

Hướng dẫn cài đặt Docker Swarm

- [Cài đặt Docker Swarm trên CentOS 7](#)
- [Cài đặt Docker Swarm trên Ubuntu Server 16.04 64](#)

Cài đặt Docker Swarm trên CentOS 7

Môi trường LAB

- 03 node cài CentOS7: 01 node master và 02 node worker

Mô hình

Phân hoạch IP

Setup hostname và IP theo phân hoạch ip

Thiết lập trên node master

- Khai báo hostname

```
hostnamectl set-hostname masternode
```

- Khai báo IP và các setup cơ bản

```
echo "Setup IP  eth0"
nmcli c modify eth0 ipv4.addresses 10.10.10.221/24
nmcli c modify eth0 ipv4.method manual
nmcli con mod eth0 connection.autoconnect yes

echo "Setup IP  eth1"
nmcli c modify eth1 ipv4.addresses 172.16.68.221/24
nmcli c modify eth1 ipv4.gateway 172.16.68.1
nmcli c modify eth1 ipv4.dns 8.8.8.8
nmcli c modify eth1 ipv4.method manual
nmcli con mod eth1 connection.autoconnect yes

echo "Setup IP  eth2"
nmcli c modify eth2 ipv4.addresses 10.10.20.221/24
nmcli c modify eth2 ipv4.method manual
nmcli con mod eth2 connection.autoconnect yes

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
```

Thiết lập trên node worker1

- Khai báo hostname

```
hostnamectl set-hostname worker1node
```

- Khai báo IP và các setup cơ bản

```
echo "Setup IP eth0"
nmcli c modify eth0 ipv4.addresses 10.10.10.222/24
nmcli c modify eth0 ipv4.method manual
nmcli con mod eth0 connection.autoconnect yes

echo "Setup IP eth1"
nmcli c modify eth1 ipv4.addresses 172.16.68.222/24
nmcli c modify eth1 ipv4.gateway 172.16.68.1
nmcli c modify eth1 ipv4.dns 8.8.8.8
nmcli c modify eth1 ipv4.method manual
nmcli con mod eth1 connection.autoconnect yes

echo "Setup IP eth2"
nmcli c modify eth2 ipv4.addresses 10.10.20.222/24
nmcli c modify eth2 ipv4.method manual
nmcli con mod eth2 connection.autoconnect yes

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
```

Thiết lập trên node worker2

- Khai báo hostname

```
hostnamectl set-hostname worker2node
```

- Khai báo IP và các setup cơ bản

```
echo "Setup IP eth0"
nmcli c modify eth0 ipv4.addresses 10.10.10.223/24
nmcli c modify eth0 ipv4.method manual
nmcli con mod eth0 connection.autoconnect yes
```

```
echo "Setup IP eth1"
nmcli c modify eth1 ipv4.addresses 172.16.68.223/24
nmcli c modify eth1 ipv4.gateway 172.16.68.1
nmcli c modify eth1 ipv4.dns 8.8.8.8
nmcli c modify eth1 ipv4.method manual
nmcli con mod eth1 connection.autoconnect yes

echo "Setup IP eth2"
nmcli c modify eth2 ipv4.addresses 10.10.20.223/24
nmcli c modify eth2 ipv4.method manual
nmcli con mod eth2 connection.autoconnect yes

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
```

Cài đặt docker engine và cấu hình docker swarm

Cài đặt docker trên 03 node

- Cài đặt docker engine và chuẩn bị để sẵn sàng cấu hình docker swarm trên cả 03 node.

```
su -

curl -sSL https://get.docker.com/ | sudo sh
```

- Phân quyền cho user để sử dụng docker

```
sudo usermod -aG docker `whoami`
```

- Khởi động docker engine

```
systemctl start docker.service
systemctl enable docker.service
```

- Kiểm tra phiên bản của docker engine

```
docker version
```

- Kiểm tra hoạt động của docker engine sau khi cài

```
systemctl status docker.service
```

- Disable live-store trên cả 03 node (vì docker swarm không dùng tùy chọn này). Soạn file `/etc/docker/daemon.json` với nội dung dưới

```
cat <<EOF> /etc/docker/daemon.json
{
    "live-restore": false
}
EOF
```

- Khởi động lại docker engine

```
systemctl restart docker
```

Cấu hình docker swarm

- Đứng trên node master thực hiện lệnh dưới để thiết lập docker swarm `docker swarm init --advertise-addr eth1`

- Trong đó:

- Nếu có nhiều NICs thì cần chỉ định thêm tùy chọn `--advertise-addr` để chỉ ra tên của interfaces mà docker swarm sẽ dùng, các node worker sẽ dùng IP này để join vào cluster.
- Kết quả của lệnh trên như bên dưới, lưu ý dòng thông báo trong kết quả nhé. Dòng này để sử dụng trên các node worker. `[root@masternode ~]# docker swarm init --advertise-addr eth1`
`Swarm initialized: current node (yio0waboc34i8xqh86zt2rdat) is now a manager.`

To add a worker to this swarm, run the following command:

```
docker swarm join --token
SWMTKN-1-1w88l3qp3q5312lvf6nvkscuky03f99iwd4u6wighokzy4xomf-lio10gmxni8pmdajsoflgqy41
172.16.68.221:2377
```

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

```
[root@masternode ~]#
```

- Đứng trên node `worker1` và `worker2` để join vào node master. Thực hiện cả trên 02 node worker còn lại. `docker swarm join --token`

```
SWMTKN-1-1w88l3qp3q5312lvf6nvkscuky03f99iwd4u6wighokzy4xomf-lio10gmxni8pmdajsoflgqy41
172.16.68.221:2377
```

- Kết quả sẽ có thông báo như sau: `[root@worker1node ~]# docker swarm join --token SWMTKN-1-1w88l3qp3q5312lvf6nvkscuky03f99iwd4u6wighokzy4xomf-lio10gmxni8pmdajsoflgqy41 172.16.68.221:2377`
`This node joined a swarm as a worker.`

- Đứng trên node master thực hiện lệnh `docker node ls` để kiểm tra xem các node worker đã join hay chưa. Nếu chưa ổn thì kiểm tra kỹ lại các bước ở trên. `docker node ls`
 - Kết quả
- ```
[root@masternode ~]# docker node ls
```

| ID                          | HOSTNAME    | STATUS | AVAILABILITY |
|-----------------------------|-------------|--------|--------------|
| MANAGER STATUS              |             |        |              |
| yio0waboc34i8xqh86zt2rdat * | masternode  | Ready  | Active       |
| Leader                      |             |        |              |
| oh19lt8vnic067jep1ro8nw3r   | worker1node | Ready  | Active       |
| kesl0xhuj9w6yl84og4x2sd3y   | worker2node | Ready  | Active       |

```
[root@masternode ~]#
```

## Kiểm tra hoạt động của cụm docker swarm vừa dựng.

- Tạo file `Dockerfile` với nội dung bên dưới trên tất cả các node. Ở đây tôi sẽ tạo ra một images chạy web server, lưu ý, tôi sẽ tạo các nội dung các web server khác nhau với 03 node để khi kiểm tra sẽ thấy các kết quả của container trên từng node.
- Trên node master `cat <<EOF> /root/Dockerfile`

```
FROM centos
MAINTAINER hocchudong <admin@hocchudong.com>
RUN yum -y install httpd
RUN echo "Node master Hello DockerFile" > /var/www/html/index.html
EXPOSE 80
CMD ["-D", "FOREGROUND"]
ENTRYPOINT ["/usr/sbin/httpd"]
EOF
```
- Trên node worker1 `cat <<EOF> /root/Dockerfile`

```
FROM centos
MAINTAINER hocchudong <admin@hocchudong.com>
RUN yum -y install httpd
RUN echo "Node worker1 Hello DockerFile" > /var/www/html/index.html
EXPOSE 80
CMD ["-D", "FOREGROUND"]
ENTRYPOINT ["/usr/sbin/httpd"]
EOF
```
- Trên node worker2 `cat <<EOF> /root/Dockerfile`

```
FROM centos
MAINTAINER hocchudong <admin@hocchudong.com>
RUN yum -y install httpd
RUN echo "Node worker2 Hello DockerFile" > /var/www/html/index.html
EXPOSE 80
CMD ["-D", "FOREGROUND"]
```

```
ENTRYPOINT ["/usr/sbin/httpd"]
```

```
EOF
```

- Thực hiện build image với dockerfile vừa tạo ở trên trên cả 03 node (lưu ý dấu . nhé, lúc này đang đứng tại thư mục root) `docker build -t web_server:latest .`
- Kiểm tra images sau khi build xong dockerfile ở trên `docker images`

- Kết quả:

- Complete!

```
Removing intermediate container ac34ac2bf2bf
```

```
---> 1f52eba3ee41
```

```
Step 4/7 : RUN echo "Hello DockerFile" > /var/www/html/index.html
```

```
---> Running in 1100a7cddd06
```

```
Removing intermediate container 1100a7cddd06
```

```
---> cdd86cafcdc8
```

```
Step 5/7 : EXPOSE 80
```

```
---> Running in 262d31a60118
```

```
Removing intermediate container 262d31a60118
```

```
---> d0ecbae79e34
```

```
Step 6/7 : CMD ["-D", "FOREGROUND"]
```

```
---> Running in e9af0ad1d386
```

```
Removing intermediate container e9af0ad1d386
```

```
---> 5a9e18361f4d
```

```
Step 7/7 : ENTRYPOINT ["/usr/sbin/httpd"]
```

```
---> Running in c95a17b69cb6
```

```
Removing intermediate container c95a17b69cb6
```

```
---> bbc76a4873a4
```

```
Successfully built bbc76a4873a4
```

```
Successfully tagged web_server:latest
```

- Tạo container từ image ở trên với số lượng bản sao là 03. Lúc này đứng trên node master thực hiện các lệnh dưới. `docker service create --name swarm_cluster --replicas=3 -p 80:80`

```
web_server:latest
```

- Kết quả như sau:

- [root@masternode ~]# `docker service create --name swarm_cluster --replicas=3 -p 80:80`

```
web_server:latest
```

```
image web_server:latest could not be accessed on a registry to record
its digest. Each node will access web_server:latest independently,
possibly leading to different nodes running different
versions of the image.
```

```
uccie2ympqo426qhmh63tnfc9
```

```
overall progress: 3 out of 3 tasks
```

```
1/3: running [=====>]
```

```
2/3: running [=====>]
```

```
3/3: running [=====>]
```

verify: Service converged

- Kiểm tra lại kết quả bằng lệnh `docker service ls` [root@masternode ~]# `docker service ls`

```
ID NAME MODE REPLICAS IMAGE
PORTS
uccie2ympqo4 swarm_cluster replicated 3/3
web_server:latest *:80->80/tcp
[root@masternode ~]#
```

- Kiểm tra sâu hơn bên trong của cluster `docker service inspect swarm_cluster --pretty`

- Kết quả

- [root@masternode ~]# `docker service inspect swarm_cluster --pretty`  
[root@masternode ~]# `docker service inspect swarm_cluster --pretty`

```
ID: uccie2ympqo426qhmh63tnfc9
Name: swarm_cluster
Service Mode: Replicated
 Replicas: 3
Placement:
UpdateConfig:
 Parallelism: 1
 On failure: pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Update order: stop-first
RollbackConfig:
 Parallelism: 1
 On failure: pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Rollback order: stop-first
ContainerSpec:
 Image: web_server:latest
Resources:
Endpoint Mode: vip
Ports:
 PublishedPort = 80
 Protocol = tcp
 TargetPort = 80
 PublishMode = ingress
[root@masternode ~]#
```

- Kiểm tra trạng thái của các service bằng lệnh `docker service ps swarm_cluster`, các container sẽ nằm đều trên các node, kết quả như bên dưới. [root@masternode ~]# `docker service ps swarm_cluster`

| ID            | NAME                       | IMAGE             | NODE        |
|---------------|----------------------------|-------------------|-------------|
| DESIRED STATE | CURRENT STATE              | ERROR             | PORTS       |
| op3x52jzqmir  | swarm_cluster.1            | web_server:latest | worker1node |
| Running       | Running about a minute ago |                   |             |
| trm5frr8hqaw  | swarm_cluster.2            | web_server:latest | worker2node |
| Running       | Running about a minute ago |                   |             |
| 6tf95kywbjgm  | swarm_cluster.3            | web_server:latest | masternode  |
| Running       | Running about a minute ago |                   |             |

[root@masternode ~]#

- Có thể kiểm tra các container trên từng node bằng lệnh `docker ps`, kết quả là thực hiện kiểm tra trên worker1 và worker2 [root@worker1node ~]# `docker ps`

| CONTAINER ID      | IMAGE             | COMMAND                                   | CREATED            |
|-------------------|-------------------|-------------------------------------------|--------------------|
| STATUS            | PORTS             | NAMES                                     |                    |
| a136bedc42b7      | web_server:latest | "/usr/sbin/httpd -D ..."                  | About a minute ago |
| Up About a minute | 80/tcp            | swarm_cluster.1.op3x52jzqmir3o0dawz7hn4tn |                    |

[root@worker2node ~]# `docker ps`

| CONTAINER ID | IMAGE             | COMMAND                                   | CREATED       |
|--------------|-------------------|-------------------------------------------|---------------|
| STATUS       | PORTS             | NAMES                                     |               |
| 143f360db9ca | web_server:latest | "/usr/sbin/httpd -D ..."                  | 3 minutes ago |
| Up 3 minutes | 80/tcp            | swarm_cluster.2.trm5frr8hqawlz14m18fm3g3c |               |

- Thử thay đổi số lượng container bằng lệnh. Lúc này container sẽ được tăng lên. `docker service scale swarm_cluster=4`
  - Kiểm tra lại bằng lệnh `docker service ps swarm_cluster`
  - Khi tắt thử các container trên một trong các node, sẽ có container mới được sinh ra để đảm bảo số container đúng với thiết lập.

Cài đặt Docker Swarm trên Ubuntu Server 16.04 64

## Hướng dẫn cài đặt docker swarm trên Ubuntu Server 16.04 64 bit

### Chuẩn bị

- 03 máy server



# Các bước cài đặt

## Đặt IP cho từng máy

- Đặt IP cho máy master `cicd1`. Node này đóng vai trò master `cat << EOF >`

```
/etc/network/interfaces
```

```
This file describes the network interfaces available on your system
and how to activate them. For more information, see interfaces(5).
```

```
source /etc/network/interfaces.d/*
```

```
The loopback network interface
```

```
auto lo
```

```
iface lo inet loopback
```

```
The primary network interface
```

```
auto ens4
```

```
iface ens4 inet static
```

```
address 172.16.68.152
```

```
netmask 255.255.255.0
```

```
gateway 172.16.68.1
```

```
dns-nameservers 8.8.8.8
```

```
EOF
```

- Đặt IP cho máy master `cicd2`. Node này đóng vai trò worker `cat << EOF >`

```
/etc/network/interfaces
```

```
This file describes the network interfaces available on your system
```

```
and how to activate them. For more information, see interfaces(5).
```

```
source /etc/network/interfaces.d/*
```

```
The loopback network interface
```

```
auto lo
```

```
iface lo inet loopback
```

```
The primary network interface
```

```
auto ens4
```

```
iface ens4 inet static
```

```
address 172.16.68.153
```

```
netmask 255.255.255.0
gateway 172.16.68.1
dns-nameservers 8.8.8.8
EOF
```

- Đặt IP cho máy master `cicd3`. Node này đóng vai trò worker `cat << EOF >`

```
/etc/network/interfaces
```

```
This file describes the network interfaces available on your system
and how to activate them. For more information, see interfaces(5).
```

```
source /etc/network/interfaces.d/*
```

```
The loopback network interface
```

```
auto lo
```

```
iface lo inet loopback
```

```
The primary network interface
```

```
auto ens4
```

```
iface ens4 inet static
```

```
address 172.16.68.154
```

```
netmask 255.255.255.0
```

```
gateway 172.16.68.1
```

```
dns-nameservers 8.8.8.8
```

```
EOF
```

## Cài đặt các thành phần của docker

Lưu ý: cài lên tất cả các node

### Cài đặt docker engine

- Cài đặt docker engine `su -`

```
curl -sSL https://get.docker.com/ | sudo sh
```

- Phân quyền `sudo usermod -aG docker `whoami``

- Khởi động lại service `systemctl start docker.service`

```
systemctl enable docker.service
```

- Kiểm tra trạng thái của docker `systemctl status docker.service`

- Kiểm tra lại phiên bản của docker `docker version`

## Cài đặt docker compose

Lưu ý: cài lên tất cả các node

- Cài đặt docker compose `sudo curl -L https://github.com/docker/compose/releases/download/1.18.0/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose`  
`sudo chmod +x /usr/local/bin/docker-compose`
- Kiểm tra phiên bản của docker-compose `docker-compose --version`

## Thực hiện cài đặt docker swarm\_cluster

### Thiết lập docker swarm

- Đứng trên node có vai trò là master (manager) để thiết lập swarm, trong ví dụ này là node có tên `cicd1`

```
docker swarm init --advertise-addr 172.16.68.152
```

- Trong đó:
  - 172.16.68.152: là IP của node master (chính là node `cicd1`)

Sau khi chạy lệnh trên, màn hình sẽ trả về kết quả và hướng dẫn các thao tác để join các node còn lại vào cụm docker swarm. Sử dụng lệnh trên màn hình trả về để thao tác trên 02 node `cicd2` và `cicd3` còn lại.

Join các node `cicd2` và `cicd3` vào cụm cluster.

- Đứng trên các node `cicd2` `docker swarm join --token SWMTKN-1-3ibshsl050pg7op2i6ychyeoiqlv4qnvqcdvrpfiis9tiw5qb1-3lheft3zx76q68oj3n3dp20kn 172.16.68.152:2377`
- Đứng trên các node `cicd3` `docker swarm join --token SWMTKN-1-3ibshsl050pg7op2i6ychyeoiqlv4qnvqcdvrpfiis9tiw5qb1-3lheft3zx76q68oj3n3dp20kn 172.16.68.152:2377`
- Kết quả trả về như dưới là ok: `This node joined a swarm as a worker.`
- Quay trở lại node master và thực hiện lệnh dưới để kiểm tra xem các node đã join vào cluster hay chưa `docker node ls`

- Kết quả trả về như dưới là thành công

| ID                           | HOSTNAME | STATUS | AVAILABILITY |
|------------------------------|----------|--------|--------------|
| MANAGER STATUS               |          |        |              |
| paicdiqjmlyxv2jrr0wqg5vcqq * | cicd1    | Ready  | Active       |
| Leader                       |          |        |              |
| ep01sgz4qsasncztzc7hmmwzss   | cicd2    | Ready  | Active       |
| olovbr7afm8bc24sl95rwdika    | cicd3    | Ready  | Active       |
| root@cicd1:~#                |          |        |              |
| root@cicd1:~#                |          |        |              |

## Kiểm tra hoạt động của docker swarm cluster.

Sau khi cài đặt docker swarm xong, chúng ta cần kiểm tra hoạt động cơ bản của chúng. Các bước thực hiện này sẽ làm tại node master. Có các thao tác kiểm tra như sau:

- Kiểm tra xem các service đang hoạt động trong cụm cluster docker swarm, kết quả trả về sẽ thông tin các service trong cụm cluster. Trong lần cài đặt đầu tiên, chúng ta sẽ không thấy service nào hoạt động cả. `docker service ls`

- Tạo service chạy web nginx với số lượng nhân bản là 02. `docker service create --name my-web --publish 8080:80 --replicas 2 nginx`

- Kết quả của lệnh trên như sau:

```

zr9md8qlu5ynie3wlez3qe2uy
overall progress: 2 out of 2 tasks
1/2: running [=====>]
2/2: running [=====>]
verify: Service converged

```

- Trong lệnh trên, ta đứng trên node master và ra lệnh cho cụm cluster docker swarn tạo ra 02 container chạy dịch vụ là nginx và phân phối chúng lên các node slave. Số bản replicas này mục tiêu là để dự phòng cho service nginx.

- Tham số `--publish 8080:80` có nghĩa là ta sẽ sử dụng port 8080 để truy cập vào web, docker sẽ forward vào port 80 bên trong container. `http://172.16.68.152:8080`

`http://172.16.68.153:8080`

`http://172.16.68.154:8080`

- Tới bước này ta có thể sử dụng IP của các máy `cicd1` hoặc `cicd2` hoặc `cicd3` với port 8080 để truy cập vào container vừa được tạo ở trên.

- Ta có thể kiểm tra lại service bằng lệnh `docker service ls`

- Kết quả như sau

```

root@cicd1:~# docker service ls

```

| ID           | NAME   | MODE       | REPLICAS | IMAGE |
|--------------|--------|------------|----------|-------|
| PORTS        |        |            |          |       |
| zr9md8qlu5yn | my-web | replicated | 2/2      |       |

```
nginx:latest *:8080->80/tcp
```

```
root@cicd1:~#
```

- **Sử dụng lệnh** `docker service ps <ten_service>` để kiểm tra service đang chạy. `docker service ps my-web`

- Kết quả như dưới (quan sát kết quả để biết thêm các tham số)

- `root@cicd1:~# docker service ps my-web`

| ID            | NAME                  | IMAGE        | NODE  |
|---------------|-----------------------|--------------|-------|
| DESIRED STATE | CURRENT STATE         | ERROR        | PO    |
| RTS           |                       |              |       |
| uo4o9xdi3y3z  | my-web.1              | nginx:latest | cicd2 |
| Running       | Running 7 minutes ago |              |       |
| dqh0e6w2y7kz  | my-web.2              | nginx:latest | cicd1 |
| Running       | Running 7 minutes ago |              |       |