



# OpenShift Container Platform 4.14

## 在 AWS 上安装

在 Amazon Web Services 上安装 OpenShift Container Platform



# OpenShift Container Platform 4.14 在 AWS 上安装

---

在 Amazon Web Services 上安装 OpenShift Container Platform

## Legal Notice

Copyright © 2025 Red Hat, Inc.

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.

## Abstract

本文档论述了如何在 Amazon Web Services 上安装 OpenShift Container Platform。

# Table of Contents

|                                                  |           |
|--------------------------------------------------|-----------|
| <b>第 1 章 准备在 AWS 上安装</b>                         | <b>8</b>  |
| 1.1. 先决条件                                        | 8         |
| 1.2. 在 AWS 上安装 OPENSIFT CONTAINER PLATFORM 的要求   | 8         |
| 1.3. 选择在 AWS 上安装 OPENSIFT CONTAINER PLATFORM 的方法 | 8         |
| 1.4. 后续步骤                                        | 9         |
| <b>第 2 章 配置 AWS 帐户</b>                           | <b>10</b> |
| 2.1. 配置路由 53 (ROUTE 53)                          | 10        |
| 2.2. AWS 帐户限值                                    | 11        |
| 2.3. IAM 用户所需的 AWS 权限                            | 12        |
| 2.4. 创建 IAM 用户                                   | 21        |
| 2.5. IAM 策略和 AWS 身份验证                            | 22        |
| 2.6. 支持的 AWS MARKETPLACE 区域                      | 25        |
| 2.7. 支持的 AWS 区域                                  | 26        |
| 2.8. 后续步骤                                        | 27        |
| <b>第 3 章 在 AWS 上快速安装集群</b>                       | <b>29</b> |
| 3.1. 先决条件                                        | 29        |
| 3.2. OPENSIFT CONTAINER PLATFORM 互联网访问           | 29        |
| 3.3. 为集群节点 SSH 访问生成密钥对                           | 29        |
| 3.4. 获取安装程序                                      | 31        |
| 3.5. 部署集群                                        | 32        |
| 3.6. 通过下载二进制文件安装 OPENSIFT CLI                    | 34        |
| 3.7. 使用 CLI 登录集群                                 | 36        |
| 3.8. 使用 WEB 控制台登录到集群                             | 36        |
| 3.9. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问  | 37        |
| 3.10. 后续步骤                                       | 37        |
| <b>第 4 章 使用自定义在 AWS 上安装集群</b>                    | <b>38</b> |
| 4.1. 先决条件                                        | 38        |
| 4.2. OPENSIFT CONTAINER PLATFORM 互联网访问           | 38        |
| 4.3. 为集群节点 SSH 访问生成密钥对                           | 38        |
| 4.4. 获取 AWS MARKETPLACE 镜像                       | 40        |
| 4.5. 获取安装程序                                      | 41        |
| 4.6. 创建安装配置文件                                    | 42        |
| 4.7. 通过下载二进制文件安装 OPENSIFT CLI                    | 49        |
| 4.8. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案      | 51        |
| 4.9. 部署集群                                        | 62        |
| 4.10. 使用 CLI 登录集群                                | 63        |
| 4.11. 使用 WEB 控制台登录到集群                            | 64        |
| 4.12. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问 | 65        |
| 4.13. 后续步骤                                       | 65        |
| <b>第 5 章 使用自定义网络在 AWS 上安装集群</b>                  | <b>66</b> |
| 5.1. 先决条件                                        | 66        |
| 5.2. OPENSIFT CONTAINER PLATFORM 互联网访问           | 66        |
| 5.3. 为集群节点 SSH 访问生成密钥对                           | 66        |
| 5.4. 获取安装程序                                      | 68        |
| 5.5. 网络配置阶段                                      | 69        |
| 5.6. 创建安装配置文件                                    | 69        |
| 5.7. 通过下载二进制文件安装 OPENSIFT CLI                    | 77        |
| 5.8. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案      | 79        |

|                                                  |            |
|--------------------------------------------------|------------|
| 5.9. CLUSTER NETWORK OPERATOR 配置                 | 89         |
| 5.10. 指定高级网络配置                                   | 96         |
| 5.11. 在新 AWS 集群上配置 INGRESS CONTROLLER 网络负载均衡     | 97         |
| 5.12. 使用 OVN-KUBERNETES 配置混合网络                   | 98         |
| 5.13. 部署集群                                       | 100        |
| 5.14. 使用 CLI 登录集群                                | 101        |
| 5.15. 使用 WEB 控制台登录到集群                            | 102        |
| 5.16. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问 | 103        |
| 5.17. 后续步骤                                       | 103        |
| <b>第 6 章 在受限网络中的 AWS 上安装集群</b>                   | <b>104</b> |
| 6.1. 先决条件                                        | 104        |
| 6.2. 关于在受限网络中安装                                  | 104        |
| 6.3. 关于使用自定义 VPC                                 | 105        |
| 6.4. OPENSIFT CONTAINER PLATFORM 互联网访问           | 109        |
| 6.5. 为集群节点 SSH 访问生成密钥对                           | 109        |
| 6.6. 创建安装配置文件                                    | 110        |
| 6.7. 通过下载二进制文件安装 OPENSIFT CLI                    | 118        |
| 6.8. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案      | 120        |
| 6.9. 部署集群                                        | 131        |
| 6.10. 使用 CLI 登录集群                                | 132        |
| 6.11. 禁用默认的 OPERATORHUB 目录源                      | 133        |
| 6.12. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问 | 133        |
| 6.13. 后续步骤                                       | 133        |
| <b>第 7 章 在 AWS 上将集群安装到现有的 VPC 中</b>              | <b>135</b> |
| 7.1. 先决条件                                        | 135        |
| 7.2. 关于使用自定义 VPC                                 | 135        |
| 7.3. OPENSIFT CONTAINER PLATFORM 互联网访问           | 140        |
| 7.4. 为集群节点 SSH 访问生成密钥对                           | 140        |
| 7.5. 获取安装程序                                      | 142        |
| 7.6. 创建安装配置文件                                    | 143        |
| 7.7. 通过下载二进制文件安装 OPENSIFT CLI                    | 152        |
| 7.8. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案      | 153        |
| 7.9. 部署集群                                        | 164        |
| 7.10. 使用 CLI 登录集群                                | 166        |
| 7.11. 使用 WEB 控制台登录到集群                            | 166        |
| 7.12. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问 | 167        |
| 7.13. 后续步骤                                       | 167        |
| <b>第 8 章 在 AWS 上安装私有集群</b>                       | <b>169</b> |
| 8.1. 先决条件                                        | 169        |
| 8.2. 私有集群                                        | 169        |
| 8.3. 关于使用自定义 VPC                                 | 170        |
| 8.4. OPENSIFT CONTAINER PLATFORM 互联网访问           | 174        |
| 8.5. 为集群节点 SSH 访问生成密钥对                           | 174        |
| 8.6. 获取安装程序                                      | 176        |
| 8.7. 手动创建安装配置文件                                  | 177        |
| 8.8. 通过下载二进制文件安装 OPENSIFT CLI                    | 185        |
| 8.9. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案      | 187        |
| 8.10. 部署集群                                       | 197        |
| 8.11. 使用 CLI 登录集群                                | 199        |
| 8.12. 使用 WEB 控制台登录到集群                            | 199        |
| 8.13. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问 | 200        |

|                                                    |            |
|----------------------------------------------------|------------|
| 8.14. 后续步骤                                         | 201        |
| <b>第 9 章 在 AWS 上将集群安装到一个政府区域</b>                   | <b>202</b> |
| 9.1. 先决条件                                          | 202        |
| 9.2. AWS 政府区域                                      | 202        |
| 9.3. 安装要求                                          | 202        |
| 9.4. 私有集群                                          | 202        |
| 9.5. 关于使用自定义 VPC                                   | 204        |
| 9.6. OPENSIFT CONTAINER PLATFORM 互联网访问             | 207        |
| 9.7. 为集群节点 SSH 访问生成密钥对                             | 208        |
| 9.8. 获取 AWS MARKETPLACE 镜像                         | 209        |
| 9.9. 获取安装程序                                        | 210        |
| 9.10. 手动创建安装配置文件                                   | 211        |
| 9.11. 通过下载二进制文件安装 OPENSIFT CLI                     | 219        |
| 9.12. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案       | 221        |
| 9.13. 部署集群                                         | 232        |
| 9.14. 使用 CLI 登录集群                                  | 233        |
| 9.15. 使用 WEB 控制台登录到集群                              | 234        |
| 9.16. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问   | 235        |
| 9.17. 后续步骤                                         | 235        |
| <b>第 10 章 在 AWS 上将集群安装到 SECRET 或 TOP SECRET 区域</b> | <b>236</b> |
| 10.1. 先决条件                                         | 236        |
| 10.2. AWS SECRET 区域                                | 236        |
| 10.3. 安装要求                                         | 236        |
| 10.4. 私有集群                                         | 237        |
| 10.5. 关于使用自定义 VPC                                  | 238        |
| 10.6. OPENSIFT CONTAINER PLATFORM 互联网访问            | 242        |
| 10.7. 在 AWS 中上传自定义 RHCOS AMI                       | 243        |
| 10.8. 为集群节点 SSH 访问生成密钥对                            | 245        |
| 10.9. 获取安装程序                                       | 246        |
| 10.10. 手动创建安装配置文件                                  | 247        |
| 10.11. 通过下载二进制文件安装 OPENSIFT CLI                    | 254        |
| 10.12. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案      | 256        |
| 10.13. 部署集群                                        | 267        |
| 10.14. 使用 CLI 登录集群                                 | 268        |
| 10.15. 使用 WEB 控制台登录到集群                             | 269        |
| 10.16. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问  | 269        |
| 10.17. 后续步骤                                        | 270        |
| <b>第 11 章 在 AWS CHINA 上安装集群</b>                    | <b>271</b> |
| 11.1. 先决条件                                         | 271        |
| 11.2. 安装要求                                         | 271        |
| 11.3. OPENSIFT CONTAINER PLATFORM 互联网访问            | 271        |
| 11.4. 私有集群                                         | 272        |
| 11.5. 关于使用自定义 VPC                                  | 273        |
| 11.6. 为集群节点 SSH 访问生成密钥对                            | 277        |
| 11.7. 在 AWS 中上传自定义 RHCOS AMI                       | 278        |
| 11.8. 获取安装程序                                       | 281        |
| 11.9. 手动创建安装配置文件                                   | 281        |
| 11.10. 通过下载二进制文件安装 OPENSIFT CLI                    | 290        |
| 11.11. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案      | 291        |
| 11.12. 部署集群                                        | 302        |
| 11.13. 使用 CLI 登录集群                                 | 304        |

|                                                                 |            |
|-----------------------------------------------------------------|------------|
| 11.14. 使用 WEB 控制台登录到集群                                          | 304        |
| 11.15. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问               | 305        |
| 11.16. 后续步骤                                                     | 305        |
| <b>第 12 章 使用 CLOUDFORMATION 模板在 AWS 中用户置备的基础架构上安装集群 .....</b>   | <b>307</b> |
| 12.1. 先决条件                                                      | 307        |
| 12.2. OPENSIFT CONTAINER PLATFORM 互联网访问                         | 307        |
| 12.3. 具有用户置备基础架构的集群的要求                                          | 308        |
| 12.4. 所需的 AWS 基础架构组件                                            | 311        |
| 12.5. 获取 AWS MARKETPLACE 镜像                                     | 328        |
| 12.6. 获取安装程序                                                    | 328        |
| 12.7. 为集群节点 SSH 访问生成密钥对                                         | 329        |
| 12.8. 创建用于 AWS 的安装文件                                            | 331        |
| 12.9. 提取基础架构名称                                                  | 338        |
| 12.10. 在 AWS 中创建 VPC                                            | 339        |
| 12.11. 在 AWS 中创建网络和负载均衡组件                                       | 346        |
| 12.12. 在 AWS 中创建安全组和角色                                          | 358        |
| 12.13. 使用流元数据访问 RHCOS AMI                                       | 371        |
| 12.14. AWS 基础架构的 RHCOS AMI                                      | 372        |
| 12.15. 在 AWS 中创建 BOOTSTRAP 节点                                   | 377        |
| 12.16. 在 AWS 中创建 CONTROL PLANE 机器                               | 386        |
| 12.17. 在 AWS 中创建 WORKER 节点                                      | 395        |
| 12.18. 使用用户置备的基础架构在 AWS 上初始化 BOOTSTRAP 序列                       | 401        |
| 12.19. 通过下载二进制文件安装 OPENSIFT CLI                                 | 402        |
| 12.20. 使用 CLI 登录集群                                              | 404        |
| 12.21. 批准机器的证书签名请求                                              | 404        |
| 12.22. 初始 OPERATOR 配置                                           | 407        |
| 12.23. 删除 BOOTSTRAP 资源：                                         | 409        |
| 12.24. 创建 INGRESS DNS 记录                                        | 410        |
| 12.25. 在用户置备的基础架构上完成 AWS 安装                                     | 412        |
| 12.26. 使用 WEB 控制台登录到集群                                          | 413        |
| 12.27. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问               | 414        |
| 12.28. 其他资源                                                     | 414        |
| 12.29. 后续步骤                                                     | 414        |
| <b>第 13 章 在带有 AWS LOCAL ZONES 上的 WORKER 节点的 AWS 上安装集群 .....</b> | <b>416</b> |
| 13.1. 先决条件                                                      | 416        |
| 13.2. AWS 本地区域中的集群限制                                            | 417        |
| 13.3. OPENSIFT CONTAINER PLATFORM 互联网访问                         | 417        |
| 13.4. 获取 AWS MARKETPLACE 镜像                                     | 418        |
| 13.5. 通过下载二进制文件安装 OPENSIFT CLI                                  | 419        |
| 13.6. 准备安装                                                      | 420        |
| 13.7. 选择 AWS 本地区域                                               | 428        |
| 13.8. AWS 本地区域的集群安装选项                                           | 429        |
| 13.9. 在 AWS 本地区域中快速安装集群                                         | 429        |
| 13.10. 在带有 LOCAL ZONE 子网的现有 VPC 上安装集群                           | 431        |
| 13.11. 部署集群                                                     | 442        |
| 13.12. 使用 CLI 登录集群                                              | 444        |
| 13.13. 使用 WEB 控制台登录到集群                                          | 444        |
| 13.14. 验证使用边缘计算池创建的节点                                           | 445        |
| 13.15. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问               | 446        |
| 13.16. 后续步骤                                                     | 446        |
| <b>第 14 章 在带有用户置备的受限网络中的 AWS 上安装集群 .....</b>                    | <b>447</b> |



|                                                                  |            |
|------------------------------------------------------------------|------------|
| 14.1. 先决条件                                                       | 447        |
| 14.2. 关于在受限网络中安装                                                 | 448        |
| 14.3. OPENSIFT CONTAINER PLATFORM 互联网访问                          | 448        |
| 14.4. 具有用户置备基础架构的集群的要求                                           | 448        |
| 14.5. 所需的 AWS 基础架构组件                                             | 451        |
| 14.6. 为集群节点 SSH 访问生成密钥对                                          | 468        |
| 14.7. 创建用于 AWS 的安装文件                                             | 470        |
| 14.8. 提取基础架构名称                                                   | 478        |
| 14.9. 在 AWS 中创建 VPC                                              | 479        |
| 14.10. 在 AWS 中创建网络和负载均衡组件                                        | 486        |
| 14.11. 在 AWS 中创建安全组和角色                                           | 497        |
| 14.12. 使用流元数据访问 RHCOS AMI                                        | 510        |
| 14.13. AWS 基础架构的 RHCOS AMI                                       | 511        |
| 14.14. 在 AWS 中创建 BOOTSTRAP 节点                                    | 514        |
| 14.15. 在 AWS 中创建 CONTROL PLANE 机器                                | 523        |
| 14.16. 在 AWS 中创建 WORKER 节点                                       | 533        |
| 14.17. 使用用户置备的基础架构在 AWS 上初始化 BOOTSTRAP 序列                        | 538        |
| 14.18. 使用 CLI 登录到集群                                              | 539        |
| 14.19. 批准机器的证书签名请求                                               | 539        |
| 14.20. 初始 OPERATOR 配置                                            | 542        |
| 14.21. 删除 BOOTSTRAP 资源：                                          | 545        |
| 14.22. 创建 INGRESS DNS 记录                                         | 545        |
| 14.23. 在用户置备的基础架构上完成 AWS 安装                                      | 548        |
| 14.24. 使用 WEB 控制台登录到集群                                           | 549        |
| 14.25. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问                | 550        |
| 14.26. 其他资源                                                      | 550        |
| 14.27. 后续步骤                                                      | 550        |
| <b>第 15 章 在带有 AWS OUTPOSTS 中的远程 WORKER 的 AWS 上安装集群</b>           | <b>551</b> |
| 15.1. 先决条件                                                       | 551        |
| 15.2. 关于使用自定义 VPC                                                | 551        |
| 15.3. OPENSIFT CONTAINER PLATFORM 互联网访问                          | 556        |
| 15.4. 为集群节点 SSH 访问生成密钥对                                          | 557        |
| 15.5. 获取安装程序                                                     | 558        |
| 15.6. 集群安装的最低资源要求                                                | 559        |
| 15.7. 识别 AWS OUTPOSTS 实例类型                                       | 560        |
| 15.8. 创建安装配置文件                                                   | 561        |
| 15.9. 生成清单文件                                                     | 565        |
| 15.10. 通过下载二进制文件安装 OPENSIFT CLI                                  | 567        |
| 15.11. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案                    | 569        |
| 15.12. 部署集群                                                      | 579        |
| 15.13. 使用 CLI 登录集群                                               | 581        |
| 15.14. 使用 WEB 控制台登录到集群                                           | 581        |
| 15.15. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问                | 582        |
| 15.16. 集群限制                                                      | 582        |
| 15.17. 后续步骤                                                      | 583        |
| <b>第 16 章 在 AWS 上安装三节点集群</b>                                     | <b>584</b> |
| 16.1. 配置三节点集群                                                    | 584        |
| 16.2. 后续步骤                                                       | 585        |
| <b>第 17 章 在 AWS 上卸载集群</b>                                        | <b>586</b> |
| 17.1. 删除使用安装程序置备的基础架构的集群                                         | 586        |
| 17.2. 使用 CLOUD CREDENTIAL OPERATOR 实用程序删除 AMAZON WEB SERVICES 资源 | 586        |

|                                     |            |
|-------------------------------------|------------|
| 17.3. 使用配置的 AWS LOCAL ZONE 基础架构删除集群 | 587        |
| <b>第 18 章 AWS 的安装配置参数 .....</b>     | <b>589</b> |
| 18.1. AWS 可用的安装配置参数                 | 589        |



# 第 1 章 准备在 AWS 上安装

## 1.1. 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。
- 您可以阅读[选择集群安装方法并为用户准备它](#)的文档。

## 1.2. 在 AWS 上安装 OPENSIFT CONTAINER PLATFORM 的要求

在 Amazon Web Services (AWS) 上安装 OpenShift Container Platform 前，您必须先创建一个 AWS 帐户。如需有关配置帐户、帐户限值、帐户权限、IAM 用户设置和支持的 AWS 区域的详情，请参阅[配置 AWS 帐户](#)。

如果环境中无法访问云身份和访问管理(IAM) API，或者不想将管理员级别的凭证 secret 存储在 **kube-system** 命名空间中，请参阅为 [AWS 手动创建长期凭证](#)，或将 [AWS 集群配置为使用 Amazon Web Services Security Token Service \(AWS STS\)的短期凭证](#)。

## 1.3. 选择在 AWS 上安装 OPENSIFT CONTAINER PLATFORM 的方法

您可以在安装程序置备的基础架构或用户置备的基础架构上安装 OpenShift Container Platform。默认安装类型使用安装程序置备的基础架构，安装程序会在其中为集群置备底层基础架构。您还可以在您置备的基础架构上安装 OpenShift Container Platform。如果不使用安装程序置备的基础架构，您必须自己管理和维护集群资源。

如需有关安装程序置备和用户置备的安装过程的更多信息，请参阅 [安装过程](#)。

### 1.3.1. 在单一节点上安装集群

在单一节点上安装 OpenShift Container Platform 可降低高可用性和大型集群的一些要求。但是，您必须[满足在单一节点上安装的要求](#)，以及在[云供应商上安装单节点 OpenShift 的额外要求](#)。在满足了在单一节点安装要求后，请按照在 [AWS 中安装自定义的集群](#) 来安装集群。在单一节点上安装 OpenShift Container Platform 集群时，[手动安装单节点 OpenShift](#) 部分包含了一个示例 **install-config.yaml** 文件。

### 1.3.2. 在安装程序置备的基础架构上安装集群

您可以使用以下方法之一在 OpenShift Container Platform 安装程序置备的 AWS 基础架构上安装集群：

- [在 AWS 上快速安装集群](#)：您可以在由 OpenShift Container Platform 安装程序置备的 AWS 基础架构上安装 OpenShift Container Platform。您可以使用默认配置选项快速安装集群。
- [在 WS 上安装自定义集群](#)：您可以在安装程序置备的 AWS 基础架构上安装自定义集群。安装程序允许在安装阶段应用一些自定义。其它自定义选项可在[安装后](#)使用。
- [使用自定义网络在 AWS 上安装集群](#)：您可以在安装过程中自定义 OpenShift Container Platform 网络配置，以便集群可以与现有的 IP 地址分配共存,并遵循您的网络要求。
- [在受限网络中的 AWS 上安装集群](#)：您可以使用安装发行内容的内部镜像在 AWS 上安装 OpenShift Container Platform。您可以使用此方法安装不需要活跃互联网连接的集群来获取软件组件。

- **在现有 Virtual Private Cloud 上安装集群**：您可以在现有 AWS Virtual Private Cloud (VPC) 上安装 OpenShift Container Platform。如果您按照公司的说明设置了限制，可以使用这个安装方法，例如在创建新帐户或基础架构时的限制。
- **在现有 VPC 上安装私有集群**：您可以在现有 AWS VPC 上安装私有集群。您可以使用此方法将 OpenShift Container Platform 部署到互联网中不可见的内部网络中。
- **在 WS 上将集群安装到一个政府或机密区域**：OpenShift Container Platform 可以部署到 AWS 区域，这些区域是为需要运行云中敏感工作负载的美国政府机构、州和本地级别的政府机构、企业和其他需要运行敏感工作负载的美国客户准备的。

### 1.3.3. 在用户置备的基础架构上安装集群

您可以使用以下方法之一在您置备的 AWS 基础架构上安装集群：

- **在您提供的 AWS 基础架构上安装集群**：您可以在您提供的 AWS 基础架构上安装 OpenShift Container Platform。您可以使用提供的 CloudFormation 模板来创建 AWS 资源堆栈，这些资源代表 OpenShift Container Platform 安装所需的每个组件。
- **在带用户置备的受限网络中的 AWS 上安装集群**：您可以使用安装发行内容的内部镜像在 AWS 基础架构上安装 OpenShift Container Platform。您可以使用此方法安装不需要活跃互联网连接的集群来获取软件组件。您还可以使用此安装方法确保集群只使用满足您机构对外部内容控制的容器镜像。虽然您可以使用镜像内容安装 OpenShift Container Platform，但您的集群仍需要访问互联网才能使用 AWS API。

## 1.4. 后续步骤

- **配置 AWS 帐户**

## 第 2 章 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。

### 2.1. 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。

#### 2.1.1. 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。

## 2.2. 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</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，worker 机器使用 <b>m6i.large</b> 实例，bootstrap 和 control plane 机器使用 <b>m6i.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>  <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> |
| 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>为额外的机器和 ELB 负载均衡器（它们由集群的使用以及部署的工作负载创建）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。                                                                                                                                                                                                                                                                                                                                                                                         |

## 2.3. IAM 用户所需的 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 2.1. 安装所需的 EC2 权限

- **ec2:AttachNetworkInterface**
- **ec2:AuthorizeSecurityGroupEgress**
- **ec2:AuthorizeSecurityGroupIngress**
- **ec2:CopyImage**
- **ec2>CreateNetworkInterface**
- **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:DescribeInstanceTypes**
- **ec2:DescribeInternetGateways**
- **ec2:DescribeKeyPairs**

- **ec2:DescribeNatGateways**
- **ec2:DescribeNetworkAcls**
- **ec2:DescribeNetworkInterfaces**
- **ec2:DescribePrefixLists**
- **ec2:DescribeRegions**
- **ec2:DescribeRouteTables**
- **ec2:DescribeSecurityGroupRules**
- **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:RevokeSecurityGroupEgress**
- **ec2:RevokeSecurityGroupIngress**
- **ec2:RunInstances**
- **ec2:TerminateInstances**

#### 例 2.2. 安装过程中创建网络资源所需的权限

- **ec2:AllocateAddress**
- **ec2:AssociateAddress**
- **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 Virtual Private Cloud (VPC)，您的帐户不需要这些权限来创建网络资源。

#### 例 2.3. 安装所需的 Elastic Load Balancing 权限(ELB)

- **elasticloadbalancing:AddTags**
- **elasticloadbalancing:ApplySecurityGroupsToLoadBalancer**
- **elasticloadbalancing:AttachLoadBalancerToSubnets**
- **elasticloadbalancing:ConfigureHealthCheck**
- **elasticloadbalancing>CreateListener**
- **elasticloadbalancing>CreateLoadBalancer**
- **elasticloadbalancing>CreateLoadBalancerListeners**
- **elasticloadbalancing>CreateTargetGroup**
- **elasticloadbalancing>DeleteLoadBalancer**
- **elasticloadbalancing:DeregisterInstancesFromLoadBalancer**
- **elasticloadbalancing:DeregisterTargets**
- **elasticloadbalancing:DescribeInstanceHealth**
- **elasticloadbalancing:DescribeListeners**

- **elasticloadbalancing:DescribeLoadBalancerAttributes**
- **elasticloadbalancing:DescribeLoadBalancers**
- **elasticloadbalancing:DescribeTags**
- **elasticloadbalancing:DescribeTargetGroupAttributes**
- **elasticloadbalancing:DescribeTargetHealth**
- **elasticloadbalancing:ModifyLoadBalancerAttributes**
- **elasticloadbalancing:ModifyTargetGroup**
- **elasticloadbalancing:ModifyTargetGroupAttributes**
- **elasticloadbalancing:RegisterInstancesWithLoadBalancer**
- **elasticloadbalancing:RegisterTargets**
- **elasticloadbalancing:SetLoadBalancerPoliciesOfListener**



### 重要

OpenShift Container Platform 使用 ELB 和 ELBv2 API 服务来置备负载均衡器。权限列表显示这两个服务所需的权限。AWS web 控制台中存在一个已知问题，其中这两个服务都使用相同的 **elasticloadbalancing** 操作前缀，但无法识别相同的操作。您可以忽略有关服务没有识别某些 **elasticloadbalancing** 操作的警告。

### 例 2.4. 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 帐户中创建负载均衡器，IAM 用户还需要 **iam:CreateServiceLinkedRole** 权限。

#### 例 2.5. 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 2.6. 安装所需的 Amazon Simple Storage Service (S3) 权限

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

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

#### 例 2.7. 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 2.8. 删除基本集群资源所需的权限

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

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

#### 例 2.9. 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:ReleaseAddress**
- **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。您的帐户只需要有 **tag:UntagResources** 权限就能删除网络资源。

#### 例 2.10. 使用自定义密钥管理服务 (KMS) 密钥安装集群的可选权限

- **kms:CreateGrant**
- **kms:Decrypt**
- **kms:DescribeKey**
- **kms:Encrypt**
- **kms:GenerateDataKey**
- **kms:GenerateDataKeyWithoutPlainText**
- **kms:ListGrants**
- **kms:RevokeGrant**

#### 例 2.11. 使用共享实例角色删除集群所需的权限

- **iam:UntagRole**

#### 例 2.12. 创建清单所需的额外 IAM 和 S3 权限

- **iam:GetUserPolicy**
- **iam:ListAccessKeys**
- **iam:PutUserPolicy**
- **iam:TagUser**
- **s3:AbortMultipartUpload**
- **s3:GetBucketPublicAccessBlock**
- **s3:ListBucket**
- **s3:ListBucketMultipartUploads**
- **s3:PutBucketPublicAccessBlock**
- **s3:PutLifecycleConfiguration**





### 注意

如果您要使用 mint 模式管理云供应商凭证，IAM 用户还需要 The `iam:CreateAccessKey` 和 `iam:CreateUser` 权限。

#### 例 2.13. 实例的可选权限和安装配额检查

- `ec2:DescribeInstanceTypeOfferings`
- `servicequotas:ListAWSDefaultServiceQuotas`

#### 例 2.14. 在共享 VPC 上安装集群时集群所有者帐户的可选权限

- `sts:AssumeRole`

## 2.4. 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。



### 重要

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

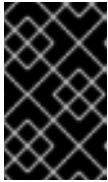
1. 指定 IAM 用户名并选择 **Programmatic access**。
2. 附加 **AdministratorAccess** 策略，以确保帐户有充足的权限来创建集群。此策略让集群能够为每个 OpenShift Container Platform 组件授予凭证。集群只为组件授予它们需要的凭证。



### 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。

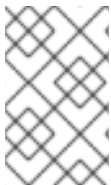


### 重要

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。集群将继续使用您当前的 AWS 凭证为集群的整个生命周期创建 AWS 资源，因此您必须使用基于密钥的长期凭证。

## 2.5. IAM 策略和 AWS 身份验证

默认情况下，安装程序会为集群操作所需的权限为 bootstrap、control plane 和计算实例创建实例配置集。



### 注意

要在单节点 OpenShift 集群中启用从 Amazon Elastic Container Registry (ECR) 拉取镜像作为安装后的任务，您必须将 **AmazonEC2ContainerRegistryReadOnly** 策略添加到与集群的 control plane 角色关联的 IAM 角色中。

但是，您可以创建自己的 IAM 角色，并将其指定为安装过程的一部分。您可能需要指定自己的角色来部署集群或在安装后管理集群。例如：

- 机构的安全策略要求您使用更严格的权限集来安装集群。
- 安装后，集群使用需要访问其他服务的 Operator 配置。

如果选择指定自己的 IAM 角色，您可以执行以下步骤：

- 从默认策略开始，并根据需要进行调整。如需更多信息，请参阅“IAM 实例配置集的默认权限”。
- 使用 AWS Identity and Access Management Access Analyzer (IAM Access Analyzer) 创建一个基于集群的活动的策略模板。如需更多信息，请参阅“使用 AWS IAM 分析器来创建策略模板”。

### 2.5.1. IAM 实例配置集的默认权限

默认情况下，安装程序会为集群操作所需的权限为 bootstrap、control plane 和 worker 实例创建 IAM 实例配置集。

以下列表指定 control plane 和计算机器的默认权限：

#### 例 2.15. control plane 实例配置集的默认 IAM 角色权限

- **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**

#### 例 2.16. 计算实例配置集的默认 IAM 角色权限

- **ec2:DescribeInstances**
- **ec2:DescribeRegions**

### 2.5.2. 指定现有的 IAM 角色

您可以使用 **install-config.yaml** 文件为 control plane 和计算实例指定现有 IAM 角色，而不是让安装程序创建具有默认权限的 IAM 实例配置集。

#### 先决条件

- 您有一个现有的 **install-config.yaml** 文件。

#### 流程

1. 使用计算机器的现有角色更新 **compute.platform.aws.iamRole**。

#### 带有计算实例的 IAM 角色的 **install-config.yaml** 文件示例

```
compute:
  - hyperthreading: Enabled
    name: worker
  platform:
    aws:
      iamRole: ExampleRole
```

2. 使用 control plane 机器的现有角色更新 **controlPlane.platform.aws.iamRole**。

#### 带有 control plane 实例的 IAM 角色的 **install-config.yaml** 文件示例

```
controlPlane:
  hyperthreading: Enabled
  name: master
  platform:
    aws:
      iamRole: ExampleRole
```

3. 保存文件并在安装 OpenShift Container Platform 集群时引用。



#### 注意

要在安装集群后更改或更新 IAM 帐户，请参阅 [RHOC 4 AWS cloud-credentials access key is expired](#) (Red Hat Knowledgebase)。

## 其他资源

- 请参阅[部署集群](#)。

### 2.5.3. 使用 AWS IAM 分析器创建策略模板

control plane 和计算实例配置集需要的最小权限集取决于如何为每日操作配置集群。

要确定集群实例需要哪些权限的一种方法是使用 AWS Identity and Access Management Access Analyzer (IAM Access Analyzer) 创建策略模板：

- 策略模板包含集群在指定时间段内使用的权限。
- 然后，您可以使用模板来创建具有细粒度权限的策略。

## 流程

整个过程可以是：

1. 确保启用了 CloudTrail。CloudTrail 记录 AWS 帐户中的所有操作和事件，包括创建策略模板所需的 API 调用。如需更多信息，请参阅 AWS 文档的[使用 CloudTrail](#)。
2. 为 control plane 实例创建实例配置集，并为计算实例创建一个实例配置集。确保为每个角色分配一个 permissive 策略，如 PowerUserAccess。如需更多信息，请参阅 AWS 文档的[创建实例配置集角色](#)。
3. 在开发环境中安装集群，并根据需要进行配置。务必在生产环境中部署集群托管的所有应用程序。
4. 全面测试集群。测试集群可确保记录所有必需的 API 调用。
5. 使用 IAM Access Analyzer 为每个实例配置集创建策略模板。如需更多信息，请参阅 AWS 文档[基于 CloudTrail 日志生成策略](#)。
6. 创建并为每个实例配置文件添加一个精细的策略。
7. 从每个实例配置集中删除 permissive 策略。
8. 使用现有实例配置集和新策略部署生产集群。



## 注意

您可以在策略中添加 [IAM 条件](#)，使其更严格且符合您的机构安全要求。

## 2.6. 支持的 AWS MARKETPLACE 区域

使用 AWS Marketplace 镜像安装 OpenShift Container Platform 集群可供购买北美提供的客户提供。

虽然优惠必须在北美购买，但您可以将集群部署到以下任何一种支持：

- 公开
- GovCloud



### 注意

AWS secret 区域或中国区域不支持使用 AWS Marketplace 镜像部署 OpenShift Container Platform 集群。

## 2.7. 支持的 AWS 区域

您可以将 OpenShift Container Platform 集群部署到以下区域。



### 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。

### 2.7.1. AWS 公共区域

支持以下 AWS 公共区域：

- **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-south-2** (Hyderabad)
- **ap-southeast-1** (Singapore)
- **ap-southeast-2** (Sydney)
- **ap-southeast-3** (Jakarta)
- **ap-southeast-4** (Melbourne)
- **ca-central-1** (Central)
- **ca-west-1** (Calgary)
- **eu-central-1** (Frankfurt)
- **eu-central-2** (Zurich)
- **eu-north-1** (Stockholm)
- **eu-south-1** (Milan)
- **eu-south-2** (Spain)

- **eu-west-1** (Ireland)
- **eu-west-2** (London)
- **eu-west-3** (Paris)
- **me-central-1** (UAE)
- **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)

### 2.7.2. AWS GovCloud 区域

支持以下 AWS GovCloud 区域：

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

### 2.7.3. AWS SC2S 和 C2S secret 区域

支持以下 AWS secret 区域：

- **us-isob-east-1** Secret Commercial Cloud Services (SC2S)
- **us-iso-east-1** Commercial Cloud Services (C2S)

### 2.7.4. AWS 中国区域

支持以下 AWS 中国区域：

- **cn-north-1** (Beijing)
- **cn-northwest-1** (Ningxia)

## 2.8. 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[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](#)

- [在带有 AWS Outposts 中的远程 worker 的 AWS 上安装集群](#)



## 第 3 章 在 AWS 上快速安装集群

在 OpenShift Container Platform 版本 4.14 中，您可以使用默认配置选项在 Amazon Web Services (AWS) 上安装集群。

### 3.1. 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。
- 您可以阅读[选择集群安装方法并为用户准备它](#)的文档。
- 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。集群将继续使用您当前的 AWS 凭证为集群的整个生命周期创建 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，[将其配置为允许集群需要访问的站点](#)。

### 3.2. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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

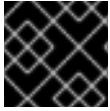
如果您的集群无法直接访问互联网，则可以在置备的某些类型的基础架构上执行受限网络安装。在此过程中，您可以下载所需的内容，并使用它为镜像 registry 填充安装软件包。对于某些安装类型，集群要安装到的环境不需要访问互联网。在更新集群前，您要更新镜像 registry 的内容。

### 3.3. 为集群节点 SSH 访问生成密钥对

在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 **core** 用户的 `~/.ssh/authorized_keys` 列表中，这将启用免密码身份验证。

将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。

如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。`./openshift-install gather` 命令还需要在集群节点上设置 SSH 公钥。

**重要**

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。

**注意**

您必须使用本地密钥，而不是使用特定平台方法配置 [的密钥](#)，如 [AWS 密钥对](#)。

**流程**

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，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_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。

**注意**

如果您计划在 **x86\_64**、**ppc64le** 和 **s390x** 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 **ed25519** 算法的密钥。相反，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

3. 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。

**注意**

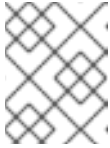
在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- a. 如果 **ssh-agent** 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

**输出示例**

```
Agent pid 31874
```



### 注意

如果集群处于 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。

4. 将 SSH 私钥添加到 **ssh-agent**：

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

- 1 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

### 输出示例

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

### 后续步骤

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

## 3.4. 获取安装程序

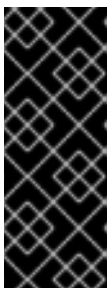
在安装 OpenShift Container Platform 前，将安装文件下载到您用于安装的主机上。

### 先决条件

- 您有一台运行 Linux 或 macOS 的计算机，至少有 1.2 GB 本地磁盘空间。

### 流程

1. 进入 Red Hat Hybrid Cloud Console 上的 [Cluster Type](#) 页。如果您有红帽帐户，请使用您的凭证登录。如果没有，请创建一个帐户。
2. 在页的 **Run it yourself** 部分中选择您的基础架构供应商。
3. 从 **OpenShift 安装程序** 下的下拉菜单中选择您的主机操作系统和架构，然后点 **下载安装程序**。
4. 将下载的文件保存在要存储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几个文件。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。删除集群需要这两个文件。
- 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，请为特定云供应商完成 OpenShift Container Platform 卸载流程。

5. 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
```

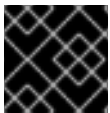
6. 从 [Red Hat OpenShift Cluster Manager](#) 下载安装 pull secret。此 pull secret 允许您与所含授权机构提供的服务进行身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

## 提示

另外，您还可以从[红帽客户门户网站](#)检索安装程序，您可以在其中指定要下载的安装程序版本。但是，您需要有一个有效的订阅才能访问此页。

## 3.5. 部署集群

您可以在兼容云平台上安装 OpenShift Container Platform。



### 重要

在初始安装过程中，您只能运行安装程序的 **create cluster** 命令一次。

## 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。
- 已确认主机上的云供应商帐户具有部署集群的正确权限。权限不正确的帐户会导致安装过程失败，并显示包括缺失权限的错误消息。

## 流程

1. 进入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>**，请指定要存储安装程序创建的文件的目录名称。

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

在指定目录时：

- 验证该目录是否具有**执行**权限。在安装目录中运行 Terraform 二进制文件需要这个权限。
  - 使用空目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。
2. 在提示符处提供值：
    - a. 可选：选择用于访问集群机器的 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. 为集群输入描述性名称。
  - g. 粘贴 [Red Hat OpenShift Cluster Manager](#) 中的 **pull secret**。
3. 可选：从您用来安装集群的 IAM 帐户删除或禁用 **AdministratorAccess** 策略。

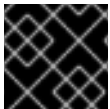
**注意**

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

**验证**

当集群部署成功完成时：

- 终端会显示用于访问集群的说明，包括指向 Web 控制台和 **kubeadmin** 用户的凭证的链接。
- 凭证信息还会输出到 **<installation\_directory>/openshift\_install.log**。

**重要**

不要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。

**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: "password"
INFO Time elapsed: 36m22s
```



## 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。一个例外是，您必须手动批准待处理的 **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。

## 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](#)。

## 3.6. 通过下载二进制文件安装 OPENSIFT CLI

您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

### 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. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。
5. 解包存档：

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

6. 将 **oc** 二进制文件放到 **PATH** 中的目录中。  
要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 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**)二进制文件。

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
4. 使用 ZIP 程序解压存档。
5. 将 **oc** 二进制文件移到 **PATH** 中的目录中。  
要查看您的 **PATH**，请打开命令提示并执行以下命令：

```
C:\> path
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 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**)二进制文件。

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。



#### 注意

对于 macOS arm64，请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

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**，请打开终端并执行以下命令：

```
$ echo $PATH
```

#### 验证



- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

### 3.7. 使用 CLI 登录集群

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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

#### 先决条件

- 已部署 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. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

#### 输出示例

```
system:admin
```

```
/validating-an-installation.adoc
```

### 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

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 控制台的更多详情](#)，请参阅 [访问 Web 控制台](#)。

## 3.9. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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](#)

## 3.10. 后续步骤

- [验证安装](#)
- [自定义集群](#)
- [远程健康报告](#)
- [删除云供应商凭证](#)

## 第 4 章 使用自定义在 AWS 上安装集群

在 OpenShift Container Platform 版本 4.14 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。



### 注意

OpenShift Container Platform 安装配置的作用范围被特意设计为较小。它旨在简化操作并确保成功。在安装完成后，您可以进行更多的 OpenShift Container Platform 配置任务。

### 4.1. 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。
- 您可以阅读[选择集群安装方法并为用户准备它](#)的文档。
- 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，[将其配置为允许集群需要访问的站点](#)。

### 4.2. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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型的基础架构上执行受限网络安装。在此过程中，您可以下载所需的内容，并使用它为镜像 registry 填充安装软件包。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。在更新集群前，您要更新镜像 registry 的内容。

### 4.3. 为集群节点 SSH 访问生成密钥对

在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 `core` 用户的 `~/.ssh/authorized_keys` 列表中，这将启用免密码身份验证。

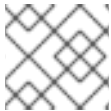
将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。

如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。`./openshift-install gather` 命令还需要在集群节点上设置 SSH 公钥。



### 重要

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。



### 注意

您必须使用本地密钥，而不是使用特定平台方法配置 [的密钥](#)，如 [AWS 密钥对](#)。

## 流程

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，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_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。



### 注意

如果您计划在 **x86\_64**、**ppc64le** 和 **s390x** 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 **ed25519** 算法的密钥。相反，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

3. 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。



### 注意

在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- a. 如果 **ssh-agent** 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

### 输出示例

```
Agent pid 31874
```



### 注意

如果集群处于 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。

4. 将 SSH 私钥添加到 **ssh-agent**：

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

- 1 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

### 输出示例

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

### 后续步骤

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

## 4.4. 获取 AWS MARKETPLACE 镜像

如果要使用 AWS Marketplace 镜像部署 OpenShift Container Platform 集群，您必须首先通过 AWS 订阅。订阅提供的提供的 AMI ID 可让您使用安装程序用来部署 worker 节点的 AMI ID。



### 注意

您应该只修改计算机器的 RHCOS 镜像以使用 AWS Marketplace 镜像。control plane 机器和基础架构节点不需要 OpenShift Container Platform 订阅，且默认使用公共 RHCOS 默认镜像，这不会对您的 AWS 账单造成订阅成本。因此，您不应该修改集群默认引导镜像或 control plane 引导镜像。将 AWS Marketplace 镜像应用到它们时，会生成无法恢复的额外许可成本。

### 先决条件

- 有 AWS 账户购买的产品。此帐户不必与用于安装集群的帐户相同。

### 流程

1. 从 [AWS Marketplace](#) 完成 OpenShift Container Platform 订阅。
2. 记录特定区域的 AMI ID。作为安装过程的一部分，您必须在部署集群前使用这个值更新 **install-config.yaml** 文件。

### 使用 AWS Marketplace worker 节点的 install-config.yaml 文件示例

```

apiVersion: v1
baseDomain: example.com
compute:
- hyperthreading: Enabled
  name: worker
  platform:
    aws:
      amiID: ami-06c4d345f7c207239 1
      type: m5.4xlarge
    replicas: 3
metadata:
  name: test-cluster
platform:
  aws:
    region: us-east-2 2
sshKey: ssh-ed25519 AAAA...
pullSecret: '{"auths": ...}'

```

**1** 来自 AWS Marketplace 订阅的 AMI ID。

**2** 您的 AMI ID 与特定的 AWS 区域相关联。在创建安装配置文件时，请确保选择配置订阅时指定的同一 AWS 区域。

## 4.5. 获取安装程序

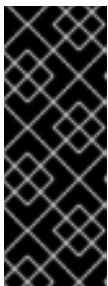
在安装 OpenShift Container Platform 前，将安装文件下载到您用于安装的主机上。

### 先决条件

- 您有一台运行 Linux 或 macOS 的计算机，至少有 1.2 GB 本地磁盘空间。

### 流程

1. 进入 Red Hat Hybrid Cloud Console 上的 [Cluster Type](#) 页。如果您有红帽帐户，请使用您的凭证登录。如果没有，请创建一个帐户。
2. 在页的 **Run it yourself** 部分中选择您的基础架构供应商。
3. 从 **OpenShift 安装程序** 下的下拉菜单中选择您的主机操作系统和架构，然后点**下载安装程序**。
4. 将下载的文件保存在要存储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几个文件。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。删除集群需要这两个文件。
- 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，请为特定云供应商完成 OpenShift Container Platform 卸载流程。

5. 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
```

6. 从 [Red Hat OpenShift Cluster Manager](#) 下载安装 [pull secret](#)。此 pull secret 允许您与所含授权机构提供的服务进行身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

## 提示

另外，您还可以从[红帽客户门户网站](#)检索安装程序，您可以在其中指定要下载的安装程序版本。但是，您需要有一个有效的订阅才能访问此页。

## 4.6. 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

- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。

### 流程

1. 创建 **install-config.yaml** 文件。
  - a. 进入包含安装程序的目录并运行以下命令：

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

- 1** 对于 **<installation\_directory>**，请指定要存储安装程序创建的文件的目录名称。

在指定目录时：

- 验证该目录是否具有执行权限。在安装目录中运行 Terraform 二进制文件需要这个权限。
- 使用空目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。



### 注意

始终删除 **~/.powervs** 目录，以避免重复使用过时的配置。运行以下命令：

```
$ rm -rf ~/.powervs
```

- b. 在提示符处，提供云的配置详情：
  - i. 可选：选择用于访问集群机器的 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. 为集群输入描述性名称。
2. 修改 **install-config.yaml** 文件。您可以在"安装配置参数"部分找到有关可用参数的更多信息。



注意

如果要安装三节点集群，请确保将 **compute.replicas** 参数设置为 **0**。这样可确保集群的 control plane 可以调度。如需更多信息，请参阅"在 AWS 上安装三节点集群"。

3. 备份 **install-config.yaml** 文件，以便您可以使用它安装多个集群。



重要

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

其他资源

- [AWS 的安装配置参数](#)

4.6.1. 集群安装的最低资源要求

每台集群机器都必须满足以下最低要求：

表 4.1. 最低资源要求

| 机器                   | 操作系统  | vCPU [1] | 虚拟内存  | Storage | 每秒输入/输出 (IOPS) [2] |
|----------------------|-------|----------|-------|---------|--------------------|
| bootstrap            | RHCOS | 4        | 16 GB | 100 GB  | 300                |
| Control plane (控制平面) | RHCOS | 4        | 16 GB | 100 GB  | 300                |



| 机器      | 操作系统                             | vCPU [1] | 虚拟内存 | Storage | 每秒输入/输出 (IOPS) [2] |
|---------|----------------------------------|----------|------|---------|--------------------|
| Compute | RHCOS、<br>RHEL 8.6 及更<br>新版本 [3] | 2        | 8 GB | 100 GB  | 300                |

1. 当未启用并发多线程 (SMT) 或超线程时，一个 vCPU 相当于一个物理内核。启用后，使用以下公式来计算对应的比例：（每个内核数的线程）× sockets = vCPU。
2. OpenShift Container Platform 和 Kubernetes 对磁盘性能非常敏感，建议使用更快的存储速度，特别是 control plane 节点上需要 10 ms p99 fsync 持续时间的 etcd。请注意，在许多云平台上，存储大小和 IOPS 可一起扩展，因此您可能需要过度分配存储卷来获取足够的性能。
3. 与所有用户置备的安装一样，如果您选择在集群中使用 RHEL 计算机器，则负责所有操作系统生命周期管理和维护，包括执行系统更新、应用补丁和完成所有其他必要的任务。RHEL 7 计算机器的使用已弃用，并已在 OpenShift Container Platform 4.10 及更新的版本中删除。



注意

从 OpenShift Container Platform 版本 4.13 开始，RHCOS 基于 RHEL 版本 9.2，它更新了微架构要求。以下列表包含每个架构需要的最小指令集架构 (ISA)：

- x86-64 体系结构需要 x86-64-v2 ISA
- ARM64 架构需要 ARMv8.0-A ISA
- IBM Power 架构需要 Power 9 ISA
- s390x 架构需要 z14 ISA

如需更多信息，请参阅 [RHEL 架构](#)。

如果平台的实例类型满足集群机器的最低要求，则 OpenShift Container Platform 支持使用它。

其他资源

- [优化存储](#)

4.6.2. 为 AWS 测试的实例类型

以下 Amazon Web Services(AWS) 实例类型已经过 OpenShift Container Platform 测试。



注意

将以下图中包含的机器类型用于 AWS 实例。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装"最小资源要求"中列出的最少资源要求匹配。

例 4.1. 基于 64 位 x86 架构的机器类型

- c4.\*



- c5.\*
- c5a.\*
- i3.\*
- m4.\*
- m5.\*
- m5a.\*
- m6i.\*
- r4.\*
- r5.\*
- r5a.\*
- r6i.\*
- t3.\*
- t3a.\*

#### 4.6.3. 在 64 位 ARM 基础架构上为 AWS 测试过的实例类型

OpenShift Container Platform 中已经测试了以下 Amazon Web Services (AWS) 64 位 ARM 实例类型。



##### 注意

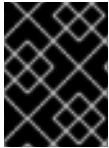
使用 AWS ARM 实例的以下图中包含的机器类型。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 4.2. 基于 64 位 ARM 架构的机器类型

- c6g.\*
- m6g.\*
- r8g.\*

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

您可以自定义安装配置文件 (**install-config.yaml**)，以指定有关 OpenShift Container Platform 集群平台的更多详细信息，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，并进行修改。

```

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
      metadataService:
        authentication: Optional 7
        type: m6i.xlarge
      replicas: 3
compute: 8
- hyperthreading: Enabled 9
  name: worker
  platform:
    aws:
      rootVolume:
        iops: 2000
        size: 500
        type: io1 10
      metadataService:
        authentication: Optional 11
        type: c5.4xlarge
      zones:
        - us-west-2c
      replicas: 3
metadata:
  name: test-cluster 12
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OVNKubernetes 13
  serviceNetwork:
    - 172.30.0.0/16
platform:
  aws:
    region: us-west-2 14

```

```

propagateUserTags: true 15
userTags:
  adminContact: jdoe
  costCenter: 7536
amiID: ami-0c5d3e03c0ab9b19a 16
serviceEndpoints: 17
  - name: ec2
    url: https://vpce-id.ec2.us-west-2.vpce.amazonaws.com
fips: false 18
sshKey: ssh-ed25519 AAAA... 19
pullSecret: '{"auths": ...}' 20

```

1 12 14 20 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。默认情况下，CCO 使用 **kube-system** 命名空间中的 root 凭证来动态尝试决定凭证的功能。有关 CCO 模式的详情，请参阅 [身份验证和授权指南](#) 中的 "About the Cloud Credential Operator" 部分。

3 8 15 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。仅使用一个 control plane 池。

5 9 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置为 **Disabled** 来禁用它。如果在某些集群机器中禁用并发多线程，则必须在所有集群机器中禁用它。



### 重要

如果您禁用并发多线程，请确保您的容量规划考虑机器性能显著降低的情况。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 10 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**。

7 11 是否需要 [Amazon EC2 实例元数据服务 v2](#) (IMDSv2)。为了要求 IMDSv2，请将参数值设置为 **Required**。要允许使用 IMDSv1 和 IMDSv2，请将参数值设置为 **Optional**。如果没有指定值，则允许 IMDSv1 和 IMDSv2。



### 注意

在集群安装过程中设置的 control plane 机器的 IMDS 配置只能使用 AWS CLI 更改。可以使用计算机器集来更改计算机器的 IMDS 配置。

13 要安装的集群网络插件。支持的值有 **OVNKubernetes** 和 **OpenShiftSDN**。默认值为 **OVNKubernetes**。

16 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。

17 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。

- 18** 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



### 重要

要为集群启用 FIPS 模式，您必须从配置为以 FIPS 模式操作的 Red Hat Enterprise Linux (RHEL) 计算机运行安装程序。有关在 RHEL 中配置 FIPS 模式的更多信息，请参阅[在 FIPS 模式中安装该系统](#)。当以 FIPS 模式运行 Red Hat Enterprise Linux (RHEL) 或 Red Hat Enterprise Linux CoreOS (RHCOS) 时，OpenShift Container Platform 核心组件使用 RHEL 加密库，在 x86\_64、ppc64le 和 s390x 架构上提交到 NIST FIPS 140-2/140-3 Validation。

- 19** 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。

## 4.6.5. 在安装过程中配置集群范围的代理

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，并确定它们中的任何站点是否需要绕过代理。默认情况下，所有集群出口流量都经过代理，包括对托管云供应商 API 的调用。如果需要，您将在 **Proxy** 对象的 **spec.noProxy** 字段中添加站点来绕过代理。



### 注意

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

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

### 流程

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
```

```
noProxy: ec2.<aws_region>.amazonaws.com,elasticloadbalancing.
<aws_region>.amazonaws.com,s3.<aws_region>.amazonaws.com ❸
additionalTrustBundle: | ❹
-----BEGIN CERTIFICATE-----
<MY_TRUSTED_CA_CERT>
-----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> ❺
```

- ❶ 用于创建集群外 HTTP 连接的代理 URL。URL 方案必须是 **http**。
- ❷ 用于创建集群外 HTTPS 连接的代理 URL。
- ❸ 要从代理中排除的目标域名、IP 地址或其他网络 CIDR 的逗号分隔列表。在域前面加上 **.** 以仅匹配子域。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。如果您已将 Amazon **EC2**、**Elastic Load Balancing** 和 **S3** VPC 端点添加到 VPC 中，您必须将这些端点添加到 **noProxy** 字段。
- ❹ 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，其包含代理 HTTPS 连接所需的一个或多个额外 CA 证书。然后，Cluster Network Operator 会创建 **trusted-ca-bundle** 配置映射，将这些内容与 Red Hat Enterprise Linux CoreOS (RHCOS) 信任捆绑包合并，**Proxy** 对象的 **trustedCA** 字段中也会引用此配置映射。**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。
- ❺ 可选：决定 **Proxy** 对象的配置以引用 **trustedCA** 字段中 **user-ca-bundle** 配置映射的策略。允许的值是 **Proxyonly** 和 **Always**。仅在配置了 **http/https** 代理时，使用 **Proxyonly** 引用 **user-ca-bundle** 配置映射。使用 **Always** 始终引用 **user-ca-bundle** 配置映射。默认值为 **Proxyonly**。



#### 注意

安装程序不支持代理的 **readinessEndpoints** 字段。



#### 注意

如果安装程序超时，重启并使用安装程序的 **wait-for** 命令完成部署。例如：

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

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**。

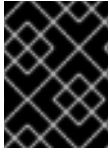


#### 注意

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

## 4.7. 通过下载二进制文件安装 OPENSIFT CLI

您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

## 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. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。
5. 解包存档：

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

6. 将 **oc** 二进制文件放到 **PATH** 中的目录中。  
要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

## 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**)二进制文件。

### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
4. 使用 ZIP 程序解压存档。
5. 将 **oc** 二进制文件移到 **PATH** 中的目录中。  
要查看您的 **PATH**，请打开命令提示并执行以下命令：

```
C:\> path
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 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**)二进制文件。

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。



#### 注意

对于 macOS arm64，请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

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**，请打开终端并执行以下命令：

```
$ echo $PATH
```

#### 验证

- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

## 4.8. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案

默认情况下，管理员 secret 存储在 **kube-system** 项目中。如果您在 **install-config.yaml** 文件中将 **credentialsMode** 参数配置为 **Manual**，则必须使用以下替代方案之一：

- 要手动管理长期云凭证，请按照[手动创建长期凭证](#)中的步骤操作。
- 要实现在集群外为各个组件管理的短期凭证，请按照[配置 AWS 集群以使用短期凭证](#)中的步骤操作。

### 4.8.1. 手动创建长期凭证

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

- 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

- 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

- 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/ {print $3}')
```

- 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 自定义资源 (CR) 列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \ ❶
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \ ❷
  --to=<path_to_directory_for_credentials_requests> \ ❸
```

❶ **--included** 参数仅包含特定集群配置所需的清单。

❷ 指定 **install-config.yaml** 文件的位置。

❸ 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

此命令为每个 **CredentialsRequest** 对象创建一个 YAML 文件。

### CredentialsRequest 对象示例

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
...
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
```



```

statementEntries:
- effect: Allow
  action:
  - iam:GetUser
  - iam:GetUserPolicy
  - iam:ListAccessKeys
  resource: "*"
...

```

5. 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。

### 带有 secret 的 CredentialsRequest 对象示例

```

apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
...
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
    - effect: Allow
      action:
      - s3:CreateBucket
      - s3:DeleteBucket
      resource: "*"
    ...
  secretRef:
    name: <component_secret>
    namespace: <component_namespace>
...

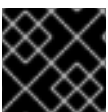
```

### Secret 对象示例

```

apiVersion: v1
kind: Secret
metadata:
  name: <component_secret>
  namespace: <component_namespace>
data:
  aws_access_key_id: <base64_encoded_aws_access_key_id>
  aws_secret_access_key: <base64_encoded_aws_secret_access_key>

```



#### 重要

在升级使用手动维护凭证的集群前，您必须确保 CCO 处于可升级状态。

## 4.8.2. 将 AWS 集群配置为使用短期凭证

要安装配置为使用 AWS 安全令牌服务 (STS) 的集群，您必须配置 CCO 实用程序并为集群创建所需的 AWS 资源。

#### 4.8.2.1. 配置 Cloud Credential Operator 工具

当 Cloud Credential Operator(CCO)以手动模式运行时，要从集群外部创建和管理云凭证，提取并准备 CCO 实用程序(**ccctl**)二进制文件。



##### 注意

**ccctl** 工具是在 Linux 环境中运行的 Linux 二进制文件。

##### 先决条件

- 您可以访问具有集群管理员权限的 OpenShift Container Platform 帐户。
- 已安装 OpenShift CLI(**oc**)。
- 您已为 **ccctl** 工具创建了用于以下权限的 AWS 帐户：

##### 例 4.3. 所需的 AWS 权限

###### 所需的 iam 权限

- **iam:CreateOpenIDConnectProvider**
- **iam:CreateRole**
- **iam>DeleteOpenIDConnectProvider**
- **iam>DeleteRole**
- **iam>DeleteRolePolicy**
- **iam:GetOpenIDConnectProvider**
- **iam:GetRole**
- **iam:GetUser**
- **iam:ListOpenIDConnectProviders**
- **iam:ListRolePolicies**
- **iam:ListRoles**
- **iam:PutRolePolicy**
- **iam:TagOpenIDConnectProvider**
- **iam:TagRole**

###### 所需的 s3 权限

- **s3:CreateBucket**

- **s3:DeleteBucket**
- **s3:DeleteObject**
- **s3:GetBucketAcl**
- **s3:GetBucketTagging**
- **s3:GetObject**
- **s3:GetObjectAcl**
- **s3:GetObjectTagging**
- **s3:ListBucket**
- **s3:PutBucketAcl**
- **s3:PutBucketPolicy**
- **s3:PutBucketPublicAccessBlock**
- **s3:PutBucketTagging**
- **s3:PutObject**
- **s3:PutObjectAcl**
- **s3:PutObjectTagging**

所需的 cloudfront 权限

- **cloudfront:ListCloudFrontOriginAccessIdentities**
- **cloudfront:ListDistributions**
- **cloudfront:ListTagsForResource**

如果您计划通过公共 CloudFront 发行版 URL 将 OIDC 配置存储在 IAM 身份提供程序访问的私有 S3 存储桶中，则运行 **ccctl** 工具的 AWS 帐户需要以下额外权限：

#### 例 4.4. 使用 CloudFront 私有 S3 存储桶的额外权限

- **cloudfront:CreateCloudFrontOriginAccessIdentity**
- **cloudfront:CreateDistribution**
- **cloudfront>DeleteCloudFrontOriginAccessIdentity**
- **cloudfront>DeleteDistribution**
- **cloudfront:GetCloudFrontOriginAccessIdentity**
- **cloudfront:GetCloudFrontOriginAccessIdentityConfig**

- **cloudfront:GetDistribution**
- **cloudfront:TagResource**
- **cloudfront:UpdateDistribution**



### 注意

在使用 **ccoctl aws create-all** 命令处理凭证请求时，这些额外权限支持使用 **--create-private-s3-bucket** 选项。

## 流程

1. 运行以下命令，为 OpenShift Container Platform 发行镜像设置变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像获取 CCO 容器镜像：

```
$ CCO_IMAGE=$(oc adm release info --image-for='cloud-credential-operator'
$RELEASE_IMAGE -a ~/.pull-secret)
```



### 注意

确保 **\$RELEASE\_IMAGE** 的架构与将使用 **ccoctl** 工具的环境架构相匹配。

3. 运行以下命令，将 CCO 容器镜像中的 **ccoctl** 二进制文件提取到 OpenShift Container Platform 发行镜像中：

```
$ oc image extract $CCO_IMAGE --file="/usr/bin/ccoctl" -a ~/.pull-secret
```

4. 运行以下命令更改权限以使 **ccoctl** 可执行：

```
$ chmod 775 ccoctl
```

## 验证

- 要验证 **ccoctl** 是否准备就绪，可以尝试显示帮助文件。运行命令时使用相对文件名，例如：

```
$ ./ccoctl.rhel9
```

## 输出示例

```
OpenShift credentials provisioning tool
```

```
Usage:
ccoctl [command]
```

```
Available Commands:
alibabacloud Manage credentials objects for alibaba cloud
```

```
aws      Manage credentials objects for AWS cloud
azure    Manage credentials objects for Azure
gcp      Manage credentials objects for Google cloud
help     Help about any command
ibmcloud Manage credentials objects for IBM Cloud
nutanix  Manage credentials objects for Nutanix
```

**Flags:**

```
-h, --help  help for ccoctl
```

Use "ccoctl [command] --help" for more information about a command.

#### 4.8.2.2. 使用 Cloud Credential Operator 实用程序创建 AWS 资源

创建 AWS 资源时有以下选项：

- 您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。这是创建资源的最快速方法。请参阅 [使用单个命令创建 AWS 资源](#)。
- 如果您需要在修改 AWS 资源前查看 **ccoctl** 工具创建的 JSON 文件，或者 **ccoctl** 工具用于创建 AWS 资源的过程无法自动满足组织的要求，您可以单独创建 AWS 资源。请参阅 [单独创建 AWS 资源](#)。

##### 4.8.2.2.1. 使用单个命令创建 AWS 资源

如果 **ccoctl** 工具用于创建 AWS 资源的过程自动满足机构的要求，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。

否则，您可以单独创建 AWS 资源。如需更多信息，请参阅“单独创建 AWS 资源”。

**注意**

默认情况下，**ccoctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 `<path_to_ccoctl_output_dir>` 来引用这个目录。

**先决条件**

您必须：

- 提取并准备好 **ccoctl** 二进制文件。

**流程**

1. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
```

```
--included \ ❶
--install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \ ❷
--to=<path_to_directory_for_credentials_requests> \ ❸
```

- ❶ **--included** 参数仅包含特定集群配置所需的清单。
- ❷ 指定 **install-config.yaml** 文件的位置。
- ❸ 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。



### 注意

此命令可能需要一些时间才能运行。

3. 运行以下命令，使用 **ccctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccctl aws create-all \
--name=<name> \ ❶
--region=<aws_region> \ ❷
--credentials-requests-dir=<path_to_credentials_requests_directory> \ ❸
--output-dir=<path_to_ccctl_output_dir> \ ❹
--create-private-s3-bucket \ ❺
```

- ❶ 指定用于标记创建用于跟踪的任何云资源的名称。
- ❷ 指定在其中创建云资源的 AWS 区域。
- ❸ 指定包含组件 **CredentialsRequest** 对象文件的目录。
- ❹ 可选：指定您希望 **ccctl** 实用程序在其中创建对象的目录。默认情况下，实用程序在运行命令的目录中创建对象。
- ❺ 可选：默认情况下，**ccctl** 实用程序将 OpenID Connect (OIDC) 配置文件存储在公共 S3 存储桶中，并使用 S3 URL 作为公共 OIDC 端点。要将 OIDC 配置存储在 IAM 身份提供程序通过公共 CloudFront 发行版 URL 访问的专用 S3 存储桶中，请使用 **--create-private-s3-bucket** 参数。



### 注意

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

## 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccctl_output_dir>/manifests
```

## 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

#### 4.8.2.2.2. 单独创建 AWS 资源

您可以使用 **ccctl** 工具单独创建 AWS 资源。这个选项对于在不同用户或部门之间创建这些资源的组织可能很有用。

否则，您可以使用 **ccctl aws create-all** 命令自动创建 AWS 资源。如需更多信息，请参阅“使用单个命令创建 AWS 资源”。



#### 注意

默认情况下，**ccctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccctl\_output\_dir>** 来引用这个目录。

有些 **ccctl** 命令会发出 AWS API 调用来创建或修改 AWS 资源。您可以使用 **--dry-run** 标志来避免 API 调用。使用此标志可在本地文件系统中创建 JSON 文件。您可以使用 **--cli-input-json** 参数查看和修改 JSON 文件，然后使用 AWS CLI 工具应用它们。

#### 先决条件

- 提取并准备 **ccctl** 二进制文件。

#### 流程

- 运行以下命令，生成用于为集群设置 OpenID Connect 供应商的公共和私有 RSA 密钥文件：

```
$ ccctl aws create-key-pair
```

#### 输出示例

```
2021/04/13 11:01:02 Generating RSA keypair
2021/04/13 11:01:03 Writing private key to /<path_to_ccctl_output_dir>/serviceaccount-signer.private
2021/04/13 11:01:03 Writing public key to /<path_to_ccctl_output_dir>/serviceaccount-signer.public
2021/04/13 11:01:03 Copying signing key for use by installer
```

其中 **serviceaccount-signer.private** 和 **serviceaccount-signer.public** 是生成的密钥文件。

此命令还会在 **/<path\_to\_ccctl\_output\_dir>/tls/bound-service-account-signing-key.key** 中创建集群在安装过程中所需的私钥。

2. 运行以下命令，在 AWS 上创建 OpenID Connect 身份提供程序和 S3 存储桶：

```
$ ccoctl aws create-identity-provider \
  --name=<name> \ ❶
  --region=<aws_region> \ ❷
  --public-key-file=<path_to_ccoctl_output_dir>/serviceaccount-signer.public ❸
```

- ❶ **<name>** 是用于标记为跟踪而创建的云资源的名称。
- ❷ **<aws-region>** 是将在其中创建云资源的 AWS 区域。
- ❸ **<path\_to\_ccoctl\_output\_dir>** 是 **ccoctl aws create-key-pair** 命令生成的公钥文件的路径。

### 输出示例

```
2021/04/13 11:16:09 Bucket <name>-oidc created
2021/04/13 11:16:10 OpenID Connect discovery document in the S3 bucket <name>-oidc at
.well-known/openid-configuration updated
2021/04/13 11:16:10 Reading public key
2021/04/13 11:16:10 JSON web key set (JWKS) in the S3 bucket <name>-oidc at keys.json
updated
2021/04/13 11:16:18 Identity Provider created with ARN: arn:aws:iam::
<aws_account_id>:oidc-provider/<name>-oidc.s3.<aws_region>.amazonaws.com
```

其中 **openid-configuration** 是发现文档和 **key.json** 是一个 JSON Web 密钥集文件。

此命令还会在 **/<path\_to\_ccoctl\_output\_dir>/manifests/cluster-authentication-02-config.yaml** 中创建 YAML 配置文件。此文件为集群生成的服务帐户令牌设置签发者 URL 字段，以便 AWS IAM 身份提供程序信任令牌。

3. 为集群中的每个组件创建 IAM 角色：

- a. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

- b. 从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \ ❶
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
  ❷
  --to=<path_to_directory_for_credentials_requests> ❸
```

- ❶ **--included** 参数仅包含特定集群配置所需的清单。
- ❷ 指定 **install-config.yaml** 文件的位置。
- ❸ 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。



c. 运行以下命令，使用 **ccctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccctl aws create-iam-roles \
  --name=<name> \
  --region=<aws_region> \
  --credentials-requests-dir=<path_to_credentials_requests_directory> \
  --identity-provider-arn=arn:aws:iam::<aws_account_id>:oidc-provider/<name>-oidc.s3.
  <aws_region>.amazonaws.com
```



### 注意

对于使用其他 IAM API 端点的 AWS 环境（如 GovCloud），还必须使用 **--region** 参数指定您的区域。

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

对于每个 **CredentialsRequest** 对象，**ccctl** 创建一个带有信任策略的 IAM 角色，该角色与指定的 OIDC 身份提供程序相关联，以及来自 OpenShift Container Platform 发行镜像的每个 **CredentialsRequest** 对象中定义的权限策略。

## 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccctl_output_dir>/manifests
```

## 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

### 4.8.2.3. 整合 Cloud Credential Operator 实用程序清单

要为单个组件在集群外实现短期安全凭证，您必须将创建 Cloud Credential Operator 实用程序 (**ccctl**) 的清单文件移到安装程序的正确目录中。

## 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您已配置了 Cloud Credential Operator 实用程序 (**ccctl**)。

- 已使用 **ccoctl** 工具创建了集群所需的云供应商资源。

## 流程

1. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，将 **ccoctl** 工具生成的清单复制到安装程序创建的 **manifests** 目录中：

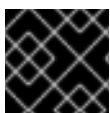
```
$ cp /<path_to_ccoctl_output_dir>/manifests/* ./manifests/
```

4. 将包含私钥的 **tls** 目录复制到安装目录中：

```
$ cp -a /<path_to_ccoctl_output_dir>/tls .
```

## 4.9. 部署集群

您可以在兼容云平台上安装 OpenShift Container Platform。



### 重要

在初始安装过程中，您只能运行安装程序的 **create cluster** 命令一次。

### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。
- 已确认主机上的云供应商帐户具有部署集群的正确权限。权限不正确的帐户会导致安装过程失败，并显示包括缺失权限的错误消息。

## 流程

1. 进入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`，而不是 `info`。

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



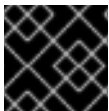
### 注意

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

## 验证

当集群部署成功完成时：

- 终端会显示用于访问集群的说明，包括指向 Web 控制台和 **kubeadmin** 用户的凭证的链接。
- 凭证信息还会输出到 `<installation_directory>/openshift_install.log`。



### 重要

不要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。

## 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: "password"
INFO Time elapsed: 36m22s
```



### 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。一个例外是，您必须手动批准待处理的 **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。

## 4.10. 使用 CLI 登录集群

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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

### 先决条件

- 已部署 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. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

## 输出示例

```
system:admin
```

/validating-an-installation.adoc

## 4.11. 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 控制台的更多详情](#)，请参阅 [访问 Web 控制台](#)。

## 4.12. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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](#)。

## 4.13. 后续步骤

- [验证安装](#)
- [自定义集群](#)
- [远程健康报告](#)
- [删除云供应商凭证](#)

## 第 5 章 使用自定义网络在 AWS 上安装集群

在 OpenShift Container Platform 版本 4.14 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。

### 5.1. 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。
- 您可以阅读[选择集群安装方法并为用户准备它的文档](#)。
- 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。集群将继续使用您当前的 AWS 凭证为集群的整个生命周期创建 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，[将其配置为允许集群需要访问的站点](#)。

### 5.2. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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型的基础架构上执行受限网络安装。在此过程中，您可以下载所需的内容，并使用它为镜像 registry 填充安装软件包。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。在更新集群前，您要更新镜像 registry 的内容。

### 5.3. 为集群节点 SSH 访问生成密钥对

在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 **core** 用户的 `~/.ssh/authorized_keys` 列表中，这将启用免密码身份验证。

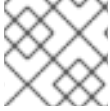
将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。

如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。`./openshift-install gather` 命令还需要在集群节点上设置 SSH 公钥。



### 重要

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。



### 注意

您必须使用本地密钥，而不是使用特定平台方法配置 [的密钥](#)，如 [AWS 密钥对](#)。

## 流程

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，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_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。



### 注意

如果您计划在 **x86\_64**、**ppc64le** 和 **s390x** 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 **ed25519** 算法的密钥。相反，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

3. 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。



### 注意

在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- a. 如果 **ssh-agent** 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

## 输出示例

Agent pid 31874



### 注意

如果集群处于 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。

4. 将 SSH 私钥添加到 **ssh-agent**：

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

- 1 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

### 输出示例

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

### 后续步骤

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

## 5.4. 获取安装程序

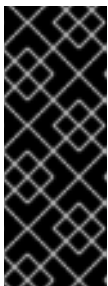
在安装 OpenShift Container Platform 前，将安装文件下载到您用于安装的主机上。

### 先决条件

- 您有一台运行 Linux 或 macOS 的计算机，至少有 1.2 GB 本地磁盘空间。

### 流程

1. 进入 Red Hat Hybrid Cloud Console 上的 [Cluster Type](#) 页。如果您有红帽帐户，请使用您的凭证登录。如果没有，请创建一个帐户。
2. 在页的 **Run it yourself** 部分中选择您的基础架构供应商。
3. 从 **OpenShift 安装程序** 下的下拉菜单中选择您的主机操作系统和架构，然后点 **下载安装程序**。
4. 将下载的文件保存在要存储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几个文件。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。删除集群需要这两个文件。
- 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，请为特定云供应商完成 OpenShift Container Platform 卸载流程。

5. 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
```

6. 从 [Red Hat OpenShift Cluster Manager](#) 下载安装 [pull secret](#)。此 pull secret 允许您与所含授权机构提供的服务进行身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

## 提示

另外，您还可以从[红帽客户门户网站](#)检索安装程序，您可以在其中指定要下载的安装程序版本。但是，您需要有一个有效的订阅才能访问此页。

## 5.5. 网络配置阶段

OpenShift Container Platform 安装前有两个阶段，您可以在其中自定义网络配置。

### 第 1 阶段

在创建清单文件前，您可以自定义 **install-config.yaml** 文件中的以下与网络相关的字段：

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

如需更多信息，请参阅“安装配置参数”。



### 注意

将 **networking.machineNetwork** 设置为与首选子网所在的无类别域间路由 (CIDR) 匹配。



### 重要

CIDR 范围 **172.17.0.0/16** 由 **libVirt** 保留。对于集群中的网络，您无法使用与 **172.17.0.0/16** CIDR 范围重叠的任何其他 CIDR 范围。

### 第 2 阶段

运行 **openshift-install create** 清单创建 清单文件后，您可以只使用您要修改的字段定义自定义 Cluster Network Operator 清单。您可以使用清单指定高级网络配置。

在阶段 2 中，您无法覆盖 **install-config.yaml** 文件中在第 1 阶段指定的值。但是，您可以在第 2 阶段自定义网络插件。

## 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

- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。

## 流程

### 1. 创建 `install-config.yaml` 文件。

#### a. 进入包含安装程序的目录并运行以下命令：

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

**1** 对于 `<installation_directory>`，请指定要存储安装程序创建的文件的目录名称。

在指定目录时：

- 验证该目录是否具有执行权限。在安装目录中运行 Terraform 二进制文件需要这个权限。
- 使用空目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。



#### 注意

始终删除 `~/.powervs` 目录，以避免重复使用过时的配置。运行以下命令：

```
$ rm -rf ~/.powervs
```

#### b. 在提示符处，提供云的配置详情：

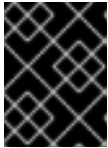
##### i. 可选：选择用于访问集群机器的 SSH 密钥。



#### 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. 为集群输入描述性名称。
2. 修改 `install-config.yaml` 文件。您可以在“安装配置参数”部分找到有关可用参数的更多信息。
  3. 备份 `install-config.yaml` 文件，以便您可以使用它安装多个集群。



### 重要

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

### 其他资源

- [AWS 的安装配置参数](#)

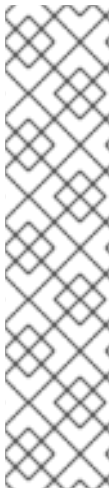
### 5.6.1. 集群安装的最低资源要求

每台集群机器都必须满足以下最低要求：

表 5.1. 最低资源要求

| 机器                   | 操作系统                     | vCPU [1] | 虚拟内存  | Storage | 每秒输入/输出 (IOPS) [2] |
|----------------------|--------------------------|----------|-------|---------|--------------------|
| bootstrap            | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Control plane (控制平面) | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Compute              | RHCOS、RHEL 8.6 及更新版本 [3] | 2        | 8 GB  | 100 GB  | 300                |

1. 当未启用并发多线程 (SMT) 或超线程时，一个 vCPU 相当于一个物理内核。启用后，使用以下公式来计算对应的比例： $(\text{每个内核数的线程}) \times \text{sockets} = \text{vCPU}$ 。
2. OpenShift Container Platform 和 Kubernetes 对磁盘性能非常敏感，建议使用更快的存储速度，特别是 control plane 节点上需要 10 ms p99 fsync 持续时间的 etcd。请注意，在许多云平台上，存储大小和 IOPS 可一起扩展，因此您可能需要过度分配存储卷来获取足够的性能。
3. 与所有用户置备的安装一样，如果您选择在集群中使用 RHEL 计算机，则负责所有操作系统生命周期管理和维护，包括执行系统更新、应用补丁和完成所有其他必要的任务。RHEL 7 计算机器的使用已弃用，并已在 OpenShift Container Platform 4.10 及更新的版本中删除。



### 注意

从 OpenShift Container Platform 版本 4.13 开始，RHCOS 基于 RHEL 版本 9.2，它更新了微架构要求。以下列表包含每个架构需要的最小指令集架构 (ISA)：

- x86-64 体系结构需要 x86-64-v2 ISA
- ARM64 架构需要 ARMv8.0-A ISA
- IBM Power 架构需要 Power 9 ISA
- s390x 架构需要 z14 ISA

如需更多信息，请参阅 [RHEL 架构](#)。

如果平台的实例类型满足集群机器的最低要求，则 OpenShift Container Platform 支持使用它。

## 其他资源

- [优化存储](#)

### 5.6.2. 为 AWS 测试的实例类型

以下 Amazon Web Services(AWS) 实例类型已经过 OpenShift Container Platform 测试。



#### 注意

将以下图中包含的机器类型用于 AWS 实例。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 5.1. 基于 64 位 x86 架构的机器类型

- c4.\*
- c5.\*
- c5a.\*
- i3.\*
- m4.\*
- m5.\*
- m5a.\*
- m6i.\*
- r4.\*
- r5.\*
- r5a.\*
- r6i.\*
- t3.\*
- t3a.\*

### 5.6.3. 在 64 位 ARM 基础架构上为 AWS 测试过的实例类型

OpenShift Container Platform 中已经测试了以下 Amazon Web Services (AWS) 64 位 ARM 实例类型。



#### 注意

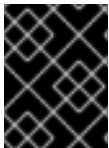
使用 AWS ARM 实例的以下图中包含的机器类型。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

**例 5.2. 基于 64 位 ARM 架构的机器类型**

- c6g.\*
- m6g.\*
- r8g.\*

**5.6.4. AWS 的自定义 install-config.yaml 文件示例**

您可以自定义安装配置文件 (**install-config.yaml**)，以指定有关 OpenShift Container Platform 集群平台的更多详细信息，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，并进行修改。

```

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
    metadataService:
      authentication: Optional 7
      type: m6i.xlarge
    replicas: 3
compute: 8
- hyperthreading: Enabled 9
  name: worker
  platform:
    aws:
      rootVolume:
        iops: 2000
        size: 500
        type: io1 10
      metadataService:
        authentication: Optional 11
      type: c5.4xlarge
      zones:
      - us-west-2c
    replicas: 3

```

```

metadata:
  name: test-cluster 12
networking: 13
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OVNKubernetes 14
  serviceNetwork:
    - 172.30.0.0/16
platform:
  aws:
    region: us-west-2 15
    propagateUserTags: true 16
    userTags:
      adminContact: jdoe
      costCenter: 7536
    amiID: ami-0c5d3e03c0ab9b19a 17
    serviceEndpoints: 18
      - name: ec2
        url: https://vpce-id.ec2.us-west-2.vpce.amazonaws.com
  fips: false 19
  sshKey: ssh-ed25519 AAAA... 20
  pullSecret: '{"auths": ...}' 21

```

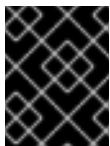
1 12 15 21 必需。安装程序会提示您输入这个值。

2 可选：添加此参数来强制 Cloud Credential Operator (CCO) 使用指定的模式。默认情况下，CCO 使用 **kube-system** 命名空间中的 root 凭证来动态尝试决定凭证的功能。有关 CCO 模式的详情，请参阅 [身份验证和授权指南](#) 中的 "About the Cloud Credential Operator" 部分。

3 8 13 16 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。仅使用一个 control plane 池。

5 9 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置为 **Disabled** 来禁用它。如果在某些集群机器中禁用并发多线程，则必须在所有集群机器中禁用它。



### 重要

如果您禁用并发多线程，请确保您的容量规划考虑机器性能显著降低的情况。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 10 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**。

7 11 是否需要 [Amazon EC2 实例元数据服务 v2](#) (IMDSv2)。为了要求 IMDSv2，请将参数值设置为 **Required**。要允许使用 IMDSv1 和 IMDSv2，请将参数值设置为 **Optional**。如果没有指定值，则允许 IMDSv1 和 IMDSv2。

**注意**

在集群安装过程中设置的 control plane 机器的 IMDS 配置只能使用 AWS CLI 更改。可以使用计算机器集来更改计算机器的 IMDS 配置。

- 14 要安装的集群网络插件。支持的值有 **OVNKubernetes** 和 **OpenShiftSDN**。默认值为 **OVNKubernetes**。
- 17 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。
- 18 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。
- 19 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**

要为集群启用 FIPS 模式，您必须从配置为以 FIPS 模式操作的 Red Hat Enterprise Linux (RHEL) 计算机运行安装程序。有关在 RHEL 中配置 FIPS 模式的更多信息，请参阅[在 FIPS 模式中安装该系统](#)。当以 FIPS 模式运行 Red Hat Enterprise Linux (RHEL) 或 Red Hat Enterprise Linux CoreOS (RHCOS) 时，OpenShift Container Platform 核心组件使用 RHEL 加密库，在 x86\_64、ppc64le 和 s390x 架构上提交到 NIST FIPS 140-2/140-3 Validation。

- 20 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。

### 5.6.5. 在安装过程中配置集群范围的代理

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，并确定它们中的任何站点是否需要绕过代理。默认情况下，所有集群出口流量都经过代理，包括对托管云供应商 API 的调用。如果需要，您将在 **Proxy** 对象的 **spec.noProxy** 字段中添加站点来绕过代理。





## 注意

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

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

## 流程

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: ec2.<aws_region>.amazonaws.com,elasticloadbalancing.
    <aws_region>.amazonaws.com,s3.<aws_region>.amazonaws.com 3
  additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

- 1 用于创建集群外 HTTP 连接的代理 URL。URL 方案必须是 **http**。
- 2 用于创建集群外 HTTPS 连接的代理 URL。
- 3 要从代理中排除的目标域名、IP 地址或其他网络 CIDR 的逗号分隔列表。在域前面加上 **.** 以仅匹配子域。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。如果您已将 Amazon **EC2**、**Elastic Load Balancing** 和 **S3** VPC 端点添加到 VPC 中，您必须将这些端点添加到 **noProxy** 字段。
- 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，其包含代理 HTTPS 连接所需的一个或多个额外 CA 证书。然后，Cluster Network Operator 会创建 **trusted-ca-bundle** 配置映射，将这些内容与 Red Hat Enterprise Linux CoreOS (RHCOS) 信任捆绑包合并，**Proxy** 对象的 **trustedCA** 字段中也会引用此配置映射。**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。
- 5 可选：决定 **Proxy** 对象的配置以引用 **trustedCA** 字段中 **user-ca-bundle** 配置映射的策略。允许的值是 **Proxyonly** 和 **Always**。仅在配置了 **http/https** 代理时，使用 **Proxyonly** 引用 **user-ca-bundle** 配置映射。使用 **Always** 始终引用 **user-ca-bundle** 配置映射。默认值为 **Proxyonly**。



## 注意

安装程序不支持代理的 **readinessEndpoints** 字段。



**注意**

如果安装程序超时，重启并使用安装程序的 **wait-for** 命令完成部署。例如：

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

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

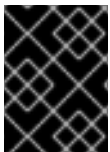
安装程序会创建一个名为 **cluster** 的集群范围代理，该代理使用提供的 **install-config.yaml** 文件中的代理设置。如果没有提供代理设置，仍然会创建一个 **cluster Proxy** 对象，但它会有一个空 **spec**。

**注意**

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

## 5.7. 通过下载二进制文件安装 OPENSIFT CLI

您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

### 5.7.1. 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. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。
5. 解包存档：

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

6. 将 **oc** 二进制文件放到 **PATH** 中的目录中。  
要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

**验证**

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

■

```
$ oc <command>
```

### 5.7.2. 在 Windows 上安装 OpenShift CLI

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

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
4. 使用 ZIP 程序解压存档。
5. 将 **oc** 二进制文件移到 **PATH** 中的目录中。  
要查看您的 **PATH**，请打开命令提示并执行以下命令：

```
C:\> path
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 5.7.3. 在 macOS 上安装 OpenShift CLI

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

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。



#### 注意

对于 macOS arm64，请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

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**，请打开终端并执行以下命令：

```
$ echo $PATH
```

#### 验证

- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

## 5.8. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案

默认情况下，管理员 secret 存储在 **kube-system** 项目中。如果您在 **install-config.yaml** 文件中将 **credentialsMode** 参数配置为 **Manual**，则必须使用以下替代方案之一：

- 要手动管理长期云凭证，请按照[手动创建长期凭证](#)中的步骤操作。
- 要实现在集群外为各个组件管理的短期凭证，请按照[配置 AWS 集群以使用短期凭证](#)中的步骤操作。

### 5.8.1. 手动创建长期凭证

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. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

#### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

4. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 自定义资源 (CR) 列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included 1 \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml 2 \
  --to=<path_to_directory_for_credentials_requests> 3
```

- 1** **--included** 参数仅包含特定集群配置所需的清单。

- 2 指定 **install-config.yaml** 文件的位置。
- 3 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

此命令为每个 **CredentialsRequest** 对象创建一个 YAML 文件。

### CredentialsRequest 对象示例

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

5. 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。

### 带有 secret 的 CredentialsRequest 对象示例

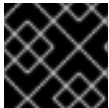
```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
...
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
      - effect: Allow
        action:
          - s3:CreateBucket
          - s3>DeleteBucket
        resource: "*"
  ...
  secretRef:
    name: <component_secret>
    namespace: <component_namespace>
  ...
```

## Secret 对象示例

```

apiVersion: v1
kind: Secret
metadata:
  name: <component_secret>
  namespace: <component_namespace>
data:
  aws_access_key_id: <base64_encoded_aws_access_key_id>
  aws_secret_access_key: <base64_encoded_aws_secret_access_key>

```



### 重要

在升级使用手动维护凭证的集群前，您必须确保 CCO 处于可升级状态。

## 5.8.2. 将 AWS 集群配置为使用短期凭证

要安装配置为使用 AWS 安全令牌服务 (STS) 的集群，您必须配置 CCO 实用程序并为集群创建所需的 AWS 资源。

### 5.8.2.1. 配置 Cloud Credential Operator 工具

当 Cloud Credential Operator (CCO) 以手动模式运行时，要从集群外部创建和管理云凭证，提取并准备 CCO 实用程序 (**ccctl**) 二进制文件。



### 注意

**ccctl** 工具是在 Linux 环境中运行的 Linux 二进制文件。

### 先决条件

- 您可以访问具有集群管理员权限的 OpenShift Container Platform 帐户。
- 已安装 OpenShift CLI (**oc**)。
- 您已为 **ccctl** 工具创建了用于以下权限的 AWS 帐户：

#### 例 5.3. 所需的 AWS 权限

##### 所需的 iam 权限

- **iam:CreateOpenIDConnectProvider**
- **iam:CreateRole**
- **iam>DeleteOpenIDConnectProvider**
- **iam>DeleteRole**
- **iam>DeleteRolePolicy**
- **iam:GetOpenIDConnectProvider**
- **iam:GetRole**

- **iam:GetUser**
- **iam:ListOpenIDConnectProviders**
- **iam:ListRolePolicies**
- **iam:ListRoles**
- **iam:PutRolePolicy**
- **iam:TagOpenIDConnectProvider**
- **iam:TagRole**

#### 所需的 s3 权限

- **s3:CreateBucket**
- **s3:DeleteBucket**
- **s3:DeleteObject**
- **s3:GetBucketAcl**
- **s3:GetBucketTagging**
- **s3:GetObject**
- **s3:GetObjectAcl**
- **s3:GetObjectTagging**
- **s3:ListBucket**
- **s3:PutBucketAcl**
- **s3:PutBucketPolicy**
- **s3:PutBucketPublicAccessBlock**
- **s3:PutBucketTagging**
- **s3:PutObject**
- **s3:PutObjectAcl**
- **s3:PutObjectTagging**

#### 所需的 cloudfront 权限

- **cloudfront:ListCloudFrontOriginAccessIdentities**
- **cloudfront:ListDistributions**
- **cloudfront:ListTagsForResource**

如果您计划通过公共 CloudFront 发行版 URL 将 OIDC 配置存储在 IAM 身份提供程序访问的私有 S3 存储桶中，则运行 **ccoctl** 工具的 AWS 帐户需要以下额外权限：

#### 例 5.4. 使用 CloudFront 私有 S3 存储桶的额外权限

- **cloudfront:CreateCloudFrontOriginAccessIdentity**
- **cloudfront:CreateDistribution**
- **cloudfront>DeleteCloudFrontOriginAccessIdentity**
- **cloudfront>DeleteDistribution**
- **cloudfront:GetCloudFrontOriginAccessIdentity**
- **cloudfront:GetCloudFrontOriginAccessIdentityConfig**
- **cloudfront:GetDistribution**
- **cloudfront:TagResource**
- **cloudfront:UpdateDistribution**



#### 注意

在使用 **ccoctl aws create-all** 命令处理凭证请求时，这些额外权限支持使用 **--create-private-s3-bucket** 选项。

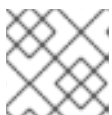
## 流程

1. 运行以下命令，为 OpenShift Container Platform 发行镜像设置变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像获取 CCO 容器镜像：

```
$ CCO_IMAGE=$(oc adm release info --image-for='cloud-credential-operator'
$RELEASE_IMAGE -a ~/.pull-secret)
```



#### 注意

确保 **\$RELEASE\_IMAGE** 的架构与将使用 **ccoctl** 工具的环境架构相匹配。

3. 运行以下命令，将 CCO 容器镜像中的 **ccoctl** 二进制文件提取到 OpenShift Container Platform 发行镜像中：

```
$ oc image extract $CCO_IMAGE --file="/usr/bin/ccoctl" -a ~/.pull-secret
```

4. 运行以下命令更改权限以使 **ccoctl** 可执行：

```
$ chmod 775 ccoctl
```

## 验证

- 要验证 **ccoctl** 是否准备就绪，可以尝试显示帮助文件。运行命令时使用相对文件名，例如：

```
$ ./ccoctl.rhel9
```

### 输出示例

OpenShift credentials provisioning tool

Usage:

**ccoctl** [command]

Available Commands:

alibabacloud Manage credentials objects for alibaba cloud  
aws Manage credentials objects for AWS cloud  
azure Manage credentials objects for Azure  
gcp Manage credentials objects for Google cloud  
help Help about any command  
ibmcloud Manage credentials objects for IBM Cloud  
nutanix Manage credentials objects for Nutanix

Flags:

-h, --help help for ccoctl

Use "**ccoctl** [command] --help" for more information about a command.

## 5.8.2.2. 使用 Cloud Credential Operator 实用程序创建 AWS 资源

创建 AWS 资源时有以下选项：

- 您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。这是创建资源的最快速方法。请参阅[使用单个命令创建 AWS 资源](#)。
- 如果您需要在修改 AWS 资源前查看 **ccoctl** 工具创建的 JSON 文件，或者 **ccoctl** 工具用于创建 AWS 资源的过程无法自动满足组织的要求，您可以单独创建 AWS 资源。请参阅[单独创建 AWS 资源](#)。

### 5.8.2.2.1. 使用单个命令创建 AWS 资源

如果 **ccoctl** 工具用于创建 AWS 资源的过程自动满足机构的要求，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。

否则，您可以单独创建 AWS 资源。如需更多信息，请参阅[单独创建 AWS 资源](#)。



#### 注意

默认情况下，**ccoctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccoctl\_output\_dir>** 来引用这个目录。

## 先决条件

您必须：



- 提取并准备好 **ccoctl** 二进制文件。

## 流程

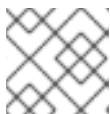
1. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \1
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \2
  --to=<path_to_directory_for_credentials_requests> \3
```

- 1 **--included** 参数仅包含特定集群配置所需的清单。
- 2 指定 **install-config.yaml** 文件的位置。
- 3 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。



### 注意

此命令可能需要一些时间才能运行。

3. 运行以下命令，使用 **ccoctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccoctl aws create-all \
  --name=<name> \1
  --region=<aws_region> \2
  --credentials-requests-dir=<path_to_credentials_requests_directory> \3
  --output-dir=<path_to_ccoctl_output_dir> \4
  --create-private-s3-bucket \5
```

- 1 指定用于标记创建用于跟踪的任何云资源的名称。
- 2 指定在其中创建云资源的 AWS 区域。
- 3 指定包含组件 **CredentialsRequest** 对象文件的目录。
- 4 可选：指定您希望 **ccoctl** 实用程序在其中创建对象的目录。默认情况下，实用程序在运行命令的目录中创建对象。
- 5 可选：默认情况下，**ccoctl** 实用程序将 OpenID Connect (OIDC) 配置文件存储在公共 S3 存储桶中，并使用 S3 URL 作为公共 OIDC 端点。要将 OIDC 配置存储在 IAM 身份提供程序通过公共 CloudFront 发行版 URL 访问的专用 S3 存储桶中，请使用 **--create-private-s3-bucket** 参数。



### 注意

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

### 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 `<path_to_ccoctl_output_dir>/manifests` 目录中的文件：

```
$ ls <path_to_ccoctl_output_dir>/manifests
```

### 输出示例

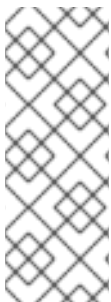
```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

#### 5.8.2.2.2. 单独创建 AWS 资源

您可以使用 **ccoctl** 工具单独创建 AWS 资源。这个选项对于在不同用户或部门之间创建这些资源的组织可能很有用。

否则，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。如需更多信息，请参阅“使用单个命令创建 AWS 资源”。



### 注意

默认情况下，**ccoctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 `<path_to_ccoctl_output_dir>` 来引用这个目录。

有些 **ccoctl** 命令会发出 AWS API 调用来创建或修改 AWS 资源。您可以使用 **--dry-run** 标志来避免 API 调用。使用此标志可在本地文件系统中创建 JSON 文件。您可以使用 **--cli-input-json** 参数查看和修改 JSON 文件，然后使用 AWS CLI 工具应用它们。

### 先决条件

- 提取并准备 **ccoctl** 二进制文件。

### 流程

- 运行以下命令，生成用于为集群设置 OpenID Connect 供应商的公共和私有 RSA 密钥文件：

```
$ ccoctl aws create-key-pair
```

## 输出示例

```
2021/04/13 11:01:02 Generating RSA keypair
2021/04/13 11:01:03 Writing private key to /<path_to_ccoctl_output_dir>/serviceaccount-signer.private
2021/04/13 11:01:03 Writing public key to /<path_to_ccoctl_output_dir>/serviceaccount-signer.public
2021/04/13 11:01:03 Copying signing key for use by installer
```

其中 **serviceaccount-signer.private** 和 **serviceaccount-signer.public** 是生成的密钥文件。

此命令还会在 **/<path\_to\_ccoctl\_output\_dir>/tls/bound-service-account-signing-key.key** 中创建集群在安装过程中所需的私钥。

2. 运行以下命令，在 AWS 上创建 OpenID Connect 身份提供程序和 S3 存储桶：

```
$ ccoctl aws create-identity-provider \
  --name=<name> \ 1
  --region=<aws_region> \ 2
  --public-key-file=<path_to_ccoctl_output_dir>/serviceaccount-signer.public 3
```

- 1** **<name>** 是用于标记为跟踪而创建的云资源的名称。
- 2** **<aws-region>** 是将在其中创建云资源的 AWS 区域。
- 3** **<path\_to\_ccoctl\_output\_dir>** 是 **ccoctl aws create-key-pair** 命令生成的公钥文件的路径。

## 输出示例

```
2021/04/13 11:16:09 Bucket <name>-oidc created
2021/04/13 11:16:10 OpenID Connect discovery document in the S3 bucket <name>-oidc at .well-known/openid-configuration updated
2021/04/13 11:16:10 Reading public key
2021/04/13 11:16:10 JSON web key set (JWKS) in the S3 bucket <name>-oidc at keys.json updated
2021/04/13 11:16:18 Identity Provider created with ARN: arn:aws:iam::<aws_account_id>:oidc-provider/<name>-oidc.s3.<aws_region>.amazonaws.com
```

其中 **openid-configuration** 是发现文档和 **key.json** 是一个 JSON Web 密钥集文件。

此命令还会在 **/<path\_to\_ccoctl\_output\_dir>/manifests/cluster-authentication-02-config.yaml** 中创建 YAML 配置文件。此文件为集群生成的服务帐户令牌设置签发者 URL 字段，以便 AWS IAM 身份提供程序信任令牌。

3. 为集群中的每个组件创建 IAM 角色：

- a. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

- b. 从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \ ❶
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
  ❷
  --to=<path_to_directory_for_credentials_requests> ❸
```

- ❶ **--included** 参数仅包含特定集群配置所需的清单。
- ❷ 指定 **install-config.yaml** 文件的位置。
- ❸ 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

c. 运行以下命令，使用 **ccocctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccocctl aws create-iam-roles \
  --name=<name> \
  --region=<aws_region> \
  --credentials-requests-dir=<path_to_credentials_requests_directory> \
  --identity-provider-arn=arn:aws:iam::<aws_account_id>:oidc-provider/<name>-oidc.s3.
  <aws_region>.amazonaws.com
```



### 注意

对于使用其他 IAM API 端点的 AWS 环境（如 GovCloud），还必须使用 **--region** 参数指定您的区域。

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

对于每个 **CredentialsRequest** 对象，**ccocctl** 创建一个带有信任策略的 IAM 角色，该角色与指定的 OIDC 身份提供程序相关联，以及来自 OpenShift Container Platform 发行镜像的每个 **CredentialsRequest** 对象中定义的权限策略。

## 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccocctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccocctl_output_dir>/manifests
```

## 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
```

```

openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml

```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

### 5.8.2.3. 整合 Cloud Credential Operator 实用程序清单

要为单个组件在集群外实现短期安全凭证，您必须将创建 Cloud Credential Operator 实用程序 (**ccctl**) 的清单文件移到安装程序的正确目录中。

#### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您已配置了 Cloud Credential Operator 实用程序 (**ccctl**)。
- 已使用 **ccctl** 工具创建了集群所需的云供应商资源。

#### 流程

1. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

#### 配置文件片段示例

```

apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...

```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，将 **ccctl** 工具生成的清单复制到安装程序创建的 **manifests** 目录中：

```
$ cp /<path_to_ccctl_output_dir>/manifests/* ./manifests/
```

4. 将包含私钥的 **tls** 目录复制到安装目录中：

```
$ cp -a /<path_to_ccctl_output_dir>/tls .
```

## 5.9. CLUSTER NETWORK OPERATOR 配置

集群网络的配置作为 Cluster Network Operator(CNO)配置的一部分指定，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：

**clusterNetwork**

从中分配 Pod IP 地址的 IP 地址池。

**serviceNetwork**

服务的 IP 地址池。

**defaultNetwork.type**

集群网络插件，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插件配置。

**5.9.1. Cluster Network Operator 配置对象**

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

表 5.2. Cluster Network Operator 配置对象

| 字段                          | 类型     | 描述                                                                                                                                                                                                     |
|-----------------------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>metadata.name</b>        | 字符串    | CNO 对象的名称。这个名称始终是 <b>集群</b> 。                                                                                                                                                                          |
| <b>spec.clusterNetwork</b>  | array  | <p>用于指定从哪些 IP 地址块分配 Pod IP 地址以及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>             |
| <b>spec.serviceNetwork</b>  | array  | <p>服务的 IP 地址块。OpenShift SDN 和 OVN-Kubernetes 网络插件只支持服务网络的一个 IP 地址块。例如：</p> <pre>spec:   serviceNetwork:     - 172.30.0.0/14</pre> <p>您只能在创建清单前在 <b>install-config.yaml</b> 文件中自定义此字段。该值在清单文件中是只读的。</p> |
| <b>spec.defaultNetwork</b>  | object | 为集群网络配置网络插件。                                                                                                                                                                                           |
| <b>spec.kubeProxyConfig</b> | object | 此对象的字段指定 kube-proxy 配置。如果使用 OVN-Kubernetes 集群网络供应商，则 kube-proxy 配置不会起作用。                                                                                                                               |



### 重要

对于需要在多个网络间部署对象的集群，请确保为 **install-config.yaml** 文件中定义的每种网络类型指定与 **clusterNetwork.hostPrefix** 参数相同的值。为每个 **clusterNetwork.hostPrefix** 参数设置不同的值可能会影响 OVN-Kubernetes 网络插件，其中插件无法有效地在不同节点间路由对象流量。

#### 5.9.1.1. defaultNetwork 对象配置

下表列出了 **defaultNetwork** 对象的值：

表 5.3. defaultNetwork 对象

| 字段                         | 类型     | 描述                                                                                                                                                                                                                                                                                                                           |
|----------------------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>type</b>                | 字符串    | <b>OpenShiftSDN 或 OVNKubernetes</b> 。Red Hat OpenShift Networking 网络插件在安装过程中被选择。您可以通过从 OpenShift SDN 迁移到 OVN-Kubernetes 来更改这个值。 <div>  <div> <b>注意</b><br/>           OpenShift Container Platform 默认使用 OVN-Kubernetes 网络插件。 </div> </div> |
| <b>openshiftSDNConfig</b>  | object | 此对象仅对 OpenShift SDN 网络插件有效。                                                                                                                                                                                                                                                                                                  |
| <b>ovnKubernetesConfig</b> | object | 此对象仅对 OVN-Kubernetes 网络插件有效。                                                                                                                                                                                                                                                                                                 |

#### 5.9.1.1.1. 配置 OpenShift SDN 网络插件

下表描述了 OpenShift SDN 网络插件的配置字段：

表 5.4. openshiftSDNConfig object

| 字段        | 类型  | 描述                                                                                                                                                            |
|-----------|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>模式</b> | 字符串 | 配置 OpenShift SDN 的网络隔离模式。默认值为 <b>NetworkPolicy</b> 。<br><br><b>Multitenant</b> 和 <b>Subnet</b> 值可用于向后兼容 OpenShift Container Platform 3.x，但不建议使用。此值在集群安装后无法更改。 |

| 字段        | 类型      | 描述                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mtu       | integer | <p>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。</p> <p>如果自动探测的值不是您期望的值，请确认节点上主网络接口上的 MTU 是否正确。您不能使用这个选项更改节点上主网络接口的 MTU 值。</p> <p>如果集群中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>。</p> <p>您可以在集群安装过程中或安装后任务设置值。如需更多信息，请参阅 OpenShift Container Platform 网络文档中的"删除集群网络 MTU"。</p> |
| vxlanPort | integer | <p>用于所有 VXLAN 数据包的端口。默认值为 <b>4789</b>。此值在集群安装后无法更改。</p> <p>如果您在虚拟环境中运行，且现有节点是另一个 VXLAN 网络的一部分，则可能需要更改此设置。例如，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。</p> <p>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。</p>                                                                                     |

5.9.1.1.2. 配置 OVN-Kubernetes 网络插件

下表描述了 OVN-Kubernetes 网络插件的配置字段：

表 5.5. ovnKubernetesConfig object

| 字段  | 类型      | 描述                                                                                                                                                                                                                                                                                                |
|-----|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mtu | integer | <p>Geneve（通用网络虚拟化封装）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值，请确认节点上主网络接口上的 MTU 是否正确。您不能使用这个选项更改节点上主网络接口的 MTU 值。</p> <p>如果集群中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 100。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> |



| 字段                       | 类型             | 描述                                                                                                                                                                                                                                             |
|--------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>genevePort</b>        | <b>integer</b> | 用于所有 Geneve 数据包的端口。默认值为 <b>6081</b> 。此值在集群安装后无法更改。                                                                                                                                                                                             |
| <b>ipsecConfig</b>       | <b>object</b>  | 指定一个空对象来启用 IPsec 加密。                                                                                                                                                                                                                           |
| <b>ipv4</b>              | <b>object</b>  | 为 IPv4 设置指定配置对象。                                                                                                                                                                                                                               |
| <b>ipv6</b>              | <b>object</b>  | 为 IPv6 设置指定配置对象。                                                                                                                                                                                                                               |
| <b>policyAuditConfig</b> | <b>object</b>  | 指定用于自定义网络策略审计日志的配置对象。如果未设置，则使用默认的审计日志设置。                                                                                                                                                                                                       |
| <b>gatewayConfig</b>     | <b>object</b>  | <p>可选：指定一个配置对象来自定义如何将出口流量发送到节点网关。</p> <div>  <div> <p><b>注意</b></p> <p>在迁移出口流量时，工作负载和服务流量会受到一定影响，直到 Cluster Network Operator (CNO) 成功推出更改。</p> </div> </div> |

表 5.6. ovnKubernetesConfig.ipv4 object

| 字段                                 | 类型  | 描述                                                                                                                                                                                                                                                                                                                                                      |
|------------------------------------|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>internalTransitSwitchSubnet</b> | 字符串 | <p>如果您的现有网络基础架构与 <b>100.88.0.0/16</b> IPv4 子网重叠，您可以指定不同的 IP 地址范围供 OVN-Kubernetes 使用。启用东西流量的分布式传输交换机的子网。此子网不能与 OVN-Kubernetes 或主机本身使用的任何其他子网重叠。必须足够大，以适应集群中的每个节点一个 IP 地址。</p> <p>默认值为 <b>100.88.0.0/16</b>。</p>                                                                                                                                          |
| <b>internalJoinSubnet</b>          | 字符串 | <p>如果您的现有网络基础架构与 <b>100.64.0.0/16</b> IPv4 子网重叠，您可以指定不同的 IP 地址范围供 OVN-Kubernetes 使用。您必须确保 IP 地址范围没有与 OpenShift Container Platform 安装使用的任何其他子网重叠。IP 地址范围必须大于可添加到集群的最大节点数。例如，如果 <b>clusterNetwork.cidr</b> 值为 <b>10.128.0.0/14</b>，并且 <b>clusterNetwork.hostPrefix</b> 值为 <b>/23</b>，则最大节点数量为 <b>2^(23-14)=512</b>。</p> <p>默认值为 <b>100.64.0.0/16</b>。</p> |

表 5.7. ovnKubernetesConfig.ipv6 object

| 字段                                 | 类型     | 描述                                                                                                                                                                                                                 |
|------------------------------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>internalTransitSwitchSubnet</b> | string | <p>如果您的现有网络基础架构与 <b>fd98::/48</b> IPv6 子网重叠，您可以指定不同的 IP 地址范围供 OVN-Kubernetes 使用。您必须确保 IP 地址范围没有与 OpenShift Container Platform 安装使用的任何其他子网重叠。IP 地址范围必须大于可添加到集群的最大节点数。</p> <p>在安装后无法更改此字段。默认值为 <b>fd98::/48</b>。</p> |
| <b>internalJoinSubnet</b>          | 字符串    | <p>如果您的现有网络基础架构与 <b>fd98::/64</b> IPv6 子网重叠，您可以指定不同的 IP 地址范围供 OVN-Kubernetes 使用。您必须确保 IP 地址范围没有与 OpenShift Container Platform 安装使用的任何其他子网重叠。IP 地址范围必须大于可添加到集群的最大节点数。</p> <p>默认值为 <b>fd98::/64</b>。</p>             |

表 5.8. policyAuditConfig object

| 字段                    | 类型  | 描述                                                                                                                                                                                                                                                                                                                                        |
|-----------------------|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>rateLimit</b>      | 整数  | 每个节点每秒生成一次的消息数量上限。默认值为每秒 <b>20</b> 条消息。                                                                                                                                                                                                                                                                                                   |
| <b>maxFileSize</b>    | 整数  | 审计日志的最大大小，以字节为单位。默认值为 <b>50000000</b> 或 50 MB。                                                                                                                                                                                                                                                                                            |
| <b>maxLogFiles</b>    | 整数  | 保留的日志文件的最大数量。                                                                                                                                                                                                                                                                                                                             |
| <b>目的地</b>            | 字符串 | <p>以下附加审计日志目标之一：</p> <p><b>libc</b><br/>主机上的 journald 进程的 libc <b>syslog ()</b> 函数。</p> <p><b>UDP:&lt;host&gt;:&lt;port&gt;</b><br/>一个 syslog 服务器。将 <b>&lt;host&gt;:&lt;port&gt;</b> 替换为 <b>syslog 服务器的主机和端口</b>。</p> <p><b>Unix:&lt;file&gt;</b><br/>由 <b>&lt;file&gt;</b> 指定的 Unix 域套接字文件。</p> <p><b>null</b><br/>不要将审计日志发送到任何其他目标。</p> |
| <b>syslogFacility</b> | 字符串 | syslog 工具，如 <b>kern</b> ，如 RFC5424 定义。默认值为 <b>local0</b> 。                                                                                                                                                                                                                                                                                |

表 5.9. gatewayConfig object

| 字段                    | 类型     | 描述                                                                                                                                                                                                                                                                  |
|-----------------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>routingViaHost</b> | 布尔值    | <p>将此字段设置为 <b>true</b>，将来自 pod 的出口流量发送到主机网络堆栈。对于依赖于在内核路由表中手动配置路由的高级别安装和应用程序，您可能需要将出口流量路由到主机网络堆栈。默认情况下，出口流量在 OVN 中进行处理以退出集群，不受内核路由表中的特殊路由的影响。默认值为 <b>false</b>。</p> <p>此字段与 Open vSwitch 硬件卸载功能有交互。如果将此字段设置为 <b>true</b>，则不会获得卸载的性能优势，因为主机网络堆栈会处理出口流量。</p>           |
| <b>ipForwarding</b>   | object | <p>您可以使用 <b>Network</b> 资源中的 <b>ipForwarding</b> 规格来控制 OVN-Kubernetes 管理接口上所有流量的 IP 转发。指定 <b>Restricted</b> 只允许 Kubernetes 相关流量的 IP 转发。指定 <b>Global</b> 以允许转发所有 IP 流量。对于新安装，默认值为 <b>Restricted</b>。对于 OpenShift Container Platform 4.14 的更新，默认值为 <b>Global</b>。</p> |
| <b>ipv4</b>           | object | <p>可选：指定一个对象来为主机配置内部 OVN-Kubernetes 伪装地址，以服务 IPv4 地址的流量。</p>                                                                                                                                                                                                        |
| <b>ipv6</b>           | object | <p>可选：指定一个对象来为主机配置内部 OVN-Kubernetes 伪装地址，以服务 IPv6 地址的流量。</p>                                                                                                                                                                                                        |

表 5.10. gatewayConfig.ipv4 对象

| 字段                              | 类型  | 描述                                                                                    |
|---------------------------------|-----|---------------------------------------------------------------------------------------|
| <b>internalMasqueradeSubnet</b> | 字符串 | <p>内部使用的伪装 IPv4 地址，以启用主机服务流量。主机配置了这些 IP 地址和共享网关网桥接口。默认值为 <b>169.254.169.0/29</b>。</p> |

表 5.11. gatewayConfig.ipv6 对象

| 字段                              | 类型  | 描述                                                                              |
|---------------------------------|-----|---------------------------------------------------------------------------------|
| <b>internalMasqueradeSubnet</b> | 字符串 | <p>内部使用的伪装 IPv6 地址，以启用主机服务流量。主机配置了这些 IP 地址和共享网关网桥接口。默认值为 <b>fd69::/125</b>。</p> |

## 启用 IPsec 的 OVN-Kubernetes 配置示例

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

5.9.1.2. kubeProxyConfig 对象配置（仅限 OpenShiftSDN 容器网络接口）

kubeProxyConfig 对象的值在下表中定义：

表 5.12. kubeProxyConfig object

| 字段                                      | 类型    | 描述                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| iptablesSyncPeriod                      | 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 时间包</a> 文档。</p> <div><b>注意</b><p>由于 OpenShift Container Platform 4.3 及更高版本中引进了性能改进，不再需要调整 <b>iptablesSyncPeriod</b> 参数。</p></div> |
| proxyArguments.iptables-min-sync-period | array | <p>刷新 <b>iptables</b> 规则前的最短持续时间。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> <div><pre>kubeProxyConfig:   proxyArguments:     iptables-min-sync-period:       - 0s</pre></div>  |

5.10. 指定高级网络配置

您可以使用网络插件的高级网络配置将集群集成到现有网络环境中。您只能在安装集群前指定高级网络配置。



**重要**

不支持通过修改安装程序创建的 OpenShift Container Platform 清单文件来自定义网络配置。支持应用您创建的清单文件，如以下流程中所示。

先决条件

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

流程

- 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** **<installation\_directory>** 指定包含集群的 **install-config.yaml** 文件的目录名称。

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

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

3. 在 **cluster-network-03-config.yml** 文件中指定集群的高级网络配置，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
```

#### 为 OVN-Kubernetes 网络供应商启用 IPsec

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

4. 可选：备份 **manifests/cluster-network-03-config.yml** 文件。在创建 Ignition 配置文件时，安装程序会消耗 **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](#)。

## 5.11. 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。

2. 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
```

- 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，几个网络配置文件位于 **manifests/** 目录中，如下所示：

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

### 输出示例

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

3. 在编辑器中打开 **cluster-ingress-default-ingresscontroller.yaml** 文件，并输入描述您想要的 Operator 配置的自定义资源（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
```

4. 保存 **cluster-ingress-default-ingresscontroller.yaml** 文件并退出文本编辑器。
5. 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。

## 5.12. 使用 OVN-KUBERNETES 配置混合网络

您可以将集群配置为使用 OVN-Kubernetes 网络插件的混合网络。这允许支持不同节点网络配置的混合集群。



## 注意

此配置是在同一集群中同时运行 Linux 和 Windows 节点所必需的。

## 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/** 目录中 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 不能与 **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 节点所需要的，且不得为任何其他云供应商配置。自定义端口可以是除默认 **4789** 端口外的任何打开的端口。有关此要求的更多信息，请参阅 Microsoft 文档中的 [Pod 到主机间的 pod 连接性](#)。



### 注意

Windows Server Long-Term Servicing Channel (LTSC) : Windows Server 2019 在带有自定义 **hybridOverlayVXLANPort** 值的集群中不被支持，因为这个 Windows server 版本不支持选择使用自定义的 VXLAN 端口。

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](#)。

## 5.13. 部署集群

您可以在兼容云平台上安装 OpenShift Container Platform。



### 重要

在初始安装过程中，您只能运行安装程序的 **create cluster** 命令一次。

### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。
- 已确认主机上的云供应商帐户具有部署集群的正确权限。权限不正确的帐户会导致安装过程失败，并显示包括缺失权限的错误消息。

### 流程

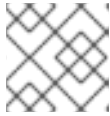
1. 进入包含安装程序的目录并初始化集群部署：

```
$ ./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**，而不是 **info**。



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



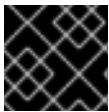
### 注意

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

## 验证

当集群部署成功完成时：

- 终端会显示用于访问集群的说明，包括指向 Web 控制台和 **kubeadmin** 用户的凭证的链接。
- 凭证信息还会输出到 `<installation_directory>/openshift_install.log`。

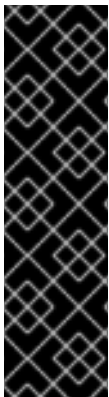


### 重要

不要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。

## 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: "password"
INFO Time elapsed: 36m22s
```



### 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。一个例外是，您必须手动批准待处理的 **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 小时时间进行轮转。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。

## 5.14. 使用 CLI 登录集群

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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

### 先决条件

- 已部署 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. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

### 输出示例

```
system:admin
```

/validating-an-installation.adoc

## 5.15. 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 控制台的更多详情](#)，请参阅 [访问 Web 控制台](#)。

## 5.16. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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](#)。

## 5.17. 后续步骤

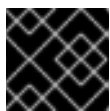
- [验证安装](#).
- [自定义集群](#)
- [远程健康报告](#).
- [删除云供应商凭证](#)

## 第 6 章 在受限网络中的 AWS 上安装集群

在 OpenShift Container Platform 版本 4.14 中，您可以通过在现有 Amazon Virtual Private Cloud (VPC) 上创建安装发行内容的内部镜像在受限网络中的 Amazon Web Services (AWS) 上安装集群。

### 6.1. 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。
- 您可以阅读有关 [选择集群安装方法的文档](#)，并为用户准备它。
- 您已 [将断开连接的安装的镜像镜像](#) 到 registry，并获取了 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。

- AWS 中有一个现有的 VPC。当使用安装程序置备的基础架构安装到受限网络时，无法使用安装程序置备的 VPC。您必须使用用户置备的 VPC 来满足以下要求之一：
  - 包含镜像 registry
  - 具有防火墙规则或对等连接来访问其他位置托管的镜像 registry
- 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。集群将继续使用您当前的 AWS 凭证为集群的整个生命周期创建 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

如果要配置代理，请务必查看此站点列表。

### 6.2. 关于在受限网络中安装

在 OpenShift Container Platform 4.14 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。受限网络安装可以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。

如果您选择在云平台中执行受限网络安装，您仍需要访问其云 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、Nutanix 或 VMware vSphere 上安装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，以镜像 OpenShift 镜像 registry 的内容并包含安装介质。您可以在镜像主机上创建此 registry，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。

### 6.2.1. 其他限制

受限网络中的集群有以下额外限制和限制：

- **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。

## 6.3. 关于使用自定义 VPC

在 OpenShift Container Platform 4.14 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。

### 6.3.1. 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。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](#) 以及 [处理 VPC 和子网](#)。

安装程序无法：

- 分割网络范围供集群使用。
- 设置子网的路由表。
- 设置 VPC 选项，如 DHCP。

您必须在安装集群前完成这些任务。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 [您的 VPC 的路由表](#)。

您的 VPC 必须满足以下特征：

- VPC 不能使用 **kubernetes.io/cluster/.\*: owned, Name**, 和 **openshift.io/cluster** 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。您不能使用 **Name** 标签，因为它与 EC2 **Name** 字段重叠，且安装失败。
- 您需要在您的 VPC 中启用 **enableDnsSupport** 和 **enableDnsHostnames** 属性，以便集群可以使用附加到 VPC 中的 Route 53 区来解析集群内部的 DNS 记录。请参阅 AWS 文档中的 [您的 VPC 中的 DNS 支持](#) 部分。  
如果要使用您自己的 Route 53 托管私有区，您必须在安装集群前将现有托管区与 VPC 相关联。您可以使用 **install-config.yaml** 文件中的 **platform.aws.hostedZone** 和 **platform.aws.hostedZoneRole** 字段定义托管区。您可以通过与安装集群的帐户共享来使用来自另一个帐户的私有托管区。如果使用另一个帐户的私有托管区，则必须使用 **Passthrough** 或 **Manual** 凭证模式。

如果您在断开连接的环境中工作，则无法访问 EC2、ELB 和 S3 端点的公共 IP 地址。根据您要在安装过程中限制互联网流量的级别，有以下配置选项：

#### 6.3.1.1. 选项 1：创建 VPC 端点

创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

通过这个选项，网络流量在 VPC 和所需的 AWS 服务之间保持私有。

#### 6.3.1.2. 选项 2：创建一个没有 VPC 端点的代理

作为安装过程的一部分，您可以配置 HTTP 或 HTTPS 代理。使用此选项时，互联网流量会通过代理访问所需的 AWS 服务。

#### 6.3.1.3. 选项 3：创建带有 VPC 端点的代理

作为安装过程的一部分，您可以使用 VPC 端点配置 HTTP 或 HTTPS 代理。创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

在 **install-config.yaml** 文件中配置代理时，将这些端点添加到 **noProxy** 字段。通过这个选项，代理会阻止集群直接访问互联网。但是，您的 VPC 和所需的 AWS 服务之间网络流量保持私有。

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      |

| 组件   | AWS 类型                                                                                                                                                                     | 描述                                                                                |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| 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。 |

### 6.3.2. 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。

### 6.3.3. 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。

### 6.3.4. 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)。



## 6.4. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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。

## 6.5. 为集群节点 SSH 访问生成密钥对

在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 **core** 用户的 `~/.ssh/authorized_keys` 列表中，这将启用免密码身份验证。

将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。

如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。`./openshift-install gather` 命令还需要在集群节点上设置 SSH 公钥。



### 重要

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。



### 注意

您必须使用本地密钥，而不是使用特定平台方法配置 [的密钥](#)，如 [AWS 密钥对](#)。

### 流程

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，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_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。



### 注意

如果您计划在 **x86\_64**、**ppc64le** 和 **s390x** 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 **ed25519** 算法的密钥。相反，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

3. 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。



#### 注意

在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- a. 如果 `ssh-agent` 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

#### 输出示例

```
Agent pid 31874
```



#### 注意

如果集群处于 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。

4. 将 SSH 私钥添加到 `ssh-agent`：

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

- 1** 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

#### 输出示例

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

#### 后续步骤

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

## 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

- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。对于受限网络安装，这些文件位于您的镜像主机上。
- 您有创建镜像 registry 期间生成的 **imageContentSources** 值。
- 您已获取了镜像 registry 的证书内容。

## 流程

### 1. 创建 **install-config.yaml** 文件。

#### a. 进入包含安装程序的目录并运行以下命令：

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

**1** 对于 **<installation\_directory>**，请指定要存储安装程序创建的文件目录名称。

在指定目录时：

- 验证该目录是否具有执行权限。在安装目录中运行 Terraform 二进制文件需要这个权限。
- 使用空目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。



#### 注意

始终删除 **~/.powervs** 目录，以避免重复使用过时的配置。运行以下命令：

```
$ rm -rf ~/.powervs
```

#### b. 在提示符处，提供云的配置详情：

##### i. 可选：选择用于访问集群机器的 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. 为集群输入描述性名称。

2. 编辑 **install-config.yaml** 文件，以提供在受限网络中安装所需的额外信息。

- a. 更新 **pullSecret** 值，使其包含 registry 的身份验证信息：

```
pullSecret: '{"auths":{"<mirror_host_name>:5000":{"auth":"<credentials>","email":  
"you@example.com"}}}'
```

对于 **<mirror\_host\_name>**，请指定您在镜像 registry 证书中指定的 registry 域名；对于 **<credentials>**，请指定您的镜像 registry 的 base64 编码用户名和密码。

- b. 添加 **additionalTrustBundle** 参数和值。

[illegible]

该值必须是您用于镜像 registry 的证书文件内容。证书文件可以是现有的可信证书颁发机构，也可以是您为镜像 registry 生成的自签名证书。

- c. 在以下位置定义 VPC 安装集群的子网：

```
subnets:
- subnet-1
- subnet-2
- subnet-3
```

- d. 添加镜像内容资源，类似于以下 YAML 摘录：

```
imageContentSources:
- mirrors:
  - <mirror_host_name>:5000/<repo_name>/release
    source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - <mirror_host_name>:5000/<repo_name>/release
    source: registry.redhat.io/ocp/release
```

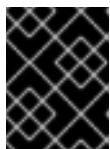
对于这些值，请使用您在创建镜像 registry 时记录的 **imageContentSources**。

- e. 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. 对您需要的 **install-config.yaml** 文件进行任何其他修改。您可以在 [安装配置参数部分](#)找到有关可用参数的更多信息。
4. 备份 **install-config.yaml** 文件，以便您可以使用它安装多个集群。



## 重要

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

### 其他资源

- [AWS 的安装配置参数](#)

### 6.6.1. 集群安装的最低资源要求

每台集群机器都必须满足以下最低要求：

表 6.1. 最低资源要求

| 机器                   | 操作系统                     | vCPU [1] | 虚拟内存  | Storage | 每秒输入/输出 (IOPS) [2] |
|----------------------|--------------------------|----------|-------|---------|--------------------|
| bootstrap            | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Control plane (控制平面) | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Compute              | RHCOS、RHEL 8.6 及更新版本 [3] | 2        | 8 GB  | 100 GB  | 300                |

1. 当未启用并发多线程 (SMT) 或超线程时，一个 vCPU 相当于一个物理内核。启用后，使用以下公式来计算对应的比例： $(\text{每个内核数的线程}) \times \text{sockets} = \text{vCPU}$ 。
2. OpenShift Container Platform 和 Kubernetes 对磁盘性能非常敏感，建议使用更快的存储速度，特别是 control plane 节点上需要 10 ms p99 fsync 持续时间的 etcd。请注意，在许多云平台上，存储大小和 IOPS 可一起扩展，因此您可能需要过度分配存储卷来获取足够的性能。
3. 与所有用户置备的安装一样，如果您选择在集群中使用 RHEL 计算机，则负责所有操作系统生命周期管理和维护，包括执行系统更新、应用补丁和完成所有其他必要的任务。RHEL 7 计算机器的使用已弃用，并已在 OpenShift Container Platform 4.10 及更新的版本中删除。



## 注意

从 OpenShift Container Platform 版本 4.13 开始，RHCOS 基于 RHEL 版本 9.2，它更新了微架构要求。以下列表包含每个架构需要的最小指令集架构 (ISA)：

- x86-64 体系结构需要 x86-64-v2 ISA
- ARM64 架构需要 ARMv8.0-A ISA
- IBM Power 架构需要 Power 9 ISA
- s390x 架构需要 z14 ISA

如需更多信息，请参阅 [RHEL 架构](#)。

如果平台的实例类型满足集群机器的最低要求，则 OpenShift Container Platform 支持使用它。

## 其他资源

- [优化存储](#)

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

您可以自定义安装配置文件 (**install-config.yaml**)，以指定有关 OpenShift Container Platform 集群平台的更多详细信息，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，并进行修改。

```
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
      metadataService:
        authentication: Optional 7
        type: m6i.xlarge
      replicas: 3
compute: 8
- hyperthreading: Enabled 9
  name: worker
  platform:
    aws:
      rootVolume:
        iops: 2000
        size: 500
        type: io1 10
      metadataService:
        authentication: Optional 11
        type: c5.4xlarge
      zones:
      - us-west-2c
      replicas: 3
metadata:
  name: test-cluster 12
networking:
```

```

clusterNetwork:
- cidr: 10.128.0.0/14
  hostPrefix: 23
machineNetwork:
- cidr: 10.0.0.0/16
networkType: OVNKubernetes 13
serviceNetwork:
- 172.30.0.0/16
platform:
aws:
  region: us-west-2 14
  propagateUserTags: true 15
  userTags:
    adminContact: jdoe
    costCenter: 7536
  subnets: 16
  - subnet-1
  - subnet-2
  - subnet-3
  amiID: ami-0c5d3e03c0ab9b19a 17
  serviceEndpoints: 18
  - name: ec2
    url: https://vpce-id.ec2.us-west-2.vpce.amazonaws.com
  hostedZone: Z3URY6TWQ91KVV 19
fips: false 20
sshKey: ssh-ed25519 AAAA... 21
pullSecret: '{"auths":{"<local_registry>":{"auth": "<credentials>","email": "you@example.com"}}}' 22
additionalTrustBundle: | 23
  -----BEGIN CERTIFICATE-----
  <MY_TRUSTED_CA_CERT>
  -----END CERTIFICATE-----
imageContentSources: 24
- 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

```

1 12 14 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。默认情况下，CCO 使用 **kube-system** 命名空间中的 root 凭证来动态尝试决定凭证的功能。有关 CCO 模式的详情，请参阅 *身份验证和授权指南* 中的 "About the Cloud Credential Operator" 部分。

3 8 15 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。仅使用一个 control plane 池。

5 9 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置为 **Disabled** 来禁用它。如果在某些集群机器中禁用并发多线程，则必须在所有集群机器中禁用它。



**重要**

如果您禁用并发多线程，请确保您的容量规划考虑机器性能显著降低的情况。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 10** 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**。

- 7 11** 是否需要 [Amazon EC2 实例元数据服务 v2](#) (IMDSv2)。为了要求 IMDSv2，请将参数值设置为 **Required**。要允许使用 IMDSv1 和 IMDSv2，请将参数值设置为 **Optional**。如果没有指定值，则允许 IMDSv1 和 IMDSv2。

**注意**

在集群安装过程中设置的 control plane 机器的 IMDS 配置只能使用 AWS CLI 更改。可以使用计算机器集来更改计算机器的 IMDS 配置。

- 13** 要安装的集群网络插件。支持的值有 **OVNKubernetes** 和 **OpenShiftSDN**。默认值为 **OVNKubernetes**。
- 16** 如果您提供自己的 VPC，为集群使用的每个可用区指定子网。
- 17** 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。
- 18** 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。
- 19** 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。
- 20** 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**

要为集群启用 FIPS 模式，您必须从配置为以 FIPS 模式操作的 Red Hat Enterprise Linux (RHEL) 计算机运行安装程序。有关在 RHEL 中配置 FIPS 模式的更多信息，请参阅[在 FIPS 模式中安装该系统](#)。当以 FIPS 模式运行 Red Hat Enterprise Linux (RHEL) 或 Red Hat Enterprise Linux CoreOS (RHCOS) 时，OpenShift Container Platform 核心组件使用 RHEL 加密库，在 x86\_64、ppc64le 和 s390x 架构上提交到 NIST FIPS 140-2/140-3 Validation。

- 21** 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。

- 22** 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 **registry.example.com:5000**。对于 **<credentials>**，请为您的镜像 registry 指定 base64 编码的用户名和密码。



- 23 提供用于镜像 registry 的证书文件内容。
- 24 提供命令输出中的 **imageContentSources** 部分来 镜像存储库。

### 6.6.3. 在安装过程中配置集群范围的代理

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，并确定它们中的任何站点是否需要绕过代理。默认情况下，所有集群出口流量都经过代理，包括对托管云供应商 API 的调用。如果需要，您将在 **Proxy** 对象的 **spec.noProxy** 字段中添加站点来绕过代理。



#### 注意

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

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

#### 流程

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: ec2.<aws_region>.amazonaws.com,elasticloadbalancing.
    <aws_region>.amazonaws.com,s3.<aws_region>.amazonaws.com 3
  additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

- 1 用于创建集群外 HTTP 连接的代理 URL。URL 方案必须是 **http**。
- 2 用于创建集群外 HTTPS 连接的代理 URL。
- 3 要从代理中排除的目标域名、IP 地址或其他网络 CIDR 的逗号分隔列表。在域前面加上 **.** 以仅匹配子域。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。如果您已将 Amazon **EC2**、**Elastic Load Balancing** 和 **S3** VPC 端点添加到 VPC 中，您必须将这些端点添加到 **noProxy** 字段。

- 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，其包含代理 HTTPS 连接所需的一个或多个额外 CA 证书。然后，Cluster Network
- 5 可选：决定 **Proxy** 对象的配置以引用 **trustedCA** 字段中 **user-ca-bundle** 配置映射的策略。允许的值是 **Proxyonly** 和 **Always**。仅在配置了 **http/https** 代理时，使用 **Proxyonly** 引用 **user-ca-bundle** 配置映射。使用 **Always** 始终引用 **user-ca-bundle** 配置映射。默认值为 **Proxyonly**。



#### 注意

安装程序不支持代理的 **readinessEndpoints** 字段。



#### 注意

如果安装程序超时，重启并使用安装程序的 **wait-for** 命令完成部署。例如：

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

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。

## 6.7. 通过下载二进制文件安装 OPENSIFT CLI

您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

### 6.7.1. 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. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。

## 5. 解包存档：

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

6. 将 **oc** 二进制文件放到 **PATH** 中的目录中。

要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

## 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

## 6.7.2. 在 Windows 上安装 OpenShift CLI

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

## 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
4. 使用 ZIP 程序解压存档。
5. 将 **oc** 二进制文件移到 **PATH** 中的目录中。  
要查看您的 **PATH**，请打开命令提示并执行以下命令：

```
C:\> path
```

## 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

## 6.7.3. 在 macOS 上安装 OpenShift CLI

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

## 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。

**注意**

对于 macOS arm64, 请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

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**, 请打开终端并执行以下命令：

```
$ echo $PATH
```

**验证**

- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

## 6.8. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案

默认情况下, 管理员 secret 存储在 **kube-system** 项目中。如果您在 **install-config.yaml** 文件中将 **credentialsMode** 参数配置为 **Manual**, 则必须使用以下替代方案之一：

- 要手动管理长期云凭证, 请按照[手动创建长期凭证](#)中的步骤操作。
- 要实现在集群外为各个组件管理的短期凭证, 请按照[配置 AWS 集群以使用短期凭证](#)中的步骤操作。

### 6.8.1. 手动创建长期凭证

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. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**, 请修改值, 如下所示：

**配置文件片段示例**

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件, 请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令, 使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

- 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 自定义资源 (CR) 列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \ ❶
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \ ❷
  --to=<path_to_directory_for_credentials_requests> \ ❸
```

- ❶ **--included** 参数仅包含特定集群配置所需的清单。
- ❷ 指定 **install-config.yaml** 文件的位置。
- ❸ 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

此命令为每个 **CredentialsRequest** 对象创建一个 YAML 文件。

### CredentialsRequest 对象示例

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
...
spec:
  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。

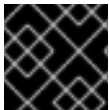
### 带有 secret 的 CredentialsRequest 对象示例

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
...
```

```
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
      - effect: Allow
        action:
          - s3:CreateBucket
          - s3:DeleteBucket
        resource: "*"
    ...
  secretRef:
    name: <component_secret>
    namespace: <component_namespace>
  ...
```

### Secret 对象示例

```
apiVersion: v1
kind: Secret
metadata:
  name: <component_secret>
  namespace: <component_namespace>
data:
  aws_access_key_id: <base64_encoded_aws_access_key_id>
  aws_secret_access_key: <base64_encoded_aws_secret_access_key>
```



#### 重要

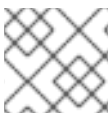
在升级使用手动维护凭证的集群前，您必须确保 CCO 处于可升级状态。

## 6.8.2. 将 AWS 集群配置为使用短期凭证

要安装配置为使用 AWS 安全令牌服务 (STS) 的集群，您必须配置 CCO 实用程序并为集群创建所需的 AWS 资源。

### 6.8.2.1. 配置 Cloud Credential Operator 工具

当 Cloud Credential Operator (CCO) 以手动模式运行时，要从集群外部创建和管理云凭证，提取并准备 CCO 实用程序 (**ccctl**) 二进制文件。



#### 注意

**ccctl** 工具是在 Linux 环境中运行的 Linux 二进制文件。

#### 先决条件

- 您可以访问具有集群管理员权限的 OpenShift Container Platform 帐户。
- 已安装 OpenShift CLI (**oc**)。
- 您已为 **ccctl** 工具创建了用于以下权限的 AWS 帐户：

#### 例 6.1. 所需的 AWS 权限

### 所需的 iam 权限

- **iam:CreateOpenIDConnectProvider**
- **iam:CreateRole**
- **iam>DeleteOpenIDConnectProvider**
- **iam>DeleteRole**
- **iam>DeleteRolePolicy**
- **iam:GetOpenIDConnectProvider**
- **iam:GetRole**
- **iam:GetUser**
- **iam:ListOpenIDConnectProviders**
- **iam:ListRolePolicies**
- **iam:ListRoles**
- **iam:PutRolePolicy**
- **iam:TagOpenIDConnectProvider**
- **iam:TagRole**

### 所需的 s3 权限

- **s3:CreateBucket**
- **s3>DeleteBucket**
- **s3>DeleteObject**
- **s3:GetBucketAcl**
- **s3:GetBucketTagging**
- **s3:GetObject**
- **s3:GetObjectAcl**
- **s3:GetObjectTagging**
- **s3:ListBucket**
- **s3:PutBucketAcl**
- **s3:PutBucketPolicy**
- **s3:PutBucketPublicAccessBlock**

- **s3:PutBucketTagging**
- **s3:PutObject**
- **s3:PutObjectAcl**
- **s3:PutObjectTagging**

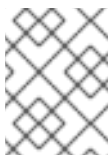
所需的 **cloudfront** 权限

- **cloudfront:ListCloudFrontOriginAccessIdentities**
- **cloudfront:ListDistributions**
- **cloudfront:ListTagsForResource**

如果您计划通过公共 CloudFront 发行版 URL 将 OIDC 配置存储在 IAM 身份提供程序访问的私有 S3 存储桶中，则运行 **ccctl** 工具的 AWS 帐户需要以下额外权限：

#### 例 6.2. 使用 CloudFront 私有 S3 存储桶的额外权限

- **cloudfront:CreateCloudFrontOriginAccessIdentity**
- **cloudfront:CreateDistribution**
- **cloudfront>DeleteCloudFrontOriginAccessIdentity**
- **cloudfront>DeleteDistribution**
- **cloudfront:GetCloudFrontOriginAccessIdentity**
- **cloudfront:GetCloudFrontOriginAccessIdentityConfig**
- **cloudfront:GetDistribution**
- **cloudfront:TagResource**
- **cloudfront:UpdateDistribution**



#### 注意

在使用 **ccctl aws create-all** 命令处理凭证请求时，这些额外权限支持使用 **--create-private-s3-bucket** 选项。

## 流程

1. 运行以下命令，为 OpenShift Container Platform 发行镜像设置变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像获取 CCO 容器镜像：

■



```
$ CCO_IMAGE=$(oc adm release info --image-for='cloud-credential-operator'
$RELEASE_IMAGE -a ~/.pull-secret)
```



### 注意

确保 **\$RELEASE\_IMAGE** 的架构与将使用 **ccoctl** 工具的环境架构相匹配。

- 运行以下命令，将 CCO 容器镜像中的 **ccoctl** 二进制文件提取到 OpenShift Container Platform 发行镜像中：

```
$ oc image extract $CCO_IMAGE --file="/usr/bin/ccoctl" -a ~/.pull-secret
```

- 运行以下命令更改权限以使 **ccoctl** 可执行：

```
$ chmod 775 ccoctl
```

### 验证

- 要验证 **ccoctl** 是否准备就绪，可以尝试显示帮助文件。运行命令时使用相对文件名，例如：

```
$ ./ccoctl.rhel9
```

### 输出示例

```
OpenShift credentials provisioning tool

Usage:
  ccoctl [command]

Available Commands:
  alibabacloud  Manage credentials objects for alibaba cloud
  aws           Manage credentials objects for AWS cloud
  azure         Manage credentials objects for Azure
  gcp           Manage credentials objects for Google cloud
  help          Help about any command
  ibmcloud      Manage credentials objects for IBM Cloud
  nutanix       Manage credentials objects for Nutanix

Flags:
  -h, --help  help for ccoctl

Use "ccoctl [command] --help" for more information about a command.
```

#### 6.8.2.2. 使用 Cloud Credential Operator 实用程序创建 AWS 资源

创建 AWS 资源时有以下选项：

- 您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。这是创建资源的最快速方法。请参阅 [使用单个命令创建 AWS 资源](#)。

- 如果您需要在修改 AWS 资源前查看 **ccctl** 工具创建的 JSON 文件，或者 **ccctl** 工具用于创建 AWS 资源的过程无法自动满足组织的要求，您可以单独创建 AWS 资源。请参阅 [单独创建 AWS 资源](#)。

#### 6.8.2.2.1. 使用单个命令创建 AWS 资源

如果 **ccctl** 工具用于创建 AWS 资源的过程自动满足机构的要求，您可以使用 **ccctl aws create-all** 命令自动创建 AWS 资源。

否则，您可以单独创建 AWS 资源。如需更多信息，请参阅“单独创建 AWS 资源”。



#### 注意

默认情况下，**ccctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccctl\_output\_dir>** 来引用这个目录。

#### 先决条件

您必须：

- 提取并准备好 **ccctl** 二进制文件。

#### 流程

1. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included 1 \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml 2 \
  --to=<path_to_directory_for_credentials_requests> 3
```

- 1** **--included** 参数仅包含特定集群配置所需的清单。
- 2** 指定 **install-config.yaml** 文件的位置。
- 3** 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。



#### 注意

此命令可能需要一些时间才能运行。

3. 运行以下命令，使用 **ccctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccctl aws create-all \
```

```
--name=<name> \ ❶
--region=<aws_region> \ ❷
--credentials-requests-dir=<path_to_credentials_requests_directory> \ ❸
--output-dir=<path_to_ccoctl_output_dir> \ ❹
--create-private-s3-bucket ❺
```

- ❶ 指定用于标记创建用于跟踪的任何云资源的名称。
- ❷ 指定在其中创建云资源的 AWS 区域。
- ❸ 指定包含组件 **CredentialsRequest** 对象文件的目录。
- ❹ 可选：指定您希望 **ccoctl** 实用程序在其中创建对象的目录。默认情况下，实用程序在运行命令的目录中创建对象。
- ❺ 可选：默认情况下，**ccoctl** 实用程序将 OpenID Connect (OIDC) 配置文件存储在公共 S3 存储桶中，并使用 S3 URL 作为公共 OIDC 端点。要将 OIDC 配置存储在 IAM 身份提供程序通过公共 CloudFront 发行版 URL 访问的专用 S3 存储桶中，请使用 **--create-private-s3-bucket** 参数。



### 注意

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

## 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccoctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccoctl_output_dir>/manifests
```

## 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

### 6.8.2.2.2. 单独创建 AWS 资源

您可以使用 **ccoctl** 工具单独创建 AWS 资源。这个选项对于在不同用户或部门之间创建这些资源的组织可能很有用。

否则，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。如需更多信息，请参阅“使用单个命令创建 AWS 资源”。



### 注意

默认情况下，**ccoctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccoctl\_output\_dir>** 来引用这个目录。

有些 **ccoctl** 命令会发出 AWS API 调用来创建或修改 AWS 资源。您可以使用 **--dry-run** 标志来避免 API 调用。使用此标志可在本地文件系统中创建 JSON 文件。您可以使用 **--cli-input-json** 参数查看和修改 JSON 文件，然后使用 AWS CLI 工具应用它们。

### 先决条件

- 提取并准备 **ccoctl** 二进制文件。

### 流程

- 运行以下命令，生成用于为集群设置 OpenID Connect 供应商的公共和私有 RSA 密钥文件：

```
$ ccoctl aws create-key-pair
```

#### 输出示例

```
2021/04/13 11:01:02 Generating RSA keypair
2021/04/13 11:01:03 Writing private key to /<path_to_ccoctl_output_dir>/serviceaccount-signer.private
2021/04/13 11:01:03 Writing public key to /<path_to_ccoctl_output_dir>/serviceaccount-signer.public
2021/04/13 11:01:03 Copying signing key for use by installer
```

其中 **serviceaccount-signer.private** 和 **serviceaccount-signer.public** 是生成的密钥文件。

此命令还会在 **/<path\_to\_ccoctl\_output\_dir>/tls/bound-service-account-signing-key.key** 中创建集群在安装过程中所需的私钥。

- 运行以下命令，在 AWS 上创建 OpenID Connect 身份提供程序和 S3 存储桶：

```
$ ccoctl aws create-identity-provider \
  --name=<name> \ ❶
  --region=<aws_region> \ ❷
  --public-key-file=<path_to_ccoctl_output_dir>/serviceaccount-signer.public ❸
```

❶ **<name>** 是用于标记为跟踪而创建的云资源的名称。

❷ **<aws-region>** 是将在其中创建云资源的 AWS 区域。

❸ **<path\_to\_ccoctl\_output\_dir>** 是 **ccoctl aws create-key-pair** 命令生成的公钥文件的路径。

#### 输出示例

```

2021/04/13 11:16:09 Bucket <name>-oidc created
2021/04/13 11:16:10 OpenID Connect discovery document in the S3 bucket <name>-oidc at
.well-known/openid-configuration updated
2021/04/13 11:16:10 Reading public key
2021/04/13 11:16:10 JSON web key set (JWKS) in the S3 bucket <name>-oidc at keys.json
updated
2021/04/13 11:16:18 Identity Provider created with ARN: arn:aws:iam::
<aws_account_id>:oidc-provider/<name>-oidc.s3.<aws_region>.amazonaws.com

```

其中 **openid-configuration** 是发现文档和 **key.json** 是一个 JSON Web 密钥集文件。

此命令还会在 `/<path_to_ccoctl_output_dir>/manifests/cluster-authentication-02-config.yaml` 中创建 YAML 配置文件。此文件为集群生成的服务帐户令牌设置签发者 URL 字段，以便 AWS IAM 身份提供程序信任令牌。

### 3. 为集群中的每个组件创建 IAM 角色：

- a. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/' {print $3})
```

- b. 从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```

$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \1
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
  \2
  --to=<path_to_directory_for_credentials_requests> \3

```

**1** **--included** 参数仅包含特定集群配置所需的清单。

**2** 指定 **install-config.yaml** 文件的位置。

**3** 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

- c. 运行以下命令，使用 **ccoctl** 工具处理所有 **CredentialsRequest** 对象：

```

$ ccoctl aws create-iam-roles \
  --name=<name> \
  --region=<aws_region> \
  --credentials-requests-dir=<path_to_credentials_requests_directory> \
  --identity-provider-arn=arn:aws:iam::<aws_account_id>:oidc-provider/<name>-oidc.s3.
<aws_region>.amazonaws.com

```



### 注意

对于使用其他 IAM API 端点的 AWS 环境（如 GovCloud），还必须使用 **--region** 参数指定您的区域。

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

对于每个 **CredentialsRequest** 对象，**ccctl** 创建一个带有信任策略的 IAM 角色，该角色与指定的 OIDC 身份提供程序相关联，以及来自 OpenShift Container Platform 发行镜像的每个 **CredentialsRequest** 对象中定义的权限策略。

### 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccctl_output_dir>/manifests
```

### 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

### 6.8.2.3. 整合 Cloud Credential Operator 实用程序清单

要为单个组件在集群外实现短期安全凭证，您必须将创建 Cloud Credential Operator 实用程序 (**ccctl**) 的清单文件移到安装程序的正确目录中。

### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您已配置了 Cloud Credential Operator 实用程序 (**ccctl**)。
- 已使用 **ccctl** 工具创建了集群所需的云供应商资源。

### 流程

- 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

- 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

- 运行以下命令，将 **ccocctl** 工具生成的清单复制到安装程序创建的 **manifests** 目录中：

```
$ cp /<path_to_ccocctl_output_dir>/manifests/* ./manifests/
```

- 将包含私钥的 **tls** 目录复制到安装目录中：

```
$ cp -a /<path_to_ccocctl_output_dir>/tls .
```

## 6.9. 部署集群

您可以在兼容云平台上安装 OpenShift Container Platform。



### 重要

在初始安装过程中，您只能运行安装程序的 **create cluster** 命令一次。

### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。
- 已确认主机上的云供应商帐户具有部署集群的正确权限。权限不正确的帐户会导致安装过程失败，并显示包括缺失权限的错误消息。

### 流程

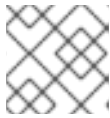
- 进入包含安装程序的目录并初始化集群部署：

```
$ ./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**，而不是 **info**。

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



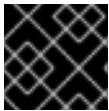
### 注意

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

## 验证

当集群部署成功完成时：

- 终端会显示用于访问集群的说明，包括指向 Web 控制台和 **kubeadmin** 用户的凭证的链接。
- 凭证信息还会输出到 `<installation_directory>/openshift_install.log`。

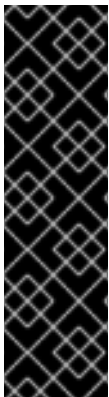


### 重要

不要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。

## 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: "password"
INFO Time elapsed: 36m22s
```



### 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。一个例外是，您必须手动批准待处理的 **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。

## 6.10. 使用 CLI 登录集群

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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

### 先决条件

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

### 流程



1. 导出 **kubeadmin** 凭证：

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

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

2. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

#### 输出示例

```
system:admin
```

## 6.11. 禁用默认的 OPERATORHUB 目录源

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 → Configuration → OperatorHub** 页面中，点 **Sources** 选项卡，您可以在其中创建、更新、删除、禁用和启用单独的源。

## 6.12. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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](#)

## 6.13. 后续步骤

- [验证安装](#)。
- [自定义集群](#)。

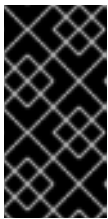
- 为 Cluster Samples Operator 和 **must-gather** 工具 [配置镜像流](#)。
- [在受限网络中使用 Operator Lifecycle Manager \(OLM\)](#)
- [配置额外的信任存储](#)
- [远程健康报告](#)

## 第 7 章 在 AWS 上将集群安装到现有的 VPC 中

在 OpenShift Container Platform 版本 4.14 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，您可以进一步自定义这些基础架构。要自定义安装，请在安装集群前修改 `install-config.yaml` 文件中的参数。

### 7.1. 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。
- 您可以阅读[选择集群安装方法并为用户准备它](#)的文档。
- 已将 [AWS 帐户配置](#)为托管集群。
- 如果现有的 VPC 由与集群不同的帐户所有，您可以在帐户间[共享 VPC](#)。



#### 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，[将其配置为允许集群需要访问的站点](#)。

### 7.2. 关于使用自定义 VPC

在 OpenShift Container Platform 4.14 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。

#### 7.2.1. 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。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](#) 以及 [处理 VPC 和子网](#)。

安装程序无法：

- 分割网络范围供集群使用。
- 设置子网的路由表。
- 设置 VPC 选项，如 DHCP。

您必须在安装集群前完成这些任务。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 [您的 VPC 的路由表](#)。

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私有子网。有关此类配置的示例，请参阅 AWS 文档中的 [具有公共和私有子网\(NAT\)的 VPC](#)。记录每个子网 ID。完成安装要求您在 `install-config.yaml` 文件的 `platform` 部分输入这些值。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 必须附加一个公共互联网网关。对于每个可用区：
  - 公共子网需要路由到互联网网关。
  - 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`、`Name` 和 `openshift.io/cluster` 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。您不能使用 `Name` 标签，因为它与 EC2 `Name` 字段重叠，且安装失败。
- 您需要在您的 VPC 中启用 `enableDnsSupport` 和 `enableDnsHostnames` 属性，以便集群可以使用附加到 VPC 中的 Route 53 区来解析集群内部的 DNS 记录。请参阅 AWS 文档中的 [您的 VPC 中的 DNS 支持](#) 部分。  
如果要使用您自己的 Route 53 托管私有区，您必须在安装集群前将现有托管区与 VPC 相关联。您可以使用 `install-config.yaml` 文件中的 `platform.aws.hostedZone` 和 `platform.aws.hostedZoneRole` 字段定义托管区。您可以通过与安装集群的帐户共享来使用来自另一个帐户的私有托管区。如果使用另一个帐户的私有托管区，则必须使用 `Passthrough` 或 `Manual` 凭证模式。

如果您在断开连接的环境中工作，则无法访问 EC2、ELB 和 S3 端点的公共 IP 地址。根据您要在安装过程中限制互联网流量的级别，有以下配置选项：

### 7.2.1.1. 选项 1：创建 VPC 端点

创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

通过这个选项，网络流量在 VPC 和所需的 AWS 服务之间保持私有。

#### 7.2.1.2. 选项 2：创建一个没有 VPC 端点的代理

作为安装过程的一部分，您可以配置 HTTP 或 HTTPS 代理。使用此选项时，互联网流量会通过代理访问所需的 AWS 服务。

#### 7.2.1.3. 选项 3：创建带有 VPC 端点的代理

作为安装过程的一部分，您可以使用 VPC 端点配置 HTTP 或 HTTPS 代理。创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

在 `install-config.yaml` 文件中配置代理时，将这些端点添加到 `noProxy` 字段。通过这个选项，代理会阻止集群直接访问互联网。但是，您的 VPC 和所需的 AWS 服务之间网络流量保持私有。

### 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：                                                                                                   |             |
|        |                                                                                                                                                                                                                                                                                                                                       | 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。                                   |             |

7.2.2. 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。

### 7.2.3. 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。

### 7.2.4. 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)。

### 7.2.5. AWS 安全组

默认情况下，安装程序会创建安全组并将其附加到 control plane 和计算机器。不可修改与默认安全组关联的规则。

但是，您可以将与现有 VPC 关联的其他现有 AWS 安全组应用到 control plane 和计算机器。应用自定义安全组可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

作为安装过程的一部分，您可以在部署集群前通过修改 **install-config.yaml** 文件来应用自定义安全组。

如需更多信息，请参阅“将现有 AWS 安全组应用到集群”。

### 7.2.6. 在安装到共享 VPC 时修改信任策略

如果使用共享 VPC 安装集群，您可以使用 **Passthrough** 或 **Manual** 凭证模式。您必须在拥有 VPC 的帐户的信任策略中添加用于安装集群的 IAM 角色作为主体。

如果使用 **Passthrough** 模式，请将创建集群的帐户的 Amazon 资源名称 (ARN)（如 **arn:aws:iam::123456789012:user/clustercreator**）添加到信任策略作为主体。

如果使用 **Manual** 模式，请将创建集群的帐户的 ARN，以及集群所有者帐户中的 ingress operator 角色的 ARN，如 `arn:aws:iam::123456789012:role/<cluster-name>-openshift-ingress-operator-cloud-credentials`，添加到信任策略作为主体。

您必须在策略中添加以下操作：

#### 例 7.1. 共享 VPC 安装所需的操作

- `route53:ChangeResourceRecordSets`
- `route53:ListHostedZones`
- `route53:ListHostedZonesByName`
- `route53:ListResourceRecordSets`
- `route53:ChangeTagsForResource`
- `route53:GetAccountLimit`
- `route53:GetChange`
- `route53:GetHostedZone`
- `route53:ListTagsForResource`
- `route53:UpdateHostedZoneComment`
- `tag:GetResources`
- `tag:UntagResources`

## 7.3. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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型的基础架构上执行受限网络安装。在此过程中，您可以下载所需的内容，并使用它为镜像 registry 填充安装软件包。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。在更新集群前，您要更新镜像 registry 的内容。

## 7.4. 为集群节点 SSH 访问生成密钥对



在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 **core** 用户的 `~/.ssh/authorized_keys` 列表中，这将启用免密码身份验证。

将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。

如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。`./openshift-install gather` 命令还需要在集群节点上设置 SSH 公钥。



### 重要

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。



### 注意

您必须使用本地密钥，而不是使用特定平台方法配置 [的密钥](#)，如 [AWS 密钥对](#)。

## 流程

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，请创建一个。例如，在使用 Linux 操作系统的计算机上运行以下命令：

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

- ❶ 指定新 SSH 密钥的路径和文件名，如 `~/.ssh/id_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。



### 注意

如果您计划在 **x86\_64**、**ppc64le** 和 **s390x** 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 **ed25519** 算法的密钥。相反，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

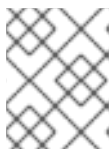
2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

3. 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。



### 注意

在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- a. 如果 **ssh-agent** 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

#### 输出示例

```
Agent pid 31874
```



#### 注意

如果集群处于 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。

4. 将 SSH 私钥添加到 **ssh-agent**：

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

- 1 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

#### 输出示例

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

#### 后续步骤

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

## 7.5. 获取安装程序

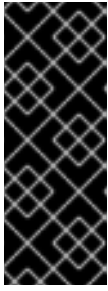
在安装 OpenShift Container Platform 前，将安装文件下载到您用于安装的主机上。

#### 先决条件

- 您有一台运行 Linux 或 macOS 的计算机，至少有 1.2 GB 本地磁盘空间。

#### 流程

1. 进入 Red Hat Hybrid Cloud Console 上的 [Cluster Type](#) 页。如果您有红帽帐户，请使用您的凭证登录。如果没有，请创建一个帐户。
2. 在页的 **Run it yourself** 部分中选择您的基础架构供应商。
3. 从 **OpenShift 安装程序** 下的下拉菜单中选择您的主机操作系统和架构，然后点**下载安装程序**。
4. 将下载的文件保存在要存储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几个文件。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。删除集群需要这两个文件。
- 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，请为特定云供应商完成 OpenShift Container Platform 卸载流程。

5. 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
```

6. 从 [Red Hat OpenShift Cluster Manager](#) 下载安装 pull secret。此 pull secret 允许您与所含授权机构提供的服务进行身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

## 提示

另外，您还可以从[红帽客户门户网站](#)检索安装程序，您可以在其中指定要下载的安装程序版本。但是，您需要有一个有效的订阅才能访问此页。

## 7.6. 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

- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。

### 流程

1. 创建 **install-config.yaml** 文件。

a. 进入包含安装程序的目录并运行以下命令：

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

**1** 对于 **<installation\_directory>**，请指定要存储安装程序创建的文件目录名称。

在指定目录时：

- 验证该目录是否具有执行权限。在安装目录中运行 Terraform 二进制文件需要这个权限。
- 使用空目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。



注意

始终删除 `~/.powervs` 目录，以避免重复使用过时的配置。运行以下命令：

```
$ rm -rf ~/.powervs
```

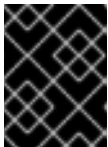
- b. 在提示符处，提供云的配置详情：
  - i. 可选：选择用于访问集群机器的 SSH 密钥。



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. 为集群输入描述性名称。
- 2. 修改 **install-config.yaml** 文件。您可以在“安装配置参数”部分找到有关可用参数的更多信息。
- 3. 备份 **install-config.yaml** 文件，以便您可以使用它安装多个集群。



重要

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

其他资源

- [AWS 的安装配置参数](#)

7.6.1. 集群安装的最低资源要求

每台集群机器都必须满足以下最低要求：

表 7.1. 最低资源要求

| 机器        | 操作系统  | vCPU [1] | 虚拟内存  | Storage | 每秒输入/输出 (IOPS) [2] |
|-----------|-------|----------|-------|---------|--------------------|
| bootstrap | RHCOS | 4        | 16 GB | 100 GB  | 300                |

| 机器                   | 操作系统                     | vCPU [1] | 虚拟内存  | Storage | 每秒输入/输出 (IOPS) [2] |
|----------------------|--------------------------|----------|-------|---------|--------------------|
| Control plane (控制平面) | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Compute              | RHCOS、RHEL 8.6 及更新版本 [3] | 2        | 8 GB  | 100 GB  | 300                |

1. 当未启用并发多线程 (SMT) 或超线程时，一个 vCPU 相当于一个物理内核。启用后，使用以下公式来计算对应的比例：（每个内核数的线程）× sockets = vCPU。
2. OpenShift Container Platform 和 Kubernetes 对磁盘性能非常敏感，建议使用更快的存储速度，特别是 control plane 节点上需要 10 ms p99 fsync 持续时间的 etcd。请注意，在许多云平台上，存储大小和 IOPS 可一起扩展，因此您可能需要过度分配存储卷来获取足够的性能。
3. 与所有用户置备的安装一样，如果您选择在集群中使用 RHEL 计算机器，则负责所有操作系统生命周期管理和维护，包括执行系统更新、应用补丁和完成所有其他必要的任务。RHEL 7 计算机器的使用已弃用，并已在 OpenShift Container Platform 4.10 及更新的版本中删除。

### 注意

从 OpenShift Container Platform 版本 4.13 开始，RHCOS 基于 RHEL 版本 9.2，它更新了微架构要求。以下列表包含每个架构需要的最小指令集架构 (ISA)：

- x86-64 体系结构需要 x86-64-v2 ISA
- ARM64 架构需要 ARMv8.0-A ISA
- IBM Power 架构需要 Power 9 ISA
- s390x 架构需要 z14 ISA

如需更多信息，请参阅 [RHEL 架构](#)。

如果平台的实例类型满足集群机器的最低要求，则 OpenShift Container Platform 支持使用它。

### 其他资源

- [优化存储](#)

## 7.6.2. 为 AWS 测试的实例类型

以下 Amazon Web Services(AWS) 实例类型已经过 OpenShift Container Platform 测试。

### 注意

将以下图中包含的机器类型用于 AWS 实例。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

### 例 7.2. 基于 64 位 x86 架构的机器类型

- c4.\*
- c5.\*
- c5a.\*
- i3.\*
- m4.\*
- m5.\*
- m5a.\*
- m6i.\*
- r4.\*
- r5.\*
- r5a.\*
- r6i.\*
- t3.\*
- t3a.\*

### 7.6.3. 在 64 位 ARM 基础架构上为 AWS 测试过的实例类型

OpenShift Container Platform 中已经测试了以下 Amazon Web Services (AWS) 64 位 ARM 实例类型。



#### 注意

使用 AWS ARM 实例的以下图中包含的机器类型。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

### 例 7.3. 基于 64 位 ARM 架构的机器类型

- c6g.\*
- m6g.\*
- r8g.\*

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

您可以自定义安装配置文件 (**install-config.yaml**)，以指定有关 OpenShift Container Platform 集群平台的更多详细信息，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，并进行修改。

```

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
      metadataService:
        authentication: Optional 7
        type: m6i.xlarge
      replicas: 3
compute: 8
  - hyperthreading: Enabled 9
    name: worker
    platform:
      aws:
        rootVolume:
          iops: 2000
          size: 500
          type: io1 10
        metadataService:
          authentication: Optional 11
          type: c5.4xlarge
        zones:
          - us-west-2c
        replicas: 3
metadata:
  name: test-cluster 12
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OVNKubernetes 13
  serviceNetwork:
    - 172.30.0.0/16
platform:
  aws:
    region: us-west-2 14

```

```

propagateUserTags: true 15
userTags:
  adminContact: jdoe
  costCenter: 7536
subnets: 16
- subnet-1
- subnet-2
- subnet-3
amiID: ami-0c5d3e03c0ab9b19a 17
serviceEndpoints: 18
- name: ec2
  url: https://vpce-id.ec2.us-west-2.vpce.amazonaws.com
hostedZone: Z3URY6TWQ91KVV 19
fips: false 20
sshKey: ssh-ed25519 AAAA... 21
pullSecret: '{"auths": ...}' 22

```

1 12 14 22 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。默认情况下，CCO 使用 **kube-system** 命名空间中的 root 凭证来动态尝试决定凭证的功能。有关 CCO 模式的详情，请参阅 [身份验证和授权指南](#) 中的 "About the Cloud Credential Operator" 部分。

3 8 15 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。仅使用一个 control plane 池。

5 9 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置为 **Disabled** 来禁用它。如果在某些集群机器中禁用并发多线程，则必须在所有集群机器中禁用它。



### 重要

如果您禁用并发多线程，请确保您的容量规划考虑机器性能显著降低的情况。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 10 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。

7 11 是否需要 [Amazon EC2 实例元数据服务 v2](#) (IMDSv2)。为了要求 IMDSv2，请将参数值设置为 **Required**。要允许使用 IMDSv1 和 IMDSv2，请将参数值设置为 **Optional**。如果没有指定值，则允许 IMDSv1 和 IMDSv2。



### 注意

在集群安装过程中设置的 control plane 机器的 IMDS 配置只能使用 AWS CLI 更改。可以使用计算机器集来更改计算机器的 IMDS 配置。

13 要安装的集群网络插件。支持的值有 **OVNKubernetes** 和 **OpenShiftSDN**。默认值为 **OVNKubernetes**。



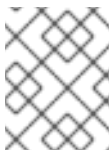
- 16 如果您提供自己的 VPC，为集群使用的每个可用区指定子网。
- 17 用于为集群引导机器的 AMI ID。如果设置，AMI 必须属于与集群相同的区域。
- 18 AWS 服务端点。在安装到未知 AWS 区域时，需要自定义端点。端点 URL 必须使用 **https** 协议，主机必须信任该证书。
- 19 您现有 Route 53 私有托管区的 ID。提供现有的托管区需要您提供自己的 VPC，托管区已在安装集群前与 VPC 关联。如果未定义，安装程序会创建一个新的托管区。
- 20 是否启用或禁用 FIPS 模式。默认情况下不启用 FIPS 模式。如果启用了 FIPS 模式，运行 OpenShift 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

要为集群启用 FIPS 模式，您必须从配置为以 FIPS 模式操作的 Red Hat Enterprise Linux (RHEL) 计算机运行安装程序。有关在 RHEL 中配置 FIPS 模式的更多信息，请参阅[在 FIPS 模式中安装该系统](#)。当以 FIPS 模式运行 Red Hat Enterprise Linux (RHEL) 或 Red Hat Enterprise Linux CoreOS (RHCOS) 时，OpenShift Container Platform 核心组件使用 RHEL 加密库，在 x86\_64、ppc64le 和 s390x 架构上提交到 NIST FIPS 140-2/140-3 Validation。

- 21 您可以选择提供您用来访问集群中机器的 **sshKey** 值。



### 注意

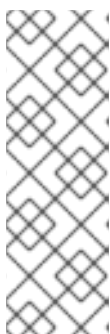
对于您要在其上执行安装调试或灾难恢复的生产环境 OpenShift Container Platform 集群，请指定 **ssh-agent** 进程使用的 SSH 密钥。

## 7.6.5. 在安装过程中配置集群范围的代理

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，并确定它们中的任何站点是否需要绕过代理。默认情况下，所有集群出口流量都经过代理，包括对托管云供应商 API 的调用。如果需要，您将在 **Proxy** 对象的 **spec.noProxy** 字段中添加站点来绕过代理。



### 注意

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

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

## 流程

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> ❶
  httpsProxy: https://<username>:<pswd>@<ip>:<port> ❷
  noProxy: ec2.<aws_region>.amazonaws.com,elasticloadbalancing.
    <aws_region>.amazonaws.com,s3.<aws_region>.amazonaws.com ❸
  additionalTrustBundle: | ❹
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> ❺
```

- ❶ 用于创建集群外 HTTP 连接的代理 URL。URL 方案必须是 **http**。
- ❷ 用于创建集群外 HTTPS 连接的代理 URL。
- ❸ 要从代理中排除的目标域名、IP 地址或其他网络 CIDR 的逗号分隔列表。在域前面加上 **.** 以仅匹配子域。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。如果您已将 Amazon **EC2**、**Elastic Load Balancing** 和 **S3** VPC 端点添加到 VPC 中，您必须将这些端点添加到 **noProxy** 字段。
- ❹ 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，其包含代理 HTTPS 连接所需的一个或多个额外 CA 证书。然后，Cluster Network Operator 会创建 **trusted-ca-bundle** 配置映射，将这些内容与 Red Hat Enterprise Linux CoreOS (RHCOS) 信任捆绑包合并，**Proxy** 对象的 **trustedCA** 字段中也会引用此配置映射。**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。
- ❺ 可选：决定 **Proxy** 对象的配置以引用 **trustedCA** 字段中 **user-ca-bundle** 配置映射的策略。允许的值是 **Proxyonly** 和 **Always**。仅在配置了 **http/https** 代理时，使用 **Proxyonly** 引用 **user-ca-bundle** 配置映射。使用 **Always** 始终引用 **user-ca-bundle** 配置映射。默认值为 **Proxyonly**。



### 注意

安装程序不支持代理的 **readinessEndpoints** 字段。



### 注意

如果安装程序超时，重启并使用安装程序的 **wait-for** 命令完成部署。例如：

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

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。

### 7.6.6. 将现有 AWS 安全组应用到集群

将现有 AWS 安全组应用到 control plane 和计算机器可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

#### 先决条件

- 您已在 AWS 中创建安全组。如需更多信息，请参阅使用 [安全组的 AWS 文档](#)。
- 安全组必须与您要将集群部署到的现有 VPC 关联。安全组不能与另一个 VPC 关联。
- 您有一个现有的 **install-config.yaml** 文件。

#### 流程

1. 在 **install-config.yaml** 文件中，编辑 **compute.platform.aws.additionalSecurityGroupIDs** 参数，为您的计算机器指定一个或多个自定义安全组。
2. 编辑 **controlPlane.platform.aws.additionalSecurityGroupIDs** 参数，为您的 control plane 机器指定一个或多个自定义安全组。
3. 保存文件并在部署集群时引用。

#### 指定自定义安全组的 **install-config.yaml** 文件示例

```
# ...
compute:
- hyperthreading: Enabled
  name: worker
  platform:
    aws:
      additionalSecurityGroupIDs:
        - sg-1 ❶
        - sg-2
  replicas: 3
controlPlane:
  hyperthreading: Enabled
  name: master
  platform:
    aws:
      additionalSecurityGroupIDs:
        - sg-3
        - sg-4
  replicas: 3
platform:
  aws:
    region: us-east-1
    subnets: ❷
    - subnet-1
    - subnet-2
    - subnet-3
```

- 1 在 Amazon EC2 控制台中显示时指定安全组的名称，包括 **sg** 前缀。
- 2 指定集群使用的每个可用区的子网。

## 7.7. 通过下载二进制文件安装 OPENSIFT CLI

您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

### 7.7.1. 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. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。
5. 解包存档：

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

6. 将 **oc** 二进制文件放到 **PATH** 中的目录中。  
要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 7.7.2. 在 Windows 上安装 OpenShift CLI

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

#### 流程

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

2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
4. 使用 ZIP 程序解压存档。
5. 将 **oc** 二进制文件移到 **PATH** 中的目录中。  
要查看您的 **PATH**，请打开命令提示并执行以下命令：

```
C:\> path
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 7.7.3. 在 macOS 上安装 OpenShift CLI

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

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。



#### 注意

对于 macOS arm64，请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

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**，请打开终端并执行以下命令：

```
$ echo $PATH
```

#### 验证

- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

### 7.8. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案

默认情况下，管理员 secret 存储在 **kube-system** 项目中。如果您在 **install-config.yaml** 文件中将 **credentialsMode** 参数配置为 **Manual**，则必须使用以下替代方案之一：

- 要手动管理长期云凭证，请按照[手动创建长期凭证](#)中的步骤操作。
- 要实现在集群外为各个组件管理的短期凭证，请按照[配置 AWS 集群以使用短期凭证](#)中的步骤操作。

### 7.8.1. 手动创建长期凭证

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. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

#### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/' {print $3})
```

4. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 自定义资源 (CR) 列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included ❶ \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml ❷ \
  --to=<path_to_directory_for_credentials_requests> ❸
```

**❶** **--included** 参数仅包含特定集群配置所需的清单。

**❷** 指定 **install-config.yaml** 文件的位置。

**❸** 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

此命令为每个 **CredentialsRequest** 对象创建一个 YAML 文件。

#### CredentialsRequest 对象示例

```

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

```

5. 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。

### 带有 secret 的 CredentialsRequest 对象示例

```

apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
  ...
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
      - effect: Allow
        action:
          - s3:CreateBucket
          - s3:DeleteBucket
        resource: "*"
    ...
  secretRef:
    name: <component_secret>
    namespace: <component_namespace>
  ...

```

### Secret 对象示例

```

apiVersion: v1
kind: Secret
metadata:
  name: <component_secret>
  namespace: <component_namespace>

```

```
data:
  aws_access_key_id: <base64_encoded_aws_access_key_id>
  aws_secret_access_key: <base64_encoded_aws_secret_access_key>
```



### 重要

在升级使用手动维护凭证的集群前，您必须确保 CCO 处于可升级状态。

## 7.8.2. 将 AWS 集群配置为使用短期凭证

要安装配置为使用 AWS 安全令牌服务 (STS) 的集群，您必须配置 CCO 实用程序并为集群创建所需的 AWS 资源。

### 7.8.2.1. 配置 Cloud Credential Operator 工具

当 Cloud Credential Operator (CCO) 以手动模式运行时，要从集群外部创建和管理云凭证，提取并准备 CCO 实用程序 (**ccctl**) 二进制文件。



### 注意

**ccctl** 工具是在 Linux 环境中运行的 Linux 二进制文件。

### 先决条件

- 您可以访问具有集群管理员权限的 OpenShift Container Platform 帐户。
- 已安装 OpenShift CLI (**oc**)。
- 您已为 **ccctl** 工具创建了用于以下权限的 AWS 帐户：

#### 例 7.4. 所需的 AWS 权限

##### 所需的 iam 权限

- **iam:CreateOpenIDConnectProvider**
- **iam:CreateRole**
- **iam>DeleteOpenIDConnectProvider**
- **iam>DeleteRole**
- **iam>DeleteRolePolicy**
- **iam:GetOpenIDConnectProvider**
- **iam:GetRole**
- **iam:GetUser**
- **iam:ListOpenIDConnectProviders**
- **iam:ListRolePolicies**
- **iam:ListRoles**



- **iam:PutRolePolicy**
- **iam:TagOpenIDConnectProvider**
- **iam:TagRole**

#### 所需的 s3 权限

- **s3:CreateBucket**
- **s3:DeleteBucket**
- **s3:DeleteObject**
- **s3:GetBucketAcl**
- **s3:GetBucketTagging**
- **s3:GetObject**
- **s3:GetObjectAcl**
- **s3:GetObjectTagging**
- **s3:ListBucket**
- **s3:PutBucketAcl**
- **s3:PutBucketPolicy**
- **s3:PutBucketPublicAccessBlock**
- **s3:PutBucketTagging**
- **s3:PutObject**
- **s3:PutObjectAcl**
- **s3:PutObjectTagging**

#### 所需的 cloudfront 权限

- **cloudfront:ListCloudFrontOriginAccessIdentities**
- **cloudfront:ListDistributions**
- **cloudfront:ListTagsForResource**

如果您计划通过公共 CloudFront 发行版 URL 将 OIDC 配置存储在 IAM 身份提供程序访问的私有 S3 存储桶中，则运行 **ccctl** 工具的 AWS 帐户需要以下额外权限：

#### 例 7.5. 使用 CloudFront 私有 S3 存储桶的额外权限

- **cloudfront:CreateCloudFrontOriginAccessIdentity**

- **cloudfront:CreateDistribution**
- **cloudfront:DeleteCloudFrontOriginAccessIdentity**
- **cloudfront:DeleteDistribution**
- **cloudfront:GetCloudFrontOriginAccessIdentity**
- **cloudfront:GetCloudFrontOriginAccessIdentityConfig**
- **cloudfront:GetDistribution**
- **cloudfront:TagResource**
- **cloudfront:UpdateDistribution**



### 注意

在使用 **ccoctl aws create-all** 命令处理凭证请求时，这些额外权限支持使用 **--create-private-s3-bucket** 选项。

## 流程

1. 运行以下命令，为 OpenShift Container Platform 发行镜像设置变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像获取 CCO 容器镜像：

```
$ CCO_IMAGE=$(oc adm release info --image-for='cloud-credential-operator'
$RELEASE_IMAGE -a ~/.pull-secret)
```



### 注意

确保 **\$RELEASE\_IMAGE** 的架构与将使用 **ccoctl** 工具的环境架构相匹配。

3. 运行以下命令，将 CCO 容器镜像中的 **ccoctl** 二进制文件提取到 OpenShift Container Platform 发行镜像中：

```
$ oc image extract $CCO_IMAGE --file="/usr/bin/ccoctl" -a ~/.pull-secret
```

4. 运行以下命令更改权限以使 **ccoctl** 可执行：

```
$ chmod 775 ccoctl
```

## 验证

- 要验证 **ccoctl** 是否准备就绪，可以尝试显示帮助文件。运行命令时使用相对文件名，例如：

```
$ ./ccoctl.rhel9
```

## 输出示例

```

OpenShift credentials provisioning tool

Usage:
  ccctl [command]

Available Commands:
  alibabacloud  Manage credentials objects for alibaba cloud
  aws           Manage credentials objects for AWS cloud
  azure         Manage credentials objects for Azure
  gcp           Manage credentials objects for Google cloud
  help          Help about any command
  ibmcloud      Manage credentials objects for IBM Cloud
  nutanix       Manage credentials objects for Nutanix

Flags:
  -h, --help  help for ccctl

Use "ccctl [command] --help" for more information about a command.

```

### 7.8.2.2. 使用 Cloud Credential Operator 实用程序创建 AWS 资源

创建 AWS 资源时有以下选项：

- 您可以使用 **ccctl aws create-all** 命令自动创建 AWS 资源。这是创建资源的最快速方法。请参阅 [使用单个命令创建 AWS 资源](#)。
- 如果您需要在修改 AWS 资源前查看 **ccctl** 工具创建的 JSON 文件，或者 **ccctl** 工具用于创建 AWS 资源的过程无法自动满足组织的要求，您可以单独创建 AWS 资源。请参阅 [单独创建 AWS 资源](#)。

#### 7.8.2.2.1. 使用单个命令创建 AWS 资源

如果 **ccctl** 工具用于创建 AWS 资源的过程自动满足机构的要求，您可以使用 **ccctl aws create-all** 命令自动创建 AWS 资源。

否则，您可以单独创建 AWS 资源。如需更多信息，请参阅“单独创建 AWS 资源”。



#### 注意

默认情况下，**ccctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccctl\_output\_dir>** 来引用这个目录。

## 先决条件

您必须：

- 提取并准备好 **ccctl** 二进制文件。

## 流程

1. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/' {print $3})
```

- 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \1
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \2
  --to=<path_to_directory_for_credentials_requests> \3
```

- included** 参数仅包含特定集群配置所需的清单。
- 指定 **install-config.yaml** 文件的位置。
- 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。



#### 注意

此命令可能需要一些时间才能运行。

- 运行以下命令，使用 **ccctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccctl aws create-all \
  --name=<name> \1
  --region=<aws_region> \2
  --credentials-requests-dir=<path_to_credentials_requests_directory> \3
  --output-dir=<path_to_ccctl_output_dir> \4
  --create-private-s3-bucket \5
```

- 指定用于标记创建用于跟踪的任何云资源的名称。
- 指定在其中创建云资源的 AWS 区域。
- 指定包含组件 **CredentialsRequest** 对象文件的目录。
- 可选：指定您希望 **ccctl** 实用程序在其中创建对象的目录。默认情况下，实用程序在运行命令的目录中创建对象。
- 可选：默认情况下，**ccctl** 实用程序将 OpenID Connect (OIDC) 配置文件存储在公共 S3 存储桶中，并使用 S3 URL 作为公共 OIDC 端点。要将 OIDC 配置存储在 IAM 身份提供程序通过公共 CloudFront 发行版 URL 访问的专用 S3 存储桶中，请使用 **--create-private-s3-bucket** 参数。



#### 注意

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

## 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 `<path_to_ccoctl_output_dir>/manifests` 目录中的文件：

```
$ ls <path_to_ccoctl_output_dir>/manifests
```

## 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

### 7.8.2.2.2. 单独创建 AWS 资源

您可以使用 **ccoctl** 工具单独创建 AWS 资源。这个选项对于在不同用户或部门之间创建这些资源的组织可能很有用。

否则，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。如需更多信息，请参阅“使用单个命令创建 AWS 资源”。



## 注意

默认情况下，**ccoctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 `<path_to_ccoctl_output_dir>` 来引用这个目录。

有些 **ccoctl** 命令会发出 AWS API 调用来创建或修改 AWS 资源。您可以使用 **--dry-run** 标志来避免 API 调用。使用此标志可在本地文件系统中创建 JSON 文件。您可以使用 **--cli-input-json** 参数查看和修改 JSON 文件，然后使用 AWS CLI 工具应用它们。

## 先决条件

- 提取并准备 **ccoctl** 二进制文件。

## 流程

- 运行以下命令，生成用于为集群设置 OpenID Connect 供应商的公共和私有 RSA 密钥文件：

```
$ ccoctl aws create-key-pair
```

## 输出示例

```
2021/04/13 11:01:02 Generating RSA keypair
2021/04/13 11:01:03 Writing private key to /<path_to_ccoctl_output_dir>/serviceaccount-signer.private
```

```
2021/04/13 11:01:03 Writing public key to /<path_to_ccoctl_output_dir>/serviceaccount-
signer.public
2021/04/13 11:01:03 Copying signing key for use by installer
```

其中 **serviceaccount-signer.private** 和 **serviceaccount-signer.public** 是生成的密钥文件。

此命令还会在 **/<path\_to\_ccoctl\_output\_dir>/tls/bound-service-account-signing-key.key** 中创建集群在安装过程中所需的私钥。

2. 运行以下命令，在 AWS 上创建 OpenID Connect 身份提供程序和 S3 存储桶：

```
$ ccoctl aws create-identity-provider \
  --name=<name> ❶
  --region=<aws_region> ❷
  --public-key-file=<path_to_ccoctl_output_dir>/serviceaccount-signer.public ❸
```

❶ **<name>** 是用于标记为跟踪而创建的云资源的名称。

❷ **<aws-region>** 是将在其中创建云资源的 AWS 区域。

❸ **<path\_to\_ccoctl\_output\_dir>** 是 **ccoctl aws create-key-pair** 命令生成的公钥文件的路径。

## 输出示例

```
2021/04/13 11:16:09 Bucket <name>-oidc created
2021/04/13 11:16:10 OpenID Connect discovery document in the S3 bucket <name>-oidc at
.well-known/openid-configuration updated
2021/04/13 11:16:10 Reading public key
2021/04/13 11:16:10 JSON web key set (JWKS) in the S3 bucket <name>-oidc at keys.json
updated
2021/04/13 11:16:18 Identity Provider created with ARN: arn:aws:iam::
<aws_account_id>:oidc-provider/<name>-oidc.s3.<aws_region>.amazonaws.com
```

其中 **openid-configuration** 是发现文档和 **key.json** 是一个 JSON Web 密钥集文件。

此命令还会在 **/<path\_to\_ccoctl\_output\_dir>/manifests/cluster-authentication-02-config.yaml** 中创建 YAML 配置文件。此文件为集群生成的服务帐户令牌设置签发者 URL 字段，以便 AWS IAM 身份提供程序信任令牌。

3. 为集群中的每个组件创建 IAM 角色：

- a. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

- b. 从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included ❶
```

```
--install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
```

2

```
--to=<path_to_directory_for_credentials_requests> 3
```

1

--included 参数仅包含特定集群配置所需的清单。

2

指定 **install-config.yaml** 文件的位置。

3

指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

c. 运行以下命令，使用 **ccoctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccoctl aws create-iam-roles \
  --name=<name> \
  --region=<aws_region> \
  --credentials-requests-dir=<path_to_credentials_requests_directory> \
  --identity-provider-arn=arn:aws:iam::<aws_account_id>:oidc-provider/<name>-oidc.s3.
  <aws_region>.amazonaws.com
```



### 注意

对于使用其他 IAM API 端点的 AWS 环境（如 GovCloud），还必须使用 **--region** 参数指定您的区域。

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

对于每个 **CredentialsRequest** 对象，**ccoctl** 创建一个带有信任策略的 IAM 角色，该角色与指定的 OIDC 身份提供程序相关联，以及来自 OpenShift Container Platform 发行镜像的每个 **CredentialsRequest** 对象中定义的权限策略。

## 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccoctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccoctl_output_dir>/manifests
```

## 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

### 7.8.2.3. 整合 Cloud Credential Operator 实用程序清单

要为单个组件在集群外实现短期安全凭证，您必须将创建 Cloud Credential Operator 实用程序 (**ccctl**) 的清单文件移到安装程序的正确目录中。

#### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您已配置了 Cloud Credential Operator 实用程序 (**ccctl**)。
- 已使用 **ccctl** 工具创建了集群所需的云供应商资源。

#### 流程

1. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

#### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，将 **ccctl** 工具生成的清单复制到安装程序创建的 **manifests** 目录中：

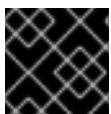
```
$ cp /<path_to_ccctl_output_dir>/manifests/* ./manifests/
```

4. 将包含私钥的 **tls** 目录复制到安装目录中：

```
$ cp -a /<path_to_ccctl_output_dir>/tls .
```

## 7.9. 部署集群

您可以在兼容云平台上安装 OpenShift Container Platform。



#### 重要

在初始安装过程中，您只能运行安装程序的 **create cluster** 命令一次。

#### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。



- 已确认主机上的云供应商帐户具有部署集群的正确权限。权限不正确的帐户会导致安装过程失败，并显示包括缺失权限的错误消息。

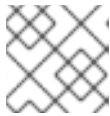
## 流程

1. 进入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**，而不是 **info**。

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



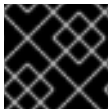
### 注意

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

## 验证

当集群部署成功完成时：

- 终端会显示用于访问集群的说明，包括指向 Web 控制台和 **kubeadmin** 用户的凭证的链接。
- 凭证信息还会输出到 **<installation\_directory>/./openshift\_install.log**。



### 重要

不要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。

## 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: "password"
INFO Time elapsed: 36m22s
```



## 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。一个例外是，您必须手动批准待处理的 **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。

## 7.10. 使用 CLI 登录集群

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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

### 先决条件

- 已部署 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. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

### 输出示例

```
system:admin
```

/validating-an-installation.adoc

## 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 控制台的更多详情](#)，请参阅 [访问 Web 控制台](#)。

## 7.12. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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](#)。

## 7.13. 后续步骤

- [验证安装](#)。
- [自定义集群](#)。

- [远程健康报告](#)
- [删除云供应商凭证](#)

## 第 8 章 在 AWS 上安装私有集群

在 OpenShift Container Platform 版本 4.14 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，您可以进一步自定义这些基础架构。要自定义安装，请在安装集群前修改 `install-config.yaml` 文件中的参数。

### 8.1. 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。
- 您可以阅读[选择集群安装方法并为用户准备它的文档](#)。
- 已将 [AWS 帐户配置](#) 为托管集群。



#### 重要

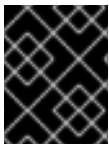
如果您的计算机上存储有 AWS 配置集，则不要在使用多因素验证设备的同时使用您生成的临时会话令牌。在集群的整个生命周期中，集群会持续使用您的当前 AWS 凭证来创建 AWS 资源，因此您必须使用长期凭证。要生成适当的密钥，请参阅 AWS 文档中的[管理 IAM 用户的访问密钥](#)。您可在运行安装程序时提供密钥。

- 如果使用防火墙，[将其配置为允许集群需要访问的站点](#)。

### 8.2. 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。在部署集群时，私有集群会将 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。



#### 重要

如果集群有任何公共子网，管理员创建的负载均衡器服务可能会公开访问。为确保集群安全性，请验证这些服务是否已明确标注为私有。

要部署私有集群，您必须：

- 使用满足您的要求的现有网络。集群资源可能会在网络上的其他集群间共享。
- 从有权访问的机器中部署：
  - 您置备的云的 API 服务。
  - 您调配的网络上的主机。
  - 用于获取安装介质的互联网。

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。

#### 8.2.1. 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。

#### 8.2.1.1. 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。

## 8.3. 关于使用自定义 VPC

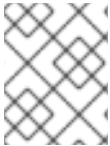
在 OpenShift Container Platform 4.14 中，您可以在 Amazon Web Services (AWS) 的现有 Amazon Virtual Private Cloud (VPC) 中将集群部署到现有子网中。通过将 OpenShift Container Platform 部署到现有的 AWS VPC 中，您可能会避开新帐户中的限制，或者更容易地利用公司所设置的操作限制。如果您无法获得您自己创建 VPC 所需的基础架构创建权限，请使用这个安装选项。

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。

#### 8.3.1. 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。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](#) 以及 [处理 VPC 和子网](#)。

安装程序无法：

- 分割网络范围供集群使用。
- 设置子网的路由表。
- 设置 VPC 选项，如 DHCP。

您必须在安装集群前完成这些任务。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 [您的 VPC 的路由表](#)。

您的 VPC 必须满足以下特征：

- VPC 不能使用 **kubernetes.io/cluster/.\*: owned, Name**, 和 **openshift.io/cluster** 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。您不能使用 **Name** 标签，因为它与 EC2 **Name** 字段重叠，且安装失败。
- 您需要在您的 VPC 中启用 **enableDnsSupport** 和 **enableDnsHostnames** 属性，以便集群可以使用附加到 VPC 中的 Route 53 区来解析集群内部的 DNS 记录。请参阅 AWS 文档中的 [您的 VPC 中的 DNS 支持](#) 部分。  
如果要使用您自己的 Route 53 托管私有区，您必须在安装集群前将现有托管区与 VPC 相关联。您可以使用 **install-config.yaml** 文件中的 **platform.aws.hostedZone** 和 **platform.aws.hostedZoneRole** 字段定义托管区。您可以通过与安装集群的帐户共享来使用来自另一个帐户的私有托管区。如果使用另一个帐户的私有托管区，则必须使用 **Passthrough** 或 **Manual** 凭证模式。

如果您在断开连接的环境中工作，则无法访问 EC2、ELB 和 S3 端点的公共 IP 地址。根据您要在安装过程中限制互联网流量的级别，有以下配置选项：

#### 8.3.1.1. 选项 1：创建 VPC 端点

创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

通过这个选项，网络流量在 VPC 和所需的 AWS 服务之间保持私有。

#### 8.3.1.2. 选项 2：创建一个没有 VPC 端点的代理

作为安装过程的一部分，您可以配置 HTTP 或 HTTPS 代理。使用此选项时，互联网流量会通过代理访问所需的 AWS 服务。

8.3.1.3. 选项 3：创建带有 VPC 端点的代理

作为安装过程的一部分，您可以使用 VPC 端点配置 HTTP 或 HTTPS 代理。创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

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

在 `install-config.yaml` 文件中配置代理时，将这些端点添加到 `noProxy` 字段。通过这个选项，代理会阻止集群直接访问互联网。但是，您的 VPC 和所需的 AWS 服务之间网络流量保持私有。

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>• <code>AWS::EC2::VPC</code></li><li>• <code>AWS::EC2::VPCEndpoint</code></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>• <code>AWS::EC2::Subnet</code></li><li>• <code>AWS::EC2::SubnetNetworkACLAssociation</code></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>• <code>AWS::EC2::InternetGateway</code></li><li>• <code>AWS::EC2::VPCGatewayAttachment</code></li><li>• <code>AWS::EC2::RouteTable</code></li><li>• <code>AWS::EC2::Route</code></li><li>• <code>AWS::EC2::SubnetRouteTableAssociation</code></li><li>• <code>AWS::EC2::NatGateway</code></li><li>• <code>AWS::EC2::EIP</code></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>• <code>AWS::EC2::NetworkACL</code></li><li>• <code>AWS::EC2::NetworkACLEntry</code></li></ul>                                                                                                                                                                                                                                            | 您必须允许 VPC 访问下列端口：                                                                                                   |            |
|        |                                                                                                                                                                                                                                                                                                                                                                                 | 端口                                                                                                                  | 原因         |
|        |                                                                                                                                                                                                                                                                                                                                                                                 | 80                                                                                                                  | 入站 HTTP 流量 |



| 组件   | AWS 类型                                                                                                                                                                     | 描述                                                                                |             |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------|
|      |                                                                                                                                                                            | 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。 |             |

### 8.3.2. 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。

### 8.3.3. 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。

### 8.3.4. 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)。

### 8.3.5. AWS 安全组

默认情况下，安装程序会创建安全组并将其附加到 control plane 和计算机器。不可修改与默认安全组关联的规则。

但是，您可以将与现有 VPC 关联的其他现有 AWS 安全组应用到 control plane 和计算机器。应用自定义安全组可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

作为安装过程的一部分，您可以在部署集群前通过修改 `install-config.yaml` 文件来应用自定义安全组。

如需更多信息，请参阅“将现有 AWS 安全组应用到集群”。

## 8.4. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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

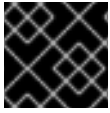
如果您的集群无法直接访问互联网，则可以在置备的某些类型的基础架构上执行受限网络安装。在此过程中，您可以下载所需的内容，并使用它为镜像 registry 填充安装软件包。对于某些安装类型，集群要安装到的环境不需要访问互联网。在更新集群前，您要更新镜像 registry 的内容。

## 8.5. 为集群节点 SSH 访问生成密钥对

在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 `core` 用户的 `~/.ssh/authorized_keys` 列表中，这将启用免密码身份验证。

将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。

如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。`./openshift-install gather` 命令还需要在集群节点上设置 SSH 公钥。

**重要**

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。

**注意**

您必须使用本地密钥，而不是使用特定平台方法配置 [的密钥](#)，如 [AWS 密钥对](#)。

**流程**

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，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_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。

**注意**

如果您计划在 **x86\_64**、**ppc64le** 和 **s390x** 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 **ed25519** 算法的密钥。相反，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

3. 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。

**注意**

在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- a. 如果 **ssh-agent** 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

**输出示例**

```
Agent pid 31874
```



### 注意

如果集群处于 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。

4. 将 SSH 私钥添加到 **ssh-agent**：

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

- 1 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

### 输出示例

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

### 后续步骤

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

## 8.6. 获取安装程序

在安装 OpenShift Container Platform 前，将安装文件下载到您用于安装的主机上。

### 先决条件

- 您有一台运行 Linux 或 macOS 的计算机，至少有 1.2 GB 本地磁盘空间。

### 流程

1. 进入 Red Hat Hybrid Cloud Console 上的 [Cluster Type](#) 页。如果您有红帽帐户，请使用您的凭证登录。如果没有，请创建一个帐户。
2. 在页的 **Run it yourself** 部分中选择您的基础架构供应商。
3. 从 **OpenShift 安装程序** 下的下拉菜单中选择您的主机操作系统和架构，然后点 **下载安装程序**。
4. 将下载的文件保存在要存储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几个文件。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。删除集群需要这两个文件。
- 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，请为特定云供应商完成 OpenShift Container Platform 卸载流程。

5. 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
```

6. 从 [Red Hat OpenShift Cluster Manager 下载安装 pull secret](#)。此 pull secret 允许您与所含授权机构提供的服务进行身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

## 提示

另外，您还可以从[红帽客户门户网站](#)检索安装程序，您可以在其中指定要下载的安装程序版本。但是，您需要有一个有效的订阅才能访问此页。

## 8.7. 手动创建安装配置文件

安装私有 OpenShift Container Platform 集群时，您必须手动生成安装配置文件。

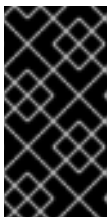
### 先决条件

- 在本地机器上有一个 SSH 公钥用于安装程序。您可以使用密钥在集群节点上进行 SSH 身份验证，以进行调试和灾难恢复。
- 已获取 OpenShift Container Platform 安装程序和集群的 pull secret。

### 流程

1. 创建一个安装目录来存储所需的安装资产：

```
$ mkdir <installation_directory>
```



#### 重要

您必须创建一个目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。

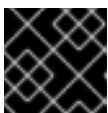
2. 对提供的 **install-config.yaml** 文件模板示例进行自定义，并将文件保存到 **<installation\_directory>** 中。



#### 注意

此配置文件必须命名为 **install-config.yaml**。

3. 备份 **install-config.yaml** 文件，以便使用它来安装多个集群。



#### 重要

现在备份 **install-config.yaml** 文件，因为安装过程会在下一步中消耗该文件。

### 其他资源

- [AWS 的安装配置参数](#)

#### 8.7.1. 集群安装的最低资源要求

每台集群机器都必须满足以下最低要求：

表 8.1. 最低资源要求

| 机器                  | 操作系统                     | vCPU [1] | 虚拟内存  | Storage | 每秒输入/输出 (IOPS) [2] |
|---------------------|--------------------------|----------|-------|---------|--------------------|
| bootstrap           | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Control plane（控制平面） | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Compute             | RHCOS、RHEL 8.6 及更新版本 [3] | 2        | 8 GB  | 100 GB  | 300                |

1. 当未启用并发多线程 (SMT) 或超线程时，一个 vCPU 相当于一个物理内核。启用后，使用以下公式来计算对应的比例：（每个内核数的线程）× sockets = vCPU。
2. OpenShift Container Platform 和 Kubernetes 对磁盘性能非常敏感，建议使用更快的存储速度，特别是 control plane 节点上需要 10 ms p99 fsync 持续时间的 etcd。请注意，在许多云平台上，存储大小和 IOPS 可一起扩展，因此您可能需要过度分配存储卷来获取足够的性能。
3. 与所有用户置备的安装一样，如果您选择在集群中使用 RHEL 计算机器，则负责所有操作系统生命周期管理和维护，包括执行系统更新、应用补丁和完成所有其他必要的任务。RHEL 7 计算机器的使用已弃用，并已在 OpenShift Container Platform 4.10 及更新的版本中删除。



注意

从 OpenShift Container Platform 版本 4.13 开始，RHCOS 基于 RHEL 版本 9.2，它更新了微架构要求。以下列表包含每个架构需要的最小指令集架构 (ISA)：

- x86-64 体系结构需要 x86-64-v2 ISA
- ARM64 架构需要 ARMv8.0-A ISA
- IBM Power 架构需要 Power 9 ISA
- s390x 架构需要 z14 ISA

如需更多信息，请参阅 [RHEL 架构](#)。

如果平台的实例类型满足集群机器的最低要求，则 OpenShift Container Platform 支持使用它。

其他资源

- [优化存储](#)

8.7.2. 为 AWS 测试的实例类型

以下 Amazon Web Services(AWS) 实例类型已经过 OpenShift Container Platform 测试。



### 注意

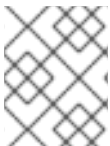
将以下图中包含的机器类型用于 AWS 实例。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 8.1. 基于 64 位 x86 架构的机器类型

- c4.\*
- c5.\*
- c5a.\*
- i3.\*
- m4.\*
- m5.\*
- m5a.\*
- m6i.\*
- r4.\*
- r5.\*
- r5a.\*
- r6i.\*
- t3.\*
- t3a.\*

#### 8.7.3. 在 64 位 ARM 基础架构上为 AWS 测试过的实例类型

OpenShift Container Platform 中已经测试了以下 Amazon Web Services (AWS) 64 位 ARM 实例类型。



### 注意

使用 AWS ARM 实例的以下图中包含的机器类型。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 8.2. 基于 64 位 ARM 架构的机器类型

- c6g.\*
- m6g.\*
- r8g.\*

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

您可以自定义安装配置文件 (**install-config.yaml**)，以指定有关 OpenShift Container Platform 集群平台的更多详细信息，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，并进行修改。

```
apiVersion: v1
baseDomain: example.com ❶
credentialsMode: Mint ❷
controlPlane: ❸ ❹
  hyperthreading: Enabled ❺
  name: master
  platform:
    aws:
      zones:
      - us-west-2a
      - us-west-2b
      rootVolume:
        iops: 4000
        size: 500
        type: io1 ❻
      metadataService:
        authentication: Optional ❼
        type: m6i.xlarge
      replicas: 3
compute: ❸
  - hyperthreading: Enabled ❾
    name: worker
    platform:
      aws:
        rootVolume:
          iops: 2000
          size: 500
          type: io1 ❿
        metadataService:
          authentication: Optional ❾
          type: c5.4xlarge
        zones:
        - us-west-2c
      replicas: 3
metadata:
  name: test-cluster ❿
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OVNKubernetes ⓫
```



```

serviceNetwork:
- 172.30.0.0/16
platform:
aws:
  region: us-west-2 14
  propagateUserTags: true 15
  userTags:
    adminContact: jdoe
    costCenter: 7536
  subnets: 16
  - subnet-1
  - subnet-2
  - subnet-3
  amiID: ami-0c5d3e03c0ab9b19a 17
  serviceEndpoints: 18
  - name: ec2
    url: https://vpce-id.ec2.us-west-2.vpce.amazonaws.com
  hostedZone: Z3URY6TWQ91KVV 19
fips: false 20
sshKey: ssh-ed25519 AAAA... 21
publish: Internal 22
pullSecret: '{"auths": ...}' 23

```

1 12 14 23 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。默认情况下，CCO 使用 **kube-system** 命名空间中的 root 凭证来动态尝试决定凭证的功能。有关 CCO 模式的详情，请参阅 [身份验证和授权指南](#) 中的 "About the Cloud Credential Operator" 部分。

3 8 15 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。仅使用一个 control plane 池。

5 9 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置为 **Disabled** 来禁用它。如果在某些集群机器中禁用并发多线程，则必须在所有集群机器中禁用它。



### 重要

如果您禁用并发多线程，请确保您的容量规划考虑机器性能显著降低的情况。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 10 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。

7 11 是否需要 [Amazon EC2 实例元数据服务 v2](#) (IMDSv2)。为了要求 IMDSv2，请将参数值设置为 **Required**。要允许使用 IMDSv1 和 IMDSv2，请将参数值设置为 **Optional**。如果没有指定值，则允许 IMDSv1 和 IMDSv2。

**注意**

在集群安装过程中设置的 control plane 机器的 IMDS 配置只能使用 AWS CLI 更改。可以使用计算机器集来更改计算机器的 IMDS 配置。

- 13 要安装的集群网络插件。支持的值有 **OVNKubernetes** 和 **OpenShiftSDN**。默认值为 **OVNKubernetes**。
- 16 如果您提供自己的 VPC，为集群使用的每个可用区指定子网。
- 17 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。
- 18 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。
- 19 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。
- 20 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**

要为集群启用 FIPS 模式，您必须从配置为以 FIPS 模式操作的 Red Hat Enterprise Linux (RHEL) 计算机运行安装程序。有关在 RHEL 中配置 FIPS 模式的更多信息，请参阅[在 FIPS 模式中安装该系统](#)。当以 FIPS 模式运行 Red Hat Enterprise Linux (RHEL) 或 Red Hat Enterprise Linux CoreOS (RHCOS) 时，OpenShift Container Platform 核心组件使用 RHEL 加密库，在 x86\_64、ppc64le 和 s390x 架构上提交到 NIST FIPS 140-2/140-3 Validation。

- 21 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。

- 22 如何发布集群的面向用户的端点。将 **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**。

### 8.7.5. 在安装过程中配置集群范围的代理

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，并确定它们中的任何站点是否需要绕过代理。默认情况下，所有集群出口流量都经过代理，包括对托管云供应商 API 的调用。如果需要，您将在 **Proxy** 对象的 **spec.noProxy** 字段中添加站点来绕过代理。



### 注意

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

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

### 流程

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: ec2.<aws_region>.amazonaws.com,elasticloadbalancing.
    <aws_region>.amazonaws.com,s3.<aws_region>.amazonaws.com 3
  additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

- 1 用于创建集群外 HTTP 连接的代理 URL。URL 方案必须是 **http**。
- 2 用于创建集群外 HTTPS 连接的代理 URL。
- 3 要从代理中排除的目标域名、IP 地址或其他网络 CIDR 的逗号分隔列表。在域前面加上 **.** 以仅匹配子域。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。如果您已将 Amazon **EC2**、**Elastic Load Balancing** 和 **S3** VPC 端点添加到 VPC 中，您必须将这些端点添加到 **noProxy** 字段。
- 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，其包含代理 HTTPS 连接所需的一个或多个额外 CA 证书。然后，Cluster Network Operator 会创建 **trusted-ca-bundle** 配置映射，将这些内容与 Red Hat Enterprise Linux CoreOS (RHCOS) 信任捆绑包合并，**Proxy** 对象的 **trustedCA** 字段中也会引用此配置映射。**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。
- 5 可选：决定 **Proxy** 对象的配置以引用 **trustedCA** 字段中 **user-ca-bundle** 配置映射的策略。允许的值是 **Proxyonly** 和 **Always**。仅在配置了 **http/https** 代理时，使用 **Proxyonly** 引用 **user-ca-bundle** 配置映射。使用 **Always** 始终引用 **user-ca-bundle** 配置映射。默认值为 **Proxyonly**。

**注意**

安装程序不支持代理的 **readinessEndpoints** 字段。

**注意**

如果安装程序超时，重启并使用安装程序的 **wait-for** 命令完成部署。例如：

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

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。

### 8.7.6. 将现有 AWS 安全组应用到集群

将现有 AWS 安全组应用到 control plane 和计算机器可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

#### 先决条件

- 您已在 AWS 中创建安全组。如需更多信息，请参阅使用 [安全组的 AWS 文档](#)。
- 安全组必须与您要将集群部署到的现有 VPC 关联。安全组不能与另一个 VPC 关联。
- 您有一个现有的 **install-config.yaml** 文件。

#### 流程

1. 在 **install-config.yaml** 文件中，编辑 **compute.platform.aws.additionalSecurityGroupIDs** 参数，为您的计算机器指定一个或多个自定义安全组。
2. 编辑 **controlPlane.platform.aws.additionalSecurityGroupIDs** 参数，为您的 control plane 机器指定一个或多个自定义安全组。
3. 保存文件并在部署集群时引用。

#### 指定自定义安全组的 **install-config.yaml** 文件示例

```
# ...
compute:
- hyperthreading: Enabled
  name: worker
  platform:
    aws:
      additionalSecurityGroupIDs:
        - sg-1 1
        - sg-2
```

```

replicas: 3
controlPlane:
  hyperthreading: Enabled
  name: master
  platform:
    aws:
      additionalSecurityGroupIDs:
        - sg-3
        - sg-4
replicas: 3
platform:
  aws:
    region: us-east-1
    subnets: 2
      - subnet-1
      - subnet-2
      - subnet-3

```

- 1 在 Amazon EC2 控制台中显示时指定安全组的名称，包括 **sg** 前缀。
- 2 指定集群使用的每个可用区的子网。

## 8.8. 通过下载二进制文件安装 OPENSIFT CLI

您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

### 8.8.1. 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. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。
5. 解包存档：

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

6. 将 **oc** 二进制文件放到 **PATH** 中的目录中。  
要查看您的 **PATH**，请执行以下命令：

■

```
$ echo $PATH
```

## 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 8.8.2. 在 Windows 上安装 OpenShift CLI

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

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
4. 使用 ZIP 程序解压存档。
5. 将 **oc** 二进制文件移到 **PATH** 中的目录中。  
要查看您的 **PATH**，请打开命令提示并执行以下命令：

```
C:\> path
```

## 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 8.8.3. 在 macOS 上安装 OpenShift CLI

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

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。



#### 注意

对于 macOS arm64，请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

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**，请打开终端并执行以下命令：

```
$ echo $PATH
```

验证

- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

## 8.9. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案

默认情况下，管理员 secret 存储在 **kube-system** 项目中。如果您在 **install-config.yaml** 文件中将 **credentialsMode** 参数配置为 **Manual**，则必须使用以下替代方案之一：

- 要手动管理长期云凭证，请按照[手动创建长期凭证](#)中的步骤操作。
- 要实现在集群外为各个组件管理的短期凭证，请按照[配置 AWS 集群以使用短期凭证](#)中的步骤操作。

### 8.9.1. 手动创建长期凭证

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. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

#### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/' {print $3})
```

4. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 自定义资源 (CR) 列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
```



```
--credentials-requests \
--included \ ❶
--install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \ ❷
--to=<path_to_directory_for_credentials_requests> ❸
```

- ❶ **--included** 参数仅包含特定集群配置所需的清单。
- ❷ 指定 **install-config.yaml** 文件的位置。
- ❸ 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

此命令为每个 **CredentialsRequest** 对象创建一个 YAML 文件。

### CredentialsRequest 对象示例

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

5. 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。

### 带有 secret 的 CredentialsRequest 对象示例

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
...
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
      - effect: Allow
        action:
          - s3:CreateBucket
```



```

- s3:DeleteBucket
  resource: "*"
...
secretRef:
  name: <component_secret>
  namespace: <component_namespace>
...

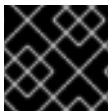
```

### Secret 对象示例

```

apiVersion: v1
kind: Secret
metadata:
  name: <component_secret>
  namespace: <component_namespace>
data:
  aws_access_key_id: <base64_encoded_aws_access_key_id>
  aws_secret_access_key: <base64_encoded_aws_secret_access_key>

```



#### 重要

在升级使用手动维护凭证的集群前，您必须确保 CCO 处于可升级状态。

## 8.9.2. 将 AWS 集群配置为使用短期凭证

要安装配置为使用 AWS 安全令牌服务 (STS) 的集群，您必须配置 CCO 实用程序并为集群创建所需的 AWS 资源。

### 8.9.2.1. 配置 Cloud Credential Operator 工具

当 Cloud Credential Operator (CCO) 以手动模式运行时，要从集群外部创建和管理云凭证，提取并准备 CCO 实用程序 (**ccoctl**) 二进制文件。



#### 注意

**ccoctl** 工具是在 Linux 环境中运行的 Linux 二进制文件。

#### 先决条件

- 您可以访问具有集群管理员权限的 OpenShift Container Platform 帐户。
- 已安装 OpenShift CLI (**oc**)。
- 您已为 **ccoctl** 工具创建了用于以下权限的 AWS 帐户：

#### 例 8.3. 所需的 AWS 权限

##### 所需的 iam 权限

- **iam:CreateOpenIDConnectProvider**
- **iam:CreateRole**
- **iam>DeleteOpenIDConnectProvider**

- **iam:DeleteRole**
- **iam:DeleteRolePolicy**
- **iam:GetOpenIDConnectProvider**
- **iam:GetRole**
- **iam:GetUser**
- **iam:ListOpenIDConnectProviders**
- **iam:ListRolePolicies**
- **iam:ListRoles**
- **iam:PutRolePolicy**
- **iam:TagOpenIDConnectProvider**
- **iam:TagRole**

#### 所需的 s3 权限

- **s3:CreateBucket**
- **s3:DeleteBucket**
- **s3:DeleteObject**
- **s3:GetBucketAcl**
- **s3:GetBucketTagging**
- **s3:GetObject**
- **s3:GetObjectAcl**
- **s3:GetObjectTagging**
- **s3:ListBucket**
- **s3:PutBucketAcl**
- **s3:PutBucketPolicy**
- **s3:PutBucketPublicAccessBlock**
- **s3:PutBucketTagging**
- **s3:PutObject**
- **s3:PutObjectAcl**
- **s3:PutObjectTagging**

### 所需的 cloudfront 权限

- **cloudfront:ListCloudFrontOriginAccessIdentities**
- **cloudfront:ListDistributions**
- **cloudfront:ListTagsForResource**

如果您计划通过公共 CloudFront 发行版 URL 将 OIDC 配置存储在 IAM 身份提供程序访问的私有 S3 存储桶中，则运行 **ccoctl** 工具的 AWS 帐户需要以下额外权限：

### 例 8.4. 使用 CloudFront 私有 S3 存储桶的额外权限

- **cloudfront:CreateCloudFrontOriginAccessIdentity**
- **cloudfront:CreateDistribution**
- **cloudfront>DeleteCloudFrontOriginAccessIdentity**
- **cloudfront>DeleteDistribution**
- **cloudfront:GetCloudFrontOriginAccessIdentity**
- **cloudfront:GetCloudFrontOriginAccessIdentityConfig**
- **cloudfront:GetDistribution**
- **cloudfront:TagResource**
- **cloudfront:UpdateDistribution**



### 注意

在使用 **ccoctl aws create-all** 命令处理凭证请求时，这些额外权限支持使用 **--create-private-s3-bucket** 选项。

## 流程

1. 运行以下命令，为 OpenShift Container Platform 发行镜像设置变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/' {print $3})
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像获取 CCO 容器镜像：

```
$ CCO_IMAGE=$(oc adm release info --image-for='cloud-credential-operator'
$RELEASE_IMAGE -a ~/.pull-secret)
```



### 注意

确保 **\$RELEASE\_IMAGE** 的架构与将使用 **ccoctl** 工具的环境架构相匹配。

- 运行以下命令，将 CCO 容器镜像中的 **ccoctl** 二进制文件提取到 OpenShift Container Platform 发行镜像中：

```
$ oc image extract $CCO_IMAGE --file="/usr/bin/ccoctl" -a ~/.pull-secret
```

- 运行以下命令更改权限以使 **ccoctl** 可执行：

```
$ chmod 775 ccoctl
```

## 验证

- 要验证 **ccoctl** 是否准备就绪，可以尝试显示帮助文件。运行命令时使用相对文件名，例如：

```
$ ./ccoctl.rhel9
```

## 输出示例

```
OpenShift credentials provisioning tool

Usage:
  ccoctl [command]

Available Commands:
  alibabacloud Manage credentials objects for alibaba cloud
  aws           Manage credentials objects for AWS cloud
  azure        Manage credentials objects for Azure
  gcp          Manage credentials objects for Google cloud
  help         Help about any command
  ibmcloud     Manage credentials objects for IBM Cloud
  nutanix      Manage credentials objects for Nutanix

Flags:
  -h, --help  help for ccoctl

Use "ccoctl [command] --help" for more information about a command.
```

### 8.9.2.2. 使用 Cloud Credential Operator 实用程序创建 AWS 资源

创建 AWS 资源时有以下选项：

- 您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。这是创建资源的最快速方法。请参阅 [使用单个命令创建 AWS 资源](#)。
- 如果您需要在修改 AWS 资源前查看 **ccoctl** 工具创建的 JSON 文件，或者 **ccoctl** 工具用于创建 AWS 资源的过程无法自动满足组织的要求，您可以单独创建 AWS 资源。请参阅 [单独创建 AWS 资源](#)。

#### 8.9.2.2.1. 使用单个命令创建 AWS 资源

如果 **ccoctl** 工具用于创建 AWS 资源的过程自动满足机构的要求，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。

否则，您可以单独创建 AWS 资源。如需更多信息，请参阅“单独创建 AWS 资源”。



## 注意

默认情况下，**ccctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccctl\_output\_dir>** 来引用这个目录。

## 先决条件

您必须：

- 提取并准备好 **ccctl** 二进制文件。

## 流程

1. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/' {print $3})
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \1
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \2
  --to=<path_to_directory_for_credentials_requests> \3
```

- 1 **--included** 参数仅包含特定集群配置所需的清单。
- 2 指定 **install-config.yaml** 文件的位置。
- 3 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。



## 注意

此命令可能需要一些时间才能运行。

3. 运行以下命令，使用 **ccctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccctl aws create-all \
  --name=<name> \1
  --region=<aws_region> \2
  --credentials-requests-dir=<path_to_credentials_requests_directory> \3
  --output-dir=<path_to_ccctl_output_dir> \4
  --create-private-s3-bucket \5
```

- 1 指定用于标记创建用于跟踪的任何云资源的名称。
- 2 指定在其中创建云资源的 AWS 区域。
- 3 指定包含组件 **CredentialsRequest** 对象文件的目录。

- 4 可选：指定您希望 **ccoctl** 实用程序在其中创建对象的目录。默认情况下，实用程序在运行命令的目录中创建对象。
- 5 可选：默认情况下，**ccoctl** 实用程序将 OpenID Connect (OIDC) 配置文件存储在公共 S3 存储桶中，并使用 S3 URL 作为公共 OIDC 端点。要将 OIDC 配置存储在 IAM 身份提供程序通过公共 CloudFront 发行版 URL 访问的专用 S3 存储桶中，请使用 **--create-private-s3-bucket** 参数。



### 注意

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

## 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccoctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccoctl_output_dir>/manifests
```

## 输出示例

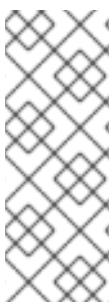
```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

### 8.9.2.2.2. 单独创建 AWS 资源

您可以使用 **ccoctl** 工具单独创建 AWS 资源。这个选项对于在不同用户或部门之间创建这些资源的组织可能很有用。

否则，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。如需更多信息，请参阅“使用单个命令创建 AWS 资源”。



### 注意

默认情况下，**ccoctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccoctl\_output\_dir>** 来引用这个目录。

有些 **ccoctl** 命令会发出 AWS API 调用来创建或修改 AWS 资源。您可以使用 **--dry-run** 标志来避免 API 调用。使用此标志可在本地文件系统中创建 JSON 文件。您可以使用 **--cli-input-json** 参数查看和修改 JSON 文件，然后使用 AWS CLI 工具应用它们。

## 先决条件

- 提取并准备 **ccctl** 二进制文件。

## 流程

1. 运行以下命令，生成用于为集群设置 OpenID Connect 供应商的公共和私有 RSA 密钥文件：

```
$ ccctl aws create-key-pair
```

### 输出示例

```
2021/04/13 11:01:02 Generating RSA keypair
2021/04/13 11:01:03 Writing private key to /<path_to_ccctl_output_dir>/serviceaccount-
signer.private
2021/04/13 11:01:03 Writing public key to /<path_to_ccctl_output_dir>/serviceaccount-
signer.public
2021/04/13 11:01:03 Copying signing key for use by installer
```

其中 **serviceaccount-signer.private** 和 **serviceaccount-signer.public** 是生成的密钥文件。

此命令还会在 **/<path\_to\_ccctl\_output\_dir>/tls/bound-service-account-signing-key.key** 中创建集群在安装过程中所需的私钥。

2. 运行以下命令，在 AWS 上创建 OpenID Connect 身份提供程序和 S3 存储桶：

```
$ ccctl aws create-identity-provider \
  --name=<name> ❶
  --region=<aws_region> ❷
  --public-key-file=<path_to_ccctl_output_dir>/serviceaccount-signer.public ❸
```

❶ **<name>** 是用于标记为跟踪而创建的云资源的名称。

❷ **<aws-region>** 是将来要在其中创建云资源的 AWS 区域。

❸ **<path\_to\_ccctl\_output\_dir>** 是 **ccctl aws create-key-pair** 命令生成的公钥文件的路径。

### 输出示例

```
2021/04/13 11:16:09 Bucket <name>-oidc created
2021/04/13 11:16:10 OpenID Connect discovery document in the S3 bucket <name>-oidc at
.well-known/openid-configuration updated
2021/04/13 11:16:10 Reading public key
2021/04/13 11:16:10 JSON web key set (JWKS) in the S3 bucket <name>-oidc at keys.json
updated
2021/04/13 11:16:18 Identity Provider created with ARN: arn:aws:iam::
<aws_account_id>:oidc-provider/<name>-oidc.s3.<aws_region>.amazonaws.com
```

其中 **openid-configuration** 是发现文档和 **key.json** 是一个 JSON Web 密钥集文件。

此命令还会在 **/<path\_to\_ccctl\_output\_dir>/manifests/cluster-authentication-02-config.yaml** 中创建 YAML 配置文件。此文件为集群生成的服务帐户令牌设置签发者 URL 字段，以便 AWS IAM 身份提供程序信任令牌。

## 3. 为集群中的每个组件创建 IAM 角色：

- a. 运行以下命令，使用安装文件中的发行镜像设置
- \$RELEASE\_IMAGE**
- 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

- b. 从 OpenShift Container Platform 发行镜像中提取
- CredentialsRequest**
- 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included 1 \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
2 \
  --to=<path_to_directory_for_credentials_requests> 3
```

**1** **--included** 参数仅包含特定集群配置所需的清单。

**2** 指定 **install-config.yaml** 文件的位置。

**3** 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

- c. 运行以下命令，使用
- ccoctl**
- 工具处理所有
- CredentialsRequest**
- 对象：

```
$ ccoctl aws create-iam-roles \
  --name=<name> \
  --region=<aws_region> \
  --credentials-requests-dir=<path_to_credentials_requests_directory> \
  --identity-provider-arn=arn:aws:iam::<aws_account_id>:oidc-provider/<name>-oidc.s3.
  <aws_region>.amazonaws.com
```

**注意**

对于使用其他 IAM API 端点的 AWS 环境（如 GovCloud），还必须使用 **--region** 参数指定您的区域。

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

对于每个 **CredentialsRequest** 对象，**ccoctl** 创建一个带有信任策略的 IAM 角色，该角色与指定的 OIDC 身份提供程序相关联，以及来自 OpenShift Container Platform 发行镜像的每个 **CredentialsRequest** 对象中定义的权限策略。

**验证**

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccoctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccoctl_output_dir>/manifests
```

**输出示例**



```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

### 8.9.2.3. 整合 Cloud Credential Operator 实用程序清单

要为单个组件在集群外实现短期安全凭证，您必须将创建 Cloud Credential Operator 实用程序 (**ccctl**) 的清单文件移到安装程序的正确目录中。

#### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您已配置了 Cloud Credential Operator 实用程序 (**ccctl**)。
- 已使用 **ccctl** 工具创建了集群所需的云供应商资源。

#### 流程

1. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

#### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，将 **ccctl** 工具生成的清单复制到安装程序创建的 **manifests** 目录中：

```
$ cp /<path_to_ccctl_output_dir>/manifests/* ./manifests/
```

4. 将包含私钥的 **tls** 目录复制到安装目录中：

```
$ cp -a /<path_to_ccctl_output_dir>/tls .
```

## 8.10. 部署集群

您可以在兼容云平台上安装 OpenShift Container Platform。



### 重要

在初始安装过程中，您只能运行安装程序的 **create cluster** 命令一次。

### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。
- 已确认主机上的云供应商帐户具有部署集群的正确权限。权限不正确的帐户会导致安装过程失败，并显示包括缺失权限的错误消息。

### 流程

1. 进入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>**，请指定自定义 **./install-config.yaml** 文件的位置。

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

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。

### 验证

当集群部署成功完成时：

- 终端会显示用于访问集群的说明，包括指向 Web 控制台和 **kubeadmin** 用户的凭证的链接。
- 凭证信息还会输出到 **<installation\_directory>/./openshift\_install.log**。



### 重要

不要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。

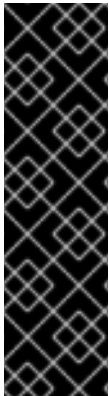
### 输出示例

```
...
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: "password"
```

```
INFO Time elapsed: 36m22s
```



## 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。一个例外是，您必须手动批准待处理的 **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 小时时间进行轮转。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。

## 8.11. 使用 CLI 登录集群

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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

### 先决条件

- 已部署 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. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

### 输出示例

```
system:admin
```

```
/validating-an-installation.adoc
```

## 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。

生产名称

## 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 控制台的更多详情](#)，请参阅 [访问 Web 控制台](#)。

## 8.13. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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](#)。

## 8.14. 后续步骤

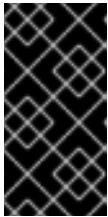
- [验证安装.](#)
- [自定义集群](#)
- [远程健康报告](#)
- [删除云供应商凭证](#)

## 第 9 章 在 AWS 上将集群安装到一个政府区域

在 OpenShift Container Platform 版本 4.14 中，您可以在 Amazon Web Services (AWS) 上将集群安装到一个政府区域。要配置区域，在安装集群前修改 `install-config.yaml` 文件中的参数。

### 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。
- 您可以阅读[选择集群安装方法并为用户准备它的文档](#)。
- 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，[将其配置为允许集群需要访问的站点](#)。

### 9.2. AWS 政府区域

OpenShift Container Platform 支持将集群部署到 [AWS GovCloud\(US\)](#) 区域。

支持以下 AWS GovCloud 分区：

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

### 9.3. 安装要求

在安装集群前，您必须：

- 提供托管集群的现有的私有 AWS VPC 和子网。  
AWS GovCloud 的 Route 53 不支持公共区。因此，当您部署到 AWS 政府区域时，集群必须是私有的。
- 手动创建安装配置文件 (`install-config.yaml`)。

### 9.4. 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 区域的 Route 53 不支持公共区。因此，如果集群部署到 AWS GovCloud 区域，则集群必须是私有的。

默认情况下，OpenShift Container Platform 被置备为使用可公开访问的 DNS 和端点。在部署集群时，私有集群会将 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。



### 重要

如果集群有任何公共子网，管理员创建的负载均衡器服务可能会公开访问。为确保集群安全性，请验证这些服务是否已明确标注为私有。

要部署私有集群，您必须：

- 使用满足您的要求的现有网络。集群资源可能会在网络上的其他集群间共享。
- 从有权访问的机器中部署：
  - 您置备的云的 API 服务。
  - 您调配的网络上的主机。
  - 用于获取安装介质的互联网。

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。

## 9.4.1. 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。

### 9.4.1.1. 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。

## 9.5. 关于使用自定义 VPC

在 OpenShift Container Platform 4.14 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。

### 9.5.1. 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。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](#) 以及 [处理 VPC 和子网](#)。

安装程序无法：

- 分割网络范围供集群使用。
- 设置子网的路由表。
- 设置 VPC 选项，如 DHCP。

您必须在安装集群前完成这些任务。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 [您的 VPC 的路由表](#)。

您的 VPC 必须满足以下特征：

- VPC 不能使用 **kubernetes.io/cluster/.\*: owned, Name**, 和 **openshift.io/cluster** 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。您不能使用 **Name** 标签，因为它与 EC2 **Name** 字段重叠，且安装失败。



- 您需要在您的 VPC 中启用 **enableDnsSupport** 和 **enableDnsHostnames** 属性，以便集群可以使用附加到 VPC 中的 Route 53 区来解析集群内部的 DNS 记录。请参阅 AWS 文档中的[您的 VPC 中的 DNS 支持](#)部分。
- 如果要使用您自己的 Route 53 托管私有区，您必须在安装集群前将现有托管区与 VPC 相关联。您可以使用 **install-config.yaml** 文件中的 **platform.aws.hostedZone** 和 **platform.aws.hostedZoneRole** 字段定义托管区。您可以通过与安装集群的帐户共享来使用来自另一个帐户的私有托管区。如果使用另一个帐户的私有托管区，则必须使用 **Passthrough** 或 **Manual** 凭证模式。

如果您在断开连接的环境中工作，则无法访问 EC2、ELB 和 S3 端点的公共 IP 地址。根据您要在安装过程中限制互联网流量的级别，有以下配置选项：

#### 9.5.1.1. 选项 1：创建 VPC 端点

创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

通过这个选项，网络流量在 VPC 和所需的 AWS 服务之间保持私有。

#### 9.5.1.2. 选项 2：创建一个没有 VPC 端点的代理

作为安装过程的一部分，您可以配置 HTTP 或 HTTPS 代理。使用此选项时，互联网流量会通过代理访问所需的 AWS 服务。

#### 9.5.1.3. 选项 3：创建带有 VPC 端点的代理

作为安装过程的一部分，您可以使用 VPC 端点配置 HTTP 或 HTTPS 代理。创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

在 **install-config.yaml** 文件中配置代理时，将这些端点添加到 **noProxy** 字段。通过这个选项，代理会阻止集群直接访问互联网。但是，您的 VPC 和所需的 AWS 服务之间网络流量保持私有。

### 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：                                                                                                   |             |
|        |                                                                                                                                                                                                                                                                                                                                       | 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。                                   |             |

### 9.5.2. 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。

### 9.5.3. 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：不同的个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。

### 9.5.4. 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)。

### 9.5.5. AWS 安全组

默认情况下，安装程序会创建安全组并将其附加到 control plane 和计算机器。不可修改与默认安全组关联的规则。

但是，您可以将与现有 VPC 关联的其他现有 AWS 安全组应用到 control plane 和计算机器。应用自定义安全组可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

作为安装过程的一部分，您可以在部署集群前通过修改 **install-config.yaml** 文件来应用自定义安全组。

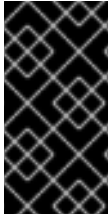
如需更多信息，请参阅“将现有 AWS 安全组应用到集群”。

## 9.6. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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

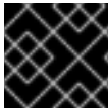
如果您的集群无法直接访问互联网，则可以在置备的某些类型的基础架构上执行受限网络安装。在此过程中，您可以下载所需的内容，并使用它为镜像 registry 填充安装软件包。对于某些安装类型，集群要安装到的环境不需要访问互联网。在更新集群前，您要更新镜像 registry 的内容。

## 9.7. 为集群节点 SSH 访问生成密钥对

在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 **core** 用户的 `~/.ssh/authorized_keys` 列表中，这将启用免密码身份验证。

将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。

如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。`./openshift-install gather` 命令还需要在集群节点上设置 SSH 公钥。



### 重要

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。



### 注意

您必须使用本地密钥，而不是使用特定平台方法配置的密钥，如 [AWS 密钥对](#)。

### 流程

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，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_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。



### 注意

如果您计划在 **x86\_64**、**ppc64le** 和 **s390x** 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 **ed25519** 算法的密钥。相反，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

## 2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

3. 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。

**注意**

在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- a. 如果 `ssh-agent` 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

**输出示例**

```
Agent pid 31874
```

**注意**

如果集群处于 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。

4. 将 SSH 私钥添加到 `ssh-agent`：

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

- ❶ 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

**输出示例**

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

**后续步骤**

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

## 9.8. 获取 AWS MARKETPLACE 镜像

如果要使用 AWS Marketplace 镜像部署 OpenShift Container Platform 集群，您必须首先通过 AWS 订阅。订阅提供的提供的 AMI ID 可让您使用安装程序用来部署 worker 节点的 AMI ID。



## 注意

您应该只修改计算机器的 RHCOS 镜像以使用 AWS Marketplace 镜像。control plane 机器和基础架构节点不需要 OpenShift Container Platform 订阅，且默认使用公共 RHCOS 默认镜像，这不会对 AWS bill 造成订阅成本。因此，您不应该修改集群默认引导镜像或 control plane 引导镜像。将 AWS Marketplace 镜像应用到它们时，会生成无法恢复的额外许可成本。

## 先决条件

- 有 AWS 账户购买的产品。此帐户不必与用于安装集群的帐户相同。

## 流程

1. 从 [AWS Marketplace](#) 完成 OpenShift Container Platform 订阅。
2. 记录特定区域的 AMI ID。作为安装过程的一部分，您必须在部署集群前使用这个值更新 **install-config.yaml** 文件。

## 使用 AWS Marketplace worker 节点的 install-config.yaml 文件示例

```
apiVersion: v1
baseDomain: example.com
compute:
- hyperthreading: Enabled
  name: worker
  platform:
    aws:
      amiID: ami-06c4d345f7c207239 1
      type: m5.4xlarge
      replicas: 3
metadata:
  name: test-cluster
platform:
  aws:
    region: us-east-2 2
sshKey: ssh-ed25519 AAAA...
pullSecret: '{"auths": ...}'
```

- 1** 来自 AWS Marketplace 订阅的 AMI ID。
- 2** 您的 AMI ID 与特定的 AWS 区域相关联。在创建安装配置文件时，请确保选择配置订阅时指定的同一 AWS 区域。

## 9.9. 获取安装程序

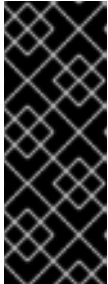
在安装 OpenShift Container Platform 前，将安装文件下载到您用于安装的主机上。

## 先决条件

- 您有一台运行 Linux 或 macOS 的计算机，至少有 1.2 GB 本地磁盘空间。

## 流程

1. 进入 Red Hat Hybrid Cloud Console 上的 [Cluster Type](#) 页。如果您有红帽帐户，请使用您的凭证登录。如果没有，请创建一个帐户。
2. 在页的 **Run it yourself** 部分中选择您的基础架构供应商。
3. 从 **OpenShift 安装程序** 下的下拉菜单中选择您的主机操作系统和架构，然后点**下载安装程序**。
4. 将下载的文件保存在要存储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几个文件。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。删除集群需要这两个文件。
- 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，请为特定云供应商完成 OpenShift Container Platform 卸载流程。

5. 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
```

6. 从 [Red Hat OpenShift Cluster Manager](#) 下载安装 [pull secret](#)。此 pull secret 允许您与所含授权机构提供的服务进行身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

### 提示

另外，您还可以从[红帽客户门户网站](#)检索安装程序，您可以在其中指定要下载的安装程序版本。但是，您需要有一个有效的订阅才能访问此页。

## 9.10. 手动创建安装配置文件

安装集群需要您手动生成安装配置文件。

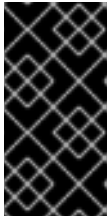
### 先决条件

- 在本地机器上有一个 SSH 公钥用于安装程序。您可以使用密钥在集群节点上进行 SSH 身份验证，以进行调试和灾难恢复。
- 已获取 OpenShift Container Platform 安装程序和集群的 pull secret。

### 流程

1. 创建一个安装目录来存储所需的安装资产：

```
$ mkdir <installation_directory>
```



重要

您必须创建一个目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。

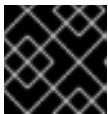
- 2. 对提供的 `install-config.yaml` 文件模板示例进行自定义，并将文件保存到 `<installation_directory>` 中。



注意

此配置文件必须命名为 `install-config.yaml`。

- 3. 备份 `install-config.yaml` 文件，以便使用它来安装多个集群。



重要

现在备份 `install-config.yaml` 文件，因为安装过程会在下一步中消耗该文件。

其他资源

- [AWS 的安装配置参数](#)

9.10.1. 集群安装的最低资源要求

每台集群机器都必须满足以下最低要求：

表 9.1. 最低资源要求

| 机器                   | 操作系统                     | vCPU [1] | 虚拟内存  | Storage | 每秒输入/输出 (IOPS) [2] |
|----------------------|--------------------------|----------|-------|---------|--------------------|
| bootstrap            | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Control plane (控制平面) | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Compute              | RHCOS、RHEL 8.6 及更新版本 [3] | 2        | 8 GB  | 100 GB  | 300                |

- 1. 当未启用并发多线程 (SMT) 或超线程时，一个 vCPU 相当于一个物理内核。启用后，使用以下公式来计算对应的比例： $(\text{每个内核数的线程}) \times \text{sockets} = \text{vCPU}$ 。
- 2. OpenShift Container Platform 和 Kubernetes 对磁盘性能非常敏感，建议使用更快的存储速度，特别是 control plane 节点上需要 10 ms p99 fsync 持续时间的 etcd。请注意，在许多云平台上，存储大小和 IOPS 可一起扩展，因此您可能需要过度分配存储卷来获取足够的性能。



3. 与所有用户置备的安装一样，如果您选择在集群中使用 RHEL 计算机器，则负责所有操作系统生命周期管理和维护，包括执行系统更新、应用补丁和完成所有其他必要的任务。RHEL 7 计算机器的使用已弃用，并已在 OpenShift Container Platform 4.10 及更新的版本中删除。



### 注意

从 OpenShift Container Platform 版本 4.13 开始，RHCOS 基于 RHEL 版本 9.2，它更新了微架构要求。以下列表包含每个架构需要的最小指令集架构 (ISA)：

- x86-64 体系结构需要 x86-64-v2 ISA
- ARM64 架构需要 ARMv8.0-A ISA
- IBM Power 架构需要 Power 9 ISA
- s390x 架构需要 z14 ISA

如需更多信息，请参阅 [RHEL 架构](#)。

如果平台的实例类型满足集群机器的最低要求，则 OpenShift Container Platform 支持使用它。

### 其他资源

- [优化存储](#)

### 9.10.2. 为 AWS 测试的实例类型

以下 Amazon Web Services(AWS) 实例类型已经过 OpenShift Container Platform 测试。



### 注意

将以下图中包含的机器类型用于 AWS 实例。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 9.1. 基于 64 位 x86 架构的机器类型

- c4.\*
- c5.\*
- c5a.\*
- i3.\*
- m4.\*
- m5.\*
- m5a.\*
- m6i.\*
- r4.\*

- **r5.\***
- **r5a.\***
- **r6i.\***
- **t3.\***
- **t3a.\***

### 9.10.3. 在 64 位 ARM 基础架构上为 AWS 测试过的实例类型

OpenShift Container Platform 中已经测试了以下 Amazon Web Services (AWS) 64 位 ARM 实例类型。



#### 注意

使用 AWS ARM 实例的以下图中包含的机器类型。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 9.2. 基于 64 位 ARM 架构的机器类型

- **c6g.\***
- **m6g.\***
- **r8g.\***

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

您可以自定义安装配置文件 (**install-config.yaml**)，以指定有关 OpenShift Container Platform 集群平台的更多详细信息，或修改所需参数的值。



#### 重要

此示例 YAML 文件仅供参考。使用它作为资源，在您手动创建的安装配置文件中输入参数值。

```
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
    metadataService:
      authentication: Optional 7
      type: m6i.xlarge
    replicas: 3
  compute: 8
  - hyperthreading: Enabled 9
  name: worker
  platform:
    aws:
      rootVolume:
        iops: 2000
        size: 500
        type: io1 10
      metadataService:
        authentication: Optional 11
        type: c5.4xlarge
      zones:
        - us-gov-west-1c
      replicas: 3
  metadata:
    name: test-cluster 12
  networking:
    clusterNetwork:
      - cidr: 10.128.0.0/14
        hostPrefix: 23
    machineNetwork:
      - cidr: 10.0.0.0/16
    networkType: OVNKubernetes 13
    serviceNetwork:
      - 172.30.0.0/16
  platform:
    aws:
      region: us-gov-west-1 14
      propagateUserTags: true 15
      userTags:
        adminContact: jdoe
        costCenter: 7536
      subnets: 16
      - subnet-1
      - subnet-2
      - subnet-3
      amiID: ami-0c5d3e03c0ab9b19a 17
      serviceEndpoints: 18
      - name: ec2
        url: https://vpce-id.ec2.us-west-2.vpce.amazonaws.com
      hostedZone: Z3URY6TWQ91KVV 19
  fips: false 20
  sshKey: ssh-ed25519 AAAA... 21
  publish: Internal 22
  pullSecret: '{"auths": ...}' 23

```

- 1 12 14 23 必需。
- 2 可选：添加此参数来强制 Cloud Credential Operator (CCO) 使用指定的模式。默认情况下，CCO 使用 **kube-system** 命名空间中的 root 凭证来动态尝试决定凭证的功能。有关 CCO 模式的详情，请参阅 *身份验证和授权指南* 中的 "About the Cloud Credential Operator" 部分。
- 3 8 15 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。仅使用一个 control plane 池。
- 5 9 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置为 **Disabled** 来禁用它。如果在某些集群机器中禁用并发多线程，则必须在所有集群机器中禁用它。



### 重要

如果您禁用并发多线程，请确保您的容量规划考虑机器性能显著降低的情况。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 10 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**。
- 7 11 是否需要 [Amazon EC2 实例元数据服务 v2 \(IMDSv2\)](#)。为了要求 IMDSv2，请将参数值设置为 **Required**。要允许使用 IMDSv1 和 IMDSv2，请将参数值设置为 **Optional**。如果没有指定值，则允许 IMDSv1 和 IMDSv2。



### 注意

在集群安装过程中设置的 control plane 机器的 IMDS 配置只能使用 AWS CLI 更改。可以使用计算机器集来更改计算机器的 IMDS 配置。

- 13 要安装的集群网络插件。支持的值有 **OVNKubernetes** 和 **OpenShiftSDN**。默认值为 **OVNKubernetes**。
- 16 如果您提供自己的 VPC，为集群使用的每个可用区指定子网。
- 17 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。
- 18 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。
- 19 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。
- 20 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

要为集群启用 FIPS 模式，您必须从配置为以 FIPS 模式操作的 Red Hat Enterprise Linux (RHEL) 计算机运行安装程序。有关在 RHEL 中配置 FIPS 模式的更多信息，请参阅[在 FIPS 模式中安装该系统](#)。当以 FIPS 模式运行 Red Hat Enterprise Linux (RHEL) 或 Red Hat Enterprise Linux CoreOS (RHCOS) 时，OpenShift Container Platform 核心组件使用 RHEL 加密库，在 x86\_64、ppc64le 和 s390x 架构上提交到 NIST FIPS 140-2/140-3 Validation。

21

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。

22

如何发布集群的面向用户的端点。将 **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**。

## 9.10.5. 在安装过程中配置集群范围的代理

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，并确定它们中的任何站点是否需要绕过代理。默认情况下，所有集群出口流量都经过代理，包括对托管云供应商 API 的调用。如果需要，您将在 **Proxy** 对象的 **spec.noProxy** 字段中添加站点来绕过代理。



### 注意

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

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

### 流程

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: ec2.<aws_region>.amazonaws.com,elasticloadbalancing.
```

```
<aws_region>.amazonaws.com,s3.<aws_region>.amazonaws.com 3
additionalTrustBundle: | 4
-----BEGIN CERTIFICATE-----
<MY_TRUSTED_CA_CERT>
-----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

- 1 用于创建集群外 HTTP 连接的代理 URL。URL 方案必须是 **http**。
- 2 用于创建集群外 HTTPS 连接的代理 URL。
- 3 要从代理中排除的目标域名、IP 地址或其他网络 CIDR 的逗号分隔列表。在域前面加上 **.** 以仅匹配子域。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。如果您已将 Amazon **EC2**、**Elastic Load Balancing** 和 **S3** VPC 端点添加到 VPC 中，您必须将这些端点添加到 **noProxy** 字段。
- 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，其包含代理 HTTPS 连接所需的一个或多个额外 CA 证书。然后，Cluster Network Operator 会创建 **trusted-ca-bundle** 配置映射，将这些内容与 Red Hat Enterprise Linux CoreOS (RHCOS) 信任捆绑包合并，**Proxy** 对象的 **trustedCA** 字段中也会引用此配置映射。**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。
- 5 可选：决定 **Proxy** 对象的配置以引用 **trustedCA** 字段中 **user-ca-bundle** 配置映射的策略。允许的值是 **Proxyonly** 和 **Always**。仅在配置了 **http/https** 代理时，使用 **Proxyonly** 引用 **user-ca-bundle** 配置映射。使用 **Always** 始终引用 **user-ca-bundle** 配置映射。默认值为 **Proxyonly**。



#### 注意

安装程序不支持代理的 **readinessEndpoints** 字段。



#### 注意

如果安装程序超时，重启并使用安装程序的 **wait-for** 命令完成部署。例如：

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

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。

### 9.10.6. 将现有 AWS 安全组应用到集群

将现有 AWS 安全组应用到 control plane 和计算机器可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

## 先决条件

- 您已在 AWS 中创建安全组。如需更多信息，请参阅使用 [安全组的 AWS 文档](#)。
- 安全组必须与您要部署集群的现有 VPC 关联。安全组不能与另一个 VPC 关联。
- 您有一个现有的 `install-config.yaml` 文件。

## 流程

1. 在 `install-config.yaml` 文件中，编辑 `compute.platform.aws.additionalSecurityGroupIDs` 参数，为您的计算机指定一个或多个自定义安全组。
2. 编辑 `controlPlane.platform.aws.additionalSecurityGroupIDs` 参数，为您的 control plane 机器指定一个或多个自定义安全组。
3. 保存文件并在部署集群时引用。

## 指定自定义安全组的 `install-config.yaml` 文件示例

```
# ...
compute:
- hyperthreading: Enabled
  name: worker
  platform:
    aws:
      additionalSecurityGroupIDs:
        - sg-1 ❶
        - sg-2
  replicas: 3
controlPlane:
  hyperthreading: Enabled
  name: master
  platform:
    aws:
      additionalSecurityGroupIDs:
        - sg-3
        - sg-4
  replicas: 3
platform:
  aws:
    region: us-east-1
    subnets: ❷
    - subnet-1
    - subnet-2
    - subnet-3
```

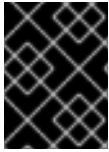
❶ 在 Amazon EC2 控制台中显示时指定安全组的名称，包括 `sg` 前缀。

❷ 指定集群使用的每个可用区的子网。

## 9.11. 通过下载二进制文件安装 OPENSIFT CLI



您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

## 9.11.1. 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. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。
5. 解包存档：

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

6. 将 **oc** 二进制文件放到 **PATH** 中的目录中。  
要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

## 9.11.2. 在 Windows 上安装 OpenShift CLI

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

### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
4. 使用 ZIP 程序解压存档。
5. 将 **oc** 二进制文件移到 **PATH** 中的目录中。  
要查看您的 **PATH**，请打开命令提示并执行以下命令：



```
C:\> path
```

## 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 9.11.3. 在 macOS 上安装 OpenShift CLI

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

## 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。



## 注意

对于 macOS arm64，请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

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**，请打开终端并执行以下命令：

```
$ echo $PATH
```

## 验证

- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

### 9.12. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案

默认情况下，管理员 secret 存储在 **kube-system** 项目中。如果您在 **install-config.yaml** 文件中将 **credentialsMode** 参数配置为 **Manual**，则必须使用以下替代方案之一：

- 要手动管理长期云凭证，请按照[手动创建长期凭证](#)中的步骤操作。
- 要实现在集群外为各个组件管理的短期凭证，请按照 [Incorporating the Cloud Credential Operator 实用程序清单](#)中的步骤操作。

#### 9.12.1. 手动创建长期凭证

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. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/' {print $3})
```

4. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 自定义资源 (CR) 列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included ❶ \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml ❷ \
  --to=<path_to_directory_for_credentials_requests> ❸
```

**❶** **--included** 参数仅包含特定集群配置所需的清单。

**❷** 指定 **install-config.yaml** 文件的位置。

**❸** 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

此命令为每个 **CredentialsRequest** 对象创建一个 YAML 文件。

### CredentialsRequest 对象示例

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
  ...
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
```

```
statementEntries:
- effect: Allow
  action:
  - iam:GetUser
  - iam:GetUserPolicy
  - iam:ListAccessKeys
  resource: "*"
...
```

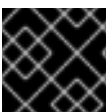
5. 在之前生成的 **openshift-install** 清单目录中为 secret 创建 YAML 文件。secret 必须使用在 **spec.secretRef** 中为每个 **CredentialsRequest** 定义的命名空间和 secret 名称存储。

### 带有 secret 的 CredentialsRequest 对象示例

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
...
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
    - effect: Allow
      action:
      - s3:CreateBucket
      - s3:DeleteBucket
      resource: "*"
    ...
  secretRef:
    name: <component_secret>
    namespace: <component_namespace>
...
```

### Secret 对象示例

```
apiVersion: v1
kind: Secret
metadata:
  name: <component_secret>
  namespace: <component_namespace>
data:
  aws_access_key_id: <base64_encoded_aws_access_key_id>
  aws_secret_access_key: <base64_encoded_aws_secret_access_key>
```



#### 重要

在升级使用手动维护凭证的集群前，您必须确保 CCO 处于可升级状态。

## 9.12.2. 将 AWS 集群配置为使用短期凭证

要安装配置为使用 AWS 安全令牌服务 (STS) 的集群，您必须配置 CCO 实用程序并为集群创建所需的 AWS 资源。

### 9.12.2.1. 配置 Cloud Credential Operator 工具

当 Cloud Credential Operator(CCO)以手动模式运行时，要从集群外部创建和管理云凭证，提取并准备 CCO 实用程序(**ccctl**)二进制文件。



#### 注意

**ccctl** 工具是在 Linux 环境中运行的 Linux 二进制文件。

#### 先决条件

- 您可以访问具有集群管理员权限的 OpenShift Container Platform 帐户。
- 已安装 OpenShift CLI(**oc**)。
- 您已为 **ccctl** 工具创建了用于以下权限的 AWS 帐户：

#### 例 9.3. 所需的 AWS 权限

##### 所需的 iam 权限

- **iam:CreateOpenIDConnectProvider**
- **iam:CreateRole**
- **iam>DeleteOpenIDConnectProvider**
- **iam>DeleteRole**
- **iam>DeleteRolePolicy**
- **iam:GetOpenIDConnectProvider**
- **iam:GetRole**
- **iam:GetUser**
- **iam:ListOpenIDConnectProviders**
- **iam:ListRolePolicies**
- **iam:ListRoles**
- **iam:PutRolePolicy**
- **iam:TagOpenIDConnectProvider**
- **iam:TagRole**

##### 所需的 s3 权限

- **s3:CreateBucket**

- **s3:DeleteBucket**
- **s3:DeleteObject**
- **s3:GetBucketAcl**
- **s3:GetBucketTagging**
- **s3:GetObject**
- **s3:GetObjectAcl**
- **s3:GetObjectTagging**
- **s3:ListBucket**
- **s3:PutBucketAcl**
- **s3:PutBucketPolicy**
- **s3:PutBucketPublicAccessBlock**
- **s3:PutBucketTagging**
- **s3:PutObject**
- **s3:PutObjectAcl**
- **s3:PutObjectTagging**

所需的 cloudfront 权限

- **cloudfront:ListCloudFrontOriginAccessIdentities**
- **cloudfront:ListDistributions**
- **cloudfront:ListTagsForResource**

如果您计划通过公共 CloudFront 发行版 URL 将 OIDC 配置存储在 IAM 身份提供程序访问的私有 S3 存储桶中，则运行 **ccoctl** 工具的 AWS 帐户需要以下额外权限：

#### 例 9.4. 使用 CloudFront 私有 S3 存储桶的额外权限

- **cloudfront:CreateCloudFrontOriginAccessIdentity**
- **cloudfront:CreateDistribution**
- **cloudfront>DeleteCloudFrontOriginAccessIdentity**
- **cloudfront>DeleteDistribution**
- **cloudfront:GetCloudFrontOriginAccessIdentity**
- **cloudfront:GetCloudFrontOriginAccessIdentityConfig**

- **cloudfront:GetDistribution**
- **cloudfront:TagResource**
- **cloudfront:UpdateDistribution**



### 注意

在使用 **ccoctl aws create-all** 命令处理凭证请求时，这些额外权限支持使用 **--create-private-s3-bucket** 选项。

## 流程

1. 运行以下命令，为 OpenShift Container Platform 发行镜像设置变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像获取 CCO 容器镜像：

```
$ CCO_IMAGE=$(oc adm release info --image-for='cloud-credential-operator'
$RELEASE_IMAGE -a ~/.pull-secret)
```



### 注意

确保 **\$RELEASE\_IMAGE** 的架构与将使用 **ccoctl** 工具的环境架构相匹配。

3. 运行以下命令，将 CCO 容器镜像中的 **ccoctl** 二进制文件提取到 OpenShift Container Platform 发行镜像中：

```
$ oc image extract $CCO_IMAGE --file="/usr/bin/ccoctl" -a ~/.pull-secret
```

4. 运行以下命令更改权限以使 **ccoctl** 可执行：

```
$ chmod 775 ccoctl
```

## 验证

- 要验证 **ccoctl** 是否准备就绪，可以尝试显示帮助文件。运行命令时使用相对文件名，例如：

```
$ ./ccoctl.rhel9
```

## 输出示例

```
OpenShift credentials provisioning tool
```

```
Usage:
ccoctl [command]
```

```
Available Commands:
alibabacloud Manage credentials objects for alibaba cloud
```

```
aws      Manage credentials objects for AWS cloud
azure    Manage credentials objects for Azure
gcp      Manage credentials objects for Google cloud
help     Help about any command
ibmcloud Manage credentials objects for IBM Cloud
nutanix  Manage credentials objects for Nutanix
```

**Flags:**

```
-h, --help  help for ccoctl
```

Use "ccoctl [command] --help" for more information about a command.

### 9.12.2.2. 使用 Cloud Credential Operator 实用程序创建 AWS 资源

创建 AWS 资源时有以下选项：

- 您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。这是创建资源的最快速方法。请参阅[使用单个命令创建 AWS 资源](#)。
- 如果您需要在修改 AWS 资源前查看 **ccoctl** 工具创建的 JSON 文件，或者 **ccoctl** 工具用于创建 AWS 资源的过程无法自动满足组织的要求，您可以单独创建 AWS 资源。请参阅[单独创建 AWS 资源](#)。

#### 9.12.2.2.1. 使用单个命令创建 AWS 资源

如果 **ccoctl** 工具用于创建 AWS 资源的过程自动满足机构的要求，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。

否则，您可以单独创建 AWS 资源。如需更多信息，请参阅["单独创建 AWS 资源"](#)。

**注意**

默认情况下，**ccoctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccoctl\_output\_dir>** 来引用这个目录。

**先决条件**

您必须：

- 提取并准备好 **ccoctl** 二进制文件。

**流程**

1. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
```

```
--included \ ❶
--install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \ ❷
--to=<path_to_directory_for_credentials_requests> \ ❸
```

- ❶ **--included** 参数仅包含特定集群配置所需的清单。
- ❷ 指定 **install-config.yaml** 文件的位置。
- ❸ 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。



### 注意

此命令可能需要一些时间才能运行。

3. 运行以下命令，使用 **ccctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccctl aws create-all \
  --name=<name> \ ❶
  --region=<aws_region> \ ❷
  --credentials-requests-dir=<path_to_credentials_requests_directory> \ ❸
  --output-dir=<path_to_ccctl_output_dir> \ ❹
  --create-private-s3-bucket \ ❺
```

- ❶ 指定用于标记创建用于跟踪的任何云资源的名称。
- ❷ 指定在其中创建云资源的 AWS 区域。
- ❸ 指定包含组件 **CredentialsRequest** 对象文件的目录。
- ❹ 可选：指定您希望 **ccctl** 实用程序在其中创建对象的目录。默认情况下，实用程序在运行命令的目录中创建对象。
- ❺ 可选：默认情况下，**ccctl** 实用程序将 OpenID Connect (OIDC) 配置文件存储在公共 S3 存储桶中，并使用 S3 URL 作为公共 OIDC 端点。要将 OIDC 配置存储在 IAM 身份提供程序通过公共 CloudFront 发行版 URL 访问的专用 S3 存储桶中，请使用 **--create-private-s3-bucket** 参数。



### 注意

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

## 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccctl_output_dir>/manifests
```

## 输出示例



```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

#### 9.12.2.2.2. 单独创建 AWS 资源

您可以使用 **ccoctl** 工具单独创建 AWS 资源。这个选项对于在不同用户或部门之间创建这些资源的组织可能很有用。

否则，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。如需更多信息，请参阅“使用单个命令创建 AWS 资源”。



#### 注意

默认情况下，**ccoctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccoctl\_output\_dir>** 来引用这个目录。

有些 **ccoctl** 命令会发出 AWS API 调用来创建或修改 AWS 资源。您可以使用 **--dry-run** 标志来避免 API 调用。使用此标志可在本地文件系统中创建 JSON 文件。您可以使用 **--cli-input-json** 参数查看和修改 JSON 文件，然后使用 AWS CLI 工具应用它们。

#### 先决条件

- 提取并准备 **ccoctl** 二进制文件。

#### 流程

- 运行以下命令，生成用于为集群设置 OpenID Connect 供应商的公共和私有 RSA 密钥文件：

```
$ ccoctl aws create-key-pair
```

#### 输出示例

```
2021/04/13 11:01:02 Generating RSA keypair
2021/04/13 11:01:03 Writing private key to /<path_to_ccoctl_output_dir>/serviceaccount-signer.private
2021/04/13 11:01:03 Writing public key to /<path_to_ccoctl_output_dir>/serviceaccount-signer.public
2021/04/13 11:01:03 Copying signing key for use by installer
```

其中 **serviceaccount-signer.private** 和 **serviceaccount-signer.public** 是生成的密钥文件。

此命令还会在 **/<path\_to\_ccoctl\_output\_dir>/tls/bound-service-account-signing-key.key** 中创建集群在安装过程中所需的私钥。

2. 运行以下命令，在 AWS 上创建 OpenID Connect 身份提供程序和 S3 存储桶：

```
$ ccoctl aws create-identity-provider \
  --name=<name> \ ❶
  --region=<aws_region> \ ❷
  --public-key-file=<path_to_ccoctl_output_dir>/serviceaccount-signer.public ❸
```

- ❶ <name> 是用于标记为跟踪而创建的云资源的名称。
- ❷ <aws-region> 是将来要在其中创建云资源的 AWS 区域。
- ❸ <path\_to\_ccoctl\_output\_dir> 是 **ccoctl aws create-key-pair** 命令生成的公钥文件的路径。

### 输出示例

```
2021/04/13 11:16:09 Bucket <name>-oidc created
2021/04/13 11:16:10 OpenID Connect discovery document in the S3 bucket <name>-oidc at
.well-known/openid-configuration updated
2021/04/13 11:16:10 Reading public key
2021/04/13 11:16:10 JSON web key set (JWKS) in the S3 bucket <name>-oidc at keys.json
updated
2021/04/13 11:16:18 Identity Provider created with ARN: arn:aws:iam::
<aws_account_id>:oidc-provider/<name>-oidc.s3.<aws_region>.amazonaws.com
```

其中 **openid-configuration** 是发现文档和 **key.json** 是一个 JSON Web 密钥集文件。

此命令还会在 **/<path\_to\_ccoctl\_output\_dir>/manifests/cluster-authentication-02-config.yaml** 中创建 YAML 配置文件。此文件为集群生成的服务帐户令牌设置签发者 URL 字段，以便 AWS IAM 身份提供程序信任令牌。

3. 为集群中的每个组件创建 IAM 角色：

- a. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

- b. 从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \ ❶
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
  ❷
  --to=<path_to_directory_for_credentials_requests> ❸
```

- ❶ **--included** 参数仅包含特定集群配置所需的清单。
- ❷ 指定 **install-config.yaml** 文件的位置。
- ❸ 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

c. 运行以下命令，使用 **ccctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccctl aws create-iam-roles \
  --name=<name> \
  --region=<aws_region> \
  --credentials-requests-dir=<path_to_credentials_requests_directory> \
  --identity-provider-arn=arn:aws:iam::<aws_account_id>:oidc-provider/<name>-oidc.s3.
  <aws_region>.amazonaws.com
```



### 注意

对于使用其他 IAM API 端点的 AWS 环境（如 GovCloud），还必须使用 **--region** 参数指定您的区域。

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

对于每个 **CredentialsRequest** 对象，**ccctl** 创建一个带有信任策略的 IAM 角色，该角色与指定的 OIDC 身份提供程序相关联，以及来自 OpenShift Container Platform 发行镜像的每个 **CredentialsRequest** 对象中定义的权限策略。

## 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccctl_output_dir>/manifests
```

## 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

### 9.12.2.3. 整合 Cloud Credential Operator 实用程序清单

要为单个组件在集群外实现短期安全凭证，您必须将创建 Cloud Credential Operator 实用程序 (**ccctl**) 的清单文件移到安装程序的正确目录中。

## 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您已配置了 Cloud Credential Operator 实用程序 (**ccctl**)。

- 已使用 **ccoctl** 工具创建了集群所需的云供应商资源。

## 流程

1. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，将 **ccoctl** 工具生成的清单复制到安装程序创建的 **manifests** 目录中：

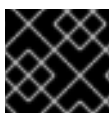
```
$ cp /<path_to_ccoctl_output_dir>/manifests/* ./manifests/
```

4. 将包含私钥的 **tls** 目录复制到安装目录中：

```
$ cp -a /<path_to_ccoctl_output_dir>/tls .
```

## 9.13. 部署集群

您可以在兼容云平台上安装 OpenShift Container Platform。



### 重要

在初始安装过程中，您只能运行安装程序的 **create cluster** 命令一次。

### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。
- 已确认主机上的云供应商帐户具有部署集群的正确权限。权限不正确的帐户会导致安装过程失败，并显示包括缺失权限的错误消息。

## 流程

1. 进入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`，而不是 `info`。

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



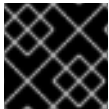
### 注意

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

## 验证

当集群部署成功完成时：

- 终端会显示用于访问集群的说明，包括指向 Web 控制台和 **kubeadmin** 用户的凭证的链接。
- 凭证信息还会输出到 `<installation_directory>/openshift_install.log`。

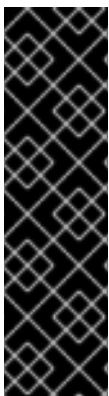


### 重要

不要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。

## 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: "password"
INFO Time elapsed: 36m22s
```



### 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。一个例外是，您必须手动批准待处理的 **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。

## 9.14. 使用 CLI 登录集群

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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

### 先决条件

- 已部署 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. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

## 输出示例

```
system:admin
```

/validating-an-installation.adoc

## 9.15. 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。

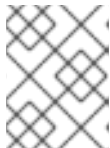
### 先决条件

- 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
```

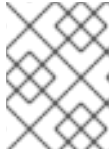


### 注意

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 控制台的更多详情](#)，请参阅 [访问 Web 控制台](#)。

## 9.16. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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](#)。

## 9.17. 后续步骤

- [验证安装](#)
- [自定义集群](#)
- [远程健康报告](#)
- [删除云供应商凭证](#)

## 第 10 章 在 AWS 上将集群安装到 SECRET 或 TOP SECRET 区域

在 OpenShift Container Platform 版本 4.14 中，您可以在 Amazon Web Services (AWS) 上将集群安装到以下 secret 区域：

- Secret Commercial Cloud Services (SC2S)
- 商业云服务 (C2S)

要在任一区域中配置集群，请在安装集群前更改 **install config.yaml** 文件中的参数。

### 10.1. 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。
- 您可以阅读[选择集群安装方法并为用户准备它的文档](#)。
- 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，[将其配置为允许集群需要访问的站点](#)。

### 10.2. AWS SECRET 区域

支持以下 AWS secret 分区：

- **us-isob-east-1** (SC2S)
- **us-iso-east-1** (C2S)



#### 注意

AWS SC2S 和 C2S 区域的最大支持 MTU 与 AWS 商业区域不同。有关在安装过程中配置 MTU 的更多信息，请参阅 [使用网络自定义在 AWS 中安装集群中的 Cluster Network Operator 配置对象](#)部分。

### 10.3. 安装要求

红帽不会为 AWS Secret 和 Top Secret 区域发布 Red Hat Enterprise Linux CoreOS (RHCOS) Amazon Machine Image。

在安装集群前，您必须：

- 上传自定义 RHCOS AMI。
- 手动创建安装配置文件 (**install-config.yaml**)。



- 在安装配置文件中指定 AWS 区域和附带的自定义 AMI。

您不能使用 OpenShift Container Platform 安装程序创建安装配置文件。安装程序不会列出没有原生支持 RHCOS AMI 的 AWS 区域。



### 重要

您还必须在 `install-config.yaml` 文件的 `additionalTrustBundle` 字段中定义自定义 CA 证书，因为 AWS API 需要自定义 CA 信任捆绑包。要允许安装程序访问 AWS API，还必须在运行安装程序的机器上定义 CA 证书。您必须将 CA 捆绑包添加到机器上的信任存储中，使用 `AWS_CA_BUNDLE` 环境变量，或者在 AWS 配置文件的 `ca_bundle` 字段中定义 CA 捆绑包。

## 10.4. 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 顶级 Secret 区域的 Route 53 不支持公共区。因此，如果集群部署到 AWS 顶级 Secret 区域，则集群必须是私有的。

默认情况下，OpenShift Container Platform 被置备为使用可公开访问的 DNS 和端点。在部署集群时，私有集群会将 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，且无法在互联网中看到。



### 重要

如果集群有任何公共子网，管理员创建的负载均衡器服务可能会公开访问。为确保集群安全性，请验证这些服务是否已明确标注为私有。

要部署私有集群，您必须：

- 使用满足您的要求的现有网络。集群资源可能会在网络上的其他集群间共享。
- 从有权访问的机器中部署：
  - 您置备的云的 API 服务。
  - 您调配的网络上的主机。
  - 用于获取安装介质的互联网。

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。

### 10.4.1. 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。

#### 10.4.1.1. 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。

## 10.5. 关于使用自定义 VPC

在 OpenShift Container Platform 4.14 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。

### 10.5.1. 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。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](#) 以及 [处理 VPC 和子网](#)。

安装程序无法：

- 分割网络范围供集群使用。
- 设置子网的路由表。
- 设置 VPC 选项，如 DHCP。

您必须在安装集群前完成这些任务。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 [您的 VPC 的路由表](#)。

您的 VPC 必须满足以下特征：

- VPC 不能使用 **kubernetes.io/cluster/.\*: owned, Name**, 和 **openshift.io/cluster** 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。您不能使用 **Name** 标签，因为它与 EC2 **Name** 字段重叠，且安装失败。
- 您需要在您的 VPC 中启用 **enableDnsSupport** 和 **enableDnsHostnames** 属性，以便集群可以使用附加到 VPC 中的 Route 53 区来解析集群内部的 DNS 记录。请参阅 AWS 文档中的 [您的 VPC 中的 DNS 支持](#) 部分。  
如果要使用您自己的 Route 53 托管私有区，您必须在安装集群前将现有托管区与 VPC 相关联。您可以使用 **install-config.yaml** 文件中的 **platform.aws.hostedZone** 和 **platform.aws.hostedZoneRole** 字段定义托管区。您可以通过与安装集群的帐户共享来使用来自另一个帐户的私有托管区。如果使用另一个帐户的私有托管区，则必须使用 **Passthrough** 或 **Manual** 凭证模式。

SC2S 或 C2S 区域中的一个集群无法访问 EC2、ELB 和 S3 端点的公共 IP 地址。根据您要在安装过程中限制互联网流量的级别，有以下配置选项：

#### 10.5.1.1. 选项 1：创建 VPC 端点

创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

##### SC2S

- **elasticloadbalancing.<aws\_region>.sc2s.sgov.gov**
- **ec2.<aws\_region>.sc2s.sgov.gov**
- **s3.<aws\_region>.sc2s.sgov.gov**

##### C2S

- **elasticloadbalancing.<aws\_region>.c2s.ic.gov**
- **ec2.<aws\_region>.c2s.ic.gov**
- **s3.<aws\_region>.c2s.ic.gov**

通过这个选项，网络流量在 VPC 和所需的 AWS 服务之间保持私有。

### 10.5.1.2. 选项 2：创建一个没有 VPC 端点的代理

作为安装过程的一部分，您可以配置 HTTP 或 HTTPS 代理。使用此选项时，互联网流量会通过代理访问所需的 AWS 服务。

### 10.5.1.3. 选项 3：创建带有 VPC 端点的代理

作为安装过程的一部分，您可以使用 VPC 端点配置 HTTP 或 HTTPS 代理。创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

#### SC2S

- **elasticloadbalancing.<aws\_region>.sc2s.sgov.gov**
- **ec2.<aws\_region>.sc2s.sgov.gov**
- **s3.<aws\_region>.sc2s.sgov.gov**

#### C2S

- **elasticloadbalancing.<aws\_region>.c2s.ic.gov**
- **ec2.<aws\_region>.c2s.ic.gov**
- **s3.<aws\_region>.c2s.ic.gov**

在 **install-config.yaml** 文件中配置代理时，将这些端点添加到 **noProxy** 字段。通过这个选项，代理会阻止集群直接访问互联网。但是，您的 VPC 和所需的 AWS 服务之间网络流量保持私有。

### 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：                                                                                                   |             |
|        |                                                                                                                                                                                                                                                                                                                                       | 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。                                   |             |

### 10.5.2. 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。

### 10.5.3. 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：不同的个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。

### 10.5.4. 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)。

### 10.5.5. AWS 安全组

默认情况下，安装程序会创建安全组并将其附加到 control plane 和计算机器。不可修改与默认安全组关联的规则。

但是，您可以将与现有 VPC 关联的其他现有 AWS 安全组应用到 control plane 和计算机器。应用自定义安全组可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

作为安装过程的一部分，您可以在部署集群前通过修改 **install-config.yaml** 文件来应用自定义安全组。

如需更多信息，请参阅“将现有 AWS 安全组应用到集群”。

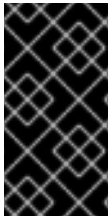
## 10.6. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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型的基础架构上执行受限网络安装。在此过程中，您可以下载所需的内容，并使用它为镜像 registry 填充安装软件包。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。在更新集群前，您要更新镜像 registry 的内容。

## 10.7. 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
```

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
```

3. 将上传至 Amazon S3 的 RHCOS 版本导出为环境变量：

```
$ export RHCOS_VERSION=<version> 1
```

1 1 1 RHCOS VMDK 版本，如 **4.14.0**。

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-4.7.0-x86_64-aws.x86_64",
      "ImportTaskId": "import-snap-fh6i8uil",
      "SnapshotTaskDetail": {
        "Description": "rhcos-4.7.0-x86_64-aws.x86_64",
        "DiskImageSize": 819056640.0,
        "Format": "VMDK",
        "SnapshotId": "snap-06331325870076318",
        "Status": "completed",
        "UserBucket": {
          "S3Bucket": "external-images",
          "S3Key": "rhcos-4.7.0-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>}' 4
```

- 1 RHCOS VMDK 架构类型，如 **x86\_64**, **aarch64**, **s390x**, 或 **ppc64le**。
- 2 来自导入快照的 **Description**。
- 3 RHCOS AMI 的名称。
- 4 导入的快照中的 **SnapshotID**。

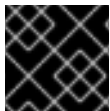
如需了解更多有关这些 API 的信息，请参阅 AWS 文档 [导入快照](#) 和 [创建由 EBS 支持的 AMI](#)。

## 10.8. 为集群节点 SSH 访问生成密钥对

在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 **core** 用户的 `~/.ssh/authorized_keys` 列表中，这将启用免密码身份验证。

将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。

如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。`./openshift-install gather` 命令还需要在集群节点上设置 SSH 公钥。



### 重要

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。



### 注意

您必须使用本地密钥，而不是使用特定平台方法配置的密钥，如 [AWS 密钥对](#)。

### 流程

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，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_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。



### 注意

如果您计划在 **x86\_64**、**ppc64le** 和 **s390x** 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 **ed25519** 算法的密钥。相反，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

## 2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

3. 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。

**注意**

在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- a. 如果 `ssh-agent` 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

**输出示例**

```
Agent pid 31874
```

**注意**

如果集群处于 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。

4. 将 SSH 私钥添加到 `ssh-agent`：

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

- 1** 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

**输出示例**

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

**后续步骤**

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

## 10.9. 获取安装程序

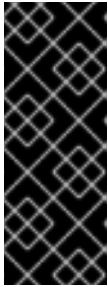
在安装 OpenShift Container Platform 前，将安装文件下载到您用于安装的主机上。

**先决条件**

- 您有一台运行 Linux 或 macOS 的计算机，至少有 1.2 GB 本地磁盘空间。

## 流程

1. 进入 Red Hat Hybrid Cloud Console 上的 [Cluster Type](#) 页。如果您有红帽帐户，请使用您的凭证登录。如果没有，请创建一个帐户。
2. 在页的 **Run it yourself** 部分中选择您的基础架构供应商。
3. 从 **OpenShift 安装程序** 下的下拉菜单中选择您的主机操作系统和架构，然后点**下载安装程序**。
4. 将下载的文件保存在要存储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几个文件。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。删除集群需要这两个文件。
- 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，请为特定云供应商完成 OpenShift Container Platform 卸载流程。

5. 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
```

6. 从 [Red Hat OpenShift Cluster Manager](#) 下载安装 [pull secret](#)。此 pull secret 允许您与所含授权机构提供的服务进行身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

## 提示

另外，您还可以从[红帽客户门户网站](#)检索安装程序，您可以在其中指定要下载的安装程序版本。但是，您需要有一个有效的订阅才能访问此页。

## 10.10. 手动创建安装配置文件

安装集群需要您手动生成安装配置文件。

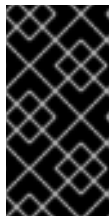
### 先决条件

- 您上传了一个自定义 RHCOS AMI。
- 在本地机器上有一个 SSH 公钥用于安装程序。您可以使用密钥在集群节点上进行 SSH 身份验证，以进行调试和灾难恢复。
- 已获取 OpenShift Container Platform 安装程序和集群的 pull secret。

## 流程

1. 创建一个安装目录来存储所需的安装资产：

```
$ mkdir <installation_directory>
```

**重要**

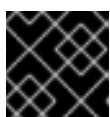
您必须创建一个目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。

2. 对提供的 **install-config.yaml** 文件模板示例进行自定义，并将文件保存到 **<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，以便使用它来安装多个集群。

**重要**

现在备份 **install-config.yaml** 文件，因为安装过程会在下一步中消耗该文件。

**其他资源**

- [AWS 的安装配置参数](#)

**10.10.1. 为 AWS 测试的实例类型**

以下 Amazon Web Services(AWS) 实例类型已经过 OpenShift Container Platform 测试。

**注意**

将以下图中包含的机器类型用于 AWS 实例。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

**例 10.1. 基于 64 位 x86 架构的机器类型用于 secret 区域**

- **c4.\***
- **c5.\***
- **i3.\***
- **m4.\***
- **m5.\***
- **r4.\***
- **r5.\***
- **t3.\***

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

您可以自定义安装配置文件 (**install-config.yaml**)，以指定有关 OpenShift Container Platform 集群平台的更多详细信息，或修改所需参数的值。



### 重要

此示例 YAML 文件仅供参考。使用它作为资源，在您手动创建的安装配置文件中输入参数值。

```
apiVersion: v1
baseDomain: example.com 1
credentialsMode: Mint 2
controlPlane: 3 4
  hyperthreading: Enabled 5
  name: master
  platform:
    aws:
      zones:
      - us-iso-east-1a
      - us-iso-east-1b
      rootVolume:
        iops: 4000
        size: 500
        type: io1 6
      metadataService:
        authentication: Optional 7
        type: m6i.xlarge
      replicas: 3
compute: 8
- hyperthreading: Enabled 9
  name: worker
  platform:
    aws:
      rootVolume:
        iops: 2000
        size: 500
        type: io1 10
      metadataService:
        authentication: Optional 11
        type: c5.4xlarge
      zones:
      - us-iso-east-1a
      - us-iso-east-1b
      replicas: 3
metadata:
  name: test-cluster 12
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
```

```

networkType: OVNKubernetes 13
serviceNetwork:
- 172.30.0.0/16
platform:
aws:
  region: us-iso-east-1 14
  propagateUserTags: true 15
  userTags:
    adminContact: jdoe
    costCenter: 7536
  subnets: 16
  - subnet-1
  - subnet-2
  - subnet-3
  amiID: ami-96c6f8f7 17 18
  serviceEndpoints: 19
  - name: ec2
    url: https://vpce-id.ec2.us-west-2.vpce.amazonaws.com
  hostedZone: Z3URY6TWQ91KVV 20
fips: false 21
sshKey: ssh-ed25519 AAAA... 22
publish: Internal 23
pullSecret: '{"auths": ...}' 24
additionalTrustBundle: | 25
  -----BEGIN CERTIFICATE-----
  <MY_TRUSTED_CA_CERT>
  -----END CERTIFICATE-----

```

1 12 14 17 24 必需。

2 可选：添加此参数来强制 Cloud Credential Operator (CCO) 使用指定的模式。默认情况下，CCO 使用 **kube-system** 命名空间中的 root 凭证来动态尝试决定凭证的功能。有关 CCO 模式的详情，请参阅 [身份验证和授权指南](#) 中的 "About the Cloud Credential Operator" 部分。

3 8 15 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。仅使用一个 control plane 池。

5 9 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置为 **Disabled** 来禁用它。如果在某些集群机器中禁用并发多线程，则必须在所有集群机器中禁用它。



### 重要

如果您禁用并发多线程，请确保您的容量规划考虑机器性能显著降低的情况。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 10 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。

7 11

是否需要 [Amazon EC2 实例元数据服务 v2](#) (IMDSv2)。为了要求 IMDSv2，请将参数值设置为 **Required**。要允许使用 IMDSv1 和 IMDSv2，请将参数值设置为 **Optional**。如果没有指定值，则允



### 注意

在集群安装过程中设置的 control plane 机器的 IMDS 配置只能使用 AWS CLI 更改。可以使用计算机器集来更改计算机器的 IMDS 配置。

- 13 要安装的集群网络插件。支持的值有 **OVNKubernetes** 和 **OpenShiftSDN**。默认值为 **OVNKubernetes**。
- 16 如果您提供自己的 VPC，为集群使用的每个可用区指定子网。
- 18 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。
- 19 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。
- 20 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。
- 21 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

要为集群启用 FIPS 模式，您必须从配置为以 FIPS 模式操作的 Red Hat Enterprise Linux (RHEL) 计算机运行安装程序。有关在 RHEL 中配置 FIPS 模式的更多信息，请参阅[在 FIPS 模式中安装该系统](#)。当以 FIPS 模式运行 Red Hat Enterprise Linux (RHEL) 或 Red Hat Enterprise Linux CoreOS (RHCOS) 时，OpenShift Container Platform 核心组件使用 RHEL 加密库，在 x86\_64、ppc64le 和 s390x 架构上提交到 NIST FIPS 140-2/140-3 Validation。

- 22 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。

- 23 如何发布集群的面向用户的端点。将 **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**。
- 25 自定义 CA 证书。当部署到 SC2S 或 C2S 区域时这是必需的，因为 AWS API 需要自定义 CA 信任捆绑包。

## 10.10.3. 在安装过程中配置集群范围的代理

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，并确定它们中的任何站点是否需要绕过代理。默认情况下，所有集群出口流量都经过代理，包括对托管云供应商 API 的调用。如果需要，您将在 **Proxy** 对象的 **spec.noProxy** 字段中添加站点来绕过代理。



### 注意

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

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

## 流程

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> ❶
  httpsProxy: https://<username>:<pswd>@<ip>:<port> ❷
  noProxy: ec2.<aws_region>.amazonaws.com,elasticloadbalancing.
    <aws_region>.amazonaws.com,s3.<aws_region>.amazonaws.com ❸
  additionalTrustBundle: | ❹
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> ❺
```

- ❶ 用于创建集群外 HTTP 连接的代理 URL。URL 方案必须是 **http**。
- ❷ 用于创建集群外 HTTPS 连接的代理 URL。
- ❸ 要从代理中排除的目标域名、IP 地址或其他网络 CIDR 的逗号分隔列表。在域前面加上 **.** 以仅匹配子域。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。如果您已将 Amazon **EC2**、**Elastic Load Balancing** 和 **S3** VPC 端点添加到 VPC 中，您必须将这些端点添加到 **noProxy** 字段。
- ❹ 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，其包含代理 HTTPS 连接所需的一个或多个额外 CA 证书。然后，Cluster Network Operator 会创建 **trusted-ca-bundle** 配置映射，将这些内容与 Red Hat Enterprise Linux CoreOS (RHCOS) 信任捆绑包合并，**Proxy** 对象的 **trustedCA** 字段中也会引用此配置映射。**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。
- ❺ 可选：决定 **Proxy** 对象的配置以引用 **trustedCA** 字段中 **user-ca-bundle** 配置映射的策略。允许的值是 **Proxyonly** 和 **Always**。仅在配置了 **http/https** 代理时，使用 **Proxyonly** 引用 **user-ca-bundle** 配置映射。使用 **Always** 始终引用 **user-ca-bundle** 配置映射。默认



值为 **Proxyonly**。



### 注意

安装程序不支持代理的 **readinessEndpoints** 字段。



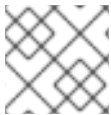
### 注意

如果安装程序超时，重启并使用安装程序的 **wait-for** 命令完成部署。例如：

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

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。

## 10.10.4. 将现有 AWS 安全组应用到集群

将现有 AWS 安全组应用到 control plane 和计算机器可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

### 先决条件

- 您已在 AWS 中创建安全组。如需更多信息，请参阅使用 [安全组的 AWS 文档](#)。
- 安全组必须与您要将集群部署到的现有 VPC 关联。安全组不能与另一个 VPC 关联。
- 您有一个现有的 **install-config.yaml** 文件。

### 流程

1. 在 **install-config.yaml** 文件中，编辑 **compute.platform.aws.additionalSecurityGroupIDs** 参数，为您的计算机器指定一个或多个自定义安全组。
2. 编辑 **controlPlane.platform.aws.additionalSecurityGroupIDs** 参数，为您的 control plane 机器指定一个或多个自定义安全组。
3. 保存文件并在部署集群时引用。

## 指定自定义安全组的 **install-config.yaml** 文件示例

```
# ...
compute:
- hyperthreading: Enabled
  name: worker
  platform:
    aws:
```

```

    additionalSecurityGroupIDs:
      - sg-1 1
      - sg-2
    replicas: 3
  controlPlane:
    hyperthreading: Enabled
    name: master
    platform:
      aws:
        additionalSecurityGroupIDs:
          - sg-3
          - sg-4
        replicas: 3
    platform:
      aws:
        region: us-east-1
        subnets: 2
          - subnet-1
          - subnet-2
          - subnet-3

```

**1** 在 Amazon EC2 控制台中显示时指定安全组的名称，包括 **sg** 前缀。

**2** 指定集群使用的每个可用区的子网。

## 10.11. 通过下载二进制文件安装 OPENSIFT CLI

您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

### 10.11.1. 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. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。
5. 解包存档：

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

- 将 **oc** 二进制文件放到 **PATH** 中的目录中。  
要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 10.11.2. 在 Windows 上安装 OpenShift CLI

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

#### 流程

- 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
- 从 **版本** 下拉列表中选择适当的版本。
- 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
- 使用 ZIP 程序解压存档。
- 将 **oc** 二进制文件移到 **PATH** 中的目录中。  
要查看您的 **PATH**，请打开命令提示并执行以下命令：

```
C:\> path
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 10.11.3. 在 macOS 上安装 OpenShift CLI

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

#### 流程

- 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
- 从 **版本** 下拉列表中选择适当的版本。
- 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。



#### 注意

对于 macOS arm64，请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

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**，请打开终端并执行以下命令：

```
$ echo $PATH
```

## 验证

- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

## 10.12. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案

默认情况下，管理员 secret 存储在 **kube-system** 项目中。如果您在 **install-config.yaml** 文件中将 **credentialsMode** 参数配置为 **Manual**，则必须使用以下替代方案之一：

- 要手动管理长期云凭证，请按照[手动创建长期凭证](#)中的步骤操作。
- 要实现在集群外为各个组件管理的短期凭证，请按照[配置 AWS 集群以使用短期凭证](#)中的步骤操作。

### 10.12.1. 手动创建长期凭证

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. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

4. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 自定义资源 (CR) 列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \ ❶
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \ ❷
  --to=<path_to_directory_for_credentials_requests> \ ❸
```

- ❶ **--included** 参数仅包含特定集群配置所需的清单。
- ❷ 指定 **install-config.yaml** 文件的位置。
- ❸ 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

此命令为每个 **CredentialsRequest** 对象创建一个 YAML 文件。

### CredentialsRequest 对象示例

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

5. 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。

### 带有 secret 的 CredentialsRequest 对象示例

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
...
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
```

```

- effect: Allow
  action:
  - s3:CreateBucket
  - s3:DeleteBucket
  resource: "*"
...
secretRef:
  name: <component_secret>
  namespace: <component_namespace>
...

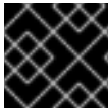
```

### Secret 对象示例

```

apiVersion: v1
kind: Secret
metadata:
  name: <component_secret>
  namespace: <component_namespace>
data:
  aws_access_key_id: <base64_encoded_aws_access_key_id>
  aws_secret_access_key: <base64_encoded_aws_secret_access_key>

```



#### 重要

在升级使用手动维护凭证的集群前，您必须确保 CCO 处于可升级状态。

## 10.12.2. 将 AWS 集群配置为使用短期凭证

要安装配置为使用 AWS 安全令牌服务 (STS) 的集群，您必须配置 CCO 实用程序并为集群创建所需的 AWS 资源。

### 10.12.2.1. 配置 Cloud Credential Operator 工具

当 Cloud Credential Operator (CCO) 以手动模式运行时，要从集群外部创建和管理云凭证，提取并准备 CCO 实用程序 (**ccoctl**) 二进制文件。



#### 注意

**ccoctl** 工具是在 Linux 环境中运行的 Linux 二进制文件。

#### 先决条件

- 您可以访问具有集群管理员权限的 OpenShift Container Platform 帐户。
- 已安装 OpenShift CLI (**oc**)。
- 您已为 **ccoctl** 工具创建了用于以下权限的 AWS 帐户：

#### 例 10.2. 所需的 AWS 权限

##### 所需的 iam 权限

- **iam:CreateOpenIDConnectProvider**

- **iam:CreateRole**
- **iam>DeleteOpenIDConnectProvider**
- **iam>DeleteRole**
- **iam>DeleteRolePolicy**
- **iam:GetOpenIDConnectProvider**
- **iam:GetRole**
- **iam:GetUser**
- **iam:ListOpenIDConnectProviders**
- **iam:ListRolePolicies**
- **iam:ListRoles**
- **iam:PutRolePolicy**
- **iam:TagOpenIDConnectProvider**
- **iam:TagRole**

#### 所需的 s3 权限

- **s3:CreateBucket**
- **s3>DeleteBucket**
- **s3>DeleteObject**
- **s3:GetBucketAcl**
- **s3:GetBucketTagging**
- **s3:GetObject**
- **s3:GetObjectAcl**
- **s3:GetObjectTagging**
- **s3:ListBucket**
- **s3:PutBucketAcl**
- **s3:PutBucketPolicy**
- **s3:PutBucketPublicAccessBlock**
- **s3:PutBucketTagging**
- **s3:PutObject**

- **s3:PutObjectAcl**
- **s3:PutObjectTagging**

所需的 **cloudfront** 权限

- **cloudfront:ListCloudFrontOriginAccessIdentities**
- **cloudfront:ListDistributions**
- **cloudfront:ListTagsForResource**

如果您计划通过公共 CloudFront 发行版 URL 将 OIDC 配置存储在 IAM 身份提供程序访问的私有 S3 存储桶中，则运行 **ccoctl** 工具的 AWS 帐户需要以下额外权限：

#### 例 10.3. 使用 CloudFront 私有 S3 存储桶的额外权限

- **cloudfront:CreateCloudFrontOriginAccessIdentity**
- **cloudfront:CreateDistribution**
- **cloudfront>DeleteCloudFrontOriginAccessIdentity**
- **cloudfront>DeleteDistribution**
- **cloudfront:GetCloudFrontOriginAccessIdentity**
- **cloudfront:GetCloudFrontOriginAccessIdentityConfig**
- **cloudfront:GetDistribution**
- **cloudfront:TagResource**
- **cloudfront:UpdateDistribution**



#### 注意

在使用 **ccoctl aws create-all** 命令处理凭证请求时，这些额外权限支持使用 **--create-private-s3-bucket** 选项。

## 流程

1. 运行以下命令，为 OpenShift Container Platform 发行镜像设置变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像获取 CCO 容器镜像：

```
$ CCO_IMAGE=$(oc adm release info --image-for='cloud-credential-operator'
$RELEASE_IMAGE -a ~/.pull-secret)
```





### 注意

确保 **\$RELEASE\_IMAGE** 的架构与将使用 **ccoctl** 工具的环境架构相匹配。

- 运行以下命令，将 CCO 容器镜像中的 **ccoctl** 二进制文件提取到 OpenShift Container Platform 发行镜像中：

```
$ oc image extract $CCO_IMAGE --file="/usr/bin/ccoctl" -a ~/.pull-secret
```

- 运行以下命令更改权限以使 **ccoctl** 可执行：

```
$ chmod 775 ccoctl
```

### 验证

- 要验证 **ccoctl** 是否准备就绪，可以尝试显示帮助文件。运行命令时使用相对文件名，例如：

```
$ ./ccoctl.rhel9
```

### 输出示例

```
OpenShift credentials provisioning tool
```

```
Usage:
```

```
ccoctl [command]
```

```
Available Commands:
```

```
alibabacloud Manage credentials objects for alibaba cloud
```

```
aws          Manage credentials objects for AWS cloud
```

```
azure        Manage credentials objects for Azure
```

```
gcp          Manage credentials objects for Google cloud
```

```
help         Help about any command
```

```
ibmcloud     Manage credentials objects for IBM Cloud
```

```
nutanix      Manage credentials objects for Nutanix
```

```
Flags:
```

```
-h, --help  help for ccoctl
```

```
Use "ccoctl [command] --help" for more information about a command.
```

## 10.12.2.2. 使用 Cloud Credential Operator 实用程序创建 AWS 资源

创建 AWS 资源时有以下选项：

- 您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。这是创建资源的最快速方法。请参阅[使用单个命令创建 AWS 资源](#)。
- 如果您需要在修改 AWS 资源前查看 **ccoctl** 工具创建的 JSON 文件，或者 **ccoctl** 工具用于创建 AWS 资源的过程无法自动满足组织的要求，您可以单独创建 AWS 资源。请参阅[单独创建 AWS 资源](#)。

### 10.12.2.2.1. 使用单个命令创建 AWS 资源

如果 **ccoctl** 工具用于创建 AWS 资源的过程自动满足机构的要求，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。

否则，您可以单独创建 AWS 资源。如需更多信息，请参阅“单独创建 AWS 资源”。



### 注意

默认情况下，**ccoctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccoctl\_output\_dir>** 来引用这个目录。

## 先决条件

您必须：

- 提取并准备好 **ccoctl** 二进制文件。

## 流程

- 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/ {print $3}')
```

- 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \1
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \2
  --to=<path_to_directory_for_credentials_requests> \3
```

- included** 参数仅包含特定集群配置所需的清单。
- 指定 **install-config.yaml** 文件的位置。
- 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。



### 注意

此命令可能需要一些时间才能运行。

- 运行以下命令，使用 **ccoctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccoctl aws create-all \
  --name=<name> \1
  --region=<aws_region> \2
  --credentials-requests-dir=<path_to_credentials_requests_directory> \3
  --output-dir=<path_to_ccoctl_output_dir> \4
  --create-private-s3-bucket \5
```

- 1 指定用于标记创建用于跟踪的任何云资源的名称。
- 2 指定在其中创建云资源的 AWS 区域。
- 3 指定包含组件 **CredentialsRequest** 对象文件的目录。
- 4 可选：指定您希望 **ccctl** 实用程序在其中创建对象的目录。默认情况下，实用程序在运行命令的目录中创建对象。
- 5 可选：默认情况下，**ccctl** 实用程序将 OpenID Connect (OIDC) 配置文件存储在公共 S3 存储桶中，并使用 S3 URL 作为公共 OIDC 端点。要将 OIDC 配置存储在 IAM 身份提供程序通过公共 CloudFront 发行版 URL 访问的专用 S3 存储桶中，请使用 **--create-private-s3-bucket** 参数。



### 注意

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

### 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccctl_output_dir>/manifests
```

### 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

#### 10.12.2.2.2. 单独创建 AWS 资源

您可以使用 **ccctl** 工具单独创建 AWS 资源。这个选项对于在不同用户或部门之间创建这些资源的组织可能很有用。

否则，您可以使用 **ccctl aws create-all** 命令自动创建 AWS 资源。如需更多信息，请参阅“使用单个命令创建 AWS 资源”。



## 注意

默认情况下，**ccctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccctl\_output\_dir>** 来引用这个目录。

有些 **ccctl** 命令会发出 AWS API 调用来创建或修改 AWS 资源。您可以使用 **--dry-run** 标志来避免 API 调用。使用此标志可在本地文件系统中创建 JSON 文件。您可以使用 **--cli-input-json** 参数查看和修改 JSON 文件，然后使用 AWS CLI 工具应用它们。

## 先决条件

- 提取并准备 **ccctl** 二进制文件。

## 流程

- 运行以下命令，生成用于为集群设置 OpenID Connect 供应商的公共和私有 RSA 密钥文件：

```
$ ccctl aws create-key-pair
```

## 输出示例

```
2021/04/13 11:01:02 Generating RSA keypair
2021/04/13 11:01:03 Writing private key to /<path_to_ccctl_output_dir>/serviceaccount-signer.private
2021/04/13 11:01:03 Writing public key to /<path_to_ccctl_output_dir>/serviceaccount-signer.public
2021/04/13 11:01:03 Copying signing key for use by installer
```

其中 **serviceaccount-signer.private** 和 **serviceaccount-signer.public** 是生成的密钥文件。

此命令还会在 **/<path\_to\_ccctl\_output\_dir>/tls/bound-service-account-signing-key.key** 中创建集群在安装过程中所需的私钥。

- 运行以下命令，在 AWS 上创建 OpenID Connect 身份提供程序和 S3 存储桶：

```
$ ccctl aws create-identity-provider \
  --name=<name> \ ❶
  --region=<aws_region> \ ❷
  --public-key-file=<path_to_ccctl_output_dir>/serviceaccount-signer.public ❸
```

- ❶ **<name>** 是用于标记为跟踪而创建的云资源的名称。
- ❷ **<aws-region>** 是将在其中创建云资源的 AWS 区域。
- ❸ **<path\_to\_ccctl\_output\_dir>** 是 **ccctl aws create-key-pair** 命令生成的公钥文件的路径。

## 输出示例

```
2021/04/13 11:16:09 Bucket <name>-oidc created
2021/04/13 11:16:10 OpenID Connect discovery document in the S3 bucket <name>-oidc at .well-known/openid-configuration updated
2021/04/13 11:16:10 Reading public key
```

```
2021/04/13 11:16:10 JSON web key set (JWKS) in the S3 bucket <name>-oidc at keys.json
updated
2021/04/13 11:16:18 Identity Provider created with ARN: arn:aws:iam::
<aws_account_id>:oidc-provider/<name>-oidc.s3.<aws_region>.amazonaws.com
```

其中 **openid-configuration** 是发现文档和 **key.json** 是一个 JSON Web 密钥集文件。

此命令还会在 `/<path_to_ccoctl_output_dir>/manifests/cluster-authentication-02-config.yaml` 中创建 YAML 配置文件。此文件为集群生成的服务帐户令牌设置签发者 URL 字段，以便 AWS IAM 身份提供程序信任令牌。

### 3. 为集群中的每个组件创建 IAM 角色：

- a. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

- b. 从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included ❶ \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
  ❷ \
  --to=<path_to_directory_for_credentials_requests> ❸
```

❶ **--included** 参数仅包含特定集群配置所需的清单。

❷ 指定 **install-config.yaml** 文件的位置。

❸ 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

- c. 运行以下命令，使用 **ccoctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccoctl aws create-iam-roles \
  --name=<name> \
  --region=<aws_region> \
  --credentials-requests-dir=<path_to_credentials_requests_directory> \
  --identity-provider-arn=arn:aws:iam::<aws_account_id>:oidc-provider/<name>-oidc.s3.
<aws_region>.amazonaws.com
```



### 注意

对于使用其他 IAM API 端点的 AWS 环境（如 GovCloud），还必须使用 **--region** 参数指定您的区域。

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

对于每个 **CredentialsRequest** 对象，**ccocctl** 创建一个带有信任策略的 IAM 角色，该角色与指定的 OIDC 身份提供程序相关联，以及来自 OpenShift Container Platform 发行镜像的每个 **CredentialsRequest** 对象中定义的权限策略。

## 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 `<path_to_ccocctl_output_dir>/manifests` 目录中的文件：

```
$ ls <path_to_ccocctl_output_dir>/manifests
```

## 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

### 10.12.2.3. 整合 Cloud Credential Operator 实用程序清单

要为单个组件在集群外实现短期安全凭证，您必须将创建 Cloud Credential Operator 实用程序 (**ccocctl**) 的清单文件移到安装程序的正确目录中。

## 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您已配置了 Cloud Credential Operator 实用程序 (**ccocctl**)。
- 已使用 **ccocctl** 工具创建了集群所需的云供应商资源。

## 流程

- 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

- 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 `<installation_directory>` 是安装程序在其中创建文件的目录。

- 运行以下命令，将 `ccoctl` 工具生成的清单复制到安装程序创建的 `manifests` 目录中：

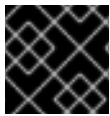
```
$ cp /<path_to_ccoctl_output_dir>/manifests/* ./manifests/
```

- 将包含私钥的 `tls` 目录复制到安装目录中：

```
$ cp -a /<path_to_ccoctl_output_dir>/tls .
```

## 10.13. 部署集群

您可以在兼容云平台上安装 OpenShift Container Platform。



### 重要

在初始安装过程中，您只能运行安装程序的 `create cluster` 命令一次。

### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。
- 已确认主机上的云供应商帐户具有部署集群的正确权限。权限不正确的帐户会导致安装过程失败，并显示包括缺失权限的错误消息。

### 流程

- 进入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>`，请指定自定义 `./install-config.yaml` 文件的位置。

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

- 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。

### 验证

当集群部署成功完成时：

- 终端会显示用于访问集群的说明，包括指向 Web 控制台和 `kubeadmin` 用户的凭证的链接。
- 凭证信息还会输出到 `<installation_directory>/./openshift_install.log`。

**重要**

不要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。

**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: "password"
INFO Time elapsed: 36m22s
```

**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。一个例外是，您必须手动批准待处理的 **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 小时时间进行轮转。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。

**10.14. 使用 CLI 登录集群**

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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

**先决条件**

- 已部署 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. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

**输出示例**



```
system:admin
```

```
/validating-an-installation.adoc
```

## 10.15. 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

- [访问Web控制台](#)

## 10.16. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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

- [关于远程健康监控](#)

## 10.17. 后续步骤

- [验证安装](#)
- [自定义集群](#)
- [远程健康报告](#)
- [删除云供应商凭证](#)

## 第 11 章 在 AWS CHINA 上安装集群

在 OpenShift Container Platform 版本 4.14 中，您可以将集群安装到以下 Amazon Web Services (AWS) 中国区域：

- **cn-north-1** (Beijing)
- **cn-northwest-1** (Ningxia)

### 11.1. 先决条件

- 您有一个 Internet Content Provider (ICP) 许可证。
- 您可以参阅有关 [OpenShift Container Platform 安装和更新](#) 流程的详细信息。
- 您可以阅读[选择集群安装方法并为用户准备它的文档](#)。
- 已将 [AWS 帐户配置](#)为托管集群。
- 如果使用防火墙，[将其配置为允许集群需要访问的站点](#)。



#### 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。

### 11.2. 安装要求

红帽没有发布 AWS China 区域的 Red Hat Enterprise Linux CoreOS (RHCOS) Amazon 机器镜像。

在安装集群前，您必须：

- 上传自定义 RHCOS AMI。
- 手动创建安装配置文件 (`install-config.yaml`)。
- 在安装配置文件中指定 AWS 区域和附带的自定义 AMI。

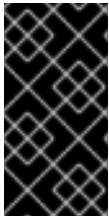
您不能使用 OpenShift Container Platform 安装程序创建安装配置文件。安装程序不会列出没有原生支持 RHCOS AMI 的 AWS 区域。

### 11.3. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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。



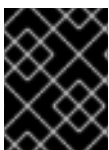
### 重要

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型的基础架构上执行受限网络安装。在此过程中，您可以下载所需的内容，并使用它为镜像 registry 填充安装软件包。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。

## 11.4. 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。在部署集群时，私有集群会将 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，且无法在互联网中看到。



### 重要

如果集群有任何公共子网，管理员创建的负载均衡器服务可能会公开访问。为确保集群安全性，请验证这些服务是否已明确标注为私有。

要部署私有集群，您必须：

- 使用满足您的要求的现有网络。集群资源可能会在网络上的其他集群间共享。
- 从有权访问的机器中部署：
  - 您置备的云的 API 服务。
  - 您调配的网络上的主机。
  - 用于获取安装介质的互联网。

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，此计算机可以是云网络上的堡垒主机。



### 注意

AWS China 不支持 VPC 和您的网络之间的 VPN 连接。有关 Beijing 和 Ningxia 地区的 Amazon VPC 服务的更多信息，请参阅 AWS China 的 [Amazon Virtual Private Cloud](#) 文档。

### 11.4.1. 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。

#### 11.4.1.1. 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。

## 11.5. 关于使用自定义 VPC

在 OpenShift Container Platform 4.14 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。

### 11.5.1. 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。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](#) 以及 [处理 VPC 和子网](#)。

安装程序无法：

- 分割网络范围供集群使用。
- 设置子网的路由表。
- 设置 VPC 选项，如 DHCP。

您必须在安装集群前完成这些任务。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 [您的 VPC 的路由表](#)。

您的 VPC 必须满足以下特征：

- VPC 不能使用 **kubernetes.io/cluster/.\*: owned, Name, 和 openshift.io/cluster** 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。您不能使用 **Name** 标签，因为它与 EC2 **Name** 字段重叠，且安装失败。
- 您需要在您的 VPC 中启用 **enableDnsSupport** 和 **enableDnsHostnames** 属性，以便集群可以使用附加到 VPC 中的 Route 53 区来解析集群内部的 DNS 记录。请参阅 AWS 文档中的 [您的 VPC 中的 DNS 支持](#) 部分。  
如果要使用您自己的 Route 53 托管私有区，您必须在安装集群前将现有托管区与 VPC 相关联。您可以使用 **install-config.yaml** 文件中的 **platform.aws.hostedZone** 和 **platform.aws.hostedZoneRole** 字段定义托管区。您可以通过与安装集群的帐户共享来使用来自另一个帐户的私有托管区。如果使用另一个帐户的私有托管区，则必须使用 **Passthrough** 或 **Manual** 凭证模式。

如果您在断开连接的环境中工作，则无法访问 EC2、ELB 和 S3 端点的公共 IP 地址。根据您要在安装过程中限制互联网流量的级别，有以下配置选项：

#### 11.5.1.1. 选项 1：创建 VPC 端点

创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com.cn**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

通过这个选项，网络流量在 VPC 和所需的 AWS 服务之间保持私有。

#### 11.5.1.2. 选项 2：创建一个没有 VPC 端点的代理

作为安装过程的一部分，您可以配置 HTTP 或 HTTPS 代理。使用此选项时，互联网流量会通过代理访问所需的 AWS 服务。

#### 11.5.1.3. 选项 3：创建带有 VPC 端点的代理

作为安装过程的一部分，您可以使用 VPC 端点配置 HTTP 或 HTTPS 代理。创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com.cn**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

在 `install-config.yaml` 文件中配置代理时，将这些端点添加到 `noProxy` 字段。通过这个选项，代理会阻止集群直接访问互联网。但是，您的 VPC 和所需的 AWS 服务之间网络流量保持私有。

## 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      |
|        |                                                                                                                                                                                                                                                                                                                                       |                                                                                                                     |             |

| 组件   | AWS 类型                                                                                                                                                                     | 描述                                                                                |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| 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。 |

### 11.5.2. 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。

### 11.5.3. 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：不同的个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。

### 11.5.4. 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)。



### 11.5.5. AWS 安全组

默认情况下，安装程序会创建安全组并将其附加到 control plane 和计算机器。不可修改与默认安全组关联的规则。

但是，您可以将与现有 VPC 关联的其他现有 AWS 安全组应用到 control plane 和计算机器。应用自定义安全组可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

作为安装过程的一部分，您可以在部署集群前通过修改 `install-config.yaml` 文件来应用自定义安全组。

如需更多信息，请参阅“将现有 AWS 安全组应用到集群”。

### 11.6. 为集群节点 SSH 访问生成密钥对

在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 `core` 用户的 `~/.ssh/authorized_keys` 列表中，这将启用免密码身份验证。

将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。

如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。`./openshift-install gather` 命令还需要在集群节点上设置 SSH 公钥。



#### 重要

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。



#### 注意

您必须使用本地密钥，而不是使用特定平台方法配置的密钥，如 AWS 密钥对。

#### 流程

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，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_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。



#### 注意

如果您计划在 `x86_64`、`ppc64le` 和 `s390x` 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 `ed25519` 算法的密钥。相反，创建一个使用 `rsa` 或 `ecdsa` 算法的密钥。

2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

3. 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。



#### 注意

在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- a. 如果 `ssh-agent` 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

#### 输出示例

```
Agent pid 31874
```



#### 注意

如果集群处于 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。

4. 将 SSH 私钥添加到 `ssh-agent`：

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

- 1** 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

#### 输出示例

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

#### 后续步骤

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

## 11.7. 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> ❶
```

- ❶ 包含 AWS 凭证的 AWS 配置集名称，如 **beijingadmin**。

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> ❶
```

- ❶ AWS 区域，如 **cn-north-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 **4.14.0**。

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-4.7.0-x86_64-aws.x86_64",
      "ImportTaskId": "import-snap-fh6i8uil",
      "SnapshotTaskDetail": {
        "Description": "rhcos-4.7.0-x86_64-aws.x86_64",
        "DiskImageSize": 819056640.0,
        "Format": "VMDK",
        "SnapshotId": "snap-06331325870076318",
        "Status": "completed",
        "UserBucket": {
          "S3Bucket": "external-images",
          "S3Key": "rhcos-4.7.0-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**, **aarch64**, **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](#)。

## 11.8. 获取安装程序

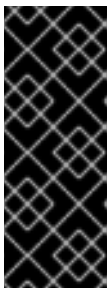
在安装 OpenShift Container Platform 前，将安装文件下载到您用于安装的主机上。

### 先决条件

- 您有一台运行 Linux 或 macOS 的计算机，至少有 1.2 GB 本地磁盘空间。

### 流程

1. 进入 Red Hat Hybrid Cloud Console 上的 [Cluster Type](#) 页。如果您有红帽帐户，请使用您的凭证登录。如果没有，请创建一个帐户。
2. 在页的 **Run it yourself** 部分中选择您的基础架构供应商。
3. 从 **OpenShift 安装程序** 下的下拉菜单中选择您的主机操作系统和架构，然后点 **下载安装程序**。
4. 将下载的文件保存在要存储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几个文件。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。删除集群需要这两个文件。
- 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，请为特定云供应商完成 OpenShift Container Platform 卸载流程。

5. 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
```

6. 从 [Red Hat OpenShift Cluster Manager](#) 下载安装 [pull secret](#)。此 pull secret 允许您与所含授权机构提供的服务进行身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

### 提示

另外，您还可以从[红帽客户门户网站](#)检索安装程序，您可以在其中指定要下载的安装程序版本。但是，您需要有一个有效的订阅才能访问此页。

## 11.9. 手动创建安装配置文件

安装集群需要您手动生成安装配置文件。

### 先决条件

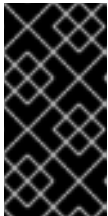
- 您上传了一个自定义 RHCOS AMI。
- 在本地机器上有一个 SSH 公钥用于安装程序。您可以使用密钥在集群节点上进行 SSH 身份验证，以进行调试和灾难恢复。

- 已获取 OpenShift Container Platform 安装程序和集群的 pull secret。

## 流程

1. 创建一个安装目录来存储所需的安装资产：

```
$ mkdir <installation_directory>
```



### 重要

您必须创建一个目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。

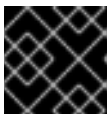
2. 对提供的 **install-config.yaml** 文件模板示例进行自定义，并将文件保存到 **<installation\_directory>** 中。



### 注意

此配置文件必须命名为 **install-config.yaml**。

3. 备份 **install-config.yaml** 文件，以便使用它来安装多个集群。



### 重要

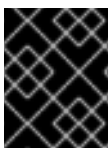
现在备份 **install-config.yaml** 文件，因为安装过程会在下一步中消耗该文件。

## 其他资源

- [AWS 的安装配置参数](#)

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

您可以自定义安装配置文件 (**install-config.yaml**)，以指定有关 OpenShift Container Platform 集群平台的更多详细信息，或修改所需参数的值。



### 重要

此示例 YAML 文件仅供参考。使用它作为资源，在您手动创建的安装配置文件中输入参数值。

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

```

- cn-north-1a
- cn-north-1b
rootVolume:
  iops: 4000
  size: 500
  type: io1 6
metadataService:
  authentication: Optional 7
  type: m6i.xlarge
replicas: 3
compute: 8
- hyperthreading: Enabled 9
  name: worker
  platform:
    aws:
      rootVolume:
        iops: 2000
        size: 500
        type: io1 10
      metadataService:
        authentication: Optional 11
        type: c5.4xlarge
      zones:
        - cn-north-1a
      replicas: 3
  metadata:
    name: test-cluster 12
  networking:
    clusterNetwork:
      - cidr: 10.128.0.0/14
        hostPrefix: 23
    machineNetwork:
      - cidr: 10.0.0.0/16
    networkType: OVNKubernetes 13
    serviceNetwork:
      - 172.30.0.0/16
  platform:
    aws:
      region: cn-north-1 14
      propagateUserTags: true 15
      userTags:
        adminContact: jdoe
        costCenter: 7536
      subnets: 16
      - subnet-1
      - subnet-2
      - subnet-3
      amiID: ami-96c6f8f7 17 18
      serviceEndpoints: 19
      - name: ec2
        url: https://vpce-id.ec2.cn-north-1.vpce.amazonaws.com.cn
      hostedZone: Z3URY6TWQ91KVV 20
  fips: false 21

```



sshKey: ssh-ed25519 AAAA... **22**

publish: Internal **23**

pullSecret: '{"auths": ...}' **24**

**1 12 14 17 24** 必需。

**2** 可选：添加此参数来强制 Cloud Credential Operator (CCO) 使用指定的模式。默认情况下，CCO 使用 **kube-system** 命名空间中的 root 凭证来动态尝试决定凭证的功能。有关 CCO 模式的详情，请参阅 [身份验证和授权指南](#) 中的 "About the Cloud Credential Operator" 部分。

**3 8 15** 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。仅使用一个 control plane 池。

**5 9** 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置为 **Disabled** 来禁用它。如果在某些集群机器中禁用并发多线程，则必须在所有集群机器中禁用它。



### 重要

如果您禁用并发多线程，请确保您的容量规划考虑机器性能显著降低的情况。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 10** 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**。

**7 11** 是否需要 [Amazon EC2 实例元数据服务 v2 \(IMDSv2\)](#)。为了要求 IMDSv2，请将参数值设置为 **Required**。要允许使用 IMDSv1 和 IMDSv2，请将参数值设置为 **Optional**。如果没有指定值，则允许 IMDSv1 和 IMDSv2。



### 注意

在集群安装过程中设置的 control plane 机器的 IMDS 配置只能使用 AWS CLI 更改。可以使用计算机器集来更改计算机器的 IMDS 配置。

**13** 要安装的集群网络插件。支持的值有 **OVNKubernetes** 和 **OpenShiftSDN**。默认值为 **OVNKubernetes**。

**16** 如果您提供自己的 VPC，为集群使用的每个可用区指定子网。

**18** 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。

**19** 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。

**20** 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。

**21** 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

要为集群启用 FIPS 模式，您必须从配置为以 FIPS 模式操作的 Red Hat Enterprise Linux (RHEL) 计算机运行安装程序。有关在 RHEL 中配置 FIPS 模式的更多信息，请参阅[在 FIPS 模式中安装该系统](#)。当以 FIPS 模式运行 Red Hat Enterprise Linux (RHEL) 或 Red Hat Enterprise Linux CoreOS (RHCOS) 时，OpenShift Container Platform 核心组件使用 RHEL 加密库，在 x86\_64、ppc64le 和 s390x 架构上提交到 NIST FIPS 140-2/140-3 Validation。

22

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。

23

如何发布集群的面向用户的端点。将 **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**。

## 11.9.2. 集群安装的最低资源要求

每台集群机器都必须满足以下最低要求：

表 11.1. 最低资源要求

| 机器                   | 操作系统                     | vCPU [1] | 虚拟内存  | Storage | 每秒输入/输出 (IOPS) [2] |
|----------------------|--------------------------|----------|-------|---------|--------------------|
| bootstrap            | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Control plane (控制平面) | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Compute              | RHCOS、RHEL 8.6 及更新版本 [3] | 2        | 8 GB  | 100 GB  | 300                |

1. 当未启用并发多线程 (SMT) 或超线程时，一个 vCPU 相当于一个物理内核。启用后，使用以下公式来计算对应的比例：（每个内核数的线程）× sockets = vCPU。
2. OpenShift Container Platform 和 Kubernetes 对磁盘性能非常敏感，建议使用更快的存储速度，特别是 control plane 节点上需要 10 ms p99 fsync 持续时间的 etcd。请注意，在许多云平台上，存储大小和 IOPS 可一起扩展，因此您可能需要过度分配存储卷来获取足够的性能。
3. 与所有用户置备的安装一样，如果您选择在集群中使用 RHEL 计算机，则负责所有操作系统生命周期管理和维护，包括执行系统更新、应用补丁和完成所有其他必要的任务。RHEL 7 计算机器的使用已弃用，并已在 OpenShift Container Platform 4.10 及更新的版本中删除。



## 注意

从 OpenShift Container Platform 版本 4.13 开始，RHCOS 基于 RHEL 版本 9.2，它更新了微架构要求。以下列表包含每个架构需要的最小指令集架构 (ISA)：

- x86-64 体系结构需要 x86-64-v2 ISA
- ARM64 架构需要 ARMv8.0-A ISA
- IBM Power 架构需要 Power 9 ISA
- s390x 架构需要 z14 ISA

如需更多信息，请参阅 [RHEL 架构](#)。

如果平台的实例类型满足集群机器的最低要求，则 OpenShift Container Platform 支持使用它。

## 其他资源

- [优化存储](#)

### 11.9.3. 为 AWS 测试的实例类型

以下 Amazon Web Services(AWS) 实例类型已经过 OpenShift Container Platform 测试。



## 注意

将以下图中包含的机器类型用于 AWS 实例。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 11.1. 基于 64 位 x86 架构的机器类型

- **c4.\***
- **c5.\***
- **c5a.\***
- **i3.\***
- **m4.\***
- **m5.\***
- **m5a.\***
- **m6i.\***
- **r4.\***
- **r5.\***
- **r5a.\***
- **r6i.\***

- t3.\*
- t3a.\*

#### 11.9.4. 在 64 位 ARM 基础架构上为 AWS 测试过的实例类型

OpenShift Container Platform 中已经测试了以下 Amazon Web Services (AWS) 64 位 ARM 实例类型。



##### 注意

使用 AWS ARM 实例的以下图中包含的机器类型。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 11.2. 基于 64 位 ARM 架构的机器类型

- c6g.\*
- m6g.\*
- r8g.\*

#### 11.9.5. 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，并确定它们中的任何站点是否需要绕过代理。默认情况下，所有集群出口流量都经过代理，包括对托管云供应商 API 的调用。如果需要，您将在 **Proxy** 对象的 **spec.noProxy** 字段中添加站点来绕过代理。



##### 注意

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

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

##### 流程

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

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

```

proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> ❶
  httpsProxy: https://<username>:<pswd>@<ip>:<port> ❷
  noProxy: ec2.<aws_region>.amazonaws.com,elasticloadbalancing.
    <aws_region>.amazonaws.com,s3.<aws_region>.amazonaws.com ❸
  additionalTrustBundle: | ❹
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> ❺

```

- ❶ 用于创建集群外 HTTP 连接的代理 URL。URL 方案必须是 **http**。
- ❷ 用于创建集群外 HTTPS 连接的代理 URL。
- ❸ 要从代理中排除的目标域名、IP 地址或其他网络 CIDR 的逗号分隔列表。在域前面加上 **.** 以仅匹配子域。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。如果您已将 Amazon **EC2**、**Elastic Load Balancing** 和 **S3** VPC 端点添加到 VPC 中，您必须将这些端点添加到 **noProxy** 字段。
- ❹ 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，其包含代理 HTTPS 连接所需的一个或多个额外 CA 证书。然后，Cluster Network Operator 会创建 **trusted-ca-bundle** 配置映射，将这些内容与 Red Hat Enterprise Linux CoreOS (RHCOS) 信任捆绑包合并，**Proxy** 对象的 **trustedCA** 字段中也会引用此配置映射。**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。
- ❺ 可选：决定 **Proxy** 对象的配置以引用 **trustedCA** 字段中 **user-ca-bundle** 配置映射的策略。允许的值是 **Proxyonly** 和 **Always**。仅在配置了 **http/https** 代理时，使用 **Proxyonly** 引用 **user-ca-bundle** 配置映射。使用 **Always** 始终引用 **user-ca-bundle** 配置映射。默认值为 **Proxyonly**。



### 注意

安装程序不支持代理的 **readinessEndpoints** 字段。



### 注意

如果安装程序超时，重启并使用安装程序的 **wait-for** 命令完成部署。例如：

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

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。

### 11.9.6. 将现有 AWS 安全组应用到集群

将现有 AWS 安全组应用到 control plane 和计算机器可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

#### 先决条件

- 您已在 AWS 中创建安全组。如需更多信息，请参阅使用 [安全组的 AWS 文档](#)。
- 安全组必须与您要将集群部署到的现有 VPC 关联。安全组不能与另一个 VPC 关联。
- 您有一个现有的 `install-config.yaml` 文件。

#### 流程

1. 在 `install-config.yaml` 文件中，编辑 `compute.platform.aws.additionalSecurityGroupIDs` 参数，为您的计算机器指定一个或多个自定义安全组。
2. 编辑 `controlPlane.platform.aws.additionalSecurityGroupIDs` 参数，为您的 control plane 机器指定一个或多个自定义安全组。
3. 保存文件并在部署集群时引用。

#### 指定自定义安全组的 `install-config.yaml` 文件示例

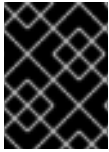
```
# ...
compute:
- hyperthreading: Enabled
  name: worker
  platform:
    aws:
      additionalSecurityGroupIDs:
        - sg-1 ❶
        - sg-2
  replicas: 3
controlPlane:
  hyperthreading: Enabled
  name: master
  platform:
    aws:
      additionalSecurityGroupIDs:
        - sg-3
        - sg-4
  replicas: 3
platform:
  aws:
    region: us-east-1
    subnets: ❷
    - subnet-1
    - subnet-2
    - subnet-3
```

❶ 在 Amazon EC2 控制台中显示时指定安全组的名称，包括 `sg` 前缀。

❷ 指定集群使用的每个可用区的子网。

## 11.10. 通过下载二进制文件安装 OPENSIFT CLI

您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

### 11.10.1. 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. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。
5. 解包存档：

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

6. 将 **oc** 二进制文件放到 **PATH** 中的目录中。  
要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 11.10.2. 在 Windows 上安装 OpenShift CLI

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

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
4. 使用 ZIP 程序解压存档。

- 将 **oc** 二进制文件移到 **PATH** 中的目录中。  
要查看您的 **PATH**，请打开命令提示并执行以下命令：

```
C:\> path
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 11.10.3. 在 macOS 上安装 OpenShift CLI

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

#### 流程

- 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
- 从 **版本** 下拉列表中选择适当的版本。
- 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。



#### 注意

对于 macOS arm64，请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

- 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**，请打开终端并执行以下命令：

```
$ echo $PATH
```

#### 验证

- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

### 11.11. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案

默认情况下，管理员 secret 存储在 **kube-system** 项目中。如果您在 **install-config.yaml** 文件中将 **credentialsMode** 参数配置为 **Manual**，则必须使用以下替代方案之一：

- 要手动管理长期云凭证，请按照[手动创建长期凭证](#)中的步骤操作。
- 要实现在集群外为各个组件管理的短期凭证，请按照[配置 AWS 集群以使用短期凭证](#)中的步骤操作。

### 11.11.1. 手动创建长期凭证

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. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

#### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/ {print $3}')
```

4. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 自定义资源 (CR) 列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included 1 \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml 2 \
  --to=<path_to_directory_for_credentials_requests> 3
```

**1** **--included** 参数仅包含特定集群配置所需的清单。

**2** 指定 **install-config.yaml** 文件的位置。

**3** 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

此命令为每个 **CredentialsRequest** 对象创建一个 YAML 文件。

#### CredentialsRequest 对象示例

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
```



```

...
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
      - effect: Allow
        action:
          - iam:GetUser
          - iam:GetUserPolicy
          - iam:ListAccessKeys
        resource: "*"
...

```

5. 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。

### 带有 secret 的 CredentialsRequest 对象示例

```

apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
...
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
      - effect: Allow
        action:
          - s3:CreateBucket
          - s3>DeleteBucket
        resource: "*"
    ...
  secretRef:
    name: <component_secret>
    namespace: <component_namespace>
...

```

### Secret 对象示例

```

apiVersion: v1
kind: Secret
metadata:
  name: <component_secret>
  namespace: <component_namespace>
data:
  aws_access_key_id: <base64_encoded_aws_access_key_id>
  aws_secret_access_key: <base64_encoded_aws_secret_access_key>

```

**重要**

在升级使用手动维护凭证的集群前，您必须确保 CCO 处于可升级状态。

**11.11.2. 将 AWS 集群配置为使用短期凭证**

要安装配置为使用 AWS 安全令牌服务 (STS) 的集群，您必须配置 CCO 实用程序并为集群创建所需的 AWS 资源。

**11.11.2.1. 配置 Cloud Credential Operator 工具**

当 Cloud Credential Operator (CCO) 以手动模式运行时，要从集群外部创建和管理云凭证，提取并准备 CCO 实用程序 (**ccoctl**) 二进制文件。

**注意**

**ccoctl** 工具是在 Linux 环境中运行的 Linux 二进制文件。

**先决条件**

- 您可以访问具有集群管理员权限的 OpenShift Container Platform 帐户。
- 已安装 OpenShift CLI (**oc**)。
- 您已为 **ccoctl** 工具创建了用于以下权限的 AWS 帐户：

**例 11.3. 所需的 AWS 权限****所需的 iam 权限**

- **iam:CreateOpenIDConnectProvider**
- **iam:CreateRole**
- **iam>DeleteOpenIDConnectProvider**
- **iam>DeleteRole**
- **iam>DeleteRolePolicy**
- **iam:GetOpenIDConnectProvider**
- **iam:GetRole**
- **iam:GetUser**
- **iam:ListOpenIDConnectProviders**
- **iam:ListRolePolicies**
- **iam:ListRoles**
- **iam:PutRolePolicy**
- **iam:TagOpenIDConnectProvider**

- **iam:TagRole**

#### 所需的 s3 权限

- **s3:CreateBucket**
- **s3:DeleteBucket**
- **s3:DeleteObject**
- **s3:GetBucketAcl**
- **s3:GetBucketTagging**
- **s3:GetObject**
- **s3:GetObjectAcl**
- **s3:GetObjectTagging**
- **s3:ListBucket**
- **s3:PutBucketAcl**
- **s3:PutBucketPolicy**
- **s3:PutBucketPublicAccessBlock**
- **s3:PutBucketTagging**
- **s3:PutObject**
- **s3:PutObjectAcl**
- **s3:PutObjectTagging**

#### 所需的 cloudfront 权限

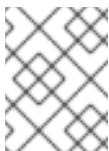
- **cloudfront:ListCloudFrontOriginAccessIdentities**
- **cloudfront:ListDistributions**
- **cloudfront:ListTagsForResource**

如果您计划通过公共 CloudFront 发行版 URL 将 OIDC 配置存储在 IAM 身份提供程序访问的私有 S3 存储桶中，则运行 **ccoctl** 工具的 AWS 帐户需要以下额外权限：

#### 例 11.4. 使用 CloudFront 私有 S3 存储桶的额外权限

- **cloudfront:CreateCloudFrontOriginAccessIdentity**
- **cloudfront:CreateDistribution**
- **cloudfront>DeleteCloudFrontOriginAccessIdentity**

- **cloudfront:DeleteDistribution**
- **cloudfront:GetCloudFrontOriginAccessIdentity**
- **cloudfront:GetCloudFrontOriginAccessIdentityConfig**
- **cloudfront:GetDistribution**
- **cloudfront:TagResource**
- **cloudfront:UpdateDistribution**



### 注意

在使用 **ccoctl aws create-all** 命令处理凭证请求时，这些额外权限支持使用 **--create-private-s3-bucket** 选项。

## 流程

1. 运行以下命令，为 OpenShift Container Platform 发行镜像设置变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像获取 CCO 容器镜像：

```
$ CCO_IMAGE=$(oc adm release info --image-for='cloud-credential-operator'
$RELEASE_IMAGE -a ~/.pull-secret)
```



### 注意

确保 **\$RELEASE\_IMAGE** 的架构与将使用 **ccoctl** 工具的环境架构相匹配。

3. 运行以下命令，将 CCO 容器镜像中的 **ccoctl** 二进制文件提取到 OpenShift Container Platform 发行镜像中：

```
$ oc image extract $CCO_IMAGE --file="/usr/bin/ccoctl" -a ~/.pull-secret
```

4. 运行以下命令更改权限以使 **ccoctl** 可执行：

```
$ chmod 775 ccoctl
```

## 验证

- 要验证 **ccoctl** 是否准备就绪，可以尝试显示帮助文件。运行命令时使用相对文件名，例如：

```
$ ./ccoctl.rhel9
```

### 输出示例

```
OpenShift credentials provisioning tool
```

Usage:  
ccoctl [command]

Available Commands:

alibabacloud Manage credentials objects for alibaba cloud  
aws Manage credentials objects for AWS cloud  
azure Manage credentials objects for Azure  
gcp Manage credentials objects for Google cloud  
help Help about any command  
ibmcloud Manage credentials objects for IBM Cloud  
nutanix Manage credentials objects for Nutanix

Flags:

-h, --help help for ccoctl

Use "ccoctl [command] --help" for more information about a command.

### 11.11.2.2. 使用 Cloud Credential Operator 实用程序创建 AWS 资源

创建 AWS 资源时有以下选项：

- 您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。这是创建资源的最快速方法。请参阅 [使用单个命令创建 AWS 资源](#)。
- 如果您需要在修改 AWS 资源前查看 **ccoctl** 工具创建的 JSON 文件，或者 **ccoctl** 工具用于创建 AWS 资源的过程无法自动满足组织的要求，您可以单独创建 AWS 资源。请参阅 [单独创建 AWS 资源](#)。

#### 11.11.2.2.1. 使用单个命令创建 AWS 资源

如果 **ccoctl** 工具用于创建 AWS 资源的过程自动满足机构的要求，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。

否则，您可以单独创建 AWS 资源。如需更多信息，请参阅“单独创建 AWS 资源”。



#### 注意

默认情况下，**ccoctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccoctl\_output\_dir>** 来引用这个目录。

#### 先决条件

您必须：

- 提取并准备好 **ccoctl** 二进制文件。

#### 流程

1. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \ ❶
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \ ❷
  --to=<path_to_directory_for_credentials_requests> \ ❸
```

- ❶ **--included** 参数仅包含特定集群配置所需的清单。
- ❷ 指定 **install-config.yaml** 文件的位置。
- ❸ 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。



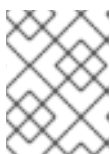
### 注意

此命令可能需要一些时间才能运行。

3. 运行以下命令，使用 **ccctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccctl aws create-all \
  --name=<name> \ ❶
  --region=<aws_region> \ ❷
  --credentials-requests-dir=<path_to_credentials_requests_directory> \ ❸
  --output-dir=<path_to_ccctl_output_dir> \ ❹
  --create-private-s3-bucket \ ❺
```

- ❶ 指定用于标记创建用于跟踪的任何云资源的名称。
- ❷ 指定在其中创建云资源的 AWS 区域。
- ❸ 指定包含组件 **CredentialsRequest** 对象文件的目录。
- ❹ 可选：指定您希望 **ccctl** 实用程序在其中创建对象的目录。默认情况下，实用程序在运行命令的目录中创建对象。
- ❺ 可选：默认情况下，**ccctl** 实用程序将 OpenID Connect (OIDC) 配置文件存储在公共 S3 存储桶中，并使用 S3 URL 作为公共 OIDC 端点。要将 OIDC 配置存储在 IAM 身份提供程序通过公共 CloudFront 发行版 URL 访问的专用 S3 存储桶中，请使用 **--create-private-s3-bucket** 参数。



### 注意

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 `<path_to_ccoctl_output_dir>/manifests` 目录中的文件：

```
$ ls <path_to_ccoctl_output_dir>/manifests
```

### 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

#### 11.11.2.2.2. 单独创建 AWS 资源

您可以使用 **ccoctl** 工具单独创建 AWS 资源。这个选项对于在不同用户或部门之间创建这些资源的组织可能很有用。

否则，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。如需更多信息，请参阅“使用单个命令创建 AWS 资源”。



### 注意

默认情况下，**ccoctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 `<path_to_ccoctl_output_dir>` 来引用这个目录。

有些 **ccoctl** 命令会发出 AWS API 调用来创建或修改 AWS 资源。您可以使用 **--dry-run** 标志来避免 API 调用。使用此标志可在本地文件系统中创建 JSON 文件。您可以使用 **--cli-input-json** 参数查看和修改 JSON 文件，然后使用 AWS CLI 工具应用它们。

### 先决条件

- 提取并准备 **ccoctl** 二进制文件。

### 流程

1. 运行以下命令，生成用于为集群设置 OpenID Connect 供应商的公共和私有 RSA 密钥文件：

```
$ ccoctl aws create-key-pair
```

### 输出示例

```
2021/04/13 11:01:02 Generating RSA keypair
2021/04/13 11:01:03 Writing private key to /<path_to_ccoctl_output_dir>/serviceaccount-signer.private
```

```
2021/04/13 11:01:03 Writing public key to /<path_to_ccoctl_output_dir>/serviceaccount-
signer.public
2021/04/13 11:01:03 Copying signing key for use by installer
```

其中 **serviceaccount-signer.private** 和 **serviceaccount-signer.public** 是生成的密钥文件。

此命令还会在 **/<path\_to\_ccoctl\_output\_dir>/tls/bound-service-account-signing-key.key** 中创建集群在安装过程中所需的私钥。

2. 运行以下命令，在 AWS 上创建 OpenID Connect 身份提供程序和 S3 存储桶：

```
$ ccoctl aws create-identity-provider \
  --name=<name> ❶
  --region=<aws_region> ❷
  --public-key-file=<path_to_ccoctl_output_dir>/serviceaccount-signer.public ❸
```

❶ **<name>** 是用于标记为跟踪而创建的云资源的名称。

❷ **<aws-region>** 是将在其中创建云资源的 AWS 区域。

❸ **<path\_to\_ccoctl\_output\_dir>** 是 **ccoctl aws create-key-pair** 命令生成的公钥文件的路径。

### 输出示例

```
2021/04/13 11:16:09 Bucket <name>-oidc created
2021/04/13 11:16:10 OpenID Connect discovery document in the S3 bucket <name>-oidc at
.well-known/openid-configuration updated
2021/04/13 11:16:10 Reading public key
2021/04/13 11:16:10 JSON web key set (JWKS) in the S3 bucket <name>-oidc at keys.json
updated
2021/04/13 11:16:18 Identity Provider created with ARN: arn:aws:iam::
<aws_account_id>:oidc-provider/<name>-oidc.s3.<aws_region>.amazonaws.com
```

其中 **openid-configuration** 是发现文档和 **key.json** 是一个 JSON Web 密钥集文件。

此命令还会在 **/<path\_to\_ccoctl\_output\_dir>/manifests/cluster-authentication-02-config.yaml** 中创建 YAML 配置文件。此文件为集群生成的服务帐户令牌设置签发者 URL 字段，以便 AWS IAM 身份提供程序信任令牌。

3. 为集群中的每个组件创建 IAM 角色：

- a. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/ {print $3}')
```

- b. 从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included ❶
```



```
--install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
```

2

```
--to=<path_to_directory_for_credentials_requests> 3
```

1

--included 参数仅包含特定集群配置所需的清单。

2

指定 **install-config.yaml** 文件的位置。

3

指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

c. 运行以下命令，使用 **ccctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccctl aws create-iam-roles \
  --name=<name> \
  --region=<aws_region> \
  --credentials-requests-dir=<path_to_credentials_requests_directory> \
  --identity-provider-arn=arn:aws:iam::<aws_account_id>:oidc-provider/<name>-oidc.s3.
  <aws_region>.amazonaws.com
```



### 注意

对于使用其他 IAM API 端点的 AWS 环境（如 GovCloud），还必须使用 **--region** 参数指定您的区域。

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

对于每个 **CredentialsRequest** 对象，**ccctl** 创建一个带有信任策略的 IAM 角色，该角色与指定的 OIDC 身份提供程序相关联，以及来自 OpenShift Container Platform 发行镜像的每个 **CredentialsRequest** 对象中定义的权限策略。

## 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccctl_output_dir>/manifests
```

## 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

### 11.11.2.3. 整合 Cloud Credential Operator 实用程序清单

要为单个组件在集群外实现短期安全凭证，您必须将创建 Cloud Credential Operator 实用程序 (**ccctl**) 的清单文件移到安装程序的正确目录中。

#### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您已配置了 Cloud Credential Operator 实用程序 (**ccctl**)。
- 已使用 **ccctl** 工具创建了集群所需的云供应商资源。

#### 流程

1. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

#### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，将 **ccctl** 工具生成的清单复制到安装程序创建的 **manifests** 目录中：

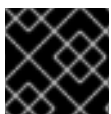
```
$ cp /<path_to_ccctl_output_dir>/manifests/* ./manifests/
```

4. 将包含私钥的 **tls** 目录复制到安装目录中：

```
$ cp -a /<path_to_ccctl_output_dir>/tls .
```

## 11.12. 部署集群

您可以在兼容云平台上安装 OpenShift Container Platform。



#### 重要

在初始安装过程中，您只能运行安装程序的 **create cluster** 命令一次。

#### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。

- 已确认主机上的云供应商帐户具有部署集群的正确权限。权限不正确的帐户会导致安装过程失败，并显示包括缺失权限的错误消息。

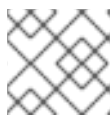
## 流程

1. 进入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>，请指定自定义 `./install-config.yaml` 文件的位置。
- ❷ 要查看不同的安装详情，请指定 **warn**、**debug** 或 **error**，而不是 **info**。

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。

## 验证

当集群部署成功完成时：

- 终端会显示用于访问集群的说明，包括指向 Web 控制台和 **kubeadmin** 用户的凭证的链接。
- 凭证信息还会输出到 `<installation_directory>/openshift_install.log`。



### 重要

不要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。

## 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: "password"
INFO Time elapsed: 36m22s
```



## 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。一个例外是，您必须手动批准待处理的 **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。

## 11.13. 使用 CLI 登录集群

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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

### 先决条件

- 已部署 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. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

### 输出示例

```
system:admin
```

/validating-an-installation.adoc

## 11.14. 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。

## 11.15. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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

- 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](#)。
- 有关 Telemetry 服务的更多信息，请参阅[关于远程健康监控](#)。

## 11.16. 后续步骤

- [验证安装](#)。
- [自定义集群](#)。
- [远程健康报告](#)

- [删除云供应商凭证](#)

## 第 12 章 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 4.14 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。

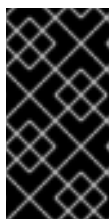


### 重要

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。

### 12.1. 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。
- 您可以阅读[选择集群安装方法并为用户准备它的文档](#)。
- 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。集群将继续使用您当前的 AWS 凭证为集群的整个生命周期创建 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

如果要配置代理，请务必查看此[站点列表](#)。

- 如果环境中无法访问云身份和访问管理(IAM)API，或者不想将管理员级别的凭证 secret 存储在 `kube-system` 命名空间中，您可以 [手动创建和维护 IAM 凭证](#)。

### 12.2. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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型的基础架构上执行受限网络安装。在此过程中，您可以下载所需的内容，并使用它为镜像 registry 填充安装软件包。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。在更新集群前，您要更新镜像 registry 的内容。

12.3. 具有用户置备基础架构的集群的要求

对于包含用户置备的基础架构的集群，您必须部署所有所需的机器。

本节论述了在用户置备的基础架构上部署 OpenShift Container Platform 的要求。

12.3.1. 集群安装所需的机器

最小的 OpenShift Container Platform 集群需要以下主机：

表 12.1. 最低所需的主机

| 主机                      | 描述                                                                                                |
|-------------------------|---------------------------------------------------------------------------------------------------|
| 一个临时 bootstrap 机器       | 集群需要 bootstrap 机器在三台 control plane 机器上部署 OpenShift Container Platform 集群。您可在安装集群后删除 bootstrap 机器。 |
| 三台 control plane 机器     | control plane 机器运行组成 control plane 的 Kubernetes 和 OpenShift Container Platform 服务。                |
| 至少两台计算机器，也称为 worker 机器。 | OpenShift Container Platform 用户请求的工作负载在计算机器上运行。                                                   |



**重要**

要保持集群的高可用性，请将独立的物理主机用于这些集群机器。

bootstrap 和 control plane 机器必须使用 Red Hat Enterprise Linux CoreOS(RHCOS)作为操作系统。但是，计算机器可以在 Red Hat Enterprise Linux CoreOS(RHCOS)、Red Hat Enterprise Linux(RHEL) 8.6 和更高的版本。

请注意，RHCOS 基于 Red Hat Enterprise Linux(RHEL) 9.2，并继承其所有硬件认证和要求。查看 [红帽企业 Linux 技术功能和限制](#)。

12.3.2. 集群安装的最低资源要求

每台集群机器都必须满足以下最低要求：

表 12.2. 最低资源要求



| 机器                   | 操作系统                     | vCPU [1] | 虚拟内存  | Storage | 每秒输入/输出 (IOPS) [2] |
|----------------------|--------------------------|----------|-------|---------|--------------------|
| bootstrap            | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Control plane (控制平面) | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Compute              | RHCOS、RHEL 8.6 及更新版本 [3] | 2        | 8 GB  | 100 GB  | 300                |

1. 当未启用并发多线程 (SMT) 或超线程时，一个 vCPU 相当于一个物理内核。启用后，使用以下公式来计算对应的比例：（每个内核数的线程）× sockets = vCPU。
2. OpenShift Container Platform 和 Kubernetes 对磁盘性能非常敏感，建议使用更快的存储速度，特别是 control plane 节点上需要 10 ms p99 fsync 持续时间的 etcd。请注意，在许多云平台上，存储大小和 IOPS 可一起扩展，因此您可能需要过度分配存储卷来获取足够的性能。
3. 与所有用户置备的安装一样，如果您选择在集群中使用 RHEL 计算机器，则负责所有操作系统生命周期管理和维护，包括执行系统更新、应用补丁和完成所有其他必要的任务。RHEL 7 计算机器的使用已弃用，并已在 OpenShift Container Platform 4.10 及更新的版本中删除。



### 注意

从 OpenShift Container Platform 版本 4.13 开始，RHCOS 基于 RHEL 版本 9.2，它更新了微架构要求。以下列表包含每个架构需要的最小指令集架构 (ISA)：

- x86-64 体系结构需要 x86-64-v2 ISA
- ARM64 架构需要 ARMv8.0-A ISA
- IBM Power 架构需要 Power 9 ISA
- s390x 架构需要 z14 ISA

如需更多信息，请参阅 [RHEL 架构](#)。

如果平台的实例类型满足集群机器的最低要求，则 OpenShift Container Platform 支持使用它。

### 其他资源

- [优化存储](#)

### 12.3.3. 为 AWS 测试的实例类型

以下 Amazon Web Services(AWS) 实例类型已经过 OpenShift Container Platform 测试。



### 注意

将以下图中包含的机器类型用于 AWS 实例。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 12.1. 基于 64 位 x86 架构的机器类型

- c4.\*
- c5.\*
- c5a.\*
- i3.\*
- m4.\*
- m5.\*
- m5a.\*
- m6i.\*
- r4.\*
- r5.\*
- r5a.\*
- r6i.\*
- t3.\*
- t3a.\*

#### 12.3.4. 在 64 位 ARM 基础架构上为 AWS 测试过的实例类型

OpenShift Container Platform 中已经测试了以下 Amazon Web Services (AWS) 64 位 ARM 实例类型。



### 注意

使用 AWS ARM 实例的以下图中包含的机器类型。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 12.2. 基于 64 位 ARM 架构的机器类型

- c6g.\*
- m6g.\*
- r8g.\*

### 12.3.5. 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。

## 12.4. 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。

### 12.4.1. 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中工作，则无法访问 EC2、ELB 和 S3 端点的公共 IP 地址。根据您要在安装过程中限制互联网流量的级别，有以下配置选项：

#### 12.4.1.1. 选项 1：创建 VPC 端点

创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- `ec2.<aws_region>.amazonaws.com`

- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

通过这个选项，网络流量在 VPC 和所需的 AWS 服务之间保持私有。

#### 12.4.1.2. 选项 2：创建一个没有 VPC 端点的代理

作为安装过程的一部分，您可以配置 HTTP 或 HTTPS 代理。使用此选项时，互联网流量会通过代理访问所需的 AWS 服务。

#### 12.4.1.3. 选项 3：创建带有 VPC 端点的代理

作为安装过程的一部分，您可以使用 VPC 端点配置 HTTP 或 HTTPS 代理。创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

在 **install-config.yaml** 文件中配置代理时，将这些端点添加到 **noProxy** 字段。通过这个选项，代理会阻止集群直接访问互联网。但是，您的 VPC 和所需的 AWS 服务之间网络流量保持私有。

### 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：                                                                                                   |             |
|        |                                                                                                                                                                                                                                                                                                                                       | 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（如果安装程序置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。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。              |
| 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。              |
| 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。  |

| 组件    | AWS 类型                                          | 描述           |
|-------|-------------------------------------------------|--------------|
| 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> | <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::SecurityGroup</b> | <b>icmp</b> | <b>0</b>     |
|                               |                                | <b>tcp</b>  | <b>22</b>    |
| <b>BootstrapSecurityGroup</b> | <b>AWS::EC2::SecurityGroup</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>MasterIngressEtcd</b>        | etcd      | <b>tcp</b> | <b>2379- 2380</b> |
| <b>MasterIngressVxlan</b>       | Vxlan 数据包 | <b>udp</b> | <b>4789</b>       |
| <b>MasterIngressWorkerVxlan</b> | Vxlan 数据包 | <b>udp</b> | <b>4789</b>       |

| 入口组                                       | 描述                                 | IP 协议      | 端口范围                 |
|-------------------------------------------|------------------------------------|------------|----------------------|
| <b>MasterIngressInternal</b>              | 内部集群通信和 Kubernetes 代理指标            | <b>tcp</b> | <b>9000 - 9999</b>   |
| <b>MasterIngressWorkerInternal</b>        | 内部集群通信                             | <b>tcp</b> | <b>9000 - 9999</b>   |
| <b>MasterIngressKube</b>                  | kubernetes kubelet、调度程序 and 控制器管理器 | <b>tcp</b> | <b>10250 - 10259</b> |
| <b>MasterIngressWorkerKube</b>            | kubernetes kubelet、调度程序 and 控制器管理器 | <b>tcp</b> | <b>10250 - 10259</b> |
| <b>MasterIngressIngressServices</b>       | Kubernetes 入口服务                    | <b>tcp</b> | <b>30000 - 32767</b> |
| <b>MasterIngressWorkerIngressServices</b> | Kubernetes 入口服务                    | <b>tcp</b> | <b>30000 - 32767</b> |
| <b>MasterIngressGeneve</b>                | Geneve 包                           | <b>udp</b> | <b>6081</b>          |
| <b>MasterIngressWorkerGeneve</b>          | Geneve 包                           | <b>udp</b> | <b>6081</b>          |
| <b>MasterIngressIpsecIke</b>              | IPsec IKE 数据包                      | <b>udp</b> | <b>500</b>           |
| <b>MasterIngressWorkerIpsecIke</b>        | IPsec IKE 数据包                      | <b>udp</b> | <b>500</b>           |
| <b>MasterIngressIpsecNat</b>              | IPsec NAT-T 数据包                    | <b>udp</b> | <b>4500</b>          |
| <b>MasterIngressWorkerIpsecNat</b>        | IPsec NAT-T 数据包                    | <b>udp</b> | <b>4500</b>          |
| <b>MasterIngressIpsecEsp</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>MasterIngressWorkerIpsecEsp</b>           | IPsec ESP 数据包   | <b>50</b>  | <b>All</b>           |
| <b>MasterIngressInternalUDP</b>              | 内部集群通信          | <b>udp</b> | <b>9000 - 9999</b>   |
| <b>MasterIngressWorkerInternalUDP</b>        | 内部集群通信          | <b>udp</b> | <b>9000 - 9999</b>   |
| <b>MasterIngressIngressServicesUDP</b>       | Kubernetes 入口服务 | <b>udp</b> | <b>30000 - 32767</b> |
| <b>MasterIngressWorkerIngressServicesUDP</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>WorkerIngressVxlan</b>          | Vxlan 数据包                      | <b>udp</b> | <b>4789</b>        |
| <b>WorkerIngressWorkerVxlan</b>    | Vxlan 数据包                      | <b>udp</b> | <b>4789</b>        |
| <b>WorkerIngressInternal</b>       | 内部集群通信                         | <b>tcp</b> | <b>9000 - 9999</b> |
| <b>WorkerIngressWorkerInternal</b> | 内部集群通信                         | <b>tcp</b> | <b>9000 - 9999</b> |
| <b>WorkerIngressKube</b>           | Kubernetes kubelet、调度程序和控制器管理器 | <b>tcp</b> | <b>10250</b>       |
| <b>WorkerIngressWorkerKube</b>     | Kubernetes kubelet、调度程序和控制器管理器 | <b>tcp</b> | <b>10250</b>       |

| 入口组                                       | 描述              | IP 协议      | 端口范围                 |
|-------------------------------------------|-----------------|------------|----------------------|
| <b>WorkerIngressIngressServices</b>       | Kubernetes 入口服务 | <b>tcp</b> | <b>30000 - 32767</b> |
| <b>WorkerIngressWorkerIngressServices</b> | Kubernetes 入口服务 | <b>tcp</b> | <b>30000 - 32767</b> |
| <b>WorkerIngressGeneve</b>                | Geneve 包        | <b>udp</b> | <b>6081</b>          |
| <b>WorkerIngressMasterGeneve</b>          | Geneve 包        | <b>udp</b> | <b>6081</b>          |
| <b>WorkerIngressIpsecIke</b>              | IPsec IKE 数据包   | <b>udp</b> | <b>500</b>           |
| <b>WorkerIngressMasterIpsecIke</b>        | IPsec IKE 数据包   | <b>udp</b> | <b>500</b>           |
| <b>WorkerIngressIpsecNat</b>              | IPsec NAT-T 数据包 | <b>udp</b> | <b>4500</b>          |
| <b>WorkerIngressMasterIpsecNat</b>        | IPsec NAT-T 数据包 | <b>udp</b> | <b>4500</b>          |
| <b>WorkerIngressIpsecEsp</b>              | IPsec ESP 数据包   | <b>50</b>  | <b>All</b>           |
| <b>WorkerIngressMasterIpsecEsp</b>        | IPsec ESP 数据包   | <b>50</b>  | <b>All</b>           |
| <b>WorkerIngressInternalUDP</b>           | 内部集群通信          | <b>udp</b> | <b>9000 - 9999</b>   |
| <b>WorkerIngressMasterInternalUDP</b>     | 内部集群通信          | <b>udp</b> | <b>9000 - 9999</b>   |

| 入口组                                   | 描述              | IP 协议 | 端口范围          |
|---------------------------------------|-----------------|-------|---------------|
| WorkerIngressIngressServicesUDP       | Kubernetes 入口服务 | udp   | 30000 - 32767 |
| WorkerIngressMasterIngressServicesUDP | 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。

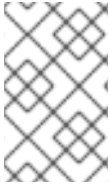
| 角色        | 影响    | 操作                     | 资源 |
|-----------|-------|------------------------|----|
| Master    | Allow | ec2:*                  | *  |
|           | Allow | elasticloadbalancing:* | *  |
|           | Allow | iam:PassRole           | *  |
|           | Allow | s3:GetObject           | *  |
| Worker    | Allow | ec2:Describe*          | *  |
| bootstrap | Allow | ec2:Describe*          | *  |
|           | Allow | ec2:AttachVolume       | *  |
|           | Allow | ec2:DetachVolume       | *  |

12.4.2. 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 control plane 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）。这些机器不受计算机器集的管控。

12.4.3. IAM 用户所需的 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 12.3. 安装所需的 EC2 权限

- **ec2:AttachNetworkInterface**
- **ec2:AuthorizeSecurityGroupEgress**
- **ec2:AuthorizeSecurityGroupIngress**
- **ec2:CopyImage**
- **ec2>CreateNetworkInterface**
- **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:DescribeInstanceTypes**
- **ec2:DescribeInternetGateways**
- **ec2:DescribeKeyPairs**

- **ec2:DescribeNatGateways**
- **ec2:DescribeNetworkAcls**
- **ec2:DescribeNetworkInterfaces**
- **ec2:DescribePrefixLists**
- **ec2:DescribeRegions**
- **ec2:DescribeRouteTables**
- **ec2:DescribeSecurityGroupRules**
- **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:RevokeSecurityGroupEgress**
- **ec2:RevokeSecurityGroupIngress**
- **ec2:RunInstances**
- **ec2:TerminateInstances**

#### 例 12.4. 安装过程中创建网络资源所需的权限

- **ec2:AllocateAddress**
- **ec2:AssociateAddress**
- **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 Virtual Private Cloud (VPC)，您的帐户不需要这些权限来创建网络资源。

#### 例 12.5. 安装所需的 Elastic Load Balancing 权限(ELB)

- **elasticloadbalancing:AddTags**
- **elasticloadbalancing:ApplySecurityGroupsToLoadBalancer**
- **elasticloadbalancing:AttachLoadBalancerToSubnets**
- **elasticloadbalancing:ConfigureHealthCheck**
- **elasticloadbalancing>CreateListener**
- **elasticloadbalancing>CreateLoadBalancer**
- **elasticloadbalancing>CreateLoadBalancerListeners**
- **elasticloadbalancing>CreateTargetGroup**
- **elasticloadbalancing>DeleteLoadBalancer**
- **elasticloadbalancing:DeregisterInstancesFromLoadBalancer**
- **elasticloadbalancing:DeregisterTargets**
- **elasticloadbalancing:DescribeInstanceHealth**
- **elasticloadbalancing:DescribeListeners**

- `elasticloadbalancing:DescribeLoadBalancerAttributes`
- `elasticloadbalancing:DescribeLoadBalancers`
- `elasticloadbalancing:DescribeTags`
- `elasticloadbalancing:DescribeTargetGroupAttributes`
- `elasticloadbalancing:DescribeTargetHealth`
- `elasticloadbalancing:ModifyLoadBalancerAttributes`
- `elasticloadbalancing:ModifyTargetGroup`
- `elasticloadbalancing:ModifyTargetGroupAttributes`
- `elasticloadbalancing:RegisterInstancesWithLoadBalancer`
- `elasticloadbalancing:RegisterTargets`
- `elasticloadbalancing:SetLoadBalancerPoliciesOfListener`



### 重要

OpenShift Container Platform 使用 ELB 和 ELBv2 API 服务来置备负载均衡器。权限列表显示这两个服务所需的权限。AWS web 控制台中存在一个已知问题，其中这两个服务都使用相同的 **elasticloadbalancing** 操作前缀，但无法识别相同的操作。您可以忽略有关服务没有识别某些 **elasticloadbalancing** 操作的警告。

### 例 12.6. 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 帐户中创建负载均衡器，IAM 用户还需要 **iam:CreateServiceLinkedRole** 权限。

#### 例 12.7. 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 12.8. 安装所需的 Amazon Simple Storage Service (S3) 权限

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



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

#### 例 12.9. 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 12.10. 删除基本集群资源所需的权限

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

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

#### 例 12.11. 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:ReleaseAddress**
- **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。您的帐户只需要有 **tag:UntagResources** 权限就能删除网络资源。

#### 例 12.12. 使用自定义密钥管理服务 (KMS) 密钥安装集群的可选权限

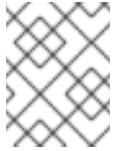
- **kms:CreateGrant**
- **kms:Decrypt**
- **kms:DescribeKey**
- **kms:Encrypt**
- **kms:GenerateDataKey**
- **kms:GenerateDataKeyWithoutPlainText**
- **kms:ListGrants**
- **kms:RevokeGrant**

#### 例 12.13. 使用共享实例角色删除集群所需的权限

- **iam:UntagRole**

#### 例 12.14. 创建清单所需的额外 IAM 和 S3 权限

- **iam:GetUserPolicy**
- **iam:ListAccessKeys**
- **iam:PutUserPolicy**
- **iam:TagUser**
- **s3:AbortMultipartUpload**
- **s3:GetBucketPublicAccessBlock**
- **s3:ListBucket**
- **s3:ListBucketMultipartUploads**
- **s3:PutBucketPublicAccessBlock**
- **s3:PutLifecycleConfiguration**

**注意**

如果您要使用 mint 模式管理云供应商凭证，IAM 用户还需要 `The iam:CreateAccessKey` 和 `iam:CreateUser` 权限。

**例 12.15. 实例的可选权限和安装配额检查**

- `ec2:DescribeInstanceTypeOfferings`
- `servicequotas:ListAWSDefaultServiceQuotas`

**例 12.16. 在共享 VPC 上安装集群时集群所有者帐户的可选权限**

- `sts:AssumeRole`

## 12.5. 获取 AWS MARKETPLACE 镜像

如果要使用 AWS Marketplace 镜像部署 OpenShift Container Platform 集群，您必须首先通过 AWS 订阅。订阅提供的提供的 AMI ID 可让您使用安装程序用来部署 worker 节点的 AMI ID。

**注意**

您应该只修改计算机器的 RHCOS 镜像以使用 AWS Marketplace 镜像。control plane 机器和基础架构节点不需要 OpenShift Container Platform 订阅，且默认使用公共 RHCOS 默认镜像，这不会对您的 AWS 账单造成订阅成本。因此，您不应该修改集群默认引导镜像或 control plane 引导镜像。将 AWS Marketplace 镜像应用到它们时，会生成无法恢复的额外许可成本。

**先决条件**

- 有 AWS 账户购买的产品。此帐户不必与用于安装集群的帐户相同。

**流程**

1. 从 [AWS Marketplace](#) 完成 OpenShift Container Platform 订阅。
2. 记录特定区域的 AMI ID。如果使用 CloudFormation 模板来部署 worker 节点，您必须更新 `worker0.type.properties.ImageID` 参数。

## 12.6. 获取安装程序

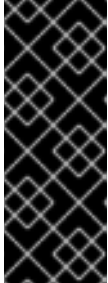
在安装 OpenShift Container Platform 前，将安装文件下载到您用于安装的主机上。

**先决条件**

- 您有一台运行 Linux 或 macOS 的计算机，至少有 1.2 GB 本地磁盘空间。

**流程**

1. 进入 Red Hat Hybrid Cloud Console 上的 [Cluster Type](#) 页。如果您有红帽帐户，请使用您的凭证登录。如果没有，请创建一个帐户。
2. 在页的 **Run it yourself** 部分中选择您的基础架构供应商。
3. 从 **OpenShift 安装程序** 下的下拉菜单中选择您的主机操作系统和架构，然后点击下载安装程序。
4. 将下载的文件保存在要存储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几个文件。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。删除集群需要这两个文件。
- 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，请为特定云供应商完成 OpenShift Container Platform 卸载流程。

5. 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
```

6. 从 [Red Hat OpenShift Cluster Manager](#) 下载安装 [pull secret](#)。此 pull secret 允许您与所含授权机构提供的服务进行身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

### 提示

另外，您还可以从[红帽客户门户网站](#)检索安装程序，您可以在其中指定要下载的安装程序版本。但是，您需要有一个有效的订阅才能访问此页。

## 12.7. 为集群节点 SSH 访问生成密钥对

在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 **core** 用户的 `~/.ssh/authorized_keys` 列表中，这将启用免密码身份验证。

将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。

如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。`./openshift-install gather` 命令还需要在集群节点上设置 SSH 公钥。



### 重要

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。



### 注意

您必须使用本地密钥，而不是使用特定平台方法配置的密钥，如 [AWS 密钥对](#)。

### 流程

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，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_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。



#### 注意

如果您计划在 **x86\_64**、**ppc64le** 和 **s390x** 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 **ed25519** 算法的密钥。相反，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

3. 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。



#### 注意

在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- a. 如果 **ssh-agent** 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

#### 输出示例

```
Agent pid 31874
```



#### 注意

如果集群处于 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。

4. 将 SSH 私钥添加到 **ssh-agent**：

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

- 1 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

## 输出示例

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

## 后续步骤

- 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，则必须为安装程序提供密钥。

## 12.8. 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。您还可以选择在安装准备阶段首先设置独立 **var** 分区。

### 12.8.1. 可选：创建独立 **/var** 分区

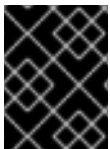
建议安装程序将 OpenShift Container Platform 的磁盘分区保留给安装程序。然而，在有些情况下您可能需要在文件系统的一部分中创建独立分区。

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

- /var/lib/containers**：保存随着系统中添加更多镜像和容器而增长的容器相关内容。
- /var/lib/etcd**：保存您可能希望独立保留的数据，比如 etcd 存储的性能优化。
- /var**：保存您可能希望独立保留的数据，以满足审计等目的。

通过单独存储 **/var** 目录的内容，可以更轻松地根据需要为区域扩展存储，并在以后重新安装 OpenShift Container Platform，并保持该数据的完整性。使用这个方法，您不必再次拉取所有容器，在更新系统时也不必复制大量日志文件。

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** 准备阶段插入一个创建的机器配置清单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**，以在 **manifest** 和 **openshift** 子目录中创建一组文件。在系统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. 创建用于配置额外分区的 Butane 配置。例如，将文件命名为 **\$HOME/clusterconfig/98-var-partition.bu**，将磁盘设备名称改为 **worker** 系统上存储设备的名称，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：

```
variant: openshift
version: 4.14.0
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 98-var-partition
storage:
  disks:
    - device: /dev/disk/by-id/<device_name> ❶
      partitions:
        - label: var
          start_mib: <partition_start_offset> ❷
          size_mib: <partition_size> ❸
          number: 5
      filesystems:
        - device: /dev/disk/by-partlabel/var
          path: /var
          format: xfs
          mount_options: [defaults, prjquota] ❹
          with_mount_unit: true
```

- ❶ 要分区的磁盘的存储设备名称。
- ❷ 在引导磁盘中添加数据分区时，推荐最少使用 25000 MiB(Mebibytes)。root 文件系统会自动调整大小以填充所有可用空间（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 以兆字节为单位的数据分区大小。
- 4 对于用于容器存储的文件系统，必须启用 **prjquota** 挂载选项。



### 注意

当创建单独的 **/var** 分区时，如果不同的实例类型没有相同的设备名称，则无法为 worker 节点使用不同的实例类型。

5. 从 Butane 配置创建一个清单，并将它保存到 **clusterconfig/openshift** 目录中。例如，运行以下命令：

```
$ butane $HOME/clusterconfig/98-var-partition.bu -o $HOME/clusterconfig/openshift/98-var-partition.yaml
```

6. 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。

## 12.8.2. 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. 进入包含安装程序的目录并运行以下命令：

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

- 1 对于 **<installation\_directory>**，请指定要存储安装程序创建的文件的目录名称。



### 重要

指定一个空目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。

b. 在提示符处，提供云的配置详情：

i. 可选：选择用于访问集群机器的 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. 为集群输入描述性名称。

vii. 粘贴 [Red Hat OpenShift Cluster Manager](#) 中的 **pull secret**。

2. 如果要安装三节点集群，请通过将 **compute.replicas** 参数设置为 **0** 来修改 **install-config.yaml** 文件。这样可确保集群的 control plane 可以调度。如需更多信息，请参阅“在 AWS 上安装三节点集群”。

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](#)。

### 12.8.3. 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，并确定它们中的任何站点是否需要绕过代理。默认情况下，所有集群出口流量都经过代理，包括对托管云供应商 API 的调用。如果需要，您将在 **Proxy** 对象的 **spec.noProxy** 字段中添加站点来绕过代理。



### 注意

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

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

## 流程

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: ec2.<aws_region>.amazonaws.com,elasticloadbalancing.
    <aws_region>.amazonaws.com,s3.<aws_region>.amazonaws.com 3
  additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

- 1 用于创建集群外 HTTP 连接的代理 URL。URL 方案必须是 **http**。
- 2 用于创建集群外 HTTPS 连接的代理 URL。
- 3 要从代理中排除的目标域名、IP 地址或其他网络 CIDR 的逗号分隔列表。在域前面加上 **.** 以仅匹配子域。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。如果您已将 Amazon **EC2**、**Elastic Load Balancing** 和 **S3** VPC 端点添加到 VPC 中，您必须将这些端点添加到 **noProxy** 字段。
- 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，其包含代理 HTTPS 连接所需的一个或多个额外 CA 证书。然后，Cluster Network Operator 会创建 **trusted-ca-bundle** 配置映射，将这些内容与 Red Hat Enterprise Linux CoreOS (RHCOS) 信任捆绑包合并，**Proxy** 对象的 **trustedCA** 字段中也会引用此配置映射。**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。

5

可选：决定 **Proxy** 对象的配置以引用 **trustedCA** 字段中 **user-ca-bundle** 配置映射的策略。允许的值是 **Proxyonly** 和 **Always**。仅在配置了 **http/https** 代理时，使用 **Proxyonly**



### 注意

安装程序不支持代理的 **readinessEndpoints** 字段。



### 注意

如果安装程序超时，重启并使用安装程序的 **wait-for** 命令完成部署。例如：

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

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。

## 12.8.4. 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手动启动集群机器，因此您必须生成 Kubernetes 清单和 Ignition 配置文件来配置机器。

安装配置文件转换为 Kubernetes 清单。清单嵌套到 Ignition 配置文件中，稍后用于配置集群机器。



### 重要

- OpenShift Container Platform 安装程序生成的 Ignition 配置文件包含 24 小时后过期的证书，然后在该时进行续订。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。一个例外是，您必须手动批准待处理的 **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 小时时间进行轮转。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. 进入包含 OpenShift Container Platform 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>**，请指定包含您创建的 **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 control plane 机器集的 Kubernetes 清单文件：

```
$ rm -f <installation_directory>/openshift/99_openshift-machine-api_master-control-plane-machine-set.yaml
```

4. 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
```



### 重要

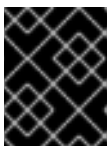
如果在用户置备的基础架构上安装集群时禁用了 **MachineAPI** 功能，则必须删除定义 worker 机器的 Kubernetes 清单文件。否则，集群将无法安装。

由于您要自行创建和管理 worker 机器，因此不需要初始化这些机器。



### 警告

如果您要安装一个三节点集群，请跳过以下步骤，以便可以调度 control plane 节点。



### 重要

当您将 control plane 节点从默认的不可调度配置为可以调度时，需要额外的订阅。这是因为 control plane 节点变为计算节点。

5. 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。

6. 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。

如果这样做，您必须在后续步骤中手动添加入口 DNS 记录。

7. 要创建 Ignition 配置文件，请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。

为安装目录中的 `bootstrap`、`control plane` 和计算节点创建 Ignition 配置文件。`kubeadmin-password` 和 `kubeconfig` 文件在 `./<installation_directory>/auth` 目录中创建：

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

## 12.9. 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

- 已获取 OpenShift Container Platform 安装程序和集群的 pull secret。
- 已为集群生成 Ignition 配置文件。
- 已安装 `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 ❶
```

- ❶ 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 ❶
```

- ❶ 此命令的输出是您的集群名称和随机字符串。

## 12.10. 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" 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> 1
--template-body file://<template>.yaml 2
--parameters file://<parameters>.json 3
```

- 1 **<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。
- 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-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：

|       |            |
|-------|------------|
| VpcId | 您的 VPC ID。 |
|-------|------------|



|                         |            |
|-------------------------|------------|
| <b>PublicSubnetIds</b>  | 新公共子网的 ID。 |
| <b>PrivateSubnetIds</b> | 新专用子网的 ID。 |

### 12.10.1. 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 12.17. VPC 的 CloudFormation 模板

AWS::CloudFormation::Template: 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:
      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
VpcId: !Ref VPC

Outputs:
VpcId:
  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"]]
    ]
PublicRouteTableId:
  Description: Public Route table ID
  Value: !Ref PublicRouteTable

```

## 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。

## 12.11. 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> ❶
```

- ❶ 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", ❶
    "ParameterValue": "mycluster" ❷
  },
  {
    "ParameterKey": "InfrastructureName", ❸
    "ParameterValue": "mycluster-<random_string>" ❹
  },
  {
    "ParameterKey": "HostedZoneId", ❺
    "ParameterValue": "<random_string>" ❻
  },
  {
    "ParameterKey": "HostedZoneName", ❼
    "ParameterValue": "example.com" ❽
  },
  {
    "ParameterKey": "PublicSubnets", ❾
    "ParameterValue": "subnet-<random_string>" ❿
  },
  {
    "ParameterKey": "PrivateSubnets", 11
    "ParameterValue": "subnet-<random_string>" 12
  }
]
```

```

    },
    {
      "ParameterKey": "VpcId", 13
      "ParameterValue": "vpc-<random_string>" 14
    }
  ]

```

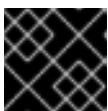
- 1 一个简短的、代表集群的名称用于主机名等。
  - 2 指定您为集群生成 **install-config.yaml** 文件时所用的集群名称。
  - 3 您的 Ignition 配置文件中为集群编码的集群基础架构名称。
  - 4 指定从 Ignition 配置文件元数据中提取的基础架构名称，其格式为 **<cluster-name>-<random-string>**。
  - 5 用来注册目标的 Route 53 公共区 ID。
  - 6 指定 Route 53 公共区 ID，其格式与 **Z21IXYZABCZ2A4** 类似。您可以从 AWS 控制台获取这个值。
  - 7 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 政府或 **secret** 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>RegisterNlbIpTargetsLambda</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。   |

### 12.11.1. 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 12.18. 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
    Z21XYZABCZ2A4.
    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 VpcId
      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](#)

RegisterTargetLambdalaRole:  
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

```

#### RegisterNlbIpTargets:

```

Type: "AWS::Lambda::Function"
Properties:
  Handler: "index.handler"
  Role:
    Fn::GetAtt:
      - "RegisterTargetLambdalaRole"
      - "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: "python3.8"
  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:
- "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: "python3.8"

```



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

如果要集群部署到 AWS 政府或 secret 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](#)。

## 12.12. 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

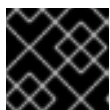
- 已配置了一个 AWS 帐户。
- 您可以通过运行 **aws configure**，将 AWS 密钥和区域添加到本地 AWS 配置集中。
- 已为集群生成 Ignition 配置文件。
- 您在 AWS 中创建并配置了 VPC 及相关子网。

## 流程

1. 创建一个 JSON 文件，其包含模板所需的参数值：

```
[
  {
    "ParameterKey": "InfrastructureName", ❶
    "ParameterValue": "mycluster-<random_string>" ❷
  },
  {
    "ParameterKey": "VpcCidr", ❸
    "ParameterValue": "10.0.0.0/16" ❹
  },
  {
    "ParameterKey": "PrivateSubnets", ❺
    "ParameterValue": "subnet-<random_string>" ❻
  },
  {
    "ParameterKey": "VpcId", ❼
    "ParameterValue": "vpc-<random_string>" ❽
  }
]
```

- 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>MasterSecurityGroup</b><br><b>Id</b> | Master 安全组 ID    |
| <b>WorkerSecurityGroup</b><br><b>Id</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 |

### 12.12.1. 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 12.19. 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])|(V(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  
VpcId: !Ref VpcId

**MasterIngressEtc:**

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

**MasterIngressIpsecIke:**

Type: AWS::EC2::SecurityGroupIngress  
Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId  
 SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
 Description: IPsec IKE packets  
 FromPort: 500  
 ToPort: 500  
 IpProtocol: udp

#### MasterIngressIpsecNat:

Type: AWS::EC2::SecurityGroupIngress  
 Properties:  
 GroupId: !GetAtt MasterSecurityGroup.GroupId  
 SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
 Description: IPsec NAT-T packets  
 FromPort: 4500  
 ToPort: 4500  
 IpProtocol: udp

#### MasterIngressIpsecEsp:

Type: AWS::EC2::SecurityGroupIngress  
 Properties:  
 GroupId: !GetAtt MasterSecurityGroup.GroupId  
 SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
 Description: IPsec ESP packets  
 IpProtocol: 50

#### MasterIngressWorkerIpsecIke:

Type: AWS::EC2::SecurityGroupIngress  
 Properties:  
 GroupId: !GetAtt MasterSecurityGroup.GroupId  
 SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
 Description: IPsec IKE packets  
 FromPort: 500  
 ToPort: 500  
 IpProtocol: udp

#### MasterIngressWorkerIpsecNat:

Type: AWS::EC2::SecurityGroupIngress  
 Properties:  
 GroupId: !GetAtt MasterSecurityGroup.GroupId  
 SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
 Description: IPsec NAT-T packets  
 FromPort: 4500  
 ToPort: 4500  
 IpProtocol: udp

#### MasterIngressWorkerIpsecEsp:

Type: AWS::EC2::SecurityGroupIngress  
 Properties:  
 GroupId: !GetAtt MasterSecurityGroup.GroupId  
 SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
 Description: IPsec ESP packets  
 IpProtocol: 50

#### 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

WorkerIngressIpsecIke:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
  GroupId: !GetAtt WorkerSecurityGroup.GroupId  
  SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
  Description: IPsec IKE packets  
  FromPort: 500  
  ToPort: 500  
  IpProtocol: udp

WorkerIngressIpsecNat:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
  GroupId: !GetAtt WorkerSecurityGroup.GroupId  
  SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
  Description: IPsec NAT-T packets  
  FromPort: 4500  
  ToPort: 4500  
  IpProtocol: udp

WorkerIngressIpsecEsp:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
  GroupId: !GetAtt WorkerSecurityGroup.GroupId  
  SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId  
  Description: IPsec ESP packets  
  IpProtocol: 50

WorkerIngressMasterIpsecIke:

Type: AWS::EC2::SecurityGroupIngress  
Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId  
 SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
 Description: IPsec IKE packets  
 FromPort: 500  
 ToPort: 500  
 IpProtocol: udp

#### WorkerIngressMasterIpsecNat:

Type: AWS::EC2::SecurityGroupIngress  
 Properties:  
 GroupId: !GetAtt WorkerSecurityGroup.GroupId  
 SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
 Description: IPsec NAT-T packets  
 FromPort: 4500  
 ToPort: 4500  
 IpProtocol: udp

#### WorkerIngressMasterIpsecEsp:

Type: AWS::EC2::SecurityGroupIngress  
 Properties:  
 GroupId: !GetAtt WorkerSecurityGroup.GroupId  
 SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
 Description: IPsec ESP packets  
 IpProtocol: 50

#### 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。

## 12.13. 使用流元数据访问 RHCOS AMI

在 OpenShift Container Platform 中，*流元数据*以 JSON 格式提供与 RHCOS 相关的标准化元数据，并将元数据注入集群中。流元数据是一种稳定的格式，支持多种架构，旨在自我记录以维护自动化。

您可以使用 **openshift-install** 的 **coreos print-stream-json** 子命令访问流元数据格式的引导镜像的信息。此命令提供了一种以可脚本、机器可读格式打印流元数据的方法。

对于用户置备的安装，**openshift-install** 二进制文件包含对经过测试用于 OpenShift Container Platform 的 RHCOS 引导镜像版本的引用，如 AWS AMI。

## 流程

要解析流元数据，请使用以下方法之一：

- 在 Go 程序中使用位于 <https://github.com/coreos/stream-metadata-go> 的正式 **stream-metadata-go** 库。您还可以查看库中的示例代码。
- 在 Python 或 Ruby 等其他编程语言中使用您首选编程语言的 JSON 库。
- 在处理 JSON 数据的命令行工具中，如 **jq**：
  - 为 AWS 区域输出当前的 **x86\_64** 或 **aarch64** AMI，如 **us-west-1**：

### 对于 x86\_64

```
$ openshift-install coreos print-stream-json | jq -r
'.architectures.x86_64.images.aws.regions["us-west-1"].image'
```

### 输出示例

```
ami-0d3e625f84626bbda
```

### 对于 aarch64

```
$ openshift-install coreos print-stream-json | jq -r
'.architectures.aarch64.images.aws.regions["us-west-1"].image'
```

输出示例

```
ami-0af1d3b7fa5be2131
```

这个命令的输出是您指定的架构和 **us-west-1** 区域的 AWS AMI ID。AMI 必须与集群属于同一区域。

12.14. AWS 基础架构的 RHCOS AMI

红帽提供了对可手动为 OpenShift Container Platform 节点指定的各种 AWS 区域和实例架构有效的 Red Hat Enterprise Linux CoreOS(RHCOS)AMI。



注意

通过导入您自己的 AMI，您还可以安装到没有公布的 RHCOS AMI 的区域。

表 12.3. x86\_64 RHCOS AMIs

| AWS 区          | AWS AMI               |
|----------------|-----------------------|
| af-south-1     | ami-01860370941726bdd |
| ap-east-1      | ami-05bc702cdaf7e4251 |
| ap-northeast-1 | ami-098932fd93c15690d |
| ap-northeast-2 | ami-006f4e02d97910a36 |
| ap-northeast-3 | ami-0c4bd5b1724f82273 |
| ap-south-1     | ami-0cbf22b638724853d |
| ap-south-2     | ami-031f4d165f4b429c4 |
| ap-southeast-1 | ami-0dc3e381a731ab9fc |
| ap-southeast-2 | ami-032ae8d0f287a66a6 |
| ap-southeast-3 | ami-0393130e034b86423 |
| ap-southeast-4 | ami-0b38f776bded7d7d7 |
| ca-central-1   | ami-058ea81b3a1d17edd |
| eu-central-1   | ami-011010debd974a250 |
| eu-central-2   | ami-0623b105ae811a5e2 |



| AWS 区         | AWS AMI               |
|---------------|-----------------------|
| eu-north-1    | ami-0c4bb9ce04f3526d4 |
| eu-south-1    | ami-06c29eccd3d74df52 |
| eu-south-2    | ami-00e0b5f3181a3f98b |
| eu-west-1     | ami-087bfa513dc600676 |
| eu-west-2     | ami-0ebad59c0e9554473 |
| eu-west-3     | ami-074e63b65eaf83f96 |
| me-central-1  | ami-0179d6ae1d908ace9 |
| me-south-1    | ami-0b60c75273d3efcd7 |
| sa-east-1     | ami-0913cbfbfa9a7a53c |
| us-east-1     | ami-0f71dcd99e6a1cd53 |
| us-east-2     | ami-0545fae7edbbbf061 |
| us-gov-east-1 | ami-081eabdc478e501e5 |
| us-gov-west-1 | ami-076102c394767f319 |
| us-west-1     | ami-0609e4436c4ae5eff |
| us-west-2     | ami-0c5d3e03c0ab9b19a |

表 12.4. aarch64 RHCOS AMI

| AWS 区          | AWS AMI               |
|----------------|-----------------------|
| af-south-1     | ami-08dd66a61a2caa326 |
| ap-east-1      | ami-0232cd715f8168c34 |
| ap-northeast-1 | ami-0bc0b17618da96700 |
| ap-northeast-2 | ami-0ee505fb62eed2fd6 |
| ap-northeast-3 | ami-0462cd2c3b7044c77 |

| AWS 区域         | AWS AMI               |
|----------------|-----------------------|
| ap-south-1     | ami-0e0b4d951b43adc58 |
| ap-south-2     | ami-06d457b151cc0e407 |
| ap-southeast-1 | ami-0874e1640dfc15f17 |
| ap-southeast-2 | ami-05f215734ceb0f5ad |
| ap-southeast-3 | ami-073030df265c88b3f |
| ap-southeast-4 | ami-043f4c40a6fc3238a |
| ca-central-1   | ami-0840622f99a32f586 |
| eu-central-1   | ami-09a5e6ebe667ae6b5 |
| eu-central-2   | ami-0835cb1bf387e609a |
| eu-north-1     | ami-069ddbda521a10a27 |
| eu-south-1     | ami-09c5cc21026032b4c |
| eu-south-2     | ami-0c36ab2a8bbeed045 |
| eu-west-1      | ami-0d2723c8228cb2df3 |
| eu-west-2      | ami-0abd66103d069f9a8 |
| eu-west-3      | ami-08c7249d59239fc5c |
| me-central-1   | ami-0685f33ebb18445a2 |
| me-south-1     | ami-0466941f4e5c56fe6 |
| sa-east-1      | ami-08cdc0c8a972f4763 |
| us-east-1      | ami-0d461970173c4332d |
| us-east-2      | ami-0e9cdc0e85e0a6aeb |
| us-gov-east-1  | ami-0b896df727672ce09 |
| us-gov-west-1  | ami-0b896df727672ce09 |

| AWS 区     | AWS AMI               |
|-----------|-----------------------|
| us-west-1 | ami-027b7cc5f4c74e6c1 |
| us-west-2 | ami-0b189d89b44bdfbf2 |

### 12.14.1. 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。

如果您要部署到 AWS SDK 不支持的区域，且您没有指定自定义 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。

### 12.14.2. 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
```

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
```

3. 将上传至 Amazon S3 的 RHCOS 版本导出为环境变量：

```
$ export RHCOS_VERSION=<version> 1
```

1 1 1 RHCOS VMDK 版本，如 **4.14.0**。

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-4.7.0-x86_64-aws.x86_64",
      "ImportTaskId": "import-snap-fh6i8uil",
      "SnapshotTaskDetail": {
        "Description": "rhcos-4.7.0-x86_64-aws.x86_64",
        "DiskImageSize": 819056640.0,
        "Format": "VMDK",
        "SnapshotId": "snap-06331325870076318",
        "Status": "completed",
        "UserBucket": {
          "S3Bucket": "external-images",
          "S3Key": "rhcos-4.7.0-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**, **aarch64**, **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](#)。

## 12.15. 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。您可以按照以下方法：

- 为集群提供 **bootstrap.ign** Ignition 配置文件的位置。此文件位于您的安装目录中。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。
- 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

- 运行以下命令来创建存储桶：

```
$ aws s3 mb s3://<cluster-name>-infra ❶
```

- ❶ **<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。

如果需要，您必须为 S3 存储桶使用预签名 URL，而不是 **s3://** 模式：

- 部署到具有与 AWS SDK 不同端点的区域。
- 部署代理。
- 提供您自己的自定义端点。

- 运行以下命令，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 ❶
```

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

- 运行以下命令验证文件是否已上传：

```
$ aws s3 ls s3://<cluster-name>-infra/
```

## 输出示例

```
2019-04-03 16:15:16    314878 bootstrap.ign
```



## 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": "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:<aws_region>:<account_number>:function:
<dns_stack_name>-RegisterNlbTargets-<random_string>" 18
},
{
  "ParameterKey": "ExternalApiTargetGroupArn", 19
  "ParameterValue": "arn:aws:elasticloadbalancing:<aws_region>:
<account_number>:targetgroup/<dns_stack_name>-Exter-<random_string>" 20
},
{
  "ParameterKey": "InternalApiTargetGroupArn", 21
  "ParameterValue": "arn:aws:elasticloadbalancing:<aws_region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<random_string>" 22
},
{
  "ParameterKey": "InternalServiceTargetGroupArn", 23
  "ParameterValue": "arn:aws:elasticloadbalancing:<aws_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 根据您的选择的架构，当前 Red Hat Enterprise Linux CoreOS (RHCOS) AMI 用于 bootstrap 节点。

- 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**。
5. 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。
  6. 可选：如果要使用代理部署集群，您必须更新模板中的 ignition 以添加 **ignition.config.proxy** 字段。另外，如果您已将 Amazon EC2、Elastic Load Balancing 和 S3 VPC 端点添加到 VPC 中，您必须将这些端点添加到 **noProxy** 字段。
  7. 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> 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-bootstrap**。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-bootstrap/12944486-2add-11eb-9dee-12dace8e3a83
```

### 8. 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。 |

## 12.15.1. 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 12.20. 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

**BootstrapInstanceType:**

Description: Instance type for the bootstrap EC2 instance

Default: "i3.large"

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
          - RegisterNLbpTargetsLambdaArn
          - 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: !Ref BootstrapInstanceType
    NetworkInterfaces:
      - AssociatePublicIpAddress: "true"
        DeviceIndex: "0"

```

```

GroupSet:
  - !Ref "BootstrapSecurityGroup"
  - !Ref "MasterSecurityGroupId"
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](#)。

## 12.16. 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", 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": "" 26
},
{
  "ParameterKey": "AutoRegisterELB", 27
  "ParameterValue": "yes" 28
},
{
  "ParameterKey": "RegisterNlbTargetsLambdaArn", 29
  "ParameterValue": "arn:aws:lambda:<aws_region>:<account_number>:function:
<dns_stack_name>-RegisterNlbTargets-<random_string>" 30
},
{
  "ParameterKey": "ExternalApiTargetGroupArn", 31
  "ParameterValue": "arn:aws:elasticloadbalancing:<aws_region>:
<account_number>:targetgroup/<dns_stack_name>-Exter-<random_string>" 32
},
{
  "ParameterKey": "InternalApiTargetGroupArn", 33

```

```

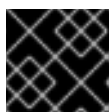
    "ParameterValue": "arn:aws:elasticloadbalancing:<aws_region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<random_string>" 34
  },
  {
    "ParameterKey": "InternalServiceTargetGroupArn", 35
    "ParameterValue": "arn:aws:elasticloadbalancing:<aws_region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<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 根据您的选择的架构，当前用于 control plane 机器的 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 根据您的选择的架构，用于 control plane 机器的 AWS 实例类型。



- 26 实例类型值与 control plane 机器的最低资源要求对应。例如，**m6i.xlarge** 是 AMD64 的类型，**m6g.xlarge** 是 ARM64 的类型。
  - 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 **RegisterNlbTargetsLambda** 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>
```

## 12.16.1. 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 12.21. 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: ""
    Description: unused
    Type: String
  PrivateHostedZoneId:
    Default: ""
    Description: unused
    Type: String
  PrivateHostedZoneName:
    Default: ""
    Description: unused
    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](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

**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: "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"
AutoRegisterELB:
  default: "Use Provided ELB Automation"

```

Conditions:

```
DoRegistration: !Equals ["yes", !Ref AutoRegisterELB]
```

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:
    - AssociatePublicIpAddress: "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

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。

## 12.17. 在 AWS 中创建 WORKER 节点

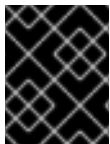
您可以在 Amazon Web Services (AWS) 中创建 worker 节点，供集群使用。



### 注意

如果您要安装三节点集群，请跳过这一步。三节点集群包含三个 control plane 机器，它们也可以充当计算机器。

您可以使用提供的 CloudFormation 模板和自定义参数文件创建代表 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", ❶
    "ParameterValue": "mycluster-<random_string>" ❷
  },
  {
    "ParameterKey": "RhcosAmi", ❸
    "ParameterValue": "ami-<random_string>" ❹
  },
  {
    "ParameterKey": "Subnet", ❺
    "ParameterValue": "subnet-<random_string>" ❻
  }
]
```



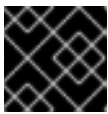
```

    },
    {
      "ParameterKey": "WorkerSecurityGroupd", 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": "" 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 根据您的选择的架构，当前 Red Hat Enterprise Linux CoreOS (RHCOS) AMI 用于 worker 节点。
- 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 **WorkerSecurityGroupd** 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 根据您选择的架构，用于计算机器的 AWS 实例类型。
  - 16 实例类型值对应于计算机器的最低资源要求。例如，**m6i.large** 是 AMD64 的类型，**m6g.large** 是 ARM64 的类型。
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. 可选：如果将 **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. 可选：如果您使用 AWS Marketplace 镜像部署，请使用从订阅获取的 AMI ID 更新 **Worker0.type.properties.ImageID** 参数。
  5. 使用 CloudFormation 模板创建代表 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
```



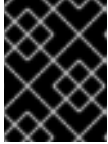
### 注意

CloudFormation 模板会创建一个堆栈，它代表一个 worker 节点。

6. 确认模板组件已存在：

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

7. 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。

**12.17.1. 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 12.22. 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 worker nodes into.
    Type: AWS::EC2::Subnet::Id
  WorkerSecurityGroupId:
    Description: The worker security group ID to associate with worker 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 worker nodes.
    Type: String
  WorkerInstanceType:
    Default: m5.large
    Type: String

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:
      - AssociatePublicIpAddress: "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

## 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。

## 12.18. 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

- 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。

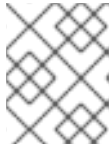
2 要查看不同的安装详情，请指定 **warn**、**debug** 或 **error**，而不是 **info**。

### 输出示例

```
INFO Waiting up to 20m0s for the Kubernetes API at
```

```
https://api.mycluster.example.com:6443...
INFO API v1.27.3 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

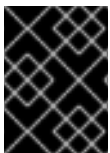
在 control plane 初始化后，它会设置计算节点，并以 Operator 的形式安装其他服务。

## 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。

## 12.19. 通过下载二进制文件安装 OPENSIFT CLI

您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

### 12.19.1. 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. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。
5. 解包存档：

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

6. 将 **oc** 二进制文件放到 **PATH** 中的目录中。  
要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

## 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 12.19.2. 在 Windows 上安装 OpenShift CLI

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

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
4. 使用 ZIP 程序解压存档。
5. 将 **oc** 二进制文件移到 **PATH** 中的目录中。  
要查看您的 **PATH**，请打开命令提示并执行以下命令：

```
C:\> path
```

## 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

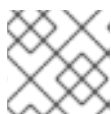
```
C:\> oc <command>
```

### 12.19.3. 在 macOS 上安装 OpenShift CLI

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

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。



#### 注意

对于 macOS arm64，请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

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**，请打开终端并执行以下命令：

```
$ echo $PATH
```

## 验证

- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

## 12.20. 使用 CLI 登录集群

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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

### 先决条件

- 已部署 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. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

### 输出示例

```
system:admin
```

## 12.21. 批准机器的证书签名请求

当您添加机器到集群时，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。必须首先批准客户端请求，然后批准服务器请求。

### 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   v1.27.3
master-1  Ready    master   63m   v1.27.3
master-2  Ready    master   64m   v1.27.3
```

输出中列出了您创建的所有机器。



#### 注意

在有些 CSR 被批准前，前面的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，在您添加的机器的所有待处理 CSR 都处于 **Pending** 状态后，请批准集群机器的 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它们，证书将会轮转，每个节点会存在多个证书。您必须批准所有这些证书。批准客户端 CSR 后，Kubelet 为服务证书创建一个二级 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，则后续提供证书续订请求由 **machine-approver** 自动批准。



#### 注意

对于在未启用机器 API 的平台上运行的集群，如裸机和其他用户置备的基础架构，您必须实施一种方法来自动批准 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。该方法必须监视新的 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。

- 要单独批准，请对每个有效的 CSR 运行以下命令：

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

**1** **<csr\_name>** 是当前 CSR 列表中 CSR 的名称。

- 要批准所有待处理的 CSR，请运行以下命令：

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{{"\n"}}{{end}}{{end}}' | xargs --no-run-if-empty oc adm certificate approve
```



### 注意

在有些 CSR 被批准前，一些 Operator 可能无法使用。

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

```
$ oc get csr
```

### 输出示例

```
NAME      AGE   REQUESTOR                                     CONDITION
csr-bfd72 5m26s system:node:ip-10-0-50-126.us-east-2.compute.internal
Pending
csr-c57lv 5m26s system:node:ip-10-0-95-157.us-east-2.compute.internal
Pending
...
```

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

- 要单独批准，请对每个有效的 CSR 运行以下命令：

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

**1** **<csr\_name>** 是当前 CSR 列表中 CSR 的名称。

- 要批准所有待处理的 CSR，请运行以下命令：

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

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

```
$ oc get nodes
```

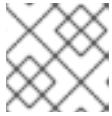
### 输出示例

```
NAME      STATUS   ROLES    AGE   VERSION
master-0  Ready   master   73m   v1.27.3
master-1  Ready   master   73m   v1.27.3
```

```

master-2 Ready   master 74m v1.27.3
worker-0 Ready   worker 11m v1.27.3
worker-1 Ready   worker 11m v1.27.3

```



### 注意

批准服务器 CSR 后可能需要几分钟时间让机器过渡到 **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](#)。

## 12.22. 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

- 观察集群组件上线：

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

### 输出示例

| NAME                          | VERSION | AVAILABLE | PROGRESSING | DEGRADED | SINCE |
|-------------------------------|---------|-----------|-------------|----------|-------|
| authentication                | 4.14.0  | True      | False       | False    | 19m   |
| baremetal                     | 4.14.0  | True      | False       | False    | 37m   |
| cloud-credential              | 4.14.0  | True      | False       | False    | 40m   |
| cluster-autoscaler            | 4.14.0  | True      | False       | False    | 37m   |
| config-operator               | 4.14.0  | True      | False       | False    | 38m   |
| console                       | 4.14.0  | True      | False       | False    | 26m   |
| csi-snapshot-controller       | 4.14.0  | True      | False       | False    | 37m   |
| dns                           | 4.14.0  | True      | False       | False    | 37m   |
| etcd                          | 4.14.0  | True      | False       | False    | 36m   |
| image-registry                | 4.14.0  | True      | False       | False    | 31m   |
| ingress                       | 4.14.0  | True      | False       | False    | 30m   |
| insights                      | 4.14.0  | True      | False       | False    | 31m   |
| kube-apiserver                | 4.14.0  | True      | False       | False    | 26m   |
| kube-controller-manager       | 4.14.0  | True      | False       | False    | 36m   |
| kube-scheduler                | 4.14.0  | True      | False       | False    | 36m   |
| kube-storage-version-migrator | 4.14.0  | True      | False       | False    | 37m   |
| machine-api                   | 4.14.0  | True      | False       | False    | 29m   |
| machine-approver              | 4.14.0  | True      | False       | False    | 37m   |
| machine-config                | 4.14.0  | True      | False       | False    | 36m   |
| marketplace                   | 4.14.0  | True      | False       | False    | 37m   |
| monitoring                    | 4.14.0  | True      | False       | False    | 29m   |
| network                       | 4.14.0  | True      | False       | False    | 38m   |
| node-tuning                   | 4.14.0  | True      | False       | False    | 37m   |
| openshift-apiserver           | 4.14.0  | True      | False       | False    | 32m   |

|                                          |        |      |       |       |     |
|------------------------------------------|--------|------|-------|-------|-----|
| openshift-controller-manager             | 4.14.0 | True | False | False | 30m |
| openshift-samples                        | 4.14.0 | True | False | False | 32m |
| operator-lifecycle-manager               | 4.14.0 | True | False | False | 37m |
| operator-lifecycle-manager-catalog       | 4.14.0 | True | False | False | 37m |
| operator-lifecycle-manager-packageserver | 4.14.0 | True | False | False | 32m |
| service-ca                               | 4.14.0 | True | False | False | 38m |
| storage                                  | 4.14.0 | True | False | False | 37m |

## 2. 配置不可用的 Operator。

### 12.22.1. 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。

显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，这仅适用于非生产集群。

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[为 AWS 用户置备的基础架构配置 registry](#)。

#### 12.22.1.1. 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](#)。

#### 12.22.1.2. 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 时会丢失所有镜像。

#### 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这个命令，**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。

#### 12.23. 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。

## 12.24. 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>
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
```

3. 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' ❶
```

- ❶ 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。

4. 获取集群域的公共托管区 ID :

```
$ aws route53 list-hosted-zones-by-name \
  --dns-name "<domain_name>" \ ❶
  --query 'HostedZones[? Config.PrivateZone != `true` && Name ==
`<domain_name>.`].Id' ❷
  --output text
```

- ❶ ❷ 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**。

5. 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 '{ ❶
> "Changes": [
> {
>   "Action": "CREATE",
>   "ResourceRecordSet": {
>     "Name": "\\052.apps.<cluster_domain>", ❷
>     "Type": "A",
>     "AliasTarget":{
>       "HostedZoneId": "<hosted_zone_id>", ❸
>       "DNSName": "<external_ip>.", ❹
>       "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。请确定在该参数值中包含最后的句点 (.)。

## 12.25. 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

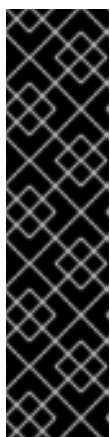
- 在包含安装程序的目录中完成集群安装：

```
$ ./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: "password"
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。一个例外是，您必须手动批准待处理的 **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。

/validating-an-installation.adoc

## 12.26. 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 控制台的更多详情](#)，请参阅 [访问 Web 控制台](#)。

## 12.27. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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](#)。

## 12.28. 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](#)。

## 12.29. 后续步骤

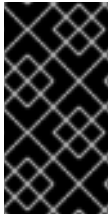
- [验证安装](#)。

- [自定义集群](#)
- [远程健康报告](#)
- [删除云供应商凭证](#)

## 第 13 章 在带有 AWS LOCAL ZONES 上的 WORKER 节点的 AWS 上安装集群

您可以通过在 `install-config.yaml` 文件的边缘计算池中设置区名称，或者在列出 Local Zone 子网的现有 VPC 上安装集群，在 Amazon Web Services (AWS) Local Zones 中安装 OpenShift Container Platform 集群。

AWS Local Zones 是一种基础架构类型，可将云资源放在满足要求的区域。如需更多信息，请参阅 [AWS 区域文档](#)。

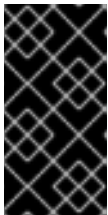


### 重要

执行安装程序置备的基础架构安装的步骤仅作为示例。在现有 VPC 上安装集群需要了解云供应商和 OpenShift Container Platform 安装过程。您可以使用 CloudFormation 模板来帮助完成这些步骤，或者帮助您建模您自己的集群安装。您可以决定使用其他方法生成这些资源，而不使用 CloudFormation 模板来创建资源。

### 13.1. 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。
- 您可以阅读[选择集群安装方法并为用户准备它的文档](#)。
- 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。集群将继续使用您当前的 AWS 凭证为集群的整个生命周期创建 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。

- 您记下了区域和支持的 [AWS 区域位置](#)，在其中创建网络资源。
- 您可以阅读每个 AWS 本地区域位置的[功能](#)。
- 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

如果要配置代理，请务必查看此站点列表。

- 为创建集群的用户添加权限，以使用 `ec2:ModifyAvailabilityZoneGroup` 修改 Local Zone 组。例如：

#### 附加到用户或组的 permissive IAM 策略示例

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Action": [
    "ec2:ModifyAvailabilityZoneGroup"
  ],
  "Effect": "Allow",
  "Resource": "*"
}
```

## 13.2. AWS 本地区中的集群限制

当您试图在 Amazon Web Services (AWS) 本地区中使用默认安装配置部署集群时，有一些限制。

### 重要

以下列出了在 AWS Local Zones 中部署集群时的详细信息限制：

- 在 Local Zone 中的一个 Amazon EC2 示例和 Region 中的一个 Amazon EC2 实例之间的最大传输单元 (MTU) 为 **1300**。这会导致集群范围的网络 MTU 根据部署中使用的网络插件而改变。
- 网络资源（如 Network Load Balancer (NLB), Classic Load Balancer, 和 Network Address Translation (NAT) Gateways）在 AWS Local Zones 中不会被全局支持。
- 对于 AWS 上的 OpenShift Container Platform 集群，AWS Elastic Block Storage (EBS) **gp3** 类型卷是节点卷和存储类的默认设置。此卷类型在 Local Zone 位置没有全局可用。默认情况下，在 Local Zones 中运行的节点使用 **gp2** EBS 卷进行部署。在 Local Zone 节点上创建工作负载时，必须设置 **gp2-csi StorageClass**。

如果您希望安装程序为 OpenShift Container Platform 集群自动创建 Local Zone 子网，则使用此方法会有针对特定配置的限制。

### 重要

当您将安装程序设置为自动为 OpenShift Container Platform 集群创建子网时，会有以下配置限制：

- AWS Local Zone 的专用子网与父区的路由表关联，以便每个专用子网都可以将出口流量路由到互联网。如果在集群安装过程中没有此路由表，则专用子网将与 Virtual Private Cloud (VPC) 中的第一个可用私有路由表相关联。此方法仅对 OpenShift Container Platform 集群中的 AWS Local Zones 子网有效。

### 其他资源

- [存储类](#)
- [Ingress Controller 分片](#)

## 13.3. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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型的基础架构上执行受限网络安装。在此过程中，您可以下载所需的内容，并使用它为镜像 registry 填充安装软件包。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。在更新集群前，您要更新镜像 registry 的内容。

## 13.4. 获取 AWS MARKETPLACE 镜像

如果要使用 AWS Marketplace 镜像部署 OpenShift Container Platform 集群，您必须首先通过 AWS 订阅。订阅提供的提供的 AMI ID 可让您使用安装程序用来部署 worker 节点的 AMI ID。



### 注意

您应该只修改计算机器的 RHCOS 镜像以使用 AWS Marketplace 镜像。control plane 机器和基础架构节点不需要 OpenShift Container Platform 订阅，且默认使用公共 RHCOS 默认镜像，这不会对您的 AWS 账单造成订阅成本。因此，您不应该修改集群默认引导镜像或 control plane 引导镜像。将 AWS Marketplace 镜像应用到它们时，会生成无法恢复的额外许可成本。

### 先决条件

- 有 AWS 账户购买的产品。此帐户不必与用于安装集群的帐户相同。

### 流程

1. 从 [AWS Marketplace](#) 完成 OpenShift Container Platform 订阅。
2. 记录特定区域的 AMI ID。作为安装过程的一部分，您必须在部署集群前使用这个值更新 **install-config.yaml** 文件。

### 使用 AWS Marketplace worker 节点的 install-config.yaml 文件示例

```
apiVersion: v1
baseDomain: example.com
compute:
- hyperthreading: Enabled
  name: worker
  platform:
    aws:
      amiID: ami-06c4d345f7c207239 1
      type: m5.4xlarge
      replicas: 3
  metadata:
    name: test-cluster
  platform:
    aws:
```

```
region: us-east-2
sshKey: ssh-ed25519 AAAA...
pullSecret: '{"auths": ...}'
```

- 1 来自 AWS Marketplace 订阅的 AMI ID。
- 2 您的 AMI ID 与特定的 AWS 区域相关联。在创建安装配置文件时，请确保选择配置订阅时指定的同一 AWS 区域。

## 13.5. 通过下载二进制文件安装 OPENSIFT CLI

您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

### 13.5.1. 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. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。
5. 解包存档：

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

6. 将 **oc** 二进制文件放到 **PATH** 中的目录中。  
要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 13.5.2. 在 Windows 上安装 OpenShift CLI

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

## 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
4. 使用 ZIP 程序解压存档。
5. 将 **oc** 二进制文件移到 **PATH** 中的目录中。  
要查看您的 **PATH**，请打开命令提示并执行以下命令：

```
C:\> path
```

## 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 13.5.3. 在 macOS 上安装 OpenShift CLI

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

## 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。



### 注意

对于 macOS arm64，请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

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**，请打开终端并执行以下命令：

```
$ echo $PATH
```

## 验证

- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

### 13.6. 准备安装



在将节点扩展到本地区前，您必须为集群安装环境准备某些资源。

### 13.6.1. 获取安装程序

在安装 OpenShift Container Platform 前，将安装文件下载到您用于安装的主机上。

#### 先决条件

- 您有一台运行 Linux 或 macOS 的计算机，至少有 1.2 GB 本地磁盘空间。

#### 流程

1. 进入 Red Hat Hybrid Cloud Console 上的 [Cluster Type](#) 页。如果您有红帽帐户，请使用您的凭证登录。如果没有，请创建一个帐户。
2. 在页的 **Run it yourself** 部分中选择您的基础架构供应商。
3. 从 **OpenShift 安装程序** 下的下拉菜单中选择您的主机操作系统和架构，然后点 **下载安装程序**。
4. 将下载的文件保存在要存储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几个文件。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。删除集群需要这两个文件。
- 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，请为特定云供应商完成 OpenShift Container Platform 卸载流程。

5. 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
```

6. 从 [Red Hat OpenShift Cluster Manager](#) 下载安装 [pull secret](#)。此 pull secret 允许您与所含授权机构提供的服务进行身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

#### 提示

另外，您还可以从[红帽客户门户网站](#)检索安装程序，您可以在其中指定要下载的安装程序版本。但是，您需要有一个有效的订阅才能访问此页。

### 13.6.2. 为集群节点 SSH 访问生成密钥对

在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 **core** 用户的 **~/.ssh/authorized\_keys** 列表中，这将启用免密码身份验证。

将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。

如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。**./openshift-install gather** 命令还需要在集群节点上设置 SSH 公钥。

**重要**

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。

**注意**

您必须使用本地密钥，而不是使用特定平台方法配置 [的密钥](#)，如 [AWS 密钥对](#)。

**流程**

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，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_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。

**注意**

如果您计划在 **x86\_64**、**ppc64le** 和 **s390x** 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 **ed25519** 算法的密钥。相反，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

3. 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。

**注意**

在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- a. 如果 **ssh-agent** 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

**输出示例**

```
Agent pid 31874
```

**注意**

如果集群处于 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。

4. 将 SSH 私钥添加到 **ssh-agent**：

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

- 1 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

**输出示例**

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

**后续步骤**

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

**13.6.3. 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。您还可以选择在安装准备阶段首先设置独立 **var** 分区。

**13.6.4. 集群安装的最低资源要求**

每台集群机器都必须满足以下最低要求：

**表 13.1. 最低资源要求**

| 机器                   | 操作系统                     | vCPU [1] | 虚拟内存  | Storage | 每秒输入/输出 (IOPS) [2] |
|----------------------|--------------------------|----------|-------|---------|--------------------|
| bootstrap            | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Control plane (控制平面) | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Compute              | RHCOS、RHEL 8.6 及更新版本 [3] | 2        | 8 GB  | 100 GB  | 300                |

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

2. OpenShift Container Platform 和 Kubernetes 对磁盘性能非常敏感，建议使用更快的存储速度，特别是 control plane 节点上需要 10 ms p99 fsync 持续时间的 etcd。请注意，在许多云平台上，存储大小和 IOPS 可一起扩展，因此您可能需要过度分配存储卷来获取足够的性能。
3. 与所有用户置备的安装一样，如果您选择在集群中使用 RHEL 计算机，则负责所有操作系统生命周期管理和维护，包括执行系统更新、应用补丁和完成所有其他必要的任务。RHEL 7 计算机器的使用已弃用，并已在 OpenShift Container Platform 4.10 及更新的版本中删除。

### 注意

从 OpenShift Container Platform 版本 4.13 开始，RHCOS 基于 RHEL 版本 9.2，它更新了微架构要求。以下列表包含每个架构需要的最小指令集架构 (ISA)：

- x86-64 体系结构需要 x86-64-v2 ISA
- ARM64 架构需要 ARMv8.0-A ISA
- IBM Power 架构需要 Power 9 ISA
- s390x 架构需要 z14 ISA

如需更多信息，请参阅 [RHEL 架构](#)。

如果平台的实例类型满足集群机器的最低要求，则 OpenShift Container Platform 支持使用它。

### 13.6.5. 为 AWS 测试的实例类型

以下 Amazon Web Services (AWS) 实例类型已使用 OpenShift Container Platform 测试，以用于 AWS Local Zones。

### 注意

将以下图中包含的机器类型用于 AWS 实例。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 13.1. 基于 AWS 本地区的 64 位 x86 架构的机器类型

- c5.\*
- c5d.\*
- m6i.\*
- m5.\*
- r5.\*
- t3.\*

### 其他资源

- 如需有关 AWS Local Zones 和支持的实例类型和服务的更多信息，请参阅 AWS 文档中的 [AWS Local Zones 功能](#)。

### 13.6.6. 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. 进入包含安装程序的目录并运行以下命令：

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

- 1** 对于 **<installation\_directory>**，请指定要存储安装程序创建的文件目录名称。



#### 重要

指定一个空目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。

- b. 在提示符处，提供云的配置详情：

- i. 可选：选择用于访问集群机器的 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。



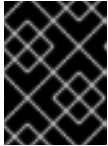
#### 注意

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。您指定的区域必须是包含您为 AWS 帐户选择的区域相同的区域。

- 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

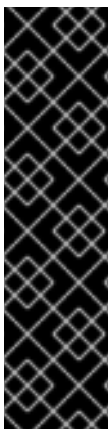
**install-config.yaml** 文件会在安装过程中消耗掉。如果要重复使用该文件，您必须立即备份该文件。

## 13.6.7. Edge 计算池和 AWS 本地区域

边缘 worker 节点是在 AWS Local Zones 位置中运行的污点 worker 节点。

在部署使用 Local Zones 的集群时，请考虑以下点：

- 本地区域中的 Amazon EC2 实例比可用区中的 Amazon EC2 实例的成本更高。
- 应用程序和最终用户之间的延迟较低在本地区域中，延迟可能会因位置而异。例如，在 Local Zones 和 Availability Zones 混合了入口流量时，一些工作负载会有一个延迟影响。



### 重要

通常，本地区域中的 Amazon EC2 实例和 Region 中的 Amazon EC2 实例之间的最大传输单元 (MTU) 为 1300。如需更多信息，请参阅 AWS 文档中的 [Local Zones 如何工作](#)。对于开销，集群网络 MTU 必须总是小于 EC2 MTU。具体开销由您的网络插件决定，例如：

- OVN-Kubernetes: **100 字节**
- OpenShift SDN: **50 字节**

网络插件可以提供额外的功能，如 IPsec，它们还必须减少 MTU。如需更多信息，请参阅文档。

OpenShift Container Platform 4.12 引入了一个新的计算池 *edge*，用于在远程区中使用。边缘计算池配置在 AWS Local Zones 位置之间很常见。由于 Local Zone 资源上的 EC2 和 EBS 等资源的类型和大小限制，默认的实例类型可能与传统的 worker 池不同。

Local Zone 位置的默认 Elastic Block Store (EBS) 是 **gp2**，它与常规 worker 池不同。根据区上的实例产品，边缘计算池中的每个 Local Zone 使用的实例类型可能与 worker 池不同。

边缘计算池创建新的标签，供开发人员用来将应用程序部署到 AWS Local Zones 节点上。新标签包括：

- **node-role.kubernetes.io/edge=""**
- **machine.openshift.io/zone-type=local-zone**
- **machine.openshift.io/zone-group=\$ZONE\_GROUP\_NAME**

默认情况下，边缘计算池的机器集定义 **NoSchedule** 污点，以防止常规工作负载分散到 Local Zone 实例中。只有用户在 pod 规格中定义容限时，用户才能运行用户工作负载。

以下示例演示了使用边缘机器池的 `install-config.yaml` 文件。

### 使用自定义实例类型使用边缘池的配置

```
apiVersion: v1
baseDomain: devcluster.openshift.com
metadata:
  name: ipi-localzone
compute:
- name: edge
  platform:
    aws:
      type: m5.4xlarge
platform:
  aws:
    region: us-west-2
pullSecret: '{"auths": ...}'
sshKey: ssh-ed25519 AAAA...
```

实例类型因位置而异。要验证集群运行的 Local Zone 中的可用性，请参阅 AWS 文档。

### 使用自定义 EBS 类型使用边缘池的配置

```
apiVersion: v1
baseDomain: devcluster.openshift.com
metadata:
  name: ipi-localzone
compute:
- name: edge
  platform:
    aws:
      rootVolume:
        type: gp3
        size: 120
platform:
  aws:
    region: us-west-2
pullSecret: '{"auths": ...}'
sshKey: ssh-ed25519 AAAA...
```

EBS 类型因位置而异。检查 AWS 文档，以验证集群在其中运行的 Local Zone 中的可用性。

### 使用自定义安全组使用边缘池的配置

```
apiVersion: v1
baseDomain: devcluster.openshift.com
metadata:
  name: ipi-localzone
compute:
- name: edge
  platform:
    aws:
      additionalSecurityGroupIDs:
        - sg-1 
        - sg-2
```



```
platform:
  aws:
    region: us-west-2
pullSecret: '{"auths": ...}'
sshKey: ssh-ed25519 AAAA...
```

- 1 在 Amazon EC2 控制台中显示时指定安全组的名称，包括 **sg** 前缀。

## 其他资源

- [更改集群网络的 MTU](#)
- [启用 IPsec 加密](#)
- [了解污点和容限](#)

## 13.7. 选择 AWS 本地区域

如果您计划在 AWS Local Zones 中创建子网，您必须单独选择每个 zone group。

### 先决条件

- 已安装 AWS CLI。
- 您已决定要部署 OpenShift Container Platform 集群的 AWS 区域。
- 您已将 permissive IAM 策略附加到选择 zone 组的用户或组帐户。例如，请考虑以下 IAM 策略示例：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:ModifyAvailabilityZoneGroup"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

### 流程

1. 运行以下命令，列出 AWS 区域中可用的区域：

```
$ aws --region "<value_of_AWS_Region>" ec2 describe-availability-zones \
  --query 'AvailabilityZones[].[ZoneName: ZoneName, GroupName: GroupName, Status: OptInStatus]'\
  --filters Name=zone-type,Values=local-zone \
  --all-availability-zones
```

根据 AWS 区域，可用区列表可能比较长。该命令返回以下字段：



**ZoneName**

Local Zone 的名称。

**GroupName**

组成区域的组。要选择区域，请保存名称。

**Status**

Local Zone 组的状态。如果状态不是 **opted-in**，您必须通过运行以下命令在 **GroupName** 中选择。

2. 运行以下命令，选择 AWS 帐户上的 zone 组：

```
$ aws ec2 modify-availability-zone-group \
  --group-name "<value_of_GroupName>" ❶
  --opt-in-status opted-in
```

❶ 对于 **<value\_of\_GroupName>**，指定您要在其中创建子网的 Local Zone 的组名称。例如，指定 **us-east-1-nyc-1** 以使用区域 **us-east-1-nyc-1a** (US East New York)。

## 13.8. AWS 本地区域的集群安装选项

要在 AWS 基础架构上的 AWS Local Zones 环境中安装 OpenShift Container Platform 集群，请选择以下安装选项之一：

- 安装集群来快速将 worker 扩展到边缘计算池，其中安装程序会自动为 OpenShift Container Platform 集群创建资源。
- 在 AWS 上安装集群到现有的 VPC 中，您必须将 Local Zone 子网添加到 **install-config.yaml** 文件中。

### 后续步骤

选择以下选项之一在 AWS Local Zones 环境中安装 OpenShift Container Platform 集群：

- [在 AWS 本地区域中快速安装集群](#)
- [在带有定义的 Local Zone 子网的现有 VPC 上安装集群](#)

## 13.9. 在 AWS 本地区域中快速安装集群

对于 OpenShift Container Platform 4.14，您可以在 Amazon Web Services (AWS) 上快速安装集群，以将计算节点扩展到 Local Zone 位置。通过使用此安装路由，安装程序会为您在配置文件中定义的每个 Local Zone 创建网络资源和 Local Zone 子网。要自定义安装，您必须在部署集群前修改 **install-config.yaml** 文件中的参数。

### 13.9.1. 修改安装配置文件以使用 AWS 本地区域

修改 **install-config.yaml** 文件，使其包含 AWS Local Zones。

#### 先决条件

- 您已配置了 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。
- 您已阅读了在指定安装程序为 OpenShift Container Platform 集群自动创建子网时应用的配置限制。请参阅 AWS Local Zones 中名为 "Cluster limitations" 的部分。
- 您选择每个区的 Local Zone 组。
- 您使用"创建安装配置文件"流程创建了 **install-config.yaml** 文件。

## 流程

1. 通过在边缘计算池的 **platform.aws.zones** 属性中指定 Local Zone 名称来修改 **install-config.yaml** 文件。例如：

```
...
platform:
  aws:
    region: <region_name> ❶
  compute:
  - name: edge
    platform:
      aws:
        zones: ❷
        - <local_zone_name>
#...
```

- ❶ AWS 区域名称。
- ❷ 必须属于同一 AWS 区域的 Local Zone 名称列表。

在 **us-west-2** AWS 区域上安装集群的配置示例，将边缘节点扩展到 **Los Angeles** 和 **Las Vegas** 位置中的 Local Zones。

```
apiVersion: v1
baseDomain: example.com
metadata:
  name: cluster-name
platform:
  aws:
    region: us-west-2
  compute:
  - name: edge
    platform:
      aws:
        zones:
        - us-west-2-lax-1a
        - us-west-2-lax-1b
        - us-west-2-las-1a
  pullSecret: '{"auths": ...}'
  sshKey: 'ssh-ed25519 AAAA...'
#...
```

2. 部署集群。

## 其他资源

- [创建安装配置文件](#)
- [AWS 本地区中的集群限制](#)

## 后续步骤

- [部署集群](#)

## 13.10. 在带有 LOCAL ZONE 子网的现有 VPC 上安装集群

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

在 AWS 上安装集群到现有的 VPC 中，需要使用 AWS Local Zones 将 worker 扩展到 Cloud Infrastructure 边缘。

Local Zone 子网将常规 worker 节点扩展到边缘网络。每个边缘 worker 节点都运行一个用户工作负载。创建 Amazon Web Service (AWS) Local Zone 环境并部署了集群后，您可以使用边缘 worker 节点在 Local Zone 子网中创建用户工作负载。

您可以使用提供的 CloudFormation 模板来创建 VPC 和公共子网。另外，您可以修改模板来自定义模板，或使用其包含的信息根据公司的策略创建 AWS 对象。



### 注意

如果要创建专用子网，您必须修改提供的 CloudFormation 模板或创建自己的模板。

### 13.10.1. 在 AWS 中创建 VPC

您可以在 OpenShift Container Platform 集群的 Amazon Web Services (AWS) 中为每个 Local Zone 位置创建一个 Virtual Private Cloud (VPC) 和子网，以将 worker 节点扩展到边缘位置。您可以进一步自定义 VPC 以满足您的要求，包括 VPN、路由表和添加新的 Local Zone 子网，这些子网没有包括在初始部署中。

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。
- 您可以选择 AWS 帐户上的 AWS 区域区域。

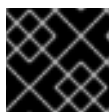
## 流程

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": "3" 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. 运行以下命令，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-1:123456789012:stack/cluster-vpc/dbedae40-2fd3-11eb-820e-12a48460849f
```

4. 运行以下命令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。      |
| <b>PublicRouteTableId</b> | 新公共路由表 ID 的 ID。 |

### 13.10.2. 在 AWS 本地区中创建子网

在为 OpenShift Container Platform 集群配置 worker machineset 之前，您必须在 AWS Local Zones 中创建子网。

您必须为每个要部署 worker 节点的 Local Zone 重复以下流程。

您可以使用提供的 CloudFormation 模板和自定义参数文件来创建代表子网的 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。
- 您已选择 Local Zone 组。

#### 流程

1. 创建一个 JSON 文件，其包含模板所需的参数值：

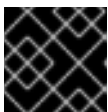
```
[
  {
    "ParameterKey": "VpcId",
```

```

    "ParameterValue": "<value_of_VpcId>" ❶
  },
  {
    "ParameterKey": "PublicRouteTableId",
    "ParameterValue": "<value_of_PublicRouteTableId>" ❷
  },
  {
    "ParameterKey": "ZoneName",
    "ParameterValue": "<value_of_ZoneName>" ❸
  },
  {
    "ParameterKey": "SubnetName",
    "ParameterValue": "<value_of_SubnetName>"
  },
  {
    "ParameterKey": "PublicSubnetCidr",
    "ParameterValue": "10.0.192.0/20" ❹
  }
]

```

- ❶ 指定 VPC ID，它是 VPC 模板输出中的值 **VpcId**。
  - ❷ 指定路由表 ID，这是 VPC 的 CloudFormation 堆栈中的 **PublicRouteTableId** 的值。
  - ❸ 指定 AWS Local Zone 名称，这是您在 "Opting to AWS Local Zones" 部分中检索的 **AvailabilityZones** 对象中的 **ZoneName** 字段的值。
  - ❹ 指定用于创建 Local Zone 子网的 CIDR 块。这个块必须是 VPC CIDR 块 **VpcCidr** 的一部分。
2. 复制本主题的子网的 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. 运行以下命令，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 <subnet_stack_name> \ ❶
  --template-body file://<template>.yaml \ ❷
  --parameters file://<parameters>.json ❸

```

- ❶ **<subnet\_stack\_name>** 是 CloudFormation 堆栈的名称，如 **cluster-lz-<local\_zone\_shortname>**。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-1:123456789012:stack/<subnet_stack_name>/dbedae40-2fd3-11eb-820e-12a48460849f
```

4. 运行以下命令确认模板组件已存在：

```
$ aws cloudformation describe-stacks --stack-name <subnet_stack_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>PublicSubnetIds</b> | 新公共子网的 ID。 |
|------------------------|------------|

### 13.10.3. 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 13.2. 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){1,3}(0-9){1,3}(0-9){1,3}|(0-9){1,3}(0-9){1,3}(0-9){1,3}|(0-9){1,3}(0-9){1,3}(0-9){1,3})$
    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:
  - !Ref VpcId

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
VpcId: !Ref VPC

Outputs:
VpcId:
  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"]]
    ]
PublicRouteTableId:
  Description: Public Route table ID
  Value: !Ref PublicRouteTable

```

#### 13.10.4. AWS 安全组

默认情况下，安装程序会创建安全组并将其附加到 control plane 和计算机器。不可修改与默认安全组关联的规则。

但是，您可以将与现有 VPC 关联的其他现有 AWS 安全组应用到 control plane 和计算机器。应用自定义安全组可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

作为安装过程的一部分，您可以在部署集群前通过修改 **install-config.yaml** 文件来应用自定义安全组。

如需更多信息，请参阅["边缘计算池和 AWS 本地区域"](#)。

#### 13.10.5. 使用 AWS 本地区域的子网的 CloudFormation 模板

您可以使用以下 CloudFormation 模板来部署使用 AWS Local Zones 的 OpenShift Container Platform 集群所需的子网。

**例 13.3. 子网的 CloudFormation 模板**

```

# CloudFormation template used to create Local Zone subnets and dependencies
AWSTemplateFormatVersion: 2010-09-09
Description: Template for create Public Local Zone subnets

Parameters:
  VpcId:
    Description: VPC Id
    Type: String
  ZoneName:
    Description: Local Zone Name (Example us-east-1-nyc-1a)
    Type: String
  SubnetName:
    Description: Local Zone Name (Example cluster-public-us-east-1-nyc-1a)
    Type: String
  PublicRouteTableId:
    Description: Public Route Table ID to associate the Local Zone subnet
    Type: String
  PublicSubnetCidr:
    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: 10.0.128.0/20
    Description: CIDR block for Public Subnet
    Type: String

Resources:
  PublicSubnet:
    Type: "AWS::EC2::Subnet"
    Properties:
      VpcId: !Ref VpcId
      CidrBlock: !Ref PublicSubnetCidr
      AvailabilityZone: !Ref ZoneName
      Tags:
        - Key: Name
          Value: !Ref SubnetName
        - Key: kubernetes.io/cluster/unmanaged
          Value: "true"

  PublicSubnetRouteTableAssociation:
    Type: "AWS::EC2::SubnetRouteTableAssociation"
    Properties:
      SubnetId: !Ref PublicSubnet
      RouteTableId: !Ref PublicRouteTableId

Outputs:
  PublicSubnetIds:
    Description: Subnet IDs of the public subnets.
    Value:
      !Join ["", [!Ref PublicSubnet]]

```

- 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。

### 13.10.6. 修改安装配置文件以使用 AWS 本地区域子网

修改 **install-config.yaml** 文件，使其包含 AWS Local Zones 子网。

#### 先决条件

- 您使用“在 AWS Local Zones 中创建子网”流程创建子网。
- 您使用“创建安装配置文件”流程创建了 **install-config.yaml** 文件。

#### 流程

- 通过在 **platform.aws.subnets** 属性中指定 Local Zone 子网来修改 **install-config.yaml** 配置文件，如下例所示：

```
...
platform:
  aws:
    region: us-west-2
    subnets: ①
    - publicSubnetId-1
    - publicSubnetId-2
    - publicSubnetId-3
    - privateSubnetId-1
    - privateSubnetId-2
    - privateSubnetId-3
    - publicSubnetId-LocalZone-1
...
```

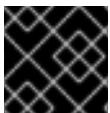
- ① 在可用性和本地区域中创建的子网列表。

#### 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](#)。

## 13.11. 部署集群

您可以在兼容云平台上安装 OpenShift Container Platform。



#### 重要

在初始安装过程中，您只能运行安装程序的 **create cluster** 命令一次。

#### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。
- 已确认主机上的云供应商帐户具有部署集群的正确权限。权限不正确的帐户会导致安装过程失败，并显示包括缺失权限的错误消息。

## 流程

1. 进入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>，请指定自定义 `./install-config.yaml` 文件的位置。
- ❷ 要查看不同的安装详情，请指定 **warn**、**debug** 或 **error**，而不是 **info**。

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。

## 验证

当集群部署成功完成时：

- 终端会显示用于访问集群的说明，包括指向 Web 控制台和 **kubeadmin** 用户的凭证的链接。
- 凭证信息还会输出到 `<installation_directory>/openshift_install.log`。



### 重要

不要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。

## 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: "password"
INFO Time elapsed: 36m22s
```



### 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。一个例外是，您必须手动批准待处理的 **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。

## 后续步骤

- [在 AWS 本地区域中创建用户工作负载](#)

## 13.12. 使用 CLI 登录集群

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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

### 先决条件

- 已部署 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. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

### 输出示例

```
system:admin
```

/validating-an-installation.adoc

## 13.13. 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 控制台的更多详情](#)，请参阅 [访问 Web 控制台](#)。

## 13.14. 验证使用边缘计算池创建的节点

安装使用 AWS Local Zones 的集群后，检查安装时由机器集清单创建的机器状态。

1. 要检查从添加到 **install-config.yaml** 文件中的子网中创建的机器集，请运行以下命令：

```
$ oc get machineset -n openshift-machine-api
```

**输出示例**

| NAME                                | DESIRED | CURRENT | READY | AVAILABLE | AGE  |
|-------------------------------------|---------|---------|-------|-----------|------|
| cluster-7xw5g-edge-us-east-1-nyc-1a | 1       | 1       | 1     | 1         | 3h4m |
| cluster-7xw5g-worker-us-east-1a     | 1       | 1       | 1     | 1         | 3h4m |
| cluster-7xw5g-worker-us-east-1b     | 1       | 1       | 1     | 1         | 3h4m |
| cluster-7xw5g-worker-us-east-1c     | 1       | 1       | 1     | 1         | 3h4m |

2. 要检查从机器集创建的机器，请运行以下命令：

```
$ oc get machines -n openshift-machine-api
```

**输出示例**

| NAME                                             | PHASE   | TYPE        | REGION    | ZONE       | AGE |
|--------------------------------------------------|---------|-------------|-----------|------------|-----|
| cluster-7xw5g-edge-us-east-1-nyc-1a-wbclh-nyc-1a | Running | c5d.2xlarge | us-east-1 | us-east-1- | 3h  |

```

cluster-7xw5g-master-0      Running  m6i.xlarge  us-east-1  us-east-1a  3h4m
cluster-7xw5g-master-1      Running  m6i.xlarge  us-east-1  us-east-1b  3h4m
cluster-7xw5g-master-2      Running  m6i.xlarge  us-east-1  us-east-1c  3h4m
cluster-7xw5g-worker-us-east-1a-rtp45  Running  m6i.xlarge  us-east-1  us-east-1a  3h
cluster-7xw5g-worker-us-east-1b-glm7c  Running  m6i.xlarge  us-east-1  us-east-1b  3h
cluster-7xw5g-worker-us-east-1c-qfvz4  Running  m6i.xlarge  us-east-1  us-east-1c  3h

```

3. 要检查具有边缘角色的节点，请运行以下命令：

```
$ oc get nodes -l node-role.kubernetes.io/edge
```

#### 输出示例

```

NAME                                STATUS  ROLES    AGE  VERSION
ip-10-0-207-188.ec2.internal  Ready  edge,worker  172m  v1.25.2+d2e245f

```

## 13.15. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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

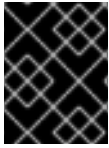
- 有关 Telemetry 服务的更多信息，请参阅[关于远程健康监控](#)。

## 13.16. 后续步骤

- [在 AWS 本地区中创建用户工作负载](#)
- [验证安装](#)
- [自定义集群](#)
- [远程健康报告](#)

## 第 14 章 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 4.14 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。

### 14.1. 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。
- 您可以阅读[选择集群安装方法并为用户准备它](#)的文档。
- 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。

- 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。集群将继续使用您当前的 AWS 凭证为集群的整个生命周期创建 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

如果要配置代理，请务必查看此站点列表。

- 如果环境中无法访问云身份和访问管理(IAM)API，或者不想将管理员级别的凭证 secret 存储在 **kube-system** 命名空间中，您可以 [手动创建和维护 IAM 凭证](#)。

## 14.2. 关于在受限网络中安装

在 OpenShift Container Platform 4.14 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。受限网络安装可以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。

如果您选择在云平台中执行受限网络安装，您仍需要访问其云 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、Nutanix 或 VMware vSphere 上安装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，以镜像 OpenShift 镜像 registry 的内容并包含安装介质。您可以在镜像主机上创建此 registry，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。完成此测试安装后，您可以更轻松地隔离和排除在受限网络中安装过程中可能出现的任何问题。

### 14.2.1. 其他限制

受限网络中的集群有以下额外限制和限制：

- **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。

## 14.3. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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。

## 14.4. 具有用户置备基础架构的集群的要求

对于包含用户置备的基础架构的集群，您必须部署所有所需的机器。

本节论述了在用户置备的基础架构上部署 OpenShift Container Platform 的要求。

### 14.4.1. 集群安装所需的机器

最小的 OpenShift Container Platform 集群需要以下主机：

表 14.1. 最低所需的主机

| 主机                      | 描述                                                                                                |
|-------------------------|---------------------------------------------------------------------------------------------------|
| 一个临时 bootstrap 机器       | 集群需要 bootstrap 机器在三台 control plane 机器上部署 OpenShift Container Platform 集群。您可在安装集群后删除 bootstrap 机器。 |
| 三台 control plane 机器     | control plane 机器运行组成 control plane 的 Kubernetes 和 OpenShift Container Platform 服务。                |
| 至少两台计算机器，也称为 worker 机器。 | OpenShift Container Platform 用户请求的工作负载在计算机器上运行。                                                   |

**重要**

要保持集群的高可用性，请将独立的物理主机用于这些集群机器。

bootstrap 和 control plane 机器必须使用 Red Hat Enterprise Linux CoreOS(RHCOS)作为操作系统。但是，计算机器可以在 Red Hat Enterprise Linux CoreOS(RHCOS)、Red Hat Enterprise Linux(RHEL) 8.6 和更高的版本。

请注意，RHCOS 基于 Red Hat Enterprise Linux(RHEL) 9.2，并继承其所有硬件认证和要求。查看 [红帽企业 Linux 技术功能和限制](#)。

**14.4.2. 集群安装的最低资源要求**

每台集群机器都必须满足以下最低要求：

表 14.2. 最低资源要求

| 机器                   | 操作系统                     | vCPU [1] | 虚拟内存  | Storage | 每秒输入/输出 (IOPS) [2] |
|----------------------|--------------------------|----------|-------|---------|--------------------|
| bootstrap            | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Control plane (控制平面) | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Compute              | RHCOS、RHEL 8.6 及更新版本 [3] | 2        | 8 GB  | 100 GB  | 300                |

1. 当未启用并发多线程 (SMT) 或超线程时，一个 vCPU 相当于一个物理内核。启用后，使用以下公式来计算对应的比例：（每个内核数的线程）× sockets = vCPU。
2. OpenShift Container Platform 和 Kubernetes 对磁盘性能非常敏感，建议使用更快的存储速度，特别是 control plane 节点上需要 10 ms p99 fsync 持续时间的 etcd。请注意，在许多云平台上，存储大小和 IOPS 可一起扩展，因此您可能需要过度分配存储卷来获取足够的性能。

- 与所有用户置备的安装一样，如果您选择在集群中使用 RHEL 计算机，则负责所有操作系统生命周期管理和维护，包括执行系统更新、应用补丁和完成所有其他必要的任务。RHEL 7 计算机器的使用已弃用，并已在 OpenShift Container Platform 4.10 及更新的版本中删除。



## 注意

从 OpenShift Container Platform 版本 4.13 开始，RHCOS 基于 RHEL 版本 9.2，它更新了微架构要求。以下列表包含每个架构需要的最小指令集架构 (ISA)：

- x86-64 体系结构需要 x86-64-v2 ISA
- ARM64 架构需要 ARMv8.0-A ISA
- IBM Power 架构需要 Power 9 ISA
- s390x 架构需要 z14 ISA

如需更多信息，请参阅 [RHEL 架构](#)。

如果平台的实例类型满足集群机器的最低要求，则 OpenShift Container Platform 支持使用它。

## 其他资源

- [优化存储](#)

### 14.4.3. 为 AWS 测试的实例类型

以下 Amazon Web Services(AWS) 实例类型已经过 OpenShift Container Platform 测试。



## 注意

将以下图中包含的机器类型用于 AWS 实例。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 14.1. 基于 64 位 x86 架构的机器类型

- c4.\*
- c5.\*
- c5a.\*
- i3.\*
- m4.\*
- m5.\*
- m5a.\*
- m6i.\*
- r4.\*

- r5.\*
- r5a.\*
- r6i.\*
- t3.\*
- t3a.\*

#### 14.4.4. 在 64 位 ARM 基础架构上为 AWS 测试过的实例类型

OpenShift Container Platform 中已经测试了以下 Amazon Web Services (AWS) 64 位 ARM 实例类型。



#### 注意

使用 AWS ARM 实例的以下图中包含的机器类型。如果您使用没有在图中列出的实例类型，请确保使用的实例大小与集群安装“最小资源要求”中列出的最少资源要求匹配。

#### 例 14.2. 基于 64 位 ARM 架构的机器类型

- c6g.\*
- m6g.\*
- r8g.\*

#### 14.4.5. 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。

### 14.5. 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。

#### 14.5.1. 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中工作，则无法访问 EC2、ELB 和 S3 端点的公共 IP 地址。根据您要在安装过程中限制互联网流量的级别，有以下配置选项：

##### 14.5.1.1. 选项 1：创建 VPC 端点

创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

通过这个选项，网络流量在 VPC 和所需的 AWS 服务之间保持私有。

##### 14.5.1.2. 选项 2：创建一个没有 VPC 端点的代理

作为安装过程的一部分，您可以配置 HTTP 或 HTTPS 代理。使用此选项时，互联网流量会通过代理访问所需的 AWS 服务。

##### 14.5.1.3. 选项 3：创建带有 VPC 端点的代理

作为安装过程的一部分，您可以使用 VPC 端点配置 HTTP 或 HTTPS 代理。创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

在 `install-config.yaml` 文件中配置代理时，将这些端点添加到 `noProxy` 字段。通过这个选项，代理会阻止集群直接访问互联网。但是，您的 VPC 和所需的 AWS 服务之间网络流量保持私有。



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      |
|        |                                                                                                                                                                                                                                                                                                                                       |                                                                                                                     |             |

| 组件   | AWS 类型                                                                                                                                                               | 描述                                                                                |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| 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（如果安装程序置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。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。             |
| 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。             |

| 组件           | AWS 类型                                           | 描述                        |
|--------------|--------------------------------------------------|---------------------------|
| 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::SecurityGroup</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::SecurityGroup</b> | <b>icmp</b> | <b>0</b>     |
|                            |                                |             |              |

| 组                      | 类型                      | IP 协议 | 端口范围  |
|------------------------|-------------------------|-------|-------|
|                        |                         | tcp   | 22    |
| BootstrapSecurityGroup | AWS::EC2::SecurityGroup | 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          |
|------------------------------------|------------------------------------|-------|---------------|
| MasterIngressEtc                   | etcd                               | tcp   | 2379- 2380    |
| MasterIngressVxlan                 | Vxlan 数据包                          | udp   | 4789          |
| MasterIngressWorkerVxlan           | Vxlan 数据包                          | udp   | 4789          |
| MasterIngressInternal              | 内部集群通信和 Kubernetes 代理指标            | tcp   | 9000 - 9999   |
| MasterIngressWorkerInternal        | 内部集群通信                             | tcp   | 9000 - 9999   |
| MasterIngressKube                  | kubernetes kubelet、调度程序 and 控制器管理器 | tcp   | 10250 - 10259 |
| MasterIngressWorkerKube            | kubernetes kubelet、调度程序 and 控制器管理器 | tcp   | 10250 - 10259 |
| MasterIngressIngressServices       | Kubernetes 入口服务                    | tcp   | 30000 - 32767 |
| MasterIngressWorkerIngressServices | Kubernetes 入口服务                    | tcp   | 30000 - 32767 |
| MasterIngressGeneve                | Geneve 包                           | udp   | 6081          |

| 入口组                                          | 描述              | IP 协议      | 端口范围                 |
|----------------------------------------------|-----------------|------------|----------------------|
| <b>MasterIngressWorkerGeneve</b>             | Geneve 包        | <b>udp</b> | <b>6081</b>          |
| <b>MasterIngressIpsecIke</b>                 | IPsec IKE 数据包   | <b>udp</b> | <b>500</b>           |
| <b>MasterIngressWorkerIpsecIke</b>           | IPsec IKE 数据包   | <b>udp</b> | <b>500</b>           |
| <b>MasterIngressIpsecNat</b>                 | IPsec NAT-T 数据包 | <b>udp</b> | <b>4500</b>          |
| <b>MasterIngressWorkerIpsecNat</b>           | IPsec NAT-T 数据包 | <b>udp</b> | <b>4500</b>          |
| <b>MasterIngressIpsecEsp</b>                 | IPsec ESP 数据包   | <b>50</b>  | <b>All</b>           |
| <b>MasterIngressWorkerIpsecEsp</b>           | IPsec ESP 数据包   | <b>50</b>  | <b>All</b>           |
| <b>MasterIngressInternalUDP</b>              | 内部集群通信          | <b>udp</b> | <b>9000 - 9999</b>   |
| <b>MasterIngressWorkerInternalUDP</b>        | 内部集群通信          | <b>udp</b> | <b>9000 - 9999</b>   |
| <b>MasterIngressIngressServicesUDP</b>       | Kubernetes 入口服务 | <b>udp</b> | <b>30000 - 32767</b> |
| <b>MasterIngressWorkerIngressServicesUDP</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</b>        | 内部集群通信                             | <b>tcp</b> | <b>9000 - 9999</b>   |
| <b>WorkerIngress Kube</b>                  | Kubernetes kubelet、调度程序 and 控制器管理器 | <b>tcp</b> | <b>10250</b>         |
| <b>WorkerIngress WorkerKube</b>            | Kubernetes kubelet、调度程序 and 控制器管理器 | <b>tcp</b> | <b>10250</b>         |
| <b>WorkerIngress IngressServices</b>       | Kubernetes 入口服务                    | <b>tcp</b> | <b>30000 - 32767</b> |
| <b>WorkerIngress WorkerIngressServices</b> | Kubernetes 入口服务                    | <b>tcp</b> | <b>30000 - 32767</b> |
| <b>WorkerIngress Geneve</b>                | Geneve 包                           | <b>udp</b> | <b>6081</b>          |
| <b>WorkerIngress MasterGeneve</b>          | Geneve 包                           | <b>udp</b> | <b>6081</b>          |
| <b>WorkerIngress IpsecIke</b>              | IPsec IKE 数据包                      | <b>udp</b> | <b>500</b>           |
| <b>WorkerIngress MasterIpsecIke</b>        | IPsec IKE 数据包                      | <b>udp</b> | <b>500</b>           |
| <b>WorkerIngress IpsecNat</b>              | IPsec NAT-T 数据包                    | <b>udp</b> | <b>4500</b>          |
| <b>WorkerIngress MasterIpsecNat</b>        | IPsec NAT-T 数据包                    | <b>udp</b> | <b>4500</b>          |

| 入口组                                   | 描述              | IP 协议 | 端口范围          |
|---------------------------------------|-----------------|-------|---------------|
| WorkerIngressIpsecEsp                 | IPsec ESP 数据包   | 50    | All           |
| WorkerIngressMasterIpsecEsp           | IPsec ESP 数据包   | 50    | All           |
| WorkerIngressInternalUDP              | 内部集群通信          | udp   | 9000 - 9999   |
| WorkerIngressMasterInternalUDP        | 内部集群通信          | udp   | 9000 - 9999   |
| WorkerIngressIngressServicesUDP       | Kubernetes 入口服务 | udp   | 30000 - 32767 |
| WorkerIngressMasterIngressServicesUDP | 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*          | *  |

| 角色        | 影响           | 操作                      | 资源 |
|-----------|--------------|-------------------------|----|
| bootstrap | <b>Allow</b> | <b>ec2:Describe*</b>    | *  |
|           | <b>Allow</b> | <b>ec2:AttachVolume</b> | *  |
|           | <b>Allow</b> | <b>ec2:DetachVolume</b> | *  |

### 14.5.2. 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 control plane 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）。这些机器不受计算机器集的管控。

### 14.5.3. IAM 用户所需的 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 14.3. 安装所需的 EC2 权限

- **ec2:AttachNetworkInterface**
- **ec2:AuthorizeSecurityGroupEgress**
- **ec2:AuthorizeSecurityGroupIngress**
- **ec2:CopyImage**
- **ec2>CreateNetworkInterface**
- **ec2>CreateSecurityGroup**
- **ec2:CreateTags**
- **ec2>CreateVolume**
- **ec2>DeleteSecurityGroup**
- **ec2>DeleteSnapshot**

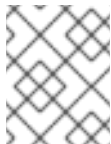


- **ec2:DeleteTags**
- **ec2:DeregisterImage**
- **ec2:DescribeAccountAttributes**
- **ec2:DescribeAddresses**
- **ec2:DescribeAvailabilityZones**
- **ec2:DescribeDhcpOptions**
- **ec2:DescribeImages**
- **ec2:DescribeInstanceAttribute**
- **ec2:DescribeInstanceCreditSpecifications**
- **ec2:DescribeInstances**
- **ec2:DescribeInstanceTypes**
- **ec2:DescribeInternetGateways**
- **ec2:DescribeKeyPairs**
- **ec2:DescribeNatGateways**
- **ec2:DescribeNetworkAcls**
- **ec2:DescribeNetworkInterfaces**
- **ec2:DescribePrefixLists**
- **ec2:DescribeRegions**
- **ec2:DescribeRouteTables**
- **ec2:DescribeSecurityGroupRules**
- **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:RevokeSecurityGroupEgress**
- **ec2:RevokeSecurityGroupIngress**
- **ec2:RunInstances**
- **ec2:TerminateInstances**

#### 例 14.4. 安装过程中创建网络资源所需的权限

- **ec2:AllocateAddress**
- **ec2:AssociateAddress**
- **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 Virtual Private Cloud (VPC)，您的帐户不需要这些权限来创建网络资源。

## 例 14.5. 安装所需的 Elastic Load Balancing 权限(ELB)

- `elasticloadbalancing:AddTags`
- `elasticloadbalancing:ApplySecurityGroupsToLoadBalancer`
- `elasticloadbalancing:AttachLoadBalancerToSubnets`
- `elasticloadbalancing:ConfigureHealthCheck`
- `elasticloadbalancing>CreateListener`
- `elasticloadbalancing>CreateLoadBalancer`
- `elasticloadbalancing>CreateLoadBalancerListeners`
- `elasticloadbalancing>CreateTargetGroup`
- `elasticloadbalancing>DeleteLoadBalancer`
- `elasticloadbalancing:DeregisterInstancesFromLoadBalancer`
- `elasticloadbalancing:DeregisterTargets`
- `elasticloadbalancing:DescribeInstanceHealth`
- `elasticloadbalancing:DescribeListeners`
- `elasticloadbalancing:DescribeLoadBalancerAttributes`
- `elasticloadbalancing:DescribeLoadBalancers`
- `elasticloadbalancing:DescribeTags`
- `elasticloadbalancing:DescribeTargetGroupAttributes`
- `elasticloadbalancing:DescribeTargetHealth`
- `elasticloadbalancing:ModifyLoadBalancerAttributes`
- `elasticloadbalancing:ModifyTargetGroup`
- `elasticloadbalancing:ModifyTargetGroupAttributes`
- `elasticloadbalancing:RegisterInstancesWithLoadBalancer`
- `elasticloadbalancing:RegisterTargets`
- `elasticloadbalancing:SetLoadBalancerPoliciesOfListener`

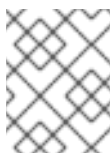


### 重要

OpenShift Container Platform 使用 ELB 和 ELBv2 API 服务来置备负载均衡器。权限列表显示这两个服务所需的权限。AWS web 控制台中存在一个已知问题，其中这两个服务都使用相同的 `elasticloadbalancing` 操作前缀，但无法识别相同的操作。您可以忽略有关服务没有识别某些 `elasticloadbalancing` 操作的警告。

#### 例 14.6. 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 帐户中创建负载均衡器，IAM 用户还需要 **iam:CreateServiceLinkedRole** 权限。

#### 例 14.7. 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 14.8. 安装所需的 Amazon Simple Storage Service (S3) 权限

- `s3:CreateBucket`
- `s3:DeleteBucket`
- `s3:GetAccelerateConfiguration`
- `s3:GetBucketAcl`
- `s3:GetBucketCors`
- `s3:GetBucketLocation`
- `s3:GetBucketLogging`
- `s3:GetBucketObjectLockConfiguration`
- `s3:GetBucketPolicy`
- `s3:GetBucketRequestPayment`
- `s3:GetBucketTagging`
- `s3:GetBucketVersioning`
- `s3:GetBucketWebsite`
- `s3:GetEncryptionConfiguration`
- `s3:GetLifecycleConfiguration`
- `s3:GetReplicationConfiguration`
- `s3:ListBucket`
- `s3:PutBucketAcl`
- `s3:PutBucketTagging`

- **s3:PutEncryptionConfiguration**

#### 例 14.9. 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 14.10. 删除基本集群资源所需的权限

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

#### 例 14.11. 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:ReleaseAddress**
- **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。您的帐户只需要有 **tag:UntagResources** 权限就能删除网络资源。

#### 例 14.12. 使用自定义密钥管理服务 (KMS) 密钥安装集群的可选权限

- **kms:CreateGrant**
- **kms:Decrypt**
- **kms:DescribeKey**
- **kms:Encrypt**
- **kms:GenerateDataKey**
- **kms:GenerateDataKeyWithoutPlainText**
- **kms:ListGrants**
- **kms:RevokeGrant**

#### 例 14.13. 使用共享实例角色删除集群所需的权限

- **iam:UntagRole**

#### 例 14.14. 创建清单所需的额外 IAM 和 S3 权限

- **iam:GetUserPolicy**
- **iam:ListAccessKeys**
- **iam:PutUserPolicy**
- **iam:TagUser**
- **s3:AbortMultipartUpload**
- **s3:GetBucketPublicAccessBlock**
- **s3:ListBucket**
- **s3:ListBucketMultipartUploads**
- **s3:PutBucketPublicAccessBlock**
- **s3:PutLifecycleConfiguration**



#### 注意

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

#### 例 14.15. 实例的可选权限和安装配额检查

- **ec2:DescribeInstanceTypeOfferings**
- **servicequotas:ListAWSDefaultServiceQuotas**

#### 例 14.16. 在共享 VPC 上安装集群时集群所有者帐户的可选权限

- **sts:AssumeRole**

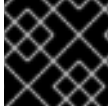
## 14.6. 为集群节点 SSH 访问生成密钥对

在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 **core** 用户的 **~/.ssh/authorized\_keys** 列表中，这将启用免密码身份验证。

将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。



如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。`./openshift-install gather` 命令还需要在集群节点上设置 SSH 公钥。



### 重要

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。



### 注意

您必须使用本地密钥，而不是使用特定平台方法配置 [的密钥](#)，如 [AWS 密钥对](#)。

## 流程

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，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_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。



### 注意

如果您计划在 **x86\_64**、**ppc64le** 和 **s390x** 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 **ed25519** 算法的密钥。相反，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

3. 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。



### 注意

在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- a. 如果 **ssh-agent** 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

## 输出示例

Agent pid 31874



### 注意

如果集群处于 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。

4. 将 SSH 私钥添加到 **ssh-agent**：

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

- 1 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

### 输出示例

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

### 后续步骤

- 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，则必须为安装程序提供密钥。

## 14.7. 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。您还可以选择在安装准备阶段首先设置独立 **var** 分区。

### 14.7.1. 可选：创建独立 **/var** 分区

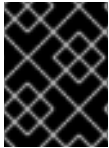
建议安装程序将 OpenShift Container Platform 的磁盘分区保留给安装程序。然而，在有些情况下您可能需要在文件系统的一部分中创建独立分区。

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

- **/var/lib/containers**：保存随着系统中添加更多镜像和容器而增长的容器相关内容。
- **/var/lib/etcd**：保存您可能希望独立保留的数据，比如 etcd 存储的性能优化。
- **/var**：保存您可能希望独立保留的数据，以满足审计等目的。

通过单独存储 **/var** 目录的内容，可以更轻松地根据需要为区域扩展存储，并在以后重新安装 OpenShift Container Platform，并保持该数据的完整性。使用这个方法，您不必再次拉取所有容器，在更新系统时也不必复制大量日志文件。

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** 准备阶段插入一个创建的机器配置清单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**，以在 **manifest** 和 **openshift** 子目录中创建一组文件。在系统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. 创建用于配置额外分区的 Butane 配置。例如，将文件命名为 **\$HOME/clusterconfig/98-var-partition.bu**，将磁盘设备名称改为 **worker** 系统上存储设备的名称，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：

```
variant: openshift
version: 4.14.0
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 98-var-partition
storage:
  disks:
    - device: /dev/disk/by-id/<device_name> ❶
      partitions:
        - label: var
          start_mib: <partition_start_offset> ❷
          size_mib: <partition_size> ❸
```

```

    number: 5
  filesystems:
    - device: /dev/disk/by-partlabel/var
      path: /var
      format: xfs
      mount_options: [defaults, prjquota] 4
      with_mount_unit: true

```

- 1 要分区的磁盘的存储设备名称。
- 2 在引导磁盘中添加数据分区时，推荐最少使用 25000 MiB(Mebibytes)。root 文件系统会自动调整大小以填充所有可用空间（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 以兆字节为单位的数据分区大小。
- 4 对于用于容器存储的文件系统，必须启用 **prjquota** 挂载选项。



### 注意

当创建单独的 **/var** 分区时，如果不同的实例类型没有相同的设备名称，则无法为 worker 节点使用不同的实例类型。

5. 从 Butane 配置创建一个清单，并将它保存到 **clusterconfig/openshift** 目录中。例如，运行以下命令：

```
$ butane $HOME/clusterconfig/98-var-partition.bu -o $HOME/clusterconfig/openshift/98-var-partition.yaml
```

6. 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。

## 14.7.2. 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. 进入包含安装程序的目录并运行以下命令：

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

**1** 对于 `<installation_directory>`，请指定要存储安装程序创建的文件目录名称。



#### 重要

指定一个空目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。

#### b. 在提示符处，提供云的配置详情：

##### i. 可选：选择用于访问集群机器的 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额外信息。

#### 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>`，请指定本地 registry 名称。对于 `<credentials>`，请指定提供本地 registry 凭证的凭据。

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 **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。证书文件可以是现有的可信证书颁发机构，也可以是您为镜像 registry 生成的自签名证书。

[illegible]

- 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
```

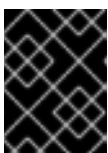
使用命令输出中的 **imageContentSources** 部分来镜像（mirror）仓库，或您从您进入受限网络的介质中的内容时使用的值。

- 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](#)。

### 14.7.3. 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，并确定它们中的任何站点是否需要绕过代理。默认情况下，所有集群出口流量都经过代理，包括对托管云供应商 API 的调用。如果需要，您将在 **Proxy** 对象的 **spec.noProxy** 字段中添加站点来绕过代理。



### 注意

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

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

### 流程

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: ec2.<aws_region>.amazonaws.com,elasticloadbalancing.
    <aws_region>.amazonaws.com,s3.<aws_region>.amazonaws.com 3
  additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

- 1 用于创建集群外 HTTP 连接的代理 URL。URL 方案必须是 **http**。
- 2 用于创建集群外 HTTPS 连接的代理 URL。
- 3 要从代理中排除的目标域名、IP 地址或其他网络 CIDR 的逗号分隔列表。在域前面加上 **.** 以仅匹配子域。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。如果您已将 Amazon **EC2**、**Elastic Load Balancing** 和 **S3** VPC 端点添加到 VPC 中，您必须将这些端点添加到 **noProxy** 字段。
- 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，其包含代理 HTTPS 连接所需的一个或多个额外 CA 证书。然后，Cluster Network Operator 会创建 **trusted-ca-bundle** 配置映射，将这些内容与 Red Hat Enterprise Linux CoreOS (RHCOS) 信任捆绑包合并，**Proxy** 对象的 **trustedCA** 字段中也会引用此配置映射。**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。
- 5 可选：决定 **Proxy** 对象的配置以引用 **trustedCA** 字段中 **user-ca-bundle** 配置映射的策略。允许的值是 **Proxyonly** 和 **Always**。仅在配置了 **http/https** 代理时，使用 **Proxyonly** 引用 **user-ca-bundle** 配置映射。使用 **Always** 始终引用 **user-ca-bundle** 配置映射。默认值为 **Proxyonly**。



**注意**

安装程序不支持代理的 **readinessEndpoints** 字段。

**注意**

如果安装程序超时，重启并使用安装程序的 **wait-for** 命令完成部署。例如：

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

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。

#### 14.7.4. 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手动启动集群机器，因此您必须生成 Kubernetes 清单和 Ignition 配置文件来配置机器。

安装配置文件转换为 Kubernetes 清单。清单嵌套到 Ignition 配置文件中，稍后用于配置集群机器。

**重要**

- OpenShift Container Platform 安装程序生成的 Ignition 配置文件包含 24 小时后过期的证书，然后在该时进行续订。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。一个例外是，您必须手动批准待处理的 **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，这些文件位于您的镜像主机上。
- 已创建 **install-config.yaml** 安装配置文件。

#### 流程

1. 进入包含 OpenShift Container Platform 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>**，请指定包含您创建的 **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
```

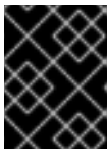
通过删除这些文件，您可以防止集群自动生成 control plane 机器。

3. 删除定义 control plane 机器集的 Kubernetes 清单文件：

```
$ rm -f <installation_directory>/openshift/99_openshift-machine-api_master-control-plane-machine-set.yaml
```

4. 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
```



### 重要

如果在用户置备的基础架构上安装集群时禁用了 **MachineAPI** 功能，则必须删除定义 worker 机器的 Kubernetes 清单文件。否则，集群将无法安装。

由于您要自行创建和管理 worker 机器，因此不需要初始化这些机器。

5. 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。
6. 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。

如果这样做，您必须在后续步骤中手动添加入口 DNS 记录。

7. 要创建 Ignition 配置文件，请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。

为安装目录中的 bootstrap、control plane 和计算节点创建 Ignition 配置文件。**kubeadmin-password** 和 **kubeconfig** 文件在 **./<installation\_directory>/auth** 目录中创建：

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

## 其他资源

- [手动创建长期凭证](#)

## 14.8. 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

- 已获取 OpenShift Container Platform 安装程序和集群的 pull secret。
- 已为集群生成 Ignition 配置文件。
- 已安装 **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 ❶
```

- ❶ 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 ❶
```

- ❶ 此命令的输出是您的集群名称和随机字符串。

## 14.9. 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。

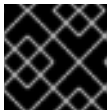
### 流程

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> 1
--template-body file://<template>.yaml 2
--parameters file://<parameters>.json 3
```

- 1 **<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。
- 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-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。 |

14.9.1. 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 14.17. 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:
    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
      VpcId: !Ref VPC
Outputs:
  VpcId:
    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"]]
    ]
PublicRouteTableId:
  Description: Public Route table ID
  Value: !Ref PublicRouteTable

```

## 14.10. 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", ❶
    "ParameterValue": "mycluster" ❷
  },
  {
    "ParameterKey": "InfrastructureName", ❸
    "ParameterValue": "mycluster-<random_string>" ❹
  },
  {
    "ParameterKey": "HostedZoneId", ❺
    "ParameterValue": "<random_string>" ❻
  },
  {
    "ParameterKey": "HostedZoneName", ❼
    "ParameterValue": "example.com" ❽
  },
  {
    "ParameterKey": "PublicSubnets", ❾
    "ParameterValue": "subnet-<random_string>" ❿
  },
  {
    "ParameterKey": "PrivateSubnets", 11
    "ParameterValue": "subnet-<random_string>" 12
  },
  {
    "ParameterKey": "VpcId", 13
    "ParameterValue": "vpc-<random_string>" 14
  }
]
```

❶ 一个简短的、代表集群的名称用于主机名等。

❷ 指定您为集群生成 **install-config.yaml** 文件时所用的集群名称。

❸ 您的 Ignition 配置文件中为集群编码的集群基础架构名称。

❹ 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>**。

❺ 用来注册目标的 Route 53 公共区 ID。

❻ 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。

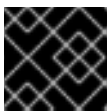
- 7 用来注册目标的 Route 53 区。
  - 8 指定您为集群生成 **install-config.yaml** 文件时所用的 Route 53 基域。请勿包含 AWS 控制台中显示的结尾句点 (.)。
  - 9 为 VPC 创建的公共子网。
  - 10 指定 VPC 的 CloudFormation 模板输出的 **PublicSubnetIds** 值。
  - 11 为 VPC 创建的专用子网。
  - 12 指定 VPC 的 CloudFormation 模板输出的 **PrivateSubnetIds** 值。
  - 13 为集群创建的 VPC。
  - 14 指定 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

如果要部署到 AWS 政府或 `secret` 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> 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-dns**。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** 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>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。                        |

14.10.1. 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 14.18. 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&lt;AWS::EC2::Subnet::Id&gt;

**PrivateSubnets:**

Description: The internal subnets.

Type: List&lt;AWS::EC2::Subnet::Id&gt;

**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
- HostedZoneld
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"
  HostedZoneld:
    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 VpcId
        VPCRegion: !Ref "AWS::Region"

```

```

ExternalApiServerRecord:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    Comment: Alias record for the API server
    HostedZoneld: !Ref HostedZoneld

```

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`

RegisterTargetLambdalaRole:

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:
      - "RegisterTargetLambdaramRole"
      - "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: "python3.8"
        Timeout: 120

RegisterSubnetTagsLambdaramRole:
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: "python3.8"
    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]]  
 RegisterNlbTargetsLambda:  
 Description: Lambda ARN useful to help register or deregister IP targets for these load balancers.  
 Value: !GetAtt RegisterNlbTargets.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

如果要集群部署到 AWS 政府或 `secret` 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](#)。

## 14.11. 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。

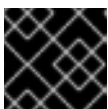
## 流程

1. 创建一个 JSON 文件，其包含模板所需的参数值：

```
[
  {
    "ParameterKey": "InfrastructureName", ❶
    "ParameterValue": "mycluster-<random_string>" ❷
  },
  {
    "ParameterKey": "VpcCidr", ❸
    "ParameterValue": "10.0.0.0/16" ❹
  },
  {
    "ParameterKey": "PrivateSubnets", ❺
    "ParameterValue": "subnet-<random_string>" ❻
  },
  {
    "ParameterKey": "VpcId", ❼
    "ParameterValue": "vpc-<random_string>" ❽
  }
]
```

- 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 |

#### 14.11.1. 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 14.19. 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])(\V(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
- 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

MasterIngressIpsecIke:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: IPsec IKE packets  
FromPort: 500  
ToPort: 500  
IpProtocol: udp

MasterIngressIpsecNat:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: IPsec NAT-T packets  
FromPort: 4500  
ToPort: 4500  
IpProtocol: udp

MasterIngressIpsecEsp:

Type: AWS::EC2::SecurityGroupIngress  
Properties:  
GroupId: !GetAtt MasterSecurityGroup.GroupId  
SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId  
Description: IPsec ESP packets  
IpProtocol: 50

MasterIngressWorkerIpsecIke:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: IPsec IKE packets

FromPort: 500

ToPort: 500

IpProtocol: udp

MasterIngressWorkerIpsecNat:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: IPsec NAT-T packets

FromPort: 4500

ToPort: 4500

IpProtocol: udp

MasterIngressWorkerIpsecEsp:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt MasterSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: IPsec ESP packets

IpProtocol: 50

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

#### WorkerIngressIpsecIke:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: IPsec IKE packets

FromPort: 500

ToPort: 500

IpProtocol: udp

WorkerIngressIpsecNat:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: IPsec NAT-T packets

FromPort: 4500

ToPort: 4500

IpProtocol: udp

WorkerIngressIpsecEsp:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt WorkerSecurityGroup.GroupId

Description: IPsec ESP packets

IpProtocol: 50

WorkerIngressMasterIpsecIke:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: IPsec IKE packets

FromPort: 500

ToPort: 500

IpProtocol: udp

WorkerIngressMasterIpsecNat:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: IPsec NAT-T packets

FromPort: 4500

ToPort: 4500

IpProtocol: udp

WorkerIngressMasterIpsecEsp:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt WorkerSecurityGroup.GroupId

SourceSecurityGroupId: !GetAtt MasterSecurityGroup.GroupId

Description: IPsec ESP packets

IpProtocol: 50

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

```

## 14.12. 使用流元数据访问 RHCOS AMI

在 OpenShift Container Platform 中，流元数据以 JSON 格式提供与 RHCOS 相关的标准化元数据，并将元数据注入集群中。流元数据是一种稳定的格式，支持多种架构，旨在自我记录以维护自动化。

您可以使用 **openshift-install** 的 **coreos print-stream-json** 子命令访问流元数据格式的引导镜像的信息。此命令提供了一种以可脚本、机器可读格式打印流元数据的方法。

对于用户置备的安装，**openshift-install** 二进制文件包含对经过测试用于 OpenShift Container Platform 的 RHCOS 引导镜像版本的引用，如 AWS AMI。

### 流程

要解析流元数据，请使用以下方法之一：

- 在 Go 程序中使用位于 <https://github.com/coreos/stream-metadata-go> 的正式 **stream-metadata-go** 库。您还可以查看库中的示例代码。
- 在 Python 或 Ruby 等其他编程语言中使用您首选编程语言的 JSON 库。
- 在处理 JSON 数据的命令行工具中，如 **jq**：
  - 为 AWS 区域输出当前的 **x86\_64** 或 **aarch64** AMI，如 **us-west-1**：

对于 **x86\_64**

```
$ openshift-install coreos print-stream-json | jq -r
'.architectures.x86_64.images.aws.regions["us-west-1"].image'
```

输出示例

```
ami-0d3e625f84626bbda
```

对于 **aarch64**

```
$ openshift-install coreos print-stream-json | jq -r
'.architectures.aarch64.images.aws.regions["us-west-1"].image'
```

输出示例

```
ami-0af1d3b7fa5be2131
```

这个命令的输出是您指定的架构和 **us-west-1** 区域的 AWS AMI ID。AMI 必须与集群属于同一区域。

### 14.13. AWS 基础架构的 RHCOS AMI

红帽提供了对可手动为 OpenShift Container Platform 节点指定的各种 AWS 区域和实例架构有效的 Red Hat Enterprise Linux CoreOS(RHCOS)AMI。



**注意**

通过导入您自己的 AMI，您还可以安装到没有公布的 RHCOS AMI 的区域。

表 14.3. x86\_64 RHCOS AMIs

| AWS 区          | AWS AMI               |
|----------------|-----------------------|
| af-south-1     | ami-01860370941726bdd |
| ap-east-1      | ami-05bc702cdaf7e4251 |
| ap-northeast-1 | ami-098932fd93c15690d |

| AWS 区域         | AWS AMI               |
|----------------|-----------------------|
| ap-northeast-2 | ami-006f4e02d97910a36 |
| ap-northeast-3 | ami-0c4bd5b1724f82273 |
| ap-south-1     | ami-0cbf22b638724853d |
| ap-south-2     | ami-031f4d165f4b429c4 |
| ap-southeast-1 | ami-0dc3e381a731ab9fc |
| ap-southeast-2 | ami-032ae8d0f287a66a6 |
| ap-southeast-3 | ami-0393130e034b86423 |
| ap-southeast-4 | ami-0b38f776bded7d7d7 |
| ca-central-1   | ami-058ea81b3a1d17edd |
| eu-central-1   | ami-011010debd974a250 |
| eu-central-2   | ami-0623b105ae811a5e2 |
| eu-north-1     | ami-0c4bb9ce04f3526d4 |
| eu-south-1     | ami-06c29eccd3d74df52 |
| eu-south-2     | ami-00e0b5f3181a3f98b |
| eu-west-1      | ami-087bfa513dc600676 |
| eu-west-2      | ami-0ebad59c0e9554473 |
| eu-west-3      | ami-074e63b65eaf83f96 |
| me-central-1   | ami-0179d6ae1d908ace9 |
| me-south-1     | ami-0b60c75273d3efcd7 |
| sa-east-1      | ami-0913cbfbfa9a7a53c |
| us-east-1      | ami-0f71dcd99e6a1cd53 |
| us-east-2      | ami-0545fae7edbbbf061 |

| AWS 区         | AWS AMI               |
|---------------|-----------------------|
| us-gov-east-1 | ami-081eabdc478e501e5 |
| us-gov-west-1 | ami-076102c394767f319 |
| us-west-1     | ami-0609e4436c4ae5eff |
| us-west-2     | ami-0c5d3e03c0ab9b19a |

表 14.4. aarch64 RHCOS AMI

| AWS 区          | AWS AMI               |
|----------------|-----------------------|
| af-south-1     | ami-08dd66a61a2caa326 |
| ap-east-1      | ami-0232cd715f8168c34 |
| ap-northeast-1 | ami-0bc0b17618da96700 |
| ap-northeast-2 | ami-0ee505fb62eed2fd6 |
| ap-northeast-3 | ami-0462cd2c3b7044c77 |
| ap-south-1     | ami-0e0b4d951b43adc58 |
| ap-south-2     | ami-06d457b151cc0e407 |
| ap-southeast-1 | ami-0874e1640dfc15f17 |
| ap-southeast-2 | ami-05f215734ceb0f5ad |
| ap-southeast-3 | ami-073030df265c88b3f |
| ap-southeast-4 | ami-043f4c40a6fc3238a |
| ca-central-1   | ami-0840622f99a32f586 |
| eu-central-1   | ami-09a5e6ebe667ae6b5 |
| eu-central-2   | ami-0835cb1bf387e609a |
| eu-north-1     | ami-069ddbda521a10a27 |
| eu-south-1     | ami-09c5cc21026032b4c |

| AWS 区         | AWS AMI               |
|---------------|-----------------------|
| eu-south-2    | ami-0c36ab2a8bbeed045 |
| eu-west-1     | ami-0d2723c8228cb2df3 |
| eu-west-2     | ami-0abd66103d069f9a8 |
| eu-west-3     | ami-08c7249d59239fc5c |
| me-central-1  | ami-0685f33ebb18445a2 |
| me-south-1    | ami-0466941f4e5c56fe6 |
| sa-east-1     | ami-08cdc0c8a972f4763 |
| us-east-1     | ami-0d461970173c4332d |
| us-east-2     | ami-0e9cdc0e85e0a6aeb |
| us-gov-east-1 | ami-0b896df727672ce09 |
| us-gov-west-1 | ami-0b896df727672ce09 |
| us-west-1     | ami-027b7cc5f4c74e6c1 |
| us-west-2     | ami-0b189d89b44bdfbf2 |

## 14.14. 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。您可以按照以下方法：

- 为集群提供 **bootstrap.ign** Ignition 配置文件的位置。此文件位于您的安装目录中。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。
- 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. 运行以下命令来创建存储桶：

```
$ aws s3 mb s3://<cluster-name>-infra ❶
```

- ❶ **<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。

如果需要，您必须为 S3 存储桶使用预签名 URL，而不是 **s3://** 模式：

- 部署到具有与 AWS SDK 不同端点的区域。
- 部署代理。
- 提供您自己的自定义端点。

2. 运行以下命令，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 ❶
```

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

3. 运行以下命令验证文件是否已上传：

```
$ aws s3 ls s3://<cluster-name>-infra/
```

## 输出示例

```
2019-04-03 16:15:16 314878 bootstrap.ign
```



## 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。

4. 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": "AllowedBootstrapSshCidr", ❺
    "ParameterValue": "0.0.0.0/0" ❻
  },
  {
    "ParameterKey": "PublicSubnet", ❼
    "ParameterValue": "subnet-<random_string>" ❽
  },
  {
    "ParameterKey": "MasterSecurityGroup", ❾
    "ParameterValue": "sg-<random_string>" ❿
  },
  {
    "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:<aws_region>:<account_number>:function:
<dns_stack_name>-RegisterNlbTargets-<random_string>" 18
  },
  {
    "ParameterKey": "ExternalApiTargetGroupArn", 19
    "ParameterValue": "arn:aws:elasticloadbalancing:<aws_region>:
<account_number>:targetgroup/<dns_stack_name>-Exter-<random_string>" 20
  },
  {
    "ParameterKey": "InternalApiTargetGroupArn", 21
    "ParameterValue": "arn:aws:elasticloadbalancing:<aws_region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<random_string>" 22
  },
  {
    "ParameterKey": "InternalServiceTargetGroupArn", 23
    "ParameterValue": "arn:aws:elasticloadbalancing:<aws_region>:"
```



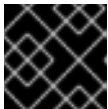
```

    <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 根据您的架构，当前 Red Hat Enterprise Linux CoreOS (RHCOS) AMI 用于 bootstrap 节点。
- 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`。
5. 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。
6. 可选：如果要使用代理部署集群，您必须更新模板中的 `ignition` 以添加 `ignition.config.proxy` 字段。另外，如果您已将 Amazon EC2、Elastic Load Balancing 和 S3 VPC 端点添加到 VPC 中，您必须将这些端点添加到 `noProxy` 字段。
7. 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
```

8. 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。 |

**Bootstrap  
PrivatelP**

bootstrap 节点专用 IP 地址。

**14.14.1. 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 14.20. 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

RegisterNLBpTargetsLambdaArn:

```

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
BootstrapInstanceType:
Description: Instance type for the bootstrap EC2 instance
Default: "i3.large"
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
  - RegisterNlbIpTargetsLambdaArn
  - 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: !Ref BootstrapInstanceType
  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](#)。

## 14.15. 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": "RhcosAmi", 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": "" 26
},
{
  "ParameterKey": "AutoRegisterELB", 27
  "ParameterValue": "yes" 28

```



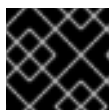
```

    },
    {
      "ParameterKey": "RegisterNlbTargetsLambdaArn", 29
      "ParameterValue": "arn:aws:lambda:<aws_region>:<account_number>:function:
<dns_stack_name>-RegisterNlbTargets-<random_string>" 30
    },
    {
      "ParameterKey": "ExternalApiTargetGroupArn", 31
      "ParameterValue": "arn:aws:elasticloadbalancing:<aws_region>:
<account_number>:targetgroup/<dns_stack_name>-Exter-<random_string>" 32
    },
    {
      "ParameterKey": "InternalApiTargetGroupArn", 33
      "ParameterValue": "arn:aws:elasticloadbalancing:<aws_region>:
<account_number>:targetgroup/<dns_stack_name>-Inter-<random_string>" 34
    },
    {
      "ParameterKey": "InternalServiceTargetGroupArn", 35
      "ParameterValue": "arn:aws:elasticloadbalancing:<aws_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 根据您的架构，当前用于 control plane 机器的 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 根据您的架构，用于 control plane 机器的 AWS 实例类型。
  - 26 实例类型值与 control plane 机器的最低资源要求对应。例如，**m6i.xlarge** 是 AMD64 的类型，**m6g.xlarge** 是 ARM64 的类型。
  - 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>
```

## 14.15.1. 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 14.21. 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: ""
    Description: unused
    Type: String
  PrivateHostedZoneId:
    Default: ""
```

Description: unused  
 Type: String  
 PrivateHostedZoneName:  
 Default: ""  
 Description: unused  
 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  
  
 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: "Load Balancer Automation"

Parameters:

- AutoRegisterELB

- RegisterNLbpTargetsLambdaArn

- 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"

AutoRegisterELB:  
default: "Use Provided ELB Automation"

Conditions:

DoRegistration: !Equals ["yes", !Ref AutoRegisterELB]

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:

- AssociatePublicIpAddress: "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,
        }
      - {
        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:
      - AssociatePublicIp: "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

```

```

Outputs:
  PrivateIPs:

```



Description: The control-plane node private IP addresses.

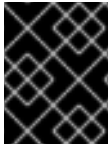
Value:

```
!Join [
  "",
  [!GetAtt Master0.PrivateIp, !GetAtt Master1.PrivateIp, !GetAtt Master2.PrivateIp]
]
```

## 14.16. 在 AWS 中创建 WORKER 节点

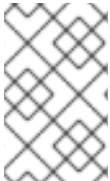
您可以在 Amazon Web Services (AWS) 中创建 worker 节点，供集群使用。

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", ❶
    "ParameterValue": "mycluster-<random_string>" ❷
  },
  {
    "ParameterKey": "Rhc Ami", ❸
```

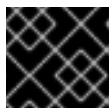
```

    "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": "" 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 根据您的选择的架构，当前 Red Hat Enterprise Linux CoreOS (RHCOS) AMI 用于 worker 节点。
- 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 根据您选择的架构，用于计算机器的 AWS 实例类型。
  - 16 实例类型值对应于计算机器的最低资源要求。例如，**m6i.large** 是 AMD64 的类型，**m6g.large** 是 ARM64 的类型。
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. 可选：如果将 **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. 可选：如果您使用 AWS Marketplace 镜像部署，请使用从订阅获取的 AMI ID 更新 **Worker0.type.properties.ImageID** 参数。
  5. 使用 CloudFormation 模板创建代表 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
```



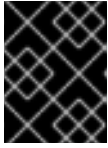
### 注意

CloudFormation 模板会创建一个堆栈，它代表一个 worker 节点。

6. 确认模板组件已存在：

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

- 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。

## 14.16.1. 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 14.22. 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 worker nodes into.
    Type: AWS::EC2::Subnet::Id
  WorkerSecurityGroupId:
    Description: The worker security group ID to associate with worker 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 worker nodes.
    Type: String
  WorkerInstanceType:
    Default: m5.large
    Type: String

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:
      - AssociatePublicIpAddress: "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

## 14.17. 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**，而不是 **info**。

### 输出示例

```

INFO Waiting up to 20m0s for the Kubernetes API at

```

```
https://api.mycluster.example.com:6443...
INFO API v1.27.3 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](#)。

## 14.18. 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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

### 先决条件

- 已部署 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. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

### 输出示例

```
system:admin
```

## 14.19. 批准机器的证书签名请求

当您添加机器到集群时，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。必须首先批准客户端请求，然后批准服务器请求。

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 | v1.27.3 |
| master-1 | Ready  | master | 63m | v1.27.3 |
| master-2 | Ready  | master | 64m | v1.27.3 |

输出中列出了您创建的所有机器。



注意

在有些 CSR 被批准前，前面的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-bootstrap | Pending   |
| csr-8vnps | 15m | system:serviceaccount:openshift-machine-config-operator:node-bootstrap | Pending   |
| ...       |     |                                                                        |           |

在本例中，两台机器加入集群。您可能会在列表中看到更多已批准的 CSR。

3. 如果 CSR 没有获得批准，在您添加的机器的所有待处理 CSR 都处于 **Pending** 状态 后，请批准集群机器的 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它们，证书将会轮转，每个节点会存在多个证书。您必须批准所有这些证书。批准客户端 CSR 后，Kubelet 为服务证书创建一个二级 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，则后续提供证书续订请求由 **machine-approver** 自动批准。





### 注意

对于在未启用机器 API 的平台上运行的集群，如裸机和其他用户置备的基础架构，您必须实施一种方法来自动批准 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。该方法必须监视新的 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。

- 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：

- 要单独批准，请对每个有效的 CSR 运行以下命令：

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

**1** **<csr\_name>** 是当前 CSR 列表中 CSR 的名称。

- 要批准所有待处理的 CSR，请运行以下命令：

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

6. 批准所有客户端和服务端 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 | v1.27.3 |
| master-1 | Ready  | master | 73m | v1.27.3 |
| master-2 | Ready  | master | 74m | v1.27.3 |
| worker-0 | Ready  | worker | 11m | v1.27.3 |
| worker-1 | Ready  | worker | 11m | v1.27.3 |



注意

批准服务端 CSR 后可能需要几分钟时间让机器过渡到 **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](#)。

14.20. 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

- 观察集群组件上线：

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

输出示例

| NAME                    | VERSION | AVAILABLE | PROGRESSING | DEGRADED | SINCE |
|-------------------------|---------|-----------|-------------|----------|-------|
| authentication          | 4.14.0  | True      | False       | False    | 19m   |
| baremetal               | 4.14.0  | True      | False       | False    | 37m   |
| cloud-credential        | 4.14.0  | True      | False       | False    | 40m   |
| cluster-autoscaler      | 4.14.0  | True      | False       | False    | 37m   |
| config-operator         | 4.14.0  | True      | False       | False    | 38m   |
| console                 | 4.14.0  | True      | False       | False    | 26m   |
| csi-snapshot-controller | 4.14.0  | True      | False       | False    | 37m   |
| dns                     | 4.14.0  | True      | False       | False    | 37m   |
| etcd                    | 4.14.0  | True      | False       | False    | 36m   |
| image-registry          | 4.14.0  | True      | False       | False    | 31m   |
| ingress                 | 4.14.0  | True      | False       | False    | 30m   |
| insights                | 4.14.0  | True      | False       | False    | 31m   |
| kube-apiserver          | 4.14.0  | True      | False       | False    | 26m   |
| kube-controller-manager | 4.14.0  | True      | False       | False    | 36m   |

|                                          |        |      |       |       |     |
|------------------------------------------|--------|------|-------|-------|-----|
| kube-scheduler                           | 4.14.0 | True | False | False | 36m |
| kube-storage-version-migrator            | 4.14.0 | True | False | False | 37m |
| machine-api                              | 4.14.0 | True | False | False | 29m |
| machine-approver                         | 4.14.0 | True | False | False | 37m |
| machine-config                           | 4.14.0 | True | False | False | 36m |
| marketplace                              | 4.14.0 | True | False | False | 37m |
| monitoring                               | 4.14.0 | True | False | False | 29m |
| network                                  | 4.14.0 | True | False | False | 38m |
| node-tuning                              | 4.14.0 | True | False | False | 37m |
| openshift-apiserver                      | 4.14.0 | True | False | False | 32m |
| openshift-controller-manager             | 4.14.0 | True | False | False | 30m |
| openshift-samples                        | 4.14.0 | True | False | False | 32m |
| operator-lifecycle-manager               | 4.14.0 | True | False | False | 37m |
| operator-lifecycle-manager-catalog       | 4.14.0 | True | False | False | 37m |
| operator-lifecycle-manager-packageserver | 4.14.0 | True | False | False | 32m |
| service-ca                               | 4.14.0 | True | False | False | 38m |
| storage                                  | 4.14.0 | True | False | False | 37m |

## 2. 配置不可用的 Operator。

### 14.20.1. 禁用默认的 OperatorHub 目录源

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** → **Configuration** → **OperatorHub** 页面中，点 **Sources** 选项卡，您可以在其中创建、更新、删除、禁用和启用单独的源。

### 14.20.2. 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。

显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，这仅适用于非生产集群。

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。

#### 14.20.2.1. 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](#)。

### 14.20.2.2. 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 时会丢失所有镜像。

## 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这个命令，**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。

## 14.21. 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。

## 14.22. 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>
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            |                                      |                            |

### 3. 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
```

- 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。

### 4. 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' ❷
--output text
```

- ❶ ❷ 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**。

5. 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 '{ ❶
> "Changes": [
> {
>   "Action": "CREATE",
>   "ResourceRecordSet": {
>     "Name": "\\052.apps.<cluster_domain>", ❷
>     "Type": "A",
>     "AliasTarget": {
>       "HostedZoneId": "<hosted_zone_id>", ❸
>       "DNSName": "<external_ip>.", ❹
>       "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 '{ ❶
> "Changes": [
> {
>   "Action": "CREATE",
>   "ResourceRecordSet": {
>     "Name": "\\052.apps.<cluster_domain>", ❷
>     "Type": "A",
>     "AliasTarget": {
```

```
> "HostedZoneId": "<hosted_zone_id>", 3
> "DNSName": "<external_ip>.", 4
> "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 (.)。

## 14.23. 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：

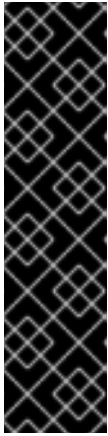
```
$ ./openshift-install --dir <installation_directory> wait-for install-complete 1
```

- 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: "password"
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。一个例外是，您必须手动批准待处理的 **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。

2. 在 [Cluster registration](#) 页面注册您的集群。

/validating-an-installation.adoc

## 14.24. 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](#) 的更多详情，请参阅 [访问 Web 控制台](#)。

## 14.25. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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](#)

## 14.26. 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](#)。

## 14.27. 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](#)。
- [在受限网络中使用 Operator Lifecycle Manager \(OLM\)](#)
- [配置额外的信任存储](#)
- [远程健康报告](#)
- [注册断开连接的集群](#)
- [删除云供应商凭证](#)

## 第 15 章 在带有 AWS OUTPOSTS 中的远程 WORKER 的 AWS 上安装集群

在 OpenShift Container Platform 版本 4.14 中，您可以使用 AWS Outposts 中运行的远程 worker 在 Amazon Web Services (AWS) 上安装集群。这可以通过自定义默认 AWS 安装并执行一些手动步骤来实现。



### 重要

在 AWS Outposts 上使用远程 worker 在 AWS 上安装集群只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅[技术预览功能支持范围](#)。

有关 AWS Outposts 的更多信息，请参阅[AWS Outposts 文档](#)。



### 重要

要在 AWS Outposts 中使用远程 worker 安装集群，所有 worker 实例都必须位于同一 Outpost 实例中，且不能位于 AWS 区域。集群无法在 AWS Outposts 和 AWS 区域中具有实例。另外，它还必须遵循 control plane 节点不能可以调度。

### 15.1. 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。
- 您可以阅读[选择集群安装方法并为用户准备它的文档](#)。
- 已将 [AWS 帐户配置](#) 为托管集群。
- 您对实例类型熟悉您使用的 AWS Outpost 实例。这可以通过 [get-outpost-instance-types AWS CLI 命令](#) 进行验证
- 熟悉 AWS Outpost 实例详情，如 OutpostArn 和 AvailabilityZone。这可以通过 [list-outposts AWS CLI 命令](#) 进行验证



### 重要

由于集群使用提供的 AWS 凭证为其整个生命周期创建 AWS 资源，所以凭证必须基于密钥且长期。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。有关生成适当密钥的更多信息，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。

- 您可以访问 Amazon Web Services (AWS) 中的现有 Amazon Virtual Private Cloud (VPC)。如需更多信息，请参阅[“使用自定义 VPC”一节](#)。
- 如果使用防火墙，则会 [将其配置为允许集群需要访问的站点](#)。

### 15.2. 关于使用自定义 VPC

OpenShift Container Platform 4.14 安装程序无法在 AWS Outposts 上自动部署 AWS 子网，因此您需要手动配置 VPC。因此，您必须在 Amazon Web Services (AWS) 的现有 Amazon Virtual Private Cloud (VPC) 中将集群部署到现有子网中。另外，您可以通过将 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。

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。

### 15.2.1. 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。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](#) 以及 [处理 VPC 和子网](#)。

安装程序无法：

- 分割网络范围供集群使用。
- 设置子网的路由表。
- 设置 VPC 选项，如 DHCP。

您必须在安装集群前完成这些任务。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 [您的 VPC 的路由表](#)。

您的 VPC 必须满足以下特征：



#### 注意

要允许在 AWS Outposts 中使用远程 worker 创建 OpenShift Container Platform，您必须在 AWS Outpost 实例中创建一个专用子网，以便创建工作负载实例，并在 AWS 区域中的一个专用子网来创建 control plane 实例。如果您在区域中指定多个专用子网，则 control plane 实例将在这些子网中分发。您还需要在用于专用子网的每个可用区中创建公共子网，包括 Outpost 专用子网，因为网络 Load Balancers 将在 API 服务器和 Ingress 网络的 AWS 区域中创建，作为集群安装的一部分。可以在与 Outpost 专用子网相同的可用区中创建 AWS 区域专用子网。

- 在 AWS 区域中为 control plane 使用的每个可用区创建一个公共和私有子网。每个可用区都不包含 AWS 区域中的一个公共子网。有关此类配置的示例，请参阅 AWS 文档中的[具有公共和私有子网\(NAT\)的 VPC](#)。

要在 AWS Outposts 中创建专用子网，首先需要确保 Outpost 实例位于所需的可用区中。然后，您可以通过添加 Outpost ARN，在 Outpost 实例中在该可用区中创建专用子网。确保在同一可用区中创建的 AWS 区域中存在另一个公共子网。

记录每个子网 ID。完成安装要求您在 AWS 区域（在 `install-config.yaml` 文件的 `platform` 部分中）输入所有子网 ID，并将 worker `machineset` 更改为使用 Outpost 中创建的专用子网 ID。请参阅 AWS 文档中的[查找子网 ID](#)。



### 重要

如果您需要在 AWS Outposts 中创建公共子网，请验证此子网没有用于网络或经典 LoadBalancer，否则 LoadBalancer 创建会失败。要达到此目的，`kubernetes.io/cluster/.*-outposts: owned` 的特殊标签必须包含在子网中。

- 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 必须附加一个公共互联网网关。对于每个可用区：
  - 公共子网需要路由到互联网网关。
  - 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。



### 注意

要通过本地网络访问本地集群，VPC 必须与您的 Outpost 的本地网关路由表关联。如需更多信息，请参阅 AWS Outposts 用户指南中的[VPC 关联](#)。

- VPC 不能使用 `kubernetes.io/cluster/.*: owned, Name`，和 `openshift.io/cluster` 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。您不能使用 `Name` 标签，因为它与 EC2 `Name` 字段重叠，且安装失败。
- 您需要在您的 VPC 中启用 `enableDnsSupport` 和 `enableDnsHostnames` 属性，以便集群可以使用附加到 VPC 中的 Route 53 区来解析集群内部的 DNS 记录。请参阅 AWS 文档中的[您的 VPC 中的 DNS 支持](#)部分。  
如果要使用您自己的 Route 53 托管私有区，您必须在安装集群前将现有托管区与 VPC 相关联。您可以使用 `install-config.yaml` 文件中的 `platform.aws.hostedZone` 和 `platform.aws.hostedZoneRole` 字段定义托管区。您可以通过与安装集群的帐户共享来使用来自另一个帐户的私有托管区。如果使用另一个帐户的私有托管区，则必须使用 `Passthrough` 或 `Manual` 凭证模式。

#### 15.2.1.1. 选项 1：创建 VPC 端点

创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- `ec2.<aws_region>.amazonaws.com`
- `elasticloadbalancing.<aws_region>.amazonaws.com`

- **s3.<aws\_region>.amazonaws.com**

通过这个选项，网络流量在 VPC 和所需的 AWS 服务之间保持私有。

15.2.1.2. 选项 2：创建一个没有 VPC 端点的代理

作为安装过程的一部分，您可以配置 HTTP 或 HTTPS 代理。使用此选项时，互联网流量会通过代理访问所需的 AWS 服务。

15.2.1.3. 选项 3：创建带有 VPC 端点的代理

作为安装过程的一部分，您可以使用 VPC 端点配置 HTTP 或 HTTPS 代理。创建 VPC 端点，并将其附加到集群使用的子网。将端点命名为如下：

- **ec2.<aws\_region>.amazonaws.com**
- **elasticloadbalancing.<aws\_region>.amazonaws.com**
- **s3.<aws\_region>.amazonaws.com**

在 `install-config.yaml` 文件中配置代理时，将这些端点添加到 `noProxy` 字段。通过这个选项，代理会阻止集群直接访问互联网。但是，您的 VPC 和所需的 AWS 服务之间网络流量保持私有。

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：                                                                                                                                          |             |
|        |                                                                                                                                                                                                                                                                                                                                       | 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。要启用在 Outpost 中运行的远程 worker，VPC 必须包含位于 Outpost 实例的专用子网，以及位于对应 AWS 区域的专用子网。如果您使用专用子网，必须为其提供适当的路由和表。 |             |

### 15.2.2. 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。每个可用区都仅包含一个公共子网，以及 AWS 区域中的一个专用子网（未在 Outpost 实例中创建）。安装 Outpost 实例的可用区应该在 Outpost 实例中包含一个外部专用子网。
- 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。

### 15.2.3. 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：不同的个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。

### 15.2.4. 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)。

### 15.2.5. AWS 安全组

默认情况下，安装程序会创建安全组并将其附加到 control plane 和计算机器。不可修改与默认安全组关联的规则。

但是，您可以将与现有 VPC 关联的其他现有 AWS 安全组应用到 control plane 和计算机器。应用自定义安全组可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

作为安装过程的一部分，您可以在部署集群前通过修改 **install-config.yaml** 文件来应用自定义安全组。

如需更多信息，请参阅“将现有 AWS 安全组应用到集群”。

## 15.3. OPENSIFT CONTAINER PLATFORM 互联网访问

在 OpenShift Container Platform 4.14 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](https://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型的基础架构上执行受限网络安装。在此过程中，您可以下载所需的内容，并使用它为镜像 registry 填充安装软件包。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。在更新集群前，您要更新镜像 registry 的内容。

## 15.4. 为集群节点 SSH 访问生成密钥对

在 OpenShift Container Platform 安装过程中，您可以为安装程序提供 SSH 公钥。密钥通过它们的 Ignition 配置文件传递给 Red Hat Enterprise Linux CoreOS(RHCOS)节点，用于验证对节点的 SSH 访问。密钥添加到每个节点上 **core** 用户的 `~/.ssh/authorized_keys` 列表中，这将启用免密码身份验证。

将密钥传递给节点后，您可以使用密钥对作为用户 **核心** 通过 SSH 连接到 RHCOS 节点。若要通过 SSH 访问节点，必须由 SSH 为您的本地用户管理私钥身份。

如果要通过 SSH 连接到集群节点来执行安装调试或灾难恢复，则必须在安装过程中提供 SSH 公钥。`./openshift-install gather` 命令还需要在集群节点上设置 SSH 公钥。



### 重要

不要在生产环境中跳过这个过程，在生产环境中需要灾难恢复和调试。



### 注意

您必须使用本地密钥，而不是使用特定平台方法配置 [的密钥](#)，如 [AWS 密钥对](#)。

### 流程

1. 如果您在本地计算机上没有可用于在集群节点上进行身份验证的现有 SSH 密钥对，请创建一个。例如，在使用 Linux 操作系统的计算机上运行以下命令：

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

- ❶ 指定新 SSH 密钥的路径和文件名，如 `~/.ssh/id_ed25519`。如果您已有密钥对，请确保您的公钥位于 `~/.ssh` 目录中。



### 注意

如果您计划在 **x86\_64**、**ppc64le** 和 **s390x** 架构上安装使用 RHEL 加密库（这些加密库已提交给 NIST 用于 FIPS 140-2/140-3 验证）的 OpenShift Container Platform 集群，则不要创建使用 **ed25519** 算法的密钥。相反，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

2. 查看公共 SSH 密钥：

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

例如，运行以下命令来查看 `~/.ssh/id_ed25519.pub` 公钥：

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

- 将 SSH 私钥身份添加到本地用户的 SSH 代理（如果尚未添加）。在集群节点上，或者要使用 `./openshift-install gather` 命令，需要对该密钥进行 SSH 代理管理，才能在集群节点上进行免密码 SSH 身份验证。



#### 注意

在某些发行版中，自动管理默认 SSH 私钥身份，如 `~/.ssh/id_rsa` 和 `~/.ssh/id_dsa`。

- 如果 `ssh-agent` 进程尚未为您的本地用户运行，请将其作为后台任务启动：

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

#### 输出示例

```
Agent pid 31874
```



#### 注意

如果集群处于 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。

- 将 SSH 私钥添加到 `ssh-agent`：

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

- ❶ 指定 SSH 私钥的路径和文件名，如 `~/.ssh/id_ed25519.pub`

#### 输出示例

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

#### 后续步骤

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

## 15.5. 获取安装程序

在安装 OpenShift Container Platform 前，将安装文件下载到您用于安装的主机上。

#### 先决条件

- 您有一台运行 Linux 或 macOS 的计算机，至少有 1.2 GB 本地磁盘空间。

#### 流程

1. 进入 Red Hat Hybrid Cloud Console 上的 [Cluster Type](#) 页。如果您有红帽帐户，请使用您的凭证登录。如果没有，请创建一个帐户。
2. 在页的 **Run it yourself** 部分中选择您的基础架构供应商。
3. 从 **OpenShift 安装程序** 下的下拉菜单中选择您的主机操作系统和架构，然后点**下载安装程序**。
4. 将下载的文件保存在要存储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几个文件。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。删除集群需要这两个文件。
- 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，请为特定云供应商完成 OpenShift Container Platform 卸载流程。

5. 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
```

6. 从 [Red Hat OpenShift Cluster Manager](#) 下载安装 [pull secret](#)。此 pull secret 允许您与所含授权机构提供的服务进行身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

### 提示

另外，您还可以从[红帽客户门户网站](#)检索安装程序，您可以在其中指定要下载的安装程序版本。但是，您需要有一个有效的订阅才能访问此页。

## 15.6. 集群安装的最低资源要求

每台集群机器都必须满足以下最低要求：

表 15.1. 最低资源要求

| 机器                   | 操作系统                     | vCPU [1] | 虚拟内存  | Storage | 每秒输入/输出 (IOPS) [2] |
|----------------------|--------------------------|----------|-------|---------|--------------------|
| bootstrap            | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Control plane (控制平面) | RHCOS                    | 4        | 16 GB | 100 GB  | 300                |
| Compute              | RHCOS、RHEL 8.6 及更新版本 [3] | 2        | 8 GB  | 100 GB  | 300                |

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

2. OpenShift Container Platform 和 Kubernetes 对磁盘性能非常敏感，建议使用更快的存储速度，特别是 control plane 节点上需要 10 ms p99 fsync 持续时间的 etcd。请注意，在许多云平台上，存储大小和 IOPS 可一起扩展，因此您可能需要过度分配存储卷来获取足够的性能。
3. 与所有用户置备的安装一样，如果您选择在集群中使用 RHEL 计算机，则负责所有操作系统生命周期管理和维护，包括执行系统更新、应用补丁和完成所有其他必要的任务。RHEL 7 计算机器的使用已弃用，并已在 OpenShift Container Platform 4.10 及更新的版本中删除。



### 注意

从 OpenShift Container Platform 版本 4.13 开始，RHCOS 基于 RHEL 版本 9.2，它更新了微架构要求。以下列表包含每个架构需要的最小指令集架构 (ISA)：

- x86-64 体系结构需要 x86-64-v2 ISA
- ARM64 架构需要 ARMv8.0-A ISA
- IBM Power 架构需要 Power 9 ISA
- s390x 架构需要 z14 ISA

如需更多信息，请参阅 [RHEL 架构](#)。

如果平台的实例类型满足集群机器的最低要求，则 OpenShift Container Platform 支持使用它。

### 其他资源

- [优化存储](#)

## 15.7. 识别 AWS OUTPOSTS 实例类型

AWS Outposts rack 目录包括支持最新 Intel 电源 EC2 实例类型具有或不使用本地实例存储的选项。确定 AWS Outpost 实例中配置了哪些实例类型。作为安装过程的一部分，您必须使用安装程序用于部署 worker 节点的实例类型更新 `install-config.yaml` 文件。

### 流程

运行以下命令，使用 AWS CLI 获取支持的实例类型列表：

```
$ aws outposts get-outpost-instance-types --outpost-id <outpost_id> 1
```

- 1** 对于 `<outpost_id>`，请指定 worker 实例的 AWS 帐户中使用的 Outpost ID



### 重要

当您为 AWS Outpost 实例购买容量时，您可以指定每个服务器提供的 EC2 容量布局。每台服务器支持单类实例类型。布局可以提供单个实例类型或多个实例类型。专用主机允许您更改您为初始布局选择的任何内容。如果您分配主机以支持整个容量的单个实例类型，则只能从该主机启动单个实例类型。

AWS Outposts 中支持的实例类型可能会改变。如需更多信息，您可以检查 AWS Outposts 文档中的 [Compute](#) 和 [Storage](#) 页面。

## 15.8. 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

- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。

### 流程

#### 1. 创建 `install-config.yaml` 文件。

##### a. 进入包含安装程序的目录并运行以下命令：

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

**1** 对于 `<installation_directory>`，请指定要存储安装程序创建的文件的目录名称。

在指定目录时：

- 验证该目录是否具有执行权限。在安装目录中运行 Terraform 二进制文件需要这个权限。
- 使用空目录。有些安装资产，如 bootstrap X.509 证书的过期间隔较短，因此不得重复使用安装目录。如果要重复使用另一个集群安装中的单个文件，您可以将它们复制到您的目录中。但是，安装资产的文件名可能会在发行版本间有所变化。从以前的 OpenShift Container Platform 版本中复制安装文件时请小心。



### 注意

始终删除 `~/.powervs` 目录，以避免重复使用过时的配置。运行以下命令：

```
$ rm -rf ~/.powervs
```

##### b. 在提示符处，提供云的配置详情：

##### i. 可选：选择用于访问集群机器的 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. 为集群输入描述性名称。

## 2. 修改 `install-config.yaml` 文件。AWS Outposts 安装有以下限制，需要手动修改 `install-config.yaml` 文件：

- 与提供接近规模的 AWS 区域不同，AWS Outposts 被其置备的容量、EC2 系列和生成、配置的实例大小以及尚未被其他工作负载消耗的计算容量的限制。因此，在创建新的 OpenShift Container Platform 集群时，您需要在配置文件的 `compute.platform.aws.type` 部分中提供支持的实例类型。
- 当使用在 AWS Outposts 中运行的远程 worker 部署 OpenShift Container Platform 集群时，只有一个可用区可用于计算实例 - 创建 Outpost 实例的可用区。因此，在创建新的 OpenShift Container Platform 集群时，建议您在配置文件中的 `compute.platform.aws.zones` 部分中提供相关的可用区，以便将计算实例限制到此可用区。
- AWS Outposts 服务不支持 Amazon Elastic Block Store (EBS) gp3 卷。此卷类型是 OpenShift Container Platform 集群使用的默认类型。因此，在创建新的 OpenShift Container Platform 集群时，必须将 `compute.platform.aws.rootVolume.type` 部分中的卷类型改为 gp2。您将找到有关如何更改这些值的更多信息。

## 3. 备份 `install-config.yaml` 文件，以便您可以使用它安装多个集群。



### 重要

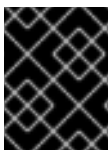
`install-config.yaml` 文件会在安装过程中消耗掉。如果要重复使用此文件，必须现在备份。

## 其他资源

- [AWS 的安装配置参数](#)

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

您可以自定义安装配置文件 (`install-config.yaml`)，以指定有关 OpenShift Container Platform 集群平台的更多详细信息，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，并进行修改。

```
apiVersion: v1
baseDomain: example.com 1
credentialsMode: Mint 2
controlPlane: 3 4
  hyperthreading: Enabled 5
  name: master
  platform: {}
  replicas: 3
compute: 6
- hyperthreading: Enabled 7
  name: worker
  platform:
    aws:
```



```

    type: m5.large 8
    zones:
      - us-east-1a 9
    rootVolume:
      type: gp2 10
      size: 120
    replicas: 3
  metadata:
    name: test-cluster 11
  networking:
    clusterNetwork:
      - cidr: 10.128.0.0/14
        hostPrefix: 23
    machineNetwork:
      - cidr: 10.0.0.0/16
    networkType: OVNKubernetes 12
    serviceNetwork:
      - 172.30.0.0/16
  platform:
    aws:
      region: us-west-2 13
      propagateUserTags: true 14
      userTags:
        adminContact: jdoe
        costCenter: 7536
    subnets: 15
      - subnet-1
      - subnet-2
      - subnet-3
  sshKey: ssh-ed25519 AAAA... 16
  pullSecret: '{"auths": ...}' 17

```

**1 11 13 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。默认情况下，CCO 使用 **kube-system** 命名空间中的 root 凭证来动态尝试决定凭证的功能。有关 CCO 模式的详情，请参阅 [身份验证和授权指南](#) 中的 "About the Cloud Credential Operator" 部分。

**3 6 14** 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。仅使用一个 control plane 池。

**5 7** 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置为 **Disabled** 来禁用它。如果在某些集群机器中禁用并发多线程，则必须在所有集群机器中禁用它。



### 重要

如果您禁用并发多线程，请确保您的容量规划考虑机器性能显著降低的情况。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**。

- 8 对于在 AWS Outpost 实例中运行的计算实例，在 AWS Outpost 实例中指定受支持的实例类型。
- 9 对于在 AWS Outpost 实例中运行的计算实例，请指定 Outpost 实例所在的可用区。
- 10 对于在 AWS Outpost 实例中运行的计算实例，请指定卷类型 gp2，以避免使用不支持的 gp3 卷类型。
- 12 要安装的集群网络插件。支持的值有 **OVNKubernetes** 和 **OpenShiftSDN**。默认值为 **OVNKubernetes**。
- 15 如果您提供自己的 VPC，为集群使用的每个可用区指定子网。
- 16 您可选择提供用于访问集群中机器的 **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。

## 15.8.2. 将现有 AWS 安全组应用到集群

将现有 AWS 安全组应用到 control plane 和计算机器可帮助您满足机构的安全需求，在这种情况下，您需要控制这些机器的传入或传出流量。

### 先决条件

- 您已在 AWS 中创建安全组。如需更多信息，请参阅使用 [安全组的 AWS 文档](#)。
- 安全组必须与您要部署集群的现有 VPC 关联。安全组不能与另一个 VPC 关联。
- 您有一个现有的 **install-config.yaml** 文件。

### 流程

1. 在 **install-config.yaml** 文件中，编辑 **compute.platform.aws.additionalSecurityGroupIDs** 参数，为您的计算机器指定一个或多个自定义安全组。
2. 编辑 **controlPlane.platform.aws.additionalSecurityGroupIDs** 参数，为您的 control plane 机器指定一个或多个自定义安全组。
3. 保存文件并在部署集群时引用。

### 指定自定义安全组的 **install-config.yaml** 文件示例

```
# ...
compute:
- hyperthreading: Enabled
  name: worker
  platform:
    aws:
      additionalSecurityGroupIDs:
        - sg-1 1
        - sg-2
```



```

replicas: 3
controlPlane:
  hyperthreading: Enabled
  name: master
  platform:
    aws:
      additionalSecurityGroupIDs:
        - sg-3
        - sg-4
replicas: 3
platform:
  aws:
    region: us-east-1
    subnets: 2
    - subnet-1
    - subnet-2
    - subnet-3

```

1 在 Amazon EC2 控制台中显示时指定安全组的名称，包括 **sg** 前缀。

2 指定集群使用的每个可用区的子网。

## 15.9. 生成清单文件

使用安装程序在 `asset` 目录中生成一组清单文件。需要清单文件来指定用于 worker 机器的 AWS Outposts 子网，并指定网络供应商所需的设置。

如果您计划重复使用 **install-config.yaml** 文件，请在生成清单文件前创建一个备份文件。

### 流程

1. 可选：创建 **install-config.yaml** 文件的备份副本：

```
$ cp install-config.yaml install-config.yaml.backup
```

2. 在资产目录中生成一组清单：

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

此命令显示以下消息：

### 输出示例

```

INFO Consuming Install Config from target directory
INFO Manifests created in: <installation_directory>/manifests and
<installation_directory>/openshift

```

该命令生成以下清单文件：

### 输出示例

```
$ tree
.
```

```

├── manifests
│   ├── cluster-config.yaml
│   ├── cluster-dns-02-config.yml
│   ├── cluster-infrastructure-02-config.yml
│   ├── cluster-ingress-02-config.yml
│   ├── cluster-network-01-crd.yml
│   ├── cluster-network-02-config.yml
│   ├── cluster-proxy-01-config.yaml
│   ├── cluster-scheduler-02-config.yml
│   ├── cvo-overrides.yaml
│   ├── kube-cloud-config.yaml
│   ├── kube-system-configmap-root-ca.yaml
│   ├── machine-config-server-tls-secret.yaml
│   └── openshift-config-secret-pull-secret.yaml
└── openshift
    ├── 99_cloud-creds-secret.yaml
    ├── 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
    ├── 99_openshift-cluster-api_master-user-data-secret.yaml
    ├── 99_openshift-cluster-api_worker-machineset-0.yaml
    ├── 99_openshift-cluster-api_worker-user-data-secret.yaml
    ├── 99_openshift-machineconfig_99-master-ssh.yaml
    ├── 99_openshift-machineconfig_99-worker-ssh.yaml
    ├── 99_role-cloud-creds-secret-reader.yaml
    └── openshift-install-manifests.yaml

```

### 15.9.1. 修改清单文件



#### 注意

AWS Outposts 环境有以下限制，需要在清单文件中手动修改：

- 网络连接的最大传输单元 (MTU) 是可通过连接传递的最大允许的数据包的大小（以字节为单位）。Outpost 服务链接支持最大数据包大小为 1300 字节。有关服务链接的更多信息，请参阅 [Outpost 连接 AWS 区域](#)

您将找到有关如何更改这些值的更多信息。

- 将 Outpost 子网用于 worker **machineset**

修改以下文件：`<installation_directory>/openshift/99_openshift-cluster-api_worker-machineset-0.yaml`。找到子网 ID，并将其替换为 Outpost 中创建的专用子网的 ID。因此，所有 worker 机器将在 Outpost 中创建。

- 为网络提供程序指定 MTU 值

Outpost 服务链接支持最大数据包大小为 1300 字节。需要修改网络提供程序的 MTU 以符合此要求。在 manifests 目录下创建一个名为 `cluster-network-03-config.yml` 的新文件

如果使用 OpenShift SDN 网络供应商，请将 MTU 值设置为 1250

```

apiVersion: operator.openshift.io/v1
kind: Network
metadata:

```

```
name: cluster
spec:
  defaultNetwork:
    openshiftSDNConfig:
      mtu: 1250
```

如果使用 OVN-Kubernetes 网络供应商，请将 MTU 值设置为 1200

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    ovnKubernetesConfig:
      mtu: 1200
```

## 15.10. 通过下载二进制文件安装 OPENSIFT CLI

您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

### 15.10.1. 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. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。
5. 解包存档：

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

6. 将 **oc** 二进制文件放到 **PATH** 中的目录中。  
要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

### 15.10.2. 在 Windows 上安装 OpenShift CLI

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

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
4. 使用 ZIP 程序解压存档。
5. 将 **oc** 二进制文件移到 **PATH** 中的目录中。  
要查看您的 **PATH**，请打开命令提示并执行以下命令：

```
C:\> path
```

#### 验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

### 15.10.3. 在 macOS 上安装 OpenShift CLI

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

#### 流程

1. 导航到红帽客户门户网站上的 [OpenShift Container Platform 下载页面](#)。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。



#### 注意

对于 macOS arm64，请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

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**，请打开终端并执行以下命令：

```
$ echo $PATH
```

## 验证

- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

## 15.11. 在 KUBE-SYSTEM 项目中存储管理员级别的 SECRET 的替代方案

默认情况下，管理员 secret 存储在 **kube-system** 项目中。如果您在 **install-config.yaml** 文件中将 **credentialsMode** 参数配置为 **Manual**，则必须使用以下替代方案之一：

- 要手动管理长期云凭证，请按照[手动创建长期凭证](#)中的步骤操作。
- 要实现在集群外为各个组件管理的短期凭证，请按照[配置 AWS 集群以使用短期凭证](#)中的步骤操作。

### 15.11.1. 手动创建长期凭证

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. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

4. 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 自定义资源 (CR) 列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included 1 \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml 2 \
  --to=<path_to_directory_for_credentials_requests> 3
```

- 1 **--included** 参数仅包含特定集群配置所需的清单。
- 2 指定 **install-config.yaml** 文件的位置。
- 3 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

此命令为每个 **CredentialsRequest** 对象创建一个 YAML 文件。

### CredentialsRequest 对象示例

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

5. 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。

### 带有 secret 的 CredentialsRequest 对象示例

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <component_credentials_request>
  namespace: openshift-cloud-credential-operator
...
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
      - effect: Allow
        action:
          - s3:CreateBucket
          - s3:DeleteBucket
        resource: "*"
    ...
  secretRef:
```

```
name: <component_secret>
namespace: <component_namespace>
...
```

### Secret 对象示例

```
apiVersion: v1
kind: Secret
metadata:
  name: <component_secret>
  namespace: <component_namespace>
data:
  aws_access_key_id: <base64_encoded_aws_access_key_id>
  aws_secret_access_key: <base64_encoded_aws_secret_access_key>
```



#### 重要

在升级使用手动维护凭证的集群前，您必须确保 CCO 处于可升级状态。

## 15.11.2. 将 AWS 集群配置为使用短期凭证

要安装配置为使用 AWS 安全令牌服务 (STS) 的集群，您必须配置 CCO 实用程序并为集群创建所需的 AWS 资源。

### 15.11.2.1. 配置 Cloud Credential Operator 工具

当 Cloud Credential Operator (CCO) 以手动模式运行时，要从集群外部创建和管理云凭证，提取并准备 CCO 实用程序 (**ccctl**) 二进制文件。



#### 注意

**ccctl** 工具是在 Linux 环境中运行的 Linux 二进制文件。

### 先决条件

- 您可以访问具有集群管理员权限的 OpenShift Container Platform 帐户。
- 已安装 OpenShift CLI (**oc**)。
- 您已为 **ccctl** 工具创建了用于以下权限的 AWS 帐户：

#### 例 15.1. 所需的 AWS 权限

##### 所需的 iam 权限

- **iam:CreateOpenIDConnectProvider**
- **iam:CreateRole**
- **iam>DeleteOpenIDConnectProvider**
- **iam>DeleteRole**
- **iam>DeleteRolePolicy**

- **iam:GetOpenIDConnectProvider**
- **iam:GetRole**
- **iam:GetUser**
- **iam:ListOpenIDConnectProviders**
- **iam:ListRolePolicies**
- **iam:ListRoles**
- **iam:PutRolePolicy**
- **iam:TagOpenIDConnectProvider**
- **iam:TagRole**

#### 所需的 s3 权限

- **s3:CreateBucket**
- **s3:DeleteBucket**
- **s3:DeleteObject**
- **s3:GetBucketAcl**
- **s3:GetBucketTagging**
- **s3:GetObject**
- **s3:GetObjectAcl**
- **s3:GetObjectTagging**
- **s3:ListBucket**
- **s3:PutBucketAcl**
- **s3:PutBucketPolicy**
- **s3:PutBucketPublicAccessBlock**
- **s3:PutBucketTagging**
- **s3:PutObject**
- **s3:PutObjectAcl**
- **s3:PutObjectTagging**

#### 所需的 cloudfront 权限

- **cloudfront:ListCloudFrontOriginAccessIdentities**



- **cloudfront:ListDistributions**
- **cloudfront:ListTagsForResource**

如果您计划通过公共 CloudFront 发行版 URL 将 OIDC 配置存储在 IAM 身份提供程序访问的私有 S3 存储桶中，则运行 **ccctl** 工具的 AWS 帐户需要以下额外权限：

#### 例 15.2. 使用 CloudFront 私有 S3 存储桶的额外权限

- **cloudfront:CreateCloudFrontOriginAccessIdentity**
- **cloudfront:CreateDistribution**
- **cloudfront>DeleteCloudFrontOriginAccessIdentity**
- **cloudfront>DeleteDistribution**
- **cloudfront:GetCloudFrontOriginAccessIdentity**
- **cloudfront:GetCloudFrontOriginAccessIdentityConfig**
- **cloudfront:GetDistribution**
- **cloudfront:TagResource**
- **cloudfront:UpdateDistribution**



#### 注意

在使用 **ccctl aws create-all** 命令处理凭证请求时，这些额外权限支持使用 **--create-private-s3-bucket** 选项。

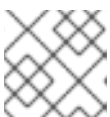
## 流程

1. 运行以下命令，为 OpenShift Container Platform 发行镜像设置变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 运行以下命令，从 OpenShift Container Platform 发行镜像获取 CCO 容器镜像：

```
$ CCO_IMAGE=$(oc adm release info --image-for='cloud-credential-operator'
$RELEASE_IMAGE -a ~/.pull-secret)
```



#### 注意

确保 **\$RELEASE\_IMAGE** 的架构与将使用 **ccctl** 工具的环境架构相匹配。

3. 运行以下命令，将 CCO 容器镜像中的 **ccctl** 二进制文件提取到 OpenShift Container Platform 发行镜像中：

```
$ oc image extract $CCO_IMAGE --file="/usr/bin/ccctl" -a ~/.pull-secret
```

- 运行以下命令更改权限以使 **ccctl** 可执行：

```
$ chmod 775 ccctl
```

## 验证

- 要验证 **ccctl** 是否准备就绪，可以尝试显示帮助文件。运行命令时使用相对文件名，例如：

```
$ ./ccctl.rhel9
```

## 输出示例

```
OpenShift credentials provisioning tool
```

```
Usage:
```

```
ccctl [command]
```

```
Available Commands:
```

```
alibabacloud Manage credentials objects for alibaba cloud
```

```
aws          Manage credentials objects for AWS cloud
```

```
azure        Manage credentials objects for Azure
```

```
gcp          Manage credentials objects for Google cloud
```

```
help         Help about any command
```

```
ibmcloud     Manage credentials objects for IBM Cloud
```

```
nutanix      Manage credentials objects for Nutanix
```

```
Flags:
```

```
-h, --help  help for ccctl
```

```
Use "ccctl [command] --help" for more information about a command.
```

### 15.11.2.2. 使用 Cloud Credential Operator 实用程序创建 AWS 资源

创建 AWS 资源时有以下选项：

- 您可以使用 **ccctl aws create-all** 命令自动创建 AWS 资源。这是创建资源的最快速方法。请参阅 [使用单个命令创建 AWS 资源](#)。
- 如果您需要在修改 AWS 资源前查看 **ccctl** 工具创建的 JSON 文件，或者 **ccctl** 工具用于创建 AWS 资源的过程无法自动满足组织的要求，您可以单独创建 AWS 资源。请参阅 [单独创建 AWS 资源](#)。

#### 15.11.2.2.1. 使用单个命令创建 AWS 资源

如果 **ccctl** 工具用于创建 AWS 资源的过程自动满足机构的要求，您可以使用 **ccctl aws create-all** 命令自动创建 AWS 资源。

否则，您可以单独创建 AWS 资源。如需更多信息，请参阅“单独创建 AWS 资源”。



## 注意

默认情况下，**ccctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccctl\_output\_dir>** 来引用这个目录。

## 先决条件

您必须：

- 提取并准备好 **ccctl** 二进制文件。

## 流程

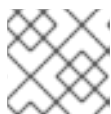
- 运行以下命令，使用安装文件中的发行镜像设置 **\$RELEASE\_IMAGE** 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk 'release image/' {print $3})
```

- 运行以下命令，从 OpenShift Container Platform 发行镜像中提取 **CredentialsRequest** 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \1
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \2
  --to=<path_to_directory_for_credentials_requests> \3
```

- included** 参数仅包含特定集群配置所需的清单。
- 指定 **install-config.yaml** 文件的位置。
- 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。



## 注意

此命令可能需要一些时间才能运行。

- 运行以下命令，使用 **ccctl** 工具处理所有 **CredentialsRequest** 对象：

```
$ ccctl aws create-all \
  --name=<name> \1
  --region=<aws_region> \2
  --credentials-requests-dir=<path_to_credentials_requests_directory> \3
  --output-dir=<path_to_ccctl_output_dir> \4
  --create-private-s3-bucket \5
```

- 指定用于标记创建用于跟踪的任何云资源的名称。
- 指定在其中创建云资源的 AWS 区域。
- 指定包含组件 **CredentialsRequest** 对象文件的目录。

- 4 可选：指定您希望 **ccoctl** 实用程序在其中创建对象的目录。默认情况下，实用程序在运行命令的目录中创建对象。
- 5 可选：默认情况下，**ccoctl** 实用程序将 OpenID Connect (OIDC) 配置文件存储在公共 S3 存储桶中，并使用 S3 URL 作为公共 OIDC 端点。要将 OIDC 配置存储在 IAM 身份提供程序通过公共 CloudFront 发行版 URL 访问的专用 S3 存储桶中，请使用 **--create-private-s3-bucket** 参数。



### 注意

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

### 验证

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccoctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccoctl_output_dir>/manifests
```

### 输出示例

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

#### 15.11.2.2.2. 单独创建 AWS 资源

您可以使用 **ccoctl** 工具单独创建 AWS 资源。这个选项对于在不同用户或部门之间创建这些资源的组织可能很有用。

否则，您可以使用 **ccoctl aws create-all** 命令自动创建 AWS 资源。如需更多信息，请参阅“使用单个命令创建 AWS 资源”。



### 注意

默认情况下，**ccoctl** 在运行命令的目录中创建对象。要在其他目录中创建对象，请使用 **--output-dir** 标志。此流程使用 **<path\_to\_ccoctl\_output\_dir>** 来引用这个目录。

有些 **ccoctl** 命令会发出 AWS API 调用来创建或修改 AWS 资源。您可以使用 **--dry-run** 标志来避免 API 调用。使用此标志可在本地文件系统中创建 JSON 文件。您可以使用 **--cli-input-json** 参数查看和修改 JSON 文件，然后使用 AWS CLI 工具应用它们。

### 先决条件

- 提取并准备 **ccctl** 二进制文件。

## 流程

1. 运行以下命令，生成用于为集群设置 OpenID Connect 供应商的公共和私有 RSA 密钥文件：

```
$ ccctl aws create-key-pair
```

### 输出示例

```
2021/04/13 11:01:02 Generating RSA keypair
2021/04/13 11:01:03 Writing private key to /<path_to_ccctl_output_dir>/serviceaccount-signer.private
2021/04/13 11:01:03 Writing public key to /<path_to_ccctl_output_dir>/serviceaccount-signer.public
2021/04/13 11:01:03 Copying signing key for use by installer
```

其中 **serviceaccount-signer.private** 和 **serviceaccount-signer.public** 是生成的密钥文件。

此命令还会在 **/<path\_to\_ccctl\_output\_dir>/tls/bound-service-account-signing-key.key** 中创建集群在安装过程中所需的私钥。

2. 运行以下命令，在 AWS 上创建 OpenID Connect 身份提供程序和 S3 存储桶：

```
$ ccctl aws create-identity-provider \
  --name=<name> ①
  --region=<aws_region> ②
  --public-key-file=<path_to_ccctl_output_dir>/serviceaccount-signer.public ③
```

① **<name>** 是用于标记为跟踪而创建的云资源的名称。

② **<aws-region>** 是将来要在其中创建云资源的 AWS 区域。

③ **<path\_to\_ccctl\_output\_dir>** 是 **ccctl aws create-key-pair** 命令生成的公钥文件的路径。

### 输出示例

```
2021/04/13 11:16:09 Bucket <name>-oidc created
2021/04/13 11:16:10 OpenID Connect discovery document in the S3 bucket <name>-oidc at .well-known/openid-configuration updated
2021/04/13 11:16:10 Reading public key
2021/04/13 11:16:10 JSON web key set (JWKS) in the S3 bucket <name>-oidc at keys.json updated
2021/04/13 11:16:18 Identity Provider created with ARN: arn:aws:iam::
<aws_account_id>:oidc-provider/<name>-oidc.s3.<aws_region>.amazonaws.com
```

其中 **openid-configuration** 是发现文档和 **key.json** 是一个 JSON Web 密钥集文件。

此命令还会在 **/<path\_to\_ccctl\_output\_dir>/manifests/cluster-authentication-02-config.yaml** 中创建 YAML 配置文件。此文件为集群生成的服务帐户令牌设置签发者 URL 字段，以便 AWS IAM 身份提供程序信任令牌。

## 3. 为集群中的每个组件创建 IAM 角色：

- a. 运行以下命令，使用安装文件中的发行镜像设置
- \$RELEASE\_IMAGE**
- 变量：

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

- b. 从 OpenShift Container Platform 发行镜像中提取
- CredentialsRequest**
- 对象列表：

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included 1 \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
2 \
  --to=<path_to_directory_for_credentials_requests> 3
```

- 1** **--included** 参数仅包含特定集群配置所需的清单。
- 2** 指定 **install-config.yaml** 文件的位置。
- 3** 指定要存储 **CredentialsRequest** 对象的目录的路径。如果指定的目录不存在，这个命令会创建它。

- c. 运行以下命令，使用
- ccoctl**
- 工具处理所有
- CredentialsRequest**
- 对象：

```
$ ccoctl aws create-iam-roles \
  --name=<name> \
  --region=<aws_region> \
  --credentials-requests-dir=<path_to_credentials_requests_directory> \
  --identity-provider-arn=arn:aws:iam::<aws_account_id>:oidc-provider/<name>-oidc.s3.
  <aws_region>.amazonaws.com
```

**注意**

对于使用其他 IAM API 端点的 AWS 环境（如 GovCloud），还必须使用 **--region** 参数指定您的区域。

如果您的集群使用 **TechPreviewNoUpgrade** 功能集启用的技术预览功能，则必须包含 **--enable-tech-preview** 参数。

对于每个 **CredentialsRequest** 对象，**ccoctl** 创建一个带有信任策略的 IAM 角色，该角色与指定的 OIDC 身份提供程序相关联，以及来自 OpenShift Container Platform 发行镜像的每个 **CredentialsRequest** 对象中定义的权限策略。

**验证**

- 要验证 OpenShift Container Platform secret 是否已创建，列出 **<path\_to\_ccoctl\_output\_dir>/manifests** 目录中的文件：

```
$ ls <path_to_ccoctl_output_dir>/manifests
```

**输出示例**

```
cluster-authentication-02-config.yaml
openshift-cloud-credential-operator-cloud-credential-operator-iam-ro-creds-credentials.yaml
openshift-cloud-network-config-controller-cloud-credentials-credentials.yaml
openshift-cluster-api-capamanager-bootstrap-credentials-credentials.yaml
openshift-cluster-csi-drivers-ebs-cloud-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-aws-cloud-credentials-credentials.yaml
```

您可以通过查询 AWS 来验证是否已创建 IAM 角色。如需更多信息，请参阅有关列出 IAM 角色的 AWS 文档。

### 15.11.2.3. 整合 Cloud Credential Operator 实用程序清单

要为单个组件在集群外实现短期安全凭证，您必须将创建 Cloud Credential Operator 实用程序 (**ccctl**) 的清单文件移到安装程序的正确目录中。

#### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您已配置了 Cloud Credential Operator 实用程序 (**ccctl**)。
- 已使用 **ccctl** 工具创建了集群所需的云供应商资源。

#### 流程

1. 如果您没有将 **install-config.yaml** 配置文件中的 **credentialsMode** 参数设置为 **Manual**，请修改值，如下所示：

#### 配置文件片段示例

```
apiVersion: v1
baseDomain: example.com
credentialsMode: Manual
# ...
```

2. 如果您之前还没有创建安装清单文件，请运行以下命令：

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

其中 **<installation\_directory>** 是安装程序在其中创建文件的目录。

3. 运行以下命令，将 **ccctl** 工具生成的清单复制到安装程序创建的 **manifests** 目录中：

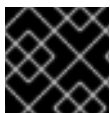
```
$ cp /<path_to_ccctl_output_dir>/manifests/* ./manifests/
```

4. 将包含私钥的 **tls** 目录复制到安装目录中：

```
$ cp -a /<path_to_ccctl_output_dir>/tls .
```

## 15.12. 部署集群

您可以在兼容云平台上安装 OpenShift Container Platform。



### 重要

在初始安装过程中，您只能运行安装程序的 **create cluster** 命令一次。

### 先决条件

- 您已使用托管集群的云平台配置了帐户。
- 您有 OpenShift Container Platform 安装程序和集群的 pull secret。
- 已确认主机上的云供应商帐户具有部署集群的正确权限。权限不正确的帐户会导致安装过程失败，并显示包括缺失权限的错误消息。

### 流程

1. 进入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**，而不是 **info**。

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



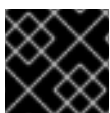
### 注意

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

### 验证

当集群部署成功完成时：

- 终端会显示用于访问集群的说明，包括指向 Web 控制台和 **kubeadmin** 用户的凭证的链接。
- 凭证信息还会输出到 **<installation\_directory>/./openshift\_install.log**。



### 重要

不要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。

### 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: "password"
```

```
INFO Time elapsed: 36m22s
```



## 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。一个例外是，您必须手动批准待处理的 **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 小时时间进行轮转。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。

## 15.13. 使用 CLI 登录集群

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** 文件包含有关集群的信息，供 CLI 用于将客户端连接到正确的集群和 API 服务器。该文件特定于集群，在 OpenShift Container Platform 安装过程中创建。

### 先决条件

- 已部署 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. 验证您可以使用导出的配置成功运行 **oc** 命令：

```
$ oc whoami
```

### 输出示例

```
system:admin
```

```
/validating-an-installation.adoc
```

## 15.14. 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。

本文档为

### 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。

## 15.15. OPENSIFT CONTAINER PLATFORM 的 TELEMETRY 访问

在 OpenShift Container Platform 4.14 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

- 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](#)。
- 有关 Telemetry 服务的更多信息，请参阅[关于远程健康监控](#)。

## 15.16. 集群限制

重要

AWS Outposts 不支持 Network Load Balancer (NLB) 和 Classic Load Balancer。创建集群后，会在 AWS 区域中创建所有 Load Balancers。要使用在 Outpost 实例中创建的 Load Balancer，应使用 Application Load Balancer。可以使用 AWS Load Balancer Operator 来实现这一目标。

如果要使用位于 outpost 实例中的公共子网作为 ALB，则需要删除之前在 VPC 创建过程中添加的特殊标签 (**kubernetes.io/cluster/rhcs-outposts: owned**)。这将阻止您创建类型为 LoadBalancer 的新服务 (Network Load Balancer)。

如需更多信息，请参阅[了解 AWS Load Balancer Operator](#)

重要

使用 AWS Elastic Block Store 限制的持久性存储

- AWS Outposts 不支持 Amazon Elastic Block Store (EBS) gp3 卷。安装后，集群包含两个存储类 - gp3-csi 和 gp2-csi，gp3-csi 是默认存储类。总是使用 gp2-csi 非常重要。您可以使用以下 OpenShift CLI (oc) 命令更改默认存储类：

```
$ oc annotate --overwrite storageclass gp3-csi storageclass.kubernetes.io/is-default-class=false
$ oc annotate --overwrite storageclass gp2-csi storageclass.kubernetes.io/is-default-class=true
```

- 要在 Outpost 实例中创建卷，CSI 驱动程序根据存储在 CSINode 对象上的拓扑键决定 Outpost ARN。为确保 CSI 驱动程序使用正确的拓扑值，需要使用 **WaitForConsumer** 卷绑定模式，并避免在创建的任何新存储类上设置允许拓扑。

## 15.17. 后续步骤

- [验证安装](#)
- [自定义集群](#)
- [远程健康报告](#)
- [删除云供应商凭证](#)

## 第 16 章 在 AWS 上安装三节点集群

在 OpenShift Container Platform 版本 4.14 中，您可以在 Amazon Web Services (AWS) 上安装三节点集群。三节点集群包含三个 control plane 机器，它们也可以充当计算机器。这种类型的集群提供了一个较小的、效率更高的集群，供集群管理员和开发人员用于测试、开发和生产。

您可以使用安装程序置备或用户置备的基础架构安装三节点集群。



### 注意

不支持使用 AWS Marketplace 镜像部署三节点集群。

### 16.1. 配置三节点集群

在部署集群前，您可以通过将 **install-config.yaml** 文件中的 worker 节点数量设置为 **0** 来配置三节点集群。将 worker 节点数量设置为 **0** 可确保 control plane 机器可以调度。这允许调度应用程序工作负载从 control plane 节点运行。



### 注意

因为应用程序工作负载从 control plane 节点运行，所以需要额外的订阅，因为 control plane 节点被视为计算节点。

#### 先决条件

- 您有一个现有的 **install-config.yaml** 文件。

#### 流程

1. 将 **install-config.yaml** 文件中的计算副本数量设置为 **0**，如以下 **compute** 小节中所示：

#### 三节点集群的 **install-config.yaml** 文件示例

```
apiVersion: v1
baseDomain: example.com
compute:
- name: worker
  platform: {}
  replicas: 0
# ...
```

2. 如果您使用用户置备的基础架构部署集群：
  - 创建 Kubernetes 清单文件后，请确保在 **cluster-scheduler-02-config.yml** 文件中将 **spec.mastersSchedulable** 参数设置为 **true**。您可以在 **<installation\_directory>/manifests** 中找到此文件。如需更多信息，请参阅“使用 CloudFormation 模板在 AWS 中用户置备的基础架构上安装集群”中的“创建 Kubernetes 清单和 Ignition 配置文件”。
  - 不要创建额外的 worker 节点。

#### 三节点集群的 **cluster-scheduler-02-config.yml** 文件示例

```
apiVersion: config.openshift.io/v1
kind: Scheduler
metadata:
  creationTimestamp: null
  name: cluster
spec:
  mastersSchedulable: true
  policy:
    name: ""
status: {}
```

## 16.2. 后续步骤

- [使用自定义在 AWS 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](#)

## 第 17 章 在 AWS 上卸载集群

您可以删除部署到 Amazon Web Services (AWS) 的集群。

### 17.1. 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未正确删除的资源，特别是在用户置备基础架构(UPI)集群中。可能存在安装程序未创建或安装程序无法访问的资源。

#### 先决条件

- 有用于部署集群的安装程序副本。
- 有创建集群时安装程序生成的文件。

#### 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**，而不是 **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。

### 17.2. 使用 CLOUD CREDENTIAL OPERATOR 实用程序删除 AMAZON WEB SERVICES 资源

卸载使用在集群外管理的短期凭证的 OpenShift Container Platform 集群后，您可以使用 CCO 实用程序 (**ccctl**) 删除 **ccctl** 在安装过程中创建的 Amazon Web Services (AWS) 资源。

#### 先决条件

- 提取并准备 **ccctl** 二进制文件。
- 卸载使用短期凭证的 AWS 上的 OpenShift Container Platform 集群。

#### 流程

- 运行以下命令，删除 **ccctl** 创建的 AWS 资源：

```
$ ccctl aws delete \
  --name=<name> \ 1
  --region=<aws_region> 2
```

**1** **<name>** 与最初用于创建和标记云资源的名称匹配。

**2** **<aws\_region>** 是要删除云资源的 AWS 区域。

### 输出示例

```
2021/04/08 17:50:41 Identity Provider object .well-known/openid-configuration deleted from
the bucket <name>-oidc
2021/04/08 17:50:42 Identity Provider object keys.json deleted from the bucket <name>-oidc
2021/04/08 17:50:43 Identity Provider bucket <name>-oidc deleted
2021/04/08 17:51:05 Policy <name>-openshift-cloud-credential-operator-cloud-credential-o
associated with IAM Role <name>-openshift-cloud-credential-operator-cloud-credential-o
deleted
2021/04/08 17:51:05 IAM Role <name>-openshift-cloud-credential-operator-cloud-credential-
o deleted
2021/04/08 17:51:07 Policy <name>-openshift-cluster-csi-drivers-ebs-cloud-credentials
associated with IAM Role <name>-openshift-cluster-csi-drivers-ebs-cloud-credentials deleted
2021/04/08 17:51:07 IAM Role <name>-openshift-cluster-csi-drivers-ebs-cloud-credentials
deleted
2021/04/08 17:51:08 Policy <name>-openshift-image-registry-installer-cloud-credentials
associated with IAM Role <name>-openshift-image-registry-installer-cloud-credentials
deleted
2021/04/08 17:51:08 IAM Role <name>-openshift-image-registry-installer-cloud-credentials
deleted
2021/04/08 17:51:09 Policy <name>-openshift-ingress-operator-cloud-credentials associated
with IAM Role <name>-openshift-ingress-operator-cloud-credentials deleted
2021/04/08 17:51:10 IAM Role <name>-openshift-ingress-operator-cloud-credentials deleted
2021/04/08 17:51:11 Policy <name>-openshift-machine-api-aws-cloud-credentials associated
with IAM Role <name>-openshift-machine-api-aws-cloud-credentials deleted
2021/04/08 17:51:11 IAM Role <name>-openshift-machine-api-aws-cloud-credentials deleted
2021/04/08 17:51:39 Identity Provider with ARN arn:aws:iam::<aws_account_id>:oidc-
provider/<name>-oidc.s3.<aws_region>.amazonaws.com deleted
```

### 验证

- 要验证资源是否已删除，请查询 AWS。如需更多信息，请参阅 AWS 文档。

## 17.3. 使用配置的 AWS LOCAL ZONE 基础架构删除集群

在 Amazon Web Services (AWS) 上安装集群后，进入现有的 Virtual Private Cloud (VPC)，并为每个 Local Zone 位置设置子网，您可以删除集群以及与之关联的任何 AWS 资源。

该流程中的示例假设您使用 CloudFormation 模板创建了 VPC 及其子网。

### 先决条件

- 您知道创建网络期间使用的 CloudFormation 堆栈 **<local\_zone\_stack\_name>** 和 **<vpc\_stack\_name>** 的名称。您需要堆栈的名称来删除集群。
- 您可以访问包含安装程序创建的安装文件的目录的权限。
- 您的帐户包含一个策略，为您提供了删除 CloudFormation 堆栈的权限。

## 流程

1. 进入包含存储的安装程序的目录，并使用 **destroy cluster** 命令删除集群：

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

- 1** 对于 **<installation\_directory>**，请指定保存安装程序创建的任何文件的目录。
- 2** 要查看不同的日志详细信息，请指定 **error**、**info** 或 **warn**，而不是 **debug**。

2. 删除 Local Zone 子网的 CloudFormation 堆栈：

```
$ aws cloudformation delete-stack --stack-name <local_zone_stack_name>
```

3. 删除代表 VPC 的资源堆栈：

```
$ aws cloudformation delete-stack --stack-name <vpc_stack_name>
```

## 验证

- 通过在 AWS CLI 中执行以下命令，检查您是否删除了堆栈资源。AWS CLI 输出没有模板组件。

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

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

## 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](#)。
- [选择 AWS 本地区域](#)
- [AWS 本地区域可用位置](#)
- [AWS Local Zones 功能](#)



## 第 18 章 AWS 的安装配置参数

在 AWS 上部署 OpenShift Container Platform 集群前，您可以提供参数来自定义集群和托管它的平台。在创建 `install-config.yaml` 文件时，您可以通过命令行为所需参数提供值。然后，您可以修改 `install-config.yaml` 文件以进一步自定义集群。

### 18.1. AWS 可用的安装配置参数

下表指定您可以在安装过程中设置所需的、可选和 AWS 特定安装配置参数。



**重要**

安装后，您无法更改 `install-config.yaml` 文件中的这些参数。

#### 18.1.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 18.1. 所需的参数

| 参数                           | 描述                                                                                                                                                                        | 值                                   |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| <code>apiVersion:</code>     | <code>install-config.yaml</code> 内容的 API 版本。当前版本为 <b>v1</b> 。安装程序可能还支持旧的 API 版本。                                                                                          | 字符串                                 |
| <code>baseDomain:</code>     | 云供应商的基域。基域用于创建到 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> 。   |
| <code>metadata:</code>       | Kubernetes 资源 <b>ObjectMeta</b> ，其中只消耗 <b>name</b> 参数。                                                                                                                    | 对象                                  |
| <code>metadata: name:</code> | 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(.)字符串，如 <b>dev</b> 。 |

| 参数                                                                                            | 描述                                                                                                                                                                                                                                                              | 值                                                                                                                                                                                                                                                                                        |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  platform:   | 要执行安装的具体平台配置： <b>alibabacloud</b> , <b>aws</b> , <b>baremetal</b> , <b>azure</b> , <b>gcp</b> , <b>ibmcloud</b> , <b>nutanix</b> , <b>openstack</b> , <b>powervs</b> , <b>vsphere</b> , or <b>{}</b> 。有关 <b>platform.&lt;platform&gt;</b> 参数的更多信息，请参考下表中您的特定平台。 | 对象                                                                                                                                                                                                                                                                                       |
|  pullSecret: | <a href="#">从 Red Hat OpenShift Cluster Manager 获取 pull secret</a> ，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> |

18.1.2. 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，或者配置与默认值不同的 IP 地址块。

仅支持 IPv4 地址。



**注意**

Red Hat OpenShift Data Foundation 灾难恢复解决方案不支持 Globalnet。对于区域灾难恢复场景，请确保为每个集群中的集群和服务网络使用非重叠的专用 IP 地址。

表 18.2. 网络参数

| 参数                                                                                              | 描述       | 值                                                                                                                                                 |
|-------------------------------------------------------------------------------------------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------|
|  networking: | 集群网络的配置。 | <div><b>注意</b><br/>您不能在安装后更改 <b>networking</b> 对象指定的参数。</div> |

| 参数                                                           | 描述                                                                                                                                                                         | 值                                                                                                                                                                                        |
|--------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>networking:<br/>networkType:</code>                    | 要安装的 Red Hat OpenShift Networking 网络插件。                                                                                                                                    | <b>OpenShiftSDN</b> 或 <b>OVNKubernetes</b> 。 <b>OpenShiftSDN</b> 是 all-Linux 网络的 CNI 插件。 <b>OVNKubernetes</b> 是 Linux 网络和包含 Linux 和 Windows 服务器的混合网络的 CNI 插件。默认值为 <b>OVNKubernetes</b> 。 |
| <code>networking:<br/>clusterNetwork:</code>                 | pod 的 IP 地址块。<br><br>默认值为 <b>10.128.0.0/14</b> ，主机前缀为 <b>/23</b> 。<br><br>如果您指定了多个 IP 地址块，块不得重叠。                                                                           | 对象数组。例如：<br><br><code>networking:<br/>clusterNetwork:<br/>- cidr: 10.128.0.0/14<br/>hostPrefix: 23</code>                                                                                |
| <code>networking:<br/>clusterNetwork:<br/>cidr:</code>       | 使用 <b>networking.clusterNetwork</b> 时需要此项。IP 地址块。<br><br>IPv4 网络。                                                                                                          | 无类别域间路由(CIDR)表示法中的 IP 地址块。IPv4 块的前缀长度介于 <b>0 到 32</b> 之间。                                                                                                                                |
| <code>networking:<br/>clusterNetwork:<br/>hostPrefix:</code> | 分配给每个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 given <b>cidr</b> 中分配 a/ <b>23</b> 子网。 <b>hostPrefix</b> 值 <b>23</b> 提供 510 ( $2^{(32 - 23)} - 2$ ) pod IP 地址。 | 子网前缀。<br><br>默认值为 <b>23</b> 。                                                                                                                                                            |
| <code>networking:<br/>serviceNetwork:</code>                 | 服务的 IP 地址块。默认值为 <b>172.30.0.0/16</b> 。<br><br>OpenShift SDN 和 OVN-Kubernetes 网络插件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：<br><br><code>networking:<br/>serviceNetwork:<br/>- 172.30.0.0/16</code>                                                                                          |
| <code>networking:<br/>machineNetwork:</code>                 | 机器的 IP 地址块。<br><br>如果您指定了多个 IP 地址块，块不得重叠。                                                                                                                                  | 对象数组。例如：<br><br><code>networking:<br/>machineNetwork:<br/>- cidr: 10.0.0.0/16</code>                                                                                                     |

| 参数                                                           | 描述                                                                                                                                                                                                                          | 值                                                                                                                                                                                                                                                          |
|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>networking:</b><br><b>machineNetwork:</b><br><b>cidr:</b> | 使用 <b>networking.machineNetwork</b> 时需要此项。IP 地址块。对于 libvirt 和 IBM Power® Virtual Server 以外的所有平台，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> 。对于 IBM Power® Virtual Server，默认值为 <b>192.168.0.0/24</b> 。 | CIDR 表示法中的 IP 网络块。<br><br>例如： <b>10.0.0.0/16</b> 。<br><br> <div> <b>注意</b><br/><br/>           将 <b>networking.machineNetwork</b> 设置为与首选 NIC 所在的 CIDR 匹配。         </div> |

### 18.1.3. 可选的配置参数

下表描述了可选的安装配置参数：

表 18.3. 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。配置代理时，也可以使用此信任捆绑包。                                                     | 字符串   |
| <b>capabilities:</b>                                          | 控制可选核心组件的安装。您可以通过禁用可选组件来减少 OpenShift Container Platform 集群的空间。如需更多信息，请参阅 安装中的“集群功能”页面。                  | 字符串数组 |
| <b>capabilities:</b><br><b>baselineCapabilitySet:</b>         | 选择要启用的一组初始可选功能。有效值为 <b>None</b> 、 <b>v4.11</b> 、 <b>v4.12</b> 和 <b>vCurrent</b> 。默认值为 <b>vCurrent</b> 。 | 字符串   |
| <b>capabilities:</b><br><b>additionalEnabledCapabilities:</b> | 将可选功能集合扩展到您在 <b>baselineCapabilitySet</b> 中指定的范围。您可以在该参数中指定多个功能。                                        | 字符串数组 |

| 参数                                        | 描述                                                                                                                                                                                                                                    | 值                                                                                                  |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <code>cpuPartitioningMode:</code>         | 启用工作负载分区，它会隔离 OpenShift Container Platform 服务、集群管理工作负载和基础架构 pod，以便在保留的一组 CPU 上运行。您只能在安装过程中启用工作负载分区。您不能在安装后禁用它。虽然此字段启用工作负载分区，但它不会将工作负载配置为使用特定的 CPU。如需更多信息，请参阅 <i>Scalability and Performance</i> 部分中的 <i>Workload partitioning</i> 页面。 | <b>None</b> 或 <b>AllNodes.None</b> 是默认值。                                                           |
| <code>compute:</code>                     | 组成计算节点的机器的配置。                                                                                                                                                                                                                         | <b>MachinePool</b> 对象的数组。                                                                          |
| <code>compute:<br/>architecture:</code>   | 决定池中机器的指令集架构。目前，不支持具有不同架构的集群。所有池都必须指定相同的架构。有效值为 <b>amd64</b> 和 <b>arm64</b> 。并非所有安装选项都支持 64 位 ARM 架构。要验证您的平台上是否支持您的安装选项，请参阅 <i>选择集群安装方法并为用户准备它中的不同平台支持的安装方法</i> 。                                                                     | 字符串                                                                                                |
| <code>compute:<br/>hyperthreading:</code> | <p>是否在计算机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>  <div> <p><b>重要</b></p> <p>如果您禁用并发多线程，请确保您的容量规划考虑机器性能显著降低的情况。</p> </div> </div>        | <b>enabled</b> 或 <b>Disabled</b>                                                                   |
| <code>compute:<br/>name:</code>           | 使用 <b>compute</b> 时需要此项。机器池的名称。                                                                                                                                                                                                       | <b>worker</b>                                                                                      |
| <code>compute:<br/>platform:</code>       | 使用 <b>compute</b> 时需要此项。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>alibabacloud, aws, azure, gcp, ibmcloud, nutanix, openstack, powervs, vsphere</b> , 或 <b>{}</b> |
| <code>compute:<br/>replicas:</code>       | 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> 。                                                                |

| 参数                                             | 描述                                                                                                                                                                                                                                                   | 值                                                                                                  |
|------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <code>featureSet:</code>                       | 为功能集启用集群。功能集是 OpenShift Container Platform 功能的集合，默认情况下不启用。有关在安装过程中启用功能集的更多信息，请参阅“使用功能门启用功能”。                                                                                                                                                         | 字符串.要启用的功能集的名称，如 <b>TechPreviewNoUpgrade</b> 。                                                     |
| <code>controlPlane:</code>                     | 组成 control plane 的机器的配置。                                                                                                                                                                                                                             | <b>MachinePool</b> 对象的数组。                                                                          |
| <code>controlPlane:<br/>architecture:</code>   | 决定池中机器的指令集合架构。目前，不支持具有不同架构的集群。所有池都必须指定相同的架构。有效值为 <b>amd64</b> 和 <b>arm64</b> 。并非所有安装选项都支持 64 位 ARM 架构。要验证您的平台上是否支持您的安装选项，请参阅 <i>选择集群安装方法并为用户准备它中的不同平台支持的安装方法</i> 。                                                                                   | 字符串                                                                                                |
| <code>controlPlane:<br/>hyperthreading:</code> | <p>是否在 control plane 机器上启用或禁用并发多 <b>线程或超线程</b>。默认情况下，启用并发多线程以提高机器内核的性能。</p> <div>  <div> <p><b>重要</b></p> <p>如果您禁用并发多线程，请确保您的容量规划考虑机器性能显著降低的情况。</p> </div> </div> | <b>enabled</b> 或 <b>Disabled</b>                                                                   |
| <code>controlPlane:<br/>name:</code>           | 使用 <b>controlPlane</b> 时需要此项。机器池的名称。                                                                                                                                                                                                                 | <b>master</b>                                                                                      |
| <code>controlPlane:<br/>platform:</code>       | 使用 <b>controlPlane</b> 时需要此项。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>alibabacloud, aws, azure, gcp, ibmcloud, nutanix, openstack, powervs, vsphere</b> , 或 <b>{}</b> |
| <code>controlPlane:<br/>replicas:</code>       | 要置备的 control plane 机器数量。                                                                                                                                                                                                                             | 部署单节点 OpenShift 时支持的值为 <b>3</b> 或 <b>1</b> 。                                                       |
| <code>credentialsMode:</code>                  | Cloud Credential Operator(CCO)模式。如果没有指定模式，CCO 会动态尝试决定提供的凭证的功能，在支持多个模式的平台上首选 mint 模式。                                                                                                                                                                 | <b>Mint</b> 、 <b>Passthrough</b> 、 <b>Manual</b> 或空字符串(“”)。 <sup>[1]</sup>                         |

| 参数                           | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 值                                                        |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| fips:                        | <p>启用或禁用 FIPS 模式。默认值为 <b>false</b>（禁用）。如果启用 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 RHCOS 提供的加密模块。</p> <div><div></div><div><p><b>重要</b></p><p>要为集群启用 FIPS 模式，您必须从配置为以 FIPS 模式操作的 Red Hat Enterprise Linux (RHEL) 计算机运行安装程序。有关在 RHEL 中配置 FIPS 模式的更多信息，请参阅 将 RHEL 切换到 FIPS 模式。</p><p>当以 FIPS 模式运行 Red Hat Enterprise Linux (RHEL) 或 Red Hat Enterprise Linux CoreOS (RHCOS) 时，OpenShift Container Platform 核心组件使用 RHEL 加密库，在 x86_64、ppc64le 和 s390x 架构上提交到 NIST FIPS 140-2/140-3 Validation。</p><p>如果使用 Azure File 存储，则无法启用 FIPS 模式。</p></div></div> | false 或 true                                             |
| imageContentSources:         | release-image 内容的源和存储库。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 对象数组。包括一个 <b>source</b> 以及可选的 <b>mirrors</b> ，如本表的以下行所述。 |
| imageContentSources: source: | 使用 <b>imageContentSources</b> 时需要此项。指定用户在镜像拉取规格中引用的存储库。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 字符串                                                      |

| 参数                                                                                                                 | 描述                                                                                                                                                                                                                                           | 值                                                                                            |
|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  imageContentSources:<br>mirrors: | 指定可能还包含同一镜像的一个或多个存储库。                                                                                                                                                                                                                        | 字符串数组                                                                                        |
|  platform:<br>aws:<br>lbType:     | 在 AWS 中设置 NLB 负载均衡器类型需要此项。有效值为 <b>Classic</b> 或 <b>NLB</b> 。如果没有指定值，安装程序将默认为 <b>Classic</b> 。安装程序设置 ingress 集群配置对象中提供的值。如果您没有为其他 Ingress Controller 指定负载均衡器类型，它们将使用此参数中设置的类型。                                                                | <b>Classic</b> 或 <b>NLB</b> .默认值为 <b>Classic</b> 。                                           |
|  publish:                         | 如何发布或公开集群的面向用户的端点，如 Kubernetes API、OpenShift 路由。                                                                                                                                                                                             | <b>内部或外部</b> .要部署无法从互联网访问的私有集群，请将 <b>publish</b> 设置为 <b>Internal</b> 。默认值为 <b>External</b> 。 |
|  sshKey:                          | <p>用于验证对集群机器的访问的 SSH 密钥。</p> <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> | 例如， <b>sshKey: ssh-ed25519 AAAA..</b>                                                        |

1. 不是所有 CCO 模式都支持所有云供应商。有关 CCO 模式的更多信息，请参阅 *身份验证和授权* 内容中的“管理云供应商凭证”条目。



**注意**

如果您的 AWS 帐户启用了服务控制策略 (SCP)，必须将 **credentialsMode** 参数配置为 **Mint**、**Passthrough** 或 **Manual**。



**重要**

将此参数设置为 **Manual** 可启用在 **kube-system** 项目中存储管理员级别的 secret 的替代方案，这需要额外的配置步骤。如需更多信息，请参阅“在 kube-system 项目中存储管理员级别的 secret”。

18.1.4. 可选的 AWS 配置参数

下表描述了可选的 AWS 配置参数：



表 18.4. 可选的 AWS 参数

| 参数                                                                                                                                  | 描述                                                                                              | 值                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <code>compute:</code><br><code>platform:</code><br><code>aws:</code><br><code>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。如需可用的 AMI ID，请参阅 <i>RHCOS AMIs for AWS infrastructure</i> 。 |
| <code>compute:</code><br><code>platform:</code><br><code>aws:</code><br><code>iamRole:</code>                                       | 一个已存在的 AWS IAM 角色应用到计算机器池实例配置集。您可以使用这些字段与命名方案匹配，并为您的 IAM 角色包含预定义的权限界限。如果未定义，安装程序会创建一个新的 IAM 角色。 | 有效 AWS IAM 角色的名称。                                                                           |
| <code>compute:</code><br><code>platform:</code><br><code>aws:</code><br><br><code>rootVolume:</code><br><code>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:</code><br><code>platform:</code><br><code>aws:</code><br><br><code>rootVolume:</code><br><code>size:</code>          | 以 GiB 为单位的根卷大小。                                                                                 | 整数，如 <b>500</b> 。                                                                           |
| <code>compute:</code><br><code>platform:</code><br><code>aws:</code><br><br><code>rootVolume:</code><br><code>type:</code>          | 根卷的类型。                                                                                          | 有效的 <a href="#">AWS EBS 卷类型</a> ，如 <b>io1</b> 。                                             |
| <code>compute:</code><br><code>platform:</code><br><code>aws:</code><br><br><code>rootVolume:</code><br><br><code>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 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:</code><br><code>platform:</code><br><code>aws:</code><br><code>type:</code>                                      | 计算机器的 EC2 实例类型。                                                                                             | 有效的 AWS 实例类型，如 <b>m4.2xlarge</b> 。请参阅"使用自定义在 AWS 上安装集群"页中的 "AWS 的经过测试的实例类型"表。               |
| <code>compute:</code><br><code>platform:</code><br><code>aws:</code><br><code>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>controlPlane:</code><br><code>platform:</code><br><code>aws:</code><br><code>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。如需可用的 AMI ID，请参阅 <i>RHCOS AMIs for AWS infrastructure</i> 。 |
| <code>controlPlane:</code><br><code>platform:</code><br><code>aws:</code><br><code>iamRole:</code>                              | 应用到 control plane 机器池实例配置集的已存在的 AWS IAM 角色。您可以使用这些字段与命名方案匹配，并为您的 IAM 角色包含预定义的权限界限。如果未定义，安装程序会创建一个新的 IAM 角色。 | 有效 AWS IAM 角色的名称。                                                                           |
| <code>controlPlane:</code><br><code>platform:</code><br><code>aws:</code><br><br><code>rootVolume:</code><br><code>iops:</code> | 为 control plane 机器上的根卷保留的每秒输入/输出操作 (IOPS)。                                                                  | 整数，如 <b>4000</b> 。                                                                          |
| <code>controlPlane:</code><br><code>platform:</code><br><code>aws:</code><br><br><code>rootVolume:</code><br><code>size:</code> | control plane 机器的根卷大小（以 GiB 为单位）。                                                                           | 整数，如 <b>500</b> 。                                                                           |

| 参数                                                                      | 描述                                                                                                                 | 值                                                                                           |
|-------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| controlPlane:<br>platform:<br>aws:<br><br>rootVolume:<br>type:          | control plane 机器的根卷的类型。                                                                                            | 有效的 <a href="#">AWS EBS 卷类型</a> ，如 <b>io1</b> 。                                             |
| controlPlane:<br>platform:<br>aws:<br><br>rootVolume:<br><br>kmsKeyARN: | 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> 。                                                         |
| controlPlane:<br>platform:<br>aws:<br>type:                             | control plane 机器的 EC2 实例类型。                                                                                        | 有效的 AWS 实例类型，如 <b>m6i.xlarge</b> 。请参阅"使用自定义在 AWS 上安装集群"页中的 "AWS 的经过测试的实例类型"表。               |
| controlPlane:<br>platform:<br>aws:<br>zones:                            | 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。                            |
| platform:<br>aws:<br>amiID:                                             | 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。如需可用的 AMI ID，请参阅 <i>RHCOS AMIs for AWS infrastructure</i> 。 |
| platform:<br>aws:<br><br>hostedZone:                                    | 集群的现有 Route 53 私有托管区。您只能在提供自己的 VPC 时使用已存在的托管区。安装前，托管区必须已经与用户提供的 VPC 关联。另外，托管区的域必须是集群域或集群域的父域。如果未定义，安装程序会创建一个新的托管区。 | 字符串，如 <b>Z3URY6TWQ91KVV</b> 。                                                               |

| 参数                                                       | 描述                                                                                                                                  | 值                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>platform: aws: hostedZoneRole:</pre>                | 在包含指定托管区的帐户中现有 IAM 角色的 Amazon 资源名称(ARN)。在对托管区执行操作时，安装程序和集群操作器会假定此角色。仅在将集群安装到共享 VPC 时使用此参数。                                          | 字符串，如<br><b>arn:aws:iam::1234567890:role/shared-vpc-role</b> 。                                                                                                                                                                                                                                                                                                                                                                                  |
| <pre>platform: aws: region:</pre>                        | 安装程序在其中创建所有集群资源的 AWS 区域。                                                                                                            | <p>任何有效的 <a href="#">AWS 区域</a>，如 <b>us-east-1</b>。您可以运行以下命令来使用 AWS CLI 访问您选择的实例类型可用的区域：</p> <pre>\$ aws ec2 describe-instance-type-offerings --filters Name=instance-type,Values=c7g.xlarge</pre> <div>  <p><b>重要</b></p> <p>在基于 ARM 的 AWS 实例上运行时，请确保进入 AWS Graviton 处理器可用的区域。请参阅 AWS 文档中的<a href="#">全局可用性</a>映射。目前，只有一些区域才提供 AWS Graviton3 处理器。</p> </div> |
| <pre>platform: aws: serviceEndpoints: - name: url:</pre> | AWS 服务端点名称和 URL。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> 名称和有效的 <a href="#">AWS 服务端点 URL</a> 。                                                                                                                                                                                                                                                                                                                                                                              |
| <pre>platform: aws: userTags:</pre>                      | 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。                                                                                                      | <p>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>。</p> <div>  <p><b>注意</b></p> <p>您可以在安装过程中最多添加 25 个用户定义的标签。剩余的 25 个标签是为 OpenShift Container Platform 保留的。</p> </div>                                                                                                              |

| 参数                                                                | 描述                                                                                                                                                                                                                                      | 值                                                              |
|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| <div>platform:<br/>aws:<br/><br/>propagateUserTags:</div>         | 指示集群 Operator 中的标记，在 Operator 创建的 AWS 资源标签中包含指定的用户标签。                                                                                                                                                                                   | 布尔值，如 <b>true</b> 或 <b>false</b> 。                             |
| <div>platform:<br/>aws:<br/>subnets:</div>                        | <p>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。</p> <p>对于标准集群，为每个可用区指定一个公共和私有子网。</p> <p>对于私有集群，为每个可用区指定一个私有子网。</p> <p>对于使用 AWS Local Zones 的集群，您必须将 AWS Local Zone 子网添加到此列表中，以确保创建边缘机器池。</p> | 有效的子网 ID。                                                      |
| <div>platform:<br/>aws:<br/><br/>preserveBootstrapIgnition:</div> | 防止在完成 bootstrap 后删除 S3 存储桶。                                                                                                                                                                                                             | <b>true</b> 或 <b>false</b> 。默认值为 <b>false</b> ，这会导致 S3 存储桶被删除。 |