Pre-requisites To Install Kubernetes

Master:

- 2 GB RAM
- 2 Cores of CPU

Slave/ Node:

- 1 GB RAM
- 1 Core of CPU

Pre-Installation Steps On Both Master & Slave (To Install Kubernetes)

The following steps have to be executed on both the master and node machines

$ sudo su
# apt-get update

Turn Off Swap Space

swapoff -a
nano /etc/fstab
If swap is enabled just # it and exit

Update The Hostnames

To change the hostname of both machines, run the below command to open the file and subsequently rename the master machine to ‘kmaster’ and your node machine to ‘knode’.

Install OpenSSH-Server

Now we have to install openshh-server. Run the following command:
sudo apt-get install openssh-server

Install Docker

Now we have to install Docker because Docker images will be used for managing the containers in the cluster. Run the following commands:
sudo su
apt-get update
apt-get install -y docker.io

Next we have to install these 3 essential components for setting up Kubernetes environment: kubeadm, kubectl, and kubelet.
Run the following commands before installing the Kubernetes environment.

apt-get update && apt-get install -y apt-transport-https curl
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
deb http://apt.kubernetes.io/ kubernetes-xenial main
EOF
apt-get update

Install kubeadm, Kubelet And Kubectl

Now its time to install the 3 essential components. ***Kubelet*** is the lowest level component in Kubernetes. It's responsible for what's running on an individual machine. ***Kuebadm*** is used for administrating the Kubernetes cluster. ***Kubectl*** is used for controlling the configurations on various nodes inside the cluster.

apt-get install -y kubelet kubeadm kubectl

Updating Kubernetes Configuration

Next, we will change the configuration file of Kubernetes. Run the following command:

nano /etc/systemd/system/kubelet.service.d/10-kubeadm.conf

This will open a text editor, enter the following line after the last “Environment Variable”:

Environment=”cgroup-driver=systemd/cgroup-driver=cgroupfs”

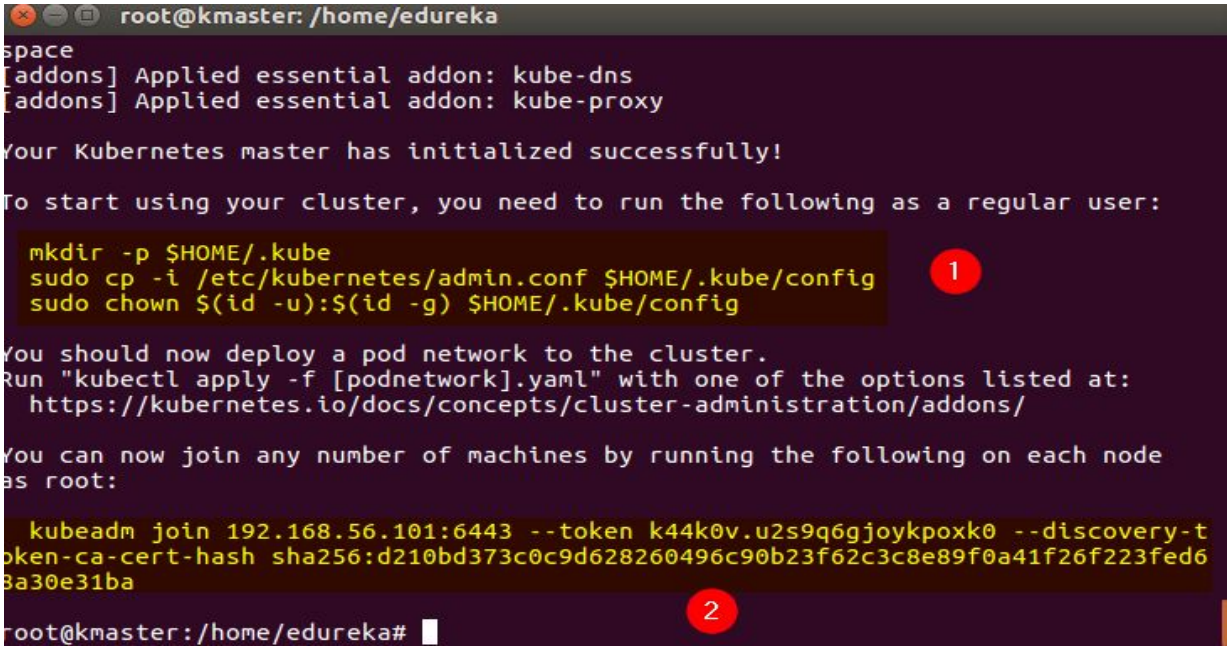
As of now, only the Kubernetes environment has been setup. But now, it is time to install Kubernetes completely, by moving onto the next 2 phases, where we will individually set the configurations in both machines.

Steps Only For Kubernetes Master VM (kmaster)

Note: These steps will only be executed on the master node (kmaster VM).
Step 1: We will now start our Kubernetes cluster from the master’s machine. Run the following command:

kubeadm init --apiserver-advertise-address=<ip-address-of-kmaster-vm> --pod-network-cidr=192.168.0.0/16

You will get the below output. The commands marked as (1), execute them as a non-root user. This will enable you to use kubectl from the CLI
The command marked as (2) should also be saved for future. This will be used to join nodes to your cluster



Step 2: As mentioned before, run the commands from the above output as a non-root user

```
$ mkdir -p $HOME/.kube
$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

To verify, if kubectl is working or not, run the following command:

```
$ kubectl get pods -o wide --all-namespaces
```

Step 3: You will notice from the previous command, that all the pods are running except one: 'kube-dns'. For resolving this we will install a pod network. To install the CALICO pod network, run the following command:

```
$ kubectl apply -f
https://docs.projectcalico.org/v3.0/getting-started/kubernetes/installation/hosted/kubeadm/1.7/calico.yaml
```

After some time, you will notice that all pods shift to the running state

Step 4: Next, we will install the dashboard. To install the Dashboard, run the following command:

$ kubectl create -f https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml

Step 5: Your dashboard is now ready with it's the pod in the running state.

Step 6: By default dashboard will not be visible on the Master VM. Run the following command in the command line:

```
$ kubectl proxy
```

To view the dashboard in the browser, navigate to the following address in the browser of your Master VM:
http://<mater-ip>:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/
You will then be prompted with this page, to enter the credentials:

Step 7: In this step, we will create the service account for the dashboard and get it's credentials.
Note: Run all these commands in a new terminal, or your kubectl proxy command will stop.
Run the following commands:
1. This command will create a service account for dashboard in the default namespace

```
$ kubectl create serviceaccount dashboard -n default
```

2. This command will add the cluster binding rules to your dashboard account

```
$ kubectl create clusterrolebinding dashboard-admin -n default \
  --clusterrole=cluster-admin \
  --serviceaccount=default:dashboard
```

3. This command will give you the token required for your dashboard login

$ kubectl get secret $(kubectl get serviceaccount dashboard -o jsonpath="{.secrets[0].name}") -o jsonpath="{.data.token}" | base64 --decode

4. Copy this token and paste it in Dashboard Login Page, by selecting token option

Steps For Only Kubernetes Node VM (knode)

It is time to get your node, to join the cluster! This is probably the only step that you will be doing on the node, after installing kubernetes on it.
Run the join command that you saved, when you ran 'kubeadm init' command on the master.

Note: Run this command with “sudo”.

sudo kubeadm join --apiserver-advertise-address=<ip-address-of-the master>
--pod-network-cidr=192.168.0.0/16