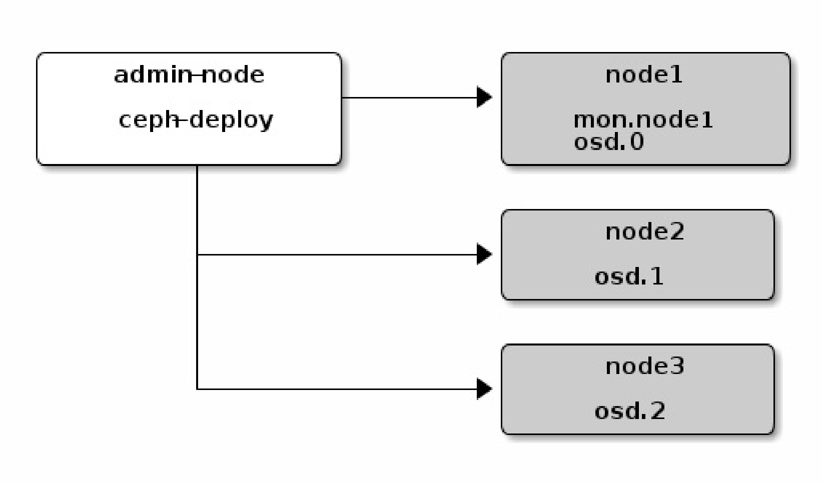
STORAGE CLUSTER QUICK START

If you haven't completed your Preflight Checklist, do that first. This **Quick Start** sets up a Ceph Storage Cluster using `ceph-deploy` on your admin node. Create a three Ceph Node cluster so you can explore Ceph functionality.



As a first exercise, create a Ceph Storage Cluster with one Ceph Monitor and three Ceph OSD Daemons. Once the cluster reaches a `active + clean` state, expand it by adding a fourth Ceph OSD Daemon, a Metadata Server and two more Ceph Monitors. For best results, create a directory on your admin node for maintaining the configuration files and keys that `ceph-deploy` generates for your cluster.

```
mkdir my-cluster
cd my-cluster
```

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

Important: 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.

STARTING OVER

If at any point you run into trouble and you want to start over, execute the following to purge the Ceph packages, and erase all its data and configuration:

```
ceph-deploy purge {ceph-node} [{ceph-node}]
ceph-deploy purgedata {ceph-node} [{ceph-node}]
ceph-deploy forgetkeys
rm ceph.*
```

If you execute `purge`, you must re-install Ceph. The last `rm` command removes any files that were written out by ceph-deploy locally during a previous installation.

CREATE A CLUSTER

On your admin node from the directory you created for holding your configuration details, perform the following steps using `ceph-deploy`.

1. Create the cluster.

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

   Specify node(s) as hostname, fqdn or hostname:fqdn. For example:

```
ceph-deploy new node1
```

Check the output of ceph-deploy with ls and cat in the current directory. You should see a Ceph configuration file (ceph.conf), a monitor secret keyring (ceph.mon.keyring), and a log file for the new cluster. See ceph-deploy new -h for additional details.

2. If you have more than one network interface, add the public network setting under the [global] section of your Ceph configuration file. See the Network Configuration Reference for details.

```
public network = {ip-address}/{bits}
```

For example,:

```
public network = 10.1.2.0/24
```

to use IPs in the 10.1.2.0/24 (or 10.1.2.0/255.255.255.0) network.

3. If you are deploying in an IPv6 environment, add the following to ceph.conf in the local directory:

```
echo ms bind ipv6 = true >> ceph.conf
```

4. Install Ceph packages.:

```
ceph-deploy install {ceph-node} [...]
```

For example:

```
ceph-deploy install node1 node2 node3
```

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

5. Deploy the initial monitor(s) and gather the keys:

```
ceph-deploy mon create-initial
```

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

- ceph.client.admin.keyring
- ceph.bootstrap-mgr.keyring
- ceph.bootstrap-osd.keyring
- ceph.bootstrap-mds.keyring
- ceph.bootstrap-rgw.keyring
- ceph.bootstrap-rbd.keyring

Note: If this process fails with a message similar to “Unable to find /etc/ceph/ceph.client.admin.keyring”, please ensure that the IP listed for the monitor node in ceph.conf is the Public IP, not the Private IP.

1. 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.

```
ceph-deploy admin {ceph-node(s)}
```

For example:

```
ceph-deploy admin node1 node2 node3
```

2. Deploy a manager daemon. (Required only for luminous+ builds):

```
ceph-deploy mgr create node1  *Required only for luminous+ builds, i.e >= 12.x builds*
```

3. Add three OSDs. For the purposes of these instructions, we assume you have an unused disk in each node called /dev/vdb. *Be sure that the device is not currently in use and does not contain any important data.*

    ceph-deploy osd create –data {device} {ceph-node}

   For example:

```
ceph-deploy osd create --data /dev/vdb node1
ceph-deploy osd create --data /dev/vdb node2
ceph-deploy osd create --data /dev/vdb node3
```

4. Check your cluster's health.

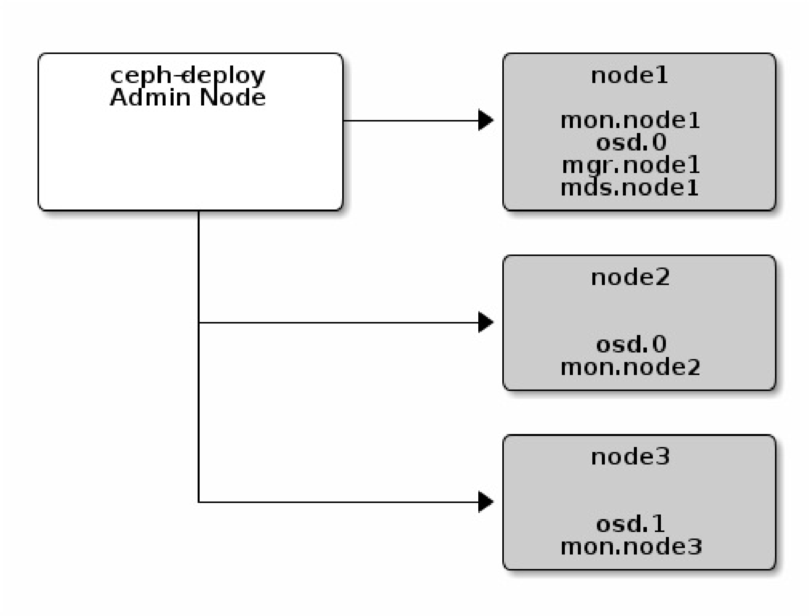
```
ssh node1 sudo ceph health
```

   Your cluster should report HEALTH_OK. You can view a more complete cluster status with:

```
ssh node1 sudo ceph -s
```

EXPANDING YOUR CLUSTER

Once you have a basic cluster up and running, the next step is to expand cluster. Add a Ceph Metadata Server to node1. Then add a Ceph Monitor and Ceph Manager to node2 and node3 to improve reliability and availability.



ADD A METADATA SERVER

To use CephFS, you need at least one metadata server. Execute the following to create a metadata server:

```
ceph-deploy mds create {ceph-node}
```

For example:

```
ceph-deploy mds create node1
```

ADDING MONITORS

A Ceph Storage Cluster requires at least one Ceph Monitor and Ceph Manager to run. For high availability, Ceph Storage Clusters typically run multiple Ceph Monitors so that the failure of a single Ceph Monitor will not bring down the Ceph Storage Cluster. Ceph uses the Paxos algorithm, which requires a majority of monitors (i.e., greather than *N/2* where *N* is the number of monitors) to form a quorum. Odd numbers of monitors tend to be better, although this is not required.

Add two Ceph Monitors to your cluster:

```
ceph-deploy mon add {ceph-nodes}
```

For example:

```
ceph-deploy mon add node2 node3
```

Once you have added your new Ceph Monitors, Ceph will begin synchronizing the monitors and form a quorum. You can check the quorum status by executing the following:

```
ceph quorum_status --format json-pretty
```

Tip: When you run Ceph with multiple monitors, you SHOULD install and configure NTP on each monitor host. Ensure that the monitors are NTP peers.

ADDING MANAGERS

The Ceph Manager daemons operate in an active/standby pattern. Deploying additional manager daemons ensures that if one daemon or host fails, another one can take over without interrupting service.

To deploy additional manager daemons:

```
ceph-deploy mgr create node2 node3
```

You should see the standby managers in the output from:

```
ssh node1 sudo ceph -s
```

ADD AN RGW INSTANCE

To use the Ceph Object Gateway component of Ceph, you must deploy an instance of RGW. Execute the following to create an new instance of RGW:

```
ceph-deploy rgw create {gateway-node}
```

For example:

```
ceph-deploy rgw create node1
```

By default, the RGW instance will listen on port 7480. This can be changed by editing ceph.conf on the node running the RGW as follows:

```
[client]
rgw frontends = civetweb port=80
```

To use an IPv6 address, use:

```
[client]
rgw frontends = civetweb port=[::]:80
```

STORING/RETRIEVING OBJECT DATA

To store object data in the Ceph Storage Cluster, a Ceph client must:

1. Set an object name
2. Specify a pool

The Ceph Client retrieves the latest cluster map and the CRUSH algorithm calculates how to map the object to a placement group, and then calculates how to assign the placement group to a Ceph OSD Daemon dynamically. To find the object location, all you need is the object name and the pool name. For example:

```
ceph osd map {poolname} {object-name}
```

Exercise: Locate an Object

As an exercise, lets create an object. Specify an object name, a path to a test file containing some object data and a pool name using the `rados put` command on the command line. For example:

```
echo {Test-data} > testfile.txt
ceph osd pool create mytest 8
rados put {object-name} {file-path} --pool=mytest
rados put test-object-1 testfile.txt --pool=mytest
```

To verify that the Ceph Storage Cluster stored the object, execute the following:

```
rados -p mytest ls
```

Now, identify the object location:

```
ceph osd map {pool-name} {object-name}
ceph osd map mytest test-object-1
```

Ceph should output the object’s location. For example:

```
osdmap e537 pool 'mytest' (1) object 'test-object-1' -> pg 1.d1743484 (1.4) -> up [1,0] act
```

To remove the test object, simply delete it using the `rados rm` command.

For example:

```
rados rm test-object-1 --pool=mytest
```

To delete the `mytest` pool:

```
ceph osd pool rm mytest
```

(For safety reasons you will need to supply additional arguments as prompted; deleting pools destroys data.)

As the cluster evolves, the object location may change dynamically. One benefit of Ceph’s dynamic rebalancing is that Ceph relieves you from having to perform data migration or balancing manually.