Create a KVM Based Debian 7 OpenStack Cloud Image

Monday, April 7, 2014

Unlike Red Hat (https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=16952), Fedora (http://fedoraproject.org/en/get-fedora#clouds), CentOS (http://repos.fedorapeople.org/repos/openstack/guest-images/), and Canonical (http://cloud-images.ubuntu.com/releases/), Debian does not create and distribute OpenStack cloud images. It is up to you to create your own.

I have seen steps using packer templates (https://github.com/osuosl/packer-templates) or virt-builder (http://rwmj.wordpress.com/2014/02/13/virt-builder-installing-cloud-init-in-a-debian-7-guest/) to create Debian 7 OpenStack cloud images, but I have not seen a step-by-step manual process to understand how it all works.

This post will be that step-by-step manual process of creating a Debian 7 OpenStack cloud image. Many of the steps were derived from the links mentioned above as well as other links in the **References** section below.

The following steps will require you to have access to a Linux distribution that has **KVM/QEMU** and **libvirt** installed.

Install KVM/QEMU and libvirt

If you happen to be using a Linux distribution that does not have any virtualization packages installed, the following packages will be needed depending on which Linux distribution you are running.

CentOS/RHEL/Fedora

Install the necessary repository packages:

```
yum groupinstall "Virtualization" "Virtualization Platform"
```

Enable and start the **libvirtd** service on CentOS 6 or RHEL 6:

```
chkconfig libvirtd on

service libvirtd start
```

Enable and start the **libvirtd** service on Fedora 17 and newer:

```
systemctl enable libvirtd

systemctl start libvirtd
```

Debian/Ubuntu

Install the necessary repository packages:

```
sudo apt-get install qemu-kvm libvirt-bin virt-manager
```

Start the **libvirt-bin** service:

```
service libvirt-bin start
```

Create the Virtual Machine

You can create the Debian 7 virtual machine using the following command. It will pull everything it needs from the official Debian repositories. Depending on the speed of your internet connection, the install may take anywhere from 5 - 20 minutes:

```
sudo virt-install --virt-type kvm --name debian-7 --ram 1024 --location=http://ftp
.debian.org/debian/dists/wheezy/main/installer-amd64/ --disk path=/tmp/debian-7-am
d64-vm.img,size=5 --network network=default --graphics vnc,listen=0.0.0.0 --noauto
console --os-type=linux --os-variant=debianwheezy --extra-args="priority=critical
interface=auto debian-installer/language=en debian-installer/country=US debian-ins
taller/locale=en_US keymap=us preseed/url=http://public.thornelabs.net/debian-7-wh
eezy-preseed.seed"
```

Once the install finishes, the virtual machine will automatically shutdown. Start it back up and proceed to the next section.

Post-install Steps

With the base install of Debian 7 in place, it is time for you to configure the operating system to work with OpenStack.

Using the KVM console or SSH, login as the **root** user (password is **debian**).

Add the Debian Wheezy Backports repository to apt so you can install the **cloud-init** packages:

```
echo 'deb http://ftp.debian.org/debian wheezy-backports main' >> /etc/apt/sources.
list
```

Update apt-get so it sees the new repository:

```
apt-get update
```

Install all of the necessary cloud-init packages along with some other packages:

```
apt-get install sudo curl wget rsync cloud-init cloud-utils cloud-initramfs-growro
ot
```

Replace the default **/etc/cloud/cloud.cfg** with the following contents:

```
# The top level settings are used as module
# and system configuration.

# A set of users which may be applied and/or used by various modules
# when a 'default' entry is found it will reference the 'default_user'
# from the distro configuration specified below
users:
   - default

# If this is set, 'root' will not be able to ssh in and they
# will get a message to login instead as the above $user (ubuntu)
disable_root: true

# This will cause the set+update hostname module to not operate (if true)
preserve_hostname: false

# Example datasource config
# datasource:
# Ec2:
# metadata_urls: [ 'blah.com' ]
# timeout: 5 # (defaults to 50 seconds)
# max_wait: 10 # (defaults to 120 seconds)

# The modules that run in the 'init' stage
cloud_init_modules:
 - migrator
 - bootcmd
 - write-files
 - resizefs
 - set_hostname
 - update_hostname
 - update_etc_hosts
 - ca-certs
 - rsyslog
 - users-groups
 - ssh

# The modules that run in the 'config' stage
cloud_config_modules:
# Emit the cloud config ready event
# this can be used by upstart jobs for 'start on cloud-config'.
 - emit_upstart
 - mounts
```

```
 - ssh-import-id
 - locale
 - set-passwords
 - grub-dpkg
 - apt-pipelining
 - apt-configure
 - package-update-upgrade-install
 - landscape
 - timezone
 - puppet
 - chef
 - salt-minion
 - mcollective
 - disable-ec2-metadata
 - runcmd
 - byobu

# The modules that run in the 'final' stage
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 and/or distro specific settings
# (not accessible to handlers/transforms)
system_info:
   # This will affect which distro class gets used
   distro: debian
   # Default user name + that default users groups (if added/used)
   default_user:
     name: debian
     lock_passwd: True
     gecos: Debian
     shell: /bin/bash
     groups: [adm, audio, cdrom, dialout, floppy, video, plugdev, dip, sudo]
     sudo: ["ALL=(ALL) NOPASSWD: ALL"]
   # Other config here will be given to the distro class and/or path classes
   paths:
      cloud_dir: /var/lib/cloud/
      templates_dir: /etc/cloud/templates/
      upstart_dir: /etc/init/
   package_mirrors:
     - arches: [default]
```

```
    failsafe:
      primary: http://ftp.debian.org/debian
```

Change the GRUB configuration so the OpenStack console outputs the boot logs:

```
sed -i -e 's/^GRUB_CMDLINE_LINUX_DEFAULT=.*/GRUB_CMDLINE_LINUX_DEFAULT="console=tt
yS0"/' /etc/default/grub
```

Update GRUB to apply the changes:

```
update-grub
```

Change the networking configuration to work with cloud-init:

```
sed -i 's/allow-hotplug eth0/auto eth0/' /etc/network/interfaces
```

Clean up apt:

```
apt-get autoremove
apt-get autoclean
apt-get clean
```

Remove everything in the **/tmp** directory:

```
rm -rf /tmp/*
```

Clean up the last logged in users logs:

```
rm -f /var/log/wtmp /var/log/btmp
```

Clean up the command history:

```
history -c
```

Shutdown the virtual machine:

```
shutdown -h now
```

Compress the Image

The `virt-install` command above created the virtual machine's backing disk file at **/tmp/debian-7-amd64-vm.img** and the final file size will be about 5 GB. Typically, OpenStack images are not this large, and there is a lot of useless space within the current image. So, use the `qemu-img` command to compress the image to about 500 MB:

```
qemu-img convert -c /tmp/debian-7-amd64-vm.img -O qcow2 /tmp/debian-7-amd64.img
```

Upload the Image into the Glance Repository

With the final image created, it is time to upload it to the Glance Repository in your OpenStack environment.

You can upload the image into the Glance Repository from a local file path using the following command:

```
glance image-create --name debian-7-amd64 --disk-format=qcow2 --container-format=b
are --file /tmp/debian-7-amd64.img
```

Or, you can upload the image into the Glance Repository from a URL using the following command:

```
glance image-create --name debian-7-amd64 --disk-format=qcow2 --container-format=b
are --copy-from http://example.com/debian-7-amd64.img
```

With the image now uploaded into your Glance Repository, you can begin creating OpenStack Instances from it.

Your SSH public key will be injected into the **debian** user's directory. So, when you login via SSH, login as that user. If you want to change this user modify the **name** parameter in the **default_user** section of the cloud-init config above.

References

openstack-debian.sh (https://github.com/osuosl/packer-templates/blob/master/scripts/openstack-debian.sh)

preseed-debian2.cfg (https://github.com/osuosl/packer-templates/blob/master/http/preseed-debian2.cfg)

virt-builder: Installing cloud-init in a Debian 7 guest (http://rwmj.wordpress.com/2014/02/13/virt-builder-installing-cloud-init-in-a-debian-7-guest/)

how to shrink the image requirements for disk (https://ask.openstack.org/en/question/8050/how-to-shrink-the-image-requirements-for-disk/)

Chapter 3. OpenStack Linux image requirements (http://docs.openstack.org/grizzly/openstack-image/content/ch_openstack_images.html)

Example: Ubuntu image (http://docs.openstack.org/image-guide/content/ubuntu-image.html)

preseeding debian netinstall does not get preseed file via http (http://serverfault.com/questions/414718/preseeding-debian-netinstall-does-not-get-preseed-file-via-http)

recent posts

Create a KVM Based CentOS 6 OpenStack Cloud Image (/2014/09/24/create-a-kvm-based-centos-6-openstack-cloud-image.html)

Reconfigure a Dell RAID (PERC) Controller with racadm (/2014/09/03/reconfigure-a-dell-raid-perc-controller-with-racadm.html)

Configure Multiple Network Interfaces on an OpenStack Instance (/2014/09/03/configure-multiple-network-interfaces-on-an-openstack-instance.html)

Delete Duplicate OpenStack Hypervisors and Services (/2014/08/03/delete-duplicate-openstack-hypervisors-and-services.html)

Questions to Ask to Determine if Your Application is Cloud Ready (/2014/08/03/questions-to-ask-to-determine-if-your-application-is-cloud-ready.html)

other posts

Understanding the Chef Environment File in Rackspace Private Cloud v4.2.x powered by OpenStack Havana (/2014/02/17/understanding-the-chef-environment-file-in-rackspace-private-cloud-v42x-powered-by-openstack-havana.html)

Use OpenStack Swift as a Backend Store for Glance (/2014/08/03/use-openstack-swift-as-a-backend-store-for-glance.html)

Create a Ubuntu Server 12.04 LTS Vagrant Base Box from Scratch Using VMware Fusion (/2013/09/12/create-a-ubuntu-server-1204-lts-vagrant-base-box-from-scratch-using-vmware-fusion.html)

RHEL 5 and RHEL 6 Core vs Base Package Group (/2013/01/23/rhel-5-and-rhel-6-core-vs-base-package-group.html)

MegaCli64 Command Usage Cheat Sheet (/2013/02/01/megacli64-command-usage-cheat-sheet.html)

---

about

I wanted a website. But I didn't want just another website. I wanted a website that I would use. I wanted a website that other people would use and, hopefully, benefit from.

I had three main goals that I want this website to accomplish: 1) provide me an outlet to develop my writing (always a work in progress), 2) make public all of the documentation stored on my hard drive, 3) figure out if I can make money from a website. Have I accomplished these goals? Not yet. I'm not sure I ever will. But it is just that which motivates me to keep working towards them.

blogs you should follow

Ken Hui's Cloud Architect Musings (http://cloudarchitectmusings.com/)

Cody Bunch's Professional OpenStack (http://openstack.prov12n.com)

Scott Lowe's Blog (http://blog.scottlowe.org/)

Major Hayden's Blog (http://major.io/)

Kevin Jackson's Collection of OpenStack Posts (http://openstackr.wordpress.com/)

Dan Walsh's Blog (http://danwalsh.livejournal.com/)

contact

@jameswthorne (https://twitter.com/jameswthorne) on Twitter

jthorne on **irc.freenode.net**

Thorne**Labs** is created using Twitter Bootstrap (http://getbootstrap.com/) and Jekyll (http://jekyllrb.com/) and is hosted on the Rackspace Public Cloud (http://www.rackspace.com/cloud/).