**Setting up a highly available Kubernetes cluster**

Disable the swap on both the master and worker nodes to ensure that the setup is successful.

**System updates**

First I have to update the Linux system with all the security patches and ensure my system is up-to date. Updated the system using the following command:

sudo yum update -y

**Install and Set Up the Master and Worker Nodes**

Before setting up the master and the worker, I have to install Docker.

**Install and Set Up Docker**

By executing this command Docker will be installed:

sudo yum install -y docker

Now to enable and start Docker as a service we use the following command.

sudo systemctl enable docker && sudo systemctl start docker

To verify the Docker version running execute this command.

sudo docker version

Install Kubernetes Clusters

To get the latest packages for Kubernetes we need to configure our yum repository.

sudo bash -c '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>

exclude=kube\*

EOF'

Disable SELinux to prevent communication between the nodes.

sudo setenforce 0

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

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

After the installation is completed, let’s enable the kubelet as a service.

sudo systemctl enable kubelet && sudo systemctl start kubelet

Master Node Set Up

Allow 6443 and 10250 from firewalld on master node

sudo firewall-cmd --permanent --add-port=6443/tcp && sudo firewall-cmd --permanent --add-port=10250/tcp && sudo firewall-cmd --reload

Set virtual IP between the nodes.

sudo bash -c 'cat <<EOF > /etc/sysctl.d/k8s.conf

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

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

EOF'

Apply the changes using the following command:

sudo sysctl --system

Load br\_netfilter module

sudo lsmod | grep br\_netfilter

**Configure Kubernetes Master Node**

Now that the installation is complete, it’s high time we do the configuration. We need to get the all the images that will be needed for Kubernetes initialization.

sudo kubeadm config images pull

After all the images are pulled, let’s get started with the cluster setup.

sudo kubeadm init --pod-network-cidr=10.244.0.0/16

Adding cluster settings to regular user to be able to access Kubernetes cluster locally.

mkdir -p $HOME/.kube

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

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

Applying Network setting to the nodes

kubectl apply -f <https://raw.githubusercontent.com/coreos/flannel/a70459be0084506e4ec919aa1c114638878db11b/Documentation/kube-flannel.yml>

**Worker Node Set Up**

We have to apply these steps to our two worker nodes. Use the information received on the master node.

sudo kubeadm join

To check if nodes have been joined, we use this:

kubectl get nodes

The output should have the status and roles of the node.