**What is Kubernetes?**

Kubernetes is a free and open-source container management system that provides a platform for deployment automation, scaling, and operations of application containers across clusters of host computers. With Kubernetes, you can freely make use of the hybrid,on-premise, and public cloud infrastructure in order to run deployment tasks of your organization.

**Pre-requisites To Install Kubernetes**

Since we are dealing with VMs, we recommend the following settings for the VMs:-

*Master*:

* 1 GB RAM
* 1 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. Let’s call the the master as ‘kmaster‘ and node as ‘knode‘.

First, login as ‘sudo’ user because the following set of commands need to be executed with ‘sudo’ permissions. Then, update your ‘apt-get’ repository.

$ sudo su

# apt-get update

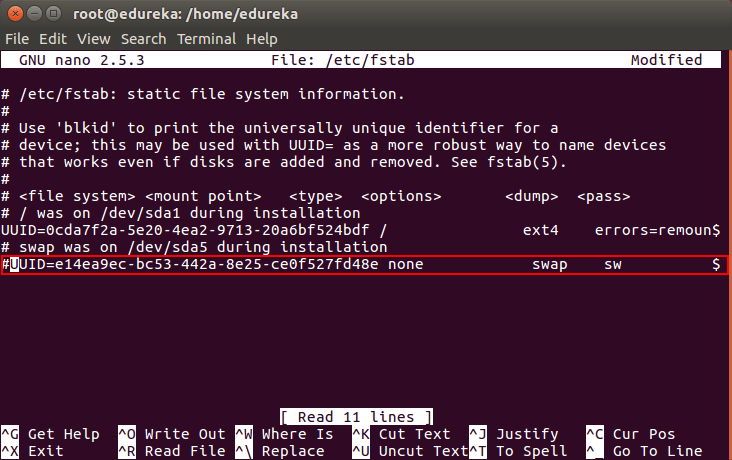
**Note**: After logging-in as ‘sudo’ user, note that your shell symbol will change to ‘#’ from ‘$’.

### **Turn Off Swap Space**

Next, we have to turn off the swap space because Kubernetes will start throwing random errors otherwise. After that you need to open the ‘fstab’ file and comment out the line which has mention of swap partition.

# swapoff -a

# nano /etc/fstab

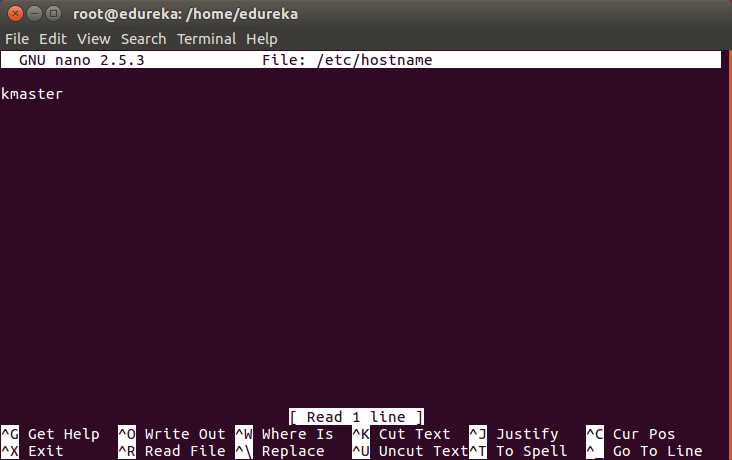


Then press ‘Ctrl+X’, then press ‘Y’ and then press ‘Enter’ to Save the file.

### **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’.

# nano /etc/hostname



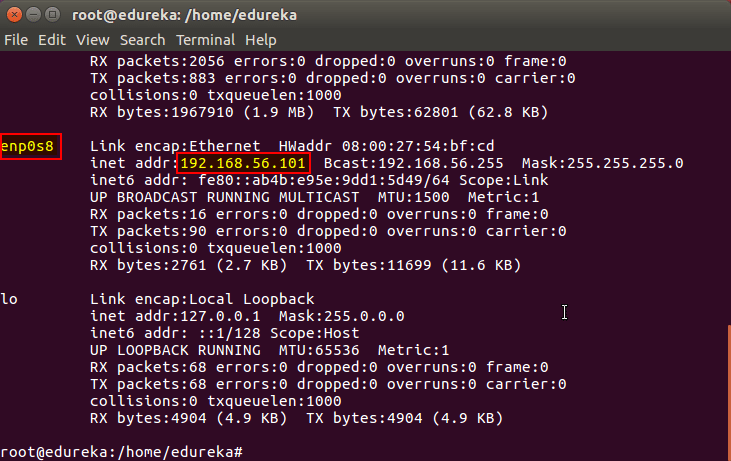
Then press ‘Ctrl+X’, then press ‘Y’ and then press ‘Enter’ to Save the file.

### **Update The Hosts File With IPs Of Master & Node**

Run the following command on both machines to note the IP addresses of each.

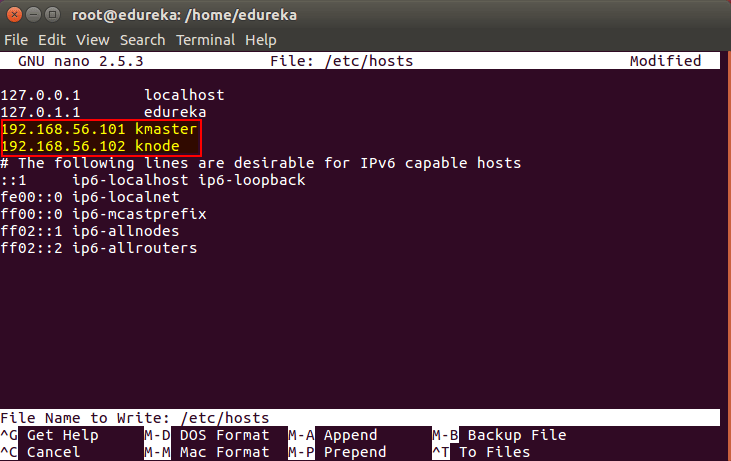
# ifconfig

Make a note of the IP address from the output of the above command. The IP address which has to be copied should be under “enp0s8”, as shown in the screenshot below.



Now go to the ‘hosts’ file on both the master and node and add an entry specifying their respective IP addresses along with their names ‘kmaster’ and ‘knode’. This is used for referencing them in the cluster. It should look like the below screenshot on both the machines.

# nano /etc/hosts



Then press ‘Ctrl+X’, then press ‘Y’ and then press ‘Enter’ to Save the file.

### **Setting Static IP Addresses**

Next, we will make the IP addresses used above, static for the VMs. We can do that by modifying the network interfaces file. Run the following command to open the file:

# nano /etc/network/interfaces

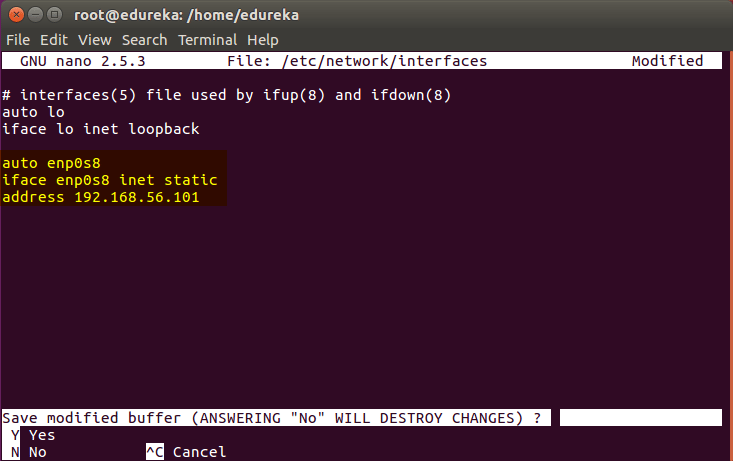
Now enter the following lines in the file.

auto enp0s8

iface enp0s8 inet static

address <IP-Address-Of-VM>

It will look something like the below screenshot.



Then press ‘Ctrl+X’, then press ‘Y’ and then press ‘Enter’ to Save the file.

After this, restart your machine(s).

### **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

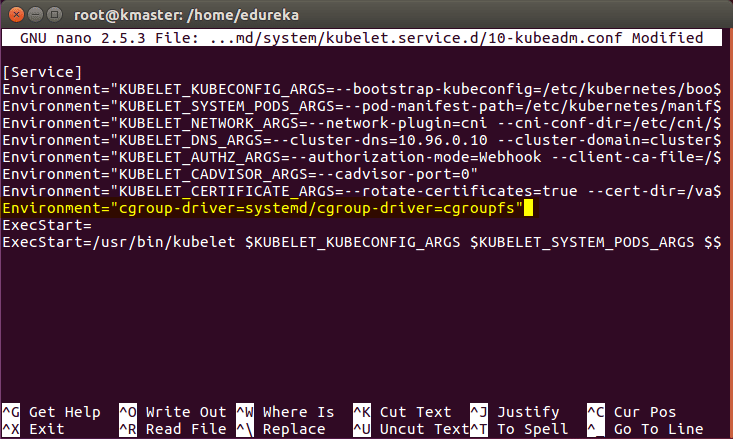
### **Updating Kubernetes Configuration**

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”



Now press Ctrl+X, then press Y, and then press Enter to Save.

***Voila!*** You have successfully installed Kubernetes on both the machines now!

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.

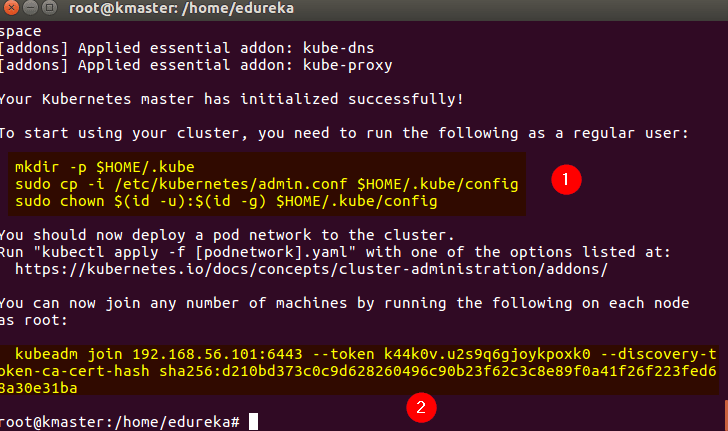
## ****Steps Only For Kubernetes Master VM (kmaster)****

***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

1. 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
2. The command marked as (2) should also be saved for future. This will be used to join nodes to your cluster

****

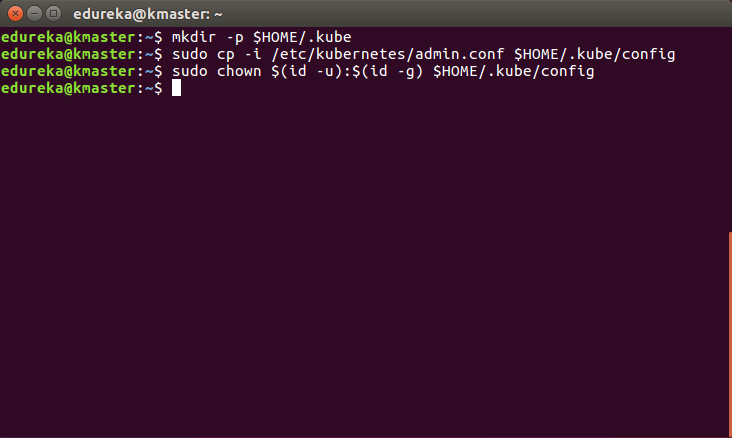
**Step 2**: As mentioned before, run the commands from the above output as a non-root user

$ mkdir -p $HOME/.kube

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

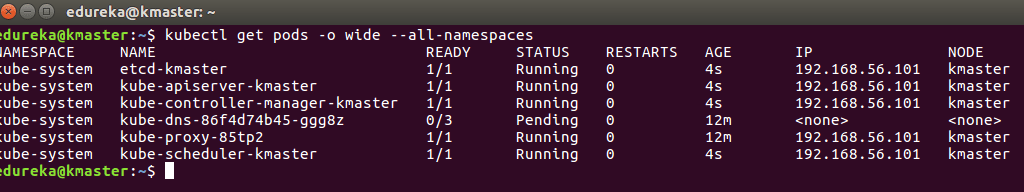
$ sudo chown $(id -u):$(id -g) $HOME/.kube/config

It should look like this:



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

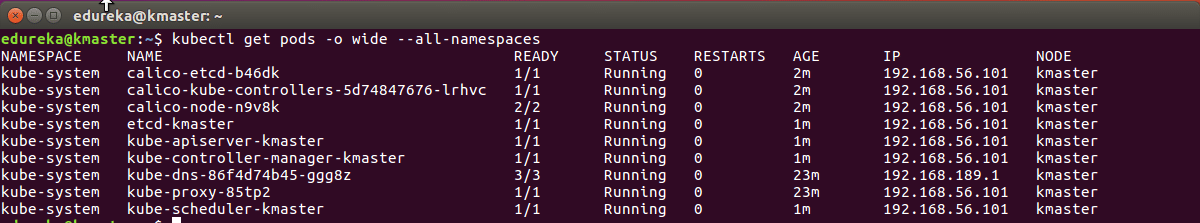
$ kubectl get pods -o wide --all-namespaces

****

**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/ku

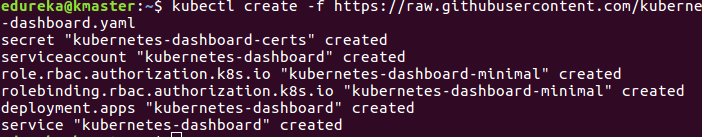
After some time, you will notice that all pods shift to the running state

****

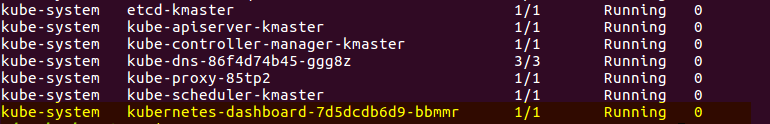
**Step 4**: Next, we will install the dashboard. To install the Dashboard, run the following command:

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

It will look something like this:

****

**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:

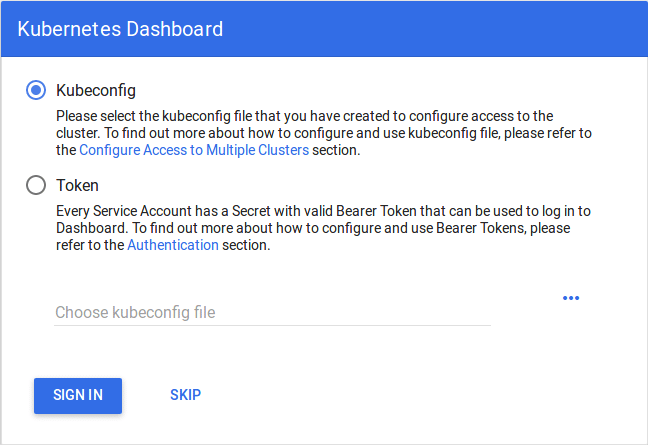
$ kubectl proxy

Then you will get something like this:

kubectl proxy - install kubernetes - edureka

To view the dashboard in the browser, navigate to the following address in the browser of your Master VM: http://localhost: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

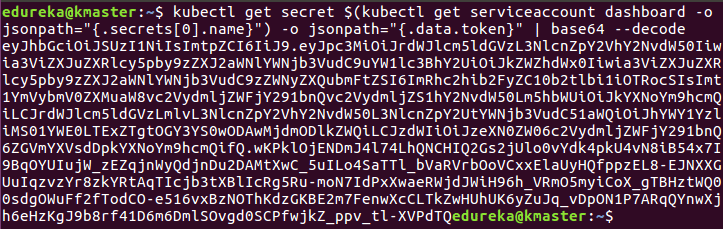
--clusterrole=cluster-admin

--serviceaccount=default:dashboard

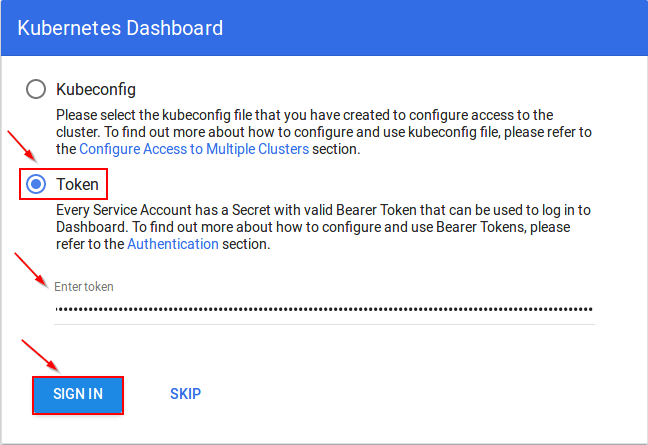
3. This command will give you the token required for your dashboard login

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

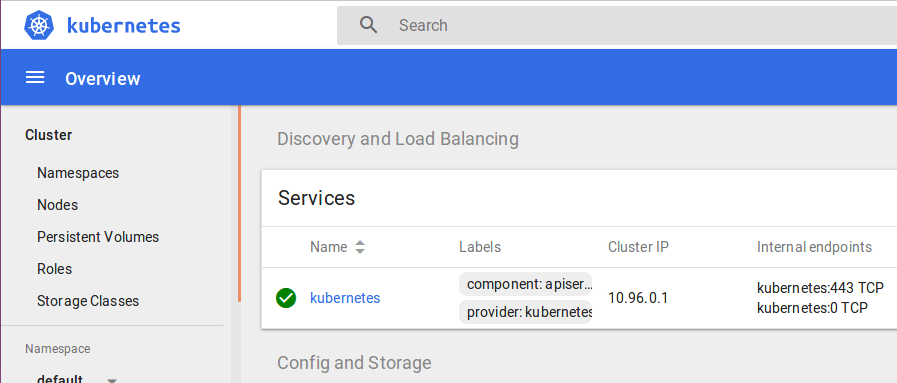
You should get the token like this:



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



5. You have successfully logged into your dashboard!

****

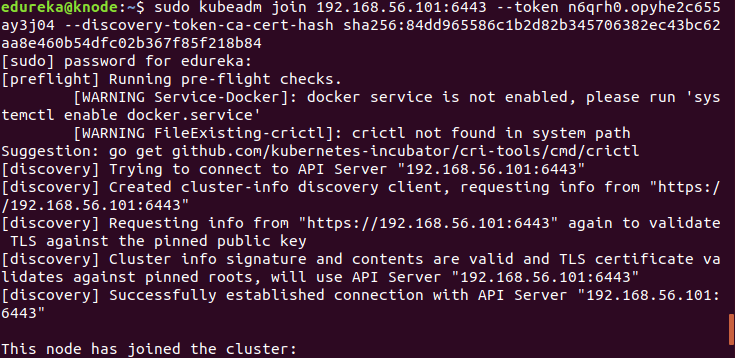
## ****Steps For Only Kubernetes Node VM (knode)****

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

******

***Bingo!*** Your Kubernetes Cluster is ready if you get something similar to the above screenshot.