Ceph config (Openstack Smokey)

PW Owned by Paul Wilson

Last updated: Jan 13, 2023 • 12 min read • 1 person viewed

Ceph (disk object storage cluster)

Ceph is typically deployed with the Openstack cluster and JUJU however you can deploy it manually if you intend on making a standalone ceph cluster.

See: TBD build page for openstack

JUJU Addon Ceph Dashboard

Ceph Dashboard

A monitoring web UI is available in the form of the upstream Ceph Dashboard. It resides in the same model alongside the existing Ceph applications.

The dashboard is deployed using the ceph-dashboard charm. It works in conjunction with the openstack-loadbalancer charm, which in turn utilises the hacluster charm.

The ceph-dashboard charm is also compatible with Prometheus (for Ceph metric gathering) and Grafana (for displaying Prometheus data). This optional enhancement results in Grafana graphs being embedded within the dashboard.

Note:
The ceph-dashboard charm is currently in tech-preview.

Deployment

We are assuming a pre-existing Ceph cluster.

Deploy the ceph-dashboard as a subordinate to the ceph-mon charm.

```
juju deploy cs:~openstack-charmers/ceph-dashboard
juju add-relation ceph-dashboard:dashboard ceph-mon:dashboard
```

Copy

Note: TLS is a requirement for this charm. Enable it by adding a relation to the vault application:

```
juju add-relation ceph-dashboard:certificates vault:certificates
```

Copy

The dashboard is load balanced using VIPs and implemented via the openstack-loadbalancer and hacluster charms (use one or more space-separated VIP addressees local to your own environment):

```
juju deploy -n 3 cs:~openstack-charmers/openstack-loadbalancer
juju config openstack-loadbalancer vip=10.5.20.200
juju deploy hacluster openstack-loadbalancer-hacluster
juju add-relation openstack-loadbalancer:ha openstack-loadbalancer-
```

Copy

Finally, to actually enable load balancing, add a relation between the openstack-loadbalancer and ceph-dashboard applications:

```
juju add-relation ceph-dashboard:loadbalancer openstack-loadbalance
```

Copy

Accessing the dashboard

Add a dashboard user by applying charm action `add-user` to any ceph-dashboard unit:

```
juju run-action --wait ceph-dashboard/0 add-user username=admin rol
```

Copy

Note:

See the Ceph documentation on the different role types available.

This user's password is included in the command's output:

```
unit-ceph-dashboard-0:
  UnitId: ceph-dashboard/0
  id: "26"
  results:
    password: nMbKY95LmYvP
  status: completed
  timing:
    completed: 2021-10-26 16:58:49 +0000 UTC
    enqueued: 2021-10-26 16:58:47 +0000 UTC
    started: 2021-10-26 16:58:48 +0000 UTC
```

Copy

The web UI is available on the configured VIP and on port 8443:

```
https://10.5.20.200:8443
```

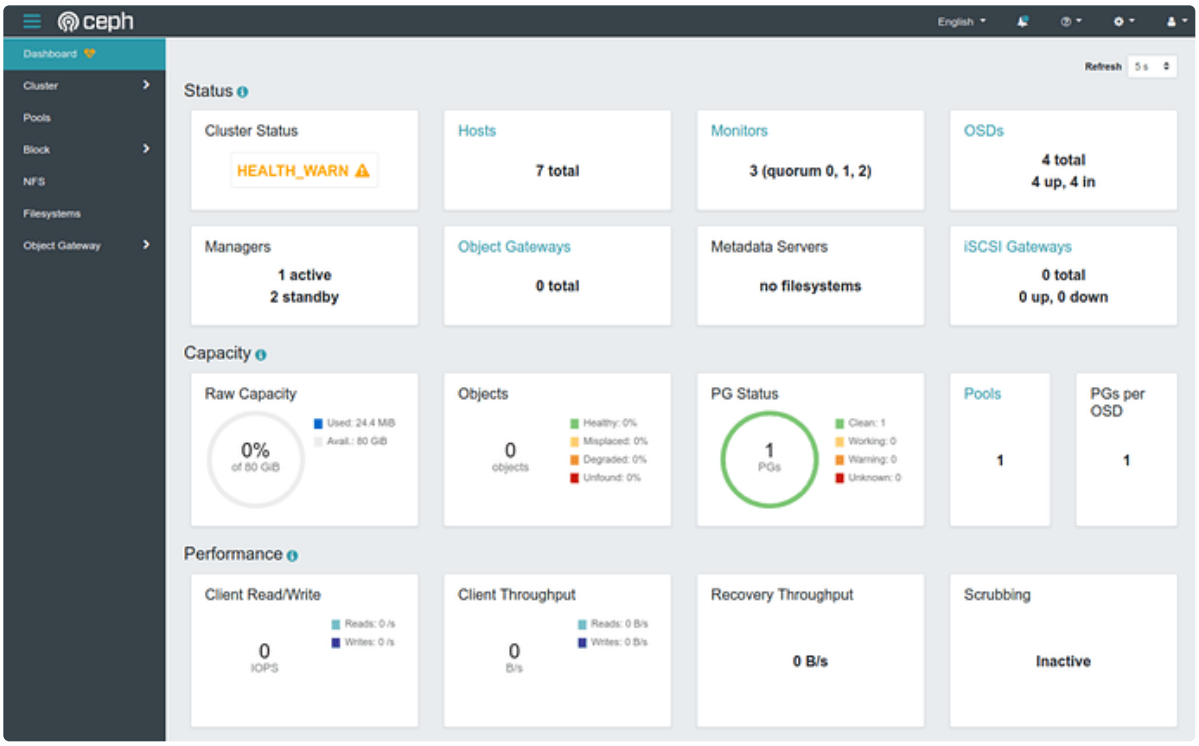
Copy

This should bring you to a login page:

In this example the credentials are:

Username: admin

Password: nMbKY95LmYvP



Manual Ceph Config

Ceph is a cluster-based storage system. Ceph is a software-defined storage solution designed to address the object, block, and file storage needs of data centers adopting open source as the new norm for high-growth block storage, object stores, and data lakes.

Prerequisites

The following are required to install the ceph cluster software on any host

- Python 3
- Systemd
- Podman or Docker for running containers (docker is installed with the ceph package
- Time synchronization (such as chrony or NTP)
- LVM2 for provisioning storage devices

These can be installed with this command

```
sudo apt install -y python3 systemd ntp lvm2
```

Copy

Installation

Installation of ceph is not super difficult however the SSH keys and users for ceph can be an issue. ALL default communication between the hosts is done with root, there is an option to change the default account for ceph but it is in TODO atm. (see below)

Enable the ssh with ROOT

```
Sudo nano /etc/ssh/sshd_config
```

Copy

Look for the # authentication part of the config gile and add the line "PermitRootLogin yes" EX:

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Copy

restart SSH

```
sudo service sshd restart
sudo service ssh restart
```

Copy

Configure SSH connections and configure ssh to allow root connections per host. make sure you do all this as SUDO SU (ROOT user)

Create key

```
ssh-keygen -b 4096
```

Copy

Copy key

```
ssh-copy-id root@%SYSTEM%
EX:
ssh-copy-id root@ceph-1
```

Copy

Do the above for all ceph systems in the cluster including itself

Add ceph ubuntu repo

```
curl --silent https://download.ceph.com/keys/release.asc | sudo apt
sudo apt-add-repository https://download.ceph.com/debian-pacific
```

Copy

Install cephadm (gui)

```
sudo apt install -y cephadm
```

Copy

This installs the main portion of ceph on each host. Each host must have this done to it however one host is bootstrapped as the initial cluster admin and other hosts are added later. MAKE SURE you only do this total config on ONE host.

Bootstrapping and initial host setup

The first step in creating a new Ceph cluster is running the `cephadm bootstrap` command on the Ceph cluster's first host. The act of running the `cephadm bootstrap` command on the Ceph cluster's first host creates the Ceph cluster's first

"monitor daemon", and that monitor daemon needs an IP address. You must pass the IP address of the Ceph cluster's first host to the `ceph bootstrap` command, so you'll need to know the IP address of that host.

- If there are multiple networks and interfaces, be sure to choose one that will be accessible by any host accessing the Ceph cluster. (TODO set cluster replication on 10g int)

Run the `ceph bootstrap` command:

```
1  cephadm bootstrap --mon-ip *<mon-ip>*
```

Copy

This command will:

- Create a monitor and manager daemon for the new cluster on the localhost.
- Generate a new SSH key for the Ceph cluster and add it to the root user's `/root/.ssh/authorized_keys` file.
- Write a copy of the public key to `/etc/ceph/ceph.pub` .
- Write a minimal configuration file to `/etc/ceph/ceph.conf` . This file is needed to communicate with the new cluster.
- Write a copy of the `client.admin` administrative (privileged!) secret key to `/etc/ceph/ceph.client.admin.keyring` .
- Add the `_admin` label to the bootstrap host. By default, any host with this label will (also) get a copy of `/etc/ceph/ceph.conf` and `/etc/ceph/ceph.client.admin.keyring` .

(TODO) Larger Ceph clusters perform better when (external to the Ceph cluster) public network traffic is separated from (internal to the Ceph cluster) cluster traffic. The internal cluster traffic handles replication, recovery, and heartbeats between OSD daemons. You can define the cluster network by supplying the `--cluster-network` option to the `bootstrap` subcommand. This parameter must define a subnet in CIDR notation (for example `10.90.90.0/24` or `fe80::/64`).

(TODO) The `--ssh-user *<user>*` option makes it possible to choose which ssh user cephadm will use to connect to hosts. The associated ssh key will be added to

`/home/*<user>*/.ssh/authorized_keys`. The user that you designate with this option must have passwordless sudo access.

Enable CLI

Cephadm does not require any Ceph packages to be installed on the host. However, we recommend enabling easy access to the `ceph` command. There are several ways to do this:

The `cephadm shell` command launches a bash shell in a container with all of the Ceph packages installed. By default, if configuration and keyring files are found in `/etc/ceph` on the host, they are passed into the container environment so that the shell is fully functional. Note that when executed on a MON host, `cephadm shell` will infer the `config` from the MON container instead of using the default configuration. If `--mount <path>` is given, then the host `<path>` (file or directory) will appear under `/mnt` inside the container:

```
1  # cephadm shell
```

Copy

To execute `ceph` commands, you can also run commands like this:

```
1  # cephadm shell -- ceph -s
```

Copy

You can install the `ceph-common` package, which contains all of the ceph commands, including `ceph`, `rbd`, `mount.ceph` (for mounting CephFS file systems), etc.:

```
1  # cephadm add-repo --release octopus
2  # cephadm install ceph-common
```

Copy

Confirm that the `ceph` command is accessible with:

```
1  # ceph -v
```

Copy

Confirm that the `ceph` command can connect to the cluster and also its status with:

```
# ceph status
```

Copy

The Graphina dashboard will error unless it is allowed to access not on SSL because this is typically local

```
#ceph dashboard set-grafana-api-ssl-verify False
```

Copy

this should get all the Graphina graphs working.

Adding hosts

Hosts must have these Requirements installed. Hosts without all the necessary requirements will fail to be added to the cluster.

To add each new host to the cluster, perform two steps:

Install the cluster's public SSH key in the new host's root user's `authorized_keys` file:

```
# ssh-copy-id -f -i /etc/ceph/ceph.pub root@*<new-host>*
```

Copy

For example:

```
# ssh-copy-id -f -i /etc/ceph/ceph.pub root@ceph-2
# ssh-copy-id -f -i /etc/ceph/ceph.pub root@ceph-3
```

Copy

Tell Ceph that the new node is part of the cluster:

```
# ceph orch host add *<newhost>* [*<ip>*] [*<label1> ...*]
```

Copy

For example:

```
# ceph orch host add ceph-2 10.0.7.21
# ceph orch host add ceph-3 10.0.7.22
```

Copy

It is best to explicitly provide the host IP address. If an IP is not provided, then the host name will be immediately resolved via DNS and that IP will be used.

One or more labels can also be included to immediately label the new host. For example, by default the `_admin` label will make cephadm maintain a copy of the `ceph.conf` file and a `client.admin` keyring file in `/etc/ceph`:

```
# ceph orch host add ceph-4 10.0.7.23 --labels _admin
```

Copy

Removing hosts

A host can safely be removed from the cluster once all daemons are removed from it.

To drain all daemons from a host do the following:

```
# ceph orch host drain *<host>*
```

Copy

The '_no_schedule' label will be applied to the host. See Special host labels

All osds on the host will be scheduled to be removed. You can check osd removal progress with the following:

```
# ceph orch osd rm status
```

Copy

You can check if there are no deamons left on the host with the following:

```
# ceph orch ps <host>
```

Copy

Once all daemons are removed you can remove the host with the following:

```
# ceph orch host rm <host>
```

Copy

If a host is offline you can force the removal with the following

```
# ceph orch host rm <host> --offline --force
```

Copy

Adding OSD's

To add storage to the cluster, either tell Ceph to consume any available and unused device:

```
# ceph orch apply osd --all-available-devices
```

Copy

With existing clusters and expanding, drives use the GUI to add OSD devices or refer to References (TODO instructions)

Troubleshooting

Some odd errors I have found associated with the setup so far

Bad or leftover drive data

causing the system to not be able to write new info to the physical disks

1. SSH to the offending system
2. Determine which system drives are the / and boot disks. and avoid doing the following to
3. list drives with lsblk
4. Select the drives that are not storing / and boot and run the following command

```
sudo wupefs -a /dev/sd*
```

Copy

If the drive has no partitions or if it is formatted wrong another option is the following

```
1  sudo ceph-volume lvm zap /dev/sd*
```

Copy

Silencing alarms for ceph crashes

if ceph experiences a crash you must acknowledge the crash

list any crashes with `ceph crash ls` then select the id of the crash alert and you can look up info on the crash with `ceph crash info <ID>` to acknowledge you can use `ceph crash archive <ID>` or `ceph crash archive-all`

Stuck or failed OSD

Find the failed osd by checking the logs on each server and look for the following

```
1  loaded failed failed    Ceph osd.1 for a22dbafc-4342-11ec-8229-0cc4
```

Copy

once you have found the failed ceph OSD on the host it lives on you can remove it with the following command (make sure this is done on the host that the failed one exists on)

```
1  cephadm rm-daemon --name osd.29
```

Copy

Ceph config with juju Openstack but 2 data pools

Configuring ceph with juju but adding a separate pool for data storage is possible but a bit complicated. the following is how to. keep in mind you need to make sure that Openstack is settled and online from the original push before you attempt to configure the other pools.

Prerequisites

Openstack must be pushed with all drives needed for instance creation. for example the current smokey config consists as follows

Drive-name/phys location	Drive type	Drive size	Drive use /
/dev/sda	SSD	500gb	boot, /
/dev/sdb	SSD	1tb	cinder-ceph
/dev/sdc	SSD	1tb	cinder-ceph
/dev/sdd	SSD	1tb	cinder-ceph
/dev/sde	HDD	2tb	cinder-ceph
/dev/sdf	HDD	2tb	cinder-ceph
/dev/sdg	HDD	2tb	cinder-ceph

when you configure the hardware the drives must coincide with the physical slots on boot order or the ceph system will not work and dual pool will fail.

Configure the juju push for Openstack and only use the default cinder-ceph drives, in the case of the current smokey push we only set sdb,sdc,sdd as drives available for ceph-osd in the openstack.yaml deployment. Once Openstack is deployed and operational you can begin the process of adding the other drives for the additional pool.

Adding the additional pool

the first step (unknown if needed) is to zap all of the disks on the osd systems. From juju use the following command for each ceph-osd and each drive respectively.

Adding the hardware for the additional pool

```
1  juju run-action --wait ceph-osd/X zap-disk i-really-mean-it=true de
```

Copy

Once the disks are zapped on each machine you can go into ceph-dashboard and add the drives in ceph-osd by adding the missing drive names and paths to the config and apply it.

It will look like this when done

Once the drives are added and all exciting items are complete and idle in juju status, you can then start working on the ceph config. you will need to ssh into the leader

ceph-mon. usually with the command

```
1  juju ssh ceph-mon/leader
```

Copy

once logged in change to root with sudo su, all commands must be run as root.

Configuring ceph-mon crush map rules

check to see if the drives are online and listing by type (in this case we are separating the ssd and hdd by type)

```
1  ceph osd tree
```

Copy

using this will list the available OSD's and you can see if they are listed as hdd and sdd.

EX:

Once you see them added and in the pool it is time for rule creation. You must make 2 new rules. one for ssd's and one for hdd's, make sure you apply the rules for the hdd first then the ssd rules. it is not critical but it is helpful for later modification

the format is:

```
ceph osd crush rule create-replicated <rulesetname> default <failur
```

Copy

so for the hdd system we do:

```
ceph osd crush rule create-replicated highcapacitypool default host
```

Copy

for ssd:

```
ceph osd crush rule create-replicated highspeedpool default host ss
```

Copy

Once the rules have been applied you need to print crush rules to make sure they are correct.

```
ceph osd crush rule dump
```

Copy

should print the following

```
}
# rules
rule replicated_rule {
  id 0
  type replicated
  step take default
  step chooseleaf firstn 0 type host
  step emit
}
```

```
rule highcapacitypool {
  id 1
  type replicated
  step take default class hdd
  step chooseleaf firstn 0 type host
  step emit
}
rule highspeedpool {
  id 2
  type replicated
  step take default class ssd
  step chooseleaf firstn 0 type host
  step emit
}
```

Copy

once this is confirmed you need to modify the crush map. but the map is compiled and will need to be decompiled to configure the pool drive availability.

Crush map modification

GET A CRUSH MAP

To get the CRUSH map for your cluster, execute the following:

```
ceph osd getcrushmap -o {compiled-crushmap-filename}
```

Copy

Ceph will output (-o) a compiled CRUSH map to the filename you specified. Since the CRUSH map is in a compiled form, you must decompile it first before you can edit it.

DECOMPILE A CRUSH MAP

To decompile a CRUSH map, execute the following:

```
crushtool -d {compiled-crushmap-filename} -o {decompiled-crushmap-f
```

Copy

Once the map is decompiled you need to change the default mapping to use ssd's and then comment out the SSD rile created previously.

```
}
# rules
rule replicated_rule {
  id 0
  type replicated
  step take default class ssd
  step chooseleaf firstn 0 type host
  step emit
}
rule highcapacitypool {
  id 1
  type replicated
  step take default class hdd
  step chooseleaf firstn 0 type host
  step emit
}
#rule highspeedpool {
#  id 2
#  type replicated
#  step take default class ssd
#  step chooseleaf firstn 0 type host
#  step emit
#}
```

Copy

once the modification is completed you recompile the crush map and set ceph to the new map.

RECOMPILE A CRUSH MAP

To compile a CRUSH map, execute the following:

```
crushtool -c {decompiled-crushmap-filename} -o {compiled-crushmap-f
```

Copy

SET THE CRUSH MAP

To set the CRUSH map for your cluster, execute the following:

```
ceph osd setcrushmap -i {compiled-crushmap-filename}
```

Copy

once the crush map is complete you need to create a pool that is connected to the new drive set.

make sure the ruleset that is selected is the pool desired. in this case the "replicated_rule" set is now ssd only and the “highcapacitypool” is the hdd set

once the pool is created you need to associate it with juju and openstack

Associating 2nd pool to openstack

first you need to create a cinder-ceph charm with the name of the pool you created.

```
juju deploy cinder-ceph cinder-ceph-%poolname%
```

Copy

then add the relations needed to nova-compute, ceph, cinder etc.

```
juju add-relation cinder cinder-ceph-%poolname%
juju add-relation nova-compute cinder-ceph-%poolname%
juju add-relation ceph-mon:client cinder-ceph-%poolname%:ceph
```

If the last relation does not work try the following:

```
juju add-relation ceph-mon:client cinder-ceph-hdd-lfs:ceph
```

Copy

once juju is finished adding the relations the pool should show in juju status

when the new pool is online and active you can add it to openstack.

connect to openstack cli: (see openstack cli: http://wikijs.mws.local/en/Dev_Ops/OpenStack/Smokey/Overcloud/OS-CLI-Setup)

```
source ~/openstack-bundles/stable/openstack-base/openrc
```

Copy

then add the volume to the stack with the following.

```
openstack volume type create --public --description "SSD, backend c
```

Copy

and connect that volume type to the volume created in juju and ceph

```
openstack volume type set  --property volume_backend_name="%poolnam
```

Copy

once this is done openstack will have the 2nd pool as an option for adding a volume to.

Change Openstack default image pool

First check that the pools have been created

```
cinder type-list
```

You should see any volumes you have added with the openstack pool creation

you will need to then list the OpenStack id's with the following command

```
openstack project list
```

This should show you something like the following

```
+----------------------------------+----------+
| ID                               | Name     |
+----------------------------------+----------+
| 1b7cd18a9ee74c97af195b855f21a214 | admin    |
| 3c99f7a8e91d408f8535a169f86b05e0 | services |
| 94fbfae968b64957b52e1538c016e214 | services |
| fc4a6804e7df4b43a197584e227cb57c | admin    |
+----------------------------------+----------+
```

Once you ave these you can check the default pool to make sure it is not the pool you want you can do this by running the cinder command

```
cinder type-default
```

Then if the default is not the pool you want use the command to change it.once the command is accepted make sure you run the type-default command again to check

```
cinder default-type-set <type-name> <project-id>
```

References

Installing on ubuntu: https://www.brightbox.com/blog/2021/04/26/cephadm-ubuntu-focal-20-04/

SSH key auth: https://www.linode.com/docs/guides/use-public-key-authentication-with-ssh/

Deploying a ceph cluster: https://docs.ceph.com/en/pacific/cephadm/install/

Host management: https://docs.ceph.com/en/pacific/cephadm/host-management/#cephadm-adding-hosts

Troubleshooting: https://docs.ceph.com/en/octopus/cephadm/troubleshooting/

Wiping drive clean: https://serverfault.com/questions/250839/deleting-all-partitions-from-the-command-line

Replacing OSD drive: https://docs.ceph.com/en/latest/rados/operations/add-or-rm-osds/

crash mitigation: https://it-ops.dev/ceph-daemons-have-recently-crashed

dashboard for ceph: https://github.com/ceph/ceph/blob/master/doc/mgr/dashboard.rst

Graphina troubleshooting: https://github.com/ceph/ceph/blob/master/doc/mgr/dashboard.rst

Adding pool to openstack: https://cloud.garr.it/support/kb/ceph/add_rbd_pool_to_openstack/

creating separate pools: https://alanxelsys.files.wordpress.com/2019/02/configuring-separate-ssd-and-hdd-pools-with-ceph-mimic.pdf

editing crush map: https://docs.ceph.com/en/latest/rados/operations/crush-map-edits/

pool documentation: https://docs.ceph.com/en/latest/rados/operations/pools/#:~:text=To%20remove%20a%20pool%20the,Monitor%20Configuration%20for%20more%20information.&text=If%20no%20other%20pools%20use,that%20rule%20from%20the%20cluster.

Default cinder config: https://docs.openstack.org/cinder/latest/admin/default-volume-types.html

Openstack project commands: https://docs.openstack.org/python-openstackclient/queens/cli/command-objects/project.html

+ Add label

Be the first to add a reaction