Kubernetes Lab

Set hostname on three machines

\#   hostnamectl set-hostname master/node1/node2

\#   Edit the IP for ping in /etc/hosts.

\#   ssh key-gen

\#   ssh-copy-id root@node1/node2

\#   yum update

\#   systemctl disable firewalld

\#   nano /etc/selinux/config   set enforcing to disabled

\#   swapoff –a   Swapoff **disables swapping on the specified devices and files**.
    When the -a flag is given, swapping is disabled on all known swap devices and
    files (as found in /proc/swaps or /etc/fstab).

\#   sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab   you can do it in single command.

\#   modprobe overlay

\#   modprobe br_netfilter

\#   tee /etc/sysctl.d/kubernetes.conf<<EOF
    > net.bridge.bridge-nf-call-ip6tables = 1
    > net.bridge.bridge-nf-call-iptables = 1
    > net.ipv4.ip_forward = 1
    > EOF

\#   sysctl –system

\#   tee /etc/yum.repos.d/kubernetes.repo<<EOF
    > [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 clean all && yum -y makecache

\#   yum -y install epel-release git curl wget kubelet kubeadm kubectl
    --disableexcludes=kubernetes

\#   systemctl start kubelet

\#   systemctl enable kubelet

\#   OS=CentOS_7

\#   VERSION=1.22

curl -L -o /etc/yum.repos.d/devel:kubic:libcontainers:stable.repo https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/$OS/devel:kubic:libcontainers:stable.repo

curl -L -o /etc/yum.repos.d/devel:kubic:libcontainers:stable:cri-o:$VERSION.repo https://download.opensuse.org/repositories/devel:kubic:libcontainers:stable:cri-o:$VERSION/$OS/devel:kubic:libcontainers:stable:cri-o:$VERSION.repo

yum remove docker-ce docker-ce-cli containerd.io

yum install cri-o –y

systemctl daemon-reload

systemctl start crio

systemctl enable crio

++++++++++++++++++++++++++On Master++++++++++++++++++++++++++

lsmod | grep br_netfilter

kubeadm config images pull

kubeadm init --pod-network-cidr=10.85.0.0/16 --upload-certs --control-plane-endpoint=master   ip assigned to pod and it generated a token which is used to join any number of worker nodes.

nano join-token   paste the token into this file.

Certificate Signing Request(CSR) is **a block encrypted text which is given to Certificate Authority when applying for SSL Certificate**. Generation of Certificate Signing Request(CSR) for Secure Sockets Layer(SSL) is common in Linux on various distributions.

RBAC rules
Create the Boostrap Token to be used by Nodes (Kubelets) to invoke Certificate API

----

chmod u+x join-token
rsync join-token root@node1:/root/
rsync join-token root@node2:/root/
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   tells the nodes information
In node1 and node2
./join-token
In master run
kubectl get nodes   YOU ARE ABLE TO SEE THE ALL NODES

Install network plug-in   In this guide we'll use [Calico](). You can choose any other [supported network plugins]().

\# kubectl create -f https://docs.projectcalico.org/manifests/tigera-operator.yaml

\# kubectl create -f https://docs.projectcalico.org/manifests/custom-resources.yaml

\# kubectl get pods --all-namespaces

\# kubectl get pods --all-namespaces -o wide