**TÀI LIỆU CÀI ĐẶT, VẬN HÀNH CEPH STORAGE CLUSTER**

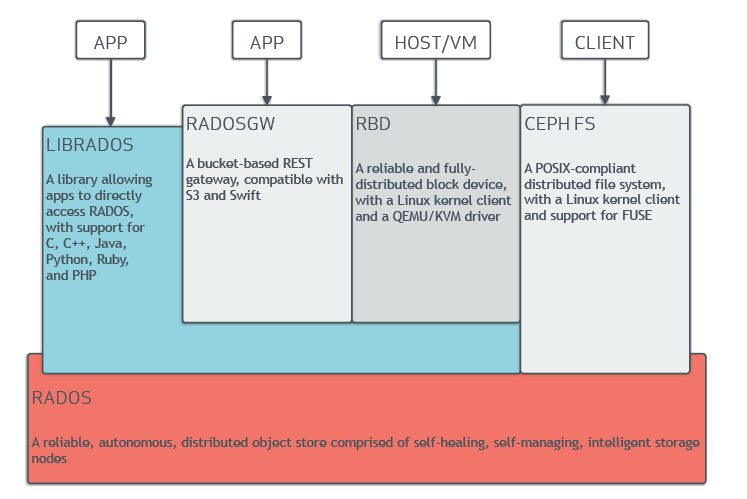
**MỤC LỤC :**

1. **TỔNG QUAN VỀ CEPH :**
2. **KIẾN TRÚC CEPH :**
3. **CÀI ĐẶT CEPH :**
4. **CẤU HÌNH CEPH:**

# TỔNG QUAN VỀ CEPH :

CEPH là giải pháp mã nguồn mở để xây dựng hạ tầng lưu trữ phân tán , ổn định , hiệu năng cao . Nền tảng của CEPH là object , từ đó hình thành nên các dạng block , file . Định dạng dữ liệu block , file đều được lưu dưới dạng object trong placement group của Ceph cluster .

1. **KIẾN TRÚC CEPH :**



Ceph cung cấp 2 thành phần chính là RADOS ( A Scalable, Reliable Storage Service for Petabyte-scale Storage Clusters) và LIBRADOS

* 1. **RADOS :**

RADOS là trái tim của hệ thống lưu trữ CEPH. RADOS cung cấp tất cả các tính năng của Ceph, gồm lưu trữ  phân tán object,tính sẵn sàng cao, tin cậy, tự sửa lỗi, tự quản lý,... lớp RADOS giữ vai trò đặc biệt quan trọng trong kiến trúc Ceph. Các phương thức truy xuất Ceph, như RBD, CephFS, RADOSGW và librados, đều hoạt động trên lớp RADOS. Khi Ceph cluster nhận một yêu cầu ghi từ người dùng, thuật toán CRUSH (Controlled, scalable, decentralized placement of replicated data) tính toán vị trí và thiết bị mà dữ liệu sẽ được ghi vào. Các thông tin này được đưa lên lớp RADOS để xử lý. Dựa vào quy tắc của CRUSH, RADOS phân tán dữ liệu lên tất cả các node dưới dạng object, các object này được lưu tại các OSD.

RADOS, khi cấu hình với số nhân bản nhiều hơn hai, sẽ chịu trách nhiệm về độ tin cậy của dữ liệu. Nó sao chép object, tạo các bản sao và lưu trữ tại các node khác nhau, do đó các bản ghi giống nhau không nằm trên cùng 1 node. RADOS đảm bảo có nhiều hơn một bản copy của object trong RADOS cluster. RADOS cũng đảm bảo object luôn nhất quán. Trong trường hợp object không nhất quán, tiến trình khôi phục sẽ chạy. Tiến trình này chạy tự động và trong suốt với người dùng, do đó mang lại khả năng tự sửa lỗi và tự quẩn lý cho Ceph. RADOS có 2 phần: phần thấp không tương tác trực tiếp với giao diện người dùng, và phần cao hơn có tất cả giao diện người dùng.

RADOS có 2 thành phần chính là OSD và Monitor :

* **OSD (Object storage devices ):**

Một Ceph cluster bao gồm nhiều OSD. Ceph lưu trữ dữ liệu dưới dạng object trên các ổ đĩa vật lý.

* Object :

Với các tác vụ đọc hoặc ghi, client gửi yêu cầu tới node monitor để lấy cluster map sau đó tương tác trực tiếp với OSD ko cần sự can thiệp của monitor.

Object được phân tán lưu trên nhiều OSD, mỗi OSD là primary OSD cho một số object và là secondary OSD cho object khác để tăng tính sẵn sàng và khả năng chống chịu lỗi. Khi primary OSD bị lỗi thì secondary OSD được đẩy lên làm primary OSD. Quá trình này trong suốt với ngưới dùng.

* **Ceph OSD File System**

Ceph OSD gồm ổ cứng vật lý, Linux filesystem trên nó và Ceph OSD Service. Linux filesystem của Ceph cần hỗ trợ extended attribute (XATTRs). Các thuộc tính của filesystem này cung cấp các thông tin về trạng thái object, metadata, snapshot và ACL cho Ceph OSD daemon, hỗ trợ việc quản lý dữ liêu. LinuxFileSystem có thể là Btrfs, XFS hay Ext4.

* **Ceph OSD Journal**

Ceph dùng các journaling filesystem là XFS cho OSD. Trước khi commit dữ liệu vào backing store, Ceph ghi dữ liệu vào một vùng lưu trữ tên là journal trước, vùng này hoạt động như là một phân vùng đệm (buffer), Journal nằm cùng hoặc khác đĩa với OSD, trên một SSD riêng hoặc một phân vùng, thậm chí là một file riêng trong filesystem. Với cơ chế này, Ceph ghi mọi thứ vào journal, rồi mới ghi vào backing storage.

Một dữ liệu ghi vào journal sẽ được lưu tại đây trong lúc syncs xuống backing store, mặc định là 5 giây chạy 1 lần. 10 GB là dung lượng phổ biến của journal, tuy nhiên journal càng lớn càng tốt. Ceph dùng journal để tăng tốc và đảm bảo tính nhất quán. Journal cho phép Ceph OSD thực hiện các tác vụ ghi nhỏ nhanh chóng; một tác vụ ghi ngẫu nhiên sẽ được ghi xuống journal theo kiểu tuần tự, sau đó được flush xuống filesystem. Điều này cho phép filesystem có thời gian để gộp các tác vụ ghi vào ổ đĩa. Hiệu năng sẽ tăng lên rõ rệt khi journal được tạo trên SSD.

Khuyến nghị, không nên vượt quá tỉ lệ 5 OSD/1 Journal disk khi dùng SSD làm journal. Khi SSD Journal bị lỗi, toàn bộ các OSD có journal trên SSD đó sẽ bị lỗi. Với Btrfs, việc này sẽ không xảy ra, bởi Btrfs hỗ trợ copy-on-write, chỉ ghi xuống các dữ liệu thay đổi, mà không tác động vào dữ liệu cũ, khi journal bị lỗi, dữ liệu trên OSD vẫn tồn tại.

* **Ceph Disk**

Mặc định Ceph đã có khả năng nhân bản để bảo vệ dữ liệu, do đó không cần làm RAID với các dữ liệu đã được nhân bản đó.

Phương pháp nhân bản dữ liệu của Ceph khong yêu câu một ổ cứng trống cùng dung lượng ổ hỏng. Nó dùng đường truyền mạng để khôi phục dữ liệu trên ổ cứng lỗi từ nhiều node khác. Trong quá trình khôi phục dữ liệu, dựa vào tỉ lệ nhân bản và số PGs(Placement Group), hầu như toàn bộ các node sẽ tham gia vào quá trình khôi phục, giúp quá trình này diễn ra nhanh hơn.

* **Ceph Monitor**

Ceph monitor chịu trách nhiệm giám sát tình trạng của toàn hệ thống. Nó hoạt động như các daemon duy trì sự kết nối trong cluster bằng cách chứa các thông tin cơ bản về cluster, tình trạng các node lưu trữ và thông tin cấu hình cluster. Ceph monitor thực hiện điều này bằng cách duy trì các cluster map. Các cluster map này bao gồm monitor, OSD, PG, CRUSH và MDS map.

    Monitor map: map này lưu giữ thông tin về các node monitor, gồm CEPH Cluster ID, monitor hostname, địa chỉ IP và số port. Nó cũng giữ epoch (phiên bản map tại một thời điểm) hiện tại để tạo map và thông tin về lần thay đổi map cuối cùng.

    OSD map: map này lưu giữ các trường như cluster ID, epoch cho việc tạo map OSD và lần thay đổi cuối., và thông tin liên quan đến pool như tên, ID, loại, mức nhân bản và PG. Nó cũng lưu các thông tin OSD như tình trạng, trọng số, thông tin host OSD.

    PG map: map này lưu giữ các phiên bản của PG (thành phần quản lý các object trong ceph), timestamp, bản OSD map cuối cùng, tỉ lệ đầy và gần đầy dung lượng. Nó cũng lưu các ID của PG, object count, tình trạng hoạt động và srub (hoạt động kiểm tra tính nhất quán của dữ liệu lưu trữ).

    CRUSH map: map này lưu các thông tin của các thiết bị lưu trữ trong Cluster, các rule cho tưng vùng lưu trữ.

    MDS map: lưu thông tin về thời gian tạo và chỉnh sửa, dữ liệu và metadata pool ID, cluster MDS count, tình trạng hoạt động của MDS, epoch của MDS map hiện tại.

Số monitor trong hệ thống nên là số lẻ, ít nhất là 1 node, và khuyến nghị là 3 node. Một monitor node sẽ là leader, và các node còn lại sẽ đưa đưa lên làm leader nếu node ban đầu bị lỗi.

Monitor daemon có thể chạy cùng trên OSD node. Tuy nhiên, cần trang bị nhiều CPU, RAM và ổ cứng hơn để lưu monitor logs. Đối với các hệ thống lớn, nên sử dụng node monitor chuyên dụng

* 1. **LIBRADOS :**

Librados là thư viện C cho phép ứng dụng làm việc trực tiếp với RADOS, bypass qua các lớp khác để tương tác với Ceph Cluster.Librados là thư viện cho RADOS, cung cấp các hàm API, giúp ứng dụng tương tác trực tiếp và truy xuất song song vào cluster.Ứng dụng có thể mở rộng các giao thức của nó để truy cập vào RADOS bằng cách sử dụng librados. Các thư viện tương tự cũng sẵn sàng cho C++, Java, Python, Ruby, PHP. librados là nền tảng cho các service giao diện khác chạy bên trên, gồm Ceph block device, Ceph filesystem, Ceph RADOS Gateway. librados cung cấp rất nhiều API, các phương thức lưu trữ key/value trong object. API hỗ trợ atomic-single-object bằng cách update dữ liệu, key và các thuộc tính.

Ceph cung cấp 3 loại storage trên thư viện librados , đó là block device , filesystem và object gateway storage

* **Ceph Block Storage Device**

Ceph Block Device có tên là RADOS block device (RBD); cung cấp block storage cho hypervisor và máy ảo. Ceph RBD driver được tích hợp với Linux kernel (từ bản 2.6.39) và hỗ trợ QEMU/KVM.

Khi Ceph block device được map vào máy chủ Linux, nó có thể được sử dụng như một phân vùng RAW hoặc cố thể định dạng theo các loại filesystem phổ biến.

Ceph đã được tích hợp chặt chẽ với các nền tảng Cloud như OpenStack. Ceph cung cấp backend là block device để lưu trữ volume máy ảo và OS image choCinder và Glance. Các volume và image này là thin provisioned, có nghĩa chỉ lưu trữ các dữ liệu object bị thay đổi, giúp tiết kiệm tài nguyên lưu trữ.

Tính năng copy-on write và cloning của Ceph giúp tạo hàng tram máy ảo trong thời gian ngắn . RBD Cũng hỗ trợ snapshot , lưu trữ trạng thái máy ảo . Ceph cũng là backend cho máy ảo , giúp di chuyển máy ảo giữa các node . Các hypervisor như QEMU , KVM , XEN có thể boot máy ảo từ volume nằm trên ceph .

RBD sử dụng thư viện librbd để tận dụng các tiện ích của RADOS và cung cấp tính tin cậy, phân tán và khả năng lưu trữ dựa trên object.RBD hỗ trợ update vào các object. Client có thể ghi, nối thêm, cắt xén vào các object đang có. Do đó RBD là giải pháp lưu trữ tối ưu cho volume máy ảo.

* **Ceph Object Gateway**

Ceph Object Gateway, hay RADOS Gateway, là một proxy chuyển các request HTTP thành các RADOS request và ngược lại, cung cấp RESTful object storage, tương thích với S3 và Swift. Ceph Object Storage sử dụng Ceph Object Gateway Daemon (radosgw) để tương tác với librgw và Ceph cluster, librados. Nó sử dụng một module FastCGI là libfcgi, và có thể sử dụng với bất cứ Server web tương thích với FASTCGI nào. Ceph Object Store hỗ trợ 3 giao diện sau:

    S3: Cung cấp Amazon S3 RESTful API.

    Swift: Cung cấp OpenStack Swift API. Ceph Object Gateway có thể thay thê Swift.

    Admin: Hỗ trợ quản trị Ceph Cluster thông qua HTTP RESTful API.

* **CephFS**

Ceph Filesystem (CephFS) là một POSIX- compliant filesystem sử dụng Ceph Storage Cluster để lưu data. CephFS kế thừa các tính năng từ RADOS.

Để sử dụng CephFS cần một Ceph MDS hay Metadata Server là daemon cho Ceph filesystem (CephFS). MDS là thành phần duy nhất trong ceph chưa production, hiện chỉ 1 ceph MDS daemon hoạt động tại 1 thời điểm. MDS không lưu dữ liệu local. Nếu 1 MDS daemon lỗi, ta có thể tạo lại trên bất cứ hệ thống nào mà cluster có thể truy cập. Các metadata server daemon được cấu hình là active-passive. Primary MDS là acive, còn các node khác chạy standby.

Thư viện libcephfs hỗ trợ Linux kernel driver, người dùng có thể dùng phương thức mounting filesystem qua lệnh mount. Nó hỗ trợ CIFS và SMB. CephFS hỗ trợ filesystem in userspace(FUSE) dùng module cephfuse. Nó cũng hỗ trợ ứng dụng tương tác trực tiếpvới RADOS cluster dùng thư viện libcephfs.

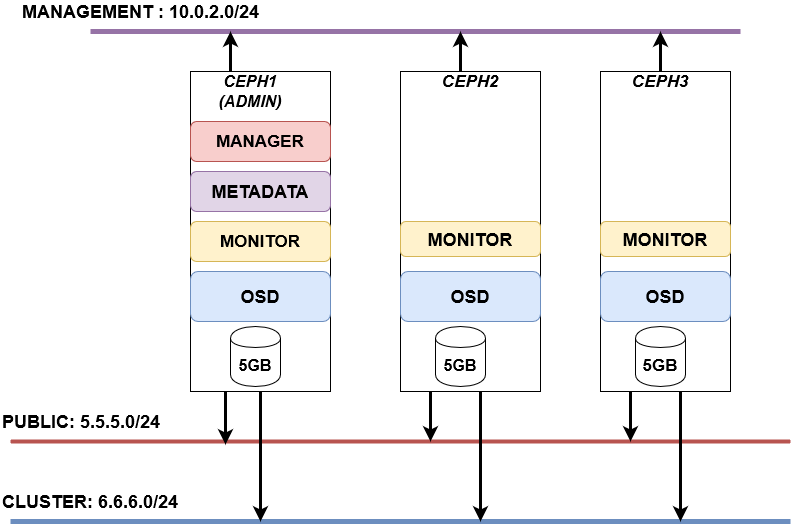
1. **CÀI ĐẶT CEPH :**
   1. **YÊU CẦU VỀ TÀI NGUYÊN :**

## Minimum Hardware Recommendations

Ceph can run on inexpensive commodity hardware. Small production clusters and development clusters can run successfully with modest hardware.

| **Process** | **Criteria** | **Minimum Recommended** |
| --- | --- | --- |
| ceph-osd | Processor | * 1x 64-bit AMD-64 * 1x 32-bit ARM dual-core or better * 1x i386 dual-core |
| RAM | ~1GB for 1TB of storage per daemon |
| Volume Storage | 1x storage drive per daemon |
| Journal | 1x SSD partition per daemon (optional) |
| Network | 2x 1GB Ethernet NICs |
| ceph-mon | Processor | * 1x 64-bit AMD-64/i386 * 1x 32-bit ARM dual-core or better * 1x i386 dual-core |
| RAM | 1 GB per daemon |
| Disk Space | 10 GB per daemon |
| Network | 2x 1GB Ethernet NICs |
| ceph-mds | Processor | * 1x 64-bit AMD-64 quad-core * 1x 32-bit ARM quad-core * 1x i386 quad-core |
| RAM | 1 GB minimum per daemon |
| Disk Space | 1 MB per daemon |
| Network | 2x 1GB Ethernet NICs |

* 1. **CHUẨN BỊ :**
* Ở phần cài đặt này ta sử dụng 4 máy ảo , 3 máy chạy cluster storage, 1 máy làm client .
* 3 máy trong cluster có ổ cứng ít nhất trên 2GB trở lên , không tính ổ cài hệ điều hành .
  1. **CÀI ĐẶT :**
* Mô hình :

****

* + 1. **CÀI ĐẶT CHO CLUSTER :**
* **Thiết lập hostname, IP cho node CEPH1 :**

- Login với tài khoản root và thực hiện các lệnh dưới.

```

yum update -y

```

- Cài đặt các gói phần mềm bổ trợ :

```

yum install epel-release -y

yum update -y

yum install wget byobu curl git byobu python-setuptools python-virtualenv -y

```

- Thiết lập hostname cho CEPH1 :

```

hostnamectl set-hostname ceph1

```

- Thiết lập IP cho các card mạng, trong hướng dẫn này, 5.5.5.0/24 sẽ ra internet để tải các gói cài đặt.

```

vi ifcfg-enp0s7

```TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=static

IPADDR=10.0.2.7

DEFROUTE=yes

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=enp0s7

DEVICE=enp0s3

ONBOOT=yes```

vi ifcfg-enp0s8

```TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=static

IPADDR=5.5.5.9

DEFROUTE=yes

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=enp0s7

ONBOOT=yes```

vi ifcfg-enp0s9

  ```TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=static

IPADDR=6.6.6.13

DEFROUTE=yes

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=enp0s9

ONBOOT=yes```

```

- Cấu hình chế độ firewall để tiện trong môi trường lab. Trong môi trường production cần bật firewall hoặc iptables hoặc có biện pháp xử lý khác tương ứng để đảm bảo các vấn đề về an toàn.

```

sudo systemctl disable firewalld

sudo systemctl stop firewalld

sudo systemctl disable NetworkManager

sudo systemctl stop NetworkManager

sudo systemctl enable network

sudo systemctl start network

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config

echo "net.ipv6.conf.all.disable\_ipv6 = 1" >> /etc/sysctl.conf

```

- Khai báo file /etc/hosts. Việc này rất quan trọng vì CEPH sẽ sử dụng hostname trong các bước tới để cấu hình và kết nối khi thực hiện.

```

vi /etc/hosts

127.0.0.1 ceph1 localhost

5.5.5.9 ceph1

5.5.5.8 ceph3

5.5.5.7 ceph2

5.5.5.4 client1

10.0.2.7 ceph1

10.0.2.5 ceph3

10.0.2.6 ceph2

10.0.2.15 client1

```

- Cài đặt NTP, trong hướng dẫn này sử dụng chronyd thay cho ntpd. Việc đồng bộ thời gian cũng là quan trọng khi triển khai CEPH. Hãy đảm bảo timezone và thời gian được đồng bộ để đúng với hệ thống của bạn.

```

yum install -y chronyd

systemctl enable chronyd.service

systemctl start chronyd.service

systemctl restart chronyd.service

chronyc sources

```

- Khởi động lại node CEPH1 và chuyển sang CEPH2 thực hiện tiếp.

```

init 6

```

* **THIẾT LẬP IP , HOSTNAME CHO NODE CEPH2 :**

- Login với tài khoản root và thực hiện các lệnh dưới.

- Cập nhật các gói phần mềm.

```

yum update -y

```

- Cài đặt các gói phần mềm bổ trợ

```

yum install epel-release -y

yum update -y

yum install wget byobu curl git byobu python-setuptools python-virtualenv -y

```

- Thiết lập hostname cho CEPH2

```

hostnamectl set-hostname ceph2

```

- Thiết lập IP Add, trong hướng dẫn này, VLAN 192.168.98.0/24 sẽ ra internet để tải các gói cài đặt.

```

echo "Setup IP  eth0"

nmcli con modify eth0 ipv4.addresses 192.168.98.86/24

nmcli con modify eth0 ipv4.gateway 192.168.98.1

nmcli con modify eth0 ipv4.dns 8.8.8.8

nmcli con modify eth0 ipv4.method manual

nmcli con mod eth0 connection.autoconnect yes

echo "Setup IP  eth1"

nmcli con modify eth1 ipv4.addresses 192.168.62.86/24

nmcli con modify eth1 ipv4.method manual

nmcli con mod eth1 connection.autoconnect yes

echo "Setup IP  eth2"

nmcli con modify eth2 ipv4.addresses 192.168.63.86/24

nmcli con modify eth2 ipv4.method manual

nmcli con mod eth2 connection.autoconnect yes

```

- Cấu hình chế độ firewall để tiện trong môi trường lab. Trong môi trường production cần bật firewall hoặc iptables hoặc có biện pháp xử lý khác tương ứng để đảm bảo các vấn đề về an toàn.

```

sudo systemctl disable firewalld

sudo systemctl stop firewalld

sudo systemctl disable NetworkManager

sudo systemctl stop NetworkManager

sudo systemctl enable network

sudo systemctl start network

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config

echo "net.ipv6.conf.all.disable\_ipv6 = 1" >> /etc/sysctl.conf

```

- Khai báo file /etc/hosts. Việc này rất quan trọng vì CEPH sẽ sử dụng hostname trong các bước tới để cấu hình và kết nối khi thực hiện.

```

vi /etc/hosts

127.0.0.1 ceph2 localhost

5.5.5.9 ceph1

5.5.5.8 ceph3

5.5.5.7 ceph2

5.5.5.4 client1

10.0.2.7 ceph1

10.0.2.5 ceph3

10.0.2.6 ceph2

10.0.2.15 client1

```

- Cài đặt NTP, trong hướng dẫn này sử dụng chronyd thay cho ntpd. Việc đồng bộ thời gian cũng là quan trọng khi triển khai CEPH. Hãy đảm bảo timezone và thời gian được đồng bộ để đúng với hệ thống của bạn.

```

yum install -y chronyd

systemctl enable chronyd.service

systemctl start chronyd.service

systemctl restart chronyd.service

chronyc sources

```

- Khởi động lại node CEPH2 và chuyển sang CEPH3 thực hiện.

```

init 6

```

* **THIẾT LẬP HOSTNAME,IP CHO NODE CEPH3:**

- Login với tài khoản root và thực hiện các lệnh dưới.

- Cập nhật các gói phần mềm.

```

yum update -y

```

- Cài đặt các gói bổ trợ

```

yum install epel-release -y

yum update -y

yum install wget byobu curl git byobu python-setuptools python-virtualenv -y

```

- Thiết lập hostname cho CEPH3

```

hostnamectl set-hostname ceph3

```

- Thiết lập IP Add, trong hướng dẫn này, VLAN 192.168.98.0/24 sẽ ra internet để tải các gói cài đặt.

```

echo "Setup IP  eth0"

nmcli con modify eth0 ipv4.addresses 192.168.98.87/24

nmcli con modify eth0 ipv4.gateway 192.168.98.1

nmcli con modify eth0 ipv4.dns 8.8.8.8

nmcli con modify eth0 ipv4.method manual

nmcli con mod eth0 connection.autoconnect yes

echo "Setup IP  eth1"

nmcli con modify eth1 ipv4.addresses 192.168.62.87/24

nmcli con modify eth1 ipv4.method manual

nmcli con mod eth1 connection.autoconnect yes

echo "Setup IP  eth2"

nmcli con modify eth2 ipv4.addresses 192.168.63.87/24

nmcli con modify eth2 ipv4.method manual

nmcli con mod eth2 connection.autoconnect yes

```

- Cấu hình chế độ firewall để tiện trong môi trường lab. Trong môi trường production cần bật firewall hoặc iptables hoặc có biện pháp xử lý khác tương ứng để đảm bảo các vấn đề về an toàn.

```

sudo systemctl disable firewalld

sudo systemctl stop firewalld

sudo systemctl disable NetworkManager

sudo systemctl stop NetworkManager

sudo systemctl enable network

sudo systemctl start network

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config

echo "net.ipv6.conf.all.disable\_ipv6 = 1" >> /etc/sysctl.conf

```

- Khai báo file /etc/hosts. Việc này rất quan trọng vì CEPH sẽ sử dụng hostname trong các bước tới để cấu hình và kết nối khi thực hiện.

```

vi /etc/hosts

127.0.0.1 ceph3 localhost

5.5.5.9 ceph1

5.5.5.8 ceph3

5.5.5.7 ceph2

5.5.5.4 client1

10.0.2.7 ceph1

10.0.2.5 ceph3

10.0.2.6 ceph2

10.0.2.15 client1

```

- Cài đặt NTP, trong hướng dẫn này sử dụng chronyd thay cho ntpd. Việc đồng bộ thời gian cũng là quan trọng khi triển khai CEPH. Hãy đảm bảo timezone và thời gian được đồng bộ để đúng với hệ thống của bạn.

```

yum install -y chronyd

systemctl enable chronyd.service

systemctl start chronyd.service

systemctl restart chronyd.service

chronyc sources

```

- Khởi động lại node CEPH2 và chuyển sang CEPH1 thực hiện các cấu hình tiếp theo. Login với các IP đã đặt cho CEPH trước đó.

* **TẠO USER CÀI ĐẶT CEPH VÀ KHAI BÁO REPO:**

**### TẠO USER CÀI ĐẶT TRÊN CẢ 3 NODE CEPH1,CEPH2,CEPH3 :**

- Bước này được thực hiện trên cả 03 node CPEH.

- Tạo user là cephuser với mật khẩu là matkhau2019@

```

useradd cephuser; echo 'matkhau2019@' | passwd cephuser --stdin

```

- Cấp quyền sudo cho user cephuser

```

echo "cephuser ALL = (root) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/cephuser

chmod 0440 /etc/sudoers.d/cephuser

```

- Khai báo repo cho ceph nautilus , thực hiện hiện khai báo repo cho ceph nautilus trên tất cả các node CEPH.

```

cat <<EOF> /etc/yum.repos.d/ceph.repo

[ceph]

name=Ceph packages for $basearch

baseurl=https://download.ceph.com/rpm-nautilus/el7/x86\_64/

enabled=1

priority=2

gpgcheck=1

gpgkey=https://download.ceph.com/keys/release.asc

[ceph-noarch]

name=Ceph noarch packages

baseurl=https://download.ceph.com/rpm-nautilus/el7/noarch

enabled=1

priority=2

gpgcheck=1

gpgkey=https://download.ceph.com/keys/release.asc

[ceph-source]

name=Ceph source packages

baseurl=https://download.ceph.com/rpm-nautilus/el7/SRPMS

enabled=0

priority=2

gpgcheck=1

gpgkey=https://download.ceph.com/keys/release.asc

EOF

```

- Thực hiện update sau khi khai bao repo

```

yum update -y

```

**CÀI ĐẶT CEPH DEPLOY VÀ CẤU HÌNH**

- Cài đặt ceph-deploy , thực hiện việc cài đặt này trên node CEPH1

- Trong một số mô hình, node cài đặt ceph-deploy được gọi là node admin. Trong hướng dẫn này, ta sẽ sử dụng node ceph1 chính là node ceph admin. Thực hiện việc này bằng tài khoản root.

> Lưu ý :  trong hướng dẫn này chỉ cần đứng trên ceph1 thực hiện, một số thao tác trên node CEPH2, CEPH3 sẽ thực hiện từ xa ngay trên CEPH1 .

```

sudo yum install -y epel-release

sudo yum install -y ceph-deploy

```

- Chuyển sang tài khoản cephuser

```

su - cephuser

```

- Tạo ssh key, sau đó copy sang các node còn lại, nhập mật khẩu của user cephuser ở trên khi được hỏi tại màn hình. Lưu ý không dùng sudo với lệnh ssh-keygen

```

ssh-keygen

```

- Nhấn Enter để mặc định các tham số, bước này sẽ sinh ra private key và public key cho user cephuser. Sau đó tiến hành các lệnh dưới để copy public key sang các node.

- Nhập mật khẩu của user cephuser tạo ở các node trước đó trong bước trên.

```

ssh-copy-id cephuser@ceph1

ssh-copy-id cephuser@ceph2

ssh-copy-id cephuser@ceph3

```

- Tạo thư mục chứa các file cấu hình khi cài đặt CEPH

```

cd ~

mkdir my-cluster

cd my-cluster

```

- Khai báo các node ceph trong cluser.

```

ceph-deploy new ceph1 ceph2 ceph3

```

- Lệnh trên sẽ sinh ra các file cấu hình trong thư mục hiện tại, kiểm tra bằng lệnh ls – alh

```

cephuser@ceph1:~/my-cluster$ ls -alh

total 188K

drwxrwxr-x 2 cephuser cephuser   75 Sep  6 23:14 .

drwx------ 5 cephuser cephuser  151 Sep  6 23:14 ..

-rw-rw-r-- 1 cephuser cephuser  418 Sep  6 23:15 ceph.conf

-rw-rw-r-- 1 cephuser cephuser 177K Sep  6 23:19 ceph-deploy-ceph.log

-rw------- 1 cephuser cephuser   73 Sep  6 23:14 ceph.mon.keyring

```

- Khai báo thêm các tùy chọn cho việc triển khai, vận hành CEPH vào file ceph.conf này trước khi cài đặt các gói cần thiết cho ceph trên các node. Lưu ý các tham số về network.

- Ta sẽ dụng mạng 5.5.5.0/24 cho đường truy cập của các client (Hay gọi là ceph public. mạng 6.6.6.0/24 cho đường replicate dữ liệu, các dữ liệu sẽ được sao chép & nhân bản qua vlan này.

```

echo "public network = 192.168.62.0/24" >> ceph.conf

echo "cluster network = 192.168.63.0/24" >> ceph.conf

echo "osd objectstore = bluestore"  >> ceph.conf

echo "mon\_allow\_pool\_delete = true"  >> ceph.conf

echo "osd pool default size = 3"  >> ceph.conf

echo "osd pool default min size = 1"  >> ceph.conf

```

- Bắt đầu cài đặt phiên bản CEPH Nautilus lên các node ceph1, ceph2, ceph3. Lệnh dưới sẽ cài đặt lần lượt lên các node.

```

ceph-deploy install --release nautilus ceph1 ceph2 ceph3

```

- Kết quả của lệnh trên sẽ hiển thị như bên dưới, trong đó có phiên bản của ceph được cài trên các node.

```

.......

[2019-09-06 23:16:46,139][ceph1][INFO  ] Running command: sudo ceph --version

[2019-09-06 23:16:46,262][ceph1][DEBUG ] ceph version 14.2.3 (0f776cf838a1ae3130b2b73dc26be9c95c6ccc39) nautilus (stable)

.......

[2019-09-06 23:18:18,771][ceph2][INFO  ] Running command: sudo ceph --version

[2019-09-06 23:18:18,944][ceph2][DEBUG ] ceph version 14.2.3 (0f776cf838a1ae3130b2b73dc26be9c95c6ccc39) nautilus (stable)

.......

[2019-09-06 23:19:46,856][ceph3][INFO  ] Running command: sudo ceph --version

[2019-09-06 23:19:47,028][ceph3][DEBUG ] ceph version 14.2.3 (0f776cf838a1ae3130b2b73dc26be9c95c6ccc39) nautilus (stable)

```

- Thiết lập thành phần MON cho CEPH. Trong hướng dẫn này khai báo 03 node đều có thành phần MON của CEPH.

```

ceph-deploy mon create-initial

```

- Kết quả sinh ra các file trong thư mục my-cluster :

```

cephuser@ceph1:~/my-cluster$ ls -alh

total 348K

drwxrwxr-x 2 cephuser cephuser  244 Sep  6 23:26 .

drwx------ 5 cephuser cephuser  151 Sep  6 23:14 ..

-rw------- 1 cephuser cephuser  113 Sep  6 23:26 ceph.bootstrap-mds.keyring

-rw------- 1 cephuser cephuser  113 Sep  6 23:26 ceph.bootstrap-mgr.keyring

-rw------- 1 cephuser cephuser  113 Sep  6 23:26 ceph.bootstrap-osd.keyring

-rw------- 1 cephuser cephuser  113 Sep  6 23:26 ceph.bootstrap-rgw.keyring

-rw------- 1 cephuser cephuser  151 Sep  6 23:26 ceph.client.admin.keyring

-rw-rw-r-- 1 cephuser cephuser  418 Sep  6 23:15 ceph.conf

-rw-rw-r-- 1 cephuser cephuser 214K Sep  6 23:26 ceph-deploy-ceph.log

-rw------- 1 cephuser cephuser   73 Sep  6 23:14 ceph.mon.keyring

```

- Thực hiện copy file ceph.client.admin.keyring sang các node trong cụm ceph cluster. File này sẽ được copy vào thư mục /etc/ceph/ trên các node.

```

ceph-deploy admin ceph1 ceph2 ceph3

```

- Kết quả của lệnh trên như sau

```

cephuser@ceph1:~/my-cluster$ ceph-deploy admin ceph1 ceph2 ceph3

[ceph\_deploy.conf][DEBUG ] found configuration file at: /home/cephuser/.cephdeploy.conf

[ceph\_deploy.cli][INFO  ] Invoked (2.0.1): /bin/ceph-deploy admin ceph1 ceph2 ceph3

[ceph\_deploy.cli][INFO  ] ceph-deploy options:

[ceph\_deploy.cli][INFO  ]  username                      : None

[ceph\_deploy.cli][INFO  ]  verbose                       : False

[ceph\_deploy.cli][INFO  ]  overwrite\_conf                : False

[ceph\_deploy.cli][INFO  ]  quiet                         : False

[ceph\_deploy.cli][INFO  ]  cd\_conf                       : <ceph\_deploy.conf.cephdeploy.Conf instance at 0x7f3c031d6248>

[ceph\_deploy.cli][INFO  ]  cluster                       : ceph

[ceph\_deploy.cli][INFO  ]  client                        : ['ceph1', 'ceph2', 'ceph3']

[ceph\_deploy.cli][INFO  ]  func                          : <function admin at 0x7f3c03a77230>

[ceph\_deploy.cli][INFO  ]  ceph\_conf                     : None

[ceph\_deploy.cli][INFO  ]  default\_release               : False

[ceph\_deploy.admin][DEBUG ] Pushing admin keys and conf to ceph1

[ceph1][DEBUG ] connection detected need for sudo

[ceph1][DEBUG ] connected to host: ceph1

[ceph1][DEBUG ] detect platform information from remote host

[ceph1][DEBUG ] detect machine type

[ceph1][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf

[ceph\_deploy.admin][DEBUG ] Pushing admin keys and conf to ceph2

[ceph2][DEBUG ] connection detected need for sudo

[ceph2][DEBUG ] connected to host: ceph2

[ceph2][DEBUG ] detect platform information from remote host

[ceph2][DEBUG ] detect machine type

[ceph2][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf

[ceph\_deploy.admin][DEBUG ] Pushing admin keys and conf to ceph3

[ceph3][DEBUG ] connection detected need for sudo

[ceph3][DEBUG ] connected to host: ceph3

[ceph3][DEBUG ] detect platform information from remote host

[ceph3][DEBUG ] detect machine type

[ceph3][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf

```

- Đứng trên node ceph1 phân quyền cho file /etc/ceph/ceph.client.admin.keyring cho cả 03 node.

```

ssh cephuser@ceph1 'sudo chmod +r /etc/ceph/ceph.client.admin.keyring'

ssh cephuser@ceph2 'sudo chmod +r /etc/ceph/ceph.client.admin.keyring'

ssh cephuser@ceph3 'sudo chmod +r /etc/ceph/ceph.client.admin.keyring'

```

* **KHAI BÁO CÁC OSD CHO NODE CEPH .**

- Đứng trên node ceph1 và thực hiện khai báo các OSD disk. Bước này sẽ thực hiện format các disk trên cả 3 node và join chúng vào làm các OSD (Thành phần chứa dữ liệu của CEPH).

```

ceph-deploy osd create --data /dev/sdb ceph1

ceph-deploy osd create --data /dev/sdb ceph2

ceph-deploy osd create --data /dev/sdb ceph3

```

> Lưu ý :

> Cần ít nhất 2GB trở lên để tạo osd

- Tới đây các bước cơ bản cấu hình ceph cluser đã hoàn tất. Kiểm tra trạng thái của cụm cluster ceph bằng lệnh ceph -s . Kết quả lệnh ceph -s sẽ như sau

```

cephuser@ceph1:~/my-cluster$ ceph -s

  cluster:

    id:     9691b2b5-a858-45b7-a239-4de3e1ff69c6

    health: HEALTH\_WARN

            no active mgr

  services:

    mon: 3 daemons, quorum ceph1,ceph2,ceph3 (age 4m)

    mgr: no daemons active

    osd: 8 osds: 8 up (since 6s), 8 in (since 6s)

  data:

    pools:   0 pools, 0 pgs

    objects: 0 objects, 0 B

    usage:   0 B used, 0 B / 0 B avail

    pgs:

```

= Ta thấy trạng thái sẽ là HEALTH\_WARN, lý do là vì ceph-mgr chưa được enable. Tiếp theo ta sẽ xử lý để kích hoạt ceph-mgr

* **Cấu hình manager và dashboad cho ceph cluster**

- Ceph-dashboard là một thành phần thuộc ceph-mgr. Trong bản Nautilus thì thành phần dashboard được cả tiến khá lớn. Cung cấp nhiều quyền hạn thao tác với CEPH hơn các bản trước đó (thành phần này được đóng góp chính bởi team SUSE).

- Thực hiện trên node ceph1 việc cài đặt này.

- Trong bản ceph nautilus 14.2.3 (tính đến ngày 06.09.2019), khi cài ceph dashboard theo các cách cũ gặp một vài vấn đề, cách xử lý như sau.

- Cài thêm các gói bổ trợ trước khi cài

```

sudo yum install -y python-jwt python-routes

```

- Tải ceph-dashboad ceph-mgr-dashboard-14.2.4-0.el7.noarch.rpm và ceph-grafana-dashboards-14.2.4-0.el7.noarch.rpm. Lưu ý nên đúng phiên bản với ceph ở trên.

```

sudo rpm -Uvh http://download.ceph.com/rpm-nautilus/el7/noarch/ceph-grafana-dashboards-14.2.4-0.el7.noarch.rpm

sudo rpm -Uvh http://download.ceph.com/rpm-nautilus/el7/noarch/ceph-mgr-dashboard-14.2.4-0.el7.noarch.rpm

```

- Thực hiện kích hoạt ceph-mgr và ceph-dashboard

```

ceph-deploy mgr create ceph1 ceph2 ceph3

yum install ceph-mgr-dashboard -y

ceph mgr module enable dashboard

ceph dashboard create-self-signed-cert

ceph mgr services

```

> Lưu ý :

> - Cài đặt yum install ceph-mgr-dashboard -y trên cả ceph01 và ceph02 ceph3

- Tạo cert cho ceph-dashboad

```

sudo ceph dashboard create-self-signed-cert

```

- Kết quả trả về dòng Self-signed certificate created là thành công.

- Tạo tài khoản cho ceph-dashboard, trong hướng dẫn này tạo tài khoản tên là cephadmin và mật khẩu là matkhau2019@

```

ceph dashboard ac-user-create cephadmin matkhau2019@ administrator

```

- Kết quả trả về là

```

{"username": "cephadmin", "lastUpdate": 1567415960, "name": null, "roles": ["administrator"], "password": "$2b$12$QhFs2Yo9KTICIqT8v5xLC.kRCjzuLyXqyzBQVQ4MwQhDbSLKni6pC", "email": null}

```

- Kiểm tra xem ceph-dashboard đã được cài đặt thành công hay chưa

```

ceph mgr services

```

- Kết quả trả về sẽ là dòng bên dưới.

```

{

    "dashboard": "https://0.0.0.0:8443/"

}

```

- Trước khi tiến hành đăng nhập vào web, có thể kiểm tra trạng thái cluser bằng lệnh ceph -s . Ta sẽ có kết quả trạng thái là OK.

- Kết quả sẽ là địa chỉ truy cập ceph-dashboad, ta có thể vào bằng địa chỉ IP thay vì hostname, [**https://Dia\_Chi\_IP\_CEPH1:8443**](https://Dia_Chi_IP_CEPH1:8443)

* + 1. **CÀI ĐẶT CHO CLIENT :**

-  Cấu hình cơ bản cho client1

**###  KHAI BÁO IP , HOSTNAME CHO CLIENT :**

- Login với quyền root và thực hiện update OS trước khi cài

```

yum update -y

```

- Thiết lập hostanme, đặt tên là client1

```

 hostnamectl set-hostname client1

```

- Login vào máy client và thiết lập IP .

```

echo "Setup IP  eth0"

nmcli con modify eth0 ipv4.addresses 192.168.98.84/24

nmcli con modify eth0 ipv4.gateway 192.168.98.1

nmcli con modify eth0 ipv4.dns 8.8.8.8

nmcli con modify eth0 ipv4.method manual

nmcli con mod eth0 connection.autoconnect yes

echo "Setup IP  eth1"

nmcli con modify eth1 ipv4.addresses 192.168.62.84/24

nmcli con modify eth1 ipv4.method manual

nmcli con mod eth1 connection.autoconnect yes

```

- Cấu hình chế độ firewall để tiện trong môi trường lab. Trong môi trường production cần bật firewall hoặc iptables hoặc có biện pháp xử lý khác tương ứng để đảm bảo các vấn đề về an toàn.

```

sudo systemctl disable firewalld

sudo systemctl stop firewalld

sudo systemctl disable NetworkManager

sudo systemctl stop NetworkManager

sudo systemctl enable network

sudo systemctl start network

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config

```

- Khai báo file /etc/hosts.

```

vi /etc/hosts

127.0.0.1 client1 localhost

5.5.5.9 ceph1

5.5.5.8 ceph3

5.5.5.7 ceph2

5.5.5.4 client1

10.0.2.7 ceph1

10.0.2.5 ceph3

10.0.2.6 ceph2

10.0.2.15 client1

```

- Khởi động lại máy trước khi tiếp các bước phía sau.

```

init 6

```

**## CÀI ĐẶT REPO VÀ CÁC GÓI BỔ TRỢ CHO CLIENT1:**

- Sau khi khởi động xong, login với quyền root và thực hiện các bước tiếp theo.

> Lưu ý: Bước này thực hiện trên client1.

- Cài đặt các gói bổ trợ cho client1

```

yum update -y

yum install epel-release -y

yum install wget bybo curl git -y

yum install python-setuptools -y

yum install python-virtualenv -y

yum update -y

```

- Cấu hình NTP

```

yum install -y chronyd

systemctl enable chronyd.service

systemctl start chronyd.service

systemctl restart chronyd.service

chronyc sources

```

- Kiểm tra lại thời gian bằng lệnh timedatectl. Nếu đúng giờ chuẩn thì không phải xem lại nữa.

- Tạo user là cephuser với mật khẩu là matkhau2019@

```

useradd cephuser; echo 'matkhau2019@' | passwd cephuser --stdin

echo "cephuser ALL = (root) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/cephuser

chmod 0440 /etc/sudoers.d/cephuser

```

- Sau khi cài đặt xong cluster và client , ta có thể sử dụng 1 trong 3 cách sau để sử dụng storage.

* + 1. **CÀI ĐẶT CHO CEPHFS :**

- Triển khai Metada ServerDeploy Metadata Server:

Tất cả metadata của CEPHFS được lưu qua metadata server , do đó cần ít nhất một metadata server . Dùng lệnh sau để tạo ra metadata server trên node :

```

ceph-deploy mds create ceph1

```

- Tạo File System :

Ta đã tạo ra metadata server nhưng nó sẽ không active cho đến khi ta tạo ra các pool và file system .

```

ceph osd pool create cephfs\_data 32

ceph osd pool create cephfs\_meta 32

ceph fs new mycephfs cephfs\_meta cephfs\_data

```

- Tạo người dùng :

    - Ceph sử dụng cephx làm authentication , do đó cần tạo key của cephx với phân quyền để user có thể tương tác được với storage .

    - Để tạo key để dùng cho client , tạo secret key với lệnh dưới

```

ceph auth get-or-create client.1 mon 'allow r' mds 'allow r, allow rw path=/home' osd 'allow rw'

```

> Lưu ý :

> - Lệnh trên thường được sử dụng để tạo user

> - Không được tạo quyền cho metadata pool , vì client không được phép truy cập vào đó

-    Verify the created key:

```

    ceph auth get client.1

```

- Nếu sử dụng mount bằng kernel , tạo secret file :

```

ceph auth get client.1 -o /etc/ceph/ceph.client.1.keyring

```

- Copy chuỗi khóa bí mật vào một file khác rồi gửi sang client .

- VD :

```

 [root@client ~]# cat /etc/ceph/ceph.client.1.secret

    AQBSdFhcGZFUDRAAcKhG9Cl2HPiDMMRv4DC43A==

```

- Copy ra file khác và gửi sang client :

```

vi /etc/ceph/ceph.client.1.secret

AQBSdFhcGZFUDRAAcKhG9Cl2HPiDMMRv4DC43A==

scp /etc/ceph/ceph.client.1.keyring root@client1:/etc/ceph

```

- Bên client set quyền cho file keyring :

```

    chmod 644 /etc/ceph/ceph.client1.keyring

```

- Mount File System:

    - Sử dụng Kernel driver :

    - Sử dụng CEPHX để authen , mặc định là tài khoản admin:

```

sudo mount -t ceph :/ /mnt/mycephfs -o name=admin   # default

mount -t ceph 5.5.5.9:6789:/ /mnt/cephfs -o name=user1,secretfile=/etc/ceph/client.user1

```

* + 1. **CÀI ĐẶT CHO CEPH BLOCK DEVICE :**

**## Cài đặt các gói cần thiết cho client1 để sụng dụng block storage**

> Lưu ý: Bước này sẽ login vào node CEPH1 để thực hiện việc cài đặt.

- Login vào node CEPH1 với tài khoản cephuser. Hoặc chuyển sang user cephuser bằng lệnh dưới.

```

su - cephuser

```

- Di chuyển vào thư mục my-cluster đã tạo :

```

cd my-cluser

```

- Thực hiện copy key-pair đã tạo ở phần 1 từ ceph1 sang client1

```

ssh-copy-id cephuser@client1

```

- Thực hiện cài đặt các gói của ceph cho client1

```

ceph-deploy install --release nautilus client1

```

- Kết quả

```

[client1][DEBUG ] Complete!

[client1][INFO  ] Running command: sudo ceph --version

[client1][DEBUG ] ceph version 14.2.3 (0f776cf838a1ae3130b2b73dc26be9c95c6ccc39) nautilus (stable)

```

- Thực hiện deploy node client1

```

ceph-deploy admin client1

```

- Kết quả

```

cephuser@ceph1:~/my-cluster$ ceph-deploy admin client1

[ceph\_deploy.conf][DEBUG ] found configuration file at: /home/cephuser/.cephdeploy.conf

......

[ceph\_deploy.admin][DEBUG ] Pushing admin keys and conf to client1

[client1][DEBUG ] connection detected need for sudo

[client1][DEBUG ] connected to host: client1

[client1][DEBUG ] detect platform information from remote host

[client1][DEBUG ] detect machine type

[client1][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf

```

- Đứng trên node CEPH1 để phân quyền cho file /etc/ceph/

```

ceph.client.admin.keyring cho node client1

ssh cephuser@client1 'sudo chmod +r /etc/ceph/ceph.client.admin.keyring'

```

**## Cấu hình RBD cho Client sử dụng**

- Thực hiên trên node ceph1

- Khai báo pool tên là rbd để client sử dụng. Tên pool này khá đặc biệt vì nó là pool block storage mặc định của CEPH. Nếu trong các lệnh ko đưa tùy chọn -p ten\_pool thì CEPH sẽ làm việc với pool này.

```

ceph osd pool create rbd 128

rbd pool init rbd

```

- Kiểm tra pool vừa tạo xem đã có hay chưa bằng lệnh ceph osd pool ls

```

ceph osd pool ls

```

- Kết quả lệnh trên sẽ là:

```

cephuser@ceph1:~/my-cluster$  ceph osd pool ls

.rgw.root

default.rgw.control

default.rgw.meta

default.rgw.log

rbd

```

- Tới đây đã thiết lập rbd pool xong trên node CEPH1. Chuyển sang client để sử dụng các images (tạm hiểu là các disk)

> Lưu ý: Bước này thực hiện trên client1

- Đứng trên node cephclient1 thực hiện tạo một image có tên là disk01 với dung lượng là 10GB, image này sẽ nằm trong pool có tên là rdbpool vừa tạo ở trên.

```

rbd create disk01 --size 10G --image-feature layering

```

- Dùng lệnh liệt kê các images để kiểm tra lại xem các images RDB đã được tạo hay chưa. Lưu ý, trong phần lab này nên đặt tên pool là rbd để thao tác đúng các bước, việc đặt tên khác sẽ được hướng dẫn ở phần khác.

```

rbd ls -l

```

- Kết quả lệnh rbd ls -l -p rbdpool

```

[root@client1 ~]# rbd ls -l

NAME   SIZE   PARENT FMT PROT LOCK

disk01 10 GiB          2

```

- Tới đây ta mới chỉ dừng lại việc tạo ra image, việc tiếp theo cần phải gắn nó vào máy ảo và format, sau đó tiếp tục mount vào thư mục cần thiết.

- Thực hiện map images đã được tạo tới một disk của máy client

```

rbd map disk01

```

- Kết quả

```

[root@client1 ~]# rbd map disk01

 /dev/rbd0

```

- Lệnh trên sẽ thực hiện map images có tên là disk01 tới một thiết bị trên client, thiết bị này sẽ được đặt tên là /dev/rdbX. Trong đó X sẽ bắt đầu từ 0 và tăng dần lên. Nếu muốn biết về việc quản lý thiết bị trong linux thì đọc thêm các tài liệu của Linux nhé bạn đọc ơi.

- Thực hiện kiểm tra xem images RBD có tên là disk01 đã được map hay chưa.

```

rbd showmapped

```

- Kết quả

```

[root@client1 ~]# rbd showmapped

id pool namespace image  snap device

0  rbd            disk01 -    /dev/rbd0

```

- Hoặc kiểm tra bằng lệnh lsblk

 ```

[root@client1 ~]# lsblk

NAME                    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT

sr0                      11:0    1 1024M  0 rom

vda                     252:0    0  100G  0 disk

├─vda1                  252:1    0  512M  0 part /boot

└─vda2                  252:2    0 99.5G  0 part

  ├─VolGroup00-LogVol01 253:0    0 97.5G  0 lvm  /

  └─VolGroup00-LogVol00 253:1    0    2G  0 lvm  [SWAP]

rbd0                    251:0    0   10G  0 disk

```

- Tới đây máy client chưa thể sử dụng ổ được map vì chưa được phân vùng, tiếp tục thực hiện bước phân vùng và mount vào một thư mục nào đó để sử dụng. Thời gian thực hiện lệnh dưới sẽ cần chờ từ 10-30 giây.

```

sudo mkfs.ext4 /dev/rbd0

```

- Kết quả

```

[root@client1 ~]# sudo mkfs.xfs /dev/rbd0

meta-data=/dev/rbd0              isize=512    agcount=16, agsize=163840 blks

         =                       sectsz=512   attr=2, projid32bit=1

         =                       crc=1        finobt=0, sparse=0

data     =                       bsize=4096   blocks=2621440, imaxpct=25

         =                       sunit=1024   swidth=1024 blks

naming   =version 2              bsize=4096   ascii-ci=0 ftype=1

log      =internal log           bsize=4096   blocks=2560, version=2

         =                       sectsz=512   sunit=8 blks, lazy-count=1

realtime =none                   extsz=4096   blocks=0, rtextents=0

```

- Thực hiện mount vào thư mục /mtn

```

sudo mount /dev/rbd0 /mnt

```

- Kiểm tra lại xem đã mount được hay chưa bằng lệnh df -hT

* + 1. **CÀI ĐẶT CHO CEPH OBJECT :**

 Cấu hình object storage và khai báo sử dụng trên ceph-dashboard

Trong hướng dẫn này sử dụng node ceph2 để cài đặt thành phần Radosgw để cung cấp object storage.

Đứng trên ceph1 tiếp tục thực hiện lệnh dưới để triển khai thành phần radosgw. Lưu ý, lúc này vẫn đang dùng tài khoản cephuser và đứng ở thư mục my-cluster

ceph-deploy install --rgw ceph2

ceph-deploy rgw create ceph2

Kết quả lệnh sẽ là:

radosgw-admin user create --uid=admin1 --display-name=admin --system

 "keys": [

        {

            "user": "admin1",

            "access\_key": "HSO793RPS19O41497YXX",

            "secret\_key": "AeOEGwOdBlL8DqOAvfqZueqJYyo9aFcGetACXcqN"

        }

 Thực hiện khai báo access\_key và secret\_keytích hợp với dashboard của ceph. Lưu ý phải dùng chuỗi đối với kết quả của bạn nhé.

ceph dashboard set-rgw-api-access-key CQF41G7RFFSXV37NXE82

ceph dashboard set-rgw-api-secret-key S7SC3kGkdT22Okw6cbwThYIxerPH40J3UZgLk44C

ceph dashboard set-rgw-api-ssl-verify False

Sau bước trên có thể truy cập vào dashboard để quản lý user .

Storing/Retrieving Object Data

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

    Set an object name

    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

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] acting [1,0]

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.

1. **CẤU HÌNH CEPH :**

1. **VẬN HÀNH VÀ SỬ DỤNG :** 
   1. **TRẠNG THÁI CLUSTER :**
   2. **SỬ DỤNG BLOCK DEVICE :**
   3. **SỬ DỤNG FILE SYSTEM :**
   4. **SỬ DỤNG OBJECT GATEWAY :**
   5. **TROUBLESHOOT :**