



Red Hat Enterprise Linux 8

Configuração e gerenciamento de nuvens - para RHEL 8

Utilização de nuvens para automatizar a inicialização de instâncias de nuvens

Red Hat Enterprise Linux 8 Configuração e gerenciamento de nuvens - para RHEL 8

Utilização de nuvens para automatizar a inicialização de instâncias de nuvens

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.

Nota Legal

Copyright © 2021 | You need to change the HOLDER entity in the en-US/Configuring_and_Managing_cloud-init_for_RHEL_8.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.

Resumo

Você pode usar o cloud-init para automatizar a inicialização de instâncias de nuvens. Você pode instalar o pacote cloud-init em sua máquina virtual, ou você pode escolher uma imagem do Red Hat Enterprise Linux que inclua cloud-init já instalado. Você pode usar o cloud-init com uma série de produtos Red Hat.

TORNANDO O CÓDIGO ABERTO MAIS INCLUSIVO	3
FORNECENDO FEEDBACK SOBRE A DOCUMENTAÇÃO DA RED HAT	4
CAPÍTULO 1. INTRODUÇÃO AO CLOUD-INIT	5
1.1. CONFIGURAÇÃO EM NUVENS	5
1.2. A NUVEM OPERA EM ETAPAS	6
1.3. OS MÓDULOS DE NUVEM SÃO EXECUTADOS EM FASES	6
1.4. ATUA SOBRE OS DADOS DO USUÁRIO, METADADOS E DADOS DO FORNECEDOR	7
1.5. NUVEM - IDENTIFICA A PLATAFORMA DE NUVENS	7
CAPÍTULO 2. SUPORTE DE CHAPÉU VERMELHO PARA A NUVEM	9
2.1. DIRETÓRIOS E ARQUIVOS IMPORTANTES NA NUVEM	9
2.2. PRODUTOS RED HAT QUE USAM O SISTEMA DE NUVENS	10
2.3. A RED HAT SUPORTA ESTES MÓDULOS COM NUVENS	10
2.4. O ARQUIVO DEFAULT CLOUD.CFG	13
2.5. O DIRETÓRIO CLOUD.CFG.D	16
2.6. O ARQUIVO PADRÃO 05_LOGGING.CFG	16
2.7. O LAYOUT DO DIRETÓRIO DE NUVENS /VAR/LIB/CLOUD	18
CAPÍTULO 3. CONFIGURANDO A NUVEM-EM-IT	20
3.1. CRIAÇÃO DE UMA MÁQUINA VIRTUAL QUE INCLUI UMA FONTE DE DADOS NOCLOUD	20
3.2. EXPIRANDO UMA SENHA DE USUÁRIO NA NUVEM COM A NUVEM	22
3.3. ALTERANDO UM NOME DE USUÁRIO PADRÃO COM A NUVEM	22
3.4. ESTABELECENDO UMA SENHA DE RAIZ COM A NUVEM	23
3.5. GERENCIANDO ASSINATURAS DA RED HAT COM O CLOUD-INIT	24
3.6. ADICIONANDO USUÁRIOS E OPÇÕES DE USUÁRIO COM A NUVEM	25
3.7. EXECUTANDO OS PRIMEIROS COMANDOS DE INICIALIZAÇÃO COM A NUVEM	26
3.8. ACRESCENTAR PROCESSADORES ADICIONAIS COM NUVENS	27
3.9. CONFIGURANDO UMA CONFIGURAÇÃO DE REDE ESTÁTICA COM A NUVEM	27
3.10. CONFIGURANDO APENAS UM USUÁRIO RAIZ COM A NUVEM	28
3.11. INSTALAÇÃO DE ARMAZENAMENTO COM ARMAZENAMENTO DE CONTÊINERES NA NUVEM	29
3.12. MUDANDO O LOCAL DO SISTEMA COM A NUVEM	30
3.13. ROTEIROS COM NUVENS E CONCHAS	30
3.14. IMPEDINDO QUE A NUVEM-INIT ATUALIZE OS ARQUIVOS DE CONFIGURAÇÃO	31
3.15. MODIFICANDO UMA VM CRIADA A PARTIR DE UMA IMAGEM DE CONVIDADO DA KVM DEPOIS QUE A NUVEM ESTIVER EM FUNCIONAMENTO	31
3.16. MODIFICAÇÃO DE UMA VM PARA UMA FONTE DE DADOS ESPECÍFICA DEPOIS QUE A NUVEM ESTIVER EM FUNCIONAMENTO	32
3.17. RESOLUÇÃO DE PROBLEMAS DE NUVENS	33

TORNANDO O CÓDIGO ABERTO MAIS INCLUSIVO

A Red Hat tem o compromisso de substituir a linguagem problemática em nosso código, documentação e propriedades da web. Estamos começando com estes quatro termos: master, slave, blacklist e whitelist. Por causa da enormidade deste esforço, estas mudanças serão implementadas gradualmente ao longo de vários lançamentos futuros. Para mais detalhes, veja a [mensagem de nosso CTO Chris Wright](#).

FORNECENDO FEEDBACK SOBRE A DOCUMENTAÇÃO DA RED HAT

Agradecemos sua contribuição em nossa documentação. Por favor, diga-nos como podemos melhorá-la. Para fazer isso:

- Para comentários simples sobre passagens específicas:
 1. Certifique-se de que você está visualizando a documentação no formato *Multi-page HTML*. Além disso, certifique-se de ver o botão **Feedback** no canto superior direito do documento.
 2. Use o cursor do mouse para destacar a parte do texto que você deseja comentar.
 3. Clique no pop-up **Add Feedback** que aparece abaixo do texto destacado.
 4. Siga as instruções apresentadas.
- Para enviar comentários mais complexos, crie um bilhete Bugzilla:
 1. Ir para o site da [Bugzilla](#).
 2. Como Componente, use **Documentation**.
 3. Preencha o campo **Description** com sua sugestão de melhoria. Inclua um link para a(s) parte(s) relevante(s) da documentação.
 4. Clique em **Submit Bug**.

CAPÍTULO 1. INTRODUÇÃO AO CLOUD-INIT

cloud-init é um pacote de software que automatiza a inicialização de instâncias de nuvem durante a inicialização do sistema. Você pode configurar **cloud-init** para realizar uma variedade de tarefas. Algumas tarefas de exemplo que **cloud-init** pode realizar incluem:

- Configuração de um nome de anfitrião
- Instalação de pacotes em uma instância
- Roteiros em execução
- Suprimindo o comportamento padrão da máquina virtual

Onde você obtém sua imagem para configurar **cloud-init** depende de como você pretende utilizá-la.

- O pacote **cloud-init** está instalado no KVM Guest Images que você pode baixar do [Portal do Cliente da Red Hat](#). Quando você lança uma instância, **cloud-init** está habilitado. As imagens KVM Guest Images que você baixa do Portal do Cliente da Red Hat são destinadas ao uso com a Red Hat Virtualization (RHV) e a Plataforma Red Hat OpenStack (RHOSP). Você também pode criar uma imagem do zero para a RHV e a RHOSP.
- Outra opção é fazer o download de uma imagem ISO do Portal do Cliente da Red Hat ou criar uma. Neste caso, você precisa instalar **cloud-init** em sua imagem ISO.
- Se você pretende usar uma imagem com um fornecedor de nuvens (por exemplo, AWS ou Azure), use o Red Hat Image Builder para criar a imagem. As imagens do Image Builder são personalizadas para uso em provedores de nuvens específicos. Os tipos de imagem AMI, VHD, e qcow2 incluem **cloud-init** já instalados. Consulte [Composição de uma imagem personalizada do sistema RHEL](#) para obter informações sobre o Image Builder.

A maioria das plataformas de nuvem suportam **cloud-init**, embora os procedimentos de configuração e as opções suportadas variem. Alternativamente, você pode configurar **cloud-init** para um ambiente NoCloud.

Você pode configurar **cloud-init** em uma máquina virtual (VM) e depois usar essa VM como modelo para VMs adicionais ou clusters de VMs.

Produtos específicos da Red Hat (por exemplo, a [Virtualização da Red Hat](#)) têm procedimentos documentados para configurar **cloud-init** para uso com esses produtos.

Este documento se refere à documentação **cloud-init** em vários lugares. Consulte a documentação referida **cloud-init** para obter informações completas em **cloud-init**.

Pré-requisitos

- Cadastre-se para uma conta [no Portal do Cliente Red Hat](#).

Recursos adicionais

- [documentação nebulosa](#)

1.1. CONFIGURAÇÃO EM NUVENS

cloud-init usa instruções de arquivo em formato YAML para realizar tarefas. Você decide a configuração inicial que deseja que **cloud-init** realize, fornecendo instruções dentro dos arquivos YAML. Quando uma

instância inicializa, o serviço **cloud-init** inicia e procura e executa as instruções. Tarefas concluídas durante o primeiro boot ou nos boots subsequentes de seu VM, com base em sua configuração **cloud-init**.

Você define as tarefas, configurando o arquivo `/etc/cloud/cloud.cfg` e adicionando diretivas sob o diretório `/etc/cloud/cloud.cfg.d/`.

- O arquivo **cloud.cfg** inclui diretivas, tais como as de acesso e autenticação do usuário e informações do sistema.
O arquivo também inclui módulos padrão e opcionais para **cloud-init**. Os módulos são executados em ordem dentro de três fases que incluem a fase de inicialização **cloud-init**, a fase de configuração, e a fase final. Dentro do arquivo **cloud.cfg**, os módulos para as três fases estão listados em **cloud_init_modules**, **cloud_config_modules**, e **cloud_final_modules**, respectivamente.
- O diretório **cloud.cfg.d** é onde você pode adicionar diretivas adicionais para **cloud-init**. Quando você adiciona diretivas ao diretório **cloud.cfg.d**, normalmente as adiciona a um arquivo chamado `*.cfg`, e você sempre inclui **#cloud-config** no topo do arquivo.

1.2. A NUVEM OPERA EM ETAPAS

cloud-init opera em cinco estágios durante uma inicialização do sistema. Essas etapas determinam se **cloud-init** funciona e onde encontra suas fontes de dados, entre outras tarefas. Segue um breve resumo das etapas.

1. A etapa do gerador **cloud-init**, através do serviço **systemd**, determina se deve executar **cloud-init** no momento da inicialização.
2. Durante a etapa local, **cloud-init** encontra fontes de dados locais e aplica a configuração da rede.
3. Durante a etapa da rede, **cloud-init** processa os dados do usuário e executa os módulos listados em **cloud_init_modules** em seu arquivo **cloud.cfg**. Você pode ativar, desativar ou adicionar módulos à seção **cloud_init_modules**.
4. Durante a etapa de configuração, **cloud-init** executa os módulos listados em **cloud_config_modules** em seu arquivo **cloud.cfg**. Você pode ativar, desativar ou adicionar módulos à seção **cloud_config_modules**.
5. Durante a etapa final, **cloud-init** pode executar o que você incluiu em **cloud_final_modules** em seu arquivo **cloud.cfg**. Você pode incluir instalações de pacotes que você normalmente executaria após a inicialização de um sistema e também pode incluir plug-ins de gerenciamento de configuração e scripts do usuário. Você pode ativar, desativar ou adicionar módulos à seção **cloud_final_modules**.

As cinco etapas de inicialização estão descritas na seção de documentação **cloud-init** [Boot Stages \(Etapas de inicialização\)](#).

1.3. OS MÓDULOS DE NUVEM SÃO EXECUTADOS EM FASES

Quando **cloud-init** é executado, ele executa os módulos dentro de **cloud.cfg** em ordem em três fases que incluem a fase de rede (**cloud_init_modules**), a fase de configuração (**cloud_config_modules**), e a fase final (**cloud_final_modules**). Quando **cloud-init** roda pela primeira vez em uma VM, todos os módulos que você configurou rodam em suas respectivas fases. Em uma execução posterior de **cloud-**

init, se um módulo roda dentro de uma fase depende do *module frequency* do módulo individual. Alguns módulos rodam toda vez que **cloud-init** roda; alguns módulos só rodam na primeira vez que **cloud-init** roda, mesmo que a identificação da instância mude.



NOTA

Uma identificação de instância identifica de forma única uma instância. Quando um ID de instância muda, **cloud-init** trata a instância como uma nova instância.

Segue uma breve descrição dos valores *module frequency*.

- *Per instance* significa que o módulo funciona na primeira inicialização de uma instância. Por exemplo, se você clonar uma instância ou criar uma nova instância a partir de uma imagem salva, os módulos designados como por instância rodam novamente.
- *Per once* significa que o módulo funciona apenas uma vez. Por exemplo, se você clonar uma instância ou criar uma nova instância a partir de uma imagem salva, os módulos designados por uma vez não serão executados novamente nessas instâncias.
- *Per always* significa que o módulo funciona em todas as botas.



NOTA

Você pode substituir a frequência de um módulo ao configurar o módulo ou ao usar a linha de comando.

1.4. ATUA SOBRE OS DADOS DO USUÁRIO, METADADOS E DADOS DO FORNECEDOR

cloud-init consome e atua sobre dados de usuários, metadados e dados de fornecedores.

- Os dados do usuário incluem diretrizes que você especifica no arquivo **cloud.cfg** e no diretório **cloud.cfg.d**, por exemplo, os dados do usuário podem incluir arquivos para executar, pacotes para instalar e scripts shell. Consulte a seção de documentação **cloud-init** [Formatos de dados do usuário](#) para obter informações sobre os tipos de dados do usuário que o **cloud-init** permite.
- Os metadados incluem dados associados a uma fonte de dados específica, por exemplo, os metadados podem incluir um nome de servidor e uma identificação de instância. Se você estiver usando uma plataforma de nuvem específica, a plataforma determina onde suas instâncias encontram dados e metadados do usuário. Sua plataforma pode exigir que você adicione metadados e dados do usuário a um serviço HTTP; neste caso, quando **cloud-init** é executado, ele consome metadados e dados do usuário do serviço HTTP.
- Os dados do fornecedor são opcionalmente fornecidos pela organização (por exemplo, um provedor de nuvem) e incluem informações que podem personalizar a imagem para melhor se ajustar ao ambiente onde a imagem é executada. **cloud-init** atua sobre dados opcionais do fornecedor e dados do usuário após ler quaisquer metadados e inicializar o sistema. Por padrão, os dados do fornecedor são executados na primeira inicialização. Você pode desativar a execução dos dados do fornecedor.
Consulte a seção de documentação [Metadados de Instância cloud-init](#) para uma descrição dos metadados; [Fontes de dados](#) para uma lista de fontes de dados; e [Dados do fornecedor](#) para mais informações sobre os dados do fornecedor.

1.5. NUVEM - IDENTIFICA A PLATAFORMA DE NUVENS

cloud-init tenta identificar a plataforma de nuvens usando o script **ds-identify**. O script é executado na primeira inicialização de uma instância.

A adição de uma diretiva de fonte de dados pode economizar tempo quando **cloud-init** é executado. Você adicionaria a diretiva no arquivo **/etc/cloud/cloud.cfg** ou no diretório **/etc/cloud/cloud.cfg.d**. Segue uma amostra.

```
data_source_list:[Ec2]
```

Além de adicionar a diretiva para sua plataforma em nuvem, você pode configurar ainda mais **cloud-init** adicionando detalhes adicionais de configuração, tais como URLs de metadados.

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

Após **cloud-init** rodar, você pode visualizar um arquivo de log (**run/cloud-init/ds-identify.log**) que fornece informações detalhadas sobre a plataforma.

Recursos adicionais

- [Fontes de dados](#)
- [Que fonte de dados estou usando?](#)

CAPÍTULO 2. SUPORTE DE CHAPÉU VERMELHO PARA A NUVEM

Este capítulo cobre o suporte da Red Hat para **cloud-init**. Ele inclui informações sobre os produtos Red Hat que usam **cloud-init**, **cloud-init** módulos que a Red Hat suporta, e diretórios e arquivos padrão.

2.1. DIRETÓRIOS E ARQUIVOS IMPORTANTES NA NUVEM

A tabela a seguir inclui diretórios e arquivos importantes. Revise estes diretórios e arquivos; eles permitem que você execute tarefas como:

- Configurando **cloud-init**
- Encontrando informações sobre sua configuração após **cloud-init** ter rodado
- Examinando arquivos de log
- Encontrando modelos

Dependendo de seu cenário e fonte de dados, pode haver arquivos e diretórios adicionais importantes para sua configuração.

Tabela 2.1. diretórios e arquivos de nuvens

Diretório ou Arquivo	Descrição
/etc/cloud/cloud.cfg	O arquivo cloud.cfg inclui a configuração básica cloud-init e lhe permite saber em que fase cada módulo funciona.
/etc/cloud/cloud.cfg.d	O diretório cloud.cfg.d é onde você pode adicionar diretivas adicionais para cloud-init .
/var/lib/cloud	Quando cloud-init é executado, ele cria um layout de diretório sob /var/lib/cloud . O layout inclui diretórios e arquivos que dão informações específicas sobre a configuração de sua instância.
/usr/share/doc/cloud-init/examples	O diretório examples inclui vários exemplos. Você pode usá-los para ajudar a modelar suas próprias diretrizes.
/etc/cloud/templates	Este diretório inclui modelos que você pode habilitar em cloud-init para certos cenários. Os modelos fornecem a direção para habilitação.
/var/log/cloud-init.log	O arquivo cloud-init.log fornece informações de registro úteis para a depuração.
/run/cloud-init	O diretório /run/cloud-init inclui informações logadas sobre sua fonte de dados e o script ds-identify .

2.2. PRODUTOS RED HAT QUE USAM O SISTEMA DE NUVENS

Você pode usar **cloud-init** com os seguintes produtos Red Hat.

- **Red Hat Virtualization.** Uma vez instalado **cloud-init** em uma VM, você pode criar um modelo e aproveitar as funções **cloud-init** para todas as VMs criadas a partir desse modelo. Consulte o tópico [Usando o Cloud-Init para automatizar a configuração de máquinas virtuais](#) para obter informações sobre como usar **cloud-init** com as VMs.
- **Red Hat OpenStack Platform.** Você pode usar **cloud-init** para ajudar a configurar imagens para o OpenStack. Consulte o [Guia de Instâncias e Imagens](#) para obter mais informações.
- **Red Hat CloudForms.** Você pode usar **cloud-init** para fornecer VMs para Red Hat CloudForms. Consulte os [Modelos de Personalização para Máquina Virtual e Provisionamento de Instância](#) para mais informações.
- **Red Hat Satellite.** Você pode usar **cloud-init** com o Red Hat Satellite. Consulte [Preparando Imagens de Nuvem na Virtualização da Red Hat](#) para mais informações.
- **Red Hat OpenShift.** Você pode usar **cloud-init** quando você criar VMs para OpenShift. Consulte [Criar máquinas virtuais](#) para obter mais informações.

2.3. A RED HAT SUPORTA ESTES MÓDULOS COM NUVENS

A Red Hat suporta a maioria dos módulos **cloud-init**. Os módulos individuais podem conter múltiplas opções de configuração. A tabela a seguir lista todos os módulos **cloud-init** que a Red Hat suporta atualmente e fornece uma breve descrição e a frequência do módulo padrão. Consulte a seção [Módulos](#) na seção de Documentação de Nuvem para obter descrições completas e opções para estes módulos.

Tabela 2.2. Módulos de nuvem suportados

módulo Nuvem-Init	Descrição	Frequência do Módulo Padrão
bootcmd	Executa comandos no início do processo de inicialização	por sempre
ca_certs	Acrescenta certificados CA	por instância
debug	Permite ou desativa a saída de informações internas para auxiliar na depuração	por instância
disable_ec2_metadata	Habilita ou desabilita os metadados do AWS EC2	por sempre
disk_setup	Configura mesas divisórias e sistemas de arquivos simples	por instância
última_mensagem	Especifica a mensagem de saída uma vez concluída cloud-init	por sempre
foo	Exemplo mostra a estrutura do módulo (Módulo não faz nada)	por instância

módulo Nuvem-Init	Descrição	Frequência do Módulo Padrão
growpart	Redimensiona as partições para preencher o espaço disponível em disco	por sempre
chaves_para_console	Permite o controle de impressões digitais e chaves que podem ser escritas no console	por instância
paisagem	Instala e configura um cliente paisagístico	por instância
locale	Configura o locale do sistema e o aplica em todo o sistema	por instância
mcollective	Instala, configura e inicia mcollective	por instância
migrante	Move as versões antigas de cloud-init para versões mais recentes	por sempre
monta	Configura pontos de montagem e arquivos swap	por instância
phone_home	Coloca os dados em um host remoto após a conclusão da inicialização	por instância
power_state_change	Completa o desligamento e reinicialização após todos os módulos de configuração terem sido executados	por instância
fantoche	Instala e configura o fantoche	por instância
resizefs	Redimensiona um sistema de arquivo para utilizar todo o espaço disponível em uma partição	por sempre
resolve_conf	Configura resolv.conf	por instância
rh_subscrição	Registra um sistema Red Hat Enterprise Linux	por instância

módulo Nuvem-Init	Descrição	Frequência do Módulo Padrão
rightscale_userdata	Adiciona suporte para ganchos de configuração RightScale para cloud-init	por instância
rsyslog	Configura o registro remoto do sistema usando rsyslog	por instância
runcmd	Executa comandos arbitrários	por instância
minio_de_sal	Instala, configura e inicia salt minion	por instância
scripts_per_boot	Roda por roteiros de inicialização	por sempre
scripts_per_instance	Roda por roteiro de instância	por instância
scripts_per_once	Executa scripts uma vez	por uma vez
scripts_utilizador	Executa scripts de usuário	por instância
scripts_vendedor	Executa scripts de fornecedores	por instância
o domínio_semente	Fornece dados aleatórios de sementes	por instância
set_hostname	Define o nome do host e o nome de domínio totalmente qualificado (FQDN)	por sempre
set_passwords	Define senhas de usuários e ativa ou desativa a autenticação de senhas SSH	por instância
ssh_authkey_fingerprints	Logs impressões digitais de chaves SSH do usuário	por instância
ssh_import_id	Importação de chaves SSH	por instância
ssh	Configura o SSH, e hospeda e autoriza chaves SSH	por instância
fuso horário	Define o fuso horário do sistema	por instância
update_etc_hosts	Atualizações /etc/hosts	por sempre
update_hostname	Atualiza o nome do host e FQDN	por sempre

módulo Nuvem-Init	Descrição	Frequência do Módulo Padrão
grupos_de_utilizadores	Configura usuários e grupos	por instância
write_files	Escreve arquivos arbitrários	por instância
yum_add_repo	Adiciona a configuração do repositório yum ao sistema	por sempre

A tabela a seguir lista os módulos que a Red Hat não suporta atualmente.

Tabela 2.3. Módulos não suportados

Módulo
apt_configure
apt_pipeline
byobu
chef
emitir_upstart
grub_dpkg
ubuntu_init_switch

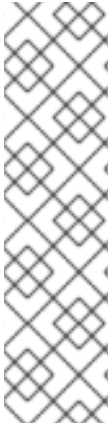
2.4. O ARQUIVO DEFAULT CLOUD.CFG

O arquivo `/etc/cloud/cloud.cfg` lista os módulos que compõem a configuração básica para **cloud-init**.

Os módulos no arquivo são os módulos padrão para **cloud-init**. Você pode configurar os módulos para seu ambiente ou remover os módulos que você não precisa. Os módulos que estão incluídos em **cloud.cfg** não fazem necessariamente nada ao serem listados no arquivo. Você precisa configurá-los individualmente se quiser que eles realizem ações durante uma das fases do **cloud-init**.

O arquivo **cloud.cfg** fornece a cronologia de execução dos módulos individuais. Você pode adicionar módulos adicionais a **cloud.cfg** desde que a Red Hat suporte os módulos que você deseja adicionar.

O que se segue é o conteúdo padrão do arquivo para o Red Hat Enterprise Linux (RHEL).



NOTA

- Os módulos são executados na ordem indicada em **cloud.cfg**. Normalmente não se altera esta ordem.
- As diretrizes **cloud.cfg** podem ser anuladas pelos dados do usuário.
- Ao executar **cloud-init** manualmente, você pode substituir **cloud.cfg** com opções de linha de comando.
- Cada módulo inclui suas próprias opções de configuração, onde você pode adicionar informações específicas.

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: ~ **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
- yum-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 Especifica o usuário padrão para o sistema. Consulte [Usuários e Grupos](#) para mais informações.
- 2 Habilita ou desabilita o login de raiz. Consulte [Chaves Autorizadas](#) para mais informações.
- 3 Especifica se **ssh** está configurado para aceitar a autenticação de senha. Consulte [Definir senhas](#) para obter mais informações.
- 4 Configura os pontos de montagem; deve ser uma lista contendo seis valores. Consulte [Montagens](#) para obter mais informações.
- 5 Especifica se é necessário remover as chaves SSH padrão do host. Consulte [Host Keys](#) para obter mais informações.
- 6 Especifica os tipos chave a gerar. Consulte [Host Keys](#) para obter mais informações.
- 7 **cloud-init** funciona em vários estágios de inicialização. Defina esta opção para que **cloud-init** possa registrar todas as etapas em seu arquivo de log. Encontre mais informações sobre esta opção no arquivo **cloud-config.txt** no diretório **usr/share/doc/cloud-init/examples**.
- 8 Habilita ou desabilita a personalização do VMware vSphere
- 9 Os módulos desta seção são serviços que funcionam quando o serviço **cloud-init** é iniciado, no início do processo de inicialização.
- 10 Estes módulos são executados durante a configuração **cloud-init**, após o boot inicial.
- 11 Estes módulos funcionam na fase final de **cloud-init**, após o término da configuração.

- 12 Especifica detalhes sobre o usuário padrão. Consulte [Usuários e Grupos](#) para obter mais informações.
- 13 Especifica a distribuição
- 14 Especifica o diretório principal que contém **cloud-init**-specific subdiretórios. Consulte o [layout do diretório](#) para obter mais informações.
- 15 Especifica onde residem os gabaritos
- 16 O nome do serviço SSH

Recursos adicionais

- [Onde estão os arquivos de configuração?](#)
- [Módulos](#)

2.5. O DIRETÓRIO CLOUD.CFG.D

cloud-init age de acordo com as diretrizes que você fornece e configura. Tipicamente, essas diretrizes estão incluídas no diretório **cloud.cfg.d**.



NOTA

Enquanto você pode configurar módulos adicionando diretivas de dados do usuário dentro do arquivo **cloud.cfg**, como melhor prática considere deixar **cloud.cfg** não modificado. Acrescente suas diretrizes ao diretório **/etc/cloud/cloud.cfg.d**. Adicionar diretivas a este diretório pode facilitar futuras modificações e atualizações.

Há várias maneiras de acrescentar diretrizes. Você pode incluir diretivas em um arquivo chamado ***.cfg**, que inclui o título **#cloud-config**. Normalmente, o diretório conteria vários arquivos ***.cfg**. Há outras opções para adicionar diretivas, por exemplo, você pode adicionar um script de dados do usuário. Consulte os [formatos de dados do usuário](#) para obter mais informações.

Recursos adicionais

- [Onde estão os arquivos de configuração?](#)
- [Exemplos de configuração em nuvem](#)

2.6. O ARQUIVO PADRÃO 05_LOGGING.CFG

O arquivo **05_logging.cfg** define as informações de registro para **cloud-init**. O diretório **/etc/cloud/cloud.cfg.d** inclui este arquivo, juntamente com outras diretrizes **cloud-init** que você adiciona.

cloud-init utiliza por padrão a configuração de registro em **05_logging.cfg**. O que segue é o conteúdo default do arquivo para o Red Hat Enterprise Linux (RHEL).

```
## 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'}
```

Recursos adicionais

- [Logging](#)

2.7. O LAYOUT DO DIRETÓRIO DE NUVENS /VAR/LIB/CLOUD

Quando **cloud-init** é executado pela primeira vez, ele cria um layout de diretório que inclui informações sobre sua instância e a configuração de **cloud-init**.

O que se segue é o modelo de layout do diretório para **cloud-init**.

O diretório pode incluir diretórios opcionais, tais como **/scripts/vendor**.

```
/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

Recursos adicionais

- [Layout do diretório](#)

CAPÍTULO 3. CONFIGURANDO A NUVEM-EM-IT

Este capítulo inclui exemplos das tarefas de configuração mais comuns para **cloud-init**.

Sua configuração **cloud-init** pode exigir que você adicione diretrizes ao arquivo **cloud.cfg** e ao diretório **cloud.cfg.d**. Alternativamente, sua fonte de dados específica pode exigir que você adicione diretivas aos arquivos, como um arquivo de dados do usuário e um arquivo de metadados. Uma fonte de dados pode exigir que você carregue suas diretivas em um servidor HTTP. Verifique os requisitos de sua fonte de dados e adicione diretivas de acordo.

3.1. CRIAÇÃO DE UMA MÁQUINA VIRTUAL QUE INCLUI UMA FONTE DE DADOS NOCLOUD

O que se segue é um exemplo de procedimento para a criação de uma nova VM que inclui **cloud-init**. Neste procedimento, você cria um arquivo **meta-data** e **user-data**. Seu arquivo **meta-data** inclui detalhes de instância. Seu arquivo **user-data** inclui informações para criar um usuário e conceder acesso. Você então inclui esses arquivos em uma nova imagem ISO, e anexa o arquivo ISO a uma nova VM criada a partir de uma imagem KVM Guest Image. Neste cenário, a fonte de dados é NoCloud.

Procedimento

1. Crie um diretório com o nome **cloudinitiso** e vá para ele.

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

2. Crie um arquivo chamado **meta-data**. Adicione as seguintes informações ao arquivo.

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

3. Crie um arquivo chamado **user-data**. Inclua as seguintes informações no arquivo.

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



NOTA

A linha final do arquivo **user-data** faz referência a uma chave pública SSH. Encontre suas chaves públicas SSH em **~/.ssh/id_rsa.pub**. Ao tentar este procedimento de amostra, modifique a linha para incluir uma de suas chaves públicas.

4. Use o comando **genisoimage** para criar uma imagem ISO que inclua **user-data** e **meta-data**.

```
# 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)
```

- Baixe uma imagem do KVM Guest Image do Portal do Cliente da Red Hat para o diretório **/var/lib/libvirt/images**.
- Criar uma nova VM da KVM Guest Image usando o comando **virt-install**. Inclua a imagem ISO que você criou como anexo à imagem.

```
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
```

- Acesse sua imagem como **cloud-user**. Sua senha é **cilogon**.

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

Passos de verificação

- Verifique o status **cloud-init** para ver se completou suas tarefas.

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

- cloud-init** cria o layout do diretório **cloud-init** sob **/var/lib/cloud** quando ele é executado, e atualiza ou muda certos conteúdos de diretório com base nas diretrizes que você especificou. Por exemplo, você pode confirmar que a fonte de dados é **NoCloud**, verificando o arquivo de fonte de dados.

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

cloud-init copia os dados do usuário para **/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
```

Estas são amostras. O layout do diretório **cloud-init** inclui muito mais informações.



NOTA

Para o OpenStack, o [Guia de Instâncias e Imagens](#) inclui informações para a configuração de uma instância usando **cloud-init**. Veja [Criando uma instância personalizada](#) para procedimentos específicos.

Recursos adicionais

- [NoCloud](#)

3.2. EXPIRANDO UMA SENHA DE USUÁRIO NA NUVEM COM A NUVEM

Você pode forçar **cloud-user** a mudar a senha **cloud-user** no primeiro login. Execute o seguinte procedimento para expirar uma senha.

Procedimento

1. Dependendo das exigências de sua fonte de dados, abra seu arquivo de dados do usuário para edição, ou adicione a seguinte diretiva ao diretório **cloud.cfg.d**.



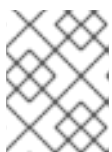
NOTA

Todas as diretrizes de usuário incluem **#cloud-config** no topo do arquivo para que **cloud-init** reconheça o arquivo como contendo as diretrizes de usuário. Quando você incluir diretivas no diretório **cloud.cfg.d**, nomeie o arquivo ***.cfg**, e inclua sempre **#cloud-config** no topo do arquivo.

2. Mude a linha **chpasswd: {expire: False}** para **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
```

Isto funciona para expirar a senha porque **password** e **chpasswd** operam no usuário padrão, a menos que você indique o contrário.



NOTA

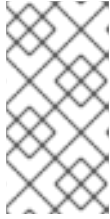
Este é um cenário global. Quando você define **chpasswd** para **True**, todos os usuários que você cria precisam mudar suas senhas quando fazem o login.

3.3. ALTERANDO UM NOME DE USUÁRIO PADRÃO COM A NUVEM

Você pode mudar o nome de usuário padrão para algo que não seja **cloud-user**.

Procedimento

1. Dependendo das exigências de sua fonte de dados, abra seu arquivo de dados do usuário para edição, ou adicione a seguinte diretiva ao diretório **cloud.cfg.d**.



NOTA

Todas as diretrizes de usuário incluem **#cloud-config** no topo do arquivo para que **cloud-init** reconheça o arquivo como contendo as diretrizes de usuário. Quando você incluir diretivas no diretório **cloud.cfg.d**, nomeie o arquivo ***.cfg**, e inclua sempre **#cloud-config** no topo do arquivo.

2. Adicione a linha **user: <username>**, substituindo <username> pelo novo nome de usuário padrão.

```
#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. ESTABELECENDO UMA SENHA DE RAIZ COM A NUVEM

Para definir a senha raiz, crie uma lista de usuários.

Procedimento

1. Dependendo das exigências de sua fonte de dados, abra seu arquivo de dados do usuário para edição, ou adicione a seguinte diretiva ao diretório **cloud.cfg.d**.



NOTA

Todas as diretrizes de usuário incluem **#cloud-config** no topo do arquivo para que **cloud-init** reconheça o arquivo como contendo as diretrizes de usuário. Quando você incluir diretivas no diretório **cloud.cfg.d**, nomeie o arquivo ***.cfg**, e inclua sempre **#cloud-config** no topo do arquivo.

2. Crie uma lista de usuários na seção **chpasswd** do arquivo. O formato é mostrado no exemplo a seguir.



NOTA

O espaço branco é significativo. Não inclua o espaço branco antes ou depois dos dois pontos em sua lista de usuários. Se você incluir espaço em branco, a senha é definida com um espaço dentro.

```
#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
```



NOTA

Se você usar este método para definir a senha do usuário, você deve definir **all passwords** nesta seção.

3.5. GERENCIANDO ASSINATURAS DA RED HAT COM O CLOUD-INIT

Você pode usar a diretiva **rh_subscription** para registrar seu sistema. Amostras a seguir. Para cada assinatura, você editaria seus dados de usuário.

Procedimento

O exemplo a seguir utiliza as opções **auto-attach** e **service-level**.

- Em **rh_subscription**, adicione seu **username** e **password**, defina **auto-attach** para **True**, e defina **service-level** para **self-support**.

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



NOTA

A opção **service-level** requer que você utilize a opção **auto-attach**.

O exemplo a seguir utiliza as opções **activation-key** e **org**.

- Em **rh_subscription**, adicione seu número **activation key** e **org** e defina **auto-attach** para **True**.

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

O exemplo a seguir acrescenta um pool de assinaturas.

- Em **rh_subscription**, adicione seu **username**, **password**, e o número do pool.

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

```
add-pool: XYZ01234567
```



NOTA

Esta amostra é o equivalente do comando **subscription-manager attach --pool=XYZ01234567**.

O exemplo a seguir estabelece o nome de um servidor hospedeiro no arquivo **/etc/rhsm/rhsm.conf**.

- Em **rh_subscription**, adicione seu **username**, **password**, **server-hostname**, e defina **auto-attach** para **True**.

```
rh_subscription:
username: sample@redhat.com
password: 'password'
server-hostname: test.example.com
auto-attach: True
```

3.6. ADICIONANDO USUÁRIOS E OPÇÕES DE USUÁRIO COM A NUVEM

Você cria e descreve os usuários em uma seção **users**. Você pode modificar a seção para adicionar mais usuários à configuração inicial de seu sistema, e você pode definir opções adicionais de usuário.

Se você adicionar a seção **users**, você também deverá definir as opções padrão do usuário nesta seção.

Procedimento

1. Dependendo das exigências de sua fonte de dados, abra seu arquivo de dados do usuário para edição, ou adicione a seguinte diretiva ao diretório **cloud.cfg.d**.



NOTA

Todas as diretrizes de usuário incluem **#cloud-config** no topo do arquivo para que **cloud-init** reconheça o arquivo como contendo as diretrizes de usuário. Quando você incluir diretivas no diretório **cloud.cfg.d**, nomeie o arquivo ***.cfg**, e inclua sempre **#cloud-config** no topo do arquivo.

2. Adicionar ou modificar a seção **users** para adicionar usuários.
 - Se você quiser que **cloud-user** seja o usuário padrão criado junto com os outros usuários especificados, certifique-se de adicionar **default** como a primeira entrada na seção. Se não for a primeira entrada, **cloud-user** não será criado.
 - Por padrão, os usuários são rotulados como **unconfined_u** se não houver um valor **selinux-user**.

```
#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
```



NOTA

- O exemplo coloca o usuário **user2** em dois grupos, **users** e **wheel**.
- A partir de **cloud-init 0.7.5**, a lista de grupos não suporta o espaço branco. Consulte a [BZ 1126365](#) para mais informações.

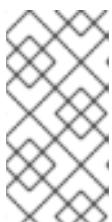
3.7. EXECUTANDO OS PRIMEIROS COMANDOS DE INICIALIZAÇÃO COM A NUVEM

Você pode usar as seções **runcmd** e **bootcmd** para executar comandos durante a inicialização e inicialização.

A seção **bootcmd** é executada no início do processo de inicialização e, por padrão, é executada em cada inicialização. A seção **runcmd** é executada perto do final do processo e só é executada durante a primeira inicialização e inicialização.

Procedimento

1. Dependendo das exigências de sua fonte de dados, abra seu arquivo de dados do usuário para edição, ou adicione a seguinte diretiva ao diretório **cloud.cfg.d**.



NOTA

Todas as diretrizes de usuário incluem **#cloud-config** no topo do arquivo para que **cloud-init** reconheça o arquivo como contendo as diretrizes de usuário. Quando você incluir diretivas no diretório **cloud.cfg.d**, nomeie o arquivo ***.cfg**, e inclua sempre **#cloud-config** no topo do arquivo.

2. Adicione as seções para **bootcmd** e **runcmd**; inclua comandos que você deseja que **cloud-init** execute.

```
#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. ACRESCENTAR PROCESSADORES ADICIONAIS COM NUVENS

Você pode configurar um usuário como um sudoer, adicionando uma entrada em **sudo** e **groups** à seção **users**.

Procedimento

1. Dependendo das exigências de sua fonte de dados, abra seu arquivo de dados do usuário para edição, ou adicione a seguinte diretiva ao diretório **cloud.cfg.d**.



NOTA

Todas as diretrizes de usuário incluem **#cloud-config** no topo do arquivo para que **cloud-init** reconheça o arquivo como contendo as diretrizes de usuário. Quando você incluir diretivas no diretório **cloud.cfg.d**, nomeie o arquivo ***.cfg**, e inclua sempre **#cloud-config** no topo do arquivo.

2. Adicione uma entrada em **sudo** e especifique o acesso do usuário. Por exemplo, **sudo: ALL=(ALL) NOPASSWD:ALL** permite o acesso irrestrito de um usuário.
3. Adicione uma entrada em **groups** e especifique os grupos que incluem o usuário.

```

#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. CONFIGURANDO UMA CONFIGURAÇÃO DE REDE ESTÁTICA COM A NUVEM

Você pode configurar sua configuração de rede com **cloud-init**, adicionando uma seção **network.interfaces** aos seus metadados.

O Red Hat Enterprise Linux fornece seu serviço de rede default através do **NetworkManager**, que é um daemon dinâmico de controle e configuração de rede que mantém os dispositivos e conexões de rede

ativos e ativos quando disponíveis. Consulte o [Getting Started with NetworkManager](#) para obter mais informações em **NetworkManager**.

Sua fonte de dados pode fornecer uma configuração de rede. Consulte a seção de documentação **cloud-init Fontes de configuração de rede** para obter mais informações.

Se você não especificar nenhuma configuração de rede para **cloud-init** e não tiver desativado a configuração de rede, **cloud-init** tenta determinar se algum dispositivo anexo tem uma conexão. Se encontrar um dispositivo conectado, ele gera uma configuração de rede que emite um pedido DHCP na interface. Consulte a seção de documentação **cloud-init Fallback Network Configuration** para obter mais informações.

Procedimento

O exemplo a seguir acrescenta uma configuração de rede estática.

1. Dependendo das exigências de sua fonte de dados, abra seu arquivo de dados do usuário para edição, ou adicione a seguinte diretiva ao diretório **cloud.cfg.d**.



NOTA

Todas as diretrizes de usuário incluem **#cloud-config** no topo do arquivo para que **cloud-init** reconheça o arquivo como contendo as diretrizes de usuário. Quando você incluir diretivas no diretório **cloud.cfg.d**, nomeie o arquivo ***.cfg**, e inclua sempre **#cloud-config** no topo do arquivo.

2. Adicione uma seção **network-interfaces**.

```
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
```



NOTA

Você pode desativar uma configuração de rede adicionando as seguintes informações aos seus metadados.

```
network
  config: disabled
```

Recursos adicionais

- [Configuração de rede](#)
- [NoCloud](#)

3.10. CONFIGURANDO APENAS UM USUÁRIO RAIZ COM A NUVEM

Você pode configurar seus dados de usuário para que você tenha um usuário `root` e nenhum outro usuário.

Procedimento

1. Dependendo das exigências de sua fonte de dados, abra seu arquivo de dados do usuário para edição, ou adicione a seguinte diretiva ao diretório `cloud.cfg.d`.



NOTA

Todas as diretrizes de usuário incluem `#cloud-config` no topo do arquivo para que `cloud-init` reconheça o arquivo como contendo as diretrizes de usuário. Quando você incluir diretivas no diretório `cloud.cfg.d`, nomeie o arquivo `*.cfg`, e inclua sempre `#cloud-config` no topo do arquivo.

2. Criar uma entrada para o usuário `root` na seção `users`.
O exemplo simples que se segue inclui uma seção `users` com apenas a opção `name`.

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

3. Opcionalmente, configure chaves SSH para o usuário `root`.

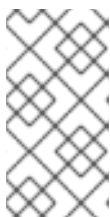
```
users:
  - name: root
    ssh_pwauth: True
    ssh_authorized_keys:
      - ssh-rsa AA..vz user@domain.com
```

3.11. INSTALAÇÃO DE ARMAZENAMENTO COM ARMAZENAMENTO DE CONTÊINERES NA NUVEM

Você pode configurar o armazenamento referenciando o utilitário `container-storage-setup` dentro do módulo `write_files`.

Procedimento

1. Dependendo das exigências de sua fonte de dados, abra seu arquivo de dados do usuário para edição, ou adicione a seguinte diretiva ao diretório `cloud.cfg.d`.



NOTA

Todas as diretrizes de usuário incluem `#cloud-config` no topo do arquivo para que `cloud-init` reconheça o arquivo como contendo as diretrizes de usuário. Quando você incluir diretivas no diretório `cloud.cfg.d`, nomeie o arquivo `*.cfg`, e inclua sempre `#cloud-config` no topo do arquivo.

- Adicionar ou modificar o módulo **write_files** para incluir o caminho para o utilitário **container-storage-setup**.

O exemplo a seguir define o tamanho do volume lógico raiz para 6GB, em vez dos 3GB padrão.

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



NOTA

Antes da RHEL 7.4, **container-storage-setup** foi chamado **docker-storage-setup**. Se você estiver usando OverlayFS para armazenamento, a partir da RHEL 7.4 você pode agora usar esse tipo de sistema de arquivo com SELinux em modo de aplicação.

3.12. MUDANDO O LOCAL DO SISTEMA COM A NUVEM

Você pode configurar a localização do sistema com o módulo **locale**.

Procedimento

- Dependendo dos requisitos de sua fonte de dados, abra seu arquivo de meta-dados para edição, ou adicione a seguinte diretiva ao arquivo **cloud.cfg** ou ao diretório **cloud.cfg.d**.
- Adicionar a diretiva **locale**, especificando a localização. A seguinte amostra estabelece o **locale** para **ja_JP** (Japão) com a codificação **UTF-8**.

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

Recursos adicionais

- [Locale](#)

3.13. ROTEIROS COM NUVENS E CONCHAS

Você pode adicionar valores de lista ou valores de string para **bootcmd** ou **runcmd**. Você também pode fornecer um script de shell dentro de seus dados de usuário.

- Se você usar um valor de lista para **bootcmd** ou **runcmd**, cada item da lista é executado por sua vez usando **execve**.
- Se você usar um valor de string, então a string inteira é executada como um script de shell.
- Se você quiser usar **cloud-init** para executar um script shell, você pode fornecer um script shell (completo com shebang (#!)) em vez de fornecer **cloud-init** com um arquivo **.yaml**.

Consulte os [comandos Executar na primeira inicialização](#) para obter exemplos de como colocar scripts shell em **bootcmd** e **runcmd**.

3.14. IMPEDINDO QUE A NUVEM-INIT ATUALIZE OS ARQUIVOS DE CONFIGURAÇÃO

Quando você cria ou restaura uma instância a partir de uma imagem de backup, a identificação da instância muda. A mudança no ID da instância pode fazer com que **cloud-init** atualize os arquivos de configuração.

Execute o seguinte procedimento para garantir que **cloud-init** não atualize certos arquivos de configuração quando você criar ou restaurar a partir de backup.

Procedimento

1. Abra o arquivo `/etc/cloud/cloud.cfg` para edição.
2. Comente ou remova a configuração que você não deseja que **cloud-init** atualize ao restaurar sua instância.
Por exemplo, para evitar a atualização do arquivo chave SSH, remova **-ssh** da seção **cloud_init_modules**.

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

Etapas de verificação

Você pode verificar quais arquivos de configuração **cloud-init** foi atualizado. Para fazer isso, examine o arquivo `/var/log/cloud/cloud-init.log`. Os arquivos atualizados são registrados durante a inicialização de uma instância com mensagens que começam com **Writing to**. Segue um exemplo.

```
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. MODIFICANDO UMA VM CRIADA A PARTIR DE UMA IMAGEM DE CONVIDADO DA KVM DEPOIS QUE A NUVEM ESTIVER EM FUNCIONAMENTO

Esta seção fornece um procedimento de amostra para quando você quiser modificar sua configuração **cloud-init** antes de executar novamente **cloud-init**. Quando você lança uma VM que inclui o pacote **cloud-init** instalado e habilitado, **cloud-init** é executado em seu estado padrão naquela inicialização inicial de sua VM.

Procedimento

1. Faça o login em seu VM.
2. Adicionar ou alterar diretivas, por exemplo, modificar o arquivo **cloud.cfg** no diretório **/etc/cloud** ou adicionar diretivas ao diretório **/etc/cloud/cloud.cfg.d**.
3. Execute o comando **cloud-init clean** para limpar os diretórios de forma que **cloud-init** possa ser executado novamente. Você também pode executar os seguintes comandos como root para limpar o VM.

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



NOTA

Você pode salvar a imagem limpa como uma nova imagem e usar essa imagem para várias VMs. As novas VMs rodam **cloud-init** usando sua configuração atualizada **cloud-init**.

4. Repetição **cloud-init** ou reinicialização da VM.
cloud-init, implementando as mudanças de configuração que você fez.

3.16. MODIFICAÇÃO DE UMA VM PARA UMA FONTE DE DADOS ESPECÍFICA DEPOIS QUE A NUVEM ESTIVER EM FUNCIONAMENTO

Esta seção fornece um procedimento de amostra para quando você quiser modificar sua configuração **cloud-init** antes de executar novamente **cloud-init**. O procedimento a seguir usa o OpenStack como exemplo. Observe que o procedimento varia de acordo com sua fonte de dados.

Procedimento

1. Criar e lançar uma instância para a Plataforma OpenStack. Consulte [Instâncias de Máquina Virtual](#) para obter informações sobre como criar instâncias para o OpenStack. Neste exemplo, nossa máquina virtual inclui **cloud-init**, que funciona na inicialização da máquina virtual.
2. Adicionar ou alterar diretrizes. Por exemplo, modificar o arquivo **user-data.file** que está armazenado no servidor HTTP OpenStack.
3. Limpar a máquina virtual. Execute os seguintes comandos como raiz.

```
\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`
```



NOTA

Você pode salvar a imagem limpa como uma nova imagem e usar essa imagem para várias máquinas virtuais. As novas máquinas virtuais rodam **cloud-init** usando sua configuração atualizada **cloud-init**.

4. Re-executar **cloud-init** ou reiniciar a máquina virtual.
A nuvem volta a funcionar, implementando as mudanças de configuração que você fez.

3.17. RESOLUÇÃO DE PROBLEMAS DE NUVENS

Você pode solucionar problemas em sua instância depois de **cloud-init** ter sido executado, examinando seus arquivos de configuração e de registro. Uma vez identificado o problema, você pode reexecutar **cloud-init** em sua instância.

Você pode rodar **cloud-init** a partir da linha de comando usando o comando **cloud-init**. Para visualizar a sintaxe do comando, juntamente com uma descrição dos argumentos e subcomandos opcionais, execute o comando **cloud-init --help**. A sintaxe básica é a seguinte.

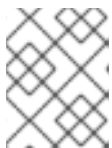
```
cloud-init [-h] [--version] [--file FILES] [--debug] [--force]
{init,modules,single,query,dhclient-hook,features,analyze,devel,collect-logs,clean,status}
```

O procedimento que se segue oferece idéias para identificar problemas com **cloud-init** e amostras para a execução do programa novamente.

Procedimento

1. Reveja os arquivos de configuração do **cloud-init**.
 - a. Examine o arquivo de configuração **/etc/cloud/cloud.cfg**. Verifique quais módulos estão incluídos em **cloud_init_modules**, **cloud_config_modules**, e **cloud_final_modules**.
 - b. Verifique as diretrizes (arquivos*.**cfg**) no diretório **/etc/cloud/cloud.cfg.d**.
2. Consulte os arquivos **/var/log/cloud-init.log** e **/var/log/cloud-init-output.log** para obter detalhes sobre uma questão específica. Por exemplo, se o problema foi que a partição raiz não foi automaticamente estendida, verifique as mensagens de registro para **growpart**. Se o sistema de arquivo não foi estendido, verifique as mensagens de log para **resizefs**. Segue um exemplo de comando **grep** para **resizefs**.

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



NOTA

growpart não suporta LVM. Se sua partição raiz for baseada em LVM, a partição raiz não é automaticamente estendida na primeira inicialização.

3. Rerun **cloud-init**. A seguir, exemplos de cenários. Executar comandos como raiz.
 - Repetição **cloud-init** apenas com os módulos **init**.


```
/usr/bin/cloud-init -d init
```
 - Reexecutar **cloud-init** com todos os módulos em sua configuração.


```
/usr/bin/cloud-init -d módulos
```
 - Eliminar o cache **cloud-init** e forçar **cloud-init** a funcionar após a inicialização.


```
rm -rf /var/lib/cloud/* && /usr/bin/cloud-init -d init
```
 - Execute os seguintes comandos para limpar diretórios e simular uma instância limpa.

-

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

- Execute os seguintes comandos para reexecutar **cloud-init**.

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

Recursos adicionais

- [Interface CLI](#)