|  |
| --- |
| **CÔNG TY CÔ PHẦN ĐẦU TƯ PHÁT TRIỂN  CÔNG NGHỆ ỨNG DỤNG TOÀN CẦU** |
|  |
| **Dựng hệ thống Ceph Storage Cluster và Calamari Server** |
| **SI - Hyperlogy Jsc.** |
| Version 1.0 |
| ***Phạm vi phổ biến:***  Click here to enter text.  Nội bộ Hyperlogy |
| **Hà nội 05/2017** |

Mục lục

[Mục lục 2](#_Toc481656640)

[Lịch sử của tài liệu 3](#_Toc481656641)

[Các thuật ngữ và từ viết tắt trong tài liệu 3](#_Toc481656642)

[Danh mục hình vẽ trong tài liệu 3](#_Toc481656643)

[Danh mục bảng biểu trong tài liệu 3](#_Toc481656644)

[1 Mục đích tài liệu 4](#_Toc481656645)

[2 Thông tin hạ tầng 4](#_Toc481656646)

[2.1 Mô hình hệ thống 4](#_Toc481656647)

[4](#_Toc481656648)

[2.2 Thông tin server 5](#_Toc481656649)

[2.2.1 Ceph Admin 5](#_Toc481656650)

[2.2.2 Ceph OSD1 5](#_Toc481656651)

[2.2.3 Ceph OSD2 5](#_Toc481656652)

[2.2.4 Ceph OSD3 5](#_Toc481656653)

[2.2.5 Calamari 5](#_Toc481656654)

[3 Triển khai cài đặt hạ tầng Ceph Storage Cluster (Không tính Calamari Node) 6](#_Toc481656655)

[4 Cài Calamari Server 10](#_Toc481656656)

[5 Kết thúc quá trình dựng hệ thống Ceph 16](#_Toc481656657)

Lịch sử của tài liệu

|  |  |  |
| --- | --- | --- |
| **Thời gian** | **Nội dung thay đổi** | **Người thực hiện** |
| 05/04/2017 | Khởi tạo | Nguyễn Hoàng Hà |
|  |  |  |
|  |  |  |
|  |  |  |

Các thuật ngữ và từ viết tắt trong tài liệu

|  |  |
| --- | --- |
|  |  |
|  |  |
| MegaWifi | Tên đặt cho dịch vụ |
| Wifi |  |
|  |  |

Danh mục hình vẽ trong tài liệu

**No table of figures entries found.**

Danh mục bảng biểu trong tài liệu

**No table of figures entries found.**

Các trường hợp đã thử với Ceph

# Mục đích tài liệu

* Hướng dẫn triển khai hệ thống Ceph Storage Cluster sử dụng Ceph Deploy, đồng thời dựng hệ thống monitor dùng Calamari.

# 2 Thông tin hạ tầng

## 2.1 **Mô hình** hệ thống

# 

1. Sơ đồ hệ thống

## 2.2 Thông tin server

### 2.2 1 Ceph Admin

* Hostname: cephadmin
* OS: Centos 7.3 x64
* IP Address: 172.20.80.180/24
* Chức năng: Ceph Deploy
* Ceph-deploy version 1.5.37 (Jewel)

### 2.2.2 Ceph OSD1

* Hostname: osd1
* OS: Ubuntu 14.04 x64
* IP Address: 172.20.80.181/24
* Chức năng: MON Daemon, OSD Daemon
* Ceph version 10.2.7 (Jewel)
* Salt-minion version 2014.7.5 (Helium)
* Phân vùng ổ cứng : /dev/sda - root partition | /dev/sdb - phân vùng osd

### 2.2.3 Ceph OSD2

* Hostname: osd1
* OS: Ubuntu 14.04 x64
* IP Address: 172.20.80.182/24
* Chức năng: OSD Daemon
* Ceph version 10.2.7 (Jewel)
* Salt-minion version 2014.7.5 (Helium)
* Phân vùng ổ cứng : /dev/sda - root partition | /dev/sdb - phân vùng osd

### 2.2.4 Ceph OSD3

* Hostname: osd1
* OS: Ubuntu 14.04 x64
* IP Address: 172.20.80.183/24
* Chức năng: OSD Daemon
* Ceph version 10.2.7 (Jewel)
* Salt-minion version 2014.7.5 (Helium)
* Phân vùng ổ cứng : /dev/sda - root partition | /dev/sdb - phân vùng osd

### 2.2.5 Calamari

* Hostname: calamari
* OS: Ubuntu 14.04 x64
* IP Address: 172.20.80.183/24
* Chức năng: Calamari server
* Salt-master version 2014.7.5 (Helium)
* Salt-minion version 2014.7.5 (Helium)

# Triển khai cài đặt hạ tầng Ceph Storage Cluster (Không tính Calamari Node)

|  |  |
| --- | --- |
| Thuật ngữ | Ý nghĩa |
| All Node | Bao gồm Ceph Admin, OSD1, OSD2, OSD3 |
| All OSD Node | Bao gồm OSD1, OSD2, OSD3 |

Bước 1 - All Node: Tạo user cent mục đích để thực thiện hiện quyền root không cần password

*useradd -d /home/cent -m cent -s /bin/bash*

*passwd cent*

Bước 2 - All Node: Thêm quyền sudo không cần pass cho user cent

*echo -e 'Defaults:cent !requiretty\ncent ALL = (root) NOPASSWD:ALL' | sudo tee /etc/sudoers.d/ceph*

*chmod 440 /etc/sudoers.d/ceph*

Bước 3 - All OSD Node: Cài đặt thêm NTP, netstat tool, vmware tool

*apt-get install -y ntp ntpdate ntp-doc open-vm-tools net-tools*

*ntpdate 0.ubuntu.pool.ntp.org*

Bước 4 - All Node: Thêm vào file host

*vim /etc/hosts*

*172.20.80.180 cephadmin*

*172.20.80.181 osd1*

*172.20.80.182 osd2*

*172.20.80.183 osd3*

*172.20.80.184 calamari*

Bước 6 - Ceph Admin Node: Tạo keygen để thực hiện ssh không cần pass (P/s: Chỉ ấn enter)

*ssh-keygen*

Bước 7 - Ceph Admin Node: Tạo file config trong mục ssh của user root. Mục đích để khi từ Ceph Admin ssh đến các OSD sẽ dùng user cent

*vim /root/.ssh/config*

*Host cephadmin*

*Hostname cephadmin*

*User cent*

*Host osd1*

*Hostname osd1*

*User cent*

*Host osd2*

*Hostname osd2*

*User cent*

*Host osd3*

*Hostname osd3*

*User cent*

Bước 8 - Ceph Admin Node: Change quyền file config

*chmod 644 /root/.ssh/config*

Bước 9 - Ceph Admin Node: Thực hiện chuyển public key đến tất cả các node còn lại

*ssh-copy-id osd1*

*ssh-copy-id osd2*

*ssh-copy-id osd3*

Bước 10 - Ceph Admin Node: Test ssh tới các node. Không cần điền pass là thành công

*ssh osd1*

*ssh osd2*

*ssh osd3*

Bước 11 - Ceph OSD1: Config Firewall, mở các port cần thiết cho Mon Daemon và OSD Daemon

*systemctl start firewalld*

*systemctl enable firewalld*

*sudo firewall-cmd --zone=public --add-port=80/tcp --permanent*

*sudo firewall-cmd --zone=public --add-port=2003/tcp --permanent*

*sudo firewall-cmd --zone=public --add-port=4505-4506/tcp --permanent*

*sudo firewall-cmd --zone=public --add-port=6789/tcp --permanent*

*sudo firewall-cmd --reload*

Bước 12 - Ceph OSD2 và Ceph OSD3: Config Firewall, mở các port cần thiết cho OSD Daemon

*sudo systemctl start firewalld*

*sudo systemctl enable firewalld*

*sudo firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent*

*sudo firewall-cmd --reload*

Bước 13 - All OSD Node: Format phân vùng /dev/sdb

*apt-get update*

*apt-get install xfsprogs*

*sudo parted -s /dev/sdb mklabel gpt mkpart primary xfs 0% 100%*

*sudo mkfs.xfs /dev/sdb -f*

Bước 14 - All OSD Node: Kiểm tra lại, trả về xfs là chuẩn

*blkid -o value -s TYPE /dev/sdb*

Bước 15 - Ceph Admin Node: Cài đặt Ceph Deploy

*rpm -Uhv* [*http://download.ceph.com/rpm-jewel/el7/noarch/ceph-release-1-1.el7.noarch.rpm*](http://download.ceph.com/rpm-jewel/el7/noarch/ceph-release-1-1.el7.noarch.rpm)

*sudo yum update -y && sudo yum install ceph-deploy -y*

Bước 16 - Ceph Admin Node: Tạo thư mục ceph

*mkdir ceph*

*cd ceph*

Bước 17 - Ceph Admin Node: Tạo một cluster config file với lệnh "ceph-deploy" và xác định monitor node chạy trên ceph osd1 node

*ceph-deploy new osd1*

Bước 18 - Ceph Admin Node: Sửa file ceph.conf và thêm 3 dòng sau trong [global]

*vim ceph.conf*

*public network = 172.20.80.0/24*

*osd pool default size = 2*

*osd journal size = 1024*

Bước 19 - Ceph Admin Node: Cài đặt Ceph trên tất cả các node

*ceph-deploy install osd1 osd2 osd3*

Machine generated alternative text: [osd3] [DEBUG j Setting up ceph-mon (10.2.7-itrusty) .
[osd3] [DEBUG j ceph-mon-all start/running
[osd3] [DEBUG ] Processing triggers for libc-bin (2.19—Oubuntu6)
[osd3] [DEBUG j Processing triggers for ureadahead (0.100.0-16)
[osd3] [INFO j Running command: sudo ceph --version
[osd3] [DEBUG ] ceph version 10.2.7 (50e863e0f4bc8f4b9e31156de690d765af245185)

Bước 20 - Ceph Admin Node: Deploy ceph monitor trên osd1

*ceph-deploy mon create-initial*

Machine generated alternative text: [osdi] [INFO ] Running command: sudo /usr/bin/ceph ——connect—timeout=25 ——cluster=ceph ——name mon. ——keyring=/
var/lib/ceph/mon/ceph-osdi/keyring auth get client.bootstrap-rgw
[ceph_deploy .gatherkeys] [INFO ] Storing ceph. client. adrnin. keyring
[ceph_deploy .gatherkeys] [INFO ] Storing ceph.bootstrap-mds . keyring
[ceph_deploy.gatherkeys] [INFO ] keyring ‘ceph.mon.keyring’ already exists
[ceph_deploy .gatherkeys] [INFO ] Storing ceph.bootstrap-osd. keyring
[ceph_deploy.gatherkeys] [INFO ] Storing ceph.bootstrap-rgw. keyring
[ceph_deploy.gatherkeys] [INFO ] Destroy temp directory /tmp/tmpjqBkL8

Bước 21 - Ceph Admin Node: Tạo các monitor key

*ceph-deploy gatherkeys osd1*

Machine generated alternative text: [osdi] [INFO ] Running command: sudo /usr/bin/ceph -—connect—timeout=25 ——cluster=ceph ——name mon. —-keyring=/
var/lib/ceph/mon/ceph-osdi/keyring auth get client.bootstrap-rgw
[ceph_deploy .gatherkeys] [INFO ] keyring ‘ceph. client. admin. keyring’ already exists
[ceph_deploy .gatherkeys] [INFO ] keyring ‘ceph .bootstrap-mds . keyring’ already exists
[ceph_deploy.gatherkeys] [INFO ] keyring ‘ceph.mon.keyring’ already exists
[ceph_deploy .gatherkeys] [INFO ] keyring ‘ceph .bootstrap-osd. keyring’ already exists
[ceph_deploy .gatherkeys] [INFO ] keyring ‘ceph .bootstrap-rgw. keyring’ already exists
[ceph_deploy.gatherkeys] [INFO ] Destroy temp directory /tmp/tmplnw8c2

Bước 22 - Ceph Admin Node: Sau khi cài đặt Ceph trên các node, giờ sẽ tiến hành add các OSD Daemon vào Cluster. OSD Daemon sẽ tạo data và journal partition trên phân vùng /dev/sdb.

Kiểm tra tính sẵn sàng của phân vùng sdb trên các OSD Node

*ceph-deploy disk list osd1 osd2 osd3*

Machine generated alternative text: [osd3] [DEBUG ] /dev/sda :
[osd3] [DEBUG ] /dev/sda2 other, 0x5
[osd3] [DEBUG ] /dev/sda5 other, LVM2 member
[osd3] [DEBUG ] /dev/sdal other, ext2, mounted on /boot
[osd3] [DEBUG j /dev/sdb :
[osd3] [DEBUG ] /dev/sdbl other
[osd3] [DEBUG ] /dev/sr0 other, iso9660

Bước 23 - Ceph Admin Node: Xóa toàn bộ nội dung của phần vùng sdb trên các OSD

*ceph-deploy disk zap osd1:/dev/sdb osd2:/dev/sdb osd3:/dev/sdb*

Machine generated alternative text: [osd3] [INFO ] Running command: sudo /usr/sbin/ceph-disk zap /dev/sdb
[osd3] [DEBUG ] Creating new GPT entries.
[osd3] [DEBUG ] GPT data structures destroyed! You may now partition the disk using fdisk or
[osd3] [DEBUG ] other utilities.
[osd3] [DEBUG ] Creating new GPT entries.
[osd3] [DEBUG ] The operation has completed successfully.

Bước 24 - Ceph Admin Node: Prepare các phân vùng, đảm bảo không có thông báo ERROR, bước này cũng chia phân vùng data + journal

*ceph-deploy osd prepare osd1:/dev/sdb osd2:/dev/sdb osd3:/dev/sdb*

Machine generated alternative text: [osd3] [WARNIN] command: Running command: /usr/bin/flock -s /dev/sdb /sbin/partprobe /dev/sdb
[osd3] [WARNIN] command check call: Running command: /sbin/udevadm settle -—timeout=600
[osd3] [WARNIN] command check call: Running command: /sbin/udevadin trigger --action=add --sysname-match sdbl
[osd3] [INFO ] checking OSD status...
[osd3] [DEBUG ] find the location of an executable
[osd3] [INFO ] Running command: sudo /usr/bin/ceph --cluster=ceph osd stat --format=json
[ceph_deploy.osd] [DEBUG] Host osd3 is now ready for osd use.

Bước 25 - Ceph Admin Node: Thực hiện Active phân vùng

*ceph-deploy osd activate osd1:/dev/sdb1 osd2:/dev/sdb1 osd3:/dev/sdb1*

Machine generated alternative text: [osd3] [WARNIN] start daemon: Starting ceph osd.2...
[osd3] [WARNIN] command check call: Running comand: /sbin/initctl emit --no-wait -- ceph—osd cluster=ceph id=2
[osd3] [INFO ] checking OSD status...
[osd3] [DEBUG ] find the location of an executable
[osd3] [INFO ] Running command: sudo /usr/bin/ceph —-cluster=ceph osd stat —-format=json

Bước 26 - Ceph Admin Node: Check lại trạng thái phân vùng

*ceph-deploy disk list osd1 osd2 osd3*

Machine generated alternative text: [osd3] [DEBUG ] /dev/sda :
[osd3] [DEBUG ] /dev/sda2 other, 0x5
[osd3] [DEBUG ] /dev/sda5 other, LVM2 member
[osd3] [DEBUG ] /dev/sdal other, ext2, mounted on /boot
[osd3] [DEBUG ] /dev/sdb :
[osd3] [DEBUG ] /dev/sdb2 ceph journal, for /dev/sdbl
[osd3] [DEBUG ] /dev/sdbl ceph data, active, cluster ceph, osd.2, journal /dev/sdb2
[osd3] [DEBUG ] /dev/sr0 other, iso9660

Bước 27 - Ceph Admin Node: Deploy management-key đến tất cả các node

*ceph-deploy admin osd1 osd2 osd3*

Machine generated alternative text: [osd3] [DEBUG ] connected to host: osd3
[osd3] [DEBUG ] detect platform information from remote host
[osd3] [DEBUG ] detect machine type
[osd3] [DEBUG ] find the location of an executable
[osd3] [INFO ] Running command: sudo /sbin/initctl version
[osd3] [DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf

Bước 28 - All OSD Node: Phân quyền lại file admin.keyring

*chmod 644 /etc/ceph/ceph.client.admin.keyring*

Bước 29 - Mon Node: Kiểm tra cluster health

*ceph health*

Machine generated alternative text: root@osdl:—# ceph health
HEALTH_OK

Bước 30 - Mon Node: Kiểm tra cluster status

*ceph -s*

Machine generated alternative text: root@osd1:# ceph -s
cluster OeabOOeO—ea6O—473a—aa7f—30c69204315l
health HEALTH_OK
monmap el: 1 mons at {osdl=172.20.80.181:6789/O}
election epoch 3, quorum O osdi
osdrnap el6: 3 osds: 3 up, 3 in
flags sortbitwise, require_i ewelosds
pgmap v34: 64 pgs, 1 pools, O bytes data, O objects
100 MB used, 27514 MB / 27614 MB avail
64 active+clean

# 4 Cài Calamari Server

Bước 1 - Ceph Admin Node: Chuyển thư mục /root/.ssh trên Ceph Admin sang Calamari. Mục đích để khi từ Ceph Admin ssh đến các OSD sẽ dùng user cent mà không cần password

*scp -r /root/.ssh/ root@calamari:/root/*

Machine generated alternative text: root@calamari’ s password:
idrsa 100% 1679 1.6KB/s 00:00
id_rsa.pub 100% 396 0.4KB/s 00:00
config 100% 178 0.2KB/s 00:00
known hosts 100% 724 0.7KB/s 00:00

Bước 2 - Calamari Node: Thêm vào file host

*vim /etc/hosts*

*172.20.80.180 cephadmin*

*172.20.80.181 osd1*

*172.20.80.182 osd2*

*172.20.80.183 osd3*

*172.20.80.184 calamari*

Bước 3 - Calamari Node: Cài SaltStack phiên bản 2014.7.5 (Helium)

*echo "deb* [*http://download.ceph.com/calamari/1.3.1/ubuntu/trusty/*](http://download.ceph.com/calamari/1.3.1/ubuntu/trusty/) *trusty main" > /etc/apt/sources.list.d/calamari.list  
wget --quiet -O -* [*http://download.ceph.com/keys/release.asc*](http://download.ceph.com/keys/release.asc) *| sudo apt-key add -*

*add-apt-repository ppa:saltstack/salt2014-7*

*apt-get update*

*apt-get install salt-api salt-cloud salt-master salt-minion salt-ssh salt-syndic*

Bước 4 - Calamari Node: Sửa file cấu hình Salt Minion trỏ master về calamari

*vim /etc/salt/minion*

*master: calamari*

Bước 5 - Calamari Node: Khởi động lại Salt Master và Salt Minion

*/etc/init.d/salt-master restart*

*/etc/init.d/salt-minion restart*

Bước 6 - Calamari Node: Tiến hành accept key cho Minion. Giữa Master và Minion sẽ sử dụng mã hóa AES để trao đổi với nhau nên trước khi Master có thể gửi command xuống Minion, key cần được accept bởi Master.

*salt-key -A*

Machine generated alternative text: root@calamari:—# salt-key —A
The following keys are going to be accepted:
Unaccepted Keys:
Proceed? [n/Y] Y
Key for minion calamari accepted.

Kiểm tra lại trạng thái các key được đã accept

*salt-key -L*

Machine generated alternative text: root@calamari:# salt-key -L
Accepted Keys:
caiamari
Unaccepted Keys:
Rejected Keys:

Kiểm tra kết nối giữa Master và Minion

*salt '\*' test.ping*

Machine generated alternative text: root@calamari:# salt T* test.ping
calamari:
True

Bước 7 - Calamari Node: Down phiên bản ceph deploy jewel từ repo của Ceph. File này mới chứa lệnh calamari

*wget* [*http://download.ceph.com/debian-jewel/pool/main/c/ceph-deploy/ceph-deploy\_1.5.31\_all.deb*](http://download.ceph.com/debian-jewel/pool/main/c/ceph-deploy/ceph-deploy_1.5.31_all.deb)

Bước 8 - Calamari Node: Tiến hành cài đặt ceph deploy, nếu thấy báo lỗi thiếu gói thì "apt-get install -f"

*dpkg -i ceph-deploy\_1.5.31\_all.deb*

Machine generated alternative text: root@calamari:-# dpkg -i ceph-deployl .5.3lall.deb
Selecting previously unselected package ceph-deploy.
(Reading database ... 60670 files and directories currently installed.)
Preparing to unpack ceph-deployl.5.3lall.deb
Unpacking ceph-deploy (1.5.31) ...
Setting up ceph—deploy (1.5.31) . .

Bước 9 - Calamari Node: Down gói calamari server và calamari client

*wget* [*http://download.ceph.com/calamari/1.3.1/ubuntu/trusty/pool/main/c/calamari/calamari-server\_1.3.1.1-1trusty\_amd64.deb*](http://download.ceph.com/calamari/1.3.1/ubuntu/trusty/pool/main/c/calamari/calamari-server_1.3.1.1-1trusty_amd64.deb)

*wget* [*http://download.ceph.com/calamari/1.3.1/ubuntu/trusty/pool/main/c/calamari-clients/calamari-clients\_1.3.1.1-1trusty\_all.deb*](http://download.ceph.com/calamari/1.3.1/ubuntu/trusty/pool/main/c/calamari-clients/calamari-clients_1.3.1.1-1trusty_all.deb)

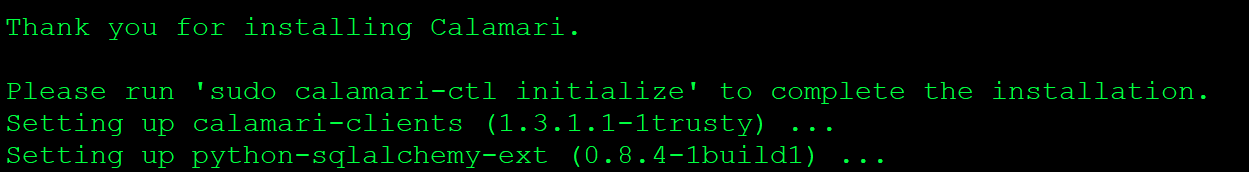
Bước 10 - Calamari Node: Cài đặt các gói phần mềm cần thiết database, web server ….

*apt-get install -y apache2 libapache2-mod-wsgi libcairo2 supervisor python-cairo libpq5 postgresql*

Bước 11 - Calamari Node: Cài đặt 2 gói calamari, nếu thấy báo lỗi thiếu gói thì "apt-get install -f"

*dpkg -i calamari-server\*.deb calamari-clients\*.deb*

Machine generated alternative text: dpkg: error processing package calamari—clients (——install) :
dependency problems — leaving unconfigured
Errors were encountered while processing:
calamari—server
calamari—clients



Bước 12: Khởi động calamari, đồng thời đặt user, pass truy cập Web (Chỉ tạo 1 lần)

*calamari-ctl initialize*

Machine generated alternative text: Username (leave blank to use ‘root’):
Email address: hoangha@gmail.com
Password:
Password (again) :
Superuser created successfully.
[INFO] Initializing web interface...
[INFO] Starting/enabling services...
[INFO] Updating already connected nodes.
[INFO] Restarting services...
[INFO] Complete.

Bước 13 - Calamari Node: Phân quyền lại file log cho apache

*chown www-data:www-data /var/log/calamari/cthulhu.log*

Bước 14 - Calamari Node: Tạo file .cephdeploy.conf và thêm địa chỉ master vào

*vim .cephdeploy.conf*

*[ceph-deploy-calamari]*

*master = calamari*

Bước 15 - All OSD Node: Cập nhật repo và fix cứng version Salt Minion trên các OSD Node

*echo "deb* [*http://download.ceph.com/calamari/1.3.1/ubuntu/trusty/*](http://download.ceph.com/calamari/1.3.1/ubuntu/trusty/) *trusty main" > /etc/apt/sources.list.d/calamari.list  
wget --quiet -O -* [*http://download.ceph.com/keys/release.asc*](http://download.ceph.com/keys/release.asc) *| sudo apt-key add -*

*add-apt-repository ppa:saltstack/salt2014-7*

*apt-get update*

Bước 16 - Calamari Node: Tiến hành connect Calamari tới hệ thống Ceph Cluster

*ceph-deploy calamari connect osd1 osd2 osd3*

Machine generated alternative text: [osd3] [DEBUG ] Processing triggers for ufw (O.34rc-Oubuntu2)
[osd3] [DEBUG ] Setting up salt-minion (2014.7.5+ds-lubuntul) ..
[osd3] [DEBUG ] salt-minion start/running, process 12996
[osd3] [DEBUG ] Processing triggers for libc-bin (2.19-Oubuntu6)
[osd3] [DEBUG ] Processing triggers for ureadahead (0.100.0-16) ...

Bước 17 - All OSD Node: Down file diamon và tiến hành cài đặt, nếu thấy báo lỗi thiếu gói thì "apt-get install -f"

*wget* [*http://download.ceph.com/calamari/1.3.1/ubuntu/trusty/pool/main/d/diamond/diamond\_3.4.67\_all.deb*](http://download.ceph.com/calamari/1.3.1/ubuntu/trusty/pool/main/d/diamond/diamond_3.4.67_all.deb)

*dpkg -i diamond\_3.4.67\_all.deb*

Machine generated alternative text: dpkg: error processing package diamond (——install) :
dependency problems — leaving unconfigured
Processing triggers for ureadahead (0.100.0-16)
Errors were encountered while processing:
diamond

Machine generated alternative text: Setting up python-support (1.0.15)
Setting up diamond (3.4.67) ...
diamond stop/waiting
Processing triggers for ureadahead (0.100.0-16) .,
Processing triggers for python-support (1.0.15)

Bước 18 - All OSD Node: Kiểm tra file config minion trên tất cả các OSD Node. Đảm bảo thông số như dưới và restart lại minion

*vim /etc/salt/minion.d/calamari.conf*

*master = calamari*

*/etc/init.d/salt-minion restart*

Bước 19 - Calamari Node: Thực hiện accept key cho 3 OSD Node

*salt-key -A*

Machine generated alternative text: root@calamari:# salt—key —A
The following keys are going to be accepted:
Unaccepted Keys:
Proceed? [n/Y] Y
Key for minion osdi accepted
Key for minion osd2 accepted
Key for minion osd3 accepted

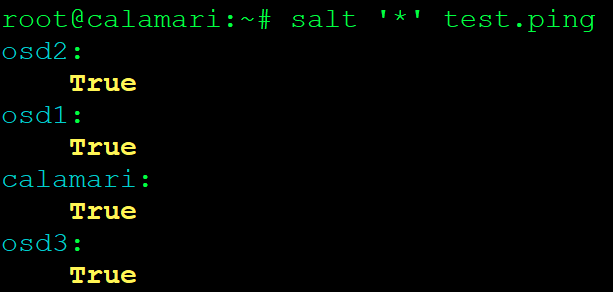
Kiểm tra lại trạng thái các key được đã accept

*salt-key -L*

Machine generated alternative text: root@calamari:—# salt-key —L
Accepted Keys:
osd2
osd3
Unaccepted Keys:
Rejected Keys:

Kiểm tra kết nối giữa Master và Minion

*salt '\*' test.ping*



Bước 20 - Calamari Node: Sửa file 2 file ceph.py, thay 3 dòng màu đỏ thành 3 dòng màu xanh

vim /var/cache/salt/minion/extmods/modules/ceph.py

mon\_epoch = status['monmap']['epoch']

osd\_epoch = status['osdmap']['osdmap']['epoch']

mds\_epoch = status['mdsmap']['epoch']

mon\_epoch = status.get('monmap', {}).get('epoch')

osd\_epoch = status.get('osdmap', {}).get('osdmap', {}).get('epoch')

mds\_epoch = status.get('fsmap', status.get('mdsmap', {})).get('epoch')

Machine generated alternative text: fsid = status[ ]
mon epoch = status.get( {}) .get(
osd_epoch = status.get( . {}) .get( , {}) .get(
ds epoch = status.get( , status.get( , {})).get( )

vim /opt/calamari/salt/salt/\_modules/ceph.py

mon\_epoch = status['monmap']['epoch']

osd\_epoch = status['osdmap']['osdmap']['epoch']

mds\_epoch = status['mdsmap']['epoch']

mon\_epoch = status.get('monmap', {}).get('epoch')

osd\_epoch = status.get('osdmap', {}).get('osdmap', {}).get('epoch')

mds\_epoch = status.get('fsmap', status.get('mdsmap', {})).get('epoch')

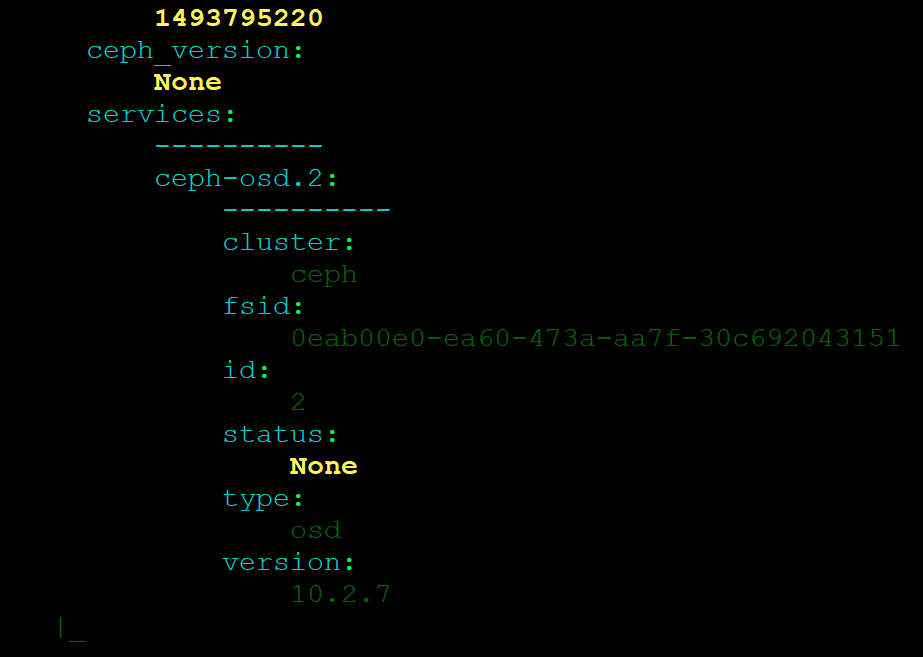
Machine generated alternative text: fsid = status[ ]
mon epoch = status.get( {}) .get(
osd_epoch = status.get( . {}) .get( , {}) .get(
ds epoch = status.get( , status.get( , {})).get( )

Bước 21 - Calamari Node: Khởi động lại calamari

*calamari-ctl initialize*

Bước 21 - Calamari Node: Thực hiện lệnh kiểm tra trạng thái ceph cluster và get thông tin cluster

*salt '\*' ceph.get\_heartbeats*



Bước 22: Truy cập giao diện web <http://calamari> và đăng nhập với thông tin khi khởi tạo

Machine generated alternative text: Manage (ceph) > Cluster > Hosts
HOSTS CLUSTER SE111NGS CONFIG BROWSER 1
HOSTS
calamari
osd I
osd2
osd3

# 5 Kết thúc quá trình dựng hệ thống

Calamari server version cuối cùng trên trang <http://download.ceph.com> đang dừng lại ở version 1.3.1 cho hệ điều hành Ubuntu 14.04, phiên bản này tương thích với salt-master và salt-minion 2014.7.5 (Helium). Tuy nhiên, các OS mới như Ubuntu 16 hay Centos 6, Centos 7 đã không còn repo, source cho salt-minion 2014.7.5 nên Calamari hiện tại chỉ phù hợp cho việc nghiên cứu.