ADDING/REMOVING MONITORS

When you have a cluster up and running, you may add or remove monitors from the cluster at runtime. To bootstrap a monitor, see Manual Deployment or Monitor Bootstrap.

ADDING MONITORS

Ceph monitors are light-weight processes that maintain a master copy of the cluster map. You can run a cluster with 1 monitor. We recommend at least 3 monitors for a production cluster. Ceph monitors use a variation of the Paxos protocol to establish consensus about maps and other critical information across the cluster. Due to the nature of Paxos, Ceph requires a majority of monitors running to establish a quorum (thus establishing consensus).

It is advisable to run an odd-number of monitors but not mandatory. An odd-number of monitors has a higher resiliency to failures than an even-number of monitors. For instance, on a 2 monitor deployment, no failures can be tolerated in order to maintain a quorum; with 3 monitors, one failure can be tolerated; in a 4 monitor deployment, one failure can be tolerated; with 5 monitors, two failures can be tolerated. This is why an odd-number is advisable. Summarizing, Ceph needs a majority of monitors to be running (and able to communicate with each other), but that majority can be achieved using a single monitor, or 2 out of 2 monitors, 2 out of 3, 3 out of 4, etc.

For an initial deployment of a multi-node Ceph cluster, it is advisable to deploy three monitors, increasing the number two at a time if a valid need for more than three exists.

Since monitors are light-weight, it is possible to run them on the same host as an OSD; however, we recommend running them on separate hosts, because fsync issues with the kernel may impair performance.

> **Note:** A *majority* of monitors in your cluster must be able to reach each other in order to establish a quorum.

DEPLOY YOUR HARDWARE

If you are adding a new host when adding a new monitor, see Hardware Recommendations for details on minimum recommendations for monitor hardware. To add a monitor host to your cluster, first make sure you have an up-to-date version of Linux installed (typically Ubuntu 16.04 or RHEL 7).

Add your monitor host to a rack in your cluster, connect it to the network and ensure that it has network connectivity.

INSTALL THE REQUIRED SOFTWARE

For manually deployed clusters, you must install Ceph packages manually. See Installing Packages for details. You should configure SSH to a user with password-less authentication and root permissions.

ADDING A MONITOR (MANUAL)

This procedure creates a `ceph-mon` data directory, retrieves the monitor map and monitor keyring, and adds a `ceph-mon` daemon to your cluster. If this results in only two monitor daemons, you may add more monitors by repeating this procedure until you have a sufficient number of `ceph-mon` daemons to achieve a quorum.

At this point you should define your monitor's id. Traditionally, monitors have been named with single letters (a, b, c, ...), but you are free to define the id as you see fit. For the purpose of this document, please take into account that `{mon-id}` should be the id you chose, without the `mon.` prefix (i.e., `{mon-id}` should be the a on `mon.a`).

1. Create the default directory on the machine that will host your new monitor.

```
ssh {new-mon-host}
sudo mkdir /var/lib/ceph/mon/ceph-{mon-id}
```

2. Create a temporary directory `{tmp}` to keep the files needed during this process. This directory should be different from the monitor's default directory created in the previous step, and can be removed after all the steps are executed.

```
mkdir {tmp}
```

3. Retrieve the keyring for your monitors, where {tmp} is the path to the retrieved keyring, and {key-filename} is the name of the file containing the retrieved monitor key.

```
ceph auth get mon. -o {tmp}/{key-filename}
```

4. Retrieve the monitor map, where {tmp} is the path to the retrieved monitor map, and {map-filename} is the name of the file containing the retrieved monitor monitor map.

```
ceph mon getmap -o {tmp}/{map-filename}
```

5. Prepare the monitor's data directory created in the first step. You must specify the path to the monitor map so that you can retrieve the information about a quorum of monitors and their fsid. You must also specify a path to the monitor keyring:

```
sudo ceph-mon -i {mon-id} --mkfs --monmap {tmp}/{map-filename} --keyring {tmp}/{key-filen
```

6. Start the new monitor and it will automatically join the cluster. The daemon needs to know which address to bind to, either via --public-addr {ip:port} or by setting mon addr in the appropriate section of ceph.conf. For example:

```
ceph-mon -i {mon-id} --public-addr {ip:port}
```

REMOVING MONITORS

When you remove monitors from a cluster, consider that Ceph monitors use PAXOS to establish consensus about the master cluster map. You must have a sufficient number of monitors to establish a quorum for consensus about the cluster map.

REMOVING A MONITOR (MANUAL)

This procedure removes a ceph-mon daemon from your cluster. If this procedure results in only two monitor daemons, you may add or remove another monitor until you have a number of ceph-mon daemons that can achieve a quorum.

1. Stop the monitor.

```
service ceph -a stop mon.{mon-id}
```

2. Remove the monitor from the cluster.

```
ceph mon remove {mon-id}
```

3. Remove the monitor entry from ceph.conf.

REMOVING MONITORS FROM AN UNHEALTHY CLUSTER

This procedure removes a ceph-mon daemon from an unhealthy cluster, for example a cluster where the monitors cannot form a quorum.

1. Stop all ceph-mon daemons on all monitor hosts.

```
ssh {mon-host}
service ceph stop mon || stop ceph-mon-all
# and repeat for all mons
```

2. Identify a surviving monitor and log in to that host.

```
ssh {mon-host}
```

3. Extract a copy of the monmap file.

```
ceph-mon -i {mon-id} --extract-monmap {map-path}
# in most cases, that's
ceph-mon -i `hostname` --extract-monmap /tmp/monmap
```

4. Remove the non-surviving or problematic monitors. For example, if you have three monitors, `mon.a`, `mon.b`, and `mon.c`, where only `mon.a` will survive, follow the example below:

```
monmaptool {map-path} --rm {mon-id}
# for example,
monmaptool /tmp/monmap --rm b
monmaptool /tmp/monmap --rm c
```

5. Inject the surviving map with the removed monitors into the surviving monitor(s). For example, to inject a map into monitor `mon.a`, follow the example below:

```
ceph-mon -i {mon-id} --inject-monmap {map-path}
# for example,
ceph-mon -i a --inject-monmap /tmp/monmap
```

6. Start only the surviving monitors.
7. Verify the monitors form a quorum (`ceph -s`).
8. You may wish to archive the removed monitors' data directory in `/var/lib/ceph/mon` in a safe location, or delete it if you are confident the remaining monitors are healthy and are sufficiently redundant.

CHANGING A MONITOR'S IP ADDRESS

Important: Existing monitors are not supposed to change their IP addresses.

Monitors are critical components of a Ceph cluster, and they need to maintain a quorum for the whole system to work properly. To establish a quorum, the monitors need to discover each other. Ceph has strict requirements for discovering monitors.

Ceph clients and other Ceph daemons use `ceph.conf` to discover monitors. However, monitors discover each other using the monitor map, not `ceph.conf`. For example, if you refer to Adding a Monitor (Manual) you will see that you need to obtain the current monmap for the cluster when creating a new monitor, as it is one of the required arguments of `ceph-mon -i {mon-id} --mkfs`. The following sections explain the consistency requirements for Ceph monitors, and a few safe ways to change a monitor's IP address.

CONSISTENCY REQUIREMENTS

A monitor always refers to the local copy of the monmap when discovering other monitors in the cluster. Using the monmap instead of `ceph.conf` avoids errors that could break the cluster (e.g., typos in `ceph.conf` when specifying a monitor address or port). Since monitors use monmaps for discovery and they share monmaps with clients and other Ceph daemons, the monmap provides monitors with a strict guarantee that their consensus is valid.

Strict consistency also applies to updates to the monmap. As with any other updates on the monitor, changes to the monmap always run through a distributed consensus algorithm called Paxos. The monitors must agree on each update to the monmap, such as adding or removing a monitor, to ensure that each monitor in the quorum has the same version of the monmap. Updates to the monmap are incremental so that monitors have the latest agreed upon version, and a set of previous versions, allowing a monitor that has an older version of the monmap to catch up with the current state of the cluster.

If monitors discovered each other through the Ceph configuration file instead of through the monmap, it would introduce additional risks because the Ceph configuration files are not updated and distributed automatically. Monitors might inadvertently use an older `ceph.conf` file, fail to recognize a monitor, fall out of a quorum, or develop a situation where Paxos is not able to determine the current state of the system accurately. Consequently, making changes to an existing monitor's IP address must be done with great care.

CHANGING A MONITOR'S IP ADDRESS (THE RIGHT WAY)

Changing a monitor's IP address in `ceph.conf` only is not sufficient to ensure that other monitors in the cluster will receive the update. To change a monitor's IP address, you must add a new monitor with the IP address you want to use (as described in Adding a Monitor (Manual)), ensure that the new monitor successfully joins the quorum; then, remove the monitor that uses the old IP address. Then, update the `ceph.conf` file to ensure that clients and other daemons know the IP address of the new monitor.

For example, lets assume there are three monitors in place, such as

```
[mon.a]
        host = host01
        addr = 10.0.0.1:6789
[mon.b]
        host = host02
        addr = 10.0.0.2:6789
[mon.c]
        host = host03
        addr = 10.0.0.3:6789
```

To change `mon.c` to `host04` with the IP address `10.0.0.4`, follow the steps in Adding a Monitor (Manual) by adding a new monitor `mon.d`. Ensure that `mon.d` is running before removing `mon.c`, or it will break the quorum. Remove `mon.c` as described on Removing a Monitor (Manual). Moving all three monitors would thus require repeating this process as many times as needed.

CHANGING A MONITOR'S IP ADDRESS (THE MESSY WAY)

There may come a time when the monitors must be moved to a different network, a different part of the datacenter or a different datacenter altogether. While it is possible to do it, the process becomes a bit more hazardous.

In such a case, the solution is to generate a new monmap with updated IP addresses for all the monitors in the cluster, and inject the new map on each individual monitor. This is not the most user-friendly approach, but we do not expect this to be something that needs to be done every other week. As it is clearly stated on the top of this section, monitors are not supposed to change IP addresses.

Using the previous monitor configuration as an example, assume you want to move all the monitors from the `10.0.0.x` range to `10.1.0.x`, and these networks are unable to communicate. Use the following procedure:

1. Retrieve the monitor map, where `{tmp}` is the path to the retrieved monitor map, and `{filename}` is the name of the file containing the retrieved monitor monitor map.

   ```
   ceph mon getmap -o {tmp}/{filename}
   ```

2. The following example demonstrates the contents of the monmap.

   ```
   $ monmaptool --print {tmp}/{filename}

   monmaptool: monmap file {tmp}/{filename}
   epoch 1
   fsid 224e376d-c5fe-4504-96bb-ea6332a19e61
   last_changed 2012-12-17 02:46:41.591248
   created 2012-12-17 02:46:41.591248
   0: 10.0.0.1:6789/0 mon.a
   1: 10.0.0.2:6789/0 mon.b
   2: 10.0.0.3:6789/0 mon.c
   ```

3. Remove the existing monitors.

   ```
   $ monmaptool --rm a --rm b --rm c {tmp}/{filename}

   monmaptool: monmap file {tmp}/{filename}
   monmaptool: removing a
   monmaptool: removing b
   monmaptool: removing c
   ```

```
monmaptool: writing epoch 1 to {tmp}/{filename} (0 monitors)
```

4. Add the new monitor locations.

```
$ monmaptool --add a 10.1.0.1:6789 --add b 10.1.0.2:6789 --add c 10.1.0.3:6789 {tmp}/{fil

monmaptool: monmap file {tmp}/{filename}
monmaptool: writing epoch 1 to {tmp}/{filename} (3 monitors)
```

5. Check new contents.

```
$ monmaptool --print {tmp}/{filename}

monmaptool: monmap file {tmp}/{filename}
epoch 1
fsid 224e376d-c5fe-4504-96bb-ea6332a19e61
last_changed 2012-12-17 02:46:41.591248
created 2012-12-17 02:46:41.591248
0: 10.1.0.1:6789/0 mon.a
1: 10.1.0.2:6789/0 mon.b
2: 10.1.0.3:6789/0 mon.c
```

At this point, we assume the monitors (and stores) are installed at the new location. The next step is to propagate the modified monmap to the new monitors, and inject the modified monmap into each new monitor.

1. First, make sure to stop all your monitors. Injection must be done while the daemon is not running.

2. Inject the monmap.

```
ceph-mon -i {mon-id} --inject-monmap {tmp}/{filename}
```

3. Restart the monitors.

After this step, migration to the new location is complete and the monitors should operate successfully.