

镜像制作

知识点	OpenStack镜像制作		
提交人	赵明俊	涉及内容	Glance

详细内容：

- 一、简述
- 二、准备环境
- 三、Linux镜像制作
 - （一）启动虚机
 - 1. 准备iso镜像文件
 - 2. 创建一块虚拟磁盘
 - 3. 启动镜像
 - 4. 安装系统
 - （二）安装软件包
 - 1. 安装基本软件包
 - 2. 安装cloud-init相关软件包
 - 2.5修改代码
 - 3. 修改配置文件
 - （三）其它系统配置
 - （四）定制化配置
 - （五）处理镜像
- 四、Windows镜像制作
 - （一）准备文件
 - （二）启动虚拟机
 - （三）配置系统
- 五、测试镜像
- 六、发布镜像
 - （一）发布准备
 - （二）生成镜像校验文件
 - （三）发布镜像
 - （四）验证

一、简述

制作openstack镜像，主要分为

1. 准备虚拟机运行环境
2. 安装一个虚拟机系统
3. 安装必要的软件包
4. 必要的系统配置
5. 系统定制化配置
6. 压缩镜像
7. 测试镜像
8. 发布镜像

本文档以CentOS6. 5为例，讲解如何制作Openstack镜像。

不同的Linux发行版的具体操作不一样，但大致上是一样的。

二、准备环境

建议使用一台物理机来配置kvm环境，需要cpu支持intel VT等硬件虚拟化功能。也可以用虚拟机来制作镜像，同样需要开启硬件虚拟化功能。

建议安装CentOS 7操作系统，确保网络及软件源配置正确。

可通过如下方法来判断

```
cat /proc/cpuinfo | grep vmx
```

确认有结果输出。则表示该机器支持或已配置intel VT
vmx(intel)或svm(AMD) 支持虚拟技术

安装软件包

```
yum install libvirt qemu-kvm virt-install bridge-utils qemu-img virt-manager libguestfs
```

启动服务

```
systemctl start libvirtd
```

三、Linux镜像制作

（一）启动虚拟机

1. 准备iso镜像文件

/tmp/CentOS-6.5-x86_64-minimal.iso

2. 创建一块虚拟磁盘

```
qemu-img create -f raw /tmp/centos-6.5.raw 20G
```

3. 启动镜像

```
virt-install --virt-type kvm --name centos-6.5 --ram 1024 \  
--disk /tmp/centos-6.5.raw,format=raw \  
--network network=default \  
--graphics vnc,listen=0.0.0.0 --noautoconsole \  
--os-type=linux --os-variant=rhel6 \  
--cdrom=/tmp/CentOS-6.5-x86_64-minimal.iso
```

4. 安装系统

使用vnc登入机器，按照正常的步骤安装系统。分区设置为单个/分区，不设置swap分区。vnc的ip为镜像宿主机的ip，端口可以ps找到，默认是5900

也可以使用virsh从命令行进行安装。部分操作命令如下

```
# 查看当前虚拟机  
virsh list [--all]  
# 使用virsh从console登入机器  
virsh console centos-6.5  
# 启动虚拟机  
virsh start centos-6.5  
# 查看vnc端口  
virsh vncdisplay centos-6.5
```

系统安装完成之后，重启虚拟机。

（二）安装软件包

如果虚拟机可以联网，则可以不用配置软件源，直接使用默认的即可。

1. 安装基本软件包

```
# 安装NetworkManager，用于网卡的自动发现及管理
# 安装acpid，用于虚拟机的电源管理
# 安装epel-release，添加epel源

yum install NetworkManager acpid epel-release

# 开机启动服务
chkconfig acpid on
chkconfig NetworkManager on
```

2. 安装cloud-init相关软件包

```
# 安装cloud-init，用于注入密码/密钥和主机名
# 安装qemu-guest-agent，用于在面板更新密码/密钥
# 安装cloud-utils，用于更改虚拟机根分区大小（可选安装，需要启用epel源）

yum install cloud-init qemu-guest-agent cloud-utils

# 开机启动服务（有的linux发行版默认不开机自启这些服务，需要手动设置开机自启）
chkconfig cloud-init on
chkconfig cloud-init-local on
chkconfig cloud-config on
chkconfig cloud-final on
chkconfig qemu-ga on    CENTOS7 为qemu-guest-agent
```

2.5修改代码

UOS3.0默认没有向虚拟机提供userdata，所以在UOS3.0下支持密码注入，需要修改cloud-init的代码实现

代码1：

/usr/lib/python2.7/site-packages/cloudinit/cloud.py

在合适的位置加入如下代码

```
def get_admin_pass(self):
    return self.datasource.get_admin_pass()
```

代码2：

/usr/lib/python2.7/site-packages/cloudinit/config/cc_set_admin_pass.py （新建文件）

写入如下代码：

```

from cloudinit import util
from string import letters, digits
# We are removing certain 'painful' letters/numbers
PW_SET = (letters.translate(None, 'loLOl') +
          digits.translate(None, '0l'))

def handle(_name, cfg, cloud, log, args):
    errors = []
    try:
        log.debug("Getting admin password from cloud")
        admin_pass = cloud.get_admin_pass()
        log.debug("Changing password for root")
        if admin_pass is None:
            admin_pass = rand_user_password()
        util.subp(['chpasswd'], 'root:%s' % admin_pass)
    except Exception as e:
        errors.append(e)
        util.logexc(log, "Failed to set passwords for root")
    if len(errors):
        log.debug("%s errors occurred, re-raising the last one", len(errors))
        raise errors[-1]

def rand_user_password(pwlen=9):
    return util.rand_str(pwlen, select_from=PW_SET)

```

如果虚拟机默认使用的是Python3，则将上面的代码替换为：

```

from cloudinit import util
from string import ascii_letters, digits                                     #Python3重写了string类
# We are removing certain 'painful' letters/numbers
PW_SET = (ascii_letters.translate('loLOl') +
          digits.translate('0l'))

def handle(_name, cfg, cloud, log, args):
    errors = []
    try:
        log.debug("Getting admin password from cloud")
        admin_pass = cloud.get_admin_pass()
        log.debug("Changing password for root")
        if admin_pass is None:
            admin_pass = rand_user_password()
        util.subp(['chpasswd'], 'root:%s' % admin_pass)
    except Exception as e:
        errors.append(e)
        util.logexc(log, "Failed to set passwords for root")
    if len(errors):
        log.debug("%s errors occurred, re-raising the last one", len(errors))
        raise errors[-1]

def rand_user_password(pwlen=9):
    return util.rand_str(pwlen, select_from=PW_SET)

```

代码3:

/usr/lib/python2.7/site-packages/cloudinit/sources/_init_.py

在合适的位置加入如下代码：

```
def get_admin_pass(self):  
    if not self.metadata or 'admin_pass' not in self.metadata:  
        return None  
    return str(self.metadata['admin_pass'])
```

3. 修改配置文件

cloud-init的配置文件

```
/etc/cloud/cloud.cfg
```

```

users:
  - default
disable_root: 0 #默认此处为1，需要更改
ssh_pwauth: 0
locale_configfile: /etc/sysconfig/i18n
mount_default_fields: [~, ~, 'auto', 'defaults,nofail', '0', '2']
resize_rootfs_tmp: /dev
ssh_deletekeys: 0
ssh_genkeytypes: ~
syslog_fix_perms: ~
cloud_init_modules:
  - migrator
  - bootcmd
  - write-files
  - growpart
  - resizefs
  - set_hostname
  - set_admin_pass #加入此行，即调用上面加入的代码

  - update_hostname
  - update_etc_hosts
  - rsyslog
  - users-groups
  - ssh
cloud_config_modules:
  - mounts
  - locale
  - set-passwords
  - 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
system_info:
  default_user:
    name: centos
    lock_passwd: true
    gecos: Cloud User
    groups: [wheel, adm]
    sudo: ["ALL=(ALL) NOPASSWD:ALL"]
    shell: /bin/bash
  distro: rhel
  paths:
    cloud_dir: /var/lib/cloud
    templates_dir: /etc/cloud/templates
  ssh_svcname: sshd
# vim:syntax=yaml

```

qemu-guest-agent的配置文件

```
/etc/sysconfig/qemu-ga
```

```
TRANSPORT_METHOD="virtio-serial"  
DEVPATH="/dev/virtio-ports/org.qemu.guest_agent.0"  
LOGFILE="/var/log/qemu-ga/qemu-ga.log"  
PIDFILE="/var/run/qemu-ga.pid"  
BLACKLIST_RPC="" #默认配置需要修改此行，清除引号中内容  
FSFREEZE_HOOK_ENABLE=0
```

（三）其它系统配置

禁用默认的zeroconf route

```
echo "NOZEROCONF=yes" >> /etc/sysconfig/network
```

修改kernel启动参数，编辑/boot/grub/grub.conf，在kernel所在行添加console字段

```
kernel ... console=ttyS0,115200n8 console=tty0
```

关闭开机启动服务

```
chkconfig iptables off  
chkconfig iptables6 off  
chkconfig postfix off
```

关闭selinux

```
/etc/selinux/config
```

```
...  
SELINUX=disabled  
...  
  
#setenforce 0
```

（四）定制化配置

对系统进行定制化配置，配置完后关闭虚拟机。

```
poweroff
```

（五）处理镜像

初始化镜像文件，删除网卡MAC等信息

```
virt-sysprep -d centos-6.5
```

Undefine虚拟机

```
virsh undefine centos-6.5
```

转换和压缩镜像文件

```
virt-sparsify --convert qcow2 --compress /tmp/centos-6.5.raw /tmp/centos-6.5.qcow2
```

四、Windows镜像制作

（一）准备文件

1. 准备iso镜像文件

/tmp/windows2008R2-x64.iso

2. 准备virtio驱动iso

下载地址: <https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/archive-virtio/virtio-win-0.1.118-2/virtio-win.iso>

3. 准备cloud-init软件

下载地址: https://cloudbase.it/downloads/CloudbaseInitSetup_0_9_8_x64.msi

4. 创建一块虚拟磁盘

```
# qemu-img create -f raw /tmp/win2k8.raw 40G
```

（二）启动虚拟机

5. 启动镜像（由于加载了两个光驱，如果不能启动系统安装盘的光驱，调换一下参数）

```
# virt-install --name win2k8 --ram 2048 --vcpus 2 \
  --network network=default,model=virtio \
  --graphics vnc,listen=0.0.0.0 --noautoconsole \
  --disk path=/tmp/win2k8.raw,device=disk,bus=virtio \
  --cdrom=/tmp/virtio-win.iso \
  --disk path=/tmp/windows2008-x64.iso,device=cdrom \
  --os-type windows --os-variant win2k8
```

6. 安装系统

安装系统过程中

默认情况下Windows检测不到可用的安装磁盘。需要加载驱动，选择光驱的viostor目录，按照系统版本加载相应的驱动
如安装Win8R2 时手动加载磁盘驱动：浏览 E:\viostor\2k8R2\amd64\viostor.inf

7. 完成系统安装，使用Administrator登录系统

（三）配置系统

8. 继续完成virtio驱动的安装

打开一个cmd窗口


```
> cd C:\windows\system32
> PnPutil.exe -i -a E:\viostor\2k8r2\amd64\viostor.inf
> PnPutil.exe -i -a E:\vioserial\2k8R2\amd64\vioser.inf
> PnPutil.exe -i -a E:\vioscsi\2k8R2\amd64\vioscsi.inf
> PnPutil.exe -i -a E:\viorng\2k8R2\amd64\viorng.inf
> PnPutil.exe -i -a E:\qemupcserial\qemupcserial.inf
> PnPutil.exe -i -a E:\NetKVM\2k8R2\amd64\netkvm.inf
> PnPutil.exe -i -a E:\Balloon\2k8R2\amd64\balloon.inf
```

也可以在硬件管理器，选择没有驱动的设备，安装驱动

PCI简易通信控制器安装vioserial驱动

PCI控制器安装Balloon驱动

以太网控制器安装NetKVM驱动

9. 安装qemu-ga

按照系统架构安装virtio驱动光盘里面的qemu-ga。

10. 安装Cloudbase-init

安装过程中的几个需要配置的地方

```
* Username: Administrator
* Serial port for logging: COM1
```

安装的最后一步按照需要选择是否进行sysprep，做完sysprep后系统会关机，所以应该最后安装cloudbase-init。

11. 配置Cloud-init（一般情况下不需要更改，使用默认的就可以）

配置文件在Cloud-init的安装目录下

配置文件参考：

```
[DEFAULT]
username=Administrator
groups=Administrators
inject_user_password=true
network_adapter=Red Hat VirtIO Ethernet Adapter
config_drive_raw_hhd=true
config_drive_cdrom=true
bsdtar_path=C:\Program Files (x86)\Cloudbase Solutions\Cloudbase-Init\bin\bsdtar.exe
verbose=true
debug=true
logdir=C:\Program Files (x86)\Cloudbase Solutions\Cloudbase-Init\log\
logfile=cloudbase-init.log
logging_serial_port_settings=COM1,115200,N,8
mtu_use_dhcp_config=false
ntp_use_dhcp_config=false
local_scripts_path=C:\Program Files (x86)\Cloudbase Solutions\Cloudbase-Init\LocalScripts\
metadata_services=cloudbaseinit.metadata.services.configdrive.ConfigDriveService
allow_reboot=false
stop_service_on_exit=false
```

12. 关闭虚拟机，导出镜像文件，转换格式，可参考上文Linux镜像制作相关步骤

五、测试镜像

将cloudmetadata.iso作为虚拟机唯一的光驱连接到关闭状态的虚拟机，启动虚拟机后，检查虚拟机是否可以被注入hostname和密码，hostname应该被注入为l
t，密码为Ustack2016

六、发布镜像

（一）发布准备

登录`server-241.0.XX.ustack.in`(XX为region name)，将制作好的镜像上传到目录`/var/image/extra/`

重命名镜像，示例如下

发行版---详细版本号---制作日期（日期+字母）.qcow2
CentOS---7.1-64bit---2016-03-20-a.qcow2

说明：

发行版 和 详细版本号 用于镜像分类

制作日期用来标志唯一镜像

格式必须满足上述示例要求

对`/var/image/extra/uiu/meta-info/`目录内文件的说明

我们通过该目录预定义一些镜像的属性，meta-info内含各个镜像的基本属性，当新添加发行版或新的大版本时，需要创建关联的文件

`ls meta-info/CentOS/`

`7.1-64bit`

主要看EXPECTED_SIZE，这个要和raw格式镜像的大小匹配

需要注意的是image_meta里的pass不一定是初始密码（安装了cloudinit的镜像，初始密码由cloudinit读取metadata获得）

display_name是面板上所展示的名字

image_name_order和image_label_order分别用于定义面板上镜像的显示顺序的

（二）生成镜像校验文件

```
# md5sum CentOS---7.1-64bit---2016-03-20-a.qcow2 > CentOS---7.1-64bit---2016-03-20-a.qcow2.md5sum
```

（三）发布镜像

进入目录`/var/image/extra/uiu/`，执行`publish_extra_img`上传镜像

```
# source /root/openrc  
# ./publish_extra_img /var/image/extra/CentOS---7.1-64bit---2016-03-20-a.qcow2
```

（四）验证

glance命令

```
# 查看镜像列表  
glance image-list  
# 查看具体镜像  
glance image-show [IMAGE_ID]  
# 修改如何镜像属性  
glance help image-update  
# 删除镜像  
glance image-delete [IMAGE_ID]
```

在web界面使用上传的镜像新建虚拟机，测试是否可以正常使用