## Semester: V Name of Student:

**Academic Year: 2021-22 Student ID: Class / Branch: TE IT**

## Subject: Advanced Devops Lab (ADL) Name of Instructor: Prof. Manasi Choche



**EXPERIMENT NO. 03**

**Aim: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.**

## Kubernetes Cluster on Linux Machines

**Step:1**

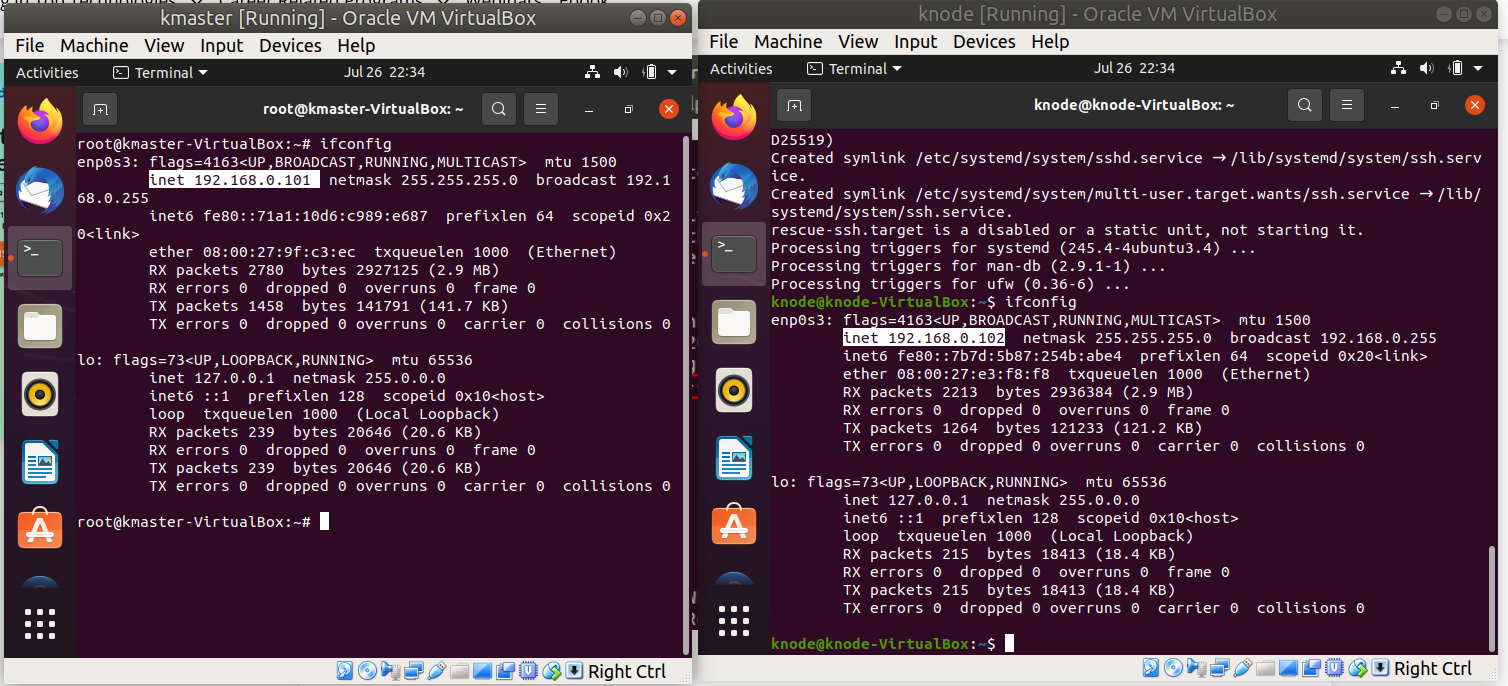
In Order to configure Kubernetes Cluster we require 2 systems we can consider it as one master and one slave.

I have launch two Virtual machines master as kmaster and slave as knode as shown in Fig.

Configure static IP as:

**Master: 192.168.0.101 (kmaster)**

**Slave: 192.168.0.102 (knode)**

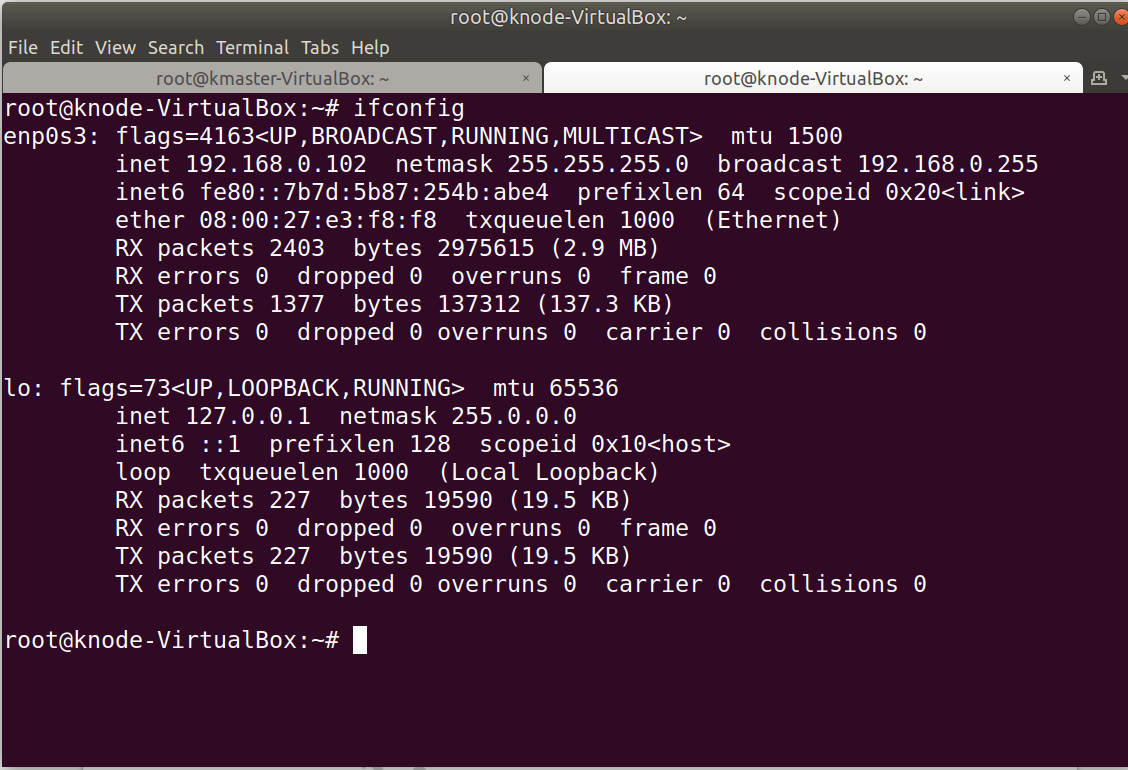
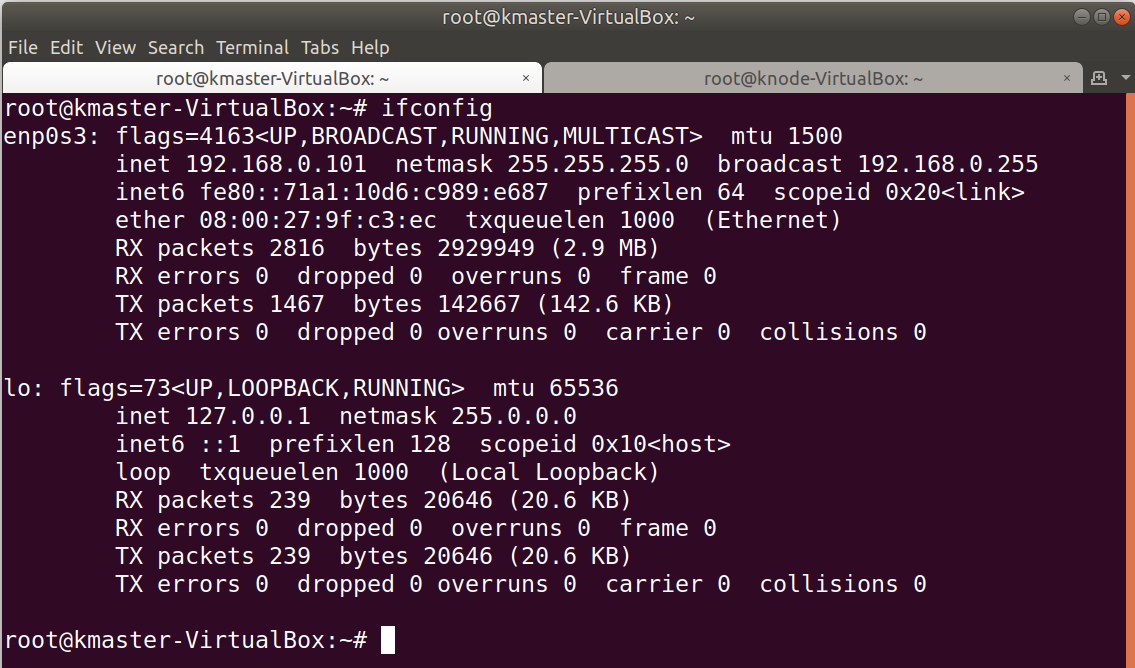


**Master: 192.168.0.101 (kmaster) Slave: 192.168.0.102 (knode)**

### Step 2: Install OpenSSH-Server (In Master and Slave)

Now we have to install openshh-server. Run the following command:

**#sudo apt-get install openssh-server**



**For easy to connect and perform I have taken access of VMs on my machine for further steps.**

### Step:3 Install Docker (In Master and Slave)

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**





**Step: 4** Next we have to install these 3 essential components for setting up Kubernetes environment: kubeadm, kubectl, and kubelet. (In Master and Slave)

Run the following commands before installing the Kubernetes environment.





[**root@kmaster-VirtualBox**](about:blank)**:~# curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -**

[**root@kmaster-VirtualBox**](about:blank)**:~# cat <<EOF >/etc/apt/sources.list.d/kubernetes.list**

**>deb http://apt.kubernetes.io/ kubernetes-xenial main**

**>EOF**

[**root@kmaster-VirtualBox**](about:blank)**:~# apt-get update**

**Step 5: Install kubeadm, Kubelet And Kubectl (In Master and Slave)**

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.

**root@kmaster-VirtualBox:~# apt-get install -y kubelet kubeadm kubectl**

**root@knode-VirtualBox:~# apt-get install -y kubelet kubeadm kubectl**

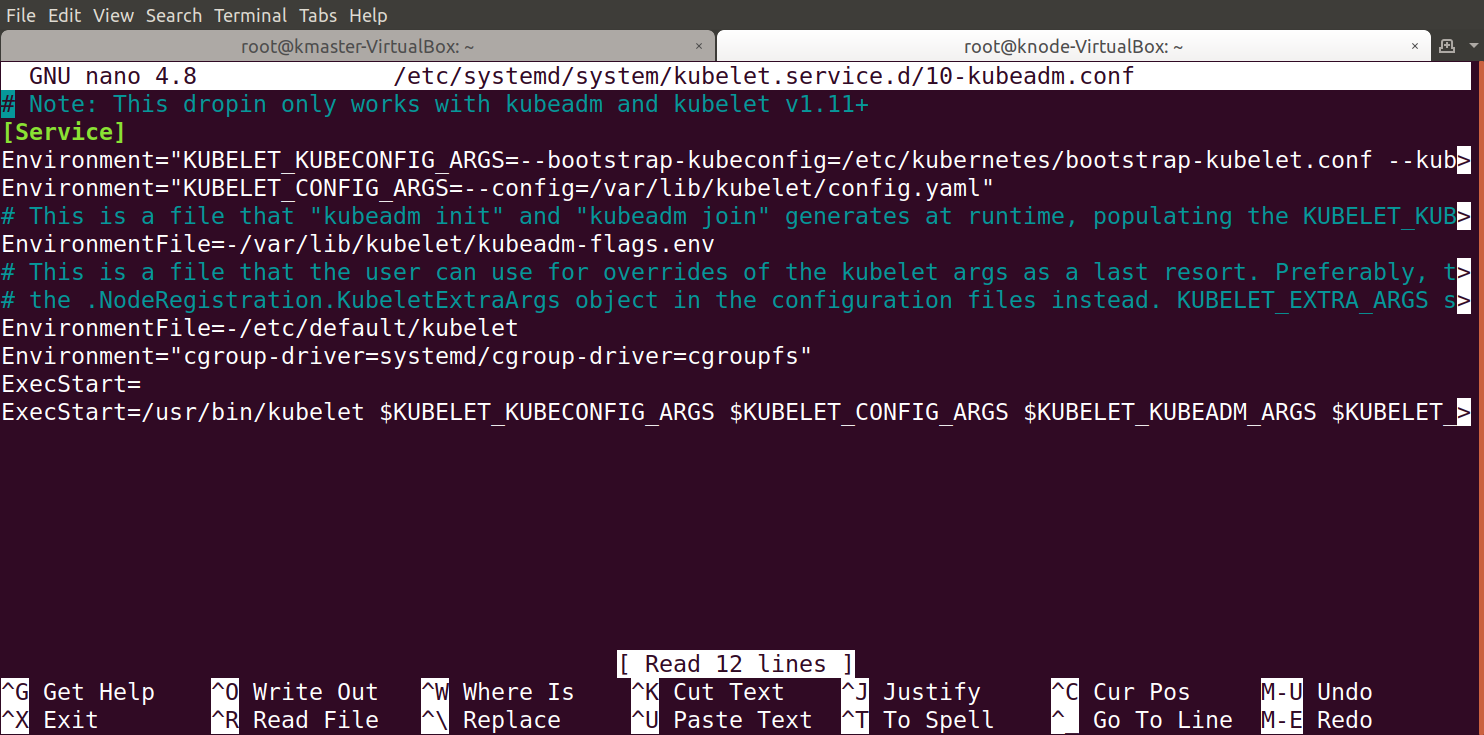
**Step 6: Updating Kubernetes Configuration (In Master and Slave)**

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

[**root@knode-VirtualBox**](about:blank)**:~# 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”**



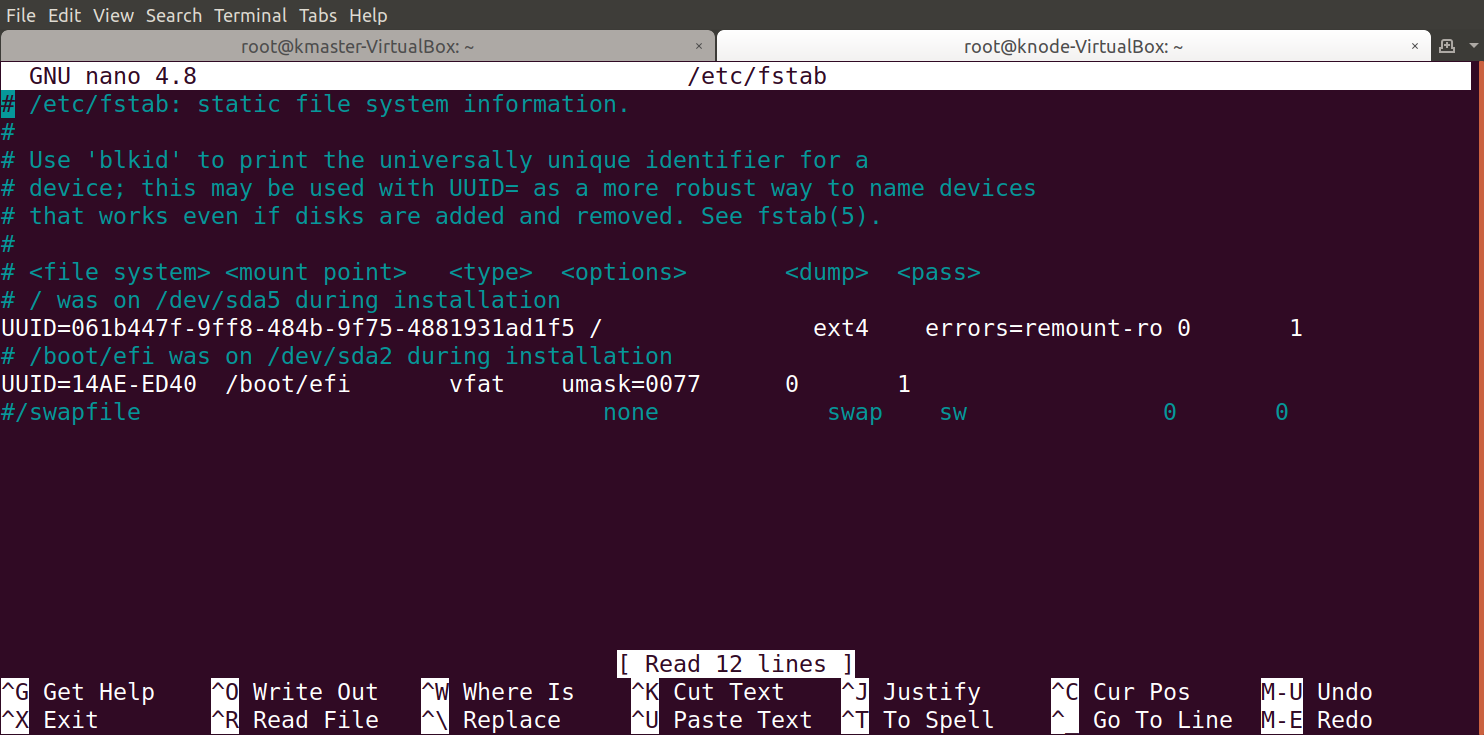
Now press Ctrl+X, then press Y, and then press Enter to Save.

You have successfully installed Kubernetes on both the machines now!

**Step 7: we will individually set the configurations in both machines. (In Master and Slave)** Comment on swap entry last line.

**root@kmaster-VirtualBox:~# swapoff -a**

**root@kmaster-VirtualBox:~# nano /etc/fstab**



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

**Step 1:** We will now start our Kubernetes cluster from the master’s machine. Run the following command:

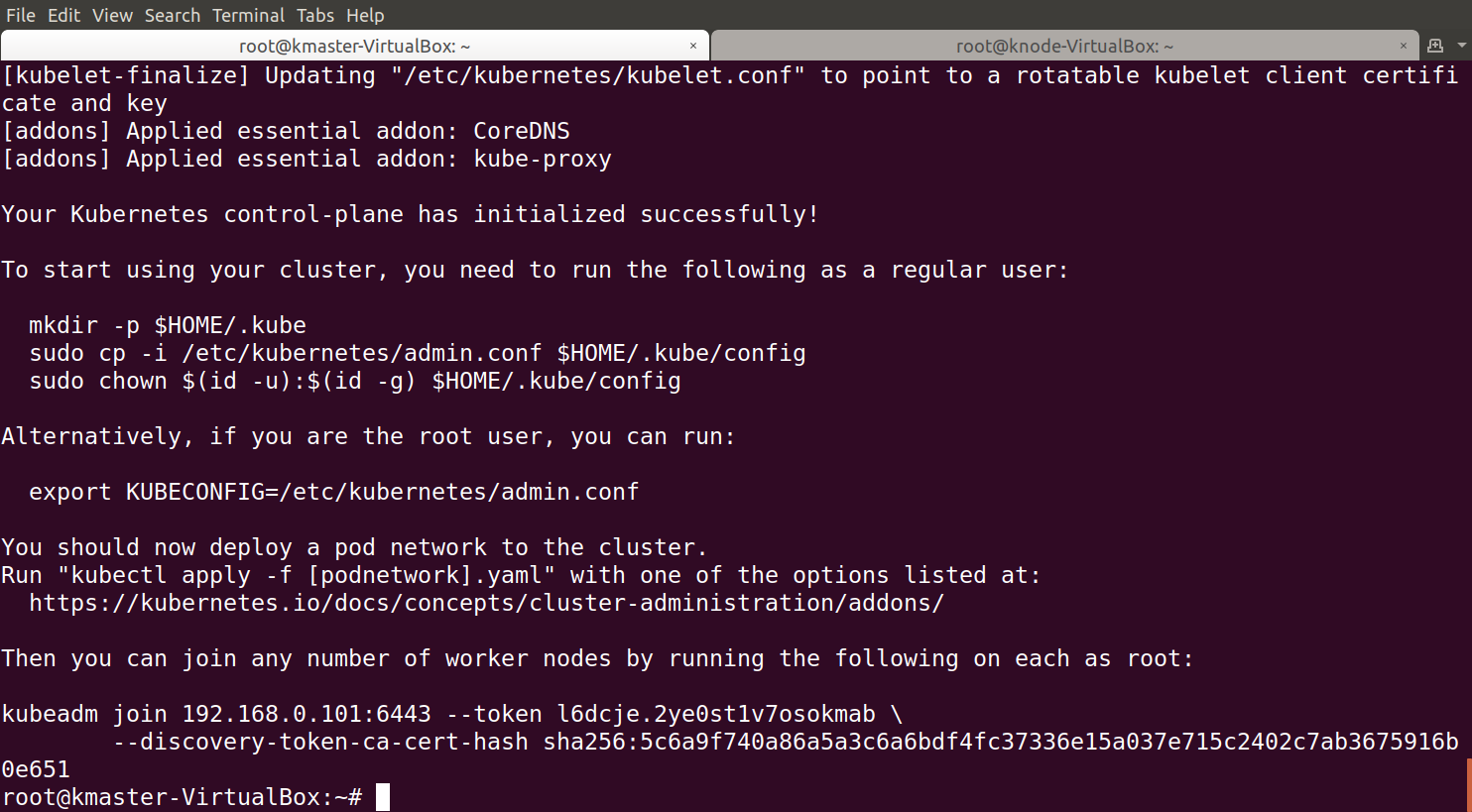
**root@kmaster-VirtualBox:~# kubeadm init –apiserver-advertise-address=192.168.0.101--pod-network-cidr=192.168.0.0/16**

**root@kmaster-VirtualBox:~# kubeadm init --apiserver-advertise-address=192.168.0.101 --pod-network-cidr=192.168.0.0/16 --ignore-preflight-errors=NumCPU**

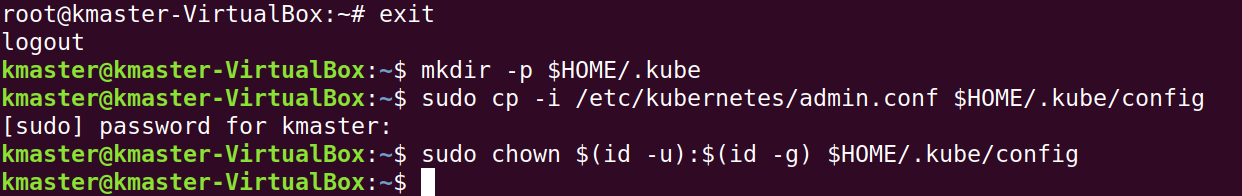
You will get the below output.

1. 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

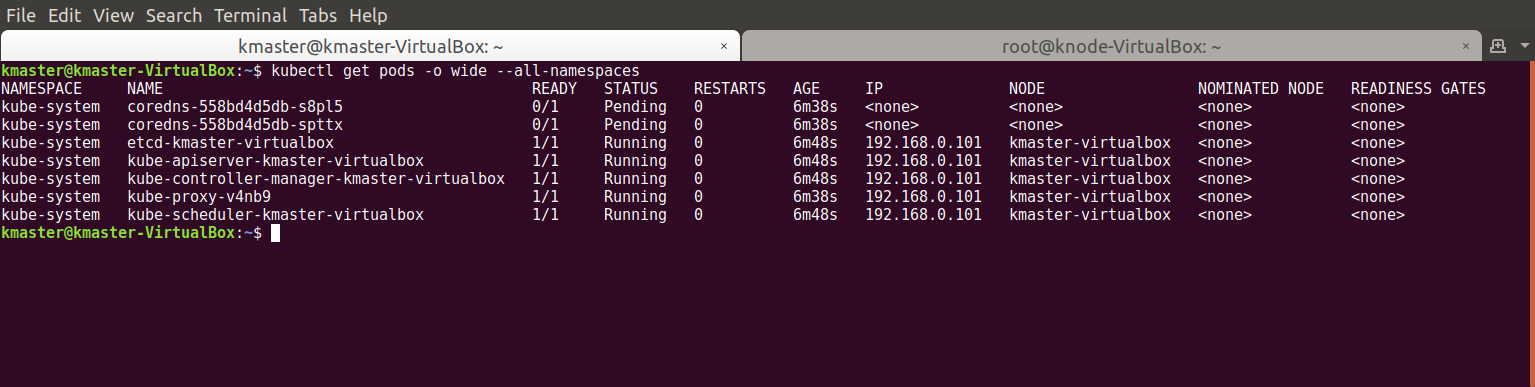


**Step 2:** As mentioned before, run the commands from the above output as a non-root user



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

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



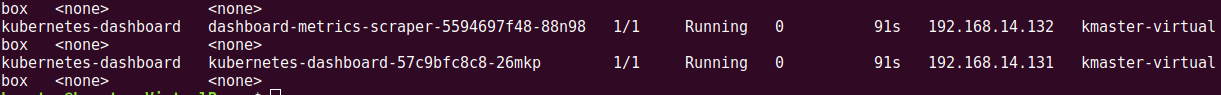
**Step 3**: You will notice from the previous command, that all the pods are running except one: ‘kube-dns’. For resolving this we will install a pod network. To install the CALICO pod network, run the following command:

**kmaster@kmaster-VirtualBox:~$ kubectl apply -f** [**https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml**](https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml)

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

**kmaster@kmaster-VirtualBox:~$ kubectl apply -f** [**https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.yaml**](https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.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:

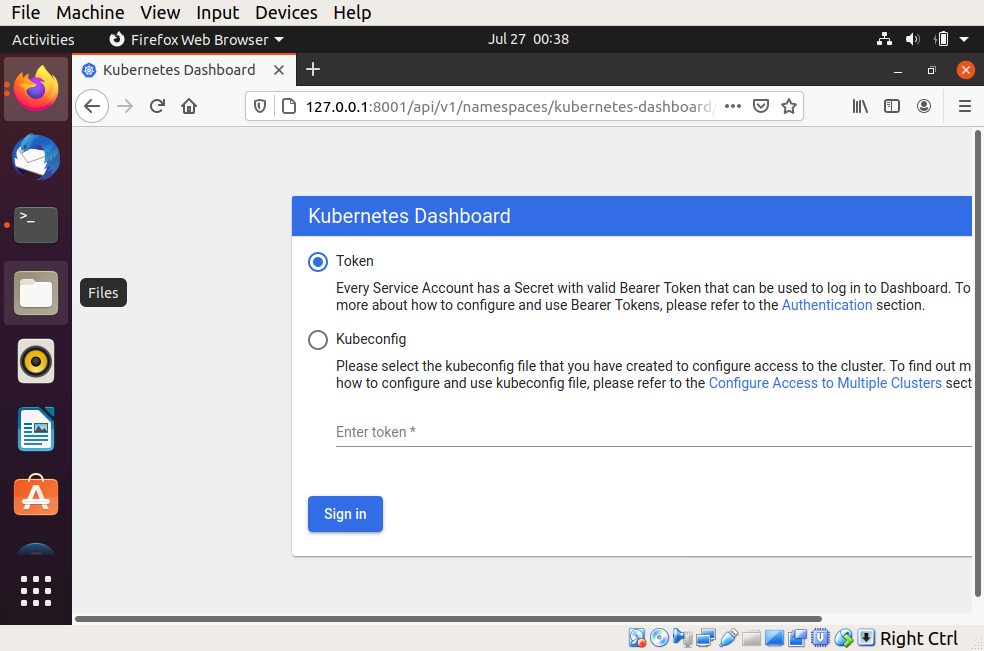
**kmaster@kmaster-VirtualBox:~$ kubectl proxy**

Then you will get something like this:



**In Master Browser put this URL:**

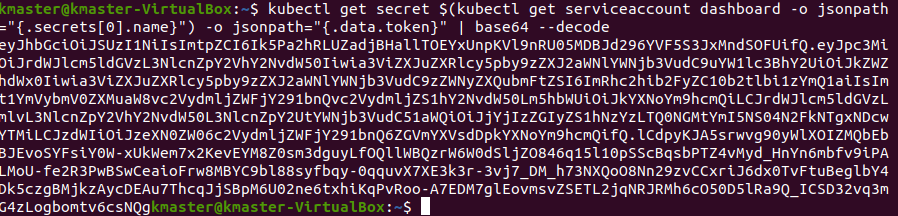
[**http://127.0.0.1:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/login**](http://127.0.0.1:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/login)  
You will then be prompted with this page, to enter the credentials:



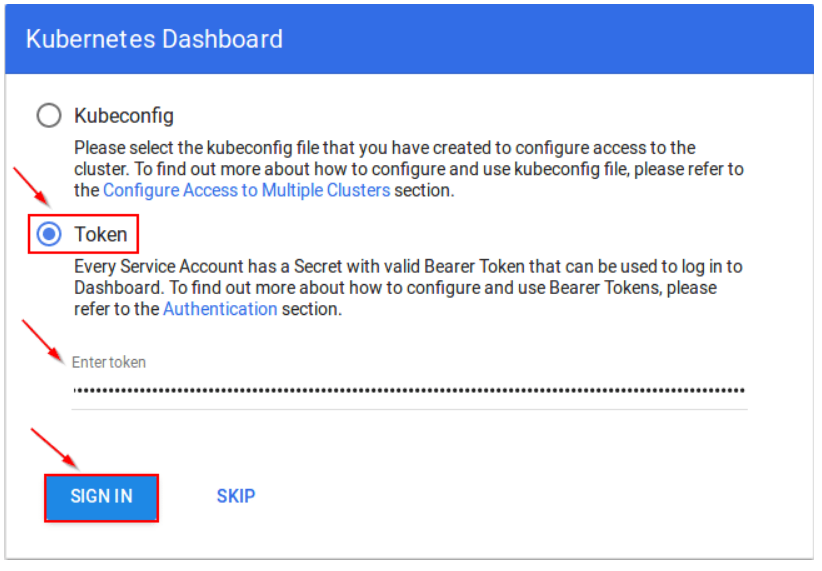
**Step 7**: In this step, we will create the service account for the dashboard and get it’s credentials.  
Note: Run all these commands in a new terminal, or your kubectl proxy command will stop.

Run the following commands:

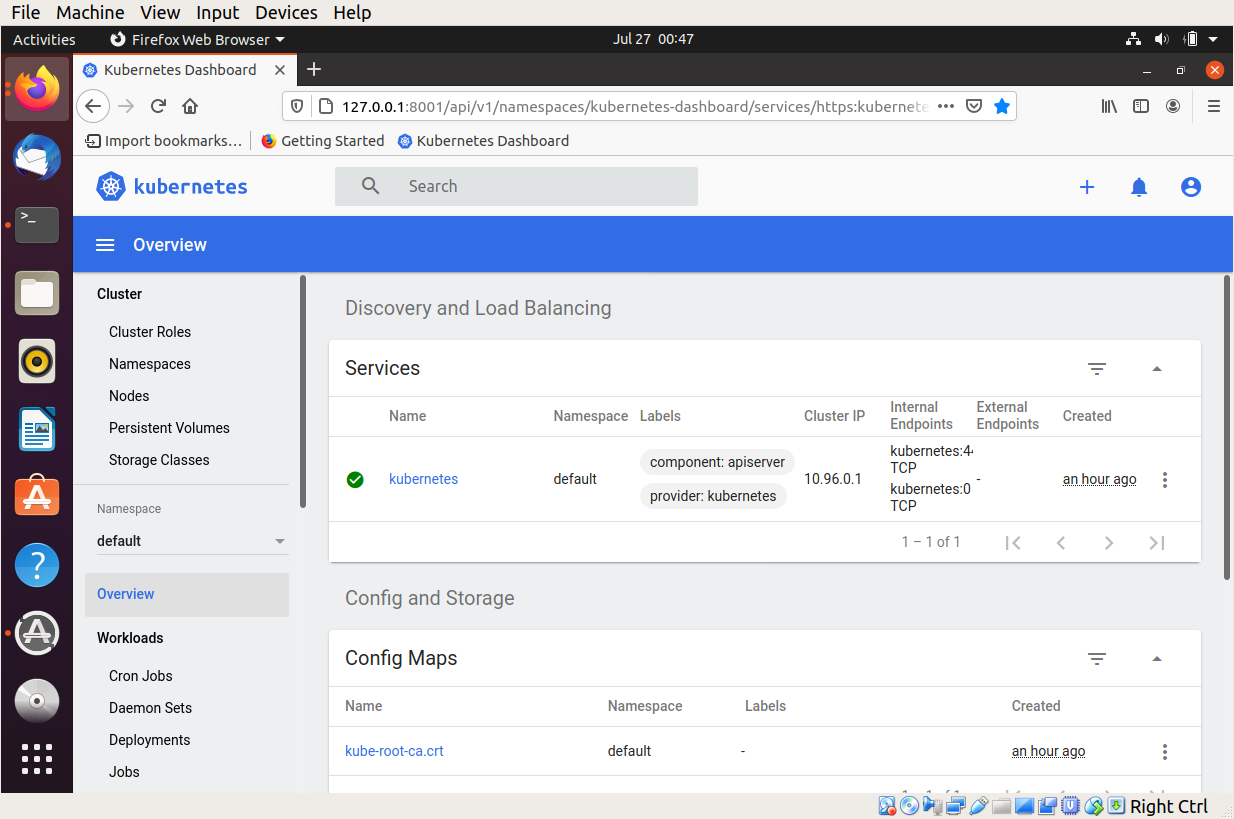
1. This command will create a service account for dashboard in the default namespace



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



3. You have successfully logged into your dashboard!



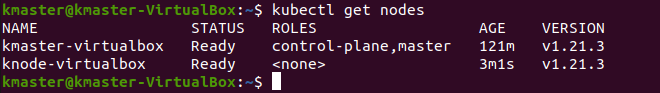
## Steps For Only Kubernetes Node VM (knode)

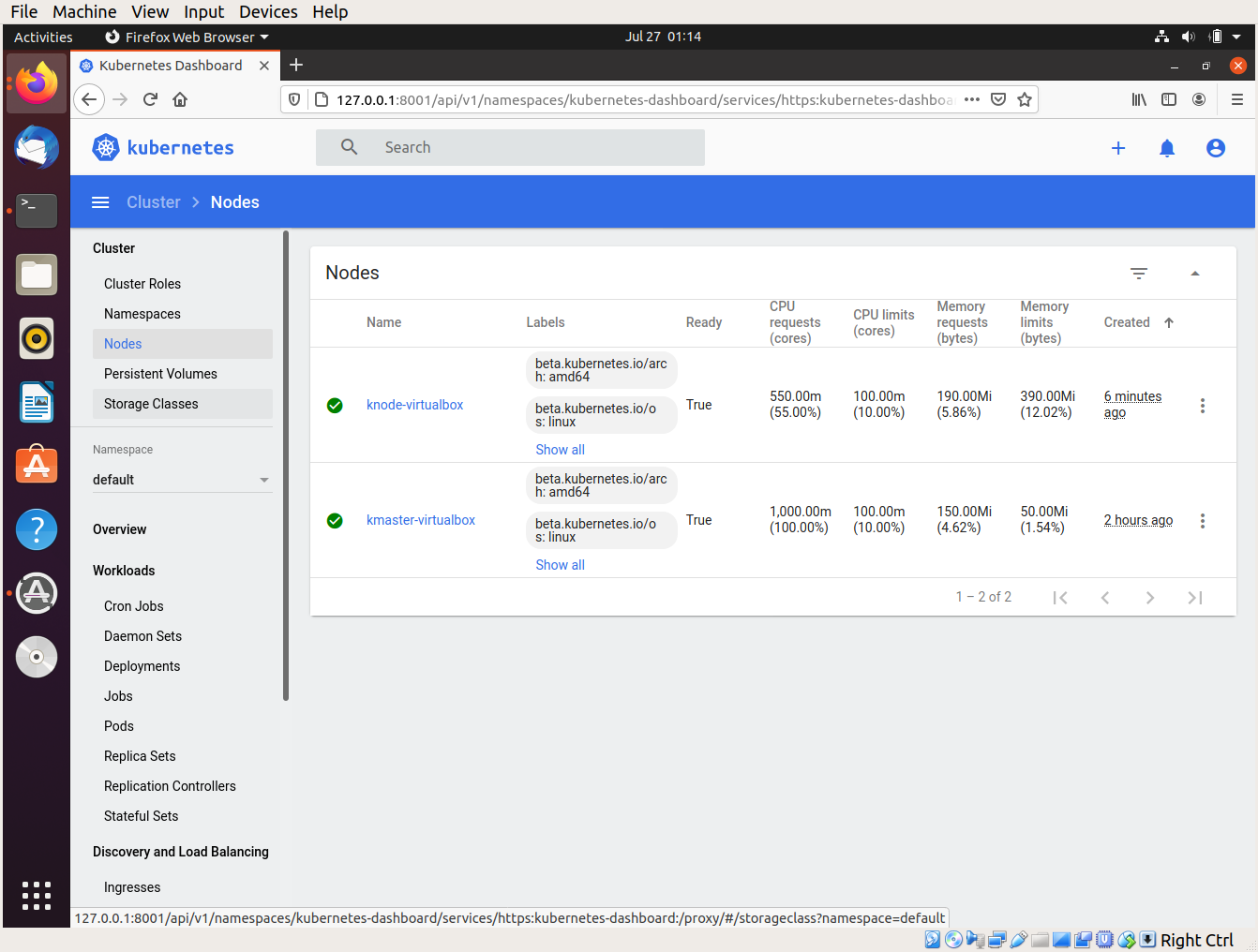
It is time to get your node, to join the cluster! This is probably the only step that you will be doing on the node, after installing kubernetes on it.

Run the join command that you saved, when you ran ‘kubeadm init’ command on the master.

**Note:** Run this command with “sudo”.

[**root@kmaster-VirtualBox**](about:blank)**:~# sudo kubeadm join 192.168.0.101:6443 --token l6dcje.2ye0st1v7osokmab \ --discovery-token-ca-cert-hash sha256:5c6a9f740a86a5a3c6a6bdf4fc37336e15a037e715c2402c7ab3675916b0e651**





**Conclusion: Write your own findings.**