

! Notice

This document is for a development version of Ceph.

[Report a Documentation Bug](#)

ADDING/REMOVING MONITORS

It is possible to add Monitors to a running cluster as long as redundancy is maintained. To bootstrap a Monitor, see [Manual Deployment](#) or [Monitor Bootstrap](#).

ADDING MONITORS

Ceph Monitors serve as the single source of truth for the cluster map. It is possible to run a cluster with only one Monitor, but for a production cluster it is recommended to have at least three Monitors provisioned and in quorum. Ceph Monitors use a variation of the [Paxos](#) algorithm to maintain consensus about maps and about other critical information across the cluster. Due to the nature of Paxos, Ceph is able to maintain quorum (and thus establish consensus) only if a majority of the Monitors are `active`.

It is best to run an odd number of Monitors. This is because a cluster that is running an odd number of Monitors is more resilient than a cluster running an even number. For example, in a two-Monitor deployment, no failures can be tolerated if quorum is to be maintained; in a three-Monitor deployment, one failure can be tolerated; in a four-Monitor deployment, one failure can be tolerated; and in a five-Monitor deployment, two failures can be tolerated. In general, a cluster running an odd number of Monitors is best because it avoids what is called the *split brain* phenomenon. In short, Ceph is able to operate only if a majority of Monitors are `active` and able to communicate with each other. For example: there must be a single Monitor, two out of two Monitors, two out of three Monitors, three out of five Monitors, or the like.

For small or non-critical deployments of multi-node Ceph clusters, it is recommended to deploy three Monitors. For larger clusters or for clusters that are intended to survive a double failure, it is recommended to deploy five Monitors. Only in rare circumstances is there any justification for deploying seven or more Monitors.

has disadvantages: for example: `fsync` issues with the kernel might weaken performance, Monitor and OSD daemons might be inactive at the same time and cause disruption if the node crashes, is rebooted, or is taken down for maintenance. Because of these risks, it is instead recommended to run Monitors and Managers on dedicated hosts.

! Note

A *majority* of Monitors in your cluster must be able to reach each other in order for quorum to be established.

DEPLOYING YOUR HARDWARE

Some operators choose to add a new Monitor host at the same time that they add a new Monitor. For details on the minimum recommendations for Monitor hardware, see [Hardware Recommendations](#). Before adding a Monitor host to the cluster, make sure that there is an up-to-date version of Linux installed.

Add the newly installed Monitor host to a rack in your cluster, connect the host to the network, and make sure that the host has network connectivity.

INSTALLING THE REQUIRED SOFTWARE

In manually deployed clusters, it is necessary to install Ceph packages manually. For details, see [Installing Packages](#). Configure SSH so that it can be used by a user that has passwordless authentication and root permissions.

ADDING A MONITOR (MANUAL)

The procedure in this section creates a `ceph-mon` data directory, retrieves both the monitor map and the monitor keyring, and adds a `ceph-mon` daemon to the cluster. The procedure might result in a Ceph cluster that contains only two Monitor daemons. To add more Monitors until there are enough `ceph-mon` daemons to establish quorum, repeat the procedure.

This is a good point at which to define the new Monitor's `id`. Monitors have often been named with single letters (`a`, `b`, `c`, etc.), but you are free to define the `id` however you like. In this document, `{mon-id}` refers to the `id` exclusive of the `mon.` prefix: for example, if `a` has been chosen as the `id` of a Monitor, then `{mon-id}` is `a`. ???

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

2. Create a temporary directory `{tmp}` that will contain the files needed during this procedure. This directory should be different from the data directory created in the previous step. Because this is a temporary directory, it can be removed after the procedure is complete:

```
$ mkdir {tmp}
```

3. Retrieve the keyring for your Monitors (`{tmp}` is the path to the retrieved keyring and `{key-filename}` is the name of the file that contains the retrieved Monitor key):

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

4. Retrieve the monitor map (`{tmp}` is the path to the retrieved monitor map and `{map-filename}` is the name of the file that contains the retrieved monitor map):

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

5. Prepare the Monitor's data directory, which was created in the first step. The following command must specify the path to the monitor map (so that information about a quorum of Monitors and their `fsid`s can be retrieved) and specify the path to the monitor keyring:

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

6. Start the new Monitor. It will automatically join the cluster. To provide information to the daemon about which address to bind to, use either the `--public-addr {ip}` option or the `--public-network {network}` option. For example:

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

REMOVING MONITORS

When Monitors are removed from a cluster, it is important to remember that Ceph uses the Paxos to maintain consensus about the cluster map. Such consensus is possible only if the number of Monitors is sufficient to establish quorum.

The procedure in this section removes a `ceph-mon` daemon from the cluster. The procedure might result in a Ceph cluster that contains a number of Monitors insufficient to maintain quorum, so plan carefully. When replacing an old Monitor with a new Monitor, add the new Monitor first, wait for quorum to be established, and then remove the old Monitor. This ensures that quorum is not lost.

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 the `ceph.conf` file:

REMOVING MONITORS FROM AN UNHEALTHY CLUSTER

The procedure in this section removes a `ceph-mon` daemon from an unhealthy cluster (for example, a cluster whose Monitors are unable to form a quorum).

1. Stop all `ceph-mon` daemons on all Monitor hosts:

```
$ ssh {mon-host}  
$ systemctl stop ceph-mon.target
```

Repeat this step on every Monitor host.

2. Identify a surviving Monitor and log in to the Monitor's host:

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

3. Extract a copy of the `monmap` file by running a command of the following form:

```
$ ceph-mon -i {mon-id} --extract-monmap {map-path}
```

`/tmp/monmap` is the `{map-path}`:

```
$ ceph-mon -i `hostname` --extract-monmap /tmp/monmap
```

4. Remove the non-surviving or otherwise problematic Monitors:

```
$ monmaptool {map-path} --rm {mon-id}
```

For example, suppose that there are three Monitors—`mon.a`, `mon.b`, and `mon.c`—and that only `mon.a` will survive:

```
$ monmaptool /tmp/monmap --rm b
$ monmaptool /tmp/monmap --rm c
```

5. Inject the surviving map that includes the removed Monitors into the monmap of the surviving Monitor(s):

```
$ ceph-mon -i {mon-id} --inject-monmap {map-path}
```

Continuing with the above example, inject a map into Monitor `mon.a` by running the following command:

```
$ ceph-mon -i a --inject-monmap /tmp/monmap
```

6. Start only the surviving Monitors.
7. Verify that the Monitors form a quorum by running the command `ceph -s`.
8. The data directory of the removed Monitors is in `/var/lib/ceph/mon`: either archive this data directory in a safe location or delete this data directory. However, do not delete it unless you are confident that the remaining Monitors are healthy and sufficiently redundant. Make sure that there is enough room for the live DB to expand and compact, and make sure that there is also room for an archived copy of the DB. The archived copy can be compressed.

CHANGING A MONITOR'S IP ADDRESS

❗ Important

Existing Monitors are not supposed to change their IP addresses.

the monitors maintain quorum, and quorum can be established only if the monitors have discovered each other by means of their IP addresses. Ceph has strict requirements on the discovery of Monitors.

Although the `ceph.conf` file is used by Ceph clients and other Ceph daemons to discover Monitors, the monitor map is used by Monitors to discover each other. This is why it is necessary to obtain the current `monmap` at the time a new Monitor is created: as can be seen above in [Adding a Monitor \(Manual\)](#), the `monmap` is one of the arguments required by the `ceph-mon -i {mon-id} --mkfs` command. The following sections explain the consistency requirements for Ceph Monitors, and also explain a number of safe ways to change a Monitor's IP address.

CONSISTENCY REQUIREMENTS

When a Monitor discovers other Monitors in the cluster, it always refers to the local copy of the Monitor map. Using the Monitor map instead of using the `ceph.conf` file avoids errors that could break the cluster (for example, typos or other slight errors in `ceph.conf` when a Monitor address or port is specified). Because Monitors use monitor maps for discovery and because they share monitor maps with Ceph clients and other Ceph daemons, the monitor map 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.

There are additional advantages to using the monitor map rather than `ceph.conf` when Monitors discover each other. Because `ceph.conf` is not automatically updated and distributed, its use would bring certain risks: Monitors might use an outdated `ceph.conf` file, might fail to recognize a specific Monitor, might fall out of quorum, and might develop a situation in which [Paxos](#) is unable to accurately ascertain the current state of the system. Because of these risks, any changes to an existing Monitor's IP address must be made with great care.

If a Monitor's IP address is changed only in the `ceph.conf` file, there is no guarantee that the other Monitors in the cluster will receive the update. For this reason, the preferred method to change a Monitor's IP address is as follows: add a new Monitor with the desired IP address (as described in [Adding a Monitor \(Manual\)](#)), make sure that the new Monitor successfully joins the quorum, remove the Monitor that is using the old IP address, and update the `ceph.conf` file to ensure that clients and other daemons are made aware of the new Monitor's IP address.

For example, suppose that there are three Monitors in place:

```
[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` so that its name is `host04` and its IP address is `10.0.0.4`: (1) follow the steps in [Adding a Monitor \(Manual\)](#) to add a new Monitor `mon.d`, (2) make sure that `mon.d` is running before removing `mon.c` or else quorum will be broken, and (3) follow the steps in [Removing a Monitor \(Manual\)](#) to remove `mon.c`. To move all three Monitors to new IP addresses, repeat this process.

CHANGING A MONITOR'S IP ADDRESS (ADVANCED METHOD)

There are cases in which the method outlined in [Changing a Monitor's IP address \(Preferred Method\)](#) cannot be used. For example, it might be necessary to move the cluster's Monitors to a different network, to a different part of the datacenter, or to a different datacenter altogether. It is still possible to change the Monitors' IP addresses, but a different method must be used.

For such cases, a new monitor map with updated IP addresses for every Monitor in the cluster must be generated and injected on each Monitor. Although this method is not particularly easy, such a major migration is unlikely to be a routine task. As stated at the beginning of this section, existing Monitors are not supposed to change their IP addresses.

(Preferred Method). Suppose that all of the Monitors are to be moved from the `10.0.0.x` range to the `10.1.0.x` range, and that these networks are unable to communicate. Carry out the following procedure:

1. Retrieve the monitor map (`{tmp}` is the path to the retrieved monitor map, and `{filename}` is the name of the file that contains the retrieved monitor map):

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

2. Check the contents of the monitor map:

```
$ 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 from the monitor map:

```
$ 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 to the monitor map:

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

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

5. Check the new contents of the monitor map:

```
$ 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 that the Monitors (and stores) have been installed at the new location. Next, propagate the modified monitor map to the new Monitors, and inject the modified monitor map into each new Monitor.

1. Make sure all of your Monitors have been stopped. Never inject into a Monitor while the Monitor daemon is running.
2. Inject the monitor map:

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

3. Restart all of the Monitors.

Migration to the new location is now complete. The Monitors should operate successfully.

USING CEPHADM TO CHANGE THE PUBLIC NETWORK

OVERVIEW

The procedure in this overview section provides only the broad outlines of using `cephadm` to change the public network.

1. Create backups of all keyrings, configuration files, and the current monmap.
2. Stop the cluster and disable `ceph.target` to prevent the daemons from starting.
3. Move the servers and power them on.

EXAMPLE PROCEDURE

❗ Note

In this procedure, the “old network” has addresses of the form `10.10.10.0/24` and the “new network” has addresses of the form `192.168.160.0/24`.

1. Enter the shell of the first Monitor:

```
# cephadm shell --name mon.reef1
```

2. Extract the current monmap from `mon.reef1`:

```
# ceph-mon -i reef1 --extract-monmap monmap
```

3. Print the content of the monmap:

```
# monmaptool --print monmap
```

```
monmaptool: monmap file monmap
epoch 5
fsid 2851404a-d09a-11ee-9aaa-fa163e2de51a
last_changed 2024-02-21T09:32:18.292040+0000
created 2024-02-21T09:18:27.136371+0000
min_mon_release 18 (reef)
election_strategy: 1
0: [v2:10.10.10.11:3300/0,v1:10.10.10.11:6789/0] mon.reef1
1: [v2:10.10.10.12:3300/0,v1:10.10.10.12:6789/0] mon.reef2
2: [v2:10.10.10.13:3300/0,v1:10.10.10.13:6789/0] mon.reef3
```

4. Remove Monitors with old addresses:

```
# monmaptool --rm reef1 --rm reef2 --rm reef3 monmap
```

5. Add Monitors with new addresses:

```
reef2 [v2:192.168.160.12:3300/0,v1:192.168.160.12:6789/0] --addv reef3  
[v2:192.168.160.13:3300/0,v1:192.168.160.13:6789/0] monmap
```

6. Verify that the changes to the monmap have been made successfully:

```
# monmaptool --print monmap
```

```
monmaptool: monmap file monmap  
epoch 4  
fsid 2851404a-d09a-11ee-9aaa-fa163e2de51a  
last_changed 2024-02-21T09:32:18.292040+0000  
created 2024-02-21T09:18:27.136371+0000  
min_mon_release 18 (reef)  
election_strategy: 1  
0: [v2:192.168.160.11:3300/0,v1:192.168.160.11:6789/0] mon.reef1  
1: [v2:192.168.160.12:3300/0,v1:192.168.160.12:6789/0] mon.reef2  
2: [v2:192.168.160.13:3300/0,v1:192.168.160.13:6789/0] mon.reef3
```

7. Inject the new monmap into the Ceph cluster:

```
# ceph-mon -i reef1 --inject-monmap monmap
```

8. Repeat the steps above for all other Monitors in the cluster.

9. Update `/var/lib/ceph/{FSID}/mon.{MON}/config`.

10. Start the Monitors.

11. Update the ceph `public_network`:

```
# ceph config set mon public_network 192.168.160.0/24
```

12. Update the configuration files of the Managers (`/var/lib/ceph/{FSID}/mgr.{mgr}/config`) and start them. Orchestrator will now be available, but it will attempt to connect to the old network because the host list contains the old addresses.

13. Update the host addresses by running commands of the following form:

```
# ceph orch host set-addr reef1 192.168.160.11  
# ceph orch host set-addr reef2 192.168.160.12  
# ceph orch host set-addr reef3 192.168.160.13
```

14. Wait a few minutes for the orchestrator to connect to each host.

15. Reconfigure the OSDs so that their config files are automatically updated:

The `<service_id>` value can be obtained from the output of the command `ceph orch ls`
`osd`.

The above procedure was developed by Eugen Block and was successfully tested in February 2024 on Ceph version 18.2.1 (Reef).

Brought to you by the Ceph Foundation

The Ceph Documentation is a community resource funded and hosted by the non-profit [Ceph Foundation](#). If you would like to support this and our other efforts, please consider [joining now](#).