Red Hat Enterprise Linux 8

为 RHEL 8 配置和管理 cloud-init

自动 Red Hat Enterprise Linux 云实例的初始化

Last Updated: 2024-06-27

Red Hat Enterprise Linux 8 为 RHEL 8 配置和管理 cloud-init

自动 Red Hat Enterprise Linux 云实例的初始化

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

摘要

您可以使用 cloud-init 软件包高效地创建 RHEL 的多个云实例。这允许在各种云平台上一致且可重复部署 RHEL。在以下章节中，您可以了解更多有关以下信息： cloud-init 的工作原理 如何使用 cloud-init 启动云实例 红帽支持 cloud-init 的哪些用途

目录

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. 单击顶部导航栏中的 **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。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，而不是使用内部硬件。

1.1. 在公有云中使用 RHEL 的好处

RHEL 作为公共云平台上的云实例与内部物理系统或虚拟机(VM)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、电源和维护要求。

  相反，在公有云平台上，您可以直接向云提供商支付使用云实例的费用。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。云实例也在云提供商控制台中远程操作，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来连接实例的 *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变得不稳定或被破坏，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，但可能并非是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共云的好处中受益。

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：
  - 具有高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恢复。

有潜在问题的用例

- 您正在运行一个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并不划算。
- 您有预算方面的硬限制。在本地数据中心中维护您的部署通常具有更大的灵活性，但与公有云相比，您对最大资源成本有更多的控制。

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带来有关涉及的变化的担忧。以下是最常见的问题。

作为云实例，我的 RHEL 是否与本地虚拟机工作不同？

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的 RHEL 虚拟机相同，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，而不是私有编排接口，来管理云资源。
- 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有云提供商的兼容性。

与本地服务器相比，我的数据在公有云中是否保持安全？

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

RHEL 公共云实例的一般安全性如下：

- 您的公有云供应商负责云 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中运行实例。

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 实例的云提供商有：
   - Amazon Web Services (AWS)
     - 如需更多信息，请参阅 在 Amazon Web Services 上部署 RHEL 8 。
   - Google Cloud Platform (GCP)
     - 如需更多信息，请参阅 在 Google Cloud Platform 上部署 RHEL 8 。
   - Microsoft Azure
     - 如需更多信息，请参阅 在 Microsoft Azure 上部署 RHEL 8 。
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

要在公有云平台上部署 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，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。

直接从云提供商市场购买 RHEL 实例。

- 您按小时为使用的服务后付费。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。

其他资源

- 什么是黄金镜像？

第 2 章 CLOUD-INIT 简介

cloud-init 工具在系统引导过程中自动初始化云实例。您可以配置 **cloud-init** 来执行各种任务：

- 配置主机名
- 在实例上安装软件包
- 运行脚本
- 限制默认虚拟机的行为

先决条件

- 注册一个红帽客户门户网站（Red Hat Customer Portal） 帐户。

cloud-init 在各种 RHEL 镜像中提供。例如：

- 如果您从 红帽客户门户网站 下载了 KVM 客户机镜像，则该镜像预安装了 **cloud-init** 软件包。启动实例后，**cloud-init** 软件包变为启用。红帽客户门户网站上的 KVM 客户机镜像旨在用于 Red Hat Virtualization (RHV)、Red Hat OpenStack Platform (RHOSP)和 Red Hat OpenShift Virtualization。
- 您还可以从红帽客户门户网站下载 RHEL ISO 镜像，以创建自定义客户机镜像。在这种情况下，您需要在自定义的客户机镜像上安装 **cloud-init** 软件包。
- 如果您需要使用来自云服务提供商（如 AWS 或 Azure）的镜像，请使用 *RHEL 镜像构建器* 来创建镜像。镜像构建器镜像是为特定云提供商自定义的。以下镜像类型包括已安装的 **cloud-init** ：
  - Amazon Machine Image (AMI)
  - 虚拟硬盘(VHD)
  - QEMU copy-on-write (qcow2)
    有关 RHEL 镜像构建器的详情，请参阅 制作自定义的 RHEL 系统镜像。

大多数云平台都支持 **cloud-init**，但配置过程和支持的选项有所不同。或者，您可以为 NoCloud 环境配置 **cloud-init**。

另外，您可以在一个虚拟机上配置 **cloud-init**，然后将该虚拟机用作模板来创建其它虚拟机或虚拟机的集群。

特定的红帽产品（如 Red Hat Virtualization ）记录了为这些产品配置 **cloud-init** 的流程。

2.1. CLOUD-INIT 配置的概述

cloud-init 工具使用 YAML 格式的配置文件，来将用户定义的任务应用到实例。当实例启动时，**cloud-init** 服务启动，并执行 YAML 文件中的指令。根据配置，任务要么在第一次引导过程中完成，要么在虚拟机的后续启动时完成。

要定义特定的任务，请配置 **/etc/cloud/cloud.cfg** 文件，并在 **/etc/cloud/cloud.cfg.d/** 目录下添加指令。

- **cloud.cfg** 文件包含各种系统配置的指令，如用户访问、身份验证和系统信息。
  文件还包括 **cloud-init** 的默认和可选模块。这些模块在以下阶段中按顺序执行：**cloud-init** 初始化阶段 ..配置阶段 ..最终阶段。

+ 在 **cloud.cfg** 文件中，这三个阶段的模块分别列在 **cloud_init_modules**、**cloud_config_modules** 和 **cloud_final_modules** 下。

- 您可以在 **cloud.cfg.d** 目录中为 **cloud-init** 添加其它指令。当向 **cloud.cfg.d** 目录中添加指令时，您需要将它们添加到名为 ***.cfg** 的自定义文件中，并且在文件的顶部始终包含 **#cloud-config**。

2.2. CLOUD-INIT 以阶段形式运行

在系统引导过程中，**cloud-init** 工具在五个阶段运行，以确定 **cloud-init** 是否运行，以及在哪里找到数据源和其它任务。这些阶段包括：

1. **生成器阶段** ：通过使用 **systemd** 服务，此阶段决定在引导时是否运行 **cloud-init** 工具。
2. **本地阶段** ：**cloud-init** 搜索本地数据源，并应用网络配置，包括基于 DHCP 的回退机制。
3. **网络阶段** ：**cloud-init** 通过运行 **/etc/cloud/cloud.cfg** 文件中 **cloud_init_modules** 下列出的模块来处理用户数据。您可以在 **cloud_init_modules** 部分种添加、删除、启用或禁用模块。
4. **配置 阶段:cloud-init** 运行 **/etc/cloud/cloud.cfg** 文件中 **cloud_config_modules** 部分下列出的模块。您可以在 **cloud_config_modules** 部分种添加、删除、启用或禁用模块。
5. **最后阶段** ：**cloud-init** 运行 **/etc/cloud/cloud.cfg** 文件的 **cloud_final_modules** 部分中包含的模块和配置。它可以包括特定软件包的安装，以及触发配置管理插件和用户定义的脚本。您可以在 **cloud_final_modules** 部分种添加、删除、启用或禁用模块。

其他资源

- cloud-init 的引导阶段

2.3. CLOUD-INIT 模块分阶段执行

当 **cloud-init** 运行时，它会在三个阶段中按顺序执行 **cloud.cfg** 中的模块：

1. 网络阶段(**cloud_init_modules**)
2. 配置阶段(**cloud_config_modules**)
3. 最终阶段(**cloud_final_modules**)

当首次在虚拟机上运行 **cloud-init** 时，您配置的所有模块都会在相应的阶段中运行。在后续的 **cloud-init** 运行中，某个模块是否在一个阶段中运行取决于单独模块中的*模块频率*。有些模块在每次运行 **cloud-init** 时都会运行，一些模块只在 **cloud-init** 第一次运行时运行，即使实例 ID 发生了变化。

注意

用于唯一标识实例的实例 ID。当实例 ID 发生变化时，**cloud-init** 将实例视为新实例。

可能的*模块频率*值如下：

- **Per instance** 意味着模块在实例第一次引导时运行。例如，如果您克隆一个实例或从保存的镜像创建了一个新实例，指定为 per instance 的模块会再次运行。
- **Per once** 表示模块只运行一次。例如，如果您克隆实例或从保存的镜像创建新实例，指定为 “Per once” 的实例不会在这些实例上再次运行。

- **Per always** 表示模块在每次引导时都运行。

注意

您可在配置模块时或使用命令行覆盖模块频率的设置。

2.4. CLOUD-INIT 使用用户数据、元数据和厂商数据

cloud-init 消耗的数据源是用户数据、元数据和厂商数据。

- 用户数据包括您在 **cloud.cfg** 文件和 **cloud.cfg.d** 目录中指定的指令，例如：用户数据可以包括要运行的文件、要安装的软件包和 shell 脚本。有关 **cloud-init** 允许的用户数据类型的信息，请参阅 **cloud-init** 文档部分 User-Data 格式。

- 元数据包括与特定数据源关联的数据，例如：元数据可以包括服务器名称和实例 ID。如果您使用一个特定的云平台，该平台将决定您的实例在何处查找用户数据和元数据。您的平台可能要求您将元数据和用户数据添加到 HTTP 服务；在这种情况下，当 **cloud-init** 运行它时，cloud-init 会使用来自 HTTP 服务的元数据和用户数据。

- 供应商数据由机构（例如云提供商）可选择地提供，并包含可自定义镜像以更好地适应镜像运行的环境的信息。在读取任何元数据并初始化系统后，**cloud-init** 对可选的厂商数据和用户数据进行操作。默认情况下，厂商数据会在第一次引导时运行。您可以禁用厂商数据执行。
  如需元数据的描述，请参阅 **cloud-init** 文档部分 实例元数据 ；如需数据源的列表，请参阅 数据源 ；如需有关供应商数据的更多信息，请参阅 供应商数据。

2.5. CLOUD-INIT 标识云平台

cloud-init 尝试使用脚本 **ds-identify** 来识别云平台。该脚本在一个实例第一次引导时运行。

添加一个数据源指令可在 **cloud-init** 运行时节省时间。您可以在 **/etc/cloud/cloud.cfg** 文件中或者在 **/etc/cloud/cloud.cfg.d** 目录中添加该指令。例如：

```
datasource_list:[Ec2]
```

除了为云平台添加此指令外，您还可以通过添加额外的配置详情，如元数据 URL 来进一步配置 **cloud-init**。

```
datasource_list: [Ec2]
datasource:
  Ec2:
    metadata_urls: ['http://169.254.169.254']
```

在 **cloud-init** 运行后，您可以查看日志文件(**run/cloud-init/ds-identify.log**)，其中提供有关平台的详细信息。

其他资源

- datasources
- 如何识别正在使用的数据源
- 如何调试用户数据？

2.6. 其他资源

- cloud-init 的上游文档

第 3 章 红帽对 CLOUD-INIT 的支持

红帽支持 **cloud-init** 实用程序、**cloud-init** 模块以及各种红帽产品中的默认目录和文件。

3.1. CLOUD-INIT 重要目录和文件

通过使用下表中的目录和文件，您可以执行以下任务：

- 配置 **cloud-init**
- **cloud-init** 运行后查找配置的信息
- 检查日志文件
- 查找模板

根据您的场景和数据源，可能还会有其他对您配置很重要的文件和目录。

表 3.1. cloud-init 目录和文件

目录或文件	描述
/etc/cloud/cloud.cfg	**cloud.cfg** 文件包含基本的 **cloud-init** 配置，您可以了解到模块会在哪个阶段运行。
/etc/cloud/cloud.cfg.d	**cloud.cfg.d** 目录，您可以在其中为 **cloud-init** 添加附加指令。
/var/lib/cloud	当 **cloud-init** 运行时，它会在 **/var/lib/cloud** 下创建一个目录布局。布局包括特定于您的实例配置的目录和文件。
/usr/share/doc/cloud-init/examples	**examples** 目录包含多个示例。您可以使用它们来帮助建模您自己的指令。
/etc/cloud/templates	这个目录包括您可以在特定情况下，在 **cloud-init** 中启用的模板。模板提供启用的指示。
/var/log/cloud-init.log	**cloud-init.log** 文件提供了有助于调试的日志信息。
/run/cloud-init	**/run/cloud-init** 目录包含有关数据源和 **ds-identify** 脚本的日志信息。

3.2. 使用 CLOUD-INIT 的红帽产品

您可以将 **cloud-init** 与以下红帽产品一起使用：

- Red Hat Virtualization。在虚拟机上安装 **cloud-init** 后，您可以创建一个模板，并在从该模板创建的所有虚拟机上使用 **cloud-init** 功能。有关将 **cloud-init** 用于虚拟机的信息，请参阅使用 Cloud-Init 自动化虚拟机的配置。

- **Red Hat OpenStack Platform。**您可以使用 **cloud-init** 来帮助为 OpenStack 配置镜像。如需更多信息，请参阅 实例和镜像指南。
- **Red Hat Satellite。**您可以将 **cloud-init** 与 Red Hat Satellite 搭配使用。如需更多信息，参阅 Red Hat Virtualization 中准备 Cloud-init 镜像 。
- **Red Hat OpenShift。**您可以为 OpenShift 创建虚拟机时使用 **cloud-init**。如需更多信息，参阅创建虚拟机。

3.3. 红帽支持这些 CLOUD-INIT 模块

红帽支持大多数 **cloud-init** 模块。单个模块可以包含多个配置选项。在下表中，您可以找到红帽当前支持的所有 **cloud-init** 模块，并提供简短描述和默认模块频率。有关 这些模块 的完整描述和选项，请参阅 **cloud-init** 文档部分中的模块。

表 3.2. 支持的 cloud-init 模块

cloud-init 模块	描述	默认模块频率
bootcmd	在引导过程早期运行命令	per always
ca_certs	添加 CA 证书	per instance
debug	启用或禁用内部信息输出来帮助调试	per instance
disable_ec2_metadata	启用或禁用 AWS EC2 元数据	per always
disk_setup	配置简单的分区表和文件系统	per instance
final_message	指定 **cloud-init** 完成后的输出消息	per always
foo	用于显示模块结构的示例（模块什么都不做）	per instance
growpart	重新定义分区大小以填充可用磁盘空间	per always
keys_to_console	允许控制可写入控制台的指纹和密钥	per instance
landscape	安装并配置 landscape 客户端	per instance
locale	配置系统区域设置并应用系统范围	per instance
mcollective	安装、配置和启动 **mcollective**	per instance
migrator	将旧版本的 **cloud-init** 移到更新的版本	per always

cloud-init 模块	描述	默认模块频率
mounts	配置挂载点和交换文件	per instance
phone_home	引导完成后将数据发送到远程主机	per instance
power_state_change	在所有配置模块运行后完成关闭并重启	per instance
puppet	安装并配置 puppet	per instance
resizefs	重新定义文件系统大小以使用分区上的所有可用空间	per always
resolv_conf	配置 **resolv.conf**	per instance
rh_subscription	注册 Red Hat Enterprise Linux 系统	per instance
rightscale_userdata	为 **cloud-init** 添加对 RightScale 配置 hook 的支持	per instance
rsyslog	使用 **rsyslog** 配置远程系统日志	per instance
runcmd	运行任意命令	per instance
salt_minion	安装、配置和启动 **salt minion**	per instance
scripts_per_boot	每个引导脚本运行	per always
scripts_per_instance	每个实例脚本运行	per instance
scripts_per_once	运行脚本一次	per once
scripts_user	运行用户脚本	per instance
scripts_vendor	运行厂商脚本	per instance
seed_random	提供随机 seed 数据	per instance
set_hostname	设置主机名和完全限定域名（FQDN）	per always
set_passwords	设置用户密码并启用或禁用 SSH 密码验证	per instance
ssh_authkey_fingerprints	用户 SSH 密钥的日志指纹	per instance

cloud-init 模块	描述	默认模块频率
ssh_import_id	导入 SSH 密钥	per instance
ssh	配置 SSH，主机及授权 SSH 密钥	per instance
timezone	设置系统时区	per instance
update_etc_hosts	更新 **/etc/hosts**	per always
update_hostname	更新主机名和 FQDN	per always
users_groups	配置用户和组	per instance
write_files	写入任意文件	per instance
yum_add_repo	在系统中添加 yum 软件仓库配置	per always

红**帽不支持** 以下模块列表：

表 3.3. 不支持的模块

模块
apt_configure
apt_pipeline
byobu
chef
emit_upstart
grub_dpkg
ubuntu_init_switch

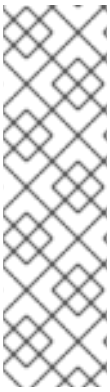
3.4. 默认的 CLOUD.CFG 文件

/etc/cloud/cloud.cfg 文件列出了由 **cloud-init** 基本配置组成的模块。

文件中的模块是 **cloud-init** 的默认模块。您可以为环境配置模块或删除您不需要的模块。包含在 **cloud.cfg** 中的模块不一定通过在文件中列出而执行任何操作。如果您希望在 **cloud-init** 的一个阶段中执行操作，则需要单独配置它们。

cloud.cfg 文件提供运行各个模块的时序。只要红帽支持这些模块，您就可以在 **cloud.cfg** 中添加额外的模块。

Red Hat Enterprise Linux(RHEL)文件的默认内容如下：



注意

- 模块按照 **cloud.cfg** 中给出的顺序运行。您通常不会更改这个顺序。
- **cloud.cfg** 指令可以被用户数据覆盖。
- 当手动运行 **cloud-init** 时，您可以使用命令行选项覆盖 **cloud.cfg**。
- 每个模块都有其自身的配置选项，您可以在其中添加特定信息。
- 为确保配置的最佳功能，首选使用带有下划线(_)而非短划线(-)的模块名称。

```
users: 1
 - default

disable_root: true 2
resize_rootfs_tmp: /dev
ssh_pwauth:   false 3

mount_default_fields: [~, ~, 'auto', 'defaults,nofail,x-systemd.requires=cloud-init.service', '0', '2'] 4
ssh_deletekeys:   true 5
ssh_genkeytypes: ['rsa', 'ecdsa', 'ed25519'] 6
syslog_fix_perms: ~ 7
disable_vmware_customization: false 8

cloud_init_modules: 9
  - migrator
  - seed_random
  - bootcmd
  - write_files
  - growpart
  - resizefs
  - disk_setup
  - mounts
  - set_hostname
  - update_hostname
  - update_etc_hosts
  - ca_certs
  - rsyslog
  - users_groups
  - ssh

cloud_config_modules: 10
  - ssh_import_id
  - locale
  - set_passwords
  - rh_subscription
  - spacewalk
  - yum_add_repo
  - ntp
  - timezone
  - disable_ec2_metadata
```

```
  - runcmd

cloud_final_modules: 11
  - package_update_upgrade_install
  - write_files_deferred
  - puppet
  - chef
  - ansible
  - mcollective
  - salt_minion
  - reset_rmc
  - rightscale_userdata
  - scripts_vendor
  - scripts_per_once
  - scripts_per_boot
  - scripts_per_instance
  - scripts_user
  - ssh_authkey_fingerprints
  - keys_to_console
  - install_hotplug
  - phone_home
  - final_message
  - power_state_change

system_info:
  default_user: 12
    name: cloud-user
    lock_passwd: true
    gecos: Cloud User
    groups: [adm, systemd-journal]
    sudo: ["ALL=(ALL) NOPASSWD:ALL"]
    shell: /bin/bash
   distro: rhel 13
   network:
    renderers: ['sysconfig', 'eni', 'netplan', 'network-manager', 'networkd']
  paths:
    cloud_dir: /var/lib/cloud 14
    templates_dir: /etc/cloud/templates 15
  ssh_svcname: sshd 16

# vim:syntax=yaml
```

1. 指定系统的默认用户。详情请参考 用户和组群。
2. 启用或禁用 root 登录。如需更多信息，请参阅认证密钥。
3. 指定 **ssh** 是否配置为接受密码身份验证。详情请参考设定密码。
4. 配置挂载点 ; 必须是一个包含六个值的列表。详情请参考挂载。
5. 指定是否删除默认主机 SSH 密钥。详情请参考主机密钥。
6. 指定要生成的密钥类型。详情请参考主机密钥。请注意，对于 RHEL 8.4 和更早版本，此行的默认值为 ~。

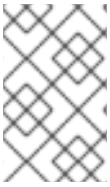
7. **cloud-init** 在引导过程的多个阶段运行。设置这个选项，以便 **cloud-init** 可以将所有阶段记录到其日志文件中。在 **usr/share/doc/cloud-init/examples** 目录中的 **cloud-config.txt** 文件中查找有关这个
8. 启用或禁用 VMware vSphere 自定义
9. 本节中的模块是在引导过程早期，在 **cloud-init** 服务启动时运行的服务。
10. 这些模块在 **cloud-init** 配置期间运行，在初始引导后运行。
11. 这些模块在配置完成后在 **cloud-init** 的最终阶段中运行。
12. 指定默认用户的详情。详情请参考用户和组群。
13. 指定发布
14. 指定包含 **cloud-init** 特定子目录的主目录。如需更多信息，请参阅 目录布局。
15. 指定模板所处的位置
16. SSH 服务的名称

其它资源

- 模块

3.5. CLOUD.CFG.D 目录

cloud-init 遵循您提供和配置的指令。通常，这些指令包含在 **cloud.cfg.d** 目录中。



注意

虽然您可以通过在 **cloud.cfg** 文件中添加用户数据指令来配置模块，但基于最佳实践，最好不要修改 **cloud.cfg**。将您的指令添加到 **/etc/cloud/cloud.cfg.d** 目录中。在这个目录中添加指令可方便将来的修改和升级。

可以通过多种方法添加指令。您可以在名为 ***.cfg** 的文件中包含指令，其中包括标题 **#cloud-config**。通常，该目录会包含多个 ***cfg** 文件。添加指令的其它选项，例如：您可以添加用户数据脚本。详情请参考 User-Data Formats。

其它资源

- 云配置示例

3.6. 默认 05_LOGGING.CFG 文件

05_logging.cfg 文件设置 **cloud-init** 的日志信息。**/etc/cloud/cloud.cfg.d** 目录包括了此文件，以及您添加的其他 **cloud-init** 指令。

cloud-init 默认使用 **05_logging.cfg** 中的日志配置。Red Hat Enterprise Linux(RHEL)文件的默认内容如下：

```
## This yaml formatted config file handles setting
## logger information.  The values that are necessary to be set
```

```
## are seen at the bottom.  The top '_log' are only used to remove
## redundancy in a syslog and fallback-to-file case.
##
## The 'log_cfgs' entry defines a list of logger configs
## Each entry in the list is tried, and the first one that
## works is used.  If a log_cfg list entry is an array, it will
## be joined with '\n'.
_log:
 - &log_base |
   [loggers]
   keys=root,cloudinit

   [handlers]
   keys=consoleHandler,cloudLogHandler

   [formatters]
   keys=simpleFormatter,arg0Formatter

   [logger_root]
   level=DEBUG
   handlers=consoleHandler,cloudLogHandler

   [logger_cloudinit]
   level=DEBUG
   qualname=cloudinit
   handlers=
   propagate=1

   [handler_consoleHandler]
   class=StreamHandler
   level=WARNING
   formatter=arg0Formatter
   args=(sys.stderr,)

   [formatter_arg0Formatter]
   format=%(asctime)s - %(filename)s[%(levelname)s]: %(message)s

   [formatter_simpleFormatter]
   format=[CLOUDINIT] %(filename)s[%(levelname)s]: %(message)s
 - &log_file |
   [handler_cloudLogHandler]
   class=FileHandler
   level=DEBUG
   formatter=arg0Formatter
   args=('/var/log/cloud-init.log',)
 - &log_syslog |
   [handler_cloudLogHandler]
   class=handlers.SysLogHandler
   level=DEBUG
   formatter=simpleFormatter
   args=("/dev/log", handlers.SysLogHandler.LOG_USER)

log_cfgs:
# Array entries in this list will be joined into a string
# that defines the configuration.
#
```

```
# If you want logs to go to syslog, uncomment the following line.
# - [ *log_base, *log_syslog ]
#
# The default behavior is to just log to a file.
# This mechanism that does not depend on a system service to operate.
 - [ *log_base, *log_file ]
# A file path can also be used.
# - /etc/log.conf

# This tells cloud-init to redirect its stdout and stderr to
# 'tee -a /var/log/cloud-init-output.log' so the user can see output
# there without needing to look on the console.
output: {all: '| tee -a /var/log/cloud-init-output.log'}
```

其他资源

- 日志

3.7. CLOUD-INIT /VAR/LIB/CLOUD 目录布局

当 **cloud-init** 首次运行时，它会创建一个目录布局，其中包含有关您的实例和 **cloud-init** 配置的信息。

目录可以包含可选目录，如 **/scripts/vendor**。

以下是 **cloud-init** 的目录布局示例：

```
/var/lib/cloud/
    - data/
       - instance-id
       - previous-instance-id
       - previous-datasource
       - previous-hostname
       - result.json
       - set-hostname
       - status.json
    - handlers/
    - instance
       - boot-finished
       - cloud-config.txt
       - datasource
       - handlers/
       - obj.pkl
       - scripts/
       - sem/
       - user-data.txt
       - user-data.txt.i
       - vendor-data.txt
       - vendor-data.txt.i
    - instances/
        f111ee00-0a4a-4eea-9c17-3fa164739c55/
          - boot-finished
          - cloud-config.txt
          - datasource
          - handlers/
          - obj.pkl
```

```
        - scripts/
        - sem/
        - user-data.txt
        - user-data.txt.i
        - vendor-data.txt
        - vendor-data.txt.i
    - scripts/
      - per-boot/
      - per-instance/
      - per-once/
      - vendor/
    - seed/
    - sem/
      - config_scripts_per_once.once
```

其他资源

- 目录布局

第 4 章 配置 CLOUD-INIT

通过使用 **cloud-init**，可以执行各种配置任务。

您的 **cloud-init** 配置可能需要在 **cloud.cfg** 文件和 **cloud.cfg.d** 目录中添加指令。或者，您的具体数据源可能需要您在文件中添加指令，如用户数据文件和元数据文件。数据源可能需要将您的指令上传到 HTTP 服务器。检查您的数据源要求并相应地添加指令。

4.1. 为 NOCLOUD 数据源创建包含 CLOUD-INIT 的虚拟机

要创建一个包含 **cloud-init** 的新虚拟机(VM)，请创建一个 **meta-data** 文件和 **user-data** 文件。

- **meta-data** 文件包含实例详情。
- **user-data** 文件包含创建用户和授予访问权限的信息。

将这些文件包含在新 ISO 镜像中，并将 ISO 文件附加到从 KVM 客户机镜像创建的新虚拟机上。在这种情况下，数据源是 NoCloud。

流程

1. 创建一个名为 **cloudinitiso** 的目录，并将其设置为您的工作目录：

```
$ mkdir cloudinitiso
$ cd cloudinitiso
```

2. 创建 **meta-data** 文件，并添加以下信息：

```
instance-id: citest
local-hostname: citest-1
```

3. 创建 **user-data** 文件，并添加以下信息：

```
#cloud-config
password: cilogon
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
  - ssh-rsa AAA...fhHQ== sample@redhat.com
```

注意

user-data 文件的最后一行引用一个 SSH 公钥。在 **~/.ssh/id_rsa.pub** 中查找您的 SSH 公钥。在尝试这个示例步骤时，请将该行修改为包含您的一个公钥。

4. 使用 **genisoimage** 命令创建一个包含 **user-data** 和 **meta-data** 的 ISO 镜像：

```
# genisoimage -output ciiso.iso -volid cidata -joliet -rock user-data meta-data

I: -input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 331
Total directory bytes: 0
```

```
Path table size(bytes): 10
Max brk space used 0
183 extents written (0 MB)
```

5. 从红帽客户门户下载 KVM 客户机镜像到 **/var/lib/libvirt/images** 目录。

6. 使用 **virt-install** 工具从 KVM 客户机镜像创建一个新虚拟机，并将下载的镜像附加到现有镜像上：

```
# virt-install \
    --memory 4096 \
    --vcpus 4 \
    --name mytestcivm \
    --disk /var/lib/libvirt/images/rhel-8.1-x86_64-
kvm.qcow2,device=disk,bus=virtio,format=qcow2 \
    --disk /home/sample/cloudinitiso/ciiso.iso,device=cdrom \
    --os-type Linux \
    --os-variant rhel8.0 \
    --virt-type kvm \
    --graphics none \
    --import
```

7. 使用用户名 **cloud-user** 和密码 **cilogon** 登录到您的镜像：

```
citest-1 login: cloud-user
Password:
[cloud-user@citest-1 ~]$
```

验证

- 检查 **cloud-init** 状态，以确认工具已完成了其定义的任务：

```
[cloud-user@citest-1 instance]$ cloud-init status
status: done
```

- **cloud-init** 工具在运行时在 **/var/lib/cloud** 下创建 **cloud-init** 目录布局，并根据您指定的指令更新或更改某些目录内容。
  例如，您可以通过检查数据源文件来确认数据源为 **NoCloud**。

```
$ cd /var/lib/cloud/instance
$ cat datasource
DataSourceNoCloud: DataSourceNoCloud [seed=/dev/sr0][dsmode=net]
```

- **cloud-init** 将 user-data 复制到 **/var/lib/cloud/instance/user-data.txt** 中：

```
$ cat user-data.txt
#cloud-config
password: cilogon
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
  - ssh-rsa AAA...fhHQ== sample@redhat.com
```

注意

对于 OpenStack，创建和管理实例 包含有关使用 **cloud-init** 配置实例的信息。请参阅为特定流程 创建自定义实例。

其它资源

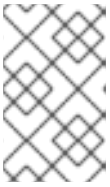
- NoCloud 数据源的上游文档

4.2. 使用 CLOUD-INIT 使云用户密码过期

要强制 **cloud-user** 在第一次登录时更改 **cloud-user** 密码，您可以将其密码设置为过期。

流程

1. 根据数据源的要求，编辑 **user-data** 文件或在 **cloud.cfg.d** 目录中添加以下指令：



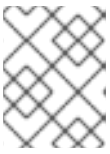
注意

所有用户指令都包括文件顶部的 **#cloud-config**，以便 **cloud-init** 将文件识别为包含用户指令。当您在 **cloud.cfg.d** 目录中包含指令时，将该文件命名为 ***.cfg**，且始终在文件的顶部包含 **#cloud-config**。

2. 将行 **chpasswd: {expire: False}** 改为 **chpasswd: {expire: True}**:

```
#cloud-config
password: mypassword
chpasswd: {expire: True}
ssh_pwauth: True
ssh_authorized_keys:
  - ssh-rsa AAA...SDvz user1@yourdomain.com
  - ssh-rsa AAB...QTuo user2@yourdomain.com
```

这可让密码过期，因为除非您另有说明，否则 **password** 和 **chpasswd** 只针对默认的用户执行操作。



注意

这是一个全局设置。当您将 **chpasswd** 设置为 **True** 时，您创建的所有用户都需要在登录时更改其密码。

4.3. 使用 CLOUD-INIT 更改默认用户名

您可以将默认用户名更改为 **cloud-user** 以外的名称。

流程

1. 根据数据源的要求，编辑 **user-data** 文件或在 **cloud.cfg.d** 目录中添加以下指令：

注意

所有用户指令都包括文件顶部的 **#cloud-config**，以便 **cloud-init** 将文件识别为包含用户指令。当您在 **cloud.cfg.d** 目录中包含指令时，将该文件命名为 ***.cfg**，且始终在文件的顶部包含 **#cloud-config**。

2. 添加行 **user: <username>** ，使用新的默认用户名替换 <username> ：

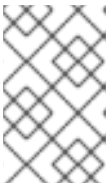
```
#cloud-config
user: username
password: mypassword
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
  - ssh-rsa AAA...SDvz user1@yourdomain.com
  - ssh-rsa AAB...QTuo user2@yourdomain.com
```

4.4. 使用 CLOUD-INIT 设置根密码

要设置根密码，创建一个用户列表。

流程

1. 根据数据源的要求，编辑 **user-data** 文件或在 **cloud.cfg.d** 目录中添加以下指令：



注意

所有用户指令都包括文件顶部的 **#cloud-config**，以便 **cloud-init** 将文件识别为包含用户指令。当您在 **cloud.cfg.d** 目录中包含指令时，将该文件命名为 ***.cfg**，且始终在文件的顶部包含 **#cloud-config**。

2. 在文件的 **chpasswd** 部分中创建一个用户列表：

注意

空格很重要。请勿在您的用户列表的冒号之前或之后包括空格。如果您包含空格，密码会被设置为包括空格。

```
#cloud-config
ssh_pwauth: True
ssh_authorized_keys:
  - ssh-rsa AAA...SDvz user1@yourdomain.com
  - ssh-rsa AAB...QTuo user2@yourdomain.com
chpasswd:
  list: |
     root:myrootpassword
     cloud-user:mypassword
  expire: False
```

注意

如果使用此方法设置用户密码，则必须在本节中设置 *all passwords*。

4.5. 使用 CLOUD-INIT 管理红帽订阅

您可以使用 **rh_subscription** 指令注册您的系统。对于每个订阅，您需要编辑用户数据。

示例 1

- 您可以使用 **auto-attach** 和 **service-level** 选项：
  在 **rh_subscription** 下，添加您的 **username** 和 **password**，将 **auto-attach** 设置为 **True**，并将 **service-level** 设置为 **self-support**。

```
rh_subscription:
  username: sample@redhat.com
  password: 'mypassword'
  auto-attach: True
  service-level: self-support
```

注意

service-level 选项要求您使用 **auto-attach** 选项。

示例 2

- 您可以使用 **activation-key** 和 **org** 选项：
  在 **rh_subscription** 下，添加您的 **activation key** 和 **org** 号，并将 **auto-attach** 设置为 **True**。

```
rh_subscription:
  activation-key: example_key
  org: 12345
  auto-attach: True
```

示例 3

- 您可以添加订阅池：
  在 **rh_subscription** 下，添加您的 **username**, **password** 和池号。

```
rh_subscription:
  username: sample@redhat.com
  password: 'password'
  add-pool: XYZ01234567
```

注意

此示例等同于 **subscription-manager attach --pool=XYZ01234567** 命令。

示例 4

- 您可以在 **/etc/rhsm/rhsm.conf** 文件中设置服务器主机名：
  在 **rh_subscription** 下，添加您的 **用户名**、**密码**、**server-hostname**，并将 **auto-attach** 设置为 **True**。

```
rh_subscription:
  username: sample@redhat.com
```

```
password: 'password'
server-hostname: test.example.com
auto-attach: True
```

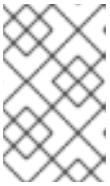
4.6. 使用 CLOUD-INIT 添加用户和用户选项

您可以在 **users** 部分中创建和描述用户。您可以修改这个部分，以在初始系统配置中添加更多用户，也可以设置其他用户选项。

如果添加 **users** 部分，还必须在本节中设置默认用户选项。

流程

1. 根据数据源的要求，编辑 **user-data** 文件或在 **cloud.cfg.d** 目录中添加以下指令：



注意

所有用户指令都包括文件顶部的 **#cloud-config**，以便 **cloud-init** 将文件识别为包含用户指令。当您在 **cloud.cfg.d** 目录中包含指令时，将该文件命名为 ***.cfg**，且始终在文件的顶部包含 **#cloud-config**。

2. 添加或修改 **users** 部分以添加用户。

- 如果您希望 **cloud-user** 与您指定的其他用户一起创建的默认用户，请确保将 **default** 添加为这个部分中的第一个条目。如果这不是第一个条目，则不会创建 **cloud-user**。
- 默认情况下，如果没有 **selinux-user** 值，用户会被标记为 **unconfined_u**。

```
#cloud-config
users:
  - default
  - name: user2
    gecos: User N. Ame
    selinux-user: staff_u
    groups: users,wheel
    ssh_pwauth: True
    ssh_authorized_keys:
      - ssh-rsa AA..vz user@domain.com
chpasswd:
  list: |
    root:password
    cloud-user:mypassword
    user2:mypassword2
  expire: False
```

注意

- 这个示例将用户 **user2** 放入两个组，即 **users** 和 **wheel**。

4.7. 使用 CLOUD-INIT 运行第一个引导命令

您可以使用 **runcmd** 和 **bootcmd** 部分在启动和初始化过程中执行命令。

bootcmd 部分在初始化过程早期执行，并且在每次引导时默认运行。**runcmd** 部分在进程末尾附近执行，且仅在第一次引导和初始化时执行。

流程

1. 根据数据源的要求，编辑 **user-data** 文件或在 **cloud.cfg.d** 目录中添加以下指令：

注意

所有用户指令都包括文件顶部的 **#cloud-config**，以便 **cloud-init** 将文件识别为包含用户指令。当您在 **cloud.cfg.d** 目录中包含指令时，将该文件命名为 ***.cfg**，且始终在文件的顶部包含 **#cloud-config**。

2. 添加 **bootcmd** 和 **runcmd** 部分；包含您希望 **cloud-init** 执行的命令。

```
#cloud-config
users:
  - default
  - name: user2
    gecos: User N. Ame
    groups: users
chpasswd:
  list: |
    root:password
    fedora:myfedpassword
    user2:mypassword2
  expire: False
bootcmd:
 - echo New MOTD >> /etc/motd
runcmd:
 - echo New MOTD2 >> /etc/motd
```

4.8. 使用 CLOUD-INIT 添加额外的 SUDOERS

您可以通过在 **users** 部分添加 **sudo** 和 **groups** 条目，将用户配置为 sudoer。

流程

1. 根据数据源的要求，编辑 **user-data** 文件或在 **cloud.cfg.d** 目录中添加以下指令：

注意

所有用户指令都包括文件顶部的 **#cloud-config**，以便 **cloud-init** 将文件识别为包含用户指令。当您在 **cloud.cfg.d** 目录中包含指令时，将该文件命名为 ***.cfg**，且始终在文件的顶部包含 **#cloud-config**。

2. 添加 **sudo** 条目并指定用户访问权限。例如，**sudo: ALL=(ALL)NOPASSWD:ALL** 允许用户进行不受限制的用户访问。

3. 添加一个 **groups** 条目，并指定包含用户的组：

```
#cloud-config
users:
```

```
  - default
  - name: user2
    gecos: User D. Two
    sudo: ["ALL=(ALL) NOPASSWD:ALL"]
    groups: wheel,adm,systemd-journal
    ssh_pwauth: True
    ssh_authorized_keys:
      - ssh-rsa AA...vz user@domain.com
chpasswd:
  list: |
    root:password
    cloud-user:mypassword
    user2:mypassword2
  expire: False
```

4.9. 使用 CLOUD-INIT 设置静态网络配置

您可以通过在元数据中添加 **network-interfaces** 部分来使用 **cloud-init** 设置网络配置。

Red Hat Enterprise Linux 通过 **NetworkManager** 提供其默认的网络服务，这是一个动态网络控制和配置守护进程，其在网络设备和连接可用时使它们保持启动和活动状态。

您的数据源可能会提供网络配置。详情请查看 **cloud-init** 网络配置源部分。

如果您没有为 **cloud-init** 指定网络配置，且没有禁用网络配置，**cloud-init** 会尝试确定任何附加的设备是否有连接。如果找到连接的设备，它会生成在接口上发出 DHCP 请求的网络配置。如需更多信息，请参阅 **cloud-init** 文档中的 Fallback Network Configuration 部分。

流程

以下示例添加了静态网络配置。

1. 根据数据源的要求，编辑 **user-data** 文件或在 **cloud.cfg.d** 目录中添加以下指令：

注意

所有用户指令都包括文件顶部的 **#cloud-config**，以便 **cloud-init** 将文件识别为包含用户指令。当您在 **cloud.cfg.d** 目录中包含指令时，将该文件命名为 ***.cfg**，且始终在文件的顶部包含 **#cloud-config**。

2. 添加 **network-interfaces** 部分。

```
network:
  version: 1
  config:
    - type: physical
      name: eth0
      subnets:
        - type: static
          address: 192.0.2.1/24
          gateway: 192.0.2.254
```

注意

您可以通过在您的元数据中添加以下信息来禁用网络配置。

```
network:
  config: disabled
```

其它资源

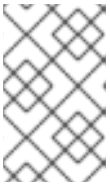
- 网络配置
- NoCloud

4.10. 使用 CLOUD-INIT 仅配置 ROOT 用户

您可以配置用户数据，以便您有一个 root 用户，而没有其他用户。

流程

1. 根据数据源的要求，编辑 **user-data** 文件或在 **cloud.cfg.d** 目录中添加以下指令：



注意

所有用户指令都包括文件顶部的 **#cloud-config**，以便 **cloud-init** 将文件识别为包含用户指令。当您在 **cloud.cfg.d** 目录中包含指令时，将该文件命名为 ***.cfg**，且始终在文件的顶部包含 **#cloud-config**。

2. 在 **users** 部分中，为 **root** 用户创建一个条目。
以下示例中包含了一个 **users** 部分，其中仅包含 **name** 选项。

```
users:
  - name: root
chpasswd:
  list: |
    root:password
  expire: False
```

3. 另外，还可为 root 用户设置 SSH 密钥。

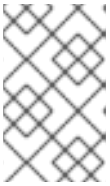
```
users:
  - name: root
    ssh_pwauth: True
    ssh_authorized_keys:
      - ssh-rsa AA..vz user@domain.com
```

4.11. 在 CLOUD-INIT 中使用 CONTAINER-STORAGE-SETUP 设置存储

您可以通过引用 **write_files** 模块中的 **container-storage-setup** 实用程序来设置存储。

流程

1. 根据数据源的要求，编辑 **user-data** 文件或在 **cloud.cfg.d** 目录中添加以下指令：

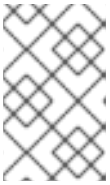


注意

所有用户指令都包括文件顶部的 **#cloud-config**，以便 **cloud-init** 将文件识别为包含用户指令。当您在 **cloud.cfg.d** 目录中包含指令时，将该文件命名为 ***.cfg**，且始终在文件的顶部包含 **#cloud-config**。

2. 添加或修改 **write_files** 模块，使其包含 **container-storage-setup** 实用程序的路径。以下示例将 root 逻辑卷的大小设置为 6 GB，而不是默认的 3 GB。

```
write_files:
  - path: /etc/sysconfig/docker-storage-setup
    permissions: 0644
    owner: root
    content: |
    ROOT_SIZE=6G
```



注意

在 RHEL 7.4 之前，**container-storage-setup** 的名称为 **docker-storage-setup**。如果您使用 OverlayFS 进行存储，从 RHEL 7.4 开始，您现在可以使用 SELinux 处于 enforcing 模式的文件系统。

4.12. 使用 CLOUD-INIT 更改系统区域设置

您可以使用 **locale** 模块配置系统位置。

流程

1. 根据数据源的要求，编辑 **meta-data** 文件。您还可以在 **cloud.cfg** 文件或 **cloud.cfg.d** 目录中添加以下指令：

2. 添加 **locale** 指令，指定位置。以下示例将 **locale** 设置设置为使用 **UTF-8** 编码的 **ja_JP** （日本）。

```
#cloud-config
locale: ja_JP.UTF-8
```

其他资源

- 设置系统区域设置

4.13. CLOUD-INIT 和 SHELL 脚本

您可以将列表值或字符串值添加到 **bootcmd** 或 **runcmd**。您还可以在 userdata 中提供 shell 脚本。

- 如果您使用 **bootcmd** 或 **runcmd** 的列表值，则每个列表项依次使用 **execve** 运行。
- 如果您使用字符串值，则整个字符串作为 shell 脚本运行。
- 如果要使用 **cloud-init** 运行 shell 脚本，您可以提供一个 shell 脚本（使用 shebang(#!)完成），而不是提供带有一个 **.yaml** 文件的 **cloud-init**。

有关如何在 **bootcmd** 和 **runcmd** 中放置 shell 脚本的示例，请参阅 在第一次引导时运行命令。

4.14. 防止 CLOUD-INIT 更新配置文件

从备份镜像创建或恢复实例时，实例 ID 会改变。通过实例 ID 中的更改，**cloud-init** 工具更新配置文件。但是，您可以确保 **cloud-init** 在从备份创建或恢复时不会更新某些配置文件。

流程

1. 编辑 **/etc/cloud/cloud.cfg** 文件，例如：

```
# vi /etc/cloud/cloud.cfg
```

2. 注释或删除在恢复实例时不需要 **cloud-init** 更新的配置。例如，为了避免更新 SSH 密钥文件，请从 **cloud_init_modules** 部分删除 **-ssh**。

```
cloud_init_modules:
 - disk_setup
 - migrator
 - bootcmd
 - write-files
 - growpart
 - resizefs
 - set_hostname
 - update_hostname
 - update_etc_hosts
 - rsyslog
 - users-groups
 # - ssh
```

验证

- 要检查 **cloud-init** 更新的配置文件，请检查 **/var/log/cloud/cloud-init.log** 文件。在实例启动期间记录更新的文件，消息以 **Writing to** 开始。例如：

```
2019-09-03 00:16:07,XXX - util.py[DEBUG]: Writing to /root/.ssh/authorized_keys - wb: [XXX] 554 bytes
2019-09-03 00:16:08,XXX - util.py[DEBUG]: Writing to /etc/ssh/sshd_config - wb: [XXX] 3905 bytes
```

4.15. 在 CLOUD-INIT 运行后修改从 KVM 客户机镜像创建的虚拟机

您可以在重新运行 **cloud-init** 工具前修改 **cloud-init** 配置。当您启动安装并启用了 **cloud-init** 软件包的虚拟机时，**cloud-init** 在虚拟机初始引导时以其默认状态运行。

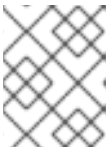
流程

1. 登录到您的虚拟机。

2. 添加或更改指令，例如修改 **/etc/cloud** 目录中的 **cloud.cfg** 文件，或在 **/etc/cloud/cloud.cfg.d** 目录中添加指令。

3. 运行 **cloud-init clean** 命令以清理目录，以便 **cloud-init** 可以再次运行。您还可以以 root 用户身份运行以下命令来清理虚拟机：

```
rm -Rf /var/lib/cloud/instances/
rm -Rf /var/lib/cloud/instance
rm -Rf /var/lib/cloud/data/
```



注意

您可以将清理的镜像保存为新镜像，并将该镜像用于多个虚拟机。新虚拟机将使用更新的 **cloud-init** 配置运行 **cloud-init**。

4. 重新运行 **cloud-init** 或重新引导虚拟机。
cloud-init 重新运行，实现您所做的配置更改。

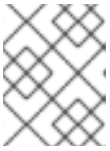
4.16. 在 CLOUD-INIT 运行后为特定的数据源修改虚拟机

您可以在重新运行 **cloud-init** 前修改 **cloud-init** 配置。此流程使用 OpenStack 作为示例数据源。请注意，执行的确切步骤会因数据源而有所不同。

流程

1. 为 OpenStack Platform 创建并启动实例。有关为 OpenStack 创建实例的详情，请参考 创建一个实例。在本例中，虚拟机(VM)包含 **cloud-init**，它在虚拟机引导时运行。
2. 添加或者更改指令。例如，修改存储在 OpenStack HTTP 服务器上的 **user-data.file** 文件。
3. 清理虚拟机。作为 root 运行以下命令。

```
# rm -rf /etc/resolv.conf /run/cloud-init
# userdel -rf cloud-user
# hostnamectl set-hostname localhost.localdomain
# rm /etc/NetworkManager/conf.d/99-cloud-init.conf
```



注意

您可以将清理的镜像保存为新镜像，并将该镜像用于多个虚拟机。新虚拟机使用更新的 **cloud-init** 配置运行 **cloud-init**。

4. 重新运行 **cloud-init** 或重新引导虚拟机。
cloud-init 重新运行，实现您所做的配置更改。

4.17. CLOUD-INIT 故障排除

运行 **cloud-init** 工具后，您可以通过检查配置和日志文件来对实例进行故障排除。在确定问题后，在您的实例上重新运行 **cloud-init**。您可以从命令行运行 **cloud-init**。详情请运行 **cloud-init --help** 命令。

流程

1. 检查 **cloud-init** 配置文件：
   a. 检查 **/etc/cloud/cloud.cfg** 配置文件。检查 **cloud_init_modules**、**cloud_config_modules** 和 **cloud_final_modules** 下包含哪些模块。
   b. 检查 **/etc/cloud/cloud.cfg.d** 目录中的指令（***.cfg** 文件）。

2. 查看 **/var/log/cloud-init.log** 和 **/var/log/cloud-init-output.log** 文件以了解特定问题的详情。例如，如果 root 分区没有自动扩展，请检查 **growpart** 工具的日志消息。如果文件系统没有扩展，请检查 **resizefs** 的日志消息。例如：

```
# grep resizefs /var/log/cloud-init.log
```

注意

growpart 不支持 LVM。如果您的 root 分区基于 LVM，在第一次引导时不会自动扩展 root 分区。

3. 以 root 用户身份重新运行 **cloud-init** 命令：

   a. 只使用 init 模块重新运行 **cloud-init** ：

   ```
   # /usr/bin/cloud-init -d init
   ```

   b. 使用配置中的所有模块重新运行 **cloud-init** ：

   ```
   # /usr/bin/cloud-init -d modules
   ```

   c. 删除 **cloud-init** 缓存，并强制 **cloud-init** 在引导后运行：

   ```
   # rm -rf /var/lib/cloud/ && /usr/bin/cloud-init -d init
   ```

   d. 清理目录并模拟一个干净的实例：

   ```
   # rm -rf /var/lib/cloud/instances/
   # rm -rf /var/lib/cloud/instance
   # rm -rf /var/lib/cloud/data/
   # reboot
   ```

   e. 重新运行 **cloud-init** 工具：

   ```
   # cloud-init init --local
   # cloud-init init
   ```

其他资源

- cloud-init cli 命令