



Red Hat Enterprise Linux 9

为 RHEL 9 配置和管理 cloud-init

使用 cloud-init 自动初始化云实例

Red Hat Enterprise Linux 9 为 RHEL 9 配置和管理 cloud-init

使用 cloud-init 自动初始化云实例

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律通告

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Configuring_and_managing_cloud-init_for_RHEL_9.ent file |.

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

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

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

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

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

Linux[®] 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 自动初始化云实例。您可以在虚拟机上安装 cloud-init 软件包，也可以选择包含 cloud-init 的 Red Hat Enterprise Linux 镜像。您可以在多个红帽产品中使用 cloud-init。

目录

让开源更具包容性	3
对红帽文档提供反馈	4
第 1 章 CLOUD-INIT 简介	5
1.1. 其他资源	5
1.2. CLOUD-INIT 配置	5
1.3. CLOUD-INIT 以阶段形式运行	6
1.4. CLOUD-INIT 模块分阶段执行	6
1.5. CLOUD-INIT 使用用户数据、元数据和厂商数据	7
1.6. CLOUD-INIT 标识云平台	7
第 2 章 红帽对 CLOUD-INIT 的支持	8
2.1. CLOUD-INIT 重要目录和文件	8
2.2. 使用 CLOUD-INIT 的红帽产品	8
2.3. 红帽支持这些 CLOUD-INIT 模块	9
2.4. 默认的 CLOUD.CFG 文件	11
2.5. CLOUD.CFG.D 目录	14
2.6. 默认 05_LOGGING.CFG 文件	14
2.7. CLOUD-INIT /VAR/LIB/CLOUD 目录布局	16
第 3 章 配置 CLOUD-INIT	18
3.1. 为 NOCLOUD 数据源创建包含 CLOUD-INIT 的虚拟机	18
3.2. 使用 CLOUD-INIT 使云用户密码过期	20
3.3. 使用 CLOUD-INIT 更改默认用户名	20
3.4. 使用 CLOUD-INIT 设置根密码	21
3.5. 使用 CLOUD-INIT 管理红帽订阅	22
3.6. 使用 CLOUD-INIT 添加用户和用户选项	23
3.7. 使用 CLOUD-INIT 运行第一个引导命令	23
3.8. 使用 CLOUD-INIT 添加额外的 SUDOERS	24
3.9. 使用 CLOUD-INIT 设置静态网络配置	25
3.10. 使用 CLOUD-INIT 仅配置 ROOT 用户	26
3.11. 在 CLOUD-INIT 中使用 CONTAINER-STORAGE-SETUP 设置存储	26
3.12. 使用 CLOUD-INIT 更改系统区域设置	27
3.13. CLOUD-INIT 和 SHELL 脚本	27
3.14. 防止 CLOUD-INIT 更新配置文件	28
3.15. 在 CLOUD-INIT 运行后修改从 KVM 客户机镜像创建的虚拟机	28
3.16. 在 CLOUD-INIT 运行后为特定的数据源修改虚拟机	29
3.17. CLOUD-INIT 故障排除	29

让开源更具包容性

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [CTO Chris Wright 的信息](#)。

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。请让我们了解如何改进文档。

- 关于特定内容的简单评论：
 1. 请确定您使用 *Multi-page HTML* 格式查看文档。另外，确定 **Feedback** 按钮出现在文档页的右上方。
 2. 用鼠标指针高亮显示您想评论的文本部分。
 3. 点在高亮文本上弹出的 **Add Feedback**。
 4. 按照显示的步骤操作。
- 要通过 Bugzilla 提交反馈，请创建一个新的 ticket：
 1. 进入 [Bugzilla](#) 网站。
 2. 在 Component 中选择 **Documentation**。
 3. 在 **Description** 中输入您要提供的信息。包括文档相关部分的链接。
 4. 点 **Submit Bug**。

第 1 章 CLOUD-INIT 简介

cloud-init 是一个软件包，可在系统引导期间自动初始化云实例。您可以配置 **cloud-init** 来执行各种任务。可以通过 **cloud-init** 执行的一些任务示例包括：

- 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

您可以从不同的地方获得配置 **cloud-init** 的镜像，具体取决于您计划如何使用它。

- **cloud-init** 软件包已安装在您从[红帽客户门户](#)下载的 KVM 客户机镜像上。当您启动实例时，**cloud-init** 会被启用。从红帽客户门户网站下载的 KVM 客户机镜像用于 Red Hat Virtualization(RHV)和 Red Hat OpenStack Platform(RHOSP)。您也可以从头开始为 RHV 和 RHOSP 创建镜像。
- 另一个选择是从红帽客户门户网站下载 ISO 镜像或创建 ISO 镜像。在这种情况下，您需要在 ISO 镜像中安装 **cloud-init**。
- 如果您计划将镜像与云供应商（如 AWS 或 Azure）搭配使用，请使用 Red Hat Image Builder 创建镜像。镜像构建器（Image Builder）镜像是针对于不同云供应商进行定制的镜像。镜像类型 AMI、VHD 和 qcow2 都已安装了 **cloud-init**。如需有关镜像构建器的信息，请参阅[自定义 RHEL 系统镜像](#)。

大部分云平台都支持 **cloud-init**，但配置过程和支持的选项有所不同。另外，您也可以为非云环境配置 **cloud-init**。

您可以在一个虚拟机上配置 **cloud-init**，然后使用该虚拟机作为其他虚拟机或虚拟机集群的模板。

特定的红帽产品（如 [Red Hat Virtualization](#)）会提供有关在其产品中配置 **cloud-init** 的信息。

本文档在多个位置上将这些信息称为 **cloud-init** 文档。有关 **cloud-init** 的完整信息，请参阅相关的 **cloud-init** 文档。

先决条件

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

1.1. 其他资源

- [cloud-init 文档](#)

1.2. CLOUD-INIT 配置

cloud-init 使用 YAML 格式的文件指令执行任务。您可以通过在 YAML 文件中提供说明来确定您希望 **cloud-init** 执行的初始配置。当实例启动时，**cloud-init** 服务将启动并执行指令。根据 **cloud-init** 配置，在第一次引导或后续启动虚拟机时完成任务。

您可以通过配置 `/etc/cloud/cloud.cfg` 文件并在 `/etc/cloud/cloud.cfg.d/` 目录下添加指令来定义任务。

- **cloud.cfg** 文件包含指令，如用于用户访问和验证以及系统信息的指令。

文件还包括 **cloud-init** 的默认和可选模块。模块在三个阶段内按顺序执行，这些阶段包括 **cloud-init** 初始化阶段、配置阶段和最终阶段。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。

- **cloud.cfg.d** 目录，您可以在其中为 **cloud-init** 添加附加指令。当您向 **cloud.cfg.d** 目录添加指令时，通常会将它们添加到名为 ***.cfg** 的文件中，您需要在每个文顶部包含 **#cloud-config**。

1.3. CLOUD-INIT 以阶段形式运行

在系统引导期间，**cloud-init** 分五个阶段运行。不同阶段会决定是否运行 **cloud-init**，找到其数据源的位置，以及其他一些任务。这些阶段包括：

1. **cloud-init** 生成器 (generator) 阶段，通过 **systemd** 服务决定是否在引导时运行 **cloud-init**。
2. 在本地 (local) 阶段，**cloud-init** 找到本地数据源并应用网络配置。
3. 在网络 (network) 阶段，**cloud-init** 处理用户数据并运行 **cloud.cfg** 文件中的 **cloud_init_modules** 下列出的模块。您可以在 **cloud_init_modules** 部分启用、禁用或添加模块。
4. 在配置 (config) 阶段，**cloud-init** 运行 **cloud.cfg** 文件中的 **cloud_config_modules** 下列出的模块。您可以在 **cloud_config_modules** 部分启用、禁用或添加模块。
5. 在最后 (final) 阶段中，**cloud-init** 可以运行 **cloud.cfg** 文件中的 **cloud_final_modules** 中包含的内容。您可以包括通常在系统引导后运行的软件包安装，也可以包括配置管理插件和用户脚本。您可以在 **cloud_final_modules** 部分启用、禁用或添加模块。

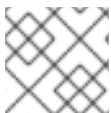
关于这五个引导阶段的相信信息包括在 **cloud-init** 文档的 [Boot Stages](#) 部分。

1.4. 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。



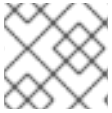
注意

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** 意味着该模块在实例的首次引导时运行。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** 表示该模块只运行一次。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** 意味着该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。

1.5. CLOUD-INIT 使用用户数据、元数据和厂商数据

cloud-init 根据用户数据、元数据和厂商数据运行。

- 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** 文档中的 [User-Data Formats](#) 部分。
- 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。
- 厂商数据由组织（如云提供商）提供，包含了可以用于自定义镜像来更好地在运行环境中运行的信息。**cloud-init** 会在读取任何元数据并初始化系统后，利用可选的供应商数据和用户数据。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。如需元数据的信息，请参阅 **cloud-init** 文档 [Instance Metadata](#)；如需数据源的列表，请参阅 [Datasources](#)；如需厂商数据的信息，请参阅 [Vendor Data](#)。

1.6. 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](#)
- [我使用什么数据源？](#)

第 2 章 红帽对 CLOUD-INIT 的支持

本章论述了红帽对 **cloud-init** 的支持。它包括使用 **cloud-init**、**cloud-init** 模块的红帽产品的信息，以及 Red Hat 支持的默认目录和文件。

2.1. CLOUD-INIT 重要目录和文件

下表包含重要的目录和文件。查看这些目录和文件；这些目录和文件允许您执行类似如下的任务：

- 配置 **cloud-init**
- **cloud-init** 运行后查找您的配置信息
- 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 2.1. 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。

2.2. 使用 CLOUD-INIT 的红帽产品

您可以将 **cloud-init** 与以下红帽产品结合使用：

- **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** 与虚拟机搭配使用的信息，请参阅[使用 Cloud-Init 自动配置虚拟机](#)。

- **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 CloudForms。**您可以使用 **cloud-init** 为 Red Hat CloudForms 调配虚拟机。如需更多信息，请参阅[虚拟机和实例置备的自定义模板](#)。
- **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](#)。

2.3. 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。下表列出了红帽目前支持的所有 **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，参考 **cloud-init** 文档部分中的[模块部分](#)

表 2.2. 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

cloud-init 模块	描述	默认模块频率
migrator	将旧版本的 cloud-init 移到更新的版本	per always
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
resolve_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

cloud-init 模块	描述	默认模块频率
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
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	向系统添加 dnf 存储库配置	per always

下表列出了红帽目前不支持的模块。

表 2.3. 不支持的模块

模块
apt_configure
apt_pipeline
byobu
chef
emit_upstart
grub_dpkg
ubuntu_init_switch

2.4. 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。

```

users: 1
- default

disable_root: 1 2
ssh_pwauth: 0 3

mount_default_fields: [~, ~, 'auto', 'defaults,nofail,x-systemd.requires=cloud-init.service', '0', '2'] 4
ssh_deletekeys: 1 5
ssh_genkeytypes: ['rsa', 'ecdsa', 'ed25519'] 6
syslog_fix_perms: ~ 7
disable_vmware_customization: false 8

cloud_init_modules: 9
- disk_setup
- migrator
- bootcmd
- write-files
- growpart
- resizefs
- set_hostname
- update_hostname
- update_etc_hosts
- rsyslog
- users-groups
- ssh

cloud_config_modules: 10
- mounts
- locale
- set-passwords
- rh_subscription
- dnf-add-repo
- package-update-upgrade-install
- timezone
- puppet

```



```

- chef
- salt-minion
- mcollective
- disable-ec2-metadata
- runcmd

cloud_final_modules: 11
- rightscale_userdata
- scripts-per-once
- scripts-per-boot
- scripts-per-instance
- scripts-user
- ssh-authkey-fingerprints
- keys-to-console
- 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
  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` 文件中找到有关此选项的更多信息。
- 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。详情请参考 [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

- [在哪里可以找到配置文件？](#)
- [模块](#)

2.5. 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

- [在哪里可以找到配置文件？](#)
- [云配置示例](#)

2.6. 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](#)

2.7. 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 3 章 配置 CLOUD-INIT

本章包含 **cloud-init** 最常见的配置任务示例。

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。

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

本节提供了创建包含 **cloud-init** 的新虚拟机(VM)的示例步骤。在此过程中，您可以创建一个 **meta-data** 和 **user-data** 文件。

- 您的 **meta-data** 文件包括了实例的详细信息。
- 您的 **user-data** 文件包含用于创建用户和授予访问权限的信息。

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

步骤

1. 创建名为 **cloudinitiso** 的目录并移动到其中。

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

2. 创建名为 **meta-data** 的文件。在该文件中添加以下信息。

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

3. 创建名为 **user-data** 的文件。在该文件中包含以下信息。

```
#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 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** 命令创建包含 **user-data** 和 **meta-data** 的 ISO 镜像。

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

l: -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)
```

- 从红帽客户门户下载 KVM 客户机镜像到 `/var/lib/libvirt/images` 目录。
- 使用 `virt-install` 命令从 KVM 客户机镜像创建新虚拟机。包含您创建的 ISO 镜像作为镜像的附件。

```
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 rhel9.0 \
  --virt-type kvm \
  --graphics none \
  --import
```

- 以 `cloud-user` 用户身份登录您的镜像。您的密码为 `cion`。

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

验证

- 检查 `cloud-init` 状态，以查看它已完成的任务。

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

- 当 `cloud-init` 在 `/var/lib/cloud` 下创建 `cloud-init` 目录布局，它会根据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`。

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

这些是示例。**cloud-init** 目录布局包含更多信息。



注意

对于 OpenStack, [实例和镜像指南](#) 包含有关使用 **cloud-init** 配置实例的信息。有关特定流程, 请参阅 [创建自定义实例](#)。

其它资源

- [NoCloud](#)

3.2. 使用 CLOUD-INIT 使云用户密码过期

您可以强制 **cloud-user** 在第一次登录时更改 **cloud-user** 密码。执行以下步骤使密码过期。

步骤

1. 根据数据源的要求, 打开您的 `user-data` 文件进行编辑, 或者在 **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. 把行 **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
```

这可使密码过期, 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

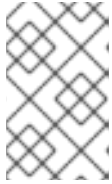
这是一个全局设置。当您将 **chpasswd** 设置为 **True** 时, 您创建的所有用户都需要在登录时更改其密码。

3.3. 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。

步骤

1. 根据数据源的要求, 打开您的 `user-data` 文件进行编辑, 或者在 **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 **user: <username>** 行，将 **<username>** 替换为新的默认用户名。

```
#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
```

3.4. 使用 CLOUD-INIT 设置根密码

要设置根密码，创建一个用户列表。

步骤

1. 根据数据源的要求，打开您的 **user-data** 文件进行编辑，或者在 **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** 部分中创建用户列表。格式在以下示例中显示。



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**

如果使用这个方法设定用户密码，您必须在这个部分中设置 **all passwords**。

3.5. 使用 CLOUD-INIT 管理红帽订阅

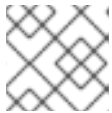
您可以使用 **rh_subscription** 指令注册您的系统。示例如下。对于每个订阅，您可以编辑您的用户数据。

步骤

以下示例使用了 **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。

以下示例使用 **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
```

以下示例添加了一个订阅池。

- 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。

以下示例在 **/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
```

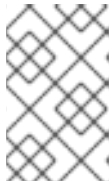
3.6. 使用 CLOUD-INIT 添加用户和用户选项

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。

步骤

1. 根据数据源的要求，打开您的 user-data 文件进行编辑，或者在 **cloud.cfg.d** 目录中添加以下指令。



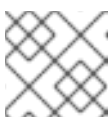
注意

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**。

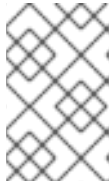
3.7. 使用 CLOUD-INIT 运行第一个引导命令

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。

步骤

1. 根据数据源的要求，打开您的 user-data 文件进行编辑，或者在 **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。

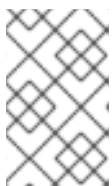
```
#cloud-config
users:
  - default
  - name: user2
    gecos: User N. Ame
    groups: users
chpasswd:
  list: |
    root:password
    fedora:myfedpassword
    user2:mypassword2
  expire: False
bootcmd:
  - echo New MOTD >> /etc/motd
runcmd:
  - echo New MOTD2 >> /etc/motd
```

3.8. 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。

步骤

1. 根据数据源的要求，打开您的 user-data 文件进行编辑，或者在 **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** 允许用户不受限制的用户访问。

3. 添加 **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
```

3.9. 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** 提供其默认网络服务，这是一个动态网络控制和配置守护进程，可在网络设备和连接可用时保持启动和激活。有关 **NetworkManager** 的详情，请参阅[使用 NetworkManager](#)

您的数据源可能会提供网络配置。如需更多信息，请参阅 **cloud-init** 文档中的[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** 会尝试确定是否存在连接的设备。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。

步骤

以下示例添加了静态网络配置。

1. 根据数据源的要求，打开您的 user-data 文件进行编辑，或者在 **cloud.cfg.d** 目录中添加以下指令。



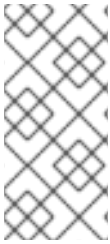
注意

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-interfaces: |
  iface eth0 inet static
  address 192.168.1.10
  network 192.168.1.0
  netmask 255.255.255.0
  broadcast 192.168.1.255
  gateway 192.168.1.254
```

```
bootcmd:
- ifdown eth0
- ifup eth0
```



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](#)

3.10. 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。

步骤

1. 根据数据源的要求，打开您的 user-data 文件进行编辑，或者在 **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。

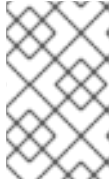
```
users:
- name: root
ssh_pwauth: True
ssh_authorized_keys:
- ssh-rsa AA..vz user@domain.com
```

3.11. 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。

步骤

1. 根据数据源的要求，打开您的 `user-data` 文件进行编辑，或者在 `cloud.cfg.d` 目录中添加以下指令。

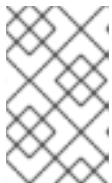


注意

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。下面的例子将根逻辑卷大小设定为 6GB，而不是默认的 3GB。

```
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。

3.12. 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。

步骤

1. 根据数据源的要求，打开您的 `meta-data` 文件进行编辑，或者在 `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

- [Locale](#)

3.13. 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` 中提供一个 shell 脚本。

- 如果您的 **bootcmd** 或 **runcmd** 使用一个列表值，则每个列表项都会依次运行，使用 **execve**。
- 如果您使用字符串值，则整个字符串将作为 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](#)。

3.14. 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。实例 ID 中的更改可能会导致 **cloud-init** 更新配置文件。

执行以下步骤以确保 **cloud-init** 在从备份创建或恢复时不会更新某些配置文件。

步骤

1. 打开 **/etc/cloud/cloud.cfg** 文件进行编辑。
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

您可以检查查看 **cloud-init** 已更新的配置文件。为此，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
```

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

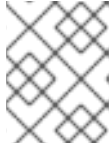
本节提供了在重新运行 **cloud-init** 前要修改 **cloud-init** 配置时的步骤示例。当您启动包含安装并启用 **cloud-init** 软件包的虚拟机时，**cloud-init** 会在虚拟机的初始引导时以默认状态运行。

步骤

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。您也可以以 `root` 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。

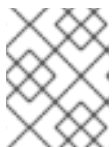
3.16. 在 CLOUD-INIT 运行后为特定的数据源修改虚拟机

本节提供了在重新运行 `cloud-init` 前要修改 `cloud-init` 配置时的步骤示例。以下操作过程使用 OpenStack 作为示例。请注意，这个过程会根据数据源的不同而有所不同。

流程

1. 为 OpenStack Platform 创建并启动实例。如需了解有关为 OpenStack 创建实例的信息，请参阅 [虚拟机实例](#)。在本示例中，虚拟机中包含 `cloud-init`，它在启动虚拟机时运行。
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。

3.17. CLOUD-INIT 故障排除

您可以通过检查配置和日志文件，在 `cloud-init` 运行后对您的实例进行故障排除。确定了问题后，您可以在实例上再次运行 `cloud-init`。

您可以使用 `cloud-init` 命令从命令行运行 `cloud-init`。要查看命令语法，以及可选参数和子命令的说明，请运行 `cloud-init --help` 命令。基本语法如下。

-

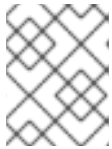
```
cloud-init [-h] [--version] [--file FILES] [--debug] [--force]
{init,modules,single,query,dhclient-hook,features,analyze,devel,collect-logs,clean,status}
```

以下流程提供了识别 **cloud-init** 问题以及用于重新运行计划的示例的建议。

步骤

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，如果问题是 `root` 分区没有自动扩展，请检查 **growpart** 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. 重新运行 **cloud-init**。以下是示例方案。作为 `root` 运行命令。

- 仅使用 `init` 模块重新运行 **cloud-init**。

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

- 使用您的配置中的所有模块重新运行 **cloud-init**。

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

- 删除 **cloud-init** 缓存并强制 **cloud-init** 在启动后运行。

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

- 运行以下命令清理目录，并模拟干净的实例。

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

- 运行以下命令以重新运行 **cloud-init**：

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

其他资源

- CLI 接口