

Openstack_v10(CL-210课程)_03

2018年3月8日 23:03

自定义镜像

在Openstack中通常情况下，使用自定义镜像都是最好的方式。

自定义镜像可以：加固服务、优化系统、安全配置、部署监控agent、设置账户等等。

Openstack支持多种虚拟机镜像格式：

1. RAW（裸盘）
2. QCOW2(Copy on write v2)
3. AMI(Amazon亚马逊镜像格式)
- 4.VHD（微软VPC格式）
- 5.VMDK（Vmware虚拟机格式）

RAW和QCOW2对比

属性	RAW	QCOW2
镜像大小	大	小
性能	好	较好
加密	不支持	可选支持
快照	不支持	支持多种快照方式
Copy on write	不支持	仅拷贝被修改的数据，原始盘不变（有效降低磁盘消耗）

Linux创建虚拟机磁盘的命令：

```
# qemu-img create [-q] [-f fmt] filename [size]
```

Supported formats:

```
vvfat vpc vmdk vhdx vdi ssh sheepdog rbd
raw host_cdrom host_floppy host_device file qed
qcow2 qcow parallels nbd iscsi gluster dmg
tftp ftps ftp https http cloop bochs blkverify blkdebug
```

Openstack组件中的Nova(Compute)负责允许虚拟机，默认Openstack使用的格式为QCOW2.

Openstack组件中的Glance(Image)负责注册镜像、保存镜像、提供镜像给账户.

diskimage-builder这个软件包可以构建自定义虚拟机镜像，
该软件可以输出多种格式的虚拟机镜像，如：RAW和QCOW2。
该软件提供了一个disk-image-create命令。

Diskimage-builder Elements:

Elements（代码集合）用来决定把哪些内容放到镜像里，以及对镜像做什么修改。

制作所有的镜像至少需要一个基础element.

Diskimage-builder提供了大量Element模板

```
# ls /usr/share/diskimage-builder/elements/
```

apt-conf	deploy	fedora	openssh-server	select-boot-kernel-initrd
apt-preferences	deploy-baremetal	fedora-minimal	opensuse	selinux-permissive
apt-sources	deploy-ironic	gentoo	opensuse-minimal	serial-console
architecture-emulation-binaries	deploy-kexec	growroot	package-installs	simple-init
baremetal	deploy-targetcli	grub2	partitioning-sfdisk	source-repositories
<u>base</u>	deploy-tgtadm	hpdsa	pip-and-virtualenv	stable-interface-names
bootloader	devuser	hwburnin	pip-cache	svc-map
cache-url	dhcp-all-interfaces	hwdiscovery	pkg-map	sysctl
centos	dib-init-system	ilo	posix	uboot
<u>centos7</u>	dib-python	install-bin	proliant-tools	ubuntu
centos-minimal	dib-run-parts	install-static	pypi	ubuntu-core
cleanup-kernel-initrd	disable-selinux	install-types	python-brickclient	ubuntu-minimal
cloud-init	dkms	ironic-agent	ramdisk	ubuntu-signed
cloud-init-datasources	docker	ironic-discoverd-ramdisk	ramdisk-base	vm
cloud-init-disable-resizefs	dpkg	iso	rax-nova-agent	yum
cloud-init-nocloud	dracut-network	local-config	redhat-common	yum-minimal
<u>debian</u>	dracut-ramdisk	manifests	<u>rhel</u>	zypper
debian-minimal	dynamic-login	mellanox	<u>rhel7</u>	zypper-minimal
debian-systemd	element-manifest	modprobe-blacklist	rhel-common	
debian-upstart	enable-serial-console	no-final-image	rpm-distro	
debootstrap	epel	oat-client	runtime-ssh-host-keys	

#tree /usr/share/diskimage-builder/elements/base/

```

/usr/share/diskimage-builder/elements/base/
├── cleanup.d
│   ├── 01-ccache
│   └── 99-tidy-logs
├── element-deps
├── environment.d
│   └── 10-ccache.bash
├── extra-data.d
│   └── 50-store-build-settings
├── install.d
│   ├── 00-baseline-environment
│   ├── 00-up-to-date
│   ├── 10-cloud-init
│   ├── 50-store-build-settings
│   └── 80-disable-rfc3041
├── package-installs.yaml
├── pkg-map
├── pre-install.d
│   └── 03-baseline-tools
├── README.rst
├── root.d
│   └── 01-ccache

```

6 directories, 15 files

登陆workstation实验:

```
[student@workstation ~]$ lab customization-img-building setup
```

//脚本会检查实验环境, 安装diskimage-build软件包, 创建账户、安全组等Openstack资源

```
[student@workstation ~]$ wget http://materials.example.com/osp-small.qcow2
```

//下载基础镜像

```
[student@workstation ~]$ cp -a /usr/share/diskimage-builder/elements/ /home/student/
```

//拷贝Elements

```
[student@workstation ~]$ mkdir -p /home/student/elements/rhel7/post-install.d/
```

//创建脚本目录(名称不能任意修改, 有要求)

```
[student@workstation ~]$ cd /home/student/elements/rhel7/post-install.d/
```

编写脚本内容如下:

```
[student@workstation post-install.d]$ cat 01-enable-service
```

```
#!/bin/bash
```

```
systemctl enable vsftpd
```

```
[student@workstation post-install.d]$ cat 02-vsftpd-disable-anonymous
```

```
#!/bin/bash
```

```
sed -i 's|^anonymous_enable=.*|anonymous_enable=NO|' /etc/vsftpd/vsftpd.conf
```

设置脚本权限（必须有可执行权限）

```
[student@workstation post-install.d]$ chmod +x /home/student/elements/rhel7/post-install.d/*
```

定义环境变量

```
[student@workstation ~]$ export NODE_DIST=rhel7
```

//基础镜像的版本

```
[student@workstation ~]$ export DIB_LOCAL_IMAGE=/home/student/osp-small.qcow2
```

//基础镜像的位置

```
[student@workstation ~]$ export DIB_YUM_REPO_CONF=/etc/yum.repos.d/openstack.repo
```

//YUM源位置，diskimage-builder会把这个配置文件复制到chroot中，用来制作虚拟机镜像时安装软件

```
[student@workstation ~]$ export ELEMENTS_PATH=/home/student/elements
```

//指定Elements的目录

自定义虚拟机镜像：

```
[student@workstation ~]$ disk-image-create vm rhel7 -t qcow2 -p vsftpd -o finance-rhel-ftp.qcow2
```

//调用vm这个Elements（自动分区）

//调用rhel7这个Elements制作基于RHEL7系统的镜像

//-t设置镜像格式为qcow2

//-p需要安装的软件有vsftpd，逗号分隔可以写多个软件包

//-o导出镜像文件

使用admin管理员登陆Web控制台创建项目与账户

```
# firefox http://172.25.250.50
```

Identity-->Projects-->Create project

Identity-->Users-->Create user

使用图形上传虚拟机镜像：

```
# firefox http://172.25.250.50
```

Admin-->Images-->Create image

设置Minimum Disk大小为10G

设置Minimum RAM大小为2048M

上传虚拟机镜像的另一种方法是命令行（需要source加载账户RC文件）

```
[student@workstation ~(admin-admin)]$ openstack image create \
```

```
> --disk-format qcow2 --min-disk 10 --min-ram 2048 \
```

```
> --file finance-rhel-ftp.qcow2 --public \
```

```
> finance-rhel-ftp
```

//上传虚拟机镜像，名称为finance-rhel-ftp，格式为qcow2，最小磁盘为10G，最小内存为2048M

//上传镜像文件为finance-rhel-ftp.qcow2

//该虚拟机镜像为public公共镜像，任何人都可以使用该镜像

使用图形创建虚拟机实例（使用前面上传的镜像）

验证虚拟机的内容是否符合自定义的要求

Details:

Instance name: FTP

Source:

Select Boot Source: 选择Image
Create New Volume: 选择No
Available:可选镜像中选择finance-rhel-ftp

Flavor:

Available: 可选主机箱中选择m1.web

Networks:

选择合适的网络（没有网络，则可以先创建一个网络）

Security Groups:

选择合适的安全组（没有，则可以先创建一个安全组）

Key Paire:

选择合适的密钥（没有，则可以先创建密钥文件）

清空实验环境#

```
[student@workstation ~]$ lab customization-img-building cleanup
```

自定义镜像的另一种方式:

使用diskimage-builder创建自定义镜像需要几分钟，还需要拷贝elements。
如果我们仅需要少量简单的自定义镜像，也可以使用更简单的guestfish工具。
libguestfs-tools-c提供了guestfish命令。

登陆workstation操作实验:

```
[student@workstation ~]$ lab customization-img-customizing setup
```

//准备实验环境

或者直接安装guestfish软件也可以软件包为libguestfs-tools-c

```
[student@workstation ~]$ yum -y install libguestfs-tools-c
```

```
[student@workstation ~]$ wget http://materials.example.com/osp-small.qcow2 -O finance-rhel-db.qcow2
```

//下载基础镜像

```
[student@workstation ~]$ guestfish -i --network -a finance-rhel-db.qcow2
```

//-i自动挂载分区

//-a加载虚拟机磁盘

//--network允许网络访问

```
> <fs> command "yum -y install mariadb mariadb-server"
> <fs> command "systemctl enable mariadb"
> <fs> command "systemctl is-enabled mariadb"
> <fs> command "yum -y install httpd"
> <fs> command "touch /var/www/html/index.html"
> <fs> command "edit /var/www/html/index.html"
```

注意：因为对文件系统中的内容修改是在guestfish环境下操作，这些变化的数据不会获得有效的SELinux标签。

比如，修改/etc/passwd，在启动虚拟机时会因为SELinux拒绝访问，而导致镜像不可用！

虽然，通过touch /.autorelabel命令，可以让虚拟机在启动时重新给文件打SELinux标签。

但是，因为该文件被内置到了虚拟机镜像模板，会导致每次启动虚拟机都会给文件重新打标签（非常耗时）。

使用如下指令立刻为虚拟机重新制作SELinux标签是比较明智的做法：

```
> <fs> selinux-relabel /etc/selinux/targeted/contexts/files/file_contexts /
> <fs> exit
```

另外，libguestfs-tools-c还提供了一个virt-customize工具

```
# virt-customize -a rhel.qcow2 \
--install aide \
--root-password password:letmein \
--run-command 'yum -y install httpd' \
--selinux-relabel
```

对比

特色	Guestfish	Virt-customize
复杂度	底层工具，稍复杂	高级工具，更简单
SELinux支持	selinux-relabel /etc/selinux/targeted/contexts/files/file_contexts /	--selinux-relabel
功能	提供更多底层工具，如分区，脚本，远程访问	安装软件、修改密码 设置主机名，设置时区等

```
[student@workstation ~]$ lab customization-img-customizing cleanup
//清除实验环境
```