QEMU AND RBD

Ceph integrates with the QEMU virtual machine. For details on QEMU, see QEMU Open Source Processor Emulator. For QEMU documentation, see QEMU Manual.

Important: To use Ceph block devices with QEMU, you must have a running Ceph cluster.

INSTALLING QEMU ON UBUNTU 12.04 PRECISE

QEMU packages are incorporated into the Ubuntu 12.04 precise distribution. To install QEMU on precise, execute the following:

```
sudo apt-get install qemu
```

INSTALLING QEMU ON EARLIER VERSIONS OF UBUNTU

For Ubuntu distributions 11.10 Oneiric and earlier, you must install the 0.15 version of QEMU or later. To build QEMU from source, use the following procedure:

```
cd {your-development-directory}
git clone git://git.qemu.org/qemu.git
cd qemu
./configure --enable-rbd
make; make install
```

CREATING IMAGES WITH QEMU

You can create a block device image from QEMU. You must specify `rbd`, the pool name, and the name of the image you wish to create. You must also specify the size of the image.

```
qemu-img create -f rbd rbd:{pool-name}/{image-name} {size}
```

For example:

```
qemu-img create -f rbd rbd:data/foo 10G
```

RESIZING IMAGES WITH QEMU

You can resize a block device image from QEMU. You must specify `rbd`, the pool name, and the name of the image you wish to resize. You must also specify the size of the image.

```
qemu-img resize -f rbd rbd:{pool-name}/{image-name} {size}
```

For example:

```
qemu-img resize -f rbd rbd:data/foo 10G
```

RETRIEVING IMAGE INFORMATION WITH QEMU

You can retrieve block device image information from QEMU. You must specify `rbd`, the pool name, and the name of the image.

```
qemu-img info -f rbd rbd:{pool-name}/{image-name}
```

For example:

```
qemu-img info -f rbd rbd:data/foo
```

RUNNING QEMU WITH RBD

QEMU can pass a block device from the host on to a guest, but since QEMU 0.15, there's no need to map an image as a block device on the host. Instead, QEMU can access an image as a virtual block device directly via `librbd`. This performs better because it avoids an additional context switch, and can take advantage of RBD caching.

You can use `qemu-img` to convert existing virtual machine images to Ceph block device images. For example, if you have a qcow2 image, you could run:

```
qemu-img convert -f qcow2 -O rbd debian_squeeze.qcow2 rbd:data/squeeze
```

To run a virtual machine booting from that image, you could run:

```
qemu -m 1024 -drive format=raw,file=rbd:data/squeeze
```

RBD caching can significantly improve performance. Since QEMU 1.2, QEMU's cache options control `librbd` caching:

```
qemu -m 1024 -drive format=rbd,file=rbd:data/squeeze,cache=writeback
```

If you have an older version of QEMU, you can set the `librbd` cache configuration (like any Ceph configuration option) as part of the 'file' parameter:

```
qemu -m 1024 -drive format=raw,file=rbd:data/squeeze:rbd_cache=true,cache=writeback
```

Important: If you set rbd_cache=true, you must set cache=writeback or risk data loss. Without cache=writeback, QEMU will not send flush requests to librbd. If QEMU exits uncleanly in this configuration, filesystems on top of rbd can be corrupted.

ENABLING DISCARD/TRIM

Since Ceph version 0.46 and QEMU version 1.1, Ceph block devices support the discard operation. This means that a guest can send TRIM requests to let a Ceph block device reclaim unused space. This can be enabled in the guest by mounting ext4 or XFS with the `discard` option.

For this to be available to the guest, it must be explicitly enabled for the block device. To do this, you must specify a `discard_granularity` associated with the drive:

```
qemu -m 1024 -drive format=raw,file=rbd:data/squeeze,id=drive1,if=none \
     -device driver=ide-hd,drive=drive1,discard_granularity=512
```

Note that this uses the IDE driver. The virtio driver does not support discard.

If using libvirt, edit your libvirt domain's configuration file using `virsh edit` to include the `xmlns:qemu` value. Then, add a `qemu:commandline` block as a child of that domain. The following example shows how to set two devices with `qemu id=` to different `discard_granularity` values.

```
<domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
        <qemu:commandline>
                <qemu:arg value='-set'/>
```

```
                <qemu:arg value='block.scsi0-0-0.discard_granularity=4096'/>
                <qemu:arg value='-set'/>
                <qemu:arg value='block.scsi0-0-1.discard_granularity=65536'/>
        </qemu:commandline>
</domain>
```

QEMU CACHE OPTIONS

QEMU's cache options correspond to the following Ceph RBD Cache settings.

Writeback:

```
rbd_cache = true
```

Writethrough:

```
rbd_cache = true
rbd_cache_max_dirty = 0
```

None:

```
rbd_cache = false
```

QEMU's cache settings override Ceph's default settings (i.e., settings that are not explicitly set in the Ceph configuration file). If you explicitly set RBD Cache settings in your Ceph configuration file, your Ceph settings override the QEMU cache settings. If you set cache settings on the QEMU command line, the QEMU command line settings override the Ceph configuration file settings.