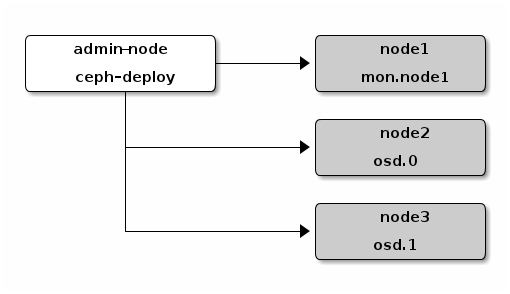
[**https://access.redhat.com/discussions/1161713**](https://access.redhat.com/discussions/1161713)

**4 Node Ceph Storage Cluster Setup**

**(ceph-admin, ceph-node1(monitor),ceph-node2,ceph-node3)**



**->Ceph Storage Cluster Setup with Four Nodes:-**

**(ceph-admin, ceph-node1(monitor), ceph-node1(OSD0), ceph-node2(OSD1)**

**Distribution ceph-version linux-kernal**

**Ubuntu**

**Trusty, 14.04LTS 0.79-0.80(trusty update),firefly 3.13**

**Note:-**

**1)Here I am going to do ceph Storage Cluster setup on VMWare virtual machines but not on physical servers.**

**->Hardware Recommendations:-**

**1)For Ceph-admin**

**RAM ->6GB, CPU Cores->4, NICs->2, Hard disk->50GB.**

**2)For Ceph-node1(Monitor)**

**RAM->6GB, CPU-Cores ->3, NICs->2, Harddisk->40GB**

**3)For OSDs(ceph-node1,ceph-node2)[Minimum we have to take 2OSDs for replication of data].**

**RAM->2GB, CPU->3cores, NICs->2, Hard disk->20GB(for OS installation only), 100GB for ceph OSD**

**Note:-**

**1) Here hardware requirements are provided by us to do Ceph storage cluster setup on VMs.**

**->System Update and Upgrade:-**

**$sudo apt-get update**

**$sudo apt-get upgrade && sudo apt-get dist-upgrade**

**On ceph-admin Node:-**

**$sudo vi /etc/hosts**

**1)In /etc/hosts file of ceph-admin node put IPs and their hostnames of monitor and ceph storage nodes.**

**->Create a Ceph user in all nodes.[Including admin node also].**

**$sudo useradd -d /home/ceph -m ceph -s /bin/bash**

**$sudo passwd ceph**

**->Add root privileges to ceph user.**

**$echo "ceph ALL = (root) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/ceph**

**$ sudo chmod 0440 /etc/sudoers.d/ceph**

**Establishing passwordless connections from admin node to mon and osd nodes.**

**$ ssh-keygen [Generate key from admin node].**

**$ ssh-copy-id ceph@[ceph-mon, ceph-osd1, ceph-osd2]**

**Verify the passwordless ssh connection from admin node to all remaining nodes.**

**->Ceph Deploy Setup**

**Ceph-deploy utility will make easy the ceph cluster deployment.**

**COMMAND description**

**new Start deploying a new cluster, and write a CLUSTER.conf**

**and keyring for it.**

**install Install Ceph packages on remote hosts.**

**mds Deploy ceph MDS on remote hosts.**

**mon Deploy ceph monitor on remote hosts.**

**gatherkeys Gather authentication keys for provisioning new nodes.**

**disk Manage disks on a remote host.**

**osd Prepare a data disk on remote host.**

**admin Push configuration and client.admin key to a remote**

**host.**

**config Push configuration file to a remote host.**

**uninstall Remove Ceph packages from remote hosts.**

**purgedata Purge (delete, destroy, discard, shred) any Ceph data**

**from /var/lib/ceph**

**purge Remove Ceph packages from remote hosts and purge all**

**data.**

**forgetkeys Remove authentication keys from the local directory.**

**pkg Manage packages on remote hosts.**

**What is Ceph-deploy utility? What it will do?**

**->Setup ceph-deploy utility on admin node:-**

[**http://docs.ceph.com/docs/master/install/get-packages/**](http://docs.ceph.com/docs/master/install/get-packages/)

**->>For Debian and Ubuntu distributions, perform the following steps:**

**a)Add the release key**

**$wget -q -O- 'https://ceph.com/git/?p=ceph.git;a=blob\_plain;f=keys/release.asc' | sudo apt-key add -**

**b)Add the Ceph packages to your repository.**

**$echo deb http://ceph.com/debian-firefly/ trusty main to /etc/apt/sources.list.d/ceph.list**

**c)Update your repository and install ceph-deploy:**

**$sudo apt-get update && sudo apt-get install ceph-deploy**

**Note:-**

**Here firefly is the ceph stable release and trusty Ubuntu release version name.**

**->Ceph Node Setup:-[On monitor and ceph-storage nodes]**

**IMP. Note:-**

**1)The admin node must be have password-less SSH access to Ceph nodes. When ceph-deploy logs in to a Ceph node as a user, that particular user must have passwordless sudo privileges.**

### **1)Install Ntp**

**$sudo apt-get install ntp(For Debian or Ubuntu)  
  
Note:-  
We recommend installing NTP on Ceph nodes (especially on Ceph Monitor nodes) to prevent issues arising from clock drift.   
  
2)Install SSH Server(On each ceph node)  
$sudo apt-get install openssh-server**

**Note:-**

**->Ensure the SSH server is running on ALL Ceph Nodes.**

**3)Create a ceph user**

**V.V.IMP Note:-**

**->The ceph-deploy utility must login to a Ceph node as a user that has passwordless sudoprivileges, because it needs to install software and configuration files without prompting for passwords.**

**->We recommend creating a Ceph user on ALL Ceph nodes in the cluster.**

**-------->>>>>>Install and Configure Ceph on Monitor node and storage nodes:-**

**1)Create a directory on your admin node node for maintaining the configuration and Keys that ceph-deploy generates for your cluster.**

**$mkdir my-cluster**

**$ cd my-cluster**

**NOTE:-**

**->The ceph-deploy utility will output files to the current directory. Ensure you are in this directory when executing ceph-deploy.**

**->Do not call Ceph-Deploy with sudo or run it as root if you are logged in as a different user, because it will not issue sudo commands needed on the remote host.**

**-->>On your admin node, perform the following steps using ceph-deploy.**

**1)Create a Cluster**

**syntax:-**

**ceph-deploy new {initial-monitor-node(s)}**

**Ex:-**

**$ceph-deploy new ceph-node1(ceph monitor node)**

**Start deploying a new cluster, and write a CLUSTER.conf and keyring for it.**

**ceph@ceph-mon:~/mycluster$ ceph-deploy new ceph-mon**

**[ceph\_deploy.cli][INFO ] Invoked (1.4.0): /usr/bin/ceph-deploy new ceph-mon**

**[ceph\_deploy.new][DEBUG ] Creating new cluster named ceph**

**[ceph\_deploy.new][DEBUG ] Resolving host ceph-mon**

**[ceph\_deploy.new][DEBUG ] Monitor ceph-mon at 192.168.2.102**

**[ceph\_deploy.new][INFO ] making sure passwordless SSH succeeds**

**[ceph\_deploy.new][DEBUG ] Monitor initial members are ['ceph-mon']**

**[ceph\_deploy.new][DEBUG ] Monitor addrs are ['192.168.2.102']**

**[ceph\_deploy.new][DEBUG ] Creating a random mon key...**

**[ceph\_deploy.new][DEBUG ] Writing initial config to ceph.conf...**

**[ceph\_deploy.new][DEBUG ] Writing monitor keyring to ceph.mon.keyring...**

**ceph@ceph-mon:~/mycluster$ ls**

**ceph.conf ceph.log ceph.mon.keyring**

**$vi ceph.conf [Default Initial ceph configuration]**

**[global]**

**fsid = 03bb4fbe-98ad-4f80-a23c-4a59675dda30**

**mon\_initial\_members = ceph-mon**

**mon\_host = 192.168.2.102**

**auth\_cluster\_required = cephx**

**auth\_service\_required = cephx**

**auth\_client\_required = cephx**

**filestore\_xattr\_use\_omap = true**

**$vi ceph.log**

**#logging info present here**

**$vi ceph.mon.keyring [random mon key generated and written to ceph.mon.keyring]**

**[mon.]**

**key = AQA3mn9WAAAAABAA9XIyLX2qkEp4eLDxAFm5pA==**

**caps mon = allow \***

**->Change the default number of replicas in ceph configuration file from 3 to 2. So that ceph can achieve active+clean state with just two OSDs.**

**[global]**

**osd pool default size = 2**

**->If you have more than one network interface , add the public network settings under the global section of your ceph configuration file.**

**[global]**

**public network = {ip-address}/{netmask}**

**[global]  
  
 *# By default, Ceph makes 3 replicas of objects. If you want to make four*   
 *# copies of an object the default value--a primary copy and three replica*   
 *# copies--reset the default values as shown in 'osd pool default size'.*  
 *# If you want to allow Ceph to write a lesser number of copies in a degraded*   
 *# state, set 'osd pool default min size' to a number less than the*  
 *# 'osd pool default size' value.*  
  
 osd pool default size = 4 # Write an object 4 times.  
 osd pool default min size = 1 # Allow writing one copy in a degraded state.  
  
 *# Ensure you have a realistic number of placement groups. We recommend*  
 *# approximately 100 per OSD. E.g., total number of OSDs multiplied by 100*   
 *# divided by the number of replicas (i.e., osd pool default size). So for*  
 *# 10 OSDs and osd pool default size = 4, we'd recommend approximately*  
 *# (100 \* 10) / 4 = 250.*  
  
 osd pool default pg num = 250  
 osd pool default pgp num = 250**

[**http://docs.ceph.com/docs/master/rados/configuration/pool-pg-config-ref/**](http://docs.ceph.com/docs/master/rados/configuration/pool-pg-config-ref/)

[**http://docs.ceph.com/docs/firefly/rados/operations/authentication/**](http://docs.ceph.com/docs/firefly/rados/operations/authentication/)

[**http://docs.ceph.com/docs/master/rados/configuration/auth-config-ref/**](http://docs.ceph.com/docs/master/rados/configuration/auth-config-ref/)

**Note:- If you use ceph-deploy Utility , Bydefault user authentication for ceph cluster CEPHX enabled by creating a monitor keyring and initial ceph.conf file[where cephx configured] when you run ceph-deploy new ceph-mon.**

**-->>Install Ceph**

**Install Ceph packages on remote hosts.**

**$ceph-deploy install ceph-admin ceph-mon ceph-osd1 ceph-osd2**

ceph@ceph-admin:~/mycluster$ ceph-deploy install ceph-osd1

[ceph\_deploy.cli][INFO ] Invoked (1.4.0): **/usr/bin/ceph-deploy install ceph-osd1**

[ceph\_deploy.install][DEBUG ] **Installing stable version emperor on cluster ceph hosts ceph-osd1**

[ceph\_deploy.install][DEBUG ] **Detecting platform for host ceph-osd1 ...**

[ceph-osd1][DEBUG ] **connected to host: ceph-osd1**

[ceph-osd1][DEBUG ] **detect platform information from remote host**

[ceph-osd1][DEBUG ] **detect machine type**

[ceph\_deploy.install][INFO ] **Distro info: Ubuntu 14.04 trusty**

[ceph-osd1][INFO ] **installing ceph on ceph-osd1**

[ceph-osd1][INFO ] Running command: **sudo env DEBIAN\_FRONTEND=noninteractive apt-get -q install --assume-yes ca-certificates**

[ceph-osd1][DEBUG ] Reading package lists...

[ceph-osd1][DEBUG ] Building dependency tree...

[ceph-osd1][DEBUG ] Reading state information...

[ceph-osd1][DEBUG ] ca-certificates is already the newest version.

[ceph-osd1][DEBUG ] 0 upgraded, 0 newly installed, 0 to remove and 67 not upgraded.

[ceph-osd1][INFO ] Running command: **sudo apt-get -q update**

[ceph-osd1][INFO ] Running command: s**udo env DEBIAN\_FRONTEND=noninteractive DEBIAN\_PRIORITY=critical apt-get -q -o Dpkg::Options::=--force-confnew**

--no-install-recommends --assume-yes install -- ceph ceph-mds ceph-common ceph-fs-common gdisk

[ceph-osd1][DEBUG ] Reading package lists...

[ceph-osd1][DEBUG ] Building dependency tree...

[ceph-osd1][DEBUG ] Reading state information...

[ceph-osd1][DEBUG ] gdisk is already the newest version.

[ceph-osd1][DEBUG ] The following extra packages will be installed:

[ceph-osd1][DEBUG ] **libcephfs1 librados2 librbd1 python-ceph python-flask python-itsdangerous**

[ceph-osd1][DEBUG ] python-werkzeug

[ceph-osd1][DEBUG ] Suggested packages:

[ceph-osd1][DEBUG ] **python-flask-doc ipython python-genshi python-lxml python-greenlet**

[ceph-osd1][DEBUG ] python-redis python-pylibmc python-memcache python-werkzeug-doc

[ceph-osd1][DEBUG ] Recommended packages:

[ceph-osd1][DEBUG ]  **ceph-fuse python-blinker python-pyinotify**

[ceph-osd1][DEBUG ] The following NEW packages will be installed:

[ceph-osd1][DEBUG ] **ceph ceph-common ceph-fs-common ceph-mds libcephfs1 librados2 librbd1**

[ceph-osd1][DEBUG ] python-ceph python-flask python-itsdangerous python-werkzeug

[ceph-osd1][DEBUG ] 0 upgraded, 11 newly installed, 0 to remove and 67 not upgraded.

[ceph-osd1][DEBUG ] Need to get 15.1 MB of archives.

[ceph-osd1][DEBUG ] After this operation, 74.1 MB of additional disk space will be used.

[ceph-osd1][DEBUG ] Get:1 http://ubuntu.cs.utah.edu/ubuntu/ trusty-updates/main librados2 amd64 0.80.10-0ubuntu1.14.04.3 [1427 kB]

[ceph-osd1][DEBUG ] Get:2 http://ubuntu.cs.utah.edu/ubuntu/ trusty-updates/main librbd1 amd64 0.80.10-0ubuntu1.14.04.3 [318 kB]

[ceph-osd1][DEBUG ] Get:3 http://ubuntu.cs.utah.edu/ubuntu/ trusty-updates/main libcephfs1 amd64 0.80.10-0ubuntu1.14.04.3 [1517 kB]

[ceph-osd1][DEBUG ] Get:4 http://ubuntu.cs.utah.edu/ubuntu/ trusty-updates/main python-werkzeug all 0.9.4+dfsg-1.1ubuntu2 [236 kB]

[ceph-osd1][DEBUG ] Get:5 http://ubuntu.cs.utah.edu/ubuntu/ trusty/main python-itsdangerous all 0.22+dfsg1-1build1 [11.5 kB]

[ceph-osd1][DEBUG ] Get:6 http://ubuntu.cs.utah.edu/ubuntu/ trusty/main python-flask all 0.10.1-2build1 [51.7 kB]

[ceph-osd1][DEBUG ] Get:7 http://ubuntu.cs.utah.edu/ubuntu/ trusty-updates/main python-ceph amd64 0.80.10-0ubuntu1.14.04.3 [35.3 kB]

[ceph-osd1][DEBUG ] Get:8 http://ubuntu.cs.utah.edu/ubuntu/ trusty-updates/main ceph-common amd64 0.80.10-0ubuntu1.14.04.3 [3983 kB]

[ceph-osd1][DEBUG ] Get:9 http://ubuntu.cs.utah.edu/ubuntu/ trusty-updates/main ceph amd64 0.80.10-0ubuntu1.14.04.3 [5396 kB]

[ceph-osd1][DEBUG ] Get:10 http://ubuntu.cs.utah.edu/ubuntu/ trusty-updates/universe ceph-fs-common amd64 0.80.10-0ubuntu1.14.04.3 [25.9 kB]

[ceph-osd1][DEBUG ] Get:11 http://ubuntu.cs.utah.edu/ubuntu/ trusty-updates/universe ceph-mds amd64 0.80.10-0ubuntu1.14.04.3 [2114 kB]

[ceph-osd1][DEBUG ] Fetched 15.1 MB in 39s (385 kB/s)

[ceph-osd1][DEBUG ] Selecting previously unselected package librados2.

[ceph-osd1][DEBUG ] (Reading database ... 119218 files and directories currently installed.)

[ceph-osd1][DEBUG ] Preparing to unpack .../librados2\_0.80.10-0ubuntu1.14.04.3\_amd64.deb ...

[ceph-osd1][DEBUG ] Unpacking librados2 (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Selecting previously unselected package librbd1.

[ceph-osd1][DEBUG ] Preparing to unpack .../librbd1\_0.80.10-0ubuntu1.14.04.3\_amd64.deb ...

[ceph-osd1][DEBUG ] Unpacking librbd1 (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Selecting previously unselected package libcephfs1.

[ceph-osd1][DEBUG ] Preparing to unpack .../libcephfs1\_0.80.10-0ubuntu1.14.04.3\_amd64.deb ...

[ceph-osd1][DEBUG ] Unpacking libcephfs1 (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Selecting previously unselected package python-werkzeug.

[ceph-osd1][DEBUG ] Preparing to unpack .../python-werkzeug\_0.9.4+dfsg-1.1ubuntu2\_all.deb ...

[ceph-osd1][DEBUG ] Unpacking python-werkzeug (0.9.4+dfsg-1.1ubuntu2) ...

[ceph-osd1][DEBUG ] Selecting previously unselected package python-itsdangerous.

[ceph-osd1][DEBUG ] Preparing to unpack .../python-itsdangerous\_0.22+dfsg1-1build1\_all.deb ...

[ceph-osd1][DEBUG ] Unpacking python-itsdangerous (0.22+dfsg1-1build1) ...

[ceph-osd1][DEBUG ] Selecting previously unselected package python-flask.

[ceph-osd1][DEBUG ] Preparing to unpack .../python-flask\_0.10.1-2build1\_all.deb ...

[ceph-osd1][DEBUG ] Unpacking python-flask (0.10.1-2build1) ...

[ceph-osd1][DEBUG ] Selecting previously unselected package python-ceph.

[ceph-osd1][DEBUG ] Preparing to unpack .../python-ceph\_0.80.10-0ubuntu1.14.04.3\_amd64.deb ...

[ceph-osd1][DEBUG ] Unpacking python-ceph (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Selecting previously unselected package ceph-common.

[ceph-osd1][DEBUG ] Preparing to unpack .../ceph-common\_0.80.10-0ubuntu1.14.04.3\_amd64.deb ...

[ceph-osd1][DEBUG ] Unpacking ceph-common (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Selecting previously unselected package ceph.

[ceph-osd1][DEBUG ] Preparing to unpack .../ceph\_0.80.10-0ubuntu1.14.04.3\_amd64.deb ...

[ceph-osd1][DEBUG ] Unpacking ceph (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Selecting previously unselected package ceph-fs-common.

[ceph-osd1][DEBUG ] Preparing to unpack .../ceph-fs-common\_0.80.10-0ubuntu1.14.04.3\_amd64.deb ...

[ceph-osd1][DEBUG ] Unpacking ceph-fs-common (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Selecting previously unselected package ceph-mds.

[ceph-osd1][DEBUG ] Preparing to unpack .../ceph-mds\_0.80.10-0ubuntu1.14.04.3\_amd64.deb ...

[ceph-osd1][DEBUG ] Unpacking ceph-mds (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Processing triggers for man-db (2.6.7.1-1ubuntu1) ...

[ceph-osd1][DEBUG ] Processing triggers for ureadahead (0.100.0-16) ...

[ceph-osd1][DEBUG ] Setting up librados2 (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Setting up librbd1 (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Setting up libcephfs1 (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Setting up python-werkzeug (0.9.4+dfsg-1.1ubuntu2) ...

[ceph-osd1][DEBUG ] Setting up python-itsdangerous (0.22+dfsg1-1build1) ...

[ceph-osd1][DEBUG ] Setting up python-flask (0.10.1-2build1) ...

[ceph-osd1][DEBUG ] Setting up python-ceph (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Setting up ceph-common (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Setting up ceph (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] ceph-all start/running

[ceph-osd1][DEBUG ] Setting up ceph-fs-common (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] Processing triggers for ureadahead (0.100.0-16) ...

[ceph-osd1][DEBUG ] Setting up ceph-mds (0.80.10-0ubuntu1.14.04.3) ...

[ceph-osd1][DEBUG ] ceph-mds-all start/running

[ceph-osd1][DEBUG ] Processing triggers for libc-bin (2.19-0ubuntu6.6) ...

[ceph-osd1][DEBUG ] Processing triggers for ureadahead (0.100.0-16) ...

[ceph-osd1][INFO ] Running command: **sudo ceph --version**

[ceph-osd1][DEBUG ] **ceph version 9.1.0-420-ge3921a8** (e3921a8396870be4a38ce1f1b6c35bc0829dbb68)

**Note:--**

**$ceph-deploy install ceph-admin ceph-mon ceph-osd1 ceph-osd2[you may get the below issue]**

**[ceph-single][ERROR ] RuntimeError: command returned non-zero exit status: 100**

**[ceph\_deploy][ERROR ] RuntimeError: Failed to execute command: env DEBIAN\_FRONTEND=noninteractive DEBIAN\_PRIORITY=critical apt-get --assume-yes -q**

**--no-install-recommends install -o Dpkg::Options::=--force-confnew ceph ceph-mds radosgw**

**W: Failed to fetch https://download.ceph.com/debian-infernalis/dists/trusty/main/binary-amd64/Packages Failed to connect to download.ceph.com port**

**443: Network is unreachable**

**W: Failed to fetch https://download.ceph.com/debian-infernalis/dists/trusty/main/binary-i386/Packages Failed to connect to download.ceph.com port**

**443: Network is unreachable**

**E: Some index files failed to download. They have been ignored, or old ones used instead.**

>**The ceph-deploy utility will install Ceph on each node.**

**->If you use ceph-deploypurge, you must re-execute this step to re-install Ceph.**

**-->>Add the initial monitor(s) and gather the keys:**

Deploy ceph monitor on remote hosts.

**$ceph-deploy mon create ceph-mon**

**ceph@ceph-mon:~/mycluster$ ceph-deploy mon create ceph-mon**

**[ceph\_deploy.cli][INFO ] Invoked (1.4.0): /usr/bin/ceph-deploy mon create ceph-mon**

**[ceph\_deploy.mon][DEBUG ] Deploying mon, cluster ceph hosts ceph-mon**

**[ceph\_deploy.mon][DEBUG ] detecting platform for host ceph-mon ...**

**[ceph-mon][DEBUG ] connected to host: ceph-mon**

**[ceph-mon][DEBUG ] detect platform information from remote host**

**[ceph-mon][DEBUG ] detect machine type**

**[ceph\_deploy.mon][INFO ] distro info: Ubuntu 14.04 trusty**

**[ceph-mon][DEBUG ] determining if provided host has same hostname in remote**

**[ceph-mon][DEBUG ] get remote short hostname**

**[ceph-mon][DEBUG ] deploying mon to ceph-mon**

**[ceph-mon][DEBUG ] get remote short hostname**

**[ceph-mon][DEBUG ] remote hostname: ceph-mon**

**[ceph-mon][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf**

**[ceph-mon][DEBUG ] create the mon path if it does not exist**

**[ceph-mon][DEBUG ] checking for done path: /var/lib/ceph/mon/ceph-ceph-mon/done**

**[ceph-mon][DEBUG ] done path does not exist: /var/lib/ceph/mon/ceph-ceph-mon/done**

**[ceph-mon][INFO ] creating keyring file: /var/lib/ceph/tmp/ceph-ceph-mon.mon.keyring**

**[ceph-mon][DEBUG ] create the monitor keyring file**

**[ceph-mon][INFO ] Running command: sudo ceph-mon --cluster ceph --mkfs -i ceph-mon --keyring /var/lib/ceph/tmp/ceph-ceph-mon.mon.keyring**

**[ceph-mon][DEBUG ] ceph-mon: mon.noname-a 192.168.2.102:6789/0 is local, renaming to mon.ceph-mon**

**[ceph-mon][DEBUG ] ceph-mon: set fsid to 03bb4fbe-98ad-4f80-a23c-4a59675dda30**

**[ceph-mon][DEBUG ] ceph-mon: created monfs at /var/lib/ceph/mon/ceph-ceph-mon for mon.ceph-mon**

**[ceph-mon][INFO ] unlinking keyring file /var/lib/ceph/tmp/ceph-ceph-mon.mon.keyring**

**[ceph-mon][DEBUG ] create a done file to avoid re-doing the mon deployment**

**[ceph-mon][DEBUG ] create the init path if it does not exist**

**[ceph-mon][DEBUG ] locating the `service` executable...**

**[ceph-mon][INFO ] Running command: sudo initctl emit ceph-mon cluster=ceph id=ceph-mon**

**[ceph-mon][INFO ] Running command: sudo ceph --cluster=ceph --admin-daemon /var/run/ceph/ceph-mon.ceph-mon.asok mon\_status**

**[ceph-mon][DEBUG ] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**[ceph-mon][DEBUG ] status for monitor: mon.ceph-mon**

**[ceph-mon][DEBUG ] {**

**[ceph-mon][DEBUG ] "election\_epoch": 2,**

**[ceph-mon][DEBUG ] "extra\_probe\_peers": [],**

**[ceph-mon][DEBUG ] "monmap": {**

**[ceph-mon][DEBUG ] "created": "0.000000",**

**[ceph-mon][DEBUG ] "epoch": 1,**

**[ceph-mon][DEBUG ] "fsid": "03bb4fbe-98ad-4f80-a23c-4a59675dda30",**

**[ceph-mon][DEBUG ] "modified": "0.000000",**

**[ceph-mon][DEBUG ] "mons": [**

**[ceph-mon][DEBUG ] {**

**[ceph-mon][DEBUG ] "addr": "192.168.2.102:6789/0",**

**[ceph-mon][DEBUG ] "name": "ceph-mon",**

**[ceph-mon][DEBUG ] "rank": 0**

**[ceph-mon][DEBUG ] }**

**[ceph-mon][DEBUG ] ]**

**[ceph-mon][DEBUG ] },**

**[ceph-mon][DEBUG ] "name": "ceph-mon",**

**[ceph-mon][DEBUG ] "outside\_quorum": [],**

**[ceph-mon][DEBUG ] "quorum": [**

**[ceph-mon][DEBUG ] 0**

**[ceph-mon][DEBUG ] ],**

**[ceph-mon][DEBUG ] "rank": 0,**

**[ceph-mon][DEBUG ] "state": "leader",**

**[ceph-mon][DEBUG ] "sync\_provider": []**

**[ceph-mon][DEBUG ] }**

**[ceph-mon][DEBUG ] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**[ceph-mon][INFO ] monitor: mon.ceph-mon is running**

**[ceph-mon][INFO ] Running command: sudo ceph --cluster=ceph --admin-daemon /var/run/ceph/ceph-mon.ceph-mon.asok mon\_status**

**Gather authentication keys for provisioning new nodes.**

**ceph@ceph-mon:~/mycluster$ ceph-deploy gatherkeys ceph-mon**

**[ceph\_deploy.cli][INFO ] Invoked (1.4.0): /usr/bin/ceph-deploy gatherkeys ceph-mon**

**[ceph\_deploy.gatherkeys][DEBUG ] Checking ceph-mon for /etc/ceph/ceph.client.admin.keyring**

**[ceph-mon][DEBUG ] connected to host: ceph-mon**

**[ceph-mon][DEBUG ] detect platform information from remote host**

**[ceph-mon][DEBUG ] detect machine type**

**[ceph-mon][DEBUG ] fetch remote file**

**[ceph\_deploy.gatherkeys][DEBUG ] Got ceph.client.admin.keyring key from ceph-mon.**

**[ceph\_deploy.gatherkeys][DEBUG ] Have ceph.mon.keyring**

**[ceph\_deploy.gatherkeys][DEBUG ] Checking ceph-mon for /var/lib/ceph/bootstrap-osd/ceph.keyring**

**[ceph-mon][DEBUG ] connected to host: ceph-mon**

**[ceph-mon][DEBUG ] detect platform information from remote host**

**[ceph-mon][DEBUG ] detect machine type**

**[ceph-mon][DEBUG ] fetch remote file**

**[ceph\_deploy.gatherkeys][DEBUG ] Got ceph.bootstrap-osd.keyring key from ceph-mon.**

**[ceph\_deploy.gatherkeys][DEBUG ] Checking ceph-mon for /var/lib/ceph/bootstrap-mds/ceph.keyring**

**[ceph-mon][DEBUG ] connected to host: ceph-mon**

**[ceph-mon][DEBUG ] detect platform information from remote host**

**[ceph-mon][DEBUG ] detect machine type**

**[ceph-mon][DEBUG ] fetch remote file**

**[ceph\_deploy.gatherkeys][DEBUG ] Got ceph.bootstrap-mds.keyring key from ceph-mon.**

**ceph@ceph-mon:~/mycluster$ ls**

**ceph.bootstrap-mds.keyring ceph.bootstrap-osd.keyring ceph.client.admin.keyring ceph.conf ceph.log ceph.mon.keyring**

**->Once you complete the process, your local directory should have the following keyrings:**

* **{cluster-name}.client.admin.keyring**
* **{cluster-name}.bootstrap-osd.keyring**
* **{cluster-name}.bootstrap-mds.keyring**
* **{cluster-name}.bootstrap-rgw.keyring**

**Note:-**

**->The bootstrap-rgw keyring is only created during installation of clusters running Hammer or newer.**

**-->>Add two OSDs**

**1)Login to the Ceph Nodes and create a directory for the Ceph OSD Daemon.**

**$ssh ceph-node2**

**$sudo mkdir /var/local/osd0**

**exit**

**$ssh ceph-node3**

**$sudo mkdir /var/local/osd1**

**exit**

**NOTE:-**

**->For fast setup, We uses a directory rather than an entire disk per Ceph OSD Daemon**

**->**[**http://docs.ceph.com/docs/master/rados/deployment/ceph-deploy-osd/**](http://docs.ceph.com/docs/master/rados/deployment/ceph-deploy-osd/)

**->See the above link for details on using separate disks/partitions for OSDs and journals. Login to the Ceph Nodes and create a directory for the Ceph OSD Daemon.**

**->From your admin node, use ceph-deploy to prepare the OSDs.**

**Prepare a data disk on remote host.**

**ceph@ceph-mon:~/mycluster$ ceph-deploy osd prepare ceph-osd1:/var/local/osd0 ceph-osd1:/var/local/osd1**

[ceph\_deploy.cli][INFO ] Invoked (1.4.0): /usr/bin/ceph-deploy osd prepare ceph-osd1:/var/local/osd0 ceph-osd1:/var/local/osd1

[ceph\_deploy.osd][DEBUG ] Preparing cluster ceph disks ceph-osd1:/var/local/osd0: ceph-osd1:/var/local/osd1:

[ceph-osd1][DEBUG ] connected to host: ceph-osd1

[ceph-osd1][DEBUG ] detect platform information from remote host

[ceph-osd1][DEBUG ] detect machine type

[ceph\_deploy.osd][INFO ] Distro info: Ubuntu 14.04 trusty

[ceph\_deploy.osd][DEBUG ] Deploying osd to ceph-osd1

[ceph-osd1][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf

[ceph-osd1][WARNIN] osd keyring does not exist yet, creating one

[ceph-osd1][DEBUG ] create a keyring file

[ceph-osd1][INFO ] Running command: sudo udevadm trigger --subsystem-match=block --action=add

[ceph\_deploy.osd][DEBUG ] Preparing host ceph-osd1 disk /var/local/osd0 journal None activate False

[ceph-osd1][INFO ] Running command: sudo ceph-disk-prepare --fs-type xfs --cluster ceph -- /var/local/osd0

[ceph\_deploy.osd][DEBUG ] Host ceph-osd1 is now ready for osd use.

[ceph-osd1][DEBUG ] connected to host: ceph-osd1

[ceph-osd1][DEBUG ] detect platform information from remote host

[ceph-osd1][DEBUG ] detect machine type

[ceph\_deploy.osd][INFO ] Distro info: Ubuntu 14.04 trusty

[ceph\_deploy.osd][DEBUG ] Preparing host ceph-osd1 disk /var/local/osd1 journal None activate False

[ceph-osd1][INFO ] Running command: sudo ceph-disk-prepare --fs-type xfs --cluster ceph -- /var/local/osd1

[ceph\_deploy.osd][DEBUG ] **Host ceph-osd1 is now ready for osd use.**

**ceph@ceph-mon:~/mycluster$ ceph-deploy osd prepare ceph-osd2:/var/local/osd2 ceph-osd2:/var/local/osd3**

**>Finally, activate the OSDs**

**$ ceph-deploy osd activate ceph-osd1:/var/local/osd0 ceph-osd1:/var/local/osd1**

**$ ceph-deploy osd activate ceph-osd2:/var/local/osd2 ceph-osd2:/var/local/osd3**

**-->Use ceph-deploy to copy the configuration file and admin key to your admin node and your Ceph Nodes so that you can use the ceph CLI without having to specify the monitor address and ceph.client.admin.keyring each time you execute a command.**

**admin-> Push configuration and client.admin key to a remote host.**

**ceph@ceph-mon:~/mycluster$ ceph-deploy admin ceph-mon ceph-osd1 ceph-osd2**

[ceph\_deploy.cli][INFO ] Invoked (1.4.0): /usr/bin/ceph-deploy admin ceph-mon ceph-osd1 ceph-osd2

[ceph\_deploy.admin][DEBUG ] Pushing admin keys and conf to ceph-mon

[ceph-mon][DEBUG ] connected to host: ceph-mon

[ceph-mon][DEBUG ] detect platform information from remote host

[ceph-mon][DEBUG ] detect machine type

[ceph-mon][DEBUG ] get remote short hostname

[ceph-mon][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf

[ceph\_deploy.admin][DEBUG ] Pushing admin keys and conf to ceph-osd1

[ceph-osd1][DEBUG ] connected to host: ceph-osd1

[ceph-osd1][DEBUG ] detect platform information from remote host

[ceph-osd1][DEBUG ] detect machine type

[ceph-osd1][DEBUG ] get remote short hostname

[ceph-osd1][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf

[ceph\_deploy.admin][DEBUG ] Pushing admin keys and conf to ceph-osd2

[ceph-osd2][DEBUG ] connected to host: ceph-osd2

[ceph-osd2][DEBUG ] detect platform information from remote host

[ceph-osd2][DEBUG ] detect machine type

[ceph-osd2][DEBUG ] get remote short hostname

[ceph-osd2][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf

**->When ceph-deploy is talking to the local admin host (ceph-admin), it must be reachable by its hostname. If necessary, modify /etc/hosts to add the name of the admin host.**

**->Ensure that you have the correct permissions for the ceph.client.admin.keyring.**

**$sudo chmod +r /etc/ceph/ceph.client.admin.keyring**

**ceph@ceph-mon:/etc/ceph$ ls**

**ceph.client.admin.keyring ceph.conf rbdmap**

**ceph@ceph-osd2:/etc/ceph$ ls**

**ceph.client.admin.keyring ceph.conf rbdmap**

**ceph@ceph-osd1:/etc/ceph$ ls**

**ceph.client.admin.keyring ceph.conf rbdmap**

**ceph@ceph-mon:~$ ceph -w**

**cluster 03bb4fbe-98ad-4f80-a23c-4a59675dda30**

**health HEALTH\_OK**

**monmap e1: 1 mons at {ceph-mon=192.168.2.102:6789/0}, election epoch 2, quorum 0 ceph-mon**

**osdmap e13: 4 osds: 4 up, 4 in**

**pgmap v25: 192 pgs, 3 pools, 0 bytes data, 0 objects**

**62069 MB used, 149 GB / 221 GB avail**

**192 active+clean**

**2015-12-27 14:26:12.893477 mon.0 [INF] pgmap v25: 192 pgs: 192 active+clean; 0 bytes data, 62069 MB used, 149 GB / 221 GB avail**

**Bydefault, Monitor has taken 6789 port.**

[**http://docs.ceph.com/docs/master/glossary/#term-ceph-storage-cluster**](http://docs.ceph.com/docs/master/glossary/#term-ceph-storage-cluster)