**KUBERNETS INSTALLATION**

**STEP 1 🡪** Make 4 MC in **t2.medium** and in **Configure Security Group** add **All TCP.**

**STEP 2 🡪** Name all 4 MC like **Master Node, Worker 1, Worker 2, Worker 3.**

**STEP 3 🡪 NOTE:- There are different ports for Master Node and Worker Node you can check them by this link** [**https://kubernetes.io/docs/reference/ports-and-protocols/**](https://kubernetes.io/docs/reference/ports-and-protocols/) **But for now we allowed all ports.**

**STEP 4 🡪 Set all MC on Mobaxterm then** Set Hostname for all MC EX.- hostnamectl set-hostname master.example.com, Hostnamectl set-hostname worker**a**.example.com, Hostnamectl set-hostname worker**b**.example.com, Hostnamectl set-hostname worker**c**.example.com

**STEP 5 🡪 Run these commands on all 4 machines**

* **lsmod | grep br\_netfilte** {To check wether the br\_netfilter module is enabled or not}
* **modprobe br\_netfilter** {To enable this br\_netfilter module}
* cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf

br\_netfilter

EOF

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

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

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

EOF

sudo sysctl –system

* **lsmod | grep br\_netfilter** {To check wether the br\_netfilter module is is enabled or not}

**STEP 6 🡪 Installing Runtime** (To run containers in Pods, Kubernetes uses a [container runtime](https://kubernetes.io/docs/setup/production-environment/container-runtimes).)

Installing Runtime Means Install DOCKER.

**Run these commands on all 4 machines.**

* yum install -y yum-utils
* yum-config-manager \

--add-repo \

<https://download.docker.com/linux/centos/docker-ce.repo>

* yum install docker-ce -y
* systemctl enable –-now docker
* systemctl status docker

**STEP 7 🡪 Run these commands on all 4 machines.**

**Now make a file**

* vim /etc/docker/daemon.json

Now enter this data in this file

* {

“exec-opts”: [“native.cgroupdriver=systemd”]

}

* systemctl restart docker

## STEP 8 🡪 Installing kubeadm, kubelet and kubectl

**Run these commands on all 4 machines.**

* cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-\$basearch

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

exclude=kubelet kubeadm kubectl

EOF

*# Set SELinux in permissive mode (effectively disabling it)*

sudo setenforce 0

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

sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes

sudo systemctl enable --now kubelet

* systemctl status kubelet
* systemctl restart kubelet

**STEP 9 🡪** [Using kubeadm to Create a Cluster](https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/)

**Run these commands only on Master Node**

* kubeadm init --pod-network-cidr=172.25.0.0/16

**NOTE:- If you have any issue to run kubeadm init command, you can reset kubeadm by this command otherwise don’t run this command.**

* kubeadm reset

**STEP 10 🡪 Run these commands only on Worker Node**

**After running kubeadm init command you will get two line at the end that will looks like this**

**“kubeadm join 1.1.1.1:6443 -–token ………………………………………………………………\**

**--discover-token-ca-cert-hash Sha...............………………………………………………………………………”**

**Copy these two lines and paste as command and run.**

**STEP 11 🡪 Run these commands only on Master node**

**And run this command also that will you get after running kubeadm init command**

* export KUBECONFIG=/etc/Kubernetes/admin.conf

**STEP 12 🡪 Now we will configure network-plugin/Networking.**

**For that we will install a pod network add-on.**

**We have many networking models like.**

* **ACI**
* **ANTREA**
* **CALICO**
* **CILIUM**
* **FLANNEL (we will use this for now )**
* **JAGUAR**

**Run these commands only on Master node.**

* kubectl apply -f <https://raw.githubusercontent.com/flannel-io/flannel/master/Documentation/kube-flannel.yml>

**kubectl tab completion**

**First Method**

* kubectl completion bash | sudo tee /etc/bash\_completion.d/kubectl > /dev/null
* echo 'alias k=kubectl' >>~/.bashrc

echo 'complete -F \_\_start\_kubectl k' >>~/.bashrc

**Second Method(If first method don’t work)**

* **yum install bash-completion -y**

**Now you are ready to work on your cluster.**

**You can access your cluster from outside also. There is no necessity to work through only master node.**