## Step 1 – Install Docker Runtime (on 3 servers/machines/VMs)

Note: These commands are for RHEL 7 only. If you are using any other operating system then check the documentation pertaining to that OS to get the commands to install docker, kubeadm, kubelet and kubectl

```
sudo yum -y upgrade
sudo yum install -y epel-release
sudo yum install -y yum-utils device-mapper-persistent-data-lvm2
sudo yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
sudo yum install -y docker-ce
```

## Step 2 – Install Kubeadm, Kubelet, Kubectl **(on all Nodes)**

- Edit the file "/etc/yum.repos.d/kubernetes.repo". Copy and paste the following lines (better to type the lines)

```
[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*
```

- Set SELinux in permissive mode and turn swap off. Run the following commands

```
sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/'
/etc/selinux/config
sudo yum install -y kubeadm kubelet kubectl --disableexcludes=kubernetes
sudo systemctl enable --now kubelet
sudo swapoff -a
```

- Install kubeadm, kubelet and kubectl and enable kubelet

```
sudo yum install -y kubeadm kubelet kubectl --disableexcludes=Kubernetes
sudo systemctl enable --now kubelet
```

- Following steps are needed only for RHEL7. Edit the file /etc/sysctl.d/k8s.conf and paste the following lines

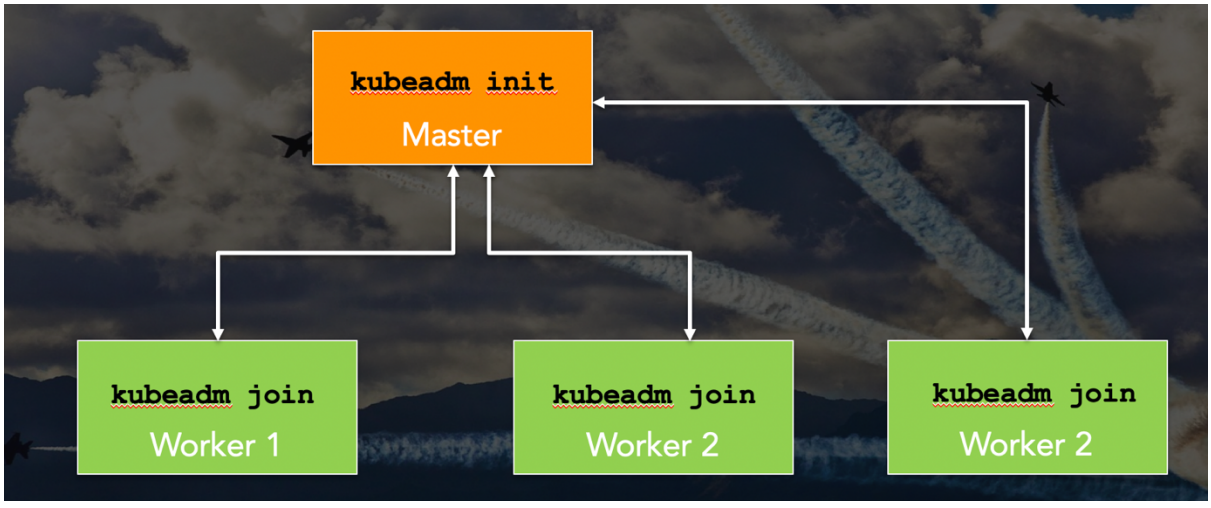
  sudo vi /etc/sysctl.d/k8s.conf

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

```
** Run the following commands **
sudo sysctl --system
sudo modprobe br_netfilter
lsmod | grep br_netfilter          ##(this should give you some output)
```

## Step 3 – Set up a cluster using kubeadm

We will create cluster with 2 worker nodes as below.




Start "Docker" service on all the nodes –
**sudo systemctl start docker.service**

Check if Docker service is running or not
**sudo service docker status**

Run commands as mentioned below:

1. On MASTER Node > **`sudo kubeadm init --pod-network-cidr 10.168.0.0/16`**

2. On MASTER Node > This is for Pod network plugin. We will be using Flannel for this. Don't worry too much about it. Run the command –

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

3. On WORKER Nodes > Whatever output you get with "**`sudo kubeadm join`** …"

4. To check the status of the nodes run "**`kubectl get nodes`**" on master node.

## Step 4 – Optional Troubleshooting

While running "Kubectl get nodes" command if you get the following error

```
The connection to the server localhost:8080 was refused - did you specify the right host or port?
```

Then run the following commands in sequence. These commands are enabling you to work as a K8S admin.

```
sudo cp /etc/kubernetes/admin.conf $HOME/
sudo chown $(id -u):$(id -g) $HOME/admin.conf
export KUBECONFIG=$HOME/admin.conf
```

NOTE: In case your Kubernetes cluster does not come up after restarting EC2 instances or Google Compute Engine instances and kubectl command gives error, follow the steps below to fix the issue:

- Start Docker Service on all nodes
- Check if kubectl (kubectl get nodes) command works or not. If it DOES NOT then only follow the steps below.

- Disable swap
- Enable kubelet service on all nodes
- Try kubectl command again