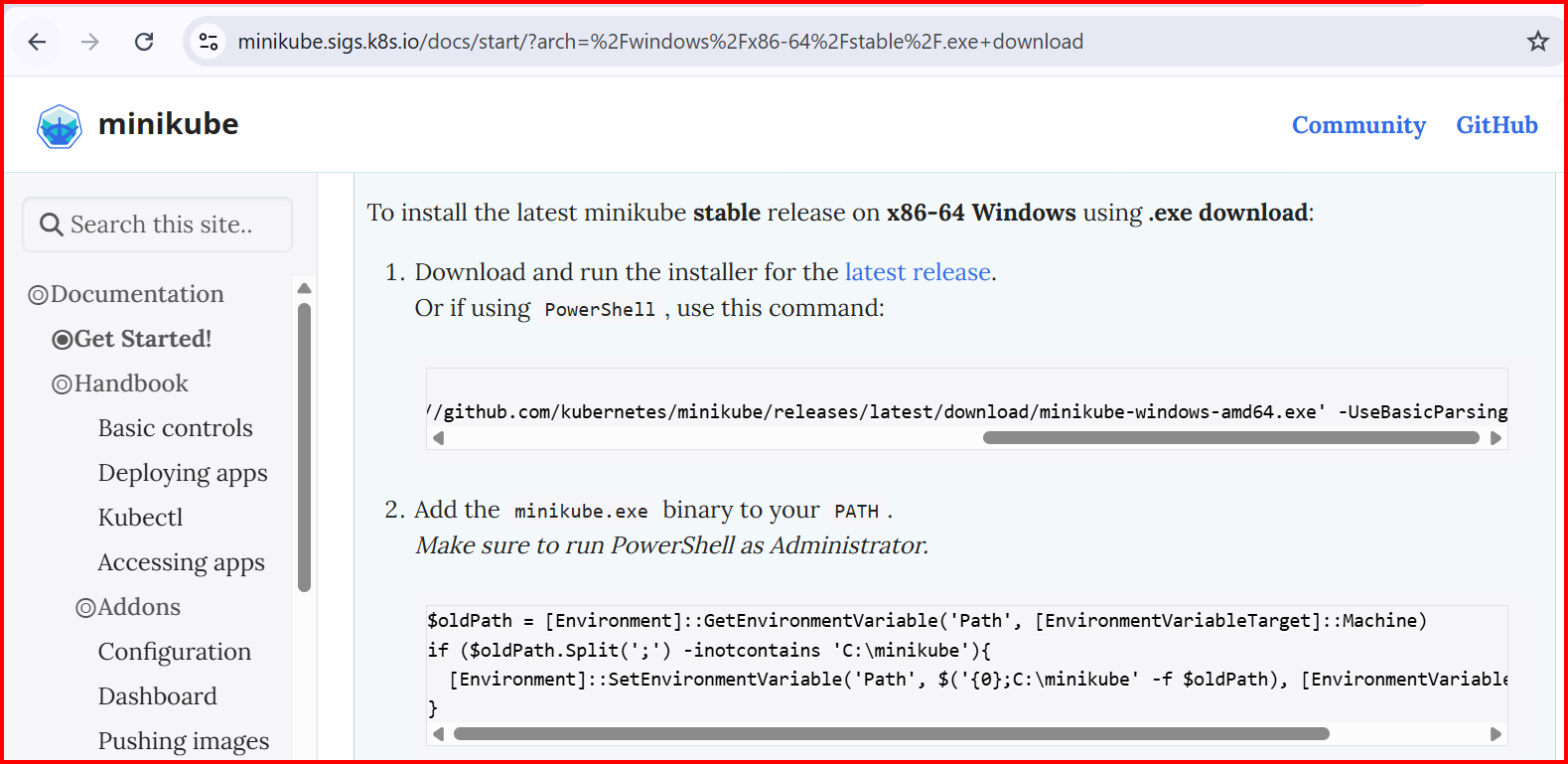
**Kubernetes-01**

1. **Setup Minikube in your local machine.**
2. **Setup k8s master and two worker nodes on ubuntu.**
3. **Run one nginx pod.**
4. **Mug up Master and slave components on k8s.**

**1.**

Minikube is a lightweight tool that lets you run a single-node  
Kubernetes cluster locally — great for learning,   
development, and testing Kubernetes.  
Prerequisites  
1.Before installing Minikube, make sure you have:  
•Virtualization support (e.g., Docker, KVM, VirtualBox)  
2.Install Docker on windows.  
https://docs.docker.com/desktop/install/windows-install/  
3.Install Oracle box on windows  
https://adamtheautomator.com/install-virtualbox-on-windows-10/  
4.Install Minikube using powershell  
[**https://minikube.sigs.k8s.io/docs/start/**](https://minikube.sigs.k8s.io/docs/start/)

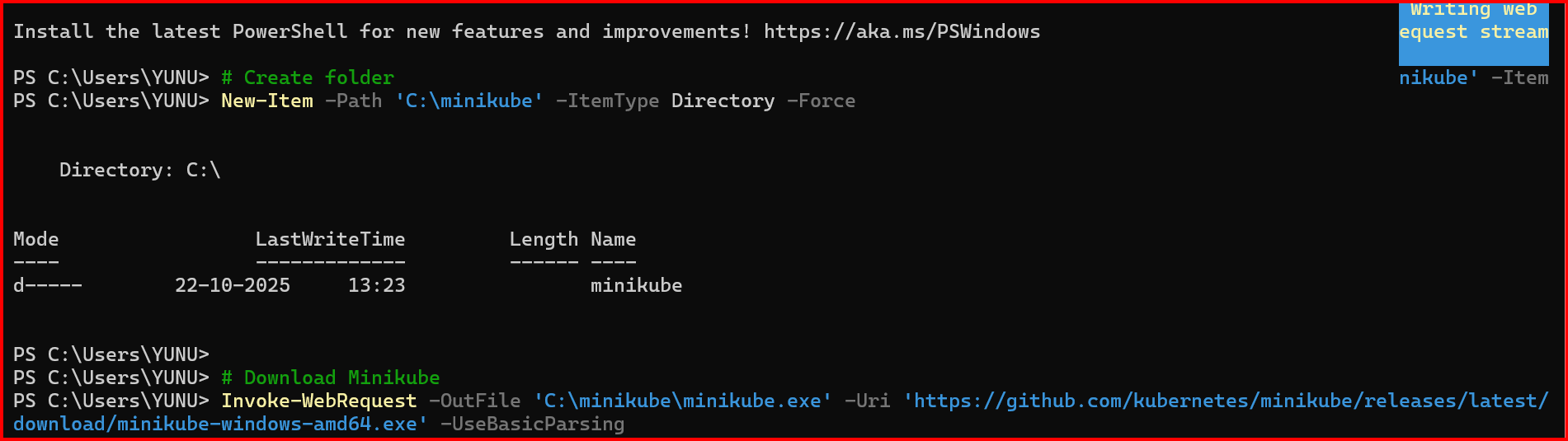
****

**# Create folder**

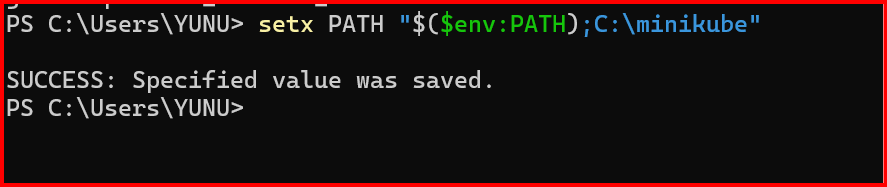
New-Item -Path 'C:\minikube' -ItemType Directory -Force

**# Download Minikube**

Invoke-WebRequest -OutFile 'C:\minikube\minikube.exe' -Uri 'https://github.com/kubernetes/minikube/releases/latest/download/minikube-windows-amd64.exe' -UseBasicParsing

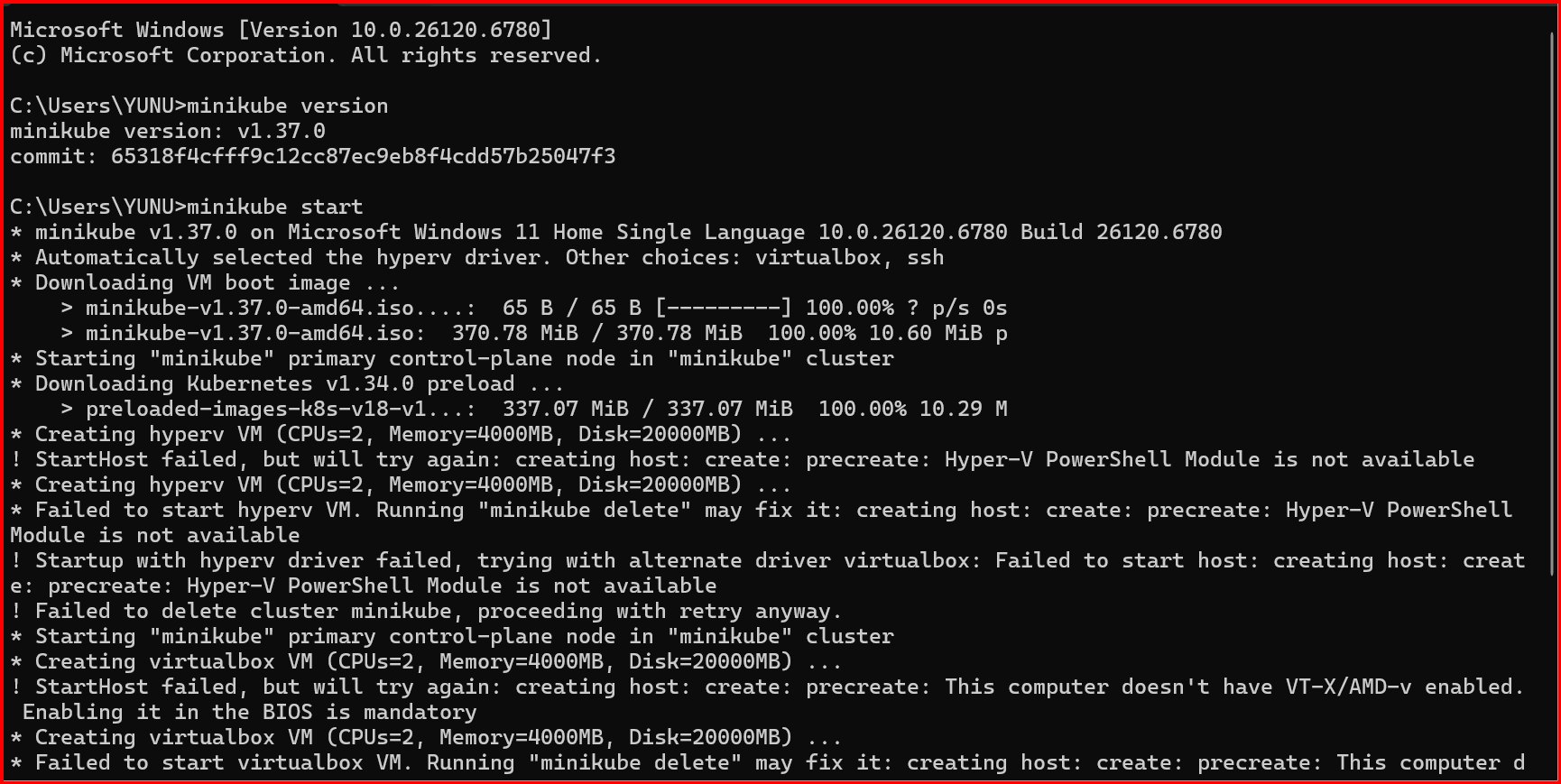
****

**setx PATH "$($env:PATH);C:\minikube"  
close this window and open**

****

**minikube version**

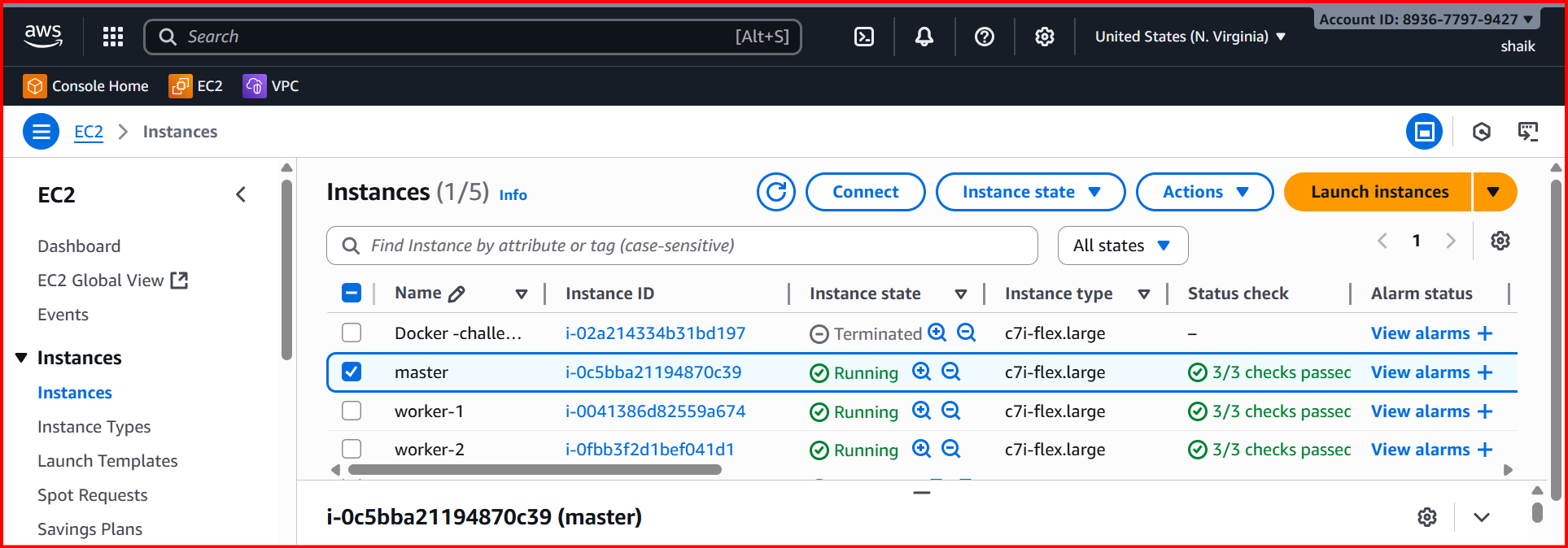
**minikube start**

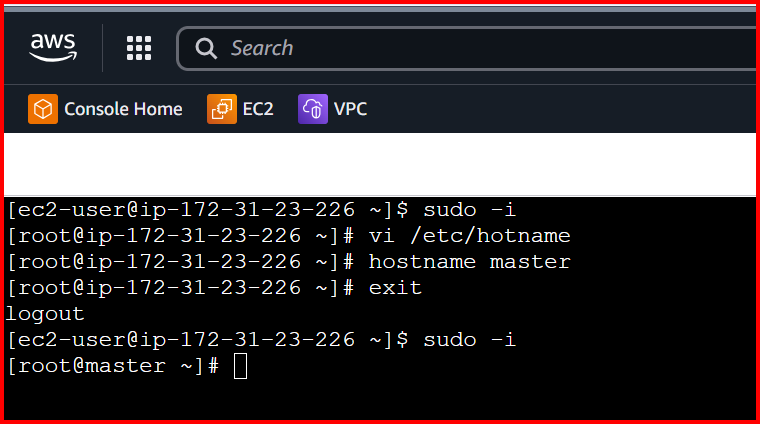
****

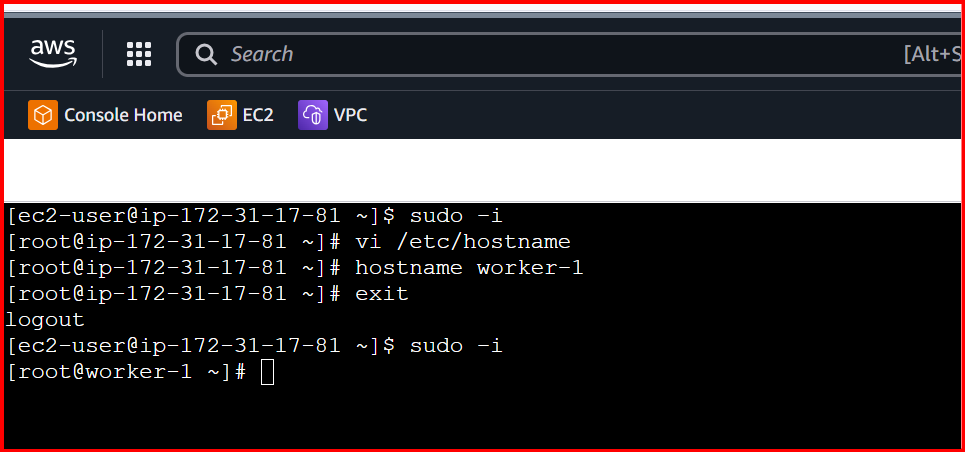
**2, Setup k8s master and two worker nodes on ubuntu.**

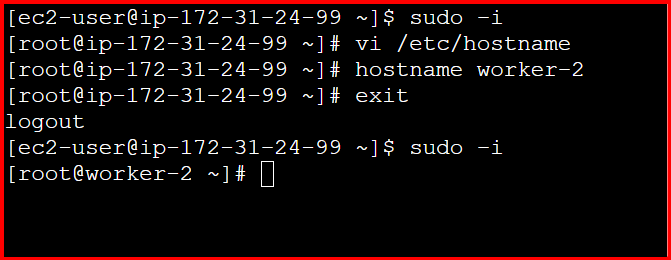
launch 3 instances with 20gb memory

Change the hostname  
vi /etc/hostname  
hostname master  
similarly do it workers hostnames





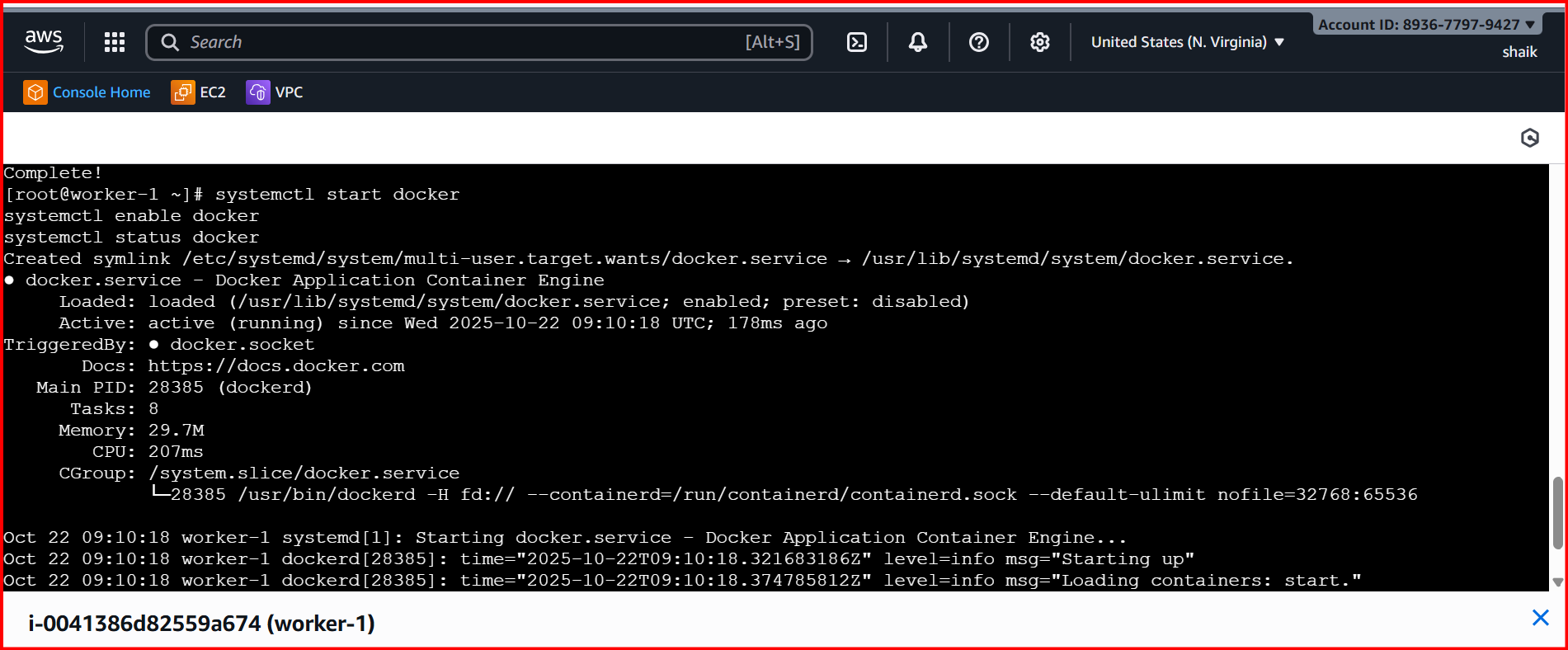




**Install docker on all 3 instances**

**yum –y install docker**

**systemctl start docker  
systemctl enable docker  
systemctl status docker**

****

**To setup k8s cluster we have different tools like kops and kubeadm**

**sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config**

**vi /etc/yum.repos.d/kubernetes.repo**

**Add**

**[kubernetes]**

**name=Kubernetes**

**baseurl=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/**

**enabled=1**

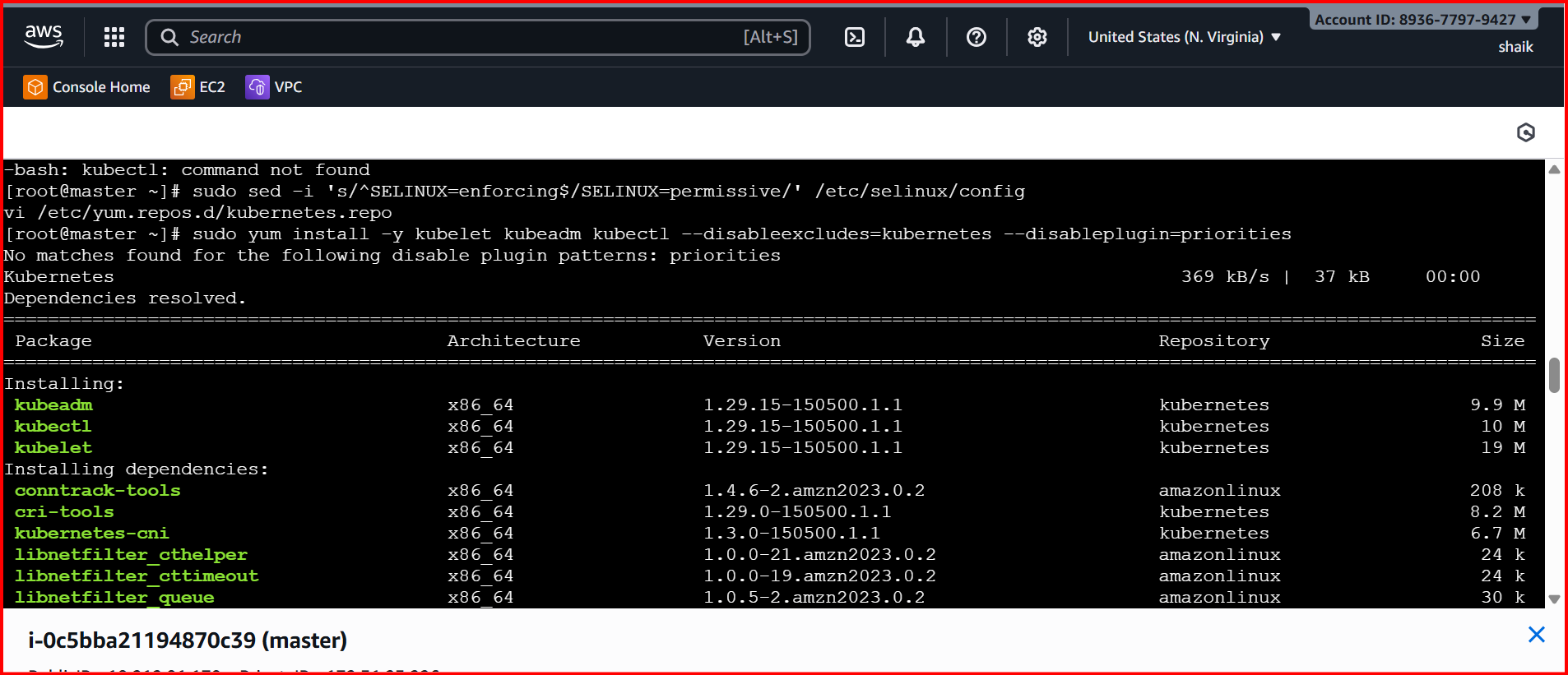
**gpgcheck=1**

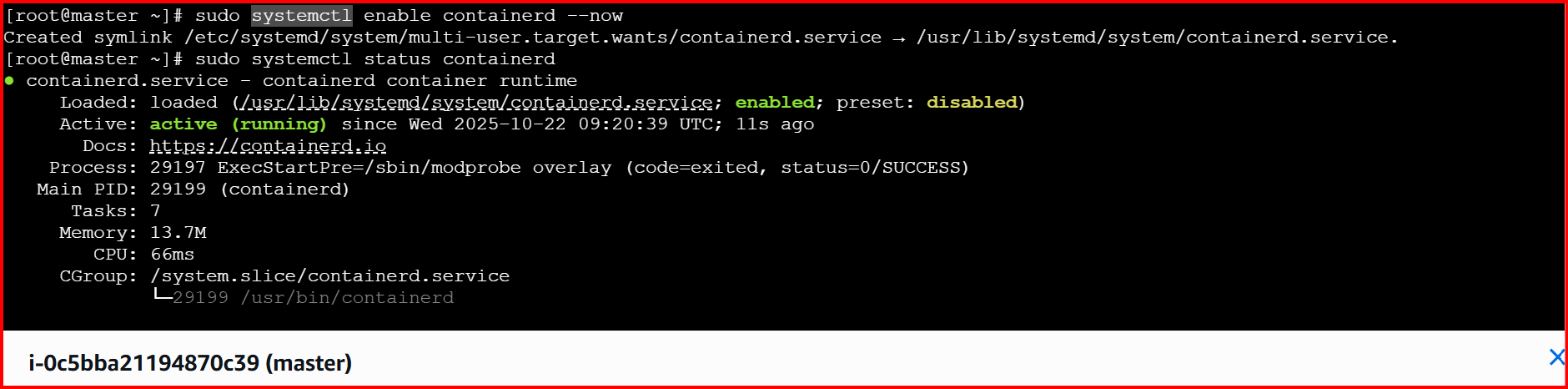
**gpgkey=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/repodata/repomd.xml.key**

**exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni**

**For installation execute the below command**

**sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes --disableplugin=priorities**

****

****

**Now we need to setup networking and configure master and workernodes**

**kubeadm init --pod-network-cidr=10.244.0.0/16**

**If the above command is not working**

**Enable br\_netfilter module**

**sudo modprobe br\_netfilter**

**lsmod | grep br\_netfilter**

**# Set bridge and IPv4 forwarding parameters**

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

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

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

**net.ipv4.ip\_forward = 1**

**EOF**

**# Apply changes immediately**

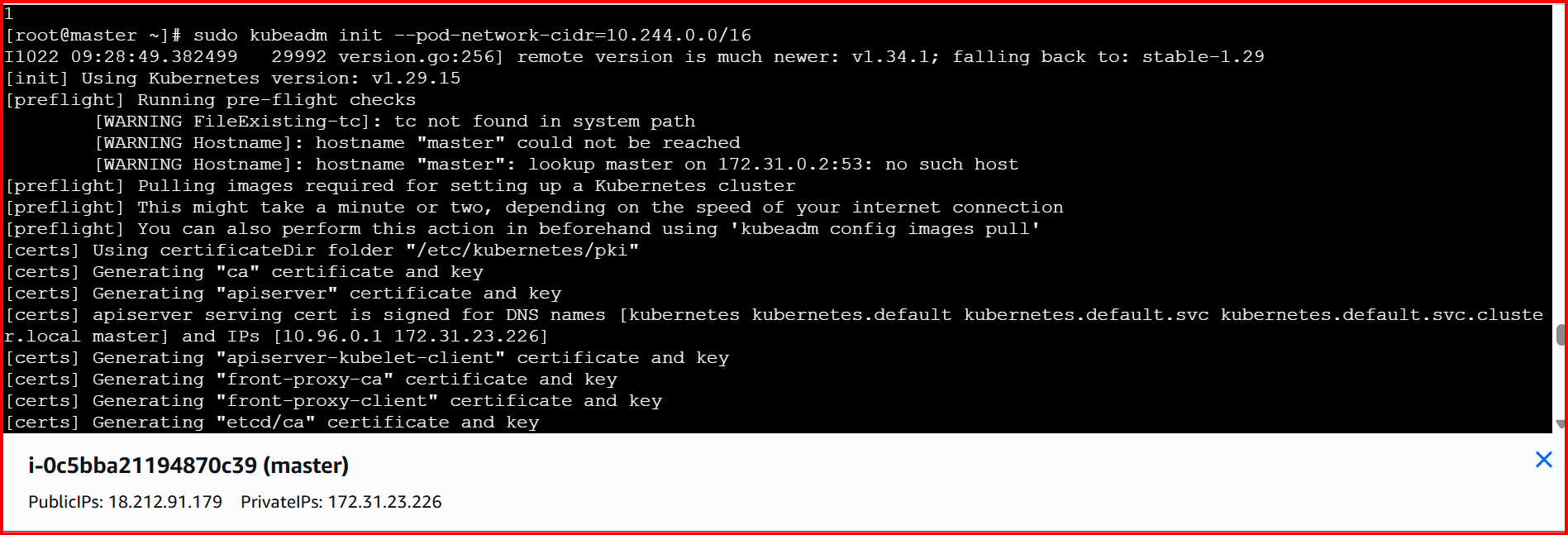
**sudo sysctl --system**

You can verify

cat /proc/sys/net/bridge/bridge-nf-call-iptables

cat /proc/sys/net/ipv4/ip\_forward

sudo kubeadm init --pod-network-cidr=10.244.0.0/16



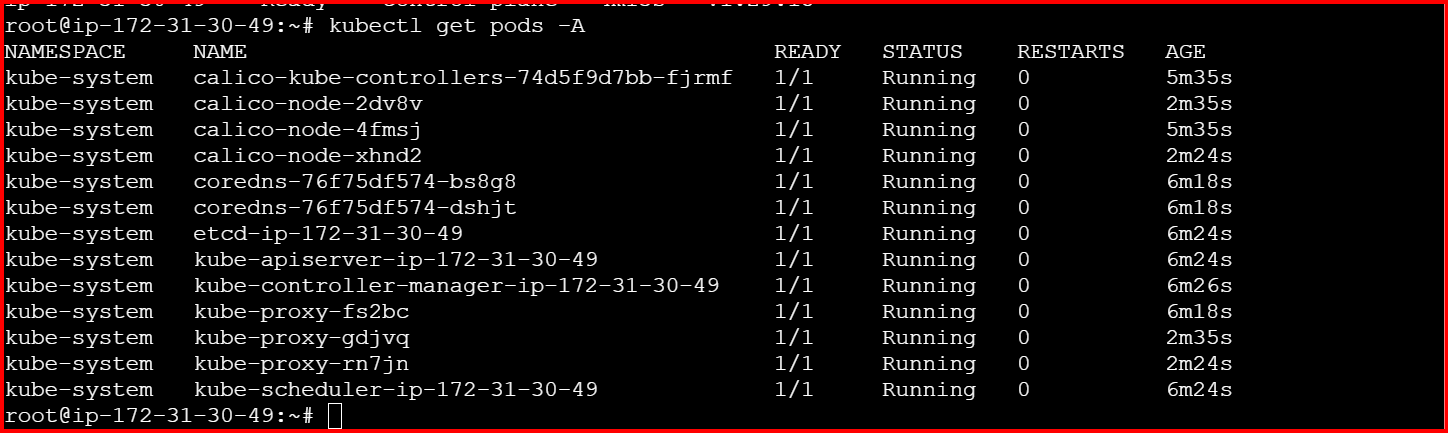
**mkdir -p $HOME/.kube**

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

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

****

**kubectl get pods -A**

****

**3.Run one nginx pod**

**Create a pod YML for Nginx**

**Vi Nginx-pod.yml**

apiVersion: v1

kind: Pod

metadata:

name: nginx-pod

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:latest

ports:

- containerPort: 80

**Kubectl apply -f ngnix-pod.yml**

**Kubectl get pods**

