OPENSTACK® Cinder QoS Instructions

Prepared by: Walter Bentley, Rackspace

These instructions are intended to provide step-by-step instructions to create one type of QoS specs using new Cinder functionality. The set of instructions are based on the OpenStack Juno release and may need revision for future OpenStack releases.

Prerequisite Configurations

Configure and enable multiple-storage back ends
To enable a multiple-storage back ends, you must set the enabled_backends flag in the cinder.conf file. Below is an example of the block to add to the cinder.conf file.

```
enabled_backends=ssd,sata
[ssd]
volume_driver=cinder.volume.drivers.lvm.LVMISCSIDriver
volume_backend_name=LVM_iSCSI
volume_group=cinder-volumes

[sata]
volume_driver=cinder.volume.drivers.lvm.LVMISCSIDriver
volume_backend_name=LVM_iSCSI_2
volume_group=cinder-volumes2
```

Using the OpenStack CLI

Create volume type
```
$ cinder type-create <volume type name>

$ cinder type-key <volume type name> set volume_backend_name=<backend name>
```

Working examples below:
```
$ cinder type-create ssd_high-iops
$ cinder type-key ssd_high-iops set volume_backend_name=LVM_iSCSI
```

**Take note of the ID of the volume type just created.*

Create QoS for volume type
```
$ cinder qos-create <QoS name> consumer="front-end" total_iops_sec=<iops>
```

Associate QoS with the volume type
```
$ cinder qos-associate <QoS specs> <volume type ID>
```

Create new volume
```
$ cinder create <size> --volume-type <volume type name> --display-name <display-name>
```

Boot an instance to attach volume to
```
$ nova boot --image <image> --flavor <flavor> <instance name>
```

Attach volume to instance after instance is active, and volume is available
```
$ nova volume-attach <instance name> <display-name>
```

Test QoS Specs

List storage devices
```
$ sudo fdisk -l
```

Calculate I/O on volume
```
$ sudo hdparm -t <storage device name>
```