

# Optimizing VM images for OpenStack with KVM/QEMU

*Chet Burgess  
Sr. Director, Engineering  
Metacloud*

# A Few Notes

- All topics covered in this presentation assume the following software versions:
  - QEMU 1.0.0+
  - OpenStack Grizzly (2013.1)
  - libvirt 0.9.8+
  - Ubuntu 12.04 LTS
  - RHEL/CentOS 6.3
- There are a number of different ways of doing what is described in this presentation. This is simply one way of doing things based upon our experience running production clouds for our clients.

# Introduction

- Disk & Container Formats
  - RAW
  - QCOW2
  - AMI
  - Launching an Instance
- Image OS Preparation
  - cloud-init
  - Authentication Model
  - Networking
  - Hotplug Support
- Putting It All Together

# Tools For Manipulating Disk Files

- As with disk and container formats there are too many different tools to cover in one session, but here are some of our favorites:
- QEMU (<http://www.qemu.org>)
  - qemu-img – Swiss Army Knife
  - qemu-nbd – Mounting QCOW disk files
- libguestfs (<http://libguestfs.org>)
  - guestmount (requires FUSE) – Mounting various disk formats
  - virt-filesystems – Detailed disk file information

# Disk vs. Container Formats

- Disk formats store partition and block data:
  - QEMU+KVM supports a cornucopia of different disk formats, too many to cover in this presentation
  - We will be covering RAW and QCOW2
- Container formats express metadata about a VM as well as its underlying block devices:
  - Typically contains files in a disk format
  - We will be covering AMI

# RAW Disk Format

- Direct representation of a disk structure
- Can be sparse
- Widely supported by hypervisors
- Can be treated as a block device

# QCOW2 Disk Format

- QEMU Copy on Write Version 2
- Supports pre-allocation as well as on-demand allocation of blocks
- Wide range of features:
  - Read-only backing files
  - Snapshots (internal and external)
  - Compression
  - Encryption

# RAW vs QCOW2

- RAW has very little overhead and thus a performance advantage
- QCOW2 is designed for virtualization and actively developed with cloud like use cases in mind
- QCOW2 supports snapshots\*



# RAW vs QCOW2 (cont)

- Benefits of read-only backing files with QCOW2:
  - Faster instance launching
    - Common backing files
    - Size can be represented virtually
  - Increased storage efficiency
    - No duplication of “base” image data

# AMI...Identity Crisis

- It's a disk format, its RAW, it's a container, it's all of the above.
- Three distinct files
  - AMI – Amazon Machine Image
    - A raw disk image containing no partition information or boot sectors, just the file system (/dev/sda).
  - AKI – Amazon Kernel Image (vmlinuz)
  - ARI – Amazon Ramdisk Image (initrd)
- AMI is booted using the associated ARI and AKI

# Launching an Instance

- User selects an image and flavor
- Request is scheduled to nova-compute node
- nova-compute ensures the image is available locally
  - Checks in instance\_dir/\_base
  - Downloads from glance if not found
  - Converted to RAW as necessary using qemu-img
  - Image is stored in instance\_dir/\_base

# Launching an Instance (cont)

- Disk file is created at `instance_dir/instance_uuid/disk`
  - QCOW2
    - Default in nova
    - Dynamically allocated
    - Backing file of previously downloaded image
    - Disk Size Set
      - Disk size of the flavor
      - If no size in the flavor, backing file size

# Launching an Instance (cont)

- Disk file is created at instance\_dir/instance\_uuid/disk
  - RAW
    - Requires flag – “use\_cow\_images=false”
    - A copy of the image file is created
    - Disk resized using qemu-img
      - Disk size of the flavor
      - If no size in the flavor, not resized



# Image OS Preparation

- cloud-init
- Authentication Model
- Networking
- Hotplug Support

# cloud-init Overview

- Flexible, modular toolset to handle numerous instance configuration tasks
  - <http://tinyurl.com/cloudinit>
- Leverages Nova's metadata API (compatible with EC2)
  - <http://tinyurl.com/ec2metadata>
- Provides a module for dynamically resizing the root file system
  - Can only grow to the maximum size defined by the partition
  - AMI has no partition table
  - Ubuntu provides cloud-initramfs-growroot

# cloud-init Installation

- Ubuntu package installation:

```
# apt-get install cloud-init cloud-initramfs-growroot
```

- RHEL/CentOS package installation:

- Setup EPEL: <http://tinyurl.com/epelpackages>

```
# yum install cloud-init
```



# cloud-init Configuration

- Extremely basic configuration:

```
# vi /etc/cloud.cfg:
  user: cloud
  disable_root: 1
  preserve_hostname: False
```

- Verify cloud-init is configured for the “EC2” data source (Ubuntu Only):

```
# dpkg-reconfigure cloud-init
```

# Authentication Model

- Disable SSH password-based logins
- Remote root logins disallowed via SSH

```
PasswordAuthentication no  
PermitRootLogin no
```

- Allow anyone in the "sudo" (Ubuntu) or "wheel" (RHEL) group to obtain root without a password

```
%sudo    ALL= (ALL:ALL)  NOPASSWD:  ALL
```

# Authentication Model (cont)

- Create a "cloud" user and add it to the sudo or wheel group
- Lock the password of the "cloud" user

```
# useradd -m -G sudo -s /bin/bash cloud  
# passwd -l cloud
```

- Allow root login *without* a password from the console
  - Can only be used from the virtual console
  - Virtual console access equates to root regardless
  - Password here has no actual security benefit

```
# passwd -d root
```

# Networking

- MAC addresses are generated dynamically each time an instance is spawned
- Eliminate MAC address binding
  - udev rules

```
# rm -r /etc/udev/rules.d/70-persistent-net.rules
# rm -r /lib/udev/write_net_rules
```
  - Interface configuration (RHEL/CentOS)
    - Remove or comment out HWADDR in `/etc/sysconfig/network-scripts/ifcfg-eth*`

# Networking (cont)

- Configure the DHCP Client for persistence
  - VMs should never give up trying to obtain a DHCP lease
  - The alternative is to fall off the network and require administrative intervention

- RHEL/CentOS

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
    PERSISTENT_DHCLIENT=yes
```

- Ubuntu

```
# rm /sbin/dhclient3
```

# Hotplug Support

- Required for attaching and detaching cinder volumes without rebooting
- Make sure the following kernel modules load at boot

```
# echo 'modprobe acpihp' > /etc/rc.modules  
# echo 'modprobe pci_hotplug' >> /etc/rc.modules  
# chmod +x /etc/rc.modules
```

- Already statically compiled into some kernels

# Putting It All Together

- AMI formatted images stored in glance
  - Automatic resizing of the file system by cloud-init
- RAW backed dynamically-allocated QCOW2 instance disks
  - Faster instance launching
  - Increased storage efficiency
  - Snapshots
- cloud-init
- Properly Prepared OS

# More Information

- Video and slides will be available via the conference
  - Direct Link: **<http://tinyurl.com/mc-image-presentation>**
- Contact Information
  - My initials (cfb) at my company (metacloud.com)
  - cburgess on irc.freenode.net
    - #openstack, #openstack-dev, #openstack-nova
  - Twitter: cfbIV
- Visit me at the Metacloud booth (place)
  - Monday 2PM-4PM
  - Tuesday 3:30PM-4:30PM
  - Wednesday 4:30PM-5:30PM