Comprehensive Guide to Deploying a Ceph Quincy Cluster

Introduction

Ceph is a highly scalable, distributed storage system designed to provide excellent performance, reliability, and scalability. It is widely used in cloud computing environments, providing object, block, and file storage in a single unified platform. This guide will walk you through the process of deploying a Ceph cluster with 3 nodes, enabling you to leverage its features for your storage needs.

Pre-requisites

Before you begin, ensure you have the following:

- **Servers**: You will need a set of servers to deploy the Ceph cluster. These servers should meet the minimum hardware requirements specified by Ceph.
- **Operating System**: Ceph supports various Linux distributions such as CentOS, Ubuntu, and Debian. Ensure that your servers are running a supported version of Linux.
- **Network Configuration**: Ensure that the servers are connected to a reliable network with sufficient bandwidth and low latency.
- **Root Access**: You will need root access to the servers for installing and configuring the Ceph software.

Hardware Requirement

Component	Minimum Requirement
Operating System	Ubuntu Server 20.04 LTS or later
CPU Cores	4 >= cores per node
RAM	8 GB >= per node
Storage	Free block device /dev/sdb per node for Ceph storage
Network Connectivity	Stable network connection between nodes
Network Interfaces	1 Ethernet interface for public network access
	1 Ethernet interface for cluster network communication

Deployment Strategies

- **Node Roles**: Each node will serve as an object storage, monitor daemon, and manager daemon, ensuring redundancy and fault tolerance.
- **Network Configuration**: Configure a private cluster network (192.168.10.0/24) for Ceph communication and a public network for client access as per DHCP.

Applications

Ceph is suitable for a wide range of applications, including:

- **Cloud Storage**: Store and retrieve objects efficiently with Ceph's scalable object storage capabilities.
- **Virtualization**: Integrate with virtualization platforms like OpenStack, Proxmox, and VMware to provide block storage for virtual machines.
- **Data Analytics**: Analyse large datasets by storing them in a distributed Ceph cluster, enabling parallel processing and high throughput.
- **Backup and Disaster Recovery**: Implement backup and disaster recovery solutions with Ceph's reliable and fault-tolerant storage architecture.

Setting up a Ceph Multi-node cluster

Overview

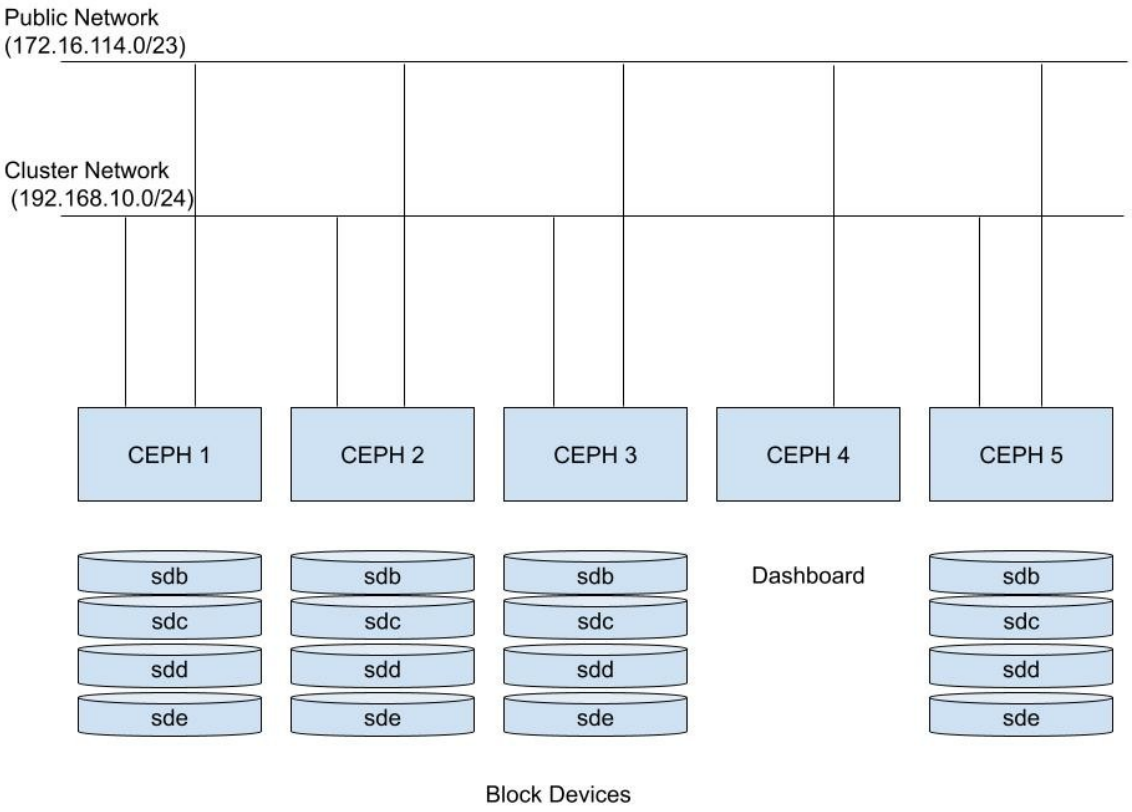
In this guide, we will walk through the process of setting up a Ceph cluster with 5 nodes. Each node will serve as an object storage, monitor daemon, or manager daemon. Additionally, each storage node will have a couple of free block devices like (/dev/sdb, /dev/sdc) for Ceph nodes.

Architecture

Cluster Architecture

Ceph Server	Public Network	Cluster Network (BOND)	Services	OSD	Storage (TB)
Ceph1	172.16.114.100	192.168.10.11	MON,OSD,MDS	4 OSD	14.4
Ceph2	172.16.114.205	192.168.10.12	MON,OSD,MDS	3 OSD	5.4
Ceph3	172.16.114.138	192.168.10.13	MON,OSD,MDS	3 OSD	1.1
Ceph4 (VM)	172.16.114.210	NIL	MGR		
Ceph5	172.16.114.98	192.168.10.14	MON,OSD	5 OSD	21.6

Cluster Diagram

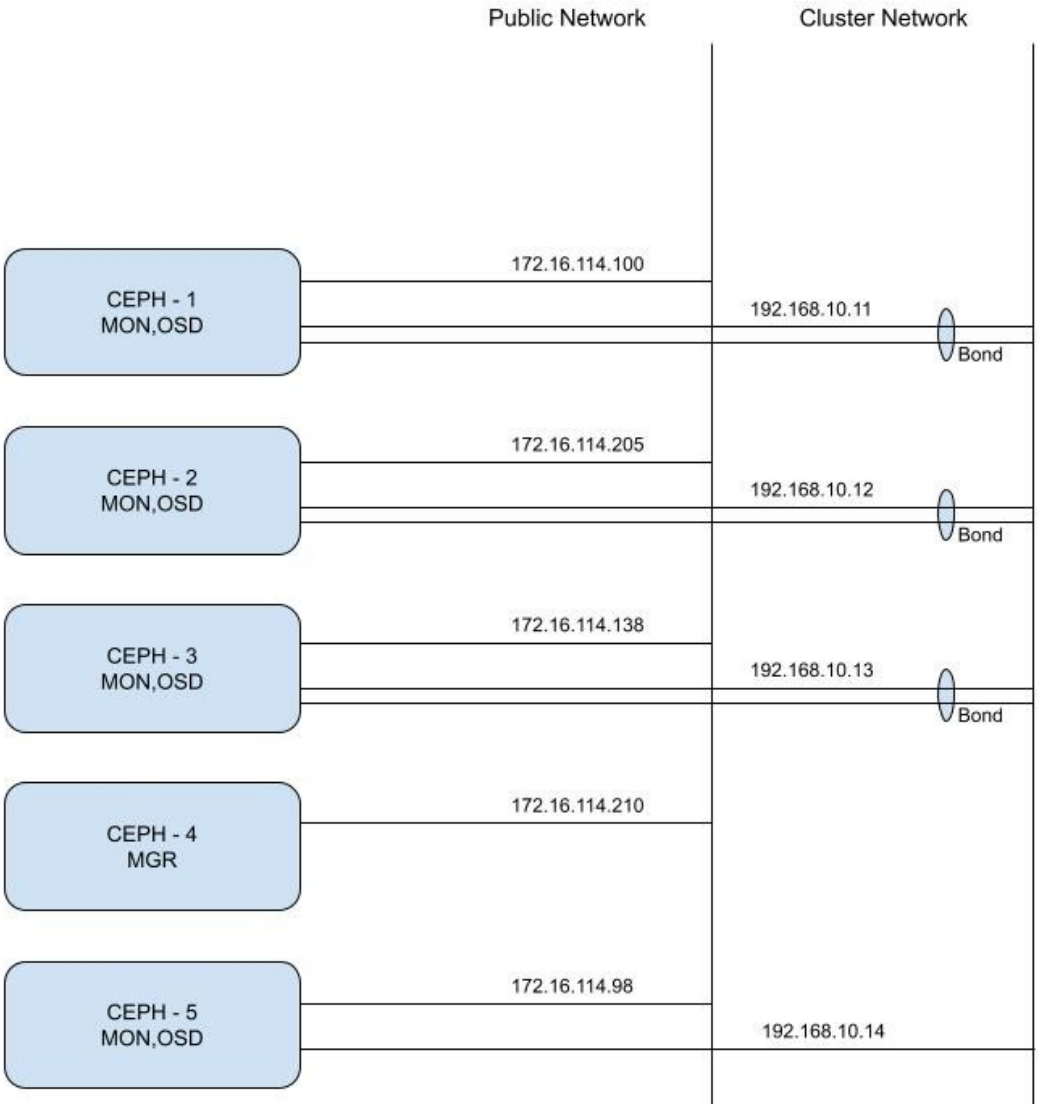


Explanation:

- **Ceph Server:** This column lists the names or identifiers of the Ceph servers within the cluster.
- **Public Network**: This column provides the IP addresses assigned to the public network interface of each Ceph server. The public network is typically used for client access and communication with external systems.
- **Cluster Network**: This column specifies the IP addresses assigned to the cluster network interface of each Ceph server. The cluster network is used for inter-node communication within the Ceph cluster. This network is configured as bond interface in each node.
- **Services**: This column indicates the services running on each Ceph server. Common services include MON (monitor), OSD (object storage daemon), MDS (metadata server), and MGR (manager). These services collectively enable the functionality of the Ceph storage cluster.
- **OSD**: This column specifies the number of OSDs (object storage daemons) running on each Ceph server. OSDs are responsible for managing the storage devices and storing data within the Ceph cluster.
- **Storage (TB)**: This column indicates the amount of storage capacity available on each Ceph server, typically measured in terabytes (TB). It represents the total storage capacity of the OSDs hosted on the respective server.

- **Ceph1**: It has public and cluster network interfaces with IP addresses 172.16.114.100 and 192.168.10.11 respectively. It runs MON, OSD, and MDS services, hosting 4 OSDs with a total storage capacity of 14.4 TB.
- **Ceph2**: Similar to Ceph1, it has public and cluster network interfaces with IP addresses 172.16.114.205 and 192.168.10.12 respectively. It also runs MON, OSD, and MDS services, hosting 3 OSDs with a total storage capacity of 5.4 TB.
- **Ceph3**: It follows a similar configuration pattern as Ceph1 and Ceph2, with public and cluster network interfaces, running MON, OSD, and MDS services. It hosts 3 OSDs with a total storage capacity of 1.1 TB.
- **Ceph4 (VM)**: Unlike the other servers, Ceph4 is identified as a virtual machine (VM). It has a public IP address but lacks a cluster network interface. It runs only the MGR service which is responsible for Ceph Dashboard.
- **Ceph5**: Similar to the others, it has public and cluster network interfaces, running MON and OSD services. It hosts 5 OSDs with a total storage capacity of 21.6 TB.

Network Diagram



Cluster Configuration

The provided commands are part of the process of setting up a Ceph cluster, specifically related to generating an SSH key pair and installing Ceph packages.

#Generate SSH key-pair on Ceph01 Node (call it Admin Node on here)

```
root@ceph01:~# ssh-keygen
```

Explanation:

- This command generates an SSH key pair on the "Ceph01" node, referred to as the "Admin Node" here. The ssh-keygen command prompts the user for inputs like the file location for the keys and passphrase (optional). It generates a public-private key pair in the default location (~/.ssh/id_rsa).

 # Install Ceph package

```
root@ceph01:~# apt update; apt -y install ceph
```

Explanation:

- apt update: This command updates the package repositories to fetch the latest package information.
- apt -y install ceph: Install the Ceph package on the "Admin Node." The -y flag automatically assumes "yes" to any prompts.

#Configure [Monitor Daemon], [Manager Daemon] on Admin Node.

```
root@ceph01:~# uuidgen
00889097-f1a1-4f7a-9bd4-4440848c8eaa
```

Explanation:

- uuidgen: Generates a new UUID (Universally Unique Identifier). UUIDs are often used in Ceph configurations to uniquely identify entities like monitors.
- The generated UUID (00889097-f1a1-4f7a-9bd4-4440848c8eaa) can be used in the Ceph configuration to uniquely identify the monitor daemon.

Note:
The provided examples and values, such as IP addresses, names, and IDs, are for reference purposes. In your Ceph configuration, these details may vary, and it is essential to use the actual values and names set up in your specific environment. Always consult official documentation for accurate information.

 # Create New Ceph Configuration File, on this example ⇒ [ceph.conf]

```
root@ceph01:~# vim /etc/ceph/ceph.conf

[global]
# specify cluster network for monitoring
cluster network = 192.168.1.0/24
# specify public network
public network = 172.16.114.0/23
```

```
# specify UUID genarated above
fsid = 00889097-f1a1-4f7a-9bd4-4440848c8eaa
# specify IP address of Monitor Daemon
mon host = 172.16.114.100
# specify Hostname of Monitor Daemon
mon initial members = ceph01
osd pool default crush rule = -1

# mon.(Node name)
[mon.ceph01]
# specify Hostname of Monitor Daemon
host = ceph01
# specify IP address of Monitor Daemon
mon addr = 172.16.114.100
# allow to delete pools
mon allow pool delete = true
```

Explanation:

- **vim /etc/ceph/ceph.conf** : This command opens the Vim text editor to create/edit the Ceph configuration file named ceph.conf. You can use any text editor of your choice, such as nano or gedit.
- The configuration file is structured with sections enclosed in square brackets ([]), specifying different components and their configurations.
- **[global] Section:**
  - **cluster network and public network:** Specify the cluster and public networks for monitoring.
  - **fsid**: Specifies the UUID generated earlier.
  - **mon host:** Specifies the IP address of the Monitor Daemon.
  - **mon initial members**: Specifies the hostname of the Monitor Daemon.
  - **osd pool default crush rule**: Specifies the crush rule for the default OSD pool.
- **[mon.ceph01] Section:**
  - Specifies the configuration for the Monitor Daemon on the node named "ceph01."
  - **host**: Specifies the hostname of the Monitor Daemon.
  - **mon addr**: Specifies the IP address of the Monitor Daemon.
  - **mon allow pool delete:** Allows deleting pools.

Note:

The ceph.conf file is a crucial configuration file for Ceph, defining various settings for the cluster. The fsid is the UUID generated earlier and is essential for uniquely identifying the cluster. The mon host and mon initial members settings specify the IP address and hostname of the Monitor Daemon. The mon.ceph01 section specifies the configuration for the Monitor Daemon on the node named "ceph01." The mon addr parameter is important for the Monitor Daemon to know its IP address. The mon allow pool delete setting allows pool deletion, which may be useful depending on cluster management requirements. After creating and editing the ceph.conf file, additional steps such as deploying monitors and OSDs are typically performed to complete the Ceph cluster setup.

 # Generate secret key for Cluster monitoring

```
root@ceph01:~# ceph-authtool --create-keyring
/etc/ceph/ceph.mon.keyring --gen-key -n mon. --cap mon 'allow
*'
```

The provided command generates a secret key for cluster monitoring and uses the ceph-authtool utility to create a keyring file. Here's an explanation with notes:

Explanation:

- **ceph-authtool**: This command is part of the Ceph utilities and is used for managing Ceph authentication keys.
- **--create-keyring /etc/ceph/ceph.mon.keyring**: Specifies the path and filename for the keyring file that will be created. In this case, it's /etc/ceph/ceph.mon.keyring.
- **--gen-key**: Generates a new key for the specified entity.
- **-n mon**.: Specifies the name of the entity for which the key is generated. In this case, it's for the monitor daemon (mon.).
- **--cap mon 'allow *'**: Sets capabilities for the monitor daemon. It grants the monitor daemon full permissions (allow *), which is necessary for monitoring operations.

Note:
The generated secret key is stored in the keyring file specified by the --create-keyring option (/etc/ceph/ceph.mon.keyring). This keyring file will be used by the monitor daemon during the Ceph cluster setup. The --cap mon 'allow *' setting grants the monitor daemon the necessary capabilities to perform monitoring tasks within the Ceph cluster. Proper management of secret keys is crucial for ensuring the security and integrity of the Ceph cluster. Keys should be stored securely and not shared unnecessarily. After generating the keyring file, additional steps such as deploying monitors and configuring authentication are typically performed to complete the Ceph cluster setup.

```
#Generate secret key for Cluster admin
```

```
root@ceph01:~# ceph-authtool --create-keyring
/etc/ceph/ceph.client.admin.keyring --gen-key -n client.admin
--cap mon 'allow *' --cap osd 'allow *' --cap mds 'allow *' --
cap mgr 'allow *'
creating /etc/ceph/ceph.client.admin.keyring
```

The provided command generates a secret key for the Cluster admin and uses the ceph-authtool utility to create a keyring file. Here's an explanation with notes:

Explanation:

- **ceph-authtool:** This command is part of the Ceph utilities and is used for managing Ceph authentication keys.
- **--create-keyring /etc/ceph/ceph.client.admin.keyring:** Specifies the path and filename for the keyring file that will be created. In this case, it's /etc/ceph/ceph.client.admin.keyring.
- **--gen-key**: Generates a new key for the specified entity.
- **-n client.admin**: Specifies the name of the entity for which the key is generated. In this case, it's for the Cluster admin (client.admin).
- **--cap mon 'allow *'**: Sets capabilities for the monitor daemon. It grants the Cluster admin full permissions (allow *) for monitoring operations.
- **--cap osd 'allow *'**: Sets capabilities for OSD (Object Storage Daemon). It grants the Cluster admin full permissions for OSD operations.
- **--cap mds 'allow *'**: Sets capabilities for MDS (Metadata Server). It grants the Cluster admin full permissions for MDS operations.
- **--cap mgr 'allow *'**: Sets capabilities for the Manager Daemon. It grants the Cluster admin full permissions for manager operations.

Note:

The generated secret key is stored in the keyring file specified by the --create-keyring option (/etc/ceph/ceph.client.admin.keyring). This keyring file will be used by the Cluster admin for various administrative operations within the Ceph cluster. The --cap options specify the capabilities granted to the Cluster admin for different Ceph components. In this case, the admin has full permissions (allow *) for monitoring, OSD, MDS, and manager operations. Proper management of secret keys is crucial for ensuring the security and integrity of the Ceph cluster. Keys should be stored securely and not shared unnecessarily.

After generating the keyring file for the Cluster admin, it can be used for administrative tasks such as configuring and managing the Ceph cluster. The provided examples and values, such as IP addresses, names, and IDs, are for reference purposes. In your Ceph configuration, these details may vary, and it is essential to use the actual values and names set up in your specific environment. Always consult official documentation for accurate information.

```
# Generate key for bootstrap

root@ceph01:~#          ceph-authtool          --create-keyring
/var/lib/ceph/bootstrap-osd/ceph.keyring      --gen-key      -n
client.bootstrap-osd  --cap  mon  'profile  bootstrap-osd'  --cap
mgr 'allow r'
creating /var/lib/ceph/bootstrap-osd/ceph.keyring
```

The provided command generates a secret key for the Ceph bootstrap process, specifically for OSD (Object Storage Daemon), and uses the ceph-authtool utility to create a keyring file. Here's an explanation with notes:

Explanation:

- **ceph-authtool**: This command is part of the Ceph utilities and is used for managing Ceph authentication keys.
- **--create-keyring /var/lib/ceph/bootstrap-osd/ceph.keyring**: Specifies the path and filename for the keyring file that will be created. In this case, it's /var/lib/ceph/bootstrap-osd/ceph.keyring.
- **--gen-key**: Generates a new key for the specified entity.
- **-n client**.bootstrap-osd: Specifies the name of the entity for which the key is generated. In this case, it's for the bootstrap process of OSD (client.bootstrap-osd).
- **--cap mon 'profile bootstrap-osd'**: Sets capabilities for the monitor daemon. It grants specific capabilities tailored for the bootstrap process of OSD.
- **--cap mgr 'allow r'**: Sets capabilities for the Manager Daemon. It grants read permissions (allow r) for manager operations.

Note:

The generated secret key is stored in the keyring file specified by the --create-keyring option (/var/lib/ceph/bootstrap-osd/ceph.keyring). This keyring file is used during the bootstrap process of the OSD. OSD nodes use this key during their initial setup to authenticate themselves with the Ceph cluster. The --cap mon 'profile bootstrap-osd' setting specifies a set of capabilities specifically tailored for the bootstrap process of OSD. It includes the necessary permissions for the OSD to register itself with the monitor. The --cap mgr 'allow r' setting grants read permissions for manager operations. This is relevant during the bootstrap process for OSD. Proper management of secret keys is crucial for ensuring the security and integrity of the Ceph cluster. Keys should be stored securely and not shared unnecessarily. After generating the keyring file for the bootstrap OSD, it is typically used in the deployment and configuration of OSD nodes in the Ceph cluster. The provided examples and values, such as IP addresses, names, and IDs, are for reference purposes. In your Ceph configuration, these details may vary, and it is essential to use the actual values and names set up in your specific environment. Always consult official documentation for accurate information.

```
# Import generated key

root@ceph01:~#  ceph-authtool  /etc/ceph/ceph.mon.keyring  --
import-keyring /etc/ceph/ceph.client.admin.keyring
```

```
importing contents of /etc/ceph/ceph.client.admin.keyring into
/etc/ceph/ceph.mon.keyring

root@ceph01:~#  ceph-authtool  /etc/ceph/ceph.mon.keyring  --
import-keyring /var/lib/ceph/bootstrap-osd/ceph.keyring
importing contents of /var/lib/ceph/bootstrap-osd/ceph.keyring
into /etc/ceph/ceph.mon.keyring
```

The provided commands use the ceph-authtool utility to import the contents of one keyring file into another. Here's an explanation with notes:

Import Generated Keys:
root@ceph01:~# ceph-authtool /etc/ceph/ceph.mon.keyring --import-keyring /etc/ceph/ceph.client.admin.keyring

Explanation:

- **ceph-authtool**: This command is used for managing Ceph authentication keys.
- **/etc/ceph/ceph.mon.keyring**: Specifies the destination keyring file where the contents will be imported.
- **--import-keyring /etc/ceph/ceph.client.admin.keyring**: Specifies the source keyring file from which contents will be imported.

Note:
This command imports the contents of the ceph.client.admin.keyring into the ceph.mon.keyring. It's a common practice to consolidate keyrings for ease of management.

Import Bootstrap OSD Keyring into Monitor Keyring:
root@ceph01:~# ceph-authtool /etc/ceph/ceph.mon.keyring --import-keyring /var/lib/ceph/bootstrap-osd/ceph.keyring

Explanation:

Similar to the previous command, this one imports the contents of the ceph.keyring for the bootstrap OSD into the ceph.mon.keyring.

Note:
The Monitor Daemon (mon.) keyring is a central keyring that often contains keys for different entities involved in Ceph operations. Importing keys into the Monitor Daemon keyring ensures that the Monitor Daemon can authenticate and authorize actions performed by other components of the Ceph cluster. The bootstrap OSD keyring is specifically used during the initial setup of OSD nodes, allowing them to authenticate with the Ceph cluster. Keyring files store authentication keys and capabilities necessary for secure communication and operation within the Ceph cluster. Proper management of keyrings is crucial for maintaining the security and integrity of the Ceph cluster. Keys should be imported and distributed appropriately based on the roles and responsibilities within the cluster.

 # Generate monitor map

```
root@ceph01:~# FSID=$(grep "^fsid" /etc/ceph/ceph.conf | awk
{'print $NF'})

root@ceph01:~#       NODENAME=$(grep       "^mon       initial"
/etc/ceph/ceph.conf | awk {'print $NF'})

root@ceph01:~# NODEIP=$(grep "^mon host" /etc/ceph/ceph.conf |
awk {'print $NF'})
```

```
root@ceph01:~# monmaptool --create --add $NODENAME $NODEIP --
fsid $FSID /etc/ceph/monmap

monmaptool: monmap file /etc/ceph/monmap
monmaptool: set fsid to 00889097-f1a1-4f7a-9bd4-4440848c8eaa
monmaptool: writing epoch 0 to /etc/ceph/monmap (1 monitors)
```

The provided commands generate a Ceph monitor map using the monmaptool utility. Here's an explanation with notes:

Explanation:

- **grep "^fsid" /etc/ceph/ceph.conf**: Searches for the line in the Ceph configuration file that starts with "fsid."
- **awk {'print $NF'}**: Extracts the last field (value) from the selected line.
- These commands extract the FSID, Node Name, and Node IP from the Ceph configuration file.

Use monmaptool to Create Monitor Map. Explanation:

- **monmaptool**: This command is used for managing Ceph monitor maps.
- **--create**: Specifies that a new monitor map should be created.
- **--add $NODENAME $NODEIP**: Adds a monitor to the monitor map. It uses the extracted Node Name and Node IP.
- **--fsid $FSID**: Sets the FSID for the monitor map.
- **/etc/ceph/monmap**: Specifies the path and filename for the generated monitor map.

Notes:
The monitor map is a critical component in the Ceph cluster, as it contains information about the monitors and their locations in the cluster. The monmaptool command creates a monitor map file (/etc/ceph/monmap) based on the provided information. The monitor map is essential for other nodes in the Ceph cluster to locate and communicate with the monitors. Proper generation and distribution of the monitor map are crucial for the correct functioning of the Ceph cluster. The map must be kept up-to-date as the cluster evolves.

 # Create a directory for Monitor Daemon in this case directory name ⇒ (Cluster Name)-(Node Name)

```
root@ceph01:~# mkdir /var/lib/ceph/mon/ceph-ceph01
```

The provided command creates a directory for the Monitor Daemon on the node named "ceph01." Here's an explanation with notes:

Explanation:

- **mkdir**: This command is used to create a new directory.
- **/var/lib/ceph/mon/ceph-ceph01**: Specifies the full path for the directory to be created.
- **/var/lib/ceph**: This is a common location for storing Ceph-related data, including monitor data.
- **/mon**: This subdirectory is used to store monitor-related data.
- **/ceph-ceph01**: The name of the directory, typically composed of the Ceph cluster name and the node name.

Notes:
The directory structure under /var/lib/ceph/mon/ is used to store monitor-related data on each node in the Ceph cluster. Creating a separate directory for each Monitor Daemon ensures proper organization of data. The name of the directory often

follows a convention, including the Ceph cluster name and the node name, to uniquely identify monitor data for that node. This directory will be used to store monitor-specific information, such as keyrings, maps, and other configuration files. Proper organisation and management of directories are essential for the correct functioning of the Ceph cluster. Each Monitor Daemon requires its own directory to store its data locally on the node.

```
# Associate key and monmap to Monitor Daemon

root@ceph01:~# ceph-mon --cluster ceph --mkfs -i $NODENAME --
monmap /etc/ceph/monmap --keyring /etc/ceph/ceph.mon.keyring

root@ceph01:~# chown ceph. /etc/ceph/ceph.*

root@ceph01:~# chown -R ceph. /var/lib/ceph/mon/ceph-ceph01
/var/lib/ceph/bootstrap-osd

root@ceph01:~# systemctl enable --now ceph-mon@$NODENAME
```

The provided commands associate keys and monitor maps to the Monitor Daemon and configure it for the Ceph cluster. Here's an explanation with notes.

Associate Key and Monmap to Monitor Daemon and Running ceph-mon to Initialise Monitor Daemon

Explanation:

- ceph-mon: This command is used for managing Ceph Monitor Daemons.
- --cluster ceph: Specifies the Ceph cluster name.
- --mkfs: Initiates the Monitor Daemon and creates necessary files and directories.
- -i $NODENAME: Specifies the name of the Monitor Daemon, which is derived from the extracted node name.
- --monmap /etc/ceph/monmap: Specifies the path to the monitor map file.
- --keyring /etc/ceph/ceph.mon.keyring: Specifies the path to the keyring file for the Monitor Daemon.

Change Ownership of Key and Configuration Files:

Explanation:

- chown: This command changes the owner of files or directories.
- ceph.: Specifies the new owner (user and group) for the files.
- /etc/ceph/ceph.*: Matches all files starting with "ceph" in the /etc/ceph/ directory.

Change Ownership of Monitor and Bootstrap OSD Directories:

Explanation:

- -R: Recursively changes ownership for all files and directories within the specified directories.
- /var/lib/ceph/mon/ceph-ceph01 /var/lib/ceph/bootstrap-osd: Specifies the directories whose ownership needs to be changed.

Enable and Start Ceph Monitor Service:

Explanation:

- systemctl enable: Enables the Ceph Monitor service to start on boot.

- --now ceph-mon@$NODENAME: Starts the Ceph Monitor service immediately for the specified Monitor Daemon.

Notes:
The ceph-mon command initializes and configures the Monitor Daemon with the provided monitor map and keyring. Changing ownership of key and configuration files to the ceph user ensures that the Ceph processes have the necessary permissions to access these files. Changing ownership of monitor and bootstrap OSD directories to the ceph user ensures proper access to data directories. Enabling and starting the Ceph Monitor service ensures that the Monitor Daemon is running and actively participating in the Ceph cluster. Proper configuration and management of the Monitor Daemon are crucial for the stability and functionality of the Ceph cluster. The Monitor Daemon plays a central role in maintaining cluster state and facilitating communication among cluster nodes.

 # Enable Messenger v2 Protocol

```
root@ceph01:~# ceph mon enable-msgr2
```

The provided command enables the Messenger v2 protocol for Ceph Monitors. Here's an explanation with notes:

Explanation:

- **ceph mon enable-msgr2**: This command is used to enable the Messenger v2 protocol specifically for Ceph Monitors.

Notes:
The Messenger v2 protocol is the second version of the Ceph Messenger protocol, which is responsible for communication between Ceph nodes in the cluster. Enabling Messenger v2 is part of the process of upgrading the Ceph cluster to use the latest communication protocol. Messenger v2 provides improvements and optimizations over the previous version, including enhanced performance and reduced latency. Enabling Messenger v2 typically involves updating the configuration and restarting Ceph components to ensure that the entire cluster uses the new protocol. It's important to follow the recommended upgrade procedures and consult the Ceph documentation for any additional steps or considerations when enabling Messenger v2. This command is typically run on the Ceph Monitor node as part of the upgrade process. Ensure that all nodes in the cluster are compatible with Messenger v2 before enabling it. Enabling the Messenger v2 protocol is a step towards keeping the Ceph cluster up-to-date with the latest features and improvements in Ceph communication.

 # Enable Placement Groups auto scale module

```
root@ceph01:~# ceph mgr module enable pg_autoscaler
```

The provided command enables the Placement Groups (PG) auto scaler module in the Ceph Manager Daemon (mgr). Here's an explanation with notes:

Explanation:

- **ceph mgr module enable pg_autoscaler**: This command enables the Placement Groups auto scaler module within the Ceph Manager Daemon.

Notes:
Placement Groups (PGs) in Ceph are logical units used for distributing and replicating data across OSDs (Object Storage Daemons). The PG auto scaler module is a Ceph Manager module designed to automatically adjust the number of Placement Groups based on the size of the Ceph cluster. This helps optimise the performance and distribution of data in the cluster. Enabling the PG auto scaler module allows the Ceph Manager to dynamically scale the number of PGs, which is beneficial for clusters that experience changes in size or workload. Auto-scaling PGs can be advantageous in large and dynamic Ceph clusters, as it helps avoid manual intervention and ensures efficient utilisation of resources. As with any Ceph configuration changes, it's important to monitor the cluster's behaviour after enabling the auto scaler module to ensure that it aligns with the cluster's performance and capacity requirements. Refer to the Ceph documentation for further details on configuring and tuning the PG auto scaler module based on specific cluster characteristics and needs.
Enabling the PG auto scaler module is a step toward automating and optimising the PG configuration in Ceph, enhancing the adaptability of the cluster to changing conditions.

Configure Ceph Manager on Node4 (Ceph04)

 # Befor configuring the ceph-mgr copy the Ceph conf and other relatd files into ceph04 node

```
root@ceph01:~#scp /etc/ceph/* ceph04:~/
```

 # Create a directory for Manager Daemon in this case directory name ⇒ (Cluster Name)-(Node Name)

```
root@ceph04:~# mkdir /var/lib/ceph/mgr/ceph-ceph04
```

The provided command creates a directory for the Manager Daemon on the node named "ceph04." Here's an explanation with notes:

Explanation:

- **mkdir**: This command is used to create a new directory.
- **/var/lib/ceph/mgr/ceph-ceph04**: Specifies the full path for the directory to be created.
- **/var/lib/ceph**: This is a common location for storing Ceph-related data, including manager data.
- **/mgr**: This subdirectory is used to store manager-related data.
- **/ceph-ceph04**: The name of the directory, typically composed of the Ceph cluster name and the node name.

Notes:

The directory structure under /var/lib/ceph/mgr/ is used to store manager-related data on each node in the Ceph cluster. Creating a separate directory for each Manager Daemon ensures proper organization of data. The name of the directory often follows a convention, including the Ceph cluster name and the node name, to uniquely identify manager data for that node. This directory will be used to store manager-specific information, such as keyrings, configuration files, and other data. Proper organisation and management of directories are essential for the correct functioning of the Ceph cluster. Each Manager Daemon requires its own directory to store its data locally on the node.

#Create auth key and start the service

```
root@ceph04:~# ceph auth get-or-create mgr.ceph04 mon 'allow profile mgr' osd 'allow *' mds 'allow *'

[mgr.ceph04]
     key = AQAaQ+Nk8Eg1EhAAjpoJVWWsqzkER2z84b050Q==

root@ceph04:~# ceph auth get-or-create mgr.ceph04 | tee /etc/ceph/ceph.mgr.admin.keyring
root@ceph04:~# cp /etc/ceph/ceph.mgr.admin.keyring /var/lib/ceph/mgr/ceph-ceph04/keyring
root@ceph04:~# chown ceph. /etc/ceph/ceph.mgr.admin.keyring
root@ceph04:~# chown -R ceph. /var/lib/ceph/mgr/ceph-ceph04
root@ceph04:~# systemctl enable --now ceph-mgr@ceph04
```

The provided commands create and configure the authentication key for the Ceph Manager Daemon. Here's an explanation with notes:

Generate Authentication Key:
Explanation:

- ceph auth get-or-create mgr.$NODENAME: This command generates or retrieves the authentication key for the Ceph Manager Daemon, identified by its node name.
- mon 'allow profile mgr' osd 'allow *' mds 'allow *': Specifies the capabilities granted to the Manager Daemon, allowing it to interact with monitors, OSDs, and MDSs.

Display and Store the Authentication Key:
Explanation:

- The authentication key is displayed in the output. It is in the form of a key-value pair.

Save Authentication Key to a Keyring File:
Explanation:

- tee /etc/ceph/ceph.mgr.admin.keyring: Saves the authentication key to the specified keyring file and also displays it in the terminal.

Copy Keyring File to Manager Daemon Directory:

- root@ceph04:~# cp /etc/ceph/ceph.mgr.admin.keyring /var/lib/ceph/mgr/ceph-ceph04/keyring

Change Ownership of Keyring File and Directory:
Explanation:

- The ownership of the keyring file is changed to the ceph user to ensure that Ceph processes have the necessary permissions to access it.
- The ownership of the Manager Daemon directory is recursively changed to the ceph user for proper access to manager-related data.

Enable and Start Ceph Manager Service:
Explanation:

- systemctl enable --now ceph-mgr@ceoh04: Enables and starts the Ceph Manager service for the specified Manager Daemon.

Notes:
The authentication key is a crucial component for securing communication between Ceph components. It should be kept confidential and not shared unnecessarily. The keyring file (ceph.mgr.admin.keyring) contains the authentication key for the Manager Daemon and is stored in a secure location. Changing ownership of the keyring file and directory ensures that the Ceph processes have the necessary permissions to access the authentication key and manager-related data. Enabling and starting the Ceph Manager service ensures that the Manager Daemon is running and actively participating in the Ceph cluster. Proper configuration and management of authentication keys are essential for securing and maintaining the integrity of the Ceph cluster. The provided examples and values, such as IP addresses, names, and IDs, are for reference purposes. In your Ceph configuration, these details may vary, and it is essential to use the actual values and names set up in your specific environment. Always consult official documentation for accurate information.

 # Return to node Ceph01 and confirm Cluster status. That's OK if [Monitor Daemon] and [Manager Daemon] are enabled like the following.

```
root@ceph01:~# ceph -s
  cluster:
    id:     00889097-f1a1-4f7a-9bd4-4440848c8eaa
    health: HEALTH_WARN
            mon is allowing insecure global_id reclaim
   services:
    mon: 1 daemons, quorum ceph01 (age 66s)
    mgr: no daemons active
    osd: 0 osds: 0 up, 0 in
```

```
data:
 pools:   0 pools, 0 pgs
 objects: 0 objects, 0 B
 usage:   0 B used, 0 B / 0 B avail
 pgs:
```

The provided command (ceph -s) is used to check the status of the Ceph cluster. The output indicates the current state of various components in the cluster. Here's an explanation with notes

Confirm Cluster Status:
Explanation:

- ceph -s: This command queries and displays the current status of the Ceph cluster.

Interpretation and Notes:
cluster Section:

- id: The unique identifier of the Ceph cluster.
- health: The overall health of the cluster. In this example, it shows a warning (HEALTH_WARN), indicating a specific issue related to insecure global_id reclaim in the Monitor Daemon.

services Section:

- mon: Indicates the status of Monitor Daemons.
- 1 daemons: There is one Monitor Daemon in the cluster.
- quorum ceph01: The monitor named ceph01 is part of the quorum (a subset of monitors that collectively agree on the state of the cluster).
- age 66s: The age of the monitor daemon, indicating how long it has been running.
- mgr: Indicates the status of Manager Daemons.
- no daemons active: There are no active Manager Daemons. This could be intentional if managers are not needed for the current configuration.
- osd: Indicates the status of Object Storage Daemons (OSDs).
- 0 osds: There are no OSDs in the cluster.
- 0 up, 0 in: None of the OSDs are up and in the cluster.

data Section:

- pools: Indicates the number of pools and placement groups (pgs) in the cluster. In this example, there are 0 pools and 0 pgs.
- objects: Indicates the number of objects and their total size in bytes. In this example, there are 0 objects and 0 bytes.
- usage: Indicates the storage usage, showing the amount used, available, and total. In this example, 0 B is used, and the total available is 0 B.
- pgs: Indicates the number of placement groups. In this example, there are no placement groups.

Notes:
The HEALTH_WARN status indicates a warning, and it's essential to investigate the specific issue mentioned (insecure global_id reclaim) to ensure proper cluster operation. The absence of active Manager Daemons (mgr: no daemons active) is acceptable if managers are not required for the current cluster configuration. The absence of OSDs (osd: 0 osds: 0 up, 0 in) is also acceptable if OSDs are yet to be added to the cluster. The absence of pools and placement groups (pools: 0 pools, 0 pgs) is expected if the cluster is newly initialized.
It's common to see warnings during the initial setup, and addressing them may involve configuring or adjusting specific parameters in the Ceph cluster. Further investigation and configuration adjustments may be necessary based on the specific

needs and requirements of the Ceph cluster. Proper understanding of the ceph -s output is essential for monitoring and troubleshooting the health and configuration of the Ceph cluster. The provided examples and values, such as IP addresses, names, and IDs, are for reference purposes. In your Ceph configuration, these details may vary, and it is essential to use the actual values and names set up in your specific environment. Always consult official documentation for accurate information.

 # Configure OSD (Object Storage Device) to each Node from Admin Node (For standalone do in same node). Make sure the separate storage available in node (in this case we have /dev/vdb)

```
root@ceph01:~# lsblk
NAME                      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0                       7:0    0 63.3M  1 loop
/snap/core20/1828
loop1                       7:1    0 63.5M  1 loop
/snap/core20/2015
loop2                       7:2    0 91.9M  1 loop
/snap/lxd/24061
loop3                       7:3    0 53.3M  1 loop
/snap/snapd/19457
loop4                       7:4    0 49.9M  1 loop
/snap/snapd/18357
sr0                        11:0    1 1024M  0 rom
vda                       252:0    0   25G  0 disk
├─vda1                    252:1    0    1M  0 part
├─vda2                    252:2    0    2G  0 part /boot
└─vda3                    252:3    0   23G  0 part
  └─ubuntu--vg-ubuntu--lv 253:0    0 11.5G  0 lvm  /
vdb                       252:16   0  100G  0 disk
```

The provided output (lsblk) displays the block devices on the system, including the available storage devices. Here's an explanation with notes:

Explanation:
Interpretation and Notes:
NAME Column:

- loop0, loop1, etc.: These are loop devices, commonly used for mounting snap packages.
- sr0: This is a read-only optical disk (CD/DVD-ROM).

vda Column:

- vda: This is the primary disk, typically associated with the root file system (/). It has partitions (vda1, vda2, vda3) and includes an LVM (Logical Volume Manager) setup.

vdb Column:

- vdb: This is the additional storage device (/dev/vdb) with a size of 100GB. This is the device which is intended to be used for configuring an OSD (Object Storage Daemon).

Notes:
The relevant storage device for configuring the OSD is /dev/vdb, which has a size of 100GB. Before configuring OSDs, ensure that the target storage device (/dev/vdb) is not in use and does not contain any important data, as configuring an OSD involves formatting the device. OSDs are responsible for storing actual data in the Ceph cluster. Each OSD corresponds to a storage device in the cluster. In a production environment, it's essential to have a separate and dedicated storage device for OSDs to ensure optimal performance and isolation. The LVM setup on vda is unrelated to the OSD configuration on vdb and is used for the root file system. The readiness of /dev/vdb for OSD configuration is indicated by its presence in the output and its unmounted status. The provided examples and values, such as IP addresses, names, and IDs, are for reference purposes. In your Ceph configuration, these details may vary, and it is essential to use the actual values and names set up in your specific environment. Always consult official documentation for accurate information.

 # Partitioning the Storage Device on each nodes (Ceph01 and Ceph02 in this case):
 # Create a GPT Partition Table:

```
root@ceph01:~# parted --script /dev/vdb 'mklabel gpt'
```

Explanation:

- parted --script /dev/vdb 'mklabel gpt': This command uses parted to create a GPT (GUID Partition Table) on the storage device /dev/vdb.

#Create a Primary Partition using the Entire Disk on each nodes (Ceph01 and Ceph02 in this case):

```
root@ceph01:~# parted --script /dev/vdb "mkpart primary 0% 100%"
```

Explanation:

- parted --script /dev/vdb "mkpart primary 0% 100%": This command uses parted to create a primary partition that spans the entire disk (0% to 100%).

#Create an OSD using ceph-volume on each nodes (Ceph01 and Ceph02 in this case):

```
root@ceph01:~# ceph-volume lvm create --data /dev/vdb1
Running command: /usr/bin/ceph-authtool --gen-print-key
Running command: /usr/bin/ceph --cluster ceph --name client.bootstrap-osd --keyring /var/lib/ceph/bootstrap-osd/ceph.keyring -i - osd new 366a0469-4a0c-40da-8a91-0591127aa93b
Running command: vgcreate --force --yes ceph-00e14c59-b49c-4849-bc8e-a83bb3e77bb0 /dev/vdb1
...
...
...
...
Running command: /usr/bin/systemctl start ceph-osd@0
--> ceph-volume lvm activate successful for osd ID: 0
--> ceph-volume lvm create successful for: /dev/vdb1
```

Explanation:
ceph-volume lvm create --data /dev/vdb1: This command uses ceph-volume to create an OSD on the specified partition (/dev/vdb1). The command involves several steps:

- Generates a key for the OSD.

- Creates a new OSD with a unique identifier (366a0469-4a0c-40da-8a91-0591127aa93b in this example).
- Creates a new LVM (Logical Volume Manager) volume group (ceph-00e14c59-b49c-4849-bc8e-a83bb3e77bb0 in this example) for storing OSD data.
- Activates the OSD and starts the corresponding service (ceph-osd@0 in this example).

Notes:
The OSD ID is 0 in this example. The unique OSD ID (366a0469-4a0c-40da-8a91-0591127aa93b) is generated during the OSD creation process. The LVM volume group (ceph-00e14c59-b49c-4849-bc8e-a83bb3e77bb0) is created to manage the storage for the OSD.
Expected Output:
The output indicates that the ceph-volume LVM activation and creation were successful. The provided examples and values, such as IP addresses, names, and IDs, are for reference purposes. In your Ceph configuration, these details may vary, and it is essential to use the actual values and names set up in your specific environment. Always consult official documentation for accurate information.

 # Confirm cluster status, that's OK if [HEALTH_OK] on Admin Node

```
root@ceph01:~# ceph -s
  cluster:
    id:     00889097-f1a1-4f7a-9bd4-4440848c8eaa
    health: HEALTH_WARN
            mon is allowing insecure global_id reclaim
            OSD count 1 < osd_pool_default_size 3

  services:
    mon: 1 daemons, quorum ceph01 (age 2m)
    mgr: ceph01(active, since 110s)
    osd: 1 osds: 1 up (since 43s), 1 in (since 43s)

  data:
    pools:   1 pools, 1 pgs
    objects: 0 objects, 0 B
    usage:   1.0 GiB used, 99 GiB / 100 GiB avail
    pgs:     100.000% pgs not active
             1 undersized+peered
```

The provided output (ceph -s) indicates the status of the Ceph cluster. Here's an explanation with notes:

Explanation:
Interpretation and Notes:

- Cluster Information:
  - id: Unique identifier for the Ceph cluster.
  - health: Current health status of the cluster. In this case, it's HEALTH_WARN, indicating a warning.

Warnings:
mon is allowing insecure global_id reclaim: This warning indicates that the Monitor Daemon (mon) is allowing insecure global ID reclaim. It's advisable to investigate and address this warning for security reasons. OSD count 1 < osd_pool_default_size 3: This warning indicates that the current number of OSDs (1) is less than the recommended default pool size (3).

- Services:

    - mon: 1 Monitor Daemon is active, and the quorum is established with ceph01 as the quorum member.
    - mgr: The Manager Daemon is active (ceph01).
    - osd: There is 1 OSD in the cluster, and it is both up and in.
- Data Information:
    - pools: There is 1 pool in the cluster.
    - objects: Currently, there are 0 objects in the cluster.
    - usage: 1.0 GiB is used, with 99 GiB available out of 100 GiB.
- PG (Placement Group) Information:
    - pgs: 100% of PGs are not active, and there is 1 undersized+peered PG.

 # Confirm OSD tree

```
root@ceph01:~# ceph osd tree
ID  CLASS  WEIGHT   TYPE NAME        STATUS  REWEIGHT  PRI-AFF
-1         0.09769  root default
-3         0.09769      host ceph01
 0    hdd  0.09769          osd.0        up   1.00000  1.00000
```

Display OSD Tree:
Explanation:

- ceph osd tree: This command displays the hierarchy and status of OSDs in the Ceph cluster.

Notes:
The output will show the OSD IDs, their weight, and their placement within the CRUSH (Controlled Replication Under Scalable Hashing) hierarchy.
The provided examples and values, such as IP addresses, names, and IDs, are for reference purposes. In your Ceph configuration, these details may vary, and it is essential to use the actual values and names set up in your specific environment. Always consult offcialdocument.

 # Confirm ceph osd

```
root@ceph01:~# ceph df
--- RAW STORAGE ---
CLASS  SIZE     AVAIL   USED     RAW USED  %RAW USED
hdd    100 GiB  99 GiB  1.6 MiB   1.0 GiB       1.00
TOTAL  100 GiB  99 GiB  1.6 MiB   1.0 GiB       1.00

--- POOLS ---
POOL                    ID   PGS   STORED   OBJECTS   USED   %USED
MAX AVAIL
device_health_metrics    1     1      0 B         0    0 B       0
31 GiB
```

Display Cluster Disk Usage:
Explanation:

- ceph df: This command provides a summary of the disk usage in the Ceph cluster.

Notes:
The output includes information such as the total used, total capacity, and overall utilization of the cluster's storage resources. It also shows the space used by RADOS (Reliable Autonomic Distributed Object Store) pools. The provided examples and values, such as IP addresses, names, and IDs, are for reference purposes. In your Ceph configuration, these

details may vary, and it is essential to use the actual values and names set up in your specific environment. Always consult official documentation for accurate information.

Add or Remove Monitor Nodes in Ceph Cluster

Introduction

This document provides step-by-step instructions for deploying and managing a Ceph cluster for your company's storage needs.

Per-requisites

Before proceeding with the deployment, ensure you have the following:

- Access to servers with root privileges.
- Connectivity between servers over SSH.
- Servers running a supported Linux distribution (e.g., CentOS, Ubuntu).
- Basic knowledge of Ceph architecture and concepts.

Adding Monitor node in Ceph Cluster

1. Transfer Public Key

```
root@ceph01:~# ssh-copy-id node04
```

Explanation: This command copies the SSH public key of the current server (node01) to the target server (node04) to enable passwordless SSH authentication.

2. Install Required Packages on new node

```
root@ceph04:~# apt update; apt -y install ceph
```

Explanation: This command updates the package repository on node04 and installs the Ceph package.

3. Configure Monitor Map on node 1

```
root@ceph01:~# FSID=$(grep "^fsid" /etc/ceph/ceph.conf | awk {'print $NF'})
root@ceph01:~# NODENAME="ceph05"
root@ceph01:~# NODEIP="172.16.114.98"
root@ceph01:~# monmaptool --add $NODENAME $NODEIP --fsid $FSID /etc/ceph/monmap
```

Explanation: This set of commands updates the monitor map (monmap) with the information of the new monitor node (node04).

4. Configure Monitor Daemon on new node

```
root@ceph01:~#              scp             /etc/ceph/ceph.conf
ceph05:/etc/ceph/ceph.conf
root@ceph01:~# scp /etc/ceph/ceph.mon.keyring ceph04:/etc/ceph
root@ceph01:~# scp /etc/ceph/monmap ceph04:/etc/ceph
root@ceph04:~#  ceph-mon  --cluster  ceph  --mkfs  -i  ceph04  --
monmap /etc/ceph/monmap --keyring /etc/ceph/ceph.mon.keyring
root@ceph04:~# chown -R ceph. /etc/ceph /var/lib/ceph/mon
root@ceph04:~#       ceph       auth       get       mon.       -o
/etc/ceph/ceph.mon.keyring
root@ceph04:~# systemctl enable --now ceph-mon@ceph04
root@ceph04:~# ceph mon enable-msgr2
root@ceph04:~# ceph -s
```

Explanation:

- The first set of commands copies necessary configuration files to node04.
- The ceph-mon command initializes the monitor daemon on node04.
- Permissions are set for the Ceph configuration directory and monitor data directory.
- The monitor key is retrieved and stored on node04.
- The monitor service is enabled and started on node04.
- The ceph -s command is used to verify the cluster status.

Removing Monitor node from Ceph Cluster

5. Removing a Monitor Daemon

```
root@node01:~# ceph mon remove ceph05
```

Explanation: This command removes the monitor daemon running on node04 from the cluster.

6. Disable Monitor Daemon

```
root@ceph04:~# systemctl disable --now ceph-mon@ceph05.service
root@ceph04:~# ceph -s
```

Explanation: These commands disable and stop the monitor daemon service on node04, and then verify the cluster status.

Notes

Ensure proper backups of configuration files before making any changes. Monitor daemon removal should be performed cautiously as it may impact cluster availability. Regularly monitor cluster health and performance for optimal operation.

Add or Remove OSDs Node in a Ceph Cluster

Introduction

This document provides step-by-step instructions for adding a new node to the OSDs (Object Storage Daemons) in a Ceph cluster and removing an existing OSD node. Each command is explained along with important notes.

Per-requisites

Before proceeding with the deployment, ensure you have the following:

- Access to servers with root privileges.
- Connectivity between servers over SSH.
- Servers running a supported Linux distribution (e.g., CentOS, Ubuntu).
- Basic knowledge of Ceph architecture and concepts.

Adding a Node to OSDs

1. Transfer Public Key

```
root@ceph01:~# ssh-copy-id ceph04
```

Explanation: This command transfers the SSH public key of the admin ceph (ceph01) to the target ceph (ceph04) to enable passwordless SSH access.

2. Install Required Packages on new node

```
root@ceph01:~# apt update; apt -y install ceph
```

Explanation: This command updates the package repository and installs the Ceph packages on the target ceph (ceph04).

3. Transfer Required Files from admin node to new node

```
root@ceph01:~#              scp             /etc/ceph/ceph.conf
ceph04:/etc/ceph/ceph.conf
root@ceph01:~#     scp     /etc/ceph/ceph.client.admin.keyring
ceph04:/etc/ceph
root@ceph01:~#   scp   /var/lib/ceph/bootstrap-osd/ceph.keyring
ceph04:/var/lib/ceph/bootstrap-osd
```

Explanation: These commands transfer the required Ceph configuration and keyring files from the admin ceph (ceph01) to the target ceph (ceph04).

4. Configure OSD

```
root@ceph04:~#       chown       ceph.       /etc/ceph/ceph.*
/var/lib/ceph/bootstrap-osd/*
root@ceph04:~# parted --script /dev/sdb 'mklabel gpt'
root@ceph04:~#  parted  --script  /dev/sdb  'mkpart  primary  0%
100%'
root@ceph04:~# ceph-volume lvm create --data /dev/sdb1"
```

Explanation: This command configures the OSD on the target ceph (ceph04). It changes ownership of Ceph configuration files, creates a partition on the specified block device (/dev/sdb), and initializes the OSD using ceph-volume.

Removing an OSD ceph

5. Mark OSD Out

```
root@ceph01:~# ceph osd out 4
```

Explanation: This command marks the OSD with ID 4 (associated with ceph04) out of the cluster, preparing it for removal.

6. Monitor Cluster Status after running [ceph osd out *], rebalancing is executed automatically. After status turns to [HEALTH_OK], disable OSD service on the target node**

```
root@ceph01:~# ceph -w
```

Explanation: This command monitors the cluster status in real-time. After marking the OSD out, the cluster will undergo automatic rebalancing.

7. Disable OSD Service

```
root@ceph04:~# systemctl disable --now ceph-osd@4.service
```

Explanation: This command disables the OSD service on the target ceph (ceph04) associated with OSD ID 4.

8. Purge OSD ceph

```
root@ceph04:~# ceph osd purge 4 --yes-i-really-mean-it
```

Explanation: This command purges the OSD ceph with ID 4 from the cluster. Use with caution as it irreversibly removes the OSD and its data.

Conclusion

Following these steps, you can successfully add a new node to the OSDs in your Ceph cluster and remove an existing OSD node. Ensure to monitor the cluster status and data redundancy during the process to maintain cluster health.