OpenShift Container Platform 4.4

在 Azure 上安装

安装 OpenShift Container Platform Azure 集群

Last Updated: 2021-03-11

OpenShift Container Platform 4.4 在 Azure 上安装

安装 OpenShift Container Platform Azure 集群

法律通告

摘要

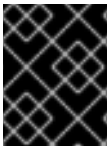
本文档提供在 Microsoft Azure 上安装和卸载 OpenShift Container Platform 集群的说明。

目录

第 1 章 在 AZURE 上安装

1.1. 配置 AZURE 帐户

在安装 OpenShift Container Platform 之前，您必须配置 Microsoft Azure 帐户。

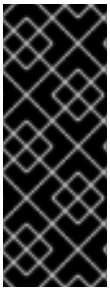


重要

所有通过公共端点提供的 Azure 资源均存在资源名称的限制，您无法创建使用某些名称的资源。如需 Azure 限制词语列表，请参阅 Azure 文档中的解决保留资源名称错误。

1.1.1. Azure 帐户限值

OpenShift Container Platform 集群使用诸多 Microsoft Azure 组件，默认的 Azure 订阅和服务限值、配额和约束会影响您安装 OpenShift Container Platform 集群的能力。



重要

默认的限制因服务类别的不同（如 Free Trial 或 Pay-As-You-Go）以及系列的不同（如 Dv2、F 或 G）而有所不同。例如，对于 Enterprise Agreement 订阅的默认限制是 350 个内核。

在 Azure 上安装默认集群前，请检查您的订阅类型的限制，如有必要，请提高帐户的配额限制。

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

组件	默认所需的组件数	默认 Azure 限值	描述

组件	默认所需的组件数	默认 Azure 限值	描述
vCPU	40	每个区域 20 个	默认集群需要 40 个 vCPU，因此您必须提高帐户限值。 默认情况下，每个集群创建以下实例： ● 一台 Bootstrap 机器，在安装后删除 ● 三个 control plane 机器 ● 三个计算（compute）机器 由于 Bootstrap 机器使用 **Standard_D4s_v3** 机器（使用 4 个 vCPU），control plane 机器使用 **Standard_D8s_v3** 虚拟机（8 个 vCPU），并且 worker 机器使用 **Standard_D4s_v3** 虚拟机（4 个 vCPU），因此默认集群需要 40 个 vCPU。bootstrap 节点 VM（使用 4 个 vCPU）只在安装过程中使用。 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，您必须进一步提高帐户的 vCPU 限值，以确保集群可以部署您需要的机器。 默认情况下，安装程序将 control plane 和 compute 机器分布到一个区域中的所有可用区。要确保集群的高可用性，请选择至少含有三个可用区的区域。如果您的区域包含的可用区少于三个，安装程序将在可用区中放置多台 control plane 机器。
VNet	1	每个区域 1000 个	每个默认集群都需要一个虚拟网络 (VNet)，此网络包括两个子网。
网络接口	6	每个区域 65,536 个	每个默认集群都需要六个网络接口。如果您要创建更多机器或者您部署的工作负载要创建负载均衡器，则集群会使用更多的网络接口。
网络安全组	2	5000	每个默认集群为 VNet 中的每个子网创建网络安全组。默认集群为 control plane 和计算节点子网创建网络安全组： **controlplane**：允许从任何位置通过端口 6443 访问 control plane 机器 **node**：允许从互联网通过端口 80 和 443 访问 worker 节点

组件	默认所需的组件数	默认 Azure 限值	描述
网络负载均衡器	3	每个区域 1000 个	每个集群都会创建以下负载均衡器： **default** 用于在 worker 机器之间对端口 80 和 443 的请求进行负载均衡的公共 IP 地址 **internal** 用于在 control plane 机器之间对端口 6443 和 22623 的请求进行负载均衡的专用 IP 地址 **external** 用于在 control plane 机器之间对端口 6443 的请求进行负载均衡的公共 IP 地址 如果您的应用程序创建了更多的 Kubernetes **LoadBalancer** 服务对象，您的集群会使用更多的负载均衡器。
公共 IP 地址	3		两个公共负载均衡器各自使用一个公共 IP 地址。bootstrap 机器也使用一个公共 IP 地址，以便您可以在安装期间通过 SSH 连接到该机器来进行故障排除。bootstrap 节点的 IP 地址仅在安装过程中使用。
专用 IP 地址	7		内部负载均衡器、三台 control plane 机器中的每一台以及三台 worker 机器中的每一台各自使用一个专用 IP 地址。

1.1.2. 在 Azure 中配置公共 DNS 区

要安装 OpenShift Container Platform，您使用的 Microsoft Azure 帐户必须在帐户中具有一个专用的公共托管 DNS 区。此区域必须对域具有权威。此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 Azure 或其他来源获取新的域和注册商。

注意

如需通过 Azure 购买域的更多信息，请参阅 Azure 文档中的购买 Azure 应用服务的自定义域名。

2. 如果您使用现有的域和注册商，请将其 DNS 迁移到 Azure。请参阅 Azure 文档中的将活动 DNS 名称迁移到 Azure 应用服务。

3. 为您的域配置 DNS。按照 Azure 文档中教程：在 Azure DNS 中托管域部分里的步骤，为您的域或子域创建一个公共托管区，提取新的权威名称服务器，并更新您的域使用的名称服务器的注册商记录。
使用合适的根域（如 **openshiftcorp.com**）或子域（如 **clusters.openshiftcorp.com**）。

4. 如果您使用子域，请按照您公司的流程将其委派记录添加到父域。

1.1.3. 提高 Azure 帐户限值

要提高帐户限值，请在 Azure 门户上提交支持请求。

注意

每一支持请求只能提高一种类型的配额。

流程

1. 从 Azure 门户，点击左下角的 **Help + suport**。
2. 点击 **New support request**，然后选择所需的值：
   a. 从 **Issue type** 列表中，选择 **Service and subscription limits (quotas)**。
   b. 从 **Subscription** 列表中，选择要修改的订阅。
   c. 从 **Quota type** 列表中，选择要提高的配额。例如，选择 **Compute-VM (cores-vCPUs) subscription limit increases** 以增加 vCPU 的数量，这是安装集群所必须的。
   d. 点击 **Next: Solutions**。
3. 在 **Problem Details** 页面中，提供您要提高配额所需的信息：
   a. 点击 **Provide details**，然后在 **Quota details** 窗口中提供所需的详情。
   b. 在 SUPPORT METHOD 和 CONTACT INFO 部分中，提供问题严重性和您的联系详情。
4. 点击 **Next: Review + create**，然后点击 **Create**。

1.1.4. 所需的 Azure 角色

Microsoft Azure 帐户必须具有您所用订阅的以下角色：

- **User Access Administrator**

要在 Azure 门户上设置角色，请参阅 Azure 文档中的使用 RBAC 和 Azure 门户管理对 Azure 资源的访问。

1.1.5. 创建服务主体

由于 OpenShift Container Platform 及其安装程序必须通过 Azure Resource Manager 创建 Microsoft Azure 资源，因此您必须创建一个能代表它的服务主体。

先决条件

- 安装或更新 Azure CLI。
- 安装**jq**软件包。
- 您的 Azure 帐户具有您所用订阅所需的角色。

流程

1. 登录 Azure CLI：

```
$ az login
```

在 Web 控制台中，使用您的凭证登录 Azure。

2. 如果您的 Azure 帐户使用订阅，请确保使用正确的订阅。

a. 查看可用帐户列表并记录您要用于集群的订阅的 **tenantId** 值：

```
$ az account list --refresh
[
  {
    "cloudName": "AzureCloud",
    "id": "9bab1460-96d5-40b3-a78e-17b15e978a80",
    "isDefault": true,
    "name": "Subscription Name",
    "state": "Enabled",
    "tenantId": "6057c7e9-b3ae-489d-a54e-de3f6bf6a8ee",
    "user": {
      "name": "you@example.com",
      "type": "user"
    }
  }
]
```

b. 查看您的活跃帐户详情，确认 **tenantId** 值与您要使用的订阅匹配：

```
$ az account show
{
  "environmentName": "AzureCloud",
  "id": "9bab1460-96d5-40b3-a78e-17b15e978a80",
  "isDefault": true,
  "name": "Subscription Name",
  "state": "Enabled",
  "tenantId": "6057c7e9-b3ae-489d-a54e-de3f6bf6a8ee", (1)
  "user": {
    "name": "you@example.com",
    "type": "user"
  }
}
```

(1) 确定 **tenantId** 参数的值是正确订阅的 UUID。

c. 如果您使用的订阅不正确，请更改活跃的订阅：

```
$ az account set -s <id> (1)
```

(1) 替换您要用于 **<id>** 的订阅的 **id** 值。

d. 如果您更改了活跃订阅，请重新显示您的帐户信息：

```
-
```

```
$ az account show

{
  "environmentName": "AzureCloud",
  "id": "33212d16-bdf6-45cb-b038-f6565b61edda",
  "isDefault": true,
  "name": "Subscription Name",
  "state": "Enabled",
  "tenantId": "8049c7e9-c3de-762d-a54e-dc3f6be6a7ee",
  "user": {
    "name": "you@example.com",
    "type": "user"
  }
}
```

3. 记录前面输出中 **tenantId** 和 **id** 参数的值。OpenShift Container Platform 安装过程中需要这些值。

4. 为您的帐户创建服务主体：

```
$ az ad sp create-for-rbac --role Contributor --name <service_principal> (1)
Changing "<service_principal>" to a valid URI of "http://<service_principal>", which is the
required format used for service principal names
Retrying role assignment creation: 1/36
Retrying role assignment creation: 2/36
Retrying role assignment creation: 3/36
Retrying role assignment creation: 4/36
{
  "appId": "8bd0d04d-0ac2-43a8-928d-705c598c6956",
  "displayName": "<service_principal>",
  "name": "http://<service_principal>",
  "password": "ac461d78-bf4b-4387-ad16-7e32e328aec6",
  "tenant": "6048c7e9-b2ad-488d-a54e-dc3f6be6a7ee"
}
```

(1) 将 **<service_principal>** 替换要您要分配给服务主体的名称。

5. 记录前面输出中 **appId** 和 **password** 参数的值。OpenShift Container Platform 安装过程中需要这些值。

6. 为服务主体授予额外权限。服务主体需要传统的 **Azure Active Directory Graph → Application.ReadWrite.OwnedBy** 权限以及集群的 **User Access Administrator** 角色，以便为其组件分配凭证。

   a. 要分配 **User Access Administrator** 角色，请运行以下命令：

```
$ az role assignment create --role "User Access Administrator" \
    --assignee-object-id $(az ad sp list --filter "appId eq '<appId>'" \ (1)
      | jq '.[0].objectId' -r)
```

(1) 将 **<appId>** 替换为服务器主体的 **appId** 参数值。

   b. 要分配 **Azure Active Directory Graph** 权限，请运行以下命令：

```
$ az ad app permission add --id <appId> \ 1
    --api 00000002-0000-0000-c000-000000000000 \
    --api-permissions 824c81eb-e3f8-4ee6-8f6d-de7f50d565b7=Role

    Invoking "az ad app permission grant --id 46d33abc-b8a3-46d8-8c84-f0fd58177435 --
api 00000002-0000-0000-c000-000000000000" is needed to make the change effective
```

1 将 **<appId>** 替换为服务器主体的 **appId** 参数值。

如需进一步了解可通过此命令授予的具体权限，请参阅 Windows Azure Active Directory 权限的 GUID 表。

c. 批准权限请求。如果您的帐户没有 Azure Active Directory 租户管理员角色，请按照您的组织的准则请租户管理员批准您的权限请求。

```
$ az ad app permission grant --id <appId> \ 1
    --api 00000002-0000-0000-c000-000000000000
```

1 将 **<appId>** 替换为服务器主体的 **appId** 参数值。

1.1.6. 支持的 Azure 区域

安装程序会根据您的订阅动态地生成可用的 Microsoft Azure 区域列表。OpenShift Container Platform 4.4.0 中已测试并验证了以下 Azure 区域：

- **australiacentral** (Australia Central)
- **australiaeast** (Australia East)
- **australiasoutheast** (Australia South East)
- **brazilsouth** (Brazil South)
- **canadacentral** (Canada Central)
- **canadaeast** (Canada East)
- **centralindia** (Central India)
- **centralus** (Central US)
- **eastasia** (East Asia)
- **eastus** (East US)
- **eastus2** (East US 2)
- **francecentral** (France Central)
- **germanywestcentral** (Germany West Central)
- **japaneast** (Japan East)
- **japanwest** (Japan West)

- **koreacentral** (Korea Central)
- **koreasouth** (Korea South)
- **northcentralus** (North Central US)
- **northeurope** (North Europe)
- **southafricanorth** (South Africa North)
- **southcentralus** (South Central US)
- **southeastasia** (Southeast Asia)
- **southindia** (South India)
- **switzerlandnorth** (Switzerland North)
- **uaenorth** (UAE North)
- **uksouth** (UK South)
- **ukwest** (UK West)
- **westcentralus** (West Central US)
- **westeurope** (West Europe)
- **westindia** (West India)
- **westus** (West US)
- **westus2** (West US 2)

1.1.7. 后续步骤

- 在 Azure 上安装 OpenShift Container Platform 集群。您可以安装自定义集群，或使用默认选项快速安装集群。

1.2. 在 AZURE 上快速安装集群

在 OpenShift Container Platform 版本 4.4 中，您可以使用默认配置选项在 Microsoft Azure 上安装集群。

1.2.1. 先决条件

- 查看有关 OpenShift Container Platform 安装和更新流程的详细信息。
- 配置一个 Azure 帐户以托管集群，并决定要将集群部署到的已测试和验证的区域。
- 如果使用防火墙，则必须将其配置为允许集群需要访问的站点。
- 如果不允许系统管理身份和访问管理（IAM），集群管理员可以手动创建和维护 IAM 凭证。手动模式也可以用于云 IAM API 无法访问的环境中。

1.2.2. OpenShift Container Platform 对互联网和 Telemetry 的访问

在 OpenShift Container Platform 4.4 中，您需要访问互联网来安装集群。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，这也需要访问互联网。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 Red Hat OpenShift Cluster Manager （OCM）。

确认 Red Hat OpenShift Cluster Manager 清单正确后，可以由 Telemetry 自动维护，也可以使用 OCM 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。

您必须具有以下互联网访问权限：

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

重要

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

1.2.3. 生成 SSH 私钥并将其添加到代理中

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

注意

在生产环境中，您需要进行灾难恢复和调试。

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

注意

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

流程

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

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

1 指定 SSH 密钥的路径和文件名，如 **~/.ssh/id_rsa**。不要指定已存在的 SSH 密钥，因为它会被覆盖。

运行此命令会在指定的位置生成不需要密码的 SSH 密钥。

2. 作为后台任务启动 **ssh-agent** 进程：

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

Agent pid 31874
```

3. 将 SSH 私钥添加到 **ssh-agent**：

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

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

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

后续步骤

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

1.2.4. 获取安装程序

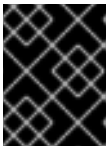
在安装 OpenShift Container Platform 之前，将安装文件下载到本地计算机上。

先决条件

- 必须从使用 Linux 或 macOS 的计算机安装集群。
- 需要 500 MB 本地磁盘空间来下载安装程序。

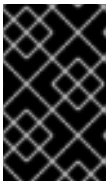
流程

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



重要

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



重要

删除安装程序创建的文件不会删除您的集群，即使集群在安装过程中失败也是如此。您必须完成针对特定云供应商的 OpenShift Container Platform 卸载流程，才能完全删除您的集群。

3. 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 <installation_program>.tar.gz
```

4. 在 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中，下载您的安装 pull secret 的 **.txt** 文件。通过此 pull secret，您可以进行所含授权机构提供的服务的身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

1.2.5. 部署集群

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

重要

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

先决条件

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

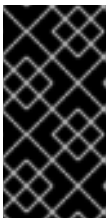
流程

1. 运行安装程序：

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

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

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



重要

指定一个空目录。一些安装信息，如 bootstrap X.509 证书，有较短的过期间隔，因此不要重复使用安装目录。如果要重复使用另一个集群安装中的个别文件，可以将其复制到您的目录中。但是，一些安装数据的文件名可能会在发行版本之间有所改变。从 OpenShift Container Platform 老版本中复制安装文件时要格外小心。

在提示符处提供值：

a. 可选：选择用来访问集群机器的 SSH 密钥。

注意

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

b. 选择 **azure** 作为目标平台。

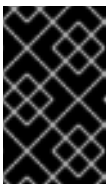
c. 如果计算机上没有 Microsoft Azure 配置集，请为您的订阅和服务主体指定以下 Azure 参数值：

- **azure subscription id**：要用于集群的订阅 ID。指定帐户输出中的 **id** 值。
- **azure tenant id**：租户 ID。指定帐户输出中的 **tenantId** 值。
- **azure service principal client id**：服务主体的 **appId** 参数值。
- **azure service principal client secret**：服务主体的 **password** 参数值。

d. 选择要在其中部署集群的区域。

e. 选择集群要部署到的基域。基域与您为集群创建的 Azure DNS 区对应。

f. 为集群输入一个描述性名称。



重要

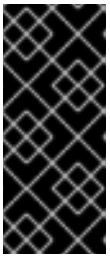
所有通过公共端点提供的 Azure 资源均存在资源名称的限制，您无法创建使用某些名称的资源。如需 Azure 限制词语列表，请参阅 Azure 文档中的解决保留资源名称错误。

g. 粘贴从 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中获取的 pull secret。

注意

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

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



重要

安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrapper** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅*从过期的 control plane 证书中恢复*的文档。

重要

您不得删除安装程序或安装程序所创建的文件。需要这两者才能删除集群。

1.2.6. 通过下载二进制文件安装 CLI

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

重要

如果安装了旧版本的 **oc**，则无法使用 OpenShift Container Platform 4.4 中的所有命令。下载并安装新版本的 **oc**。

1.2.6.1. 在 Linux 上安装 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. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。

3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Linux**，并点 **Download command-line tools**。
4. 解包存档：

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

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

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

1.2.6.2. 在 Windows 上安装 CLI

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

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Windows**，点 **Download command-line tools**。
4. 使用 ZIP 程序解压存档。
5. 把 **oc** 二进制代码放到 **PATH** 中的目录中。
要查看您的 **PATH**，请打开命令提示窗口并执行以下命令：

```
C:\> path
```

安装 CLI 后，就可以使用**oc**命令：

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

1.2.6.3. 在 macOS 上安装 CLI

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

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **MacOS**，并点 **Download command-line tools**。

4. 解包和解压存档。

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

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

1.2.7. 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** 文件包含关于集群的信息，供 CLI 用于将客户端连接到正确集群和 API 服务器。该文件特只适用于一个特定的集群，在 OpenShift Container Platform 安装过程中创建。

先决条件

- 部署一个 OpenShift Container Platform 集群。
- 安装 **oc** CLI。

流程

1. 导出 **kubeadmin** 凭证：

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

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

2. 使用导出的配置，验证能否成功运行 **oc** 命令：

```
$ oc whoami
system:admin
```

1.2.8. 后续步骤

- 自定义集群。
- 若有需要，您可以选择不使用远程健康报告。

1.3. 使用自定义在 AZURE 上安装集群

在 OpenShift Container Platform 版本 4.4 中，您可以在安装程序在 Microsoft Azure 上置备的基础架构上安装自定义的集群。要自定义安装，请在安装集群前修改 **install-config.yaml** 文件中的参数。

1.3.1. 先决条件

- 查看有关 OpenShift Container Platform 安装和更新 流程的详细信息。

- 配置一个 Azure 帐户以托管集群，并决定要将集群部署到的已测试和验证的区域。
- 如果使用防火墙，则必须将其配置为允许集群需要访问的站点。
- 如果不允许系统管理身份和访问管理（IAM），集群管理员可以手动创建和维护 IAM 凭证。手动模式也可以用于云 IAM API 无法访问的环境中。

1.3.2. OpenShift Container Platform 对互联网和 Telemetry 的访问

在 OpenShift Container Platform 4.4 中，您需要访问互联网来安装集群。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，这也需要访问互联网。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 Red Hat OpenShift Cluster Manager （OCM）。

确认 Red Hat OpenShift Cluster Manager 清单正确后，可以由 Telemetry 自动维护，也可以使用 OCM 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。

您必须具有以下互联网访问权限：

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

重要

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

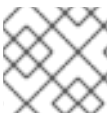
1.3.3. 生成 SSH 私钥并将其添加到代理中

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

注意

在生产环境中，您需要进行灾难恢复和调试。

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



注意

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

流程

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

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

1 指定 SSH 密钥的路径和文件名，如 **~/.ssh/id_rsa**。不要指定已存在的 SSH 密钥，因为它会被覆盖。

运行此命令会在指定的位置生成不需要密码的 SSH 密钥。

2. 作为后台任务启动 **ssh-agent** 进程：

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

Agent pid 31874
```

3. 将 SSH 私钥添加到 **ssh-agent**：

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

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

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

后续步骤

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

1.3.4. 获取安装程序

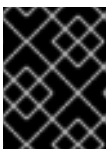
在安装 OpenShift Container Platform 之前，将安装文件下载到本地计算机上。

先决条件

- 必须从使用 Linux 或 macOS 的计算机安装集群。
- 需要 500 MB 本地磁盘空间来下载安装程序。

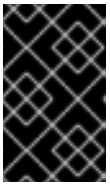
流程

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



重要

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



重要

删除安装程序创建的文件不会删除您的集群，即使集群在安装过程中失败也是如此。您必须完成针对特定云供应商的 OpenShift Container Platform 卸载流程，才能完全删除您的集群。

3. 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 <installation_program>.tar.gz
```

4. 在 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中，下载您的安装 pull secret 的 **.txt** 文件。通过此 pull secret，您可以进行所含授权机构提供的服务的身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

1.3.5. 创建安装配置文件

您可以自定义在 Microsoft Azure 上安装的 OpenShift Container Platform 集群。

先决条件

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

流程

1. 创建 **install-config.yaml** 文件。
   a. 运行以下命令：

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

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

重要

指定一个空目录。一些安装信息，如 bootstrap X.509 证书，有较短的过期间隔，因此不要重复使用安装目录。如果要重复使用另一个集群安装中的个别文件，可以将其复制到您的目录中。但是，一些安装数据的文件名可能会在发行版本之间有所改变。从 OpenShift Container Platform 老版本中复制安装文件时要格外小心。

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

注意

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

      ii. 选择 **azure** 作为目标平台。

iii. 如果计算机上没有 Microsoft Azure 配置集，请为您的订阅和服务主体指定以下 Azure 参数值：

- **azure subscription id**：要用于集群的订阅 ID。指定帐户输出中的 **id** 值。
- **azure tenant id**：租户 ID。指定帐户输出中的 **tenantId** 值。
- **azure service principal client id**：服务主体的 **appId** 参数值。
- **azure service principal client secret**：服务主体的 **password** 参数值。

iv. 选择要在其中部署集群的区域。

v. 选择集群要部署到的基域。基域与您为集群创建的 Azure DNS 区对应。

vi. 为集群输入一个描述性名称。

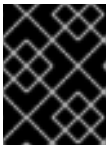
重要

所有通过公共端点提供的 Azure 资源均存在资源名称的限制，您无法创建使用某些名称的资源。如需 Azure 限制词语列表，请参阅 Azure 文档中的解决保留资源名称错误。

vii. 粘贴从 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中获取的 pull secret。

2. 修改 **install-config.yaml** 文件。您可以在**安装配置参数**部分中找到有关可用参数的更多信息。

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



重要

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

1.3.5.1. 安装配置参数

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

注意

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

表 1.1. 所需的参数

参数	描述	值

参数	描述	值
baseDomain	云供应商的基域。此值用于创建到 OpenShift Container Platform 集群组件的路由。集群的完整 DNS 名称是 **baseDomain** 和 **metadata.name** 参数值的组合，其格式为 **<metadata.name>.<baseDomain>**。	完全限定域名或子域名，如 **example.com**。
controlPlane.platform	托管 control plane 机器的云供应商。此参数值必须与 **compute.platform** 参数值匹配。	**aws**、**azure**、**gcp**、**openstack** 或 **{}**
compute.platform	托管 worker 机器的云供应商。此参数值必须与 **controlPlane.platform** 参数值匹配。	**aws**、**azure**、**gcp**、**openstack** 或 **{}**
metadata.name	集群的名称。	包含大写字母或小写字母的字符串，如 **dev**。
platform.<platform>.region	集群要部署到的区域。	云的有效区域，如 AWS 的 **us-east-1**、Azure 的 **centralus**。Red Hat OpenStack Platform (RHOSP) 不使用这个参数

pullSecret	从 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中获取的 pull secret。您可以使用此 pull secret 来进行所含授权机构提供的服务的身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。	{
		"auths":{
		"cloud.openshift.com":{
		"auth":"b3Blb=",
		"email":"you@example.com"
		},
		"quay.io":{
		"auth":"b3Blb=",
		"email":"you@example.com"
		}
		}
		}

表 1.2. 可选参数

参数	描述	值

参数	描述	值
sshKey	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。	添加到 **ssh-agent** 进程的有效本地公共 SSH 密钥。
fips	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。	**false** 或 **true**
publish	如何发布集群的面向用户的端点。	**Internal** 或 **External**。把 **publish** 设置为 **Internal** 以部署一个私有集群，它不能被互联网访问。默认值为 **External**。
compute.hyperthreading	是否在计算机器上启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。  **重要** 如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。	**Enabled** 或 **Disabled**
compute.replicas	要置备的计算机器数量，也称为 worker 机器。	大于或等于 **2** 的正整数。默认值为 **3**。

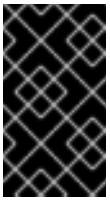
参数	描述	值
controlPlane.hyperthreading	是否在 control plane 机器上启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。  **重要** 如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。	**Enabled** 或 **Disabled**
controlPlane.replicas	要置备的 control plane 机器数量。	唯一支持的值是 **3**，它是默认值。

表 1.3. 其他 Azure 参数

参数	描述	值
machines.platform.azure.type	Azure 虚拟机实例类型。	使用 Windows 或 Linux 作为操作系统的虚拟机。请参阅 Azure 文档中的 Azure Stack 上支持的客户端操作系统。
machines.platform.azure.osDisk.diskSizeGB	虚拟机的 Azure 磁盘大小。	以 GB 为单位表示磁盘大小的整数，如 **512**。支持的最小磁盘大小为 **120**。
platform.azure.baseDomainResourceGroupName	包含基域的 DNS 区的资源组的名称。	字符串，如 **production_cluster**。
platform.azure.region	托管集群的 Azure 区域名称。	任何有效的区域名称。
platform.azure.zone	可在其中放入机器的可用区的列表。如需高可用性，请至少指定两个区域。	区域列表，如 **["1", "2", "3"]**。
platform.azure.networkResourceGroupName	包含要将集群部署到的现有 VNet 的资源组名称。这个名称不能和 **platform.azure.baseDomainResourceGroupName**相同。	字符串。
platform.azure.virtualNetwork	要将集群部署到的现有 VNet 的名称。	字符串。
platform.azure.controlPlaneSubnet	要将 control plane 机器部署到的 VNet 中现有子网的名称。	有效的 CIDR，如 **10.0.0.0/16**。

参数	描述	值
platform.azure.computeSubnet	您要将计算机器部署到的 VNet 中现有子网的名称。	有效的 CIDR，如 **10.0.0.0/16**。

注意

您无法自定义 Azure 可用区，也不能使用标签来整理用于 Azure 集群的 Azure 资源。

1.3.5.2. Azure 的自定义 install-config.yaml 文件示例

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

重要

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

```
apiVersion: v1
baseDomain: example.com (1)
controlPlane: (2)
  hyperthreading: Enabled (3) (4)
  name: master
  platform:
    azure:
      osDisk:
        diskSizeGB: 1024 (5)
      type: Standard_D8s_v3
  replicas: 3
compute: (6)
- hyperthreading: Enabled (7)
  name: worker
  platform:
    azure:
      type: Standard_D2s_v3
      osDisk:
        diskSizeGB: 512 (8)
      zones: (9)
      - "1"
      - "2"
      - "3"
  replicas: 5
metadata:
  name: test-cluster (10)
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
```

```
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  azure:
    region: centralus 11
    baseDomainResourceGroupName: resource_group 12
pullSecret: '{"auths": ...}' 13
fips: false 14
sshKey: ssh-ed25519 AAAA... 15
```

1 10 11 13 必需。安装程序会提示您输入这个值。

2 6 如果没有提供这些参数和值，安装程序会提供默认值。

3 7 **controlPlane** 部分是一个单映射，但 compute 部分是一系列映射。为满足不同数据结构的要求，**compute** 部分的第一行必须以连字符 **-** 开头，**controlPlane** 部分的第一行则不可以连字符开头。虽然这两个部分目前都定义单个机器池，但未来的 OpenShift Container Platform 版本可能会支持在安装过程中定义多个计算池。只使用一个 control plane 池。

4 是否要启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。您可以通过将参数值设为 **Disabled** 来禁用。如果您在某些集群机器上禁用并发多线程，则必须在所有集群机器上禁用。

重要

如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。如果禁用并发多线程，请使用较大的虚拟机类型，如 **Standard_D8s_v3**。

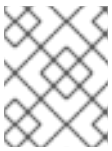
5 8 可以 GB 为单位指定要使用的磁盘大小。master 节点的最低推荐值为 1024 GB。

9 指定要将机器部署到的区域列表。如需高可用性，请至少指定两个区域。

12 指定包含基域的 DNS 区的资源组的名称。

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

15 您可以选择提供您用来访问集群中机器的 **sshKey** 值。

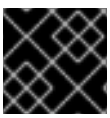


注意

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

1.3.6. 部署集群

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



重要

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

先决条件

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

流程

1. 运行安装程序：

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

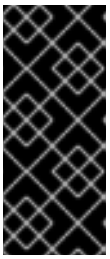
1 对于 **<installation_directory>**，请指定自定义 **./install-config.yaml** 文件的位置。

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

注意

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

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



重要

安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrapper** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅 *从过期的 control plane 证书中恢复*的文档。

重要

您不得删除安装程序或安装程序所创建的文件。需要这两者才能删除集群。

1.3.7. 通过下载二进制文件安装 CLI

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

重要

如果安装了旧版本的 **oc**，则无法使用 OpenShift Container Platform 4.4 中的所有命令。下载并安装新版本的 **oc**。

1.3.7.1. 在 Linux 上安装 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. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Linux**，并点 **Download command-line tools**。
4. 解包存档：

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

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

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

1.3.7.2. 在 Windows 上安装 CLI

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

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Windows**，点 **Download command-line tools**。
4. 使用 ZIP 程序解压存档。
5. 把 **oc** 二进制代码放到 **PATH** 中的目录中。
   要查看您的 **PATH**，请打开命令提示窗口并执行以下命令：

```
C:\> path
```

安装 CLI 后，就可以使用**oc**命令：

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

1.3.7.3. 在 macOS 上安装 CLI

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

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。

3. 在 **Command-line interface** 部分，从下拉菜单中选择 **MacOS**，并点 **Download command-line tools**。
4. 解包和解压存档。
5. 将 **oc** 二进制文件移到 PATH 的目录中。
   要查看您的 **PATH**，打开一个终端窗口并执行以下命令：

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

1.3.8. 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** 文件包含关于集群的信息，供 CLI 用于将客户端连接到正确集群和 API 服务器。该文件特只适用于一个特定的集群，在 OpenShift Container Platform 安装过程中创建。

先决条件

- 部署一个 OpenShift Container Platform 集群。
- 安装 **oc** CLI。

流程

1. 导出 **kubeadmin** 凭证：

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

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

2. 使用导出的配置，验证能否成功运行 **oc** 命令：

```
$ oc whoami
system:admin
```

1.3.9. 后续步骤

- 自定义集群。
- 若有需要，您可以选择不使用远程健康报告。

1.4. 使用网络自定义在 AZURE 上安装集群

在 OpenShift Container Platform 版本 4.4 中，您可以使用自定义的网络配置在安装程序在 Microsoft Azure 上置备的基础架构上安装集群。通过自定义网络配置，您的集群可以与环境中现有的 IP 地址分配共存，并与现有的 MTU 和 VXLAN 配置集成。

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

1.4.1. 先决条件

- 查看有关 OpenShift Container Platform 安装和更新流程的详细信息。
- 配置一个 Azure 帐户以托管集群，并决定要将集群部署到的已测试和验证的区域。
- 如果使用防火墙，则必须将其配置为允许集群需要访问的站点。
- 如果不允许系统管理身份和访问管理（IAM），集群管理员可以手动创建和维护 IAM 凭证。手动模式也可以用于云 IAM API 无法访问的环境中。

1.4.2. OpenShift Container Platform 对互联网和 Telemetry 的访问

在 OpenShift Container Platform 4.4 中，您需要访问互联网来安装集群。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，这也需要访问互联网。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 Red Hat OpenShift Cluster Manager （OCM）。

确认 Red Hat OpenShift Cluster Manager 清单正确后，可以由 Telemetry 自动维护，也可以使用 OCM 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。

您必须具有以下互联网访问权限：

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

重要

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

1.4.3. 生成 SSH 私钥并将其添加到代理中

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

注意

在生产环境中，您需要进行灾难恢复和调试。

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

注意

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

流程

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

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

1 指定 SSH 密钥的路径和文件名，如 **~/.ssh/id_rsa**。不要指定已存在的 SSH 密钥，因为它会被覆盖。

运行此命令会在指定的位置生成不需要密码的 SSH 密钥。

2. 作为后台任务启动 **ssh-agent** 进程：

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

Agent pid 31874
```

3. 将 SSH 私钥添加到 **ssh-agent**：

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

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

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

后续步骤

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

1.4.4. 获取安装程序

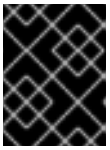
在安装 OpenShift Container Platform 之前，将安装文件下载到本地计算机上。

先决条件

- 必须从使用 Linux 或 macOS 的计算机安装集群。
- 需要 500 MB 本地磁盘空间来下载安装程序。

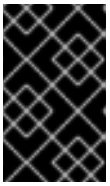
流程

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



重要

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



重要

删除安装程序创建的文件不会删除您的集群，即使集群在安装过程中失败也是如此。您必须完成针对特定云供应商的 OpenShift Container Platform 卸载流程，才能完全删除您的集群。

3. 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 <installation_program>.tar.gz
```

4. 在 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中，下载您的安装 pull secret 的 **.txt** 文件。通过此 pull secret，您可以进行所含授权机构提供的服务的身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

1.4.5. 创建安装配置文件

您可以自定义在 Microsoft Azure 上安装的 OpenShift Container Platform 集群。

先决条件

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

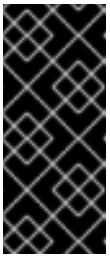
流程

1. 创建 **install-config.yaml** 文件。

   a. 运行以下命令：

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

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



重要

指定一个空目录。一些安装信息，如 bootstrap X.509 证书，有较短的过期间隔，因此不要重复使用安装目录。如果要重复使用另一个集群安装中的个别文件，可以将其复制到您的目录中。但是，一些安装数据的文件名可能会在发行版本之间有所改变。从 OpenShift Container Platform 老版本中复制安装文件时要格外小心。

   b. 在提示符处，提供您的云的配置详情：

      i. 可选：选择用来访问集群机器的 SSH 密钥。

注意

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

      ii. 选择 **azure** 作为目标平台。

iii. 如果计算机上没有 Microsoft Azure 配置集，请为您的订阅和服务主体指定以下 Azure 参数值：

- **azure subscription id**：要用于集群的订阅 ID。指定帐户输出中的 **id** 值。
- **azure tenant id**：租户 ID。指定帐户输出中的 **tenantId** 值。
- **azure service principal client id**：服务主体的 **appId** 参数值。
- **azure service principal client secret**：服务主体的 **password** 参数值。

iv. 选择要在其中部署集群的区域。

v. 选择集群要部署到的基域。基域与您为集群创建的 Azure DNS 区对应。

vi. 为集群输入一个描述性名称。

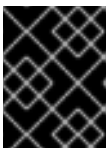
重要

所有通过公共端点提供的 Azure 资源均存在资源名称的限制，您无法创建使用某些名称的资源。如需 Azure 限制词语列表，请参阅 Azure 文档中的解决保留资源名称错误。

vii. 粘贴从 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中获取的 pull secret。

2. 修改 **install-config.yaml** 文件。您可以在**安装配置参数**部分中找到有关可用参数的更多信息。

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



重要

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

1.4.5.1. 安装配置参数

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

注意

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

表 1.4. 所需的参数

参数	描述	值

参数	描述	值
baseDomain	云供应商的基域。此值用于创建到 OpenShift Container Platform 集群组件的路由。集群的完整 DNS 名称是 **baseDomain** 和 **metadata.name** 参数值的组合，其格式为 **<metadata.name>.<baseDomain>**。	完全限定域名或子域名，如 **example.com**。
controlPlane.platform	托管 control plane 机器的云供应商。此参数值必须与 **compute.platform** 参数值匹配。	**aws**、**azure**、**gcp**、**openstack** 或 **{}**
compute.platform	托管 worker 机器的云供应商。此参数值必须与 **controlPlane.platform** 参数值匹配。	**aws**、**azure**、**gcp**、**openstack** 或 **{}**
metadata.name	集群的名称。	包含大写字母或小写字母的字符串，如 **dev**。
platform.<platform>.region	集群要部署到的区域。	云的有效区域，如 AWS 的 **us-east-1**、Azure 的 **centralus**。Red Hat OpenStack Platform (RHOSP) 不使用这个参数

pullSecret	从 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中获取的 pull secret。您可以使用此 pull secret 来进行所含授权机构提供的服务的身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。	{
		"auths":{
		"cloud.openshift.com":{
		"auth":"b3Blb=",
		"email":"you@example.com"
		},
		"quay.io":{
		"auth":"b3Blb=",
		"email":"you@example.com"
		}
		}
		}

表 1.5. 可选参数

参数	描述	值

参数	描述	值
sshKey	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。	添加到 **ssh-agent** 进程的有效本地公共 SSH 密钥。
fips	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。	**false** 或 **true**
publish	如何发布集群的面向用户的端点。	**Internal** 或 **External**。把 **publish** 设置为 **Internal** 以部署一个私有集群，它不能被互联网访问。默认值为 **External**。
compute.hyperthreading	是否在计算机器上启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。  **重要** 如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。	**Enabled** 或 **Disabled**
compute.replicas	要置备的计算机器数量，也称为 worker 机器。	大于或等于 **2** 的正整数。默认值为 **3**。

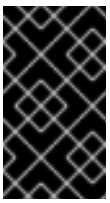
参数	描述	值
controlPlane.hyperthreading	是否在 control plane 机器上启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。  **重要** 如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。	**Enabled** 或 **Disabled**
controlPlane.replicas	要置备的 control plane 机器数量。	唯一支持的值是 **3**，它是默认值。

表 1.6. 其他 Azure 参数

参数	描述	值
machines.platform.azure.type	Azure 虚拟机实例类型。	使用 Windows 或 Linux 作为操作系统的虚拟机。请参阅 Azure 文档中的 Azure Stack 上支持的客户端操作系统。
machines.platform.azure.osDisk.diskSizeGB	虚拟机的 Azure 磁盘大小。	以 GB 为单位表示磁盘大小的整数，如 **512**。支持的最小磁盘大小为 **120**。
platform.azure.baseDomainResourceGroupName	包含基域的 DNS 区的资源组的名称。	字符串，如 **production_cluster**。
platform.azure.region	托管集群的 Azure 区域名称。	任何有效的区域名称。
platform.azure.zone	可在其中放入机器的可用区的列表。如需高可用性，请至少指定两个区域。	区域列表，如 **["1", "2", "3"]**。
platform.azure.networkResourceGroupName	包含要将集群部署到的现有 VNet 的资源组名称。这个名称不能和 **platform.azure.baseDomainResourceGroupName**相同。	字符串。
platform.azure.virtualNetwork	要将集群部署到的现有 VNet 的名称。	字符串。
platform.azure.controlPlaneSubnet	要将 control plane 机器部署到的 VNet 中现有子网的名称。	有效的 CIDR，如 **10.0.0.0/16**。

参数	描述	值
platform.azure.computeSubnet	您要将计算机器部署到的 VNet 中现有子网的名称。	有效的 CIDR，如 **10.0.0.0/16**。

注意

您无法自定义 Azure 可用区，也不能使用标签来整理用于 Azure 集群的 Azure 资源。

重要

Open Virtual Networking (OVN) Kubernetes 网络插件只是技术预览功能。技术预览功能不被红帽产品服务等级协议 (SLA) 支持，且可能在功能方面有缺陷。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。

如需 OVN 技术预览支持范围的更多信息，请参阅 https://access.redhat.com/articles/4380121。

1.4.5.2. 网络配置参数

您可以在 **install-config.yaml** 配置文件中修改集群网络配置参数。下表描述了这些参数。

注意

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

表 1.7. 所需的网络参数

参数	描述	值
networking.networkType	要部署的默认 Container Network Interface(CNI)网络供应商插件。**OpenShiftSDN** 插件是 OpenShift Container Platform 4.4 中唯一支持的插件。**OVNKubernetes** 插件在 OpenShift Container Platform 4.4 中仅以技术预览提供。	**OpenShiftSDN** 或 **OVNKubernetes**。默认值为 **OpenShiftSDN**。
networking.clusterNetwork[].cidr	从中分配 pod IP 地址的 IP 地址块。**OpenShiftSDN** 网络插件支持多个集群网络。多个集群网络的地址块不得互相重叠。请选择足够大的地址池，以适配预期的工作负载。	CIDR 格式的 IP 地址分配。默认值为 **10.128.0.0/14**。
networking.clusterNetwork[].hostPrefix	分配给每个单独节点的子网前缀长度。例如，如果 **hostPrefix** 设为 **23**，则每个节点从所给的 **cidr** 中分配一个 **/23** 子网，这样就能有 510 (2^(32 - 23) - 2) 个 pod IP 地址。	子网前缀。默认值为 **23**。

参数	描述	值
networking.serviceNetwork[]	服务的 IP 地址块。**OpenShiftSDN** 只允许一个 **serviceNetwork** 块。该地址块不得与任何其他网络块重叠。	CIDR 格式的 IP 地址分配。默认值为 **172.30.0.0/16**。
networking.machineNetwork[].cidr	OpenShift Container Platform 安装程序在安装集群时分配给节点的 IP 地址块。该地址块不得与任何其他网络块重叠。可以指定多个 CIDR 范围。	CIDR 格式的 IP 地址分配。默认值为 **10.0.0.0/16**。

1.4.5.3. Azure 的自定义 install-config.yaml 文件示例

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

重要

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

```
apiVersion: v1
baseDomain: example.com (1)
controlPlane: (2)
  hyperthreading: Enabled (3) (4)
  name: master
  platform:
    azure:
      osDisk:
        diskSizeGB: 1024 (5)
      type: Standard_D8s_v3
  replicas: 3
compute: (6)
- hyperthreading: Enabled (7)
  name: worker
  platform:
    azure:
      type: Standard_D2s_v3
      osDisk:
        diskSizeGB: 512 (8)
      zones: (9)
      - "1"
      - "2"
      - "3"
  replicas: 5
metadata:
  name: test-cluster (10)
networking: (11)
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
```

```
  - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  azure:
    region: centralus 12
    baseDomainResourceGroupName: resource_group 13
pullSecret: '{"auths": ...}' 14
fips: false 15
sshKey: ssh-ed25519 AAAA... 16
```

1 10 12 14 必需。安装程序会提示您输入这个值。

2 6 11 如果没有提供这些参数和值，安装程序会提供默认值。

3 7 **controlPlane** 部分是一个单映射，但 compute 部分是一系列映射。为满足不同数据结构的要求，**compute** 部分的第一行必须以连字符 **-** 开头，**controlPlane** 部分的第一行则不可以连字符开头。虽然这两个部分目前都定义单个机器池，但未来的 OpenShift Container Platform 版本可能会支持在安装过程中定义多个计算池。只使用一个 control plane 池。

4 是否要启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。您可以通过将参数值设为 **Disabled** 来禁用。如果您在某些集群机器上禁用并发多线程，则必须在所有集群机器上禁用。

重要

如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。如果禁用并发多线程，请使用较大的虚拟机类型，如 **Standard_D8s_v3**。

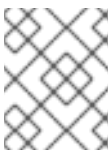
5 8 可以 GB 为单位指定要使用的磁盘大小。master 节点的最低推荐值为 1024 GB。

9 指定要将机器部署到的区域列表。如需高可用性，请至少指定两个区域。

13 指定包含基域的 DNS 区的资源组的名称。

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

16 您可以选择提供您用来访问集群中机器的 **sshKey** 值。



注意

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

1.4.6. 修改高级网络配置参数

您只能在安装集群前修改高级网络配置参数。通过自定义高级配置，您可以指定 MTU 或 VXLAN 端口，允许自定义 kube-proxy 设置，以及为 **openshiftSDNConfig** 参数指定不同的 **mode**，从而将集群整合到现有的网络环境中。

重要

不支持直接修改 OpenShift Container Platform 清单文件。

先决条件

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

流程

1. 使用以下命令来创建清单：

```
$ ./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/** 目录下，创建一个名为 **cluster-network-03-config.yml** 的文件：

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

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

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

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

输**出示例**

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

3. 在编辑器中打开 **cluster-network-03-config.yml** 文件，然后输入描述您想要的 Operator 配置的 CR：

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec: 1
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  serviceNetwork:
  - 172.30.0.0/16
  defaultNetwork:
    type: OpenShiftSDN
    openshiftSDNConfig:
```

```
    mode: NetworkPolicy
    mtu: 1450
    vxlanPort: 4789
```

1 **spec** 参数的参数仅作示例。在 CR 中为 Cluster Network Operator 指定配置。

CNO 为 CR 中的参数提供默认值，因此您必须只指定要更改的参数。

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。

1.4.7. Cluster Network Operator 配置

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

您可以通过在 CNO CR 中设置 **defaultNetwork** 参数的值，为 OpenShift Container Platform 集群指定集群网络配置。以下 CR 显示了 CNO 的默认配置，并列出了您可以配置的参数和有效的参数值：

Cluster Network Operator CR

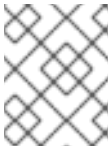
```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  clusterNetwork: 1
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  serviceNetwork: 2
  - 172.30.0.0/16
  defaultNetwork: 3
    ...
  kubeProxyConfig: 4
    iptablesSyncPeriod: 30s 5
    proxyArguments:
      iptables-min-sync-period: 6
      - 0s
```

1 2 在 **install-config.yaml** 文件中指定。

3 配置集群网络的默认 Container Network Interface(CNI)网络供应商。

4 此对象的参数指定 **kube-proxy** 配置。如果没有指定参数值，Cluster Network Operator 会使用显示的默认参数值。如果您使用 OVN-Kubernetes 默认 CNI 网络供应商，则 kube-proxy 的配置不会起作用。

5 **iptables** 规则的刷新周期。默认值为 **30s**。有效的后缀包括 **s**、**m**和 **h**，具体参见 Go 时间包文档。



注意

由于 OpenShift Container Platform 4.3 及更高版本中引进了性能上的改进，现在不再需要调整 **iptablesSyncPeriod** 参数。

6 刷新 **iptables** 规则前的最短时长。此参数确保刷新的频率不会过于频繁。有效的后缀包括 **s**、**m** 和 **h**，具体参见 Go time 软件包。

1.4.7.1. OpenShift SDN 网络供应商的配置参数

以下 YAML 对象描述了 OpenShift SDN 默认 Container Network Interface(CNI)网络供应商的配置参数。

```
defaultNetwork:
  type: OpenShiftSDN 1
  openshiftSDNConfig: 2
    mode: NetworkPolicy 3
    mtu: 1450 4
    vxlanPort: 4789 5
```

1 在 **install-config.yaml** 文件中指定。

2 只有您要覆盖部分 OpenShift SDN 配置时才需要指定。

3 配置 OpenShift SDN 的网络隔离模式。允许的值有 **Multitenant**、**Subnet** 或 **NetworkPolicy**。默认值为 **NetworkPolicy**。

4 VXLAN 覆盖网络的最大传输单元 (MTU) 。这个值通常是自动配置的；但是，如果集群中的节点没有全部使用相同的 MTU，那么您必须将此值明确设置为比最小节点 MTU 的值小 50。

5 用于所有 VXLAN 数据包的端口。默认值为 **4789**。如果您在虚拟环境中运行，并且现有节点是另一个 VXLAN 网络的一部分，那么可能需要更改此值。例如，当在 VMware NSX-T 上运行 OpenShift SDN 覆盖时，您必须为 VXLAN 选择一个备用端口，因为两个 SDN 都使用相同的默认 VXLAN 端口号。

在 Amazon Web Services (AWS) 上，您可以在端口 **9000** 和端口 **9999** 之间为 VXLAN 选择一个备用端口。

1.4.7.2. OVN-Kubernetes 网络供应商的配置参数

以下 YAML 对象描述了 OVN-Kubernetes Pod 网络供应商的配置参数：

```
defaultNetwork:
  type: OVNKubernetes 1
  ovnKubernetesConfig: 2
    mtu: 1400 3
    genevePort: 6081 4
```

1 在 **install-config.yaml** 文件中指定。

2 只有在需要覆盖部分 OVN-Kubernetes 配置时才需要指定。

3 Geneve（Generic Network Virtualization Encapsulation）覆盖网络的 MTU。这个值通常是自动配

4 Geneve 覆盖网络的 UDP 端口。

1.4.7.3. Cluster Network Operator 配置示例

下例中显示了 CNO 的一个完整 CR：

Cluster Network Operator 示例 CR

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  serviceNetwork:
  - 172.30.0.0/16
  defaultNetwork:
    type: OpenShiftSDN
    openshiftSDNConfig:
      mode: NetworkPolicy
      mtu: 1450
      vxlanPort: 4789
  kubeProxyConfig:
    iptablesSyncPeriod: 30s
    proxyArguments:
      iptables-min-sync-period:
      - 0s
```

1.4.8. 部署集群

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

重要

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

先决条件

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

流程

1. 运行安装程序：

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

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

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

注意

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

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

重要

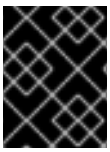
安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrapper** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅*从过期的 control plane 证书中恢复*的文档。

重要

您不得删除安装程序或安装程序所创建的文件。需要这两者才能删除集群。

1.4.9. 通过下载二进制文件安装 CLI

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



重要

如果安装了旧版本的 **oc**，则无法使用 OpenShift Container Platform 4.4 中的所有命令。下载并安装新版本的 **oc**。

1.4.9.1. 在 Linux 上安装 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. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Linux**，并点 **Download command-line tools**。
4. 解包存档：

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

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

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

1.4.9.2. 在 Windows 上安装 CLI

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

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Windows**，点 **Download command-line tools**。
4. 使用 ZIP 程序解压存档。
5. 把 **oc** 二进制代码放到 **PATH** 中的目录中。
   要查看您的 **PATH**，请打开命令提示窗口并执行以下命令：

```
C:\> path
```

安装 CLI 后，就可以使用**oc**命令：

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

1.4.9.3. 在 macOS 上安装 CLI

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

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **MacOS**，并点 **Download command-line tools**。
4. 解包和解压存档。
5. 将 **oc** 二进制文件移到 PATH 的目录中。
   要查看您的 **PATH**，打开一个终端窗口并执行以下命令：

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

1.4.10. 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** 文件包含关于集群的信息，供 CLI 用于将客户端连接到正确集群和 API 服务器。该文件特只适用于一个特定的集群，在 OpenShift Container Platform 安装过程中创建。

先决条件

- 部署一个 OpenShift Container Platform 集群。
- 安装 **oc** CLI。

流程

1. 导出 **kubeadmin** 凭证：

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

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

2. 使用导出的配置，验证能否成功运行 **oc** 命令：

```
$ oc whoami
system:admin
```

1.4.11. 后续步骤

- 自定义集群。
- 若有需要，您可以选择不使用远程健康报告。

1.5. 将 AZURE 上的集群安装到现有的 VNET

在 OpenShift Container Platform 版本 4.4 中，您可以在 Microsoft Azure 上将集群安装到现有 Azure Virtual Network (VNet) 中。安装程序会置备所需基础架构的其余部分，您可以进一步定制这些基础架构。要自定义安装，请在安装集群前修改 **install-config.yaml** 文件中的参数。

1.5.1. 先决条件

- 查看有关 OpenShift Container Platform 安装和更新流程的详细信息。
- 配置一个 Azure 帐户以托管集群，并决定要将集群部署到的已测试和验证的区域。
- 如果使用防火墙，则必须将其配置为允许集群需要访问的站点。
- 如果不允许系统管理身份和访问管理（IAM），集群管理员可以手动创建和维护 IAM 凭证。手动模式也可以用于云 IAM API 无法访问的环境中。

1.5.2. 关于为 OpenShift Container Platform 集群重复使用 VNet

在 OpenShift Container Platform 4.4 中，您可以在 Microsoft Azure 中将集群部署到现有的 Azure Virtual Network (VNet) 中。如果您这样做，还必须在 VNet 和路由规则中使用现有子网。

通过将 OpenShift Container Platform 部署到现有的 Azure VNet 中，您可能会避开新帐户中的服务限制，或者更容易地利用公司所设置的操作限制。如果您无法获得创建 VNet 所需的基础架构创建权限，则可以使用这个选项。

重要

使用现有 VNet 需要使用更新的 Azure 私有 DNS（预览）功能。如需了解有关此功能限制的更多信息，请参阅 Azure DNS 私有区的预览刷新 。

1.5.2.1. 使用 VNet 的要求

当使用现有 VNet 部署集群时，必须在安装集群前执行额外网络配置。在安装程序置备的基础架构集群中，安装程序通常会创建以下组件，但在安装到现有 VNet 时不会创建它们：

- 子网
- 路由表
- VNets
- 网络安全组

如果您使用自定义 VNet，您必须正确配置它及其子网，以便安装程序和集群使用。安装程序不能为集群分配要使用的网络范围，为子网设置路由表，或者设置类似 DHCP 的 VNet 选项，因此您必须在安装集群前配置它们。

集群必须能够访问包含现有 VNet 和子网的资源组。虽然集群创建的所有资源都放在它创建的单独资源组中，但有些网络资源则从另外一个独立的组中使用。一些集群 Operator 必须能够访问这两个资源组中的资源。例如，Machine API 控制器会为它创建的虚拟机附加 NICS，使其从网络资源组中划分子网。

您的 VNet 必须满足以下特征：

- VNet 的 CIDR 块必须包含 **Networking.machineCIDR**，它是集群机器的 IP 地址池。
- VNet 及其子网必须属于同一资源组，子网必须配置为使用 Azure 分配的 DHCP IP 地址而不是静态 IP 地址。

您必须在 VNet 中提供两个子网，一个用于 control plane 机器，一个用于计算机器。因为 Azure 在您指定的区域内的不同可用区中分发机器，所以集群将默认具有高可用性功能。

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。
- 您可以为每个可用区提供两个私有子网。
- 子网 CIDR 属于您指定的机器 CIDR。机器不会在没有为其提供私有子网的可用区中置备。如果需要，安装程序会创建管理 control plane 和 worker 节点的公共负载均衡器，Azure 会为其分配一个公共 IP 地址。

如果您销毁了使用现有 VNet 的集群，则不会删除 VNet。

1.5.2.1.1. 网络安全组要求

托管 compute 和 control plane 机器的子网的网络安全组需要特定的访问权限，以确保集群通信正确。您必须创建规则来允许访问所需的集群通信端口。

重要

在安装集群前必须先设置网络安全组规则。如果您试图在没有所需访问权限的情况下安装集群，安装程序就无法访问 Azure API，且会导致安装失败。

表 1.8. 所需端口

端口	描述	Control plane	Compute
80	允许 HTTP 流量	x	
443	允许 HTTPS 流量	x	
6443	允许与 control plane 机器通信。	x	x
22623	允许与机器配置服务器通信。	x	x

1.5.2.2. 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、存储和负载均衡器，但不能创建与网络相关的组件，如 VNets 、子网或入站规则。

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

1.5.2.3. 集群间隔离

因为集群无法修改现有子网中的网络安全组，所以无法在 VNet 中相互隔离集群。

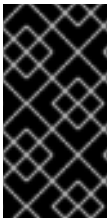
1.5.3. OpenShift Container Platform 对互联网和 Telemetry 的访问

在 OpenShift Container Platform 4.4 中，您需要访问互联网来安装集群。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，这也需要访问互联网。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 Red Hat OpenShift Cluster Manager （OCM）。

确认 Red Hat OpenShift Cluster Manager 清单正确后，可以由 Telemetry 自动维护，也可以使用 OCM 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。

您必须具有以下互联网访问权限：

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

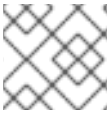


重要

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

1.5.4. 生成 SSH 私钥并将其添加到代理中

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



注意

在生产环境中，您需要进行灾难恢复和调试。

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

注意

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

流程

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

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

1 指定 SSH 密钥的路径和文件名，如 **~/.ssh/id_rsa**。不要指定已存在的 SSH 密钥，因为它会被覆盖。

运行此命令会在指定的位置生成不需要密码的 SSH 密钥。

2. 作为后台任务启动 **ssh-agent** 进程：

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

Agent pid 31874
```

3. 将 SSH 私钥添加到 **ssh-agent**：

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

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

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

后续步骤

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

1.5.5. 获取安装程序

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

先决条件

- 必须从使用 Linux 或 macOS 的计算机安装集群。
- 需要 500 MB 本地磁盘空间来下载安装程序。

流程

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

重要

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

重要

删除安装程序创建的文件不会删除您的集群，即使集群在安装过程中失败也是如此。您必须完成针对特定云供应商的 OpenShift Container Platform 卸载流程，才能完全删除您的集群。

3. 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 <installation_program>.tar.gz
```

4. 在 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中，下载您的安装 pull secret 的 **.txt** 文件。通过此 pull secret，您可以进行所含授权机构提供的服务的身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

1.5.6. 创建安装配置文件

您可以自定义在 Microsoft Azure 上安装的 OpenShift Container Platform 集群。

先决条件

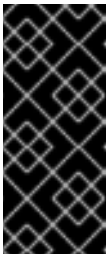
- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

流程

1. 创建 **install-config.yaml** 文件。
   a. 运行以下命令：

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

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



重要

指定一个空目录。一些安装信息，如 bootstrap X.509 证书，有较短的过期间隔，因此不要重复使用安装目录。如果要重复使用另一个集群安装中的个别文件，可以将其复制到您的目录中。但是，一些安装数据的文件名可能会在发行版本之间有所改变。从 OpenShift Container Platform 老版本中复制安装文件时要格外小心。

b. 在提示符处，提供您的云的配置详情：

i. 可选：选择用来访问集群机器的 SSH 密钥。

注意

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

ii. 选择 **azure** 作为目标平台。

iii. 如果计算机上没有 Microsoft Azure 配置集，请为您的订阅和服务主体指定以下 Azure 参数值：

- **azure subscription id**：要用于集群的订阅 ID。指定帐户输出中的 **id** 值。
- **azure tenant id**：租户 ID。指定帐户输出中的 **tenantId** 值。
- **azure service principal client id**：服务主体的 **appId** 参数值。
- **azure service principal client secret**：服务主体的 **password** 参数值。

iv. 选择要在其中部署集群的区域。

v. 选择集群要部署到的基域。基域与您为集群创建的 Azure DNS 区对应。

vi. 为集群输入一个描述性名称。

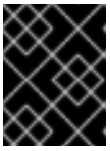
重要

所有通过公共端点提供的 Azure 资源均存在资源名称的限制，您无法创建使用某些名称的资源。如需 Azure 限制词语列表，请参阅 Azure 文档中的解决保留资源名称错误。

vii. 粘贴从 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中获取的 pull secret。

2. 修改 **install-config.yaml** 文件。您可以在**安装配置参数**部分中找到有关可用参数的更多信息。

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

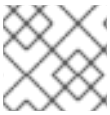


重要

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

1.5.6.1. 安装配置参数

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



注意

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

表 1.9. 所需的参数

参数	描述	值
baseDomain	云供应商的基域。此值用于创建到 OpenShift Container Platform 集群组件的路由。集群的完整 DNS 名称是 **baseDomain** 和 **metadata.name** 参数值的组合，其格式为 **<metadata.name>.<baseDomain>**。	完全限定域名或子域名，如 **example.com**。
controlPlane.platform	托管 control plane 机器的云供应商。此参数值必须与 **compute.platform** 参数值匹配。	**aws**、**azure**、**gcp**、**openstack** 或 **{}**
compute.platform	托管 worker 机器的云供应商。此参数值必须与 **controlPlane.platform** 参数值匹配。	**aws**、**azure**、**gcp**、**openstack** 或 **{}**
metadata.name	集群的名称。	包含大写字母或小写字母的字符串，如 **dev**。
platform.<platform>.region	集群要部署到的区域。	云的有效区域，如 AWS 的 **us-east-1**、Azure 的 **centralus**。Red Hat OpenStack Platform (RHOSP) 不使用这个参数

参数	描述	值

pullSecret	从 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中获取的 pull secret。您可以使用此 pull secret 来进行所含授权机构提供的服务的身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。	<pre>{
		  "auths":{
		  "cloud.openshift.com":{
		  "auth":"b3Blb=",
		  "email":"you@example.com"
		  },
		  "quay.io":{
		  "auth":"b3Blb=",
		  "email":"you@example.com"
		  }
		  }
		}</pre>

表 1.10. 可选参数

参数	描述	值
sshKey	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。	添加到 **ssh-agent** 进程的有效本地公共 SSH 密钥。
fips	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。	**false** 或 **true**
publish	如何发布集群的面向用户的端点。	**Internal** 或 **External**。把 **publish** 设置为 **Internal** 以部署一个私有集群，它不能被互联网访问。默认值为 **External**。

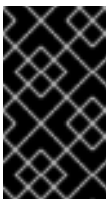
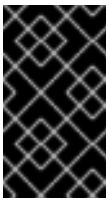
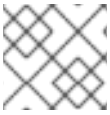
参数	描述	值
compute.hyperthreading	是否在计算机器上启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。  **重要** 如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。	**Enabled** 或 **Disabled**
compute.replicas	要置备的计算机器数量，也称为 worker 机器。	大于或等于 **2** 的正整数。默认值为 **3**。
controlPlane.hyperthreading	是否在 control plane 机器上启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。  **重要** 如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。	**Enabled** 或 **Disabled**
controlPlane.replicas	要置备的 control plane 机器数量。	唯一支持的值是 **3**，它是默认值。

表 1.11. 其他 Azure 参数

参数	描述	值
machines.platform.azure.type	Azure 虚拟机实例类型。	使用 Windows 或 Linux 作为操作系统的虚拟机。请参阅 Azure 文档中的 Azure Stack 上支持的客户端操作系统。
machines.platform.azure.osDisk.diskSizeGB	虚拟机的 Azure 磁盘大小。	以 GB 为单位表示磁盘大小的整数，如 **512**。支持的最小磁盘大小为 **120**。
platform.azure.baseDomainResourceGroupName	包含基域的 DNS 区的资源组的名称。	字符串，如 **production_cluster**。

参数	描述	值
platform.azure.region	托管集群的 Azure 区域名称。	任何有效的区域名称。
platform.azure.zone	可在其中放入机器的可用区的列表。如需高可用性，请至少指定两个区域。	区域列表，如 **["1", "2", "3"]**。
platform.azure.networkResourceGroupName	包含要将集群部署到的现有 VNet 的资源组名称。这个名称不能和 **platform.azure.baseDomainResourceGroupName**相同。	字符串。
platform.azure.virtualNetwork	要将集群部署到的现有 VNet 的名称。	字符串。
platform.azure.controlPlaneSubnet	要将 control plane 机器部署到的 VNet 中现有子网的名称。	有效的 CIDR，如 **10.0.0.0/16**。
platform.azure.computeSubnet	您要将计算机器部署到的 VNet 中现有子网的名称。	有效的 CIDR，如 **10.0.0.0/16**。



注意

您无法自定义 Azure 可用区，也不能使用标签来整理用于 Azure 集群的 Azure 资源。

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

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

重要

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

```
apiVersion: v1
baseDomain: example.com (1)
controlPlane: (2)
  hyperthreading: Enabled (3) (4)
  name: master
  platform:
    azure:
      osDisk:
        diskSizeGB: 1024 (5)
      type: Standard_D8s_v3
  replicas: 3
compute: (6)
```

```
- hyperthreading: Enabled 7
  name: worker
  platform:
    azure:
      type: Standard_D2s_v3
      osDisk:
        diskSizeGB: 512 8
      zones: 9
      - "1"
      - "2"
      - "3"
  replicas: 5
metadata:
  name: test-cluster 10
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  azure:
    region: centralus 11
    baseDomainResourceGroupName: resource_group 12
    networkResourceGroupName: vnet_resource_group 13
    virtualNetwork: vnet 14
    controlPlaneSubnet: control_plane_subnet 15
    computeSubnet: compute_subnet 16
pullSecret: '{"auths": ...}' 17
fips: false 18
sshKey: ssh-ed25519 AAAA... 19
```

1 10 11 17 必需。安装程序会提示您输入这个值。

2 6 如果没有提供这些参数和值，安装程序会提供默认值。

3 7 **controlPlane** 部分是一个单映射，但 compute 部分是一系列映射。为满足不同数据结构的要求，**compute** 部分的第一行必须以连字符 **-** 开头，**controlPlane** 部分的第一行则不可以连字符开头。虽然这两个部分目前都定义单个机器池，但未来的 OpenShift Container Platform 版本可能会支持在安装过程中定义多个计算池。只使用一个 control plane 池。

4 是否要启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。您可以通过将参数值设为 **Disabled** 来禁用。如果您在某些集群机器上禁用并发多线程，则必须在所有集群机器上禁用。

重要

如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。如果禁用并发多线程，请使用较大的虚拟机类型，如 **Standard_D8s_v3**。

5 8 可以 GB 为单位指定要使用的磁盘大小。master 节点的最低推荐值为 1024 GB。

9 指定要将机器部署到的区域列表。如需高可用性，请至少指定两个区域。

12 指定包含基域的 DNS 区的资源组的名称。

13 如果您使用现有的 如果使用现有的 VNet，请指定包含它的资源组的名称。。

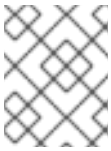
14 如果使用现有的 VNet，请指定其名称。

15 如果使用现有的 VNet，请指定托管 control plane 机器的子网名称。

16 如果使用现有的 VNet，请指定托管计算机器的子网名称。

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 加密套件，并使用由 RHCOS 提供的加密模块。

19 您可以选择提供您用来访问集群中机器的 **sshKey** 值。



注意

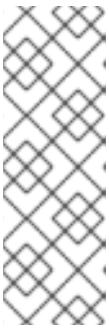
对于您要在其上执行安装调试或灾难恢复的生产环境 OpenShift Container Platform 集群，请指定 **ssh-agent** 进程使用的 SSH 密钥。

1.5.6.3. 在安装过程中配置集群范围代理

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

先决条件

- 现有的 **install-config.yaml** 文件。
- 查看集群需要访问的站点，并决定是否需要绕过代理。默认情况下代理所有集群出口流量，包括对托管云供应商 API 的调用。如果需要，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: http://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
...
```

1. 用于创建集群外 HTTP 连接的代理 URL。URL 必须是 **http**。如果您使用不要求额外代理配置但需要额外 CA 的 MITM 透明代理网络，则不得指定 **httpProxy** 值。

2. 用于创建集群外 HTTPS 连接的代理 URL。如果未指定此字段，**httpProxy** 会同时用于 HTTP 和 HTTPS 连接。如果您使用不要求额外代理配置但需要额外 CA 的 MITM 透明代理网络，则不得指定 **httpsProxy** 值。

3. 要排除代理的目标域名、域、IP 地址或其他网络 CIDR 的逗号分隔列表。域之前加上前缀 **.** 可包含该域的所有子域。使用 * 可对所有目的地绕过所有代理。

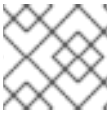
4. 如果提供，安装程序会在 **openshift-config** 命名空间中生成名为 **user-ca-bundle** 的配置映射，其包含代理 HTTPS 连接所需的一个或多个额外 CA 证书。然后，Cluster Network Operator 会创建 **trusted-ca-bundle** 配置映射，将这些内容与 Red Hat Enterprise Linux CoreOS（RHCOS）信任捆绑包合并， **Proxy** 对象的 **trustedCA** 字段中也会引用此配置映射。**additionalTrustBundle** 字段是必需的，除非代理的身份证书由来自 RHCOS 信任捆绑包的颁发机构签名。如果您使用不要求额外代理配置但需要额外 CA 的 MITM 透明代理网络，您必须提供 MITM CA 证书。

注意

安装程序不支持代理的 **readinessEndpoints** 字段。

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

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



注意

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

1.5.7. 部署集群

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

重要

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

先决条件

- 配置托管集群的云平台的帐户。

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

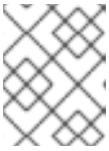
流程

1. 运行安装程序：

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

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

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



注意

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

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

重要

安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrapper** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅 *从过期的 control plane 证书中恢复* 的文档。

重要

您不得删除安装程序或安装程序所创建的文件。需要这两者才能删除集群。

1.5.8. 通过下载二进制文件安装 CLI

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

重要

如果安装了旧版本的 **oc**，则无法使用 OpenShift Container Platform 4.4 中的所有命令。下载并安装新版本的 **oc**。

1.5.8.1. 在 Linux 上安装 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. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。

3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Linux**，并点 **Download command-line tools**。
4. 解包存档：

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

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

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

1.5.8.2. 在 Windows 上安装 CLI

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

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Windows**，点 **Download command-line tools**。
4. 使用 ZIP 程序解压存档。
5. 把 **oc** 二进制代码放到 **PATH** 中的目录中。
要查看您的 **PATH**，请打开命令提示窗口并执行以下命令：

```
C:\> path
```

安装 CLI 后，就可以使用**oc**命令：

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

1.5.8.3. 在 macOS 上安装 CLI

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

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **MacOS**，并点 **Download command-line tools**。

4. 解包和解压存档。

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

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

1.5.9. 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** 文件包含关于集群的信息，供 CLI 用于将客户端连接到正确集群和 API 服务器。该文件特只适用于一个特定的集群，在 OpenShift Container Platform 安装过程中创建。

先决条件

- 部署一个 OpenShift Container Platform 集群。
- 安装 **oc** CLI。

流程

1. 导出 **kubeadmin** 凭证：

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

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

2. 使用导出的配置，验证能否成功运行 **oc** 命令：

```
$ oc whoami
system:admin
```

1.5.10. 后续步骤

- 自定义集群。
- 若有需要，您可以选择不使用远程健康报告。

1.6. 在 AZURE 上安装私有集群

在 OpenShift Container Platform 版本 4.4 中，您可以在 Microsoft Azure 上将私有集群安装到现有 Azure Virtual Network (VNet) 中。安装程序会置备所需基础架构的其余部分，您可以进一步定制这些基础架构。要自定义安装，请在安装集群前修改 **install-config.yaml** 文件中的参数。

1.6.1. 先决条件

- 查看有关 OpenShift Container Platform 安装和更新 流程的详细信息。

- 配置一个 Azure 帐户以托管集群，并决定要将集群部署到的已测试和验证的区域。
- 如果使用防火墙，则必须将其配置为允许集群需要访问的站点。
- 如果不允许系统管理身份和访问管理（IAM），集群管理员可以手动创建和维护 IAM 凭证。手动模式也可以用于云 IAM API 无法访问的环境中。

1.6.2. 私有集群

如果您的环境不需要外部互联网连接，您可以部署不公开外部端点的 OpenShift Container Platform 集群。私有集群只能从内部网络访问，且无法在互联网中看到。

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

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

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

1.6.2.1. Azure 中的私有集群

要在 Microsoft Azure 平台上创建私有集群，您必须提供一个现有的私有 VNet 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。安装程序只为内部流量配置 Ingress Operator 和 API 服务器。

根据您的网络如何连接到私有 VNET，您可能需要使用 DNS 转发程序来解析集群的专用 DNS 记录。集群的机器在内部使用 **168.63.129.16** 进行 DNS 解析。如需更多信息，请参阅Azure 文档中的 Azure Private DNS？ 和 什么是 IP 地址 168.63.129.16？ 部分 。

集群仍然需要访问互联网来访问 Azure API。

安装私有集群时不需要或创建以下项目：

- **BaseDomainResourceGroup**，因为集群不创建公共记录
- 公共 IP 地址
- 公共 DNS 记录
- 公共端点

> The cluster is configured so that the Operators do not create public records for the cluster and all cluster machines are placed in the private subnets that you specify.

1.6.2.1.1. 限制：

Azure 上的私有集群只受到与使用一个现有 VNet 相关的限制。

1.6.3. 关于为 OpenShift Container Platform 集群重复使用 VNet

在 OpenShift Container Platform 4.4 中，您可以在 Microsoft Azure 中将集群部署到现有的 Azure Virtual Network (VNet) 中。如果您这样做，还必须在 VNet 和路由规则中使用现有子网。

通过将 OpenShift Container Platform 部署到现有的 Azure VNet 中，您可能会避开新帐户中的服务限制，或者更容易地利用公司所设置的操作限制。如果您无法获得创建 VNet 所需的基础架构创建权限，则可以使用这个选项。

重要

使用现有 VNet 需要使用更新的 Azure 私有 DNS（预览）功能。如需了解有关此功能限制的更多信息，请参阅 Azure DNS 私有区的预览刷新 。

1.6.3.1. 使用 VNet 的要求

当使用现有 VNet 部署集群时，必须在安装集群前执行额外网络配置。在安装程序置备的基础架构集群中，安装程序通常会创建以下组件，但在安装到现有 VNet 时不会创建它们：

- 子网
- 路由表
- VNets
- 网络安全组

如果您使用自定义 VNet，您必须正确配置它及其子网，以便安装程序和集群使用。安装程序不能为集群分配要使用的网络范围，为子网设置路由表，或者设置类似 DHCP 的 VNet 选项，因此您必须在安装集群前配置它们。

集群必须能够访问包含现有 VNet 和子网的资源组。虽然集群创建的所有资源都放在它创建的单独资源组中，但有些网络资源则从另外一个独立的组中使用。一些集群 Operator 必须能够访问这两个资源组中的资源。例如，Machine API 控制器会为它创建的虚拟机附加 NICS，使其从网络资源组中划分子网。

您的 VNet 必须满足以下特征：

- VNet 的 CIDR 块必须包含 **Networking.machineCIDR**，它是集群机器的 IP 地址池。
- VNet 及其子网必须属于同一资源组，子网必须配置为使用 Azure 分配的 DHCP IP 地址而不是静态 IP 地址。

您必须在 VNet 中提供两个子网，一个用于 control plane 机器，一个用于计算机器。因为 Azure 在您指定的区域内的不同可用区中分发机器，所以集群将默认具有高可用性功能。

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。
- 您可以为每个可用区提供两个私有子网。
- 子网 CIDR 属于您指定的机器 CIDR。机器不会在没有为其提供私有子网的可用区中置备。如果需要，安装程序会创建管理 control plane 和 worker 节点的公共负载均衡器，Azure 会为其分配一个公共 IP 地址。

如果您销毁了使用现有 VNet 的集群，则不会删除 VNet。

1.6.3.1.1. 网络安全组要求

托管 compute 和 control plane 机器的子网的网络安全组需要特定的访问权限，以确保集群通信正确。您必须创建规则来允许访问所需的集群通信端口。

重要

在安装集群前必须先设置网络安全组规则。如果您试图在没有所需访问权限的情况下安装集群，安装程序就无法访问 Azure API，且会导致安装失败。

表 1.12. 所需端口

端口	描述	Control plane	Compute
80	允许 HTTP 流量	x	
443	允许 HTTPS 流量	x	
6443	允许与 control plane 机器通信。	x	x
22623	允许与机器配置服务器通信。	x	x

1.6.3.2. 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、存储和负载均衡器，但不能创建与网络相关的组件，如 VNets 、子网或入站规则。

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

1.6.3.3. 集群间隔离

因为集群无法修改现有子网中的网络安全组，所以无法在 VNet 中相互隔离集群。

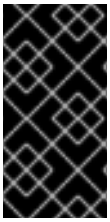
1.6.4. OpenShift Container Platform 对互联网和 Telemetry 的访问

在 OpenShift Container Platform 4.4 中，您需要访问互联网来安装集群。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，这也需要访问互联网。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 Red Hat OpenShift Cluster Manager （OCM）。

确认 Red Hat OpenShift Cluster Manager 清单正确后，可以由 Telemetry 自动维护，也可以使用 OCM 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。

您必须具有以下互联网访问权限：

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



重要

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

1.6.5. 生成 SSH 私钥并将其添加到代理中

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

注意

在生产环境中，您需要进行灾难恢复和调试。

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

注意

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

流程

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

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

1 指定 SSH 密钥的路径和文件名，如 **~/.ssh/id_rsa**。不要指定已存在的 SSH 密钥，因为它会被覆盖。

运行此命令会在指定的位置生成不需要密码的 SSH 密钥。

2. 作为后台任务启动 **ssh-agent** 进程：

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

Agent pid 31874
```

3. 将 SSH 私钥添加到 **ssh-agent**：

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

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

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

后续步骤

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

1.6.6. 获取安装程序

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

先决条件

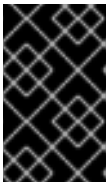
- 必须从使用 Linux 或 macOS 的计算机安装集群。
- 需要 500 MB 本地磁盘空间来下载安装程序。

流程

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

重要

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



重要

删除安装程序创建的文件不会删除您的集群，即使集群在安装过程中失败也是如此。您必须完成针对特定云供应商的 OpenShift Container Platform 卸载流程，才能完全删除您的集群。

3. 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 <installation_program>.tar.gz
```

4. 在 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中，下载您的安装 pull secret 的 **.txt** 文件。通过此 pull secret，您可以进行所含授权机构提供的服务的身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

1.6.7. 手动创建安装配置文件

对于使用用户置备的基础架构的 OpenShift Container Platform 安装，您必须手动生成安装配置文件。对于只能从内部网络访问且不能在互联网中看到的私有 OpenShift Container Platform 集群安装，您必须手动生成安装配置文件。

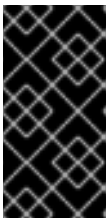
先决条件

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

流程

1. 创建用来存储您所需的安装资产的安装目录：

```
$ mkdir <installation_directory>
```



重要

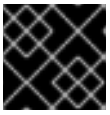
您必须创建目录。一些安装信息，如 bootstrap X.509 证书，有较短的过期间隔，因此不要重复使用安装目录。如果要重复使用另一个集群安装中的个别文件，可以将其复制到您的目录中。但是，一些安装数据的文件名可能会在发行版本之间有所改变。从 OpenShift Container Platform 老版本中复制安装文件时要格外小心。

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

注意

此配置文件必须命名为 **install-config.yaml**。

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



重要

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

1.6.7.1. 安装配置参数

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

注意

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

表 1.13. 所需的参数

参数	描述	值
baseDomain	云供应商的基域。此值用于创建到 OpenShift Container Platform 集群组件的路由。集群的完整 DNS 名称是 **baseDomain** 和 **metadata.name** 参数值的组合，其格式为 **<metadata.name>.<baseDomain>**。	完全限定域名或子域名，如 **example.com**。
controlPlane.platform	托管 control plane 机器的云供应商。此参数值必须与 **compute.platform** 参数值匹配。	**aws**、**azure**、**gcp**、**openstack** 或 **{}**

参数	描述	值
compute.platform	托管 worker 机器的云供应商。此参数值必须与 **controlPlane.platform** 参数值匹配。	**aws**、**azure**、**gcp**、**openstack** 或 **{}**
metadata.name	集群的名称。	包含大写字母或小写字母的字符串，如 **dev**。
platform.<platform>.region	集群要部署到的区域。	云的有效区域，如 AWS 的 **us-east-1**、Azure 的 **centralus**。Red Hat OpenStack Platform (RHOSP) 不使用这个参数

pullSecret	从 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中获取的 pull secret。您可以使用此 pull secret 来进行所含授权机构提供的服务的身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。	<pre>{
		  "auths":{
		  "cloud.openshift.com":{
		  "auth":"b3Blb=",
		  "email":"you@example.com"
		  },
		  "quay.io":{
		  "auth":"b3Blb=",
		  "email":"you@example.com"
		  }
		  }
		}</pre>

表 1.14. 可选参数

参数	描述	值
sshKey	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。	添加到 **ssh-agent** 进程的有效本地公共 SSH 密钥。

参数	描述	值
fips	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。	**false** 或 **true**
publish	如何发布集群的面向用户的端点。	**Internal** 或 **External**。把 **publish** 设置为 **Internal** 以部署一个私有集群，它不能被互联网访问。默认值为 **External**。
compute.hyperthreading	是否在计算机器上启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。  **重要** 如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。	**Enabled** 或 **Disabled**
compute.replicas	要置备的计算机器数量，也称为 worker 机器。	大于或等于 **2** 的正整数。默认值为 **3**。
controlPlane.hyperthreading	是否在 control plane 机器上启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。  **重要** 如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。	**Enabled** 或 **Disabled**
controlPlane.replicas	要置备的 control plane 机器数量。	唯一支持的值是 **3**，它是默认值。

表 1.15. 其他 Azure 参数

参数	描述	值
machines.platform.azure.type	Azure 虚拟机实例类型。	使用 Windows 或 Linux 作为操作系统的虚拟机。请参阅 Azure 文档中的 Azure Stack 上支持的客户端操作系统。

参数	描述	值
machines.platform.azure.osDisk.diskSizeGB	虚拟机的 Azure 磁盘大小。	以 GB 为单位表示磁盘大小的整数，如 **512**。支持的最小磁盘大小为 **120**。
platform.azure.baseDomainResourceGroupName	包含基域的 DNS 区的资源组的名称。	字符串，如 **production_cluster**。
platform.azure.region	托管集群的 Azure 区域名称。	任何有效的区域名称。
platform.azure.zone	可在其中放入机器的可用区的列表。如需高可用性，请至少指定两个区域。	区域列表，如 **["1", "2", "3"]**。
platform.azure.networkResourceGroupName	包含要将集群部署到的现有 VNet 的资源组名称。这个名称不能和 **platform.azure.baseDomainResourceGroupName**相同。	字符串。
platform.azure.virtualNetwork	要将集群部署到的现有 VNet 的名称。	字符串。
platform.azure.controlPlaneSubnet	要将 control plane 机器部署到的 VNet 中现有子网的名称。	有效的 CIDR，如 **10.0.0.0/16**。
platform.azure.computeSubnet	您要将计算机器部署到的 VNet 中现有子网的名称。	有效的 CIDR，如 **10.0.0.0/16**。

注意

您无法自定义 Azure 可用区，也不能使用标签来整理用于 Azure 集群的 Azure 资源。

1.6.7.2. Azure 的自定义 install-config.yaml 文件示例

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

重要

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

```
apiVersion: v1
baseDomain: example.com (1)
controlPlane: (2)
  hyperthreading: Enabled (3) (4)
  name: master
  platform:
```

```
    azure:
      osDisk:
        diskSizeGB: 1024 (5)
      type: Standard_D8s_v3
  replicas: 3
compute: (6)
- hyperthreading: Enabled (7)
  name: worker
  platform:
    azure:
      type: Standard_D2s_v3
      osDisk:
        diskSizeGB: 512 (8)
      zones: (9)
      - "1"
      - "2"
      - "3"
  replicas: 5
metadata:
  name: test-cluster (10)
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  azure:
    region: centralus (11)
    baseDomainResourceGroupName: resource_group (12)
    networkResourceGroupName: vnet_resource_group (13)
    virtualNetwork: vnet (14)
    controlPlaneSubnet: control_plane_subnet (15)
    computeSubnet: compute_subnet (16)
pullSecret: '{"auths": ...}' (17)
fips: false (18)
sshKey: ssh-ed25519 AAAA... (19)
publish: Internal (20)
```

(1) (10) (11) (17) 必需。安装程序会提示您输入这个值。

(2) (6) 如果没有提供这些参数和值，安装程序会提供默认值。

(3) (7) **controlPlane** 部分是一个单映射，但 compute 部分是一系列映射。为满足不同数据结构的要求，**compute** 部分的第一行必须以连字符 **-** 开头，**controlPlane** 部分的第一行则不可以连字符开头。虽然这两个部分目前都定义单个机器池，但未来的 OpenShift Container Platform 版本可能会支持在安装过程中定义多个计算池。只使用一个 control plane 池。

(4) 是否要启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。您可以通过将参数值设为 **Disabled** 来禁用。如果您在某些集群机器上禁用并发多线程，则必须在所有集群机器上禁用。

重要

如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。如果禁用并发多线程，请使用较大的虚拟机类型，如 **Standard_D8s_v3**。

5 8 可以 GB 为单位指定要使用的磁盘大小。master 节点的最低推荐值为 1024 GB。

9 指定要将机器部署到的区域列表。如需高可用性，请至少指定两个区域。

12 指定包含基域的 DNS 区的资源组的名称。

13 如果您使用现有的 如果使用现有的 VNet，请指定包含它的资源组的名称。。

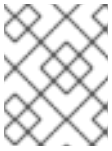
14 如果使用现有的 VNet，请指定其名称。

15 如果使用现有的 VNet，请指定托管 control plane 机器的子网名称。

16 如果使用现有的 VNet，请指定托管计算机器的子网名称。

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 加密套件，并使用由 RHCOS 提供的加密模块。

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。

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

1.6.7.3. 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

- 现有的 **install-config.yaml** 文件。
- 查看集群需要访问的站点，并决定是否需要绕过代理。默认情况下代理所有集群出口流量，包括对托管云供应商 API 的调用。如果需要，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: http://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
...
```

1. 用于创建集群外 HTTP 连接的代理 URL。URL 必须是 **http**。如果您使用不要求额外代理配置但需要额外 CA 的 MITM 透明代理网络，则不得指定 **httpProxy** 值。

2. 用于创建集群外 HTTPS 连接的代理 URL。如果未指定此字段，**httpProxy** 会同时用于 HTTP 和 HTTPS 连接。如果您使用不要求额外代理配置但需要额外 CA 的 MITM 透明代理网络，则不得指定 **httpsProxy** 值。

3. 要排除代理的目标域名、域、IP 地址或其他网络 CIDR 的逗号分隔列表。域之前加上前缀 **.** 可包含该域的所有子域。使用 * 可对所有目的地绕过所有代理。

4. 如果提供，安装程序会在 **openshift-config** 命名空间中生成名为 **user-ca-bundle** 的配置映射，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。如果您使用不要求额外代理配置但需要额外 CA 的 MITM 透明代理网络，您必须提供 MITM CA 证书。

注意

安装程序不支持代理的 **readinessEndpoints** 字段。

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

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

注意

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

1.6.8. 部署集群

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

重要

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

先决条件

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

流程

1. 运行安装程序：

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

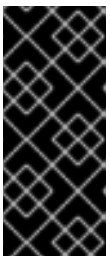
1 对于 **<installation_directory>**，请指定

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

注意

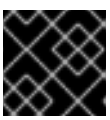
如果您在主机上配置的云供应商帐户没有足够的权限来部署集群，安装过程将会停止，并且显示缺少的权限。

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



重要

安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrapper** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅 *从过期的 control plane 证书中恢复*的文档。



重要

您不得删除安装程序或安装程序所创建的文件。需要这两者才能删除集群。

1.6.9. 通过下载二进制文件安装 CLI

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

重要

如果安装了旧版本的 **oc**，则无法使用 OpenShift Container Platform 4.4 中的所有命令。下载并安装新版本的 **oc**。

1.6.9.1. 在 Linux 上安装 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. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Linux**，并点 **Download command-line tools**。
4. 解包存档：

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

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

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

1.6.9.2. 在 Windows 上安装 CLI

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

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Windows**，点 **Download command-line tools**。
4. 使用 ZIP 程序解压存档。
5. 把 **oc** 二进制代码放到 **PATH** 中的目录中。
要查看您的 **PATH**，请打开命令提示窗口并执行以下命令：

```
C:\> path
```

安装 CLI 后，就可以使用**oc**命令：

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

1.6.9.3. 在 macOS 上安装 CLI

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

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **MacOS**，并点 **Download command-line tools**。
4. 解包和解压存档。
5. 将 **oc** 二进制文件移到 PATH 的目录中。
   要查看您的 **PATH**，打开一个终端窗口并执行以下命令：

   ```
   $ echo $PATH
   ```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

1.6.10. 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** 文件包含关于集群的信息，供 CLI 用于将客户端连接到正确集群和 API 服务器。该文件特只适用于一个特定的集群，在 OpenShift Container Platform 安装过程中创建。

先决条件

- 部署一个 OpenShift Container Platform 集群。
- 安装 **oc** CLI。

流程

1. 导出 **kubeadmin** 凭证：

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

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

2. 使用导出的配置，验证能否成功运行 **oc** 命令：

   ```
   $ oc whoami
   system:admin
   ```

1.6.11. 后续步骤

- 自定义集群。
- 若有需要，您可以选择不使用远程健康报告。

1.7. 详情请参阅在使用 ARM 模板的 AZURE 上安装集群。

在 OpenShift Container Platform 版本 4.4 中，您可以使用您提供的基础架构在 Microsoft Azure 上安装集群。

提供的几个 Azure Resource Manager（ARM） 模板可协助完成这些步骤，也可帮助您自行建模。您也可以选择通过其他方法创建所需的资源；模板仅作示例之用。

1.7.1. 先决条件

- 查看有关 OpenShift Container Platform 安装和更新流程的详细信息。
- 配置 Azure 帐户以托管集群。
- 下载 Azure CLI 并安装到您的计算机上。请参阅 Azure 文档中的安装 Azure CLI。以下文档使用 Azure CLI 版本 **2.2.0** 进行测试。Azure CLI 命令可能会根据您使用的版本的不同而不同。
- 如果使用防火墙并计划使用遥测（telemetry），您必须将其配置为允许集群需要访问的站点。
- 如果不允许系统管理身份和访问管理（IAM），集群管理员可以手动创建和维护 IAM 凭证。手动模式也可以用于云 IAM API 无法访问的环境中。

注意

如果您要配置代理，请务必也要查看此站点列表。

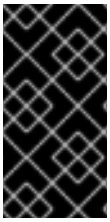
1.7.2. OpenShift Container Platform 对互联网和 Telemetry 的访问

在 OpenShift Container Platform 4.4 中，您需要访问互联网来安装集群。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，这也需要访问互联网。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 Red Hat OpenShift Cluster Manager （OCM）。

确认 Red Hat OpenShift Cluster Manager 清单正确后，可以由 Telemetry 自动维护，也可以使用 OCM 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。

您必须具有以下互联网访问权限：

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



重要

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

1.7.3. 配置 Azure 项目

在安装 OpenShift Container Platform 之前，您必须配置 Azure 项目来托管它。

重要

所有通过公共端点提供的 Azure 资源均存在资源名称的限制，您无法创建使用某些名称的资源。如需 Azure 限制词语列表，请参阅 Azure 文档中的解决保留资源名称错误。

1.7.3.1. Azure 帐户限值

OpenShift Container Platform 集群使用诸多 Microsoft Azure 组件，默认的 Azure 订阅和服务限值、配额和约束会影响您安装 OpenShift Container Platform 集群的能力。

重要

默认的限制因服务类别的不同（如 Free Trial 或 Pay-As-You-Go）以及系列的不同（如 Dv2、F 或 G）而有所不同。例如，对于 Enterprise Agreement 订阅的默认限制是 350 个内核。

在 Azure 上安装默认集群前，请检查您的订阅类型的限制，如有必要，请提高帐户的配额限制。

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

组件	默认所需的组件数	默认 Azure 限值	描述

组件	默认所需的组件数	默认 Azure 限值	描述
vCPU	40	每个区域 20 个	默认集群需要 40 个 vCPU，因此您必须提高帐户限值。 默认情况下，每个集群创建以下实例： ● 一台 Bootstrap 机器，在安装后删除 ● 三个 control plane 机器 ● 三个计算（compute）机器 由于 Bootstrap 机器使用 **Standard_D4s_v3** 机器（使用 4 个 vCPU），control plane 机器使用 **Standard_D8s_v3** 虚拟机（8 个 vCPU），并且 worker 机器使用 **Standard_D4s_v3** 虚拟机（4 个 vCPU），因此默认集群需要 40 个 vCPU。bootstrap 节点 VM（使用 4 个 vCPU）只在安装过程中使用。 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，您必须进一步提高帐户的 vCPU 限值，以确保集群可以部署您需要的机器。 默认情况下，安装程序将 control plane 和 compute 机器分布到一个区域中的所有可用区。要确保集群的高可用性，请选择至少含有三个可用区的区域。如果您的区域包含的可用区少于三个，安装程序将在可用区中放置多台 control plane 机器。
VNet	1	每个区域 1000 个	每个默认集群都需要一个虚拟网络 (VNet)，此网络包括两个子网。
网络接口	6	每个区域 65,536 个	每个默认集群都需要六个网络接口。如果您要创建更多机器或者您部署的工作负载要创建负载均衡器，则集群会使用更多的网络接口。
网络安全组	2	5000	每个默认集群为 VNet 中的每个子网创建网络安全组。默认集群为 control plane 和计算节点子网创建网络安全组： **controlplane**：允许从任何位置通过端口 6443 访问 control plane 机器 **node**：允许从互联网通过端口 80 和 443 访问 worker 节点

组件	默认所需的组件数	默认 Azure 限值	描述
网络负载均衡器	3	每个区域 1000 个	每个集群都会创建以下负载均衡器： **default** 用于在 worker 机器之间对端口 80 和 443 的请求进行负载均衡的公共 IP 地址 **internal** 用于在 control plane 机器之间对端口 6443 和 22623 的请求进行负载均衡的专用 IP 地址 **external** 用于在 control plane 机器之间对端口 6443 的请求进行负载均衡的公共 IP 地址 如果您的应用程序创建了更多的 Kubernetes **LoadBalancer** 服务对象，您的集群会使用更多的负载均衡器。
公共 IP 地址	3		两个公共负载均衡器各自使用一个公共 IP 地址。bootstrap 机器也使用一个公共 IP 地址，以便您可以在安装期间通过 SSH 连接到该机器来进行故障排除。bootstrap 节点的 IP 地址仅在安装过程中使用。
专用 IP 地址	7		内部负载均衡器、三台 control plane 机器中的每一台以及三台 worker 机器中的每一台各自使用一个专用 IP 地址。

1.7.3.2. 在 Azure 中配置公共 DNS 区

要安装 OpenShift Container Platform，您使用的 Microsoft Azure 帐户必须在帐户中具有一个专用的公共托管 DNS 区。此区域必须对域具有权威。此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 Azure 或其他来源获取新的域和注册商。

注意

如需通过 Azure 购买域的更多信息，请参阅 Azure 文档中的购买 Azure 应用服务的自定义域名。

2. 如果您使用现有的域和注册商，请将其 DNS 迁移到 Azure。请参阅 Azure 文档中的将活动 DNS 名称迁移到 Azure 应用服务。

3. 为您的域配置 DNS。按照 Azure 文档中教程：在 Azure DNS 中托管域部分里的步骤，为您的域或子域创建一个公共托管区，提取新的权威名称服务器，并更新您的域使用的名称服务器的注册商记录。
使用合适的根域（如 **openshiftcorp.com**）或子域（如 **clusters.openshiftcorp.com**）。

4. 如果您使用子域，请按照您公司的流程将其委派记录添加到父域。

您可以通过访问此 示例来创建 DNS 区域来查看 Azure 的 DNS 解决方案 。

1.7.3.3. 提高 Azure 帐户限值

要提高帐户限值，请在 Azure 门户上提交支持请求。

注意

每一支持请求只能提高一种类型的配额。

流程

1. 从 Azure 门户，点击左下角的 **Help + suport**。
2. 点击 **New support request**，然后选择所需的值：
   a. 从 **Issue type** 列表中，选择 **Service and subscription limits (quotas)**。
   b. 从 **Subscription** 列表中，选择要修改的订阅。
   c. 从 **Quota type** 列表中，选择要提高的配额。例如，选择 **Compute-VM (cores-vCPUs) subscription limit increases** 以增加 vCPU 的数量，这是安装集群所必须的。
   d. 点击 **Next: Solutions**。
3. 在 **Problem Details** 页面中，提供您要提高配额所需的信息：
   a. 点击 **Provide details**，然后在 **Quota details** 窗口中提供所需的详情。
   b. 在 SUPPORT METHOD 和 CONTACT INFO 部分中，提供问题严重性和您的联系详情。
4. 点击 **Next: Review + create**，然后点击 **Create**。

1.7.3.4. 证书签名请求管理

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

1.7.3.5. 所需的 Azure 角色

Microsoft Azure 帐户必须具有您所用订阅的以下角色：

- **User Access Administrator**

要在 Azure 门户上设置角色，请参阅 Azure 文档中的使用 RBAC 和 Azure 门户管理对 Azure 资源的访问。

1.7.3.6. 创建服务主体

由于 OpenShift Container Platform 及其安装程序必须通过 Azure Resource Manager 创建 Microsoft Azure 资源，因此您必须创建一个能代表它的服务主体。

先决条件

- 安装或更新 Azure CLI。
- 安装**jq**软件包。
- 您的 Azure 帐户具有您所用订阅所需的角色。

流程

1. 登录 Azure CLI：

```
$ az login
```

   在 Web 控制台中，使用您的凭证登录 Azure。

2. 如果您的 Azure 帐户使用订阅，请确保使用正确的订阅。

   a. 查看可用帐户列表并记录您要用于集群的订阅的 **tenantId** 值：

```
$ az account list --refresh
[
  {
    "cloudName": "AzureCloud",
    "id": "9bab1460-96d5-40b3-a78e-17b15e978a80",
    "isDefault": true,
    "name": "Subscription Name",
    "state": "Enabled",
    "tenantId": "6057c7e9-b3ae-489d-a54e-de3f6bf6a8ee",
    "user": {
      "name": "you@example.com",
      "type": "user"
    }
  }
]
```

   b. 查看您的活跃帐户详情，确认 **tenantId** 值与您要使用的订阅匹配：

```
$ az account show
{
  "environmentName": "AzureCloud",
  "id": "9bab1460-96d5-40b3-a78e-17b15e978a80",
  "isDefault": true,
  "name": "Subscription Name",
  "state": "Enabled",
  "tenantId": "6057c7e9-b3ae-489d-a54e-de3f6bf6a8ee", (1)
  "user": {
    "name": "you@example.com",
    "type": "user"
  }
}
```

   (1) 确定 **tenantId** 参数的值是正确订阅的 UUID。

   c. 如果您使用的订阅不正确，请更改活跃的订阅：

```
$ az account set -s <id> 1
```

1 替换您要用于 **<id>** 的订阅的 **id** 值。

d. 如果您更改了活跃订阅，请重新显示您的帐户信息：

```
$ az account show

{
  "environmentName": "AzureCloud",
  "id": "33212d16-bdf6-45cb-b038-f6565b61edda",
  "isDefault": true,
  "name": "Subscription Name",
  "state": "Enabled",
  "tenantId": "8049c7e9-c3de-762d-a54e-dc3f6be6a7ee",
  "user": {
    "name": "you@example.com",
    "type": "user"
  }
}
```

3. 记录前面输出中 **tenantId** 和 **id** 参数的值。OpenShift Container Platform 安装过程中需要这些值。

4. 为您的帐户创建服务主体：

```
$ az ad sp create-for-rbac --role Contributor --name <service_principal> 1
Changing "<service_principal>" to a valid URI of "http://<service_principal>", which is the
required format used for service principal names
Retrying role assignment creation: 1/36
Retrying role assignment creation: 2/36
Retrying role assignment creation: 3/36
Retrying role assignment creation: 4/36
{
  "appId": "8bd0d04d-0ac2-43a8-928d-705c598c6956",
  "displayName": "<service_principal>",
  "name": "http://<service_principal>",
  "password": "ac461d78-bf4b-4387-ad16-7e32e328aec6",
  "tenant": "6048c7e9-b2ad-488d-a54e-dc3f6be6a7ee"
}
```

1 将 **<service_principal>** 替换要您要分配给服务主体的名称。

5. 记录前面输出中 **appId** 和 **password** 参数的值。OpenShift Container Platform 安装过程中需要这些值。

6. 为服务主体授予额外权限。服务主体需要传统的 **Azure Active Directory Graph → Application.ReadWrite.OwnedBy** 权限以及集群的 **User Access Administrator** 角色，以便为其组件分配凭证。

   a. 要分配 **User Access Administrator** 角色，请运行以下命令：

```
$ az role assignment create --role "User Access Administrator" \
    --assignee-object-id $(az ad sp list --filter "appId eq '<appId>'" \ 1
      | jq '.[0].objectId' -r)
```

1 将 **<appId>** 替换为服务器主体的 **appId** 参数值。

b. 要分配 **Azure Active Directory Graph** 权限，请运行以下命令：

```
$ az ad app permission add --id <appId> \ 1
    --api 00000002-0000-0000-c000-000000000000 \
    --api-permissions 824c81eb-e3f8-4ee6-8f6d-de7f50d565b7=Role

    Invoking "az ad app permission grant --id 46d33abc-b8a3-46d8-8c84-f0fd58177435 --
api 00000002-0000-0000-c000-000000000000" is needed to make the change effective
```

1 将 **<appId>** 替换为服务器主体的 **appId** 参数值。

如需进一步了解可通过此命令授予的具体权限，请参阅 Windows Azure Active Directory 权限的 GUID 表。

c. 批准权限请求。如果您的帐户没有 Azure Active Directory 租户管理员角色，请按照您的组织的准则请租户管理员批准您的权限请求。

```
$ az ad app permission grant --id <appId> \ 1
    --api 00000002-0000-0000-c000-000000000000
```

1 将 **<appId>** 替换为服务器主体的 **appId** 参数值。

1.7.3.7. 支持的 Azure 区域

安装程序会根据您的订阅动态地生成可用的 Microsoft Azure 区域列表。OpenShift Container Platform 4.4.0 中已测试并验证了以下 Azure 区域：

- **australiacentral** (Australia Central)
- **australiaeast** (Australia East)
- **australiasoutheast** (Australia South East)
- **brazilsouth** (Brazil South)
- **canadacentral** (Canada Central)
- **canadaeast** (Canada East)
- **centralindia** (Central India)
- **centralus** (Central US)
- **eastasia** (East Asia)
- **eastus** (East US)

- **eastus2** (East US 2)
- **francecentral** (France Central)
- **germanywestcentral** (Germany West Central)
- **japaneast** (Japan East)
- **japanwest** (Japan West)
- **koreacentral** (Korea Central)
- **koreasouth** (Korea South)
- **northcentralus** (North Central US)
- **northeurope** (North Europe)
- **southafricanorth** (South Africa North)
- **southcentralus** (South Central US)
- **southeastasia** (Southeast Asia)
- **southindia** (South India)
- **switzerlandnorth** (Switzerland North)
- **uaenorth** (UAE North)
- **uksouth** (UK South)
- **ukwest** (UK West)
- **westcentralus** (West Central US)
- **westeurope** (West Europe)
- **westindia** (West India)
- **westus** (West US)
- **westus2** (West US 2)

1.7.4. 获取安装程序

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

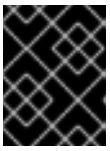
先决条件

- 必须从使用 Linux 或 macOS 的计算机安装集群。
- 需要 500 MB 本地磁盘空间来下载安装程序。

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。如果您有红帽帐号，请使用自己的凭证登录。如果没有，请创建一个帐户。

2. 进入适用于您的安装类型的页面，下载您的操作系统的安装程序，并将文件放在要保存安装配置文件的目录中。。



重要

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

重要

删除安装程序创建的文件不会删除您的集群，即使集群在安装过程中失败也是如此。您必须完成针对特定云供应商的 OpenShift Container Platform 卸载流程，才能完全删除您的集群。

3. 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 <installation_program>.tar.gz
```

4. 在 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中，下载您的安装 pull secret 的 **.txt** 文件。通过此 pull secret，您可以进行所含授权机构提供的服务的身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

1.7.5. 生成 SSH 私钥并将其添加到代理中

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

注意

在生产环境中，您需要进行灾难恢复和调试。

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

注意

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

流程

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

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

1 指定 SSH 密钥的路径和文件名，如 **~/.ssh/id_rsa**。不要指定已存在的 SSH 密钥，因为它会被覆盖。

运行此命令会在指定的位置生成不需要密码的 SSH 密钥。

2. 作为后台任务启动 **ssh-agent** 进程：

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

Agent pid 31874
```

3. 将 SSH 私钥添加到 **ssh-agent**：

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

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

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

后续步骤

- 在安装 OpenShift Container Platform 时，为安装程序提供 SSH 公钥。如果在您置备的基础架构上安装集群，您必须将此密钥提供给集群的机器。

1.7.6. 创建用于 Azure 的安装文件

要使用用户置备的基础架构在 Microsoft Azure 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。

1.7.6.1. 创建安装配置文件

您可以自定义在 Microsoft Azure 上安装的 OpenShift Container Platform 集群。

先决条件

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

流程

1. 创建 **install-config.yaml** 文件。
   a. 运行以下命令：

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

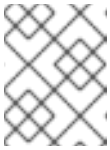
(1) 对于 **<installation_directory>**，请指定用于保存安装程序所创建的文件的目录名称。

重要

指定一个空目录。一些安装信息，如 bootstrap X.509 证书，有较短的过期间隔，因此不要重复使用安装目录。如果要重复使用另一个集群安装中的个别文件，可以将其复制到您的目录中。但是，一些安装数据的文件名可能会在发行版本之间有所改变。从 OpenShift Container Platform 老版本中复制安装文件时要格外小心。

b. 在提示符处，提供您的云的配置详情：

i. 可选：选择用来访问集群机器的 SSH 密钥。



注意

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

ii. 选择 **azure** 作为目标平台。

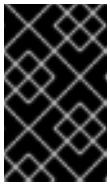
iii. 如果计算机上没有 Microsoft Azure 配置集，请为您的订阅和服务主体指定以下 Azure 参数值：

- **azure subscription id**：要用于集群的订阅 ID。指定帐户输出中的 **id** 值。
- **azure tenant id**：租户 ID。指定帐户输出中的 **tenantId** 值。
- **azure service principal client id**：服务主体的 **appId** 参数值。
- **azure service principal client secret**：服务主体的 **password** 参数值。

iv. 选择要在其中部署集群的区域。

v. 选择集群要部署到的基域。基域与您为集群创建的 Azure DNS 区对应。

vi. 为集群输入一个描述性名称。



重要

所有通过公共端点提供的 Azure 资源均存在资源名称的限制，您无法创建使用某些名称的资源。如需 Azure 限制词语列表，请参阅 Azure 文档中的解决保留资源名称错误。

vii. 粘贴从 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面中获取的 pull secret。

2. 修改 **install-config.yaml** 文件。您可以在**安装配置参数**部分中找到有关可用参数的更多信息。

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

重要

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

1.7.6.2. 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

- 现有的 **install-config.yaml** 文件。

- 查看集群需要访问的站点，并决定是否需要绕过代理。默认情况下代理所有集群出口流量，包括对托管云供应商 API 的调用。如果需要，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：

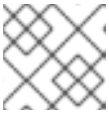
```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: http://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
...
```

1. 用于创建集群外 HTTP 连接的代理 URL。URL 必须是 **http**。如果您使用不要求额外代理配置但需要额外 CA 的 MITM 透明代理网络，则不得指定 **httpProxy** 值。

2. 用于创建集群外 HTTPS 连接的代理 URL。如果未指定此字段，**httpProxy** 会同时用于 HTTP 和 HTTPS 连接。如果您使用不要求额外代理配置但需要额外 CA 的 MITM 透明代理网络，则不得指定 **httpsProxy** 值。

3. 要排除代理的目标域名、域、IP 地址或其他网络 CIDR 的逗号分隔列表。域之前加上前缀 **.** 可包含该域的所有子域。使用 * 可对所有目的地绕过所有代理。

4. 如果提供，安装程序会在 **openshift-config** 命名空间中生成名为 **user-ca-bundle** 的配置映射，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。如果您使用不要求额外代理配置但需要额外 CA 的 MITM 透明代理网络，您必须提供 MITM CA 证书。



注意

安装程序不支持代理的 **readinessEndpoints** 字段。

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

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

注意

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

1.7.6.3. 为 ARM 模板导出常用变量

您必须导出与提供的 Azure Resource Manager（ARM）模板搭配使用的一组常用变量，它们有助于在 Microsoft Azure 上完成用户提供基础架构安装。

注意

特定的 ARM 模板可能还需要其他导出变量，这些变量在相关的程序中详细介绍。

先决条件

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

流程

1. 导出 **install-config.yaml** 中由提供的 ARM 模板使用的通用变量：

```
$ export CLUSTER_NAME=<cluster_name>(1)
$ export AZURE_REGION=<azure_region>(2)
$ export SSH_KEY=<ssh_key>(3)
$ export BASE_DOMAIN=<base_domain>(4)
$ export BASE_DOMAIN_RESOURCE_GROUP=<base_domain_resource_group>(5)
```

(1) **install-config.yaml** 文件中的**.metadata.name** 属性的值。

(2) 集群要部署到的区域，如 **centralus** 。这是来自 **install-config.yaml** 文件中的 **.platform.azure.region** 属性的值。

(3) 作为字符串的 SSH RSA 公钥文件。您必须使用引号包括 SSH 密钥，因为它包含空格。这是 **install-config.yaml 文件中的 .sshKey** 属性的值。

(4) 集群要部署到的基域。基域与您为集群创建的公共 DNS 区对应。这是 **install-config.yaml 文件中的 .baseDomain** 属性的值。

(5) 公共 DNS 区所在的资源组。这是 **install-config.yaml 文件中的 .platform.azure.baseDomainResourceGroupName** 属性的值。

例如：

```
$ export CLUSTER_NAME=test-cluster
$ export AZURE_REGION=centralus
$ export SSH_KEY="ssh-rsa xxx/xxx/xxx= user@email.com"
$ export BASE_DOMAIN=example.com
$ export BASE_DOMAIN_RESOURCE_GROUP=ocp-cluster
```

2. 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。

1.7.6.4. 创建 Kubernetes 清单和 Ignition 配置文件

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

重要

安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrapper** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅*从过期的 control plane 证书中恢复*的文档。

先决条件

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

流程

1. 为集群生成 Kubernetes 清单：

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

INFO Consuming Install Config from target directory
WARNING Making control-plane schedulable by setting MastersSchedulable to true for
Scheduler cluster settings
```

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

由于您稍后会在安装过程中自行创建计算机器，因此可以忽略这个警告。

2. 删除定义 control plane 机器的 Kubernetes 清单文件：

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

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

3. 删除定义 worker 机器的 Kubernetes 清单文件：

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

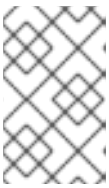
由于您要自行创建并管理 worker 机器，因此不需要初始化这些机器。

4. 修改 **<installation_directory>/manifests/cluster-scheduler-02-config.yml** Kubernetes 清单文件，以防止在 control plane 机器上调度 Ppd：

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

b. 找到 **mastersSchedulable** 参数，并将其值设为 **False**。

c. 保存并退出文件。



注意

目前，由于 Kubernetes 限制，入口负载均衡器将无法访问在 control plane 机器上运行的路由器 Pod。以后的 OpenShift Container Platform 次要版本中可能不需要这一步骤。

5. 可选：如果您不希望 Ingress Operator 代表您创建 DNS 记录，请删除 **<installation_directory>/manifests/cluster-dns-02-config.yml** DNS 配置文件中的 **privateZone** 和 **publicZone** 部分：

```
apiVersion: config.openshift.io/v1
kind: DNS
metadata:
  creationTimestamp: null
  name: cluster
spec:
  baseDomain: example.openshift.com
  privateZone: 1
    id: mycluster-100419-private-zone
  publicZone: 2
    id: example.openshift.com
status: {}
```

1 2 彻底删除这些部分。

如果您这样做，后续步骤中必须手动添加入口 DNS 记录。

6. 在用户置备的基础架构上配置 Azure 时，您必须导出清单文件中定义的一些常见变量，以备稍后在 Azure Resource Manager（ARM）模板中使用：

```
$ export INFRA_ID=<infra_id> 1
$ export RESOURCE_GROUP=<resource_group> 2
```

1 OpenShift Container Platform 集群被分配了一个标识符（**INFRA_ID**），其格式为 **<cluster_name>-<random_string>**。这将作为使用提供的 ARM 模板创建的大部分资源的基本名称。这是 **manifests/cluster-infrastructure-02-config.yml 文件中的 .status.infrastructureName** 属性的值。

2 此 Azure 部署中创建的所有资源都作为资源组的一部分。资源组名称还基于 **INFRA_ID**，格式为 **<cluster_name>-<random_string>-rg**。这是 **manifests/cluster-infrastructure-02-config.yml 文件中的 .status.platformStatus.azure.resourceGroupName** 属性的值。

7. 获取 Ignition 配置文件：

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

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

该目录中将生成以下文件：

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

1.7.7. 创建 Azure 资源组和身份

您必须创建一个 Microsoft Azure 资源组以及该资源组的身份。它们都用于在 Azure 上安装 OpenShift Container Platform 集群。

先决条件

- 配置 Azure 帐户。
- 为集群生成 Ignition 配置文件。

流程

1. 在受支持的 Azure 区域中创建资源组：

```
$ az group create --name ${RESOURCE_GROUP} --location ${AZURE_REGION}
```

2. 为资源组创建 Azure 身份：

```
$ az identity create -g ${RESOURCE_GROUP} -n ${INFRA_ID}-identity
```

这用于授予集群中 Operator 所需的访问权限。例如，这允许 Ingress Operator 创建公共 IP 及其负载均衡器。您必须将 Azure 身份分配给角色。

3. 将 Contributor 角色授予 Azure 身份：

   a. 导出 Azure 角色分配所需的以下变量：

```
$ export PRINCIPAL_ID=`az identity show -g ${RESOURCE_GROUP} -n ${INFRA_ID}-identity --query principalId --out tsv`
$ export RESOURCE_GROUP_ID=`az group show -g ${RESOURCE_GROUP} --query id --out tsv`
```

   b. 将 Contributor 角色分配给身份：

```
$ az role assignment create --assignee "${PRINCIPAL_ID}" --role 'Contributor' --scope "${RESOURCE_GROUP_ID}"
```

1.7.8. 上传 RHCOS 集群镜像和 bootstrap Ignition 配置文件

Azure 客户端不支持基于本地现有文件进行的部署，因此您必须复制 RHCOS 虚拟硬盘（VHD）集群镜像，并将 bootstrap ignition 配置文件存储在存储容器中，以便在部署过程中访问这些文件。

先决条件

- 配置 Azure 帐户。
- 为集群生成 Ignition 配置文件。

流程

1. 创建 Azure 存储帐户以存储 VHD 集群镜像：

```
$ az storage account create -g ${RESOURCE_GROUP} --location ${AZURE_REGION} --
name ${CLUSTER_NAME}sa --kind Storage --sku Standard_LRS
```

警告

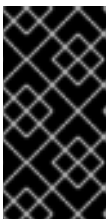
Azure 存储帐户名称的长度必须在 3 到 24 个字符之，且只使用数字和小写字母。如果您的 **CLUSTER_NAME** 变量没有遵循这些限制，您必须手动定义 Azure 存储帐户名称。如需有关 Azure 存储帐户名称限制的更多信息，请参阅 Azure 文档中的解决存储帐户名称的错误。

2. 将存储帐户密钥导出为环境变量：

```
$ export ACCOUNT_KEY=`az storage account keys list -g ${RESOURCE_GROUP} --
account-name ${CLUSTER_NAME}sa --query "[0].value" -o tsv`
```

3. 选择 RHCOS 版本以使用并导出 VHD 的 URL 到环境变量：

```
$ export VHD_URL=`curl -s https://raw.githubusercontent.com/openshift/installer/release-
4.4/data/data/rhcos.json | jq -r .azure.url`
```



重要

RHCOS 镜像可能不会随着 OpenShift Container Platform 的每一发行版本都有改变。您必须指定一个最高版本的镜像，其版本号应小于或等于您安装的 OpenShift Container Platform 版本。如果可用，请使用与 OpenShift Container Platform 版本匹配的镜像版本。

4. 将所选 VHD 复制到 blob:

```
$ az storage container create --name vhd --account-name ${CLUSTER_NAME}sa --account-
key ${ACCOUNT_KEY}
$ az storage blob copy start --account-name ${CLUSTER_NAME}sa --account-key
${ACCOUNT_KEY} --destination-blob "rhcos.vhd" --destination-container vhd --source-uri
"${VHD_URL}"
```

要跟踪 VHD 复制任务的进程，请运行这个脚本：

```
status="unknown"
while [ "$status" != "success" ]
do
  status=`az storage blob show --container-name vhd --name "rhcos.vhd" --account-name
${CLUSTER_NAME}sa --account-key ${ACCOUNT_KEY} -o tsv --query
properties.copy.status`
  echo $status
done
```

5. 创建 blob 存储容器并上传生成的 **bootstrap.ign** 文件：

```
$ az storage container create --name files --account-name ${CLUSTER_NAME}sa --
account-key ${ACCOUNT_KEY} --public-access blob
$ az storage blob upload --account-name ${CLUSTER_NAME}sa --account-key
${ACCOUNT_KEY} -c "files" -f "<installation_directory>/bootstrap.ign" -n "bootstrap.ign"
```

1.7.9. 创建 DNS 区示例

使用用户置备的基础架构的集群需要 DNS 记录。您应该选择适合您的场景的 DNS 策略。

在本例中，使用了 Azure 的 DNS 解决方案，因此您将为外部（内部网络）可见性创建一个新的公共 DNS 区域，并为内部集群解析创建一个私有 DNS 区域。

注意

公共 DNS 区域不需要与集群部署位于同一个资源组中，且可能已在您的机构中为所需基域存在。如果情况如此，您可以跳过创建公共 DNS 区这一步 ; 请确定您之前生成的安装配置反映了这种情况。

先决条件

- 配置 Azure 帐户。
- 为集群生成 Ignition 配置文件。

流程

1. 在 **BASE_DOMAIN_RESOURCE_GROUP** 环境变量中导出的资源组中创建新的公共 DNS 区域：

```
$ az network dns zone create -g ${BASE_DOMAIN_RESOURCE_GROUP} -n
${CLUSTER_NAME}.${BASE_DOMAIN}
```

   如果您使用的是公共 DNS 区域，可以跳过这一步。

2. 在与这个部署的其余部分相同的资源组中创建私有 DNS 区域：

```
$ az network private-dns zone create -g ${RESOURCE_GROUP} -n
${CLUSTER_NAME}.${BASE_DOMAIN}
```

如需了解更多信息，请参阅在 Azure 中配置公共 DNS 的信息。

1.7.10. 在 Azure 中创建 VNet

您必须在 Microsoft Azure 中创建虚拟网络（VNet），供您的 OpenShift Container Platform 集群使用。您可以对 VNet 进行定制来满足您的要求。创建 VNet 的一种方法是修改提供的 Azure Resource Manager（ARM）模板。

注意

如果不使用提供的 ARM 模板来创建 Azure 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

- 配置 Azure 帐户。
- 为集群生成 Ignition 配置文件。

流程

1. 复制 **VNet 的 ARM 模板** 一节中的模板，并将它以 **01_vnet.json** 保存到集群的安装目录中。此模板描述了集群所需的 VNet。
2. 使用 **az** CLI 创建部署：

```
$ az deployment group create -g ${RESOURCE_GROUP} \
  --template-file "<installation_directory>/01_vnet.json" \
  --parameters baseName="${INFRA_ID}" 1
```

1 资源名称使用的基本名称；这通常是集群的基础架构 ID。

3. 将 VNet 模板链接到私有 DNS 区域：

```
$ az network private-dns link vnet create -g ${RESOURCE_GROUP} -z
${CLUSTER_NAME}.${BASE_DOMAIN} -n ${INFRA_ID}-network-link -v "${INFRA_ID}-vnet"
-e false
```

1.7.10.1. VNet 的 ARM 模板

您可以使用以下 Azure Resource Manager（ARM）模板来部署 OpenShift Container Platform 集群所需的 VPC：

例 1.1. 01_vnet.json ARM 模板

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-
01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
```

```
      }
    }
  },
  "variables" : {
    "location" : "[resourceGroup().location]",
    "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
    "addressPrefix" : "10.0.0.0/16",
    "masterSubnetName" : "[concat(parameters('baseName'), '-master-subnet')]",
    "masterSubnetPrefix" : "10.0.0.0/24",
    "nodeSubnetName" : "[concat(parameters('baseName'), '-worker-subnet')]",
    "nodeSubnetPrefix" : "10.0.1.0/24",
    "controlPlaneNsgName" : "[concat(parameters('baseName'), '-controlplane-nsg')]",
    "nodeNsgName" : "[concat(parameters('baseName'), '-node-nsg')]"
  },
  "resources" : [
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/virtualNetworks",
      "name" : "[variables('virtualNetworkName')]",
      "location" : "[variables('location')]",
      "dependsOn" : [
        "[concat('Microsoft.Network/networkSecurityGroups/', variables('controlPlaneNsgName'))]",
        "[concat('Microsoft.Network/networkSecurityGroups/', variables('nodeNsgName'))]"
      ],
      "properties" : {
        "addressSpace" : {
          "addressPrefixes" : [
            "[variables('addressPrefix')]"
          ]
        },
        "subnets" : [
          {
            "name" : "[variables('masterSubnetName')]",
            "properties" : {
              "addressPrefix" : "[variables('masterSubnetPrefix')]",
              "serviceEndpoints": [],
              "networkSecurityGroup" : {
                "id" : "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('controlPlaneNsgName'))]"
              }
            }
          },
          {
            "name" : "[variables('nodeSubnetName')]",
            "properties" : {
              "addressPrefix" : "[variables('nodeSubnetPrefix')]",
              "serviceEndpoints": [],
              "networkSecurityGroup" : {
                "id" : "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('nodeNsgName'))]"
              }
            }
          }
        ]
      }
    },
```

```
    {
      "type" : "Microsoft.Network/networkSecurityGroups",
      "name" : "[variables('controlPlaneNsgName')]",
      "apiVersion" : "2018-10-01",
      "location" : "[variables('location')]",
      "properties" : {
        "securityRules" : [
          {
            "name" : "apiserver_in",
            "properties" : {
              "protocol" : "Tcp",
              "sourcePortRange" : "*",
              "destinationPortRange" : "6443",
              "sourceAddressPrefix" : "*",
              "destinationAddressPrefix" : "*",
              "access" : "Allow",
              "priority" : 101,
              "direction" : "Inbound"
            }
          }
        ]
      }
    },
    {
      "type" : "Microsoft.Network/networkSecurityGroups",
      "name" : "[variables('nodeNsgName')]",
      "apiVersion" : "2018-10-01",
      "location" : "[variables('location')]",
      "properties" : {
        "securityRules" : [
          {
            "name" : "apiserver_in",
            "properties" : {
              "protocol" : "Tcp",
              "sourcePortRange" : "*",
              "destinationPortRange" : "6443",
              "sourceAddressPrefix" : "*",
              "destinationAddressPrefix" : "*",
              "access" : "Allow",
              "priority" : 101,
              "direction" : "Inbound"
            }
          }
        ]
      }
    }
  ]
}
```

1.7.11. 为 Azure 基础架构创建 RHCOS 集群镜像

您必须对 OpenShift Container Platform 节点的 Microsoft Azure 使用有效的 Red Hat Enterprise Linux CoreOS（RHCOS）镜像。

先决条件

先决条件

- 配置 Azure 帐户。
- 为集群生成 Ignition 配置文件。
- 将 RHCOS 虚拟硬盘（VHD）集群镜像存储在 Azure 存储容器中。
- 在 Azure 存储容器中存储 bootstrap Ignition 配置文件。

流程

1. 复制**镜像存储的 ARM 模板** 部分中的模板，并将它以 **02_storage.json** 保存到集群的安装目录中。此模板描述了集群所需的镜像存储。
2. 以一个变量的形式将 RHCOS VHD blob URL 导出：

```
$ export VHD_BLOB_URL=`az storage blob url --account-name ${CLUSTER_NAME}sa --
account-key ${ACCOUNT_KEY} -c vhd -n "rhcos.vhd" -o tsv`
```

3. 部署集群镜像

```
$ az deployment group create -g ${RESOURCE_GROUP} \
  --template-file "<installation_directory>/02_storage.json" \
  --parameters vhdBlobURL="${VHD_BLOB_URL}" \ 1
  --parameters baseName="${INFRA_ID}" 2
```

1. 用于创建 master 和 worker 机器的 RHCOS VHD 的 blob URL。
2. 资源名称使用的基本名称；这通常是集群的基础架构 ID。

1.7.11.1. 镜像存储的 ARM 模板

您可以使用以下 Azure Resource Manager（ARM）模板来部署 OpenShift Container Platform 集群所需的存储的 Red Hat Enterprise Linux CoreOS（RHCOS）镜像：

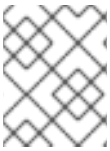
例 1.2. 02_storage.json ARM 模板

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-
01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "vhdBlobURL" : {
      "type" : "string",
      "metadata" : {
        "description" : "URL pointing to the blob where the VHD to be used to create master and
```

```
worker machines is located"
      }
    }
  },
  "variables" : {
    "location" : "[resourceGroup().location]",
    "imageName" : "[concat(parameters('baseName'), '-image')]"
  },
  "resources" : [
    {
      "apiVersion" : "2018-06-01",
      "type": "Microsoft.Compute/images",
      "name": "[variables('imageName')]",
      "location" : "[variables('location')]",
      "properties": {
        "storageProfile": {
          "osDisk": {
            "osType": "Linux",
            "osState": "Generalized",
            "blobUri": "[parameters('vhdBlobURL')]",
            "storageAccountType": "Standard_LRS"
          }
        }
      }
    }
  ]
}
```

1.7.12. 在 Azure 中创建网络和负载均衡组件

您必须在 Microsoft Azure 中配置网络和负载均衡，供您的 OpenShift Container Platform 集群使用。创建这些组件的一种方法是修改提供的 Azure Resource Manager（ARM）模板。



注意

如果不使用提供的 ARM 模板来创建 Azure 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

- 配置 Azure 帐户。
- 为集群生成 Ignition 配置文件。
- 在 Azure 中创建和配置 VNet 及相关子网。

流程

1. 复制**网络和负载均衡器的 ARM 模板**一节中的模板，并将以 **03_infra.json** 保存到集群的安装目录中。此模板描述了集群所需的网络和负载均衡对象。
2. 使用 **az** CLI 创建部署：

```
$ az deployment group create -g ${RESOURCE_GROUP} \
  --template-file "<installation_directory>/03_infra.json" \
  --parameters privateDNSZoneName="${CLUSTER_NAME}.${BASE_DOMAIN}" \ 1
  --parameters baseName="${INFRA_ID}" 2
```

1. 私有 DNS 区的名称。
2. 资源名称使用的基本名称；这通常是集群的基础架构 ID。

3. 在公共区为 API 公共负载均衡器创建一个 **api** DNS 记录。**${BASE_DOMAIN_RESOURCE_GROUP}** 变量必须指向存在公共 DNS 区的资源组。

   a. 导出以下变量：

   ```
   $ export PUBLIC_IP=`az network public-ip list -g ${RESOURCE_GROUP} --query "[?
   name=='${INFRA_ID}-master-pip'] | [0].ipAddress" -o tsv`
   ```

   b. 在一个新的公共区中创建 DNS 记录：

   ```
   $ az network dns record-set a add-record -g ${BASE_DOMAIN_RESOURCE_GROUP} -
   z ${CLUSTER_NAME}.${BASE_DOMAIN} -n api -a ${PUBLIC_IP} --ttl 60
   ```

   c. 如果您要将集群添加到现有的公共区，您可以在其中创建 DNS 记录：

   ```
   $ az network dns record-set a add-record -g ${BASE_DOMAIN_RESOURCE_GROUP} -
   z ${BASE_DOMAIN} -n api.${CLUSTER_NAME} -a ${PUBLIC_IP} --ttl 60
   ```

1.7.12.1. 网络和负载均衡器的 ARM 模板

您可以使用以下 Azure Resource Manager（ARM）模板来部署 OpenShift Container Platform 集群所需的网络对象和负载均衡器：

例 1.3. 03_infra.json ARM 模板

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-
01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "privateDNSZoneName" : {
      "type" : "string",
      "metadata" : {
        "description" : "Name of the private DNS zone"
      }
    }
  },
```

```
  "variables" : {
    "location" : "[resourceGroup().location]",
    "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
    "virtualNetworkID" : "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]",
    "masterSubnetName" : "[concat(parameters('baseName'), '-master-subnet')]",
    "masterSubnetRef" : "[concat(variables('virtualNetworkID'), '/subnets/',
variables('masterSubnetName'))]",
    "masterPublicIpAddressName" : "[concat(parameters('baseName'), '-master-pip')]",
    "masterPublicIpAddressID" : "[resourceId('Microsoft.Network/publicIPAddresses',
variables('masterPublicIpAddressName'))]",
    "masterLoadBalancerName" : "[concat(parameters('baseName'), '-public-lb')]",
    "masterLoadBalancerID" : "[resourceId('Microsoft.Network/loadBalancers',
variables('masterLoadBalancerName'))]",
    "internalLoadBalancerName" : "[concat(parameters('baseName'), '-internal-lb')]",
    "internalLoadBalancerID" : "[resourceId('Microsoft.Network/loadBalancers',
variables('internalLoadBalancerName'))]",
    "skuName": "Standard"
  },
  "resources" : [
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/publicIPAddresses",
      "name" : "[variables('masterPublicIpAddressName')]",
      "location" : "[variables('location')]",
      "sku": {
        "name": "[variables('skuName')]"
      },
      "properties" : {
        "publicIPAllocationMethod" : "Static",
        "dnsSettings" : {
          "domainNameLabel" : "[variables('masterPublicIpAddressName')]"
        }
      }
    },
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/loadBalancers",
      "name" : "[variables('masterLoadBalancerName')]",
      "location" : "[variables('location')]",
      "sku": {
        "name": "[variables('skuName')]"
      },
      "dependsOn" : [
        "[concat('Microsoft.Network/publicIPAddresses/', variables('masterPublicIpAddressName'))]"
      ],
      "properties" : {
        "frontendIPConfigurations" : [
          {
            "name" : "public-lb-ip",
            "properties" : {
              "publicIPAddress" : {
                "id" : "[variables('masterPublicIpAddressID')]"
              }
            }
          }
```

```
        ],
        "backendAddressPools" : [
          {
            "name" : "public-lb-backend"
          }
        ],
        "loadBalancingRules" : [
          {
            "name" : "api-internal",
            "properties" : {
              "frontendIPConfiguration" : {
                "id" :"[concat(variables('masterLoadBalancerID'), '/frontendIPConfigurations/public-lb-
ip')]"
              },
              "backendAddressPool" : {
                "id" : "[concat(variables('masterLoadBalancerID'), '/backendAddressPools/public-lb-
backend')]"
              },
              "protocol" : "Tcp",
              "loadDistribution" : "Default",
              "idleTimeoutInMinutes" : 30,
              "frontendPort" : 6443,
              "backendPort" : 6443,
              "probe" : {
                "id" : "[concat(variables('masterLoadBalancerID'), '/probes/api-internal-probe')]"
              }
            }
          }
        ],
        "probes" : [
          {
            "name" : "api-internal-probe",
            "properties" : {
              "protocol" : "Https",
              "port" : 6443,
              "requestPath": "/readyz",
              "intervalInSeconds" : 10,
              "numberOfProbes" : 3
            }
          }
        ]
      }
    },
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/loadBalancers",
      "name" : "[variables('internalLoadBalancerName')]",
      "location" : "[variables('location')]",
      "sku": {
        "name": "[variables('skuName')]"
      },
      "properties" : {
        "frontendIPConfigurations" : [
          {
            "name" : "internal-lb-ip",
            "properties" : {
```

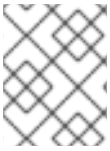
```
            "privateIPAllocationMethod" : "Dynamic",
            "subnet" : {
              "id" : "[variables('masterSubnetRef')]"
            },
            "privateIPAddressVersion" : "IPv4"
          }
        }
      ],
      "backendAddressPools" : [
        {
          "name" : "internal-lb-backend"
        }
      ],
      "loadBalancingRules" : [
        {
          "name" : "api-internal",
          "properties" : {
            "frontendIPConfiguration" : {
              "id" : "[concat(variables('internalLoadBalancerID'), '/frontendIPConfigurations/internal-lb-
ip')]"
            },
            "frontendPort" : 6443,
            "backendPort" : 6443,
            "enableFloatingIP" : false,
            "idleTimeoutInMinutes" : 30,
            "protocol" : "Tcp",
            "enableTcpReset" : false,
            "loadDistribution" : "Default",
            "backendAddressPool" : {
              "id" : "[concat(variables('internalLoadBalancerID'), '/backendAddressPools/internal-lb-
backend')]"
            },
            "probe" : {
              "id" : "[concat(variables('internalLoadBalancerID'), '/probes/api-internal-probe')]"
            }
          }
        },
        {
          "name" : "sint",
          "properties" : {
            "frontendIPConfiguration" : {
              "id" : "[concat(variables('internalLoadBalancerID'), '/frontendIPConfigurations/internal-lb-
ip')]"
            },
            "frontendPort" : 22623,
            "backendPort" : 22623,
            "enableFloatingIP" : false,
            "idleTimeoutInMinutes" : 30,
            "protocol" : "Tcp",
            "enableTcpReset" : false,
            "loadDistribution" : "Default",
            "backendAddressPool" : {
              "id" : "[concat(variables('internalLoadBalancerID'), '/backendAddressPools/internal-lb-
backend')]"
            },
            "probe" : {
```

```
            "id" : "[concat(variables('internalLoadBalancerID'), '/probes/sint-probe')]"
          }
        }
      }
    ],
    "probes" : [
      {
        "name" : "api-internal-probe",
        "properties" : {
          "protocol" : "Https",
          "port" : 6443,
          "requestPath": "/readyz",
          "intervalInSeconds" : 10,
          "numberOfProbes" : 3
        }
      },
      {
        "name" : "sint-probe",
        "properties" : {
          "protocol" : "Https",
          "port" : 22623,
          "requestPath": "/healthz",
          "intervalInSeconds" : 10,
          "numberOfProbes" : 3
        }
      }
    ]
  }
},
{
  "apiVersion": "2018-09-01",
  "type": "Microsoft.Network/privateDnsZones/A",
  "name": "[concat(parameters('privateDNSZoneName'), '/api')]",
  "location" : "[variables('location')]",
  "dependsOn" : [
    "[concat('Microsoft.Network/loadBalancers/', variables('internalLoadBalancerName'))]"
  ],
  "properties": {
    "ttl": 60,
    "aRecords": [
      {
        "ipv4Address": "
[reference(variables('internalLoadBalancerName')).frontendIPConfigurations[0].properties.privateIP
Address]"
      }
    ]
  }
},
{
  "apiVersion": "2018-09-01",
  "type": "Microsoft.Network/privateDnsZones/A",
  "name": "[concat(parameters('privateDNSZoneName'), '/api-int')]",
  "location" : "[variables('location')]",
  "dependsOn" : [
    "[concat('Microsoft.Network/loadBalancers/', variables('internalLoadBalancerName'))]"
  ],
```

```
      "properties": {
        "ttl": 60,
        "aRecords": [
          {
            "ipv4Address": "
[reference(variables('internalLoadBalancerName')).frontendIPConfigurations[0].properties.privateIP
Address]"
          }
        ]
      }
    }
  ]
}
```

1.7.13. 在 Azure 中创建 bootstrap 机器

您必须在 Microsoft Azure 中创建 bootstrap 机器，以便在 OpenShift Container Platform 集群初始化过程中使用。创建此机器的一种方法是修改提供的 Azure Resource Manager（ARM）模板。



注意

如果不使用提供的 ARM 模板来创建 bootstrap 机器，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

- 配置 Azure 帐户。
- 为集群生成 Ignition 配置文件。
- 在 Azure 中创建和配置 VNet 及相关子网。
- 在 Azure 中创建和配置联网及负载均衡器。
- 创建 control plane 和计算角色。

流程

1. 复制 **bootstrap 机器的 ARM 模板**一节中的模板，并将它以 **04_bootstrap.json** 保存到集群的安装目录中。此模板描述了集群所需的 bootstrap 机器。
2. 导出 bootstrap 机器部署所需的以下变量：

```
$ export BOOTSTRAP_URL=`az storage blob url --account-name ${CLUSTER_NAME}sa --
account-key ${ACCOUNT_KEY} -c "files" -n "bootstrap.ign" -o tsv`
$ export BOOTSTRAP_IGNITION=`jq -rcnM --arg v "2.2.0" --arg url ${BOOTSTRAP_URL}
'{ignition:{version:$v,config:{replace:{source:$url}}}}' | base64 -w0`
```

3. 使用 **az** CLI 创建部署：

```
$ az deployment group create -g ${RESOURCE_GROUP} \
  --template-file "<installation_directory>/04_bootstrap.json" \
  --parameters bootstrapIgnition="${BOOTSTRAP_IGNITION}" \ 1
```

```
--parameters sshKeyData="${SSH_KEY}" \ 2
--parameters baseName="${INFRA_ID}" 3
```

1 bootstrap 集群的 bootstrap Ignition 内容。

2 作为字符串的 SSH RSA 公钥文件。

3 资源名称使用的基本名称 ; 这通常是集群的基础架构 ID。

1.7.13.1. bootstrap 机器的 ARM 模板

您可以使用以下 Azure Resource Manager（ARM）模板来部署 OpenShift Container Platform 集群所需的 bootstrap 机器：

例 1.4. 04_bootstrap.json ARM 模板

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "bootstrapIgnition" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Bootstrap ignition content for the bootstrap cluster"
      }
    },
    "sshKeyData" : {
      "type" : "securestring",
      "metadata" : {
        "description" : "SSH RSA public key file as a string."
      }
    },
    "bootstrapVMSize" : {
      "type" : "string",
      "defaultValue" : "Standard_D4s_v3",
      "allowedValues" : [
        "Standard_A2",
        "Standard_A3",
        "Standard_A4",
        "Standard_A5",
        "Standard_A6",
        "Standard_A7",
        "Standard_A8",
        "Standard_A9",
        "Standard_A10",
```

```
        "Standard_A11",
        "Standard_D2",
        "Standard_D3",
        "Standard_D4",
        "Standard_D11",
        "Standard_D12",
        "Standard_D13",
        "Standard_D14",
        "Standard_D2_v2",
        "Standard_D3_v2",
        "Standard_D4_v2",
        "Standard_D5_v2",
        "Standard_D8_v3",
        "Standard_D11_v2",
        "Standard_D12_v2",
        "Standard_D13_v2",
        "Standard_D14_v2",
        "Standard_E2_v3",
        "Standard_E4_v3",
        "Standard_E8_v3",
        "Standard_E16_v3",
        "Standard_E32_v3",
        "Standard_E64_v3",
        "Standard_E2s_v3",
        "Standard_E4s_v3",
        "Standard_E8s_v3",
        "Standard_E16s_v3",
        "Standard_E32s_v3",
        "Standard_E64s_v3",
        "Standard_G1",
        "Standard_G2",
        "Standard_G3",
        "Standard_G4",
        "Standard_G5",
        "Standard_DS2",
        "Standard_DS3",
        "Standard_DS4",
        "Standard_DS11",
        "Standard_DS12",
        "Standard_DS13",
        "Standard_DS14",
        "Standard_DS2_v2",
        "Standard_DS3_v2",
        "Standard_DS4_v2",
        "Standard_DS5_v2",
        "Standard_DS11_v2",
        "Standard_DS12_v2",
        "Standard_DS13_v2",
        "Standard_DS14_v2",
        "Standard_GS1",
        "Standard_GS2",
        "Standard_GS3",
        "Standard_GS4",
        "Standard_GS5",
        "Standard_D2s_v3",
        "Standard_D4s_v3",
```

```
        "Standard_D8s_v3"
      ],
      "metadata" : {
        "description" : "The size of the Bootstrap Virtual Machine"
      }
    }
  },
  "variables" : {
    "location" : "[resourceGroup().location]",
    "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
    "virtualNetworkID" : "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]",
    "masterSubnetName" : "[concat(parameters('baseName'), '-master-subnet')]",
    "masterSubnetRef" : "[concat(variables('virtualNetworkID'), '/subnets/',
variables('masterSubnetName'))]",
    "masterLoadBalancerName" : "[concat(parameters('baseName'), '-public-lb')]",
    "internalLoadBalancerName" : "[concat(parameters('baseName'), '-internal-lb')]",
    "sshKeyPath" : "/home/core/.ssh/authorized_keys",
    "identityName" : "[concat(parameters('baseName'), '-identity')]",
    "vmName" : "[concat(parameters('baseName'), '-bootstrap')]",
    "nicName" : "[concat(variables('vmName'), '-nic')]",
    "imageName" : "[concat(parameters('baseName'), '-image')]",
    "controlPlaneNsgName" : "[concat(parameters('baseName'), '-controlplane-nsg')]",
    "sshPublicIpAddressName" : "[concat(variables('vmName'), '-ssh-pip')]"
  },
  "resources" : [
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/publicIPAddresses",
      "name" : "[variables('sshPublicIpAddressName')]",
      "location" : "[variables('location')]",
      "sku": {
        "name": "Standard"
      },
      "properties" : {
        "publicIPAllocationMethod" : "Static",
        "dnsSettings" : {
          "domainNameLabel" : "[variables('sshPublicIpAddressName')]"
        }
      }
    },
    {
      "apiVersion" : "2018-06-01",
      "type" : "Microsoft.Network/networkInterfaces",
      "name" : "[variables('nicName')]",
      "location" : "[variables('location')]",
      "dependsOn" : [
        "[resourceId('Microsoft.Network/publicIPAddresses', variables('sshPublicIpAddressName'))]"
      ],
      "properties" : {
        "ipConfigurations" : [
          {
            "name" : "pipConfig",
            "properties" : {
              "privateIPAllocationMethod" : "Dynamic",
              "publicIPAddress": {
```

```
            "id": "[resourceId('Microsoft.Network/publicIPAddresses',
variables('sshPublicIpAddressName'))]"
          },
          "subnet" : {
            "id" : "[variables('masterSubnetRef')]"
          },
          "loadBalancerBackendAddressPools" : [
            {
              "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('masterLoadBalancerName'), '/backendAddressPools/public-lb-backend')]"
            },
            {
              "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('internalLoadBalancerName'), '/backendAddressPools/internal-lb-backend')]"
            }
          ]
        }
      }
    ]
  },
  {
    "apiVersion" : "2018-06-01",
    "type" : "Microsoft.Compute/virtualMachines",
    "name" : "[variables('vmName')]",
    "location" : "[variables('location')]",
    "identity" : {
      "type" : "userAssigned",
      "userAssignedIdentities" : {
        "[resourceID('Microsoft.ManagedIdentity/userAssignedIdentities/',
variables('identityName'))]" : {}
      }
    },
    "dependsOn" : [
      "[concat('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
    ],
    "properties" : {
      "hardwareProfile" : {
        "vmSize" : "[parameters('bootstrapVMSize')]"
      },
      "osProfile" : {
        "computerName" : "[variables('vmName')]",
        "adminUsername" : "core",
        "customData" : "[parameters('bootstrapIgnition')]",
        "linuxConfiguration" : {
          "disablePasswordAuthentication" : true,
          "ssh" : {
            "publicKeys" : [
              {
                "path" : "[variables('sshKeyPath')]",
                "keyData" : "[parameters('sshKeyData')]"
              }
            ]
          }
```

```
          }
        },
        "storageProfile" : {
          "imageReference": {
            "id": "[resourceId('Microsoft.Compute/images', variables('imageName'))]"
          },
          "osDisk" : {
            "name": "[concat(variables('vmName'),'_OSDisk')]",
            "osType" : "Linux",
            "createOption" : "FromImage",
            "managedDisk": {
              "storageAccountType": "Premium_LRS"
            },
            "diskSizeGB" : 100
          }
        },
        "networkProfile" : {
          "networkInterfaces" : [
            {
              "id" : "[resourceId('Microsoft.Network/networkInterfaces', variables('nicName'))]"
            }
          ]
        }
      },
      {
        "apiVersion" : "2018-06-01",
        "type": "Microsoft.Network/networkSecurityGroups/securityRules",
        "name" : "[concat(variables('controlPlaneNsgName'), '/bootstrap_ssh_in')]",
        "location" : "[variables('location')]",
        "dependsOn" : [
          "[resourceId('Microsoft.Compute/virtualMachines', variables('vmName'))]"
        ],
        "properties": {
          "protocol" : "Tcp",
          "sourcePortRange" : "*",
          "destinationPortRange" : "22",
          "sourceAddressPrefix" : "*",
          "destinationAddressPrefix" : "*",
          "access" : "Allow",
          "priority" : 100,
          "direction" : "Inbound"
        }
      }
    ]
  }
```

1.7.14. 在 Azure 中创建 control plane 机器

您必须在 Microsoft Azure 中创建 control plane 机器，供您的集群使用。创建这些机器的一种方法是修改提供的 Azure Resource Manager（ARM）模板。

注意

如果不使用提供的 ARM 模板来创建 control plane 机器，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

- 配置 Azure 帐户。
- 为集群生成 Ignition 配置文件。
- 在 Azure 中创建和配置 VNet 及相关子网。
- 在 Azure 中创建和配置联网及负载均衡器。
- 创建 control plane 和计算角色。
- 创建 bootstrap 机器。

流程

1. 复制 **control plane 机器的 ARM 模板**一节中的模板，并将它以 **05_masters.json** 保存到集群的安装目录中。此模板描述了集群所需的 control plane 机器。
2. 导出 control plane 机器部署所需的以下变量：

```
$ export MASTER_IGNITION=`cat <installation_directory>/master.ign | base64`
```

3. 使用 **az** CLI 创建部署：

```
$ az deployment group create -g ${RESOURCE_GROUP} \
  --template-file "<installation_directory>/05_masters.json" \
  --parameters masterIgnition="${MASTER_IGNITION}" \ 1
  --parameters sshKeyData="${SSH_KEY}" \ 2
  --parameters privateDNSZoneName="${CLUSTER_NAME}.${BASE_DOMAIN}" 3
  --parameters baseName="${INFRA_ID}" 4
```

1. master 节点的 Ignition 内容。
2. 作为字符串的 SSH RSA 公钥文件。
3. master 节点附加的私有 DNS 区域名称。
4. 资源名称使用的基本名称；这通常是集群的基础架构 ID。

1.7.14.1. control plane 机器的 ARM 模板

您可以使用以下 Azure Resource Manager（ARM）模板来部署 OpenShift Container Platform 集群所需的 control plane 机器：

例 1.5. 05_masters.json ARM 模板

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-
```

```
01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "masterIgnition" : {
      "type" : "string",
      "metadata" : {
        "description" : "Ignition content for the master nodes"
      }
    },
    "numberOfMasters" : {
      "type" : "int",
      "defaultValue" : 3,
      "minValue" : 2,
      "maxValue" : 30,
      "metadata" : {
        "description" : "Number of OpenShift masters to deploy"
      }
    },
    "sshKeyData" : {
      "type" : "securestring",
      "metadata" : {
        "description" : "SSH RSA public key file as a string"
      }
    },
    "privateDNSZoneName" : {
      "type" : "string",
      "metadata" : {
        "description" : "Name of the private DNS zone the master nodes are going to be attached to"
      }
    },
    "masterVMSize" : {
      "type" : "string",
      "defaultValue" : "Standard_D8s_v3",
      "allowedValues" : [
        "Standard_A2",
        "Standard_A3",
        "Standard_A4",
        "Standard_A5",
        "Standard_A6",
        "Standard_A7",
        "Standard_A8",
        "Standard_A9",
        "Standard_A10",
        "Standard_A11",
        "Standard_D2",
        "Standard_D3",
        "Standard_D4",
        "Standard_D11",
        "Standard_D12",
```

```
      "Standard_D13",
      "Standard_D14",
      "Standard_D2_v2",
      "Standard_D3_v2",
      "Standard_D4_v2",
      "Standard_D5_v2",
      "Standard_D8_v3",
      "Standard_D11_v2",
      "Standard_D12_v2",
      "Standard_D13_v2",
      "Standard_D14_v2",
      "Standard_E2_v3",
      "Standard_E4_v3",
      "Standard_E8_v3",
      "Standard_E16_v3",
      "Standard_E32_v3",
      "Standard_E64_v3",
      "Standard_E2s_v3",
      "Standard_E4s_v3",
      "Standard_E8s_v3",
      "Standard_E16s_v3",
      "Standard_E32s_v3",
      "Standard_E64s_v3",
      "Standard_G1",
      "Standard_G2",
      "Standard_G3",
      "Standard_G4",
      "Standard_G5",
      "Standard_DS2",
      "Standard_DS3",
      "Standard_DS4",
      "Standard_DS11",
      "Standard_DS12",
      "Standard_DS13",
      "Standard_DS14",
      "Standard_DS2_v2",
      "Standard_DS3_v2",
      "Standard_DS4_v2",
      "Standard_DS5_v2",
      "Standard_DS11_v2",
      "Standard_DS12_v2",
      "Standard_DS13_v2",
      "Standard_DS14_v2",
      "Standard_GS1",
      "Standard_GS2",
      "Standard_GS3",
      "Standard_GS4",
      "Standard_GS5",
      "Standard_D2s_v3",
      "Standard_D4s_v3",
      "Standard_D8s_v3"
    ],
    "metadata" : {
      "description" : "The size of the Master Virtual Machines"
    }
  },
```

```
    "diskSizeGB" : {
      "type" : "int",
      "defaultValue" : 1024,
      "metadata" : {
        "description" : "Size of the Master VM OS disk, in GB"
      }
    }
  },
  "variables" : {
    "location" : "[resourceGroup().location]",
    "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
    "virtualNetworkID" : "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]",
    "masterSubnetName" : "[concat(parameters('baseName'), '-master-subnet')]",
    "masterSubnetRef" : "[concat(variables('virtualNetworkID'), '/subnets/',
variables('masterSubnetName'))]",
    "masterLoadBalancerName" : "[concat(parameters('baseName'), '-public-lb')]",
    "internalLoadBalancerName" : "[concat(parameters('baseName'), '-internal-lb')]",
    "sshKeyPath" : "/home/core/.ssh/authorized_keys",
    "identityName" : "[concat(parameters('baseName'), '-identity')]",
    "imageName" : "[concat(parameters('baseName'), '-image')]",
    "copy" : [
      {
        "name" : "vmNames",
        "count" :  "[parameters('numberOfMasters')]",
        "input" : "[concat(parameters('baseName'), '-master-', copyIndex('vmNames'))]"
      }
    ]
  },
  "resources" : [
    {
      "apiVersion" : "2018-06-01",
      "type" : "Microsoft.Network/networkInterfaces",
      "copy" : {
        "name" : "nicCopy",
        "count" : "[length(variables('vmNames'))]"
      },
      "name" : "[concat(variables('vmNames')[copyIndex()], '-nic')]",
      "location" : "[variables('location')]",
      "properties" : {
        "ipConfigurations" : [
          {
            "name" : "pipConfig",
            "properties" : {
              "privateIPAllocationMethod" : "Dynamic",
              "subnet" : {
                "id" : "[variables('masterSubnetRef')]"
              },
              "loadBalancerBackendAddressPools" : [
                {
                  "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('masterLoadBalancerName'), '/backendAddressPools/public-lb-backend')]"
                },
                {
                  "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
```

```
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('internalLoadBalancerName'), '/backendAddressPools/internal-lb-backend')]"
              }
            ]
          }
        }
      ]
    }
  },
  {
    "apiVersion": "2018-09-01",
    "type": "Microsoft.Network/privateDnsZones/SRV",
    "name": "[concat(parameters('privateDNSZoneName'), '/_etcd-server-ssl._tcp')]",
    "location" : "[variables('location')]",
    "properties": {
      "ttl": 60,
      "copy": [{
        "name": "srvRecords",
        "count": "[length(variables('vmNames'))]",
        "input": {
          "priority": 0,
          "weight" : 10,
          "port" : 2380,
          "target" : "[concat('etcd-', copyIndex('srvRecords'), '.',
parameters('privateDNSZoneName'))]"
        }
      }]
    }
  },
  {
    "apiVersion": "2018-09-01",
    "type": "Microsoft.Network/privateDnsZones/A",
    "copy" : {
      "name" : "dnsCopy",
      "count" : "[length(variables('vmNames'))]"
    },
    "name": "[concat(parameters('privateDNSZoneName'), '/etcd-', copyIndex())]",
    "location" : "[variables('location')]",
    "dependsOn" : [
      "[concat('Microsoft.Network/networkInterfaces/', concat(variables('vmNames')[copyIndex()], '-
nic'))]"
    ],
    "properties": {
      "ttl": 60,
      "aRecords": [
        {
          "ipv4Address": "[reference(concat(variables('vmNames')[copyIndex()], '-
nic')).ipConfigurations[0].properties.privateIPAddress]"
        }
      ]
    }
  },
  {
    "apiVersion" : "2018-06-01",
    "type" : "Microsoft.Compute/virtualMachines",
    "copy" : {
```

```
        "name" : "vmCopy",
        "count" : "[length(variables('vmNames'))]"
      },
      "name" : "[variables('vmNames')[copyIndex()]]",
      "location" : "[variables('location')]",
      "identity" : {
        "type" : "userAssigned",
        "userAssignedIdentities" : {
          "[resourceID('Microsoft.ManagedIdentity/userAssignedIdentities/',
variables('identityName'))]" : {}
        }
      },
      "dependsOn" : [
        "[concat('Microsoft.Network/networkInterfaces/', concat(variables('vmNames')[copyIndex()], '-
nic'))]",
        "[concat('Microsoft.Network/privateDnsZones/', parameters('privateDNSZoneName'),
'/A/etcd-', copyIndex())]",
        "[concat('Microsoft.Network/privateDnsZones/', parameters('privateDNSZoneName'),
'/SRV/_etcd-server-ssl._tcp')]"
      ],
      "properties" : {
        "hardwareProfile" : {
          "vmSize" : "[parameters('masterVMSize')]"
        },
        "osProfile" : {
          "computerName" : "[variables('vmNames')[copyIndex()]]",
          "adminUsername" : "core",
          "customData" : "[parameters('masterIgnition')]",
          "linuxConfiguration" : {
            "disablePasswordAuthentication" : true,
            "ssh" : {
              "publicKeys" : [
                {
                  "path" : "[variables('sshKeyPath')]",
                  "keyData" : "[parameters('sshKeyData')]"
                }
              ]
            }
          }
        },
        "storageProfile" : {
          "imageReference": {
            "id": "[resourceId('Microsoft.Compute/images', variables('imageName'))]"
          },
          "osDisk" : {
            "name": "[concat(variables('vmNames')[copyIndex()], '_OSDisk')]",
            "osType" : "Linux",
            "createOption" : "FromImage",
            "caching": "ReadOnly",
            "writeAcceleratorEnabled": false,
            "managedDisk": {
              "storageAccountType": "Premium_LRS"
            },
            "diskSizeGB" : "[parameters('diskSizeGB')]"
          }
        },
```

```
        "networkProfile" : {
          "networkInterfaces" : [
            {
              "id" : "[resourceId('Microsoft.Network/networkInterfaces', concat(variables('vmNames')
[copyIndex()], '-nic'))]",
              "properties": {
                "primary": false
              }
            }
          ]
        }
      }
    }
  ]
}
```

1.7.15. 等待 bootstrap 完成并删除 Azure 中的 bootstrap 资源

在 Microsoft Azure 中创建所有所需的基础架构后，请等待您通过安装程序生成的 Ignition 配置文件所置备的机器上完成 bootstrap 过程。

先决条件

- 配置 Azure 帐户。
- 为集群生成 Ignition 配置文件。
- 在 Azure 中创建和配置 VNet 及相关子网。
- 在 Azure 中创建和配置联网及负载均衡器。
- 创建 control plane 和计算角色。
- 创建 bootstrap 机器。
- 创建 control plane 机器。

流程

1. 更改到包含安装程序的目录，再运行以下命令：

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

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

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

如果命令退出且不显示 **FATAL** 警告，则代表您的 control plane 已被初始化。

2. 删除 bootstrap 资源：

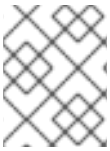
```
$ az network nsg rule delete -g ${RESOURCE_GROUP} --nsg-name ${INFRA_ID}-nsg --
```

```
name bootstrap_ssh_in
$ az vm stop -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap
$ az vm deallocate -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap
$ az vm delete -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap --yes
$ az disk delete -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap_OSDisk --no-
wait --yes
$ az network nic delete -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap-nic --no-
wait
$ az storage blob delete --account-key ${ACCOUNT_KEY} --account-name
${CLUSTER_NAME}sa --container-name files --name bootstrap.ign
$ az network public-ip delete -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap-
ssh-pip
```

1.7.16. 在 Azure 中创建额外的 worker 机器

您可以通过分散启动各个实例或利用集群外自动化流程（如自动缩放组），在 Microsoft Azure 中为您的集群创建 worker 机器。您还可以利用 OpenShift Container Platform 中的内置集群扩展机制和机器 API。

在本例中，您要使用 Azure Resource Manager（ARM） 模板来手动启动一个实例。通过在文件中添加类型为 **06_workers.json** 的其他资源，即可启动其他实例。



注意

如果不使用提供的 ARM 模板来创建 worker 机器，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

- 配置 Azure 帐户。
- 为集群生成 Ignition 配置文件。
- 在 Azure 中创建和配置 VNet 及相关子网。
- 在 Azure 中创建和配置联网及负载均衡器。
- 创建 control plane 和计算角色。
- 创建 bootstrap 机器。
- 创建 control plane 机器。

流程

1. 复制 **worker 机器的 ARM 模板**一节中的模板，并将它以 **06_workers.json** 保存到集群的安装目录中。此模板描述了集群所需的 worker 机器。
2. 导出 worker 机器部署所需的以下变量：

```
$ export WORKER_IGNITION=`cat <installation_directory>/worker.ign | base64`
```

3. 使用 **az** CLI 创建部署：

```
$ az deployment group create -g ${RESOURCE_GROUP} \
```

```
--template-file "<installation_directory>/06_workers.json" \
--parameters workerIgnition="${WORKER_IGNITION}" \ 1
--parameters sshKeyData="${SSH_KEY}" \ 2
--parameters baseName="${INFRA_ID}" 3
```

1 worker 节点的 Ignition 内容。

2 作为字符串的 SSH RSA 公钥文件。

3 资源名称使用的基本名称 ; 这通常是集群的基础架构 ID。

1.7.16.1. worker 机器的 ARM 模板

您可以使用以下 Azure Resource Manager（ARM）模板来部署 OpenShift Container Platform 集群所需的 worker 机器：

例 1.6. 06_workers.json ARM 模板

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "workerIgnition" : {
      "type" : "string",
      "metadata" : {
        "description" : "Ignition content for the worker nodes"
      }
    },
    "numberOfNodes" : {
      "type" : "int",
      "defaultValue" : 3,
      "minValue" : 2,
      "maxValue" : 30,
      "metadata" : {
        "description" : "Number of OpenShift compute nodes to deploy"
      }
    },
    "sshKeyData" : {
      "type" : "securestring",
      "metadata" : {
        "description" : "SSH RSA public key file as a string"
      }
    },
    "nodeVMSize" : {
      "type" : "string",
      "defaultValue" : "Standard_D4s_v3",
```

```
    "allowedValues" : [
      "Standard_A2",
      "Standard_A3",
      "Standard_A4",
      "Standard_A5",
      "Standard_A6",
      "Standard_A7",
      "Standard_A8",
      "Standard_A9",
      "Standard_A10",
      "Standard_A11",
      "Standard_D2",
      "Standard_D3",
      "Standard_D4",
      "Standard_D11",
      "Standard_D12",
      "Standard_D13",
      "Standard_D14",
      "Standard_D2_v2",
      "Standard_D3_v2",
      "Standard_D4_v2",
      "Standard_D5_v2",
      "Standard_D8_v3",
      "Standard_D11_v2",
      "Standard_D12_v2",
      "Standard_D13_v2",
      "Standard_D14_v2",
      "Standard_E2_v3",
      "Standard_E4_v3",
      "Standard_E8_v3",
      "Standard_E16_v3",
      "Standard_E32_v3",
      "Standard_E64_v3",
      "Standard_E2s_v3",
      "Standard_E4s_v3",
      "Standard_E8s_v3",
      "Standard_E16s_v3",
      "Standard_E32s_v3",
      "Standard_E64s_v3",
      "Standard_G1",
      "Standard_G2",
      "Standard_G3",
      "Standard_G4",
      "Standard_G5",
      "Standard_DS2",
      "Standard_DS3",
      "Standard_DS4",
      "Standard_DS11",
      "Standard_DS12",
      "Standard_DS13",
      "Standard_DS14",
      "Standard_DS2_v2",
      "Standard_DS3_v2",
      "Standard_DS4_v2",
      "Standard_DS5_v2",
      "Standard_DS11_v2",
```

```
        "Standard_DS12_v2",
        "Standard_DS13_v2",
        "Standard_DS14_v2",
        "Standard_GS1",
        "Standard_GS2",
        "Standard_GS3",
        "Standard_GS4",
        "Standard_GS5",
        "Standard_D2s_v3",
        "Standard_D4s_v3",
        "Standard_D8s_v3"
      ],
      "metadata" : {
        "description" : "The size of the each Node Virtual Machine"
      }
    }
  },
  "variables" : {
    "location" : "[resourceGroup().location]",
    "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
    "virtualNetworkID" : "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]",
    "nodeSubnetName" : "[concat(parameters('baseName'), '-worker-subnet')]",
    "nodeSubnetRef" : "[concat(variables('virtualNetworkID'), '/subnets/',
variables('nodeSubnetName'))]",
    "infraLoadBalancerName" : "[parameters('baseName')]",
    "sshKeyPath" : "/home/capi/.ssh/authorized_keys",
    "identityName" : "[concat(parameters('baseName'), '-identity')]",
    "imageName" : "[concat(parameters('baseName'), '-image')]",
    "copy" : [
      {
        "name" : "vmNames",
        "count" :  "[parameters('numberOfNodes')]",
        "input" : "[concat(parameters('baseName'), '-worker-', variables('location'), '-',
copyIndex('vmNames', 1))]"
      }
    ]
  },
  "resources" : [
    {
      "apiVersion" : "2019-05-01",
      "name" : "[concat('node', copyIndex())]",
      "type" : "Microsoft.Resources/deployments",
      "copy" : {
        "name" : "nodeCopy",
        "count" : "[length(variables('vmNames'))]"
      },
      "properties" : {
        "mode" : "Incremental",
        "template" : {
          "$schema" : "http://schema.management.azure.com/schemas/2015-01-
01/deploymentTemplate.json#",
          "contentVersion" : "1.0.0.0",
          "resources" : [
            {
              "apiVersion" : "2018-06-01",
```

```
            "type" : "Microsoft.Network/networkInterfaces",
            "name" : "[concat(variables('vmNames')[copyIndex()], '-nic')]",
            "location" : "[variables('location')]",
            "properties" : {
              "ipConfigurations" : [
                {
                  "name" : "pipConfig",
                  "properties" : {
                    "privateIPAllocationMethod" : "Dynamic",
                    "subnet" : {
                      "id" : "[variables('nodeSubnetRef')]"
                    },
                    "loadBalancerBackendAddressPools" : [
                      {
                        "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('infraLoadBalancerName'), '/backendAddressPools/', parameters('baseName'))]"
                      }
                    ]
                  }
                }
              ]
            }
          },
          {
            "apiVersion" : "2018-06-01",
            "type" : "Microsoft.Compute/virtualMachines",
            "name" : "[variables('vmNames')[copyIndex()]]",
            "location" : "[variables('location')]",
            "tags" : {
              "kubernetes.io-cluster-ffranzupi": "owned"
            },
            "identity" : {
              "type" : "userAssigned",
              "userAssignedIdentities" : {
                "[resourceID('Microsoft.ManagedIdentity/userAssignedIdentities/',
variables('identityName'))]" : {}
              }
            },
            "dependsOn" : [
              "[concat('Microsoft.Network/networkInterfaces/', concat(variables('vmNames')
[copyIndex()], '-nic'))]"
            ],
            "properties" : {
              "hardwareProfile" : {
                "vmSize" : "[parameters('nodeVMSize')]"
              },
              "osProfile" : {
                "computerName" : "[variables('vmNames')[copyIndex()]]",
                "adminUsername" : "capi",
                "customData" : "[parameters('workerIgnition')]",
                "linuxConfiguration" : {
                  "disablePasswordAuthentication" : true,
                  "ssh" : {
                    "publicKeys" : [
                      {
```

```
                    "path" : "[variables('sshKeyPath')]",
                    "keyData" : "[parameters('sshKeyData')]"
                  }
                ]
              }
            }
          },
          "storageProfile" : {
            "imageReference": {
              "id": "[resourceId('Microsoft.Compute/images', variables('imageName'))]"
            },
            "osDisk" : {
              "name": "[concat(variables('vmNames')[copyIndex()],'_OSDisk')]",
              "osType" : "Linux",
              "createOption" : "FromImage",
              "managedDisk": {
                "storageAccountType": "Premium_LRS"
              },
              "diskSizeGB": 128
            }
          },
          "networkProfile" : {
            "networkInterfaces" : [
              {
                "id" : "[resourceId('Microsoft.Network/networkInterfaces',
concat(variables('vmNames')[copyIndex()], '-nic'))]",
                "properties": {
                  "primary": true
                }
              }
            ]
          }
        }
      }
    }
  ]
}
```

1.7.17. 通过下载二进制文件安装 CLI

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

重要

如果安装了旧版本的 **oc**，则无法使用 OpenShift Container Platform 4.4 中的所有命令。下载并安装新版本的 **oc**。

1.7.17.1. 在 Linux 上安装 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. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Linux**，并点 **Download command-line tools**。
4. 解包存档：

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

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

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

1.7.17.2. 在 Windows 上安装 CLI

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

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Windows**，点 **Download command-line tools**。
4. 使用 ZIP 程序解压存档。
5. 把 **oc** 二进制代码放到 **PATH** 中的目录中。
   要查看您的 **PATH**，请打开命令提示窗口并执行以下命令：

```
C:\> path
```

安装 CLI 后，就可以使用**oc**命令：

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

1.7.17.3. 在 macOS 上安装 CLI

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

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。

2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **MacOS**，并点 **Download command-line tools**。
4. 解包和解压存档。
5. 将 **oc** 二进制文件移到 PATH 的目录中。
   要查看您的 **PATH**，打开一个终端窗口并执行以下命令：

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

1.7.18. 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** 文件包含关于集群的信息，供 CLI 用于将客户端连接到正确集群和 API 服务器。该文件特只适用于一个特定的集群，在 OpenShift Container Platform 安装过程中创建。

先决条件

- 部署一个 OpenShift Container Platform 集群。
- 安装 **oc** CLI。

流程

1. 导出 **kubeadmin** 凭证：

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

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

2. 使用导出的配置，验证能否成功运行 **oc** 命令：

```
$ oc whoami
system:admin
```

1.7.19. 批准机器的证书签名请求

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

先决条件

- 您已将机器添加到集群中。

流程

1. 确认集群可以识别这些机器：

```
# oc get nodes

NAME                 STATUS   ROLES    AGE   VERSION
master-01.example.com   Ready    master   40d   v1.17.1
master-02.example.com   Ready    master   40d   v1.17.1
master-03.example.com   Ready    master   40d   v1.17.1
worker-01.example.com   Ready    worker   40d   v1.17.1
worker-02.example.com   Ready    worker   40d   v1.17.1
```

输出将列出您创建的所有机器。

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

```
$ oc get csr

NAME        AGE     REQUESTOR                                                   CONDITION
csr-8b2br   15m     system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
csr-8vnps   15m     system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
...
```

在本例中，两台机器加入了集群。您可能会在列表中看到更多已批准的 CSR。

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

注意

由于 CSR 会自动轮转，因此请在将机器添加到集群后一小时内批准您的 CSR。如果没有在一小时内批准，证书将会轮转，每个节点将会存在多个证书。您必须批准所有这些证书。批准初始 CSR 后，集群的 **kube-controller-manager** 会自动批准后续的节点客户端 CSR。您必须实施一个方法来自动批准 kubelet 提供的证书请求。

- 若要单独批准，请对每个有效的 CSR 运行以下命令：

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

1 **<csr_name>** 是当前 CSR 列表中 CSR 的名称。

- 要批准所有待处理的 CSR，请运行以下命令：

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

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

```
$ oc get csr
```

输出示例

```
NAME        AGE     REQUESTOR                                                   CONDITION
csr-bfd72   5m26s   system:node:ip-10-0-50-126.us-east-2.compute.internal       Pending
csr-c57lv   5m26s   system:node:ip-10-0-95-157.us-east-2.compute.internal       Pending
...
```

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

- 若要单独批准，请对每个有效的 CSR 运行以下命令：

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

1 **<csr_name>** 是当前 CSR 列表中 CSR 的名称。

- 要批准所有待处理的 CSR，请运行以下命令：

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

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

```
$ oc get nodes
```

输出示例

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

注意

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

其他信息

- 如需有关 CSR 的更多信息，请参阅证书签名请求。

1.7.20. 添加 Ingress DNS 记录

如果您在创建 Kubernetes 清单并生成 Ignition 配置时删除了 DNS 区配置，您必须手动创建指向入口负载均衡器的 DNS 记录。您可以创建通配符 ***.apps.{baseDomain}.** 或具体的记录。您可以根据自己的要求使用 A、CNAME 和其他记录。

先决条件

- 已使用您置备的基础架构在 Microsoft Azure 上安装了 OpenShift Container Platform 集群。

- 安装 OpenShift CLI（**oc**）。
- 安装 **jq** 软件包。
- 安装或更新 Azure CLI。

流程

1. 确认 Ingress 路由器已创建了负载均衡器并填充 **EXTERNAL-IP** 字段：

```
$ oc -n openshift-ingress get service router-default
NAME             TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)                      AGE
router-default   LoadBalancer   172.30.20.10    35.130.120.110
80:32288/TCP,443:31215/TCP   20
```

2. 将 Ingress 路由器 IP 导出作为变量：

```
$ export PUBLIC_IP_ROUTER=`oc -n openshift-ingress get service router-default --no-
headers | awk '{print $4}'`
```

3. 在公共 DNS 区域中添加 ***.apps** 记录。

   a. 如果您要将此集群添加到新的公共区，请运行：

```
$ az network dns record-set a add-record -g ${BASE_DOMAIN_RESOURCE_GROUP} -
z ${CLUSTER_NAME}.${BASE_DOMAIN} -n *.apps -a ${PUBLIC_IP_ROUTER} --ttl 300
```

   b. 如果您要将此集群添加到已经存在的公共区中，请运行：

```
$ az network dns record-set a add-record -g ${BASE_DOMAIN_RESOURCE_GROUP} -
z ${BASE_DOMAIN} -n *.apps.${CLUSTER_NAME} -a ${PUBLIC_IP_ROUTER} --ttl 300
```

4. 在私有 DNS 区域中添加 ***.apps** 记录：

```
$ az network private-dns record-set a create -g ${RESOURCE_GROUP} -z
${CLUSTER_NAME}.${BASE_DOMAIN} -n *.apps --ttl 300
$ az network private-dns record-set a add-record -g ${RESOURCE_GROUP} -z
${CLUSTER_NAME}.${BASE_DOMAIN} -n *.apps -a ${PUBLIC_IP_ROUTER}
```

如果需要添加特定域而不使用通配符，可以为集群的每个当前路由创建条目：

```
$ oc get --all-namespaces -o jsonpath='{range .items[*]}{range .status.ingress[*]}{.host}{"\n"}{end}
{end}' routes
oauth-openshift.apps.cluster.basedomain.com
console-openshift-console.apps.cluster.basedomain.com
downloads-openshift-console.apps.cluster.basedomain.com
alertmanager-main-openshift-monitoring.apps.cluster.basedomain.com
grafana-openshift-monitoring.apps.cluster.basedomain.com
prometheus-k8s-openshift-monitoring.apps.cluster.basedomain.com
```

1.7.21. 在用户置备的基础架构上完成 Azure 安装

在 Microsoft Azure 用户置备的基础架构上启动 OpenShift Container Platform 安装后，您可以监控集群事件，直到集群就绪可用。

先决条件

- 在用户置备的 Azure 基础架构上为 OpenShift Container Platform 集群部署 bootstrap 机器。
- 安装 **oc** CLI 并登录。

流程

- 完成集群安装：

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

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

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

重要

安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrapper** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅*从过期的 control plane 证书中恢复*的文档。

1.8. 在 AZURE 上卸载集群

您可以删除部署到 Microsoft Azure 的集群。

1.8.1. 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。

先决条件

- 有部署集群时所用的安装程序副本。
- 有创建集群时安装程序所生成的文件。

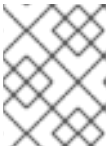
流程

1. 在用来安装集群的计算机中运行以下命令：

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

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

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



注意

您必须为集群指定包含集群定义文件的目录。安装程序需要此目录中的 **metadata.json** 文件来删除集群。

2. 可选：删除 **<installation_directory>** 目录和 OpenShift Container Platform 安装程序。