**KUBERNETES**

**PreReq:** Master node with min- 2core CPU and 4GB RAM

Worker node with the same spec.

**Need to open the following ports:**

sudo firewall-cmd --zone=public --add-port=6443/tcp --permanent

sudo firewall-cmd --zone=public --add-port=10250/tcp --permanent

sudo firewall-cmd --zone=public --add-port=8001/tcp --permanent

sudo firewall-cmd --reload

**Installation:**

***Make a host entry in both master and slaves:***

vi /etc/hosts

10.10.34.242 kubemaster

10.10.32.250 kubenode

***DISABLE SWAP***

swapoff -a -> in both master and slaves

vi /etc/fstab and comment swap entry

***#MASTER***

***Disable SELINUX***

$ setenforce 0

$ sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux

***Open FIREWALL***

$ firewall-cmd --permanent --add-port=6443/tcp

$ firewall-cmd --permanent --add-port=8001/tcp

$ firewall-cmd --permanent --add-port=443/tcp

$ firewall-cmd --permanent --add-port=2379-2380/tcp

$ firewall-cmd --permanent --add-port=10250/tcp

$ firewall-cmd --permanent --add-port=10251/tcp

$ firewall-cmd --permanent --add-port=10252/tcp

$ firewall-cmd --permanent --add-port=10255/tcp

$ firewall-cmd --reload

***Enable br\_netfilter Kernel Module for cluster communication, packets traversing the bridge.***

$ modprobe br\_netfilter

$ echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables

***Configure Kubernetes Repository***

cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

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

EOF

$ yum install kubeadm docker -y

$ systemctl restart docker && systemctl enable docker

$ systemctl restart kubelet && systemctl enable kubelet

***Initialize Kubernetes Master***

$ kubeadm init

***Execute the command to use the cluster as root user***

$ mkdir -p $HOME/.kube

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

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

***Deploy Pod Network for internal communication***

$ export kubever=$(kubectl version | base64 | tr -d '\n')

$ kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$kubever"

***Verify the status***

$ kubectl get nodes

$kubectl get pods --all-namespaces

***Creating a Dashboard UI:***

For creating dashboard

$ kubectl create –f <https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml>

To access the dashboard with full administrative permission, create a YAML file named dashboard-admin.yaml.

$ vi dashboard-admin.yaml and add the below content in it.

apiVersion: rbac.authorization.k8s.io/v1beta1

kind: ClusterRoleBinding

metadata:

name: kubernetes-dashboard

labels:

k8s-app: kubernetes-dashboard

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: cluster-admin

subjects:

- kind: ServiceAccount

name: kubernetes-dashboard

namespace: kube-system

After adding the content, run the following command to create a dashboard $ kubectl create –f dashboard-admin.yaml

***To enable Proxy***

$ nohup kubectl proxy --address="10.10.34.242" -p 8001 --accept-hosts='^\*$' &

This will be running in the background. If you want to kill the process $ ps –a and then $ kill -9 <pid>

Now, Access the dashboard using the following URL

<http://10.10.34.242:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/>

We can able to access the dashboard through the kubeconfig file or bearer token. We have already provided full admin access to dashboard service account. So just click on SKIP option to access the dashboard.

***#WORKER***

***Disable SELINUX***

$ setenforce 0

$ sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux

***Open FIREWALL***

$ firewall-cmd --permanent --add-port=10250/tcp

$ firewall-cmd --permanent --add-port=10255/tcp

$ firewall-cmd --permanent --add-port=30000-32767/tcp

$ firewall-cmd --permanent --add-port=6783/tcp

$ firewall-cmd --reload

***Enable br\_netfilter Kernel Module for cluster communication, packets traversing the bridge.***

$ modprobe br\_netfilter

$ echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables

***Configure Kubernetes Repository***

cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

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

EOF

$ yum install kubeadm docker -y

$ systemctl restart docker && systemctl enable docker

$ systemctl restart kubelet && systemctl enable kubelet

Get the join command from master and run the command in worker to get the node connect to cluster

DEPLOYING JENKINS:

TYPE: NODEPORT

$ kubectl create deployment jenkins --image=jenkins

$ kubectl create service nodeport jenkins --tcp=8080:8080 --node-port=30000

And access the service with node-ip:30000