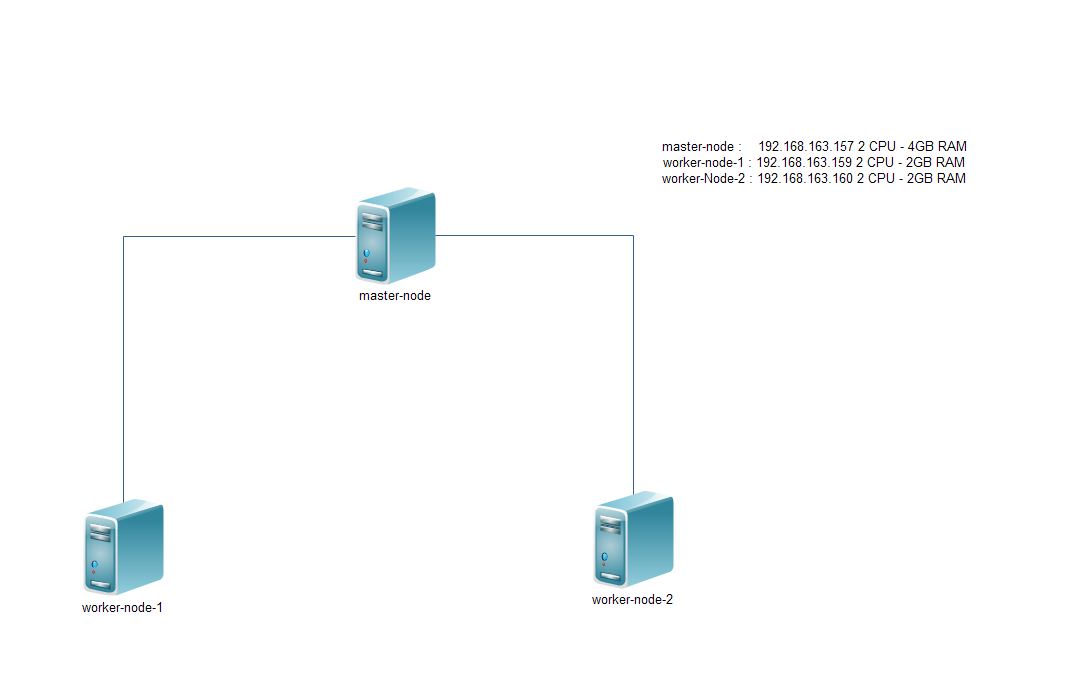
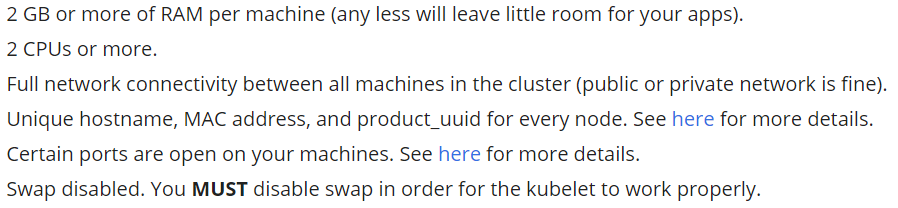
**Mô hình hệ thống**

****

**Các Yêu Cầu Cơ Bản**

****

**https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/**

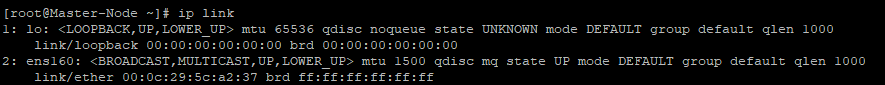
**Kiểm tra yêu cầu**

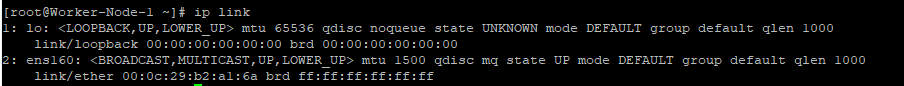
1. **Kiểm tra MAC Address**

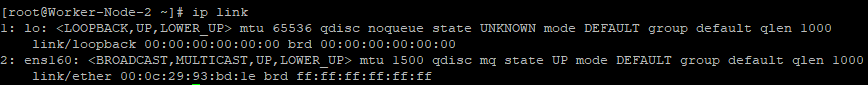
Đảm bảo MAC Address của các card mạng giữa các máy chủ node là độc lập và duy nhất. Điều này dễ xảy ra khi MAC Address trùng lặp khi chúng ta sử dụng VM được clone ra

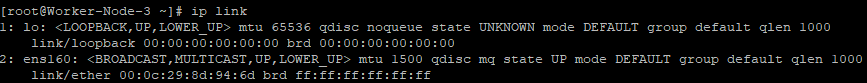
Ta sử dụng câu lệnh sau

**ip link**

****

****

****

****

1. **Kiểm tra product\_uuid**

Đảm bảo product\_uuid giữa các máy chủ node là độc lập và duy nhất

Ta sử dụng câu lệnh:

**cat /sys/class/dmi/id/product\_uuid**









1. **Đặt Hostname cho các máy**

Sử dụng câu lệnh sau để đặt hostname tương ứng với sơ đồ

**hostnamectl set-hostname <<tên máy chủ >>**

Sử dụng câu lệnh sau để xem tên máy chủ

**hostname**

****

****

****

1. **Cấu hình lại File host ( Thực hiện trên cả Master và Worker )**

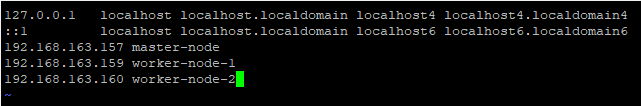
Sử dụng câu lệnh sau để vào cấu hình của host và thêm vào các dòng sau

**vi /etc/hosts**

**192.168.163.157 master-node**

**192.168.163.159 worker-node-1**

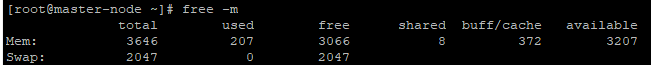
**192.168.163.160 worker-node-2**

****

1. **Tắt bộ nhớ Swap trong hệ thống ( Thực hiện trên cả Master và Worker )**

Sử dụng câu lệnh sau để check xem Swap có được dùng không

**free -m**

****

Tắt Swap tạm thời bằng câu lệnh sau

**swapoff -a**

Tắt vĩnh viễn bộ nhớ Swap bằng câu lệnh sau để vào File cấu hình và thêm comment vào dòng như hình

**vi /etc/fstab**

****

1. **Tắt Selinux ( Thực hiện trên cả Master và Worker )**

Sử dụng câu lệnh sau để vô hiệu hoá Selinux tạm thời

**setenforce 0**

Dùng câu lệnh sau để xem trạng thái Selinux

**getenforce**

****

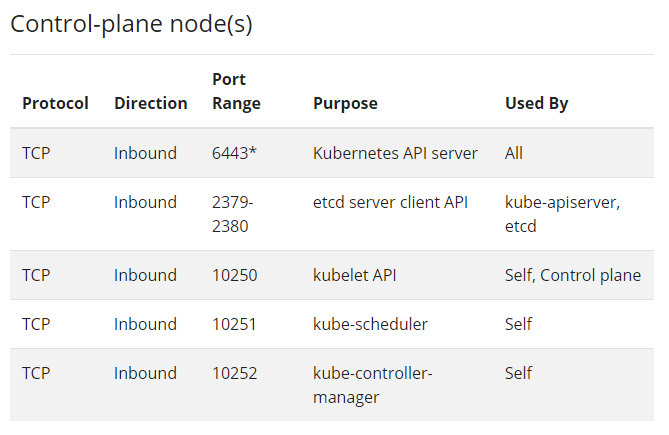
Sử dụng câu sau để thay đổi enforcing thành disabled

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

Sau đó reset lại máy

**reboot**

1. **Mở Port cần thiết cho dịch vụ kubernetess**



master-node: Thêm các Port sau 6443,2379-2380,10250,10251,10252

Sử dụng câu lệnh sau để thêm các Port cần thiết vào master-node

**firewall-cmd --add-masquerade --permanent**

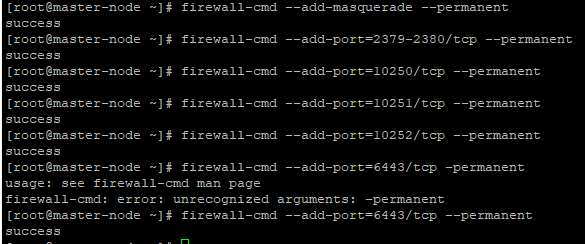
**firewall-cmd --add-port=2379-2380/tcp --permanent**

**firewall-cmd --add-port=10250/tcp --permanent**

**firewall-cmd --add-port=10251/tcp --permanent**

**firewall-cmd --add-port=10252/tcp --permanent**

**firewall-cmd --add-port=6443/tcp --permanent**

****

Khởi động lại dịch vụ firewalld

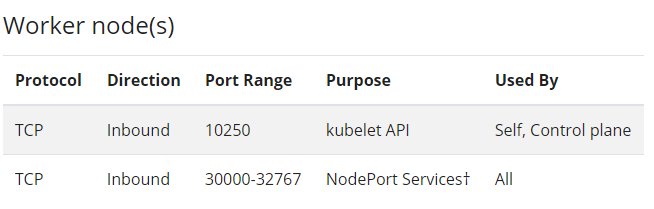
**firewall-cmd --reload**

****

Sử dụng câu lệnh sau để xem các Port đã được thêm vào

**firewall-cmd --list-ports**





worker-node: Thêm các Port sau 10250,30000-32767

Sử dụng câu lệnh sau để thêm các Port cần thiết vào worker-node

**firewall-cmd --add-masquerade --permanent**

**firewall-cmd --add-port=30000-32767/tcp --permanent**

**firewall-cmd --add-port=10250/tcp --permanent**

Khởi động lại dịch vụ firewalld

**firewall-cmd --reload**

Sử dụng câu lệnh sau để xem các Port đã được thêm vào

**firewall-cmd --list-ports**

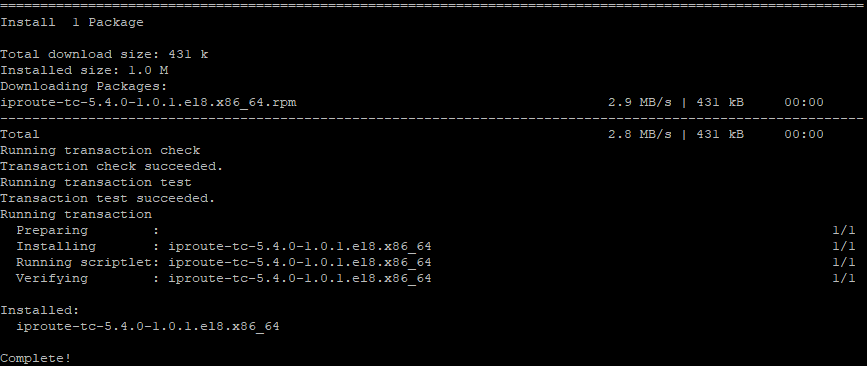




1. **Cài đặt iproute**

Sử dụng câu lệnh sau để cài đặt iproute

**dnf install -y iproute-tc**

****

1. **Cấu hình Mạng ( Thực hiện trên cả Master và Worker )**

br\_netfilter : module này cho phép giả mạo kênh truyền để cho lưu lượng mạng LAN có thể giao tiếp giữa các pod của kubernetes

Sử dụng câu lệnh sau để xem br\_netfilter có hoạt động không

**lsmod | grep br\_netfilter**



Sử dụng câu lệnh sau để kích hoạt br\_netfilter

**modprobe br\_netfilter**

Kiểm tra lại xem module br\_netfilter đã hoạt động chưa bằng câu lệnh sau

**lsmod | grep br\_netfilter**

****

Tiến hành kiểm tra xem chế độ cho phép lưu lượng bridge có được di chuyển qua iptables không bằng câu lệnh sau

**sysctl -a | grep net.bridge.bridge-nf-call-iptables**



Nếu giá trị là 1 như trên là lưu lượng bridge được phép di chuyển qua iptables

Sử dụng câu lệnh sau để tạo File cài đặt kubernetes (k8s.conf) trong sysctl

**cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf**

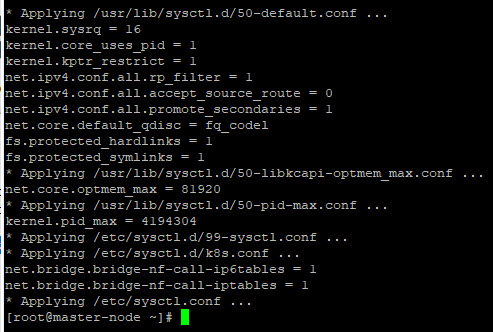
**net.bridge.bridge-nf-call-ip6tables = 1**

**net.bridge.bridge-nf-call-iptables = 1**

**EOF**

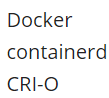
Sử dụng câu lệnh sau để tải hệ thống từ tất cả các tệp cấu hình hệ thống

**sudo sysctl --system**

****

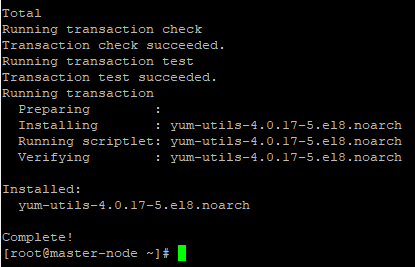
1. **Cài đặt dịch vụ Docker sử dụng repository trên cả Master-Node và Worker-Node**

Theo như ta đã biết thì kubernetes cần có container để quản lí có rất nhiều dịch vụ để tạo ra container nhưng trong bài lab này . Mình chỉ dùng Docker vì sự phổ biến rộng rãi của nó



Cài đặt 1 số thư viện cần thiết cho docker bằng câu lệnh sau

**dnf install -y yum-utils device-mapper-persistent-data lvm2**

****

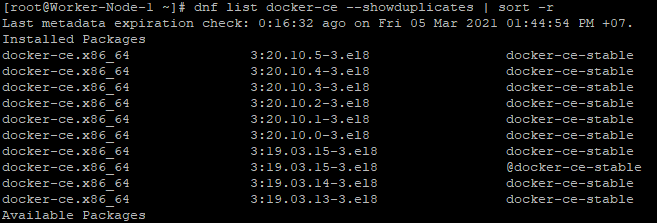
Tiến hành thêm kho lưu trữ của Docker vào vì nó không có trong danh sách các gói mặc định bằng cách sử dụng câu lệnh sau:

**yum-config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo**

****

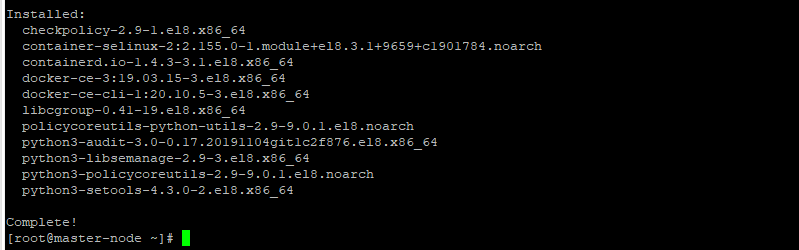
Dùng câu lệnh sau để xem phiên bản của docker

**dnf list docker-ce --showduplicates | sort -r**

****

Tiến hành cài đặt Docker bằng câu lệnh sau

**dnf -y install docker-ce-3:19.03.15-3.el8**

****

Tạo thư mục thư mục dịch vụ cho docker

**mkdir -p /etc/docker /etc/systemd/system/docker.service.d**

Thêm cấu hình daemon.json vào hệ thống

**vi /etc/docker/daemon.json**

**{**

**"exec-opts": ["native.cgroupdriver=systemd"],**

**"log-driver": "json-file",**

**"log-opts": {**

**"max-size": "100m"**

**},**

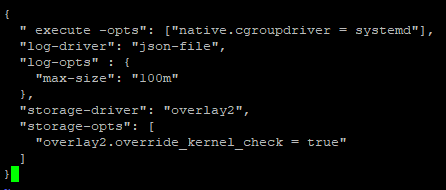
**"storage-driver": "overlay2",**

**"storage-opts": [**

**"overlay2.override\_kernel\_check=true"**

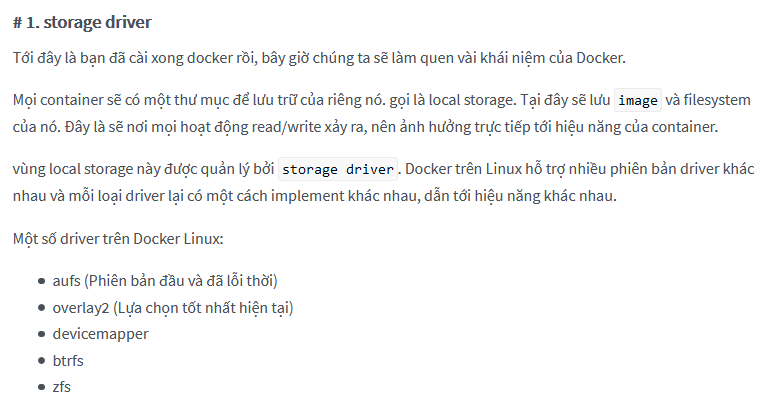
**]**

**}**

****

**systemd** là một bộ các trình quản lý hệ thống, thư viện và tiện ích được thiết kế như một nền tảng quản lý và cấu hình trung tâm cho hệ điều hành máy tính

Cấu hình trên mặc định thiết lập driver sử dụng xuất ra file log dạng json, kích thước tối đa của mỗi file là 100m. Bạn chỉ việc điều chỉnh cỡ theo nhu cầu và khởi động lại Docker



Khởi động lại dịch vụ doraemon

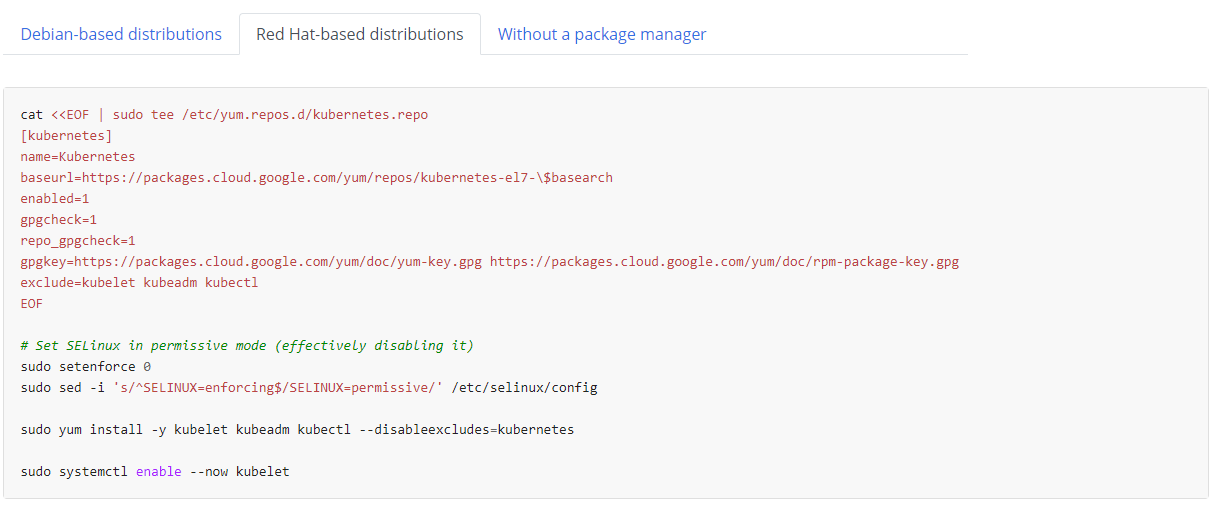
**systemctl daemon-reload**

Kích hoạt docker chạy cùng với hệ thống khi khởi động và start docker

**systemctl enable docker --now**

**systemctl start docker**

1. **Cài đặt dịch bụ Kubernetes ( Thực hiện trên cả Master và Worker )**

****

**kubeadm:** lệnh khởi động cụm kubernetes

**kubelet :** thành phần chạy trên tất cả các máy worker và thực hiện nhưng việc như khởi động pods và containers

**kubectl:** dòng lệnh sử dụng để nói chuyện với các máy worker

Tạo một tệp kho cho kubernetes bằng câu lệnh sau

**cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo**

**[kubernetes]**

**name=Kubernetes**

**baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-\$basearch**

**enabled=1**

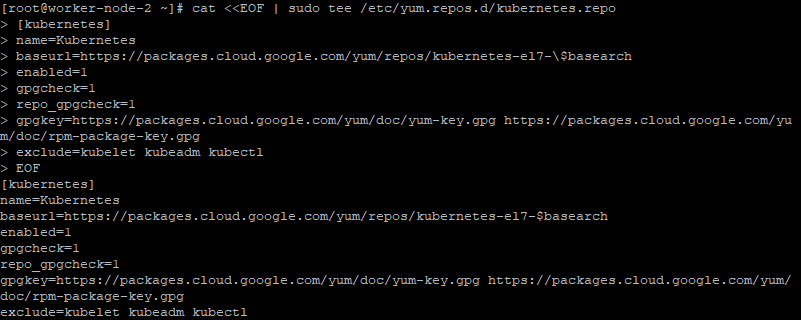
**gpgcheck=1**

**repo\_gpgcheck=1**

**gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg**

**exclude=kubelet kubeadm kubectl**

**EOF**

****

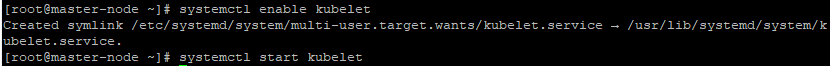
Tiến hành cài đặt Kubernetes bằng câu lệnh sau:

**dnf install -y kubelet kubeadm kubectl --disableexcludes=kubernetes**

Kích hoạt kubernetes chạy cùng với hệ thống khi khởi động và start kubernetes

**systemctl enable kubelet**

**systemctl start kubelet**

****

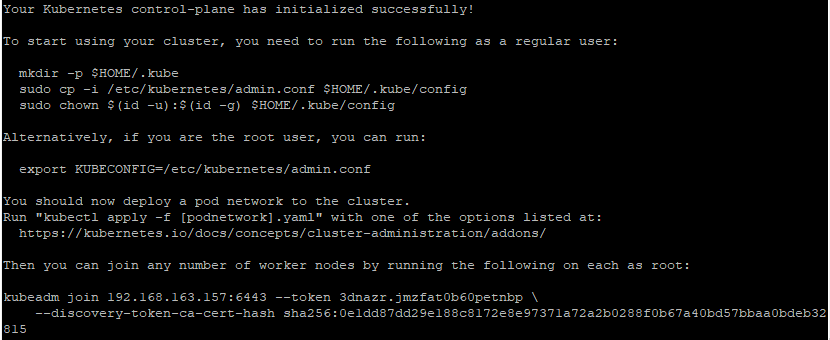
1. **Khởi tạo Kubernetes Cluster ( Chỉ thực hiện trên Master )**

Sử dụng câu lệnh sau để cấu hình kubernetes

**kubeadm init --apiserver-advertise-address 192.168.163.157 --pod-network-cidr=172.16.0.0/16**

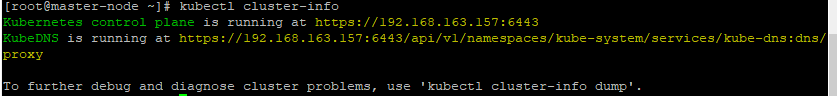
**apiserver-advertise-address :** là địa chỉ của Master-Node, địa chỉ này dùng để truyền thông đến các node còn lại

**pod-network-cidr :** chính là các IP sẽ cấp cho các pod để kết nối với nhau, do mình sẽ sử dụng Calico nên thường khai báo với dải 172.16.0.0/16

****

Sử dụng lệnh sau đểm kiểm tra thông tin của cụm kubernetes

**kubectl cluster-info**

****

Sử dụng lệnh sau để xem các Node trong cụm kubernetes

**kubectl get nodes**

****

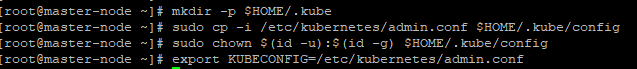
1. **Cấu hình file config để chương trình kubectl có thể tương tác Kubernetes API (Chỉ Trên Master)**

**mkdir -p $HOME/.kube**

**sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config**

**sudo chown $(id -u):$(id -g) $HOME/.kube/config**

**export KUBECONFIG=/etc/kubernetes/admin.conf**

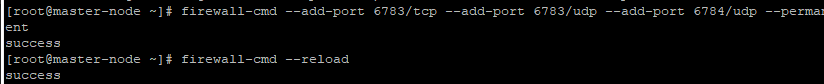
****

1. **Thêm 1 số port cần thiết cho tiện ích mạng weave-net ( cho cả Master và Worker)**

Thêm 1 số port cần thiết bằng câu lệnh sau

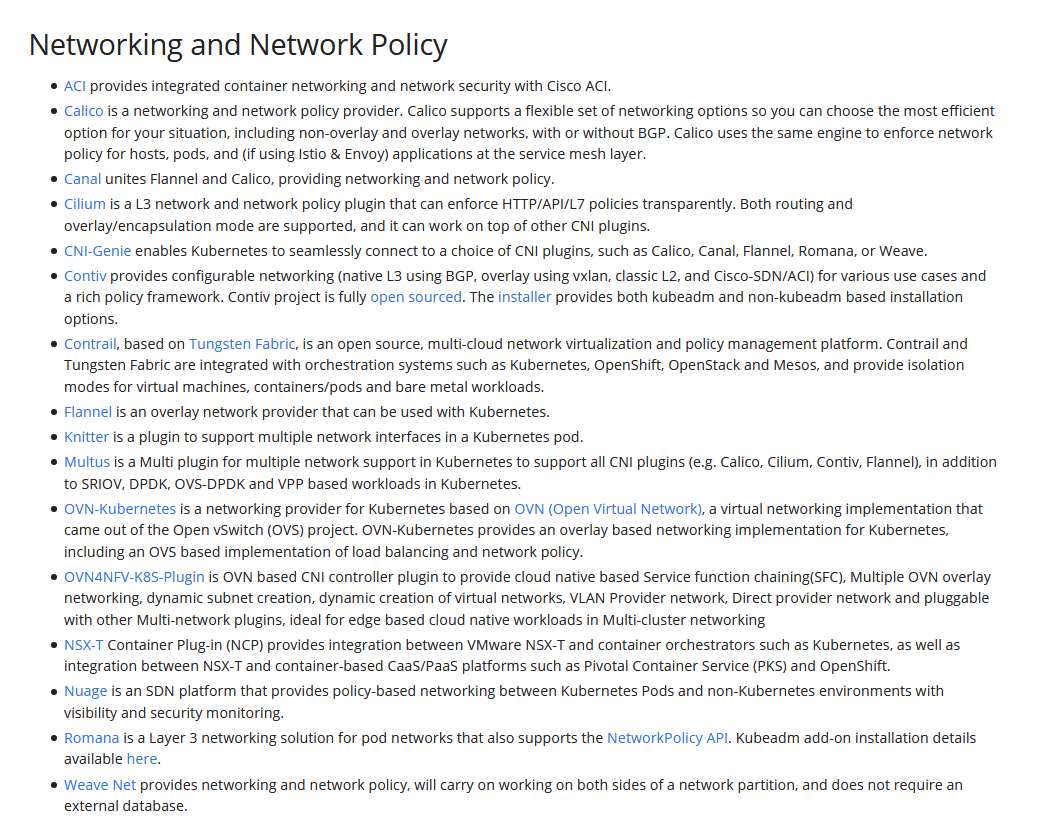
**firewall-cmd --add-port 6783/tcp --add-port 6783/udp --add-port 6784/udp --permanent**

**firewall-cmd --reload**

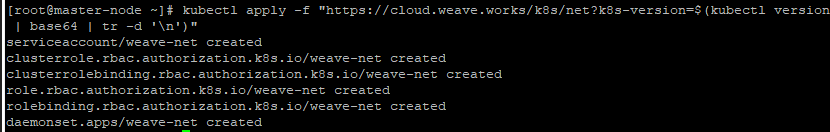
****

1. **Cài đặt Pod Network (Chỉ Trên Master)**

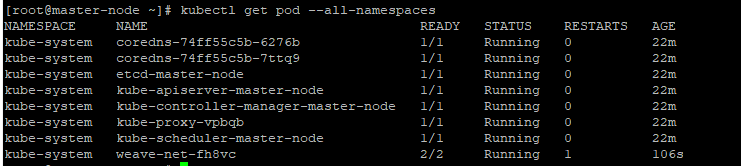
Network : Không có network riêng mà phải sử dụng network ảo từ các nguồn khác



Như chúng ta đã nói ở trên thì Kubernetes sử dụng các Network plugin để tạo mạng ảo, có rất nhiều lựa chọn ở đây mình chọn **weave-net**Chúng ta sử dụng câu lệnh sau

**kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"** ****

**kubectl get pod --all-namespaces**

****

Sau khi tiến hành xong ta sử dụng câu lệnh sau để kiểm tra Master-Node đã hoạt động chưa và đã nằm trong control-plane chưa

**kubectl get nodes**

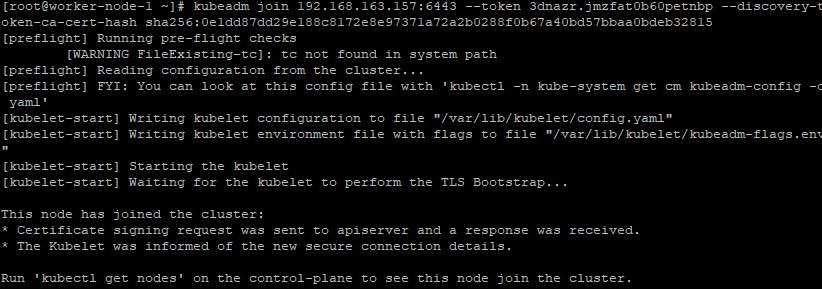
****

1. **Tiến hành tham gia tất cả các Worker-Node vào cụm Kubernetes ( Chỉ Dành Cho Worker-Node)**

Lúc nãy khi cấu hình kubernetes trên master ta có được 1 chuỗi như sau dùng để xác thực kết nối đến cụm kubernetes

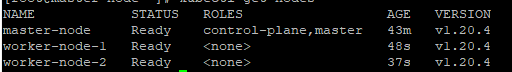
**kubeadm join 192.168.163.157:6443 --token 3dnazr.jmzfat0b60petnbp --discovery-token-ca-cert-hash sha256:0e1dd87dd29e188c8172e8e97371a72a2b0288f0b67a40bd57bbaa0bdeb32815**

Tiến hành tham gia vào cụm kubernetes từ chuỗi xác thực được cung cấp



Tiến hành kiểm tra xem các Worker-Node đã kết nối đến cụm Kubernetes thành công chưa sử dụng câu lệnh sau trên Master-Node

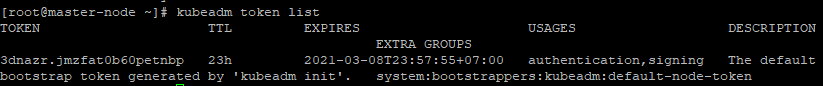
**kubectl get nodes**

****

Ta thấy rằng 3 Worker-Node đã tham gia thành công vào cụm Kubernetes

Ta có thể xem thời gian sống của token được tạo ra ở trên bằng câu lệnh sau:

**kubeadm token list**

****

Giả sử nếu như token hết hạn ta có thể sử dụng câu lệnh sau để tạo token mới

**kubeadm token create**

****

Nếu chúng ta không tìm thấy giá trị “–discovery-token-ca-cert-hash” , thì chúng ta có thể chạy lệnh sau trên Control-plane Node để lấy thông tin CA Cert Hash

**openssl x509 -pubkey -in /etc/kubernetes/pki/ca.crt | openssl rsa -pubin -outform der 2>/dev/null | openssl dgst -sha256 -hex | sed 's/^.\* //'**

****

Nếu ta thấy nhưng điều trên quá dài dòng ta có thể sử dụng câu lệnh sau

**kubeadm token create --print-join-command**

****

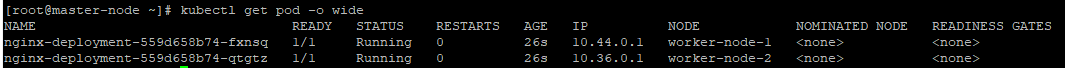
1. **Test tạo pod thử dịch vụ**

****

**kubectl apply -f https://k8s.io/examples/application/deployment-update.yaml**

****

**kubectl get pod -o wide**

****

Ngoài ra ta có thể sử dụng lệnh để dưới để xem các service nào đã sẵn sàng để deployment

**kubectl get deployment**

****

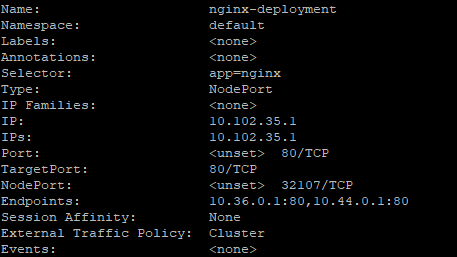
Tới bước này, chúng ta chưa thể truy cập vào các container được, cần thực hiện thêm bước deploy các container với các tùy chọn phù hợp, cụ thể như sau

**kubectl expose deploy nginx-deployment --port 80 --target-port 8080 --type NodePort**

****

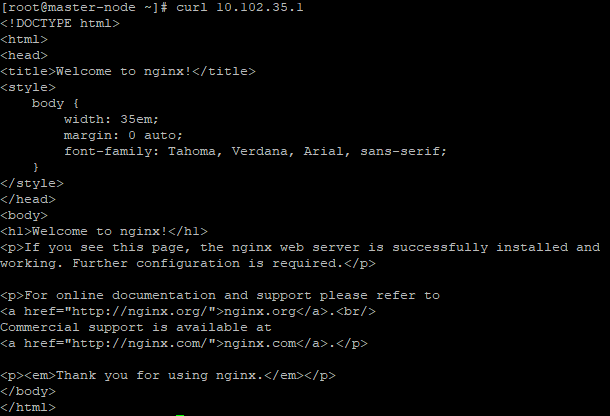
Sử dụng câu lệnh sau để xem chi tiết dịch vụ vừa được deploy

**kubectl describe service nginx-deployment**

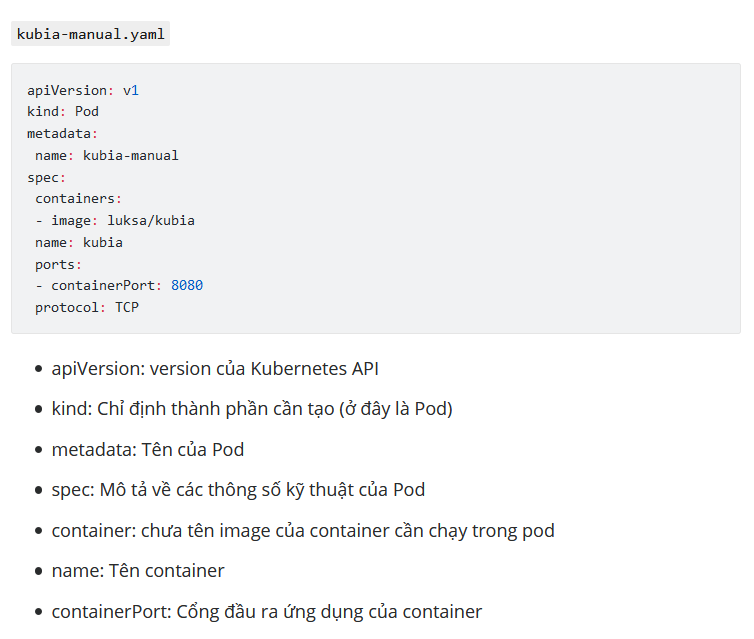
****

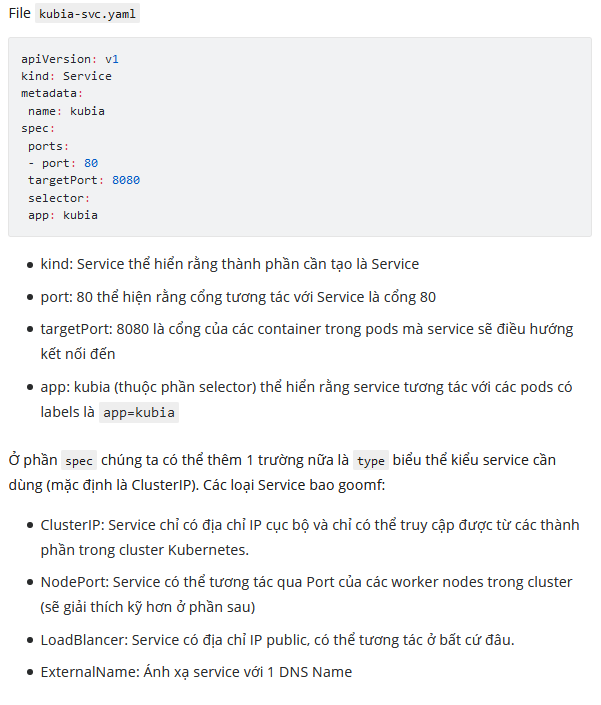
Test dịch vụ bằng câu lệnh sau

**curl 10.102.35.1**

****

**Giải thích 1 số thành phần trong file yaml**

****

**kubectl get svc**

apiVersion: v1

kind: Pod

metadata:

name: hello-httpd

spec:

containers:

- name: hello-httpd

image: httpd