



# Red Hat Enterprise Linux 9

## 在 Google Cloud Platform 上部署 RHEL 9

获取 RHEL 系统镜像并在 GCP 上创建 RHEL 实例



# Red Hat Enterprise Linux 9 在 Google Cloud Platform 上部署 RHEL 9

---

获取 RHEL 系统镜像并在 GCP 上创建 RHEL 实例

## 法律通告

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 摘要

要在公有云环境中使用 Red Hat Enterprise Linux (RHEL)，您可以在各种云平台，包括 Google Cloud Platform (GCP) 上创建和部署 RHEL 系统镜像。您还可以在 GCP 上创建和配置红帽高可用性 (HA) 集群。以下章节提供了在 GCP 上创建云 RHEL 实例和 HA 集群的说明。这些进程包括安装所需的软件包和代理、配置隔离以及安装网络资源代理。

---

# 目录

对红帽文档提供反馈 .....	3
<b>第 1 章 公有云平台上的 RHEL 简介 .....</b>	<b>4</b>
1.1. 在公有云中使用 RHEL 的好处	4
1.2. RHEL 的公有云用例	4
1.3. 迁移到公有云时的常见关注	5
1.4. 为公有云部署获取 RHEL	6
1.5. 创建 RHEL 云实例的方法	6
<b>第 2 章 使用 RHEL 镜像构建器将镜像上传到 GCP .....</b>	<b>8</b>
2.1. 使用 CLI 将 GCE 镜像上传到 GCP	8
2.2. RHEL 镜像构建器如何对不同 GCP 凭证的身份验证顺序进行排序	9
<b>第 3 章 在 GOOGLE CLOUD PLATFORM 上将 RED HAT ENTERPRISE LINUX 镜像部署为 GOOGLE COMPUTE ENGINE 实例 .....</b>	<b>11</b>
3.1. GCP 上的 RED HAT ENTERPRISE LINUX 镜像选项	11
3.2. 理解基础镜像	12
3.3. 从 ISO 镜像创建基本虚拟机	13
3.4. 将 RHEL 镜像上传到 GCP	14
3.5. 其他资源	20
<b>第 4 章 在 GOOGLE CLOUD PLATFORM 上配置红帽高可用性集群 .....</b>	<b>21</b>
4.1. 在公有云平台上使用高可用性集群的好处	21
4.2. 所需的系统软件包	22
4.3. GCP 上的 RED HAT ENTERPRISE LINUX 镜像选项	22
4.4. 安装 GOOGLE CLOUD SDK	23
4.5. 创建 GCP 镜像存储桶	24
4.6. 创建自定义虚拟私有云网络和子网	24
4.7. 准备并导入基本 GCP 镜像	24
4.8. 创建并配置基本 GCP 实例	25
4.9. 创建快照镜像	27
4.10. 创建 HA 节点模板实例和 HA 节点	28
4.11. 安装 HA 软件包和代理	29
4.12. 配置 HA 服务	29
4.13. 创建集群	30
4.14. 创建隔离设备	31
4.15. 配置 GCP-VCP-MOVE-VIP 资源代理	32
4.16. 其他资源	34



---

## 对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

### 通过 Jira 提交反馈（需要帐户）

1. 登录到 [Jira](#) 网站。
2. 在顶部导航栏中点 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您对改进的建议。包括文档相关部分的链接。
5. 点对话框底部的 **Create**。

## 第 1 章 公有云平台上的 RHEL 简介

公有云平台提供计算资源即服务。除了使用内部硬件外，您可以运行您的 IT 工作负载，包括 Red Hat Enterprise Linux (RHEL) 系统，作为公共云实例。

### 1.1. 在公有云中 使用 RHEL 的好处

RHEL 作为公有云平台上的云实例与内部物理系统或虚拟机(VM)上的 RHEL 相比有以下优点：

- **灵活精细的资源分配**

RHEL 的云实例作为虚拟机在云平台上运行，这通常意味着由云服务提供商维护远程服务器集群。因此，给实例分配硬件资源，如特定类型的 CPU 或存储，发生在软件层面上，可轻松自定义。

与本地 RHEL 系统相比，您也不会受物理主机的功能的限制。相反，您可以根据云提供商提供的选择，从各种功能中进行选择。

- **空间及成本效率**

您不需要拥有任何内部服务器来托管您的云工作负载。这可避免与物理硬件关联的空间、电源和维护的要求。

相反，在公共云平台上，您直接向云提供商支付云实例的费用。成本通常基于分配给实例的硬件以及您使用的时间。因此，您可以根据要求优化成本。

- **软件控制的配置**

云实例的整个配置都作为数据保存在云平台上，并由软件控制。因此，您可以轻松地创建、删除、克隆或迁移实例。也可以在云提供商控制台对云实例远程操作，其默认连接到远程存储。

另外，您可以随时将云实例的当前状态备份为快照。之后，您可以加载快照，将实例恢复到保存的状态。

- **与主机分离和软件兼容性**

与本地虚拟机类似，云实例上的 RHEL 客户机操作系统运行在虚拟化内核上。这个内核与主机操作系统以及用于连接实例的客户端系统分开。

因此，任何操作系统都可以安装在云实例上。这意味着，在 RHEL 公有云实例中，您可以运行无法在本地操作系统上使用的特定于 RHEL 的应用程序。

另外，即使实例的操作系统不稳定或被破坏，您的客户端系统也不会受到任何影响。

#### 其他资源

- [什么是公有云？](#)
- [什么是 Hyperscaler？](#)
- [云计算的类型](#)
- [RHEL 的公有云用例](#)
- [为公有云部署获取 RHEL](#)

### 1.2. RHEL 的公有云用例

在公有云上部署会带来许多好处，但可能并非每种场景中最有效的解决方案。如果您要评估是否将 RHEL 部署迁移到公共云，请考虑您的用例是否将从公有云的好处中受益。

### 有益的用例

- 部署公有云实例对于灵活地增加和减少部署的活跃计算能力（也称为 *扩展* 和 *缩减*）非常有效。因此，在以下情况下，建议在公有云上使用 RHEL：
  - 具有峰值工作负载和一般性能要求的集群。在资源成本方面，根据您的需求扩展和缩减可能非常高效。
  - 快速设置或扩展集群。这可避免设置本地服务器的高前期成本。
- 云实例不受本地环境中发生的情况的影响。因此，您可以使用它们进行备份和灾难恢复。

### 有潜在问题的用例

- 您正在运行无法调整的现有环境。与您当前的主机平台相比，自定义云实例以适应现有部署的特定需求可能不是很划算。
- 您受预算方面的硬限制。与公有云相比，在本地数据中心的部署通常具有较少的灵活性，但可对最大资源成本进行更多的控制。

### 后续步骤

- [为公有云部署获取 RHEL](#)

### 其他资源

- [我是否应该将应用程序迁移到云？以下是如何决定。](#)

## 1.3. 迁移到公有云时的常见关注

将 RHEL 工作负载从本地环境移到公有云平台可能会引起对所涉及的变化担忧。以下是最常见的问题。

### 我的 RHEL 的工作方式是否与作为本地虚拟机的工作方式不同？

在大部分方面，公有云平台上的 RHEL 实例的工作方式与本地主机上 RHEL 虚拟机的工作方式一样，如内部服务器。主要例外包括：

- 公有云实例使用特定于提供商的控制台接口来管理云资源，而不是使用私有编排接口。
- 某些功能（如嵌套虚拟化）可能无法正常工作。如果特定功能对部署至关重要，请提前检查该功能与您选择的公共云提供商的兼容性。

### 与本地服务器相比，我的数据在公有云中是否安全？

RHEL 云实例中的数据归您所有，您的公共云提供商对其没有任何访问权限。此外，主要的云供应商支持传输中的数据加密，这提高了将虚拟机迁移到公有云时数据的安全性。

RHEL 公有云实例的一般安全性如下：

- 您的公有云提供商负责云 hypervisor 的安全性
- 红帽在您的实例中提供 RHEL 客户机操作系统的安全功能
- 您可以在云基础架构中管理特定的安全设置和实践

## 我的地理区域对 RHEL 公有云实例的功能有何影响？

无论您所在的地理位置如何，您都可以在公有云平台上使用 RHEL 实例。因此，您可以在与内部服务器相同的区域运行实例。

但是，在物理上较远的区域中托管您的实例可能会在操作它们时造成高延迟。此外，取决于公有云提供商，某些区域可能会提供额外的功能或更具成本效益。在创建 RHEL 实例前，请查看您选择的云提供商提供的托管区域的属性。

## 1.4. 为公有云部署获取 RHEL

要在公有云环境中部署 RHEL 系统，您需要：

1. 根据您的需求和当前市场提供的，为您的使用案例选择最佳云提供商。  
当前认证的运行 RHEL 实例云的提供商有：

- [Amazon Web Services \(AWS\)](#)
- [Google Cloud Platform \(GCP\)](#)
- [Microsoft Azure](#)



### 注意

本文档专门讨论有关在 GCP 上部署 RHEL。

2. 在您选择的云平台上创建 RHEL 云实例。如需更多信息，请参阅 [创建 RHEL 云实例的方法](#)。
3. 要让您的 RHEL 部署保持最新状态，请使用 [红帽更新基础设施 \(RHUI\)](#)。

### 其他资源

- [RHUI 文档](#)
- [Red Hat Open Hybrid Cloud](#)

## 1.5. 创建 RHEL 云实例的方法

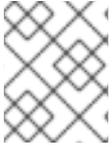
要在公有云上部署 RHEL 实例，您可以使用以下方法之一：

### 创建 RHEL 的系统镜像，并将其导入到云平台。

- 要创建系统镜像，您可以使用 [RHEL 镜像构建器](#)，也可以手动构建镜像。
- 此方法使用您现有的 RHEL 订阅，也称为 *自带订阅* (BYOS)。
- 您每年预付订阅费，并且您可以使用您的红帽客户折扣。
- 您的客户服务由红帽提供。
- 要有效地创建多个镜像，您可以使用 **cloud-init** 工具。

直接从云供应商市场购买 RHEL 实例。

- 使用这项服务，你需要按小时付费。因此，此方法也称为*随用随付*(PAYG)。
- 您的客户服务由云平台提供商提供。



### 注意

有关使用各种方法在 Google Cloud Platform 上部署 RHEL 实例的详细信息，请参阅本文档中的以下章节。

### 其他资源

- [什么是黄金镜像？](#)
- [为 RHEL 9 配置和管理 cloud-init](#)

## 第 2 章 使用 RHEL 镜像构建器将镜像上传到 GCP

使用 RHEL 镜像构建器，您可以构建 **gce** 镜像，为用户或 GCP 服务帐户提供凭证，然后将 **gce** 镜像直接上传到 GCP 环境。

### 2.1. 使用 CLI 将 GCE 镜像上传到 GCP

按照以下步骤使用凭证设置配置文件，将 **gce** 镜像上传到 GCP。



#### 警告

您无法手动将 **gce** 镜像导入到 GCP，因为镜像不能引导。您必须使用 **gcloud** 或 RHEL 镜像构建器上传它。

#### 先决条件

- 您有一个有效的 Google 帐户和凭证，来将镜像上传到 GCP。凭据可以来自用户帐户或服务帐户。与凭证关联的帐户必须至少分配以下 IAM 角色：
  - **roles/storage.admin** - 用于创建和删除存储对象
  - **roles/compute.storageAdmin** - 将虚拟机镜像导入到 Compute Engine。
- 您有一个现有的 GCP 存储桶。

#### 流程

1. 通过使用文本编辑器，创建一个具有以下内容的 **gcp-config.toml** 配置文件：

```
provider = "gcp"

[settings]
bucket = "GCP_BUCKET"
region = "GCP_STORAGE_REGION"
object = "OBJECT_KEY"
credentials = "GCP_CREDENTIALS"
```

- **GCP\_BUCKET** 指向现有的存储桶。它用于存储正在上传的镜像的中间存储对象。
- **GCP\_STORAGE\_REGION** 是常规的 Google 存储区域，是一个双区域或多区域。
- **OBJECT\_KEY** 是中间存储对象的名称。它在上传过程前不能存在，并在上传过程完成后被删除。如果对象名称不以 **.tar.gz** 结尾，则扩展会自动添加到对象名称中。
- **GCP\_CREDENTIALS** 是一个从 GCP 下载的凭证 JSON 文件的 Base64 编码方案。凭证决定了 GCP 将镜像上传到的项目。



### 注意

如果您使用不同的机制来使用 GCP 进行身份验证，在 `gcp-config.toml` 文件中指定 `GCP_CREDENTIALS` 是可选的。有关其他验证方法，请参阅 [使用 GCP 进行验证](#)。

2. 从 GCP 下载的 JSON 文件中检索 `GCP_CREDENTIALS`。

```
$ sudo base64 -w 0 cee-gcp-nasa-476a1fa485b7.json
```

3. 使用附加镜像名称和云供应商配置集创建 compose：

```
$ sudo composer-cli compose start BLUEPRINT-NAME gce IMAGE_KEY gcp-config.toml
```

镜像构建、上传和云注册过程最多可能需要十分钟才能完成。

### 验证

- 验证镜像状态为 FINISHED：

```
$ sudo composer-cli compose status
```

### 其它资源

- [身份和访问权限管理](#)
- [创建存储桶](#)

## 2.2. RHEL 镜像构建器如何对不同 GCP 凭证的身份验证顺序进行排序

您可以在 RHEL 镜像构建器中使用几种不同类型的凭证来使用 GCP 进行身份验证。如果 RHEL 镜像构建器配置被设置为使用多组凭证用 GCP 进行身份验证，它会按以下首选顺序使用凭证：

1. 在配置文件中，使用 `composer-cli` 命令指定的凭证。
2. 凭证在 `osbuild-composer` worker 配置中被配置。
3. **Google GCP SDK** 库中的 **应用程序默认凭证**，它尝试使用以下选项自动找到一个身份验证的方法：
  - a. 如果设置了 `GOOGLE_APPLICATION_CREDENTIALS` 环境变量，应用程序默认凭据会尝试加载并从文件中使用由变量指向的凭证。
  - b. 应用默认凭据尝试使用附加到运行代码的资源的服务帐户进行身份验证。例如，Google Compute Engine 虚拟机。



### 注意

您必须使用 GCP 凭证来决定将镜像上传到的 GCP 项目。因此，除非要将所有镜像上传到同一 GCP 项目，您必须使用 `composer-cli` 命令指定 `gcp-config.toml` 配置文件中的凭证。

### 2.2.1. 使用 `composer-cli` 命令指定 GCP 凭证

您可以在上传目标配置 **gcp-config.toml** 文件中指定 GCP 验证凭证。使用 Google 帐户凭证 JSON 文件的 **Base64** 编码方案来节省时间。

## 流程

1. 运行以下命令，使用存储在 **GOOGLE\_APPLICATION\_CREDENTIALS** 环境变量中的路径获取 Google 帐户凭证文件的编码内容：

```
$ base64 -w 0 "${GOOGLE_APPLICATION_CREDENTIALS}"
```

2. 在上传目标配置 **gcp-config.toml** 文件中，设置凭证：

```
provider = "gcp"

[settings]
provider = "gcp"

[settings]
...
credentials = "GCP_CREDENTIALS"
```

### 2.2.2. 在 **osbuild-composer worker** 配置中指定凭证

您可以将 GCP 身份验证凭据配置为全局用于 GCP 用于所有镜像构建。这样，如果您想要将镜像导入到同一 GCP 项目，则您可以对上传到 GCP 的所有镜像使用相同的凭据。

## 流程

- 在 **/etc/osbuild-worker/osbuild-worker.toml** worker 配置中，设置以下凭证值：

```
[gcp]
credentials = "PATH_TO_GCP_ACCOUNT_CREDENTIALS"
```

## 第 3 章 在 GOOGLE CLOUD PLATFORM 上将 RED HAT ENTERPRISE LINUX 镜像部署为 GOOGLE COMPUTE ENGINE 实例

要将 Red Hat Enterprise Linux 9 (RHEL 9) 作为 Google Compute Engine(GCE)实例部署到 Google Cloud Platform(GCP)上，请按照以下信息操作。本章：

- 讨论您选择镜像的选项
- 列出或引用主机系统和虚拟机(VM)的系统要求。
- 提供从 ISO 镜像创建自定义虚拟机的流程，将其上传到 GCE，并启动实例



### 注意

有关 GCP 红帽产品认证列表，请参阅 [Google Cloud Platform 上的红帽](#)。



### 重要

您可以从 ISO 镜像创建自定义的虚拟机，但红帽建议您使用 *Red Hat Image Builder* 产品来创建自定义的镜像以用于特定的云供应商。如需更多信息，请参阅 [制作自定义 RHEL 系统镜像](#)。

### 先决条件

- 您需要一个[红帽客户门户网站](#)帐户才能完成本章中的步骤。
- 使用 GCP 创建帐户来访问 Google Cloud Platform 控制台。如需更多信息，请参阅 [Google Cloud](#)。

## 3.1. GCP 上的 RED HAT ENTERPRISE LINUX 镜像选项

下表列出了 Google Cloud Platform 上 RHEL 9 的镜像选择以及镜像选项的区别。

表 3.1. 镜像选项

镜像选项	订阅	示例情境	注意事项
部署红帽黄金镜像。	使用您现有的红帽订阅。	在 Google Cloud Platform 上选择红帽黄金镜像。有关黄金镜像以及如何如何在 Google Cloud Platform 上访问它们的详情，请查看 <a href="#">红帽云访问参考指南</a> 。	订阅包括红帽产品成本；您需要支付 Google 其他实例的费用。红帽直接为自定义 RHEL 镜像提供支持。
部署移至 GCP 的自定义镜像。	使用您现有的红帽订阅。	上传自定义镜像并附加您的订阅。	订阅只包括红帽产品的成本；您还需要支付其他成本。红帽直接为自定义 RHEL 镜像提供支持。

镜像选项	订阅	示例情境	注意事项
部署一个包含 RHEL 的现有 GCP 镜像。	GCP 镜像包括一个红帽产品。	在 <a href="#">GCP Compute Engine</a> 上启动实例时选择 RHEL 镜像，或者从 <a href="#">Google Cloud Platform Marketplace</a> 中选择镜像。	根据 pay-as-you-go 模式每小时向 GCP 支付。这样的镜像称为 "on-demand" 镜像。GCP 通过支持协议支持 on-demand 镜像。



### 注意

您可以使用红帽镜像构建器为 GCP 创建一个自定义镜像。如需更多信息，请参阅 [制作自定义 RHEL 系统镜像](#)。



### 重要

您不能将按需实例转换为自定义 RHEL 实例。从按需镜像改为自定义 RHEL *自带订阅* (BYOS) 镜像：

1. 创建新的自定义 RHEL 实例，并从您的按需实例迁移数据。
2. 在迁移数据后取消您的 on-demand 实例以避免出现重复账单。

### 其他资源

- [公有云中的红帽](#)
- [compute Engine 镜像](#)
- [从自定义镜像创建实例](#)

## 3.2. 理解基础镜像

要从 ISO 镜像创建基础虚拟机，您可以使用预配置的基础镜像及其配置设置。

### 3.2.1. 使用自定义基础镜像

要手动配置虚拟机 (VM)，首先创建一个基础（起步）虚拟机镜像。然后，您可以修改配置设置，并添加 VM 在云上操作所需的软件包。您可在上传镜像后为特定应用程序进行额外的配置更改。

### 其它资源

- [Red Hat Enterprise Linux](#)

### 3.2.2. 虚拟机配置设置

云虚拟机必须具有以下配置设置。

表 3.2. 虚拟机配置设置

设置	建议
ssh	必须启用 SSH 来提供虚拟机的远程访问。
dhcp	应该为 dhcp 配置主虚拟适配器。

### 3.3. 从 ISO 镜像创建基本虚拟机

要从 ISO 镜像创建 RHEL 9 基础镜像，请为虚拟化启用您的主机并创建 RHEL 虚拟机(VM)。

#### 先决条件

- [虚拟化已在您的主机上启用](#)。
- 您已从[红帽客户门户网站](#)下载了最新的 Red Hat Enterprise Linux ISO 镜像，并将该镜像移到 `/var/lib/libvirt/images` 中。

#### 3.3.1. 从 RHEL ISO 镜像创建虚拟机

##### 步骤

1. 确保已为虚拟化启用主机机器。有关信息和流程，请参阅[在 RHEL 9 中启用虚拟化](#)。
2. 创建并启动基本 Red Hat Enterprise Linux 虚拟机。具体步骤请参阅[创建虚拟机](#)。
  - a. 如果使用命令行创建虚拟机，请确保将默认内存和 CPU 设置为您所需的容量。将您的虚拟网络接口设置为 `virtio`。  
例如，以下命令使用 `/home/username/Downloads/rhel9.iso` 镜像创建一个 `kvmtest` 虚拟机：

```
# virt-install \
  --name kvmtest --memory 2048 --vcpus 2 \
  --cdrom /home/username/Downloads/rhel9.iso,bus=virtio \
  --os-variant=rhel9.0
```

- b. 如果使用 web 控制台创建虚拟机，请按照[使用 web 控制台创建虚拟机](#)中的流程操作，请注意以下几个方面：
  - 不要选择 **Immediately Start VM**。
  - 将 **Memory** 大小更改为您希望的设置。
  - 在开始安装前，请确保将 **Virtual Network Interface Settings** 中的 **Model** 更改为 `virtio`，并将您的 **vCPU** 更改为您想要的虚拟机容量设置。

#### 3.3.2. 完成 RHEL 安装

要完成要在 Amazon Web Services (AWS) 上部署的 RHEL 系统 **安装**，[自定义安装概述](#) 视图，开始安装，并在虚拟机启动后启用 root 访问。

##### 流程

1. 选择您要在安装过程中使用的语言。
2. 在 **Installation Summary** 视图中：
  - a. 点 **Software Selection**, 选择 **Minimal Install**。
  - b. 点 **Done**。
  - c. 点击 **Installation Destination** 并检查 **Storage Configuration** 中的 **Custom**。
    - 验证 **/boot** 至少 500 MB。将剩余空间用于根 **/**。
    - 建议使用标准分区, 但您也可以使用逻辑卷管理 (LVM) 。
    - 您可以将 **xfs**、**ext4** 或者 **ext3** 用于文件系统。
    - 完成更改后点 **Done**。
3. 点 **Begin Installation**。
4. 设置 **Root 密码**。根据情况创建其他用户。
5. 重新启动虚拟机, 并在安装完成后以 **root** 身份登录。
6. 配置镜像。
  - a. 注册虚拟机并启用 Red Hat Enterprise Linux 9 软件仓库。

```
# subscription-manager register --auto-attach
```
  - b. 确保已安装并启用了 **cloud-init** 软件包。

```
# dnf install cloud-init
# systemctl enable --now cloud-init.service
```
7. 关闭虚拟机。

#### 其他资源

- [订阅自动附加和更新](#)
- [cloud-init 简介](#)

## 3.4. 将 RHEL 镜像上传到 GCP

要将 RHEL 9 镜像上传到 Google Cloud Platform (GCP), 请遵循本节中的步骤。

### 3.4.1. 在 GCP 上创建新项目

完成以下步骤, 来在 Google Cloud Platform(GCP)上创建一个新项目。

#### 先决条件

- 您必须拥有 GCP 帐户。如果没有, 请参阅 [Google Cloud](#) 了解更多信息。

## 步骤

1. 启动 [GCP 控制台](#)。
2. 点击 **Google Cloud Platform** 右侧的下拉菜单。
3. 在弹出菜单中点击 **NEW PROJECT**。
4. 在 **New Project** 窗口中输入新项目的名称。
5. 检查 **Organization**。如果需要，点击下拉菜单更改机构。
6. 确认您的父机构或文件夹的 **位置**。如果需要，点 **Browse** 搜索并更改这个值。
7. 点击 **CREATE** 创建新 GCP 项目。



### 注意

安装 Google Cloud SDK 后，您可以使用 **gcloud projects create** CLI 命令来创建项目。例如：

```
# gcloud projects create my-gcp-project3 --name project3
```

该示例创建了一个项目 ID 为 **my-gcp-project3**，项目名称为 **project3** 的项目。如需更多信息，请参阅 [gcloud 项目创建](#)。

## 其他资源

- [在 Google Cloud 中创建和管理资源](#)

### 3.4.2. 安装 Google Cloud SDK

完成以下步骤以安装 Google Cloud SDK。

## 步骤

1. 按照下载和提取 Google Cloud SDK 归档的 GCP 说明。详情请查看 GCP 文档中的 [Linux Quickstart](#)。
2. 按照初始化 Google Cloud SDK 的说明。



### 注意

初始化 Google Cloud SDK 后，您可以使用 **gcloud** CLI 命令来执行任务，并获取有关项目和实例的信息。例如，您可以使用 **gcloud compute project-info describe --project <project-name>** 命令来显示项目信息。

## 其它资源

- [Linux 快速入门](#)
- [gcloud 命令参考](#)
- [gcloud 命令行工具概述](#)

### 3.4.3. 为 Google Compute Engine 创建 SSH 密钥

执行以下流程，来使用 GCE 生成和注册 SSH 密钥，以便您可以使用其公共 IP 地址直接 SSH 到实例。

#### 流程

1. 使用 **ssh-keygen** 命令来生成用于 GCE 的 SSH 密钥对。

```
# ssh-keygen -t rsa -f ~/.ssh/google_compute_engine
```

2. 在 [GCP Console Dashboard 页面](#) 中，点击 Google Cloud Console banner 左侧的 **Navigation** 菜单，并选择 **Compute Engine**，然后选择 **Metadata**。
3. 点 **SSH Keys**，然后点 **Edit**。
4. 输入 `~/.ssh/google_compute_engine.pub` 文件中生成的结果，然后单击 **Save**。  
现在，您可以使用标准的 SSH 连接到您的实例。

```
# ssh -i ~/.ssh/google_compute_engine <username>@<instance_external_ip>
```



#### 注意

您可以运行 **gcloud compute config-ssh** 命令，使用实例的别名来填充配置文件。别名允许按实例名称简单的 SSH 连接。有关 **gcloud compute config-ssh** 命令的详情，请参考 [gcloud compute config-ssh](#)。

#### 其它资源

- [gcloud 计算 config-ssh](#)
- [连接到实例](#)

### 3.4.4. 在 GCP Storage 中创建存储桶

导入到 GCP 需要 GCP Storage Bucket。完成以下步骤以创建存储桶。

#### 流程

1. 如果您还没有登录到 GCP，请使用以下命令登录。

```
# gcloud auth login
```

2. 创建存储桶。

```
# gsutil mb gs://bucket_name
```



#### 注意

另外，您可以使用 Google Cloud Console 创建存储桶。如需更多信息，请[参阅创建存储桶](#)。

#### 其它资源

- [创建存储桶](#)

### 3.4.5. 转换并上传您的镜像到您的 GCP 存储桶

完成以下步骤，将您的镜像转换并上传到您的 GCP 存储桶。示例具有代表性；它们将 **qcow2** 镜像转换为 **raw** 格式，然后对该映像进行 tar 操作以便上传。

#### 步骤

1. 运行 **qemu-img** 命令来转换您的镜像。转换的映像必须具有名称 **disk.raw**。

```
# qemu-img convert -f qcow2 -O raw rhel-9.0-sample.qcow2 disk.raw
```

2. 打包镜像。

```
# tar --format=oldgnu -Sczf disk.raw.tar.gz disk.raw
```

3. 将镜像上传到之前创建的存储桶。上传可能需要几分钟时间。

```
# gsutil cp disk.raw.tar.gz gs://bucket_name
```

4. 在 **Google Cloud Platform** 主屏幕中，单击折叠菜单图标，并选择 **Storage**，然后选择 **Browser**。

5. 点存储桶的名称。  
打包的镜像列在存储桶名称下。



#### 注意

您还可以使用 **GCP 控制台** 上传您的镜像。为此，可单击存储桶的名称，然后单击 **Upload files**。

#### 其它资源

- [手动导入虚拟磁盘](#)
- [选择导入方法](#)

### 3.4.6. 从 GCP 存储桶中创建镜像

执行以下步骤从 GCP 存储桶中的对象创建镜像。

#### 流程

1. 运行以下命令来为 GCE 创建镜像。指定您要创建的镜像的名称、存储桶名称和打包的镜像的名称。

```
# gcloud compute images create my-image-name --source-uri gs://my-bucket-name/disk.raw.tar.gz
```



### 注意

另外，您可以使用 Google Cloud Console 创建镜像。如需更多信息，请参阅[创建、删除和弃用自定义镜像](#)。

2. 另外，还可在 GCP Console 中找到该镜像。
  - a. 单击 **Google Cloud Console** 标语左侧的 **Navigation** 菜单。
  - b. 选择 **Compute Engine**，然后选择 **Images**。

### 其它资源

- [创建、删除和弃用自定义镜像](#)
- [gcloud 计算镜像创建](#)

### 3.4.7. 从镜像创建 Google Compute Engine 实例

完成以下步骤，来使用 GCP 控制台配置 GCE 虚拟机实例。



### 注意

以下流程提供了使用 GCP 控制台创建基本虚拟机实例的说明。有关 GCE 虚拟机实例及其配置选项的更多信息，请参阅[创建和启动虚拟机实例](#)。

### 流程

1. 在 [GCP Console Dashboard 页面](#) 中，单击 Google **Cloud Console banner** 左侧的 **Navigation** 菜单，选择 **Compute Engine**，然后选择 **Images**。
2. 选择您的镜像。
3. 点 **Create Instance**。
4. 在 **Create an instance** 页面中，为您的实例输入一个 **Name**。
5. 选择一个 **Region** 和 **Zone**。
6. 选择满足或超过工作负载要求的**机器配置**。
7. 确保引导**磁盘**指定了您的镜像名称。
8. （可选）在 **Firewall** 下，选择 **Allow HTTP traffic** 或 **Allow HTTPS traffic**。
9. 点 **Create**。



### 注意

这些是创建基本实例所需的最小配置选项。根据您的应用程序要求查看其他选项。

10. 在**虚拟机实例**中查找您的镜像。
11. 在 GCP Console Dashboard 中，单击 Google **Cloud Console banner** 左侧的 **Navigation** 菜单，选择 **Compute Engine**，然后选择 **VM instances**。



### 注意

或者，您可以使用 **gcloud compute instances create** CLI 命令来通过镜像创建 GCE 虚拟机实例。下面是一个简单的例子。

```
gcloud compute instances create myinstance3 --zone=us-central1-a --image
test-iso2-image
```

该示例根据现有的 **test-iso2-image** 映像，在区域 **us-central1-a** 中创建名为 **myinstance3** 的虚拟机实例。如需更多信息，请参阅 [gcloud 计算实例创建](#)。

### 3.4.8. 连接到您的实例

执行以下步骤使用其公共 IP 地址连接到 GCE 实例。

#### 流程

1. 运行以下命令以确保您的实例正在运行。命令列出有关 GCE 实例的信息，包括实例是否正在运行，如果正在运行，则列出正在运行的实例的公共 IP 地址。

```
# gcloud compute instances list
```

2. 使用标准的 SSH 连接到您的实例。该示例是使用之前创建的 **google\_compute\_engine** 密钥。

```
# ssh -i ~/.ssh/google_compute_engine <user_name>@<instance_external_ip>
```



### 注意

GCP 提供了多种 SSH 到您的实例的方法。如需更多信息，请参阅[连接到实例](#)。您还可以使用之前设置的 root 帐户和密码连接到您的实例。

#### 其它资源

- [gcloud 计算实例列表](#)
- [连接到实例](#)

### 3.4.9. 附加红帽订阅

使用 **subscription-manager** 命令，您可以注册并附加红帽订阅到 RHEL 实例。

#### 先决条件

- 您必须已启用您的订阅。

#### 流程

1. 注册您的系统。

```
# subscription-manager register --auto-attach
```

2. 附加您的订阅。

- 您可以使用激活码来附加订阅。如需更多信息，请参阅[创建红帽客户门户网站激活码](#)。
  - 或者，您可以使用订阅池（池 ID）的 ID 手动附加订阅。请参阅[通过命令行附加和删除订阅](#)。
3. 可选：要在 [Red Hat Hybrid Cloud Console](#) 中收集有关实例的各种系统指标，您可以使用 [Red Hat Insights](#) 注册实例。

```
# insights-client register --display-name <display-name-value>
```

有关 Red Hat Insights 进一步配置的详情，请参考 [Red Hat Insights 的客户端配置指南](#)。

#### 其它资源

- [创建红帽客户门户网站激活码](#)
- [通过命令行附加和删除订阅](#)
- [使用并配置 Red Hat Subscription Manager](#)
- [Red Hat Insights 的客户端配置指南](#)

### 3.5. 其他资源

- [公有云中的红帽](#)
- [Google Cloud](#)

## 第 4 章 在 GOOGLE CLOUD PLATFORM 上配置红帽高可用性集群

要创建集群，当节点出现故障时，RHEL 节点会自动重新分发其工作负载，请使用 Red Hat High Availability Add-On。此类高可用性(HA)集群也可以托管在公有云平台上，包括 Google Cloud Platform (GCP)。在 GCP 上创建 RHEL HA 集群与在非云环境中创建 HA 集群类似，某些特定信息。

要使用 Google Compute Engine (GCE)虚拟机(VM)实例作为集群节点在 Google Cloud Platform (GCP) 上配置 Red Hat HA 集群，请参阅以下部分。

它们提供以下信息：

- 为 GCP 设置环境的先决条件的流程。设置完环境后，您可以创建并配置虚拟机实例。
- 特定于创建 HA 集群的流程，其将单个节点转换为 GCP 上的 HA 节点的集群。这包括在每个集群节点上安装高可用性软件包和代理、配置隔离以及安装网络资源代理的步骤。

### 先决条件

- Red Hat Enterprise Linux 9 Server: rhel-9-server-rpms/8Server/x86\_64
- Red Hat Enterprise Linux 9 服务器（高可用性）：rhel-9-server-ha-rpms/8Server/x86\_64
  - 您必须属于活跃的 GCP 项目，并有足够的权限在项目中创建资源。
  - 您的项目应具有属于虚拟机实例而非单独的用户的服务帐户。有关使用默认服务帐户而不是创建单独服务帐户的信息，请参阅[使用 Compute Engine 默认服务帐户](#)。

如果您或项目管理员创建自定义服务帐户，则应该为以下角色配置服务帐户。

- Cloud Trace Agent
- Compute Admin
- Compute Network Admin
- Cloud Datastore User
- Logging Admin
- Monitoring Editor
- Monitoring Metric Writer
- Service Account Administrator
- Storage Admin

### 4.1. 在公有云平台上使用高可用性集群的好处

高可用性(HA)集群是一组链接在一起的计算机（称为节点），以运行特定的工作负载。HA 集群的目的是在出现硬件或软件故障时提供冗余。如果 HA 集群中的节点失败，Pacemaker 集群资源管理器会将工作负载分发到其他节点，且在集群中运行的服务中不会出现显著的停机时间。

您还可以在公有云平台上运行 HA 集群。在这种情况下，您要将云中的虚拟机(VM)实例用作单独的集群节点。在公有云平台上使用 HA 集群有以下优点：

- 改进了可用性：如果出现虚拟机故障，工作负载会快速重新分发到其他节点，因此运行的服务不会中断。
- 可扩展性：在需求高时启动其他节点，在需求低时停止其他节点。
- 节约成本：采用现收现付定价时，您只需为正在运行的节点支付费用。
- 简化管理：有些公共云平台提供管理界面，以便更轻松地配置 HA 集群。

要在 Red Hat Enterprise Linux (RHEL) 系统上启用 HA，红帽提供了一个高可用性附加组件。高可用性附加组件提供了在 RHEL 系统上创建 HA 集群的所有必要组件。这些组件包括高可用性服务管理和集群管理工具。

## 其他资源

- [高可用性附加组件概述](#)

## 4.2. 所需的系统软件包

要创建并配置 RHEL 的基础镜像，主机系统必须安装了以下软件包。

表 4.1. 系统软件包

软件包	软件仓库	描述
libvirt	rhel-9-for-x86_64-appstream-rpms	用于管理平台虚拟化的开源 API、守护进程和管理工具
virt-install	rhel-9-for-x86_64-appstream-rpms	用于构建虚拟机的命令行工具
libguestfs	rhel-9-for-x86_64-appstream-rpms	用于访问和修改虚拟机文件系统的库
guestfs-tools	rhel-9-for-x86_64-appstream-rpms	虚拟机的系统管理工具；包括 <b>virt-customize</b> 实用程序

## 4.3. GCP 上的 RED HAT ENTERPRISE LINUX 镜像选项

下表列出了 Google Cloud Platform 上 RHEL 9 的镜像选择以及镜像选项的区别。

表 4.2. 镜像选项

镜像选项	订阅	示例情境	注意事项
部署红帽黄金镜像。	使用您现有的红帽订阅。	在 Google Cloud Platform 上选择红帽黄金镜像。有关黄金镜像以及如何如何在 Google Cloud Platform 上访问它们的详情，请查看 <a href="#">红帽云访问参考指南</a> 。	订阅包括红帽产品成本；您需要支付 Google 其他实例的费用。红帽直接为自定义 RHEL 镜像提供支持。

镜像选项	订阅	示例情境	注意事项
部署移至 GCP 的自定义镜像。	使用您现有的红帽订阅。	上传自定义镜像并附加您的订阅。	订阅只包括红帽产品的成本；您还需要支付其他成本。红帽直接为自定义 RHEL 镜像提供支持。
部署一个包含 RHEL 的现有 GCP 镜像。	GCP 镜像包括一个红帽产品。	在 <a href="#">GCP Compute Engine</a> 上启动实例时选择 RHEL 镜像，或者从 <a href="#">Google Cloud Platform Marketplace</a> 中选择镜像。	根据 pay-as-you-go 模式每小时向 GCP 支付。这样的镜像称为 "on-demand" 镜像。GCP 通过支持协议支持 on-demand 镜像。



### 注意

您可以使用红帽镜像构建器为 GCP 创建一个自定义镜像。如需更多信息，请参阅 [制作自定义 RHEL 系统镜像](#)。



### 重要

您不能将按需实例转换为自定义 RHEL 实例。从按需镜像改为自定义 RHEL *自带订阅* (BYOS) 镜像：

1. 创建新的自定义 RHEL 实例，并从您的按需实例迁移数据。
2. 在迁移数据后取消您的 on-demand 实例以避免出现重复账单。

### 其他资源

- [公有云中的红帽](#)
- [compute Engine 镜像](#)
- [从自定义镜像创建实例](#)

## 4.4. 安装 GOOGLE CLOUD SDK

完成以下步骤以安装 Google Cloud SDK。

### 步骤

1. 按照下载和提取 Google Cloud SDK 归档的 GCP 说明。详情请查看 GCP 文档中的 [Linux Quickstart](#)。
2. 按照初始化 Google Cloud SDK 的说明。



### 注意

初始化 Google Cloud SDK 后，您可以使用 **gcloud** CLI 命令来执行任务，并获取有关项目和实例的信息。例如，您可以使用 **gcloud compute project-info describe --project <project-name>** 命令来显示项目信息。

## 其它资源

- [Linux 快速入门](#)
- [gcloud 命令参考](#)
- [gcloud 命令行工具概述](#)

## 4.5. 创建 GCP 镜像存储桶

以下文档包含在默认位置创建 [multi-regional](#) 存储桶的最低要求。

### 先决条件

- GCP 存储工具(gcloud)

### 流程

1. 如果您还没有登录到 Google Cloud Platform，请使用以下命令登录。

```
# gcloud auth login
```

2. 创建存储桶。

```
$ gcloud mb gs://BucketName
```

例如：

```
$ gcloud mb gs://rhel-ha-bucket
```

## 其它资源

- [生成存储桶](#)

## 4.6. 创建自定义虚拟私有云网络和子网

完成以下步骤来创建自定义虚拟私有云(VPC)网络和子网。

### 流程

1. 启动 GCP 控制台。
2. 在左侧导航窗格中，选择 **Networking** 下的 **VPC networks**。
3. 点 **Create VPC Network**。
4. 输入 VPC 网络的名称。
5. 在 **New subnet** 下，在您要创建集群的区域中创建 **Custom subnet**。
6. 点 **Create**。

## 4.7. 准备并导入基本 GCP 镜像

完成以下步骤，为 GCP 准备 Red Hat Enterprise Linux 9 镜像。

### 步骤

1. 转换文件。上传到 GCP 的镜像必须是 **raw** 格式，并命名为 **disk.raw**。

```
$ qemu-img convert -f qcow2 ImageName.qcow2 -O raw disk.raw
```

2. 压缩 **raw** 文件。上传到 GCP 的镜像必须被压缩。

```
$ tar -Sczf ImageName.tar.gz disk.raw
```

3. 将压缩镜像导入到之前创建的存储桶。

```
$ gsutil cp ImageName.tar.gz gs://BucketName
```

## 4.8. 创建并配置基本 GCP 实例

完成以下步骤，创建并配置符合 GCP 操作和安全要求的 GCP 实例。

### 步骤

1. 从存储桶中压缩的文件创建镜像。

```
$ gcloud compute images create BaseImageName --source-uri
gs://BucketName/BaseImageName.tar.gz
```

例如：

```
[admin@localhost ~] $ gcloud compute images create rhel-76-server --source-uri gs://user-
rhelha/rhel-server-76.tar.gz
Created [https://www.googleapis.com/compute/v1/projects/MyProject/global/images/rhel-
server-76].
NAME          PROJECT          FAMILY DEPRECATED STATUS
rhel-76-server rhel-ha-testing-on-gcp          READY
```

2. 从镜像创建模板实例。基本 RHEL 实例所需的最小值为 n1-standard-2。如需了解更多配置选项，请参阅 [gcloud 计算实例创建](#)。

```
$ gcloud compute instances create BaseInstanceName --can-ip-forward --machine-type n1-
standard-2 --image BaseImageName --service-account ServiceAccountEmail
```

例如：

```
[admin@localhost ~] $ gcloud compute instances create rhel-76-server-base-instance --can-
ip-forward --machine-type n1-standard-2 --image rhel-76-server --service-account
account@project-name-on-gcp.iam.gserviceaccount.com
Created [https://www.googleapis.com/compute/v1/projects/rhel-ha-testing-on-gcp/zones/us-
east1-b/instances/rhel-76-server-base-instance].
NAME ZONE MACHINE_TYPE PREEMPTIBLE INTERNAL_IP EXTERNAL_IP
```

## STATUS

```
rhel-76-server-base-instance us-east1-bn1-standard-2 10.10.10.3 192.227.54.211  
RUNNING
```

3. 通过 SSH 终端会话连接到实例。

```
$ ssh root@PublicIPaddress
```

4. 更新 RHEL 软件。

- a. 使用红帽订阅管理器(RHSM)注册。
- b. 启用订阅池 ID（或使用 **--auto-attach** 命令）。
- c. 禁用所有软件仓库。

```
# subscription-manager repos --disable=*
```

- d. 启用以下软件仓库。

```
# subscription-manager repos --enable=rhel-9-server-rpms
```

- e. 运行 **dnf update** 命令。

```
# dnf update -y
```

5. 在运行的实例（原位安装）中安装 GCP Linux 客户机环境。  
具体步骤请参阅[原位安装客户机环境](#)。
6. 选择 **CentOS/RHEL** 选项。
7. 复制命令脚本，并将它粘贴到命令提示符处，来立即运行脚本。
8. 对实例进行以下配置更改。这些更改基于自定义镜像的 GCP 建议。如需更多信息，请参阅 [gcloudcompute 镜像列表](#)。
  - a. 编辑 **/etc/chrony.conf** 文件，并删除所有 NTP 服务器。
  - b. 添加以下 NTP 服务器。

```
metadata.google.internal iburst Google NTP server
```

- c. 删除任何持久的网络设备规则。

```
# rm -f /etc/udev/rules.d/70-persistent-net.rules
```

```
# rm -f /etc/udev/rules.d/75-persistent-net-generator.rules
```

- d. 将网络服务设置为自动启动。

```
# chkconfig network on
```

- e. 将 **sshd** 服务 设置为自动启动。

```
# systemctl enable sshd
# systemctl is-enabled sshd
```

- f. 将时区设为 UTC。

```
# ln -sf /usr/share/zoneinfo/UTC /etc/localtime
```

- g. (可选) 编辑 `/etc/ssh/ssh_config` 文件, 并将以下行添加到文件的末尾: 这将在较长的不活跃时间段内使您的 SSH 会话保持活跃。

```
# Server times out connections after several minutes of inactivity.
# Keep alive ssh connections by sending a packet every 7 minutes.
ServerAliveInterval 420
```

- h. 编辑 `/etc/ssh/sshd_config` 文件, 并根据需要进行以下更改: `ClientAliveInterval 420` 设置是可选的; 这会使 SSH 会话在长时间处于非活动状态期间保持活跃状态。

```
PermitRootLogin no
PasswordAuthentication no
AllowTcpForwarding yes
X11Forwarding no
PermitTunnel no
# Compute times out connections after 10 minutes of inactivity.
# Keep ssh connections alive by sending a packet every 7 minutes.
ClientAliveInterval 420
```

9. 禁用密码访问。

```
ssh_pwauth from 1 to 0.
ssh_pwauth: 0
```



### 重要

在以前的版本中, 您可以启用密码访问来允许 SSH 会话访问来配置实例。您必须禁用密码访问。所有 SSH 会话访问都必须是无密码的。

10. 从订阅管理器取消实例注册。

```
# subscription-manager unregister
```

11. 清除 shell 历史记录。为下一个流程保留实例运行。

```
# export HISTSIZE=0
```

## 4.9. 创建快照镜像

完成以下步骤以保留实例配置设置并创建快照。

### 步骤

1. 在正在运行的实例上, 将数据同步到磁盘。

```
# sync
```

2. 在您的主机系统上创建快照。

```
$ gcloud compute disks snapshot InstanceName --snapshot-names SnapshotName
```

3. 在您的主机系统上，从快照创建配置的镜像。

```
$ gcloud compute images create ConfiguredImageFromSnapshot --source-snapshot SnapshotName
```

## 其他资源

- [创建持久性磁盘快照](#)

## 4.10. 创建 HA 节点模板实例和 HA 节点

从快照配置了镜像后，您可以创建节点模板。使用此模板来创建所有 HA 节点。完成以下步骤以创建模板和 HA 节点。

### 步骤

1. 创建实例模板。

```
$ gcloud compute instance-templates create InstanceTemplateName --can-ip-forward --
machine-type n1-standard-2 --image ConfiguredImageFromSnapshot --service-account
ServiceAccountEmailAddress
```

Example:

```
[admin@localhost ~] $ gcloud compute instance-templates create rhel-91-instance-template
--can-ip-forward --machine-type n1-standard-2 --image rhel-91-gcp-image --service-account
account@project-name-on-gcp.iam.gserviceaccount.com
Created [https://www.googleapis.com/compute/v1/projects/project-name-on-
gcp/global/instanceTemplates/rhel-91-instance-template].
NAME MACHINE_TYPE PREEMPTIBLE CREATION_TIMESTAMP
rhel-91-instance-template n1-standard-2 2018-07-25T11:09:30.506-07:00
```

2. 在一个区域中创建多个节点。

```
# gcloud compute instances create NodeName01 NodeName02 --source-instance-template
InstanceTemplateName --zone RegionZone --network=NetworkName --
subnet=SubnetName
```

Example:

```
[admin@localhost ~] $ gcloud compute instances create rhel81-node-01 rhel81-node-02
rhel81-node-03 --source-instance-template rhel-91-instance-template --zone us-west1-b --
network=projectVPC --subnet=range0
Created [https://www.googleapis.com/compute/v1/projects/project-name-on-gcp/zones/us-
west1-b/instances/rhel81-node-01].
Created [https://www.googleapis.com/compute/v1/projects/project-name-on-gcp/zones/us-
```

```
west1-b/instances/rhel81-node-02].
```

```
Created [https://www.googleapis.com/compute/v1/projects/project-name-on-gcp/zones/us-west1-b/instances/rhel81-node-03].
```

NAME	ZONE	MACHINE_TYPE	PREEMPTIBLE	INTERNAL_IP	EXTERNAL_IP	STATUS
rhel81-node-01	us-west1-b	n1-standard-2		10.10.10.4	192.230.25.81	RUNNING
rhel81-node-02	us-west1-b	n1-standard-2		10.10.10.5	192.230.81.253	RUNNING
rhel81-node-03	us-east1-b	n1-standard-2		10.10.10.6	192.230.102.15	RUNNING

## 4.11. 安装 HA 软件包和代理

在所有节点上完成以下步骤。

### 流程

1. 在 Google Cloud Console 中，选择 **Compute Engine**，然后选择 **VM instances**。
2. 选择实例，单击 **SSH** 旁边的箭头，然后选择 **View gcloud** 命令选项。
3. 在命令提示符下粘贴此命令，以进行免密码访问实例。
4. 启用 sudo 帐户访问，并通过 Red Hat Subscription Manager 注册。
5. 启用订阅池 ID（或使用 **--auto-attach** 命令）。
6. 禁用所有软件仓库。

```
# subscription-manager repos --disable=*
```

7. 启用以下软件仓库。

```
# subscription-manager repos --enable=rhel-9-server-rpms
# subscription-manager repos --enable=rhel-9-for-x86_64-highavailability-rpms
```

8. 安装 **pcs pacemaker**、隔离代理和资源代理。

```
# dnf install -y pcs pacemaker fence-agents-gce resource-agents-gcp
```

9. 更新所有软件包。

```
# dnf update -y
```

## 4.12. 配置 HA 服务

在所有节点上完成以下步骤以配置 HA 服务。

### 步骤

1. 在上一步中的 **pcs** 和 **pacemaker** 安装过程中，创建了用户 **hacluster**。在所有群集节点上为用户 **hacluster** 创建密码。所有节点都使用相同的密码。

```
# passwd hacluster
```

2. 如果安装了 **firewalld** 服务，请添加 HA 服务。

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --reload
```

3. 启动 **pcs** 服务，并使其在引导时启动。

```
# systemctl start pcsd.service
# systemctl enable pcsd.service

Created symlink from /etc/systemd/system/multi-user.target.wants/pcsd.service to
/usr/lib/systemd/system/pcsd.service.
```

## 验证

1. 确保 **pcsd** 服务正在运行。

```
# systemctl status pcsd.service

pcsd.service - PCS GUI and remote configuration interface
Loaded: loaded (/usr/lib/systemd/system/pcsd.service; enabled; vendor preset: disabled)
Active: active (running) since Mon 2018-06-25 19:21:42 UTC; 15s ago
Docs: man:pcsd(8)
      man:pcs(8)
Main PID: 5901 (pcsd)
CGroup: /system.slice/pcsd.service
        └─5901 /usr/bin/ruby /usr/lib/pcsd/pcsd > /dev/null &
```

2. 编辑 **/etc/hosts** 文件。为所有节点添加 RHEL 主机名和内部 IP 地址。

## 其它资源

- [如何在 RHEL 集群节点上设置 /etc/hosts 文件？](#)

## 4.13. 创建集群

完成以下步骤以创建节点集群。

### 步骤

1. 在其中一个节点上，验证 **pcs** 用户。在该命令中指定集群中每个节点的名称。

```
# pcs host auth hostname1 hostname2 hostname3
Username: hacluster
Password:
hostname1: Authorized
hostname2: Authorized
hostname3: Authorized
```

2. 创建集群。

```
# pcs cluster setup cluster-name hostname1 hostname2 hostname3
```

## 验证

1. 运行以下命令，以便在启动时自动加入集群。

```
# pcs cluster enable --all
```

2. 启动集群。

```
# pcs cluster start --all
```

## 4.14. 创建隔离设备

完成以下步骤以创建隔离设备。

请注意，对于大多数默认配置，GCP 实例名称和 RHEL 主机名是一样的。

### 步骤

1. 获取 GCP 实例名称。请注意，以下命令的输出还显示实例的内部 ID。

```
# fence_gce --zone us-west1-b --project=rhel-ha-on-gcp -o list
```

Example:

```
[root@rhel81-node-01 ~]# fence_gce --zone us-west1-b --project=rhel-ha-testing-on-gcp
-o list
4435801234567893181,InstanceName-3
4081901234567896811,InstanceName-1
7173601234567893341,InstanceName-2
```

2. 创建隔离设备。

```
# pcs stonith create FenceDeviceName fence_gce zone=Region-Zone
project=MyProject
```

## 验证

- 验证隔离设备是否已启动。

```
# pcs status
```

例如：

```
[root@rhel81-node-01 ~]# pcs status
Cluster name: gcp-cluster
Stack: corosync
Current DC: rhel81-node-02 (version 1.1.18-11.el7_5.3-2b07d5c5a9) - partition with quorum
```

```
Last updated: Fri Jul 27 12:53:25 2018
Last change: Fri Jul 27 12:51:43 2018 by root via cibadmin on rhel81-node-01
```

```
3 nodes configured
3 resources configured
```

```
Online: [ rhel81-node-01 rhel81-node-02 rhel81-node-03 ]
```

```
Full list of resources:
```

```
us-west1-b-fence (stonith:fence_gce): Started rhel81-node-01
```

```
Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

## 4.15. 配置 GCP-VCP-MOVE-VIP 资源代理

**gcp-vpc-move-vip** 资源代理将辅助 IP 地址（别名 IP）附加到正在运行的实例。这是一个浮动 IP 地址，可在集群中的不同节点间传递。

要显示有关此资源的更多信息：

```
# pcs resource describe gcp-vpc-move-vip
```

您可以将资源代理配置为使用主子网地址范围或二级子网地址范围：

### 主子网地址范围

完成以下步骤，为主 VPC 子网配置资源。

### 流程

1. 创建 **aliasip** 资源。包括一个未使用的内部 IP 地址。在命令中包含 CIDR 块。

```
# pcs resource create aliasip gcp-vpc-move-vip alias_ip=UnusedIPAddress/CIDRblock
```

例如：

```
[root@rhel81-node-01 ~]# pcs resource create aliasip gcp-vpc-move-vip
alias_ip=10.10.10.200/32
```

2. 创建用于管理节点上 IP 的 **IPAddr2** 资源。

```
# pcs resource create vip IPAddr2 nic=interface ip=AliasIPAddress cidr_netmask=32
```

例如：

```
[root@rhel81-node-01 ~]# pcs resource create vip IPAddr2 nic=eth0 ip=10.10.10.200
cidr_netmask=32
```

3. 将网络资源分组到 **vipgrp** 下。

```
# pcs resource group add vipgrp aliasip vip
```

## 验证

1. 验证资源是否已启动，是否分组在 **vipgrp** 下。

```
# pcs status
```

2. 验证资源是否可以移到另一个节点。

```
# pcs resource move vip Node
```

例如：

```
[root@rhel81-node-01 ~]# pcs resource move vip rhel81-node-03
```

3. 验证 **vip** 是否在不同的节点上成功启动。

```
# pcs status
```

## 二级子网地址范围

完成以下步骤，为二级子网地址范围配置资源。

### 先决条件

- [您已创建了自定义网络和子网](#)
- **可选：**已安装 Google Cloud SDK。具体步骤请参阅 [安装 Google Cloud SDK](#)。但请注意，您可以在您可以在 Google Cloud web 控制台中激活的终端中使用 **gcloud** 命令。

### 流程

1. 创建二级子网地址范围。

```
# gcloud compute networks subnets update SubnetName --region RegionName --add-secondary-ranges SecondarySubnetName=SecondarySubnetRange
```

例如：

```
# gcloud compute networks subnets update range0 --region us-west1 --add-secondary-ranges range1=10.10.20.0/24
```

2. 创建 **aliasip** 资源。在二级子网地址范围内创建一个未使用的内部 IP 地址。在命令中包含 CIDR 块。

```
# pcs resource create aliasip gcp-vpc-move-vip alias_ip=UnusedIPAddress/CIDRblock
```

例如：

```
[root@rhel81-node-01 ~]# pcs resource create aliasip gcp-vpc-move-vip alias_ip=10.10.20.200/32
```

- 
- 
3. 创建用于管理节点上 IP 的 **IPAddr2** 资源。

```
# pcs resource create vip IPAddr2 nic=interface ip=AliasIPAddress cidr_netmask=32
```

例如：

```
[root@rhel81-node-01 ~]# pcs resource create vip IPAddr2 nic=eth0 ip=10.10.20.200  
cidr_netmask=32
```

- 
- 
- 
4. 将网络资源分组到 **vipgrp** 下。

```
# pcs resource group add vipgrp aliasip vip
```

## 验证

1. 验证资源是否已启动，是否分组在 **vipgrp** 下。

```
# pcs status
```

2. 验证资源是否可以移到另一个节点。

```
# pcs resource move vip Node
```

例如：

```
[root@rhel81-node-01 ~]# pcs resource move vip rhel81-node-03
```

3. 验证 **vip** 是否在不同的节点上成功启动。

```
# pcs status
```

## 4.16. 其他资源

- [RHEL 高可用性集群的支持策略 - 传输协议](#)
- [VPC 网络概述](#)
- [查看 RHEL 高可用性的组件、概念和功能 - 传输概述](#)
- [RHEL 高可用性集群设计指南 - 选择传输协议](#)