



smartx
MAKE IT SIMPLE

cloud-init_虚拟机初始化配置 指南

发布日期: 2022-08-31

MAKE IT SIMPLE

目录

关于本文档.....	3
更新信息.....	4
第 1 章 概述.....	5
cloud-init 简介.....	5
cloud-init 与客户机操作系统的兼容性.....	5
第 2 章 使用 cloud-init.....	6
第 1 步：创建空白虚拟机并安装操作系统.....	6
第 2 步：在操作系统中安装 cloud-init.....	6
使用官方软件源安装（推荐）.....	6
使用源码包安装.....	8
第 3 步：修改 cloud-init 配置文件.....	9
第 4 步：将 cloud-init 虚拟机转化为虚拟机模板.....	12
第 5 步：使用 cloud-init 模板创建虚拟机.....	12
通过 Web 控制台/CloudTower 创建虚拟机.....	12
通过 CloudTower RESTful API/ SDK 创建虚拟机.....	13
第 6 步：确认 cloud-init 配置是否生效.....	13
第 3 章 后续操作注意事项.....	14
第 4 章 附录.....	15
用户数据示例.....	15
已知问题.....	15
Q&A.....	15
版权信息.....	17

关于本文档

本文档介绍了如何使用 cloud-init 为 Linux 虚拟机进行初始化配置。

更新信息

2022-08-31: 第一版正式发布

第 1 章

概述

在创建新虚拟机时，经常需对虚拟机进行初始化设置，如设置主机名、IP 地址、SSH 秘钥等。当批量创建虚拟机时，如对每个虚拟机手动进行初始化设置，则操作较为繁琐。在此场景下，您可以配置 cloud-init，使 Linux 虚拟机在首次开机时自动完成初始化配置，提高配置效率。

cloud-init 简介

cloud-init 是一个第三方工具，可以为 Linux 虚拟机自动完成一系列系统初始化配置。cloud-init 作为一个非常驻服务，仅在虚拟机开机启动时执行，执行完成后立即退出，不会监听任何端口。

ELF 通过使用配置了 cloud-init 的虚拟机模板创建虚拟机的方式，实现对 cloud-init 的支持。使用 cloud-init 模板创建的新虚拟机包含了初始化数据，新虚拟机在第一次启动时将自动完成初始化配置。

您可以使用 cloud-init 完成如下初始化配置。

- 主机名
- 默认用户的密码
默认用户为 root。
- SSH 公钥，用于 SSH 远程登录主机
- DNS
- IP 地址，支持 DHCP 和静态 IP 配置
- 自定义用户数据，可通过设置脚本实现更多自动化配置

如需了解 cloud-init 更多信息，请参考 [cloud-init 官方文档](#)。

cloud-init 与客户机操作系统的兼容性

cloud-init 与虚拟机操作系统的兼容性情况，请参考随 SMTX OS 发布的《SMTX 虚拟机服务兼容性指南》。

第 2 章

使用 cloud-init

使用 cloud-init 请参照以下步骤进行操作。

第 1 步：创建空白虚拟机并安装操作系统

创建一个空白虚拟机，并安装所需的 Linux 系统。具体操作方法，请根据 SMTX OS 版本，参考如下手册。

- SMTX OS 4.0.x 版本，请参考《SMTX OS Web 控制台用户操作手册》的“创建虚拟机”和“为虚拟机安装操作系统”章节。
- SMTX OS 5.x.x 和 6.x.x 版本，请参考《SMTX OS 管理指南》的“创建虚拟机”和“为虚拟机安装操作系统”章节。

第 2 步：在操作系统中安装 cloud-init

在 Linux 系统中可以采用以下两种方法安装 cloud-init，建议安装 cloud-init 18.2 及以上的版本

- 使用 Linux 系统的官方软件源安装（推荐）
- 使用源码包安装

使用官方软件源安装（推荐）

CentOS/RHEL

CentOS 和 RHEL 使用官方软件源安装 cloud-init 的安装方法相同。

其中，CentOS 自带的软件源中包含 cloud-init，无需添加外部软件源。RHEL 需添加软件源，且可使用 RHEL 官方软件源或 CentOS 软件源，添加方法请参考 [Adding a Yum Repository](#)。

以 CentOS 7.4 为例，安装 cloud-init 的方法如下，请以 root 帐户执行命令。

1. 执行以下命令安装 cloud-init:

```
yum install -y cloud-init
```

2. 执行以下命令设置 cloud-init 开机自启动:

```
systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service
```

执行以下命令确认 cloud-init 是否已经设置为开机自启动:

```
systemctl list-unit-files | grep cloud
```

若输出如下结果，表明 cloud-init 服务已经设置为开机自启动。

```
[root@localhost ~]# systemctl list-unit-files | grep cloud
cloud-config.service          enabled
cloud-final.service          enabled
cloud-init-local.service     enabled
cloud-init.service           enabled
cloud-config.target          static
```

OpenSUSE

以 OpenSUSE 12 SP3 为例，安装 cloud-init 的方法如下，请以 root 帐户执行命令。

1. 执行以下命令为 OpenSUSE 12 SP3 添加 zypper 软件源：

```
zypper addrepo -fc https://download.opensuse.org/repositories/Cloud:/Tools/
SLE_12_SP3/Cloud:Tools.repo
```

说明： 软件源链接建议使用 https。若 https 不可用，可使用 http。

2. 执行以下命令安装 cloud-init：

```
zypper --gpg-auto-import-keys install cloud-init
```

根据系统环境情况的不同，界面可能会提示需要安装 python-oauthlib，若出现此提示，请输入 1，按回车键后继续安装。

```
Problem: cloud-init-19.1-7.1.x86_64 requires python-oauthlib, but this requirement cannot be provided
uninstallable providers: python-oauthlib-3.0.2-2.11.noarch[Cloud_Tools]
Solution 1: Following actions will be done:
  install python-cryptography-2.7-2.12.x86_64 (with vendor change)
  SUSE LLC <https://www.suse.com/> --> obs://build.opensuse.org/devel:languages:python
  install python-cffi-1.12.3-2.2.x86_64 (with vendor change)
  SUSE LLC <https://www.suse.com/> --> obs://build.opensuse.org/devel:languages:python
Solution 2: do not install cloud-init-19.1-7.1.x86_64
Solution 3: break cloud-init-19.1-7.1.x86_64 by ignoring some of its dependencies
Choose from above solutions by number or cancel [1/2/3/c] (c): 1
```

3. 执行以下命令将 cloud-init 服务设置为开机自启动：

```
systemctl enable cloud-init-local.service cloud-init.service cloud-
config.service cloud-final.service
```

注意：

zypper 软件源的 cloud-init 存在开机启动不生效的问题。设置开机自启动后，需手动修改 service 启动脚本，将 WantedBy=cloud-init.target 修改为 WantedBy=multi-user.target，以解决此问题。

4. 可选。若安装的 cloud-init 的版本为 cloud-init-19.1-7.1.x86_64，需执行以下命令解决该版本存在的一个问题：

```
sed -i "s/path = net.sys_dev_path(iface.name)/path = net.sys_dev_path(iface
\[\"name\\\"\\])/\" /usr/lib/python2.7/site-packages/cloudinit/net/sysconfig.py
```

Ubuntu/Debian

Ubuntu 和 Debian 使用官方软件源安装 cloud-init 的安装方法相同，且系统自带的软件源通常已提供 cloud-init 的安装包，无需额外添加软件源。

以 Ubuntu 18.04 LTS 为例，安装步骤如下，请以 root 帐户执行命令。

1. 执行以下命令安装 cloud-init:

```
apt-get install cloud-init
```

2. 编辑 /etc/network/interfaces 文件以支持 ENI 的网络配置，将其内容替换为:

```
source /etc/network/interfaces.d/*
```

3. 可选。若安装的 cloud-init 的版本为 cloud-init-19.2-36，需执行以下命令解决该版本存在的一个问题:

```
sed -i "s/for cfg_source in order/for cfg_source in available_cfgs.keys()/g" /
usr/lib/python3/dist-packages/cloudinit/stages.py
```

使用源码包安装

CentOS

以 CentOS 7.4 为例，cloud-init 的安装步骤如下所示。

1. 执行以下命令安装 pip 工具:

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
```

```
python get-pip.py
```

2. 执行以下命令下载源码包并安装 cloud-init，此处以安装 cloud-init 18.2 为例:

```
#curl -L -o cloud-init-18.2.tar.gz https://launchpad.net/cloud-init/trunk/18.2/+download/
cloud-init-18.2.tar.gz
#tar -zxf ./cloud-init-18.2.tar.gz
#(cd ./cloud-init-18.2 && pip install -r ./requirements.txt --ignore-installed && python
setup.py install)
```

3. 执行以下命令，将 cloud-init 设置为开机启动:

```
systemctl enable cloud-init-local.service cloud-init.service cloud-
config.service cloud-final.service
```

4. 执行以下命令，为 cloud-init 创建 syslog 用户:

```
useradd syslog
```

5. 执行以下命令，为 cloud-init 创建 adm 组:

```
groupadd adm
```

Ubuntu

以 Ubuntu 14.04 为例，介绍如何使用源码包为 Ubuntu 系统安装 cloud-init。

1. 执行以下命令，下载 cloud-init 源码包，此处以 cloud-init 18.5 为例。

```
cd /root/
```

```
wget https://launchpadlibrarian.net/401408239/cloud-init-18.5.tar.gz
```

2. 执行以下命令，解压 cloud-init 安装包并进入解压后的安装包目录：

```
tar -xzvf cloud-init-18.5.tar.gz
```

```
cd cloud-init-18.5
```

3. 执行以下命令，安装 Python-pip：

```
apt-get install python-pip -y
```

4. 执行以下命令，确认镜像环境的 Python 解释器版本：

```
python --version
```

- 若 Python 解释器版本大于 Python 2.6，请直接进入下一步，安装 cloud-init 依赖包。
- 若 Python 解释器为 Python 2.6 及更低版本，在安装 cloud-init 依赖包之前，需执行 `pip install 'requests<2.20.0'` 命令，使系统安装的依赖包低于 requests 2.20.0 版本。

5. 执行以下命令，安装 cloud-init 依赖包。

```
pip install -r requirements.txt --upgrade
```

6. 依次执行以下命令，安装 cloud-init：

```
python setup.py build
```

```
python setup.py install --init-system systemd
```

注意：

`--init-system` 为当前操作系统使用的自启动服务管理方式，可选参数有：`systemd`, `sysvinit`, `sysvinit_deb`, `sysvinit_freebsd`, `sysvinit_openrc`, `sysvinit_suse`, `upstart`，默认为空。请根据当前操作系统使用的自启动服务管理方式进行选择。若选择错误，cloud-init 服务无法开机自启动。本文以 `systemd` 为例。

第 3 步：修改 cloud-init 配置文件

完成 cloud-init 安装后，请参考以下说明更新 cloud-init 的配置文件。

操作步骤

1. 打开 cloud-init 配置文件 `/etc/cloud/cloud.cfg`，更新如下配置项：

- 启用 root 帐户并将其设置为 cloud-init 的默认帐户

```
default_user:
  name: root
  lock_passwd: false
```

```
disable_root: false
ssh_pwauth: true
chpasswd:
  expire: false
```

- 配置 cloud-init 的数据源为 config drive

```
datasource_list: [ConfigDrive, None]
datasource:
  ConfigDrive:
    dsmode: local
```

- 将 growpart 和 resizefs 两个项目注释掉，以禁用 cloud-init 自动扩展分区模块

```
# - growpart
# - resizefs
```

以下为基于 CentOS 7.4, cloud-init-18.5-3.el7.centos.x86_64 版本的配置文件示例：

```
users:
  - default

#修改 disable_root, ssh_pwauth, chpasswd, datasource_list, datasource ==>
disable_root: false
ssh_pwauth: true
chpasswd:
  expire: false

datasource_list: [ConfigDrive, None]
datasource:
  ConfigDrive:
    dsmode: local
#<==修改到此行结束

mount_default_fields: [~, ~, 'auto', 'defaults,nofail,x-systemd.requires=Cloud-init.service',
'0', '2']
resize_rootfs_tmp: /dev
ssh_deletekeys: 0
ssh_genkeytypes: ~
syslog_fix_perms: ~
disable_vmware_customization: false

cloud_init_modules:
  - disk_setup
  - migrator
  - bootcmd
  - write-files
#注释掉以下两个项目
# - growpart
# - resizefs
  - set_hostname
  - update_hostname
  - update_etc_hosts
  - rsyslog
  - users-groups
  - ssh

cloud_config_modules:
  - 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:
- 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:
#修改 default_user ==>
default_user:
  name: root
  lock_passwd: false
#<==修改在此行结束
distro: rhel
paths:
  cloud_dir: /var/lib/cloud
  templates_dir: /etc/cloud/templates
ssh_svcname: sshd

# vim:syntax=yaml

```

2. 打开 `/usr/lib/systemd/system/cloud-init-local.service` 配置文件，增加 `ExecStartPre=/bin/sleep 2`，以解决在部分虚拟机启动较快的场景下（例如集群开启 Boost 模式），cloud-init 服务可能在 config drive 完成挂载前启动，从而导致无法检测到 DataSource 的问题。

```

[Unit]
Description=Initial cloud-init job (pre-networking)
DefaultDependencies=no
Wants=network-pre.target
After=systemd-remount-fs.service
Requires=dbus.socket
After=dbus.socket
Before=NetworkManager.service network.service
Before=network-pre.target
Before=shutdown.target
Before=firewalld.target
Conflicts=shutdown.target
RequiresMountsFor=/var/lib/cloud
ConditionPathExists=!/etc/cloud/cloud-init.disabled
ConditionKernelCommandLine=!cloud-init=disabled

[Service]
Type=oneshot
ExecStartPre=/bin/mkdir -p /run/cloud-init
ExecStartPre=/sbin/restorecon /run/cloud-init
ExecStartPre=/usr/sbin/touch /run/cloud-init/enabled
ExecStartPre=/bin/sleep 2
ExecStart=/usr/bin/cloud-init init --local
ExecStart=/bin/touch /run/cloud-init/network-config-ready
RemainAfterExit=yes
TimeoutSec=0

# Output needs to appear in instance console output
StandardOutput=journal+console

[Install]
WantedBy=cloud-init.target
~

```

第 4 步：将 cloud-init 虚拟机转化为虚拟机模板

完成 cloud-init 安装配置后，请关闭虚拟机，并将虚拟机转化成虚拟机模板。具体操作方法，请根据 SMTX OS 版本，参考如下相关手册和章节。

- SMTX OS 4.0.x 版本，请参考《SMTX OS Web 控制台用户操作手册》的“创建虚拟机模板”章节。
- SMTX OS 5.x.x 和 6.x.x 版本，请参考《SMTX OS 管理指南》的“创建虚拟机模板”章节。

注意：

将已安装 cloud-init 的虚拟机转化成模板时，务必将已安装 cloud-init 选项标记为是，否则后续通过模板创建虚拟机时无法配置初始化数据。

第 5 步：使用 cloud-init 模板创建虚拟机

创建 cloud-init 虚拟机模板后，可使用此模板创建虚拟机。使用 cloud-init 模板除创建虚拟机时，除可配置一般的虚拟机参数，还可配置虚拟机初始化数据。

您可以通过 Web 控制台或 CloudTower 创建 cloud-init 虚拟机，也可以通过 RESTful API 和 CloudTower SDK 进行创建。本节仅描述使用 cloud-init 模板创建虚拟机时的初始化配置。使用虚拟机模板创建虚拟机的详细步骤以及选项描述，请根据 SMTX OS 版本，参考如下相关手册和章节：

- SMTX OS 4.0.x 版本，请参考《SMTX OS Web 控制台用户操作手册》的“创建虚拟机”章节。
- SMTX OS 5.x.x 和 6.x.x 版本，请参考《SMTX OS 管理指南》的“从模板创建虚拟机”章节。

通过 Web 控制台/CloudTower 创建虚拟机

通过 Web 控制台和 CloudTower 创建虚拟机的方法基本相同，此处以 CloudTower 为例。

操作步骤

1. 登录 CloudTower，在界面右上角单击**创建**，选择**从模板创建虚拟机**。
2. 进入创建界面，选择预先制作的 cloud-init 虚拟机模板，参考《SMTX OS 管理指南》，配置虚拟机的基本信息名称、计算资源、存储资源等。
3. 配置**网络设备**时，在 **Cloud-init 配置** 部分，选择虚拟机的 IP 配置方式。
 - **配置静态 IP 地址**：为虚拟机配置静态 IP 和静态路由。
 - **启用网卡 DHCP**：为虚拟机启用 DHCP 以获取动态 IP。

- **不作配置**：不使用 cloud-init 配置 IP 地址。
4. 在**初始化配置**部分，配置虚拟机初始化数据。
 - a. 填写客户机操作系统的主机名、默认账户的密码、SSH 公钥和 DNS 等。
 - b. 配置用户数据。您可以配置不超过 32 KB 的用户数据，配置示例请参考[用户数据示例](#)。
 5. 单击**创建**，完成虚拟机创建。

通过 CloudTower RESTful API/ SDK 创建虚拟机

您可以通过 CloudTower 提供的 RESTful API 和 SDK 使用模板创建虚拟机，并配置初始化数据。详情请参考 [RESTful API 描述文档](#) 的“从内容库模版创建虚拟机”章节和 [SDK 描述文档](#)。

第 6 步：确认 cloud-init 配置是否生效

使用模板创建虚拟机后，请将虚拟机开机，检查虚拟机的主机名、IP、SSH 公钥、DNS 等初始化配置是否生效。若初始化配置未生效，需参考如下步骤，检查并修改模板的 cloud-init 配置文件。

操作步骤

更新 cloud-init 配置文件需将模板重新转化成虚拟机后再进行操作。您可以通过 Web 控制台和 CloudTower 将模板转化成虚拟机。下述操作以 Web 控制台为例。

1. 登录 Web 控制台，将虚拟机模板转化成虚拟机并开机。
2. 参考[第 3 步：修改 cloud-init 配置文件](#)，更正 cloud-init 配置。
3. 删除 `/var/lib/cloud/` 和 `/etc/cloud/cloud.cfg.d/50-curtin-networking.cfg`，以重新初始化 cloud-init 的运行环境。
4. 重新将 cloud-init 虚拟机转化为模板，并使用模板再次创建虚拟机，检查虚拟机初始化配置是否生效。
5. 若仍存在问题，请重复以上步骤，直至 cloud-init 已正确配置。

第 3 章

后续操作注意事项

克隆 cloud-init 虚拟机

cloud-init 数据是虚拟机的私有数据，不会随虚拟机克隆而传播。因此，克隆安装有 cloud-init 的虚拟机时，克隆的虚拟机无法获得相同的初始化配置，建议在克隆前先从 Guest OS 中卸载 cloud-init。

卸载 cloud-init

卸载 cloud-init 依赖安装时使用的软件管理工具。例如，若通过 `yum install` 安装 cloud-init，则可通过 `yum remove` 卸载。

第 4 章

附录

用户数据示例

cloud-init 支持的用户数据可参考以下官方文档。

- [User-Data Formats](#)
- [Cloud Config Examples](#)
- [Modules](#)

本文以使用 User Data Script 和 Cloud-Config Runcmd 为例，介绍如何配置用户数据。其他复杂的应用请参考官方文档进行配置。

User Data Script

配置如下所示的 Shell Script，使虚拟机在第一次启动时执行命令 `echo "Hello World. The time is now $(date -R)!" >> /root/smartx.txt`。

```
#!/bin/sh
echo "Hello World. The time is now $(date -R)!" >> /root/smartx.txt
```

Cloud-Config Runcmd

配置如下所示的 Cloud-Config Runcmd，使虚拟机在第一次启动时执行命令 `curl https://www.smartx.com/ -o /root/smartx.html`。

```
#cloud-config
runcmd:
- [ curl, "https://www.smartx.com/", -o, /root/smartx.html ]
```

已知问题

cloud-init 配置文件路径 `/etc/cloud/` 下只能存在一份 `cloud.cfg` 配置文件，即使其他 `cloud.cfg` 配置文件在该路径下以 `cloud.cfg_bak` 的形式存在，也会导致系统开机后 `root` 用户无法使用。

Q&A

Q: Centos7.x 通过 cloud-init 配置 DNS 时，cloud-init 正常执行，为什么 DNS 没有生效?

A: 该问题可能是由于虚拟机模版 `/etc/resolv.conf` 文件已存在 3 个 DNS 配置导致，使用 `/etc/resolv.conf` 配置 DNS 的 Linux 发行版都会有此限制。cloud-init 基于 `resolv.conf` 实现 DNS 配置，且配置方式为增加 DNS，不会取代原有的 DNS。根据 [resolv.conf\(5\)](#) 的限制，

resolv.conf 中最多只能配置 3 个 DNS。因此，若虚拟机模版 `/etc/resolv.conf` 文件已配置了 3 个 DNS，通过 cloud-init 配置 DNS 将会失败，在 `/var/log/cloud-init.log` 中也会输出 `ignoring nameserver {dns_server}: adding would exceed the maximum of '3' name servers (see resolv.conf(5))` 日志。

Q: Ubuntu 18.04 及以上版本通过 cloud-init 配置 DNS 时，若未选择"通过 cloud-init 配置网络"，cloud-init 正常执行，为什么 DNS 没有生效？

A: Ubuntu 18.04 版本开始，引入 netplan 作为网络管理组件，netplan 会对每个网卡单独配置 DNS。ConfigDrive 数据格式可以适配大多数操作系统的网络组件，但对于 netplan 组件，在未选择**通过 cloud-init 配置网络**的情况下，ConfigDrive 提供的数据不满足 cloud-init netplan 模块的数据格式要求，因此 DNS Server 配置不生效。

版权信息

版权所有 © 2023 北京志凌海纳科技有限公司 (SmartX) 保留一切权利。

本文档和本文档包含的全部信息受中华人民共和国法律法规及国际公约对知识产权的管辖和保护。本文档和本文档包含的全部信息的版权由 SmartX 或其许可方所有。未经 SmartX 书面许可，不得以任何方式，包括但不限于电子、机械或光学的方式对本文档内容的部分或全部进行复制、修改、传播、经销、翻印、拆解、存储在检索系统中或其他任何形式的使用。本文档中出现的所有非 SmartX 的公司名称、产品名称和服务名称仅用于识别目的，可能是其各自所有者的注册商标、商标或服务标记，不应被视为未经其所有者书面许可而授予使用前述任何商标、标志的许可或权利，亦不代表该所有者对任何产品或服务信息的授权使用或背书。