/master/vagrantfile

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

# Tạo máy ảo từ box centos/7, gán địa chỉ IP, đặt hostname, gán 2GB bộ nhớ, 2 cpus

Vagrant.configure("2") do |config|

config.vm.box = "centos/7"

config.vm.network "private\_network", ip: "172.16.10.100"

config.vm.hostname = "master.xtl"

config.vm.provider "virtualbox" do |vb|

vb.name = "master.xtl"

vb.cpus = 2

vb.memory = "1848"

end

# Chạy file install-docker-kube.sh sau khi nạp Box

config.vm.provision "shell", path: "./../install-docker-kube.sh"

# Chạy các lệnh shell

config.vm.provision "shell", inline: <<-SHELL

# Đặt pass la a có tài khoản root và cho phép SSH

echo "a" | passwd --stdin root

sed -i 's/^PasswordAuthentication no/PasswordAuthentication yes/' /etc/ssh/sshd\_config

systemctl reload sshd

# Ghi nội dung sau ra file /etc/hosts để truy cập được các máy theo HOSTNAME

cat >>/etc/hosts<<EOF

172.16.10.100 master.xtl

172.16.10.101 worker1.xtl

172.16.10.102 worker2.xtl

EOF

SHELL

end

giải thích:

path: "./../install-docker-kube.sh"

khi chạy vagrant ta phải ở vị trí của file vagrantfile và chạy lệnh:

# vagrant up

Khi đó n sẽ tạo ra 1 file /.vagrant bên cạnh vagrantfile được tạo từ trước.

Và file install-docker-kube.sh phải để ngang hàng với vị trí của master

Worker1

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

Vagrant.configure("2") do |config|

config.vm.box = "centos"

config.vm.network "private\_network", ip: "172.16.10.101"

config.vm.hostname = "worker1.xtl"

config.vm.provider "virtualbox" do |vb|

vb.name = "worker1.xtl"

vb.cpus = 1

vb.memory = "1024"

end

config.vm.provision "shell", path: "./../install-docker-kube.sh"

config.vm.provision "shell", inline: <<-SHELL

echo "root password"

echo "a" | passwd --stdin root

sed -i 's/^PasswordAuthentication no/PasswordAuthentication yes/' /etc/ssh/sshd\_config

systemctl reload sshd

cat >>/etc/hosts<<EOF

172.16.10.100 master.xtl

172.16.10.101 worker1.xtl

172.16.10.102 worker2.xtl

EOF

SHELL

end

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

# Tạo máy ảo từ box centos/7, gán địa chỉ IP, đặt hostname, gán 2GB bộ nhớ, 2 cpus

Vagrant.configure("2") do |config|

config.vm.box = "centos/7"

config.vm.network "private\_network", ip: "172.16.10.100"

config.vm.hostname = "master.xtl"

config.vm.provider "virtualbox" do |vb|

vb.name = "master.xtl"

vb.cpus = 2

vb.memory = "2048"

end

# Chạy file install-docker-kube.sh sau khi nạp Box

config.vm.provision "shell", path: "./../install-docker-kube.sh"

# Chạy các lệnh shell

config.vm.provision "shell", inline: <<-SHELL

# Đặt pass 123 có tài khoản root và cho phép SSH

echo "123" | passwd --stdin root

sed -i 's/^PasswordAuthentication no/PasswordAuthentication yes/' /etc/ssh/sshd\_config

systemctl reload sshd

# Ghi nội dung sau ra file /etc/hosts để truy cập được các máy theo HOSTNAME

cat >>/etc/hosts<<EOF

172.16.10.100 master.xtl

172.16.10.101 worker1.xtl

172.16.10.102 worker2.xtl

EOF

SHELL

end

kubernetes-centos7/install-docker-kube.sh

#!/bin/bash

# Cập nhật 12/2019

# Cai dat Docker

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

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

yum update -y && yum install docker-ce-18.06.2.ce -y

usermod -aG docker $(whoami)

## Create /etc/docker directory.

mkdir /etc/docker

# Setup daemon.

cat > /etc/docker/daemon.json <<EOF

{

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

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

"log-opts": {

"max-size": "100m"

},

"storage-driver": "overlay2",

"storage-opts": [

"overlay2.override\_kernel\_check=true"

]

}

EOF

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

# Restart Docker

systemctl enable docker.service

systemctl daemon-reload

systemctl restart docker

# Tat SELinux

setenforce 0

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

# Tat Firewall

systemctl disable firewalld >/dev/null 2>&1

systemctl stop firewalld

# sysctl

cat >>/etc/sysctl.d/kubernetes.conf<<EOF

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

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

EOF

sysctl --system >/dev/null 2>&1

# Tat swap

sed -i '/swap/d' /etc/fstab

swapoff -a

# Add yum repo file for Kubernetes

cat >>/etc/yum.repos.d/kubernetes.repo<<EOF

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

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

EOF

yum install -y -q kubeadm kubelet kubectl

systemctl enable kubelet

systemctl start kubelet

# Configure NetworkManager before attempting to use Calico networking.

cat >>/etc/NetworkManager/conf.d/calico.conf<<EOF

[keyfile]

unmanaged-devices=interface-name:cali\*;interface-name:tunl\*

EOF

Thiết lập file chạy được

chmode +x install-docker-kube.sh

Tại thư mục kubernetes-centos7/master/ gõ lệnh vagrant để tạo máy master.xtl

vagrant up

Sau lệnh này, quá trình cài đặt diễn ra, kết thúc thì có máy ảo VirtualBox với tên master.xtl trong đó đã có Docker, kubelet đang chạy ở địa chỉ IP 172.16.10.100, hãy ssh vào máy này bằng lệnh ssh với tài khoản root có cấu hình pass là 123 ở trên.

**Khởi tạo Cluster**

Trong lệnh khởi tạo cluster có tham số --pod-network-cidr để chọn cấu hình mạng của POD, do dự định dùng Addon calico nên chọn --pod-network-cidr=192.168.0.0/16

Gõ lệnh sau để khở tạo là nút master của Cluster

kubeadm init --apiserver-advertise-address=172.16.10.100 --pod-network-cidr=192.168.0.0/16

Sau khi lệnh chạy xong, chạy tiếp cụm lệnh nó yêu cầu chạy sau khi khởi tạo- để chép file cấu hình đảm bảo trình kubectl trên máy này kết nối Cluster

mkdir -p $HOME/.kube

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

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

Tiếp đó, nó yêu cầu cài đặt một Plugin mạng trong các Plugin tại [addon](https://kubernetes.io/docs/concepts/cluster-administration/addons/), ở đây đã chọn calico, nên chạy lệnh sau để cài nó

kubectl apply -f https://docs.projectcalico.org/v3.10/manifests/calico.yaml

Gõ vài lệnh sau để kiểm tra

# Thông tin cluster

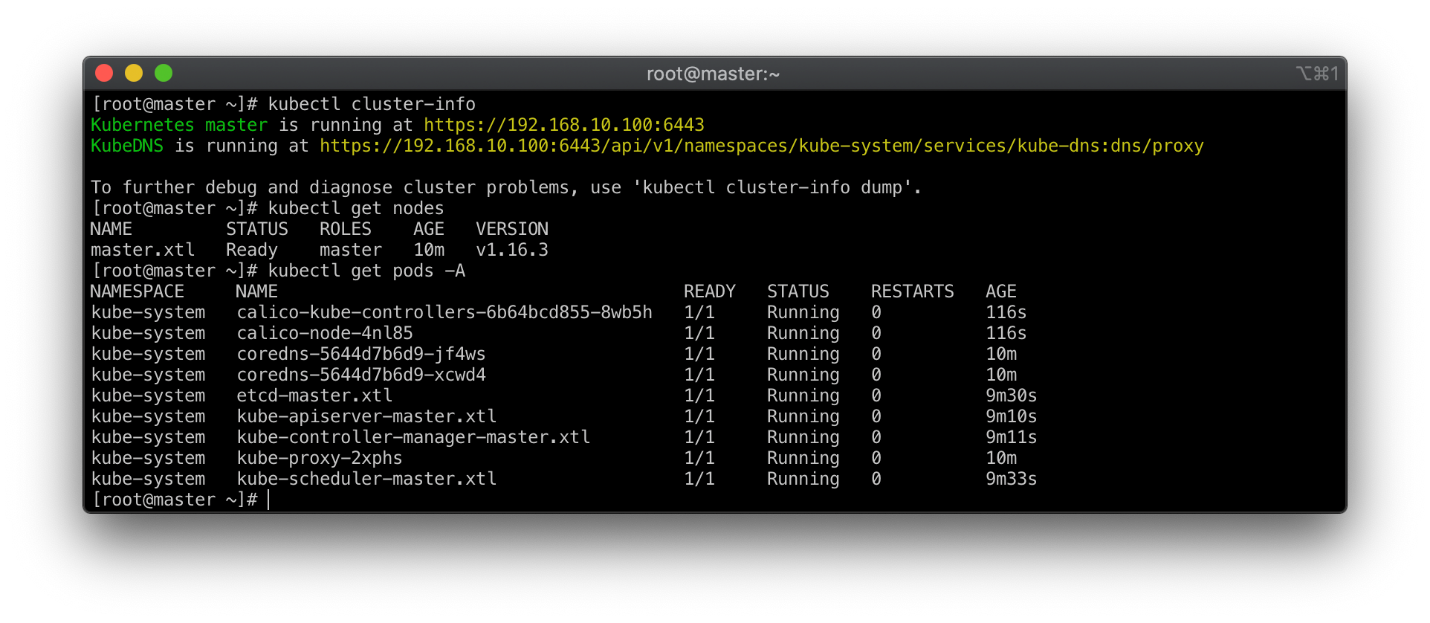
kubectl cluster-info

# Các node trong cluster

kubectl get nodes

# Các pod đang chạy trong tất cả các namespace

kubectl get pods -A



Vậy là đã có Cluster với 1 node!

Cấu hình kubectl máy trạm truy cập đến các Cluster

Chương trình client kubectl là công cụ dòng lệnh kết nối và tương tác với các Cluster Kubernetes, thường khi cài đặt Kubernetes mọi người cũng cài luôn kubectl như phần trên trên, ngay cả máy cài Docker Desktop cũng đã có kubectl. Tất nhiên, bạn có cài đặt kubectl trên một máy không Docker, không Kubernetes với mục đích chỉ dùng nó kết nối đến hệ thống Cluster từ xa. Nếu muốn cài ở máy độc lập như vậy xem tại [Intall kubectl](https://kubernetes.io/docs/tasks/tools/install-kubectl/)

**File cấu hình lệnh kubectl**

Khi thi hành kubectl, thì nó đọc file cấu hình ở đường dẫn $HOME/.kube/config để biết các thông số để kết nối đến Cluster. ($HOME là thư mục gốc dành cho user đang chạy, để biết chính xác gõ lệnh echo $HOME) - tài khoản root thì đó là /root/.kube/config

Trở lại máy Host, để xem nội dung cấu hình kubectl gõ lệnh

kubectl config view

Tại máy master ở trên, có file cấu hình cho tại /root/.kube/config, ta copy file cấu hình này ra lưu thành file config-mycluster (không ghi đè vào config hiện tại của máy HOST)

scp root@172.16.10.100:/etc/kubernetes/admin.conf ~/.kube/config-mycluster

(Nhớ thay đường dẫn theo user của bạn)

Vậy trên máy của tôi đang có 2 file cấu hình

* /User/xuanthulab/.kube/config-mycluster cấu hình kết nối đến Cluster mới tạo ở trên
* /User/xuanthulab/.kube/config cấu hình kết nối đến Cluster cục bộ của bản Kubernetes có sẵn của Docker

Nếu muốn yêu cầu kubectl sử dụng ngay file cấu hình nào đó, thì gán biến môi trường KUBECONFIG bằng đường dẫn file cấu hình, ví dụ sử dụng file cấu hình config-mycluster

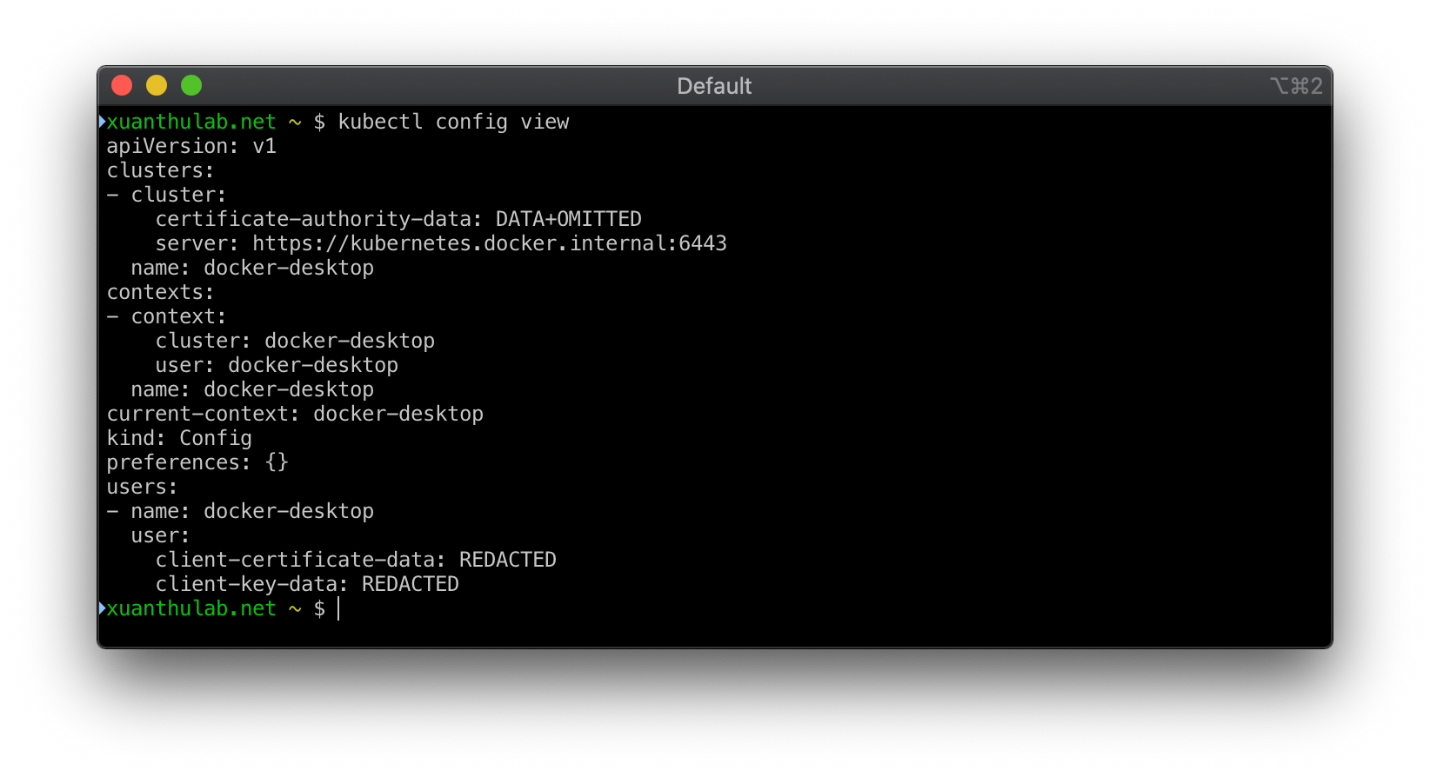
export KUBECONFIG=/Users/xuanthulab/.kube/config-mycluster

Sau lệnh đó thì kubectl sẽ dùng config-mycluster để có thông tin kết nối đến, nhưng trường hợp này chỉ có hiệu lực trong một phiên làm việc, ví dụ nếu bạn đóng terminal và mở lại thì lại phải thiết lập lại biến môi trường như trên.

**Sử dụng các context trong cấu hình kubectl**

(hãy tắt terminal và mở lại để KUBECONFIG không còn tác dụng)

Khi bạn xem nội dung config với lệnh kubectl config view, bạn thấy rằng nó khai báo có các mục cluster là thông tin của cluster với tên, user thông tin user được đăng nhập, context là ngữ cảnh sử dụng, mỗi ngữ cảnh có tên trong đó có thông tin user và cluster.



Ở file trên bạn thấy mục current-context là context với tên docker-desktop, có nghĩa là kết nối đến cluster có tên docker-desktop với user là docker-desktop

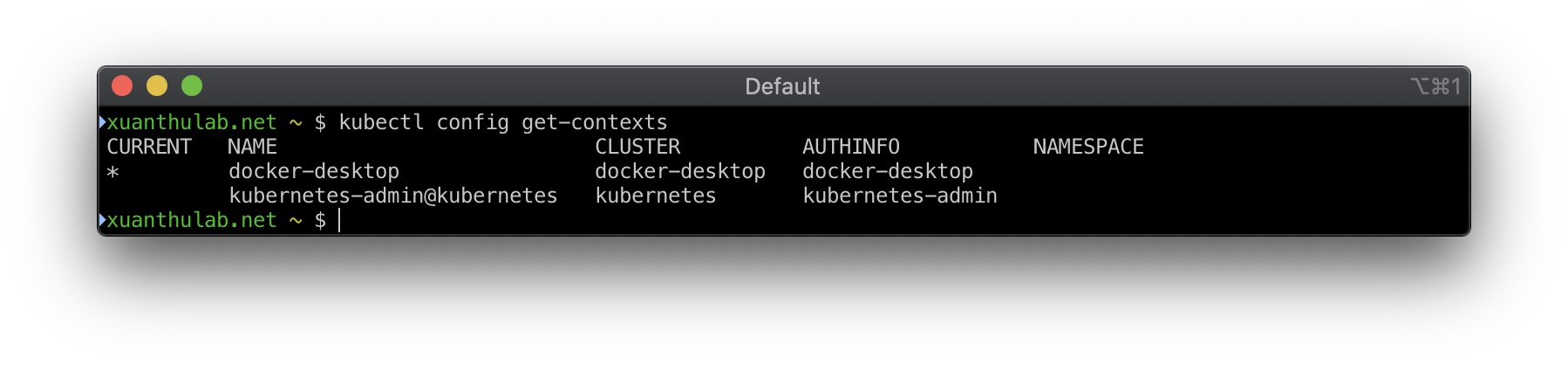
Giờ bạn sẽ thực hiện kết hợp 2 file: config và config-mycluster thành 1 và lưu trở lại config.

export KUBECONFIG=~/.kube/config:~/.kube/config-mycluster

kubectl config view --flatten > ~/.kube/config\_temp

mv ~/.kube/config\_temp ~/.kube/config

Như vậy trong file cấu hình đã có các ngữ cảnh khác nhau để sử dụng, đóng terminal và mở lại rồi gõ lệnh, có các ngữ cảnh nào



Ký hiệu \* là cho biết context hiện tại, nếu muốn chuyển làm việc sang context có tên kubernetes-admin@kubernetes (nối với cluster mới tạo ở trên) thì gõ lệnh

kubectl config use-context kubernetes-admin@kubernetes

Như vậy sử dụng context, giúp bạn lưu và chuyển đổi dễ dàng các loại kết nối đến các cluster của bạn

Cài đặt các node worker - kubernetes

Tạo thư mục kubernetes-centos7/worker1 và kubernetes-centos7/worker2 để cấu hình, tạo các file Vagrantfile trong thư mục tương ứng với nội dung

kubernetes-centos7/worker1/Vagrantfile

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

Vagrant.configure("2") do |config|

config.vm.box = "centos/7"

config.vm.network "private\_network", ip: "172.16.10.101"

config.vm.hostname = "worker1.xtl"

config.vm.provider "virtualbox" do |vb|

vb.name = "worker1.xtl"

vb.cpus = 1

vb.memory = "1024"

end

config.vm.provision "shell", path: "./../install-docker-kube.sh"

config.vm.provision "shell", inline: <<-SHELL

echo "root password"

echo "123" | passwd --stdin root

sed -i 's/^PasswordAuthentication no/PasswordAuthentication yes/' /etc/ssh/sshd\_config

systemctl reload sshd

cat >>/etc/hosts<<EOF

172.16.10.100 master.xtl

172.16.10.101 worker1.xtl

172.16.10.102 worker2.xtl

EOF

SHELL

end

kubernetes-centos7/worker2/Vagrantfile

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

Vagrant.configure("2") do |config|

config.vm.box = "centos/7"

config.vm.network "private\_network", ip: "172.16.10.102"

config.vm.hostname = "worker2.xtl"

config.vm.provider "virtualbox" do |vb|

vb.name = "worker2.xtl"

vb.cpus = 1

vb.memory = "1024"

end

config.vm.provision "shell", path: "./../install-docker-kube.sh"

config.vm.provision "shell", inline: <<-SHELL

echo "root password"

echo "123" | passwd --stdin root

sed -i 's/^PasswordAuthentication no/PasswordAuthentication yes/' /etc/ssh/sshd\_config

systemctl reload sshd

cat >>/etc/hosts<<EOF

172.16.10.100 master.xtl

172.16.10.101 worker1.xtl

172.16.10.102 worker2.xtl

EOF

SHELL

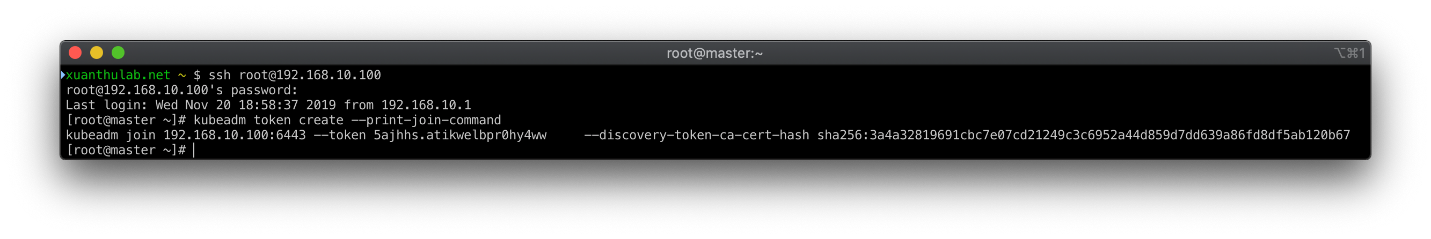
end

Sau đó vào từng thư mục, thực hiện lệnh vagrant up để tạo hai máy ảo có cài đặt docker và kubernetes, máy ảo có tên và ip tương ứng worker1.xtl (172.16.10.101), worker2.xtl (172.16.10.102)

**Kết nối Node vào Cluster**

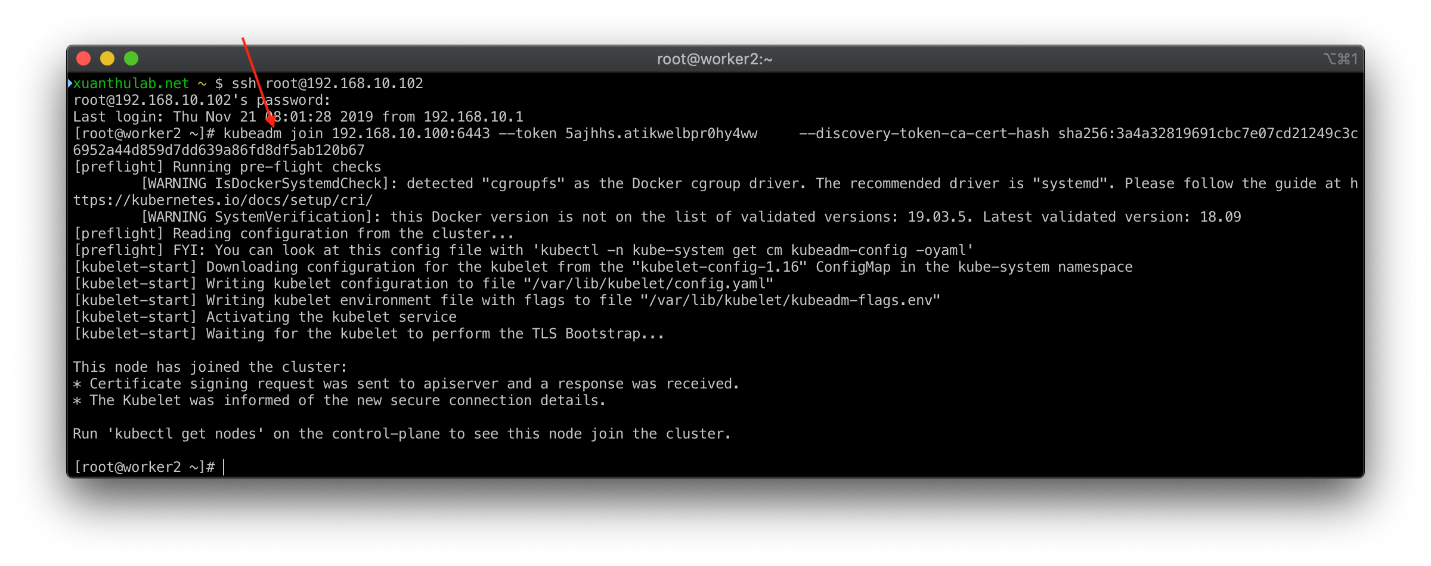
Hãy vào máy node master (bằng SSH ssh root@172.16.10.100). Thực hiện lệnh sau với Cluster để lấy lệnh kết nối

kubeadm token create --print-join-command



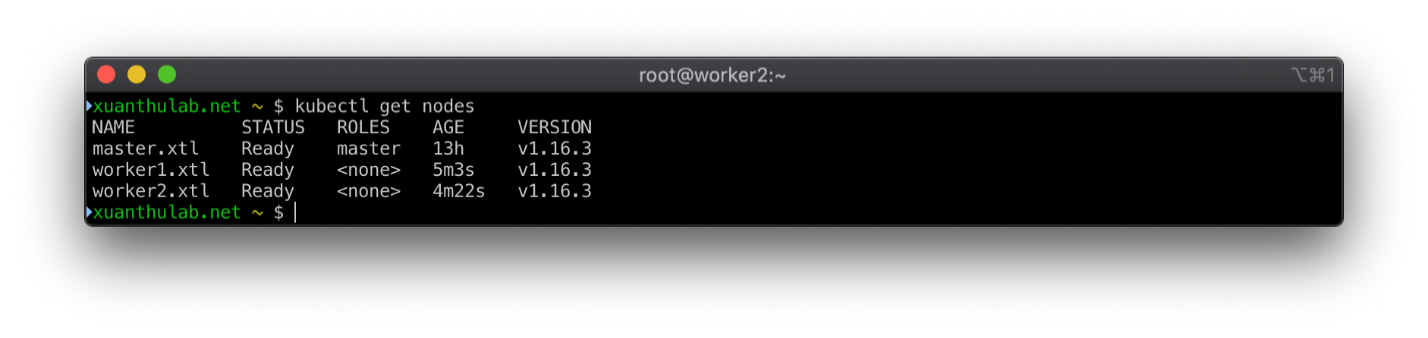
Nó cho nội dung lệnh kubeadm join ... thực hiện lệnh này trên các node worker thì node worker sẽ nối vào Cluster

SSH vào máy worker1, work2 và thực hiện kết nối



Giờ kiểm tra các node có trong Cluster

kubectl get nodes



Tổng kết lại

Đến đây bạn đã biết khởi tạo một Cluster từ Docker Destop hay một Cluster phức tạp 3 node thực thụ, tuy nhiên quá trình cài đặt vẫn chưa hoàn thành, các công cụ cần để dễ dàng làm việc với Kubernetes sẽ tiếp tục ở bài sau, nhưng hiện giờ bạn đã biết các lệnh:

# khởi tạo một Cluster

kubeadm init --apiserver-advertise-address=172.16.10.100 --pod-network-cidr=192.168.0.0/16

# Cài đặt giao diện mạng calico sử dụng bởi các Pod

kubectl apply -f https://docs.projectcalico.org/v3.10/manifests/calico.yaml

# Thông tin cluster

kubectl cluster-info

# Các node (máy) trong cluster

kubectl get nodes

# Các pod (chứa container) đang chạy trong tất cả các namespace

kubectl get pods -A

# Xem nội dung cấu hình hiện tại của kubectl

kubectl config view

# Thiết lập file cấu hình kubectl sử dụng cho 1 phiên làm việc hiện tại của termianl

export KUBECONFIG=/Users/xuanthulab/.kube/config-mycluster

# Gộp file cấu hình kubectl

export KUBECONFIG=~/.kube/config:~/.kube/config-mycluster

kubectl config view --flatten > ~/.kube/config\_temp

mv ~/.kube/config\_temp ~/.kube/config

# Các ngữ cảnh hiện có trong config

kubectl config get-contexts

# Đổi ngữ cảnh làm việc (kết nối đến cluster nào)

kubectl config use-context kubernetes-admin@kubernetes

# Lấy mã kết nối vào Cluster

kubeadm token create --print-join-command

# node worker kết nối vào Cluster

kubeadm join 172.16.10.100:6443 --token 5ajhhs.atikwelbpr0 ...