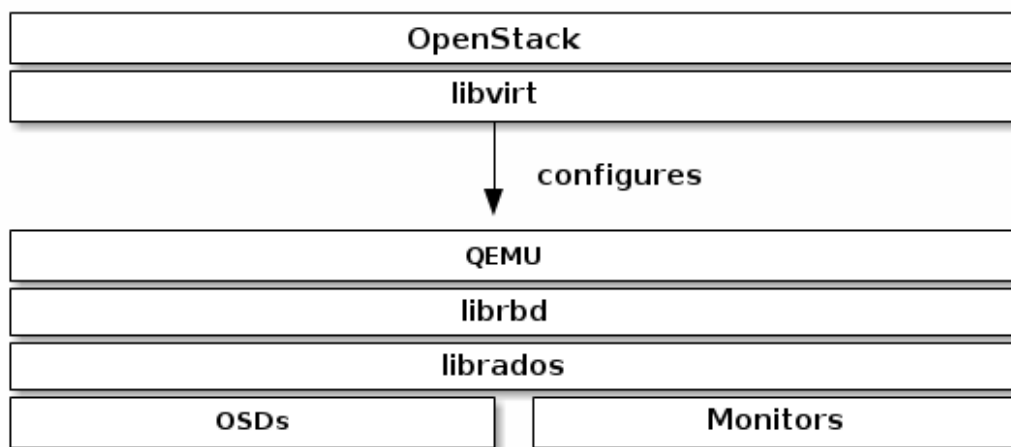


Ceph Block Devices & OpenStack

You may use Ceph Block Device images with OpenStack through libvirt, which configures the QEMU interface to librbd. Ceph stripes block device images as objects across the cluster, which means that large Ceph Block Device images have better performance than a standalone server!

To use Ceph Block Devices with OpenStack, you must install QEMU, libvirt, and OpenStack first. The following diagram depicts the OpenStack/Ceph technology stack.



To use the Ceph as the backend storage of OpenStack, you have to do some tasks at both ends, Ceph as well as OpenStack.

On Ceph end :

- Create respective pools

- Create respective pool users and authentication key-rings

On OpenStack end :

- Install required Python bindings

- Download and save respective Key-ring files and ceph.conf file

- Change ownership of key-ring file

- Make configuration changes to OpenStack to integrate with Ceph

- Restart respective OpenStack services

Configure Ceph –

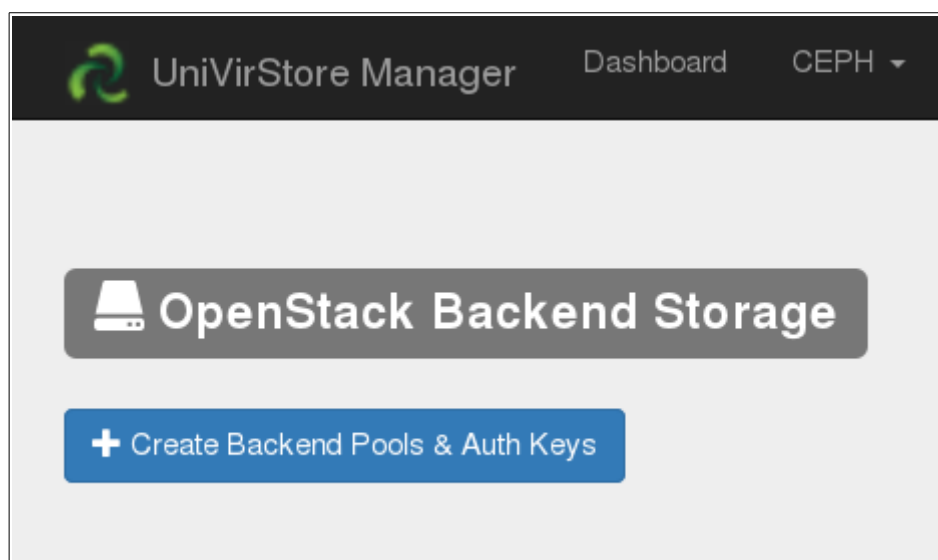
Ambedded Mars 200 UniVirStor Manager helps you to create pools, bind users to pool and create authentication key-rings for OpenStack API server nodes. It simplifies the procedure at Ceph side. You just need to configure the OpenStack by CLI.

- Create the pools

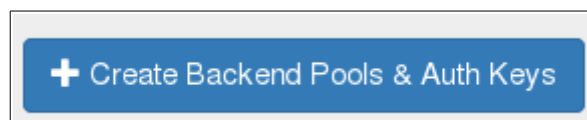
From Mars 200 UniVerStor Manager menu bar, click Openstack.



You will get to Openstack Backend Storage page.



Click Create Backend Pools and & Auth Keys button to create Pools and respective users along with Keyring.



```

pool '.glance' created
Exit status : 0
[client.glance]
    key = AQAJJYNXTR0QJxAACmNRQ3MjYy6ZR3AdWQDf+Q==
Exit status : 0
pool '.cinder' created
Exit status : 0
[client.cinder]

```

Once the process is complete, you will be presented with the page showing Ceph Users for Cinder, Glance and Nova along with keys and Download button to download the keys.

OpenStack Backend Storage				
<div>Search</div> <div> </div>				
OpenStack Component	Pool Name	CephX User Name	Key	Action
CINDER	.cinder	cinder	AQALJYNXNcD1LhAA9zwJt29mHEhB6Ra+a9sSw==	Download Key
GLANCE	.glance	glance	AQAJJYNXTR0QJxAACmNRQ3MjYy6ZR3AdWQDf+Q==	Download Key
NOVA	.nova	nova	AQANJYNX1dHLOBAAOsmMgUHb+JytSnJT0rlywA==	Download Key

OpenStack Component	Pool Name	CephX User Name	Key
CINDER	.cinder	cinder	AQALJYNXNcD1LhAA9zwJt29mHEhB6Ra+a9sSw==

Configure OpenStack –

- *Install Ceph Client Packages*

For Openstack to interact with Ceph cluster, it need Python bindings for librbd.

For **Glance** to connect to Ceph cluster, on the glance-api node, you'll need the Python bindings for librbd:

```
$sudo apt-get install python-rbd - Debian based
```

```
$sudo yum install python-rbd - RPM based
```

For **Cinder and Nova**, use both the Python bindings and the client command line tools:


```
$sudo apt-get install ceph-common - Debian based
```

```
$sudo yum install ceph-common - RPM based
```

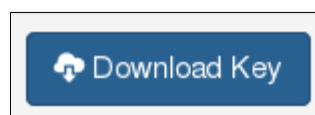
- *Download and save Keyring and Ceph.conf file –*

Once the required Python client packages are installed, you need to generate and copy the Ceph authentication key-ring and Ceph configuration file; ceph.conf to Openstack server.

Mars 200 UVS Manager help you to generate the Ceph client keyring for client.glance, client.cinder and client.nova and also the ceph.conf file, which can be downloaded and copied to the Openstack.

CephX User Name	Key	Action
cinder	AQALJYNXNcD1LhAA9zwJt29mHEhB6Ra+a9sSw==	

Click Download Key button against each user to download the compressed file, which contains the respective Ceph user authentication keyring along with the ceph.conf file, which needs to be copied to **/etc/ceph/** folder on Openstack node.



– Change Ownership of keyring files

Once the keys are copied to Openstack, all you need to do is change the Ownership to respective Openstack service users.

Glance

```
#ssh {your-glance-api-server} chown glance:glance /etc/ceph/ceph.client.glance.keyring
```

Cinder

```
#ssh {your-cinder-volume-server} chown cinder:cinder /etc/ceph/ceph.client.cinder.keyring
```

Nova

```
#ssh {your-nova-compute-server} sudo nova:nova /etc/ceph/ceph.client.nova.keyring
```

Nova & Cinder also need to store the secret key of the **client.nova** & **client.cinder** user in libvirt. The libvirt process needs it to access the cluster while attaching a block device from Cinder.

Create a temporary copy of the secret key on the nodes running Cinder and Nova-compute as the case be:

From the extracted file, open `ceph.client.{cinder|nova}.keyring` file. It will have content as shown below .(the actual content might vary depending on key.)

[client.cinder]

key = AQALJYNXNcD1LhAA9zwUj29mHEhB6Ra+a9sSw==

Copy 'AQALJYNXNcD1LhAA9zwUj29mHEhB6Ra+a9sSw==' section from keyring and paste it in a blank file named `client.cinder.key` and save the file.

Similarly for Nova, extract client.nova key and save the key to `client.nova.key` file.

– Configure Openstack Configuration files

Then, on the Cinder volume nodes and Nova Compute nodes, add the secret key to libvirt and remove the temporary copy of the key.

```
uuidgen
457eb676-33da-42ec-9a8c-9293d545c337

cat > secret.xml <<EOF
<secret ephemeral='no' private='no'>
  <uuid>457eb676-33da-42ec-9a8c-9293d545c337</uuid>
  <usage type='ceph'>
    <name>client.cinder secret</name>
  </usage>
</secret>
EOF
sudo virsh secret-define --file secret.xml
Secret 457eb676-33da-42ec-9a8c-9293d545c337 created
sudo virsh secret-set-value --secret 457eb676-33da-42ec-9a8c-9293d545c337
--base64 $(cat client.cinder.key) && rm client.cinder.key secret.xml
```

Save the uuid of the secret for configuring nova-compute later

Important: You don't necessary need the UUID on all the computer nodes. However from a platform consistency prospective, it's better to keep the same UUID.

Restart Opentack service and connect to Ceph cluster to use Ceph as backend to Glance, Cinder and Nova services.

Detailed information about configuration of Openstack with Ceph can be found at :

<http://docs.ceph.com/docs/master/rbd/rbd-openstack/>