Lab – The Plan

Welcome to **Ceph Cluster Setup**, the first module in the Red Hat Ceph Storage Test Drive.

In this lab we will:

- Install and setup a Red Hat Ceph Storage cluster
- Verify the cluster is up and running using the Calamari
- Verify the cluster is up and running using the CLI
- Create a block device
- Mount the block device
- Write and read data to/from the block device
AGENDA

• Register for Red Hat Storage Test Drive
• Launch Lab: Ceph Cluster Setup
• AWS EC2 Lab Deployment Architecture Overview
• Accessing Servers
• Install Red Hat Ceph Storage
• Setup Red Hat Ceph Storage
• Create Ceph Cluster
• Add OSD
• Cluster Administration
• Verify cluster setup using Calamari
• Storing Object Data
• Block Device Access Using RBD
Register for Red Hat Storage Test Drive

Go to the URL: https://engage.redhat.com/aws-test-drive-201308271223

ANYWHERE, ALL-THE-TIME DATA ACCESS.
Open source cloud storage with built-in automatic data replication.

RED HAT STORAGE TEST DRIVE.
The Red Hat Storage Test Drive lab on Amazon Web Services (AWS) provides you with a free hands-on experience with Red Hat Storage Server. You'll be able to explore the features and ease of use of the product in real-time. It's a quick and easy way to test functionality of the product whether you end up deploying on-premise or in the cloud.

- High availability and business continuity in AWS
- Secure enterprise file sync and share
- Large file and object storage
- Multi-media content delivery and storage

The Test Drive lab has been developed by Red Hat and AWS for educational and demonstration purposes.

Register for the test drive
Register to access the assets in our Resource Library and receive emails customized to your interests.

First name:

Last name:

Work email:
Once logged in you will see that courses available to you. In this Test Drive we are focusing on the **Ceph Training** modules. Select **Ceph Training** (highlighted) to get started.
Lab Launch Page

Select the Lab you wish to work on by selecting Start Lab button.

Important Information
- Please click on the Lab Name to get more details on the lab.
- Or
- Please click on the Start Lab button to start a lab
- Enter your Lab Access Code if requested.

Ceph Training

Class Description: This is a Ceph training class with modules covering the basic setup of a cluster, management and some of the advanced features.

Start Date: Monday, April 6, 2015 9:00AM EDT
End Date: Friday, July 31, 2015 12:00PM EDT
Location: US / EMEA
Trainer: veda shankar

<table>
<thead>
<tr>
<th>Class Labs</th>
<th>Duration (min.)</th>
<th>Start Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ceph Cluster Setup</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>Ceph Management I</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>Ceph Object Gateway and Ceph Pools</td>
<td>180</td>
<td></td>
</tr>
</tbody>
</table>

<< Return to My Classes
Lab - Initializing

At the bottom of the Lab details page you will see 4 tabs (highlighted).

**Lab Description:** Create a Ceph Storage Cluster with one management host, 3 monitor hosts and 4 OSD hosts. Test basic object and block mode access. Confirm the cluster status using Calamari UI.

**Lab Creator:** veda shankar

**Date Created:** June 15, 2015 00:50

**AWS Region:** [us-east-1] US East (N. Virginia)

*Create in Progress*...
Lab - Tabs

Each tab has specific information relating to the lab you are taking

**Ceph Concepts:** This tab contains detail on the components that make up a Ceph storage cluster

**Lab Instructions:** In this tab you will find instructions on how to set up and run the lab

**Connection:** The SSH key pairs can be found in this tab. Only needed if you have connection problems

**ADDL. INFO:** IP Addresses of the nodes in the lab
LAB - IP INFORMATION

Under **ADDL INFO** tab, you should see the IP addresses of all the nodes.

**Note:** The ManagementNodePublicIP in your lab will likely be different than appears here.
Lab - Concepts

Ceph Calamari: Calamari is a graphical management and monitoring system for Ceph storage cluster.

Ceph OSD (Object Store Device): Usually corresponding to a physical disk, an OSD is a directory on a regular filesystem that Ceph utilizes. Ceph OSDs store data on behalf of Ceph clients. Additionally, Ceph OSDs utilize the CPU and memory of Ceph nodes to perform data replication, rebalancing, recovery, monitoring and reporting functions.

Ceph Mon (Monitor): A Ceph monitor maintains a master copy of the Ceph storage cluster map with the current state of the storage cluster.

Ceph PG (Placement Groups): PGs are logical containers for objects and are assigned to OSDs. Each placement group is composed of a number of OSDs equal to either the number of replicas (in the case of replication) or the sum of K and M (in the case of erasure coding). A single OSD can have data from multiple placement groups.

Ceph Pools: A Pool is a logical partition for storing objects via the PGs assigned to a pool.

RADOS Block Device: A Ceph block device presents block storage that mounts like a physical storage drive for use by both physical and virtual systems (includes QEMU/KVM driver).

RADOS Object Storage Utility: Utility for interacting with the Ceph object storage cluster.

For more details see the Ceph Concepts guide located at the main lab page.
AWS resources launched as part of the Lab: Ceph Cluster Setup

CLUSTER NETWORK

OSD node1 10.100.1.11
  - SSD
  - HDD
  - HDD
OSD node2 10.100.1.12
  - SSD
  - HDD
  - HDD
OSD node3 10.100.1.13
  - SSD
  - HDD
  - HDD
OSD node4 10.100.1.14
  - SSD
  - HDD
  - HDD

PUBLIC NETWORK

mon1 10.100.2.11
mon2 10.100.2.12
mon3 10.100.2.13
client1 10.100.2.14

Administration / Calamari - mgmt
10.100.0.72/External IP

SSD
HDD
HDD
HDD
Logical view of Red Hat Ceph Storage cluster components

Client Interface Layer
- RADOSGW
- RBD

Objects in Pools
- Pool ID. (Hash/Object) %Num of PGs

CRUSH Ruleset
- CRUSH Map

Placement Groups
- objects in pools
- CRUSH map
- Ceph Nodes: OSD Hosts
  Monitors (mons)
Lab Setup

This is the reference host file for this lab

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4 ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
#
10.100.0.72 mgmt.ec2.internal mgmt ip-10-100-0-72.ec2.internal ip-10-100-0-72
#
10.100.1.11 node1.ec2.internal node1 ip-10-100-1-11.ec2.internal ip-10-100-1-11
10.100.1.12 node2.ec2.internal node2 ip-10-100-1-12.ec2.internal ip-10-100-1-12
10.100.1.13 node3.ec2.internal node3 ip-10-100-1-13.ec2.internal ip-10-100-1-13
10.100.1.14 node4.ec2.internal node4 ip-10-100-1-14.ec2.internal ip-10-100-1-14
#
10.100.2.11 mon1.ec2.internal mon1 ip-10-100-2-11.ec2.internal ip-10-100-2-11
10.100.2.12 mon2.ec2.internal mon2 ip-10-100-2-12.ec2.internal ip-10-100-2-12
10.100.2.13 mon3.ec2.internal mon3 ip-10-100-2-13.ec2.internal ip-10-100-2-13
#
10.100.2.14 client1.ec2.internal client1 ip-10-100-2-14.ec2.internal ip-10-100-2-14

A note about networking
This lab uses a very simple networking schema to better facilitate the learning activities. Your corporate networking setup will likely differ. Make sure the Admin node has connectivity to the server nodes (Mons, OSD’s).
**Lab Setup**

Use your SSH client of choice to access the management node via the public ip address (found at the bottom of the ADDL. INFO tab on the launch page):

```bash
ssh ceph@ManagementNodePublicIP
password = Redhat12
```

Example: `ssh ceph@54.88.26.3`

When prompted “Are you sure you want to continue connecting (yes/no)?” Enter yes

You can login to each individual node using:

- **User** – `ceph` (or root)
- **Password** – Redhat12

Note: If your SSH connection fails, do the following

- Select the **CONNECTION** tab from the main lab page
- Select the **Download PEM/PPK** button
- Select **Download PEM** (note the download location of the PEM file)
- Change to the location of the downloaded PEM file and execute the following:
  ```bash
  ssh -i name-of-file.pem ceph@ip-address-of-ManagementNodePublicIP
  ```

Example: `ssh -i qwikLABS-L12-1652.pem ceph@54.88.26.3` (your keypair and IP will differ)
Install Red Hat Ceph Storage

Install the ICE setup script:

Create the directory from which we will launch the installation

```
[ceph@mgmt ~]# sudo mkdir /mnt/ceph
```

Now we mount the .iso image file (previously downloaded for this lab)

```
[ceph@mgmt ~]# sudo mount /root/ICE-1.3/rhceph-1.3-rhel-7-x86_64-dvd.iso /mnt/ceph/
```

Launch the Red Hat Ceph Storage installer

```
[ceph@mgmt ~]# sudo yum install /mnt/ceph/Installer/ice_setup-* .rpm
```

Answer Y when prompted:

```
Install 1 Package
Total size: 132 k
Is this ok [y/d/n]: y
```

Look for Complete! upon successful installation.
Setup Red Hat Ceph Storage

Create the main Ceph setup directory

```
[ceph@mgmt ~]$ mkdir ~/ceph-config
[ceph@mgmt ~]$ cd ~/ceph-config
```

Launch the Ceph setup script

```
[ceph@mgmt ceph-config]$ sudo ice_setup -d /mnt/ceph/
```

Enter y at “do you want to continue?”

Press Enter to accept the remaining default options

**Note:** Make sure you are in the [ceph@mgmt ceph-config]$ directory when running Ceph commands

The setup script performs the following operations:
- Moves the RPMs to /opt/ICE and /opt/calamari
- Creates a .repo file for the ceph-deploy and calamari packages pointing to a local path
- Installs the Calamari server packages on the admin node
- Installs the ceph-deploy package on the admin node; and
- Writes a cephdeploy.conf file to /opt/ICE.
Create Red Hat Ceph Storage Cluster

Initialize Calamari monitoring. We will use the Ceph Calamari management interface to initialize and verify the cluster.

From the mgmt node as user ceph:

```
[ceph@mgmt ceph-config]$ sudo calamari-ctl initialize
```

Make note of the Calamari admin user you setup during initialization. You will use this user to log in to the Calamari UI. Leave the Email Address: section blank.

Use the ceph-deploy command to create the Ceph cluster:

Add the initial Monitors to establish the Ceph cluster

**Note:** During this portion of the cluster setup, enter **yes** when prompted

```
[ceph@mgmt ceph-config]$ ceph-deploy new mon1 mon2 mon3
[ceph@mgmt ceph-config]$ ceph-deploy install --repo --release=ceph-mon mon1 mon2 mon3
[ceph@mgmt ceph-config]$ ceph-deploy install --mon mon1 mon2 mon3
```

Install Ceph on the OSD nodes in the cluster

```
[ceph@mgmt ceph-config]$ ceph-deploy install --repo --release=ceph-osd node1 node2 node3 node4
[ceph@mgmt ceph-config]$ ceph-deploy install --osd node1 node2 node3 node4
```

Add the initial Monitor node

```
[ceph@mgmt ceph-config]$ ceph-deploy mon create-initial
```

Once we have added the monitors, we need to connect them to Calamari:

```
[ceph@mgmt ceph-config]$ ceph-deploy calamari connect --master mgmt.ec2.internal mon1 mon2 mon3
[ceph@mgmt ceph-config]$ ceph-deploy calamari connect --master mgmt.ec2.internal node1 node2 node3 node4
```
Add OSD

Use the ceph-deploy command to add three OSDs to the Ceph cluster. Use the list command to see what disk drives are available for each node. The disk lettering scheme is dynamic and may vary from node to node.

From the mgmt node as user ceph:
List the disks available to act as OSD devices

```
[ceph@mgmt ceph-config]$ ceph-deploy disk list node1
```

Configure the “xvdg” drive on each node to use as an OSD

```
[ceph@mgmt ceph-config]$ ceph-deploy osd --zap create node1:xvdg
[ceph@mgmt ceph-config]$ ceph-deploy disk list node2
[ceph@mgmt ceph-config]$ ceph-deploy osd --zap create node2:xvdg
[ceph@mgmt ceph-config]$ ceph-deploy disk list node3
[ceph@mgmt ceph-config]$ ceph-deploy osd --zap create node3:xvdg
```

Notes:
In this lab we are creating one OSD per host for simplicity. In a real world cluster we would configure multiple OSD’s per host.
Refer back to the AWS resources slide for a listing of disks available to each host.
Cluster Administration

Make the mgmt host an administration client for the Ceph cluster.

From the mgmt node as user ceph:

```
[ceph@mgmt ceph-config]$ ceph-deploy install --repo --release=ceph-mon mgmt
[ceph@mgmt ceph-config]$ ceph-deploy install --cli mgmt
[ceph@mgmt ceph-config]$ ceph-deploy admin mgmt
[ceph@mgmt ceph-config]$ sudo chmod +r /etc/ceph/ceph.client.admin.keyring
```

Type the following administration commands to check the status of the cluster, listing of the OSDs & pools and the status of quorum among the monitors.

```
[ceph@mgmt ceph-config]$ ceph health
Three possible values: HEALTH_OK, HEALTH_WARN, and HEALTH_ERR
[ceph@mgmt ceph-config]$ ceph --w
A more concise view of what is occurring in the cluster (use ctrl-c to end)
[ceph@mgmt ceph-config]$ ceph osd tree
A tree listing of the OSD’s and their status
[ceph@mgmt ceph-config]$ ceph osd lspools
Displays the pools configured in the cluster
[ceph@mgmt ceph-config]$ ceph quorum_status --format json-pretty
Displays the quorum status of the Monitors
```

Note: Try the previous “quorum_status” command without the “json-pretty” formatting and compare the two
Calamari – Check Cluster Status

Using a browser go to the URL of the management node. http://ManagementNodePublicIP

Use the credentials setup during the `calamari-ctl initialize` command (root/Redhat12).
Verify Lab - Add Hosts to Calamari

Select the **Add** button (highlighted) to establish the Ceph cluster and setup the Monitors.

Wait for the Cluster to finish initializing then click Close when prompted.
Calamari – Check Cluster Status

This is the view presented when the Dashboard option is selected (highlighted).

Note: As this is a lab environment there will occasionally be errors reported in the cluster. You can view the errors by selecting the Manage icon, then Logs. These errors can be ignored.

[ceph@mgmt ~] $ sudo salt '*' state.highstate
Calamari – Check Cluster Status

By selecting **Workbench** on the Calamari UI, you can check on the status of the OSDs.
Storing Object Data

Using the **RADOS** object storage utility, we will create an object on the Ceph cluster pool “data” using a test file.

First create a new pool “data” to test with
From the **mgmt** node as user **ceph**:

```
[ceph@mgmt ~]$ ceph osd pool create data 64 64
```

Create a small file
```
[ceph@mgmt ~]$ echo "hello world" > /tmp/testfile.txt
```

Use the **rados** utility to generate the object and place it in the pool **data**
```
[ceph@mgmt ~]$ rados put object-1 /tmp/testfile.txt --pool=data
```

Use the **rados** utility to view the object (object-1) in the pool **data**
```
[ceph@mgmt ~]$ rados -p data ls
```

Use the **ceph osd** command to view the object details
```
[ceph@mgmt ~]$ ceph osd map data object-1
```

You can remove the object using the command:
```
[ceph@mgmt ~]$ rados rm object-1 --pool=data
```
RED HAT CEPH STORAGE

Lab Guide – MySQL on KRBD
Lab – The Plan

Welcome to **MySQL on KRBD**, an optional module in the Red Hat Ceph Storage Test Drive.

In this lab we will:

- Install KRBD on a Ceph client machine
- Create a RBD volume to use to store InnoDB tables
- Configure Percona Server to use RBD backed storage
- Execute Sysbench load tests against MySQL on KRBD
Client1 Setup

It is assumed that the Ceph cluster is up and running in the lab environment, if that is not the case then that should be done first. If the environment has a cluster ready then you will begin by connecting to the client1 host from the mgmt node.

```
[ceph@mgmt ceph-config]$ ssh client1
```

The client node has an old version of Ceph installed, remove that before continuing:

```
[ceph@client1 ]$ sudo yum remove --y ceph
```

Next, return to the mgmt node, either by exiting or Ctrl-D. Once on the management host the Ceph repo and packages can be installed on client1:

```
[ceph@mgmt ceph-config]$ ceph-deploy install --repo --release=ceph-mon client1
[ceph@mgmt ceph-config]$ ceph-deploy install --cli client1
```

Install admin code on client1:

```
[ceph@mgmt ceph-config]$ ceph-deploy --overwrite-conf admin client1
```
RBD Setup

Now it is time to create a RBD volume to store InnoDB data.

```
[ceph@mgmt  ceph-config]$ rbd create mysql --size 10240
```

See info on the newly created volume:

```
[ceph@mgmt  ceph-config]$ rbd info mysql
```

Next, reconnect to client1 over SSH.

```
[ceph@mgmt  ceph-config]$ ssh client1
```

Now install the RBD kernel module and load it:

```
[ceph@client1]$ sudo rpm -ivh /root/ICE-1.2/kmod*.rpm
[ceph@client1]$ sudo modprobe rbd
```

Map it the volume to a device, create a filesystem on it, and mount:

```
[ceph@client1]$ sudo rbd map mysql
[ceph@client1]$ sudo mkfs.xfs /dev/rbd0
[ceph@client1]$ sudo mkdir /mnt/mysql
[ceph@client1]$ sudo mount /dev/rbd0 /mnt/mysql
```
MySQL Setup

First, stop MySQL:

[ceph@client1]$ sudo systemctl stop mysqld.service

Make sure MySQL is not running:

[ceph@client1]$ pgrep mysql

Copy the MySQL data directory to the new mount point:

[ceph@client1]$ sudo rsync -av /var/lib/mysql/ /mnt/mysql/

Change ownership to the mysql user and group:

[ceph@client1]$ sudo chown mysql:mysql /mnt/mysql
MySQL Setup Continued

Replace the MySQL configuration with one that uses the innodb_data volume mount:

```
[ceph@client1]$ sudo vi /etc/my.cnf
```

```
[client]
socket=/mnt/mysql/mysql.sock

[server]
table_open_cache = 512
thread_cache = 512
query_cache_size = 0
query_cache_type = 0
innodb_buffer_pool_size = 5120M
innodb_additional_mem_pool_size = 32M
innodb_log_file_size = 256M
innodb_log_buffer_size = 16M
innodb_flush_log_at_trx_commit = 1
innodb_lockwait_timeout = 50
innodb_doublewrite = 1
innodb_flush_method = O_DIRECT
innodb_thread_concurrency = 0
innodb_max_dirty_pages_pct = 80
datadir=/mnt/mysql
socket=/mnt/mysql/mysql.sock
```

Finally, start up MySQL, this make take a couple minutes:

```
[ceph@client1]$ sudo systemctl start mysqld.service
```
Preparing Sysbench

Set a few environmental variables for the tests:

```
[ceph@client1]$ TEST_DIR='/usr/share/doc/sysbench/tests/db/'
[ceph@client1]$ TEST=insert.lua
```

Create the sysbench test database:

```
[ceph@client1]$ sudo mysqladmin -u root create sbtest
```

Next, type out the sysbench to populate the database with inserts, note that copy/paste does not work well here:

```
[ceph@client1]$ sysbench
   --test=${TEST_DIR}/${TEST}
   --oltp-table
   --size=2000000
   --max-time=300
   --max-requests=0
   --mysql-table-engine=InnoDB
   --mysql-user=root
   --mysql-engine-trx=yes
   --num-threads=64
   --mysql-socket=/mnt/mysql/mysql.sock
   prepare
```
Running Sysbench OLTP Workload

Update the TEST environmental variable to a OLTP script:

[ceph@client1]$ TEST=oltp.lua

Finally, execute the OLTP workload with sysbench:

[ceph@client1]$ sysbench --test=${TEST_DIR}/${TEST} --oltp-table-size=2000000 --max-time=300 --max-requests=0 --mysql-table-engine=InnoDB --mysql-user=root --mysql-engine-trx=yes --num-threads=64 --mysql-socket=/mnt/mysql/mysql.sock run
End Lab

If you have completed all the lab exercises and wish to end your session, please select the End button from the launch page.

Thank You!