# cloud-init（Linux）

## 介绍

cloudinit是专为云环境中虚拟机的初始化而开发的工具，它从各种数据源读取相关数据并据此对虚拟机进行配置。常见的数据源包括：云平台的metadata服务、ConfigDrive等，常见的配置包括：设定虚拟机的hostname、hosts文件、设定用户名密码、更新apt －get的本地缓存、调整文件系统的大小（注意不是调整分区的大小）等。

官方文档：<http://cloudinit.readthedocs.io/en/latest/>

## 功能

目前的功能有：

1. 创建新用户
2. 修改用户密码
3. 添加用户到指定用户组
4. 获取用户home目录
5. 设置主机名
6. 获取网卡信息
7. 获取DHCP信息
8. 设置NTP
9. 设置静态IP
10. 判断服务是否存在
11. 获取服务状态
12. 设置服务启动模式
13. 启动服务
14. 关闭服务
15. 获取默认网关
16. 获取CD驱动
17. 获取物理磁盘
18. 创建防火墙规则
19. 删除防火墙规则

…

## 安装

centos 6.4和ubuntu server 12.04的官方源中已经包含cloudinit，直接采用yum 或者 apt -get安装即可  
源码地址：<https://launchpad.net/cloud-init>

目前支持以下操作系统：

* Ubuntu
* Fedora
* Debian
* RHEL
* CentOS
* *and more...*

说明：window使用的是cloudbase-init， linux用的是cloud-init。

## 工作原理

cloud-init 是 linux 的一个工具，当系统启动时，cloud-init 可从 nova metadata 服务或者 config drive 中获取 metadata，完成包括但不限于下面的定制化工作：

设置 default locale

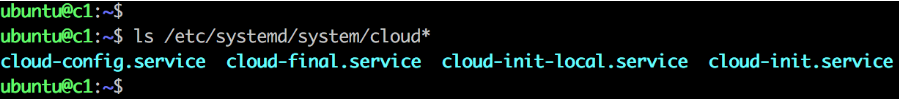
* 设置 hostname
* 添加 ssh keys到 .ssh/authorized\_keys
* 设置用户密码
* 配置网络
* 安装软件包

为了实现 instance 定制工作，cloud-init 会按 4 个阶段执行任务：

1. local
2. init
3. config
4. final

cloud-init 安装时会将这 4 个阶段执行的任务以服务的形式注册到系统中，比如在 systemd 的环境下，我们能够看到这4个阶段分别对应的服务：

1. local - cloud-init-local.service
2. init - cloud-init.service
3. config - cloud-config.service
4. final - cloud-final.service



### local阶段

作为 cloud-init 执行的第一个阶段，此时 instance 还不知道该如何配置网卡，cloud-init 的任务就是从 config drive 中获取配置信息，然后写入 /etc/network/interfaces 文件（如果是 centos 则写入 /etc/sysconfig/network-scripts/ifcfg-xxx）。

**如果没有 config drive，则将所有网卡配置成 dhcp 模式。**这是非常关键的一步，只有当网卡正确配置后，才能获取到 metadata。

### init、config 和 final 阶段

正常情况下，在这三个阶段执行之前 instance 网络已经配置好了，并且已经成功获取到 metadata。cloud-init 的配置文件 /etc/cloud/cloud.cfg 定义了三个阶段分别要执行的任务，任务以 module 形式指定。



instance 真正的定制工作就是由这些 module 完成的。module 决定做哪些定制化工作，而 metadata 则决定最终定制化的结果。

举个例子，如果 cloud.cfg 中指定了 set\_hostname 这个 module，则意味着 cloud-int 会设置 instance 的主机名，而具体设置成哪个主机名则由 metadata 中 hostname 参数决定。

有些 module 是有默认行为的，比如 growpart，如果 metadata 中没有特别指定，它会自动扩展 / 分区。

具体module设置可参考<https://cloudinit.readthedocs.io/en/latest/topics/modules.html>

## 配置方式

控制cloudinit配置行为方式主要有两种：

### cloud.cfg定义配置

可直接通过配置文件/etc/cloud/cloud.cfg加入配置，如红色部分定义的配置值

|  |
| --- |
| user: root  disable\_root: 0  manage\_etc\_hosts: True  preserve\_hostname: False  cloud\_init\_modules:  - bootcmd  - resizefs  - set\_hostname  - update\_hostname  - update\_etc\_hosts  - ca-certs  - rsyslog  - ssh  cloud\_config\_modules:  - mounts  - ssh-import-id  - locale  - set-passwords  - grub-dpkg  - landscape  - timezone  - puppet  - chef  - salt-minion  - mcollective  - disable-ec2-metadata  - runcmd  - byobu  cloud\_final\_modules:  - rightscale\_userdata  - scripts-per-once  - scripts-per-boot  - scripts-per-instance  - scripts-user  - keys-to-console  - phone-home  - final-message |

配置文件大致分为两部分，开头的变量/参数定义部分、后边要运行的模块列表（包括三大类cloud\_init\_modules、cloud\_config\_modules、cloud\_final\_modules）。

各模块在运行时，会根据之前定义的变量/参数的值，配置虚拟机的状态。

这里举一个简单的例子，update\_etc\_hosts模块

（源文件：/usr/lib/python2.7/site-packages/cloudinit/CloudConfig/cc\_update\_etc\_hosts.py）

顾名思义，该模块用来设置主机的hosts文件，其中就用到了hostname、fqdn、manage\_etc\_hosts等变量的值。模块首先尝试从cloudinit的配置文件中读取这些变量的值，如果没有定义，则尝试从其他的数据源中获取变量的值（见1.5.2描述），例如对于openstack来讲，可以从metadata service获取虚拟机的主机名。

### 通过数据源配置

通过数据源，如Openstack及EC2中的userdata。

userdata文件可以有多个格式，具体参见：

<http://cloudinit.readthedocs.org/en/latest/topics/format.html>

## 典型应用

### 设置 hostname

cloud-init 默认会将 instance 的名字设置为 hostname。但这样不太方便，有时希望能够将二者分开，可利用 cloud-init 的set\_hostname 模块实现。set\_hostname 它会查询 metadata 中 hostname 信息，默认值就是 instance 的名字。我们可以指定自己的 hostname，方法是将下面的内容传给 cloud-init：

|  |
| --- |
| #cloud-config  hostname: my1.cloudman.cc  manage\_etc\_hosts: true |

说明：

1. cloud-init 只会读取以 #cloud-config 开头的数据，所以这一行一定要写对。
2. hostname: my1.cloudman.cc 告诉 cloud-init 将 hostname 设置为 my1.cloudman.cc。
3. manage\_etc\_hosts: true 告诉 cloud-init 更新 /etc/hosts 文件。

### 定制用户初始密码

官方的 cloud image 默认只能通过 ssh key 登录。我们可以利用set-passwords 模块为用户设置密码并启用密码登录。需要传入的脚本如下：

|  |
| --- |
| #cloud-config  chpasswd:  list: |  root:123456  ubuntu:123456  expire: false  ssh\_pwauth: true |

说明：

1. root 和 ubuntu 用户密码设置为 123456。
2. ssh\_pwauth 启用密码登录。

### 安装软件

标准镜像中不可能包含我们需要的所有软件，定制安装是不可避免的。一个办法是部署完后手动安装，另一个办法是通过 package-update-upgrade-install 模块让 cloud-init 自动为我们安装。

需要传入的脚本如下：

|  |
| --- |
| #cloud-config  apt:  primary:  - arches: [default]  search:  - http://1.2.3.4  packages:  - pwgen  - pastebinit  - [libpython2.7, 2.7.3-0ubuntu3.1] |

说明：

1. apt 指定安装源的位置，这里为 http://1.2.3.4 。如果是 yum 源则用 yum\_repos 模块指定，具体用法可参看官网文档。
2. packages 指定需要安装的软件包，还可以指定具体版本。

# cloudbase-init（Window）

## 介绍

cloudbase-init 是 Windows 和其他系统的云初始化程序，可以设置主机名、创建用户、设置静态ip、设置密码等。对应的linux初始化程序是cloudinit，都是开源的Python项目。

cloudbase-init主要包括两部分services和plugins。Service主要为plugins提供数据来源，来源包括指定的云服务（openstack、ec2等）、本地配置文件（iso文件、物理磁盘）等。plugins即为执行相关操作的插件，像初始化ip、创建用户等。

具体可查看官方文档：

[http://cloudbase-init.readthedocs.io/en/latest/services.html#openstack-configuration-drive](http://cloudbase-init.readthedocs.io/en/latest/services.html%23openstack-configuration-drive)

## 功能

目前的功能有：

* 1. 网络设置
  2. 初始化主机名
  3. 初始化用户
  4. 自动扩展磁盘
  5. 设置密码

## 使用方法

首先, 下载你所需要的安装包, 之后安装并在配置文件中填入最适合你的选项. 基于当前 cloudbase-init 安装包的安装方式, 他将会在 C:\Program Files or C:\Program Files (x86) as Cloudbase Solutions\Cloudbase-Init目录中. 其中目录结构如下所示:

bin - 可执行文件和其他二进制文件.

conf - 配置文件

log - cloudbase-init相关日志.

LocalScripts - 用户提供的脚本.

安装完成之后, cloudbase-init 作为一个使用服务和通过插件去执行的两个步骤的服务, 通过这种方式去支持所有的配置. 根据平台不同的插件可能需要重启.

* 系统准备

系统准备（Sysprep）工具准备安装Windows以进行复制，审核和客户交付。 复制（也称为镜像）使您能够捕获可在整个组织中重复使用的自定义Windows映像。 Sysprep阶段使用“Unattend.xml”，这意味着使用“cloudbase-init-unattend.conf”配置文件运行服务。

* 配置文件

在已选择的安装路径下的conf目录存在两个配置文件分别叫： cloudbase-init.conf 和 cloudbase-init-unattend.conf. 可以有很多各种各样的参数用于配置你想要运行的服务和插件，以及定制用户配置。

* 模版配置文件解析

|  |
| --- |
| [DEFAULT]  # 创建什么用户以及用户属于什么组.  username=Admin  groups=Administrators  inject\_user\_password=true # 从元数据中获取密码 (非随机).  # 那个设备是可能的配置驱动 (元数据).  config\_drive\_raw\_hhd=true  config\_drive\_cdrom=true  # 与ubuntu中功能相同的tar命令默认路径.  bsdtar\_path=C:\Program Files (x86)\Cloudbase Solutions\Cloudbase-Init\bin\bsdtar.exe  # 日志等级.  verbose=true  debug=true  # 日志存放路径.  logdir=C:\Program Files (x86)\Cloudbase Solutions\Cloudbase-Init\log\  logfile=cloudbase-init-unattend.log  default\_log\_levels=comtypes=INFO,suds=INFO,iso8601=WARN  logging\_serial\_port\_settings=  # 启用最大传输单元和时间同步服务.  mtu\_use\_dhcp\_config=true  ntp\_use\_dhcp\_config=true  # 存放用户用于执行的脚本的路径.  local\_scripts\_path=C:\Program Files (x86)\Cloudbase Solutions\Cloudbase-Init\LocalScripts\  # 服务将会检测以下驱动直到某个能够成功获取元数据.  metadata\_services=cloudbaseinit.metadata.services.configdrive.ConfigDriveService,  cloudbaseinit.metadata.services.httpservice.HttpService,  cloudbaseinit.metadata.services.ec2service.EC2Service,  cloudbaseinit.metadata.services.maasservice.MaaSHttpService  # 需要执行的插件.  plugins=cloudbaseinit.plugins.common.mtu.MTUPlugin,“ cloudbaseinit.plugins.common.sethostname.SetHostNamePlugin  # 其他配置.  allow\_reboot=false # allow the service to reboot the system  stop\_service\_on\_exit=false |

cloudbase-init-unattend.conf配置文件与默认的类似以及它用户系统准备阶段. 他被用于最小化用户交互以及要求有MTU和修改主机名的插件的这种场景, 让镜像准备好进行进一步的初始化。

* 脚本执行

Cloudbase-init 能够执行用户提供的脚本, 通常放在默认路径 C:\Program Files (x86)\Cloudbase Solutions\Cloudbase-Init\LocalScripts, through a specific plugin 中执行，根据所使用的平台,这些脚本可以是MZPEs、PowerShell、Python、Batch或者Bash脚本. 用户数据插件也能够执行各种各样的脚本.

根据返回码, 你可以进行重启或者重启机器之后再执行命令:

1001 - 重启但启动后不再执行命令

1002 - 不重启，等下一次重启再执行命令

1003 - 重启并在启动时执行命令