Archiving OpenStack cloud volumes

August 2015

Author:

Jakub Kvita

Supervisor(s):

Sebastian Bukowiec

CERN openlab Summer Student Report 2015

Project Specification

CERN runs a large scale OpenStack based cloud on over 4,200 hypervisors to provide computing resources for users on demand. Critical application data can be stored on reliable disk volumes.

The project would involve investigation of solutions for long term archiving of contents and defining the necessary large scale deployment tools using Puppet along with performance tuning for production usage.

The student will gain understanding of cloud technologies, experience with OpenStack and Puppet along with storage technologies such as Ceph and TSM.

The work would also involve collaboration with large open source communities.

Abstract

This report is a summary of solutions for archivation of Cinder volumes and user-driven creation of virtual volumes backups. Solutions are based and evaluated on current CERN private cloud status and currently used environments and tools.

The explored solutions include use of TSM - tape storage and two different clusters of Ceph in two different geographical locations(Meyrin and Wigner).

Table of Contents

Introduction

Through the last years, cloud computing became popular with use across a lot of different application fields. This is caused by some of the features including better resource management and administration than different ways of computing. At the beginning cloud computing was mainly focused on providing computing power for users, but then it evolved and include delivery of all kinds of computer systems from volumes and databases to computer networks. See Figure 1. showing different types of cloud computing. Nowadays cloud computing goes hand to hand with close activities like software deployment and automation with emphasis on elasticity, scalability and usability.

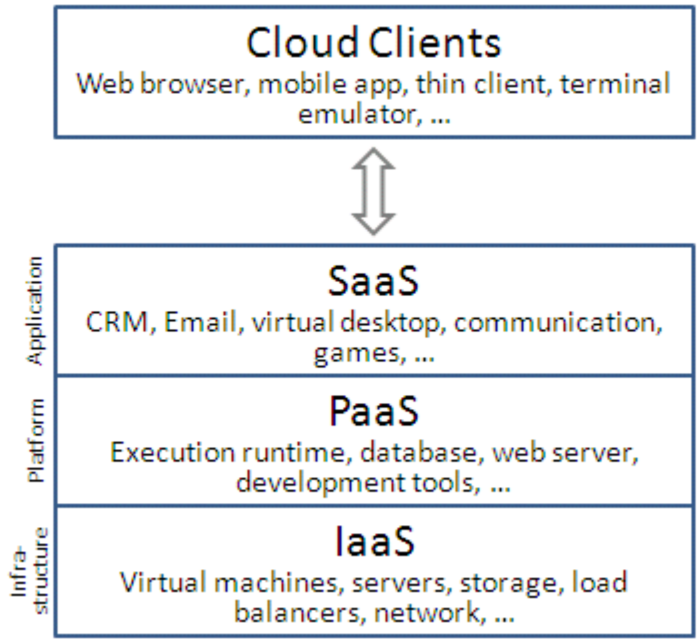
Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
Application
SaaS
CRM, Email, virtual desktop, communication, games, ...
Platform
PaaS
Execution runtime, database, web server, development tools, ...
Infra-structure
IaaS
Virtual machines, servers, storage, load balancers, network, ...

Figure 1. Different service models of cloud computing.

At CERN we are shifting towards using cloud computing with our own cloud offered to our users. There are several services provided for users including tools for private Infrastructure-as-a-Service, which is mainly served by OpenStack and its projects like Cinder, and other big data tools used apart from different purposes as backend for our OpenStack cloud.

In following chapters I will explain how I created my own working environment with small cloud, how to connect backend for virtual volume creation and how to provide Cinder backup feature for users.

OpenStack and Cinder

OpenStack[1] is one of the most used cloud computing platforms, which provides Infrastructure-as-a-Service. OpenStack consists of many projects, each one of them focused on different part of cloud management and different services. Among projects considered as core are Nova, directly working with virtual machines, Keystone, which provides authentication, Glance, dealing with images, Cinder, used for virtual volumes and some others.

In my project I focused mainly on work with Cinder, which, as mentioned earlier, is an OpenStack component for management of block storage, which can serve as persistent data storage for critical application data volume. We can connect this volume to virtual machines, detach them and move them across multiple machines in same way as USB key. If machine with attached volume fails, we lost all the data saved on it, but data stored on cinder volumes is untouched as it is not stored locally but on the network attached volume.

As Openstack consists of multiple projects connected together, its installation can be time consuming. Therefore there were created tools to package it and make it easier to deploy. One of the most used ones are Packstack[2], for deploying Openstack on Red Hat Enterprise Linux, Fedora and distributions derived from these (such as CentOS, Scientific Linux and others). Other one is Devstack[3]. Packstack is perfect for developing, testing and evaluation of setups before deploying them in production environment.

To install Packstack-based test environment inside CERN cloud I created and used following procedure:

1. Use large flavor when provisioning virtual machine in CERN cloud, because with the others creation of VMs inside won't be possible. I also used CC7 Extra image for the VM that was hosting my packstack as CC7 Base was causing many issues with volume attachment. However the time spent on debugging issues with CC7 base image allowed me to understand better Cinder and Nova components.

2. Once inside of VM, disable NetworkManager to prevent conflicts with nova-network.

```
systemctl stop NetworkManager
systemctl disable NetworkManager
systemctl enable network
```

3. Turn off base protection in yum. Some of CERN repositores are protected and are installing wrong packages.

```
sed -i 's/1/0/g' /etc/yum/pluginconf.d/protectbase.conf
```

4. Update, install Packstack and run it to install Openstack with nova-network.

```
sudo yum update -y
sudo yum install -y openstack-packstack
packstack --allinone --os-neutron-install=n
sudo yum install -y https://rdoproject.org/repos/rdo-release.rpm
sudo yum update -y
```

5. Apply the fix on nova-network to prevent getting exceptions, while running. Restart service afterwards.

```
sed -i 's/if interface:/if interface and interface !='\''lo'\'':/'
/usr/lib/python2.7/site-packages/nova/network/linux_net.py
systemctl restart openstack-nova-network
```

Before using OpenStack services it is necessary to create credentials variables for user through this command:

```
. ~/keystonerc_admin
```

In Horizon file can be downloaded from *Access & Security -> API Access -> Download OpenStack RC File*.

Status of all the services can be listed by:

```
# openstack-status

== Nova services ==
openstack-nova-api:                    active
openstack-nova-cert:                   active
openstack-nova-compute:                active
openstack-nova-network:                active
openstack-nova-scheduler:              active
openstack-nova-conductor:              active
== Glance services ==
openstack-glance-api:                  active
openstack-glance-registry:             active
== Keystone service ==
openstack-keystone:                    active    (disabled on boot)
== Horizon service ==
openstack-dashboard:                   active
== Swift services ==
openstack-swift-proxy:                 active
openstack-swift-account:               active
openstack-swift-container:             active
openstack-swift-object:                active
== Cinder services ==
openstack-cinder-api:                  active
openstack-cinder-scheduler:            active
openstack-cinder-volume:               active
openstack-cinder-backup:               active
...
```

Ceph backend

Ceph[4] is a distributed object store and file system designed to provide excellent performance, reliability and scalability. Currently we have two Ceph clusters – one in Meyrin and second in Wigner.

Meyrin cluster is older one and is currently used for storing data from CERN cloud like Glance images, Cinder volumes,… Its size is almost 4PB with close to 1 PB already used. At Figure 2. is Meyrin cluster dashboard, where is total storage used, size of Glance images and Cinder volumes, etc...

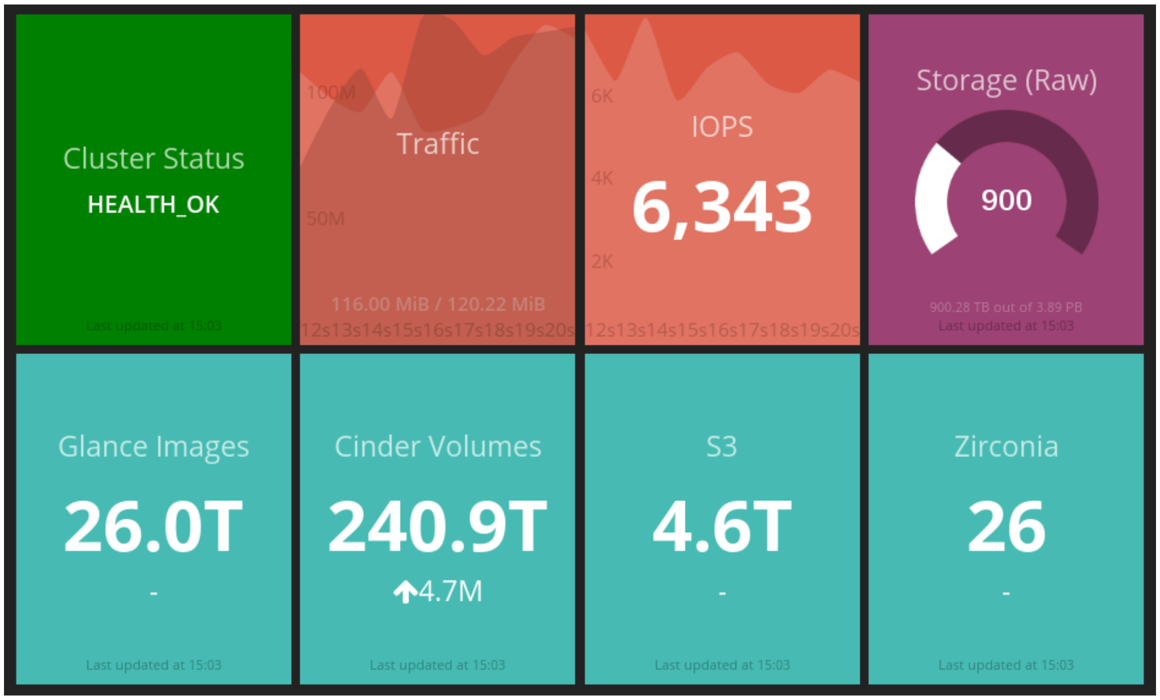

Figure 2. Meyrin cluster dashboard.

Second cluster in Wigner, Hungary is smaller with size approximately 430 TB and 150 TB usable for data storage. This cluster haven't been used yet and I was the first person who stored volumes there.

Ceph cluster serves as Cinder backend and all created volumes are in reality located there. This also mean that is necessary to configure Cinder and other components to use Ceph before we start actually working. In my setup I was using dedicated test pool of the production Ceph cluster. To configure OpenStack CERN clients to use Ceph, request access to the cluster you need and then follow these instructions:

1. Check if Epel repositories are enabled. Installing packstack is turning them off and it is necessary to have them on.

```
vim /etc/yum.repos.d/epel.repo
```

2. Install Ceph packages.

```
yum install -y libvirt python-rbd ceph-common
```

3. Create configuration files for Ceph with key you received. One global config file with reference to users config file.

```
cat > "/etc/ceph/ceph.conf" <<EOF
[global]
    mon host = <ceph_monitor>.cern.ch:<port>
[client.testing-volumes]
    log file = /var/tmp/ceph-client.\$pid.log
    keyring = /etc/ceph/ceph.client.testing.keyring
EOF
```

```
cat > /etc/ceph/ceph.client.testing.keyring <<EOF
[client.testing]
    key = INSERT_KEY_HERE
EOF
```

4. Edit Cinder config file to add Ceph backend and create category with same name and configuration for it.

```
sed -i '/^enabled_backends=/ s/$/,ceph/' /etc/cinder/cinder.conf

cat >> /etc/cinder/cinder.conf <<EOF

[ceph]
volume_driver=cinder.volume.drivers.rbd.RBDDriver
rbd_user=testing
rbd_pool=testing
volume_backend_name=ceph
rbd_ceph_conf=/etc/ceph/ceph.conf
rbd_flatten_volume_from_snapshot=false
rbd_max_clone_depth=5
rbd_store_chunk_size=4
rados_connect_timeout=-1
glance_api_version=2
rbd_secret_uuid=12345678-1234-1234-1234-000000000001
EOF
```

5. Create virsh secret for mounting volumes in Nova. Use same key as before.

```
cat > secret.xml <<EOF
<secret ephemeral='no' private='no'>
  <uuid>12345678-1234-1234-1234-000000000001</uuid>
  <usage type='ceph'>
    <name>client.testing secret</name>
  </usage>
</secret>
EOF
sudo virsh secret-define --file secret.xml
sudo   virsh   secret-set-value   –secret   12345678-1234-1234-1234-
000000000001 --base64 INSERT_KEY_HERE
```

6. Create new volume type in Cinder. Property volume_backend_name has to has same value as backend in Cinder configuration file.

```
cinder type-create ceph
cinder type-key ceph set volume_backend_name=ceph
```

7. Restart services to reload configuration files.

```
openstack-service restart
```

Now you can create Cinder volumes in Ceph and attach them in virtual machines in Nova. Example follows:

1. Create VM.

```
nova boot --image cirros --flavor m1.tiny cir1
```

2. Create Cinder volume at Ceph.

```
cinder create --volume-type ceph --display-name volume1 1
```

3. Attach volume to Nova.

   ```
   nova volume-attach cir1 volume1 auto
   ```

Cinder backups

One of the features of Cinder I was interested in is backup. Since Havana release backups were added to Cinder as tool to save state of volume and put it in different backend than actual volume is. This means that first of all, backup backend has to be configured and then, when backup is created, it is based on some volume which is then referenced as source. Incremental backups can be also created, as backup refer to each other, but this is supported only for several backends.

Common use of backups is this:

1. Create backup of available volume.

   ```
   cinder backup-create volume1
   ```

2. List backups.

   ```
   cinder backup-list
   ```

3. Restore volume from a backup.

   ```
   cinder backup-restore [--volume-id <volume>] <backup>
   ```

These commands can be used once we set backend for backups. Possible solutions for CERN usage are TSM, which is already used for storing data and Ceph. I explored both possibilities and included results in rest of this chapter.

TSM - Tivoli Storage Manager

Tivoli Storage Manager[5] is IBM product, which is a data protection platform that gives enterprises a single point of control and administration for backup and recovery. At CERN it is used together with tapes to create large storage for LHC generated data.

There is official support of TSM for OpenStack from IBM [6], with driver for Cinder backend, which is delivered together with OpenStack installation. To enable TSM as Cinder backup backend, use following recipe. Be aware that if you use Ceph as backend for your volumes, backups won't work, as it is discussed later in this chapter.

1. Follow instructions in [7] to get access to tape backups at CERN.

2. Download archive with installation files(rpms) with support for your system. Support list is in [8]. Software in [9]. For 64 bit version of CentOS 7 use these commands:

   ```
   yum install -y wget
   wget                ftp://ftp.software.ibm.com/storage/tivoli-storage-
   management/maintenance/client/v7r1/Linux/LinuxX86/BA/v712/7.1.2.0-
   TIV-TSMBAC-LinuxX86.tar
   ```

3. Untar and install necessary rmps.

   ```
   tar -xf 7.1.2.0-TIV-TSMBAC-LinuxX86.tar

   rpm      -U      gskcrypt64-8.0.50.40.linux.x86_64.rpm      gskssl64-
   8.0.50.40.linux.x86_64.rpm
   ```

```
rpm -i TIVsm-API64.x86_64.rpm
rpm -i TIVsm-BA.x86_64.rpm
```

4. Configure TSM client following [10].

5. Create configuration files from the samples included.

```
cd /opt/tivoli/tsm/client/ba/bin/
cp dsm.opt.smp dsm.opt
cp dsm.sys.smp dsm.sys
```

6. Open configuration files and append your settings. Values should be delivered to you when you requested access to TSM.

```
# cat dsm.sys
SErvername  TSM516
   COMMMethod         TCPip
   TCPPort            <port>
   TCPServeraddress   <tsm_node>.cern.ch
   NODename           jakubkvitatest
   errorLogName       /var/log/tsm/dsmerror.log

# cat dsm.opt
SErvername <TSM_NODE>
```

7. Create log file with proper access rights.

```
mkdir /var/log/tsm
touch /var/log/tsm/dsmerror.log
chmod 666 /var/log/tsm/dsmerror.log
```

8. Finally set up cinder backup in /etc/cinder/cinder.conf.

```
# Options defined in cinder.backup.drivers.tsm
#

# Volume prefix for the backup id when backing up to TSM
# (string value)
backup_tsm_volume_prefix=backup

# TSM password for the running username (string value)
backup_tsm_password=<password>

# Enable or Disable compression for backups (boolean value)
backup_tsm_compression=true


#
# Options defined in cinder.backup.manager
#

# Driver to use for backups. (string value)
# Deprecated group/name - [DEFAULT]/backup_service
#backup_driver=cinder.backup.drivers.swift
backup_driver=cinder.backup.drivers.tsm
```

9. Restart OpenStack services to reload configuration files

```
openstack-service restart
```

At this point we can create backups of volumes which are not stored at Ceph. Despite the fact that IBM provides TSM driver, communication betweend TSM and Ceph is not possible. This means that backups of Ceph volumes to TSM are not possible and according to the bug [12] reported by other people working on same solution it is a known issue for Red Hat. After discussion with developers (Jon Bernard), we found out that this could be supported by rewriting Cinder drivers, but they are not very sure, if it will happen. Even if it happen, it was not possible to release it before end of my project.

Ceph

Ceph, explained earlier at Ceph backend chapter, can be used also as a backend to Cinder backups, not just for volumes. We already know that connection to Cinder works and adding it as a backup backend is essentially similar. At CERN we have two clusters, one at Geneva and second at Wigner, Hungary, and I created testing environment with both of them connected, one for volumes and other one for backups.

To have working environment, first request access to the Wigner Ceph cluster and then with login and password follow these instructions.

1. Create second Ceph cluster user configuration file.

```
cat > "/etc/ceph/ceph.client.test-wigner.keyring" <<EOF
[client.test]
        key = INSERT_KEY_HERE
EOF
```

2. Create second config file for Ceph with following settings. Do not change/delete configuration file of already configured Ceph cluster.

```
cat > "/etc/ceph/ceph-backup.conf" <<EOF
[global]
    mon host = <mon_host_01>.cern.ch, <mon_host_02>.cern.ch

[client.test]
    log file = /var/tmp/ceph-client-wigner.$pid.log
    keyring = /etc/ceph/ceph.client.test-wigner.keyring
EOF
```

3. Edit /etc/cinder/cinder.conf with new backup backend.

```
backup_driver = cinder.backup.drivers.ceph
backup_ceph_conf=/etc/ceph/ceph-backup.conf
backup_ceph_user=test
backup_ceph_chunk_size=134217728
backup_ceph_pool=test
```

4. Restart OpenStack services to reload configuration files.

```
openstack-service restart
```

This configuration should enable backups in Cinder.

While working on this configuration I found out couple important things in how are these backups implemented and designed in Cinder.

- To create a backup, status of the volume has to be 'available'. This means volume has to be detached from virtual machine, which is not very efficient, while working. It is possible to create snapshot of attached volume, then created new volume from it, with same data in it and backup this one, but that creates unwanted meta data of the backup. Users has to be familiar with the tools to manage it properly.
- Cinder provides quotas for backups - both maximum number of backups created for user and maximum size of backups in gigabytes. Supported since Kilo release of OpenStack.
- Creation of backup in Cinder means that data from volume are streamed from the backend to Cinder node, processed in Cinder and then sent to the backup backend. This solution is not scalable and if multiple users are creating backups at same time, network bandwidth is all used just for backups, which has to be processed whole, if incremental backups are not enabled.

Conclusion

In this report I described investigation of several solutions to providing users possibility to create Cinder volumes backup. One of the solutions - TSM is not possible to use out of the box without writing any hooks, due to missing support of Ceph and TSM together at Cinder. However it is still possible to install TSM client inside the virtual machine and user could archive data to TSM with it. Second solution - Ceph cluster at Wigner, is working and can be used together with cluster at Geneva, where volumes are stored.

The second solution also requires more investigation of properties of Cinder backups. For example internal implementation of backup creation is not scalable and therefore not acceptable for usage in CERN environment. As far as actual procedure of backups from users point of view is concerned, we need to discuss more backing up volumes which are attached to running virtual machines and probably request some support of that at developer teams. Also as backup quotas were introduced in latest release, more research of what is actually working is necessary.

Most of the tools and software involved in this use case is actually under development and are not yet implemented properly. This probably means that code and interface is going to change and we should monitor development closely.

References

[1] https://www.openstack.org/

[2] https://www.rdoproject.org/

[3] http://docs.openstack.org/developer/devstack/

[4] http://ceph.com/

[5] http://www.ibm.com/software/products/cs/tivostormana

[6] https://www.ibm.com/developerworks/community/files/basic/anonymous/api/library/4a5b0e43-b165-49c7-ae33-b1480e6840cb/document/18c7869c-7941-4af4-853c-cc4840e41774/media

[7] https://twiki.cern.ch/twiki/bin/view/BackupService/BackUserGuide

[8] ftp://ftp.software.ibm.com/storage/tivoli-storage-management/maintenance/client/v7r1/Linux/LinuxX86/BA/v712/

[9] http://www-01.ibm.com/support/docview.wss?uid=swg21417165

[10] http://www-01.ibm.com/support/knowledgecenter/SSGSG7_7.1.2/com.ibm.itsm.client.doc/t_cfg_crtmodoptunix.html%23t_cfg_crtmodoptunix

[11] https://bugs.launchpad.net/cinder/+bug/1446644