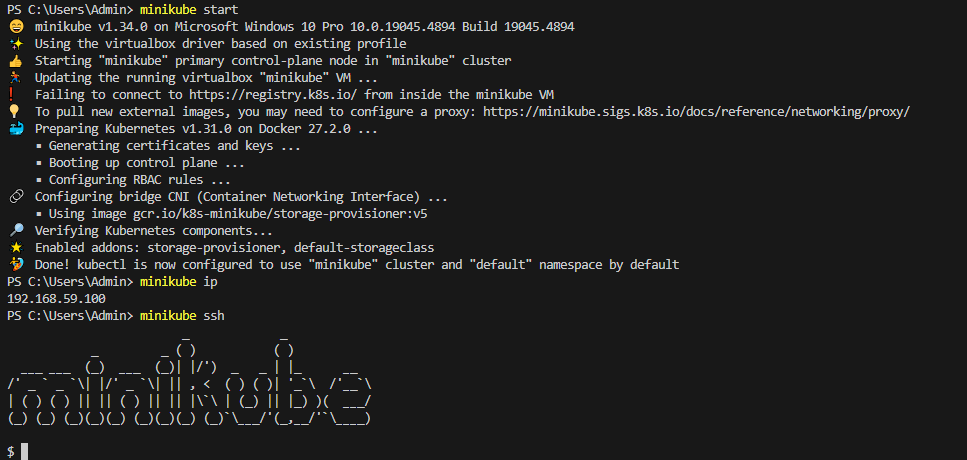
1) Setup Minikube in your local machine.



2) Setup k8s master and two worker nodes on aws ec2.  
**On Ubuntu server only**  
  
  
Step 1: Disable swap:

sudo swapoff -a

Step 2: Set up hostnames

sudo hostnamectl set-hostname "master " 🡪 For Master Node

sudo hostnamectl set-hostname "worker01 " 🡪 For Worker1 Node

sudo hostnamectl set-hostname "worker02 " 🡪 For Worker2 Node  
  
Step 3: Update the /etc/hosts File for Hostname Resolution

sudo vi /etc/hosts

Add private ip’s of nodes

Example:-  


### Step 4: Set up the IPV4 bridge on all nodes

cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf

overlay

br\_netfilter

EOF

sudo modprobe overlay

sudo modprobe br\_netfilter

# sysctl params required by setup, params persist across reboots

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

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

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

net.ipv4.ip\_forward = 1

EOF

# Apply sysctl params without reboot

sudo sysctl --system

### Step 5: Install kubelet, kubeadm, and kubectl on each node

### sudo apt-get update

### sudo apt-get install -y apt-transport-https ca-certificates curl gpg

### curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

### echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

### sudo apt-get update

### sudo apt-get install -y kubelet kubeadm kubectl

### sudo apt-mark hold kubelet kubeadm kubectl

### Step 6: Install Docker

### sudo apt install docker.io

### sudo mkdir /etc/containerd

### sudo sh -c "containerd config default > /etc/containerd/config.toml"

### sudo sed -i 's/ SystemdCgroup = false/ SystemdCgroup = true/' /etc/containerd/config.toml

### sudo systemctl restart containerd.service

### sudo systemctl restart kubelet.service

### sudo systemctl enable kubelet.service

### Step 7: Initialize the Kubernetes cluster on the master node

### On MASTER NODE:-

sudo kubeadm config images pull

### sudo kubeadm init --pod-network-cidr=10.10.0.0/16

mkdir -p $HOME/.kube

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

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

### Step 8: Configure kubectl and Calico

kubectl create -f <https://raw.githubusercontent.com/projectcalico/calico/v3.26.1/manifests/tigera-operator.yaml>

curl https://raw.githubusercontent.com/projectcalico/calico/v3.26.1/manifests/custom-resources.yaml -O

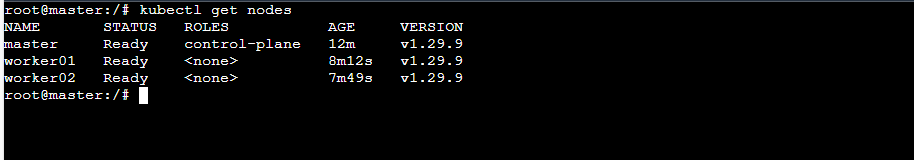
### sed -i 's/cidr: 192\.168\.0\.0\/16/cidr: 10.10.0.0\/16/g' custom-resources.yaml

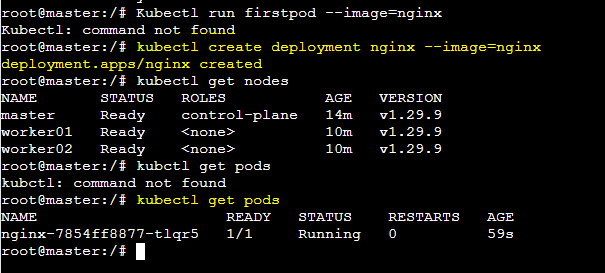
kubectl create -f custom-resources.yaml

### Step 9: Add worker nodes to the cluster

sudo kubeadm join &lt;MASTER\_NODE\_IP>:&lt;API\_SERVER\_PORT> --token &lt;TOKEN> --discovery-token-ca-cert-hash &lt;CERTIFICATE\_HASH>

### Step 10: Verify the cluster and test

kubectl get nodes  


3) Run one nginx pod.  


4) Mugup Master and slave components on k8s.  
Okay.