Docker and Kubernetes Installation Steps:

Docker installation

Please access Linux Virtual Machine assigned to you using IP address

```
ssh docker@ipaddress
```

Once logged in, please run

```
sudo -i
```

Step1:

# Add Docker's official GPG key:

```
sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc
```

# Add the repository to Apt sources:

```
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

Step3: Check if docker is up and running

```
systemctl status docker
docker ps -a
docker images
```

Kubernetes Installation

Step1:

Login to 1st VM and Setup Hostname

```
sudo -i
hostnamectl set-hostname master
bash
```

Login to 2nd VM and Setup Hostname

```
sudo -i
hostnamectl set-hostname worker1
bash
```

Login to 3rd VM and Setup Hostname

```
sudo -i
hostnamectl set-hostname worker2
bash
```

Step2: (Copy all lines in one go and Fire on all master and worker nodes)

# disable swap

```
sudo sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab
swapoff -a
```

Step3 : (Copy all lines in one go and Fire on all master and worker nodes)

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

Step4 : (Copy all lines in one go and Fire on all master and worker nodes)

```
sudo modprobe overlay
sudo modprobe br_netfilter
```

Step5 : (Copy all lines in one go and Fire on all master and worker nodes)

```
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
```

Step6 : (Copy all lines in one go and Fire on all master and worker nodes)

```
sudo sysctl --system
lsmod | grep br_netfilter
lsmod | grep overlay
sysctl net.bridge.bridge-nf-call-iptables net.bridge.bridge-nf-call-ip6tables net.ipv4.ip_forward
```

Step7 (Fire on master and all worker nodes-**line by line: Installation of container engine**)

```
apt update -y

sudo apt install -y curl gnupg2 software-properties-common apt-transport-https ca-certificates

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmour -o /etc/apt/trusted.gpg.d/docker.gpg

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

sudo apt update

sudo apt install -y containerd.io
systemctl enable containerd
systemctl restart containerd
systemctl status containerd.service

containerd config default | sudo tee /etc/containerd/config.toml >/dev/null 2>&1

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

systemctl restart containerd
sudo systemctl enable containerd
```

step8 (Fire on master and all worker nodes-**line by line – Installation of kubelet/kubeadm/kubectl**)

```
sudo apt-get update

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

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/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:/v1.30/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
sudo systemctl enable --now kubelet
```

```
 kubelet --version
 kubeadm version
kubectl version
```

Step9: (only on master node – Bootstrap Kubernetes control plane)

Please find out private Ip address of master machine and replace in second command mentioned below

```
kubeadm config images pull --cri-socket /run/containerd/containerd.sock --kubernetes-version v1.30.0

kubeadm init --pod-network-cidr=10.244.0.0/16 --upload-certs --kubernetes-version=v1.30.0 --control-plane-endpoint=privateIPofMasternode --ignore-preflight-errors=Mem --cri-socket /run/containerd/containerd.sock
```

Step10 : Only on Worker nodes : Joining workernodes to control plane

```
### From output of above Copy join(second one) command from the output of above command and paste in notepad and run it on all worker nodes (Please do not run this command on master node) .
```

Step11: (Run below commands only on master)

```
 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 nodes
```

Step12 (Only fire on master nodes – install calico CNI (Flannel + Canal)

```
kubectl create -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml

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

kubectl apply -f canal.yaml

#####and after few seconds  run below command

kubectl get nodes
```