###Perform the following steps on master

Note: Please change Node IP and Gateway as per your configuration

Step 0: IP and Hostname

nmcli connection modify ens33 ipv4.method manual ipv4.addresses 192.168.32.134/24 ipv4.gateway 192.168.32.2 ipv4.dns 8.8.8.8 ipv6.method ignore

nmcli connection up ens33

hostnamectl set-hostname manager

cat >> /etc/hosts <<EOF

192.168.32.134 manager

192.168.32.135 node1

192.168.32.136 node2

EOF

Step 1: Disable SELinux & setup firewall rules

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

setenforce 0

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

Step 2: Enable and start NTP

systemctl enable chronyd

systemctl start chronyd

systemctl status chronyd

chronyc sources

Step 3: Load Bridge Module

modprobe br\_netfilter

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

cat > /etc/sysctl.d/k8s.conf <<EOF

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

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

EOF

sysctl --system

Step 4: Turn swap off

swapoff -a

sed -e '/swap/ s/^#\*/#/' -i /etc/fstab

Step 5: Install base Packages and update the server with latest packages

yum -y install wget git net-tools bind-utils bridge-utils bash-completion kexec-tools

yum update -y

reboot

Step 6: Configure Kubernetes Repository

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

[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

EOF

Step 7: Install Kubeadm and Docker

yum install kubeadm docker -y

systemctl restart docker && systemctl enable docker

systemctl restart kubelet && systemctl enable kubelet

Step 8: Initialize Kubernetes Master with ‘kubeadm init’

Note: Do not forget to increase CPU count 2.

kubeadm init --pod-network-cidr 10.244.0.0/24 --apiserver-advertise-address=192.168.32.134

Note: Copy “kubeadm join” command from the output.

systemctl status kubelet

mkdir -p $HOME/.kube

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

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

modprobe openvswitch

sed -i '$ s/$/ --runtime-cgroups=\/systemd\/system.slice --kubelet-cgroups=\/systemd\/system.slice/' /etc/systemd/system/kubelet.service.d/10-kubeadm.conf

sed -e '/insecure-port/s/^/#/g' -i /etc/kubernetes/manifests/kube-apiserver.yaml

systemctl daemon-reload

systemctl restart kubelet

systemctl status kubelet

Step 5: Deploy pod network to the cluster

kubectl get pods -n kube-system

Note: Wait for 5 min for all services to be up.

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/bc79dd1505b0c8681ece4de4c0d86c5cd2643275/Documentation/kube-flannel.yml

kubectl get nodes

TIP:

In case if you forget to make a note of kubeadm join information, use the below command from the master server to retrieve the join information.

kubeadm token create --print-join-command