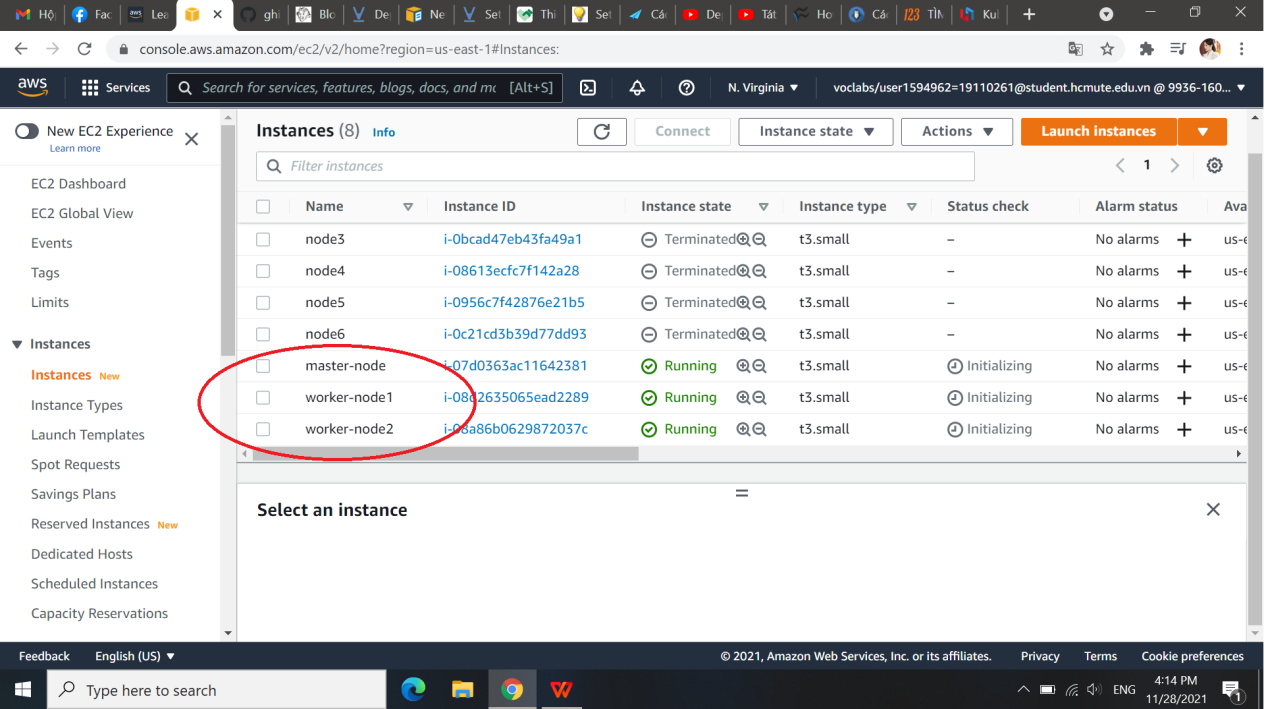
1. **Tìm hiểu Kubernetes**

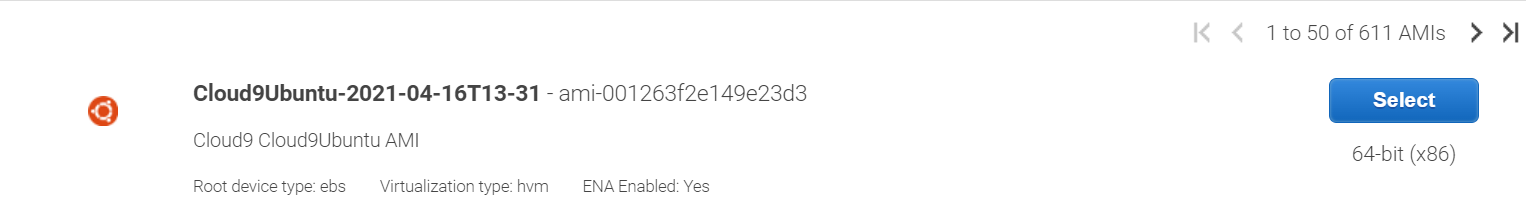
Nhóm em deploy 1 web nginx có sẵn để chạy demo tiện mô tả tính chất Kubernetes một cách rõ nét

1. Tạo cụm Kubernetes

Nhóm sử dụng dịch vụ Amazon EC2 để tiến hành tạo 3 máy ảo giống nhau



Nền tảng: Cloud9Ubuntu

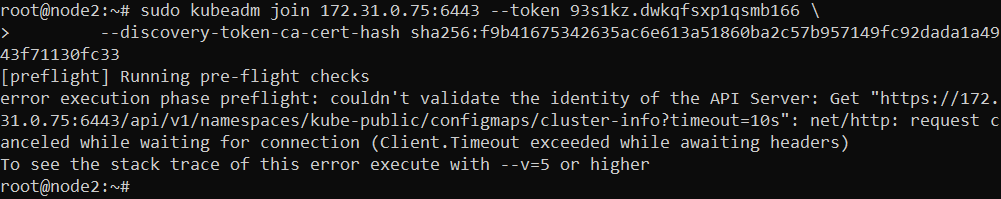


Loại máy EC2: t3.small

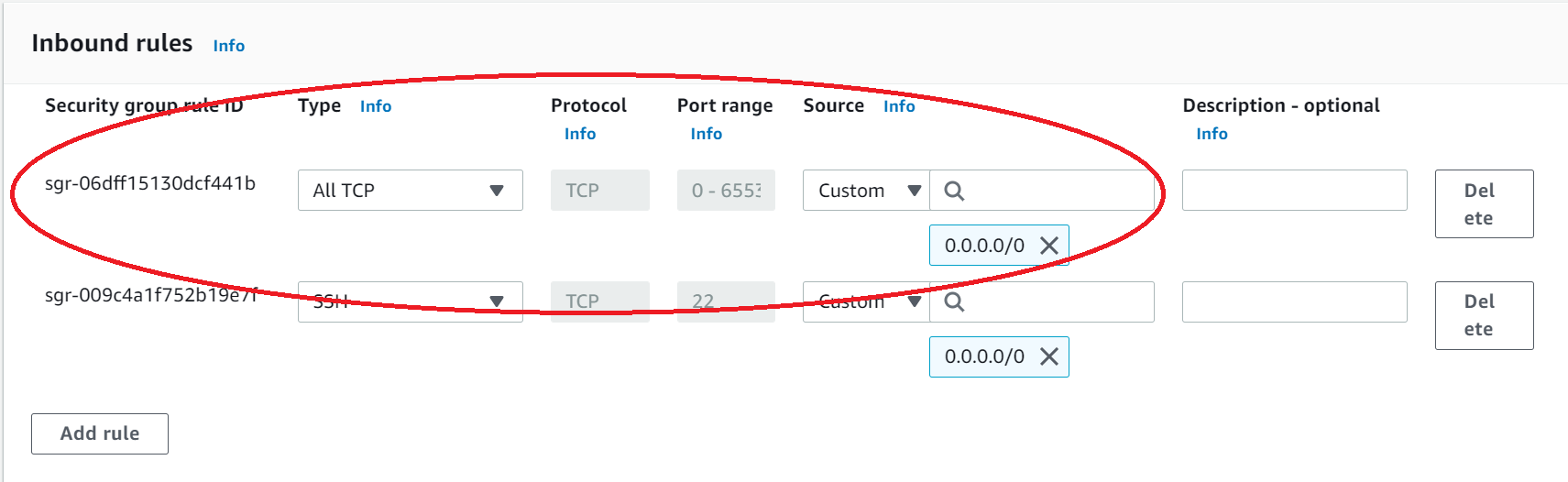
CPU: 02 vCPU

RAM: 02 GB

Khi dùng lệnh “ kubeadm join ” nút công nhân không thể tham gia vào nút chính. Nguyên nhân gốc rễ là cổng 6443, cổng này đã bị chặn trong nhóm bảo mật của AWS.

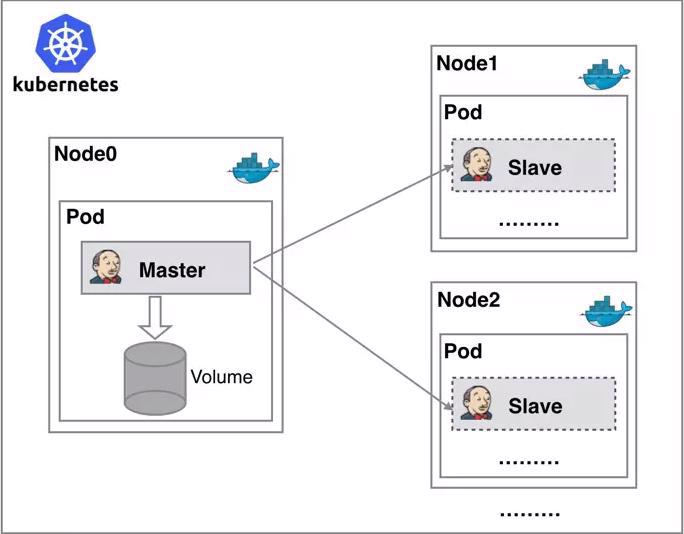


Vì thế em vào Security Groups thêm 1 rule với type: All TCP ( giải quyết lỗi )



* 1. Mô hình

-



- Máy chủ 0 = node0 có địa chỉ IP Private là 172.31.1.252

- Máy chủ 1 = node1 có địa chỉ IP Private là 172.31.7.205

- Máy chủ 2 = node2 có địa chỉ IP Private là 172.31.8.111

- Đặt tên cho các máy chủ với lệnh:

sudo hostnamectl set-hostname "master"

sudo hostnamectl set-hostname "node1"

sudo hostnamectl set-hostname "node2"

Trên cả 3 node:

**Cài đặt docker**

**Nhận khóa gpg Docker**

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

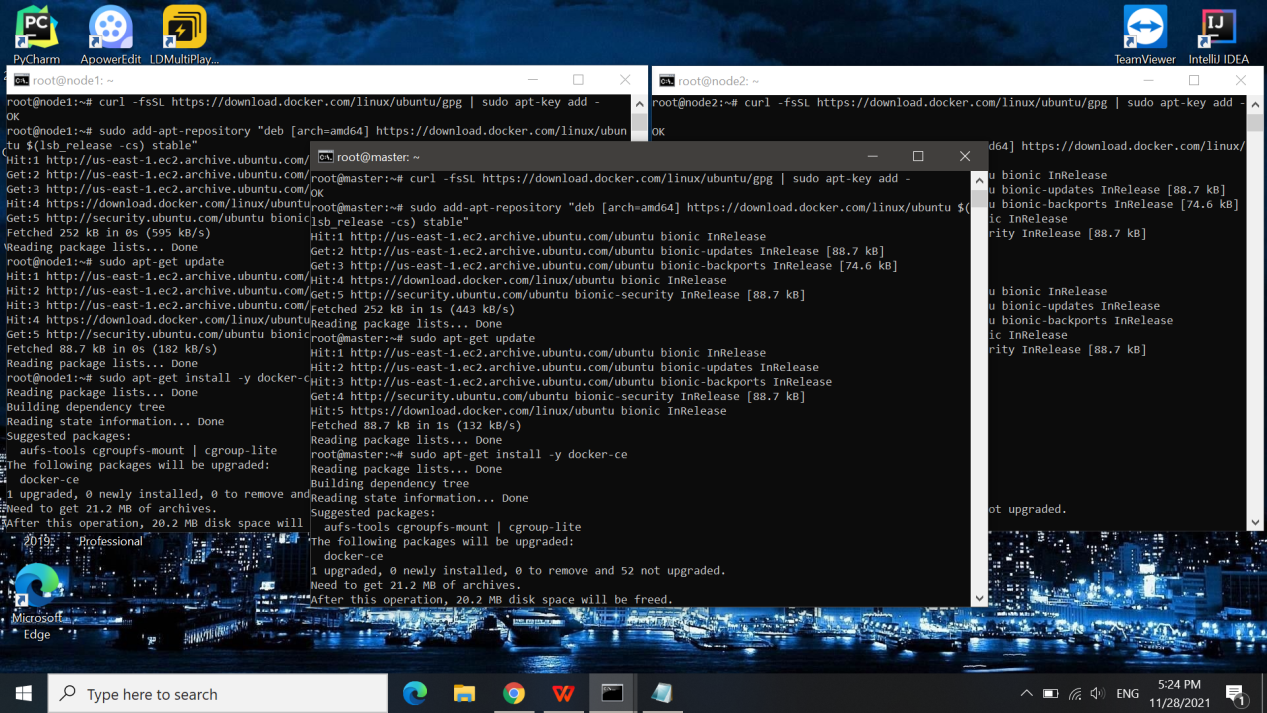
**Thêm kho lưu trữ Docker**

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt-get update ( cập nhật các gói )

**Install Docker**

sudo apt-get install -y docker-ce



**Cài đặt** **kubelet, kubeadm, kubectl**

**Nhận khóa gpg Kubernetes**

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

**Thêm kho lưu trữ Kubernetes**

cat << EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list

deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF

sudo apt-get update (cập nhật các gói )

**Install** **kubelet, kubeadm, kubectl**

sudo apt-get install -y kubelet kubeadm kubectl

**Giữ chúng ở phiên bản hiện tại**

sudo apt-mark hold docker-ce kubelet kubeadm kubectl

**Thêm quy tắc iptables vào sysctl.conf**

echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf

**Bật iptables ngay lập tức**

sudo sysctl -p

**Để tắt hoán đổi tạm thời** **(** khắc phục lỗi **kubeadm init )**

sudo swapoff -a

**Gặp lỗi với trình điều khiển cgroup**

Trình điều khiển cgroup Kubernetes đã được đặt thành hệ thống nhưng docker được đặt thành systemd. Vì vậy, em đã tạo /etc/docker/daemon.json và thêm vào bên dưới: **(** khắc phục lỗi **kubeadm init )**

sudo touch /etc/docker/daemon.json

sudo vim /etc/docker/daemon.json

{

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

}

esc --> :wq

sudo systemctl daemon-reload

sudo systemctl restart docker

sudo systemctl restart kubelet

1. Tạo Cluster

Cluster sẽ bao gồm các tài nguyên vật lý sau:

- Master node là nút kiểm soát và quản lý một tập hợp các worker node (workloads runtime) và giống như một cluster trong Kubernetes. Nó cũng nắm giữ kế hoạch tài nguyên của node để xác định hành động thích hợp cho event được kích hoạt. Nó chạy etcd, một kho lưu trữ key-value phân tán mã nguồn mở được sử dụng để lưu giữ và quản lý dữ liệu cluster giữa các thành phần lên lịch workload cho các worker node

- 2 worker node là các nút tiếp tục công việc được giao ngay cả khi master node ngừng hoạt động sau khi lập lịch hoàn tất. Các worker node là các máy chủ nơi workload của bạn (tức là các ứng dụng và dịch vụ được tích hợp sẵn) sẽ chạy. Bạn cũng có thể tăng công suất của cluster bằng cách thêm các worker.

- Sau khi tạo thành công, chúng ta sẽ có một cluster đầy đủ chức năng sẵn sàng để chạy workload (tức là các application và service được chứa trong container) nếu các máy chủ trong cluster có đủ tài nguyên CPU và RAM để các ứng dụng của bạn chạy. Sau khi bạn đã thiết lập thành công cluster, bạn có thể chạy hầu hết mọi ứng dụng UNIX truyền thống. Nó có thể được chứa trong cluster của bạn, bao gồm các ứng dụng web, cơ sở dữ liệu, daemon và các công cụ command-line.

- Bản thân cluster sẽ tiêu thụ khoảng 300-500MB bộ nhớ và 10% CPU trên mỗi node.

**Trên Master node**

**Khởi tạo cụm**

sudo kubeadm init --apiserver-advertise-address=172.31.1.252 --pod-network-cidr=10.244.0.0/16

**Thiết lập kubeconfig cục bộ**

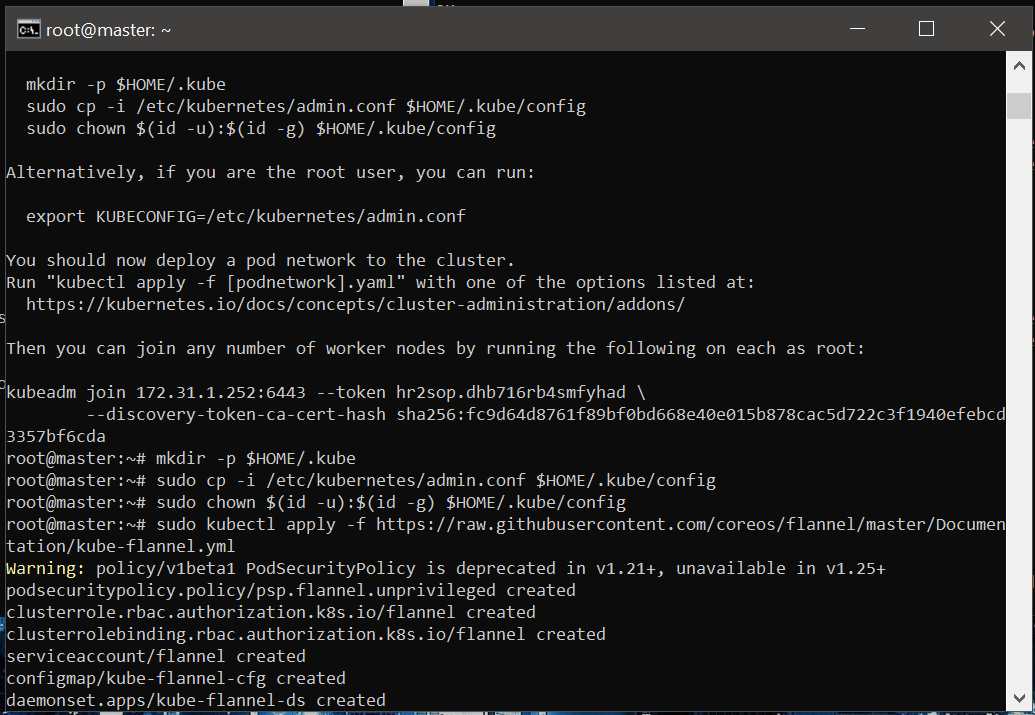
mkdir -p $ HOME / .kube

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

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

**Áp dụng lớp phủ mạng CNI Flannel**

sudo kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml



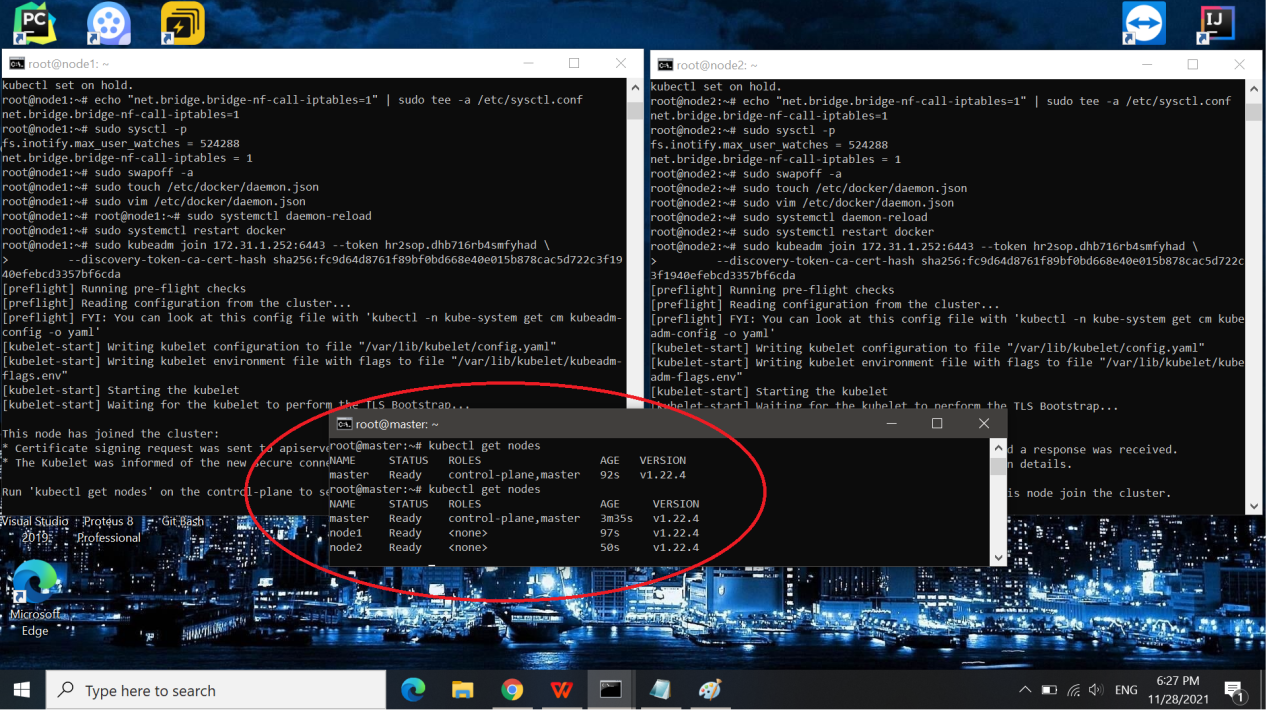
**Trên Node1 và Node 2**

**Các nút công nhân tham gia vào cụm**

sudo kubeadm join 172.31.1.252:6443 --token hr2sop.dhb716rb4smfyhad \ --discovery-token-ca-cert-hash sha256:fc9d64d8761f89bf0bd668e40e015b878cac5d722c3f1940efebcd3357bf6cda

**Kiểm tra các node trong Cluster được tạo**

kubectl get nodes



Hoàn thành thiết lập nút chính Kubernetes và nút công nhân trên EC2 của đám mây AWS.

1. Triển khai web NGINX trên Clusters vừa tạo

**Trên node master**

**Tạo triển khai nginx = cách sử dụng NGINX image**

sudo kubectl create deployment nginx --image=nginx

**Tạo 1 dịch vụ với type = NodePort**

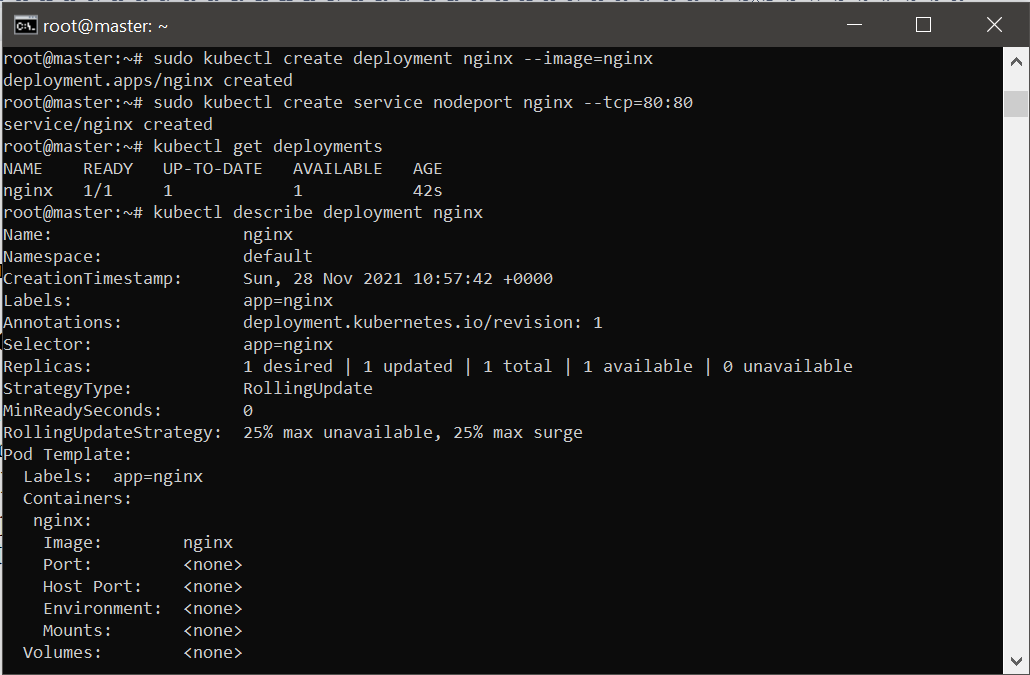
sudo kubectl create service nodeport nginx --tcp=80:80

**Xem trạng thái triển khai**

kubectl get deployments

**Xem mô tả triển khai nginx**

kubectl describe deployment nginx



**Xem các dịch vụ đang có**

kubectl get svc

**Kiểm tra số pod**

kubectl get pods

**Scale lên 10 pod**

kubectl scale --replicas=10 deployments/nginx

Các pods của nginx-deployment được Cluster chia đều qua 2 worker node.

Câu hỏi đặt ra, điều gì 1 server worker node bị lỗi or chết?

Em sẽ tiến hành thực hành để thấy rõ vấn đề này.

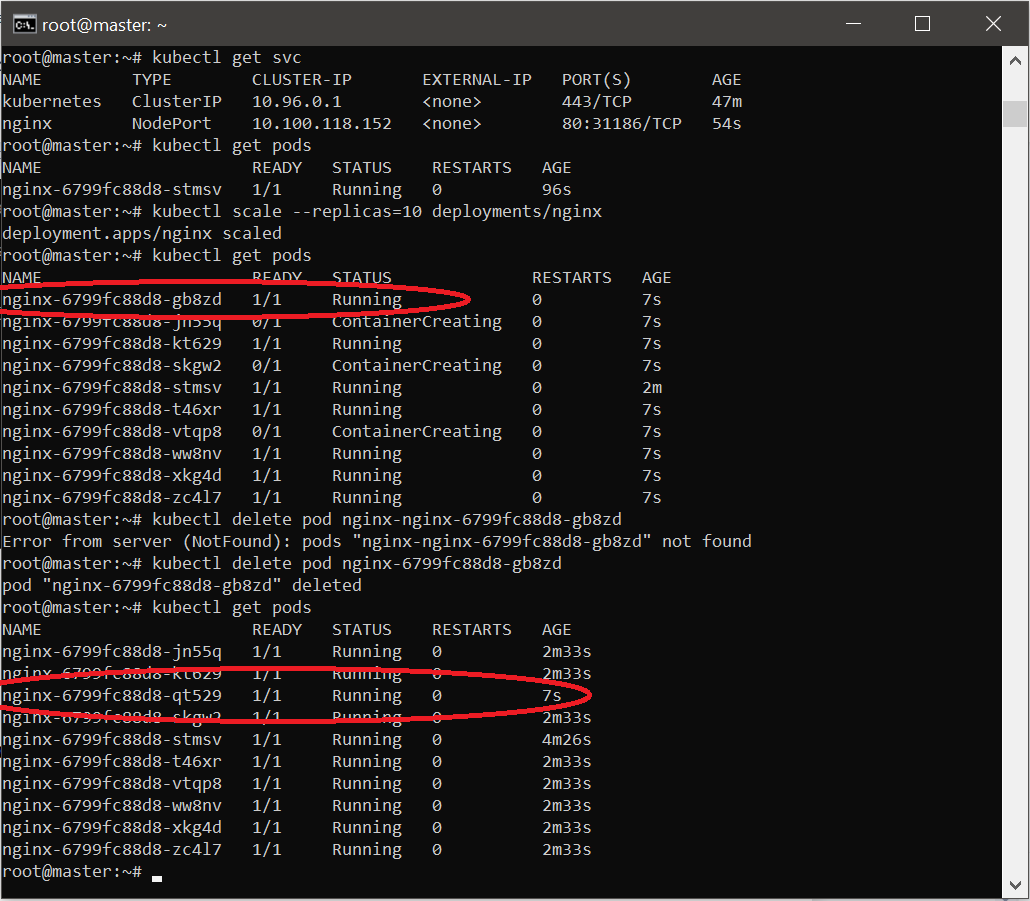
**Thử xóa 1 pods**

kubectl delete pod nginx-6799fc88d8-gb8zd

**Kiểm tra lại số pod**

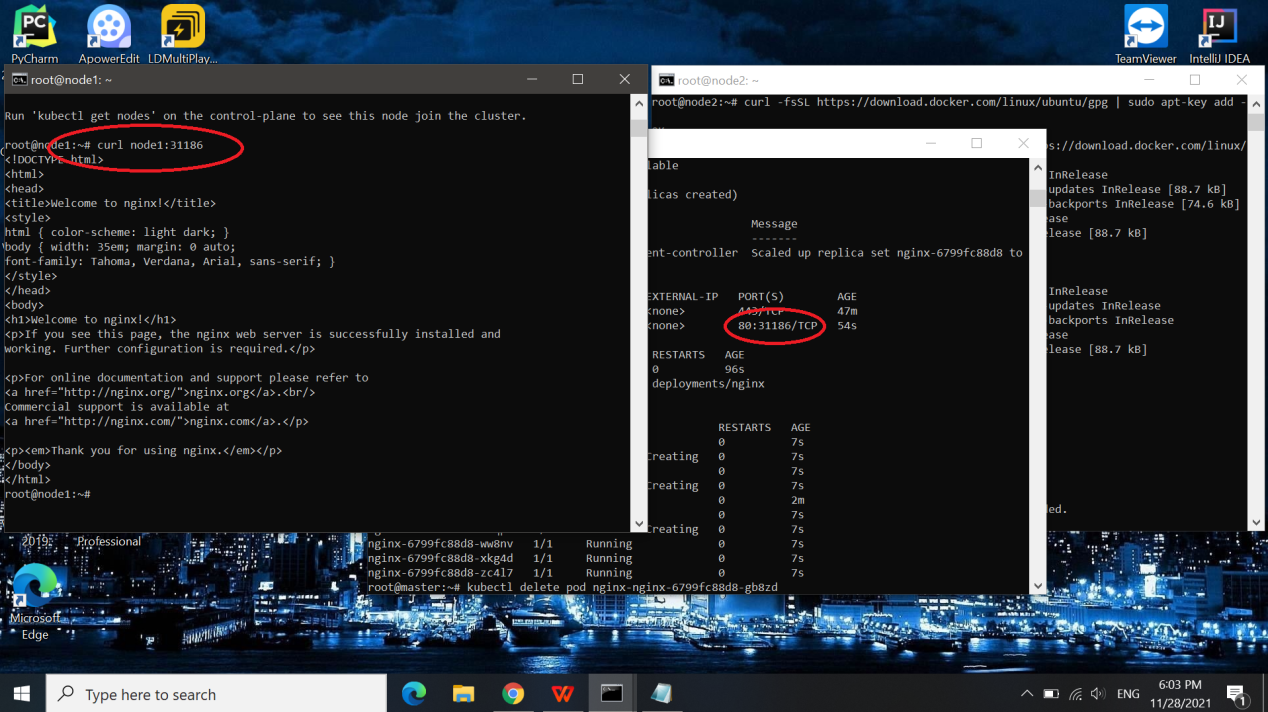
kubectl get pods

Thì thấy 1 pod nginx-6799fc88d8-gb8zd đã bị xóa và 1 pods mới nginx-6799fc88d8-qt529 được tạo ra sau 7s thay thế cho pod cũ nginx-6799fc88d8-gb8zd



Thực hiện thành công theo đúng như lí thuyết điển hình về Kubernets Cluster

Truy cập web trên woker node



**Mở trên brow****ser**

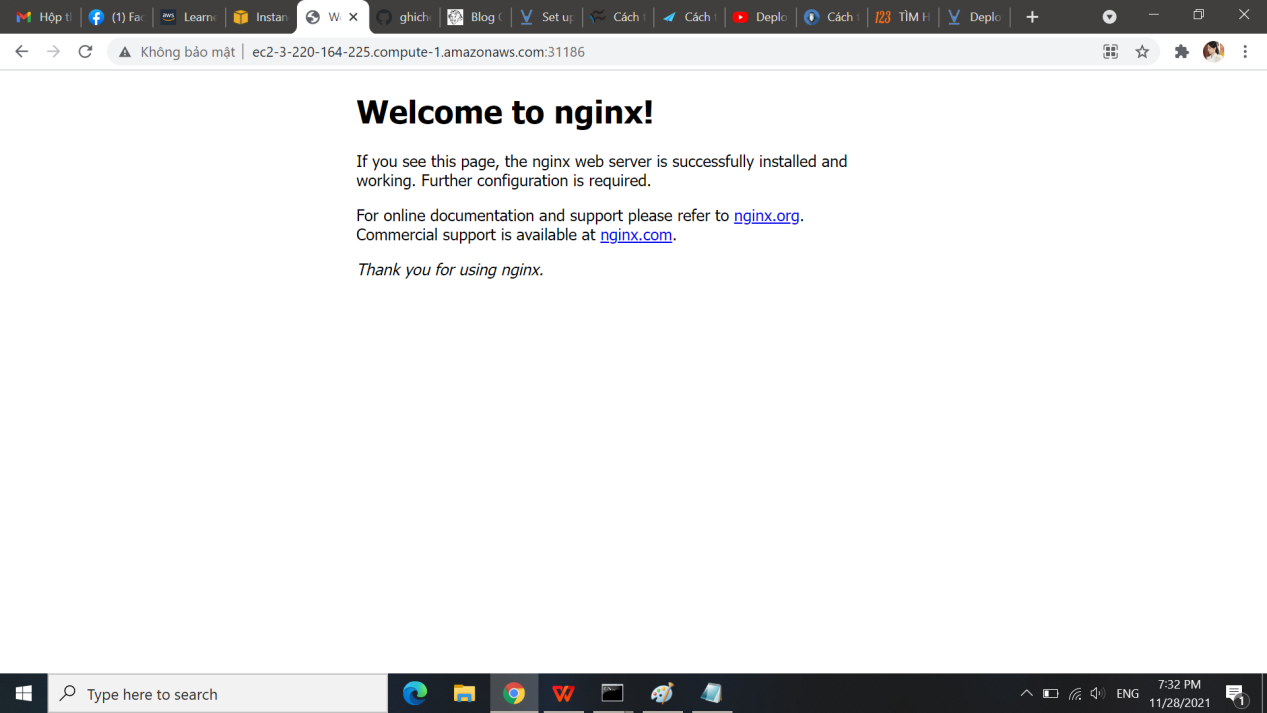
Sau khi em tìm hiểu thì

Dùng lệnh: # ip a

Thì thấy không có IP Public nào tồn tại.

Vấn đề: Vì hiện đang chạy triển khai này trên Máy ảo do nhà cung cấp đám mây công cộng cung cấp. Vì vậy, mặc dù không có giao diện cụ thể nào được chỉ định IP công cộng, nhưng nhà cung cấp máy ảo đã cấp một địa chỉ IP bên ngoài Ephemeral.

Địa chỉ IP bên ngoài tạm thời là địa chỉ IP tạm thời vẫn được gắn vào máy ảo cho đến khi phiên bản ảo bị dừng. Khi phiên bản ảo được khởi động lại, một IP bên ngoài mới sẽ được chỉ định. Về cơ bản, đó là một cách đơn giản để các nhà cung cấp dịch vụ tận dụng các IP công cộng nhàn rỗi.

Thách thức ở đây, ngoài thực tế là IP công cộng của bạn không phải là tĩnh, IP công cộng Ephemeral chỉ đơn giản là một phần mở rộng (hoặc proxy) của IP Riêng và vì lý do đó, dịch vụ sẽ chỉ được truy cập trên cổng **31186**. Điều đó có nghĩa là dịch vụ sẽ được truy cập trên URL, đó là http://ec2-3-220-164-225.compute-1.amazonaws.com:**31186**/, kiểm tra trình duyệt của mình, em đã thấy trang chào mừng. 

Em đã triển khai thành công NGINX trên cụm Kubernetes 3 nút của chúng em.

http://ec2-3-220-164-225.compute-1.amazonaws.com:31186/

Vì mỗi lần reset thì sẽ thay đổi Public IPv4 DNS nên sẽ k còn dùng được nữa.

