**Hướng dẫn cài 1 kubenetes cluster trên cụm máy ảo (VirtualBox)**

**1: Sử dụng công cụ vagrant để tương tác với VirtualBox**

**(**[**Hướng dẫn cài Vagrant tại đây**](https://xuanthulab.net/su-dung-vagrant-tao-va-quan-ly-may-ao.html)**)**

**Bật cmd lần lượt tài 3 thư mục worker1,worker2,master chạy lệnh**

vagrant init

vagrant up

**2:** [**cài kubectl trên window**](https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/)

**Note: ssh vào các máy ảo remove containerd cũ có thể bị bản cũ bị lỗi**

yum update

sudo yum remove containerd

sudo yum install containerd.io -y

sudo rm /etc/containerd/config.toml

systemctl restart containerd

**tại máy master tiến hành khởi tạo một kubelet 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

**lấy token để join node worker**

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 copy toàn bộ lệnh này ssh vào worker1 và worker2 để chạy

Note: nếu check status của docker mà lỗi trên các node thì chạy lệnh này để cài lại

sudo yum install docker-ce docker-ce-cli containerd.io docker-compose-plugin

sudo systemctl start docker

* Sau khi thành công bước này chúng ta đã có 1 kubeleters cluster

## 3. Triển khai metrics server

**metrics server** trong kubernetes ([metrics server](https://github.com/kubernetes-sigs/metrics-server)) giám sát về tài nguyên sử dụng trên cluster, cung cấp các API để các thành phần khác truy vấn đến biết được và mức độ sử dụng tài nguyên (CPU, Memory) của Pod, Node ... Cần có Metric Server để HPA hoạt động chính xác

bật cmd tại foder mestric chạy lệnh

kubectl apply -f components.yaml

**4. Triển khai 1 deployment**

Bật cmd tại forder testpod chạy lênh

kubectl apply -f pod1\_deployment.yaml

như vậy 1 deployment đã dc tạo,

kiểm tra xem pod đã chạy chưa bằng lệnh

kubectl get pod –o wide

**5. Tạo một service để có thể truy cập app từ bên ngoài**

Bật cmd tại forder testpod chạy lênh

kubectl apply -f svc\_node\_pord.yaml

service đã được tạo bây h có thể truy cập vào app với đường dẫn sau, ip là ip của 1 trong 3 node

172.16.10.100:31808/test/call

**6. Tạo một HPA để có thể autoscale pod khi cpu và ram của pod vượt ngưỡng**

Bật cmd tại forder testpod chạy lênh

kubectl apply -f hpa.yaml

Như vậy đã tạo xong 1 cluster K8s deploy 1 app java đơn giản có thể autoscale khi lượng truy cập lớn

-----------------------------------------K8S--------------------------------------------------

NOTE:

Thông tin tài liệu tham khảo có thể xem thêm [tại đây](https://xuanthulab.net/gioi-thieu-va-cai-dat-kubernetes-cluster.html)

Các khải niệm cơ bản xem tại file Kubernetes – K8S.pptx

Các lệnh kubectl để tương tác với api server có thể xem trong file note.txt