



# OpenShift Container Platform 4.6

## 在 AWS 上安装

安装 OpenShift Container Platform AWS 集群



# OpenShift Container Platform 4.6 在 AWS 上安装

---

## 安装 OpenShift Container Platform AWS 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\_AWS.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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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 AWS 上安装和卸载 OpenShift Container Platform 集群的说明。

# 目录

<b>第 1 章 在 AWS 上安装</b> .....	<b>8</b>
1.1. 配置 AWS 帐户	8
1.1.1. 配置路由 53 (Route 53)	8
1.1.1.1. AWS Route 53 的 Ingress Operator 端点配置	8
1.1.2. AWS 帐户限值	9
1.1.3. 所需的 AWS 权限	10
1.1.4. 创建 IAM 用户	18
1.1.5. 支持的 AWS 区域	19
1.1.6. 后续步骤	20
1.2. 为 AWS 手动创建 IAM	20
1.2.1. 在 kube-system 项目中存储管理员级别的 secret 的替代方案	20
1.2.2. 手动创建 IAM	21
1.2.3. 管理凭证 root secret 格式	23
1.2.4. 使用手动维护的凭证升级集群	23
1.2.5. Mint 模式	23
1.2.6. 带有删除或轮转管理员凭证的 Mint 模式	24
1.2.7. 后续步骤	24
1.3. 在 AWS 上快速安装集群	24
1.3.1. 先决条件	24
1.3.2. OpenShift Container Platform 的互联网访问	25
1.3.3. 生成 SSH 私钥并将其添加到代理中	25
1.3.4. 获取安装程序	26
1.3.5. 部署集群	27
1.3.6. 通过下载二进制文件安装 OpenShift CLI	29
1.3.6.1. 在 Linux 上安装 OpenShift CLI	29
1.3.6.2. 在 Windows 上安装 OpenShift CLI	30
1.3.6.3. 在 macOS 上安装 OpenShift CLI	30
1.3.7. 使用 CLI 登录到集群	31
1.3.8. 使用 Web 控制台登录到集群	31
1.3.9. OpenShift Container Platform 的 Telemetry 访问	32
1.3.10. 后续步骤	32
1.4. 使用自定义在 AWS 上安装集群	33
1.4.1. 先决条件	33
1.4.2. OpenShift Container Platform 的互联网访问	33
1.4.3. 生成 SSH 私钥并将其添加到代理中	33
1.4.4. 获取安装程序	35
1.4.5. 创建安装配置文件	35
1.4.5.1. 安装配置参数	36
1.4.5.1.1. 所需的配置参数	37
1.4.5.1.2. 网络配置参数	38
1.4.5.1.3. 可选配置参数	39
1.4.5.1.4. 可选的 AWS 配置参数	42
1.4.5.2. AWS 的自定义 install-config.yaml 文件示例	44
1.4.5.3. 在安装过程中配置集群范围代理	46
1.4.6. 部署集群	48
1.4.7. 通过下载二进制文件安装 OpenShift CLI	49
1.4.7.1. 在 Linux 上安装 OpenShift CLI	49
1.4.7.2. 在 Windows 上安装 OpenShift CLI	50
1.4.7.3. 在 macOS 上安装 OpenShift CLI	50
1.4.8. 使用 CLI 登录到集群	51
1.4.9. 使用 Web 控制台登录到集群	51

1.4.10. OpenShift Container Platform 的 Telemetry 访问	52
1.4.11. 后续步骤	52
1.5. 使用自定义网络在 AWS 上安装集群	52
1.5.1. 先决条件	53
1.5.2. OpenShift Container Platform 的互联网访问	53
1.5.3. 生成 SSH 私钥并将其添加到代理中	53
1.5.4. 获取安装程序	54
1.5.5. 网络配置阶段	55
1.5.6. 创建安装配置文件	56
1.5.6.1. 安装配置参数	57
1.5.6.1.1. 所需的配置参数	57
1.5.6.1.2. 网络配置参数	58
1.5.6.1.3. 可选配置参数	59
1.5.6.1.4. 可选的 AWS 配置参数	63
1.5.6.2. AWS 的自定义 install-config.yaml 文件示例	64
1.5.6.3. 在安装过程中配置集群范围代理	67
1.5.7. Cluster Network Operator 配置	68
1.5.7.1. Cluster Network Operator 配置对象	68
defaultNetwork 对象配置	69
配置 OpenShift SDN CNI 集群网络供应商	70
配置 OVN-Kubernetes CNI 集群网络供应商	71
1.5.8. 指定高级网络配置	72
1.5.9. 在新 AWS 集群上配置 Ingress Controller 网络负载均衡	73
1.5.10. 使用 OVN-Kubernetes 配置混合网络	74
1.5.11. 部署集群	76
1.5.12. 通过下载二进制文件安装 OpenShift CLI	77
1.5.12.1. 在 Linux 上安装 OpenShift CLI	77
1.5.12.2. 在 Windows 上安装 OpenShift CLI	78
1.5.12.3. 在 macOS 上安装 OpenShift CLI	78
1.5.13. 使用 CLI 登录到集群	79
1.5.14. 使用 Web 控制台登录到集群	79
1.5.15. OpenShift Container Platform 的 Telemetry 访问	80
1.5.16. 后续步骤	80
1.6. 在 AWS 上将集群安装到现有的 VPC 中	80
1.6.1. 先决条件	81
1.6.2. 关于使用自定义 VPC	81
1.6.2.1. 使用 VPC 的要求	81
1.6.2.2. VPC 验证	84
1.6.2.3. 权限划分	84
1.6.2.4. 集群间隔离	84
1.6.3. OpenShift Container Platform 的互联网访问	84
1.6.4. 生成 SSH 私钥并将其添加到代理中	85
1.6.5. 获取安装程序	86
1.6.6. 创建安装配置文件	87
1.6.6.1. 安装配置参数	88
1.6.6.1.1. 所需的配置参数	88
1.6.6.1.2. 网络配置参数	89
1.6.6.1.3. 可选配置参数	90
1.6.6.1.4. 可选的 AWS 配置参数	94
1.6.6.2. AWS 的自定义 install-config.yaml 文件示例	95
1.6.6.3. 在安装过程中配置集群范围代理	98
1.6.7. 部署集群	99
1.6.8. 通过下载二进制文件安装 OpenShift CLI	100

1.6.8.1. 在 Linux 上安装 OpenShift CLI	101
1.6.8.2. 在 Windows 上安装 OpenShift CLI	101
1.6.8.3. 在 macOS 上安装 OpenShift CLI	102
1.6.9. 使用 CLI 登录到集群	102
1.6.10. 使用 Web 控制台登录到集群	103
1.6.11. OpenShift Container Platform 的 Telemetry 访问	103
1.6.12. 后续步骤	104
1.7. 在 AWS 上安装私有集群	104
1.7.1. 先决条件	104
1.7.2. 私有集群	104
1.7.2.1. AWS 中的私有集群	105
1.7.2.1.1. 限制：	105
1.7.3. 关于使用自定义 VPC	105
1.7.3.1. 使用 VPC 的要求	105
1.7.3.2. VPC 验证	108
1.7.3.3. 权限划分	108
1.7.3.4. 集群间隔离	108
1.7.4. OpenShift Container Platform 的互联网访问	108
1.7.5. 生成 SSH 私钥并将其添加到代理中	109
1.7.6. 获取安装程序	110
1.7.7. 手动创建安装配置文件	111
1.7.7.1. 安装配置参数	111
1.7.7.1.1. 所需的配置参数	112
1.7.7.1.2. 网络配置参数	113
1.7.7.1.3. 可选配置参数	114
1.7.7.1.4. 可选的 AWS 配置参数	117
1.7.7.2. AWS 的自定义 install-config.yaml 文件示例	119
1.7.7.3. 在安装过程中配置集群范围代理	121
1.7.8. 部署集群	123
1.7.9. 通过下载二进制文件安装 OpenShift CLI	124
1.7.9.1. 在 Linux 上安装 OpenShift CLI	124
1.7.9.2. 在 Windows 上安装 OpenShift CLI	125
1.7.9.3. 在 macOS 上安装 OpenShift CLI	125
1.7.10. 使用 CLI 登录到集群	125
1.7.11. 使用 Web 控制台登录到集群	126
1.7.12. OpenShift Container Platform 的 Telemetry 访问	127
1.7.13. 后续步骤	127
1.8. 在 AWS 上将集群安装到一个政府区域	127
1.8.1. 先决条件	127
1.8.2. AWS 政府区域	128
1.8.3. 私有集群	128
1.8.3.1. AWS 中的私有集群	128
1.8.3.1.1. 限制：	129
1.8.4. 关于使用自定义 VPC	129
1.8.4.1. 使用 VPC 的要求	129
1.8.4.2. VPC 验证	131
1.8.4.3. 权限划分	132
1.8.4.4. 集群间隔离	132
1.8.5. OpenShift Container Platform 的互联网访问	132
1.8.6. 生成 SSH 私钥并将其添加到代理中	133
1.8.7. 获取安装程序	134
1.8.8. 手动创建安装配置文件	135
1.8.8.1. 安装配置参数	135

1.8.8.1.1. 所需的配置参数	136
1.8.8.1.2. 网络配置参数	137
1.8.8.1.3. 可选配置参数	138
1.8.8.1.4. 可选的 AWS 配置参数	141
1.8.8.2. AWS 的自定义 install-config.yaml 文件示例	143
1.8.8.3. 没有公布的 RHCOS AMI 的 AWS 区域	146
1.8.8.4. 在 AWS 中上传自定义 RHCOS AMI	146
1.8.8.5. 在安装过程中配置集群范围代理	148
1.8.9. 部署集群	149
1.8.10. 通过下载二进制文件安装 OpenShift CLI	151
1.8.10.1. 在 Linux 上安装 OpenShift CLI	151
1.8.10.2. 在 Windows 上安装 OpenShift CLI	152
1.8.10.3. 在 macOS 上安装 OpenShift CLI	152
1.8.11. 使用 CLI 登录到集群	152
1.8.12. 使用 Web 控制台登录到集群	153
1.8.13. OpenShift Container Platform 的 Telemetry 访问	154
1.8.14. 后续步骤	154
1.9. 使用 CLOUDFORMATION 模板在 AWS 中用户置备的基础架构上安装集群	154
1.9.1. 先决条件	155
1.9.2. OpenShift Container Platform 的互联网访问	155
1.9.3. 所需的 AWS 基础架构组件	156
1.9.3.1. 集群机器	156
1.9.3.2. 其他基础架构组件	157
1.9.3.3. 证书签名请求管理	165
1.9.3.4. 所需的 AWS 权限	165
1.9.4. 获取安装程序	173
1.9.5. 生成 SSH 私钥并将其添加到代理中	173
1.9.6. 创建用于 AWS 的安装文件	175
1.9.6.1. 可选：创建独立 /var 分区	175
1.9.6.2. 创建安装配置文件	177
1.9.6.3. 在安装过程中配置集群范围代理	178
1.9.6.4. 创建 Kubernetes 清单和 Ignition 配置文件	180
1.9.7. 提取基础架构名称	182
1.9.8. 在 AWS 中创建 VPC	182
1.9.8.1. VPC 的 CloudFormation 模板	184
1.9.9. 在 AWS 中创建网络和负载均衡组件	189
1.9.9.1. 网络和负载均衡器的 CloudFormation 模板	193
1.9.10. 在 AWS 中创建安全组和角色	201
1.9.10.1. 安全对象的 CloudFormation 模板	203
1.9.11. AWS 基础架构的 RHCOS AMI	212
1.9.11.1. 没有公布的 RHCOS AMI 的 AWS 区域	213
1.9.11.2. 在 AWS 中上传自定义 RHCOS AMI	213
1.9.12. 在 AWS 中创建 bootstrap 节点	216
1.9.12.1. bootstrap 机器的 CloudFormation 模板	220
1.9.13. 在 AWS 中创建 control plane 机器	224
1.9.13.1. control plane 机器的 CloudFormation 模板	229
1.9.14. 在 AWS 中创建 worker 节点	237
1.9.14.1. worker 机器的 CloudFormation 模板	241
1.9.15. 使用用户置备的基础架构在 AWS 上初始化 bootstrap 序列	244
1.9.16. 通过下载二进制文件安装 OpenShift CLI	245
1.9.16.1. 在 Linux 上安装 OpenShift CLI	246
1.9.16.2. 在 Windows 上安装 OpenShift CLI	246
1.9.16.3. 在 macOS 上安装 OpenShift CLI	247



1.9.17. 使用 CLI 登录到集群	247
1.9.18. 批准机器的证书签名请求	248
1.9.19. 初始 Operator 配置	250
1.9.19.1. 镜像 registry 存储配置	251
1.9.19.1.1. 为使用用户置备的基础架构的 AWS 配置 registry 存储	251
1.9.19.1.2. 在非生产集群中配置镜像 registry 存储	252
1.9.20. 删除 bootstrap 资源 :	253
1.9.21. 创建 Ingress DNS 记录	253
1.9.22. 在用户置备的基础架构上完成 AWS 安装	256
1.9.23. 使用 Web 控制台登录到集群	256
1.9.24. OpenShift Container Platform 的 Telemetry 访问	257
1.9.25. 其他资源	258
1.9.26. 后续步骤	258
1.10. 在带有用户置备的受限网络中的 AWS 上安装集群	258
1.10.1. 先决条件	258
1.10.2. 关于在受限网络中安装	259
1.10.2.1. 其他限制	259
1.10.3. OpenShift Container Platform 的互联网访问	259
1.10.4. 所需的 AWS 基础架构组件	260
1.10.4.1. 集群机器	260
1.10.4.2. 其他基础架构组件	262
1.10.4.3. 证书签名请求管理	269
1.10.4.4. 所需的 AWS 权限	269
1.10.5. 生成 SSH 私钥并将其添加到代理中	277
1.10.6. 创建用于 AWS 的安装文件	278
1.10.6.1. 可选 : 创建独立 /var 分区	278
1.10.6.2. 创建安装配置文件	281
1.10.6.3. 在安装过程中配置集群范围代理	283
1.10.6.4. 创建 Kubernetes 清单和 Ignition 配置文件	284
1.10.7. 提取基础架构名称	286
1.10.8. 在 AWS 中创建 VPC	286
1.10.8.1. VPC 的 CloudFormation 模板	288
1.10.9. 在 AWS 中创建网络和负载均衡组件	294
1.10.9.1. 网络和负载均衡器的 CloudFormation 模板	297
1.10.10. 在 AWS 中创建安全组和角色	305
1.10.10.1. 安全对象的 CloudFormation 模板	307
1.10.11. AWS 基础架构的 RHCOS AMI	316
1.10.12. 在 AWS 中创建 bootstrap 节点	317
1.10.12.1. bootstrap 机器的 CloudFormation 模板	321
1.10.13. 在 AWS 中创建 control plane 机器	325
1.10.13.1. control plane 机器的 CloudFormation 模板	330
1.10.14. 在 AWS 中创建 worker 节点	338
1.10.14.1. worker 机器的 CloudFormation 模板	342
1.10.15. 使用用户置备的基础架构在 AWS 上初始化 bootstrap 序列	345
1.10.16. 使用 CLI 登录到集群	347
1.10.17. 批准机器的证书签名请求	347
1.10.18. 初始 Operator 配置	350
1.10.18.1. 禁用默认的 OperatorHub 源	351
1.10.18.2. 镜像 registry 存储配置	351
1.10.18.2.1. 为使用用户置备的基础架构的 AWS 配置 registry 存储	351
1.10.18.2.2. 在非生产集群中配置镜像 registry 存储	352
1.10.19. 删除 bootstrap 资源 :	352
1.10.20. 创建 Ingress DNS 记录	353

1.10.21. 在用户置备的基础架构上完成 AWS 安装	355
1.10.22. 使用 Web 控制台登录到集群	356
1.10.23. OpenShift Container Platform 的 Telemetry 访问	357
1.10.24. 其他资源	357
1.10.25. 后续步骤	357
1.11. 在 AWS 上卸载集群	358
1.11.1. 删除使用安装程序置备的基础架构的集群	358



# 第 1 章 在 AWS 上安装

## 1.1. 配置 AWS 帐户

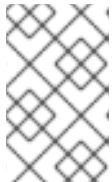
在安装 OpenShift Container Platform 之前，您必须先配置 Amazon Web Services (AWS) 帐户。

### 1.1.1. 配置路由 53 (Route 53)

要安装 OpenShift Container Platform，您使用的 Amazon Web Services (AWS) 帐户必须在 Route 53 服务中有一个专用的公共托管区。此区域必须对域具有权威。Route 53 服务为集群外部连接提供集群 DNS 解析和名称查询。

#### 流程

1. 标识您的域或子域，以及注册商 (registrar)。您可以转移现有的域和注册商，或通过 AWS 或其他来源获取新的域和注册商。



#### 注意

如果您通过 AWS 购买了一个新域，则需要一定时间来传播相关的 DNS 更改信息。有关通过 AWS 购买域的更多信息，请参阅 AWS 文档中的[使用 Amazon Route 53 注册域名](#)。

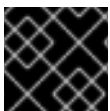
2. 如果您使用现有的域和注册商，请将其 DNS 迁移到 AWS。请参阅 AWS 文档中的[使 Amazon Route 53 成为现有域的 DNS 服务](#)。
3. 为您的域或子域创建一个公共托管区。请参阅 AWS 文档中的[创建公共托管区](#)。使用合适的根域 (如 **openshiftcorp.com**) 或子域 (如 **clusters.openshiftcorp.com**)。
4. 从托管区记录中提取新的权威名称服务器。请参阅 AWS 文档中的[获取公共托管区的名称服务器](#)。
5. 更新域所用 AWS Route 53 名称服务器的注册商记录。例如，如果您将域注册到不同帐户中的 Route 53 服务，请参阅 AWS 文档中的以下主题：[添加或更改名称服务器或粘附记录](#)。
6. 如果使用子域，请将其委托记录添加到父域中。这为子域赋予 Amazon Route 53 责任。按照父域的 DNS 供应商概述的委托程序。请参阅 [创建使用 Amazon Route 53 作为 DNS 服务的子域，而无需迁移 AWS 文档](#) 中的父域以获取示例高级流程。

#### 1.1.1.1. AWS Route 53 的 Ingress Operator 端点配置

如果您在 Amazon Web Services (AWS) GovCloud(US)US-West 或 US-East 区域中安装，Ingress Operator 使用 **us-gov-west-1** 区域用于 Route53 并标记 API 客户端。

如果配置了带有字符串 'us-gov-east-1' 的自定义端点，Ingress Operator 使用 <https://tagging.us-gov-west-1.amazonaws.com> 作为 tagging API 端点。

有关 AWS GovCloud (US) 端点的更多信息，请参阅 [AWS 文档中的有关 GovCloud\(US\)的服务端点的内容](#)。



#### 重要

在 **us-gov-east-1** 区域中安装时，AWS GovCloud 不支持私有的、断开连接的安装。

## Route 53 配置示例

```
platform:
  aws:
    region: us-gov-west-1
    serviceEndpoints:
      - name: ec2
        url: https://ec2.us-gov-west-1.amazonaws.com
      - name: elasticloadbalancing
        url: https://elasticloadbalancing.us-gov-west-1.amazonaws.com
      - name: route53
        url: https://route53.us-gov.amazonaws.com ❶
      - name: tagging
        url: https://tagging.us-gov-west-1.amazonaws.com ❷
```

❶ 对于所有两个 AWS GovCloud(US)区域，Route53 默认为 <https://route53.us-gov.amazonaws.com>。

❷ 只有 US-West 区域有标记端点。如果集群位于另一个区域，则省略此参数。

### 1.1.2. AWS 帐户限值

OpenShift Container Platform 集群使用诸多 Amazon Web Services (AWS) 组件，默认的服务限值会影响您安装 OpenShift Container Platform 集群的能力。如果您使用特定的集群配置，在某些 AWS 区域部署集群，或者从您的帐户运行多个集群，您可能需要为 AWS 帐户请求其他资源。

下表总结了 AWS 组件，它们的限值可能会影响您安装和运行 OpenShift Container Platform 集群的能力。

组件	默认可用的集群数	默认 AWS 限值	描述
实例限值	可变	可变	<p>默认情况下，每个集群创建以下实例：</p> <ul style="list-style-type: none"> <li>• 一台 Bootstrap 机器，在安装后删除</li> <li>• 三个 control plane 节点（也称为 master 节点）</li> <li>• 三个 worker 节点</li> </ul> <p>这些实例类型数量在新帐户的默认限值之内。若要部署更多 worker 节点、启用自动扩展、部署大型工作负载或使用不同的实例类型，请检查您的帐户限制，以确保集群可以部署您需要的机器。</p> <p>在大多数区域中，bootstrap 和 worker 机器使用 <b>m4.large</b> 机器，control plane 机器使用 <b>m4.xlarge</b> 实例。在一些区域，包括所有不支持这些实例类型的区域，则使用 <b>m5.large</b> 和 <b>m5.xlarge</b> 实例。</p>

组件	默认可用的集群数	默认 AWS 限值	描述
弹性 IP (EIP)	0 到 1	每个帐户 5 个 EIP	<p>要在高可用性配置中置备集群，安装程序将为<a href="#">区域中的每个可用区</a>创建一个公共和专用子网。每个专用子网都需要 <a href="#">NAT 网关</a>，每个 NAT 网关需要单独的<a href="#">弹性 IP</a>。查看<a href="#">AWS 区域图</a>来确定每个区域有多少个可用区。要利用默认高可用性，请在至少含有三个可用区的区域安装集群。要在有超过五个可用区的区域安装集群，您必须提高 EIP 限值。</p> <div style="display: flex; align-items: flex-start;">  <div> <p><b>重要</b></p> <p>要使用 <b>us-east-1</b> 区域，必须提高您帐户的 EIP 限值。</p> </div> </div>
虚拟私有云 (VPC)	5	每个区域 5 个 VPC	每个集群创建自己的 VPC。
弹性负载均衡 (ELB/NLB)	3	每个区域 20 个	在默认情况下，每个集群为 master API 服务器创建一个内部和外部网络负载均衡器，并为路由器创建一个典型的弹性负载均衡器。使用类型 <b>LoadBalancer</b> 部署更多 Kubernetes <b>Service</b> 对象将创建额外的 <a href="#">负载均衡器</a> 。
NAT 网关	5	每个可用区 5 个	集群在每个可用区中部署一个 NAT 网关。
弹性网络接口 (ENI)	至少 12 个	每个区域 350 个	<p>默认安装创建 21 个 ENI，并为区域中的每个可用区创建一个 ENI。例如，<b>us-east-1</b> 区域包含六个可用区，因此在该区域部署的集群将使用 27 个 ENI。查看<a href="#">AWS 区域图</a>来确定每个区域有多少个可用区。</p> <p>针对根据集群使用情况和部署的工作负载创建的额外机器和弹性负载均衡器，为其创建额外的 ENI。</p>
VPC 网关	20	每个帐户 20 个	每个集群创建一个 VPC 网关来访问 S3。
S3 存储桶	99	每个帐户有 100 个存储桶	因为安装过程会创建一个临时存储桶，并且每个集群中的 registry 组件会创建一个存储桶，所以您只能为每个 AWS 帐户创建 99 个 OpenShift Container Platform 集群。
安全组	250	每个帐户 2,500 个	每个集群创建 10 个不同的安全组。

### 1.1.3. 所需的 AWS 权限



## 注意

您的 IAM 用户必须在区域 **us-east-1** 中有权限 **tag:GetResources** 来删除基本集群资源。作为 AWS API 的要求的一部分，OpenShift Container Platform 安装程序在此区域中执行各种操作。

将 **AdministratorAccess** 策略附加到您在 Amazon Web Services (AWS) 中创建的 IAM 用户时，授予该用户所有需要的权限。要部署 OpenShift Container Platform 集群的所有组件，IAM 用户需要以下权限：

### 例 1.1. 安装所需的 EC2 权限

- **tag:TagResources**
- **tag:UntagResources**
- **ec2:AllocateAddress**
- **ec2:AssociateAddress**
- **ec2:AuthorizeSecurityGroupEgress**
- **ec2:AuthorizeSecurityGroupIngress**
- **ec2:CopyImage**
- **ec2>CreateNetworkInterface**
- **ec2:AttachNetworkInterface**
- **ec2:CreateSecurityGroup**
- **ec2:CreateTags**
- **ec2:CreateVolume**
- **ec2>DeleteSecurityGroup**
- **ec2>DeleteSnapshot**
- **ec2>DeleteTags**
- **ec2:DeregisterImage**
- **ec2:DescribeAccountAttributes**
- **ec2:DescribeAddresses**
- **ec2:DescribeAvailabilityZones**
- **ec2:DescribeDhcpOptions**
- **ec2:DescribeImages**
- **ec2:DescribeInstanceAttribute**
- **ec2:DescribeInstanceCreditSpecifications**

- **ec2:DescribeInstances**
- **ec2:DescribeInternetGateways**
- **ec2:DescribeKeyPairs**
- **ec2:DescribeNatGateways**
- **ec2:DescribeNetworkAcls**
- **ec2:DescribeNetworkInterfaces**
- **ec2:DescribePrefixLists**
- **ec2:DescribeRegions**
- **ec2:DescribeRouteTables**
- **ec2:DescribeSecurityGroups**
- **ec2:DescribeSubnets**
- **ec2:DescribeTags**
- **ec2:DescribeVolumes**
- **ec2:DescribeVpcAttribute**
- **ec2:DescribeVpcClassicLink**
- **ec2:DescribeVpcClassicLinkDnsSupport**
- **ec2:DescribeVpcEndpoints**
- **ec2:DescribeVpcs**
- **ec2:GetEbsDefaultKmsKeyId**
- **ec2:ModifyInstanceAttribute**
- **ec2:ModifyNetworkInterfaceAttribute**
- **ec2:ReleaseAddress**
- **ec2:RevokeSecurityGroupEgress**
- **ec2:RevokeSecurityGroupIngress**
- **ec2:RunInstances**
- **ec2:TerminateInstances**

例 1.2. 安装过程中创建网络资源所需的权限

- **ec2:AssociateDhcpOptions**
- **ec2:AssociateRouteTable**



- **ec2:AttachInternetGateway**
- **ec2:CreateDhcpOptions**
- **ec2:CreateInternetGateway**
- **ec2:CreateNatGateway**
- **ec2:CreateRoute**
- **ec2:CreateRouteTable**
- **ec2:CreateSubnet**
- **ec2:CreateVpc**
- **ec2:CreateVpcEndpoint**
- **ec2:ModifySubnetAttribute**
- **ec2:ModifyVpcAttribute**



#### 注意

如果您使用现有的 VPC，您的帐户不需要这些权限来创建网络资源。

#### 例 1.3. 安装所需的 Elastic Load Balancing 权限(ELB)

- **elasticloadbalancing:AddTags**
- **elasticloadbalancing:ApplySecurityGroupsToLoadBalancer**
- **elasticloadbalancing:AttachLoadBalancerToSubnets**
- **elasticloadbalancing:ConfigureHealthCheck**
- **elasticloadbalancing:CreateLoadBalancer**
- **elasticloadbalancing:CreateLoadBalancerListeners**
- **elasticloadbalancing>DeleteLoadBalancer**
- **elasticloadbalancing:DeregisterInstancesFromLoadBalancer**
- **elasticloadbalancing:DescribeInstanceHealth**
- **elasticloadbalancing:DescribeLoadBalancerAttributes**
- **elasticloadbalancing:DescribeLoadBalancers**
- **elasticloadbalancing:DescribeTags**
- **elasticloadbalancing:ModifyLoadBalancerAttributes**
- **elasticloadbalancing:RegisterInstancesWithLoadBalancer**

- **elasticloadbalancing:SetLoadBalancerPoliciesOfListener**

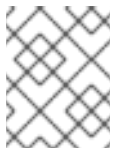
#### 例 1.4. 安装所需的 Elastic Load Balancing 权限(ELBv2)

- **elasticloadbalancing:AddTags**
- **elasticloadbalancing:CreateListener**
- **elasticloadbalancing:CreateLoadBalancer**
- **elasticloadbalancing:CreateTargetGroup**
- **elasticloadbalancing>DeleteLoadBalancer**
- **elasticloadbalancing:DeregisterTargets**
- **elasticloadbalancing:DescribeListeners**
- **elasticloadbalancing:DescribeLoadBalancerAttributes**
- **elasticloadbalancing:DescribeLoadBalancers**
- **elasticloadbalancing:DescribeTargetGroupAttributes**
- **elasticloadbalancing:DescribeTargetHealth**
- **elasticloadbalancing:ModifyLoadBalancerAttributes**
- **elasticloadbalancing:ModifyTargetGroup**
- **elasticloadbalancing:ModifyTargetGroupAttributes**
- **elasticloadbalancing:RegisterTargets**

#### 例 1.5. 安装所需的 IAM 权限

- **iam:AddRoleToInstanceProfile**
- **iam:CreateInstanceProfile**
- **iam:CreateRole**
- **iam:DeleteInstanceProfile**
- **iam>DeleteRole**
- **iam>DeleteRolePolicy**
- **iam:GetInstanceProfile**
- **iam:GetRole**
- **iam:GetRolePolicy**
- **iam:GetUser**

- **iam:ListInstanceProfilesForRole**
- **iam:ListRoles**
- **iam:ListUsers**
- **iam:PassRole**
- **iam:PutRolePolicy**
- **iam:RemoveRoleFromInstanceProfile**
- **iam:SimulatePrincipalPolicy**
- **iam:TagRole**



### 注意

如果您还没有在 AWS 帐户中创建弹性负载均衡器（ELB），IAM 用户还需要 **iam:CreateServiceLinkedRole** 权限。

#### 例 1.6. 安装所需的 Route 53 权限

- **route53:ChangeResourceRecordSets**
- **route53:ChangeTagsForResource**
- **route53:CreateHostedZone**
- **route53>DeleteHostedZone**
- **route53:GetChange**
- **route53:GetHostedZone**
- **route53:ListHostedZones**
- **route53:ListHostedZonesByName**
- **route53:ListResourceRecordSets**
- **route53:ListTagsForResource**
- **route53:UpdateHostedZoneComment**

#### 例 1.7. 安装所需的 S3 权限

- **s3:CreateBucket**
- **s3>DeleteBucket**
- **s3:GetAccelerateConfiguration**
- **s3:GetBucketAcl**

- **s3:GetBucketCors**
- **s3:GetBucketLocation**
- **s3:GetBucketLogging**
- **s3:GetBucketObjectLockConfiguration**
- **s3:GetBucketReplication**
- **s3:GetBucketRequestPayment**
- **s3:GetBucketTagging**
- **s3:GetBucketVersioning**
- **s3:GetBucketWebsite**
- **s3:GetEncryptionConfiguration**
- **s3:GetLifecycleConfiguration**
- **s3:GetReplicationConfiguration**
- **s3:ListBucket**
- **s3:PutBucketAcl**
- **s3:PutBucketTagging**
- **s3:PutEncryptionConfiguration**

#### 例 1.8. 集群 Operators 所需的 S3 权限

- **s3:DeleteObject**
- **s3:GetObject**
- **s3:GetObjectAcl**
- **s3:GetObjectTagging**
- **s3:GetObjectVersion**
- **s3:PutObject**
- **s3:PutObjectAcl**
- **s3:PutObjectTagging**

#### 例 1.9. 删除基本集群资源所需的权限

- **autoscaling:DescribeAutoScalingGroups**
- **ec2:DeleteNetworkInterface**

- **ec2:DeleteVolume**
- **elasticloadbalancing:DeleteTargetGroup**
- **elasticloadbalancing:DescribeTargetGroups**
- **iam:DeleteAccessKey**
- **iam:DeleteUser**
- **iam>ListAttachedRolePolicies**
- **iam>ListInstanceProfiles**
- **iam>ListRolePolicies**
- **iam>ListUserPolicies**
- **s3:DeleteObject**
- **s3>ListBucketVersions**
- **tag:GetResources**

#### 例 1.10. 删除网络资源所需的权限

- **ec2:DeleteDhcpOptions**
- **ec2:DeleteInternetGateway**
- **ec2:DeleteNatGateway**
- **ec2:DeleteRoute**
- **ec2:DeleteRouteTable**
- **ec2:DeleteSubnet**
- **ec2:DeleteVpc**
- **ec2:DeleteVpcEndpoints**
- **ec2:DetachInternetGateway**
- **ec2:DisassociateRouteTable**
- **ec2:ReplaceRouteTableAssociation**



#### 注意

如果您使用现有的 VPC，您的帐户不需要这些权限来删除网络资源。

#### 例 1.11. 创建清单所需的额外 IAM 和 S3 权限

- **iam:DeleteAccessKey**
- **iam:DeleteUser**
- **iam:DeleteUserPolicy**
- **iam:GetUserPolicy**
- **iam:ListAccessKeys**
- **iam:PutUserPolicy**
- **iam:TagUser**
- **iam:GetUserPolicy**
- **iam:ListAccessKeys**
- **s3:PutBucketPublicAccessBlock**
- **s3:GetBucketPublicAccessBlock**
- **s3:PutLifecycleConfiguration**
- **s3:HeadBucket**
- **s3:ListBucketMultipartUploads**
- **s3:AbortMultipartUpload**



#### 注意

如果要使用 mint 模式管理云供应商凭证，IAM 用户还需要 **iam:CreateAccessKey** and **iam:CreateUser** 权限。

#### 例 1.12. 安装时配额检查的可选权限

- **servicequotas:ListAWSDefaultServiceQuotas**

### 1.1.4. 创建 IAM 用户

每个 Amazon Web Services (AWS) 帐户都包含一个根用户帐户，它基于您用来创建帐户的电子邮件地址。这是一个高权限帐户，建议仅用于初始帐户和账单配置、创建初始用户集，以及保护帐户安全。

在安装 OpenShift Container Platform 之前，请创建一个辅助 IAM 管理用户。完成 AWS 文档中所述的在 [AWS 帐户中创建 IAM 用户](#) 流程时，请设置以下选项：

#### 流程

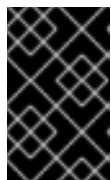
1. 指定 IAM 用户名并选择 **Programmatic access**。
2. 附加 **AdministratorAccess** 策略，以确保帐户有充足的权限来创建集群。此策略让集群能够为每个 OpenShift Container Platform 组件授予凭证。集群只为组件授予它们需要的凭证。



### 注意

虽然可以创建赋予所有所需 AWS 权限的策略并将其附加到用户，但这不是首选的选项。集群将无法为各个组件授予额外的凭证，因此所有组件都使用相同的凭证。

3. 可选：通过附加标签向用户添加元数据。
4. 确认您指定的用户名被授予了 **AdministratorAccess** 策略。
5. 记录访问密钥 ID 和 Secret 访问密钥值。在配置本地机器时，您必须使用这些值来运行安装程序。



### 重要

在部署集群时，您无法在使用多因素验证设备来验证 AWS 的同时使用您生成的临时会话令牌。在集群的整个生命周期中，集群会持续使用您的当前 AWS 凭证来创建 AWS 资源，因此您必须使用基于密钥的长期凭证。

#### 其他资源

- 有关在安装前将 Cloud Credential Operator (CCO) 设置为手动模式的步骤，请参阅[手动为 AWS 创建 IAM](#)。在无法使用云身份和访问管理 (IAM) API 的环境里，或不希望将管理员级别的凭证 secret 保存在集群 **kube-system** 项目中时，可以使用这个模式。

#### 1.1.5. 支持的 AWS 区域

您可以将 OpenShift Container Platform 集群部署到以下公共区域：



### 注意

您的 IAM 用户必须在区域 **us-east-1** 中有权限 **tag:GetResources** 来删除基本集群资源。作为 AWS API 的要求的一部分，OpenShift Container Platform 安装程序在此区域中执行各种操作。

- **af-south-1** (Cape Town)
- **ap-east-1** (Hong Kong)
- **ap-northeast-1** (Tokyo)
- **ap-northeast-2** (Seoul)
- **ap-northeast-3** (Osaka)
- **ap-south-1** (Mumbai)
- **ap-southeast-1** (Singapore)
- **ap-southeast-2** (Sydney)
- **ca-central-1** (Central)
- **eu-central-1** (Frankfurt)
- **eu-north-1** (Stockholm)

- **eu-south-1** (Milan)
- **eu-west-1** (Ireland)
- **eu-west-2** (London)
- **eu-west-3** (Paris)
- **me-south-1** (Bahrain)
- **sa-east-1** (São Paulo)
- **us-east-1** (N. Virginia)
- **us-east-2** (Ohio)
- **us-west-1** (N. California)
- **us-west-2** (Oregon)

支持以下 AWS GovCloud 区域：

- **us-gov-west-1**
- **us-gov-east-1**

### 1.1.6. 后续步骤

- 安装 OpenShift Container Platform 集群：
  - [使用安装程序置备基础架构默认选项快速安装集群](#)
  - [在安装程序置备的基础架构中使用云自定义安装集群](#)
  - [使用网络自定义在安装程序置备的基础架构上安装集群](#)
  - [使用 CloudFormation 模板在 AWS 中用户置备的基础架构上安装集群](#)

## 1.2. 为 AWS 手动创建 IAM

在无法访问云身份和访问管理 (IAM) API 的环境中，或者管理员更不希望将管理员级别的凭证 secret 存储在集群 **kube-system** 命名空间中时，可以在安装前将 Cloud Credential Operator (CCO) 放入手动模式。

### 1.2.1. 在 **kube-system** 项目中存储管理员级别的 **secret** 的替代方案

Cloud Credential Operator (CCO) 将云供应商凭证作为 Kubernetes 自定义资源定义 (CRD) 进行管理。您可以通过在 **install-config.yaml** 文件中为 **credentialsMode** 参数设置不同的值，来配置 CCO 来满足机构的安全要求。

如果您不希望在集群 **kube-system** 项目中存储管理员级别的凭证 secret，您可以在安装 OpenShift Container Platform 时选择以下选项之一：

- 手动管理云凭证：

您可以将 CCO 的 **credentialsMode** 参数设置为 **Manual** 以手动管理云凭证。使用手动模式可允许每个集群组件只拥有所需的权限，而无需在集群中存储管理员级别的凭证。如果您的环境没有



连接到云供应商公共 IAM 端点，您还可以使用此模式。但是，每次升级都必须手动将权限与新发行镜像协调。您还必须手动为每个请求它们的组件提供凭证。

- 使用 **mint** 模式安装 **OpenShift Container Platform** 后删除管理员级别的凭证 **secret**：如果您使用 CCO，并将 **credentialsMode** 参数设置为 **Mint**，您可以在安装 OpenShift Container Platform 后删除或轮转管理员级别的凭证。Mint 模式是 CCO 的默认配置。这个选项需要在安装过程中存在管理员级别的凭证。在安装过程中使用管理员级别的凭证来模拟授予某些权限的其他凭证。原始凭证 **secret** 不会永久存储在集群中。



### 注意

在非 z-stream 升级前，您必须使用管理员级别的凭证重新恢复凭证 **secret**。如果没有凭证，则可能无法进行升级。

### 其他资源

- 要了解如何在安装 OpenShift Container Platform 后轮转或删除管理员级别的凭证 **secret**，请参阅 [轮转或删除云供应商凭证](#)。
- 如需了解所有可用的 CCO 凭证模式及其支持的平台，请参阅 [Cloud Credential Operator](#)。

## 1.2.2. 手动创建 IAM

在无法访问云身份和访问管理（IAM）API 的环境中，或者管理员更不希望将管理员级别的凭证 **secret** 存储在集群 **kube-system** 命名空间中时，可以在安装前将 Cloud Credential Operator（CCO）放入手动模式。

### 流程

1. 切换到包含安装程序的目录并创建 **install-config.yaml** 文件：

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

2. 编辑 **install-config.yaml** 配置文件，把其中的 **credentialsMode** 参数设置为 **Manual**。

### 示例 **install-config.yaml** 配置文件

```
apiVersion: v1
baseDomain: cluster1.example.com
credentialsMode: Manual ❶
compute:
- architecture: amd64
  hyperthreading: Enabled
...
```

- ❶ 添加这一行将 **credentialsMode** 参数设置为 **Manual**。

3. 要生成清单，请在包含安装程序的目录中运行以下命令：

```
$ openshift-install create manifests --dir <installation_directory> ❶
```

- ❶ 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。

- 删除使用本地云凭证创建的 **admin** 凭证 `secret`。这会防止您的 **admin** 凭证存储在集群中：

```
$ rm mycluster/openshift/99_cloud-creds-secret.yaml
```

- 从包含安装程序的目录中，获取 **openshift-install** 二进制文件要使用的 OpenShift Container Platform 发行镜像详情：

```
$ openshift-install version
```

### 输出示例

```
release image quay.io/openshift-release-dev/ocp-release:4.y.z-x86_64
```

- 针对您要部署到的云，找到此发行版本镜像中的所有 **CredentialsRequests** 对象：

```
$ oc adm release extract quay.io/openshift-release-dev/ocp-release:4.y.z-x86_64 --
credentials-requests --cloud=aws
```

这会显示每个请求的详情。

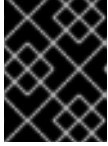
### CredentialsRequest 对象示例

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: cloud-credential-operator-iam-ro
  namespace: openshift-cloud-credential-operator
spec:
  secretRef:
    name: cloud-credential-operator-iam-ro-creds
    namespace: openshift-cloud-credential-operator
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
    - effect: Allow
      action:
      - iam:GetUser
      - iam:GetUserPolicy
      - iam:ListAccessKeys
    resource: "*"

```

- 在之前生成的 **openshift-install** 清单目录中为 `secret` 创建 YAML 文件。`secret` 必须使用在 **spec.secretRef** 中为每个 **credentialsRequest** 定义的命名空间和 `secret` 名称存储。`secret` 数据的格式因云供应商而异。
- 从包含安装程序的目录中，开始创建集群：

```
$ openshift-install create cluster --dir <installation_directory>
```



## 重要

在升级使用手动维护凭证的集群前，必须确保 CCO 处于可升级状态。详情请参阅您的云供应商的*对手动维护凭证的集群进行升级*部分的内容。

### 1.2.3. 管理凭证 root secret 格式

每个云供应商都使用 **kube-system** 命名空间中的一个凭证 root secret，用于满足所有凭证请求并创建它们相应的 secret。这可以通过 mint 新凭证 (*mint mode*)，或复制凭证 root secret (*passthrough mode*) 实现。

secret 的格式因云而异，也用于每个 **CredentialsRequest** secret。

#### Amazon Web Services(AWS)secret 格式

```
apiVersion: v1
kind: Secret
metadata:
  namespace: kube-system
  name: aws-creds
stringData:
  aws_access_key_id: <AccessKeyID>
  aws_secret_access_key: <SecretAccessKey>
```

### 1.2.4. 使用手动维护的凭证升级集群

如果在未来的发行版本中添加了凭证，则使用手动维护凭证的集群的 Cloud Credential Operator (CCO) 可升级状态会变为 **false**。对于次版本（例如从 4.5 到 4.6），这个状态会阻止升级，直到解决了更新的权限。对于 z-stream 版本（例如从 4.5.10 到 4.5.11），升级不会受阻，但必须为新版本更新凭证。

使用 Web 控制台的 **Administrator** 视角来判断 CCO 是否可以升级。

1. 导航至 **Administration** → **Cluster Settings**。
2. 要查看 CCO 状态详情，请点 **Cluster Operators** 列表中的 **cloud-credential**。
3. 如果 **Conditions** 部分中的 **Upgradeable** 状态为 **False**，请检查新发行版本的 **credentialsRequests**，并在升级前更新集群中手动维护的凭证以匹配。

除了为您要升级到的发行版本镜像创建新凭证外，还需要查看现有凭证所需的权限，并满足新发行版本中现有组件的所有新权限要求。CCO 无法检测到这些不匹配的问题，且在此情况下无法将 **upgradable** 设置为 **false**。

详情请参阅您的云供应商的*手动创建 IAM* 部分来了解如何获取和使用您的云所需的凭证。

### 1.2.5. Mint 模式

Mint 模式是 OpenShift Container Platform 的默认和推荐的 Cloud Credential Operator (CCO) 凭证模式。在这种模式中，CCO 使用提供的管理员级云凭证来运行集群。AWS、GCP 和 Azure 支持 Mint 模式。

在 mint 模式中，**admin** 凭证存储在 **kube-system** 命名空间中，然后由 CCO 使用来处理集群中的 **CredentialsRequest** 对象，并为每个对象创建具有特定权限的用户。

mint 模式的好处包括：

- 每个集群组件只有其所需权限
- 云凭证的自动、持续协调，包括升级可能需要的额外凭证或权限

mint 模式的一个缺陷是，**admin** 凭证需要存储在集群 **kube-system** 的 secret 中。

### 1.2.6. 带有删除或轮转管理员凭证的 Mint 模式

目前，只有 AWS 支持这个模式。

在这个模式中，用户使用类似正常的 mint 模式的 **admin** 凭证安装 OpenShift Container Platform。但是，此模式会在集群安装后删除 **admin** 凭证 secret。

管理员可以让 Cloud Credential Operator 自行请求只读凭证，许它验证所有 **CredentialsRequest** 对象是否有其所需的权限。因此，除非需要更改内容，否则不需要 **admin** 凭证。删除关联的凭证后，可以根据需要在底层云上销毁它。

在升级前，应该恢复 **admin** 凭证。以后，如果凭证不存在，升级可能会阻止。

**admin** 凭证不会永久存储在集群中。

这个模式仍然需要在一个短的时间内，集群中存在 **admin** 凭证。它还需要为每个升级使用 **admin** 凭证手动重新生成 secret。

### 1.2.7. 后续步骤

- 安装 OpenShift Container Platform 集群：
  - 使用安装程序置备的基础架构默认选项在 [AWS 上快速安装集群](#)
  - 在安装程序置备的基础架构中使用云自定义安装集群
  - 使用网络自定义在安装程序置备的基础架构上安装集群
  - 使用 CloudFormation 模板在 AWS 中用户置备的基础架构上安装集群

## 1.3. 在 AWS 上快速安装集群

在 OpenShift Container Platform 版本 4.6 中，您可以使用默认配置选项在 Amazon Web Services (AWS) 上安装集群。

### 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。
- [配置 AWS 帐户](#) 以托管集群。



### 重要

如果您的计算机上存储有 AWS 配置集，则不要在使用多因素验证设备的同时使用您生成的临时会话令牌。在集群的整个生命周期中，集群会持续使用您的当前 AWS 凭证来创建 AWS 资源，因此您必须使用基于密钥的长期凭证。要生成适当的密钥，请参阅 AWS 文档中的[管理 IAM 用户的访问密钥](#)。您可在运行安装程序时提供密钥。

- 如果使用防火墙，则必须[将其配置为允许集群需要访问的站点](#)。
- 如果不允许系统管理身份和访问管理（IAM），集群管理员可以[手动创建和维护 IAM 凭证](#)。手动模式也可以用于云 IAM API 无法访问的环境中。

### 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. 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。

## 1.3.4. 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

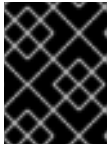
## 流程

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。

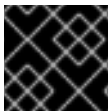
4. 提取安装程序。例如，在使用 Linux 操作系统的计算机上运行以下命令：

```
$ 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。

## 1.3.5. 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

- 配置托管集群的云平台的帐户。
- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

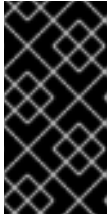
## 流程

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> \ ❶  
--log-level=info ❷
```

❶ 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。

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



## 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。

在提示符处提供值：

- a. 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。

- b. 选择 **aws** 作为目标平台。
- c. 如果计算机上没有保存 Amazon Web Services (AWS) 配置集，请为您配置用于运行安装程序的用户输入 AWS 访问密钥 ID 和 Secret 访问密钥。



## 注意

AWS 访问密钥 ID 和 secret 访问密钥存储在安装主机上当前用户主目录中的 **~/.aws/credentials** 中。如果文件中不存在导出的配置集凭证，安装程序会提示您输入凭证。您向安装程序提供的所有凭证都存储在文件中。

- d. 选择要将集群部署到的 AWS 区域。
- e. 选择您为集群配置的 Route 53 服务的基域。
- f. 为集群输入一个描述性名称。
- g. 粘贴 [Red Hat OpenShift Cluster Manager](#) 中的 **pull secret**。



## 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。

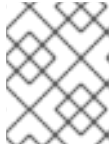
## 输出示例

```
...
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
```



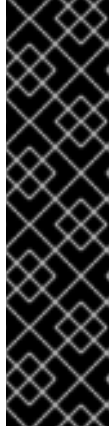
INFO Login to the console with user: "kubeadmin", and password: "4vYBz-Ee6gm-ymBZj-Wt5AL"

INFO Time elapsed: 36m22s



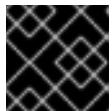
### 注意

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 Ignition 配置文件生成后的 12 小时内使用它们，因为 24 小时的证书会在集群安装后的 16 小时到 22 小时进行轮转。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因为执行了证书更新而导致安装失败的问题。



### 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。

2. 可选：从您用来安装集群的 IAM 帐户删除或禁用 **AdministratorAccess** 策略。



### 注意

只有在安装过程中才需要 **AdministratorAccess** 策略提供的升级权限。

## 其他资源

- 如需有关 AWS 配置集和凭证配置的更多信息，请参阅 [AWS 文档中的配置和凭证文件设置](#)。

## 1.3.6. 通过下载二进制文件安装 OpenShift CLI

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



### 重要

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

### 1.3.6.1. 在 Linux 上安装 OpenShift CLI

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

#### 流程

1. 进入到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。

2. 在 **Version** 下拉菜单中选择相应的版本。
3. 单击 **OpenShift v4.6 Linux** 客户端条目旁边的 **Download Now**，再保存文件。
4. 解包存档：

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

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

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 1.3.6.2. 在 Windows 上安装 OpenShift CLI

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

流程

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

```
C:\> path
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 1.3.6.3. 在 macOS 上安装 OpenShift CLI

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

流程

1. 进入到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 在 **Version** 下拉菜单中选择相应的版本。
3. 单击 **OpenShift v4.6 MacOSX** 客户端条目旁边的 **Download Now**，再保存文件。
4. 解包和解压存档。

- 将 **oc** 二进制文件移到 PATH 的目录中。  
要查看您的 **PATH**，打开一个终端窗口并执行以下命令：

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 1.3.7. 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

- 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。

- 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
```

### 1.3.8. 使用 Web 控制台登录到集群

**kubeadmin** 用户默认在 OpenShift Container Platform 安装后存在。您可以使用 OpenShift Container Platform Web 控制台以 **kubeadmin** 用户身份登录集群。

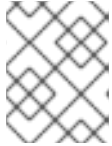
先决条件

- 有访问安装主机的访问权限。
- 您完成了集群安装，所有集群 Operator 都可用。

流程

- 从安装主机上的 **kubeadmin -password** 文件中获取 kubeadmin 用户的密码：

```
$ cat <installation_directory>/auth/kubeadmin-password
```



### 注意

另外，您还可以从安装主机上的 `<installation_directory>/openshift_install.log` 日志文件获取 **kubeadmin** 密码。

2. 列出 OpenShift Container Platform Web 控制台路由：

```
$ oc get routes -n openshift-console | grep 'console-openshift'
```



### 注意

另外，您还可以从安装主机上的 `<installation_directory>/openshift_install.log` 日志文件获取 OpenShift Container Platform 路由。

### 输出示例

```
console    console-openshift-console.apps.<cluster_name>.<base_domain>    console
https reencrypt/Redirect None
```

3. 在 Web 浏览器中导航到上一命令输出中包括的路由，以 **kubeadmin** 用户身份登录。

### 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](#)。

## 1.3.9. 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](#)。

## 1.3.10. 后续步骤

- [验证安装](#)。
- [自定义集群](#)。
- 如果需要，您可以[选择不使用远程健康报告](#)。
- 如果需要，您可以[删除云供应商凭证](#)。

## 1.4. 使用自定义在 AWS 上安装集群

在 OpenShift Container Platform 版本 4.6 中，您可以在安装程序在 Amazon Web Services (AWS) 中置备的基础架构上安装自定义集群。要自定义安装，请在安装集群前修改 `install-config.yaml` 文件中的参数。

### 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。
- [配置 AWS 帐户](#) 以托管集群。



#### 重要

如果您的计算机上存储有 AWS 配置集，则不要在使用多因素验证设备的同时使用您生成的临时会话令牌。在集群的整个生命周期中，集群会持续使用您的当前 AWS 凭证来创建 AWS 资源，因此您必须使用长期凭证。要生成适当的密钥，请参阅 AWS 文档中的[管理 IAM 用户的访问密钥](#)。您可在运行安装程序时提供密钥。

- 如果使用防火墙，则必须将其配置为允许集群需要访问的站点。
- 如果不允许系统管理身份和访问管理 (IAM)，集群管理员可以[手动创建和维护 IAM 凭证](#)。手动模式也可以用于云 IAM API 无法访问的环境中。

### 1.4.2. 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全。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。

### 1.4.3. 生成 SSH 私钥并将其添加到代理中

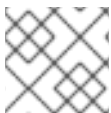
如果要在集群上执行安装调试或灾难恢复，则必须为 `ssh-agent` 和安装程序提供 SSH 密钥。您可以使用此密钥访问公共集群中的 bootstrap 机器来排除安装问题。



#### 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`

トピック

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

#### 1.4.4. 获取安装程序

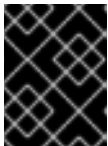
在安装 OpenShift Container Platform 之前，将安装文件下载到本地计算机上。

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。

#### 1.4.5. 创建安装配置文件

您可以自定义在 Amazon Web Services (AWS) 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 **AWS** 作为目标平台。
  - iii. 如果计算机上没有保存 Amazon Web Services (AWS) 配置集，请为您配置用于运行安装程序的用户输入 AWS 访问密钥 ID 和 Secret 访问密钥。
  - iv. 选择要将集群部署到的 AWS 区域。
  - v. 选择您为集群配置的 Route 53 服务的基域。
  - vi. 为集群输入一个描述性名称。
  - vii. 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。

#### 1.4.5.1. 安装配置参数

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



### 注意

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





## 重要

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

### 1.4.5.1.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 1.1. 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 <b>v1</b> 。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>&lt;metadata.name&gt;.&lt;baseDomain&gt;</code> 。	完全限定域名或子域名，如 <b>example.com</b> 。
<code>metadata</code>	Kubernetes 资源 <code>ObjectMeta</code> ，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 <b>dev</b> 。
<code>platform</code>	执行安装的具体平台配置： <b>aws</b> 、 <b>baremetal</b> 、 <b>azure</b> 、 <b>openstack</b> 、 <b>ovirt</b> 、 <b>vsphere</b> 。有关 <code>platform</code> . <code>&lt;platform&gt;</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
<b>pullSecret</b>	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>

#### 1.4.5.1.2. 网络配置参数

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

只支持 IPv4 地址。

表 1.2. 网络参数

参数	描述	值
<b>networking</b>	集群网络的配置。	对象  <b>注意</b> 您不能在安装后修改 <b>networking</b> 对象指定的参数。
<b>networking.networkType</b>	要安装的集群网络供应商 Container Network Interface (CNI) 插件。	<b>OpenShiftSDN</b> 或 <b>OVNKubernetes</b> 。默认值为 <b>OpenShiftSDN</b> 。
<b>networking.clusterNetwork</b>	pod 的 IP 地址块。 默认值为 <b>10.128.0.0/14</b> ，主机前缀为 <b>/23</b> 。 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>
<b>networking.clusterNetwork.cidr</b>	使用 <b>networking.clusterNetwork</b> 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 <b>0</b> 到 <b>32</b> 之间。

参数	描述	值
<b>networking.clusterNetwork.hostPrefix</b>	分配给每个单独节点的子网前缀长度。例如，如果 <b>hostPrefix</b> 设为 <b>23</b> ，则每个节点从所给的 <b>cidr</b> 中分配一个 <b>/23</b> 子网。 <b>hostPrefix</b> 值 <b>23</b> 提供 510 ( $2^{(32-23)}-2$ ) 个 pod IP 地址。	子网前缀。 默认值为 <b>23</b> 。
<b>networking.serviceNetwork</b>	服务的 IP 地址块。默认值为 <b>172.30.0.0/16</b> 。  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>
<b>networking.machineNetwork</b>	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>
<b>networking.machineNetwork.cidr</b>	使用 <b>networking.machineNetwork</b> 时需要。IP 地址块。libvirt 以外的所有平台的默认值为 <b>10.0.0.0/16</b> 。对于 libvirt，默认值为 <b>192.168.126.0/24</b> 。	CIDR 表示法中的 IP 网络块。 例如： <b>10.0.0.0/16</b> 。   <b>注意</b> 将 <b>networking.machineNetwork</b> 设置为与首选 NIC 所在的 CIDR 匹配。

#### 1.4.5.1.3. 可选配置参数

下表描述了可选安装配置参数：

表 1.3. 可选参数

参数	描述	值
<b>additionalTrustBundle</b>	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
<b>compute</b>	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
<b>compute.architecture</b>	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 <b>amd64</b> （默认值）。	字符串
<b>compute.hyperthreading</b>	<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><b>重要</b></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>	<b>Enabled</b> 或 <b>Disabled</b>
<b>compute.name</b>	使用 <b>compute</b> 时需要此值。机器池的名称。	<b>worker</b>
<b>compute.platform</b>	使用 <b>compute</b> 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 <b>controlPlane.platform</b> 参数值匹配。	<b>aws</b> 、 <b>azure</b> 、 <b>gcp</b> 、 <b>openstack</b> 、 <b>ovirt</b> 、 <b>vsphere</b> 或 <b>{}</b>
<b>compute.replicas</b>	要置备的计算机器数量，也称为 worker 机器。	大于或等于 <b>2</b> 的正整数。默认值为 <b>3</b> 。
<b>controlPlane</b>	组成 control plane 的机器的配置。	<b>MachinePool</b> 对象的数组。详情请查看以下"Machine-pool"表。
<b>controlPlane.architecture</b>	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 <b>amd64</b> （默认值）。	字符串
<b>controlPlane.hyperthreading</b>	<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><b>重要</b></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>	<b>Enabled</b> 或 <b>Disabled</b>

参数	描述	值
<b>controlPlane.name</b>	使用 <b>controlPlane</b> 时需要。机器池的名称。	<b>master</b>
<b>controlPlane.platform</b>	使用 <b>controlPlane</b> 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 <b>compute.platform</b> 参数值匹配。	<b>aws、azure、gcp、openstack、o virt、vsphere 或 {}</b>
<b>controlPlane.replicas</b>	要置备的 control plane 机器数量。	唯一支持的值是 <b>3</b> ，它是默认值。
<b>credentialsMode</b>	<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; border: 1px solid black; margin-right: 10px;"></div> <div> <p><b>注意</b></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>	<b>Mint、Passthrough、Manual</b> 或空字符串(“”)。
<b>fips</b>	<p>启用或禁用 FIPS 模式。默认为 <b>false</b> (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; border: 1px solid black; margin-right: 10px;"></div> <div> <p><b>重要</b></p> <p>只有在 <b>x86_64</b> 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: 20px; height: 20px; border: 1px solid black; margin-right: 10px;"></div> <div> <p><b>注意</b></p> <p>如果使用 Azure File 存储，则无法启用 FIPS 模式。</p> </div> </div>	<b>false</b> 或 <b>true</b>

参数	描述	值
<b>imageContentSources</b>	release-image 内容的源和仓库。	对象数组。包括一个 <b>source</b> 以及可选的 <b>mirrors</b> ，如下表所示。
<b>imageContentSources.source</b>	使用 <b>imageContentSources</b> 时需要。指定用户在镜像拉取规格中引用的仓库。	字符串
<b>imageContentSources.mirrors</b>	指定可能还包含同一镜像的一个或多个仓库。	字符串数组
<b>publish</b>	如何发布或公开集群的面向用户的端点，如 Kubernetes API、OpenShift 路由。	<b>Internal</b> 或 <b>External</b> 。把 <b>publish</b> 设置为 <b>Internal</b> 以部署一个私有集群，它不能被互联网访问。默认值为 <b>External</b> 。
<b>sshKey</b>	<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: center;">  <div> <p><b>注意</b></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 <b>ssh-agent</b> 进程使用的 SSH 密钥。</p> </div> </div>	<p>一个或多个密钥。例如：</p> <pre>sshKey:   &lt;key1&gt;   &lt;key2&gt;   &lt;key3&gt;</pre>

#### 1.4.5.1.4. 可选的 AWS 配置参数

下表描述了可选的 AWS 配置参数：

表 1.4. 可选的 AWS 参数

参数	描述	值
<b>compute.platform.aws.amIID</b>	用于为集群引导计算机器的 AWS AMI。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<b>compute.platform.aws.rootVolume.iops</b>	为根卷保留的每秒输入/输出操作 (IOPS) 数。	整数，如 <b>4000</b> 。

参数	描述	值
<code>compute.platform.aws.rootVolume.size</code>	以 GiB 为单位的根卷大小。	整数，如 <b>500</b> 。
<code>compute.platform.aws.rootVolume.type</code>	根卷的类型。	有效的 <a href="#">AWS EBS 卷类型</a> ，如 <b>io1</b> 。
<code>compute.platform.aws.rootVolume.kmsKeyARN</code>	KMS 密钥的 Amazon 资源名称（密钥 ARN）。这是使用特定 KMS 密钥加密 worker 节点的操作系统卷。	有效的 <a href="#">密钥 ID 或密钥 ARN</a> 。
<code>compute.platform.aws.type</code>	计算机器的 EC2 实例类型。	有效的 <a href="#">AWS 实例类型</a> ，如 <b>c5.9xlarge</b> 。
<code>compute.platform.aws.zones</code>	安装程序在其中为计算机器池创建机器的可用区。如果您提供自己的 VPC，则必须在那个可用域中提供一个子网。	有效 AWS 可用区的列表，如 <b>us-east-1c</b> ，以 <a href="#">YAML 序列</a> 表示。
<code>compute.aws.region</code>	安装程序在其中创建计算资源的 AWS 区域。	任何有效的 <a href="#">AWS 区域</a> ，如 <b>us-east-1</b> 。
<code>controlPlane.platform.aws.amiID</code>	用于为集群引导 control plane 机器的 AWS AMI。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<code>controlPlane.platform.aws.rootVolume.kmsKeyARN</code>	KMS 密钥的 Amazon 资源名称（密钥 ARN）。这需要使用特定的 KMS 密钥加密 control plane 节点的操作系统卷。	有效的 <a href="#">密钥 ID 和密钥 ARN</a> 。
<code>controlPlane.platform.aws.type</code>	control plane 机器的 EC2 实例类型。	有效的 <a href="#">AWS 实例类型</a> ，如 <b>c5.9xlarge</b> 。
<code>controlPlane.platform.aws.zones</code>	安装程序在其中为 control plane 机器池创建机器的可用区。	有效 AWS 可用区的列表，如 <b>us-east-1c</b> ，以 <a href="#">YAML 序列</a> 表示。
<code>controlPlane.aws.region</code>	安装程序在其中创建 control plane 资源的 AWS 区域。	有效的 <a href="#">AWS 区域</a> ，如 <b>us-east-1</b> 。

参数	描述	值
<b>platform.aws.amiID</b>	用于为集群引导所有机器的 AWS AMI。如果设置，AMI 必须属于与集群相同的区域。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<b>platform.aws.serviceEndpoints.name</b>	AWS 服务端点名称。只有在必须使用替代 AWS 端点（如 FIPS）时，才需要自定义端点。可以为 EC2、S3、IAM、Elastic Load Balancing、Tagging、Route 53 和 STS AWS 服务指定自定义 API 端点。	有效的 <a href="#">AWS 服务端点名称</a> 。
<b>platform.aws.serviceEndpoints.url</b>	AWS 服务端点 URL。URL 必须使用 <b>https</b> 协议，主机必须信任该证书。	有效的 <a href="#">AWS 服务端点 URL</a> 。
<b>platform.aws.userTags</b>	键与值的映射，安装程序将其作为标签添加到它所创建的所有资源。	任何有效的 YAML 映射，如 <b>&lt;key&gt;: &lt;value&gt;</b> 格式的键值对。如需有关 AWS 标签的更多信息，请参阅 AWS 文档中的 <a href="#">标记您的 Amazon EC2 资源</a> 。
<b>platform.aws.subnets</b>	如果您提供 VPC，而不是让安装程序为您创建 VPC，请指定要使用的集群子网。子网必须是您指定的同一 <b>machineNetwork[].cidr</b> 范围的一部分。对于标准集群，为每个可用区指定一个公共和私有子网。对于私有集群，为每个可用区指定一个私有子网。	有效的子网 ID。

### 1.4.5.2. AWS 的自定义 install-config.yaml 文件示例

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



#### 重要

此示例 YAML 文件仅供参考。您必须使用安装程序来获取 **install-config.yaml** 文件，并且修改该文件。

```
apiVersion: v1
baseDomain: example.com 1
credentialsMode: Mint 2
controlPlane: 3 4
hypertexting: Enabled 5
```



```

name: master
platform:
  aws:
    zones:
      - us-west-2a
      - us-west-2b
    rootVolume:
      iops: 4000
      size: 500
      type: io1 6
    type: m5.xlarge
  replicas: 3
compute: 7
- hyperthreading: Enabled 8
name: worker
platform:
  aws:
    rootVolume:
      iops: 2000
      size: 500
      type: io1 9
    type: c5.4xlarge
    zones:
      - us-west-2c
  replicas: 3
metadata:
  name: test-cluster 10
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  aws:
    region: us-west-2 11
  userTags:
    adminContact: jdoe
    costCenter: 7536
  amiID: ami-96c6f8f7 12
  serviceEndpoints: 13
    - name: ec2
      url: https://vpce-id.ec2.us-west-2.vpce.amazonaws.com
  fips: false 14
  sshKey: ssh-ed25519 AAAA... 15
  pullSecret: '{"auths": ...}' 16

```

**1** **10** **11** **16** 必需。安装程序会提示您输入这个值。

**2** 可选：添加此参数来强制 Cloud Credential Operator (CCO) 使用指定的模式，而不是让 CCO 动态尝试决定凭证的功能。如需有关 CCO 模式的详情，请参阅 *Red Hat Operator* 参考内容中的 *Cloud Credential Operator* 条目。

- 3 7 如果没有提供这些参数和值，安装程序会提供默认值。
- 4 **controlPlane** 部分是一个单个映射，但 **compute** 部分是一系列映射。为满足不同数据结构的要求，**compute** 部分的第一行必须以连字符 - 开头，**controlPlane** 部分的第一行则不以连字符开头。只使用一个 control plane 池。
- 5 8 是否要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。您可以通过将参数值设为 **Disabled** 来禁用。如果您在某些集群机器上禁用并发多线程，则必须在所有集群机器上禁用。



### 重要

如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。如果您对机器禁用并发多线程，请使用较大的实例类型，如 **m4.2xlarge** 或 **m5.2xlarge**。

- 6 9 要为 etcd 配置更快的存储，特别是对于较大的集群，请将存储类型设置为 **io1**，并将 **iops** 设为 **2000**。
- 12 用于为集群引导机器的 AMI ID。如果设置，AMI 必须属于与集群相同的区域。
- 13 AWS 服务端点。在安装到未知 AWS 区域时，需要自定义端点。端点 URL 必须使用 **https** 协议，主机必须信任该证书。
- 14 是否启用或禁用 FIPS 模式。默认情况下不启用 FIPS 模式。如果启用了 FIPS 模式，运行 OpenShift Container Platform 的 Red Hat Enterprise Linux CoreOS (RHCOS) 机器会绕过默认的 Kubernetes 加密套件，并使用由 RHCOS 提供的加密模块。



### 重要

只有在 **x86\_64** 架构中的 OpenShift Container Platform 部署支持 FIPS 验证的 `/Modules in Process` 加密库。

- 15 您可以选择提供您用来访问集群中机器的 **sshKey** 值。



### 注意

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

#### 1.4.5.3. 在安装过程中配置集群范围代理

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

##### 先决条件

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



## 注意

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

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

- 如果您的集群位于 AWS 上, 请将 `ec2.<region>.amazonaws.com`、`elasticloadbalancing.<region>.amazonaws.com` 和 `s3.<region>.amazonaws.com` 端点添加到 VPC 端点。需要这些端点才能完成节点到 AWS EC2 API 的请求。由于代理在容器级别而不是节点级别工作, 因此您必须通过 AWS 专用网络将这些请求路由到 AWS EC2 API。在代理服务器中的允许列表中添加 EC2 API 的公共 IP 地址是不够的。

## 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` 的配置映射来保存额外的 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, 将为 `trustedCA` 参数指定的内容与 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**。

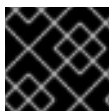


### 注意

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

## 1.4.6. 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

- 配置托管集群的云平台的帐户。
- 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。

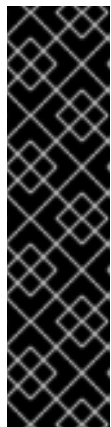
### 输出示例

```
...
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 Ignition 配置文件生成后的 12 小时内使用它们，因为 24 小时的证书会在集群安装后的 16 小时到 22 小时进行轮转。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因为执行了证书更新而导致安装失败的问题。



### 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。

2. 可选：从您用来安装集群的 IAM 帐户删除或禁用 **AdministratorAccess** 策略。



### 注意

只有在安装过程中才需要 **AdministratorAccess** 策略提供的升级权限。

## 1.4.7. 通过下载二进制文件安装 OpenShift CLI

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



### 重要

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

### 1.4.7.1. 在 Linux 上安装 OpenShift CLI

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

#### 流程

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

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

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

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

#### 1.4.7.2. 在 Windows 上安装 OpenShift CLI

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

流程

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

```
C:\> path
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

#### 1.4.7.3. 在 macOS 上安装 OpenShift CLI

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

流程

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

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

-

```
$ oc <command>
```

### 1.4.8. 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
```

### 1.4.9. 使用 Web 控制台登录到集群

**kubeadmin** 用户默认在 OpenShift Container Platform 安装后存在。您可以使用 OpenShift Container Platform Web 控制台以 **kubeadmin** 用户身份登录集群。

先决条件

- 有访问安装主机的访问权限。
- 您完成了集群安装，所有集群 Operator 都可用。

流程

1. 从安装主机上的 **kubeadmin -password** 文件中获取 kubeadmin 用户的密码：

```
$ cat <installation_directory>/auth/kubeadmin-password
```



#### 注意

另外，您还可以从安装主机上的 **<installation\_directory>/openshift\_install.log** 日志文件获取 **kubeadmin** 密码。

- 列出 OpenShift Container Platform Web 控制台路由：

```
$ oc get routes -n openshift-console | grep 'console-openshift'
```



### 注意

另外，您还可以从安装主机上的 `<installation_directory>/openshift_install.log` 日志文件获取 OpenShift Container Platform 路由。

### 输出示例

```
console    console-openshift-console.apps.<cluster_name>.<base_domain>    console
https    reencrypt/Redirect    None
```

- 在 Web 浏览器中导航到上一命令输出中包括的路由，以 **kubeadmin** 用户身份登录。

### 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](#)。

## 1.4.10. 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](#)。

## 1.4.11. 后续步骤

- [验证安装](#)。
- [自定义集群](#)。
- 如果需要，您可以[选择不使用远程健康报告](#)。
- 如果需要，您可以[删除云供应商凭证](#)。

## 1.5. 使用自定义网络在 AWS 上安装集群

在 OpenShift Container Platform 版本 4.6 中，您可以使用自定义网络配置选项在 Amazon Web Services (AWS) 上安装集群。通过自定义网络配置，您的集群可以与环境中现有的 IP 地址分配共存，并与现有的 MTU 和 VXLAN 配置集成。

大部分网络配置参数必须在安装过程中设置，只有 **kubeProxy** 配置参数可以在运行的集群中修改。



### 1.5.1. 先决条件

- 查看有关 [OpenShift Container Platform 安装和更新](#) 流程的详细信息。
- [配置 AWS 帐户](#) 以托管集群。



#### 重要

如果您的计算机上存储有 AWS 配置集，则不要在使用多因素验证设备的同时使用您生成的临时会话令牌。在集群的整个生命周期中，集群会持续使用您的当前 AWS 凭证来创建 AWS 资源，因此您必须使用基于密钥的长期凭证。要生成适当的密钥，请参阅 AWS 文档中的[管理 IAM 用户的访问密钥](#)。您可在运行安装程序时提供密钥。

- 如果使用防火墙，则必须[将其配置为允许集群需要访问的站点](#)。
- 如果不允许系统管理身份和访问管理（IAM），集群管理员可以[手动创建和维护 IAM 凭证](#)。手动模式也可以用于云 IAM API 无法访问的环境中。

### 1.5.2. 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。

### 1.5.3. 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](#)。

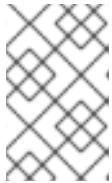
## 流程

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。

### 1.5.4. 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。

## 1.5.5. 网络配置阶段

当在安装前指定集群配置时，在安装过程中的几个阶段可以修改网络配置：

### 阶段 1

输入 `openshift-install create install-config` 命令后。在 `install-config.yaml` 文件中，您可以自定义以下与网络相关的字段：

- `networking.networkType`
- `networking.clusterNetwork`
- `networking.serviceNetwork`
- `networking.machineNetwork`  
有关这些字段的更多信息，请参阅“安装配置参数”。



### 注意

将 `networking.machineNetwork` 设置为与首选 NIC 所在的 CIDR 匹配。

## 阶段 2

输入 `openshift-install create manifests` 命令后。如果必须指定高级网络配置，在这个阶段中，只能使用您要修改的字段来定义自定义的 Cluster Network Operator 清单。

在 2 阶段，您无法覆盖 `install-config.yaml` 文件中的 1 阶段中指定的值。但是，您可以在第 2 阶段进一步自定义集群网络供应商。

### 1.5.6. 创建安装配置文件

您可以自定义在 Amazon Web Services (AWS) 上安装的 OpenShift Container Platform 集群。

#### 先决条件

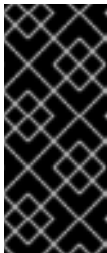
- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

#### 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 **AWS** 作为目标平台。
- iii. 如果计算机上没有保存 Amazon Web Services (AWS) 配置集，请为您配置用于运行安装程序的用户输入 AWS 访问密钥 ID 和 Secret 访问密钥。
- iv. 选择要将集群部署到的 AWS 区域。
- v. 选择您为集群配置的 Route 53 服务的基域。
- vi. 为集群输入一个描述性名称。

- vii. 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。

#### 1.5.6.1. 安装配置参数

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



### 注意

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



### 重要

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

##### 1.5.6.1.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 1.5. 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 <b>v1</b> 。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>&lt;metadata.name&gt;.&lt;baseDomain&gt;</code> 。	完全限定域名或子域名，如 <code>example.com</code> 。
<code>metadata</code>	Kubernetes 资源 <code>ObjectMeta</code> ，其中只消耗 <code>name</code> 参数。	对象

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

### 1.5.6.1.2. 网络配置参数

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

只支持 IPv4 地址。

表 1.6. 网络参数

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

参数	描述	值
<b>networking.clusterNetwork</b>	pod 的 IP 地址块。  默认值为 <b>10.128.0.0/14</b> ，主机前缀为 <b>/23</b> 。  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>
<b>networking.clusterNetwork.cidr</b>	使用 <b>networking.clusterNetwork</b> 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 <b>0</b> 到 <b>32</b> 之间。
<b>networking.clusterNetwork.hostPrefix</b>	分配给每个单独节点的子网前缀长度。 例如，如果 <b>hostPrefix</b> 设为 <b>23</b> ，则每个节点从所给的 <b>cidr</b> 中分配一个 <b>/23</b> 子网。 <b>hostPrefix</b> 值 <b>23</b> 提供 $510 (2^{(32 - 23)} - 2)$ 个 pod IP 地址。	子网前缀。  默认值为 <b>23</b> 。
<b>networking.serviceNetwork</b>	服务的 IP 地址块。默认值为 <b>172.30.0.0/16</b> 。  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>
<b>networking.machineNetwork</b>	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>
<b>networking.machineNetwork.cidr</b>	使用 <b>networking.machineNetwork</b> 时需要。IP 地址块。libvirt 以外的所有平台的默认值为 <b>10.0.0.0/16</b> 。对于 libvirt，默认值为 <b>192.168.126.0/24</b> 。	CIDR 表示法中的 IP 网络块。  例如： <b>10.0.0.0/16</b> 。   <b>注意</b>  将 <b>networking.machineNetwork</b> 设置为与首选 NIC 所在的 CIDR 匹配。



### 1.5.6.1.3. 可选配置参数

下表描述了可选安装配置参数：

表 1.7. 可选参数

参数	描述	值
<b>additionalTrustBundle</b>	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
<b>compute</b>	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。
<b>compute.architecture</b>	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 <b>amd64</b> （默认值）。	字符串
<b>compute.hyperthreading</b>	<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><b>重要</b></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>	<b>Enabled</b> 或 <b>Disabled</b>
<b>compute.name</b>	使用 <b>compute</b> 时需要此值。机器池的名称。	<b>worker</b>
<b>compute.platform</b>	使用 <b>compute</b> 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 <b>controlPlane.platform</b> 参数值匹配。	<b>aws、azure、gcp、openstack、o virt、vsphere</b> 或 <b>{}</b>
<b>compute.replicas</b>	要置备的计算机器数量，也称为 worker 机器。	大于或等于 <b>2</b> 的正整数。默认值为 <b>3</b> 。
<b>controlPlane</b>	组成 control plane 的机器的配置。	<b>MachinePool</b> 对象的数组。详情请查看以下"Machine-pool"表。
<b>controlPlane.architecture</b>	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 <b>amd64</b> （默认值）。	字符串



参数	描述	值
<b>controlPlane.hyperthreading</b>	<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><b>重要</b></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>	<b>Enabled 或 Disabled</b>
<b>controlPlane.name</b>	使用 <b>controlPlane</b> 时需要。机器池的名称。	<b>master</b>
<b>controlPlane.platform</b>	使用 <b>controlPlane</b> 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 <b>compute.platform</b> 参数值匹配。	<b>aws、azure、gcp、openstack、ovirt、vsphere 或 {}</b>
<b>controlPlane.replicas</b>	要置备的 control plane 机器数量。	唯一支持的值是 <b>3</b> ，它是默认值。
<b>credentialsMode</b>	<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> <p><b>注意</b></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>	<b>Mint、Passthrough、Manual 或空字符串("")。</b>

参数	描述	值
<b>fips</b>	<p>启用或禁用 FIPS 模式。默认为 <b>false</b>（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, black 2px, black 4px); margin-right: 10px;"></div> <div> <p><b>重要</b></p> <p>只有在 <b>x86_64</b> 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, black 2px, black 4px); margin-right: 10px;"></div> <div> <p><b>注意</b></p> <p>如果使用 Azure File 存储，则无法启用 FIPS 模式。</p> </div> </div>	<b>false</b> 或 <b>true</b>
<b>imageContentSources</b>	release-image 内容的源和仓库。	对象数组。包括一个 <b>source</b> 以及可选的 <b>mirrors</b> ，如下表所示。
<b>imageContentSources.source</b>	使用 <b>imageContentSources</b> 时需要。指定用户在镜像拉取规格中引用的仓库。	字符串
<b>imageContentSources.mirrors</b>	指定可能还包含同一镜像的一个或多个仓库。	字符串数组
<b>publish</b>	如何发布或公开集群的面向用户的端点，如 Kubernetes API、OpenShift 路由。	<b>Internal</b> 或 <b>External</b> 。把 <b>publish</b> 设置为 <b>Internal</b> 以部署一个私有集群，它不能被互联网访问。默认值为 <b>External</b> 。
<b>sshKey</b>	<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: 40px; height: 40px; background: repeating-linear-gradient(-45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px;"></div> <div> <p><b>注意</b></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 <b>ssh-agent</b> 进程使用的 SSH 密钥。</p> </div> </div>	<p>一个或多个密钥。例如：</p> <pre>sshKey:   &lt;key1&gt;   &lt;key2&gt;   &lt;key3&gt;</pre>

## 1.5.6.1.4. 可选的 AWS 配置参数

下表描述了可选的 AWS 配置参数：

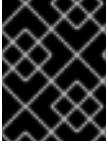
表 1.8. 可选的 AWS 参数

参数	描述	值
<code>compute.platform.aws.amiID</code>	用于为集群引导计算机器的 AWS AMI。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<code>compute.platform.aws.rootVolume.iops</code>	为根卷保留的每秒输入/输出操作 (IOPS) 数。	整数，如 <b>4000</b> 。
<code>compute.platform.aws.rootVolume.size</code>	以 GiB 为单位的根卷大小。	整数，如 <b>500</b> 。
<code>compute.platform.aws.rootVolume.type</code>	根卷的类型。	有效的 <a href="#">AWS EBS 卷类型</a> ，如 <b>io1</b> 。
<code>compute.platform.aws.rootVolume.kmsKeyARN</code>	KMS 密钥的 Amazon 资源名称 (密钥 ARN)。这是使用特定 KMS 密钥加密 worker 节点的操作系统卷。	有效的 <a href="#">密钥 ID 或密钥 ARN</a> 。
<code>compute.platform.aws.type</code>	计算机器的 EC2 实例类型。	有效的 <a href="#">AWS 实例类型</a> ，如 <b>c5.9xlarge</b> 。
<code>compute.platform.aws.zones</code>	安装程序在其中为计算机器池创建机器的可用区。如果您提供自己的 VPC，则必须在那个可用域中提供一个子网。	有效 AWS 可用区的列表，如 <b>us-east-1c</b> ，以 <a href="#">YAML 序列</a> 表示。
<code>compute.aws.region</code>	安装程序在其中创建计算资源的 AWS 区域。	任何有效的 <a href="#">AWS 区域</a> ，如 <b>us-east-1</b> 。
<code>controlPlane.platform.aws.amiID</code>	用于为集群引导 control plane 机器的 AWS AMI。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<code>controlPlane.platform.aws.rootVolume.kmsKeyARN</code>	KMS 密钥的 Amazon 资源名称 (密钥 ARN)。这需要特定的 KMS 密钥加密 control plane 节点的操作系统卷。	有效的 <a href="#">密钥 ID 和密钥 ARN</a> 。

参数	描述	值
<b>controlPlane.platform.aws.type</b>	control plane 机器的 EC2 实例类型。	有效的 <a href="#">AWS 实例类型</a> ，如 <b>c5.9xlarge</b> 。
<b>controlPlane.platform.aws.zones</b>	安装程序在其中为 control plane 机器池创建机器的可用区。	有效 AWS 可用区的列表，如 <b>us-east-1c</b> ，以 <a href="#">YAML 序列</a> 表示。
<b>controlPlane.aws.region</b>	安装程序在其中创建 control plane 资源的 AWS 区域。	有效的 <a href="#">AWS 区域</a> ，如 <b>us-east-1</b> 。
<b>platform.aws.amiID</b>	用于为集群引导所有机器的 AWS AMI。如果设置，AMI 必须属于与集群相同的区域。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<b>platform.aws.serviceEndpoints.name</b>	AWS 服务端点名称。只有在必须使用替代 AWS 端点（如 FIPS）时，才需要自定义端点。可以为 EC2、S3、IAM、Elastic Load Balancing、Tagging、Route 53 和 STS AWS 服务指定自定义 API 端点。	有效的 <a href="#">AWS 服务端点名称</a> 。
<b>platform.aws.serviceEndpoints.url</b>	AWS 服务端点 URL。URL 必须使用 <b>https</b> 协议，主机必须信任该证书。	有效的 <a href="#">AWS 服务端点 URL</a> 。
<b>platform.aws.userTags</b>	键与值的映射，安装程序将其作为标签添加到它所创建的所有资源。	任何有效的 YAML 映射，如 <b>&lt;key&gt;: &lt;value&gt;</b> 格式的键值对。如需有关 AWS 标签的更多信息，请参阅 AWS 文档中的 <a href="#">标记您的 Amazon EC2 资源</a> 。
<b>platform.aws.subnets</b>	如果您提供 VPC，而不是让安装程序为您创建 VPC，请指定要使用的集群子网。子网必须是您指定的同一 <b>machineNetwork[].cidr</b> 范围的一部分。对于标准集群，为每个可用区指定一个公共和私有子网。对于私有集群，为每个可用区指定一个私有子网。	有效的子网 ID。

### 1.5.6.2. AWS 的自定义 install-config.yaml 文件示例

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



## 重要

此示例 YAML 文件仅供参考。您必须使用安装程序来获取 `install-config.yaml` 文件，并且修改该文件。

```

apiVersion: v1
baseDomain: example.com 1
credentialsMode: Mint 2
controlPlane: 3 4
  hyperthreading: Enabled 5
  name: master
  platform:
    aws:
      zones:
      - us-west-2a
      - us-west-2b
    rootVolume:
      iops: 4000
      size: 500
      type: io1 6
    type: m5.xlarge
  replicas: 3
compute: 7
- hyperthreading: Enabled 8
  name: worker
  platform:
    aws:
      rootVolume:
        iops: 2000
        size: 500
        type: io1 9
      type: c5.4xlarge
      zones:
      - us-west-2c
    replicas: 3
  metadata:
    name: test-cluster 10
networking: 11
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  aws:
    region: us-west-2 12
  userTags:
    adminContact: jdoe
    costCenter: 7536
  amiID: ami-96c6f8f7 13
  serviceEndpoints: 14

```

```

- name: ec2
  url: https://vpce-id.ec2.us-west-2.vpce.amazonaws.com
  fips: false 15
  sshKey: ssh-ed25519 AAAA... 16
  pullSecret: '{"auths": ...}' 17

```

- 1 10 12 17** 必需。安装程序会提示您输入这个值。
- 2** 可选：添加此参数来强制 Cloud Credential Operator (CCO) 使用指定的模式，而不是让 CCO 动态尝试决定凭证的功能。如需有关 CCO 模式的详情，请参阅 *Red Hat Operator* 参考内容中的 *Cloud Credential Operator* 条目。
- 3 7 11** 如果没有提供这些参数和值，安装程序会提供默认值。
- 4** **controlPlane** 部分是一个单个映射，但 **compute** 部分是一系列映射。为满足不同数据结构的要求，**compute** 部分的第一行必须以连字符 - 开头，**controlPlane** 部分的第一行则不以连字符开头。只使用一个 control plane 池。
- 5 8** 是否要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。您可以通过将参数值设为 **Disabled** 来禁用。如果您在某些集群机器上禁用并发多线程，则必须在所有集群机器上禁用。



### 重要

如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。如果您对机器禁用并发多线程，请使用较大的实例类型，如 **m4.2xlarge** 或 **m5.2xlarge**。

- 6 9** 要为 etcd 配置更快的存储，特别是对于较大的集群，请将存储类型设置为 **io1**，并将 **iops** 设为 **2000**。
- 13** 用于为集群引导机器的 AMI ID。如果设置，AMI 必须属于与集群相同的区域。
- 14** AWS 服务端点。在安装到未知 AWS 区域时，需要自定义端点。端点 URL 必须使用 **https** 协议，主机必须信任该证书。
- 15** 是否启用或禁用 FIPS 模式。默认情况下不启用 FIPS 模式。如果启用了 FIPS 模式，运行 OpenShift Container Platform 的 Red Hat Enterprise Linux CoreOS (RHCOS) 机器会绕过默认的 Kubernetes 加密套件，并使用由 RHCOS 提供的加密模块。



### 重要

只有在 **x86\_64** 架构中的 OpenShift Container Platform 部署支持 FIPS 验证的 `/Modules in Process` 加密库。

- 16** 您可以选择提供您用来访问集群中机器的 **sshKey** 值。



### 注意

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

### 1.5.6.3. 在安装过程中配置集群范围代理

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

#### 先决条件

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



#### 注意

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

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

- 如果您的集群位于 AWS 上，请将 `ec2.<region>.amazonaws.com`、`elasticloadbalancing.<region>.amazonaws.com` 和 `s3.<region>.amazonaws.com` 端点添加到 VPC 端点。需要这些端点才能完成节点到 AWS EC2 API 的请求。由于代理在容器级别而不是节点级别工作，因此您必须通过 AWS 专用网络将这些请求路由到 AWS EC2 API。在代理服务器中的允许列表中添加 EC2 API 的公共 IP 地址是不够的。

#### 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` 的配置映射来保存额外的 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，将为 **trustedCA** 参数指定的内容与 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。

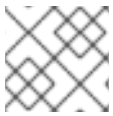


### 注意

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. Cluster Network Operator 配置

集群网络的配置作为 Cluster Network Operator (CNO) 配置的一部分被指定，并存储在名为 **cluster** 的自定义资源 (CR) 对象中。CR 指定 **operator.openshift.io** API 组中的 **Network** API 的字段。

CNO 配置会在集群安装过程中从 **Network.config.openshift.io** API 组中的 **Network** API 继承以下字段，这些字段无法更改：

### **clusterNetwork**

从中分配 pod IP 地址的 IP 地址池。

### **serviceNetwork**

服务的 IP 地址池。

### **defaultNetwork.type**

集群网络供应商，如 OpenShift SDN 或 OVN-Kubernetes。

您可以通过在名为 **cluster** 的 CNO 对象中设置 **defaultNetwork** 对象的字段来为集群指定集群网络供应商配置。

### 1.5.7.1. Cluster Network Operator 配置对象

Cluster Network Operator (CNO) 的字段在下表中描述：

表 1.9. Cluster Network Operator 配置对象

字段	类型	Description
<b>metadata.name</b>	字符串	CNO 对象的名称。这个名称始终是 <b>cluster</b> 。



字段	类型	Description
<b>spec.clusterNetwork</b>	数组	<p>用于指定从哪些 IP 地址块分配 Pod IP 地址以及分配给集群中每个节点的子网前缀长度的列表。例如：</p> <pre>spec:   clusterNetwork:   - cidr: 10.128.0.0/19     hostPrefix: 23   - cidr: 10.128.32.0/19     hostPrefix: 23</pre> <p>此值是只读的，并在 <b>install-config.yaml</b> 文件中指定。</p>
<b>spec.serviceNetwork</b>	数组	<p>服务的 IP 地址块。OpenShift SDN 和 OVN-Kubernetes Container Network Interface (CNI) 网络供应商只支持服务网络具有单个 IP 地址块。例如：</p> <pre>spec:   serviceNetwork:   - 172.30.0.0/14</pre> <p>此值是只读的，并在 <b>install-config.yaml</b> 文件中指定。</p>
<b>spec.defaultNetwork</b>	对象	为集群网络配置 Container Network Interface (CNI) 集群网络供应商。
<b>spec.kubeProxyConfig</b>	对象	此对象的字段指定 kube-proxy 配置。如果您使用 OVN-Kubernetes 集群网络供应商，则 kube-proxy 的配置不会起作用。

### defaultNetwork 对象配置

**defaultNetwork** 对象的值在下表中定义：

表 1.10. **defaultNetwork** 对象

字段	类型	Description
<b>type</b>	字符串	<p><b>OpenShiftSDN</b> 或 <b>OVNKubernetes</b>。在安装过程中选择了集群网络供应商。集群安装后无法更改这个值。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p><b>注意</b></p> <p>OpenShift Container Platform 默认使用 OpenShift SDN Container Network Interface (CNI) 集群网络供应商。</p> </div> </div>

字段	类型	Description
<b>openshiftSDNConfig</b>	对象	此对象仅对 OpenShift SDN 集群网络供应商有效。
<b>ovnKubernetesConfig</b>	对象	此对象仅对 OVN-Kubernetes 集群网络供应商有效。

### 配置 OpenShift SDN CNI 集群网络供应商

下表描述了 OpenShift SDN Container Network Interface (CNI) 集群网络供应商的配置字段。

表 1.11. **openshiftSDNConfig** 对象

字段	类型	Description
<b>mode</b>	字符串	配置 OpenShift SDN 的网络隔离模式。默认值为 <b>NetworkPolicy</b> 。  <b>Multitenant</b> 和 <b>Subnet</b> 的值可以向后兼容 OpenShift Container Platform 3.x, 但不推荐这样做。集群安装后无法更改这个值。
<b>mtu</b>	整数	VXLAN 覆盖网络的最大传输单元 (MTU)。这根据主网络接口的 MTU 自动探测。您通常不需要覆盖检测到的 MTU。  如果自动探测的值不是您期望的, 请确认节点上主网络接口中的 MTU 是正确的。您不能使用这个选项更改节点上主网络接口的 MTU 值。  如果您的集群中的不同节点需要不同的 MTU 值, 则必须将此值设置为比集群中的最低 MTU 值小 <b>50</b> 。例如, 如果集群中的某些节点的 MTU 为 <b>9001</b> , 而某些节点的 MTU 为 <b>1500</b> , 则必须将此值设置为 <b>1450</b> 。  集群安装后无法更改这个值。
<b>vxlanPort</b>	整数	用于所有 VXLAN 数据包的端口。默认值为 <b>4789</b> 。集群安装后无法更改这个值。  如果您在虚拟环境中运行, 并且现有节点是另一个 VXLAN 网络的一部分, 那么可能需要更改此值。例如, 当在 VMware NSX-T 上运行 OpenShift SDN 覆盖时, 您必须为 VXLAN 选择一个备用端口, 因为两个 SDN 都使用相同的默认 VXLAN 端口号。  在 Amazon Web Services (AWS) 上, 您可以在端口 <b>9000</b> 和端口 <b>9999</b> 之间为 VXLAN 选择一个备用端口。

### OpenShift SDN 配置示例

```
defaultNetwork:
  type: OpenShiftSDN
  openshiftSDNConfig:
```

```

mode: NetworkPolicy
mtu: 1450
vxlanPort: 4789

```

### 配置 OVN-Kubernetes CNI 集群网络供应商

下表描述了 OVN-Kubernetes CNI 集群网络供应商的配置字段。

表 1.12. `ovnKubernetesConfig` 对象

字段	类型	Description
<code>mtu</code>	整数	<p>Geneve (Generic Network Virtualization Encapsulation) 覆盖网络的最大传输单元 (MTU)。这根据主网络接口的 MTU 自动探测。您通常不需要覆盖检测到的 MTU。</p> <p>如果自动探测的值不是您期望的，请确认节点上主网络接口中的 MTU 是正确的。您不能使用这个选项更改节点上主网络接口的 MTU 值。</p> <p>如果您的集群中的不同节点需要不同的 MTU 值，则必须将此值设置为比集群中的最低 MTU 值小 <b>100</b>。例如，如果集群中的某些节点的 MTU 为 <b>9001</b>，而某些节点的 MTU 为 <b>1500</b>，则必须将此值设置为 <b>1400</b>。</p> <p>集群安装后无法更改这个值。</p>
<code>genevePort</code>	整数	<p>用于所有 Geneve 数据包的端口。默认值为 <b>6081</b>。集群安装后无法更改这个值。</p>

### OVN-Kubernetes 配置示例

```

defaultNetwork:
  type: OVNKubernetes
  ovnKubernetesConfig:
    mtu: 1400
    genevePort: 6081

```

### kubeProxyConfig 对象配置

`kubeProxyConfig` 对象的值在下表中定义：

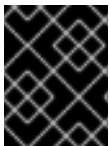
表 1.13. `kubeProxyConfig` 对象

字段	类型	Description
----	----	-------------

字段	类型	Description
<b>iptablesSyncPeriod</b>	字符串	<p><b>iptables</b> 规则的刷新周期。默认值为 <b>30s</b>。有效的后缀包括 <b>s</b>、<b>m</b> 和 <b>h</b>，具体参见 <a href="#">Go time 软件包文档</a>。</p> <p> <b>注意</b></p> <p>由于 OpenShift Container Platform 4.3 及更高版本中引进了性能上的改进，现在不再需要调整 <b>iptablesSyncPeriod</b> 参数。</p>
<b>proxyArguments.iptables-min-sync-period</b>	数组	<p>刷新 <b>iptables</b> 规则前的最短时长。此字段确保刷新的频率不会过于频繁。有效的后缀包括 <b>s</b>、<b>m</b> 和 <b>h</b>，具体参见 <a href="#">Go time 软件包</a>。默认值为：</p> <pre>kubeProxyConfig:   proxyArguments:     iptables-min-sync-period:       - 0s</pre>

### 1.5.8. 指定高级网络配置

您可以通过为集群网络供应商指定额外的配置，使用高级配置自定义将集群整合到现有网络环境中。您只能在安装集群前指定高级网络配置。



#### 重要

不支持修改安装程序创建的 OpenShift Container Platform 清单文件。支持应用您创建的清单文件，如以下流程所示。

#### 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. 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。

2. 在 **<installation\_directory>/manifests/** 目录下，为高级网络配置创建一个名为 **cluster-network-03-config.yml** 的 stub 清单文件：

```
$ cat <<EOF > <installation_directory>/manifests/cluster-network-03-config.yml
apiVersion: operator.openshift.io/v1
```

```
kind: Network
metadata:
  name: cluster
spec:
EOF
```

其中：

#### <installation\_directory>

指定包含集群的 **manifests/** 目录的目录名称。

3. 在编辑器中打开 **cluster-network-03-config.yml** 文件，并为集群指定高级网络配置，如下例所示：

#### 为 OpenShift SDN 网络供应商指定不同的 VXLAN 端口

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

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



#### 注意

有关在 AWS 中使用网络负载均衡（Network Load Balancer）的更多信息，请参阅 [使用网络负载均衡器在 AWS 上配置 Ingress 集群流量](#)。

### 1.5.9. 在新 AWS 集群上配置 Ingress Controller 网络负载均衡

您可在新集群中创建一个由 AWS Network Load Balancer（NLB）支持的 Ingress Controller。

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

在新集群中，创建一个由 AWS NLB 支持的 Ingress Controller。

1. 进入包含安装程序的目录并创建清单：

```
$ ./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。

- 在 `<installation_directory>/manifests/` 目录中创建一个名为 `cluster-ingress-default-ingresscontroller.yaml` 的文件：

```
$ touch <installation_directory>/manifests/cluster-ingress-default-ingresscontroller.yaml 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-ingress-default-ingresscontroller.yaml
```

### 输出示例

```
cluster-ingress-default-ingresscontroller.yaml
```

- 在编辑器中打开 `cluster-ingress-default-ingresscontroller.yaml` 文件，并输入描述您想要的 Operator 配置的 CR:

```
apiVersion: operator.openshift.io/v1
kind: IngressController
metadata:
  creationTimestamp: null
  name: default
  namespace: openshift-ingress-operator
spec:
  endpointPublishingStrategy:
    loadBalancer:
      scope: External
      providerParameters:
        type: AWS
      aws:
        type: NLB
    type: LoadBalancerService
```

- 保存 `cluster-ingress-default-ingresscontroller.yaml` 文件并退出文本编辑器。
- 可选：备份 `manifests/cluster-ingress-default-ingresscontroller.yaml` 文件。创建集群时，安装程序会删除 `manifests/` 目录。

## 1.5.10. 使用 OVN-Kubernetes 配置混合网络

您可以将集群配置为使用 OVN-Kubernetes 的混合网络。这允许支持不同节点网络配置的混合集群。例如：集群中运行 Linux 和 Windows 节点时需要这样做。



### 重要

您必须在安装集群过程中使用 OVN-Kubernetes 配置混合网络。您不能在安装过程中切换到混合网络。

先决条件

- 您在 `install-config.yaml` 文件中为 `networking.networkType` 参数定义了 `OVNKubernetes`。如需更多信息，请参阅有关在所选云供应商上配置 OpenShift Container Platform 网络自定义的安装文档。

## 流程

1. 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。

2. 在 `<installation_directory>/manifests/` 目录下，为高级网络配置创建一个名为 `cluster-network-03-config.yml` 的 stub 清单文件：

```
$ cat <<EOF > <installation_directory>/manifests/cluster-network-03-config.yml
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
EOF
```

其中：

### <installation\_directory>

指定包含集群的 `manifests/` 目录的目录名称。

3. 在编辑器中打开 `cluster-network-03-config.yml` 文件，并使用混合网络配置 OVN-Kubernetes，如下例所示：

## 指定混合网络配置

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    ovnKubernetesConfig:
      hybridOverlayConfig:
        hybridClusterNetwork: ①
        - cidr: 10.132.0.0/14
        hostPrefix: 23
        hybridOverlayVXLANPort: 9898 ②
```

① 指定用于额外覆盖网络上节点的 CIDR 配置。`hybridClusterNetwork` CIDR 无法与 `clusterNetwork` CIDR 重叠。

② 为额外覆盖网络指定自定义 VXLAN 端口。这是在 vSphere 上安装的集群中运行 Windows

4. 保存 `cluster-network-03-config.yml` 文件，再退出文本编辑器。
5. 可选：备份 `manifests/cluster-network-03-config.yml` 文件。创建集群时，安装程序会删除 `manifests/` 目录。

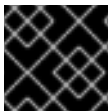


### 注意

有关在同一集群中使用 Linux 和 Windows 节点的更多信息，请参阅 [了解 Windows 容器工作负载](#)。

## 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

- 配置托管集群的云平台的帐户。
- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

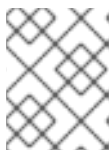
#### 流程

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
```



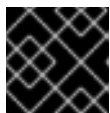
### 注意

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

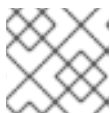
- 安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrap** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅从过期的 *control plane* 证书中恢复的文档。
- 建议您在 Ignition 配置文件生成后的 12 小时内使用它们，因为 24 小时的证书会在集群安装后的 16 小时到 22 小时进行轮转。通过在 12 小时内使用 Ignition 配置文件，您可以避免在安装过程中因为执行了证书更新而导致安装失败的问题。



### 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。

2. 可选：从您用来安装集群的 IAM 帐户删除或禁用 **AdministratorAccess** 策略。



### 注意

只有在安装过程中才需要 **AdministratorAccess** 策略提供的升级权限。

## 1.5.12. 通过下载二进制文件安装 OpenShift CLI

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



### 重要

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

### 1.5.12.1. 在 Linux 上安装 OpenShift CLI

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

#### 流程

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

#### 4. 解包存档：

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

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

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 1.5.12.2. 在 Windows 上安装 OpenShift CLI

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

#### 流程

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

```
C:\> path
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 1.5.12.3. 在 macOS 上安装 OpenShift CLI

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

#### 流程

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

■

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 1.5.13. 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
```

### 1.5.14. 使用 Web 控制台登录到集群

**kubeadmin** 用户默认在 OpenShift Container Platform 安装后存在。您可以使用 OpenShift Container Platform Web 控制台以 **kubeadmin** 用户身份登录集群。

先决条件

- 有访问安装主机的访问权限。
- 您完成了集群安装，所有集群 Operator 都可用。

流程

1. 从安装主机上的 **kubeadmin -password** 文件中获取 kubeadmin 用户的密码：

```
$ cat <installation_directory>/auth/kubeadmin-password
```



### 注意

另外，您还可以从安装主机上的 `<installation_directory>/openshift_install.log` 日志文件获取 **kubeadmin** 密码。

2. 列出 OpenShift Container Platform Web 控制台路由：

```
$ oc get routes -n openshift-console | grep 'console-openshift'
```



### 注意

另外，您还可以从安装主机上的 `<installation_directory>/openshift_install.log` 日志文件获取 OpenShift Container Platform 路由。

### 输出示例

```
console    console-openshift-console.apps.<cluster_name>.<base_domain>    console
https      reencrypt/Redirect      None
```

3. 在 Web 浏览器中导航到上一命令输出中包括的路由，以 **kubeadmin** 用户身份登录。

### 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](#)。

## 1.5.15. 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

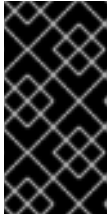
- [验证安装](#)。
- [自定义集群](#)。
- 如果需要，您可以[选择不使用远程健康报告](#)。
- 如果需要，您可以[删除云供应商凭证](#)。

## 1.6. 在 AWS 上将集群安装到现有的 VPC 中

在 OpenShift Container Platform 版本 4.6 中，您可以在 Amazon Web Services (AWS) 上将集群安装到现有 Amazon Virtual Private Cloud (VPC) 中。安装程序会置备所需基础架构的其余部分，您可以进一步定制这些基础架构。要自定义安装，请在安装集群前修改 `install-config.yaml` 文件中的参数。

### 1.6.1. 先决条件

- 查看有关 [OpenShift Container Platform 安装和更新](#) 流程的详细信息。
- [配置 AWS 帐户](#) 以托管集群。



#### 重要

如果您的计算机上存储有 AWS 配置集，则不要在使用多因素验证设备的同时使用您生成的临时会话令牌。在集群的整个生命周期中，集群会持续使用您的当前 AWS 凭证来创建 AWS 资源，因此您必须使用长期凭证。要生成适当的密钥，请参阅 AWS 文档中的[管理 IAM 用户的访问密钥](#)。您可在运行安装程序时提供密钥。

- 如果使用防火墙，则必须将其配置为允许集群需要访问的站点。
- 如果不允许系统管理身份和访问管理 (IAM)，集群管理员可以 [手动创建和维护 IAM 凭证](#)。手动模式也可以用于云 IAM API 无法访问的环境中。

### 1.6.2. 关于使用自定义 VPC

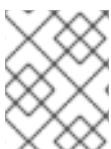
在 OpenShift Container Platform 4.6 中，您可以在 Amazon Web Services (AWS) 的现有 Amazon Virtual Private Cloud (VPC) 中将集群部署到现有子网中。通过将 OpenShift Container Platform 部署到现有的 AWS VPC 中，您可能会避开新帐户中的限制，或者更容易地利用公司所设置的操作限制。如果您无法获得您自己创建 VPC 所需的基础架构创建权限，请使用这个安装选项。

因为安装程序无法了解您现有子网中还有哪些其他组件，所以无法选择子网 CIDR。您必须为安装集群的子网配置网络。

#### 1.6.2.1. 使用 VPC 的要求

安装程序不再创建以下组件：

- 互联网网关
- NAT 网关
- 子网
- 路由表
- VPCs
- VPC DHCP 选项
- VPC 端点



#### 注意

安装程序要求您使用由云提供的 DNS 服务器。不支持使用自定义 DNS 服务器，并导致安装失败。

如果您使用自定义 VPC，您必须为安装程序和集群正确配置它及其子网。如需有关创建和管理 AWS VPC 的更多信息，请参阅 [AWS 文档中的 Amazon VPC 控制台向导配置和工作 VPC 和子网](#)。

安装程序无法：

- 细分供集群使用的网络范围。
- 为子网设置路由表。
- 设置 VPC 选项，如 DHCP。

在安装集群前，您必须完成这些任务。有关在 AWS VPC 中配置网络的更多信息，请参阅 [VPC 的 VPC 网络组件和路由表](#)。

您的 VPC 必须满足以下特征：

- 为集群使用的每个可用区创建一个公共和私有子网。每个可用区不能包含多于一个的公共子网和专用子网。有关这种类型的配置示例，请参阅 [AWS 文档中的 带有公共和私有子网\(NAT\)的 VPC](#)。  
记录每个子网 ID。完成安装需要您在 `install-config.yaml` 文件的 `platform` 部分中输入这些值。请参阅 [AWS 文档中的 查找子网 ID](#)。
- VPC 的 CIDR 块必须包含 `Networking.machineCIDR`，它是集群机器的 IP 地址池。子网 CIDR 块必须属于您指定的机器 CIDR。
- VPC 必须附加有公共互联网网关。对于每个可用区：
  - 公共子网需要路由到互联网网关的路由。
  - 公共子网需要一个具有 EIP 地址的 NAT 网关。
  - 专用子网需要路由到公共子网中的 NAT 网关。
- VPC 不能使用 `kubernetes.io/cluster/.*: owned` 标签。  
安装程序会修改子网以添加 `kubernetes.io/cluster/.*: shared` 标签，因此您的子网必须至少有一个可用的空闲标签插槽。请参阅 [AWS 文档中的 标签限制](#) 部分，以确认安装程序可以为您指定的每个子网添加标签。
- 您必须在 VPC 中启用 `enableDnsSupport` 和 `enableDnsHostnames` 属性，以便集群可以使用附加到 VPC 的 Route 53 区来解析集群内部 DNS 记录。请参阅 [AWS 文档中的您的 VPC 中的 DNS 支持部分](#)。  
如果要使用自己的 Route 53 托管私有区，您必须在安装集群前将现有托管区与 VPC 关联。您可以使用 `install-config.yaml` 文件中的 `platform.aws.hostedZone` 字段定义托管区。

如果您在断开连接的环境中工作，您将无法访问 EC2 和 ELB 端点的公共 IP 地址。要解决这个问题，您必须创建一个 VPC 端点，并将其附加到集群使用的子网。端点应命名如下：

- `ec2.<region>.amazonaws.com`
- `elasticloadbalancing.<region>.amazonaws.com`
- `s3.<region>.amazonaws.com`

## 所需的 VPC 组件

您必须提供合适的 VPC 和子网，以便与您的机器通信。

组件	AWS 类型	描述	
VPC	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::VPC</b></li> <li>● <b>AWS::EC2::VPCEndpoint</b></li> </ul>	您必须提供一个公共 VPC 供集群使用。VPC 使用引用每个子网的路由表的端点，以改进与托管在 S3 中的 registry 的通信。	
公共子网	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::Subnet</b></li> <li>● <b>AWS::EC2::SubnetNetworkACLAssociation</b></li> </ul>	您的 VPC 必须有 1 到 3 个可用区的公共子网，并将其与适当的入口规则关联。	
互联网网关	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::InternetGateway</b></li> <li>● <b>AWS::EC2::VPCGatewayAttachment</b></li> <li>● <b>AWS::EC2::RouteTable</b></li> <li>● <b>AWS::EC2::Route</b></li> <li>● <b>AWS::EC2::SubnetRouteTableAssociation</b></li> <li>● <b>AWS::EC2::NatGateway</b></li> <li>● <b>AWS::EC2::EIP</b></li> </ul>	您必须有一个公共互联网网关，以及附加到 VPC 的公共路由。在提供的模板中，每个公共子网都有一个具有 EIP 地址的 NAT 网关。这些 NAT 网关允许集群资源（如专用子网实例）访问互联网，而有些受限网络或代理场景则不需要它们。	
网络访问控制	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::NetworkACL</b></li> <li>● <b>AWS::EC2::NetworkACLEntry</b></li> </ul>	您必须允许 VPC 访问下列端口：	
		端口	原因
		80	入站 HTTP 流量
		443	入站 HTTPS 流量
		22	入站 SSH 流量
		1024 - 65535	入站临时流量
0 - 65535	出站临时流量		
专用子网	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::Subnet</b></li> <li>● <b>AWS::EC2::RouteTable</b></li> <li>● <b>AWS::EC2::SubnetRouteTableAssociation</b></li> </ul>	您的 VPC 可以具有私有子网。提供的 CloudFormation 模板可为 1 到 3 个可用区创建专用子网。如果您使用专用子网，必须为其提供适当的路由和表。	

### 1.6.2.2. VPC 验证

要确保您提供的子网适合您的环境，安装程序会确认以下信息：

- 您指定的所有子网都存在。
- 您提供了私有子网。
- 子网 CIDR 属于您指定的机器 CIDR。
- 您为每个可用区提供子网。每个可用区不包含多于一个的公共子网和私有子网。如果您使用私有集群，为每个可用区只提供一个私有子网。否则，为每个可用区提供一个公共和私有子网。
- 您可以为每个私有子网可用区提供一个公共子网。机器不会在没有为其提供私有子网的可用区中置备。

如果您销毁使用现有 VPC 的集群，VPC 不会被删除。从 VPC 中删除 OpenShift Container Platform 集群时，`kubernetes.io/cluster/.*: shared` 标签会从使用它的子网中删除。

### 1.6.2.3. 权限划分

从 OpenShift Container Platform 4.3 开始，您不需要安装程序置备的基础架构集群部署所需的所有权限。这与您所在机构可能已有的权限划分类似：不同的人可以在您的云中创建不同的资源。例如，您可以创建针对于特定应用程序的对象，如实例、存储桶和负载均衡器，但不能创建与网络相关的组件，如 VPC、子网或入站规则。

您在创建集群时使用的 AWS 凭证不需要 VPC 和 VPC 中的核心网络组件（如子网、路由表、互联网网关、NAT 和 VPN）所需的网络权限。您仍然需要获取集群中的机器需要的应用程序资源的权限，如 ELB、安全组、S3 存储桶和节点。

### 1.6.2.4. 集群间隔离

如果您将 OpenShift Container Platform 部署到现有网络中，集群服务的隔离将在以下方面减少：

- 您可以在同一 VPC 中安装多个 OpenShift Container Platform 集群。
- 整个网络允许 ICMP 入站流量。
- 整个网络都允许 TCP 22 入站流量 (SSH)。
- 整个网络都允许 control plane TCP 6443 入站流量 (Kubernetes API)。
- 整个网络都允许 control plane TCP 22623 入站流量 (MCS)。

## 1.6.3. 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。



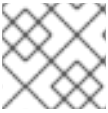


### 重要

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。

## 1.6.4. 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。

## 1.6.5. 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。

### 1.6.6. 创建安装配置文件

您可以自定义在 Amazon Web Services (AWS) 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 **AWS** 作为目标平台。
          - iii. 如果计算机上没有保存 Amazon Web Services (AWS) 配置集，请为您配置用于运行安装程序的用户输入 AWS 访问密钥 ID 和 Secret 访问密钥。
          - iv. 选择要将集群部署到的 AWS 区域。
          - v. 选择您为集群配置的 Route 53 服务的基域。
          - vi. 为集群输入一个描述性名称。
          - vii. 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。

#### 1.6.6.1. 安装配置参数

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



### 注意

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



### 重要

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

##### 1.6.6.1.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 1.14. 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 <b>v1</b> 。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>&lt;metadata.name&gt;.&lt;baseDomain&gt;</code> 。	完全限定域名或子域名，如 <code>example.com</code> 。
<code>metadata</code>	Kubernetes 资源 <b>ObjectMeta</b> ，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 <code>dev</code> 。

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

#### 1.6.6.1.2. 网络配置参数

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

只支持 IPv4 地址。

表 1.15. 网络参数

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



参数	描述	值
<b>networking.clusterNetwork</b>	pod 的 IP 地址块。  默认值为 <b>10.128.0.0/14</b> ，主机前缀为 <b>/23</b> 。  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>
<b>networking.clusterNetwork.cidr</b>	使用 <b>networking.clusterNetwork</b> 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 <b>0</b> 到 <b>32</b> 之间。
<b>networking.clusterNetwork.hostPrefix</b>	分配给每个单独节点的子网前缀长度。 例如，如果 <b>hostPrefix</b> 设为 <b>23</b> ，则每个节点从所给的 <b>cidr</b> 中分配一个 <b>/23</b> 子网。 <b>hostPrefix</b> 值 <b>23</b> 提供 510 ( $2^{(32-23)} - 2$ ) 个 pod IP 地址。	子网前缀。  默认值为 <b>23</b> 。
<b>networking.serviceNetwork</b>	服务的 IP 地址块。默认值为 <b>172.30.0.0/16</b> 。  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>
<b>networking.machineNetwork</b>	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>
<b>networking.machineNetwork.cidr</b>	使用 <b>networking.machineNetwork</b> 时需要。IP 地址块。libvirt 以外的所有平台的默认值为 <b>10.0.0.0/16</b> 。对于 libvirt，默认值为 <b>192.168.126.0/24</b> 。	CIDR 表示法中的 IP 网络块。  例如： <b>10.0.0.0/16</b> 。   <b>注意</b>  将 <b>networking.machineNetwork</b> 设置为与首选 NIC 所在的 CIDR 匹配。

## 1.6.6.1.3. 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
<b>additionalTrustBundle</b>	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
<b>compute</b>	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。
<b>compute.architecture</b>	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 <b>amd64</b> （默认值）。	字符串
<b>compute.hyperthreading</b>	<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><b>重要</b></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>	<b>Enabled</b> 或 <b>Disabled</b>
<b>compute.name</b>	使用 <b>compute</b> 时需要此值。机器池的名称。	<b>worker</b>
<b>compute.platform</b>	使用 <b>compute</b> 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 <b>controlPlane.platform</b> 参数值匹配。	<b>aws、azure、gcp、openstack、o virt、vsphere</b> 或 <b>{}</b>
<b>compute.replicas</b>	要置备的计算机器数量，也称为 worker 机器。	大于或等于 <b>2</b> 的正整数。默认值为 <b>3</b> 。
<b>controlPlane</b>	组成 control plane 的机器的配置。	<b>MachinePool</b> 对象的数组。详情请查看以下"Machine-pool"表。
<b>controlPlane.architecture</b>	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 <b>amd64</b> （默认值）。	字符串

参数	描述	值
<b>controlPlane.hyperthreading</b>	<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 style="margin-left: 10px;"> <p><b>重要</b></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>	<b>Enabled 或 Disabled</b>
<b>controlPlane.name</b>	使用 <b>controlPlane</b> 时需要。机器池的名称。	<b>master</b>
<b>controlPlane.platform</b>	使用 <b>controlPlane</b> 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 <b>compute.platform</b> 参数值匹配。	<b>aws、azure、gcp、openstack、ovirt、vsphere 或 {}</b>
<b>controlPlane.replicas</b>	要置备的 control plane 机器数量。	唯一支持的值是 <b>3</b> ，它是默认值。
<b>credentialsMode</b>	<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: center;">  <div style="margin-left: 10px;"> <p><b>注意</b></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>	<b>Mint、Passthrough、Manual 或空字符串("")。</b>



参数	描述	值
<b>fips</b>	<p>启用或禁用 FIPS 模式。默认为 <b>false</b>（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, black 2px, black 4px); margin-right: 10px;"></div> <div> <p><b>重要</b></p> <p>只有在 <b>x86_64</b> 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, black 2px, black 4px); margin-right: 10px;"></div> <div> <p><b>注意</b></p> <p>如果使用 Azure File 存储，则无法启用 FIPS 模式。</p> </div> </div>	<b>false</b> 或 <b>true</b>
<b>imageContentSources</b>	release-image 内容的源和仓库。	对象数组。包括一个 <b>source</b> 以及可选的 <b>mirrors</b> ，如下表所示。
<b>imageContentSources.source</b>	使用 <b>imageContentSources</b> 时需要。指定用户在镜像拉取规格中引用的仓库。	字符串
<b>imageContentSources.mirrors</b>	指定可能还包含同一镜像的一个或多个仓库。	字符串数组
<b>publish</b>	如何发布或公开集群的面向用户的端点，如 Kubernetes API、OpenShift 路由。	<b>Internal</b> 或 <b>External</b> 。把 <b>publish</b> 设置为 <b>Internal</b> 以部署一个私有集群，它不能被互联网访问。默认值为 <b>External</b> 。
<b>sshKey</b>	<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: 40px; height: 40px; background: repeating-linear-gradient(-45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px;"></div> <div> <p><b>注意</b></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 <b>ssh-agent</b> 进程使用的 SSH 密钥。</p> </div> </div>	<p>一个或多个密钥。例如：</p> <pre>sshKey:   &lt;key1&gt;   &lt;key2&gt;   &lt;key3&gt;</pre>

## 1.6.6.1.4. 可选的 AWS 配置参数

下表描述了可选的 AWS 配置参数：

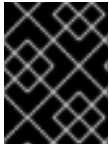
表 1.17. 可选的 AWS 参数

参数	描述	值
<code>compute.platform.aws.amiID</code>	用于为集群引导计算机器的 AWS AMI。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<code>compute.platform.aws.rootVolume.iops</code>	为根卷保留的每秒输入/输出操作 (IOPS) 数。	整数，如 <b>4000</b> 。
<code>compute.platform.aws.rootVolume.size</code>	以 GiB 为单位的根卷大小。	整数，如 <b>500</b> 。
<code>compute.platform.aws.rootVolume.type</code>	根卷的类型。	有效的 <a href="#">AWS EBS 卷类型</a> ，如 <b>io1</b> 。
<code>compute.platform.aws.rootVolume.kmsKeyARN</code>	KMS 密钥的 Amazon 资源名称 (密钥 ARN)。这是使用特定 KMS 密钥加密 worker 节点的操作系统卷。	有效的 <a href="#">密钥 ID 或密钥 ARN</a> 。
<code>compute.platform.aws.type</code>	计算机器的 EC2 实例类型。	有效的 <a href="#">AWS 实例类型</a> ，如 <b>c5.9xlarge</b> 。
<code>compute.platform.aws.zones</code>	安装程序在其中为计算机器池创建机器的可用区。如果您提供自己的 VPC，则必须在那个可用域中提供一个子网。	有效 AWS 可用区的列表，如 <b>us-east-1c</b> ，以 <a href="#">YAML 序列</a> 表示。
<code>compute.aws.region</code>	安装程序在其中创建计算资源的 AWS 区域。	任何有效的 <a href="#">AWS 区域</a> ，如 <b>us-east-1</b> 。
<code>controlPlane.platform.aws.amiID</code>	用于为集群引导 control plane 机器的 AWS AMI。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<code>controlPlane.platform.aws.rootVolume.kmsKeyARN</code>	KMS 密钥的 Amazon 资源名称 (密钥 ARN)。这需要特定的 KMS 密钥加密 control plane 节点的操作系统卷。	有效的 <a href="#">密钥 ID 和密钥 ARN</a> 。

参数	描述	值
<b>controlPlane.platform.aws.type</b>	control plane 机器的 EC2 实例类型。	有效的 <a href="#">AWS 实例类型</a> ，如 <b>c5.9xlarge</b> 。
<b>controlPlane.platform.aws.zones</b>	安装程序在其中为 control plane 机器池创建机器的可用区。	有效 AWS 可用区的列表，如 <b>us-east-1c</b> ，以 <a href="#">YAML 序列</a> 表示。
<b>controlPlane.aws.region</b>	安装程序在其中创建 control plane 资源的 AWS 区域。	有效的 <a href="#">AWS 区域</a> ，如 <b>us-east-1</b> 。
<b>platform.aws.amiID</b>	用于为集群引导所有机器的 AWS AMI。如果设置，AMI 必须属于与集群相同的区域。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<b>platform.aws.serviceEndpoints.name</b>	AWS 服务端点名称。只有在必须使用替代 AWS 端点（如 FIPS）时，才需要自定义端点。可以为 EC2、S3、IAM、Elastic Load Balancing、Tagging、Route 53 和 STS AWS 服务指定自定义 API 端点。	有效的 <a href="#">AWS 服务端点名称</a> 。
<b>platform.aws.serviceEndpoints.url</b>	AWS 服务端点 URL。URL 必须使用 <b>https</b> 协议，主机必须信任该证书。	有效的 <a href="#">AWS 服务端点 URL</a> 。
<b>platform.aws.userTags</b>	键与值的映射，安装程序将其作为标签添加到它所创建的所有资源。	任何有效的 YAML 映射，如 <b>&lt;key&gt;: &lt;value&gt;</b> 格式的键值对。如需有关 AWS 标签的更多信息，请参阅 AWS 文档中的 <a href="#">标记您的 Amazon EC2 资源</a> 。
<b>platform.aws.subnets</b>	如果您提供 VPC，而不是让安装程序为您创建 VPC，请指定要使用的集群子网。子网必须是您指定的同一 <b>machineNetwork[].cidr</b> 范围的一部分。对于标准集群，为每个可用区指定一个公共和私有子网。对于私有集群，为每个可用区指定一个私有子网。	有效的子网 ID。

### 1.6.6.2. AWS 的自定义 install-config.yaml 文件示例

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



## 重要

此示例 YAML 文件仅供参考。您必须使用安装程序来获取 `install-config.yaml` 文件，并且修改该文件。

```

apiVersion: v1
baseDomain: example.com 1
credentialsMode: Mint 2
controlPlane: 3 4
  hyperthreading: Enabled 5
  name: master
  platform:
    aws:
      zones:
        - us-west-2a
        - us-west-2b
      rootVolume:
        iops: 4000
        size: 500
        type: io1 6
      type: m5.xlarge
    replicas: 3
compute: 7
- hyperthreading: Enabled 8
  name: worker
  platform:
    aws:
      rootVolume:
        iops: 2000
        size: 500
        type: io1 9
      type: c5.4xlarge
      zones:
        - us-west-2c
    replicas: 3
metadata:
  name: test-cluster 10
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  aws:
    region: us-west-2 11
    userTags:
      adminContact: jdoe
      costCenter: 7536
  subnets: 12
    - subnet-1

```

```

- subnet-2
- subnet-3
amiID: ami-96c6f8f7 13
serviceEndpoints: 14
  - name: ec2
    url: https://vpce-id.ec2.us-west-2.vpce.amazonaws.com
hostedZone: Z3URY6TWQ91KVV 15
fips: false 16
sshKey: ssh-ed25519 AAAA... 17
pullSecret: '{"auths": ...}' 18

```

1 10 11 18 必需。安装程序会提示您输入这个值。

2 可选：添加此参数来强制 Cloud Credential Operator (CCO) 使用指定的模式，而不是让 CCO 动态尝试决定凭证的功能。如需有关 CCO 模式的详情，请参阅 *Red Hat Operator* 参考内容中的 *Cloud Credential Operator* 条目。

3 7 如果没有提供这些参数和值，安装程序会提供默认值。

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

5 8 是否要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。您可以通过将参数值设为 **Disabled** 来禁用。如果您在某些集群机器上禁用并发多线程，则必须在所有集群机器上禁用。



### 重要

如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。如果您对机器禁用并发多线程，请使用较大的实例类型，如 **m4.2xlarge** 或 **m5.2xlarge**。

6 9 要为 etcd 配置更快的存储，特别是对于较大的集群，请将存储类型设置为 **io1**，并将 **iops** 设为 **2000**。

12 如果您提供自己的 VPC，为集群使用的每个可用区指定子网。

13 用于为集群引导机器的 AMI ID。如果设置，AMI 必须属于与集群相同的区域。

14 AWS 服务端点。在安装到未知 AWS 区域时，需要自定义端点。端点 URL 必须使用 **https** 协议，主机必须信任该证书。

15 您现有 Route 53 私有托管区的 ID。提供现有的托管区需要您提供自己的 VPC，托管区已在安装集群前与 VPC 关联。如果未定义，安装程序会创建一个新的托管区。

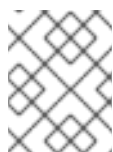
16 是否启用或禁用 FIPS 模式。默认情况下不启用 FIPS 模式。如果启用了 FIPS 模式，运行 OpenShift Container Platform 的 Red Hat Enterprise Linux CoreOS (RHCOS) 机器会绕过默认的 Kubernetes 加密套件，并使用由 RHCOS 提供的加密模块。



### 重要

只有在 **x86\_64** 架构中的 OpenShift Container Platform 部署支持 FIPS 验证的 `/Modules in Process` 加密库。

- 17 您可以选择提供您用来访问集群中机器的 `sshKey` 值。



### 注意

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

### 1.6.6.3. 在安装过程中配置集群范围代理

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

#### 先决条件

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



### 注意

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

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

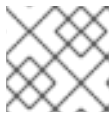
- 如果您的集群位于 AWS 上，请将 `ec2.<region>.amazonaws.com`、`elasticloadbalancing.<region>.amazonaws.com` 和 `s3.<region>.amazonaws.com` 端点添加到 VPC 端点。需要这些端点才能完成节点到 AWS EC2 API 的请求。由于代理在容器级别而不是节点级别工作，因此您必须通过 AWS 专用网络将这些请求路由到 AWS EC2 API。在代理服务器中的允许列表中添加 EC2 API 的公共 IP 地址是不够的。

#### 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** 的配置映射来保存额外的 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，将为 **trustedCA** 参数指定的内容与 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**。

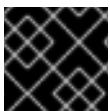


### 注意

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

## 1.6.7. 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

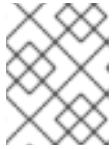
- 配置托管集群的云平台的帐户。
- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

#### 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。

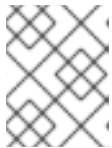
集群部署完成后，终端会显示访问集群的信息，包括指向其 Web 控制台的链接和 **kubeadmin** 用户的凭证。

### 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

```



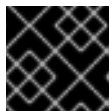
### 注意

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

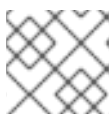
- 安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrap** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅 *从过期的 control plane 证书中恢复* 的文档。
- 建议您在 Ignition 配置文件生成后的 12 小时内使用它们，因为 24 小时的证书会在集群安装后的 16 小时到 22 小时进行轮转。通过在 12 小时内使用 Ignition 配置文件，您可以避免在安装过程中因为执行了证书更新而导致安装失败的问题。



### 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。

2. 可选：从您用来安装集群的 IAM 帐户删除或禁用 **AdministratorAccess** 策略。



### 注意

只有在安装过程中才需要 **AdministratorAccess** 策略提供的升级权限。

## 1.6.8. 通过下载二进制文件安装 OpenShift CLI

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





## 重要

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

### 1.6.8.1. 在 Linux 上安装 OpenShift CLI

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

#### 流程

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

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

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

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 1.6.8.2. 在 Windows 上安装 OpenShift CLI

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

#### 流程

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

```
C:\> path
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 1.6.8.3. 在 macOS 上安装 OpenShift CLI

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

#### 流程

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

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 1.6.9. 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
```

## 1.6.10. 使用 Web 控制台登录到集群

**kubeadmin** 用户默认在 OpenShift Container Platform 安装后存在。您可以使用 OpenShift Container Platform Web 控制台以 **kubeadmin** 用户身份登录集群。

先决条件

- 有访问安装主机的访问权限。
- 您完成了集群安装，所有集群 Operator 都可用。

流程

1. 从安装主机上的 **kubeadmin -password** 文件中获取 kubeadmin 用户的密码：

```
$ cat <installation_directory>/auth/kubeadmin-password
```



### 注意

另外，您还可以从安装主机上的 **<installation\_directory>/openshift\_install.log** 日志文件获取 **kubeadmin** 密码。

2. 列出 OpenShift Container Platform Web 控制台路由：

```
$ oc get routes -n openshift-console | grep 'console-openshift'
```



### 注意

另外，您还可以从安装主机上的 **<installation\_directory>/openshift\_install.log** 日志文件获取 OpenShift Container Platform 路由。

### 输出示例

```
console    console-openshift-console.apps.<cluster_name>.<base_domain>    console
https reencrypt/Redirect None
```

3. 在 Web 浏览器中导航到上一命令输出中包括的路由，以 **kubeadmin** 用户身份登录。

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](#)。

## 1.6.11. 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](#)。

### 1.6.12. 后续步骤

- [验证安装](#)。
- [自定义集群](#)。
- 如果需要，您可以[选择不使用远程健康报告](#)。
- 如果需要，您可以[删除云供应商凭证](#)。

## 1.7. 在 AWS 上安装私有集群

在 OpenShift Container Platform 版本 4.6 中，您可以在 Amazon Web Services (AWS) 上将私有集群安装到现有的 VPC 中。安装程序会置备所需基础架构的其余部分，您可以进一步定制这些基础架构。要自定义安装，请在安装集群前修改 `install-config.yaml` 文件中的参数。

### 1.7.1. 先决条件

- 查看有关 [OpenShift Container Platform 安装和更新](#) 流程的详细信息。
- [配置 AWS 帐户](#) 以托管集群。



#### 重要

如果您的计算机上存储有 AWS 配置集，则不要在使用多因素验证设备的同时使用您生成的临时会话令牌。在集群的整个生命周期中，集群会持续使用您的当前 AWS 凭证来创建 AWS 资源，因此您必须使用长期凭证。要生成适当的密钥，请参阅 AWS 文档中的[管理 IAM 用户的访问密钥](#)。您可在运行安装程序时提供密钥。

- 如果使用防火墙，则必须[将其配置为允许集群需要访问的站点](#)。
- 如果不允许系统管理身份和访问管理 (IAM)，集群管理员可以[手动创建和维护 IAM 凭证](#)。手动模式也可以用于云 IAM API 无法访问的环境中。

### 1.7.2. 私有集群

您可以部署不公开外部端点的私有 OpenShift Container Platform 集群。私有集群只能从内部网络访问，且无法在互联网中看到。

默认情况下，OpenShift Container Platform 被置备为使用可公开访问的 DNS 和端点。私有集群在部署集群时将 DNS、Ingress Controller 和 API 服务器设置为私有。这意味着，集群资源只能从您的内部网络访问，且不能在互联网中看到。

要部署私有集群，您必须使用符合您的要求的现有网络。您的集群资源可能会在网络中的其他集群间共享。

另外，您必须从可访问您置备的云的 API 服务、您置备的网络上的主机以及可以连接到互联网来获取安装介质的机器上部署私有集群。您可以使用符合这些访问要求的机器，并按照您的公司规定进行操作。例如，该机器可以是云网络中的堡垒主机，也可以是可通过 VPN 访问网络的机器。

### 1.7.2.1. AWS 中的私有集群

要在 Amazon Web Services (AWS) 上创建私有集群，您必须提供一个现有的私有 VPC 和子网来托管集群。安装程序还必须能够解析集群所需的 DNS 记录。安装程序将 Ingress Operator 和 API 服务器配置为只可以从私有网络访问。

集群仍然需要访问互联网来访问 AWS API。

安装私有集群时不需要或创建以下项目：

- 公共子网
- 支持公共入口的公共负载均衡器
- 与集群的 **baseDomain** 匹配的公共 Route 53 区域

安装程序会使用您指定的 **baseDomain** 来创建专用的 Route 53 区域以及集群所需的记录。集群被配置，以便 Operator 不会为集群创建公共记录，且所有集群机器都放置在您指定的私有子网中。

#### 1.7.2.1.1. 限制：

为私有集群添加公共功能的能力有限。

- 在安装后，您无法在不进行额外操作的情况下公开 Kubernetes API 端点。这些额外的操作包括为使用中的每个可用区在 VPC 中创建公共子网，创建公共负载均衡器，以及配置 control plane 安全组以便 6443 端口（Kubernetes API 端口）可以接受来自于互联网的网络流量。
- 如果使用公共服务类型负载均衡器，您必须在每个可用区中为公共子网添加 **kubernetes.io/cluster/<cluster-infra-id>: shared** 标签，以便 AWS 可使用它们来创建公共负载均衡器。

### 1.7.3. 关于使用自定义 VPC

在 OpenShift Container Platform 4.6 中，您可以在 Amazon Web Services (AWS) 的现有 Amazon Virtual Private Cloud (VPC) 中将集群部署到现有子网中。通过将 OpenShift Container Platform 部署到现有的 AWS VPC 中，您可能会避开新帐户中的限制，或者更容易地利用公司所设置的操作限制。如果您无法获得您自己创建 VPC 所需的基础架构创建权限，请使用这个安装选项。

因为安装程序无法了解您现有子网中还有哪些其他组件，所以无法选择子网 CIDR。您必须为安装集群的子网配置网络。

#### 1.7.3.1. 使用 VPC 的要求

安装程序不再创建以下组件：

- 互联网网关
- NAT 网关
- 子网
- 路由表
- VPCs
- VPC DHCP 选项

- VPC 端点



### 注意

安装程序要求您使用由云提供的 DNS 服务器。不支持使用自定义 DNS 服务器，并导致安装失败。

如果您使用自定义 VPC，您必须为安装程序和集群正确配置它及其子网。如需有关创建和管理 AWS VPC 的更多信息，请参阅 [AWS 文档中的 Amazon VPC 控制台向导配置和工作 VPC 和子网](#)。

安装程序无法：

- 细分供集群使用的网络范围。
- 为子网设置路由表。
- 设置 VPC 选项，如 DHCP。

在安装集群前，您必须完成这些任务。有关在 AWS VPC 中配置网络的更多信息，请参阅 [VPC 的 VPC 网络组件和路由表](#)。

您的 VPC 必须满足以下特征：

- VPC 不能使用 **kubernetes.io/cluster/.\*: owned** 标签。  
安装程序会修改子网以添加 **kubernetes.io/cluster/.\*: shared** 标签，因此您的子网必须至少有一个可用的空闲标签插槽。请参阅 AWS 文档中的 [标签限制](#) 部分，以确认安装程序可以为您指定的每个子网添加标签。
- 您必须在 VPC 中启用 **enableDnsSupport** 和 **enableDnsHostnames** 属性，以便集群可以使用附加到 VPC 的 Route 53 区来解析集群内部 DNS 记录。请参阅 AWS 文档中的 [您的 VPC 中的 DNS 支持](#) 部分。  
如果要使用自己的 Route 53 托管私有区，您必须在安装集群前将现有托管区与 VPC 关联。您可以使用 **install-config.yaml** 文件中的 **platform.aws.hostedZone** 字段定义托管区。
- 如果您使用具有公共访问权限的集群，您必须为每个集群使用的可用区创建一个公共和私有子网。每个可用区不能包含多于一个的公共子网和专用子网。

如果您在断开连接的环境中工作，您将无法访问 EC2 和 ELB 端点的公共 IP 地址。要解决这个问题，您必须创建一个 VPC 端点，并将其附加到集群使用的子网。端点应命名如下：

- **ec2.<region>.amazonaws.com**
- **elasticloadbalancing.<region>.amazonaws.com**
- **s3.<region>.amazonaws.com**

### 所需的 VPC 组件

您必须提供合适的 VPC 和子网，以便与您的机器通信。

组件	AWS 类型	描述
----	--------	----

组件	AWS 类型	描述	
VPC	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::VPC</b></li> <li>● <b>AWS::EC2::VPCEndpoint</b></li> </ul>	您必须提供一个公共 VPC 供集群使用。VPC 使用引用每个子网的路由表的端点，以改进与托管在 S3 中的 registry 的通信。	
公共子网	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::Subnet</b></li> <li>● <b>AWS::EC2::SubnetNetworkAclAssociation</b></li> </ul>	您的 VPC 必须有 1 到 3 个可用区的公共子网，并将其与适当的入口规则关联。	
互联网网关	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::InternetGateway</b></li> <li>● <b>AWS::EC2::VPCGatewayAttachment</b></li> <li>● <b>AWS::EC2::RouteTable</b></li> <li>● <b>AWS::EC2::Route</b></li> <li>● <b>AWS::EC2::SubnetRouteTableAssociation</b></li> <li>● <b>AWS::EC2::NatGateway</b></li> <li>● <b>AWS::EC2::EIP</b></li> </ul>	您必须有一个公共互联网网关，以及附加到 VPC 的公共路由。在提供的模板中，每个公共子网都有一个具有 EIP 地址的 NAT 网关。这些 NAT 网关允许集群资源（如专用子网实例）访问互联网，而有些受限网络或代理场景则不需要它们。	
网络访问控制	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::NetworkAcl</b></li> <li>● <b>AWS::EC2::NetworkAclEntry</b></li> </ul>	您必须允许 VPC 访问下列端口：	
		<b>端口</b>	<b>原因</b>
		<b>80</b>	入站 HTTP 流量
		<b>443</b>	入站 HTTPS 流量
		<b>22</b>	入站 SSH 流量
		<b>1024 - 65535</b>	入站临时流量
	<b>0 - 65535</b>	出站临时流量	
专用子网	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::Subnet</b></li> <li>● <b>AWS::EC2::RouteTable</b></li> <li>● <b>AWS::EC2::SubnetRouteTableAssociation</b></li> </ul>	您的 VPC 可以具有私有子网。提供的 CloudFormation 模板可为 1 到 3 个可用区创建专用子网。如果您使用专用子网，必须为其提供适当的路由和表。	

### 1.7.3.2. VPC 验证

要确保您提供的子网适合您的环境，安装程序会确认以下信息：

- 您指定的所有子网都存在。
- 您提供了私有子网。
- 子网 CIDR 属于您指定的机器 CIDR。
- 您为每个可用区提供子网。每个可用区不包含多于一个的公共子网和私有子网。如果您使用私有集群，为每个可用区只提供一个私有子网。否则，为每个可用区提供一个公共和私有子网。
- 您可以为每个私有子网可用区提供一个公共子网。机器不会在没有为其提供私有子网的可用区中置备。

如果您销毁使用现有 VPC 的集群，VPC 不会被删除。从 VPC 中删除 OpenShift Container Platform 集群时，`kubernetes.io/cluster/.*: shared` 标签会从使用它的子网中删除。

### 1.7.3.3. 权限划分

从 OpenShift Container Platform 4.3 开始，您不需要安装程序置备的基础架构集群部署所需的所有权限。这与您所在机构可能已有的权限划分类似：不同的人可以在您的云中创建不同的资源。例如，您可以创建针对于特定应用程序的对象，如实例、存储桶和负载均衡器，但不能创建与网络相关的组件，如 VPC、子网或入站规则。

您在创建集群时使用的 AWS 凭证不需要 VPC 和 VPC 中的核心网络组件（如子网、路由表、互联网网关、NAT 和 VPN）所需的网络权限。您仍然需要获取集群中的机器需要的应用程序资源的权限，如 ELB、安全组、S3 存储桶和节点。

### 1.7.3.4. 集群间隔离

如果您将 OpenShift Container Platform 部署到现有网络中，集群服务的隔离将在以下方面减少：

- 您可以在同一 VPC 中安装多个 OpenShift Container Platform 集群。
- 整个网络允许 ICMP 入站流量。
- 整个网络都允许 TCP 22 入站流量 (SSH)。
- 整个网络都允许 control plane TCP 6443 入站流量 (Kubernetes API)。
- 整个网络都允许 control plane TCP 22623 入站流量 (MCS)。

## 1.7.4. 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。

### 1.7.5. 生成 SSH 私钥并将其添加到代理中

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



## 注意

在生产环境中，您需要进行灾难恢复和调试。

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。

**1.7.6. 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**

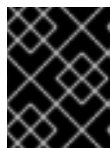
- 运行 Linux 或 macOS 的计算机，本地磁盘空间为 500 MB

**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。

### 1.7.7. 手动创建安装配置文件

对于只能从内部网络访问且不能在互联网中看到的私有 OpenShift Container Platform 集群安装，您必须手动生成安装配置文件。

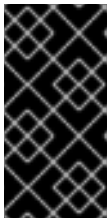
#### 先决条件

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

#### 流程

1. 创建用来存储您所需的安装资产的安装目录：

```
$ mkdir <installation_directory>
```



#### 重要

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

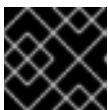
2. 自定义以下 **install-config.yaml** 文件模板，并将它保存到 **<installation\_directory>** 中。



#### 注意

此配置文件必须命名为 **install-config.yaml**。

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



#### 重要

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

#### 1.7.7.1. 安装配置参数

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



#### 注意

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



#### 重要

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

## 1.7.7.1.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 1.18. 所需的参数

参数	描述	值
<b>apiVersion</b>	<b>install-config.yaml</b> 内容的 API 版本。当前版本是 <b>v1</b> 。安装程序还可能支持旧的 API 版本。	字符串
<b>baseDomain</b>	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 <b>baseDomain</b> 和 <b>metadata.name</b> 参数值的组合，其格式为 <b>&lt;metadata.name&gt;.&lt;baseDomain&gt;</b> 。	完全限定域名或子域名，如 <b>example.com</b> 。
<b>metadata</b>	Kubernetes 资源 <b>ObjectMeta</b> ，其中只消耗 <b>name</b> 参数。	对象
<b>metadata.name</b>	集群的名称。集群的 DNS 记录是 <b>{{.metadata.name}}</b> . <b>{{.baseDomain}}</b> 的子域。	小写字母,连字符(-)和句点(.)的字符串，如 <b>dev</b> 。
<b>platform</b>	执行安装的具体平台配置： <b>aws</b> 、 <b>baremetal</b> 、 <b>azure</b> 、 <b>openstack</b> 、 <b>ovirt</b> 、 <b>vsphere</b> 。有关 <b>platform</b> 。<platform> 参数的额外信息，请参考下表来了解您的具体平台。	对象
<b>pullSecret</b>	从 Red Hat OpenShift Cluster Manager 获取 <b>pull secret</b> ，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>

## 1.7.7.1.2. 网络配置参数

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

只支持 IPv4 地址。

表 1.19. 网络参数

参数	描述	值
<b>networking</b>	集群网络的配置。	对象  <b>注意</b> 您不能在安装后修改 <b>networking</b> 对象指定的参数。
<b>networking.networkType</b>	要安装的集群网络供应商 Container Network Interface (CNI) 插件。	<b>OpenShiftSDN</b> 或 <b>OVNKubernetes</b> 。默认值为 <b>OpenShiftSDN</b> 。
<b>networking.clusterNetwork</b>	pod 的 IP 地址块。  默认值为 <b>10.128.0.0/14</b> ，主机前缀为 <b>/23</b> 。  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>
<b>networking.clusterNetwork.cidr</b>	使用 <b>networking.clusterNetwork</b> 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 <b>0</b> 到 <b>32</b> 之间。
<b>networking.clusterNetwork.hostPrefix</b>	分配给每个单独节点的子网前缀长度。 例如，如果 <b>hostPrefix</b> 设为 <b>23</b> ，则每个节点从所给的 <b>cidr</b> 中分配一个 <b>/23</b> 子网。 <b>hostPrefix</b> 值 <b>23</b> 提供 $510 (2^{(32 - 23)} - 2)$ 个 pod IP 地址。	子网前缀。  默认值为 <b>23</b> 。
<b>networking.serviceNetwork</b>	服务的 IP 地址块。默认值为 <b>172.30.0.0/16</b> 。  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>

参数	描述	值
<b>networking.machineNetwork</b>	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>
<b>networking.machineNetwork.cidr</b>	使用 <b>networking.machineNetwork</b> 时需要。IP 地址块。libvirt 以外的所有平台的默认值为 <b>10.0.0.0/16</b> 。对于 libvirt，默认值为 <b>192.168.126.0/24</b> 。	CIDR 表示法中的 IP 网络块。  例如： <b>10.0.0.0/16</b> 。   <b>注意</b>  将 <b>networking.machineNetwork</b> 设置为与首选 NIC 所在的 CIDR 匹配。

### 1.7.7.1.3. 可选配置参数

下表描述了可选安装配置参数：

表 1.20. 可选参数

参数	描述	值
<b>additionalTrustBundle</b>	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
<b>compute</b>	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。
<b>compute.architecture</b>	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 <b>amd64</b> （默认值）。	字符串
<b>compute.hyperthreading</b>	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。   <b>重要</b>  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。	<b>Enabled</b> 或 <b>Disabled</b>

参数	描述	值
<b>compute.name</b>	使用 <b>compute</b> 时需要此值。机器池的名称。	<b>worker</b>
<b>compute.platform</b>	使用 <b>compute</b> 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 <b>controlPlane.platform</b> 参数值匹配。	<b>aws、azure、gcp、openstack、o virt、vsphere 或 {}</b>
<b>compute.replicas</b>	要置备的计算机器数量，也称为 worker 机器。	大于或等于 <b>2</b> 的正整数。默认值为 <b>3</b> 。
<b>controlPlane</b>	组成 control plane 的机器的配置。	<b>MachinePool</b> 对象的数组。详情请查看以下"Machine-pool"表。
<b>controlPlane.architecture</b>	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 <b>amd64</b> （默认值）。	字符串
<b>controlPlane.hyperthread reading</b>	<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><b>重要</b></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>	<b>Enabled 或 Disabled</b>
<b>controlPlane.name</b>	使用 <b>controlPlane</b> 时需要。机器池的名称。	<b>master</b>
<b>controlPlane.platform</b>	使用 <b>controlPlane</b> 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 <b>compute.platform</b> 参数值匹配。	<b>aws、azure、gcp、openstack、o virt、vsphere 或 {}</b>
<b>controlPlane.replicas</b>	要置备的 control plane 机器数量。	唯一支持的值是 <b>3</b> ，它是默认值。

参数	描述	值
<b>credentialsMode</b>	<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><b>注意</b></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>	<b>Mint、Passthrough、Manual</b> 或空字符串("")。
<b>fips</b>	<p>启用或禁用 FIPS 模式。默认为 <b>false</b> (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><b>重要</b></p> <p>只有在 <b>x86_64</b> 架构中的 OpenShift Container Platform 部署支持 FIPS 验证的/Modules in Process 加密库。</p>  <p><b>注意</b></p> <p>如果使用 Azure File 存储，则无法启用 FIPS 模式。</p>	<b>false</b> 或 <b>true</b>
<b>imageContentSources</b>	release-image 内容的源和仓库。	对象数组。包括一个 <b>source</b> 以及可选的 <b>mirrors</b> ，如下表所示。
<b>imageContentSources.source</b>	使用 <b>imageContentSources</b> 时需要。指定用户在镜像拉取规格中引用的仓库。	字符串
<b>imageContentSources.mirrors</b>	指定可能还包含同一镜像的一个或多个仓库。	字符串数组



参数	描述	值
<b>publish</b>	如何发布或公开集群的面向用户的端点，如 Kubernetes API、OpenShift 路由。	<b>Internal</b> 或 <b>External</b> 。把 <b>publish</b> 设置为 <b>Internal</b> 以部署一个私有集群，它不能被互联网访问。默认值为 <b>External</b> 。
<b>sshKey</b>	用于验证集群机器访问的 SSH 密钥或密钥。   <div style="margin-left: 20px;"> <p><b>注意</b></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 <b>ssh-agent</b> 进程使用的 SSH 密钥。</p> </div>	一个或多个密钥。例如：  <pre>sshKey:   &lt;key1&gt;   &lt;key2&gt;   &lt;key3&gt;</pre>

#### 1.7.7.1.4. 可选的 AWS 配置参数

下表描述了可选的 AWS 配置参数：

表 1.21. 可选的 AWS 参数

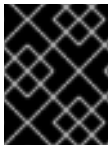
参数	描述	值
<b>compute.platform.aws.amiid</b>	用于为集群引导计算机器的 AWS AMI。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<b>compute.platform.aws.rootVolume.iops</b>	为根卷保留的每秒输入/输出操作 (IOPS) 数。	整数，如 <b>4000</b> 。
<b>compute.platform.aws.rootVolume.size</b>	以 GiB 为单位的根卷大小。	整数，如 <b>500</b> 。
<b>compute.platform.aws.rootVolume.type</b>	根卷的类型。	有效的 <a href="#">AWS EBS 卷类型</a> ，如 <b>io1</b> 。
<b>compute.platform.aws.rootVolume.kmsKeyARN</b>	KMS 密钥的 Amazon 资源名称（密钥 ARN）。这是使用特定 KMS 密钥加密 worker 节点的操作系统卷。	有效的 <a href="#">密钥 ID 或密钥 ARN</a> 。

参数	描述	值
<code>compute.platform.aws.type</code>	计算机器的 EC2 实例类型。	有效的 <a href="#">AWS 实例类型</a> ，如 <code>c5.9xlarge</code> 。
<code>compute.platform.aws.zones</code>	安装程序在其中为计算机器池创建机器的可用区。如果您提供自己的 VPC，则必须在那个可用域中提供一个子网。	有效 AWS 可用区的列表，如 <code>us-east-1c</code> ，以 <a href="#">YAML 序列</a> 表示。
<code>compute.aws.region</code>	安装程序在其中创建计算资源的 AWS 区域。	任何有效的 <a href="#">AWS 区域</a> ，如 <code>us-east-1</code> 。
<code>controlPlane.platform.aws.amiID</code>	用于为集群引导 control plane 机器的 AWS AMI。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<code>controlPlane.platform.aws.rootVolume.kmsKeyARN</code>	KMS 密钥的 Amazon 资源名称（密钥 ARN）。这需要使用特定的 KMS 密钥加密 control plane 节点的操作系统卷。	有效的 <a href="#">密钥 ID 和密钥 ARN</a> 。
<code>controlPlane.platform.aws.type</code>	control plane 机器的 EC2 实例类型。	有效的 <a href="#">AWS 实例类型</a> ，如 <code>c5.9xlarge</code> 。
<code>controlPlane.platform.aws.zones</code>	安装程序在其中为 control plane 机器池创建机器的可用区。	有效 AWS 可用区的列表，如 <code>us-east-1c</code> ，以 <a href="#">YAML 序列</a> 表示。
<code>controlPlane.aws.region</code>	安装程序在其中创建 control plane 资源的 AWS 区域。	有效的 <a href="#">AWS 区域</a> ，如 <code>us-east-1</code> 。
<code>platform.aws.amiID</code>	用于为集群引导所有机器的 AWS AMI。如果设置，AMI 必须属于与集群相同的区域。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<code>platform.aws.serviceEndpoints.name</code>	AWS 服务端点名称。只有在必须使用替代 AWS 端点（如 FIPS）时，才需要自定义端点。可以为 EC2、S3、IAM、Elastic Load Balancing、Tagging、Route 53 和 STS AWS 服务指定自定义 API 端点。	有效的 <a href="#">AWS 服务端点名称</a> 。
<code>platform.aws.serviceEndpoints.url</code>	AWS 服务端点 URL。URL 必须使用 <a href="#">https</a> 协议，主机必须信任该证书。	有效的 <a href="#">AWS 服务端点 URL</a> 。

参数	描述	值
<b>platform.aws.userTags</b>	键与值的映射，安装程序将其作为标签添加到它所创建的所有资源。	任何有效的 YAML 映射，如 <b>&lt;key&gt;: &lt;value&gt;</b> 格式的键值对。如需有关 AWS 标签的更多信息，请参阅 AWS 文档中的 <a href="#">标记您的 Amazon EC2 资源</a> 。
<b>platform.aws.subnets</b>	如果您提供 VPC，而不是让安装程序为您创建 VPC，请指定要使用的集群子网。子网必须是您指定的同一 <b>machineNetwork[].cidr</b> 范围的一部分。对于标准集群，为每个可用区指定一个公共和私有子网。对于私有集群，为每个可用区指定一个私有子网。	有效的子网 ID。

### 1.7.7.2. AWS 的自定义 install-config.yaml 文件示例

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



#### 重要

此示例 YAML 文件仅供参考。您必须使用安装程序来获取 **install-config.yaml** 文件，并且修改该文件。

```

apiVersion: v1
baseDomain: example.com 1
credentialsMode: Mint 2
controlPlane: 3 4
  hyperthreading: Enabled 5
  name: master
  platform:
    aws:
      zones:
      - us-west-2a
      - us-west-2b
    rootVolume:
      iops: 4000
      size: 500
      type: io1 6
      type: m5.xlarge
    replicas: 3
  compute: 7
  - hyperthreading: Enabled 8
  name: worker
  platform:
    aws:
      rootVolume:
        iops: 2000

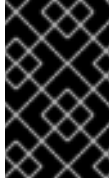
```

```

    size: 500
    type: io1 9
    type: c5.4xlarge
    zones:
    - us-west-2c
  replicas: 3
  metadata:
    name: test-cluster 10
  networking:
    clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
    machineNetwork:
    - cidr: 10.0.0.0/16
    networkType: OpenShiftSDN
    serviceNetwork:
    - 172.30.0.0/16
  platform:
    aws:
      region: us-west-2 11
      userTags:
        adminContact: jdoe
        costCenter: 7536
      subnets: 12
      - subnet-1
      - subnet-2
      - subnet-3
      amiID: ami-96c6f8f7 13
      serviceEndpoints: 14
      - name: ec2
        url: https://vpce-id.ec2.us-west-2.vpce.amazonaws.com
      hostedZone: Z3URY6TWQ91KVV 15
    fips: false 16
    sshKey: ssh-ed25519 AAAA... 17
    publish: Internal 18
    pullSecret: '{"auths": ...}' 19

```

- 1 10 11 19** 必需。安装程序会提示您输入这个值。
- 2** 可选：添加此参数来强制 Cloud Credential Operator (CCO) 使用指定的模式，而不是让 CCO 动态尝试决定凭证的功能。如需有关 CCO 模式的详情，请参阅 *Red Hat Operator* 参考内容中的 *Cloud Credential Operator* 条目。
- 3 7** 如果没有提供这些参数和值，安装程序会提供默认值。
- 4** **controlPlane** 部分是一个单个映射，但 **compute** 部分是一系列映射。为满足不同数据结构的要求，**compute** 部分的第一行必须以连字符 - 开头，**controlPlane** 部分的第一行则不以连字符开头。只使用一个 control plane 池。
- 5 8** 是否要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。您可以通过将参数值设为 **Disabled** 来禁用。如果您在某些集群机器上禁用并发多线程，则必须在所有集群机器上禁用。

**重要**

如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。如果您对机器禁用并发多线程，请使用较大的实例类型，如 **m4.2xlarge** 或 **m5.2xlarge**。

- 6** **9** 要为 etcd 配置更快的存储，特别是对于较大的集群，请将存储类型设置为 **io1**，并将 **iops** 设为 **2000**。
- 12** 如果您提供自己的 VPC，为集群使用的每个可用区指定子网。
- 13** 用于为集群引导机器的 AMI ID。如果设置，AMI 必须属于与集群相同的区域。
- 14** AWS 服务端点。在安装到未知 AWS 区域时，需要自定义端点。端点 URL 必须使用 **https** 协议，主机必须信任该证书。
- 15** 您现有 Route 53 私有托管区的 ID。提供现有的托管区需要您提供自己的 VPC，托管区已在安装集群前与 VPC 关联。如果未定义，安装程序会创建一个新的托管区。
- 16** 是否启用或禁用 FIPS 模式。默认情况下不启用 FIPS 模式。如果启用了 FIPS 模式，运行 OpenShift Container Platform 的 Red Hat Enterprise Linux CoreOS (RHCOS) 机器会绕过默认的 Kubernetes 加密套件，并使用由 RHCOS 提供的加密模块。

**重要**

只有在 **x86\_64** 架构中的 OpenShift Container Platform 部署支持 FIPS 验证的 `/Modules in Process` 加密库。

- 17** 您可以选择提供您用来访问集群中机器的 **sshKey** 值。

**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。

- 18** 如何发布集群的面向用户的端点。把 **publish** 设置为 **Internal** 以部署一个私有集群，它不能被互联网访问。默认值为 **External**。

### 1.7.7.3. 在安装过程中配置集群范围代理

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

#### 先决条件

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



## 注意

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

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

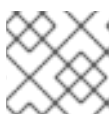
- 如果您的集群位于 AWS 上, 请将 **ec2.<region>.amazonaws.com**、**elasticloadbalancing.<region>.amazonaws.com** 和 **s3.<region>.amazonaws.com** 端点添加到 VPC 端点。需要这些端点才能完成节点到 AWS EC2 API 的请求。由于代理在容器级别而不是节点级别工作, 因此您必须通过 AWS 专用网络将这些请求路由到 AWS EC2 API。在代理服务器中的允许列表中添加 EC2 API 的公共 IP 地址是不够的。

## 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** 的配置映射来保存额外的 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, 将为 **trustedCA** 参数指定的内容与 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**。

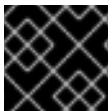


### 注意

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

## 1.7.8. 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

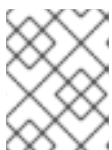
- 配置托管集群的云平台的帐户。
- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

### 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
- 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 Ignition 配置文件生成后的 12 小时内使用它们，因为 24 小时的证书会在集群安装后的 16 小时到 22 小时进行轮转。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因为执行了证书更新而导致安装失败的问题。



### 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。

## 1.7.9. 通过下载二进制文件安装 OpenShift CLI

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



### 重要

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

### 1.7.9.1. 在 Linux 上安装 OpenShift CLI

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

#### 流程

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

```
$ tar xvfz <file>
```

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

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：



```
$ oc <command>
```

### 1.7.9.2. 在 Windows 上安装 OpenShift CLI

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

#### 流程

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

```
C:\> path
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 1.7.9.3. 在 macOS 上安装 OpenShift CLI

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

#### 流程

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

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 1.7.10. 使用 CLI 登录到集群

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

#### 先决条件

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

#### 流程

1. 导出 **kubeadmin** 凭证：

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

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

2. 使用导出的配置，验证能否成功运行 **oc** 命令：

```
$ oc whoami
```

#### 输出示例

```
system:admin
```

### 1.7.11. 使用 Web 控制台登录到集群

**kubeadmin** 用户默认在 OpenShift Container Platform 安装后存在。您可以使用 OpenShift Container Platform Web 控制台以 **kubeadmin** 用户身份登录集群。

#### 先决条件

- 有访问安装主机的访问权限。
- 您完成了集群安装，所有集群 Operator 都可用。

#### 流程

1. 从安装主机上的 **kubeadmin -password** 文件中获取 kubeadmin 用户的密码：

```
$ cat <installation_directory>/auth/kubeadmin-password
```

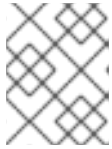


#### 注意

另外，您还可以从安装主机上的 **<installation\_directory>/openshift\_install.log** 日志文件获取 **kubeadmin** 密码。

2. 列出 OpenShift Container Platform Web 控制台路由：

```
$ oc get routes -n openshift-console | grep 'console-openshift'
```



### 注意

另外，您还可以从安装主机上的 `<installation_directory>/openshift_install.log` 日志文件获取 OpenShift Container Platform 路由。

### 输出示例

```
console console-openshift-console.apps.<cluster_name>.<base_domain> console
https reencrypt/Redirect None
```

3. 在 Web 浏览器中导航到上一命令输出中包括的路由，以 **kubeadmin** 用户身份登录。

### 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](#)。

## 1.7.12. 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](#)。

## 1.7.13. 后续步骤

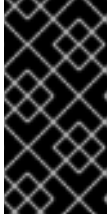
- [验证安装](#)。
- [自定义集群](#)。
- 如果需要，您可以[选择不使用远程健康报告](#)。
- 如果需要，您可以[删除云供应商凭证](#)。

## 1.8. 在 AWS 上将集群安装到一个政府区域

在 OpenShift Container Platform 版本 4.6 中，您可以在 Amazon Web Services (AWS) 上将集群安装到一个政府区域。要配置政府区域，请在安装集群前修改 `install-config.yaml` 文件中的参数。

### 1.8.1. 先决条件

- 查看有关 [OpenShift Container Platform 安装和更新](#) 流程的详细信息。
- [配置 AWS 帐户](#) 以托管集群。



### 重要

如果您的计算机上存储有 AWS 配置集，则不要在使用多因素验证设备的同时使用您生成的临时会话令牌。在集群的整个生命周期中，集群会持续使用您的当前 AWS 凭证来创建 AWS 资源，因此您必须使用长期凭证。要生成适当的密钥，请参阅 AWS 文档中的[管理 IAM 用户的访问密钥](#)。您可在运行安装程序时提供密钥。

- 如果使用防火墙，则必须将其配置为允许集群需要访问的站点。
- 如果不允许系统管理身份和访问管理（IAM），集群管理员可以[手动创建和维护 IAM 凭证](#)。手动模式也可以用于云 IAM API 无法访问的环境中。

## 1.8.2. AWS 政府区域

OpenShift Container Platform 支持将集群部署到 [AWS GovCloud\(US\)](#) 区域。AWS GovCloud 是为需要运行敏感负载的美国政府机构、企业、企业和其他美国客户特别设计的。

这些区域尚未发布 Red Hat Enterprise Linux CoreOS (RHCOS) Amazon Machine Images (AMI)，因此您必须上传属于该区的自定义 AMI。

支持以下 AWS GovCloud 分区：

- **us-gov-west-1**
- **us-gov-east-1**

由于那些区域的 RHCOS AMI 不是由红帽提供的，所以必须在 **install-config.yaml** 文件中手动配置 AWS GovCloud 区域和自定义 AMI。

## 1.8.3. 私有集群

您可以部署不公开外部端点的私有 OpenShift Container Platform 集群。私有集群只能从内部网络访问，且无法在互联网中看到。



### 注意

AWS GovCloud 的 Route 53 不支持公共区。因此，如果集群部署到 AWS 政府区域，集群必须是私有的。

默认情况下，OpenShift Container Platform 被置备为使用可公开访问的 DNS 和端点。私有集群在部署集群时将 DNS、Ingress Controller 和 API 服务器设置为私有。这意味着，集群资源只能从您的内部网络访问，且不能在互联网中看到。

要部署私有集群，您必须使用符合您的要求的现有网络。您的集群资源可能会在网络中的其他集群间共享。

另外，您必须从可访问您置备的云的 API 服务、您置备的网络上的主机以及可以连接到互联网来获取安装介质的机器上部署私有集群。您可以使用符合这些访问要求的机器，并按照您的公司规定进行操作。例如，该机器可以是云网络中的堡垒主机，也可以是可通过 VPN 访问网络的机器。

### 1.8.3.1. AWS 中的私有集群

要在 Amazon Web Services (AWS) 上创建私有集群，您必须提供一个现有的私有 VPC 和子网来托管集群。安装程序还必须能够解析集群所需的 DNS 记录。安装程序将 Ingress Operator 和 API 服务器配置为只可以从私有网络访问。

集群仍然需要访问互联网来访问 AWS API。

安装私有集群时不需要或创建以下项目：

- 公共子网
- 支持公共入口的公共负载均衡器
- 与集群的 **baseDomain** 匹配的公共 Route 53 区域

安装程序会使用您指定的 **baseDomain** 来创建专用的 Route 53 区域以及集群所需的记录。集群被配置，以便 Operator 不会为集群创建公共记录，且所有集群机器都放置在您指定的私有子网中。

#### 1.8.3.1.1. 限制：

为私有集群添加公共功能的能力有限。

- 在安装后，您无法在不进行额外操作的情况下公开 Kubernetes API 端点。这些额外的操作包括为使用中的每个可用区在 VPC 中创建公共子网，创建公共负载均衡器，以及配置 control plane 安全组以便 6443 端口（Kubernetes API 端口）可以接受来自于互联网的网络流量。
- 如果使用公共服务类型负载均衡器，您必须在每个可用区中为公共子网添加 **kubernetes.io/cluster/<cluster-infra-id>: shared** 标签，以便 AWS 可使用它们来创建公共负载均衡器。

### 1.8.4. 关于使用自定义 VPC

在 OpenShift Container Platform 4.6 中，您可以在 Amazon Web Services (AWS) 的现有 Amazon Virtual Private Cloud (VPC) 中将集群部署到现有子网中。通过将 OpenShift Container Platform 部署到现有的 AWS VPC 中，您可能会避开新帐户中的限制，或者更容易地利用公司所设置的操作限制。如果您无法获得您自己创建 VPC 所需的基础架构创建权限，请使用这个安装选项。

因为安装程序无法了解您现有子网中还有哪些其他组件，所以无法选择子网 CIDR。您必须为安装集群的子网配置网络。

#### 1.8.4.1. 使用 VPC 的要求

安装程序不再创建以下组件：

- 互联网网关
- NAT 网关
- 子网
- 路由表
- VPCs
- VPC DHCP 选项
- VPC 端点



## 注意

安装程序要求您使用由云提供的 DNS 服务器。不支持使用自定义 DNS 服务器，并导致安装失败。

如果您使用自定义 VPC，您必须为安装程序和集群正确配置它及其子网。如需有关创建和管理 AWS VPC 的更多信息，请参阅 [AWS 文档中的 Amazon VPC 控制台向导配置和工作 VPC 和子网](#)。

安装程序无法：

- 细分供集群使用的网络范围。
- 为子网设置路由表。
- 设置 VPC 选项，如 DHCP。

在安装集群前，您必须完成这些任务。有关在 AWS VPC 中配置网络的更多信息，请参阅 [VPC 的 VPC 网络组件和路由表](#)。

您的 VPC 必须满足以下特征：

- VPC 不能使用 **kubernetes.io/cluster/.\*: owned** 标签。  
安装程序会修改子网以添加 **kubernetes.io/cluster/.\*: shared** 标签，因此您的子网必须至少有一个可用的空闲标签插槽。请参阅 AWS 文档中的 [标签限制](#) 部分，以确认安装程序可以为您指定的每个子网添加标签。
- 您必须在 VPC 中启用 **enableDnsSupport** 和 **enableDnsHostnames** 属性，以便集群可以使用附加到 VPC 的 Route 53 区来解析集群内部 DNS 记录。请参阅 AWS 文档中的 [您的 VPC 中的 DNS 支持](#) 部分。  
如果要使用自己的 Route 53 托管私有区，您必须在安装集群前将现有托管区与 VPC 关联。您可以使用 **install-config.yaml** 文件中的 **platform.aws.hostedZone** 字段定义托管区。
- 如果您使用具有公共访问权限的集群，您必须为每个集群使用的可用区创建一个公共和私有子网。每个可用区不能包含多于一个的公共子网和专用子网。

如果您在断开连接的环境中工作，您将无法访问 EC2 和 ELB 端点的公共 IP 地址。要解决这个问题，您必须创建一个 VPC 端点，并将其附加到集群使用的子网。端点应命名如下：

- **ec2.<region>.amazonaws.com**
- **elasticloadbalancing.<region>.amazonaws.com**
- **s3.<region>.amazonaws.com**

## 所需的 VPC 组件

您必须提供合适的 VPC 和子网，以便与您的机器通信。

组件	AWS 类型	描述
VPC	<ul style="list-style-type: none"> <li>• <b>AWS::EC2::VPC</b></li> <li>• <b>AWS::EC2::VPCEndpoint</b></li> </ul>	您必须提供一个公共 VPC 供集群使用。VPC 使用引用每个子网的路由表的端点，以改进与托管在 S3 中的 registry 的通信。

组件	AWS 类型	描述	
公共子网	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::Subnet</b></li> <li>● <b>AWS::EC2::SubnetNetworkAclAssociation</b></li> </ul>	您的 VPC 必须有 1 到 3 个可用区的公共子网，并将其与适当的入口规则关联。	
互联网网关	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::InternetGateway</b></li> <li>● <b>AWS::EC2::VPCGatewayAttachment</b></li> <li>● <b>AWS::EC2::RouteTable</b></li> <li>● <b>AWS::EC2::Route</b></li> <li>● <b>AWS::EC2::SubnetRouteTableAssociation</b></li> <li>● <b>AWS::EC2::NatGateway</b></li> <li>● <b>AWS::EC2::EIP</b></li> </ul>	您必须有一个公共互联网网关，以及附加到 VPC 的公共路由。在提供的模板中，每个公共子网都有一个具有 EIP 地址的 NAT 网关。这些 NAT 网关允许集群资源（如专用子网实例）访问互联网，而有些受限网络或代理场景则不需要它们。	
网络访问控制	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::NetworkAcl</b></li> <li>● <b>AWS::EC2::NetworkAclEntry</b></li> </ul>	您必须允许 VPC 访问下列端口：	
		<b>端口</b>	<b>原因</b>
		<b>80</b>	入站 HTTP 流量
		<b>443</b>	入站 HTTPS 流量
		<b>22</b>	入站 SSH 流量
		<b>1024 - 65535</b>	入站临时流量
<b>0 - 65535</b>	出站临时流量		
专用子网	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::Subnet</b></li> <li>● <b>AWS::EC2::RouteTable</b></li> <li>● <b>AWS::EC2::SubnetRouteTableAssociation</b></li> </ul>	您的 VPC 可以具有私有子网。提供的 CloudFormation 模板可为 1 到 3 个可用区创建专用子网。如果您使用专用子网，必须为其提供适当的路由和表。	

#### 1.8.4.2. VPC 验证

要确保您提供的子网适合您的环境，安装程序会确认以下信息：

- 您指定的所有子网都存在。

- 您提供了私有子网。
- 子网 CIDR 属于您指定的机器 CIDR。
- 您为每个可用区提供子网。每个可用区不包含多于一个的公共子网和私有子网。如果您使用私有集群，为每个可用区只提供一个私有子网。否则，为每个可用区提供一个公共和私有子网。
- 您可以为每个私有子网可用区提供一个公共子网。机器不会在没有为其提供私有子网的可用区中置备。

如果您销毁使用现有 VPC 的集群，VPC 不会被删除。从 VPC 中删除 OpenShift Container Platform 集群时，`kubernetes.io/cluster/.*: shared` 标签会从使用它的子网中删除。

#### 1.8.4.3. 权限划分

从 OpenShift Container Platform 4.3 开始，您不需要安装程序置备的基础架构集群部署所需的所有权限。这与您所在机构可能已有的权限划分类似：不同的人可以在您的云中创建不同的资源。例如，您可以创建针对于特定应用程序的对象，如实例、存储桶和负载均衡器，但不能创建与网络相关的组件，如 VPC、子网或入站规则。

您在创建集群时使用的 AWS 凭证不需要 VPC 和 VPC 中的核心网络组件（如子网、路由表、互联网网关、NAT 和 VPN）所需的网络权限。您仍然需要获取集群中的机器需要的应用程序资源的权限，如 ELB、安全组、S3 存储桶和节点。

#### 1.8.4.4. 集群间隔离

如果您将 OpenShift Container Platform 部署到现有网络中，集群服务的隔离将在以下方面减少：

- 您可以在同一 VPC 中安装多个 OpenShift Container Platform 集群。
- 整个网络允许 ICMP 入站流量。
- 整个网络都允许 TCP 22 入站流量 (SSH)。
- 整个网络都允许 control plane TCP 6443 入站流量 (Kubernetes API)。
- 整个网络都允许 control plane TCP 22623 入站流量 (MCS)。

#### 1.8.5. 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。

### 1.8.6. 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> ❶
```

**输出示例**

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

- ❶ 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。

**1.8.7. 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。

### 1.8.8. 手动创建安装配置文件

在 Amazon Web Services (AWS) 上安装 OpenShift Container Platform 时，进入需要自定义 Red Hat Enterprise Linux CoreOS (RHCOS) AMI 的区域时，您必须手动生成安装配置文件。

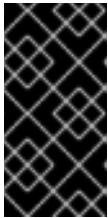
先决条件

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

流程

1. 创建用来存储您所需的安装资产的安装目录：

```
$ mkdir <installation_directory>
```



#### 重要

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

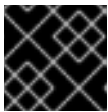
2. 自定义以下 **install-config.yaml** 文件模板，并将它保存到 **<installation\_directory>** 中。



#### 注意

此配置文件必须命名为 **install-config.yaml**。

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



#### 重要

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

#### 1.8.8.1. 安装配置参数

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



#### 注意

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



#### 重要

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

## 1.8.8.1.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 1.22. 所需的参数

参数	描述	值
<b>apiVersion</b>	<b>install-config.yaml</b> 内容的 API 版本。当前版本是 <b>v1</b> 。安装程序还可能支持旧的 API 版本。	字符串
<b>baseDomain</b>	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 <b>baseDomain</b> 和 <b>metadata.name</b> 参数值的组合，其格式为 <b>&lt;metadata.name&gt;.&lt;baseDomain&gt;</b> 。	完全限定域名或子域名，如 <b>example.com</b> 。
<b>metadata</b>	Kubernetes 资源 <b>ObjectMeta</b> ，其中只消耗 <b>name</b> 参数。	对象
<b>metadata.name</b>	集群的名称。集群的 DNS 记录是 <b>{{.metadata.name}}</b> . <b>{{.baseDomain}}</b> 的子域。	小写字母,连字符(-)和句点(.)的字符串，如 <b>dev</b> 。
<b>platform</b>	执行安装的具体平台配置： <b>aws</b> 、 <b>baremetal</b> 、 <b>azure</b> 、 <b>openstack</b> 、 <b>ovirt</b> 、 <b>vsphere</b> 。有关 <b>platform</b> 。 <b>&lt;platform&gt;</b> 参数的额外信息，请参考下表来了解您的具体平台。	对象
<b>pullSecret</b>	从 <a href="#">Red Hat OpenShift Cluster Manager</a> 获取 <b>pull secret</b> ，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>

## 1.8.8.1.2. 网络配置参数

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

只支持 IPv4 地址。

表 1.23. 网络参数

参数	描述	值
<b>networking</b>	集群网络的配置。	对象  <b>注意</b> 您不能在安装后修改 <b>networking</b> 对象指定的参数。
<b>networking.networkType</b>	要安装的集群网络供应商 Container Network Interface (CNI) 插件。	<b>OpenShiftSDN</b> 或 <b>OVNKubernetes</b> 。默认值为 <b>OpenShiftSDN</b> 。
<b>networking.clusterNetwork</b>	pod 的 IP 地址块。  默认值为 <b>10.128.0.0/14</b> ，主机前缀为 <b>/23</b> 。  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>
<b>networking.clusterNetwork.cidr</b>	使用 <b>networking.clusterNetwork</b> 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 <b>0</b> 到 <b>32</b> 之间。
<b>networking.clusterNetwork.hostPrefix</b>	分配给每个单独节点的子网前缀长度。 例如，如果 <b>hostPrefix</b> 设为 <b>23</b> ，则每个节点从所给的 <b>cidr</b> 中分配一个 <b>/23</b> 子网。 <b>hostPrefix</b> 值 <b>23</b> 提供 $510 (2^{(32 - 23)} - 2)$ 个 pod IP 地址。	子网前缀。  默认值为 <b>23</b> 。
<b>networking.serviceNetwork</b>	服务的 IP 地址块。默认值为 <b>172.30.0.0/16</b> 。  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>

参数	描述	值
<b>networking.machineNetwork</b>	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>
<b>networking.machineNetwork.cidr</b>	使用 <b>networking.machineNetwork</b> 时需要。IP 地址块。libvirt 以外的所有平台的默认值为 <b>10.0.0.0/16</b> 。对于 libvirt，默认值为 <b>192.168.126.0/24</b> 。	CIDR 表示法中的 IP 网络块。  例如： <b>10.0.0.0/16</b> 。   <b>注意</b>  将 <b>networking.machineNetwork</b> 设置为与首选 NIC 所在的 CIDR 匹配。

### 1.8.8.1.3. 可选配置参数

下表描述了可选安装配置参数：

表 1.24. 可选参数

参数	描述	值
<b>additionalTrustBundle</b>	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
<b>compute</b>	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。
<b>compute.architecture</b>	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 <b>amd64</b> （默认值）。	字符串

参数	描述	值
<b>compute.hyperthreading</b>	<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><b>重要</b></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>	<b>Enabled 或 Disabled</b>
<b>compute.name</b>	使用 <b>compute</b> 时需要此值。机器池的名称。	<b>worker</b>
<b>compute.platform</b>	使用 <b>compute</b> 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 <b>controlPlane.platform</b> 参数值匹配。	<b>aws、azure、gcp、openstack、ovirt、vsphere 或 {}</b>
<b>compute.replicas</b>	要置备的计算机器数量，也称为 worker 机器。	大于或等于 <b>2</b> 的正整数。默认值为 <b>3</b> 。
<b>controlPlane</b>	组成 control plane 的机器的配置。	<b>MachinePool</b> 对象的数组。详情请查看以下"Machine-pool"表。
<b>controlPlane.architecture</b>	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 <b>amd64</b> （默认值）。	字符串
<b>controlPlane.hyperthreading</b>	<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><b>重要</b></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>	<b>Enabled 或 Disabled</b>
<b>controlPlane.name</b>	使用 <b>controlPlane</b> 时需要。机器池的名称。	<b>master</b>

参数	描述	值
<b>controlPlane.platform</b>	使用 <b>controlPlane</b> 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 <b>compute.platform</b> 参数值匹配。	<b>aws、azure、gcp、openstack、ovirt、vsphere</b> 或 <b>{}</b>
<b>controlPlane.replicas</b>	要置备的 control plane 机器数量。	唯一支持的值是 <b>3</b> ，它是默认值。
<b>credentialsMode</b>	<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: center;">  <div> <p><b>注意</b></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>	<b>Mint、Passthrough、Manual</b> 或空字符串(“”)。
<b>fips</b>	<p>启用或禁用 FIPS 模式。默认为 <b>false</b> (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: center;">  <div> <p><b>重要</b></p> <p>只有在 <b>x86_64</b> 架构中的 OpenShift Container Platform 部署支持 FIPS 验证的 <code>/Modules in Process</code> 加密库。</p> </div> </div> <div style="display: flex; align-items: center; margin-top: 10px;">  <div> <p><b>注意</b></p> <p>如果使用 Azure File 存储，则无法启用 FIPS 模式。</p> </div> </div>	<b>false</b> 或 <b>true</b>



参数	描述	值
<b>imageContentSources</b>	release-image 内容的源和仓库。	对象数组。包括一个 <b>source</b> 以及可选的 <b>mirrors</b> ，如下表所示。
<b>imageContentSources.source</b>	使用 <b>imageContentSources</b> 时需要。指定用户在镜像拉取规格中引用的仓库。	字符串
<b>imageContentSources.mirrors</b>	指定可能还包含同一镜像的一个或多个仓库。	字符串数组
<b>publish</b>	如何发布或公开集群的面向用户的端点，如 Kubernetes API、OpenShift 路由。	<b>Internal</b> 或 <b>External</b> 。把 <b>publish</b> 设置为 <b>Internal</b> 以部署一个私有集群，它不能被互联网访问。默认值为 <b>External</b> 。
<b>sshKey</b>	<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: center;">  <div> <p><b>注意</b></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 <b>ssh-agent</b> 进程使用的 SSH 密钥。</p> </div> </div>	<p>一个或多个密钥。例如：</p> <pre>sshKey:   &lt;key1&gt;   &lt;key2&gt;   &lt;key3&gt;</pre>

#### 1.8.8.1.4. 可选的 AWS 配置参数

下表描述了可选的 AWS 配置参数：

表 1.25. 可选的 AWS 参数

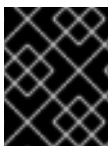
参数	描述	值
<b>compute.platform.aws.amiID</b>	用于为集群引导计算机器的 AWS AMI。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<b>compute.platform.aws.rootVolume.iops</b>	为根卷保留的每秒输入/输出操作 (IOPS) 数。	整数，如 <b>4000</b> 。

参数	描述	值
<code>compute.platform.aws.rootVolume.size</code>	以 GiB 为单位的根卷大小。	整数，如 <b>500</b> 。
<code>compute.platform.aws.rootVolume.type</code>	根卷的类型。	有效的 <a href="#">AWS EBS 卷类型</a> ，如 <b>io1</b> 。
<code>compute.platform.aws.rootVolume.kmsKeyARN</code>	KMS 密钥的 Amazon 资源名称（密钥 ARN）。这是使用特定 KMS 密钥加密 worker 节点的操作系统卷。	有效的 <a href="#">密钥 ID 或密钥 ARN</a> 。
<code>compute.platform.aws.type</code>	计算机器的 EC2 实例类型。	有效的 <a href="#">AWS 实例类型</a> ，如 <b>c5.9xlarge</b> 。
<code>compute.platform.aws.zones</code>	安装程序在其中为计算机器池创建机器的可用区。如果您提供自己的 VPC，则必须在那个可用域中提供一个子网。	有效 AWS 可用区的列表，如 <b>us-east-1c</b> ，以 <a href="#">YAML 序列</a> 表示。
<code>compute.aws.region</code>	安装程序在其中创建计算资源的 AWS 区域。	任何有效的 <a href="#">AWS 区域</a> ，如 <b>us-east-1</b> 。
<code>controlPlane.platform.aws.amiID</code>	用于为集群引导 control plane 机器的 AWS AMI。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。
<code>controlPlane.platform.aws.rootVolume.kmsKeyARN</code>	KMS 密钥的 Amazon 资源名称（密钥 ARN）。这需要使用特定的 KMS 密钥加密 control plane 节点的操作系统卷。	有效的 <a href="#">密钥 ID 和密钥 ARN</a> 。
<code>controlPlane.platform.aws.type</code>	control plane 机器的 EC2 实例类型。	有效的 <a href="#">AWS 实例类型</a> ，如 <b>c5.9xlarge</b> 。
<code>controlPlane.platform.aws.zones</code>	安装程序在其中为 control plane 机器池创建机器的可用区。	有效 AWS 可用区的列表，如 <b>us-east-1c</b> ，以 <a href="#">YAML 序列</a> 表示。
<code>controlPlane.aws.region</code>	安装程序在其中创建 control plane 资源的 AWS 区域。	有效的 <a href="#">AWS 区域</a> ，如 <b>us-east-1</b> 。
<code>platform.aws.amiID</code>	用于为集群引导所有机器的 AWS AMI。如果设置，AMI 必须属于与集群相同的区域。对于需要自定义 RHCOS AMI 的区域来说，这是必需的。	属于集合 AWS 区域的任何已发布或自定义 RHCOS AMI。

参数	描述	值
<b>platform.aws.serviceEndpoints.name</b>	AWS 服务端点名称。只有在必须使用替代 AWS 端点（如 FIPS）时，才需要自定义端点。可以为 EC2、S3、IAM、Elastic Load Balancing、Tagging、Route 53 和 STS AWS 服务指定自定义 API 端点。	有效的 <a href="#">AWS 服务端点</a> 名称。
<b>platform.aws.serviceEndpoints.url</b>	AWS 服务端点 URL。URL 必须使用 <b>https</b> 协议，主机必须信任该证书。	有效的 <a href="#">AWS 服务端点</a> URL。
<b>platform.aws.userTags</b>	键与值的映射，安装程序将其作为标签添加到它所创建的所有资源。	任何有效的 YAML 映射，如 <b>&lt;key&gt;: &lt;value&gt;</b> 格式的键值对。如需有关 AWS 标签的更多信息，请参阅 AWS 文档中的 <a href="#">标记您的 Amazon EC2 资源</a> 。
<b>platform.aws.subnets</b>	如果您提供 VPC，而不是让安装程序为您创建 VPC，请指定要使用的集群子网。子网必须是您指定的同一 <b>machineNetwork[].cidr</b> 范围的一部分。对于标准集群，为每个可用区指定一个公共和私有子网。对于私有集群，为每个可用区指定一个私有子网。	有效的子网 ID。

### 1.8.8.2. AWS 的自定义 install-config.yaml 文件示例

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



#### 重要

此示例 YAML 文件仅供参考。您必须使用安装程序来获取 **install-config.yaml** 文件，并且修改该文件。

```

apiVersion: v1
baseDomain: example.com 1
credentialsMode: Mint 2
controlPlane: 3 4
  hyperthreading: Enabled 5
name: master
platform:
  aws:
    zones:

```

```
- us-gov-west-1a
- us-gov-west-1b
rootVolume:
  iops: 4000
  size: 500
  type: io1 6
  type: m5.xlarge
replicas: 3
compute: 7
- hyperthreading: Enabled 8
  name: worker
  platform:
    aws:
      rootVolume:
        iops: 2000
        size: 500
        type: io1 9
        type: c5.4xlarge
      zones:
        - us-gov-west-1c
    replicas: 3
metadata:
  name: test-cluster 10
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  aws:
    region: us-gov-west-1
    userTags:
      adminContact: jdoe
      costCenter: 7536
    subnets: 11
    - subnet-1
    - subnet-2
    - subnet-3
    amiID: ami-96c6f8f7 12
    serviceEndpoints: 13
    - name: ec2
      url: https://vpce-id.ec2.us-west-2.vpce.amazonaws.com
    hostedZone: Z3URY6TWQ91KVV 14
  fips: false 15
  sshKey: ssh-ed25519 AAAA... 16
  publish: Internal 17
  pullSecret: '{"auths": ...}' 18
  additionalTrustBundle: | 19
```

```
-----BEGIN CERTIFICATE-----
<MY_TRUSTED_CA_CERT>
-----END CERTIFICATE-----
```

- 1 10 18** 必需。
- 2** 可选：添加此参数来强制 Cloud Credential Operator (CCO) 使用指定的模式，而不是让 CCO 动态尝试决定凭证的功能。如需有关 CCO 模式的详情，请参阅 *Red Hat Operator* 参考内容中的 *Cloud Credential Operator* 条目。
- 3 7** 如果没有提供这些参数和值，安装程序会提供默认值。
- 4** **controlPlane** 部分是一个单个映射，但 **compute** 部分是一系列映射。为满足不同数据结构的要求，**compute** 部分的第一行必须以连字符 - 开头，**controlPlane** 部分的第一行则不以连字符开头。只使用一个 control plane 池。
- 5 8** 是否要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。您可以通过将参数值设为 **Disabled** 来禁用。如果您在某些集群机器上禁用并发多线程，则必须在所有集群机器上禁用。



### 重要

如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。如果您对机器禁用并发多线程，请使用较大的实例类型，如 **m4.2xlarge** 或 **m5.2xlarge**。

- 6 9** 要为 etcd 配置更快的存储，特别是对于较大的集群，请将存储类型设置为 **io1**，并将 **iops** 设为 **2000**。
- 11** 如果您提供自己的 VPC，为集群使用的每个可用区指定子网。
- 12** 用于为集群引导机器的 AMI ID。如果设置，AMI 必须属于与集群相同的区域。
- 13** AWS 服务端点。在安装到未知 AWS 区域时，需要自定义端点。端点 URL 必须使用 **https** 协议，主机必须信任该证书。
- 14** 您现有 Route 53 私有托管区的 ID。提供现有的托管区需要您提供自己的 VPC，托管区已在安装集群前与 VPC 关联。如果未定义，安装程序会创建一个新的托管区。
- 15** 是否启用或禁用 FIPS 模式。默认情况下不启用 FIPS 模式。如果启用了 FIPS 模式，运行 OpenShift Container Platform 的 Red Hat Enterprise Linux CoreOS (RHCOS) 机器会绕过默认的 Kubernetes 加密套件，并使用由 RHCOS 提供的加密模块。



### 重要

只有在 **x86\_64** 架构中的 OpenShift Container Platform 部署支持 FIPS 验证的 `/Modules in Process` 加密库。

- 16** 您可以选择提供您用来访问集群中机器的 **sshKey** 值。



### 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。

- 17 如何发布集群的面向用户的端点。把 **publish** 设置为 **Internal** 以部署一个私有集群，它不能被互联网访问。默认值为 **External**。
- 19 自定义 CA 证书。当部署到 AWS C2S Secret 区域时，这是必需的，因为 AWS API 需要自定义 CA 信任捆绑包。

### 1.8.8.3. 没有公布的 RHCOS AMI 的 AWS 区域

您可以将 OpenShift Container Platform 集群部署到 Amazon Web Services (AWS) 区域，而无需对 Red Hat Enterprise Linux CoreOS (RHCOS) Amazon Machine Image (AMI) 或 AWS 软件开发 kit (SDK) 的原生支持。如果 AWS 区域没有可用的已公布的 AMI，您可以在安装集群前上传自定义 AMI。如果您要将集群部署到 AWS 政府区域，则需要此参数。

如果您部署到没有公布的 RHCOS AMI 的非机构区域，且您没有指定自定义的 AMI，安装程序会自动将 **us-east-1** AMI 复制到用户帐户。然后，安装程序使用默认或用户指定的密钥管理服务 (KMS) 密钥创建带有加密 EBS 卷的 control plane 机器。这允许 AMI 跟踪与公布的 RHCOS AMI 相同的进程工作流。

在集群创建过程中，无法从终端中选择没有原生支持 RHCOS AMI 的区域，因为它没有发布。但是，您可以通过在 **install-config.yaml** 文件中配置自定义 AMI 来安装到这个区域。

### 1.8.8.4. 在 AWS 中上传自定义 RHCOS AMI

如果要部署到自定义 Amazon Web Services (AWS) 区域，您必须上传属于该区域的自定义 Red Hat Enterprise Linux CoreOS (RHCOS) Amazon Machine Image (AMI)。

#### 先决条件

- 已配置了一个 AWS 帐户。
- 已使用所需的 IAM [服务角色](#) 创建 Amazon S3 存储桶。
- 将 RHCOS VMDK 文件上传到 Amazon S3。RHCOS VMDK 文件必须是小于或等于您要安装的 OpenShift Container Platform 版本的最高版本。
- 您下载了 AWS CLI 并安装到您的计算机上。请参阅[使用捆绑安装程序安装 AWS CLI](#)。

#### 流程

1. 将 AWS 配置集导出为环境变量：

```
$ export AWS_PROFILE=<aws_profile> 1
```

- 1 拥有 AWS 凭证的 AWS 配置集名称，如 **govcloud**。

2. 将与自定义 AMI 关联的区域导出为环境变量：

```
$ export AWS_DEFAULT_REGION=<aws_region> 1
```

- 1 AWS 区域，如 **us-gov-east-1**。

3. 将上传至 Amazon S3 的 RHCOS 版本导出为环境变量：

```
$ export RHCOS_VERSION=<version> ❶
```

❶ RHCOS VMDK 版本，如 **4.6.0**。

4. 将 Amazon S3 存储桶名称导出为环境变量：

```
$ export VMIMPORT_BUCKET_NAME=<s3_bucket_name>
```

5. 创建 **containers.json** 文件并定义 RHCOS VMDK 文件：

```
$ cat <<EOF > containers.json
{
  "Description": "rhcos-${RHCOS_VERSION}-x86_64-aws.x86_64",
  "Format": "vmdk",
  "UserBucket": {
    "S3Bucket": "${VMIMPORT_BUCKET_NAME}",
    "S3Key": "rhcos-${RHCOS_VERSION}-x86_64-aws.x86_64.vmdk"
  }
}
EOF
```

6. 将 RHCOS 磁盘导入为 Amazon EBS 快照：

```
$ aws ec2 import-snapshot --region ${AWS_DEFAULT_REGION} \
  --description "<description>" ❶ \
  --disk-container "file://<file_path>/containers.json" ❷
```

❶ 导入 RHCOS 磁盘的描述，如 **rhcos-\${RHCOS\_VERSION}-x86\_64-aws.x86\_64**。

❷ 描述 RHCOS 磁盘的 JSON 文件的文件路径。JSON 文件应包含您的 Amazon S3 存储桶名称和密钥。

7. 检查镜像导入的状态：

```
$ watch -n 5 aws ec2 describe-import-snapshot-tasks --region ${AWS_DEFAULT_REGION}
```

### 输出示例

```
{
  "ImportSnapshotTasks": [
    {
      "Description": "rhcos-4.6.0-x86_64-aws.x86_64",
      "ImportTaskId": "import-snap-fh6i8uil",
      "SnapshotTaskDetail": {
        "Description": "rhcos-4.6.0-x86_64-aws.x86_64",
        "DiskImageSize": 819056640.0,
        "Format": "VMDK",
        "SnapshotId": "snap-06331325870076318",
        "Status": "completed",
        "UserBucket": {
          "S3Bucket": "external-images",
          "S3Key": "rhcos-4.6.0-x86_64-aws.x86_64.vmdk"
        }
      }
    }
  ]
}
```

```

    }
  }
}
]
}

```

复制 **SnapshotId** 以注册镜像。

#### 8. 从 RHCOS 快照创建自定义 RHCOS AMI:

```

$ aws ec2 register-image \
  --region ${AWS_DEFAULT_REGION} \
  --architecture x86_64 \ ❶
  --description "rhcos-${RHCOS_VERSION}-x86_64-aws.x86_64" \ ❷
  --ena-support \
  --name "rhcos-${RHCOS_VERSION}-x86_64-aws.x86_64" \ ❸
  --virtualization-type hvm \
  --root-device-name '/dev/xvda' \
  --block-device-mappings 'DeviceName=/dev/xvda,Ebs=
{DeleteOnTermination=true,SnapshotId=<snapshot_ID>' ❹

```

- ❶ RHCOS VMDK 架构类型，如 **x86\_64**、**s390x** 或 **ppc64le**。
- ❷ 来自导入快照的 **Description**。
- ❸ RHCOS AMI 的名称。
- ❹ 导入的快照中的 **SnapshotID**。

如需了解更多有关这些 API 的信息，请参阅 AWS 文档 [导入快照](#) 和 [创建由 EBS 支持的 AMI](#)。

#### 1.8.8.5. 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)。



- 如果您的集群位于 AWS 上，请将 `ec2.<region>.amazonaws.com`、`elasticloadbalancing.<region>.amazonaws.com` 和 `s3.<region>.amazonaws.com` 端点添加到 VPC 端点。需要这些端点才能完成节点到 AWS EC2 API 的请求。由于代理在容器级别而不是节点级别工作，因此您必须通过 AWS 专用网络将这些请求路由到 AWS EC2 API。在代理服务器中的允许列表中添加 EC2 API 的公共 IP 地址是不够的。

## 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` 的配置映射来保存额外的 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，将为 `trustedCA` 参数指定的内容与 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。

## 1.8.9. 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

- 配置托管集群的云平台的帐户。
- 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。

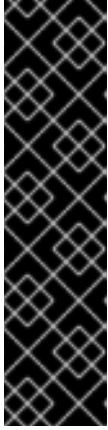
### 输出示例

```
...
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 Ignition 配置文件生成后的 12 小时内使用它们，因为 24 小时的证书会在集群安装后的 16 小时到 22 小时进行轮转。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因为执行了证书更新而导致安装失败的问题。



### 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。

2. 可选：从您用来安装集群的 IAM 帐户删除或禁用 **AdministratorAccess** 策略。

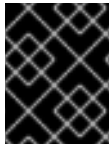


### 注意

只有在安装过程中才需要 **AdministratorAccess** 策略提供的升级权限。

## 1.8.10. 通过下载二进制文件安装 OpenShift CLI

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



### 重要

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

### 1.8.10.1. 在 Linux 上安装 OpenShift CLI

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

#### 流程

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

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

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

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 1.8.10.2. 在 Windows 上安装 OpenShift CLI

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

流程

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

```
C:\> path
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 1.8.10.3. 在 macOS 上安装 OpenShift CLI

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

流程

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

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 1.8.11. 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
```

### 1.8.12. 使用 Web 控制台登录到集群

**kubeadmin** 用户默认在 OpenShift Container Platform 安装后存在。您可以使用 OpenShift Container Platform Web 控制台以 **kubeadmin** 用户身份登录集群。

#### 先决条件

- 有访问安装主机的访问权限。
- 您完成了集群安装，所有集群 Operator 都可用。

#### 流程

1. 从安装主机上的 **kubeadmin-password** 文件中获取 kubeadmin 用户的密码：

```
$ cat <installation_directory>/auth/kubeadmin-password
```



#### 注意

另外，您还可以从安装主机上的 **<installation\_directory>/openshift\_install.log** 日志文件获取 **kubeadmin** 密码。

2. 列出 OpenShift Container Platform Web 控制台路由：

```
$ oc get routes -n openshift-console | grep 'console-openshift'
```



### 注意

另外，您还可以从安装主机上的 `<installation_directory>/openshift_install.log` 日志文件获取 OpenShift Container Platform 路由。

### 输出示例

```
console console-openshift-console.apps.<cluster_name>.<base_domain> console
https reencrypt/Redirect None
```

3. 在 Web 浏览器中导航到上一命令输出中包括的路由，以 **kubeadmin** 用户身份登录。

### 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](#)。

## 1.8.13. 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](#)。

## 1.8.14. 后续步骤

- [验证安装](#)。
- [自定义集群](#)。
- 如果需要，您可以[选择不使用远程健康报告](#)。
- 如果需要，您可以[删除云供应商凭证](#)。

## 1.9. 使用 CLOUDFORMATION 模板在 AWS 中用户置备的基础架构上安装集群

在 OpenShift Container Platform 版本 4.6 中，您可以使用您提供的基础架构在 Amazon Web Services (AWS) 上安装集群。

创建此基础架构的一种方法是使用提供的 CloudFormation 模板。您可以修改模板来自定义基础架构，或使用其包含的信息来按照公司策略创建 AWS 对象。

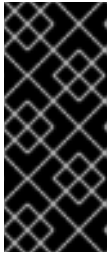


## 重要

进行用户置备的基础架构安装的步骤仅作为示例。使用您提供的基础架构安装集群需要了解云供应商和 OpenShift Container Platform 安装过程。提供的几个 CloudFormation 模板可帮助完成这些步骤，或者帮助您自行建模。您也可以自由选择通过其他方法创建所需的资源；模板仅作参考之用。

### 1.9.1. 先决条件

- 您可以参阅有关 [OpenShift Container Platform 安装和更新流程](#) 的详细信息。
- 已将 [AWS 帐户配置](#) 为托管集群。



## 重要

如果您的计算机上存储有 AWS 配置集，则不要在使用多因素验证设备的同时使用您生成的临时会话令牌。在集群的整个生命周期中，集群会持续使用您的当前 AWS 凭证来创建 AWS 资源，因此您必须使用基于密钥的长期凭证。要生成适当的密钥，请参阅 AWS 文档中的[管理 IAM 用户的访问密钥](#)。您可在运行安装程序时提供密钥。

- 您下载了 AWS CLI 并安装到您的计算机上。请参阅 AWS 文档中的[使用捆绑安装程序 \(Linux、macOS 或 Unix\) 安装 AWS CLI](#)。
- 如果使用防火墙，[将其配置为允许集群需要访问的站点](#)。



## 注意

如果您要配置代理，请务必也要查看此站点列表。

- 如果不允许系统管理身份和访问管理（IAM），集群管理员可以 [手动创建和维护 IAM 凭证](#)。手动模式也可以用于云 IAM API 无法访问的环境中。

### 1.9.2. 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。

### 1.9.3. 所需的 AWS 基础架构组件

要在 Amazon Web Services (AWS) 中用户置备的基础架构上安装 OpenShift Container Platform，您必须手动创建机器及其支持的基础架构。

如需有关不同平台集成测试的更多信息，请参阅 [OpenShift Container Platform 4.x Tested Integrations](#) 页面。

通过使用提供的 CloudFormation 模板，您可以创建代表以下组件的 AWS 资源堆栈：

- 一个 AWS Virtual Private Cloud (VPC)
- 网络和负载均衡组件
- 安全组和角色
- 一个 OpenShift Container Platform bootstrap 节点
- OpenShift Container Platform control plane 节点
- 一个 OpenShift Container Platform 计算节点

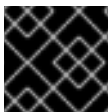
或者，您可以手动创建组件，也可以重复使用满足集群要求的现有基础架构。查看 CloudFormation 模板，了解组件如何相互连接的更多详情。

#### 1.9.3.1. 集群机器

以下机器需要 `AWS::EC2::Instance` 对象：

- bootstrap 机器。安装过程中需要此机器，但可在集群部署后删除。
- 三个 control plane 机器。control plane 机器不受机器集的管控。
- 计算机器。在安装过程中创建至少两台计算（compute）机器（也称为 worker 机器）。这些机器不受机器集的管控。

您可以通过提供的 CloudFormation 模板，为集群机器使用以下实例类型。



#### 重要

如果您的区域中没有 `m4` 实例类型，例如 `eu-west-3`，请改为使用 `m5` 类型。

表 1.26. 机器的实例类型

实例类型	bootstrap	Control plane	Compute
<code>i3.large</code>	x		
<code>m4.large</code>			x
<code>m4.xlarge</code>		x	x
<code>m4.2xlarge</code>		x	x



实例类型	bootstrap	Control plane	Compute
<b>m4.4xlarge</b>		x	x
<b>m4.8xlarge</b>		x	x
<b>m4.10xlarge</b>		x	x
<b>m4.16xlarge</b>		x	x
<b>m5.large</b>			x
<b>m5.xlarge</b>		x	x
<b>m5.2xlarge</b>		x	x
<b>m5.4xlarge</b>		x	x
<b>m5.8xlarge</b>		x	x
<b>m5.10xlarge</b>		x	x
<b>m5.16xlarge</b>		x	x
<b>m6i.xlarge</b>		x	x
<b>c4.2xlarge</b>		x	x
<b>c4.4xlarge</b>		x	x
<b>c4.8xlarge</b>		x	x
<b>r4.large</b>			x
<b>r4.xlarge</b>		x	x
<b>r4.2xlarge</b>		x	x
<b>r4.4xlarge</b>		x	x
<b>r4.8xlarge</b>		x	x
<b>r4.16xlarge</b>		x	x

您可能能够使用符合这些实例类型规格的其他实例类型。

### 1.9.3.2. 其他基础架构组件

- VPC
- DNS 条目
- 负载均衡器（典型或网络）和监听器
- 公共和专用路由 53 区域
- 安全组
- IAM 角色
- S3 存储桶

如果您在断开连接的环境或使用代理的环境中工作，则无法访问 EC2 和 ELB 端点的公共 IP 地址。要访问这些端点，您必须创建一个 VPC 端点，并将其附加到集群使用的子网。创建以下端点：

- **ec2.<region>.amazonaws.com**
- **elasticloadbalancing.<region>.amazonaws.com**
- **s3.<region>.amazonaws.com**

### 所需的 VPC 组件

您必须提供合适的 VPC 和子网，以便与您的机器通信。

组件	AWS 类型	描述
VPC	<ul style="list-style-type: none"> <li>• <b>AWS::EC2::VPC</b></li> <li>• <b>AWS::EC2::VPCEndpoint</b></li> </ul>	您必须提供一个公共 VPC 供集群使用。VPC 使用引用每个子网的路由表的端点，以改进与托管在 S3 中的 registry 的通信。
公共子网	<ul style="list-style-type: none"> <li>• <b>AWS::EC2::Subnet</b></li> <li>• <b>AWS::EC2::SubnetNetworkAclAssociation</b></li> </ul>	您的 VPC 必须有 1 到 3 个可用区的公共子网，并将其与适当的入口规则关联。
互联网网关	<ul style="list-style-type: none"> <li>• <b>AWS::EC2::InternetGateway</b></li> <li>• <b>AWS::EC2::VPCGatewayAttachment</b></li> <li>• <b>AWS::EC2::RouteTable</b></li> <li>• <b>AWS::EC2::Route</b></li> <li>• <b>AWS::EC2::SubnetRouteTableAssociation</b></li> <li>• <b>AWS::EC2::NatGateway</b></li> <li>• <b>AWS::EC2::EIP</b></li> </ul>	您必须有一个公共互联网网关，以及附加到 VPC 的公共路由。在提供的模板中，每个公共子网都有一个具有 EIP 地址的 NAT 网关。这些 NAT 网关允许集群资源（如专用子网实例）访问互联网，而有些受限网络或代理场景则不需要它们。

组件	AWS 类型	描述	
网络访问控制	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::NetworkAcl</b></li> <li>● <b>AWS::EC2::NetworkAclEntry</b></li> </ul>	您必须允许 VPC 访问下列端口：	
		端口	原因
		80	入站 HTTP 流量
		443	入站 HTTPS 流量
		22	入站 SSH 流量
		1024 - 65535	入站临时流量
0 - 65535	出站临时流量		
专用子网	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::Subnet</b></li> <li>● <b>AWS::EC2::RouteTable</b></li> <li>● <b>AWS::EC2::SubnetRouteTableAssociation</b></li> </ul>	您的 VPC 可以具有私有子网。提供的 CloudFormation 模板可为 1 到 3 个可用区创建专用子网。如果您使用专用子网，必须为其提供适当的路由和表。	

### 所需的 DNS 和负载均衡组件

您的 DNS 和负载均衡器配置需要使用公共托管区，并可使用类似安装程序使用的专用托管区（如果安装程序配备了集群的基础架构）。您必须创建一个解析到负载均衡器的 DNS 条目。**api.<cluster\_name>.<domain>** 的条目必须指向外部负载均衡器，**api-int.<cluster\_name>.<domain>** 的条目则必须指向内部负载均衡器。

集群还需要负载均衡器，以及监听端口 6443（用于 Kubernetes API 及其扩展）和端口 22623（用于新机器的 Ignition 配置文件）的监听程序。目标是 control plane 节点（也称为 master 节点）。集群外的客户端和集群内的节点都必须能够访问端口 6443。集群内的节点必须能够访问端口 22623。

组件	AWS 类型	描述
DNS	<b>AWS::Route53::HostedZone</b>	内部 DNS 的托管区。
etcd 记录集	<b>AWS::Route53::RecordSet</b>	control plane 机器的 etcd 注册记录。
公共负载均衡器	<b>AWS::ElasticLoadBalancingV2::LoadBalancer</b>	公共子网的负载均衡器。

组件	AWS 类型	描述
外部 API 服务器记录	<b>AWS::Route53::RecordSetGroup</b>	外部 API 服务器的别名记录。
外部监听程序	<b>AWS::ElasticLoadBalancingV2::Listener</b>	为外部负载均衡器监听端口 6443 的监听程序。
外部目标组	<b>AWS::ElasticLoadBalancingV2::TargetGroup</b>	外部负载均衡器的目标组。
专用负载均衡器	<b>AWS::ElasticLoadBalancingV2::LoadBalancer</b>	专用子网的负载均衡器。
内部 API 服务器记录	<b>AWS::Route53::RecordSetGroup</b>	内部 API 服务器的别名记录。
内部监听程序	<b>AWS::ElasticLoadBalancingV2::Listener</b>	为内部负载均衡器监听端口 22623 的监听程序。
内部目标组	<b>AWS::ElasticLoadBalancingV2::TargetGroup</b>	内部负载均衡器的目标组。
内部监听程序	<b>AWS::ElasticLoadBalancingV2::Listener</b>	为内部负载均衡器监听端口 6443 的监听程序。
内部目标组	<b>AWS::ElasticLoadBalancingV2::TargetGroup</b>	内部负载均衡器的目标组。

## 安全组

control plane 和 worker 机器需要访问下列端口：

组	类型	IP 协议	端口范围
<b>MasterSecurityGroup</b>	<b>AWS::EC2::Security Group</b>	<b>icmp</b>	<b>0</b>
		<b>tcp</b>	<b>22</b>
		<b>tcp</b>	<b>6443</b>
		<b>tcp</b>	<b>22623</b>
<b>WorkerSecurityGroup</b>	<b>AWS::EC2::Security Group</b>	<b>icmp</b>	<b>0</b>
		<b>tcp</b>	<b>22</b>
<b>BootstrapSecurityGroup</b>	<b>AWS::EC2::Security Group</b>	<b>tcp</b>	<b>22</b>
		<b>tcp</b>	<b>19531</b>

## control plane 入口

control plane 机器需要以下入口组。每个入口组都是 **AWS::EC2::SecurityGroupIngress** 资源。

入口组	描述	IP 协议	端口范围
<b>MasterIngress Etcd</b>	etcd	<b>tcp</b>	<b>2379- 2380</b>
<b>MasterIngress Vxlan</b>	Vxlan 数据包	<b>udp</b>	<b>4789</b>
<b>MasterIngress WorkerVxlan</b>	Vxlan 数据包	<b>udp</b>	<b>4789</b>
<b>MasterIngress Internal</b>	内部集群通信和 Kubernetes 代理指标	<b>tcp</b>	<b>9000 - 9999</b>
<b>MasterIngress WorkerInternal</b>	内部集群通信	<b>tcp</b>	<b>9000 - 9999</b>
<b>MasterIngress Kube</b>	kubernetes kubelet、调度程序和控制器管理器	<b>tcp</b>	<b>10250 - 10259</b>
<b>MasterIngress WorkerKube</b>	kubernetes kubelet、调度程序和控制器管理器	<b>tcp</b>	<b>10250 - 10259</b>

入口组	描述	IP 协议	端口范围
<b>MasterIngress IngressServices</b>	Kubernetes 入口服务	<b>tcp</b>	<b>30000 - 32767</b>
<b>MasterIngress WorkerIngress Services</b>	Kubernetes 入口服务	<b>tcp</b>	<b>30000 - 32767</b>
<b>MasterIngress Geneve</b>	Geneve 数据包	<b>udp</b>	<b>6081</b>
<b>MasterIngress WorkerGeneve</b>	Geneve 数据包	<b>udp</b>	<b>6081</b>
<b>MasterIngress IpsecIke</b>	IPsec IKE 数据包	<b>udp</b>	<b>500</b>
<b>MasterIngress WorkerIpsecIke</b>	IPsec IKE 数据包	<b>udp</b>	<b>500</b>
<b>MasterIngress IpsecNat</b>	IPsec NAT-T 数据包	<b>udp</b>	<b>4500</b>
<b>MasterIngress WorkerIpsecNat</b>	IPsec NAT-T 数据包	<b>udp</b>	<b>4500</b>
<b>MasterIngress IpsecEsp</b>	IPsec ESP 数据包	<b>50</b>	<b>All</b>
<b>MasterIngress WorkerIpsecEsp</b>	IPsec ESP 数据包	<b>50</b>	<b>All</b>
<b>MasterIngress InternalUDP</b>	内部集群通信	<b>udp</b>	<b>9000 - 9999</b>
<b>MasterIngress WorkerInternalUDP</b>	内部集群通信	<b>udp</b>	<b>9000 - 9999</b>
<b>MasterIngress IngressServicesUDP</b>	Kubernetes 入口服务	<b>udp</b>	<b>30000 - 32767</b>

入口组	描述	IP 协议	端口范围
<b>MasterIngress WorkerIngress ServicesUDP</b>	Kubernetes 入口服务	<b>udp</b>	<b>30000 - 32767</b>

## worker 入口

worker 机器需要以下入口组。每个入口组都是 **AWS::EC2::SecurityGroupIngress** 资源。

入口组	描述	IP 协议	端口范围
<b>WorkerIngress Vxlan</b>	Vxlan 数据包	<b>udp</b>	<b>4789</b>
<b>WorkerIngress WorkerVxlan</b>	Vxlan 数据包	<b>udp</b>	<b>4789</b>
<b>WorkerIngress Internal</b>	内部集群通信	<b>tcp</b>	<b>9000 - 9999</b>
<b>WorkerIngress WorkerIntern al</b>	内部集群通信	<b>tcp</b>	<b>9000 - 9999</b>
<b>WorkerIngress Kube</b>	Kubernetes kubelet、调度程序和控制器管理器	<b>tcp</b>	<b>10250</b>
<b>WorkerIngress WorkerKube</b>	Kubernetes kubelet、调度程序和控制器管理器	<b>tcp</b>	<b>10250</b>
<b>WorkerIngress IngressServic es</b>	Kubernetes 入口服务	<b>tcp</b>	<b>30000 - 32767</b>
<b>WorkerIngress WorkerIngress Services</b>	Kubernetes 入口服务	<b>tcp</b>	<b>30000 - 32767</b>
<b>WorkerIngress Geneve</b>	Geneve 数据包	<b>udp</b>	<b>6081</b>
<b>WorkerIngress MasterGeneve</b>	Geneve 数据包	<b>udp</b>	<b>6081</b>
<b>WorkerIngress IpsecIke</b>	IPsec IKE 数据包	<b>udp</b>	<b>500</b>

入口组	描述	IP 协议	端口范围
<b>WorkerIngressMasterIpsecckle</b>	IPsec IKE 数据包	udp	500
<b>WorkerIngressIpsecNat</b>	IPsec NAT-T 数据包	udp	4500
<b>WorkerIngressMasterIpsecNat</b>	IPsec NAT-T 数据包	udp	4500
<b>WorkerIngressIpsecEsp</b>	IPsec ESP 数据包	50	All
<b>WorkerIngressMasterIpsecEsp</b>	IPsec ESP 数据包	50	All
<b>WorkerIngressInternalUDP</b>	内部集群通信	udp	9000 - 9999
<b>WorkerIngressMasterInternalUDP</b>	内部集群通信	udp	9000 - 9999
<b>WorkerIngressIngressServicesUDP</b>	Kubernetes 入口服务	udp	30000 - 32767
<b>WorkerIngressMasterIngressServicesUDP</b>	Kubernetes 入口服务	udp	30000 - 32767

## 角色和实例配置集

您必须在 AWS 中为机器授予权限。提供的 CloudFormation 模板为以下 **AWS::IAM::Role** 对象授予机器 **Allow** 权限，并为每一组角色提供一个 **AWS::IAM::InstanceProfile**。如果不使用模板，您可以为机器授予以下宽泛权限或单独权限。

角色	影响	操作	资源
Master	<b>Allow</b>	<b>ec2:*</b>	*
	<b>Allow</b>	<b>elasticloadbalancing:*</b>	*

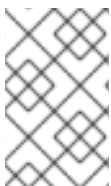


角色	影响	操作	资源
	<b>Allow</b>	<b>iam:PassRole</b>	*
	<b>Allow</b>	<b>s3:GetObject</b>	*
Worker	<b>Allow</b>	<b>ec2:Describe*</b>	*
bootstrap	<b>Allow</b>	<b>ec2:Describe*</b>	*
	<b>Allow</b>	<b>ec2:AttachVolume</b>	*
	<b>Allow</b>	<b>ec2:DetachVolume</b>	*

### 1.9.3.3. 证书签名请求管理

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

### 1.9.3.4. 所需的 AWS 权限



#### 注意

您的 IAM 用户必须在区域 **us-east-1** 中有权限 **tag:GetResources** 来删除基本集群资源。作为 AWS API 的要求的一部分，OpenShift Container Platform 安装程序在此区域中执行各种操作。

将 **AdministratorAccess** 策略附加到您在 Amazon Web Services (AWS) 中创建的 IAM 用户时，授予该用户所有需要的权限。要部署 OpenShift Container Platform 集群的所有组件，IAM 用户需要以下权限：

#### 例 1.13. 安装所需的 EC2 权限

- **tag:TagResources**
- **tag:UntagResources**
- **ec2:AllocateAddress**
- **ec2:AssociateAddress**
- **ec2:AuthorizeSecurityGroupEgress**
- **ec2:AuthorizeSecurityGroupIngress**
- **ec2:CopyImage**
- **ec2>CreateNetworkInterface**
- **ec2:AttachNetworkInterface**

- **ec2:CreateSecurityGroup**
- **ec2:CreateTags**
- **ec2:CreateVolume**
- **ec2>DeleteSecurityGroup**
- **ec2>DeleteSnapshot**
- **ec2>DeleteTags**
- **ec2:DeregisterImage**
- **ec2:DescribeAccountAttributes**
- **ec2:DescribeAddresses**
- **ec2:DescribeAvailabilityZones**
- **ec2:DescribeDhcpOptions**
- **ec2:DescribeImages**
- **ec2:DescribeInstanceAttribute**
- **ec2:DescribeInstanceCreditSpecifications**
- **ec2:DescribeInstances**
- **ec2:DescribeInternetGateways**
- **ec2:DescribeKeyPairs**
- **ec2:DescribeNatGateways**
- **ec2:DescribeNetworkAcls**
- **ec2:DescribeNetworkInterfaces**
- **ec2:DescribePrefixLists**
- **ec2:DescribeRegions**
- **ec2:DescribeRouteTables**
- **ec2:DescribeSecurityGroups**
- **ec2:DescribeSubnets**
- **ec2:DescribeTags**
- **ec2:DescribeVolumes**
- **ec2:DescribeVpcAttribute**
- **ec2:DescribeVpcClassicLink**

- **ec2:DescribeVpcClassicLinkDnsSupport**
- **ec2:DescribeVpcEndpoints**
- **ec2:DescribeVpcs**
- **ec2:GetEbsDefaultKmsKeyId**
- **ec2:ModifyInstanceAttribute**
- **ec2:ModifyNetworkInterfaceAttribute**
- **ec2:ReleaseAddress**
- **ec2:RevokeSecurityGroupEgress**
- **ec2:RevokeSecurityGroupIngress**
- **ec2:RunInstances**
- **ec2:TerminateInstances**

例 1.14. 安装过程中创建网络资源所需的权限

- **ec2:AssociateDhcpOptions**
- **ec2:AssociateRouteTable**
- **ec2:AttachInternetGateway**
- **ec2:CreateDhcpOptions**
- **ec2:CreateInternetGateway**
- **ec2:CreateNatGateway**
- **ec2:CreateRoute**
- **ec2:CreateRouteTable**
- **ec2:CreateSubnet**
- **ec2:CreateVpc**
- **ec2:CreateVpcEndpoint**
- **ec2:ModifySubnetAttribute**
- **ec2:ModifyVpcAttribute**



#### 注意

如果您使用现有的 VPC，您的帐户不需要这些权限来创建网络资源。

## 例 1.15. 安装所需的 Elastic Load Balancing 权限(ELB)

- **elasticloadbalancing:AddTags**
- **elasticloadbalancing:ApplySecurityGroupsToLoadBalancer**
- **elasticloadbalancing:AttachLoadBalancerToSubnets**
- **elasticloadbalancing:ConfigureHealthCheck**
- **elasticloadbalancing:CreateLoadBalancer**
- **elasticloadbalancing:CreateLoadBalancerListeners**
- **elasticloadbalancing>DeleteLoadBalancer**
- **elasticloadbalancing:DeregisterInstancesFromLoadBalancer**
- **elasticloadbalancing:DescribeInstanceHealth**
- **elasticloadbalancing:DescribeLoadBalancerAttributes**
- **elasticloadbalancing:DescribeLoadBalancers**
- **elasticloadbalancing:DescribeTags**
- **elasticloadbalancing:ModifyLoadBalancerAttributes**
- **elasticloadbalancing:RegisterInstancesWithLoadBalancer**
- **elasticloadbalancing:SetLoadBalancerPoliciesOfListener**

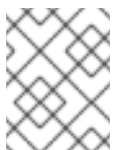
## 例 1.16. 安装所需的 Elastic Load Balancing 权限(ELBv2)

- **elasticloadbalancing:AddTags**
- **elasticloadbalancing:CreateListener**
- **elasticloadbalancing:CreateLoadBalancer**
- **elasticloadbalancing:CreateTargetGroup**
- **elasticloadbalancing>DeleteLoadBalancer**
- **elasticloadbalancing:DeregisterTargets**
- **elasticloadbalancing:DescribeListeners**
- **elasticloadbalancing:DescribeLoadBalancerAttributes**
- **elasticloadbalancing:DescribeLoadBalancers**
- **elasticloadbalancing:DescribeTargetGroupAttributes**
- **elasticloadbalancing:DescribeTargetHealth**

- **elasticloadbalancing:ModifyLoadBalancerAttributes**
- **elasticloadbalancing:ModifyTargetGroup**
- **elasticloadbalancing:ModifyTargetGroupAttributes**
- **elasticloadbalancing:RegisterTargets**

#### 例 1.17. 安装所需的 IAM 权限

- **iam:AddRoleToInstanceProfile**
- **iam:CreateInstanceProfile**
- **iam:CreateRole**
- **iam:DeleteInstanceProfile**
- **iam>DeleteRole**
- **iam>DeleteRolePolicy**
- **iam:GetInstanceProfile**
- **iam:GetRole**
- **iam:GetRolePolicy**
- **iam:GetUser**
- **iam:ListInstanceProfilesForRole**
- **iam:ListRoles**
- **iam:ListUsers**
- **iam:PassRole**
- **iam:PutRolePolicy**
- **iam:RemoveRoleFromInstanceProfile**
- **iam:SimulatePrincipalPolicy**
- **iam:TagRole**



#### 注意

如果您还没有在 AWS 帐户中创建弹性负载均衡器（ELB），IAM 用户还需要 **iam:CreateServiceLinkedRole** 权限。

#### 例 1.18. 安装所需的 Route 53 权限

- **route53:ChangeResourceRecordSets**

- **route53:ChangeTagsForResource**
- **route53:CreateHostedZone**
- **route53>DeleteHostedZone**
- **route53:GetChange**
- **route53:GetHostedZone**
- **route53:ListHostedZones**
- **route53:ListHostedZonesByName**
- **route53:ListResourceRecordSets**
- **route53:ListTagsForResource**
- **route53:UpdateHostedZoneComment**

例 1.19. 安装所需的 S3 权限

- **s3:CreateBucket**
- **s3>DeleteBucket**
- **s3:GetAccelerateConfiguration**
- **s3:GetBucketAcl**
- **s3:GetBucketCors**
- **s3:GetBucketLocation**
- **s3:GetBucketLogging**
- **s3:GetBucketObjectLockConfiguration**
- **s3:GetBucketReplication**
- **s3:GetBucketRequestPayment**
- **s3:GetBucketTagging**
- **s3:GetBucketVersioning**
- **s3:GetBucketWebsite**
- **s3:GetEncryptionConfiguration**
- **s3:GetLifecycleConfiguration**
- **s3:GetReplicationConfiguration**
- **s3:ListBucket**
- **s3:PutBucketAcl**

- **s3:PutBucketTagging**
- **s3:PutEncryptionConfiguration**

例 1.20. 集群 Operators 所需的 S3 权限

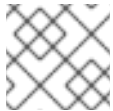
- **s3:DeleteObject**
- **s3:GetObject**
- **s3:GetObjectAcl**
- **s3:GetObjectTagging**
- **s3:GetObjectVersion**
- **s3:PutObject**
- **s3:PutObjectAcl**
- **s3:PutObjectTagging**

例 1.21. 删除基本集群资源所需的权限

- **autoscaling:DescribeAutoScalingGroups**
- **ec2:DeleteNetworkInterface**
- **ec2:DeleteVolume**
- **elasticloadbalancing:DeleteTargetGroup**
- **elasticloadbalancing:DescribeTargetGroups**
- **iam:DeleteAccessKey**
- **iam:DeleteUser**
- **iam>ListAttachedRolePolicies**
- **iam>ListInstanceProfiles**
- **iam>ListRolePolicies**
- **iam>ListUserPolicies**
- **s3:DeleteObject**
- **s3:ListBucketVersions**
- **tag:GetResources**

例 1.22. 删除网络资源所需的权限

- **ec2:DeleteDhcpOptions**
- **ec2:DeleteInternetGateway**
- **ec2:DeleteNatGateway**
- **ec2:DeleteRoute**
- **ec2:DeleteRouteTable**
- **ec2:DeleteSubnet**
- **ec2:DeleteVpc**
- **ec2:DeleteVpcEndpoints**
- **ec2:DetachInternetGateway**
- **ec2:DisassociateRouteTable**
- **ec2:ReplaceRouteTableAssociation**



#### 注意

如果您使用现有的 VPC，您的帐户不需要这些权限来删除网络资源。

#### 例 1.23. 创建清单所需的额外 IAM 和 S3 权限

- **iam:DeleteAccessKey**
- **iam:DeleteUser**
- **iam:DeleteUserPolicy**
- **iam:GetUserPolicy**
- **iam:ListAccessKeys**
- **iam:PutUserPolicy**
- **iam:TagUser**
- **iam:GetUserPolicy**
- **iam:ListAccessKeys**
- **s3:PutBucketPublicAccessBlock**
- **s3:GetBucketPublicAccessBlock**
- **s3:PutLifecycleConfiguration**
- **s3:HeadBucket**
- **s3:ListBucketMultipartUploads**



- **s3:AbortMultipartUpload**



### 注意

如果要使用 mint 模式管理云供应商凭证，IAM 用户还需要 **iam:CreateAccessKey** and **iam:CreateUser** 权限。

#### 例 1.24. 安装时配额检查的可选权限

- **servicequotas:ListAWSDefaultServiceQuotas**

## 1.9.4. 获取安装程序

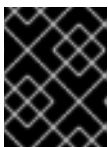
在安装 OpenShift Container Platform 之前，将安装文件下载到本地计算机上。

### 先决条件

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

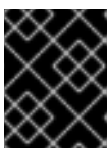
### 流程

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。

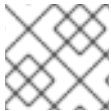
4. 提取安装程序。例如，在使用 Linux 操作系统的计算机上运行以下命令：

```
$ 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。

## 1.9.5. 生成 SSH 私钥并将其添加到代理中

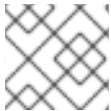
如果要在集群上执行安装调试或灾难恢复，则必须为 **ssh-agent** 和安装程序提供 SSH 密钥。您可以使用此密钥访问公共集群中的 bootstrap 机器来排除安装问题。



### 注意

在生产环境中，您需要进行灾难恢复和调试。

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。如果在您置备的基础架构上安装集群，您必须将此密钥提供给集群的机器。

## 1.9.6. 创建用于 AWS 的安装文件

要使用用户置备的基础架构在 Amazon Web Services (AWS) 上安装 OpenShift Container Platform，您必须生成并修改安装程序部署集群所需的文件，以便集群只创建要使用的机器。您要生成并自定义 **install-config.yaml** 文件、Kubernetes 清单和 Ignition 配置文件。您也可以选择在安装准备阶段首先设置独立的 **var** 分区。

### 1.9.6.1. 可选：创建独立 /var 分区

建议安装程序将 OpenShift Container Platform 的磁盘分区保留给安装程序。然而，在有些情况下您可能需要在文件系统的一部分中创建独立分区。

OpenShift Container Platform 支持添加单个分区来将存储附加到 **/var** 分区或 **/var** 的子目录。例如：

- **/var/lib/containers**：保存镜像相关的内容，随着更多镜像和容器添加到系统中，它所占用的存储会增加。
- **/var/lib/etcd**：保存您可能希望保持独立的数据，比如 etcd 存储的性能优化。
- **/var**：保存您希望独立保留的数据，用于特定目的（如审计）。

单独存储 **/var** 目录的内容可方便地根据需要对区域扩展存储，并可以在以后重新安装 OpenShift Container Platform 时保持该数据地完整。使用这个方法，您不必再次拉取所有容器，在更新系统时也无法复制大量日志文件。

因为 **/var** 在进行一个全新的 Red Hat Enterprise Linux CoreOS (RHCOS) 安装前必需存在，所以这个流程会在 OpenShift Container Platform 安装过程的 **openshift-install** 准备阶段插入的机器配置来设置独立的 **/var** 分区。



### 重要

如果按照以下步骤在此流程中创建独立 **/var** 分区，则不需要再次创建 Kubernetes 清单和 Ignition 配置文件，如本节所述。

### 流程

1. 创建存放 OpenShift Container Platform 安装文件的目录：

```
$ mkdir $HOME/clusterconfig
```

2. 运行 **openshift-install** 在 **manifest** 和 **openshift** 子目录中创建一组文件。在出现提示时回答系统问题：

```
$ openshift-install create manifests --dir $HOME/clusterconfig
```

### 输出示例

```
? SSH Public Key ...
INFO Credentials loaded from the "myprofile" profile in file "/home/myuser/.aws/credentials"
INFO Consuming Install Config from target directory
INFO Manifests created in: $HOME/clusterconfig/manifests and
$HOME/clusterconfig/openshift
```

3. 可选：确认安装程序在 **clusterconfig/openshift** 目录中创建了清单：

```
$ ls $HOME/clusterconfig/openshift/
```

### 输出示例

```
99_kubeadmin-password-secret.yaml
99_openshift-cluster-api_master-machines-0.yaml
99_openshift-cluster-api_master-machines-1.yaml
99_openshift-cluster-api_master-machines-2.yaml
...
```

4. 创建 **MachineConfig** 对象并将其添加到 **openshift** 目录中的一个文件中。例如，把文件命名为 **98-var-partition.yaml**，将磁盘设备名称改为 **worker** 系统中存储设备的名称，并根据情况设置存储大小。这个示例将 **/var** 目录放在一个单独的分区中：

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 98-var-partition
spec:
  config:
    ignition:
      version: 3.1.0
    storage:
      disks:
        - device: /dev/<device_name> 1
          partitions:
            - label: var
              startMiB: <partition_start_offset> 2
              sizeMiB: <partition_size> 3
          filesystems:
            - device: /dev/disk/by-partlabel/var
              path: /var
              format: xfs
      systemd:
        units:
          - name: var.mount 4
            enabled: true
            contents: |
              [Unit]
```

```
Before=local-fs.target
[Mount]
What=/dev/disk/by-partlabel/var
Where=/var
Options=defaults,prjquota 5
[Install]
WantedBy=local-fs.target
```

- 1 要分区的磁盘的存储设备名称。
- 2 当在引导磁盘中添加数据分区时，推荐最少使用 25000 MiB（Mebibytes）。root 文件系统会自动重新定义大小使其占据所有可用空间（最多到指定的偏移值）。如果没有指定值，或者指定的值小于推荐的最小值，则生成的 root 文件系统会太小，而在以后进行的 RHCOS 重新安装可能会覆盖数据分区的开始部分。
- 3 数据分区的大小（以兆字节为单位）。
- 4 挂载单元的名称必须与 **Where=** 指令中指定的目录匹配。例如，对于挂载于 **/var/lib/containers** 上的文件系统，该单元必须命名为 **var-lib-containers.mount**。
- 5 对于用于容器存储的文件系统，必须启用 **prjquota** 挂载选项。



### 注意

在创建独立 **/var** 分区时，如果不同的实例类型没有相同的设备名称，则无法将不同的实例类型用于 worker 节点。

5. 再次运行 **openshift-install**，从 **manifest** 和 **openshift** 子目录中的一组文件创建 Ignition 配置：

```
$ openshift-install create ignition-configs --dir $HOME/clusterconfig
$ ls $HOME/clusterconfig/
auth bootstrap.ign master.ign metadata.json worker.ign
```

现在，您可以使用 Ignition 配置文件作为安装程序的输入来安装 Red Hat Enterprise Linux CoreOS（RHCOS）系统。

### 1.9.6.2. 创建安装配置文件

生成并自定义安装程序部署集群所需的安装配置文件。

#### 先决条件

- 已获取 OpenShift Container Platform 安装程序用于用户置备的基础架构和集群的 pull secret。
- 使用红帽发布的附带 Red Hat Enterprise Linux CoreOS（RHCOS）AMI 检查您是否将集群部署到一个区域。如果您要部署到需要自定义 AMI 的区域，如 AWS GovCloud 区域，您必须手动创建 **install-config.yaml** 文件。

#### 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。

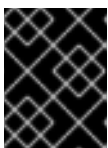
- ii. 选择 **aws** 作为目标平台。
- iii. 如果计算机上没有保存 AWS 配置集，请为您配置用于运行安装程序的用户输入 AWS 访问密钥 ID 和 secret 访问密钥。



### 注意

AWS 访问密钥 ID 和 secret 访问密钥存储在安装主机上当前用户主目录中的 **~/.aws/credentials** 中。如果文件中不存在导出的配置集凭证，安装程序会提示您输入凭证。您向安装程序提供的所有凭证都存储在文件中。

- iv. 选择要将集群部署到的 AWS 区域。
- v. 选择您为集群配置的 Route 53 服务的基域。
- vi. 为集群输入一个描述性名称。
- vii. 粘贴 [Red Hat OpenShift Cluster Manager](#) 中的 pull secret 。
2. 可选：备份 **install-config.yaml** 文件。



### 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

- 如需有关 AWS 配置集和凭证配置的更多信息，请参阅 [AWS 文档中的配置和凭证文件设置](#)。

#### 1.9.6.3. 在安装过程中配置集群范围代理

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

## 先决条件

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



## 注意

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

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

- 如果您的集群位于 AWS 上，请将 `ec2.<region>.amazonaws.com`、`elasticloadbalancing.<region>.amazonaws.com` 和 `s3.<region>.amazonaws.com` 端点添加到 VPC 端点。需要这些端点才能完成节点到 AWS EC2 API 的请求。由于代理在容器级别而不是节点级别工作，因此您必须通过 AWS 专用网络将这些请求路由到 AWS EC2 API。在代理服务器中的允许列表中添加 EC2 API 的公共 IP 地址是不够的。

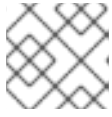
## 流程

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` 的配置映射来保存额外的 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，将为 **trustedCA** 参数指定的内容与 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。

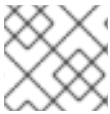


### 注意

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**。



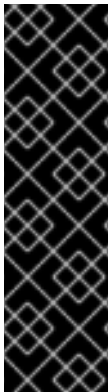
### 注意

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

#### 1.9.6.4. 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 Ignition 配置文件生成后的 12 小时内使用它们，因为 24 小时的证书会在集群安装后的 16 小时到 22 小时进行轮转。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因为执行了证书更新而导致安装失败的问题。

#### 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：



```
$ rm -f <installation_directory>/openshift/99_openshift-cluster-api_master-machines-*.yaml
```

通过删除这些文件，您可以防止集群自动生成 control plane 机器。

- 删除定义 worker 机器的 Kubernetes 清单文件：

```
$ rm -f <installation_directory>/openshift/99_openshift-cluster-api_worker-machineset-*.yaml
```

由于您要自行创建并管理 worker 机器，因此不需要初始化这些机器。

- 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：
  - 打开 `<installation_directory>/manifests/cluster-scheduler-02-config.yml` 文件。
  - 找到 `mastersSchedulable` 参数并确保它被设置为 `false`。
  - 保存并退出文件。
- 可选：如果您不希望 [Ingress Operator](#) 代表您创建 DNS 记录，请删除 `<installation_directory>/manifests/cluster-dns-02-config.yml` DNS 配置文件中的 `privateZone` 和 `publicZone` 部分：

```
apiVersion: config.openshift.io/v1
kind: DNS
metadata:
  creationTimestamp: null
  name: cluster
spec:
  baseDomain: example.openshift.com
  privateZone: ❶
    id: mycluster-100419-private-zone
  publicZone: ❷
    id: example.openshift.com
status: {}
```

❶ ❷ 完全删除此部分。

如果您这样做，后续步骤中必须手动添加入口 DNS 记录。

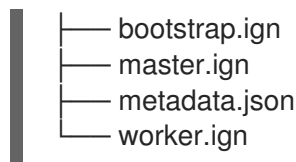
- 要创建 Ignition 配置文件，从包含安装程序的目录运行以下命令：

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

❶ 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
```



### 1.9.7. 提取基础架构名称

Ignition 配置文件包含一个唯一集群标识符，您可以使用它在 Amazon Web Services (AWS) 中唯一地标识您的集群。基础架构名称还用于在 OpenShift Container Platform 安装过程中定位适当的 AWS 资源。提供的 CloudFormation 模板包含对此基础架构名称的引用，因此您必须提取它。

#### 先决条件

- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。
- 已为集群生成 Ignition 配置文件。
- 安装了 **jq** 软件包。

#### 流程

- 要从 Ignition 配置文件元数据中提取和查看基础架构名称，请运行以下命令：

```
$ jq -r .infraID <installation_directory>/metadata.json 1
```

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

#### 输出示例

```
openshift-vw9j6 1
```

- 1** 此命令的输出是您的集群名称和随机字符串。

### 1.9.8. 在 AWS 中创建 VPC

您必须在 Amazon Web Services (AWS) 中创建 Virtual Private Cloud (VPC)，供您的 OpenShift Container Platform 集群使用。您可以自定义 VPC 来满足您的要求，包括 VPN 和路由表。

您可以使用提供的 CloudFormation 模板和自定义参数文件创建代表 VPC 的 AWS 资源堆栈。



#### 注意

如果不使用提供的 CloudFormation 模板来创建 AWS 基础架构，您必须检查提供的信息并手动创建基础架构。如果集群没有正确初始化，您可能需要联系红帽支持并提供您的安装日志。

#### 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。

- 已为集群生成 Ignition 配置文件。

## 流程

1. 创建一个 JSON 文件，其包含模板所需的参数值：

```
[
  {
    "ParameterKey": "VpcCidr", ❶
    "ParameterValue": "10.0.0.0/16" ❷
  },
  {
    "ParameterKey": "AvailabilityZoneCount", ❸
    "ParameterValue": "1" ❹
  },
  {
    "ParameterKey": "SubnetBits", ❺
    "ParameterValue": "12" ❻
  }
]
```

- ❶ VPC 的 CIDR 块。
- ❷ 以 **x.x.x.x/16-24** 格式指定 CIDR 块。
- ❸ 在其中部署 VPC 的可用区的数量。
- ❹ 指定一个 **1** 到 **3** 之间的整数。
- ❺ 各个可用区中每个子网的大小。
- ❻ 指定 **5** 到 **13** 之间的整数，其中 **5** 为 /27，**13** 为 /19。

2. 复制本主题的 VPC 的 CloudFormation 模板部分中的模板，并将它以 YAML 文件形式保存到计算机上。此模板描述了集群所需的 VPC。
3. 启动 CloudFormation 模板，以创建代表 VPC 的 AWS 资源堆栈：



### 重要

您必须在一行内输入命令。

```
$ aws cloudformation create-stack --stack-name <name> ❶
  --template-body file://<template>.yaml ❷
  --parameters file://<parameters>.json ❸
```

- ❶ **<name>** 是 CloudFormation 堆栈的名称，如 **cluster-VPC**。如果您删除集群，则需要此堆栈的名称。
- ❷ **<template>** 是您保存的 CloudFormation 模板 YAML 文件的相对路径和名称。
- ❸ **<parameters>** 是 CloudFormation 参数 JSON 文件的相对路径和名称。

## 输出示例

```
arn:aws:cloudformation:us-east-1:269333783861:stack/cluster-vpc/dbedae40-2fd3-11eb-820e-12a48460849f
```

### 4. 确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <name>
```

在 **StackStatus** 显示 **CREATE\_COMPLETE** 后，输出会显示以下参数的值。您必须将这些参数值提供给您在创建集群时要运行的其他 CloudFormation 模板：

<b>VpcId</b>	您的 VPC ID。
<b>PublicSubnetIds</b>	新公共子网的 ID。
<b>PrivateSubnetIds</b>	新专用子网的 ID。

### 1.9.8.1. VPC 的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署 OpenShift Container Platform 集群所需的 VPC。

#### 例 1.25. VPC 的 CloudFormation 模板

```
AWSTemplateFormatVersion: 2010-09-09
Description: Template for Best Practice VPC with 1-3 AZs

Parameters:
  VpcCidr:
    AllowedPattern: ^((([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])|(1[6-9]|2[0-4]))$
    ConstraintDescription: CIDR block parameter must be in the form x.x.x.x/16-24.
    Default: 10.0.0.0/16
    Description: CIDR block for VPC.
    Type: String
  AvailabilityZoneCount:
    ConstraintDescription: "The number of availability zones. (Min: 1, Max: 3)"
    MinValue: 1
    MaxValue: 3
    Default: 1
    Description: "How many AZs to create VPC subnets for. (Min: 1, Max: 3)"
    Type: Number
  SubnetBits:
    ConstraintDescription: CIDR block parameter must be in the form x.x.x.x/19-27.
    MinValue: 5
    MaxValue: 13
    Default: 12
    Description: "Size of each subnet to create within the availability zones. (Min: 5 = /27, Max: 13 = /19)"
    Type: Number
```

## Metadata:

AWS::CloudFormation::Interface:

ParameterGroups:

- Label:

default: "Network Configuration"

Parameters:

- VpcCidr

- SubnetBits

- Label:

default: "Availability Zones"

Parameters:

- AvailabilityZoneCount

ParameterLabels:

AvailabilityZoneCount:

default: "Availability Zone Count"

VpcCidr:

default: "VPC CIDR"

SubnetBits:

default: "Bits Per Subnet"

## Conditions:

DoAz3: !Equals [3, !Ref AvailabilityZoneCount]

DoAz2: !Or [!Equals [2, !Ref AvailabilityZoneCount], Condition: DoAz3]

## Resources:

VPC:

Type: "AWS::EC2::VPC"

Properties:

EnableDnsSupport: "true"

EnableDnsHostnames: "true"

CidrBlock: !Ref VpcCidr

PublicSubnet:

Type: "AWS::EC2::Subnet"

Properties:

VpcId: !Ref VPC

CidrBlock: !Select [0, !Cidr [!Ref VpcCidr, 6, !Ref SubnetBits]]

AvailabilityZone: !Select

- 0

- Fn::GetAZs: !Ref "AWS::Region"

PublicSubnet2:

Type: "AWS::EC2::Subnet"

Condition: DoAz2

Properties:

VpcId: !Ref VPC

CidrBlock: !Select [1, !Cidr [!Ref VpcCidr, 6, !Ref SubnetBits]]

AvailabilityZone: !Select

- 1

- Fn::GetAZs: !Ref "AWS::Region"

PublicSubnet3:

Type: "AWS::EC2::Subnet"

Condition: DoAz3

Properties:

VpcId: !Ref VPC

CidrBlock: !Select [2, !Cidr [!Ref VpcCidr, 6, !Ref SubnetBits]]

AvailabilityZone: !Select

```
- 2
- Fn::GetAZs: !Ref "AWS::Region"
InternetGateway:
  Type: "AWS::EC2::InternetGateway"
GatewayToInternet:
  Type: "AWS::EC2::VPCGatewayAttachment"
  Properties:
    Vpclid: !Ref VPC
    InternetGatewayId: !Ref InternetGateway
PublicRouteTable:
  Type: "AWS::EC2::RouteTable"
  Properties:
    Vpclid: !Ref VPC
PublicRoute:
  Type: "AWS::EC2::Route"
  DependsOn: GatewayToInternet
  Properties:
    RouteTableId: !Ref PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway
PublicSubnetRouteTableAssociation:
  Type: "AWS::EC2::SubnetRouteTableAssociation"
  Properties:
    SubnetId: !Ref PublicSubnet
    RouteTableId: !Ref PublicRouteTable
PublicSubnetRouteTableAssociation2:
  Type: "AWS::EC2::SubnetRouteTableAssociation"
  Condition: DoAz2
  Properties:
    SubnetId: !Ref PublicSubnet2
    RouteTableId: !Ref PublicRouteTable
PublicSubnetRouteTableAssociation3:
  Condition: DoAz3
  Type: "AWS::EC2::SubnetRouteTableAssociation"
  Properties:
    SubnetId: !Ref PublicSubnet3
    RouteTableId: !Ref PublicRouteTable
PrivateSubnet:
  Type: "AWS::EC2::Subnet"
  Properties:
    Vpclid: !Ref VPC
    CidrBlock: !Select [3, !Cidr [!Ref VpcCidr, 6, !Ref SubnetBits]]
    AvailabilityZone: !Select
- 0
- Fn::GetAZs: !Ref "AWS::Region"
PrivateRouteTable:
  Type: "AWS::EC2::RouteTable"
  Properties:
    Vpclid: !Ref VPC
PrivateSubnetRouteTableAssociation:
  Type: "AWS::EC2::SubnetRouteTableAssociation"
  Properties:
    SubnetId: !Ref PrivateSubnet
    RouteTableId: !Ref PrivateRouteTable
NAT:
  DependsOn:
```

```

- GatewayToInternet
Type: "AWS::EC2::NatGateway"
Properties:
  AllocationId:
    "Fn::GetAtt":
      - EIP
      - AllocationId
  SubnetId: !Ref PublicSubnet
EIP:
Type: "AWS::EC2::EIP"
Properties:
  Domain: vpc
Route:
Type: "AWS::EC2::Route"
Properties:
  RouteTableId:
    Ref: PrivateRouteTable
  DestinationCidrBlock: 0.0.0.0/0
  NatGatewayId:
    Ref: NAT
PrivateSubnet2:
Type: "AWS::EC2::Subnet"
Condition: DoAz2
Properties:
  Vpclid: !Ref VPC
  CidrBlock: !Select [4, !Cidr [!Ref VpcCidr, 6, !Ref SubnetBits]]
  AvailabilityZone: !Select
    - 1
    - Fn::GetAZs: !Ref "AWS::Region"
PrivateRouteTable2:
Type: "AWS::EC2::RouteTable"
Condition: DoAz2
Properties:
  Vpclid: !Ref VPC
PrivateSubnetRouteTableAssociation2:
Type: "AWS::EC2::SubnetRouteTableAssociation"
Condition: DoAz2
Properties:
  SubnetId: !Ref PrivateSubnet2
  RouteTableId: !Ref PrivateRouteTable2
NAT2:
DependsOn:
- GatewayToInternet
Type: "AWS::EC2::NatGateway"
Condition: DoAz2
Properties:
  AllocationId:
    "Fn::GetAtt":
      - EIP2
      - AllocationId
  SubnetId: !Ref PublicSubnet2
EIP2:
Type: "AWS::EC2::EIP"
Condition: DoAz2
Properties:
  Domain: vpc

```

```
Route2:
  Type: "AWS::EC2::Route"
  Condition: DoAz2
  Properties:
    RouteTableId:
      Ref: PrivateRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId:
      Ref: NAT2
PrivateSubnet3:
  Type: "AWS::EC2::Subnet"
  Condition: DoAz3
  Properties:
    VpcId: !Ref VPC
    CidrBlock: !Select [5, !Cidr [!Ref VpcCidr, 6, !Ref SubnetBits]]
    AvailabilityZone: !Select
      - 2
      - Fn::GetAZs: !Ref "AWS::Region"
PrivateRouteTable3:
  Type: "AWS::EC2::RouteTable"
  Condition: DoAz3
  Properties:
    VpcId: !Ref VPC
PrivateSubnetRouteTableAssociation3:
  Type: "AWS::EC2::SubnetRouteTableAssociation"
  Condition: DoAz3
  Properties:
    SubnetId: !Ref PrivateSubnet3
    RouteTableId: !Ref PrivateRouteTable3
NAT3:
  DependsOn:
    - GatewayToInternet
  Type: "AWS::EC2::NatGateway"
  Condition: DoAz3
  Properties:
    AllocationId:
      "Fn::GetAtt":
        - EIP3
        - AllocationId
    SubnetId: !Ref PublicSubnet3
EIP3:
  Type: "AWS::EC2::EIP"
  Condition: DoAz3
  Properties:
    Domain: vpc
Route3:
  Type: "AWS::EC2::Route"
  Condition: DoAz3
  Properties:
    RouteTableId:
      Ref: PrivateRouteTable3
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId:
      Ref: NAT3
S3Endpoint:
  Type: AWS::EC2::VPCEndpoint
```



```

Properties:
  PolicyDocument:
    Version: 2012-10-17
    Statement:
      - Effect: Allow
        Principal: '*'
        Action:
          - '*'
        Resource:
          - '*'

RouteTableIds:
- !Ref PublicRouteTable
- !Ref PrivateRouteTable
- !If [DoAz2, !Ref PrivateRouteTable2, !Ref "AWS::NoValue"]
- !If [DoAz3, !Ref PrivateRouteTable3, !Ref "AWS::NoValue"]
ServiceName: !Join
- "
- - com.amazonaws.
- !Ref 'AWS::Region'
- .s3
VpId: !Ref VPC

Outputs:
  VpId:
    Description: ID of the new VPC.
    Value: !Ref VPC
  PublicSubnetIds:
    Description: Subnet IDs of the public subnets.
    Value:
      !Join [
        ",",
        [!Ref PublicSubnet, !If [DoAz2, !Ref PublicSubnet2, !Ref "AWS::NoValue"], !If [DoAz3, !Ref
PublicSubnet3, !Ref "AWS::NoValue"]]
      ]
  PrivateSubnetIds:
    Description: Subnet IDs of the private subnets.
    Value:
      !Join [
        ",",
        [!Ref PrivateSubnet, !If [DoAz2, !Ref PrivateSubnet2, !Ref "AWS::NoValue"], !If [DoAz3, !Ref
PrivateSubnet3, !Ref "AWS::NoValue"]]
      ]

```

## 其他资源

- 您可以通过导航 [AWS CloudFormation 控制台](#) 来查看您创建的 CloudFormation 堆栈的详情。

### 1.9.9. 在 AWS 中创建网络和负载均衡组件

您必须在 OpenShift Container Platform 集群可以使用的 Amazon Web Services (AWS) 中配置网络、经典或网络负载均衡。

您可以使用提供的 CloudFormation 模板和自定义参数文件来创建 AWS 资源堆栈。堆栈代表 OpenShift Container Platform 集群所需的网络和负载均衡组件。该模板还创建一个托管区和子网标签。

您可以在单一虚拟私有云(VPC)内多次运行该模板。



### 注意

如果不使用提供的 CloudFormation 模板来创建 AWS 基础架构，您必须检查提供的信息并手动创建基础架构。如果集群没有正确初始化，您可能需要联系红帽支持并提供您的安装日志。

### 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。
- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。

### 流程

1. 获取您在 **install-config.yaml** 文件中为集群指定的 Route 53 基域的托管区 ID。您可以运行以下命令来获取托管区的详细信息：

```
$ aws route53 list-hosted-zones-by-name --dns-name <route53_domain> 1
```

- 1 对于 **<route53\_domain>**，请指定您为集群生成 **install-config.yaml** 文件时所用的 Route53 基域。

### 输出示例

```
mycluster.example.com. False 100
HOSTEDZONES 65F8F38E-2268-B835-E15C-AB55336FCBFA
/hostedzone/Z21IXYZABCZ2A4 mycluster.example.com. 10
```

在示例输出中，托管区 ID 为 **Z21IXYZABCZ2A4**。

2. 创建一个 JSON 文件，其包含模板所需的参数值：

```
[
  {
    "ParameterKey": "ClusterName", 1
    "ParameterValue": "mycluster" 2
  },
  {
    "ParameterKey": "InfrastructureName", 3
    "ParameterValue": "mycluster-<random_string>" 4
  },
  {
    "ParameterKey": "HostedZoneId", 5
    "ParameterValue": "<random_string>" 6
  },
  {
    "ParameterKey": "HostedZoneName", 7
```

```

    "ParameterValue": "example.com" 8
  },
  {
    "ParameterKey": "PublicSubnets", 9
    "ParameterValue": "subnet-<random_string>" 10
  },
  {
    "ParameterKey": "PrivateSubnets", 11
    "ParameterValue": "subnet-<random_string>" 12
  },
  {
    "ParameterKey": "VpcId", 13
    "ParameterValue": "vpc-<random_string>" 14
  }
]

```

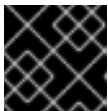
- 1 一个简短的、代表集群的名称用于主机名等。
  - 2 指定您为集群生成 `install-config.yaml` 文件时所用的集群名称。
  - 3 您的 Ignition 配置文件中为集群编码的集群基础架构名称。
  - 4 指定从 Ignition 配置文件元数据中提取的基础架构名称，其格式为 `<cluster-name>-<random-string>`。
  - 5 用来注册目标的 Route 53 公共区 ID。
  - 6 指定 Route 53 公共区 ID，其格式与 `Z21IXYZABCZ2A4` 类似。您可以从 AWS 控制台获取这个值。
  - 7 用来注册目标的 Route 53 区。
  - 8 指定您为集群生成 `install-config.yaml` 文件时所用的 Route 53 基域。请勿包含 AWS 控制台中显示的结尾句点 (.)。
  - 9 为 VPC 创建的公共子网。
  - 10 指定 VPC 的 CloudFormation 模板输出的 `PublicSubnetIds` 值。
  - 11 为 VPC 创建的专用子网。
  - 12 指定 VPC 的 CloudFormation 模板输出的 `PrivateSubnetIds` 值。
  - 13 为集群创建的 VPC。
  - 14 指定 VPC 的 CloudFormation 模板输出的 `VpcId` 值。
3. 复制本主题的网络和负载均衡器的 **CloudFormation** 模板部分中的模板，并将它以 **YAML** 文件格式保存到计算机上。此模板描述了集群所需的网络和负载均衡对象。



### 重要

如果要部署集群到 AWS 政府区域，您必须更新 CloudFormation 模板中的 `InternalApiServerRecord`，以使用 **CNAME** 记录。AWS 政府区不支持 **ALIAS** 类型的记录。

4. 启动 CloudFormation 模板，以创建 AWS 资源堆栈，该堆栈提供网络和负载均衡组件：



### 重要

您必须在一行内输入命令。

```
$ aws cloudformation create-stack --stack-name <name> ❶
  --template-body file://<template>.yaml ❷
  --parameters file://<parameters>.json ❸
  --capabilities CAPABILITY_NAMED_IAM ❹
```

- ❶ **<name>** 是 CloudFormation 堆栈的名称，如 **cluster-dns**。如果您删除集群，则需要此堆栈的名称。
- ❷ **<template>** 是您保存的 CloudFormation 模板 YAML 文件的相对路径和名称。
- ❸ **<parameters>** 是 CloudFormation 参数 JSON 文件的相对路径和名称。
- ❹ 您必须明确声明 **CAPABILITY\_NAMED\_IAM** 功能，因为提供的模板会创建一些 **AWS::IAM::Role** 资源。

### 输出示例

```
arn:aws:cloudformation:us-east-1:269333783861:stack/cluster-dns/cd3e5de0-2fd4-11eb-5cf0-12be5c33a183
```

5. 确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <name>
```

在 **StackStatus** 显示 **CREATE\_COMPLETE** 后，输出会显示以下参数的值。您必须将这些参数值提供给您在创建集群时要运行的其他 CloudFormation 模板：

<b>PrivateHostedZoneId</b>	专用 DNS 的托管区 ID。
<b>ExternalApiLoadBalancerName</b>	外部 API 负载均衡器的完整名称。
<b>InternalApiLoadBalancerName</b>	内部 API 负载均衡器的完整名称。
<b>ApiServerDnsName</b>	API 服务器的完整主机名。

<b>RegisterNLbpTargetsLambda</b>	有助于为这些负载均衡器注册/撤销注册 IP 目标的 Lambda ARN。
<b>ExternalAPITargetGroupArn</b>	外部 API 目标组的 ARN。
<b>InternalAPITargetGroupArn</b>	内部 API 目标组的 ARN。
<b>InternalServiceTargetGroupArn</b>	内部服务目标组群的 ARN。

### 1.9.9.1. 网络和负载均衡器的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署 OpenShift Container Platform 集群所需的网络对象和负载均衡器。

#### 例 1.26. 网络和负载均衡器的 CloudFormation 模板

```

AWSTemplateFormatVersion: 2010-09-09
Description: Template for OpenShift Cluster Network Elements (Route53 & LBs)

Parameters:
  ClusterName:
    AllowedPattern: ^([a-zA-Z][a-zA-Z0-9\-\_]{0,26})$
    MaxLength: 27
    MinLength: 1
    ConstraintDescription: Cluster name must be alphanumeric, start with a letter, and have a
maximum of 27 characters.
    Description: A short, representative cluster name to use for host names and other identifying
names.
    Type: String
  InfrastructureName:
    AllowedPattern: ^([a-zA-Z][a-zA-Z0-9\-\_]{0,26})$
    MaxLength: 27
    MinLength: 1
    ConstraintDescription: Infrastructure name must be alphanumeric, start with a letter, and have a
maximum of 27 characters.
    Description: A short, unique cluster ID used to tag cloud resources and identify items owned or
used by the cluster.
    Type: String
  HostedZoneId:
    Description: The Route53 public zone ID to register the targets with, such as
Z21IXYZABCZ2A4.
    Type: String
  HostedZoneName:

```

Description: The Route53 zone to register the targets with, such as example.com. Omit the trailing period.

Type: String

Default: "example.com"

PublicSubnets:

Description: The internet-facing subnets.

Type: List<AWS::EC2::Subnet::Id>

PrivateSubnets:

Description: The internal subnets.

Type: List<AWS::EC2::Subnet::Id>

VpcId:

Description: The VPC-scoped resources will belong to this VPC.

Type: AWS::EC2::VPC::Id

Metadata:

AWS::CloudFormation::Interface:

ParameterGroups:

- Label:

default: "Cluster Information"

Parameters:

- ClusterName

- InfrastructureName

- Label:

default: "Network Configuration"

Parameters:

- VpcId

- PublicSubnets

- PrivateSubnets

- Label:

default: "DNS"

Parameters:

- HostedZoneName

- HostedZoneId

ParameterLabels:

ClusterName:

default: "Cluster Name"

InfrastructureName:

default: "Infrastructure Name"

VpcId:

default: "VPC ID"

PublicSubnets:

default: "Public Subnets"

PrivateSubnets:

default: "Private Subnets"

HostedZoneName:

default: "Public Hosted Zone Name"

HostedZoneId:

default: "Public Hosted Zone ID"

Resources:

ExtApiElb:

Type: AWS::ElasticLoadBalancingV2::LoadBalancer

Properties:

Name: !Join ["-", [!Ref InfrastructureName, "ext"]]

IpAddressType: ipv4

Subnets: !Ref PublicSubnets

Type: network

IntApiElb:

Type: AWS::ElasticLoadBalancingV2::LoadBalancer

Properties:

Name: !Join ["-", [!Ref InfrastructureName, "int"]]

Scheme: internal

IpAddressType: ipv4

Subnets: !Ref PrivateSubnets

Type: network

IntDns:

Type: "AWS::Route53::HostedZone"

Properties:

HostedZoneConfig:

Comment: "Managed by CloudFormation"

Name: !Join [".", [!Ref ClusterName, !Ref HostedZoneName]]

HostedZoneTags:

- Key: Name

Value: !Join ["-", [!Ref InfrastructureName, "int"]]

- Key: !Join [""], ["kubernetes.io/cluster/", !Ref InfrastructureName]]

Value: "owned"

VPCs:

- VPCId: !Ref Vpclid

VPCRegion: !Ref "AWS::Region"

ExternalApiServerRecord:

Type: AWS::Route53::RecordSetGroup

Properties:

Comment: Alias record for the API server

HostedZoneId: !Ref HostedZoneId

RecordSets:

- Name:

!Join [

".",

["api", !Ref ClusterName, !Join [""], [!Ref HostedZoneName, "."]]],

]

Type: A

AliasTarget:

HostedZoneId: !GetAtt ExtApiElb.CanonicalHostedZoneID

DNSName: !GetAtt ExtApiElb.DNSName

InternalApiServerRecord:

Type: AWS::Route53::RecordSetGroup

Properties:

Comment: Alias record for the API server

HostedZoneId: !Ref IntDns

RecordSets:

- Name:

!Join [

".",

["api", !Ref ClusterName, !Join [""], [!Ref HostedZoneName, "."]]],

]

Type: A

AliasTarget:

HostedZoneId: !GetAtt IntApiElb.CanonicalHostedZoneID

```
DNSName: !GetAtt IntApiElb.DNSName
- Name:
  !Join [
    ".",
    ["api-int", !Ref ClusterName, !Join ["", [!Ref HostedZoneName, "."]]],
  ]
Type: A
AliasTarget:
  HostedZoneId: !GetAtt IntApiElb.CanonicalHostedZoneID
  DNSName: !GetAtt IntApiElb.DNSName
```

```
ExternalApiListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn:
          Ref: ExternalApiTargetGroup
    LoadBalancerArn:
      Ref: ExtApiElb
    Port: 6443
    Protocol: TCP
```

```
ExternalApiTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 10
    HealthCheckPath: "/readyz"
    HealthCheckPort: 6443
    HealthCheckProtocol: HTTPS
    HealthyThresholdCount: 2
    UnhealthyThresholdCount: 2
    Port: 6443
    Protocol: TCP
    TargetType: ip
    VpcId:
      Ref: VpcId
    TargetGroupAttributes:
      - Key: deregistration_delay.timeout_seconds
        Value: 60
```

```
InternalApiListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn:
          Ref: InternalApiTargetGroup
    LoadBalancerArn:
      Ref: IntApiElb
    Port: 6443
    Protocol: TCP
```

```
InternalApiTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
```



HealthCheckIntervalSeconds: 10  
HealthCheckPath: "/readyz"  
HealthCheckPort: 6443  
HealthCheckProtocol: HTTPS  
HealthyThresholdCount: 2  
UnhealthyThresholdCount: 2  
Port: 6443  
Protocol: TCP  
TargetType: ip  
VpcId:  
Ref: VpcId  
TargetGroupAttributes:  
- Key: deregistration\_delay.timeout\_seconds  
Value: 60

InternalServiceInternalListener:  
Type: AWS::ElasticLoadBalancingV2::Listener  
Properties:  
DefaultActions:  
- Type: forward  
TargetGroupArn:  
Ref: InternalServiceTargetGroup  
LoadBalancerArn:  
Ref: IntApiElb  
Port: 22623  
Protocol: TCP

InternalServiceTargetGroup:  
Type: AWS::ElasticLoadBalancingV2::TargetGroup  
Properties:  
HealthCheckIntervalSeconds: 10  
HealthCheckPath: "/healthz"  
HealthCheckPort: 22623  
HealthCheckProtocol: HTTPS  
HealthyThresholdCount: 2  
UnhealthyThresholdCount: 2  
Port: 22623  
Protocol: TCP  
TargetType: ip  
VpcId:  
Ref: VpcId  
TargetGroupAttributes:  
- Key: deregistration\_delay.timeout\_seconds  
Value: 60

RegisterTargetLambdalamRole:  
Type: AWS::IAM::Role  
Properties:  
RoleName: !Join ["-", [!Ref InfrastructureName, "nlb", "lambda", "role"]]  
AssumeRolePolicyDocument:  
Version: "2012-10-17"  
Statement:  
- Effect: "Allow"  
Principal:  
Service:  
- "lambda.amazonaws.com"

```

    Action:
      - "sts:AssumeRole"
    Path: "/"
    Policies:
      - PolicyName: !Join ["-", [!Ref InfrastructureName, "master", "policy"]]
    PolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: "Allow"
          Action:
            [
              "elasticloadbalancing:RegisterTargets",
              "elasticloadbalancing:DeregisterTargets",
            ]
          Resource: !Ref InternalApiTargetGroup
        - Effect: "Allow"
          Action:
            [
              "elasticloadbalancing:RegisterTargets",
              "elasticloadbalancing:DeregisterTargets",
            ]
          Resource: !Ref InternalServiceTargetGroup
        - Effect: "Allow"
          Action:
            [
              "elasticloadbalancing:RegisterTargets",
              "elasticloadbalancing:DeregisterTargets",
            ]
          Resource: !Ref ExternalApiTargetGroup

RegisterNlbPTargets:
  Type: "AWS::Lambda::Function"
  Properties:
    Handler: "index.handler"
    Role:
      Fn::GetAtt:
        - "RegisterTargetLambdalamRole"
        - "Arn"
    Code:
      ZipFile: |
        import json
        import boto3
        import cfnresponse
        def handler(event, context):
            elb = boto3.client('elbv2')
            if event['RequestType'] == 'Delete':
                elb.deregister_targets(TargetGroupArn=event['ResourceProperties']
[TargetArn],Targets=[{'Id': event['ResourceProperties']['TargetIp']})
            elif event['RequestType'] == 'Create':
                elb.register_targets(TargetGroupArn=event['ResourceProperties']['TargetArn'],Targets=
[{'Id': event['ResourceProperties']['TargetIp']})
                responseData = {}
                cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
event['ResourceProperties']['TargetArn']+event['ResourceProperties']['TargetIp'])
      Runtime: "python3.7"
      Timeout: 120

```

RegisterSubnetTagsLambdaRole:

Type: AWS::IAM::Role

Properties:

RoleName: !Join ["-", [!Ref InfrastructureName, "subnet-tags-lambda-role"]]

AssumeRolePolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: "Allow"

Principal:

Service:

- "lambda.amazonaws.com"

Action:

- "sts:AssumeRole"

Path: "/"

Policies:

- PolicyName: !Join ["-", [!Ref InfrastructureName, "subnet-tagging-policy"]]

PolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: "Allow"

Action:

```
[
  "ec2:DeleteTags",
  "ec2:CreateTags"
]
```

Resource: "arn:aws:ec2:\*:\*:subnet/\*"

- Effect: "Allow"

Action:

```
[
  "ec2:DescribeSubnets",
  "ec2:DescribeTags"
]
```

Resource: ""

RegisterSubnetTags:

Type: "AWS::Lambda::Function"

Properties:

Handler: "index.handler"

Role:

Fn::GetAtt:

- "RegisterSubnetTagsLambdaRole"

- "Arn"

Code:

ZipFile: |

```
import json
```

```
import boto3
```

```
import cfnresponse
```

```
def handler(event, context):
```

```
    ec2_client = boto3.client('ec2')
```

```
    if event['RequestType'] == 'Delete':
```

```
        for subnet_id in event['ResourceProperties']['Subnets']:
```

```
            ec2_client.delete_tags(Resources=[subnet_id], Tags=[{'Key': 'kubernetes.io/cluster/' +
event['ResourceProperties']['InfrastructureName']}]);
```

```
    elif event['RequestType'] == 'Create':
```

```
        for subnet_id in event['ResourceProperties']['Subnets']:
```

```
    ec2_client.create_tags(Resources=[subnet_id], Tags=[{'Key': 'kubernetes.io/cluster/' +
event['ResourceProperties']['InfrastructureName'], 'Value': 'shared'}]);
    responseData = {}
    cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
event['ResourceProperties']['InfrastructureName']+event['ResourceProperties']['Subnets'][0])
    Runtime: "python3.7"
    Timeout: 120
```

**RegisterPublicSubnetTags:**

Type: Custom::SubnetRegister

Properties:

ServiceToken: !GetAtt RegisterSubnetTags.Arn

InfrastructureName: !Ref InfrastructureName

Subnets: !Ref PublicSubnets

**RegisterPrivateSubnetTags:**

Type: Custom::SubnetRegister

Properties:

ServiceToken: !GetAtt RegisterSubnetTags.Arn

InfrastructureName: !Ref InfrastructureName

Subnets: !Ref PrivateSubnets

**Outputs:****PrivateHostedZoneId:**

Description: Hosted zone ID for the private DNS, which is required for private records.

Value: !Ref IntDns

**ExternalApiLoadBalancerName:**

Description: Full name of the external API load balancer.

Value: !GetAtt ExtApiElb.LoadBalancerFullName

**InternalApiLoadBalancerName:**

Description: Full name of the internal API load balancer.

Value: !GetAtt IntApiElb.LoadBalancerFullName

**ApiServerDnsName:**

Description: Full hostname of the API server, which is required for the Ignition config files.

Value: !Join [".", ["api-int", !Ref ClusterName, !Ref HostedZoneName]]

**RegisterNlbIpTargetsLambda:**

Description: Lambda ARN useful to help register or deregister IP targets for these load balancers.

Value: !GetAtt RegisterNlbIpTargets.Arn

**ExternalApiTargetGroupArn:**

Description: ARN of the external API target group.

Value: !Ref ExternalApiTargetGroup

**InternalApiTargetGroupArn:**

Description: ARN of the internal API target group.

Value: !Ref InternalApiTargetGroup

**InternalServiceTargetGroupArn:**

Description: ARN of the internal service target group.

Value: !Ref InternalServiceTargetGroup



## 重要

如果要部署到 AWS 政府区域，您必须更新 `InternalApiServerRecord` 以使用 `CNAME` 记录。AWS 政府区不支持 `ALIAS` 类型的记录。例如：

```
Type: CNAME
TTL: 10
ResourceRecords:
- !GetAtt IntApiElb.DNSName
```

## 其他资源

- 您可以通过导航 [AWS CloudFormation 控制台](#) 来查看您创建的 CloudFormation 堆栈的详情。
- 您可以通过导航到 [AWS Route 53 控制台](#) 来查看托管区的详情。
- 有关列出公共托管区的更多信息，请参阅 AWS 文档中的 [列出公共托管区](#)。

## 1.9.10. 在 AWS 中创建安全组和角色

您必须在 Amazon Web Services (AWS) 中创建安全组和角色，供您的 OpenShift Container Platform 集群使用。

您可以使用提供的 CloudFormation 模板和自定义参数文件来创建 AWS 资源堆栈。堆栈代表 OpenShift Container Platform 集群所需的安全组和角色。



## 注意

如果不使用提供的 CloudFormation 模板来创建 AWS 基础架构，您必须检查提供的信息并手动创建基础架构。如果集群没有正确初始化，您可能需要联系红帽支持并提供您的安装日志。

## 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 `aws configure`，将 AWS 密钥和区域添加到本地 AWS 配置集中。
- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。

## 流程

1. 创建一个 JSON 文件，其包含模板所需的参数值：

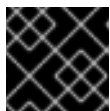
```
[
  {
    "ParameterKey": "InfrastructureName", 1
    "ParameterValue": "mycluster-<random_string>" 2
  },
  {
    "ParameterKey": "VpcCidr", 3
    "ParameterValue": "10.0.0.0/16" 4
  }
]
```

```

    },
    {
      "ParameterKey": "PrivateSubnets", ❸
      "ParameterValue": "subnet-<random_string>" ❹
    },
    {
      "ParameterKey": "VpcId", ❺
      "ParameterValue": "vpc-<random_string>" ❻
    }
  ]

```

- ❶ 您的 Ignition 配置文件中为集群编码的集群基础架构名称。
  - ❷ 指定从 Ignition 配置文件元数据中提取的基础架构名称，其格式为 **<cluster-name>-<random-string>**。
  - ❸ VPC 的 CIDR 块。
  - ❹ 指定以 **x.x.x.x/16-24** 格式定义的用于 VPC 的 CIDR 地址块。
  - ❺ 为 VPC 创建的专用子网。
  - ❻ 指定 VPC 的 CloudFormation 模板输出的 **PrivateSubnetIds** 值。
  - ❼ 为集群创建的 VPC。
  - ❽ 指定 VPC 的 CloudFormation 模板输出的 **VpcId** 值。
2. 复制本主题的安全对象的 **CloudFormation** 模板部分中的模板，并将它以 YAML 文件形式保存到计算机上。此模板描述了集群所需的安全组和角色。
  3. 启动 CloudFormation 模板，以创建代表安全组和角色的 AWS 资源堆栈：



### 重要

您必须在一行内输入命令。

```

$ aws cloudformation create-stack --stack-name <name> ❶
  --template-body file://<template>.yaml ❷
  --parameters file://<parameters>.json ❸
  --capabilities CAPABILITY_NAMED_IAM ❹

```

- ❶ **<name>** 是 CloudFormation 堆栈的名称，如 **cluster-sec**。如果您删除集群，则需要此堆栈的名称。
- ❷ **<template>** 是您保存的 CloudFormation 模板 YAML 文件的相对路径和名称。
- ❸ **<parameters>** 是 CloudFormation 参数 JSON 文件的相对路径和名称。
- ❹ 您必须明确声明 **CAPABILITY\_NAMED\_IAM** 功能，因为提供的模板会创建一些 **AWS::IAM::Role** 和 **AWS::IAM::InstanceProfile** 资源。

### 输出示例

```
arn:aws:cloudformation:us-east-1:269333783861:stack/cluster-sec/03bd4210-2ed7-11eb-6d7a-13fc0b61e9db
```

#### 4. 确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <name>
```

在 **StackStatus** 显示 **CREATE\_COMPLETE** 后，输出会显示以下参数的值。您必须将这些参数值提供给您在创建集群时要运行的其他 CloudFormation 模板：

<b>MasterSecurityGroupID</b>	Master 安全组 ID
<b>WorkerSecurityGroupID</b>	worker 安全组 ID
<b>MasterInstanceProfile</b>	Master IAM 实例配置集
<b>WorkerInstanceProfile</b>	worker IAM 实例配置集

#### 1.9.10.1. 安全对象的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署 OpenShift Container Platform 集群所需的安全对象。

##### 例 1.27. 安全对象的 CloudFormation 模板

```
AWSTemplateFormatVersion: 2010-09-09
```

```
Description: Template for OpenShift Cluster Security Elements (Security Groups & IAM)
```

```
Parameters:
```

```
InfrastructureName:
```

```
AllowedPattern: ^([a-zA-Z][a-zA-Z0-9\-\_]{0,26})$
```

```
MaxLength: 27
```

```
MinLength: 1
```

```
ConstraintDescription: Infrastructure name must be alphanumeric, start with a letter, and have a maximum of 27 characters.
```

```
Description: A short, unique cluster ID used to tag cloud resources and identify items owned or used by the cluster.
```

```
Type: String
```

```
VpcCidr:
```

```
AllowedPattern: ^(((0-9){1-9}[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.)\.{3}((0-9){1-9}[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.(1[6-9]|2[0-4])$
```

```
ConstraintDescription: CIDR block parameter must be in the form x.x.x.x/16-24.
```

```
Default: 10.0.0/16
```

```
Description: CIDR block for VPC.
```

```
Type: String
```

VpcId:  
Description: The VPC-scoped resources will belong to this VPC.  
Type: AWS::EC2::VPC::Id

PrivateSubnets:  
Description: The internal subnets.  
Type: List<AWS::EC2::Subnet::Id>

Metadata:

AWS::CloudFormation::Interface:

ParameterGroups:

- Label:  
default: "Cluster Information"
- Parameters:
  - InfrastructureName
- Label:  
default: "Network Configuration"
- Parameters:
  - VpcId
  - VpcCidr
  - PrivateSubnets

ParameterLabels:

InfrastructureName:  
default: "Infrastructure Name"

VpcId:  
default: "VPC ID"

VpcCidr:  
default: "VPC CIDR"

PrivateSubnets:  
default: "Private Subnets"

Resources:

MasterSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Cluster Master Security Group

SecurityGroupIngress:

- IpProtocol: icmp  
FromPort: 0  
ToPort: 0  
CidrIp: !Ref VpcCidr
- IpProtocol: tcp  
FromPort: 22  
ToPort: 22  
CidrIp: !Ref VpcCidr
- IpProtocol: tcp  
ToPort: 6443  
FromPort: 6443  
CidrIp: !Ref VpcCidr
- IpProtocol: tcp  
FromPort: 22623  
ToPort: 22623  
CidrIp: !Ref VpcCidr

VpcId: !Ref VpcId

WorkerSecurityGroup:

Type: AWS::EC2::SecurityGroup



## Properties:

GroupDescription: Cluster Worker Security Group

SecurityGroupIngress:

- IpProtocol: icmp

FromPort: 0

ToPort: 0

CidrIp: !Ref VpcCidr

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: !Ref VpcCidr

VpcId: !Ref VpcId

## MasterIngressEtcd:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: etcd

FromPort: 2379

ToPort: 2380

IpProtocol: tcp

## MasterIngressVxlan:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: Vxlan packets

FromPort: 4789

ToPort: 4789

IpProtocol: udp

## MasterIngressWorkerVxlan:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Vxlan packets

FromPort: 4789

ToPort: 4789

IpProtocol: udp

## MasterIngressGeneve:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: Geneve packets

FromPort: 6081

ToPort: 6081

IpProtocol: udp

## MasterIngressWorkerGeneve:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Geneve packets  
FromPort: 6081  
ToPort: 6081  
IpProtocol: udp

MasterIngressInternal:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Internal cluster communication  
FromPort: 9000  
ToPort: 9999  
IpProtocol: tcp

MasterIngressWorkerInternal:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Internal cluster communication  
FromPort: 9000  
ToPort: 9999  
IpProtocol: tcp

MasterIngressInternalUDP:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Internal cluster communication  
FromPort: 9000  
ToPort: 9999  
IpProtocol: udp

MasterIngressWorkerInternalUDP:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Internal cluster communication  
FromPort: 9000  
ToPort: 9999  
IpProtocol: udp

MasterIngressKube:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Kubernetes kubelet, scheduler and controller manager  
FromPort: 10250  
ToPort: 10259  
IpProtocol: tcp

**MasterIngressWorkerKube:**

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Kubernetes kubelet, scheduler and controller manager

FromPort: 10250

ToPort: 10259

IpProtocol: tcp

**MasterIngressIngressServices:**

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: Kubernetes ingress services

FromPort: 30000

ToPort: 32767

IpProtocol: tcp

**MasterIngressWorkerIngressServices:**

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Kubernetes ingress services

FromPort: 30000

ToPort: 32767

IpProtocol: tcp

**MasterIngressIngressServicesUDP:**

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: Kubernetes ingress services

FromPort: 30000

ToPort: 32767

IpProtocol: udp

**MasterIngressWorkerIngressServicesUDP:**

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Kubernetes ingress services

FromPort: 30000

ToPort: 32767

IpProtocol: udp

**WorkerIngressVxlan:**

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Vxlan packets  
FromPort: 4789  
ToPort: 4789  
IpProtocol: udp

WorkerIngressMasterVxlan:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt WorkerSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Vxlan packets  
FromPort: 4789  
ToPort: 4789  
IpProtocol: udp

WorkerIngressGeneve:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt WorkerSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Geneve packets  
FromPort: 6081  
ToPort: 6081  
IpProtocol: udp

WorkerIngressMasterGeneve:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt WorkerSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Geneve packets  
FromPort: 6081  
ToPort: 6081  
IpProtocol: udp

WorkerIngressInternal:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt WorkerSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Internal cluster communication  
FromPort: 9000  
ToPort: 9999  
IpProtocol: tcp

WorkerIngressMasterInternal:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt WorkerSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Internal cluster communication  
FromPort: 9000  
ToPort: 9999  
IpProtocol: tcp

WorkerIngressInternalUDP:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Internal cluster communication

FromPort: 9000

ToPort: 9999

IpProtocol: udp

WorkerIngressMasterInternalUDP:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: Internal cluster communication

FromPort: 9000

ToPort: 9999

IpProtocol: udp

WorkerIngressKube:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Kubernetes secure kubelet port

FromPort: 10250

ToPort: 10250

IpProtocol: tcp

WorkerIngressWorkerKube:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: Internal Kubernetes communication

FromPort: 10250

ToPort: 10250

IpProtocol: tcp

WorkerIngressIngressServices:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Kubernetes ingress services

FromPort: 30000

ToPort: 32767

IpProtocol: tcp

WorkerIngressMasterIngressServices:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: Kubernetes ingress services

FromPort: 30000

ToPort: 32767

IpProtocol: tcp

WorkerIngressIngressServicesUDP:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Kubernetes ingress services

FromPort: 30000

ToPort: 32767

IpProtocol: udp

WorkerIngressMasterIngressServicesUDP:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: Kubernetes ingress services

FromPort: 30000

ToPort: 32767

IpProtocol: udp

MasterIamRole:

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: "Allow"

Principal:

Service:

- "ec2.amazonaws.com"

Action:

- "sts:AssumeRole"

Policies:

- PolicyName: !Join ["-", [!Ref InfrastructureName, "master", "policy"]]

PolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: "Allow"

Action:

- "ec2:AttachVolume"

- "ec2:AuthorizeSecurityGroupIngress"

- "ec2:CreateSecurityGroup"

- "ec2:CreateTags"

- "ec2:CreateVolume"

- "ec2>DeleteSecurityGroup"

- "ec2>DeleteVolume"

- "ec2:Describe\*"

- "ec2:DetachVolume"

- "ec2:ModifyInstanceAttribute"

- "ec2:ModifyVolume"

- "ec2:RevokeSecurityGroupIngress"

- "elasticloadbalancing:AddTags"

- "elasticloadbalancing:AttachLoadBalancerToSubnets"

- "elasticloadbalancing:ApplySecurityGroupsToLoadBalancer"
- "elasticloadbalancing:CreateListener"
- "elasticloadbalancing:CreateLoadBalancer"
- "elasticloadbalancing:CreateLoadBalancerPolicy"
- "elasticloadbalancing:CreateLoadBalancerListeners"
- "elasticloadbalancing:CreateTargetGroup"
- "elasticloadbalancing:ConfigureHealthCheck"
- "elasticloadbalancing>DeleteListener"
- "elasticloadbalancing>DeleteLoadBalancer"
- "elasticloadbalancing>DeleteLoadBalancerListeners"
- "elasticloadbalancing>DeleteTargetGroup"
- "elasticloadbalancing:DeregisterInstancesFromLoadBalancer"
- "elasticloadbalancing:DeregisterTargets"
- "elasticloadbalancing:Describe\*"
- "elasticloadbalancing:DetachLoadBalancerFromSubnets"
- "elasticloadbalancing:ModifyListener"
- "elasticloadbalancing:ModifyLoadBalancerAttributes"
- "elasticloadbalancing:ModifyTargetGroup"
- "elasticloadbalancing:ModifyTargetGroupAttributes"
- "elasticloadbalancing:RegisterInstancesWithLoadBalancer"
- "elasticloadbalancing:RegisterTargets"
- "elasticloadbalancing:SetLoadBalancerPoliciesForBackendServer"
- "elasticloadbalancing:SetLoadBalancerPoliciesOfListener"
- "kms:DescribeKey"

Resource: "\*"

#### MasterInstanceProfile:

Type: "AWS::IAM::InstanceProfile"

#### Properties:

##### Roles:

- Ref: "MasterIamRole"

#### WorkerIamRole:

Type: AWS::IAM::Role

#### Properties:

##### AssumeRolePolicyDocument:

Version: "2012-10-17"

##### Statement:

- Effect: "Allow"

##### Principal:

##### Service:

- "ec2.amazonaws.com"

##### Action:

- "sts:AssumeRole"

#### Policies:

- PolicyName: !Join ["-", [!Ref InfrastructureName, "worker", "policy"]]

##### PolicyDocument:

Version: "2012-10-17"

##### Statement:

- Effect: "Allow"

##### Action:

- "ec2:DescribeInstances"
- "ec2:DescribeRegions"

Resource: "\*"

#### WorkerInstanceProfile:

```

Type: "AWS::IAM::InstanceProfile"
Properties:
  Roles:
    - Ref: "WorkerIamRole"

Outputs:
MasterSecurityGroupId:
  Description: Master Security Group ID
  Value: !GetAtt MasterSecurityGroup.GroupId

WorkerSecurityGroupId:
  Description: Worker Security Group ID
  Value: !GetAtt WorkerSecurityGroup.GroupId

MasterInstanceProfile:
  Description: Master IAM Instance Profile
  Value: !Ref MasterInstanceProfile

WorkerInstanceProfile:
  Description: Worker IAM Instance Profile
  Value: !Ref WorkerInstanceProfile

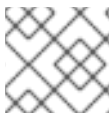
```

#### 其他资源

- 您可以通过导航 [AWS CloudFormation 控制台](#) 来查看您创建的 CloudFormation 堆栈的详情。

### 1.9.11. AWS 基础架构的 RHCOS AMI

红帽为您为 OpenShift Container Platform 节点指定的各种 Amazon Web Services (AWS) 区域提供了有效的 Red Hat Enterprise Linux CoreOS (RHCOS) AMI。



#### 注意

您还可以导入您自己的 AMI，来安装到没有发布 RHCOS AMI 的区域。

表 1.27. RHCOS AMI

AWS 区	AWS AMI
af-south-1	ami-09921c9c1c36e695c
ap-east-1	ami-01ee8446e9af6b197
ap-northeast-1	ami-04e5b5722a55846ea
ap-northeast-2	ami-0fdc25c8a0273a742
ap-south-1	ami-09e3deb397cc526a8
ap-southeast-1	ami-0630e03f75e02eec4



AWS 区	AWS AMI
<b>ap-southeast-2</b>	<b>ami-069450613262ba03c</b>
<b>ca-central-1</b>	<b>ami-012518cdbd3057dfd</b>
<b>eu-central-1</b>	<b>ami-0bd7175ff5b1aef0c</b>
<b>eu-north-1</b>	<b>ami-06c9ec42d0a839ad2</b>
<b>eu-south-1</b>	<b>ami-0614d7440a0363d71</b>
<b>eu-west-1</b>	<b>ami-01b89df58b5d4d5fa</b>
<b>eu-west-2</b>	<b>ami-06f6e31ddd554f89d</b>
<b>eu-west-3</b>	<b>ami-0dc82e2517ded15a1</b>
<b>me-south-1</b>	<b>ami-07d181e3aa0f76067</b>
<b>sa-east-1</b>	<b>ami-0cd44e6dd20e6c7fa</b>
<b>us-east-1</b>	<b>ami-04a16d506e5b0e246</b>
<b>us-east-2</b>	<b>ami-0a1f868ad58ea59a7</b>
<b>us-west-1</b>	<b>ami-0a65d76e3a6f6622f</b>
<b>us-west-2</b>	<b>ami-0dd9008abadc519f1</b>

### 1.9.11.1. 没有公布的 RHCOS AMI 的 AWS 区域

您可以将 OpenShift Container Platform 集群部署到 Amazon Web Services (AWS) 区域，而无需对 Red Hat Enterprise Linux CoreOS (RHCOS) Amazon Machine Image (AMI) 或 AWS 软件开发 kit (SDK) 的原生支持。如果 AWS 区域没有可用的已公布的 AMI，您可以在安装集群前上传自定义 AMI。如果您要将集群部署到 AWS 政府区域，则需要此参数。

如果您部署到没有公布的 RHCOS AMI 的非机构区域，且您没有指定自定义的 AMI，安装程序会自动将 **us-east-1** AMI 复制到用户帐户。然后，安装程序使用默认或用户指定的密钥管理服务 (KMS) 密钥创建带有加密 EBS 卷的 control plane 机器。这允许 AMI 跟踪与公布的 RHCOS AMI 相同的进程工作流。

在集群创建过程中，无法从终端中选择没有原生支持 RHCOS AMI 的区域，因为它没有发布。但是，您可以通过在 **install-config.yaml** 文件中配置自定义 AMI 来安装到这个区域。

### 1.9.11.2. 在 AWS 中上传自定义 RHCOS AMI

如果要部署到自定义 Amazon Web Services (AWS) 区域，您必须上传属于该区域的自定义 Red Hat Enterprise Linux CoreOS (RHCOS) Amazon Machine Image (AMI)。

## 先决条件

- 已配置了一个 AWS 帐户。
- 已使用所需的 IAM [服务角色](#) 创建 Amazon S3 存储桶。
- 将 RHCOS VMDK 文件上传到 Amazon S3。RHCOS VMDK 文件必须是小于或等于您要安装的 OpenShift Container Platform 版本的最高版本。
- 您下载了 AWS CLI 并安装到您的计算机上。请参阅[使用捆绑安装程序安装 AWS CLI](#)。

## 流程

1. 将 AWS 配置集导出为环境变量：

```
$ export AWS_PROFILE=<aws_profile> 1
```

- 1 拥有 AWS 凭证的 AWS 配置集名称，如 **govcloud**。

2. 将与自定义 AMI 关联的区域导出为环境变量：

```
$ export AWS_DEFAULT_REGION=<aws_region> 1
```

- 1 AWS 区域，如 **us-gov-east-1**。

3. 将上传至 Amazon S3 的 RHCOS 版本导出为环境变量：

```
$ export RHCOS_VERSION=<version> 1
```

- 1 RHCOS VMDK 版本，如 **4.6.0**。

4. 将 Amazon S3 存储桶名称导出为环境变量：

```
$ export VMIMPORT_BUCKET_NAME=<s3_bucket_name>
```

5. 创建 **containers.json** 文件并定义 RHCOS VMDK 文件：

```
$ cat <<EOF > containers.json
{
  "Description": "rhcos-${RHCOS_VERSION}-x86_64-aws.x86_64",
  "Format": "vmdk",
  "UserBucket": {
    "S3Bucket": "${VMIMPORT_BUCKET_NAME}",
    "S3Key": "rhcos-${RHCOS_VERSION}-x86_64-aws.x86_64.vmdk"
  }
}
EOF
```

6. 将 RHCOS 磁盘导入为 Amazon EBS 快照：

```
$ aws ec2 import-snapshot --region ${AWS_DEFAULT_REGION} \
  --description "<description>" \ ❶
  --disk-container "file://<file_path>/containers.json" ❷
```

- ❶ 导入 RHCOS 磁盘的描述，如 `rhcos-${RHCOS_VERSION}-x86_64-aws.x86_64`。
- ❷ 描述 RHCOS 磁盘的 JSON 文件的文件路径。JSON 文件应包含您的 Amazon S3 存储桶名称和密钥。

#### 7. 检查镜像导入的状态：

```
$ watch -n 5 aws ec2 describe-import-snapshot-tasks --region ${AWS_DEFAULT_REGION}
```

#### 输出示例

```
{
  "ImportSnapshotTasks": [
    {
      "Description": "rhcos-4.6.0-x86_64-aws.x86_64",
      "ImportTaskId": "import-snap-fh6i8uil",
      "SnapshotTaskDetail": {
        "Description": "rhcos-4.6.0-x86_64-aws.x86_64",
        "DiskImageSize": 819056640.0,
        "Format": "VMDK",
        "SnapshotId": "snap-06331325870076318",
        "Status": "completed",
        "UserBucket": {
          "S3Bucket": "external-images",
          "S3Key": "rhcos-4.6.0-x86_64-aws.x86_64.vmdk"
        }
      }
    }
  ]
}
```

复制 **SnapshotId** 以注册镜像。

#### 8. 从 RHCOS 快照创建自定义 RHCOS AMI:

```
$ aws ec2 register-image \
  --region ${AWS_DEFAULT_REGION} \
  --architecture x86_64 \ ❶
  --description "rhcos-${RHCOS_VERSION}-x86_64-aws.x86_64" \ ❷
  --ena-support \
  --name "rhcos-${RHCOS_VERSION}-x86_64-aws.x86_64" \ ❸
  --virtualization-type hvm \
  --root-device-name '/dev/xvda' \
  --block-device-mappings 'DeviceName=/dev/xvda,Ebs={DeleteOnTermination=true,SnapshotId=<snapshot_ID>}' ❹
```

- ❶ RHCOS VMDK 架构类型，如 `x86_64`、`s390x` 或 `ppc64le`。
- ❷ 来自导入快照的 **Description**。

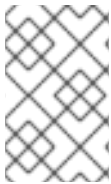
- 3 RHCOS AMI 的名称。
- 4 导入的快照中的 **SnapshotID**。

如需了解更多有关这些 API 的信息，请参阅 AWS 文档 [导入快照](#) 和 [创建由 EBS 支持的 AMI](#)。

### 1.9.12. 在 AWS 中创建 bootstrap 节点

您必须在 Amazon Web Services (AWS) 中创建 bootstrap 节点，以便在 OpenShift Container Platform 集群初始化过程中使用。

您可以使用提供的 CloudFormation 模板和自定义参数文件来创建 AWS 资源堆栈。堆栈代表 OpenShift Container Platform 安装所需的 bootstrap 节点。



#### 注意

如果不使用提供的 CloudFormation 模板来创建 bootstrap 节点，您必须检查提供的信息并手动创建基础架构。如果集群没有正确初始化，您可能需要联系红帽支持并提供您的安装日志。

#### 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。
- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。
- 您在 AWS 中创建并配置了 DNS、负载均衡器和监听程序。
- 您在 AWS 中创建了集群所需的安全组和角色。

#### 流程

1. 提供一个位置，以便向集群提供 **bootstrap.ign** Ignition 配置文件。此文件位于您的安装目录中。达成此目标的一种方式是在集群区域中创建一个 S3 存储桶，并将 Ignition 配置文件上传到其中。



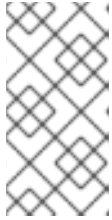
#### 重要

提供的 CloudFormation 模板假定集群的 Ignition 配置文件由 S3 存储桶提供。如果选择从其他位置提供文件，您必须修改模板。



#### 重要

如果您部署到具有与 AWS SDK 不同的端点，或者您提供自己的自定义端点的区域，则必须为 S3 存储桶使用预签名 URL 而不是 **s3://** 模式。



## 注意

bootstrap Ignition 配置文件包含 secret，如 X.509 密钥。以下步骤为 S3 存储桶提供基本安全性。若要提供额外的安全性，您可以启用 S3 存储桶策略，仅允许某些用户（如 OpenShift IAM 用户）访问存储桶中包含的对象。您可以完全避开 S3，并从 bootstrap 可访问的任意地址提供 bootstrap Ignition 配置文件。

- a. 创建存储桶：

```
$ aws s3 mb s3://<cluster-name>-infra 1
```

- 1 <cluster-name>-infra 是存储桶名称。在创建 **install-config.yaml** 文件时，将 <cluster-name> 替换为为集群指定的名称。

- b. 将 **bootstrap.ign** Ignition 配置文件上传到存储桶：

```
$ aws s3 cp <installation_directory>/bootstrap.ign s3://<cluster-name>-infra/bootstrap.ign 1
```

- 1 对于 <installation\_directory>，请指定安装文件保存到的目录的路径。

- c. 验证文件已经上传：

```
$ aws s3 ls s3://<cluster-name>-infra/
```

### 输出示例

```
2019-04-03 16:15:16 314878 bootstrap.ign
```

2. 创建一个 JSON 文件，其包含模板所需的参数值：

```
[
  {
    "ParameterKey": "InfrastructureName", 1
    "ParameterValue": "mycluster-<random_string>" 2
  },
  {
    "ParameterKey": "RhcosAmi", 3
    "ParameterValue": "ami-<random_string>" 4
  },
  {
    "ParameterKey": "AllowedBootstrapSshCidr", 5
    "ParameterValue": "0.0.0.0/0" 6
  },
  {
    "ParameterKey": "PublicSubnet", 7
    "ParameterValue": "subnet-<random_string>" 8
  },
  {
    "ParameterKey": "MasterSecurityGroup", 9
    "ParameterValue": "sg-<random_string>" 10
  }
]
```

```

},
{
  "ParameterKey": "VpcId", 11
  "ParameterValue": "vpc-<random_string>" 12
},
{
  "ParameterKey": "BootstrapIgnitionLocation", 13
  "ParameterValue": "s3://<bucket_name>/bootstrap.ign" 14
},
{
  "ParameterKey": "AutoRegisterELB", 15
  "ParameterValue": "yes" 16
},
{
  "ParameterKey": "RegisterNlbTargetsLambdaArn", 17
  "ParameterValue": "arn:aws:lambda:<region>:<account_number>:function:
<dns_stack_name>-RegisterNlbTargets-<random_string>" 18
},
{
  "ParameterKey": "ExternalApiTargetGroupArn", 19
  "ParameterValue": "arn:aws:elasticloadbalancing:<region>:
<account_number>:targetgroup/<dns_stack_name>-Exter-<random_string>" 20
},
{
  "ParameterKey": "InternalApiTargetGroupArn", 21
  "ParameterValue": "arn:aws:elasticloadbalancing:<region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<random_string>" 22
},
{
  "ParameterKey": "InternalServiceTargetGroupArn", 23
  "ParameterValue": "arn:aws:elasticloadbalancing:<region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<random_string>" 24
}
]

```

- 1 您的 Ignition 配置文件中为集群编码的集群基础架构名称。
- 2 指定从 Ignition 配置文件元数据中提取的基础架构名称，其格式为 **<cluster-name>-<random-string>**。
- 3 用于 bootstrap 节点的当前 Red Hat Enterprise Linux CoreOS (RHCOS) AMI。
- 4 指定有效的 **AWS::EC2::Image::Id** 值。
- 5 允许通过 SSH 访问 bootstrap 节点的 CIDR 块。
- 6 以 **x.x.x.x/16-24** 格式指定 CIDR 块。
- 7 与 VPC 关联的公共子网，将 bootstrap 节点启动到其中。
- 8 指定 VPC 的 CloudFormation 模板输出的 **PublicSubnetIds** 值。
- 9 master 安全组 ID（用于注册临时规则）

- 10 指定安全组和角色的 CloudFormation 模板输出的 **MasterSecurityGroupId** 值。
  - 11 创建的资源将从属于的 VPC。
  - 12 指定 VPC 的 CloudFormation 模板输出的 **VpcId** 值。
  - 13 从中获取 bootstrap Ignition 配置文件的位置。
  - 14 指定 S3 存储桶和文件名，格式为 **s3://<bucket\_name>/bootstrap.ign**。
  - 15 是否要注册网络负载均衡器 (NLB)。
  - 16 指定 **yes** 或 **no**。如果指定 **yes**，您必须提供一个 Lambda Amazon Resource Name (ARN) 值。
  - 17 NLB IP 目标注册 lambda 组的 ARN。
  - 18 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **RegisterNlbTargetsLambda** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
  - 19 外部 API 负载均衡器目标组的 ARN。
  - 20 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **ExternalApiTargetGroupArn** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
  - 21 内部 API 负载均衡器目标组群的 ARN。
  - 22 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **InternalApiTargetGroupArn** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
  - 23 内部服务负载均衡器目标组群的 ARN。
  - 24 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **InternalServiceTargetGroupArn** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
3. 复制本主题的 **Bootstrap** 机器的 **CloudFormation** 模板部分中的模板，并将它以 YAML 文件形式保存到计算机上。此模板描述了集群所需的 bootstrap 机器。
  4. 启动 CloudFormation 模板，以创建代表 bootstrap 节点的 AWS 资源堆栈：



### 重要

您必须在一行内输入命令。

```
$ aws cloudformation create-stack --stack-name <name> 1
  --template-body file://<template>.yaml 2
  --parameters file://<parameters>.json 3
  --capabilities CAPABILITY_NAMED_IAM 4
```

- 1 **<name>** 是 CloudFormation 堆栈的名称，如 **cluster-bootstrap**。如果您删除集群，则需要此堆栈的名称。
- 2 **<template>** 是您保存的 CloudFormation 模板 YAML 文件的相对路径和名称。

- 3 **<parameters>** 是 CloudFormation 参数 JSON 文件的相对路径和名称。
- 4 您必须明确声明 **CAPABILITY\_NAMED\_IAM** 功能，因为提供的模板会创建一些 **AWS::IAM::Role** 和 **AWS::IAM::InstanceProfile** 资源。

### 输出示例

```
arn:aws:cloudformation:us-east-1:269333783861:stack/cluster-bootstrap/12944486-2add-11eb-9dee-12dace8e3a83
```

5. 确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <name>
```

在 **StackStatus** 显示 **CREATE\_COMPLETE** 后，输出会显示以下参数的值。您必须将这些参数值提供给您在创建集群时要运行的其他 CloudFormation 模板：

<b>Bootstrap InstanceId</b>	bootstrap 实例 ID。
<b>Bootstrap PublicIp</b>	bootstrap 节点公共 IP 地址。
<b>Bootstrap PrivateIp</b>	bootstrap 节点专用 IP 地址。

#### 1.9.12.1. bootstrap 机器的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署 OpenShift Container Platform 集群所需的 bootstrap 机器。

##### 例 1.28. bootstrap 机器的 CloudFormation 模板

```
AWSTemplateFormatVersion: 2010-09-09
Description: Template for OpenShift Cluster Bootstrap (EC2 Instance, Security Groups and IAM)

Parameters:
  InfrastructureName:
    AllowedPattern: ^([a-zA-Z][a-zA-Z0-9\-\]{0,26})$
    MaxLength: 27
    MinLength: 1
    ConstraintDescription: Infrastructure name must be alphanumeric, start with a letter, and have a maximum of 27 characters.
    Description: A short, unique cluster ID used to tag cloud resources and identify items owned or used by the cluster.
    Type: String
  RhcosAmi:
    Description: Current Red Hat Enterprise Linux CoreOS AMI to use for bootstrap.
    Type: AWS::EC2::Image::Id
  AllowedBootstrapSshCidr:
    AllowedPattern: ^((([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])|([0-9]|1[0-9]|2[0-9]|3[0-2]))$
```



ConstraintDescription: CIDR block parameter must be in the form x.x.x.x/0-32.

Default: 0.0.0.0/0

Description: CIDR block to allow SSH access to the bootstrap node.

Type: String

PublicSubnet:

Description: The public subnet to launch the bootstrap node into.

Type: AWS::EC2::Subnet::Id

MasterSecurityGroupId:

Description: The master security group ID for registering temporary rules.

Type: AWS::EC2::SecurityGroup::Id

VpcId:

Description: The VPC-scoped resources will belong to this VPC.

Type: AWS::EC2::VPC::Id

BootstrapIgnitionLocation:

Default: s3://my-s3-bucket/bootstrap.ign

Description: Ignition config file location.

Type: String

AutoRegisterELB:

Default: "yes"

AllowedValues:

- "yes"

- "no"

Description: Do you want to invoke NLB registration, which requires a Lambda ARN parameter?

Type: String

RegisterNlbTargetsLambdaArn:

Description: ARN for NLB IP target registration lambda.

Type: String

ExternalApiTargetGroupArn:

Description: ARN for external API load balancer target group.

Type: String

InternalApiTargetGroupArn:

Description: ARN for internal API load balancer target group.

Type: String

InternalServiceTargetGroupArn:

Description: ARN for internal service load balancer target group.

Type: String

Metadata:

AWS::CloudFormation::Interface:

ParameterGroups:

- Label:

default: "Cluster Information"

Parameters:

- InfrastructureName

- Label:

default: "Host Information"

Parameters:

- RhcosAmi

- BootstrapIgnitionLocation

- MasterSecurityGroupId

- Label:

default: "Network Configuration"

Parameters:

- VpcId

- AllowedBootstrapSshCidr

- PublicSubnet

```

- Label:
  default: "Load Balancer Automation"
Parameters:
- AutoRegisterELB
- RegisterNlbTargetsLambdaArn
- ExternalApiTargetGroupArn
- InternalApiTargetGroupArn
- InternalServiceTargetGroupArn
ParameterLabels:
InfrastructureName:
  default: "Infrastructure Name"
VpcId:
  default: "VPC ID"
AllowedBootstrapSshCidr:
  default: "Allowed SSH Source"
PublicSubnet:
  default: "Public Subnet"
RhcOsAmi:
  default: "Red Hat Enterprise Linux CoreOS AMI ID"
BootstrapIgnitionLocation:
  default: "Bootstrap Ignition Source"
MasterSecurityGroupId:
  default: "Master Security Group ID"
AutoRegisterELB:
  default: "Use Provided ELB Automation"

```

Conditions:

```
DoRegistration: !Equals ["yes", !Ref AutoRegisterELB]
```

Resources:

```

BootstrapIamRole:
Type: AWS::IAM::Role
Properties:
AssumeRolePolicyDocument:
Version: "2012-10-17"
Statement:
- Effect: "Allow"
Principal:
Service:
- "ec2.amazonaws.com"
Action:
- "sts:AssumeRole"
Path: "/"
Policies:
- PolicyName: !Join ["-", [!Ref InfrastructureName, "bootstrap", "policy"]]
PolicyDocument:
Version: "2012-10-17"
Statement:
- Effect: "Allow"
Action: "ec2:Describe*"
Resource: "*"
- Effect: "Allow"
Action: "ec2:AttachVolume"
Resource: "*"
- Effect: "Allow"
Action: "ec2:DetachVolume"

```

```

    Resource: "*"
  - Effect: "Allow"
    Action: "s3:GetObject"
    Resource: "*"

```

```

BootstrapInstanceProfile:
  Type: "AWS::IAM::InstanceProfile"
  Properties:
    Path: "/"
    Roles:
      - Ref: "BootstrapIamRole"

```

```

BootstrapSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Cluster Bootstrap Security Group
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref AllowedBootstrapSshCidr
      - IpProtocol: tcp
        ToPort: 19531
        FromPort: 19531
        CidrIp: 0.0.0.0/0
    VpCid: !Ref VpCid

```

```

BootstrapInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !Ref RhcosAmi
    IamInstanceProfile: !Ref BootstrapInstanceProfile
    InstanceType: "i3.large"
    NetworkInterfaces:
      - AssociatePublicIpAddress: "true"
        DeviceIndex: "0"
        GroupSet:
          - !Ref "BootstrapSecurityGroup"
          - !Ref "MasterSecurityGroup"
        SubnetId: !Ref "PublicSubnet"
    UserData:
      Fn::Base64: !Sub
        - '{"ignition":{"config":{"replace":{"source":"${S3Loc}"}},"version":"3.1.0"}}'
        - {
            S3Loc: !Ref BootstrapIgnitionLocation
          }

```

```

RegisterBootstrapApiTarget:
  Condition: DoRegistration
  Type: Custom::NLBRegister
  Properties:
    ServiceToken: !Ref RegisterNLBTargetsLambdaArn
    TargetArn: !Ref ExternalApiTargetGroupArn
    TargetIp: !GetAtt BootstrapInstance.PrivateIp

```

```

RegisterBootstrapInternalApiTarget:

```

Condition: DoRegistration  
 Type: Custom::NLBRegister  
 Properties:  
 ServiceToken: !Ref RegisterNlbTargetsLambdaArn  
 TargetArn: !Ref InternalApiTargetGroupArn  
 TargetIp: !GetAtt BootstrapInstance.PrivateIp

RegisterBootstrapInternalServiceTarget:  
 Condition: DoRegistration  
 Type: Custom::NLBRegister  
 Properties:  
 ServiceToken: !Ref RegisterNlbTargetsLambdaArn  
 TargetArn: !Ref InternalServiceTargetGroupArn  
 TargetIp: !GetAtt BootstrapInstance.PrivateIp

#### Outputs:

BootstrapInstanceid:  
 Description: Bootstrap Instance ID.  
 Value: !Ref BootstrapInstance

BootstrapPublicIp:  
 Description: The bootstrap node public IP address.  
 Value: !GetAtt BootstrapInstance.PublicIp

BootstrapPrivateIp:  
 Description: The bootstrap node private IP address.  
 Value: !GetAtt BootstrapInstance.PrivateIp

#### 其他资源

- 您可以通过导航 [AWS CloudFormation 控制台](#) 来查看您创建的 CloudFormation 堆栈的详情。
- 如需有关 AWS 区的 Red Hat Enterprise Linux CoreOS (RHCOS) AMI 的详细信息，请参阅 [AWS 基础架构的 RHCOS AMI](#)。

### 1.9.13. 在 AWS 中创建 control plane 机器

您必须在集群要使用的 Amazon Web Services (AWS) 中创建 control plane 机器。

您可以使用提供的 CloudFormation 模板和自定义参数文件，创建代表 control plane 节点的 AWS 资源堆栈。



#### 重要

CloudFormation 模板会创建一个堆栈，它代表三个 control plane 节点。



#### 注意

如果不使用提供的 CloudFormation 模板来创建 control plane 节点，您必须检查提供的信息并手动创建基础架构。如果集群没有正确初始化，您可能需要联系红帽支持并提供您的安装日志。

#### 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。
- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。
- 您在 AWS 中创建并配置了 DNS、负载均衡器和监听程序。
- 您在 AWS 中创建了集群所需的安全组和角色。
- 已创建 bootstrap 机器。

## 流程

1. 创建一个 JSON 文件，其包含模板所需的参数值：

```
[
  {
    "ParameterKey": "InfrastructureName", ❶
    "ParameterValue": "mycluster-<random_string>" ❷
  },
  {
    "ParameterKey": "RhcosAmi", ❸
    "ParameterValue": "ami-<random_string>" ❹
  },
  {
    "ParameterKey": "AutoRegisterDNS", ❺
    "ParameterValue": "yes" ❻
  },
  {
    "ParameterKey": "PrivateHostedZoneId", ❼
    "ParameterValue": "<random_string>" ❽
  },
  {
    "ParameterKey": "PrivateHostedZoneName", ❾
    "ParameterValue": "mycluster.example.com" ❿
  },
  {
    "ParameterKey": "Master0Subnet", ⓫
    "ParameterValue": "subnet-<random_string>" ⓬
  },
  {
    "ParameterKey": "Master1Subnet", ⓭
    "ParameterValue": "subnet-<random_string>" ⓮
  },
  {
    "ParameterKey": "Master2Subnet", ⓯
    "ParameterValue": "subnet-<random_string>" ⓰
  },
  {
    "ParameterKey": "MasterSecurityGroupID", ⓱
    "ParameterValue": "sg-<random_string>" ⓲
  }
]
```

```

    },
    {
      "ParameterKey": "IgnitionLocation", 19
      "ParameterValue": "https://api-int.<cluster_name>.<domain_name>:22623/config/master"
    },
    {
      "ParameterKey": "CertificateAuthorities", 21
      "ParameterValue": "data:text/plain;charset=utf-8;base64,ABC...xYz==" 22
    },
    {
      "ParameterKey": "MasterInstanceProfileName", 23
      "ParameterValue": "<roles_stack>-MasterInstanceProfile-<random_string>" 24
    },
    {
      "ParameterKey": "MasterInstanceType", 25
      "ParameterValue": "m4.xlarge" 26
    },
    {
      "ParameterKey": "AutoRegisterELB", 27
      "ParameterValue": "yes" 28
    },
    {
      "ParameterKey": "RegisterNlbTargetsLambdaArn", 29
      "ParameterValue": "arn:aws:lambda:<region>:<account_number>:function:
<dns_stack_name>-RegisterNlbTargets-<random_string>" 30
    },
    {
      "ParameterKey": "ExternalApiTargetGroupArn", 31
      "ParameterValue": "arn:aws:elasticloadbalancing:<region>:
<account_number>:targetgroup/<dns_stack_name>-Exter-<random_string>" 32
    },
    {
      "ParameterKey": "InternalApiTargetGroupArn", 33
      "ParameterValue": "arn:aws:elasticloadbalancing:<region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<random_string>" 34
    },
    {
      "ParameterKey": "InternalServiceTargetGroupArn", 35
      "ParameterValue": "arn:aws:elasticloadbalancing:<region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<random_string>" 36
    }
  ]

```

- 1 您的 Ignition 配置文件中为集群编码的集群基础架构名称。
- 2 指定从 Ignition 配置文件元数据中提取的基础架构名称，其格式为 **<cluster-name>-<random-string>**。
- 3 用于 control plane 机器的当前 Red Hat Enterprise Linux CoreOS (RHCOS) AMI。
- 4 指定 **AWS::EC2::Image::Id** 值。
- 5 是否要执行 DNS etcd 注册。

- 6 指定 **yes** 或 **no**。如果指定 **yes**，您必须提供托管区信息。
- 7 用来注册 etcd 目标的 Route 53 专用区 ID。
- 8 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **PrivateHostedZoneId** 值。
- 9 用来注册目标的 Route 53 区。
- 10 指定 **<cluster\_name>.<domain\_name>**，其中 **<domain\_name>** 是您为集群生成 **install-config.yaml** 文件时所用的 Route 53 基域。请勿包含 AWS 控制台中显示的结尾句点 (.)。
- 11 13 15 在其中启动 control plane 机器的子网，最好是专用子网。
- 12 14 16 从 DNS 和负载均衡的 CloudFormation 模板输出的 **PrivateSubnets** 值指定子网。
- 17 与 control plane 节点（也称为 master 节点）关联的 master 安全组 ID。
- 18 指定安全组和角色的 CloudFormation 模板输出的 **MasterSecurityGroupId** 值。
- 19 从中获取 control plane Ignition 配置文件的位置。
- 20 指定生成的 Ignition 配置文件的位置，[https://api-int.<cluster\\_name>.<domain\\_name>:22623/config/master](https://api-int.<cluster_name>.<domain_name>:22623/config/master)。
- 21 要使用的 base64 编码证书颁发机构字符串。
- 22 指定安装目录中 **master.ign** 文件中的值。这个值是一个长字符串，格式为 **data:text/plain;charset=utf-8;base64,ABC...xYz==**。
- 23 与 control plane 节点关联的 IAM 配置集。
- 24 指定安全组和角色的 CloudFormation 模板输出的 **MasterInstanceProfile** 参数值。
- 25 用于 control plane 机器的 AWS 实例类型。
- 26 允许的值：
  - **m4.xlarge**
  - **m4.2xlarge**
  - **m4.4xlarge**
  - **m4.8xlarge**
  - **m4.10xlarge**
  - **m4.16xlarge**
  - **m5.xlarge**
  - **m5.2xlarge**
  - **m5.4xlarge**
  - **m5.8xlarge**
  - **m5.10xlarge**

- **m5.16xlarge**
- **m6i.xlarge**
- **c4.2xlarge**
- **c4.4xlarge**
- **c4.8xlarge**
- **r4.xlarge**
- **r4.2xlarge**
- **r4.4xlarge**
- **r4.8xlarge**
- **r4.16xlarge**



### 重要

如果您的区域中没有 **m4** 实例类型，例如 **eu-west-3**，请改为指定 **m5** 类型，如 **m5.xlarge**。

27. 是否要注册网络负载均衡器 (NLB)。
  28. 指定 **yes** 或 **no**。如果指定 **yes**，您必须提供一个 Lambda Amazon Resource Name (ARN) 值。
  29. NLB IP 目标注册 lambda 组的 ARN。
  30. 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **RegisterNlbIpTargetsLambda** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
  31. 外部 API 负载均衡器目标组的 ARN。
  32. 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **ExternalApiTargetGroupArn** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
  33. 内部 API 负载均衡器目标组群的 ARN。
  34. 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **InternalApiTargetGroupArn** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
  35. 内部服务负载均衡器目标组群的 ARN。
  36. 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **InternalServiceTargetGroupArn** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
2. 复制 **control plane** 机器的 **CloudFormation** 模板一节中的模板，并将它以 YAML 文件形式保存到计算机上。此模板描述了集群所需的 control plane 机器。
  3. 如果您将 **m5** 实例类型指定为 **MasterInstanceType** 的值，请将该实例类型添加到 CloudFormation 模板中的 **MasterInstanceType.AllowedValues** 参数。



4. 启动 CloudFormation 模板，以创建代表 control plane 节点的 AWS 资源堆栈：



### 重要

您必须在一行内输入命令。

```
$ aws cloudformation create-stack --stack-name <name> ❶
  --template-body file://<template>.yaml ❷
  --parameters file://<parameters>.json ❸
```

- ❶ **<name>** 是 CloudFormation 堆栈的名称，如 **cluster-control-plane**。如果您删除集群，则需要此堆栈的名称。
- ❷ **<template>** 是您保存的 CloudFormation 模板 YAML 文件的相对路径和名称。
- ❸ **<parameters>** 是 CloudFormation 参数 JSON 文件的相对路径和名称。

### 输出示例

```
arn:aws:cloudformation:us-east-1:269333783861:stack/cluster-control-plane/21c7e2b0-2ee2-11eb-c6f6-0aa34627df4b
```



### 注意

CloudFormation 模板会创建一个堆栈，它代表三个 control plane 节点。

5. 确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <name>
```

#### 1.9.13.1. control plane 机器的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署 OpenShift Container Platform 集群所需的 control plane 机器。

##### 例 1.29. control plane 机器的 CloudFormation 模板

```
AWSTemplateFormatVersion: 2010-09-09
Description: Template for OpenShift Cluster Node Launch (EC2 master instances)

Parameters:
  InfrastructureName:
    AllowedPattern: ^([a-zA-Z][a-zA-Z0-9\-\_]{0,26})$
    MaxLength: 27
    MinLength: 1
    ConstraintDescription: Infrastructure name must be alphanumeric, start with a letter, and have a maximum of 27 characters.
    Description: A short, unique cluster ID used to tag nodes for the kubelet cloud provider.
    Type: String
  RhcosAmi:
    Description: Current Red Hat Enterprise Linux CoreOS AMI to use for bootstrap.
```

Type: AWS::EC2::Image::Id  
AutoRegisterDNS:  
Default: "yes"  
AllowedValues:  
- "yes"  
- "no"  
Description: Do you want to invoke DNS etcd registration, which requires Hosted Zone information?  
Type: String  
PrivateHostedZoneId:  
Description: The Route53 private zone ID to register the etcd targets with, such as Z21IXYZABCZ2A4.  
Type: String  
PrivateHostedZoneName:  
Description: The Route53 zone to register the targets with, such as cluster.example.com. Omit the trailing period.  
Type: String  
Master0Subnet:  
Description: The subnets, recommend private, to launch the master nodes into.  
Type: AWS::EC2::Subnet::Id  
Master1Subnet:  
Description: The subnets, recommend private, to launch the master nodes into.  
Type: AWS::EC2::Subnet::Id  
Master2Subnet:  
Description: The subnets, recommend private, to launch the master nodes into.  
Type: AWS::EC2::Subnet::Id  
MasterSecurityGroupId:  
Description: The master security group ID to associate with master nodes.  
Type: AWS::EC2::SecurityGroup::Id  
IgnitionLocation:  
Default: https://api-int.\$CLUSTER\_NAME.\$DOMAIN:22623/config/master  
Description: Ignition config file location.  
Type: String  
CertificateAuthorities:  
Default: data:text/plain;charset=utf-8;base64,ABC...xYz==  
Description: Base64 encoded certificate authority string to use.  
Type: String  
MasterInstanceProfileName:  
Description: IAM profile to associate with master nodes.  
Type: String  
MasterInstanceType:  
Default: m5.xlarge  
Type: String  
AllowedValues:  
- "m4.xlarge"  
- "m4.2xlarge"  
- "m4.4xlarge"  
- "m4.10xlarge"  
- "m4.16xlarge"  
- "m5.xlarge"  
- "m5.2xlarge"  
- "m5.4xlarge"  
- "m5.8xlarge"  
- "m5.12xlarge"  
- "m5.16xlarge"  
- "m5a.xlarge"

- "m5a.2xlarge"
- "m5a.4xlarge"
- "m5a.8xlarge"
- "m5a.10xlarge"
- "m5a.16xlarge"
- "c4.2xlarge"
- "c4.4xlarge"
- "c4.8xlarge"
- "c5.2xlarge"
- "c5.4xlarge"
- "c5.9xlarge"
- "c5.12xlarge"
- "c5.18xlarge"
- "c5.24xlarge"
- "c5a.2xlarge"
- "c5a.4xlarge"
- "c5a.8xlarge"
- "c5a.12xlarge"
- "c5a.16xlarge"
- "c5a.24xlarge"
- "r4.xlarge"
- "r4.2xlarge"
- "r4.4xlarge"
- "r4.8xlarge"
- "r4.16xlarge"
- "r5.xlarge"
- "r5.2xlarge"
- "r5.4xlarge"
- "r5.8xlarge"
- "r5.12xlarge"
- "r5.16xlarge"
- "r5.24xlarge"
- "r5a.xlarge"
- "r5a.2xlarge"
- "r5a.4xlarge"
- "r5a.8xlarge"
- "r5a.12xlarge"
- "r5a.16xlarge"
- "r5a.24xlarge"

**AutoRegisterELB:**

Default: "yes"

AllowedValues:

- "yes"
- "no"

Description: Do you want to invoke NLB registration, which requires a Lambda ARN parameter?

Type: String

**RegisterNlbIpTargetsLambdaArn:**

Description: ARN for NLB IP target registration lambda. Supply the value from the cluster infrastructure or select "no" for AutoRegisterELB.

Type: String

**ExternalApiTargetGroupArn:**

Description: ARN for external API load balancer target group. Supply the value from the cluster infrastructure or select "no" for AutoRegisterELB.

Type: String

**InternalApiTargetGroupArn:**

Description: ARN for internal API load balancer target group. Supply the value from the cluster infrastructure or select "no" for AutoRegisterELB.

Type: String

InternalServiceTargetGroupArn:

Description: ARN for internal service load balancer target group. Supply the value from the cluster infrastructure or select "no" for AutoRegisterELB.

Type: String

Metadata:

AWS::CloudFormation::Interface:

ParameterGroups:

- Label:

default: "Cluster Information"

Parameters:

- InfrastructureName

- Label:

default: "Host Information"

Parameters:

- MasterInstanceType

- RhcosAmi

- IgnitionLocation

- CertificateAuthorities

- MasterSecurityGroupId

- MasterInstanceProfileName

- Label:

default: "Network Configuration"

Parameters:

- VpcId

- AllowedBootstrapSshCidr

- Master0Subnet

- Master1Subnet

- Master2Subnet

- Label:

default: "DNS"

Parameters:

- AutoRegisterDNS

- PrivateHostedZoneName

- PrivateHostedZoneId

- Label:

default: "Load Balancer Automation"

Parameters:

- AutoRegisterELB

- RegisterNlbTargetsLambdaArn

- ExternalApiTargetGroupArn

- InternalApiTargetGroupArn

- InternalServiceTargetGroupArn

ParameterLabels:

InfrastructureName:

default: "Infrastructure Name"

VpcId:

default: "VPC ID"

Master0Subnet:

default: "Master-0 Subnet"

Master1Subnet:

default: "Master-1 Subnet"

Master2Subnet:

```

    default: "Master-2 Subnet"
MasterInstanceType:
    default: "Master Instance Type"
MasterInstanceProfileName:
    default: "Master Instance Profile Name"
RhcOsAmi:
    default: "Red Hat Enterprise Linux CoreOS AMI ID"
BootstrapIgnitionLocation:
    default: "Master Ignition Source"
CertificateAuthorities:
    default: "Ignition CA String"
MasterSecurityGroupId:
    default: "Master Security Group ID"
AutoRegisterDNS:
    default: "Use Provided DNS Automation"
AutoRegisterELB:
    default: "Use Provided ELB Automation"
PrivateHostedZoneName:
    default: "Private Hosted Zone Name"
PrivateHostedZoneId:
    default: "Private Hosted Zone ID"

Conditions:
DoRegistration: !Equals ["yes", !Ref AutoRegisterELB]
DoDns: !Equals ["yes", !Ref AutoRegisterDNS]

Resources:
Master0:
    Type: AWS::EC2::Instance
    Properties:
        ImageId: !Ref RhcOsAmi
        BlockDeviceMappings:
            - DeviceName: /dev/xvda
              Ebs:
                  VolumeSize: "120"
                  VolumeType: "gp2"
        IamInstanceProfile: !Ref MasterInstanceProfileName
        InstanceType: !Ref MasterInstanceType
        NetworkInterfaces:
            - AssociatePublicIp: "false"
              DeviceIndex: "0"
              GroupSet:
                  - !Ref "MasterSecurityGroupId"
              SubnetId: !Ref "Master0Subnet"
        UserData:
            Fn::Base64: !Sub
                - '{"ignition":{"config":{"merge":{"source":"${SOURCE}"}}, "security":{"tls":
{"certificateAuthorities":[{"source":"${CA_BUNDLE}"}]}, "version":"3.1.0"}}'
                - {
                    SOURCE: !Ref IgnitionLocation,
                    CA_BUNDLE: !Ref CertificateAuthorities,
                }
    Tags:
        - Key: !Join ["", ["kubernetes.io/cluster/", !Ref InfrastructureName]]
          Value: "shared"

```

```

RegisterMaster0:
  Condition: DoRegistration
  Type: Custom::NLBRegister
  Properties:
    ServiceToken: !Ref RegisterNlbTargetsLambdaArn
    TargetArn: !Ref ExternalApiTargetGroupArn
    TargetIp: !GetAtt Master0.PrivateIp

```

```

RegisterMaster0InternalApiTarget:
  Condition: DoRegistration
  Type: Custom::NLBRegister
  Properties:
    ServiceToken: !Ref RegisterNlbTargetsLambdaArn
    TargetArn: !Ref InternalApiTargetGroupArn
    TargetIp: !GetAtt Master0.PrivateIp

```

```

RegisterMaster0InternalServiceTarget:
  Condition: DoRegistration
  Type: Custom::NLBRegister
  Properties:
    ServiceToken: !Ref RegisterNlbTargetsLambdaArn
    TargetArn: !Ref InternalServiceTargetGroupArn
    TargetIp: !GetAtt Master0.PrivateIp

```

```

Master1:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !Ref RhcosAmi
    BlockDeviceMappings:
      - DeviceName: /dev/xvda
        Ebs:
          VolumeSize: "120"
          VolumeType: "gp2"
    IamInstanceProfile: !Ref MasterInstanceProfileName
    InstanceType: !Ref MasterInstanceType
    NetworkInterfaces:
      - AssociatePublicIpAddress: "false"
        DeviceIndex: "0"
        GroupSet:
          - !Ref "MasterSecurityGroupId"
        SubnetId: !Ref "Master1Subnet"
    UserData:
      Fn::Base64: !Sub
        - {"ignition":{"config":{"merge":[{"source":"${SOURCE}"]},"security":{"tls":
{"certificateAuthorities":[{"source":"${CA_BUNDLE}"]},"version":"3.1.0"}}}
        - {
          SOURCE: !Ref IgnitionLocation,
          CA_BUNDLE: !Ref CertificateAuthorities,
        }
    Tags:
      - Key: !Join [ "", ["kubernetes.io/cluster/", !Ref InfrastructureName]]
        Value: "shared"

```

```

RegisterMaster1:
  Condition: DoRegistration
  Type: Custom::NLBRegister

```

## Properties:

ServiceToken: !Ref RegisterNlbTargetsLambdaArn  
 TargetArn: !Ref ExternalApiTargetGroupArn  
 TargetIp: !GetAtt Master1.PrivateIp

## RegisterMaster1InternalApiTarget:

Condition: DoRegistration  
 Type: Custom::NLBRegister  
 Properties:  
 ServiceToken: !Ref RegisterNlbTargetsLambdaArn  
 TargetArn: !Ref InternalApiTargetGroupArn  
 TargetIp: !GetAtt Master1.PrivateIp

## RegisterMaster1InternalServiceTarget:

Condition: DoRegistration  
 Type: Custom::NLBRegister  
 Properties:  
 ServiceToken: !Ref RegisterNlbTargetsLambdaArn  
 TargetArn: !Ref InternalServiceTargetGroupArn  
 TargetIp: !GetAtt Master1.PrivateIp

## Master2:

Type: AWS::EC2::Instance  
 Properties:  
 ImageId: !Ref RhcosAmi  
 BlockDeviceMappings:  
 - DeviceName: /dev/xvda  
 Ebs:  
 VolumeSize: "120"  
 VolumeType: "gp2"  
 IamInstanceProfile: !Ref MasterInstanceProfileName  
 InstanceType: !Ref MasterInstanceType  
 NetworkInterfaces:  
 - AssociatePublicIpAddress: "false"  
 DeviceIndex: "0"  
 GroupSet:  
 - !Ref "MasterSecurityGroupId"  
 SubnetId: !Ref "Master2Subnet"  
 UserData:  
 Fn::Base64: !Sub  
 - '{"ignition":{"config":{"merge":{"source":"\${SOURCE}"},"security":{"tls":{"certificateAuthorities":[{"source":"\${CA\_BUNDLE}"}]},"version":"3.1.0"}}}'  
 - {  
 SOURCE: !Ref IgnitionLocation,  
 CA\_BUNDLE: !Ref CertificateAuthorities,  
 }  
 }  
 Tags:  
 - Key: !Join [ "", ["kubernetes.io/cluster/", !Ref InfrastructureName] ]  
 Value: "shared"

## RegisterMaster2:

Condition: DoRegistration  
 Type: Custom::NLBRegister  
 Properties:  
 ServiceToken: !Ref RegisterNlbTargetsLambdaArn  
 TargetArn: !Ref ExternalApiTargetGroupArn

TargetIp: !GetAtt Master2.PrivateIp

RegisterMaster2InternalApiTarget:

Condition: DoRegistration

Type: Custom::NLBRegister

Properties:

ServiceToken: !Ref RegisterNlbTargetsLambdaArn

TargetArn: !Ref InternalApiTargetGroupArn

TargetIp: !GetAtt Master2.PrivateIp

RegisterMaster2InternalServiceTarget:

Condition: DoRegistration

Type: Custom::NLBRegister

Properties:

ServiceToken: !Ref RegisterNlbTargetsLambdaArn

TargetArn: !Ref InternalServiceTargetGroupArn

TargetIp: !GetAtt Master2.PrivateIp

EtcdSrvRecords:

Condition: DoDns

Type: AWS::Route53::RecordSet

Properties:

HostedZoneId: !Ref PrivateHostedZoneId

Name: !Join [".", ["\_etcd-server-ssl.\_tcp", !Ref PrivateHostedZoneName]]

ResourceRecords:

- !Join [  
 " ",  
 ["0 10 2380", !Join [".", ["etcd-0", !Ref PrivateHostedZoneName]]],  
 ]

- !Join [  
 " ",  
 ["0 10 2380", !Join [".", ["etcd-1", !Ref PrivateHostedZoneName]]],  
 ]

- !Join [  
 " ",  
 ["0 10 2380", !Join [".", ["etcd-2", !Ref PrivateHostedZoneName]]],  
 ]

TTL: 60

Type: SRV

Etcd0Record:

Condition: DoDns

Type: AWS::Route53::RecordSet

Properties:

HostedZoneId: !Ref PrivateHostedZoneId

Name: !Join [".", ["etcd-0", !Ref PrivateHostedZoneName]]

ResourceRecords:

- !GetAtt Master0.PrivateIp

TTL: 60

Type: A

Etcd1Record:

Condition: DoDns

Type: AWS::Route53::RecordSet

Properties:

HostedZoneId: !Ref PrivateHostedZoneId



```
Name: !Join [".", ["etcd-1", !Ref PrivateHostedZoneName]]
ResourceRecords:
- !GetAtt Master1.PrivateIp
TTL: 60
Type: A
```

```
Etcd2Record:
Condition: DoDns
Type: AWS::Route53::RecordSet
Properties:
HostedZoneId: !Ref PrivateHostedZoneId
Name: !Join [".", ["etcd-2", !Ref PrivateHostedZoneName]]
ResourceRecords:
- !GetAtt Master2.PrivateIp
TTL: 60
Type: A
```

```
Outputs:
PrivateIPs:
Description: The control-plane node private IP addresses.
Value:
!Join [
  ",",
  [!GetAtt Master0.PrivateIp, !GetAtt Master1.PrivateIp, !GetAtt Master2.PrivateIp]
]
```

#### 其他资源

- 您可以通过导航 [AWS CloudFormation 控制台](#) 来查看您创建的 CloudFormation 堆栈的详情。

### 1.9.14. 在 AWS 中创建 worker 节点

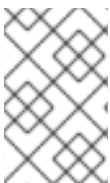
您可以在 Amazon Web Services (AWS) 中创建 worker 节点，供集群使用。

您可以使用提供的 CloudFormation 模板和自定义参数文件创建代表 worker 节点的 AWS 资源堆栈。



#### 重要

CloudFormation 模板会创建一个堆栈，它代表一个 worker 节点。您必须为每个 worker 节点创建一个堆栈。



#### 注意

如果不使用提供的 CloudFormation 模板来创建 worker 节点，您必须检查提供的信息并手动创建基础架构。如果集群没有正确初始化，您可能需要联系红帽支持并提供您的安装日志。

#### 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。

- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。
- 您在 AWS 中创建并配置了 DNS、负载均衡器和监听程序。
- 您在 AWS 中创建了集群所需的安全组和角色。
- 已创建 bootstrap 机器。
- 已创建 control plane 机器。

## 流程

1. 创建一个 JSON 文件，其包含 CloudFormation 模板需要的参数值：

```
[
  {
    "ParameterKey": "InfrastructureName", 1
    "ParameterValue": "mycluster-<random_string>" 2
  },
  {
    "ParameterKey": "RhcOsAmi", 3
    "ParameterValue": "ami-<random_string>" 4
  },
  {
    "ParameterKey": "Subnet", 5
    "ParameterValue": "subnet-<random_string>" 6
  },
  {
    "ParameterKey": "WorkerSecurityGroupID", 7
    "ParameterValue": "sg-<random_string>" 8
  },
  {
    "ParameterKey": "IgnitionLocation", 9
    "ParameterValue": "https://api-int.<cluster_name>.<domain_name>:22623/config/worker"
    10
  },
  {
    "ParameterKey": "CertificateAuthorities", 11
    "ParameterValue": "" 12
  },
  {
    "ParameterKey": "WorkerInstanceProfileName", 13
    "ParameterValue": "" 14
  },
  {
    "ParameterKey": "WorkerInstanceType", 15
    "ParameterValue": "m4.large" 16
  }
]
```

- 1 您的 Ignition 配置文件中为集群编码的集群基础架构名称。

- 2 指定从 Ignition 配置文件元数据中提取的基础架构名称，其格式为 **<cluster-name>-<random-string>**。
- 3 用于 worker 节点的当前 Red Hat Enterprise Linux CoreOS (RHCOS) AMI。
- 4 指定 **AWS::EC2::Image::Id** 值。
- 5 在其中启动 worker 节点的子网，最好是专用子网。
- 6 从 DNS 和负载均衡的 CloudFormation 模板输出的 **PrivateSubnets** 值指定子网。
- 7 与 worker 节点关联的 worker 安全组 ID。
- 8 指定安全组和角色的 CloudFormation 模板输出的 **WorkerSecurityGroupId** 值。
- 9 从中获取 bootstrap Ignition 配置文件的位置。
- 10 指定生成的 Ignition 配置的位置，[https://api-int.<cluster\\_name>.<domain\\_name>:22623/config/worker](https://api-int.<cluster_name>.<domain_name>:22623/config/worker)。
- 11 要使用的 Base64 编码证书颁发机构字符串。
- 12 指定安装目录下 **worker.ign** 文件中的值。这个值是一个长字符串，格式为 **data:text/plain;charset=utf-8;base64,ABC...xYz==**。
- 13 与 worker 节点关联的 IAM 配置集。
- 14 指定安全组和角色的 CloudFormation 模板输出的 **WorkerInstanceProfile** 参数值。
- 15 用于 control plane 机器的 AWS 实例类型。
- 16 允许的值：
  - **m4.large**
  - **m4.xlarge**
  - **m4.2xlarge**
  - **m4.4xlarge**
  - **m4.8xlarge**
  - **m4.10xlarge**
  - **m4.16xlarge**
  - **m5.large**
  - **m5.xlarge**
  - **m5.2xlarge**
  - **m5.4xlarge**
  - **m5.8xlarge**
  - **m5.10xlarge**

- **m5.16xlarge**
- **m6i.xlarge**
- **c4.2xlarge**
- **c4.4xlarge**
- **c4.8xlarge**
- **r4.large**
- **r4.xlarge**
- **r4.2xlarge**
- **r4.4xlarge**
- **r4.8xlarge**
- **r4.16xlarge**



### 重要

如果您的区域中没有 **m4** 实例类型，例如 **eu-west-3**，请改为使用 **m5** 类型。

2. 复制 **worker** 机器的 **CloudFormation** 模板一节中的模板，并将它以 **YAML** 文件形式保存到计算机上。此模板描述了集群所需的网络对象和负载均衡器。
3. 如果您将 **m5** 实例类型指定为 **WorkerInstanceType** 的值，请将该实例类型添加到 **CloudFormation** 模板中的 **WorkerInstanceType.AllowedValues** 参数。
4. 启动 **CloudFormation** 模板，以创建代表 **worker** 节点的 **AWS** 资源堆栈：



### 重要

您必须在一行内输入命令。

```
$ aws cloudformation create-stack --stack-name <name> 1
  --template-body file://<template>.yaml \ 2
  --parameters file://<parameters>.json 3
```

1 **<name>** 是 **CloudFormation** 堆栈的名称，如 **cluster-worker-1**。如果您删除集群，则需要此堆栈的名称。

2 **<template>** 是您保存的 **CloudFormation** 模板 **YAML** 文件的相对路径和名称。

3 **<parameters>** 是 **CloudFormation** 参数 **JSON** 文件的相对路径和名称。

### 输出示例

```
arn:aws:cloudformation:us-east-1:269333783861:stack/cluster-worker-1/729ee301-1c2a-11eb-348f-sd9888c65b59
```



### 注意

CloudFormation 模板会创建一个堆栈，它代表一个 worker 节点。

5. 确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <name>
```

6. 继续创建 worker 堆栈，直到为集群创建了充足的 worker 机器。您可以通过引用同一模板和参数文件并指定不同的堆栈名称来创建额外的 worker 堆栈。



### 重要

您必须至少创建两台 worker 机器，因此您必须创建至少两个使用此 CloudFormation 模板的堆栈。

#### 1.9.14.1. worker 机器的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署 OpenShift Container Platform 集群所需的 worker 机器。

##### 例 1.30. worker 机器的 CloudFormation 模板

```
AWSTemplateFormatVersion: 2010-09-09
Description: Template for OpenShift Cluster Node Launch (EC2 worker instance)

Parameters:
  InfrastructureName:
    AllowedPattern: ^([a-zA-Z][a-zA-Z0-9\-\]{0,26})$
    MaxLength: 27
    MinLength: 1
    ConstraintDescription: Infrastructure name must be alphanumeric, start with a letter, and have a
maximum of 27 characters.
    Description: A short, unique cluster ID used to tag nodes for the kubelet cloud provider.
    Type: String
  RhcosAmi:
    Description: Current Red Hat Enterprise Linux CoreOS AMI to use for bootstrap.
    Type: AWS::EC2::Image::Id
  Subnet:
    Description: The subnets, recommend private, to launch the master nodes into.
    Type: AWS::EC2::Subnet::Id
  WorkerSecurityGroupId:
    Description: The master security group ID to associate with master nodes.
    Type: AWS::EC2::SecurityGroup::Id
  IgnitionLocation:
    Default: https://api-int.$CLUSTER_NAME.$DOMAIN:22623/config/worker
    Description: Ignition config file location.
    Type: String
  CertificateAuthorities:
    Default: data:text/plain;charset=utf-8;base64,ABC...xYz==
    Description: Base64 encoded certificate authority string to use.
```

Type: String

WorkerInstanceProfileName:

Description: IAM profile to associate with master nodes.

Type: String

WorkerInstanceType:

Default: m5.large

Type: String

AllowedValues:

- "m4.large"
- "m4.xlarge"
- "m4.2xlarge"
- "m4.4xlarge"
- "m4.10xlarge"
- "m4.16xlarge"
- "m5.large"
- "m5.xlarge"
- "m5.2xlarge"
- "m5.4xlarge"
- "m5.8xlarge"
- "m5.12xlarge"
- "m5.16xlarge"
- "m5a.large"
- "m5a.xlarge"
- "m5a.2xlarge"
- "m5a.4xlarge"
- "m5a.8xlarge"
- "m5a.10xlarge"
- "m5a.16xlarge"
- "c4.large"
- "c4.xlarge"
- "c4.2xlarge"
- "c4.4xlarge"
- "c4.8xlarge"
- "c5.large"
- "c5.xlarge"
- "c5.2xlarge"
- "c5.4xlarge"
- "c5.9xlarge"
- "c5.12xlarge"
- "c5.18xlarge"
- "c5.24xlarge"
- "c5a.large"
- "c5a.xlarge"
- "c5a.2xlarge"
- "c5a.4xlarge"
- "c5a.8xlarge"
- "c5a.12xlarge"
- "c5a.16xlarge"
- "c5a.24xlarge"
- "r4.large"
- "r4.xlarge"
- "r4.2xlarge"
- "r4.4xlarge"
- "r4.8xlarge"
- "r4.16xlarge"
- "r5.large"

- "r5.xlarge"
- "r5.2xlarge"
- "r5.4xlarge"
- "r5.8xlarge"
- "r5.12xlarge"
- "r5.16xlarge"
- "r5.24xlarge"
- "r5a.large"
- "r5a.xlarge"
- "r5a.2xlarge"
- "r5a.4xlarge"
- "r5a.8xlarge"
- "r5a.12xlarge"
- "r5a.16xlarge"
- "r5a.24xlarge"
- "t3.large"
- "t3.xlarge"
- "t3.2xlarge"
- "t3a.large"
- "t3a.xlarge"
- "t3a.2xlarge"

Metadata:

AWS::CloudFormation::Interface:

ParameterGroups:

- Label:

default: "Cluster Information"

Parameters:

- InfrastructureName

- Label:

default: "Host Information"

Parameters:

- WorkerInstanceType

- RhcosAmi

- IgnitionLocation

- CertificateAuthorities

- WorkerSecurityGroupId

- WorkerInstanceProfileName

- Label:

default: "Network Configuration"

Parameters:

- Subnet

ParameterLabels:

Subnet:

default: "Subnet"

InfrastructureName:

default: "Infrastructure Name"

WorkerInstanceType:

default: "Worker Instance Type"

WorkerInstanceProfileName:

default: "Worker Instance Profile Name"

RhcosAmi:

default: "Red Hat Enterprise Linux CoreOS AMI ID"

IgnitionLocation:

default: "Worker Ignition Source"

CertificateAuthorities:

```

    default: "Ignition CA String"
  WorkerSecurityGroupId:
    default: "Worker Security Group ID"

Resources:
  Worker0:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref RHCOSAmi
      BlockDeviceMappings:
        - DeviceName: /dev/xvda
          Ebs:
            VolumeSize: "120"
            VolumeType: "gp2"
      IamInstanceProfile: !Ref WorkerInstanceProfileName
      InstanceType: !Ref WorkerInstanceType
      NetworkInterfaces:
        - AssociatePublicIp: "false"
          DeviceIndex: "0"
          GroupSet:
            - !Ref "WorkerSecurityGroup"
          SubnetId: !Ref "Subnet"
      UserData:
        Fn::Base64: !Sub
          - '{"ignition":{"config":{"merge":[{"source":"${SOURCE}"}]},"security":{"tls":{"certificateAuthorities":[{"source":"${CA_BUNDLE}"}]},"version":"3.1.0"}}}'
          - {
              SOURCE: !Ref IgnitionLocation,
              CA_BUNDLE: !Ref CertificateAuthorities,
            }
      Tags:
        - Key: !Join ["", ["kubernetes.io/cluster/", !Ref InfrastructureName]]
          Value: "shared"

Outputs:
  PrivateIP:
    Description: The compute node private IP address.
    Value: !GetAtt Worker0.PrivateIp

```

## 其他资源

- 您可以通过导航 [AWS CloudFormation 控制台](#) 来查看您创建的 CloudFormation 堆栈的详情。

## 1.9.15. 使用用户置备的基础架构在 AWS 上初始化 bootstrap 序列

在 Amazon Web Services (AWS) 中创建所有所需的基础架构后，您可以启动初始化 OpenShift Container Platform control plane 的 bootstrap 序列。

### 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。



- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。
- 您在 AWS 中创建并配置了 DNS、负载均衡器和监听程序。
- 您在 AWS 中创建了集群所需的安全组和角色。
- 已创建 bootstrap 机器。
- 已创建 control plane 机器。
- 已创建 worker 节点。

## 流程

1. 更改为包含安装程序的目录，并启动初始化 OpenShift Container Platform control plane 的 bootstrap 过程：

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

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

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

## 输出示例

```
INFO Waiting up to 20m0s for the Kubernetes API at
https://api.mycluster.example.com:6443...
INFO API v1.19.0+9f84db3 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
INFO Time elapsed: 1s
```

如果命令退出时没有 **FATAL** 警告，则 OpenShift Container Platform control plane 已被初始化。



### 注意

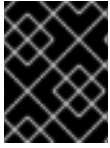
在 control plane 初始化后，它会设置计算节点，并以 Operator 的形式安装其他服务。

## 其他资源

- 如需了解在 OpenShift Container Platform 安装过程中监控安装、bootstrap 和 control plane 日志的详细信息，请参阅[监控安装进度](#)。
- 如需有关对 bootstrap 过程进行故障排除的信息，请参阅[收集 bootstrap 节点诊断数据](#)。
- 您可以使用 [AWS EC2 控制台](#) 查看正在运行的实例的详情。

## 1.9.16. 通过下载二进制文件安装 OpenShift CLI

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



### 重要

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

#### 1.9.16.1. 在 Linux 上安装 OpenShift CLI

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

##### 流程

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

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

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

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

#### 1.9.16.2. 在 Windows 上安装 OpenShift CLI

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

##### 流程

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

```
C:\> path
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 1.9.16.3. 在 macOS 上安装 OpenShift CLI

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

#### 流程

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

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 1.9.17. 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
```

### 1.9.18. 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

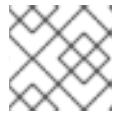
1. 确认集群可以识别这些机器：

```
$ 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：

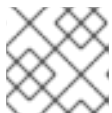
```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{\n"}\n{{end}}' | xargs oc adm certificate approve
```

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](#)。

## 1.9.19. 初始 Operator 配置

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

### 先决条件

- 您的 control plane 已初始化。

### 流程

1. 观察集群组件上线：

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

### 输出示例

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
authentication	4.6.0	True	False	False	3h56m
cloud-credential	4.6.0	True	False	False	29h
cluster-autoscaler	4.6.0	True	False	False	29h
config-operator	4.6.0	True	False	False	6h39m
console	4.6.0	True	False	False	3h59m
csi-snapshot-controller	4.6.0	True	False	False	4h12m

dns	4.6.0	True	False	False	4h15m
etcd	4.6.0	True	False	False	29h
image-registry	4.6.0	True	False	False	3h59m
ingress	4.6.0	True	False	False	4h30m
insights	4.6.0	True	False	False	29h
kube-apiserver	4.6.0	True	False	False	29h
kube-controller-manager	4.6.0	True	False	False	29h
kube-scheduler	4.6.0	True	False	False	29h
kube-storage-version-migrator	4.6.0	True	False	False	4h2m
machine-api	4.6.0	True	False	False	29h
machine-approver	4.6.0	True	False	False	6h34m
machine-config	4.6.0	True	False	False	3h56m
marketplace	4.6.0	True	False	False	4h2m
monitoring	4.6.0	True	False	False	6h31m
network	4.6.0	True	False	False	29h
node-tuning	4.6.0	True	False	False	4h30m
openshift-apiserver	4.6.0	True	False	False	3h56m
openshift-controller-manager	4.6.0	True	False	False	4h36m
openshift-samples	4.6.0	True	False	False	4h30m
operator-lifecycle-manager	4.6.0	True	False	False	29h
operator-lifecycle-manager-catalog	4.6.0	True	False	False	29h
operator-lifecycle-manager-packageserver	4.6.0	True	False	False	3h59m
service-ca	4.6.0	True	False	False	29h
storage	4.6.0	True	False	False	4h30m

## 2. 配置不可用的 Operator。

### 1.9.19.1. 镜像 registry 存储配置

Amazon Web Services 提供默认存储，这意味着 Image Registry Operator 在安装后可用。但是，如果 Registry Operator 无法创建 S3 存储桶并自动配置存储，您需要手工配置 registry 存储。

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

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

您可以在 AWS 中为用户置备的基础架构配置 registry 存储，以将 OpenShift Container Platform 部署到隐藏的区域。请参阅[AWS 用户置备的基础架构配置 registry](#)。

#### 1.9.19.1.1. 为使用用户置备的基础架构的 AWS 配置 registry 存储

在安装过程中，使用您的云凭据就可以创建一个 Amazon S3 存储桶，Registry Operator 将会自动配置存储。

如果 Registry Operator 无法创建 S3 存储桶或自动配置存储，您可以按照以下流程创建 S3 存储桶并配置存储。

#### 先决条件

- 在带有用户置备的基础架构的 AWS 上有一个集群。
- 对于 Amazon S3 存储，secret 应该包含以下两个键：
  - **REGISTRY\_STORAGE\_S3\_ACCESSKEY**

- **REGISTRY\_STORAGE\_S3\_SECRETKEY**

## 流程

如果 Registry Operator 无法创建 S3 存储桶并自动配置存储，请进行以下操作。

1. 设置一个 [Bucket Lifecycle Policy](#) 用来终止已有一天之久的未完成的分段上传操作。
2. 在 `configs.imageregistry.operator.openshift.io/cluster` 中输入存储配置：

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

## 配置示例

```
storage:
  s3:
    bucket: <bucket-name>
    region: <region-name>
```



### 警告

为了保护 AWS 中 registry 镜像的安全，[阻止对 S3 存储桶的公共访问](#)。

### 1.9.19.1.2. 在非生产集群中配置镜像 registry 存储

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

## 流程

- 将镜像 registry 存储设置为空目录：

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



### 警告

仅可为非生产集群配置这个选项。

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

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



等待几分钟，然后再次运行该命令。

### 1.9.20. 删除 bootstrap 资源：

完成集群的初始 Operator 配置后，从 Amazon Web Services (AWS) 中删除 bootstrap 资源。

先决条件

- 已为集群完成初始的 Operator 配置。

流程

1. 删除 bootstrap 资源。如果您使用了 CloudFormation 模板，请[删除其堆栈](#)：

- 使用 AWS CLI 删除堆栈：

```
$ aws cloudformation delete-stack --stack-name <name> 1
```

**1** <name> 是 bootstrap 堆栈的名称。

- 使用 [AWS CloudFormation 控制台](#) 删除堆栈。

### 1.9.21. 创建 Ingress DNS 记录

如果您删除了 DNS 区配置，请手动创建指向 Ingress 负载均衡器的 DNS 记录。您可以创建一个 wildcard 记录或具体的记录。以下流程使用了 A 记录，但您可以使用其他所需记录类型，如 CNAME 或别名。

先决条件

- 已在 Amazon Web Services (AWS) 上安装了使用您置备的基础架构的 OpenShift Container Platform 集群。
- 已安装 OpenShift CLI (**oc**)。
- 安装了 **jq** 软件包。
- 您下载了 AWS CLI 并安装到您的计算机上。请参阅[使用捆绑安装程序 \(Linux、macOS 或 Unix\) 安装 AWS CLI](#)的文档。

流程

1. 决定要创建的路由。

- 要创建一个 wildcard 记录，请使用 **\*.apps.<cluster\_name>.<domain\_name>**，其中 **<cluster\_name>** 是集群名称，**<domain\_name>** 是 OpenShift Container Platform 集群的 Route 53 基域。
- 要创建特定的记录，您必须为集群使用的每个路由创建一个记录，如下所示：

```
$ oc get --all-namespaces -o jsonpath='{range .items[*]}{range .status.ingress[*]}{.host}\n\n'} routes
```

输出示例

-

```

oauth-openshift.apps.<cluster_name>.<domain_name>
console-openshift-console.apps.<cluster_name>.<domain_name>
downloads-openshift-console.apps.<cluster_name>.<domain_name>
alertmanager-main-openshift-monitoring.apps.<cluster_name>.<domain_name>
grafana-openshift-monitoring.apps.<cluster_name>.<domain_name>
prometheus-k8s-openshift-monitoring.apps.<cluster_name>.<domain_name>

```

- 获取 Ingress Operator 负载均衡器状态，并记录其使用的外部 IP 地址值，如 **EXTERNAL-IP** 列所示：

```
$ oc -n openshift-ingress get service router-default
```

### 输出示例

```

NAME          TYPE          CLUSTER-IP    EXTERNAL-IP          PORT(S)
AGE
router-default LoadBalancer 172.30.62.215  ab3...28.us-east-2.elb.amazonaws.com
80:31499/TCP,443:30693/TCP 5m

```

- 为负载均衡器定位托管区 ID：

```
$ aws elb describe-load-balancers | jq -r '.LoadBalancerDescriptions[] | select(.DNSName == "<external_ip>").CanonicalHostedZoneNameID' 1
```

- 对于 **<external\_ip>**，请指定您获取的 Ingress Operator 负载均衡器的外部 IP 地址值。

### 输出示例

```
Z3AADJGX6KTTL2
```

这个命令的输出是负载均衡器托管区 ID。

- 获取集群域的公共托管区 ID：

```

$ aws route53 list-hosted-zones-by-name \
  --dns-name "<domain_name>" 1
  --query 'HostedZones[? Config.PrivateZone != `true` && Name ==
`<domain_name>.`].Id' 2
  --output text

```

- 对于 **<domain\_name>**，请为 OpenShift Container Platform 集群指定 Route 53 基域。

### 输出示例

```
/hostedzone/Z3URY6TWQ91KVV
```

命令输出中会显示您的域的公共托管区 ID。在本例中是 **Z3URY6TWQ91KVV**。

- 在您的私有区中添加别名记录：

```
$ aws route53 change-resource-record-sets --hosted-zone-id "<private_hosted_zone_id>" --
```

```
change-batch '{
> "Changes": [
> {
>   "Action": "CREATE",
>   "ResourceRecordSet": {
>     "Name": "\\052.apps.<cluster_domain>",
>     "Type": "A",
>     "AliasTarget": {
>       "HostedZoneId": "<hosted_zone_id>",
>       "DNSName": "<external_ip>.",
>       "EvaluateTargetHealth": false
>     }
>   }
> }
> ]
> }'
```

- ❶ 对于 **<private\_hosted\_zone\_id>**，指定 DNS 和负载均衡的 CloudFormation 模板输出的值。
- ❷ 对于 **<cluster\_domain>**，请指定用于 OpenShift Container Platform 集群的域或子域。
- ❸ 对于 **<hosted\_zone\_id>**，请为您获得的负载均衡器指定公共托管区 ID。
- ❹ 对于 **<external\_ip>**，请指定 Ingress Operator 负载均衡器的外部 IP 地址值。请确定在该参数数值中包含最后的句点 (.)。

#### 6. 在您的公共区中添加记录：

```
$ aws route53 change-resource-record-sets --hosted-zone-id "<public_hosted_zone_id>" --
change-batch '{
> "Changes": [
> {
>   "Action": "CREATE",
>   "ResourceRecordSet": {
>     "Name": "\\052.apps.<cluster_domain>",
>     "Type": "A",
>     "AliasTarget": {
>       "HostedZoneId": "<hosted_zone_id>",
>       "DNSName": "<external_ip>.",
>       "EvaluateTargetHealth": false
>     }
>   }
> }
> ]
> }'
```

- ❶ 对于 **<public\_hosted\_zone\_id>**，请为您的域指定公共托管区。
- ❷ 对于 **<cluster\_domain>**，请指定用于 OpenShift Container Platform 集群的域或子域。
- ❸ 对于 **<hosted\_zone\_id>**，请为您获得的负载均衡器指定公共托管区 ID。
- ❹ 对于 **<external\_ip>**，请指定 Ingress Operator 负载均衡器的外部 IP 地址值。请确定在该参数数值中包含最后的句点 (.)。

### 1.9.22. 在用户置备的基础架构上完成 AWS 安装

在用户置备的基础架构 Amazon Web Service (AWS) 上启动 OpenShift Container Platform 安装后，监视进程并等待安装完成。

#### 先决条件

- 您在用户置备的 AWS 基础架构上为 OpenShift Container Platform 集群删除了 bootstrap 节点。
- 已安装 **oc** CLI。

#### 流程

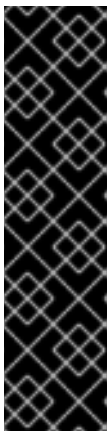
- 在包含安装程序的目录中完成集群安装：

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

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

#### 输出示例

```
INFO Waiting up to 40m0s for the cluster at https://api.mycluster.example.com:6443 to
initialize...
INFO Waiting up to 10m0s for the openshift-console route to be created...
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-Fe5en-ymBEc-
Wt6NL"
INFO Time elapsed: 1s
```



#### 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 Ignition 配置文件生成后的 12 小时内使用它们，因为 24 小时的证书会在集群安装后的 16 小时到 22 小时进行轮转。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因为执行了证书更新而导致安装失败的问题。

### 1.9.23. 使用 Web 控制台登录到集群

**kubeadmin** 用户默认在 OpenShift Container Platform 安装后存在。您可以使用 OpenShift Container Platform Web 控制台以 **kubeadmin** 用户身份登录集群。

## 先决条件

- 有访问安装主机的访问权限。
- 您完成了集群安装，所有集群 Operator 都可用。

## 流程

1. 从安装主机上的 **kubeadmin -password** 文件中获取 kubeadmin 用户的密码：

```
$ cat <installation_directory>/auth/kubeadmin-password
```



### 注意

另外，您还可以从安装主机上的 **<installation\_directory>/openshift\_install.log** 日志文件获取 **kubeadmin** 密码。

2. 列出 OpenShift Container Platform Web 控制台路由：

```
$ oc get routes -n openshift-console | grep 'console-openshift'
```



### 注意

另外，您还可以从安装主机上的 **<installation\_directory>/openshift\_install.log** 日志文件获取 OpenShift Container Platform 路由。

## 输出示例

```
console    console-openshift-console.apps.<cluster_name>.<base_domain>    console
https reencrypt/Redirect None
```

3. 在 Web 浏览器中导航到上一命令输出中包括的路由，以 **kubeadmin** 用户身份登录。

## 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](#)。

## 1.9.24. 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](#)。

### 1.9.25. 其他资源

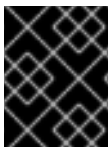
- 如需有关 AWS CloudFormation 堆栈的更多信息，请参阅 [AWS 文档中的使用堆栈](#)。

### 1.9.26. 后续步骤

- [验证安装](#)。
- [自定义集群](#)。
- 如果需要，您可以[选择不使用远程健康报告](#)。
- 如果需要，您可以[删除云供应商凭证](#)。

## 1.10. 在带有用户置备的受限网络中的 AWS 上安装集群

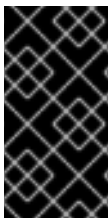
在 OpenShift Container Platform 版本 4.6 中，您可以使用您提供的基础架构和安装发行内容的内部镜像在 Amazon Web Services (AWS) 上安装集群。



### 重要

虽然您可以使用镜像安装发行内容安装 OpenShift Container Platform 集群，但您的集群仍需要访问互联网才能使用 AWS API。

创建此基础架构的一种方法是使用提供的 CloudFormation 模板。您可以修改模板来自定义基础架构，或使用其包含的信息来按照公司策略创建 AWS 对象。

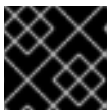


### 重要

进行用户置备的基础架构安装的步骤仅作为示例。使用您提供的基础架构安装集群需要了解云供应商和 OpenShift Container Platform 安装过程。提供的几个 CloudFormation 模板可帮助完成这些步骤，或者帮助您自行建模。您也可以自由选择通过其他方法创建所需的资源；模板仅作参考之用。

### 1.10.1. 先决条件

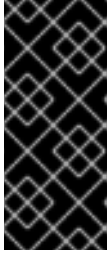
- 您在[镜像主机上创建了一个镜像 registry](#)，并获取您的 OpenShift Container Platform 版本的 `imageContentSources` 数据。



### 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。

- 您可以参阅有关 [OpenShift Container Platform 安装和更新流程](#) 的详细信息。
- 已将 AWS 帐户配置为托管集群。



### 重要

如果您的计算机上存储有 AWS 配置集，则不要在使用多因素验证设备的同时使用您生成的临时会话令牌。在集群的整个生命周期中，集群会持续使用您的当前 AWS 凭证来创建 AWS 资源，因此您必须使用基于密钥的长期凭证。要生成适当的密钥，请参阅 AWS 文档中的[管理 IAM 用户的访问密钥](#)。您可在运行安装程序时提供密钥。

- 您下载了 AWS CLI 并安装到您的计算机上。请参阅 AWS 文档中的[使用捆绑安装程序（Linux、macOS 或 Unix）安装 AWS CLI](#)。
- 如果使用防火墙并计划使用 Telemetry 服务，需要 [将防火墙配置为允许集群需要访问的站点](#)。



### 注意

如果您要配置代理，请务必也要查看此站点列表。

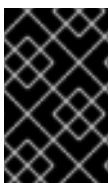
- 如果不允许系统管理身份和访问管理（IAM），集群管理员可以 [手动创建和维护 IAM 凭证](#)。手动模式也可以用于云 IAM API 无法访问的环境中。

## 1.10.2. 关于在受限网络中安装

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

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

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



### 重要

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

### 1.10.2.1. 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。

## 1.10.3. 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。

#### 1.10.4. 所需的 AWS 基础架构组件

要在 Amazon Web Services (AWS) 中用户置备的基础架构上安装 OpenShift Container Platform，您必须手动创建机器及其支持的基础架构。

如需有关不同平台集成测试的更多信息，请参阅 [OpenShift Container Platform 4.x Tested Integrations](#) 页面。

通过使用提供的 CloudFormation 模板，您可以创建代表以下组件的 AWS 资源堆栈：

- 一个 AWS Virtual Private Cloud (VPC)
- 网络和负载均衡组件
- 安全组和角色
- 一个 OpenShift Container Platform bootstrap 节点
- OpenShift Container Platform control plane 节点
- 一个 OpenShift Container Platform 计算节点

或者，您可以手动创建组件，也可以重复使用满足集群要求的现有基础架构。查看 CloudFormation 模板，了解组件如何相互连接的更多详情。

##### 1.10.4.1. 集群机器

以下机器需要 `AWS::EC2::Instance` 对象：

- bootstrap 机器。安装过程中需要此机器，但可在集群部署后删除。
- 三个 control plane 机器。control plane 机器不受机器集的管控。
- 计算机器。在安装过程中创建至少两台计算（compute）机器（也称为 worker 机器）。这些机器不受机器集的管控。

您可以通过提供的 CloudFormation 模板，为集群机器使用以下实例类型。



### 重要

如果您的区域中没有 `m4` 实例类型，例如 `eu-west-3`，请改为使用 `m5` 类型。



表 1.28. 机器的实例类型

实例类型	bootstrap	Control plane	Compute
<b>i3.large</b>	x		
<b>m4.large</b>			x
<b>m4.xlarge</b>		x	x
<b>m4.2xlarge</b>		x	x
<b>m4.4xlarge</b>		x	x
<b>m4.8xlarge</b>		x	x
<b>m4.10xlarge</b>		x	x
<b>m4.16xlarge</b>		x	x
<b>m5.large</b>			x
<b>m5.xlarge</b>		x	x
<b>m5.2xlarge</b>		x	x
<b>m5.4xlarge</b>		x	x
<b>m5.8xlarge</b>		x	x
<b>m5.10xlarge</b>		x	x
<b>m5.16xlarge</b>		x	x
<b>m6i.xlarge</b>		x	x
<b>c4.2xlarge</b>		x	x
<b>c4.4xlarge</b>		x	x
<b>c4.8xlarge</b>		x	x
<b>r4.large</b>			x
<b>r4.xlarge</b>		x	x
<b>r4.2xlarge</b>		x	x

实例类型	bootstrap	Control plane	Compute
<b>r4.4xlarge</b>		x	x
<b>r4.8xlarge</b>		x	x
<b>r4.16xlarge</b>		x	x

您可能能够使用符合这些实例类型规格的其他实例类型。

#### 1.10.4.2. 其他基础架构组件

- VPC
- DNS 条目
- 负载均衡器（典型或网络）和监听器
- 公共和专用路由 53 区域
- 安全组
- IAM 角色
- S3 存储桶

如果您在断开连接的环境或使用代理的环境中工作，则无法访问 EC2 和 ELB 端点的公共 IP 地址。要访问这些端点，您必须创建一个 VPC 端点，并将其附加到集群使用的子网。创建以下端点：

- **ec2.<region>.amazonaws.com**
- **elasticloadbalancing.<region>.amazonaws.com**
- **s3.<region>.amazonaws.com**

#### 所需的 VPC 组件

您必须提供合适的 VPC 和子网，以便与您的机器通信。

组件	AWS 类型	描述
VPC	<ul style="list-style-type: none"> <li>• <b>AWS::EC2::VPC</b></li> <li>• <b>AWS::EC2::VPCEndpoint</b></li> </ul>	您必须提供一个公共 VPC 供集群使用。VPC 使用引用每个子网的路由表的端点，以改进与托管在 S3 中的 registry 的通信。
公共子网	<ul style="list-style-type: none"> <li>• <b>AWS::EC2::Subnet</b></li> <li>• <b>AWS::EC2::SubnetNetworkACLAssociation</b></li> </ul>	您的 VPC 必须有 1 到 3 个可用区的公共子网，并将其与适当的入口规则关联。

组件	AWS 类型	描述	
互联网网关	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::InternetGateway</b></li> <li>● <b>AWS::EC2::VPCGatewayAttachment</b></li> <li>● <b>AWS::EC2::RouteTable</b></li> <li>● <b>AWS::EC2::Route</b></li> <li>● <b>AWS::EC2::SubnetRouteTableAssociation</b></li> <li>● <b>AWS::EC2::NatGateway</b></li> <li>● <b>AWS::EC2::EIP</b></li> </ul>	您必须有一个公共互联网网关，以及附加到 VPC 的公共路由。在提供的模板中，每个公共子网都有一个具有 EIP 地址的 NAT 网关。这些 NAT 网关允许集群资源（如专用子网实例）访问互联网，而有些受限网络或代理场景则不需要它们。	
网络访问控制	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::NetworkAcl</b></li> <li>● <b>AWS::EC2::NetworkAclEntry</b></li> </ul>	您必须允许 VPC 访问下列端口：	
		<b>端口</b>	<b>原因</b>
		<b>80</b>	入站 HTTP 流量
		<b>443</b>	入站 HTTPS 流量
		<b>22</b>	入站 SSH 流量
		<b>1024 - 65535</b>	入站临时流量
	<b>0 - 65535</b>	出站临时流量	
专用子网	<ul style="list-style-type: none"> <li>● <b>AWS::EC2::Subnet</b></li> <li>● <b>AWS::EC2::RouteTable</b></li> <li>● <b>AWS::EC2::SubnetRouteTableAssociation</b></li> </ul>	您的 VPC 可以具有私有子网。提供的 CloudFormation 模板可为 1 到 3 个可用区创建专用子网。如果您使用专用子网，必须为其提供适当的路由和表。	

## 所需的 DNS 和负载均衡组件

您的 DNS 和负载均衡器配置需要使用公共托管区，并可使用类似安装程序使用的专用托管区（如果安装程序配备了集群的基础架构）。您必须创建一个解析到负载均衡器的 DNS 条目。**api.<cluster\_name>.<domain>** 的条目必须指向外部负载均衡器，**api-int.<cluster\_name>.<domain>** 的条目则必须指向内部负载均衡器。

集群还需要负载均衡器，以及监听端口 6443（用于 Kubernetes API 及其扩展）和端口 22623（用于新机器的 Ignition 配置文件）的监听程序。目标是 control plane 节点（也称为 master 节点）。集群外的客户端和集群内的节点都必须能够访问端口 6443。集群内的节点必须能够访问端口 22623。

组件	AWS 类型	描述
DNS	<b>AWS::Route53::HostedZone</b>	内部 DNS 的托管区。
etcd 记录集	<b>AWS::Route53::RecordSet</b>	control plane 机器的 etcd 注册记录。
公共负载均衡器	<b>AWS::ElasticLoadBalancingV2::LoadBalancer</b>	公共子网的负载均衡器。
外部 API 服务器记录	<b>AWS::Route53::RecordSetGroup</b>	外部 API 服务器的别名记录。
外部监听程序	<b>AWS::ElasticLoadBalancingV2::Listener</b>	为外部负载均衡器监听端口 6443 的监听程序。
外部目标组	<b>AWS::ElasticLoadBalancingV2::TargetGroup</b>	外部负载均衡器的目标组。
专用负载均衡器	<b>AWS::ElasticLoadBalancingV2::LoadBalancer</b>	专用子网的负载均衡器。
内部 API 服务器记录	<b>AWS::Route53::RecordSetGroup</b>	内部 API 服务器的别名记录。
内部监听程序	<b>AWS::ElasticLoadBalancingV2::Listener</b>	为内部负载均衡器监听端口 22623 的监听程序。
内部目标组	<b>AWS::ElasticLoadBalancingV2::TargetGroup</b>	内部负载均衡器的目标组。

组件	AWS 类型	描述
内部监听程序	<b>AWS::ElasticLoadBalancingV2::Listener</b>	为内部负载均衡器监听端口 6443 的监听程序。
内部目标组	<b>AWS::ElasticLoadBalancingV2::TargetGroup</b>	内部负载均衡器的目标组。

## 安全组

control plane 和 worker 机器需要访问下列端口：

组	类型	IP 协议	端口范围
<b>MasterSecurityGroup</b>	<b>AWS::EC2::SecurityGroup</b>	icmp	0
		tcp	22
		tcp	6443
		tcp	22623
<b>WorkerSecurityGroup</b>	<b>AWS::EC2::SecurityGroup</b>	icmp	0
		tcp	22
<b>BootstrapSecurityGroup</b>	<b>AWS::EC2::SecurityGroup</b>	tcp	22
		tcp	19531

## control plane 入口

control plane 机器需要以下入口组。每个入口组都是 **AWS::EC2::SecurityGroupIngress** 资源。

入口组	描述	IP 协议	端口范围
<b>MasterIngressEtcd</b>	etcd	tcp	2379- 2380
<b>MasterIngressVxlan</b>	Vxlan 数据包	udp	4789
<b>MasterIngressWorkerVxlan</b>	Vxlan 数据包	udp	4789

入口组	描述	IP 协议	端口范围
<b>MasterIngress Internal</b>	内部集群通信和 Kubernetes 代理指标	<b>tcp</b>	<b>9000 - 9999</b>
<b>MasterIngress WorkerInternal</b>	内部集群通信	<b>tcp</b>	<b>9000 - 9999</b>
<b>MasterIngress Kube</b>	kubernetes kubelet、调度程序和控制器管理器	<b>tcp</b>	<b>10250 - 10259</b>
<b>MasterIngress WorkerKube</b>	kubernetes kubelet、调度程序和控制器管理器	<b>tcp</b>	<b>10250 - 10259</b>
<b>MasterIngress IngressServices</b>	Kubernetes 入口服务	<b>tcp</b>	<b>30000 - 32767</b>
<b>MasterIngress WorkerIngressServices</b>	Kubernetes 入口服务	<b>tcp</b>	<b>30000 - 32767</b>
<b>MasterIngress Geneve</b>	Geneve 数据包	<b>udp</b>	<b>6081</b>
<b>MasterIngress WorkerGeneve</b>	Geneve 数据包	<b>udp</b>	<b>6081</b>
<b>MasterIngress IpsecIke</b>	IPsec IKE 数据包	<b>udp</b>	<b>500</b>
<b>MasterIngress WorkerIpsecIke</b>	IPsec IKE 数据包	<b>udp</b>	<b>500</b>
<b>MasterIngress IpsecNat</b>	IPsec NAT-T 数据包	<b>udp</b>	<b>4500</b>
<b>MasterIngress WorkerIpsecNat</b>	IPsec NAT-T 数据包	<b>udp</b>	<b>4500</b>
<b>MasterIngress IpsecEsp</b>	IPsec ESP 数据包	<b>50</b>	<b>All</b>
<b>MasterIngress WorkerIpsecEsp</b>	IPsec ESP 数据包	<b>50</b>	<b>All</b>

入口组	描述	IP 协议	端口范围
<b>MasterIngress InternalUDP</b>	内部集群通信	<b>udp</b>	<b>9000 - 9999</b>
<b>MasterIngress WorkerInternal IUDP</b>	内部集群通信	<b>udp</b>	<b>9000 - 9999</b>
<b>MasterIngress IngressService esUDP</b>	Kubernetes 入口服务	<b>udp</b>	<b>30000 - 32767</b>
<b>MasterIngress WorkerIngress ServicesUDP</b>	Kubernetes 入口服务	<b>udp</b>	<b>30000 - 32767</b>

## worker 入口

worker 机器需要以下入口组。每个入口组都是 **AWS::EC2::SecurityGroupIngress** 资源。

入口组	描述	IP 协议	端口范围
<b>WorkerIngress Vxlan</b>	Vxlan 数据包	<b>udp</b>	<b>4789</b>
<b>WorkerIngress WorkerVxlan</b>	Vxlan 数据包	<b>udp</b>	<b>4789</b>
<b>WorkerIngress Internal</b>	内部集群通信	<b>tcp</b>	<b>9000 - 9999</b>
<b>WorkerIngress WorkerInternal I</b>	内部集群通信	<b>tcp</b>	<b>9000 - 9999</b>
<b>WorkerIngress Kube</b>	Kubernetes kubelet、调度程序和控制器管理器	<b>tcp</b>	<b>10250</b>
<b>WorkerIngress WorkerKube</b>	Kubernetes kubelet、调度程序和控制器管理器	<b>tcp</b>	<b>10250</b>
<b>WorkerIngress IngressService es</b>	Kubernetes 入口服务	<b>tcp</b>	<b>30000 - 32767</b>
<b>WorkerIngress WorkerIngress Services</b>	Kubernetes 入口服务	<b>tcp</b>	<b>30000 - 32767</b>

入口组	描述	IP 协议	端口范围
<b>WorkerIngressGeneve</b>	Geneve 数据包	udp	6081
<b>WorkerIngressMasterGeneve</b>	Geneve 数据包	udp	6081
<b>WorkerIngressIpsecIke</b>	IPsec IKE 数据包	udp	500
<b>WorkerIngressMasterIpsecIke</b>	IPsec IKE 数据包	udp	500
<b>WorkerIngressIpsecNat</b>	IPsec NAT-T 数据包	udp	4500
<b>WorkerIngressMasterIpsecNat</b>	IPsec NAT-T 数据包	udp	4500
<b>WorkerIngressIpsecEsp</b>	IPsec ESP 数据包	50	All
<b>WorkerIngressMasterIpsecEsp</b>	IPsec ESP 数据包	50	All
<b>WorkerIngressInternalUDP</b>	内部集群通信	udp	9000 - 9999
<b>WorkerIngressMasterInternalUDP</b>	内部集群通信	udp	9000 - 9999
<b>WorkerIngressIngressServicesUDP</b>	Kubernetes 入口服务	udp	30000 - 32767
<b>WorkerIngressMasterIngressServicesUDP</b>	Kubernetes 入口服务	udp	30000 - 32767

## 角色和实例配置集

您必须在 AWS 中为机器授予权限。提供的 CloudFormation 模板为以下 **AWS::IAM::Role** 对象授予机器 **Allow** 权限，并为每一组角色提供一个 **AWS::IAM::InstanceProfile**。如果不使用模板，您可以为机器授予以下宽泛权限或单独权限。



角色	影响	操作	资源
Master	Allow	ec2:*	*
	Allow	elasticloadbalancing:*	*
	Allow	iam:PassRole	*
	Allow	s3:GetObject	*
Worker	Allow	ec2:Describe*	*
bootstrap	Allow	ec2:Describe*	*
	Allow	ec2:AttachVolume	*
	Allow	ec2:DetachVolume	*

#### 1.10.4.3. 证书签名请求管理

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

#### 1.10.4.4. 所需的 AWS 权限



##### 注意

您的 IAM 用户必须在区域 **us-east-1** 中有权限 **tag:GetResources** 来删除基本集群资源。作为 AWS API 的要求的一部分，OpenShift Container Platform 安装程序在此区域中执行各种操作。

将 **AdministratorAccess** 策略附加到您在 Amazon Web Services (AWS) 中创建的 IAM 用户时，授予该用户所有需要的权限。要部署 OpenShift Container Platform 集群的所有组件，IAM 用户需要以下权限：

##### 例 1.31. 安装所需的 EC2 权限

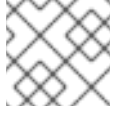
- **tag:TagResources**
- **tag:UntagResources**
- **ec2:AllocateAddress**
- **ec2:AssociateAddress**
- **ec2:AuthorizeSecurityGroupEgress**
- **ec2:AuthorizeSecurityGroupIngress**

- **ec2:CopyImage**
- **ec2:CreateNetworkInterface**
- **ec2:AttachNetworkInterface**
- **ec2:CreateSecurityGroup**
- **ec2:CreateTags**
- **ec2:CreateVolume**
- **ec2>DeleteSecurityGroup**
- **ec2>DeleteSnapshot**
- **ec2>DeleteTags**
- **ec2:DeregisterImage**
- **ec2:DescribeAccountAttributes**
- **ec2:DescribeAddresses**
- **ec2:DescribeAvailabilityZones**
- **ec2:DescribeDhcpOptions**
- **ec2:DescribeImages**
- **ec2:DescribeInstanceAttribute**
- **ec2:DescribeInstanceCreditSpecifications**
- **ec2:DescribeInstances**
- **ec2:DescribeInternetGateways**
- **ec2:DescribeKeyPairs**
- **ec2:DescribeNatGateways**
- **ec2:DescribeNetworkAcls**
- **ec2:DescribeNetworkInterfaces**
- **ec2:DescribePrefixLists**
- **ec2:DescribeRegions**
- **ec2:DescribeRouteTables**
- **ec2:DescribeSecurityGroups**
- **ec2:DescribeSubnets**
- **ec2:DescribeTags**

- **ec2:DescribeVolumes**
- **ec2:DescribeVpcAttribute**
- **ec2:DescribeVpcClassicLink**
- **ec2:DescribeVpcClassicLinkDnsSupport**
- **ec2:DescribeVpcEndpoints**
- **ec2:DescribeVpcs**
- **ec2:GetEbsDefaultKmsKeyId**
- **ec2:ModifyInstanceAttribute**
- **ec2:ModifyNetworkInterfaceAttribute**
- **ec2:ReleaseAddress**
- **ec2:RevokeSecurityGroupEgress**
- **ec2:RevokeSecurityGroupIngress**
- **ec2:RunInstances**
- **ec2:TerminateInstances**

例 1.32. 安装过程中创建网络资源所需的权限

- **ec2:AssociateDhcpOptions**
- **ec2:AssociateRouteTable**
- **ec2:AttachInternetGateway**
- **ec2:CreateDhcpOptions**
- **ec2:CreateInternetGateway**
- **ec2:CreateNatGateway**
- **ec2:CreateRoute**
- **ec2:CreateRouteTable**
- **ec2:CreateSubnet**
- **ec2:CreateVpc**
- **ec2:CreateVpcEndpoint**
- **ec2:ModifySubnetAttribute**
- **ec2:ModifyVpcAttribute**

**注意**

如果您使用现有的 VPC，您的帐户不需要这些权限来创建网络资源。

**例 1.33. 安装所需的 Elastic Load Balancing 权限(ELB)**

- **elasticloadbalancing:AddTags**
- **elasticloadbalancing:ApplySecurityGroupsToLoadBalancer**
- **elasticloadbalancing:AttachLoadBalancerToSubnets**
- **elasticloadbalancing:ConfigureHealthCheck**
- **elasticloadbalancing>CreateLoadBalancer**
- **elasticloadbalancing>CreateLoadBalancerListeners**
- **elasticloadbalancing>DeleteLoadBalancer**
- **elasticloadbalancing:DeregisterInstancesFromLoadBalancer**
- **elasticloadbalancing:DescribeInstanceHealth**
- **elasticloadbalancing:DescribeLoadBalancerAttributes**
- **elasticloadbalancing:DescribeLoadBalancers**
- **elasticloadbalancing:DescribeTags**
- **elasticloadbalancing:ModifyLoadBalancerAttributes**
- **elasticloadbalancing:RegisterInstancesWithLoadBalancer**
- **elasticloadbalancing:SetLoadBalancerPoliciesOfListener**

**例 1.34. 安装所需的 Elastic Load Balancing 权限(ELBv2)**

- **elasticloadbalancing:AddTags**
- **elasticloadbalancing>CreateListener**
- **elasticloadbalancing>CreateLoadBalancer**
- **elasticloadbalancing>CreateTargetGroup**
- **elasticloadbalancing>DeleteLoadBalancer**
- **elasticloadbalancing:DeregisterTargets**
- **elasticloadbalancing:DescribeListeners**
- **elasticloadbalancing:DescribeLoadBalancerAttributes**
- **elasticloadbalancing:DescribeLoadBalancers**

- **elasticloadbalancing:DescribeTargetGroupAttributes**
- **elasticloadbalancing:DescribeTargetHealth**
- **elasticloadbalancing:ModifyLoadBalancerAttributes**
- **elasticloadbalancing:ModifyTargetGroup**
- **elasticloadbalancing:ModifyTargetGroupAttributes**
- **elasticloadbalancing:RegisterTargets**

#### 例 1.35. 安装所需的 IAM 权限

- **iam:AddRoleToInstanceProfile**
- **iam:CreateInstanceProfile**
- **iam:CreateRole**
- **iam:DeleteInstanceProfile**
- **iam>DeleteRole**
- **iam>DeleteRolePolicy**
- **iam:GetInstanceProfile**
- **iam:GetRole**
- **iam:GetRolePolicy**
- **iam:GetUser**
- **iam:ListInstanceProfilesForRole**
- **iam:ListRoles**
- **iam:ListUsers**
- **iam:PassRole**
- **iam:PutRolePolicy**
- **iam:RemoveRoleFromInstanceProfile**
- **iam:SimulatePrincipalPolicy**
- **iam:TagRole**



#### 注意

如果您还没有在 AWS 帐户中创建弹性负载均衡器（ELB），IAM 用户还需要 **iam:CreateServiceLinkedRole** 权限。

## 例 1.36. 安装所需的 Route 53 权限

- **route53:ChangeResourceRecordSets**
- **route53:ChangeTagsForResource**
- **route53:CreateHostedZone**
- **route53>DeleteHostedZone**
- **route53:GetChange**
- **route53:GetHostedZone**
- **route53:ListHostedZones**
- **route53:ListHostedZonesByName**
- **route53:ListResourceRecordSets**
- **route53:ListTagsForResource**
- **route53:UpdateHostedZoneComment**

## 例 1.37. 安装所需的 S3 权限

- **s3:CreateBucket**
- **s3>DeleteBucket**
- **s3:GetAccelerateConfiguration**
- **s3:GetBucketAcl**
- **s3:GetBucketCors**
- **s3:GetBucketLocation**
- **s3:GetBucketLogging**
- **s3:GetBucketObjectLockConfiguration**
- **s3:GetBucketReplication**
- **s3:GetBucketRequestPayment**
- **s3:GetBucketTagging**
- **s3:GetBucketVersioning**
- **s3:GetBucketWebsite**
- **s3:GetEncryptionConfiguration**
- **s3:GetLifecycleConfiguration**

- **s3:GetReplicationConfiguration**
- **s3:ListBucket**
- **s3:PutBucketAcl**
- **s3:PutBucketTagging**
- **s3:PutEncryptionConfiguration**

例 1.38. 集群 Operators 所需的 S3 权限

- **s3:DeleteObject**
- **s3:GetObject**
- **s3:GetObjectAcl**
- **s3:GetObjectTagging**
- **s3:GetObjectVersion**
- **s3:PutObject**
- **s3:PutObjectAcl**
- **s3:PutObjectTagging**

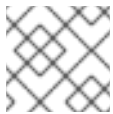
例 1.39. 删除基本集群资源所需的权限

- **autoscaling:DescribeAutoScalingGroups**
- **ec2:DeleteNetworkInterface**
- **ec2:DeleteVolume**
- **elasticloadbalancing:DeleteTargetGroup**
- **elasticloadbalancing:DescribeTargetGroups**
- **iam:DeleteAccessKey**
- **iam:DeleteUser**
- **iam>ListAttachedRolePolicies**
- **iam>ListInstanceProfiles**
- **iam>ListRolePolicies**
- **iam>ListUserPolicies**
- **s3:DeleteObject**
- **s3:ListBucketVersions**

- **tag:GetResources**

#### 例 1.40. 删除网络资源所需的权限

- **ec2:DeleteDhcpOptions**
- **ec2:DeleteInternetGateway**
- **ec2:DeleteNatGateway**
- **ec2:DeleteRoute**
- **ec2:DeleteRouteTable**
- **ec2:DeleteSubnet**
- **ec2:DeleteVpc**
- **ec2:DeleteVpcEndpoints**
- **ec2:DetachInternetGateway**
- **ec2:DisassociateRouteTable**
- **ec2:ReplaceRouteTableAssociation**



#### 注意

如果您使用现有的 VPC，您的帐户不需要这些权限来删除网络资源。

#### 例 1.41. 创建清单所需的额外 IAM 和 S3 权限

- **iam:DeleteAccessKey**
- **iam:DeleteUser**
- **iam:DeleteUserPolicy**
- **iam:GetUserPolicy**
- **iam:ListAccessKeys**
- **iam:PutUserPolicy**
- **iam:TagUser**
- **iam:GetUserPolicy**
- **iam:ListAccessKeys**
- **s3:PutBucketPublicAccessBlock**
- **s3:GetBucketPublicAccessBlock**



- **s3:PutLifecycleConfiguration**
- **s3:HeadBucket**
- **s3:ListBucketMultipartUploads**
- **s3:AbortMultipartUpload**



### 注意

如果要使用 mint 模式管理云供应商凭证，IAM 用户还需要 **iam:CreateAccessKey** and **iam:CreateUser** 权限。

#### 例 1.42. 安装时配额检查的可选项

- **servicequotas:ListAWSDefaultServiceQuotas**

### 1.10.5. 生成 SSH 私钥并将其添加到代理中

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



### 注意

在生产环境中，您需要进行灾难恢复和调试。

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](#)。

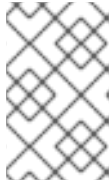
#### 流程

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

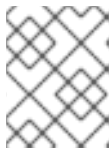
如果您计划在 **x86\_64** 架构中安装使用 FIPS 验证的/Modules in Process 加密库的 OpenShift Container Platform 集群，不要创建使用 **ed25519** 算法的密钥。反之，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

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。如果在您置备的基础架构上安装集群，您必须将此密钥提供给集群的机器。

## 1.10.6. 创建用于 AWS 的安装文件

要使用用户置备的基础架构在 Amazon Web Services (AWS) 上安装 OpenShift Container Platform，您必须生成并修改安装程序部署集群所需的文件，以便集群只创建要使用的机器。您要生成并自定义 **install-config.yaml** 文件、Kubernetes 清单和 Ignition 配置文件。您也可以选择在安装准备阶段首先设置独立的 **var** 分区。

### 1.10.6.1. 可选：创建独立 **/var** 分区

建议安装程序将 OpenShift Container Platform 的磁盘分区保留给安装程序。然而，在有些情况下您可能需要在文件系统的一部分中创建独立分区。

OpenShift Container Platform 支持添加单个分区来将存储附加到 **/var** 分区或 **/var** 的子目录。例如：

- **/var/lib/containers**：保存镜像相关的内容，随着更多镜像和容器添加到系统中，它所占用的存储会增加。

- `/var/lib/etcd` : 保存您可能希望保持独立的数据, 比如 etcd 存储的性能优化。
- `/var` : 保存您希望独立保留的数据, 用于特定目的 (如审计) 。

单独存储 `/var` 目录的内容可方便地根据需要对区域扩展存储, 并可以在以后重新安装 OpenShift Container Platform 时保持该数据地完整。使用这个方法, 您不必再次拉取所有容器, 在更新系统时也无法复制大量日志文件。

因为 `/var` 在进行一个全新的 Red Hat Enterprise Linux CoreOS (RHCOS) 安装前必需存在, 所以这个流程会在 OpenShift Container Platform 安装过程的 `openshift-install` 准备阶段插入的机器配置来设置独立的 `/var` 分区。



### 重要

如果按照以下步骤在此流程中创建独立 `/var` 分区, 则不需要再次创建 Kubernetes 清单和 Ignition 配置文件, 如本节所述。

### 流程

1. 创建存放 OpenShift Container Platform 安装文件的目录 :

```
$ mkdir $HOME/clusterconfig
```

2. 运行 `openshift-install` 在 `manifest` 和 `openshift` 子目录中创建一组文件。在出现提示时回答系统问题 :

```
$ openshift-install create manifests --dir $HOME/clusterconfig
```

### 输出示例

```
? SSH Public Key ...
INFO Credentials loaded from the "myprofile" profile in file "/home/myuser/.aws/credentials"
INFO Consuming Install Config from target directory
INFO Manifests created in: $HOME/clusterconfig/manifests and
$HOME/clusterconfig/openshift
```

3. 可选 : 确认安装程序在 `clusterconfig/openshift` 目录中创建了清单 :

```
$ ls $HOME/clusterconfig/openshift/
```

### 输出示例

```
99_kubeadmin-password-secret.yaml
99_openshift-cluster-api_master-machines-0.yaml
99_openshift-cluster-api_master-machines-1.yaml
99_openshift-cluster-api_master-machines-2.yaml
...
```

4. 创建 `MachineConfig` 对象并将其添加到 `openshift` 目录中的一个文件中。例如, 把文件命名为 `98-var-partition.yaml`, 将磁盘设备名称改为 `worker` 系统中存储设备的名称, 并根据情况设置存储大小。这个示例将 `/var` 目录放在一个单独的分区中 :

```
apiVersion: machineconfiguration.openshift.io/v1
```

```

kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 98-var-partition
spec:
  config:
    ignition:
      version: 3.1.0
    storage:
      disks:
        - device: /dev/<device_name> ❶
          partitions:
            - label: var
              startMiB: <partition_start_offset> ❷
              sizeMiB: <partition_size> ❸
          filesystems:
            - device: /dev/disk/by-partlabel/var
              path: /var
              format: xfs
      systemd:
        units:
          - name: var.mount ❹
            enabled: true
            contents: |
              [Unit]
              Before=local-fs.target
              [Mount]
              What=/dev/disk/by-partlabel/var
              Where=/var
              Options=defaults,prjquota ❺
              [Install]
              WantedBy=local-fs.target

```

- ❶ 要分区的磁盘的存储设备名称。
- ❷ 当在引导磁盘中添加数据分区时，推荐最少使用 25000 MiB (Mebibytes)。root 文件系统会自动重新定义大小使其占据所有可用空间（最多到指定的偏移值）。如果没有指定值，或者指定的值小于推荐的最小值，则生成的 root 文件系统会太小，而在以后进行的 RHCOS 重新安装可能会覆盖数据分区的开始部分。
- ❸ 数据分区的大小（以兆字节为单位）。
- ❹ 挂载单元的名称必须与 **Where=** 指令中指定的目录匹配。例如，对于挂载于 **/var/lib/containers** 上的文件系统，该单元必须命名为 **var-lib-containers.mount**。
- ❺ 对于用于容器存储的文件系统，必须启用 **prjquota** 挂载选项。



### 注意

在创建独立 **/var** 分区时，如果不同的实例类型没有相同的设备名称，则无法将不同的实例类型用于 worker 节点。

5. 再次运行 **openshift-install**，从 **manifest** 和 **openshift** 子目录中的一组文件创建 Ignition 配置：

```
$ openshift-install create ignition-configs --dir $HOME/clusterconfig
$ ls $HOME/clusterconfig/
auth bootstrap.ign master.ign metadata.json worker.ign
```

现在，您可以使用 Ignition 配置文件作为安装程序的输入来安装 Red Hat Enterprise Linux CoreOS (RHCOS) 系统。

### 1.10.6.2. 创建安装配置文件

生成并自定义安装程序部署集群所需的安装配置文件。

#### 先决条件

- 已获取 OpenShift Container Platform 安装程序用于用户置备的基础架构和集群的 pull secret。对于受限网络安装，这些文件位于您的堡垒主机上。
- 使用红帽发布的附带 Red Hat Enterprise Linux CoreOS (RHCOS) AMI 检查您是否将集群部署到一个区域。如果您要部署到需要自定义 AMI 的区域，如 AWS GovCloud 区域，您必须手动创建 `install-config.yaml` 文件。

#### 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 `aws` 作为目标平台。
- iii. 如果计算机上没有保存 AWS 配置集，请为您配置用于运行安装程序的用户输入 AWS 访问密钥 ID 和 secret 访问密钥。



## 注意

AWS 访问密钥 ID 和 secret 访问密钥存储在安装主机上当前用户主目录中的 `~/.aws/credentials` 中。如果文件中不存在导出的配置集凭证，安装程序会提示您输入凭证。您向安装程序提供的所有凭证都存储在文件中。

- iv. 选择要将集群部署到的 AWS 区域。
  - v. 选择您为集群配置的 Route 53 服务的基域。
  - vi. 为集群输入一个描述性名称。
  - vii. 粘贴 [Red Hat OpenShift Cluster Manager 中的 pull secret](#)。
2. 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":{"<local_registry>": {"auth": "<credentials>","email":
"you@example.com"}}}'
```

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

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

```
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----
  ////////////////////////////////////////////////////////////////////
  -----END CERTIFICATE-----
```

- c. 添加镜像内容资源：

```
imageContentSources:
- mirrors:
  - <local_registry>/<local_repository_name>/release
  source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - <local_registry>/<local_repository_name>/release
  source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
```

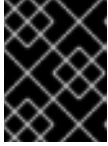
使用命令输出中的 `imageContentSources` 部分来镜像（mirror）仓库，或您从您进入受限网络的介质中的内容时使用的值。

- d. 可选：将发布策略设置为 `Internal`：

```
publish: Internal
```

通过设置这个选项，您可以创建一个内部 Ingress Controller 和一个私有负载均衡器。

3. 可选：备份 `install-config.yaml` 文件。



## 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

- 如需有关 AWS 配置集和凭证配置的更多信息，请参阅 [AWS 文档中的配置和凭证文件设置](#)。

### 1.10.6.3. 在安装过程中配置集群范围代理

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

#### 先决条件

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



## 注意

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

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

- 如果您的集群位于 AWS 上，请将 `ec2.<region>.amazonaws.com`、`elasticloadbalancing.<region>.amazonaws.com` 和 `s3.<region>.amazonaws.com` 端点添加到 VPC 端点。需要这些端点才能完成节点到 AWS EC2 API 的请求。由于代理在容器级别而不是节点级别工作，因此您必须通过 AWS 专用网络将这些请求路由到 AWS EC2 API。在代理服务器中的允许列表中添加 EC2 API 的公共 IP 地址是不够的。

#### 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** 的配置映射来保存额外的 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，将为 **trustedCA** 参数指定的内容与 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。

#### 1.10.6.4. 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 Ignition 配置文件生成后的 12 小时内使用它们，因为 24 小时的证书会在集群安装后的 16 小时到 22 小时进行轮转。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因为执行了证书更新而导致安装失败的问题。

#### 先决条件

- 已获得 OpenShift Container Platform 安装程序。对于受限网络安装，这些文件位于您的堡垒主机上。
- 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> ❶
```

- ❶ 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：

```
$ rm -f <installation_directory>/openshift/99_openshift-cluster-api_master-machines-*.yaml
```

通过删除这些文件，您可以防止集群自动生成 control plane 机器。

3. 删除定义 worker 机器的 Kubernetes 清单文件：

```
$ rm -f <installation_directory>/openshift/99_openshift-cluster-api_worker-machineset-*.yaml
```

由于您要自行创建并管理 worker 机器，因此不需要初始化这些机器。

4. 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。

5. 可选：如果您不希望 [Ingress Operator](#) 代表您创建 DNS 记录，请删除 **<installation\_directory>/manifests/cluster-dns-02-config.yml** DNS 配置文件中的 **privateZone** 和 **publicZone** 部分：

```
apiVersion: config.openshift.io/v1
kind: DNS
metadata:
  creationTimestamp: null
  name: cluster
spec:
  baseDomain: example.openshift.com
  privateZone: ❶
    id: mycluster-100419-private-zone
  publicZone: ❷
    id: example.openshift.com
status: {}
```

- ❶ ❷ 完全删除此部分。

如果您这样做，后续步骤中必须手动添加入口 DNS 记录。

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

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

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

该目录中将生成以下文件：

```

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

```

### 1.10.7. 提取基础架构名称

Ignition 配置文件包含一个唯一集群标识符，您可以使用它在 Amazon Web Services (AWS) 中唯一地标识您的集群。基础架构名称还用于在 OpenShift Container Platform 安装过程中定位适当的 AWS 资源。提供的 CloudFormation 模板包含对此基础架构名称的引用，因此您必须提取它。

#### 先决条件

- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。
- 已为集群生成 Ignition 配置文件。
- 安装了 **jq** 软件包。

#### 流程

- 要从 Ignition 配置文件元数据中提取和查看基础架构名称，请运行以下命令：

```
$ jq -r .infraID <installation_directory>/metadata.json 1
```

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

#### 输出示例

```
openshift-vw9j6 1
```

1 此命令的输出是您的集群名称和随机字符串。

### 1.10.8. 在 AWS 中创建 VPC

您必须在 Amazon Web Services (AWS) 中创建 Virtual Private Cloud (VPC)，供您的 OpenShift Container Platform 集群使用。您可以自定义 VPC 来满足您的要求，包括 VPN 和路由表。

您可以使用提供的 CloudFormation 模板和自定义参数文件创建代表 VPC 的 AWS 资源堆栈。



## 注意

如果不使用提供的 CloudFormation 模板来创建 AWS 基础架构，您必须检查提供的信息并手动创建基础架构。如果集群没有正确初始化，您可能需要联系红帽支持并提供您的安装日志。

## 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。
- 已为集群生成 Ignition 配置文件。

## 流程

1. 创建一个 JSON 文件，其包含模板所需的参数值：

```
[
  {
    "ParameterKey": "VpcCidr", ①
    "ParameterValue": "10.0.0.0/16" ②
  },
  {
    "ParameterKey": "AvailabilityZoneCount", ③
    "ParameterValue": "1" ④
  },
  {
    "ParameterKey": "SubnetBits", ⑤
    "ParameterValue": "12" ⑥
  }
]
```

- ① VPC 的 CIDR 块。
- ② 以 **x.x.x.x/16-24** 格式指定 CIDR 块。
- ③ 在其中部署 VPC 的可用区的数量。
- ④ 指定一个 **1** 到 **3** 之间的整数。
- ⑤ 各个可用区中每个子网的大小。
- ⑥ 指定 **5** 到 **13** 之间的整数，其中 **5** 为 **/27**，**13** 为 **/19**。

2. 复制本主题的 VPC 的 CloudFormation 模板部分中的模板，并将它以 YAML 文件形式保存到计算机上。此模板描述了集群所需的 VPC。
3. 启动 CloudFormation 模板，以创建代表 VPC 的 AWS 资源堆栈：



## 重要

您必须在一行内输入命令。

```
$ aws cloudformation create-stack --stack-name <name> ❶
--template-body file://<template>.yaml ❷
--parameters file://<parameters>.json ❸
```

- ❶ **<name>** 是 CloudFormation 堆栈的名称，如 **cluster-VPC**。如果您删除集群，则需要此堆栈的名称。
- ❷ **<template>** 是您保存的 CloudFormation 模板 YAML 文件的相对路径和名称。
- ❸ **<parameters>** 是 CloudFormation 参数 JSON 文件的相对路径和名称。

### 输出示例

```
arn:aws:cloudformation:us-east-1:269333783861:stack/cluster-vpc/dbedae40-2fd3-11eb-820e-12a48460849f
```

#### 4. 确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <name>
```

在 **StackStatus** 显示 **CREATE\_COMPLETE** 后，输出会显示以下参数的值。您必须将这些参数值提供给您在创建集群时要运行的其他 CloudFormation 模板：

<b>VpcId</b>	您的 VPC ID。
<b>PublicSubnetIds</b>	新公共子网的 ID。
<b>PrivateSubnetIds</b>	新专用子网的 ID。

#### 1.10.8.1. VPC 的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署 OpenShift Container Platform 集群所需的 VPC。

##### 例 1.43. VPC 的 CloudFormation 模板

```
AWSTemplateFormatVersion: 2010-09-09
Description: Template for Best Practice VPC with 1-3 AZs

Parameters:
  VpcCidr:
    AllowedPattern: ^(((0-9){1,3}[0-9]{0,2}|2[0-4][0-9]{0,1}|25[0-5])\.)\{3\}((0-9){1,3}[0-9]{0,1}|1[0-9]{2}|2[0-4][0-9]|25[0-5])(\/(1[6-9]|2[0-4]))$
    ConstraintDescription: CIDR block parameter must be in the form x.x.x.x/16-24.
    Default: 10.0.0.0/16
    Description: CIDR block for VPC.
    Type: String
  AvailabilityZoneCount:
    ConstraintDescription: "The number of availability zones. (Min: 1, Max: 3)"
    MinValue: 1
```

```

    MaxValue: 3
    Default: 1
    Description: "How many AZs to create VPC subnets for. (Min: 1, Max: 3)"
    Type: Number
  SubnetBits:
    ConstraintDescription: CIDR block parameter must be in the form x.x.x.x/19-27.
    MinValue: 5
    MaxValue: 13
    Default: 12
    Description: "Size of each subnet to create within the availability zones. (Min: 5 = /27, Max: 13 =
/19)"
    Type: Number

Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: "Network Configuration"
        Parameters:
          - VpcCidr
          - SubnetBits
      - Label:
          default: "Availability Zones"
        Parameters:
          - AvailabilityZoneCount
    ParameterLabels:
      AvailabilityZoneCount:
        default: "Availability Zone Count"
      VpcCidr:
        default: "VPC CIDR"
      SubnetBits:
        default: "Bits Per Subnet"

Conditions:
  DoAz3: !Equals [3, !Ref AvailabilityZoneCount]
  DoAz2: !Or [!Equals [2, !Ref AvailabilityZoneCount], Condition: DoAz3]

Resources:
  VPC:
    Type: "AWS::EC2::VPC"
    Properties:
      EnableDnsSupport: "true"
      EnableDnsHostnames: "true"
      CidrBlock: !Ref VpcCidr
  PublicSubnet:
    Type: "AWS::EC2::Subnet"
    Properties:
      VpcId: !Ref VPC
      CidrBlock: !Select [0, !Cidr [!Ref VpcCidr, 6, !Ref SubnetBits]]
      AvailabilityZone: !Select
        - 0
        - Fn::GetAZs: !Ref "AWS::Region"
  PublicSubnet2:
    Type: "AWS::EC2::Subnet"
    Condition: DoAz2
    Properties:

```

```
VpcId: !Ref VPC
CidrBlock: !Select [1, !Cidr [!Ref VpcCidr, 6, !Ref SubnetBits]]
AvailabilityZone: !Select
- 1
- Fn::GetAZs: !Ref "AWS::Region"
PublicSubnet3:
Type: "AWS::EC2::Subnet"
Condition: DoAz3
Properties:
VpcId: !Ref VPC
CidrBlock: !Select [2, !Cidr [!Ref VpcCidr, 6, !Ref SubnetBits]]
AvailabilityZone: !Select
- 2
- Fn::GetAZs: !Ref "AWS::Region"
InternetGateway:
Type: "AWS::EC2::InternetGateway"
GatewayToInternet:
Type: "AWS::EC2::VPCGatewayAttachment"
Properties:
VpcId: !Ref VPC
InternetGatewayId: !Ref InternetGateway
PublicRouteTable:
Type: "AWS::EC2::RouteTable"
Properties:
VpcId: !Ref VPC
PublicRoute:
Type: "AWS::EC2::Route"
DependsOn: GatewayToInternet
Properties:
RouteTableId: !Ref PublicRouteTable
DestinationCidrBlock: 0.0.0.0/0
GatewayId: !Ref InternetGateway
PublicSubnetRouteTableAssociation:
Type: "AWS::EC2::SubnetRouteTableAssociation"
Properties:
SubnetId: !Ref PublicSubnet
RouteTableId: !Ref PublicRouteTable
PublicSubnetRouteTableAssociation2:
Type: "AWS::EC2::SubnetRouteTableAssociation"
Condition: DoAz2
Properties:
SubnetId: !Ref PublicSubnet2
RouteTableId: !Ref PublicRouteTable
PublicSubnetRouteTableAssociation3:
Condition: DoAz3
Type: "AWS::EC2::SubnetRouteTableAssociation"
Properties:
SubnetId: !Ref PublicSubnet3
RouteTableId: !Ref PublicRouteTable
PrivateSubnet:
Type: "AWS::EC2::Subnet"
Properties:
VpcId: !Ref VPC
CidrBlock: !Select [3, !Cidr [!Ref VpcCidr, 6, !Ref SubnetBits]]
AvailabilityZone: !Select
- 0
```

```

- Fn::GetAZs: !Ref "AWS::Region"
PrivateRouteTable:
  Type: "AWS::EC2::RouteTable"
  Properties:
    VpcId: !Ref VPC
PrivateSubnetRouteTableAssociation:
  Type: "AWS::EC2::SubnetRouteTableAssociation"
  Properties:
    SubnetId: !Ref PrivateSubnet
    RouteTableId: !Ref PrivateRouteTable
NAT:
  DependsOn:
  - GatewayToInternet
  Type: "AWS::EC2::NatGateway"
  Properties:
    AllocationId:
      "Fn::GetAtt":
      - EIP
      - AllocationId
    SubnetId: !Ref PublicSubnet
EIP:
  Type: "AWS::EC2::EIP"
  Properties:
    Domain: vpc
Route:
  Type: "AWS::EC2::Route"
  Properties:
    RouteTableId:
      Ref: PrivateRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId:
      Ref: NAT
PrivateSubnet2:
  Type: "AWS::EC2::Subnet"
  Condition: DoAz2
  Properties:
    VpcId: !Ref VPC
    CidrBlock: !Select [4, !Cidr [!Ref VpcCidr, 6, !Ref SubnetBits]]
    AvailabilityZone: !Select
    - 1
    - Fn::GetAZs: !Ref "AWS::Region"
PrivateRouteTable2:
  Type: "AWS::EC2::RouteTable"
  Condition: DoAz2
  Properties:
    VpcId: !Ref VPC
PrivateSubnetRouteTableAssociation2:
  Type: "AWS::EC2::SubnetRouteTableAssociation"
  Condition: DoAz2
  Properties:
    SubnetId: !Ref PrivateSubnet2
    RouteTableId: !Ref PrivateRouteTable2
NAT2:
  DependsOn:
  - GatewayToInternet
  Type: "AWS::EC2::NatGateway"

```

```
Condition: DoAz2
Properties:
  AllocationId:
    "Fn::GetAtt":
      - EIP2
      - AllocationId
  SubnetId: !Ref PublicSubnet2
EIP2:
  Type: "AWS::EC2::EIP"
  Condition: DoAz2
  Properties:
    Domain: vpc
Route2:
  Type: "AWS::EC2::Route"
  Condition: DoAz2
  Properties:
    RouteTableId:
      Ref: PrivateRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId:
      Ref: NAT2
PrivateSubnet3:
  Type: "AWS::EC2::Subnet"
  Condition: DoAz3
  Properties:
    VpcId: !Ref VPC
    CidrBlock: !Select [5, !Cidr [!Ref VpcCidr, 6, !Ref SubnetBits]]
    AvailabilityZone: !Select
      - 2
      - Fn::GetAZs: !Ref "AWS::Region"
PrivateRouteTable3:
  Type: "AWS::EC2::RouteTable"
  Condition: DoAz3
  Properties:
    VpcId: !Ref VPC
PrivateSubnetRouteTableAssociation3:
  Type: "AWS::EC2::SubnetRouteTableAssociation"
  Condition: DoAz3
  Properties:
    SubnetId: !Ref PrivateSubnet3
    RouteTableId: !Ref PrivateRouteTable3
NAT3:
  DependsOn:
    - GatewayToInternet
  Type: "AWS::EC2::NatGateway"
  Condition: DoAz3
  Properties:
    AllocationId:
      "Fn::GetAtt":
        - EIP3
        - AllocationId
    SubnetId: !Ref PublicSubnet3
EIP3:
  Type: "AWS::EC2::EIP"
  Condition: DoAz3
  Properties:
```



```

    Domain: vpc
Route3:
  Type: "AWS::EC2::Route"
  Condition: DoAz3
  Properties:
    RouteTableId:
      Ref: PrivateRouteTable3
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId:
      Ref: NAT3
S3Endpoint:
  Type: AWS::EC2::VPCEndpoint
  Properties:
    PolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Principal: '*'
          Action:
            - '*'
          Resource:
            - '*'
      RouteTableIds:
        - !Ref PublicRouteTable
        - !Ref PrivateRouteTable
        - !If [DoAz2, !Ref PrivateRouteTable2, !Ref "AWS::NoValue"]
        - !If [DoAz3, !Ref PrivateRouteTable3, !Ref "AWS::NoValue"]
      ServiceName: !Join
        - "
        - - com.amazonaws.
          - !Ref 'AWS::Region'
          - .s3
      Vpclid: !Ref VPC

Outputs:
Vpclid:
  Description: ID of the new VPC.
  Value: !Ref VPC
PublicSubnetIds:
  Description: Subnet IDs of the public subnets.
  Value:
    !Join [
      ",",
      [!Ref PublicSubnet, !If [DoAz2, !Ref PublicSubnet2, !Ref "AWS::NoValue"], !If [DoAz3, !Ref
PublicSubnet3, !Ref "AWS::NoValue"]]
    ]
PrivateSubnetIds:
  Description: Subnet IDs of the private subnets.
  Value:
    !Join [
      ",",
      [!Ref PrivateSubnet, !If [DoAz2, !Ref PrivateSubnet2, !Ref "AWS::NoValue"], !If [DoAz3, !Ref
PrivateSubnet3, !Ref "AWS::NoValue"]]
    ]

```

### 1.10.9. 在 AWS 中创建网络和负载均衡组件

您必须在 OpenShift Container Platform 集群可以使用的 Amazon Web Services (AWS) 中配置网络、经典或网络负载均衡。

您可以使用提供的 CloudFormation 模板和自定义参数文件来创建 AWS 资源堆栈。堆栈代表 OpenShift Container Platform 集群所需的网络和负载均衡组件。该模板还创建一个托管区和子网标签。

您可以在单一虚拟私有云(VPC)内多次运行该模板。



#### 注意

如果不使用提供的 CloudFormation 模板来创建 AWS 基础架构，您必须检查提供的信息并手动创建基础架构。如果集群没有正确初始化，您可能需要联系红帽支持并提供您的安装日志。

#### 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。
- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。

#### 流程

1. 获取您在 **install-config.yaml** 文件中为集群指定的 Route 53 基域的托管区 ID。您可以运行以下命令来获取托管区的详细信息：

```
$ aws route53 list-hosted-zones-by-name --dns-name <route53_domain> 1
```

- 1** 对于 **<route53\_domain>**，请指定您为集群生成 **install-config.yaml** 文件时所用的 Route53 基域。

#### 输出示例

```
mycluster.example.com. False 100
HOSTEDZONES 65F8F38E-2268-B835-E15C-AB55336FCBFA
/hostedzone/Z21IXYZABCZ2A4 mycluster.example.com. 10
```

在示例输出中，托管区 ID 为 **Z21IXYZABCZ2A4**。

2. 创建一个 JSON 文件，其包含模板所需的参数值：

```
[
  {
    "ParameterKey": "ClusterName", 1
    "ParameterValue": "mycluster" 2
  },
  {
    "ParameterKey": "InfrastructureName", 3
```

```

    "ParameterValue": "mycluster-<random_string>" 4
  },
  {
    "ParameterKey": "HostedZoneId", 5
    "ParameterValue": "<random_string>" 6
  },
  {
    "ParameterKey": "HostedZoneName", 7
    "ParameterValue": "example.com" 8
  },
  {
    "ParameterKey": "PublicSubnets", 9
    "ParameterValue": "subnet-<random_string>" 10
  },
  {
    "ParameterKey": "PrivateSubnets", 11
    "ParameterValue": "subnet-<random_string>" 12
  },
  {
    "ParameterKey": "VpcId", 13
    "ParameterValue": "vpc-<random_string>" 14
  }
]

```

- 1 一个简短的、代表集群的名称用于主机名等。
- 2 指定您为集群生成 `install-config.yaml` 文件时所用的集群名称。
- 3 您的 Ignition 配置文件中为集群编码的集群基础架构名称。
- 4 指定从 Ignition 配置文件元数据中提取的基础架构名称，其格式为 `<cluster-name>-<random-string>`。
- 5 用来注册目标的 Route 53 公共区 ID。
- 6 指定 Route 53 公共区 ID，其格式与 `Z21IXYZABCZ2A4` 类似。您可以从 AWS 控制台获取这个值。
- 7 用来注册目标的 Route 53 区。
- 8 指定您为集群生成 `install-config.yaml` 文件时所用的 Route 53 基域。请勿包含 AWS 控制台中显示的结尾句点 (.)。
- 9 为 VPC 创建的公共子网。
- 10 指定 VPC 的 CloudFormation 模板输出的 `PublicSubnetIds` 值。
- 11 为 VPC 创建的专用子网。
- 12 指定 VPC 的 CloudFormation 模板输出的 `PrivateSubnetIds` 值。
- 13 为集群创建的 VPC。
- 14 指定 VPC 的 CloudFormation 模板输出的 `VpcId` 值。

- 复制本主题的网络和负载均衡器的 **CloudFormation** 模板部分中的模板，并将它以 YAML 文件形式保存到计算机上。此模板描述了集群所需的网络和负载均衡对象。



### 重要

如果要将集群部署到 AWS 政府区域，您必须更新 CloudFormation 模板中的 **InternalApiServerRecord**，以使用 **CNAME** 记录。AWS 政府区不支持 **ALIAS** 类型的记录。

- 启动 CloudFormation 模板，以创建 AWS 资源堆栈，该堆栈提供网络和负载均衡组件：



### 重要

您必须在一行内输入命令。

```
$ aws cloudformation create-stack --stack-name <name> 1
  --template-body file://<template>.yaml 2
  --parameters file://<parameters>.json 3
  --capabilities CAPABILITY_NAMED_IAM 4
```

- <name>** 是 CloudFormation 堆栈的名称，如 **cluster-dns**。如果您删除集群，则需要此堆栈的名称。
- <template>** 是您保存的 CloudFormation 模板 YAML 文件的相对路径和名称。
- <parameters>** 是 CloudFormation 参数 JSON 文件的相对路径和名称。
- 您必须明确声明 **CAPABILITY\_NAMED\_IAM** 功能，因为提供的模板会创建一些 **AWS::IAM::Role** 资源。

### 输出示例

```
arn:aws:cloudformation:us-east-1:269333783861:stack/cluster-dns/cd3e5de0-2fd4-11eb-5cf0-12be5c33a183
```

- 确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <name>
```

在 **StackStatus** 显示 **CREATE\_COMPLETE** 后，输出会显示以下参数的值。您必须将这些参数值提供给您在创建集群时要运行的其他 CloudFormation 模板：

<b>PrivateHostedZoneId</b>	专用 DNS 的托管区 ID。
<b>ExternalApiLoadBalancerName</b>	外部 API 负载均衡器的完整名称。

<b>InternalApiLoadBalancerName</b>	内部 API 负载均衡器的完整名称。
<b>ApiServerDnsName</b>	API 服务器的完整主机名。
<b>RegisterNlbTargetLambda</b>	有助于为这些负载均衡器注册/撤销注册 IP 目标的 Lambda ARN。
<b>ExternalApiTargetGroupArn</b>	外部 API 目标组的 ARN。
<b>InternalApiTargetGroupArn</b>	内部 API 目标组的 ARN。
<b>InternalServiceTargetGroupArn</b>	内部服务目标组群的 ARN。

### 1.10.9.1. 网络和负载均衡器的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署 OpenShift Container Platform 集群所需的网络对象和负载均衡器。

#### 例 1.44. 网络和负载均衡器的 CloudFormation 模板

AWSTemplateFormatVersion: [2010-09-09](#)

Description: Template for OpenShift Cluster Network Elements (Route53 & LBs)

Parameters:

ClusterName:

AllowedPattern: `^[a-zA-Z][a-zA-Z0-9-]{0,26}$`

MaxLength: [27](#)

MinLength: [1](#)

ConstraintDescription: Cluster name must be alphanumeric, start with a letter, and have a maximum of [27](#) characters.

Description: A short, representative cluster name to use for host names and other identifying names.

Type: String

InfrastructureName:

AllowedPattern: `^[a-zA-Z][a-zA-Z0-9-]{0,26}$`

MaxLength: [27](#)

MinLength: [1](#)

ConstraintDescription: Infrastructure name must be alphanumeric, start with a letter, and have a maximum of [27](#) characters.

Description: A short, unique cluster ID used to tag cloud resources and identify items owned or used by the cluster.

Type: String  
HostedZoneId:  
Description: The Route53 public zone ID to register the targets with, such as Z21IXYZABCZ2A4.  
Type: String  
HostedZoneName:  
Description: The Route53 zone to register the targets with, such as example.com. Omit the trailing period.  
Type: String  
Default: "example.com"  
PublicSubnets:  
Description: The internet-facing subnets.  
Type: List<AWS::EC2::Subnet::Id>  
PrivateSubnets:  
Description: The internal subnets.  
Type: List<AWS::EC2::Subnet::Id>  
VpcId:  
Description: The VPC-scoped resources will belong to this VPC.  
Type: AWS::EC2::VPC::Id

Metadata:

AWS::CloudFormation::Interface:

ParameterGroups:

- Label:  
default: "Cluster Information"  
Parameters:
  - ClusterName
  - InfrastructureName
- Label:  
default: "Network Configuration"  
Parameters:
  - VpcId
  - PublicSubnets
  - PrivateSubnets
- Label:  
default: "DNS"  
Parameters:
  - HostedZoneName
  - HostedZoneId

ParameterLabels:

ClusterName:  
default: "Cluster Name"

InfrastructureName:  
default: "Infrastructure Name"

VpcId:  
default: "VPC ID"

PublicSubnets:  
default: "Public Subnets"

PrivateSubnets:  
default: "Private Subnets"

HostedZoneName:  
default: "Public Hosted Zone Name"

HostedZoneId:  
default: "Public Hosted Zone ID"

Resources:

## ExtApiElb:

Type: AWS::ElasticLoadBalancingV2::LoadBalancer

## Properties:

Name: !Join ["-", [!Ref InfrastructureName, "ext"]]

IpAddressType: ipv4

Subnets: !Ref PublicSubnets

Type: network

## IntApiElb:

Type: AWS::ElasticLoadBalancingV2::LoadBalancer

## Properties:

Name: !Join ["-", [!Ref InfrastructureName, "int"]]

Scheme: internal

IpAddressType: ipv4

Subnets: !Ref PrivateSubnets

Type: network

## IntDns:

Type: "AWS::Route53::HostedZone"

## Properties:

## HostedZoneConfig:

Comment: "Managed by CloudFormation"

Name: !Join [".", [!Ref ClusterName, !Ref HostedZoneName]]

## HostedZoneTags:

- Key: Name

Value: !Join ["-", [!Ref InfrastructureName, "int"]]

- Key: !Join [""], ["kubernetes.io/cluster/", !Ref InfrastructureName]]

Value: "owned"

## VPCs:

- VPCId: !Ref Vpclid

VPCRegion: !Ref "AWS::Region"

## ExternalApiServerRecord:

Type: AWS::Route53::RecordSetGroup

## Properties:

Comment: Alias record for the API server

HostedZoneId: !Ref HostedZoneId

## RecordSets:

- Name:

!Join [

":",

["api", !Ref ClusterName, !Join [""], [!Ref HostedZoneName, "."]],

]

Type: A

## AliasTarget:

HostedZoneId: !GetAtt ExtApiElb.CanonicalHostedZoneID

DNSName: !GetAtt ExtApiElb.DNSName

## InternalApiServerRecord:

Type: AWS::Route53::RecordSetGroup

## Properties:

Comment: Alias record for the API server

HostedZoneId: !Ref IntDns

## RecordSets:

- Name:

!Join [

```

      ":",
      ["api", !Ref ClusterName, !Join ["", [!Ref HostedZoneName, "."]]],
    ]
    Type: A
    AliasTarget:
      HostedZoneId: !GetAtt IntApiElb.CanonicalHostedZoneID
      DNSName: !GetAtt IntApiElb.DNSName
  - Name:
    !Join [
      ":",
      ["api-int", !Ref ClusterName, !Join ["", [!Ref HostedZoneName, "."]]],
    ]
    Type: A
    AliasTarget:
      HostedZoneId: !GetAtt IntApiElb.CanonicalHostedZoneID
      DNSName: !GetAtt IntApiElb.DNSName

ExternalApiListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
    - Type: forward
      TargetGroupArn:
        Ref: ExternalApiTargetGroup
    LoadBalancerArn:
      Ref: ExtApiElb
    Port: 6443
    Protocol: TCP

ExternalApiTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 10
    HealthCheckPath: "/readyz"
    HealthCheckPort: 6443
    HealthCheckProtocol: HTTPS
    HealthyThresholdCount: 2
    UnhealthyThresholdCount: 2
    Port: 6443
    Protocol: TCP
    TargetType: ip
    Vpclid:
      Ref: Vpclid
    TargetGroupAttributes:
    - Key: deregistration_delay.timeout_seconds
      Value: 60

InternalApiListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
    - Type: forward
      TargetGroupArn:
        Ref: InternalApiTargetGroup
    LoadBalancerArn:
      Ref: IntApiElb

```



Port: 6443  
Protocol: TCP

InternalApiTargetGroup:

Type: AWS::ElasticLoadBalancingV2::TargetGroup

Properties:

HealthCheckIntervalSeconds: 10

HealthCheckPath: "/readyz"

HealthCheckPort: 6443

HealthCheckProtocol: HTTPS

HealthyThresholdCount: 2

UnhealthyThresholdCount: 2

Port: 6443

Protocol: TCP

TargetType: ip

VpId:

Ref: VpId

TargetGroupAttributes:

- Key: deregistration\_delay.timeout\_seconds  
Value: 60

InternalServiceInternalListener:

Type: AWS::ElasticLoadBalancingV2::Listener

Properties:

DefaultActions:

- Type: forward

TargetGroupArn:

Ref: InternalServiceTargetGroup

LoadBalancerArn:

Ref: IntApiElb

Port: 22623

Protocol: TCP

InternalServiceTargetGroup:

Type: AWS::ElasticLoadBalancingV2::TargetGroup

Properties:

HealthCheckIntervalSeconds: 10

HealthCheckPath: "/healthz"

HealthCheckPort: 22623

HealthCheckProtocol: HTTPS

HealthyThresholdCount: 2

UnhealthyThresholdCount: 2

Port: 22623

Protocol: TCP

TargetType: ip

VpId:

Ref: VpId

TargetGroupAttributes:

- Key: deregistration\_delay.timeout\_seconds  
Value: 60

RegisterTargetLambdRole:

Type: AWS::IAM::Role

Properties:

RoleName: !Join ["-", [!Ref InfrastructureName, "nlb", "lambda", "role"]]

AssumeRolePolicyDocument:

```

Version: "2012-10-17"
Statement:
- Effect: "Allow"
Principal:
Service:
- "lambda.amazonaws.com"
Action:
- "sts:AssumeRole"
Path: "/"
Policies:
- PolicyName: !Join ["-", [!Ref InfrastructureName, "master", "policy"]]
PolicyDocument:
Version: "2012-10-17"
Statement:
- Effect: "Allow"
Action:
[
"elasticloadbalancing:RegisterTargets",
"elasticloadbalancing:DeregisterTargets",
]
Resource: !Ref InternalApiTargetGroup
- Effect: "Allow"
Action:
[
"elasticloadbalancing:RegisterTargets",
"elasticloadbalancing:DeregisterTargets",
]
Resource: !Ref InternalServiceTargetGroup
- Effect: "Allow"
Action:
[
"elasticloadbalancing:RegisterTargets",
"elasticloadbalancing:DeregisterTargets",
]
Resource: !Ref ExternalApiTargetGroup

```

#### RegisterNlbTargets:

Type: "AWS::Lambda::Function"

#### Properties:

Handler: "index.handler"

#### Role:

Fn::GetAtt:

- "RegisterTargetLambdalamRole"
- "Arn"

#### Code:

ZipFile: |

```
import json
```

```
import boto3
```

```
import cfntools
```

```
def handler(event, context):
```

```
    elb = boto3.client('elbv2')
```

```
    if event['RequestType'] == 'Delete':
```

```
        elb.deregister_targets(TargetGroupArn=event['ResourceProperties']
```

```
['TargetArn'], Targets=[{'Id': event['ResourceProperties']['TargetId']})
```

```
    elif event['RequestType'] == 'Create':
```

```
        elb.register_targets(TargetGroupArn=event['ResourceProperties']['TargetArn'], Targets=
```

```

[{'Id': event['ResourceProperties']['TargetIp']}]
    responseData = {}
    cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
event['ResourceProperties']['TargetArn']+event['ResourceProperties']['TargetIp'])
    Runtime: "python3.7"
    Timeout: 120

```

RegisterSubnetTagsLambdalaRole:

Type: AWS::IAM::Role

Properties:

RoleName: !Join ["-", [!Ref InfrastructureName, "subnet-tags-lambda-role"]]

AssumeRolePolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: "Allow"

Principal:

Service:

- "lambda.amazonaws.com"

Action:

- "sts:AssumeRole"

Path: "/"

Policies:

- PolicyName: !Join ["-", [!Ref InfrastructureName, "subnet-tagging-policy"]]

PolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: "Allow"

Action:

```

[
  "ec2:DeleteTags",
  "ec2:CreateTags"
]

```

Resource: "arn:aws:ec2:\*:\*:subnet/\*"

- Effect: "Allow"

Action:

```

[
  "ec2:DescribeSubnets",
  "ec2:DescribeTags"
]

```

Resource: ""

RegisterSubnetTags:

Type: "AWS::Lambda::Function"

Properties:

Handler: "index.handler"

Role:

Fn::GetAtt:

- "RegisterSubnetTagsLambdalaRole"

- "Arn"

Code:

ZipFile: |

```
import json
```

```
import boto3
```

```
import cfnresponse
```

```
def handler(event, context):
```

```
    ec2_client = boto3.client('ec2')
```

```

    if event['RequestType'] == 'Delete':
        for subnet_id in event['ResourceProperties']['Subnets']:
            ec2_client.delete_tags(Resources=[subnet_id], Tags=[{'Key': 'kubernetes.io/cluster/' +
event['ResourceProperties']['InfrastructureName']}]);
        elif event['RequestType'] == 'Create':
            for subnet_id in event['ResourceProperties']['Subnets']:
                ec2_client.create_tags(Resources=[subnet_id], Tags=[{'Key': 'kubernetes.io/cluster/' +
event['ResourceProperties']['InfrastructureName'], 'Value': 'shared'}]);
            responseData = {}
            cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
event['ResourceProperties']['InfrastructureName']+event['ResourceProperties']['Subnets'][0])
    Runtime: "python3.7"
    Timeout: 120

```

**RegisterPublicSubnetTags:**

Type: Custom::SubnetRegister

## Properties:

ServiceToken: !GetAtt RegisterSubnetTags.Arn

InfrastructureName: !Ref InfrastructureName

Subnets: !Ref PublicSubnets

**RegisterPrivateSubnetTags:**

Type: Custom::SubnetRegister

## Properties:

ServiceToken: !GetAtt RegisterSubnetTags.Arn

InfrastructureName: !Ref InfrastructureName

Subnets: !Ref PrivateSubnets

**Outputs:****PrivateHostedZoneId:**

Description: Hosted zone ID for the private DNS, which is required for private records.

Value: !Ref IntDns

**ExternalApiLoadBalancerName:**

Description: Full name of the external API load balancer.

Value: !GetAtt ExtApiElb.LoadBalancerFullName

**InternalApiLoadBalancerName:**

Description: Full name of the internal API load balancer.

Value: !GetAtt IntApiElb.LoadBalancerFullName

**ApiServerDnsName:**

Description: Full hostname of the API server, which is required for the Ignition config files.

Value: !Join [".", ["api-int", !Ref ClusterName, !Ref HostedZoneName]]

**RegisterNlbPTargetsLambda:**

Description: Lambda ARN useful to help register or deregister IP targets for these load balancers.

Value: !GetAtt RegisterNlbPTargets.Arn

**ExternalApiTargetGroupArn:**

Description: ARN of the external API target group.

Value: !Ref ExternalApiTargetGroup

**InternalApiTargetGroupArn:**

Description: ARN of the internal API target group.

Value: !Ref InternalApiTargetGroup

**InternalServiceTargetGroupArn:**

Description: ARN of the internal service target group.

Value: !Ref InternalServiceTargetGroup



## 重要

如果要部署集群到 AWS 政府区域，您必须更新 `InternalApiServerRecord` 以使用 `CNAME` 记录。AWS 政府区不支持 `ALIAS` 类型的记录。例如：

```
Type: CNAME
TTL: 10
ResourceRecords:
- !GetAtt IntApiElb.DNSName
```

## 其他资源

- 有关列出公共托管区的更多信息，请参阅 AWS 文档中的[列出公共托管区](#)。

### 1.10.10. 在 AWS 中创建安全组和角色

您必须在 Amazon Web Services (AWS) 中创建安全组和角色，供您的 OpenShift Container Platform 集群使用。

您可以使用提供的 CloudFormation 模板和自定义参数文件来创建 AWS 资源堆栈。堆栈代表 OpenShift Container Platform 集群所需的安全组和角色。



## 注意

如果不使用提供的 CloudFormation 模板来创建 AWS 基础架构，您必须检查提供的信息并手动创建基础架构。如果集群没有正确初始化，您可能需要联系红帽支持并提供您的安装日志。

## 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 `aws configure`，将 AWS 密钥和区域添加到本地 AWS 配置集中。
- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。

## 流程

1. 创建一个 JSON 文件，其包含模板所需的参数值：

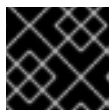
```
[
  {
    "ParameterKey": "InfrastructureName", 1
    "ParameterValue": "mycluster-<random_string>" 2
  },
  {
    "ParameterKey": "VpcCidr", 3
    "ParameterValue": "10.0.0.0/16" 4
  },
  {
    "ParameterKey": "PrivateSubnets", 5
    "ParameterValue": "subnet-<random_string>" 6
  }
]
```

```

    },
    {
      "ParameterKey": "VpcId", ⑦
      "ParameterValue": "vpc-<random_string>" ⑧
    }
  ]

```

- ① 您的 Ignition 配置文件中为集群编码的集群基础架构名称。
  - ② 指定从 Ignition 配置文件元数据中提取的基础架构名称，其格式为 **<cluster-name>-<random-string>**。
  - ③ VPC 的 CIDR 块。
  - ④ 指定以 **x.x.x.x/16-24** 格式定义的用于 VPC 的 CIDR 地址块。
  - ⑤ 为 VPC 创建的专用子网。
  - ⑥ 指定 VPC 的 CloudFormation 模板输出的 **PrivateSubnetIds** 值。
  - ⑦ 为集群创建的 VPC。
  - ⑧ 指定 VPC 的 CloudFormation 模板输出的 **VpcId** 值。
2. 复制本主题的安全对象的 CloudFormation 模板部分中的模板，并将它以 YAML 文件形式保存到计算机上。此模板描述了集群所需的安全组和角色。
  3. 启动 CloudFormation 模板，以创建代表安全组和角色的 AWS 资源堆栈：



### 重要

您必须在一行内输入命令。

```

$ aws cloudformation create-stack --stack-name <name> ①
  --template-body file://<template>.yaml ②
  --parameters file://<parameters>.json ③
  --capabilities CAPABILITY_NAMED_IAM ④

```

- ① **<name>** 是 CloudFormation 堆栈的名称，如 **cluster-sec**。如果您删除集群，则需要此堆栈的名称。
- ② **<template>** 是您保存的 CloudFormation 模板 YAML 文件的相对路径和名称。
- ③ **<parameters>** 是 CloudFormation 参数 JSON 文件的相对路径和名称。
- ④ 您必须明确声明 **CAPABILITY\_NAMED\_IAM** 功能，因为提供的模板会创建一些 **AWS::IAM::Role** 和 **AWS::IAM::InstanceProfile** 资源。

### 输出示例

```

arn:aws:cloudformation:us-east-1:269333783861:stack/cluster-sec/03bd4210-2ed7-11eb-6d7a-13fc0b61e9db

```

## 4. 确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <name>
```

在 **StackStatus** 显示 **CREATE\_COMPLETE** 后，输出会显示以下参数的值。您必须将这些参数值提供给您在创建集群时要运行的其他 CloudFormation 模板：

<b>MasterSecurityGroupID</b>	Master 安全组 ID
<b>WorkerSecurityGroupID</b>	worker 安全组 ID
<b>MasterInstanceProfile</b>	Master IAM 实例配置集
<b>WorkerInstanceProfile</b>	worker IAM 实例配置集

## 1.10.10.1. 安全对象的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署 OpenShift Container Platform 集群所需的安全对象。

## 例 1.45. 安全对象的 CloudFormation 模板

AWSTemplateFormatVersion: 2010-09-09

Description: Template for OpenShift Cluster Security Elements (Security Groups & IAM)

Parameters:

InfrastructureName:

AllowedPattern: ^([a-zA-Z][a-zA-Z0-9\-\\_]{0,26})\$

MaxLength: 27

MinLength: 1

ConstraintDescription: Infrastructure name must be alphanumeric, start with a letter, and have a maximum of 27 characters.

Description: A short, unique cluster ID used to tag cloud resources and identify items owned or used by the cluster.

Type: String

VpcCidr:

AllowedPattern: ^(((0-9)|1-9|0-9|10-9){2}|2[0-4][0-9]|25[0-5])\.\.){3}((0-9)|1-9|0-9|10-9){2}|2[0-4][0-9]|25[0-5])(\.(1[6-9]|2[0-4]))\$

ConstraintDescription: CIDR block parameter must be in the form x.x.x.x/16-24.

Default: 10.0.0.0/16

Description: CIDR block for VPC.

Type: String

VpcId:

Description: The VPC-scoped resources will belong to this VPC.

Type: AWS::EC2::VPC::Id

PrivateSubnets:

Description: The internal subnets.  
Type: List<AWS::EC2::Subnet::Id>

**Metadata:**

AWS::CloudFormation::Interface:

ParameterGroups:

- Label:

default: "Cluster Information"

Parameters:

- InfrastructureName

- Label:

default: "Network Configuration"

Parameters:

- VpcId

- VpcCidr

- PrivateSubnets

ParameterLabels:

InfrastructureName:

default: "Infrastructure Name"

VpcId:

default: "VPC ID"

VpcCidr:

default: "VPC CIDR"

PrivateSubnets:

default: "Private Subnets"

**Resources:**

MasterSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Cluster Master Security Group

SecurityGroupIngress:

- IpProtocol: icmp

FromPort: 0

ToPort: 0

CidrIp: !Ref VpcCidr

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: !Ref VpcCidr

- IpProtocol: tcp

ToPort: 6443

FromPort: 6443

CidrIp: !Ref VpcCidr

- IpProtocol: tcp

FromPort: 22623

ToPort: 22623

CidrIp: !Ref VpcCidr

VpcId: !Ref VpcId

WorkerSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Cluster Worker Security Group

SecurityGroupIngress:

- IpProtocol: icmp



FromPort: 0  
ToPort: 0  
CidrIp: !Ref VpcCidr  
- IpProtocol: tcp  
FromPort: 22  
ToPort: 22  
CidrIp: !Ref VpcCidr  
Vpclid: !Ref Vpclid

MasterIngressEtcd:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: etcd  
FromPort: 2379  
ToPort: 2380  
IpProtocol: tcp

MasterIngressVxlan:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Vxlan packets  
FromPort: 4789  
ToPort: 4789  
IpProtocol: udp

MasterIngressWorkerVxlan:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Vxlan packets  
FromPort: 4789  
ToPort: 4789  
IpProtocol: udp

MasterIngressGeneve:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Geneve packets  
FromPort: 6081  
ToPort: 6081  
IpProtocol: udp

MasterIngressWorkerGeneve:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Geneve packets  
FromPort: 6081

ToPort: 6081  
IpProtocol: udp

MasterIngressInternal:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Internal cluster communication  
FromPort: 9000  
ToPort: 9999  
IpProtocol: tcp

MasterIngressWorkerInternal:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Internal cluster communication  
FromPort: 9000  
ToPort: 9999  
IpProtocol: tcp

MasterIngressInternalUDP:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Internal cluster communication  
FromPort: 9000  
ToPort: 9999  
IpProtocol: udp

MasterIngressWorkerInternalUDP:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Internal cluster communication  
FromPort: 9000  
ToPort: 9999  
IpProtocol: udp

MasterIngressKube:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Kubernetes kubelet, scheduler and controller manager  
FromPort: 10250  
ToPort: 10259  
IpProtocol: tcp

MasterIngressWorkerKube:

Type: AWS::EC2::SecurityGroupIngress  
Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Kubernetes kubelet, scheduler and controller manager  
FromPort: 10250  
ToPort: 10259  
IpProtocol: tcp

MasterIngressIngressServices:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Kubernetes ingress services  
FromPort: 30000  
ToPort: 32767  
IpProtocol: tcp

MasterIngressWorkerIngressServices:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Kubernetes ingress services  
FromPort: 30000  
ToPort: 32767  
IpProtocol: tcp

MasterIngressIngressServicesUDP:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Kubernetes ingress services  
FromPort: 30000  
ToPort: 32767  
IpProtocol: udp

MasterIngressWorkerIngressServicesUDP:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Kubernetes ingress services  
FromPort: 30000  
ToPort: 32767  
IpProtocol: udp

WorkerIngressVxlan:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt WorkerSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Vxlan packets  
FromPort: 4789  
ToPort: 4789  
IpProtocol: udp

**WorkerIngressMasterVxlan:**

Type: AWS::EC2::SecurityGroupIngress

## Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: Vxlan packets

FromPort: 4789

ToPort: 4789

IpProtocol: udp

**WorkerIngressGeneve:**

Type: AWS::EC2::SecurityGroupIngress

## Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Geneve packets

FromPort: 6081

ToPort: 6081

IpProtocol: udp

**WorkerIngressMasterGeneve:**

Type: AWS::EC2::SecurityGroupIngress

## Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: Geneve packets

FromPort: 6081

ToPort: 6081

IpProtocol: udp

**WorkerIngressInternal:**

Type: AWS::EC2::SecurityGroupIngress

## Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Internal cluster communication

FromPort: 9000

ToPort: 9999

IpProtocol: tcp

**WorkerIngressMasterInternal:**

Type: AWS::EC2::SecurityGroupIngress

## Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: Internal cluster communication

FromPort: 9000

ToPort: 9999

IpProtocol: tcp

**WorkerIngressInternalUDP:**

Type: AWS::EC2::SecurityGroupIngress

## Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Internal cluster communication  
FromPort: 9000  
ToPort: 9999  
IpProtocol: udp

WorkerIngressMasterInternalUDP:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt WorkerSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Internal cluster communication  
FromPort: 9000  
ToPort: 9999  
IpProtocol: udp

WorkerIngressKube:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt WorkerSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Kubernetes secure kubelet port  
FromPort: 10250  
ToPort: 10250  
IpProtocol: tcp

WorkerIngressWorkerKube:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt WorkerSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Internal Kubernetes communication  
FromPort: 10250  
ToPort: 10250  
IpProtocol: tcp

WorkerIngressIngressServices:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt WorkerSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
Description: Kubernetes ingress services  
FromPort: 30000  
ToPort: 32767  
IpProtocol: tcp

WorkerIngressMasterIngressServices:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt WorkerSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: Kubernetes ingress services  
FromPort: 30000  
ToPort: 32767  
IpProtocol: tcp

WorkerIngressIngressServicesUDP:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: Kubernetes ingress services

FromPort: 30000

ToPort: 32767

IpProtocol: udp

WorkerIngressMasterIngressServicesUDP:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: Kubernetes ingress services

FromPort: 30000

ToPort: 32767

IpProtocol: udp

MasterIamRole:

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: "Allow"

Principal:

Service:

- "ec2.amazonaws.com"

Action:

- "sts:AssumeRole"

Policies:

- PolicyName: !Join ["-", [!Ref InfrastructureName, "master", "policy"]]

PolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: "Allow"

Action:

- "ec2:AttachVolume"

- "ec2:AuthorizeSecurityGroupIngress"

- "ec2:CreateSecurityGroup"

- "ec2:CreateTags"

- "ec2:CreateVolume"

- "ec2>DeleteSecurityGroup"

- "ec2>DeleteVolume"

- "ec2:Describe\*"

- "ec2:DetachVolume"

- "ec2:ModifyInstanceAttribute"

- "ec2:ModifyVolume"

- "ec2:RevokeSecurityGroupIngress"

- "elasticloadbalancing:AddTags"

- "elasticloadbalancing:AttachLoadBalancerToSubnets"

- "elasticloadbalancing:ApplySecurityGroupsToLoadBalancer"

- "elasticloadbalancing:CreateListener"

- "elasticloadbalancing:CreateLoadBalancer"

- "elasticloadbalancing:CreateLoadBalancerPolicy"

- "elasticloadbalancing:CreateLoadBalancerListeners"
- "elasticloadbalancing:CreateTargetGroup"
- "elasticloadbalancing:ConfigureHealthCheck"
- "elasticloadbalancing>DeleteListener"
- "elasticloadbalancing>DeleteLoadBalancer"
- "elasticloadbalancing>DeleteLoadBalancerListeners"
- "elasticloadbalancing>DeleteTargetGroup"
- "elasticloadbalancing:DeregisterInstancesFromLoadBalancer"
- "elasticloadbalancing:DeregisterTargets"
- "elasticloadbalancing:Describe\*"
- "elasticloadbalancing:DetachLoadBalancerFromSubnets"
- "elasticloadbalancing:ModifyListener"
- "elasticloadbalancing:ModifyLoadBalancerAttributes"
- "elasticloadbalancing:ModifyTargetGroup"
- "elasticloadbalancing:ModifyTargetGroupAttributes"
- "elasticloadbalancing:RegisterInstancesWithLoadBalancer"
- "elasticloadbalancing:RegisterTargets"
- "elasticloadbalancing:SetLoadBalancerPoliciesForBackendServer"
- "elasticloadbalancing:SetLoadBalancerPoliciesOfListener"
- "kms:DescribeKey"

Resource: "\*"

#### MasterInstanceProfile:

Type: "AWS::IAM::InstanceProfile"

Properties:

Roles:

- Ref: "MasterIamRole"

#### WorkerIamRole:

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: "Allow"

Principal:

Service:

- "ec2.amazonaws.com"

Action:

- "sts:AssumeRole"

Policies:

- PolicyName: !Join ["-", [!Ref InfrastructureName, "worker", "policy"]]

PolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: "Allow"

Action:

- "ec2:DescribeInstances"

- "ec2:DescribeRegions"

Resource: "\*"

#### WorkerInstanceProfile:

Type: "AWS::IAM::InstanceProfile"

Properties:

Roles:

- Ref: "WorkerIamRole"

**Outputs:****MasterSecurityGroupId:**

Description: Master Security Group ID

Value: !GetAtt MasterSecurityGroup.GroupId

**WorkerSecurityGroupId:**

Description: Worker Security Group ID

Value: !GetAtt WorkerSecurityGroup.GroupId

**MasterInstanceProfile:**

Description: Master IAM Instance Profile

Value: !Ref MasterInstanceProfile

**WorkerInstanceProfile:**

Description: Worker IAM Instance Profile

Value: !Ref WorkerInstanceProfile

### 1.10.11. AWS 基础架构的 RHCOS AMI

红帽为您提供为 OpenShift Container Platform 节点指定的各种 Amazon Web Services (AWS) 区域提供了有效的 Red Hat Enterprise Linux CoreOS (RHCOS) AMI。

**注意**

您还可以导入您自己的 AMI，来安装到没有发布 RHCOS AMI 的区域。

表 1.29. RHCOS AMI

AWS 区	AWS AMI
af-south-1	ami-09921c9c1c36e695c
ap-east-1	ami-01ee8446e9af6b197
ap-northeast-1	ami-04e5b5722a55846ea
ap-northeast-2	ami-0fdc25c8a0273a742
ap-south-1	ami-09e3deb397cc526a8
ap-southeast-1	ami-0630e03f75e02eec4
ap-southeast-2	ami-069450613262ba03c
ca-central-1	ami-012518cdbd3057dfd
eu-central-1	ami-0bd7175ff5b1aef0c



AWS 区	AWS AMI
eu-north-1	ami-06c9ec42d0a839ad2
eu-south-1	ami-0614d7440a0363d71
eu-west-1	ami-01b89df58b5d4d5fa
eu-west-2	ami-06f6e31ddd554f89d
eu-west-3	ami-0dc82e2517ded15a1
me-south-1	ami-07d181e3aa0f76067
sa-east-1	ami-0cd44e6dd20e6c7fa
us-east-1	ami-04a16d506e5b0e246
us-east-2	ami-0a1f868ad58ea59a7
us-west-1	ami-0a65d76e3a6f6622f
us-west-2	ami-0dd9008abadc519f1

### 1.10.12. 在 AWS 中创建 bootstrap 节点

您必须在 Amazon Web Services (AWS) 中创建 bootstrap 节点，以便在 OpenShift Container Platform 集群初始化过程中使用。

您可以使用提供的 CloudFormation 模板和自定义参数文件来创建 AWS 资源堆栈。堆栈代表 OpenShift Container Platform 安装所需的 bootstrap 节点。



#### 注意

如果不使用提供的 CloudFormation 模板来创建 bootstrap 节点，您必须检查提供的信息并手动创建基础架构。如果集群没有正确初始化，您可能需要联系红帽支持并提供您的安装日志。

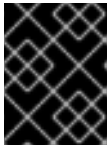
#### 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。
- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。
- 您在 AWS 中创建并配置了 DNS、负载均衡器和监听程序。

- 您在 AWS 中创建了集群所需的安全组和角色。

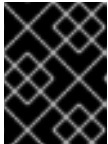
## 流程

- 提供一个位置，以便向集群提供 **bootstrap.ign** Ignition 配置文件。此文件位于您的安装目录中。达成此目标的一种方式是在集群区域中创建一个 S3 存储桶，并将 Ignition 配置文件上传到其中。



### 重要

提供的 CloudFormation 模板假定集群的 Ignition 配置文件由 S3 存储桶提供。如果选择从其他位置提供文件，您必须修改模板。



### 重要

如果您部署到具有与 AWS SDK 不同的端点，或者您提供自己的自定义端点的区域，则必须为 S3 存储桶使用预签名 URL 而不是 **s3://** 模式。



### 注意

bootstrap Ignition 配置文件包含 secret，如 X.509 密钥。以下步骤为 S3 存储桶提供基本安全性。若要提供额外的安全性，您可以启用 S3 存储桶策略，仅允许某些用户（如 OpenShift IAM 用户）访问存储桶中包含的对象。您可以完全避开 S3，并从 bootstrap 可访问的任意地址提供 bootstrap Ignition 配置文件。

- 创建存储桶：

```
$ aws s3 mb s3://<cluster-name>-infra 1
```

- 1** **<cluster-name>-infra** 是存储桶名称。在创建 **install-config.yaml** 文件时，将 **<cluster-name>** 替换为为集群指定的名称。

- 将 **bootstrap.ign** Ignition 配置文件上传到存储桶：

```
$ aws s3 cp <installation_directory>/bootstrap.ign s3://<cluster-name>-infra/bootstrap.ign 1
```

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

- 验证文件已经上传：

```
$ aws s3 ls s3://<cluster-name>-infra/
```

### 输出示例

```
2019-04-03 16:15:16 314878 bootstrap.ign
```

- 创建一个 JSON 文件，其包含模板所需的参数值：

```
[
  {
    "ParameterKey": "InfrastructureName", 1
```

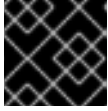
```

    "ParameterValue": "mycluster-<random_string>" 2
  },
  {
    "ParameterKey": "RhcoshAmi", 3
    "ParameterValue": "ami-<random_string>" 4
  },
  {
    "ParameterKey": "AllowedBootstrapSshCidr", 5
    "ParameterValue": "0.0.0.0/0" 6
  },
  {
    "ParameterKey": "PublicSubnet", 7
    "ParameterValue": "subnet-<random_string>" 8
  },
  {
    "ParameterKey": "MasterSecurityGroup", 9
    "ParameterValue": "sg-<random_string>" 10
  },
  {
    "ParameterKey": "VpcId", 11
    "ParameterValue": "vpc-<random_string>" 12
  },
  {
    "ParameterKey": "BootstrapIgnitionLocation", 13
    "ParameterValue": "s3://<bucket_name>/bootstrap.ign" 14
  },
  {
    "ParameterKey": "AutoRegisterELB", 15
    "ParameterValue": "yes" 16
  },
  {
    "ParameterKey": "RegisterNlbTargetsLambdaArn", 17
    "ParameterValue": "arn:aws:lambda:<region>:<account_number>:function:
<dns_stack_name>-RegisterNlbTargets-<random_string>" 18
  },
  {
    "ParameterKey": "ExternalApiTargetGroupArn", 19
    "ParameterValue": "arn:aws:elasticloadbalancing:<region>:
<account_number>:targetgroup/<dns_stack_name>-Exter-<random_string>" 20
  },
  {
    "ParameterKey": "InternalApiTargetGroupArn", 21
    "ParameterValue": "arn:aws:elasticloadbalancing:<region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<random_string>" 22
  },
  {
    "ParameterKey": "InternalServiceTargetGroupArn", 23
    "ParameterValue": "arn:aws:elasticloadbalancing:<region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<random_string>" 24
  }
]

```

1 您的 Ignition 配置文件中为集群编码的集群基础架构名称。

- 2 指定从 Ignition 配置文件元数据中提取的基础架构名称，其格式为 **<cluster-name>-<random-string>**。
  - 3 用于 bootstrap 节点的当前 Red Hat Enterprise Linux CoreOS (RHCOS) AMI。
  - 4 指定有效的 **AWS::EC2::Image::Id** 值。
  - 5 允许通过 SSH 访问 bootstrap 节点的 CIDR 块。
  - 6 以 **x.x.x.x/16-24** 格式指定 CIDR 块。
  - 7 与 VPC 关联的公共子网，将 bootstrap 节点启动到其中。
  - 8 指定 VPC 的 CloudFormation 模板输出的 **PublicSubnetIds** 值。
  - 9 master 安全组 ID（用于注册临时规则）
  - 10 指定安全组和角色的 CloudFormation 模板输出的 **MasterSecurityGroupId** 值。
  - 11 创建的资源将从属于的 VPC。
  - 12 指定 VPC 的 CloudFormation 模板输出的 **VpId** 值。
  - 13 从中获取 bootstrap Ignition 配置文件的位置。
  - 14 指定 S3 存储桶和文件名，格式为 **s3://<bucket\_name>/bootstrap.ign**。
  - 15 是否要注册网络负载均衡器 (NLB)。
  - 16 指定 **yes** 或 **no**。如果指定 **yes**，您必须提供一个 Lambda Amazon Resource Name (ARN) 值。
  - 17 NLB IP 目标注册 lambda 组的 ARN。
  - 18 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **RegisterNlbIpTargetsLambda** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
  - 19 外部 API 负载均衡器目标组的 ARN。
  - 20 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **ExternalApiTargetGroupArn** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
  - 21 内部 API 负载均衡器目标组群的 ARN。
  - 22 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **InternalApiTargetGroupArn** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
  - 23 内部服务负载均衡器目标组群的 ARN。
  - 24 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **InternalServiceTargetGroupArn** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
3. 复制本主题的 **Bootstrap** 机器的 **CloudFormation** 模板部分中的模板，并将它以 YAML 文件形式保存到计算机上。此模板描述了集群所需的 bootstrap 机器。
  4. 启动 CloudFormation 模板，以创建代表 bootstrap 节点的 AWS 资源堆栈：



## 重要

您必须在一行内输入命令。

```
$ aws cloudformation create-stack --stack-name <name> 1
  --template-body file://<template>.yaml 2
  --parameters file://<parameters>.json 3
  --capabilities CAPABILITY_NAMED_IAM 4
```

- 1 **<name>** 是 CloudFormation 堆栈的名称，如 **cluster-bootstrap**。如果您删除集群，则需要此堆栈的名称。
- 2 **<template>** 是您保存的 CloudFormation 模板 YAML 文件的相对路径和名称。
- 3 **<parameters>** 是 CloudFormation 参数 JSON 文件的相对路径和名称。
- 4 您必须明确声明 **CAPABILITY\_NAMED\_IAM** 功能，因为提供的模板会创建一些 **AWS::IAM::Role** 和 **AWS::IAM::InstanceProfile** 资源。

## 输出示例

```
arn:aws:cloudformation:us-east-1:269333783861:stack/cluster-bootstrap/12944486-2add-11eb-9dee-12dace8e3a83
```

5. 确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <name>
```

在 **StackStatus** 显示 **CREATE\_COMPLETE** 后，输出会显示以下参数的值。您必须将这些参数值提供给您在创建集群时要运行的其他 CloudFormation 模板：

<b>Bootstrap InstanceId</b>	bootstrap 实例 ID。
<b>Bootstrap PublicIp</b>	bootstrap 节点公共 IP 地址。
<b>Bootstrap PrivateIp</b>	bootstrap 节点专用 IP 地址。

### 1.10.12.1. bootstrap 机器的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署 OpenShift Container Platform 集群所需的 bootstrap 机器。

#### 例 1.46. bootstrap 机器的 CloudFormation 模板

```
AWSTemplateFormatVersion: 2010-09-09
Description: Template for OpenShift Cluster Bootstrap (EC2 Instance, Security Groups and IAM)
```

## Parameters:

## InfrastructureName:

AllowedPattern: `^[a-zA-Z][a-zA-Z0-9\-\]{0,26}$`

MaxLength: `27`

MinLength: `1`

ConstraintDescription: Infrastructure name must be alphanumeric, start with a letter, and have a maximum of `27` characters.

Description: A short, unique cluster ID used to tag cloud resources and identify items owned or used by the cluster.

Type: String

## RhcOsAmi:

Description: Current Red Hat Enterprise Linux CoreOS AMI to use for bootstrap.

Type: `AWS::EC2::Image::Id`

## AllowedBootstrapSshCidr:

AllowedPattern: `^((([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.)\.)\{3\}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\)(\{([0-9]|1[0-9]|2[0-9]|3[0-2])\})$`

ConstraintDescription: CIDR block parameter must be in the form `x.x.x.x/0-32`.

Default: `0.0.0.0/0`

Description: CIDR block to allow SSH access to the bootstrap node.

Type: String

## PublicSubnet:

Description: The public subnet to launch the bootstrap node into.

Type: `AWS::EC2::Subnet::Id`

## MasterSecurityGroupId:

Description: The master security group ID for registering temporary rules.

Type: `AWS::EC2::SecurityGroup::Id`

## VpcId:

Description: The VPC-scoped resources will belong to this VPC.

Type: `AWS::EC2::VPC::Id`

## BootstrapIgnitionLocation:

Default: `s3://my-s3-bucket/bootstrap.ign`

Description: Ignition config file location.

Type: String

## AutoRegisterELB:

Default: `"yes"`

AllowedValues:

- `"yes"`

- `"no"`

Description: Do you want to invoke NLB registration, which requires a Lambda ARN parameter?

Type: String

## RegisterNlbIpTargetsLambdaArn:

Description: ARN for NLB IP target registration lambda.

Type: String

## ExternalApiTargetGroupArn:

Description: ARN for external API load balancer target group.

Type: String

## InternalApiTargetGroupArn:

Description: ARN for internal API load balancer target group.

Type: String

## InternalServiceTargetGroupArn:

Description: ARN for internal service load balancer target group.

Type: String

## Metadata:

`AWS::CloudFormation::Interface:`

ParameterGroups:

```

- Label:
  default: "Cluster Information"
Parameters:
- InfrastructureName
- Label:
  default: "Host Information"
Parameters:
- RhcosAmi
- BootstrapIgnitionLocation
- MasterSecurityGroupId
- Label:
  default: "Network Configuration"
Parameters:
- VpcId
- AllowedBootstrapSshCidr
- PublicSubnet
- Label:
  default: "Load Balancer Automation"
Parameters:
- AutoRegisterELB
- RegisterNlbTargetsLambdaArn
- ExternalApiTargetGroupArn
- InternalApiTargetGroupArn
- InternalServiceTargetGroupArn
ParameterLabels:
InfrastructureName:
  default: "Infrastructure Name"
VpcId:
  default: "VPC ID"
AllowedBootstrapSshCidr:
  default: "Allowed SSH Source"
PublicSubnet:
  default: "Public Subnet"
RhcosAmi:
  default: "Red Hat Enterprise Linux CoreOS AMI ID"
BootstrapIgnitionLocation:
  default: "Bootstrap Ignition Source"
MasterSecurityGroupId:
  default: "Master Security Group ID"
AutoRegisterELB:
  default: "Use Provided ELB Automation"

```

Conditions:

```
DoRegistration: !Equals ["yes", !Ref AutoRegisterELB]
```

Resources:

```

BootstrapIamRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: "Allow"
          Principal:
            Service:
              - "ec2.amazonaws.com"

```

```
Action:
- "sts:AssumeRole"
Path: "/"
Policies:
- PolicyName: !Join ["-", [!Ref InfrastructureName, "bootstrap", "policy"]]
  PolicyDocument:
    Version: "2012-10-17"
    Statement:
      - Effect: "Allow"
        Action: "ec2:Describe*"
        Resource: "*"
      - Effect: "Allow"
        Action: "ec2:AttachVolume"
        Resource: "*"
      - Effect: "Allow"
        Action: "ec2:DetachVolume"
        Resource: "*"
      - Effect: "Allow"
        Action: "s3:GetObject"
        Resource: "*"

BootstrapInstanceProfile:
  Type: "AWS::IAM::InstanceProfile"
  Properties:
    Path: "/"
    Roles:
      - Ref: "BootstrapIamRole"

BootstrapSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Cluster Bootstrap Security Group
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref AllowedBootstrapSshCidr
      - IpProtocol: tcp
        ToPort: 19531
        FromPort: 19531
        CidrIp: 0.0.0.0/0
    VpCid: !Ref VpCid

BootstrapInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !Ref RHCOSAmi
    IamInstanceProfile: !Ref BootstrapInstanceProfile
    InstanceType: "i3.large"
    NetworkInterfaces:
      - AssociatePublicIpAddress: "true"
        DeviceIndex: "0"
        GroupSet:
          - !Ref "BootstrapSecurityGroup"
          - !Ref "MasterSecurityGroup"
        SubnetId: !Ref "PublicSubnet"
```



```

UserData:
  Fn::Base64: !Sub
    - '{"ignition":{"config":{"replace":{"source":"${S3Loc}"}},"version":"3.1.0"}}'
    - {
      S3Loc: !Ref BootstrapIgnitionLocation
    }

```

```

RegisterBootstrapApiTarget:
  Condition: DoRegistration
  Type: Custom::NLBRegister
  Properties:
    ServiceToken: !Ref RegisterNlbTargetsLambdaArn
    TargetArn: !Ref ExternalApiTargetGroupArn
    TargetIp: !GetAtt BootstrapInstance.PrivateIp

```

```

RegisterBootstrapInternalApiTarget:
  Condition: DoRegistration
  Type: Custom::NLBRegister
  Properties:
    ServiceToken: !Ref RegisterNlbTargetsLambdaArn
    TargetArn: !Ref InternalApiTargetGroupArn
    TargetIp: !GetAtt BootstrapInstance.PrivateIp

```

```

RegisterBootstrapInternalServiceTarget:
  Condition: DoRegistration
  Type: Custom::NLBRegister
  Properties:
    ServiceToken: !Ref RegisterNlbTargetsLambdaArn
    TargetArn: !Ref InternalServiceTargetGroupArn
    TargetIp: !GetAtt BootstrapInstance.PrivateIp

```

```

Outputs:
  BootstrapInstanceId:
    Description: Bootstrap Instance ID.
    Value: !Ref BootstrapInstance

```

```

BootstrapPublicIp:
  Description: The bootstrap node public IP address.
  Value: !GetAtt BootstrapInstance.PublicIp

```

```

BootstrapPrivateIp:
  Description: The bootstrap node private IP address.
  Value: !GetAtt BootstrapInstance.PrivateIp

```

## 其他资源

- 如需有关 AWS 区的 Red Hat Enterprise Linux CoreOS (RHCOS) AMI 的详细信息，请参阅 [AWS 基础架构的 RHCOS AMI](#)。

### 1.10.13. 在 AWS 中创建 control plane 机器

您必须在集群要使用的 Amazon Web Services (AWS) 中创建 control plane 机器。

您可以使用提供的 CloudFormation 模板和自定义参数文件，创建代表 control plane 节点的 AWS 资源堆栈。



### 重要

CloudFormation 模板会创建一个堆栈，它代表三个 control plane 节点。



### 注意

如果不使用提供的 CloudFormation 模板来创建 control plane 节点，您必须检查提供的信息并手动创建基础架构。如果集群没有正确初始化，您可能需要联系红帽支持并提供您的安装日志。

### 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。
- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。
- 您在 AWS 中创建并配置了 DNS、负载均衡器和监听程序。
- 您在 AWS 中创建了集群所需的安全组和角色。
- 已创建 bootstrap 机器。

### 流程

1. 创建一个 JSON 文件，其包含模板所需的参数值：

```
[
  {
    "ParameterKey": "InfrastructureName", 1
    "ParameterValue": "mycluster-<random_string>" 2
  },
  {
    "ParameterKey": "RhcossAmi", 3
    "ParameterValue": "ami-<random_string>" 4
  },
  {
    "ParameterKey": "AutoRegisterDNS", 5
    "ParameterValue": "yes" 6
  },
  {
    "ParameterKey": "PrivateHostedZoneId", 7
    "ParameterValue": "<random_string>" 8
  },
  {
    "ParameterKey": "PrivateHostedZoneName", 9
    "ParameterValue": "mycluster.example.com" 10
  },
],
```

```

{
  "ParameterKey": "Master0Subnet", 11
  "ParameterValue": "subnet-<random_string>" 12
},
{
  "ParameterKey": "Master1Subnet", 13
  "ParameterValue": "subnet-<random_string>" 14
},
{
  "ParameterKey": "Master2Subnet", 15
  "ParameterValue": "subnet-<random_string>" 16
},
{
  "ParameterKey": "MasterSecurityGroupId", 17
  "ParameterValue": "sg-<random_string>" 18
},
{
  "ParameterKey": "IgnitionLocation", 19
  "ParameterValue": "https://api-int.<cluster_name>.<domain_name>:22623/config/master"
20
},
{
  "ParameterKey": "CertificateAuthorities", 21
  "ParameterValue": "data:text/plain;charset=utf-8;base64,ABC...xYz==" 22
},
{
  "ParameterKey": "MasterInstanceProfileName", 23
  "ParameterValue": "<roles_stack>-MasterInstanceProfile-<random_string>" 24
},
{
  "ParameterKey": "MasterInstanceType", 25
  "ParameterValue": "m4.xlarge" 26
},
{
  "ParameterKey": "AutoRegisterELB", 27
  "ParameterValue": "yes" 28
},
{
  "ParameterKey": "RegisterNlbPTargetsLambdaArn", 29
  "ParameterValue": "arn:aws:lambda:<region>:<account_number>:function:
<dns_stack_name>-RegisterNlbPTargets-<random_string>" 30
},
{
  "ParameterKey": "ExternalApiTargetGroupArn", 31
  "ParameterValue": "arn:aws:elasticloadbalancing:<region>:
<account_number>:targetgroup/<dns_stack_name>-Exter-<random_string>" 32
},
{
  "ParameterKey": "InternalApiTargetGroupArn", 33
  "ParameterValue": "arn:aws:elasticloadbalancing:<region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<random_string>" 34
},
{

```

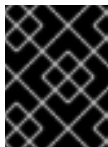
```

    "ParameterKey": "InternalServiceTargetGroupArn", 35
    "ParameterValue": "arn:aws:elasticloadbalancing:<region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<random_string>" 36
  }
]

```

- 1 您的 Ignition 配置文件中为集群编码的集群基础架构名称。
- 2 指定从 Ignition 配置文件元数据中提取的基础架构名称，其格式为 **<cluster-name>-<random-string>**。
- 3 用于 control plane 机器的当前 Red Hat Enterprise Linux CoreOS (RHCOS) AMI。
- 4 指定 **AWS::EC2::Image::Id** 值。
- 5 是否要执行 DNS etcd 注册。
- 6 指定 **yes** 或 **no**。如果指定 **yes**，您必须提供托管区信息。
- 7 用来注册 etcd 目标的 Route 53 专用区 ID。
- 8 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **PrivateHostedZoneId** 值。
- 9 用来注册目标的 Route 53 区。
- 10 指定 **<cluster\_name>.<domain\_name>**，其中 **<domain\_name>** 是您为集群生成 **install-config.yaml** 文件时所用的 Route 53 基域。请勿包含 AWS 控制台中显示的结尾句点 (.)。
- 11 13 15 在其中启动 control plane 机器的子网，最好是专用子网。
- 12 14 16 从 DNS 和负载均衡的 CloudFormation 模板输出的 **PrivateSubnets** 值指定子网。
- 17 与 control plane 节点（也称为 master 节点）关联的 master 安全组 ID。
- 18 指定安全组和角色的 CloudFormation 模板输出的 **MasterSecurityGroupId** 值。
- 19 从中获取 control plane Ignition 配置文件的位置。
- 20 指定生成的 Ignition 配置文件的位置，[https://api-int.<cluster\\_name>.<domain\\_name>:22623/config/master](https://api-int.<cluster_name>.<domain_name>:22623/config/master)。
- 21 要使用的 base64 编码证书颁发机构字符串。
- 22 指定安装目录中 **master.ign** 文件中的值。这个值是一个长字符串，格式为 **data:text/plain;charset=utf-8;base64,ABC...xYz==**。
- 23 与 control plane 节点关联的 IAM 配置集。
- 24 指定安全组和角色的 CloudFormation 模板输出的 **MasterInstanceProfile** 参数值。
- 25 用于 control plane 机器的 AWS 实例类型。
- 26 允许的值：
  - **m4.xlarge**
  - **m4.2xlarge**

- **m4.4xlarge**
- **m4.8xlarge**
- **m4.10xlarge**
- **m4.16xlarge**
- **m5.xlarge**
- **m5.2xlarge**
- **m5.4xlarge**
- **m5.8xlarge**
- **m5.10xlarge**
- **m5.16xlarge**
- **m6i.xlarge**
- **c4.2xlarge**
- **c4.4xlarge**
- **c4.8xlarge**
- **r4.xlarge**
- **r4.2xlarge**
- **r4.4xlarge**
- **r4.8xlarge**
- **r4.16xlarge**

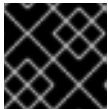


### 重要

如果您的区域中没有 **m4** 实例类型，例如 **eu-west-3**，请改为指定 **m5** 类型，如 **m5.xlarge**。

- 27** 是否要注册网络负载均衡器 (NLB)。
- 28** 指定 **yes** 或 **no**。如果指定 **yes**，您必须提供一个 Lambda Amazon Resource Name (ARN) 值。
- 29** NLB IP 目标注册 lambda 组的 ARN。
- 30** 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **RegisterNlbIpTargetsLambda** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
- 31** 外部 API 负载均衡器目标组的 ARN。
- 32** 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **ExternalApiTargetGroupArn** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。

- 33 内部 API 负载均衡器目标组群的 ARN。
  - 34 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **InternalApiTargetGroupArn** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
  - 35 内部服务负载均衡器目标组群的 ARN。
  - 36 指定 DNS 和负载均衡的 CloudFormation 模板输出的 **InternalServiceTargetGroupArn** 值。如果将集群部署到 AWS GovCloud 区域，请使用 **arn:aws-us-gov**。
2. 复制 **control plane** 机器的 CloudFormation 模板一节中的模板，并将它以 YAML 文件形式保存到计算机上。此模板描述了集群所需的 control plane 机器。
  3. 如果您将 **m5** 实例类型指定为 **MasterInstanceType** 的值，请将该实例类型添加到 CloudFormation 模板中的 **MasterInstanceType.AllowedValues** 参数。
  4. 启动 CloudFormation 模板，以创建代表 control plane 节点的 AWS 资源堆栈：



### 重要

您必须在一行内输入命令。

```
$ aws cloudformation create-stack --stack-name <name> 1
--template-body file://<template>.yaml 2
--parameters file://<parameters>.json 3
```

- 1 **<name>** 是 CloudFormation 堆栈的名称，如 **cluster-control-plane**。如果您删除集群，则需要此堆栈的名称。
- 2 **<template>** 是您保存的 CloudFormation 模板 YAML 文件的相对路径和名称。
- 3 **<parameters>** 是 CloudFormation 参数 JSON 文件的相对路径和名称。

### 输出示例

```
arn:aws:cloudformation:us-east-1:269333783861:stack/cluster-control-plane/21c7e2b0-2ee2-11eb-c6f6-0aa34627df4b
```



### 注意

CloudFormation 模板会创建一个堆栈，它代表三个 control plane 节点。

5. 确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <name>
```

#### 1.10.13.1. control plane 机器的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署 OpenShift Container Platform 集群所需的 control plane 机器。

## 例 1.47. control plane 机器的 CloudFormation 模板

AWSTemplateFormatVersion: 2010-09-09

Description: Template for OpenShift Cluster Node Launch (EC2 master instances)

Parameters:

InfrastructureName:

AllowedPattern: ^([a-zA-Z][a-zA-Z0-9\-\\_]{0,26})\$

MaxLength: 27

MinLength: 1

ConstraintDescription: Infrastructure name must be alphanumeric, start with a letter, and have a maximum of 27 characters.

Description: A short, unique cluster ID used to tag nodes for the kubelet cloud provider.

Type: String

RhcosAmi:

Description: Current Red Hat Enterprise Linux CoreOS AMI to use for bootstrap.

Type: AWS::EC2::Image::Id

AutoRegisterDNS:

Default: "yes"

AllowedValues:

- "yes"

- "no"

Description: Do you want to invoke DNS etcd registration, which requires Hosted Zone information?

Type: String

PrivateHostedZoneId:

Description: The Route53 private zone ID to register the etcd targets with, such as Z21IXYZABCZ2A4.

Type: String

PrivateHostedZoneName:

Description: The Route53 zone to register the targets with, such as cluster.example.com. Omit the trailing period.

Type: String

Master0Subnet:

Description: The subnets, recommend private, to launch the master nodes into.

Type: AWS::EC2::Subnet::Id

Master1Subnet:

Description: The subnets, recommend private, to launch the master nodes into.

Type: AWS::EC2::Subnet::Id

Master2Subnet:

Description: The subnets, recommend private, to launch the master nodes into.

Type: AWS::EC2::Subnet::Id

MasterSecurityGroupId:

Description: The master security group ID to associate with master nodes.

Type: AWS::EC2::SecurityGroup::Id

IgnitionLocation:

Default: https://api-int.\$CLUSTER\_NAME.\$DOMAIN:22623/config/master

Description: Ignition config file location.

Type: String

CertificateAuthorities:

Default: data:text/plain;charset=utf-8;base64,ABC...xYz==

Description: Base64 encoded certificate authority string to use.

Type: String

MasterInstanceProfileName:

Description: IAM profile to associate with master nodes.

Type: String

**MasterInstanceType:**

Default: m5.xlarge

Type: String

AllowedValues:

- "m4.xlarge"
- "m4.2xlarge"
- "m4.4xlarge"
- "m4.10xlarge"
- "m4.16xlarge"
- "m5.xlarge"
- "m5.2xlarge"
- "m5.4xlarge"
- "m5.8xlarge"
- "m5.12xlarge"
- "m5.16xlarge"
- "m5a.xlarge"
- "m5a.2xlarge"
- "m5a.4xlarge"
- "m5a.8xlarge"
- "m5a.10xlarge"
- "m5a.16xlarge"
- "c4.2xlarge"
- "c4.4xlarge"
- "c4.8xlarge"
- "c5.2xlarge"
- "c5.4xlarge"
- "c5.9xlarge"
- "c5.12xlarge"
- "c5.18xlarge"
- "c5.24xlarge"
- "c5a.2xlarge"
- "c5a.4xlarge"
- "c5a.8xlarge"
- "c5a.12xlarge"
- "c5a.16xlarge"
- "c5a.24xlarge"
- "r4.xlarge"
- "r4.2xlarge"
- "r4.4xlarge"
- "r4.8xlarge"
- "r4.16xlarge"
- "r5.xlarge"
- "r5.2xlarge"
- "r5.4xlarge"
- "r5.8xlarge"
- "r5.12xlarge"
- "r5.16xlarge"
- "r5.24xlarge"
- "r5a.xlarge"
- "r5a.2xlarge"
- "r5a.4xlarge"
- "r5a.8xlarge"
- "r5a.12xlarge"
- "r5a.16xlarge"
- "r5a.24xlarge"



AutoRegisterELB:

Default: "yes"

AllowedValues:

- "yes"
- "no"

Description: Do you want to invoke NLB registration, which requires a Lambda ARN parameter?

Type: String

RegisterNlbTargetsLambdaArn:

Description: ARN for NLB IP target registration lambda. Supply the value from the cluster infrastructure or select "no" for AutoRegisterELB.

Type: String

ExternalApiTargetGroupArn:

Description: ARN for external API load balancer target group. Supply the value from the cluster infrastructure or select "no" for AutoRegisterELB.

Type: String

InternalApiTargetGroupArn:

Description: ARN for internal API load balancer target group. Supply the value from the cluster infrastructure or select "no" for AutoRegisterELB.

Type: String

InternalServiceTargetGroupArn:

Description: ARN for internal service load balancer target group. Supply the value from the cluster infrastructure or select "no" for AutoRegisterELB.

Type: String

Metadata:

AWS::CloudFormation::Interface:

ParameterGroups:

- Label:

default: "Cluster Information"

Parameters:

- InfrastructureName

- Label:

default: "Host Information"

Parameters:

- MasterInstanceType

- RhcosAmi

- IgnitionLocation

- CertificateAuthorities

- MasterSecurityGroupId

- MasterInstanceProfileName

- Label:

default: "Network Configuration"

Parameters:

- VpcId

- AllowedBootstrapSshCidr

- Master0Subnet

- Master1Subnet

- Master2Subnet

- Label:

default: "DNS"

Parameters:

- AutoRegisterDNS

- PrivateHostedZoneName

- PrivateHostedZoneId

- Label:

default: "Load Balancer Automation"

Parameters:

- AutoRegisterELB
- RegisterNlbTargetsLambdaArn
- ExternalApiTargetGroupArn
- InternalApiTargetGroupArn
- InternalServiceTargetGroupArn

ParameterLabels:

InfrastructureName:  
default: "Infrastructure Name"

VpcId:  
default: "VPC ID"

Master0Subnet:  
default: "Master-0 Subnet"

Master1Subnet:  
default: "Master-1 Subnet"

Master2Subnet:  
default: "Master-2 Subnet"

MasterInstanceType:  
default: "Master Instance Type"

MasterInstanceProfileName:  
default: "Master Instance Profile Name"

RhcosAmi:  
default: "Red Hat Enterprise Linux CoreOS AMI ID"

BootstrapIgnitionLocation:  
default: "Master Ignition Source"

CertificateAuthorities:  
default: "Ignition CA String"

MasterSecurityGroupId:  
default: "Master Security Group ID"

AutoRegisterDNS:  
default: "Use Provided DNS Automation"

AutoRegisterELB:  
default: "Use Provided ELB Automation"

PrivateHostedZoneName:  
default: "Private Hosted Zone Name"

PrivateHostedZoneId:  
default: "Private Hosted Zone ID"

Conditions:

DoRegistration: !Equals ["yes", !Ref AutoRegisterELB]

DoDns: !Equals ["yes", !Ref AutoRegisterDNS]

Resources:

Master0:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref RhcosAmi

BlockDeviceMappings:

- DeviceName: /dev/xvda

Ebs:

VolumeSize: "120"

VolumeType: "gp2"

IamInstanceProfile: !Ref MasterInstanceProfileName

InstanceType: !Ref MasterInstanceType

NetworkInterfaces:

- AssociatePublicIp: "false"

```

DeviceIndex: "0"
GroupSet:
- !Ref "MasterSecurityGroupId"
SubnetId: !Ref "Master0Subnet"
UserData:
  Fn::Base64: !Sub
    - '{"ignition":{"config":{"merge":{"source":"${SOURCE}"},"security":{"tls":{"certificateAuthorities":{"source":"${CA_BUNDLE}"},"version":"3.1.0"}}}}'
    - {
      SOURCE: !Ref IgnitionLocation,
      CA_BUNDLE: !Ref CertificateAuthorities,
    }
  Tags:
    - Key: !Join [ "", ["kubernetes.io/cluster/", !Ref InfrastructureName] ]
      Value: "shared"

```

#### RegisterMaster0:

```

Condition: DoRegistration
Type: Custom::NLBRegister
Properties:
  ServiceToken: !Ref RegisterNlbTargetsLambdaArn
  TargetArn: !Ref ExternalApiTargetGroupArn
  TargetIp: !GetAtt Master0.PrivateIp

```

#### RegisterMaster0InternalApiTarget:

```

Condition: DoRegistration
Type: Custom::NLBRegister
Properties:
  ServiceToken: !Ref RegisterNlbTargetsLambdaArn
  TargetArn: !Ref InternalApiTargetGroupArn
  TargetIp: !GetAtt Master0.PrivateIp

```

#### RegisterMaster0InternalServiceTarget:

```

Condition: DoRegistration
Type: Custom::NLBRegister
Properties:
  ServiceToken: !Ref RegisterNlbTargetsLambdaArn
  TargetArn: !Ref InternalServiceTargetGroupArn
  TargetIp: !GetAtt Master0.PrivateIp

```

#### Master1:

```

Type: AWS::EC2::Instance
Properties:
  ImageId: !Ref RhcosAmi
  BlockDeviceMappings:
    - DeviceName: /dev/xvda
      Ebs:
        VolumeSize: "120"
        VolumeType: "gp2"
  IamInstanceProfile: !Ref MasterInstanceProfileName
  InstanceType: !Ref MasterInstanceType
  NetworkInterfaces:
    - AssociatePublicIpAddress: "false"
      DeviceIndex: "0"
      GroupSet:
        - !Ref "MasterSecurityGroupId"

```

```

    SubnetId: !Ref "Master1Subnet"
  UserData:
    Fn::Base64: !Sub
      - '{"ignition":{"config":{"merge":[{"source":"${SOURCE}"}]},"security":{"tls":
{"certificateAuthorities":[{"source":"${CA_BUNDLE}"}]},"version":"3.1.0"}}'
      - {
        SOURCE: !Ref IgnitionLocation,
        CA_BUNDLE: !Ref CertificateAuthorities,
      }
  Tags:
    - Key: !Join [ "", ["kubernetes.io/cluster/", !Ref InfrastructureName]]
      Value: "shared"

```

```

RegisterMaster1:
  Condition: DoRegistration
  Type: Custom::NLBRegister
  Properties:
    ServiceToken: !Ref RegisterNlbTargetsLambdaArn
    TargetArn: !Ref ExternalApiTargetGroupArn
    TargetIp: !GetAtt Master1.PrivateIp

```

```

RegisterMaster1InternalApiTarget:
  Condition: DoRegistration
  Type: Custom::NLBRegister
  Properties:
    ServiceToken: !Ref RegisterNlbTargetsLambdaArn
    TargetArn: !Ref InternalApiTargetGroupArn
    TargetIp: !GetAtt Master1.PrivateIp

```

```

RegisterMaster1InternalServiceTarget:
  Condition: DoRegistration
  Type: Custom::NLBRegister
  Properties:
    ServiceToken: !Ref RegisterNlbTargetsLambdaArn
    TargetArn: !Ref InternalServiceTargetGroupArn
    TargetIp: !GetAtt Master1.PrivateIp

```

```

Master2:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !Ref RhcosAmi
    BlockDeviceMappings:
      - DeviceName: /dev/xvda
        Ebs:
          VolumeSize: "120"
          VolumeType: "gp2"
    IamInstanceProfile: !Ref MasterInstanceProfileName
    InstanceType: !Ref MasterInstanceType
    NetworkInterfaces:
      - AssociatePublicIpAddress: "false"
        DeviceIndex: "0"
        GroupSet:
          - !Ref "MasterSecurityGroupId"
        SubnetId: !Ref "Master2Subnet"
  UserData:
    Fn::Base64: !Sub

```

```

- {"ignition":{"config":{"merge":[{"source":"${SOURCE}"]},"security":{"tls":{"certificateAuthorities":[{"source":"${CA_BUNDLE}"]}},"version":"3.1.0"}}}
- {
  SOURCE: !Ref IgnitionLocation,
  CA_BUNDLE: !Ref CertificateAuthorities,
}
Tags:
- Key: !Join [ "", ["kubernetes.io/cluster/", !Ref InfrastructureName] ]
  Value: "shared"

```

#### RegisterMaster2:

```

Condition: DoRegistration
Type: Custom::NLBRegister
Properties:
  ServiceToken: !Ref RegisterNlbTargetsLambdaArn
  TargetArn: !Ref ExternalApiTargetGroupArn
  TargetIp: !GetAtt Master2.PrivateIp

```

#### RegisterMaster2InternalApiTarget:

```

Condition: DoRegistration
Type: Custom::NLBRegister
Properties:
  ServiceToken: !Ref RegisterNlbTargetsLambdaArn
  TargetArn: !Ref InternalApiTargetGroupArn
  TargetIp: !GetAtt Master2.PrivateIp

```

#### RegisterMaster2InternalServiceTarget:

```

Condition: DoRegistration
Type: Custom::NLBRegister
Properties:
  ServiceToken: !Ref RegisterNlbTargetsLambdaArn
  TargetArn: !Ref InternalServiceTargetGroupArn
  TargetIp: !GetAtt Master2.PrivateIp

```

#### EtcdSrvRecords:

```

Condition: DoDns
Type: AWS::Route53::RecordSet
Properties:
  HostedZoneId: !Ref PrivateHostedZoneId
  Name: !Join [ ".", ["_etcd-server-ssl._tcp", !Ref PrivateHostedZoneName] ]
  ResourceRecords:
  - !Join [
    " ",
    ["0 10 2380", !Join [ ".", ["etcd-0", !Ref PrivateHostedZoneName] ]],
  ]
  - !Join [
    " ",
    ["0 10 2380", !Join [ ".", ["etcd-1", !Ref PrivateHostedZoneName] ]],
  ]
  - !Join [
    " ",
    ["0 10 2380", !Join [ ".", ["etcd-2", !Ref PrivateHostedZoneName] ]],
  ]
  TTL: 60
  Type: SRV

```

```

Etcd0Record:
  Condition: DoDns
  Type: AWS::Route53::RecordSet
  Properties:
    HostedZoneId: !Ref PrivateHostedZoneId
    Name: !Join [".", ["etcd-0", !Ref PrivateHostedZoneName]]
    ResourceRecords:
      - !GetAtt Master0.PrivateIp
    TTL: 60
    Type: A

Etcd1Record:
  Condition: DoDns
  Type: AWS::Route53::RecordSet
  Properties:
    HostedZoneId: !Ref PrivateHostedZoneId
    Name: !Join [".", ["etcd-1", !Ref PrivateHostedZoneName]]
    ResourceRecords:
      - !GetAtt Master1.PrivateIp
    TTL: 60
    Type: A

Etcd2Record:
  Condition: DoDns
  Type: AWS::Route53::RecordSet
  Properties:
    HostedZoneId: !Ref PrivateHostedZoneId
    Name: !Join [".", ["etcd-2", !Ref PrivateHostedZoneName]]
    ResourceRecords:
      - !GetAtt Master2.PrivateIp
    TTL: 60
    Type: A

Outputs:
  PrivateIPs:
    Description: The control-plane node private IP addresses.
    Value:
      !Join [
        ",",
        [!GetAtt Master0.PrivateIp, !GetAtt Master1.PrivateIp, !GetAtt Master2.PrivateIp]
      ]

```

#### 1.10.14. 在 AWS 中创建 worker 节点

您可以在 Amazon Web Services (AWS) 中创建 worker 节点，供集群使用。

您可以使用提供的 CloudFormation 模板和自定义参数文件创建代表 worker 节点的 AWS 资源堆栈。



#### 重要

CloudFormation 模板会创建一个堆栈，它代表一个 worker 节点。您必须为每个 worker 节点创建一个堆栈。



## 注意

如果不使用提供的 CloudFormation 模板来创建 worker 节点，您必须检查提供的信息并手动创建基础架构。如果集群没有正确初始化，您可能需要联系红帽支持并提供您的安装日志。

### 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。
- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。
- 您在 AWS 中创建并配置了 DNS、负载均衡器和监听程序。
- 您在 AWS 中创建了集群所需的安全组和角色。
- 已创建 bootstrap 机器。
- 已创建 control plane 机器。

### 流程

1. 创建一个 JSON 文件，其包含 CloudFormation 模板需要的参数值：

```
[
  {
    "ParameterKey": "InfrastructureName", 1
    "ParameterValue": "mycluster-<random_string>" 2
  },
  {
    "ParameterKey": "RhcosAmi", 3
    "ParameterValue": "ami-<random_string>" 4
  },
  {
    "ParameterKey": "Subnet", 5
    "ParameterValue": "subnet-<random_string>" 6
  },
  {
    "ParameterKey": "WorkerSecurityGroupId", 7
    "ParameterValue": "sg-<random_string>" 8
  },
  {
    "ParameterKey": "IgnitionLocation", 9
    "ParameterValue": "https://api-int.<cluster_name>.<domain_name>:22623/config/worker"
  },
  {
    "ParameterKey": "CertificateAuthorities", 11
    "ParameterValue": "" 12
  },
]
```

```

{
  "ParameterKey": "WorkerInstanceProfileName", 13
  "ParameterValue": "" 14
},
{
  "ParameterKey": "WorkerInstanceType", 15
  "ParameterValue": "m4.large" 16
}
]

```

- 1 您的 Ignition 配置文件中为集群编码的集群基础架构名称。
- 2 指定从 Ignition 配置文件元数据中提取的基础架构名称，其格式为 **<cluster-name>-<random-string>**。
- 3 用于 worker 节点的当前 Red Hat Enterprise Linux CoreOS (RHCOS) AMI。
- 4 指定 **AWS::EC2::Image::Id** 值。
- 5 在其中启动 worker 节点的子网，最好是专用子网。
- 6 从 DNS 和负载均衡的 CloudFormation 模板输出的 **PrivateSubnets** 值指定子网。
- 7 与 worker 节点关联的 worker 安全组 ID。
- 8 指定安全组和角色的 CloudFormation 模板输出的 **WorkerSecurityGroupId** 值。
- 9 从中获取 bootstrap Ignition 配置文件的位置。
- 10 指定生成的 Ignition 配置的位置，[https://api-int.<cluster\\_name>.<domain\\_name>:22623/config/worker](https://api-int.<cluster_name>.<domain_name>:22623/config/worker)。
- 11 要使用的 Base64 编码证书颁发机构字符串。
- 12 指定安装目录下 **worker.ign** 文件中的值。这个值是一个长字符串，格式为 **data:text/plain;charset=utf-8;base64,ABC...xYz==**。
- 13 与 worker 节点关联的 IAM 配置集。
- 14 指定安全组和角色的 CloudFormation 模板输出的 **WorkerInstanceProfile** 参数值。
- 15 用于 control plane 机器的 AWS 实例类型。
- 16 允许的值：
  - **m4.large**
  - **m4.xlarge**
  - **m4.2xlarge**
  - **m4.4xlarge**
  - **m4.8xlarge**
  - **m4.10xlarge**



- **m4.16xlarge**
- **m5.large**
- **m5.xlarge**
- **m5.2xlarge**
- **m5.4xlarge**
- **m5.8xlarge**
- **m5.10xlarge**
- **m5.16xlarge**
- **m6i.xlarge**
- **c4.2xlarge**
- **c4.4xlarge**
- **c4.8xlarge**
- **r4.large**
- **r4.xlarge**
- **r4.2xlarge**
- **r4.4xlarge**
- **r4.8xlarge**
- **r4.16xlarge**



### 重要

如果您的区域中没有 **m4** 实例类型，例如 **eu-west-3**，请改为使用 **m5** 类型。

2. 复制 **worker** 机器的 **CloudFormation** 模板一节中的模板，并将它以 **YAML** 文件形式保存到计算机上。此模板描述了集群所需的网络对象和负载均衡器。
3. 如果您将 **m5** 实例类型指定为 **WorkerInstanceType** 的值，请将该实例类型添加到 **CloudFormation** 模板中的 **WorkerInstanceType.AllowedValues** 参数。
4. 启动 **CloudFormation** 模板，以创建代表 **worker** 节点的 **AWS** 资源堆栈：



### 重要

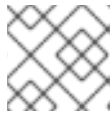
您必须在一行内输入命令。

```
$ aws cloudformation create-stack --stack-name <name> ❶
  --template-body file://<template>.yaml \ ❷
  --parameters file://<parameters>.json ❸
```

- ❶ **<name>** 是 CloudFormation 堆栈的名称，如 **cluster-worker-1**。如果您删除集群，则需要此堆栈的名称。
- ❷ **<template>** 是您保存的 CloudFormation 模板 YAML 文件的相对路径和名称。
- ❸ **<parameters>** 是 CloudFormation 参数 JSON 文件的相对路径和名称。

### 输出示例

```
arn:aws:cloudformation:us-east-1:269333783861:stack/cluster-worker-1/729ee301-1c2a-11eb-348f-sd9888c65b59
```



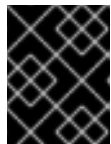
#### 注意

CloudFormation 模板会创建一个堆栈，它代表一个 worker 节点。

5. 确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <name>
```

6. 继续创建 worker 堆栈，直到为集群创建了充足的 worker 机器。您可以通过引用同一模板和参数文件并指定不同的堆栈名称来创建额外的 worker 堆栈。



#### 重要

您必须至少创建两台 worker 机器，因此您必须创建至少两个使用此 CloudFormation 模板的堆栈。

### 1.10.14.1. worker 机器的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署 OpenShift Container Platform 集群所需的 worker 机器。

#### 例 1.48. worker 机器的 CloudFormation 模板

```
AWSTemplateFormatVersion: 2010-09-09
Description: Template for OpenShift Cluster Node Launch (EC2 worker instance)

Parameters:
  InfrastructureName:
    AllowedPattern: ^([a-zA-Z][a-zA-Z0-9-]{0,26})$
    MaxLength: 27
    MinLength: 1
    ConstraintDescription: Infrastructure name must be alphanumeric, start with a letter, and have a maximum of 27 characters.
    Description: A short, unique cluster ID used to tag nodes for the kubelet cloud provider.
    Type: String
  RhcosAmi:
    Description: Current Red Hat Enterprise Linux CoreOS AMI to use for bootstrap.
```

Type: AWS::EC2::Image::Id  
Subnet:  
Description: The subnets, recommend private, to launch the master nodes into.  
Type: AWS::EC2::Subnet::Id  
WorkerSecurityGroupId:  
Description: The master security group ID to associate with master nodes.  
Type: AWS::EC2::SecurityGroup::Id  
IgnitionLocation:  
Default: [https://api-int.\\$CLUSTER\\_NAME.\\$DOMAIN:22623/config/worker](https://api-int.$CLUSTER_NAME.$DOMAIN:22623/config/worker)  
Description: Ignition config file location.  
Type: String  
CertificateAuthorities:  
Default: data:text/plain;charset=utf-8;base64,ABC...xYz==  
Description: Base64 encoded certificate authority string to use.  
Type: String  
WorkerInstanceProfileName:  
Description: IAM profile to associate with master nodes.  
Type: String  
WorkerInstanceType:  
Default: m5.large  
Type: String  
AllowedValues:  
- "m4.large"  
- "m4.xlarge"  
- "m4.2xlarge"  
- "m4.4xlarge"  
- "m4.10xlarge"  
- "m4.16xlarge"  
- "m5.large"  
- "m5.xlarge"  
- "m5.2xlarge"  
- "m5.4xlarge"  
- "m5.8xlarge"  
- "m5.12xlarge"  
- "m5.16xlarge"  
- "m5a.large"  
- "m5a.xlarge"  
- "m5a.2xlarge"  
- "m5a.4xlarge"  
- "m5a.8xlarge"  
- "m5a.10xlarge"  
- "m5a.16xlarge"  
- "c4.large"  
- "c4.xlarge"  
- "c4.2xlarge"  
- "c4.4xlarge"  
- "c4.8xlarge"  
- "c5.large"  
- "c5.xlarge"  
- "c5.2xlarge"  
- "c5.4xlarge"  
- "c5.9xlarge"  
- "c5.12xlarge"  
- "c5.18xlarge"  
- "c5.24xlarge"  
- "c5a.large"

- "c5a.xlarge"
- "c5a.2xlarge"
- "c5a.4xlarge"
- "c5a.8xlarge"
- "c5a.12xlarge"
- "c5a.16xlarge"
- "c5a.24xlarge"
- "r4.large"
- "r4.xlarge"
- "r4.2xlarge"
- "r4.4xlarge"
- "r4.8xlarge"
- "r4.16xlarge"
- "r5.large"
- "r5.xlarge"
- "r5.2xlarge"
- "r5.4xlarge"
- "r5.8xlarge"
- "r5.12xlarge"
- "r5.16xlarge"
- "r5.24xlarge"
- "r5a.large"
- "r5a.xlarge"
- "r5a.2xlarge"
- "r5a.4xlarge"
- "r5a.8xlarge"
- "r5a.12xlarge"
- "r5a.16xlarge"
- "r5a.24xlarge"
- "t3.large"
- "t3.xlarge"
- "t3.2xlarge"
- "t3a.large"
- "t3a.xlarge"
- "t3a.2xlarge"

Metadata:

AWS::CloudFormation::Interface:

ParameterGroups:

- Label:

default: "Cluster Information"

Parameters:

- InfrastructureName

- Label:

default: "Host Information"

Parameters:

- WorkerInstanceType

- RhcosAmi

- IgnitionLocation

- CertificateAuthorities

- WorkerSecurityGroupId

- WorkerInstanceProfileName

- Label:

default: "Network Configuration"

Parameters:

- Subnet

```

ParameterLabels:
  Subnet:
    default: "Subnet"
  InfrastructureName:
    default: "Infrastructure Name"
  WorkerInstanceType:
    default: "Worker Instance Type"
  WorkerInstanceProfileName:
    default: "Worker Instance Profile Name"
  RhcosAmi:
    default: "Red Hat Enterprise Linux CoreOS AMI ID"
  IgnitionLocation:
    default: "Worker Ignition Source"
  CertificateAuthorities:
    default: "Ignition CA String"
  WorkerSecurityGroupId:
    default: "Worker Security Group ID"

Resources:
  Worker0:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref RhcosAmi
      BlockDeviceMappings:
        - DeviceName: /dev/xvda
          Ebs:
            VolumeSize: "120"
            VolumeType: "gp2"
      IamInstanceProfile: !Ref WorkerInstanceProfileName
      InstanceType: !Ref WorkerInstanceType
      NetworkInterfaces:
        - AssociatePublicIp: "false"
          DeviceIndex: "0"
          GroupSet:
            - !Ref "WorkerSecurityGroupId"
          SubnetId: !Ref "Subnet"
      UserData:
        Fn::Base64: !Sub
          - '{"ignition":{"config":{"merge":[{"source":"${SOURCE}"}]},"security":{"tls":
{"certificateAuthorities":[{"source":"${CA_BUNDLE}"}]},"version":"3.1.0"}}'
          - {
              SOURCE: !Ref IgnitionLocation,
              CA_BUNDLE: !Ref CertificateAuthorities,
            }
      Tags:
        - Key: !Join ["", ["kubernetes.io/cluster/", !Ref InfrastructureName]]
          Value: "shared"

Outputs:
  PrivateIP:
    Description: The compute node private IP address.
    Value: !GetAtt Worker0.PrivateIp

```

### 1.10.15. 使用用户置备的基础架构在 AWS 上初始化 bootstrap 序列

在 Amazon Web Services (AWS) 中创建所有所需的基础架构后，您可以启动初始化 OpenShift Container Platform control plane 的 bootstrap 序列。

### 先决条件

- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。
- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。
- 您在 AWS 中创建并配置了 DNS、负载均衡器和监听程序。
- 您在 AWS 中创建了集群所需的安全组和角色。
- 已创建 bootstrap 机器。
- 已创建 control plane 机器。
- 已创建 worker 节点。

### 流程

1. 更改为包含安装程序的目录，并启动初始化 OpenShift Container Platform control plane 的 bootstrap 过程：

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

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

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

### 输出示例

```
INFO Waiting up to 20m0s for the Kubernetes API at  
https://api.mycluster.example.com:6443...  
INFO API v1.19.0+9f84db3 up  
INFO Waiting up to 30m0s for bootstrapping to complete...  
INFO It is now safe to remove the bootstrap resources  
INFO Time elapsed: 1s
```

如果命令退出时没有 **FATAL** 警告，则 OpenShift Container Platform control plane 已被初始化。



### 注意

在 control plane 初始化后，它会设置计算节点，并以 Operator 的形式安装其他服务。

### 其他资源

- 如需了解在 OpenShift Container Platform 安装过程中监控安装、bootstrap 和 control plane 日志的详细信息，请参阅[监控安装进度](#)。
- 如需有关对 bootstrap 过程进行故障排除的信息，请参阅[收集 bootstrap 节点诊断数据](#)。

### 1.10.16. 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
```

### 1.10.17. 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-bootstrapper** 服务帐户提交，并确认节点的身份。

- 若要单独批准，请对每个有效的 CSR 运行以下命令：

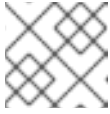
```
$ oc adm certificate approve <csr_name> 1
```

**1** <csr\_name> 是当前 CSR 列表中 CSR 的名称。



- 要批准所有待处理的 CSR，请运行以下命令：

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{"\n"}\n{{end}}\n{{end}}' | xargs --no-run-if-empty oc adm certificate approve
```



### 注意

在有些 CSR 被批准前，一些 Operator 可能无法使用。

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

```
$ oc get csr
```

### 输出示例

```
NAME      AGE   REQUESTOR                                     CONDITION
csr-bfd72 5m26s system:node:ip-10-0-50-126.us-east-2.compute.internal
Pending
csr-c57lv 5m26s system:node:ip-10-0-95-157.us-east-2.compute.internal
Pending
...
```

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

- 若要单独批准，请对每个有效的 CSR 运行以下命令：

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

**1** **<csr\_name>** 是当前 CSR 列表中 CSR 的名称。

- 要批准所有待处理的 CSR，请运行以下命令：

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

6. 批准所有客户端和服务器的 CSR 后，机器将处于 **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](#)。

## 1.10.18. 初始 Operator 配置

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

## 先决条件

- 您的 control plane 已初始化。

## 流程

1. 观察集群组件上线：

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

## 输出示例

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

2. 配置不可用的 Operator。

### 1.10.18.1. 禁用默认的 OperatorHub 源

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

#### 流程

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

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

#### 提示

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

### 1.10.18.2. 镜像 registry 存储配置

Amazon Web Services 提供默认存储，这意味着 Image Registry Operator 在安装后可用。但是，如果 Registry Operator 无法创建 S3 存储桶并自动配置存储，您需要手工配置 registry 存储。

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

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

#### 1.10.18.2.1. 为使用用户置备的基础架构的 AWS 配置 registry 存储

在安装过程中，使用您的云凭据就可以创建一个 Amazon S3 存储桶，Registry Operator 将会自动配置存储。

如果 Registry Operator 无法创建 S3 存储桶或自动配置存储，您可以按照以下流程创建 S3 存储桶并配置存储。

#### 先决条件

- 在带有用户置备的基础架构的 AWS 上有一个集群。
- 对于 Amazon S3 存储，secret 应该包含以下两个键：
  - **REGISTRY\_STORAGE\_S3\_ACCESSKEY**
  - **REGISTRY\_STORAGE\_S3\_SECRETKEY**

#### 流程

如果 Registry Operator 无法创建 S3 存储桶并自动配置存储，请进行以下操作。

1. 设置一个 [Bucket Lifecycle Policy](#) 用来终止已有一天之久的未完成的分段上传操作。
2. 在 **configs.imageregistry.operator.openshift.io/cluster** 中输入存储配置：

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

## 配置示例

```
storage:
  s3:
    bucket: <bucket-name>
    region: <region-name>
```



### 警告

为了保护 AWS 中 registry 镜像的安全，[阻止对 S3 存储桶的公共访问](#)。

### 1.10.18.2.2. 在非生产集群中配置镜像 registry 存储

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

#### 流程

- 将镜像 registry 存储设置为空目录：

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



### 警告

仅可为非生产集群配置这个选项。

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

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

等待几分钟，然后再次运行该命令。

### 1.10.19. 删除 bootstrap 资源：

完成集群的初始 Operator 配置后，从 Amazon Web Services (AWS) 中删除 bootstrap 资源。

#### 先决条件

- 已为集群完成初始的 Operator 配置。

#### 流程

1. 删除 bootstrap 资源。如果您使用了 CloudFormation 模板，请[删除其堆栈](#)：

- 使用 AWS CLI 删除堆栈：

```
$ aws cloudformation delete-stack --stack-name <name> 1
```

**1** <name> 是 bootstrap 堆栈的名称。

- 使用 [AWS CloudFormation 控制台](#) 删除堆栈。

### 1.10.20. 创建 Ingress DNS 记录

如果您删除了 DNS 区配置，请手动创建指向 Ingress 负载均衡器的 DNS 记录。您可以创建一个 wildcard 记录或具体的记录。以下流程使用了 A 记录，但您可以使用其他所需记录类型，如 CNAME 或别名。

#### 先决条件

- 已在 Amazon Web Services (AWS) 上安装了使用您置备的基础架构的 OpenShift Container Platform 集群。
- 已安装 OpenShift CLI (**oc**)。
- 安装了 **jq** 软件包。
- 您下载了 AWS CLI 并安装到您的计算机上。请参阅[使用捆绑安装程序 \(Linux、macOS 或 Unix\) 安装 AWS CLI](#)的文档。

#### 流程

1. 决定要创建的路由。

- 要创建一个 wildcard 记录，请使用 **\*.apps.<cluster\_name>.<domain\_name>**，其中 **<cluster\_name>** 是集群名称，**<domain\_name>** 是 OpenShift Container Platform 集群的 Route 53 基域。
- 要创建特定的记录，您必须为集群使用的每个路由创建一个记录，如下所示：

```
$ oc get --all-namespaces -o jsonpath='{range .items[*]}{range .status.ingress[*]}{.host}{"\n"}{end}{end}' routes
```

#### 输出示例

```
oauth-openshift.apps.<cluster_name>.<domain_name>
console-openshift-console.apps.<cluster_name>.<domain_name>
downloads-openshift-console.apps.<cluster_name>.<domain_name>
alertmanager-main-openshift-monitoring.apps.<cluster_name>.<domain_name>
grafana-openshift-monitoring.apps.<cluster_name>.<domain_name>
prometheus-k8s-openshift-monitoring.apps.<cluster_name>.<domain_name>
```

2. 获取 Ingress Operator 负载均衡器状态，并记录其使用的外部 IP 地址值，如 **EXTERNAL-IP** 列所示：

```
$ oc -n openshift-ingress get service router-default
```

## 输出示例

```

NAME          TYPE          CLUSTER-IP    EXTERNAL-IP          PORT(S)
AGE
router-default LoadBalancer 172.30.62.215  ab3...28.us-east-2.elb.amazonaws.com
80:31499/TCP,443:30693/TCP 5m

```

- 为负载均衡器定位托管区 ID :

```

$ aws elb describe-load-balancers | jq -r '.LoadBalancerDescriptions[] | select(.DNSName ==
"<external_ip>").CanonicalHostedZoneNameID' ①

```

- 对于 **<external\_ip>**, 请指定您获取的 Ingress Operator 负载均衡器的外部 IP 地址值。

## 输出示例

```
Z3AADJGX6KTTL2
```

这个命令的输出是负载均衡器托管区 ID。

- 获取集群域的公共托管区 ID :

```

$ aws route53 list-hosted-zones-by-name \
  --dns-name "<domain_name>" \ ①
  --query 'HostedZones[? Config.PrivateZone != `true` && Name ==
`<domain_name>.`].Id' ②
  --output text

```

- ② 对于 **<domain\_name>**, 请为 OpenShift Container Platform 集群指定 Route 53 基域。

## 输出示例

```
/hostedzone/Z3URY6TWQ91KVV
```

命令输出中会显示您的域的公共托管区 ID。在本例中是 **Z3URY6TWQ91KVV**。

- 在您的私有区中添加别名记录 :

```

$ aws route53 change-resource-record-sets --hosted-zone-id "<private_hosted_zone_id>" --
change-batch '{ ①
> "Changes": [
> {
>   "Action": "CREATE",
>   "ResourceRecordSet": {
>     "Name": "\\052.apps.<cluster_domain>", ②
>     "Type": "A",
>     "AliasTarget":{
>       "HostedZoneId": "<hosted_zone_id>", ③
>       "DNSName": "<external_ip>.", ④
>       "EvaluateTargetHealth": false
>     }
> }

```

```
> }
> }
> ]
>}'
```

- 1 对于 `<private_hosted_zone_id>`，指定 DNS 和负载均衡的 CloudFormation 模板输出的值。
- 2 对于 `<cluster_domain>`，请指定用于 OpenShift Container Platform 集群的域或子域。
- 3 对于 `<hosted_zone_id>`，请为您获得的负载均衡器指定公共托管区 ID。
- 4 对于 `<external_ip>`，请指定 Ingress Operator 负载均衡器的外部 IP 地址值。请确定在该参数值中包含最后的句点 (.)。

6. 在您的公共区中添加记录：

```
$ aws route53 change-resource-record-sets --hosted-zone-id "<public_hosted_zone_id>" --
change-batch '{
> "Changes": [
> {
>   "Action": "CREATE",
>   "ResourceRecordSet": {
>     "Name": "\\052.apps.<cluster_domain>",
>     "Type": "A",
>     "AliasTarget": {
>       "HostedZoneId": "<hosted_zone_id>",
>       "DNSName": "<external_ip>.",
>       "EvaluateTargetHealth": false
>     }
>   }
> }
> ]
>}'
```

- 1 对于 `<public_hosted_zone_id>`，请为您的域指定公共托管区。
- 2 对于 `<cluster_domain>`，请指定用于 OpenShift Container Platform 集群的域或子域。
- 3 对于 `<hosted_zone_id>`，请为您获得的负载均衡器指定公共托管区 ID。
- 4 对于 `<external_ip>`，请指定 Ingress Operator 负载均衡器的外部 IP 地址值。请确定在该参数值中包含最后的句点 (.)。

### 1.10.21. 在用户置备的基础架构上完成 AWS 安装

在用户置备的基础架构 Amazon Web Service (AWS) 上启动 OpenShift Container Platform 安装后，监视进程并等待安装完成。

#### 先决条件

- 您在用户置备的 AWS 基础架构上为 OpenShift Container Platform 集群删除了 bootstrap 节点。

- 已安装 **oc** CLI。

## 流程

1. 在包含安装程序的目录中完成集群安装：

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

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

## 输出示例

```
INFO Waiting up to 40m0s for the cluster at https://api.mycluster.example.com:6443 to
initialize...
INFO Waiting up to 10m0s for the openshift-console route to be created...
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-Fe5en-ymBEc-
Wt6NL"
INFO Time elapsed: 1s
```



## 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-bootstraptrapper** 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 Ignition 配置文件生成后的 12 小时内使用它们，因为 24 小时的证书会在集群安装后的 16 小时到 22 小时进行轮转。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因为执行了证书更新而导致安装失败的问题。

2. 在 [Cluster registration](#) 页面注册您的集群。

### 1.10.22. 使用 Web 控制台登录到集群

**kubeadmin** 用户默认在 OpenShift Container Platform 安装后存在。您可以使用 OpenShift Container Platform Web 控制台以 **kubeadmin** 用户身份登录集群。

#### 先决条件

- 有访问安装主机的访问权限。
- 您完成了集群安装，所有集群 Operator 都可用。

## 流程



1. 从安装主机上的 **kubeadmin -password** 文件中获取 kubeadmin 用户的密码：

```
$ cat <installation_directory>/auth/kubeadmin-password
```

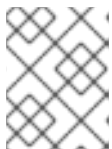


### 注意

另外，您还可以从安装主机上的 **<installation\_directory>/openshift\_install.log** 日志文件获取 **kubeadmin** 密码。

2. 列出 OpenShift Container Platform Web 控制台路由：

```
$ oc get routes -n openshift-console | grep 'console-openshift'
```



### 注意

另外，您还可以从安装主机上的 **<installation\_directory>/openshift\_install.log** 日志文件获取 OpenShift Container Platform 路由。

### 输出示例

```
console    console-openshift-console.apps.<cluster_name>.<base_domain>    console
https reencrypt/Redirect None
```

3. 在 Web 浏览器中导航到上一命令输出中包括的路由，以 **kubeadmin** 用户身份登录。

### 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](#)。

## 1.10.23. 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](#)。

## 1.10.24. 其他资源

- 如需有关 AWS CloudFormation 堆栈的更多信息，请参阅 [AWS 文档中的使用堆栈](#)。

## 1.10.25. 后续步骤

- [验证安装](#)。

- [自定义集群](#)。
- 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](#)。
- 如果需要，您可以[删除云供应商凭证](#)。

## 1.11. 在 AWS 上卸载集群

您可以删除部署到 Amazon Web Services (AWS) 的集群。

### 1.11.1. 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。

