Steps to Install Kubernetes CLuster using KUBEADM :

1)Min 3 VM's with 2core CPU & 2GB of ram or above

2)Ensure to Disable Swap, SE Linux and Firewall on all the nodes (Master & Worker Node)

3)Ensure DOcker Runtime installed & Docker service running on all the nodes (Master & Worker Node)

4)Ensure Kubeadm, Kubelet, Kubectl Installed on all the nodes (Master&Worker Node)

5)Start & Enable Kubelet Service on all the Nodes (Master & Worker Node)

6)Ensure IP FOrwarding is done on all the nodes (Master & Worker Node)

7)Initialize the Master Node (ON Master Only)

8)Configure the POD Network using flannel Network (On Master Node Only)

9)Join the worker nodes to Kubernetes Cluster

10)Now our CLuster is ready to launch an Application.

Follow the Below steps on all the Nodes (Master & Worker Node) :

#swapon -s **(To display the swap space)**

#swapoff -a **(Disables all the swap devices, But it is temporary for the session)**

NOTE : Make it permant by adding a comment to swap partition in /etc/fstab file

#getenforce **(Displays SE Linux State)**

#setenforce 0 **(Changing the SE Linux state from Enforcing to Permissive)**

#getenforce

NOTE : Go to /etc/selinux/config file and disable it :

```
#vim /etc/selinux/config
SeLinux=Enforcing ( Replace this with Disabled )
:wq!
```

NOTE : To reflect the changes Reboot the Server

```
#reboot
#swapon -s
#getenforce
```

Now we are disbaling Firewall Permanently :

```
#systemctl stop firewalld

#systemctl disable firewalld

#systemctl status firewalld
```

Installing Docker :

```
#yum install -y yum-utils

#yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

#yum-config-manager --enable docker-ce-nightly

#yum install docker-ce docker-ce-cli containerd.io

#which docker

#docker version

#systemctl status docker

#systemctl start docker && systemctl enable docker

#systemctl status docker
```

Installing Kubeadm, Kubectl and Kubelet :

Setting up Kubernetes Repository :

```
#vim /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
:wq!
```

```
#yum clean all ( Clearing the Cache )
```

```
#yum install kubeadm kubectl kubelet -y

#systemctl start kubelet

#systemctl enable kubelet

#systemctl status kubelet
```

Configuring IP Forwarding :

```
#vim /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
:wq!
```

```
#systemctl daemon-reload

#systemctl restart kubelet

#sysctl –system

#sysctl net.bridge.bridge-nf-call-iptables=1
```

```
#sysctl net.ipv4.ip_forward=1
```

```
#sysctl –system
```

```
#echo "1" /proc/sys/net/ipv4/ip_forward
```

Now intializing Master node (In my Ex my Machine iP - 192.168.45.151) Execute on only the machine you wanted to configure on Master Node :

```
#kubeadm init --pod-network-cidr=10.240.0.0/16
```

Configuring POD Network (On Master Node Only) :

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

Verify POD Network :

```
#kubectl get pods --all-namespaces
```

`#kubectl get nodes` **(Verifying the Nodes in Cluster)**

Creating New token on Master:

```
#kubeadm token create --print-join-command
```

Listing the Available tokens:

```
#kubeadm token list
```

Deleting the tokens:

```
#kubeadm token delete <tokenID>
```

Now go to Worker Nodes and add them to Kubernetes CLuster by executing the below Command :

```
#kubeadm join 192.168.45.51:6443 --token u19xca.pheytekpee3d1x0k --discovery-token-ca-cert-hash sha256:230551f5766604a0a3a2d7a682a330c605039787b9e9524f658e6336be6947b3
```

(This is FYI, DO not copy and paste the same command. You use the token generated on master. Using that join your client machine as worker node to the cluster)

Once all the WOrker Nodes are addes to cluster execute the below command on Master Node :

```
#kubectl get nodes
```