Kubernetes Cluster on Ubuntu VMs

Installation Guide



Installation Guide

**Note:** For this installation we recommend a fresh ubuntu image since Kubernetes cantake up a lot of resources.

Following are the preferable VM settings:

Master:

* 2GB RAM
* 2 Cores of CPU

Slave Node:

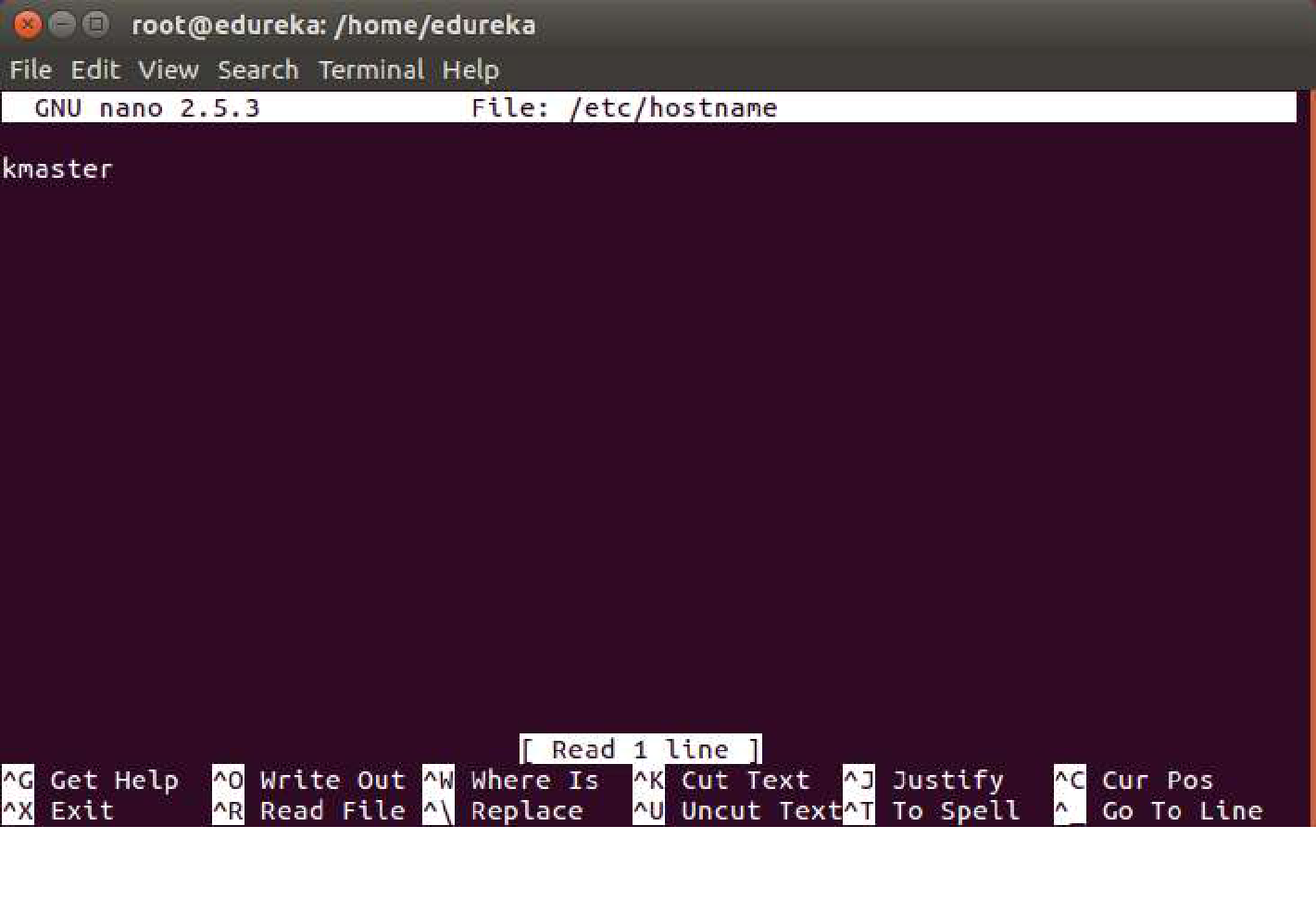
* 1 GB RAM
* 1 Core of CPU

Step 1:

1. Open your Virtual Box Manager, and create new VMs

Step 2:

1. Run the following command: nano /etc/hostname
2. Edit the name to “kmaster” for kmaster VM, and “knode” for knode VM.

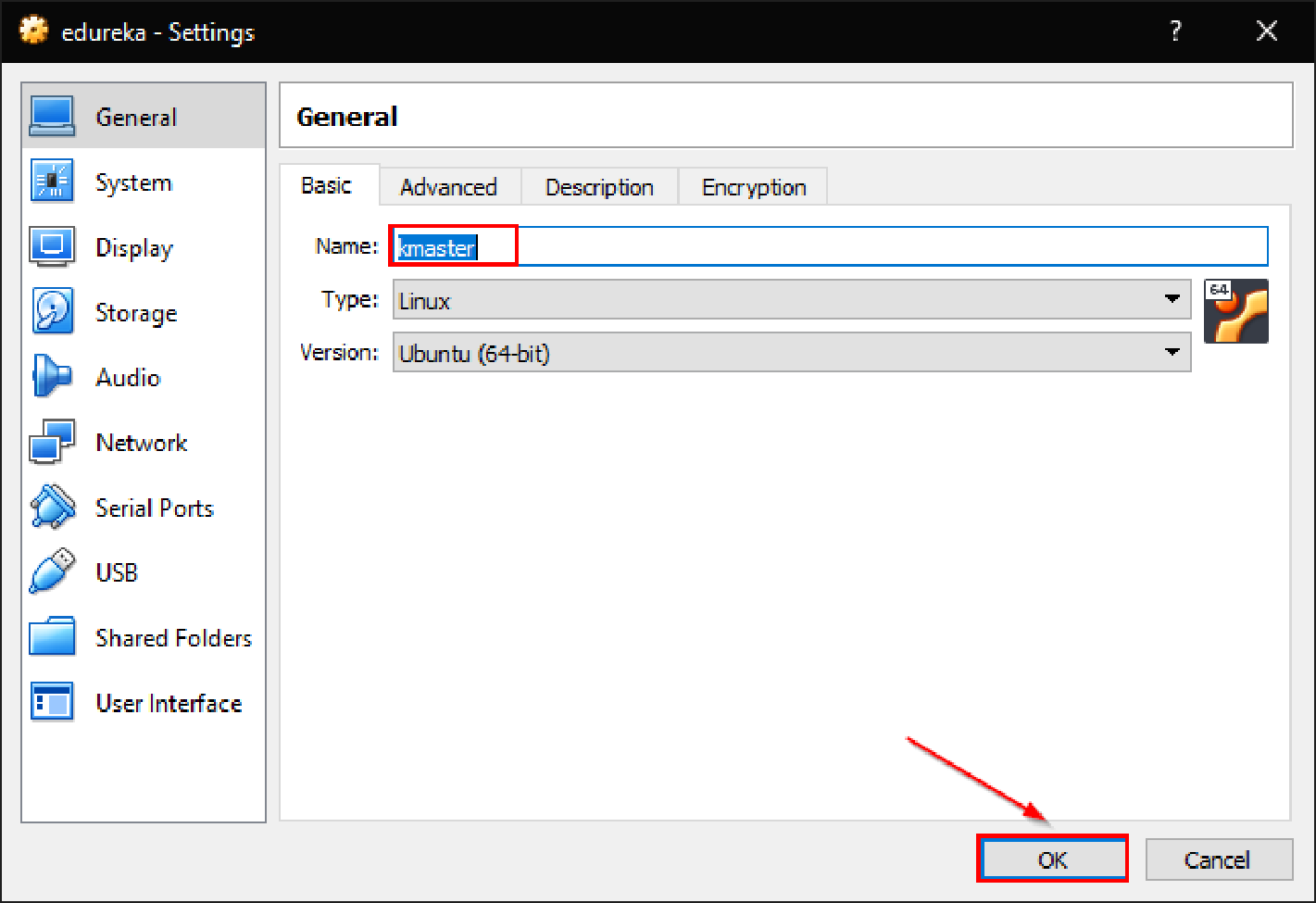


1. Press Ctrl+X, then press Y, and then press Enter to Save.

Step 3: Right Click on your VM, and click on Settings.



Step 4: Edit the name as kmaster, and click on OK



Step 5: Repeat the same steps to get a Slave Node, and name it as “knode

**Steps for Master and Slave VMs**

**Note:** These steps are common to both kmaster and knode VMs

Step 1:

1. Run the following commands:

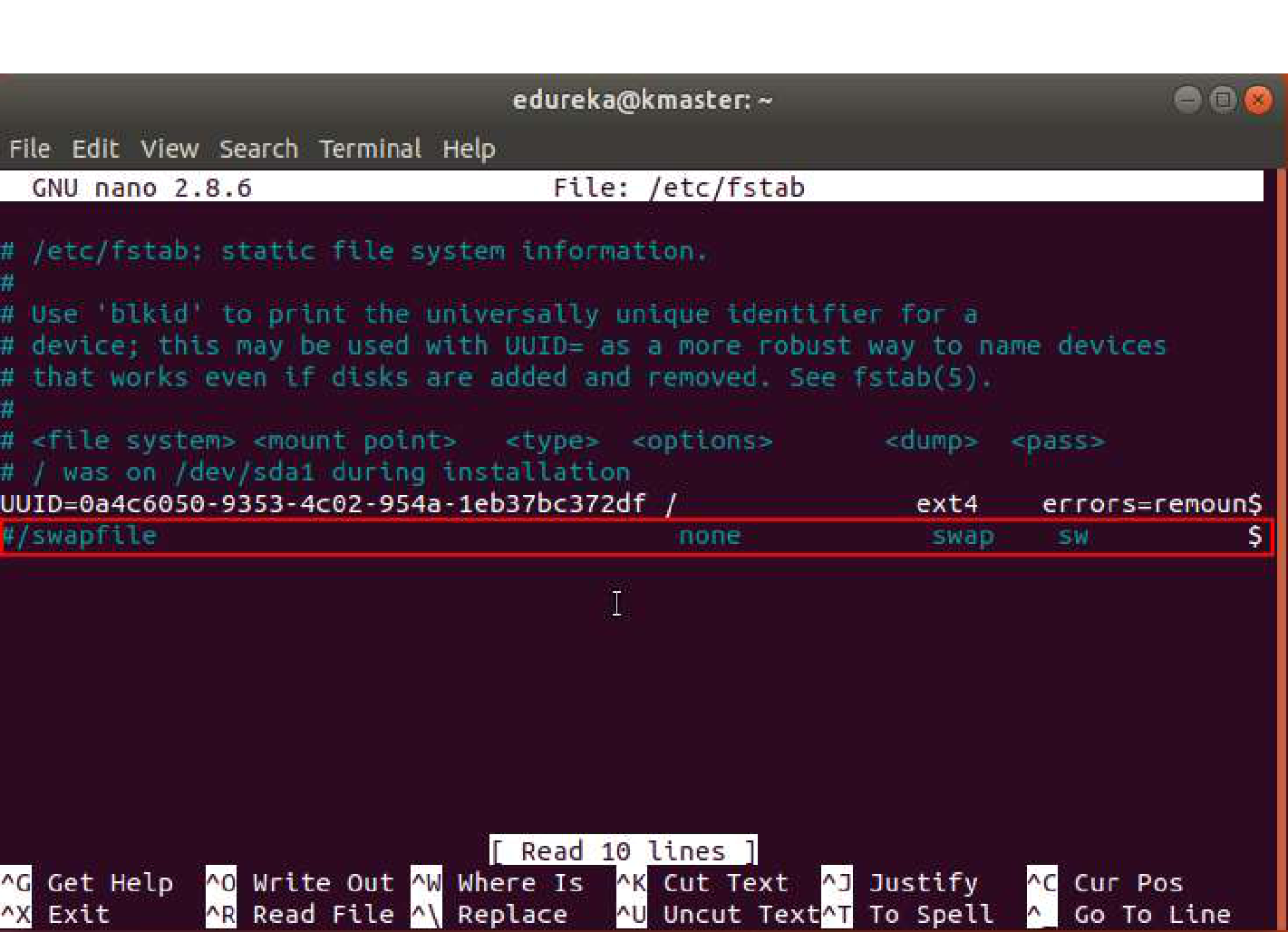
sudo su

apt-get update

swapoff -a

nano /etc/fstab

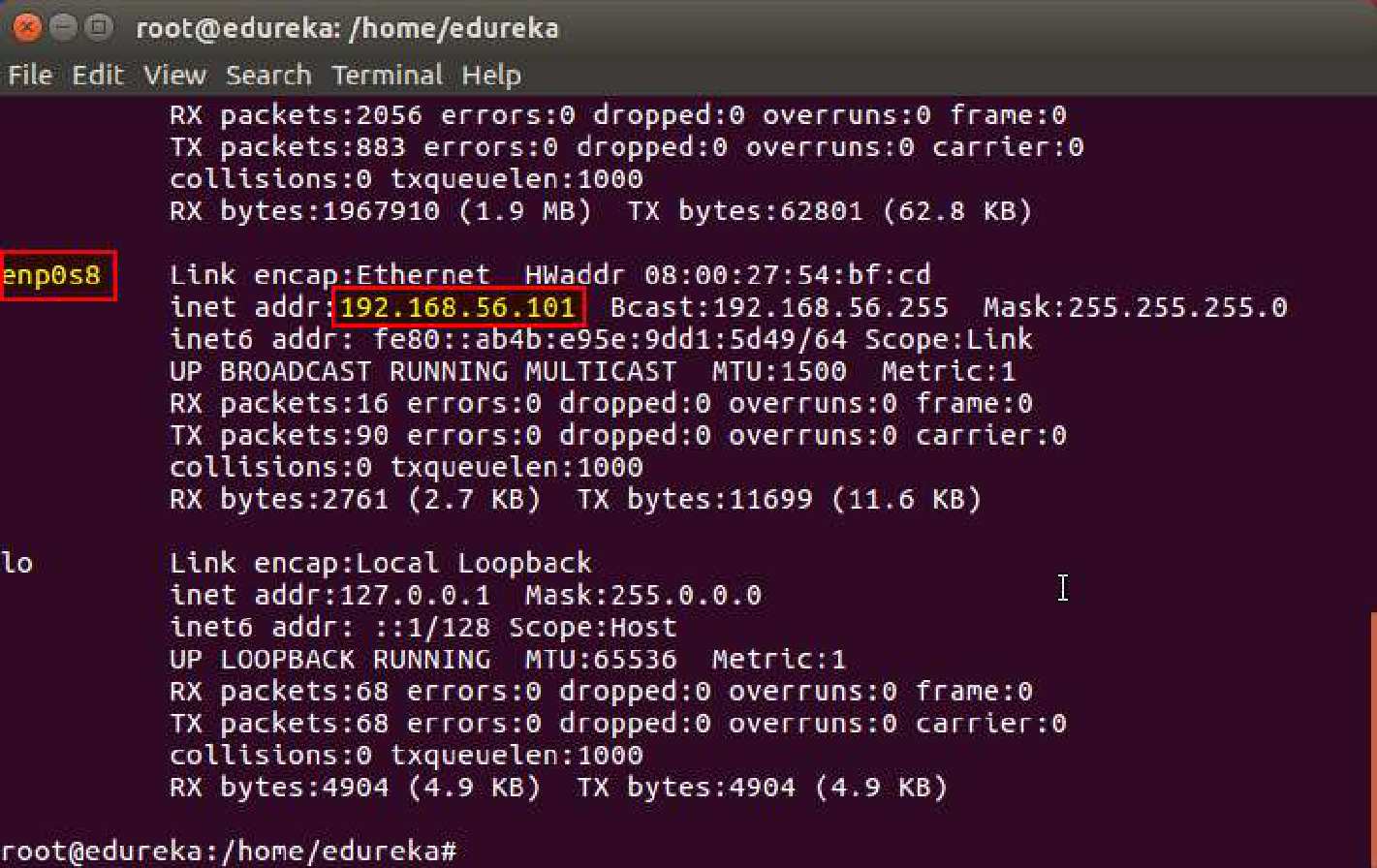
1. Add a “#” in front of the last line, to comment it.



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

Step 2:

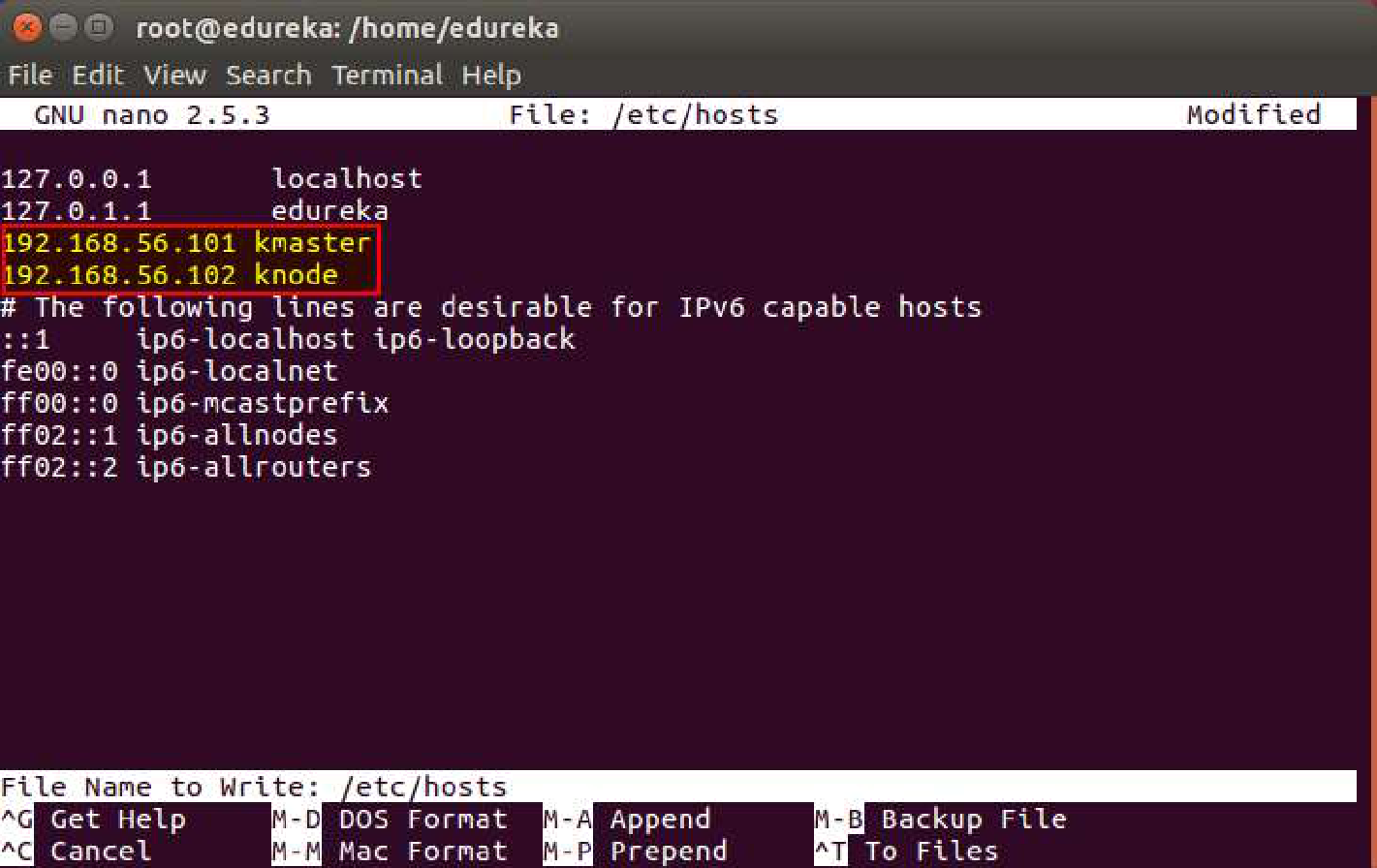
1. Run the following command: ifconfig
2. Make a note of the IP address from the output of the above command. The IP address which has to be copied should be under “enp0s8”, as shown in the screenshot below.



3. Make a note of the IP address of both the VMs using the above commands.

Step 3:

1. Run the following command: nano /etc/hosts
2. Enter the IP address of the kmaster VM and the knode VM both in this file. (This has to be done in both the VMs). In front of the IP address of master write, “kmaster”. Similarly, in front of the Node IP address write “knode”.



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

Step 4: Now we will install openssh-server. Run the following command:

sudo apt-get install openssh-server

Step 5: Next, we will install Docker. Run the following commands:

sudo su

apt-get update

apt install -y docker.io

sudo systemctl enable docker

**Step 6: Next, we will install kubeadm, kubelet and kubectl.**

**For that first run the following four commands to set the 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, kubectl and kubelet**

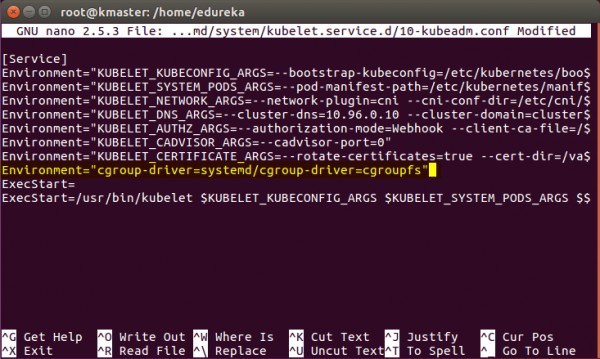
$ apt-get install -y kubelet kubeadm kubectl

**Step 7: Next, Change the config file of kubernetes, go to the following file:**

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

**This will open an editor, enter the following line after the last “Environment” Variable.**

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



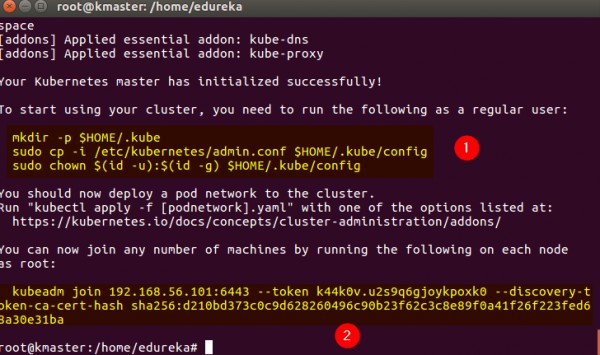
Press Ctrl+X, then press Y, and then press Enter to Save.

Step 8: **Restart your VMs for the changes to take effect.**

You have successfully installed Kubernetes on both the machines now!

Step 9: **Next, execute this only on Kubernetes master:**

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



1. **Run these commands as a non root user:**

$ mkdir -p $HOME/.kube

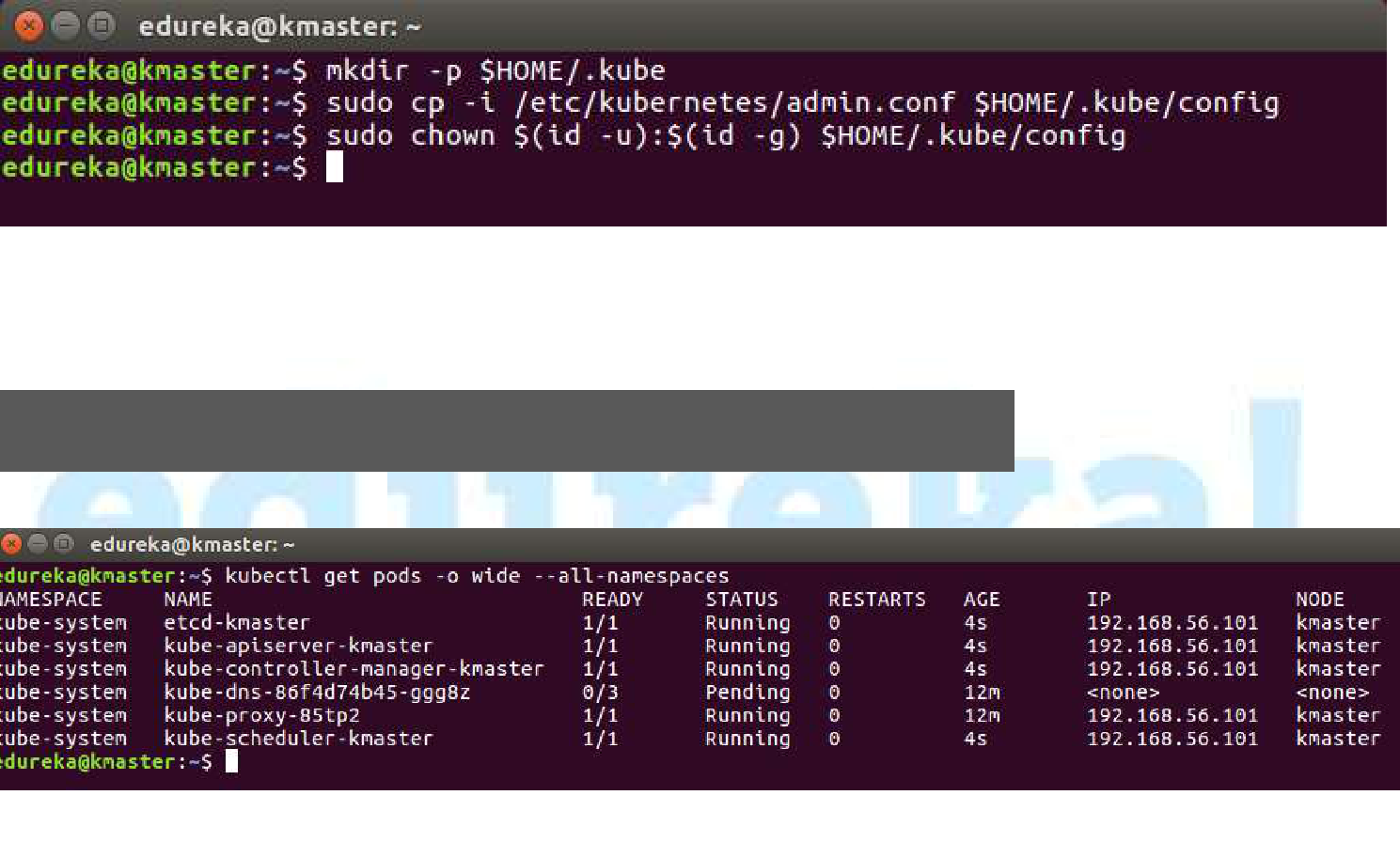
$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

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

1. **Save the second command for later and execute that on worker nodes to join the master**

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

kubectl get pods -o wide --all-namespaces

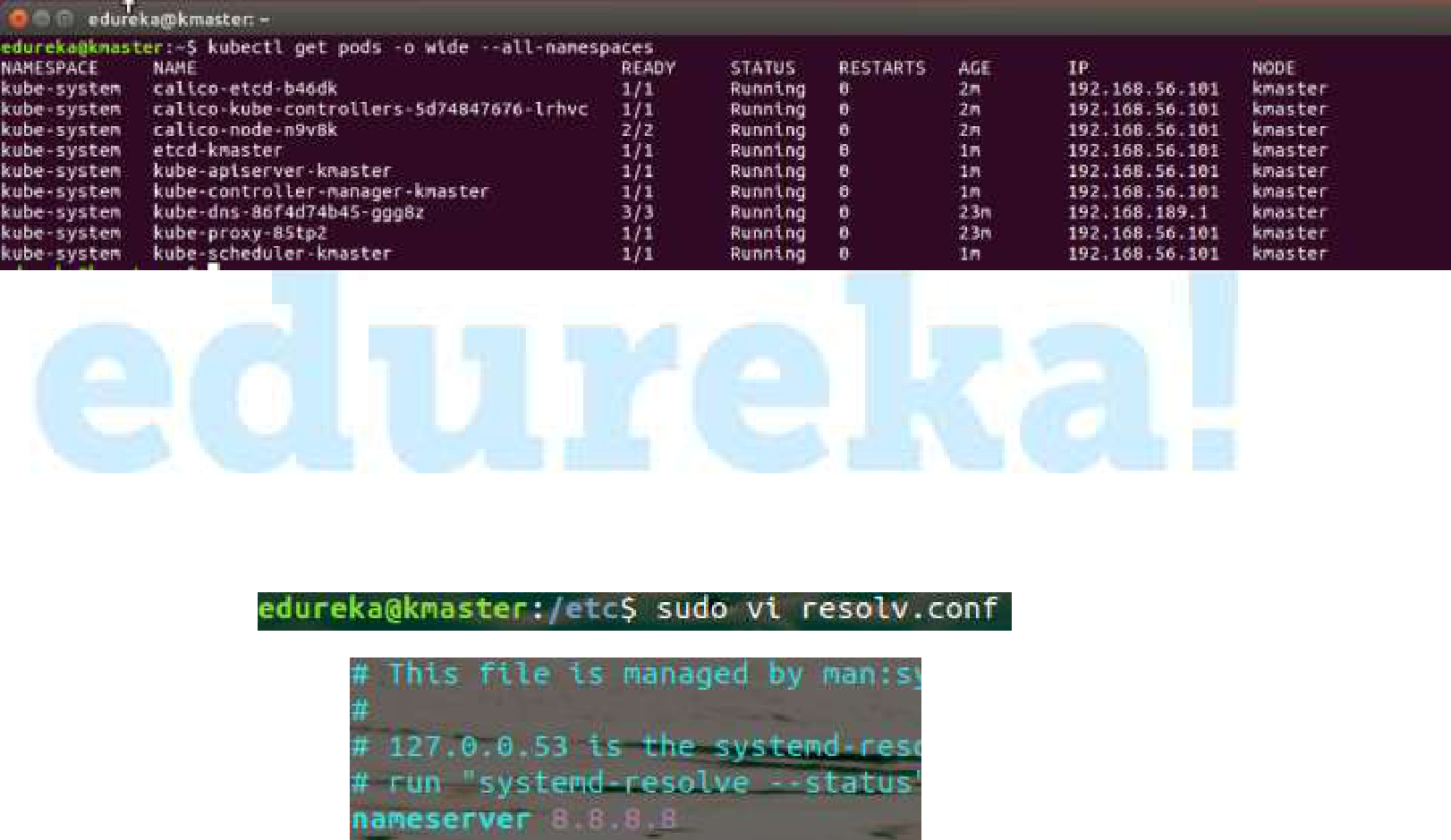


You will notice from the previous command, all the pods are running except some like kube-dns. For resolving this we will install a pod network. To install the pod network, run the following command:

kubectl apply -f <https://docs.projectcalico.org/v3.7/manifests/calico.yaml>

After some time, you will notice that all pods shift to the running state. Check that by following command

watch kubectl get pods --all-namespaces

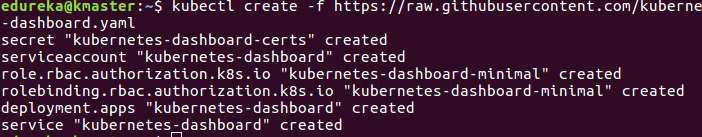


Please check if all pods are running or not. (Ctrl + C) to come out.

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

kubectl create -f

<https://raw.githubusercontent.com/kubernetes/dashboard/v1.10.1/src/deploy/recommended/kubernetes-dashboard.yaml>



Your dashboard is now ready with it’s the pod in the running state. Check by command

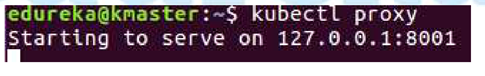
watch kubectl get pods --all-namespaces



(Ctrl + C) to come out.

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

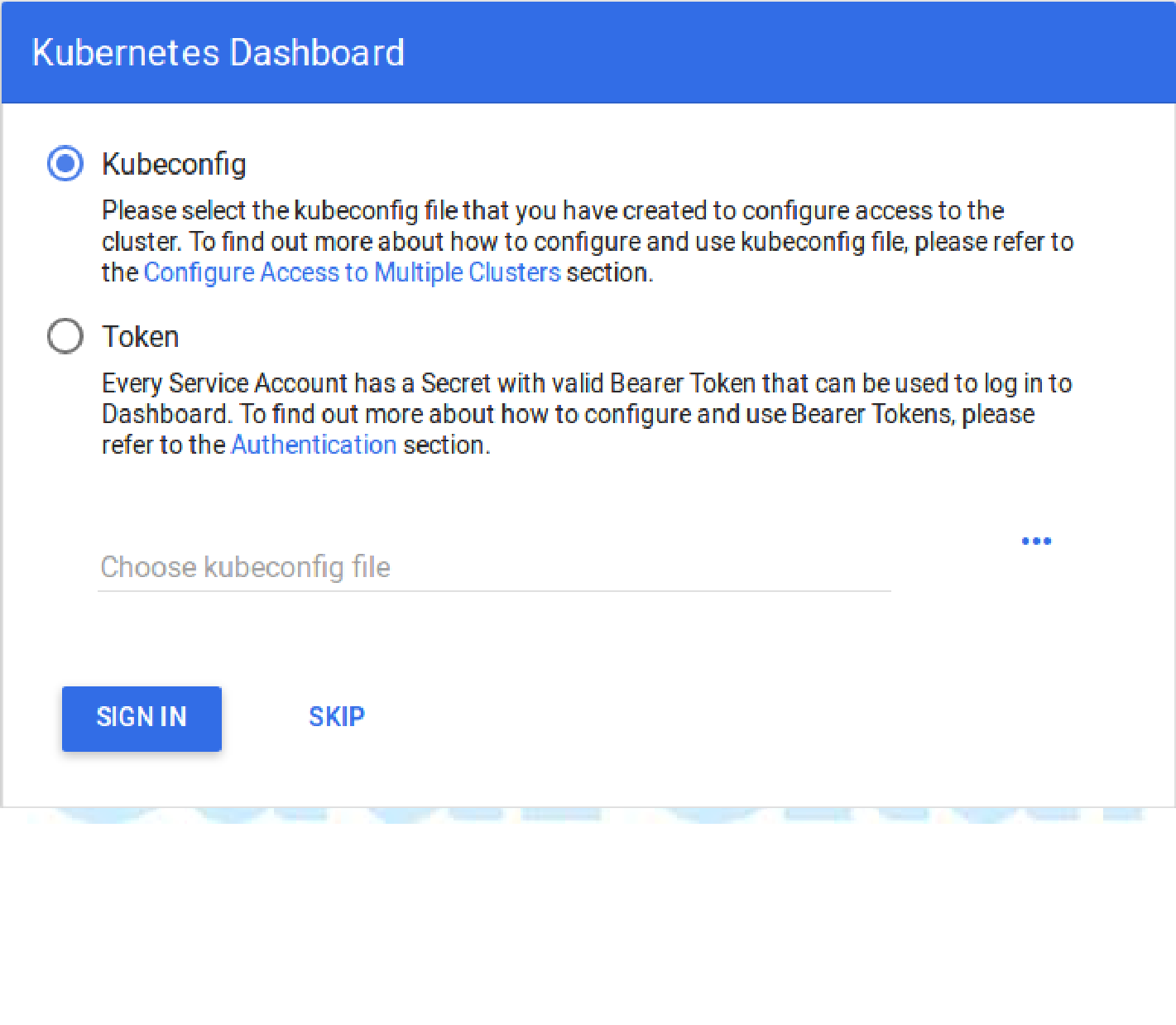
Kubectl proxy



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 be prompted with this page, to enter the credentials.



In next step, we will create the service account for the dashboard and get it’s credentials. Run the following commands

Run all these commands in a new terminal, or your kubectl proxy commandwill stop.

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

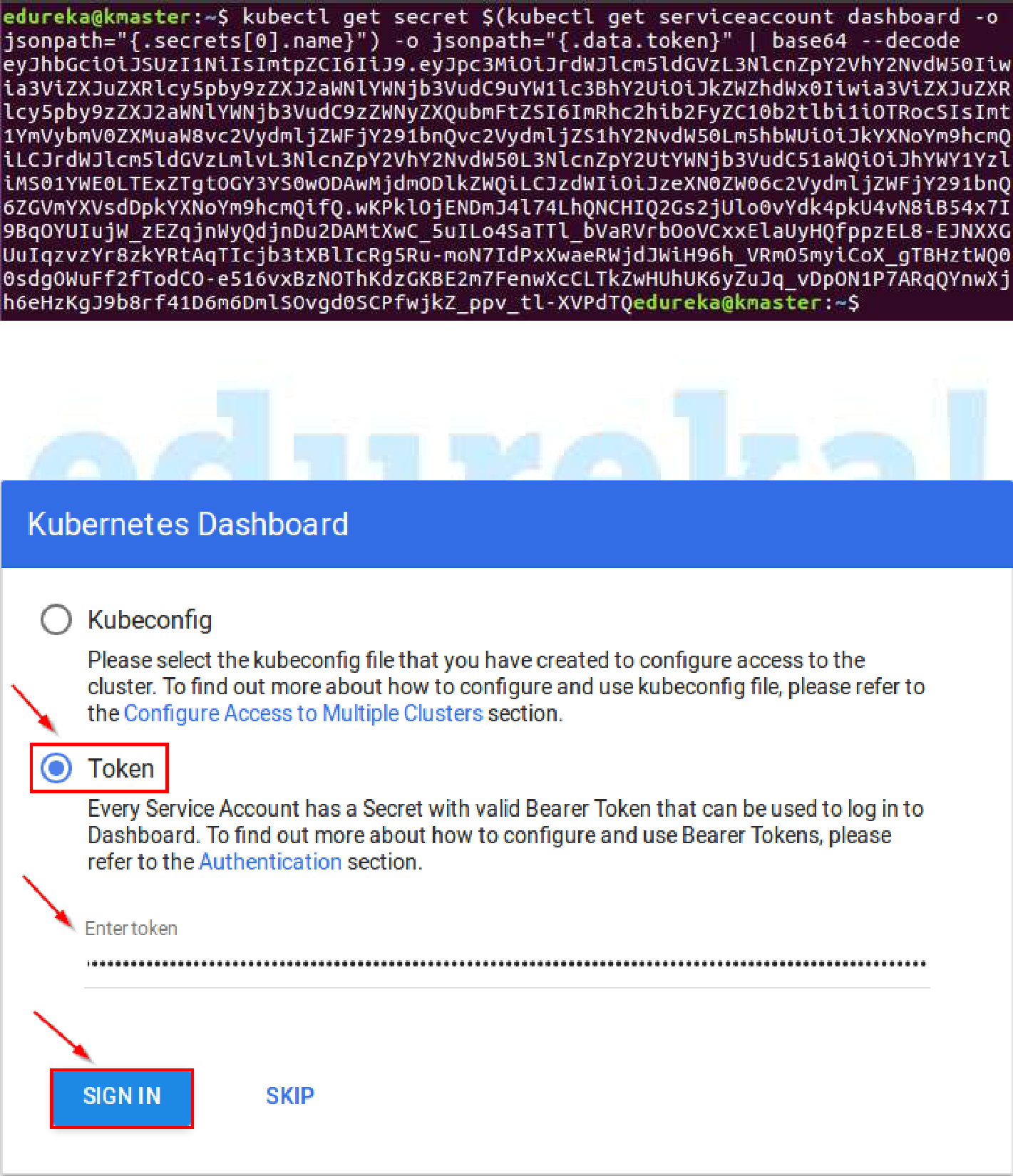
kubectl create serviceaccount dashboard -n default

1. Following command will add the cluster binding rules to your dashboard account

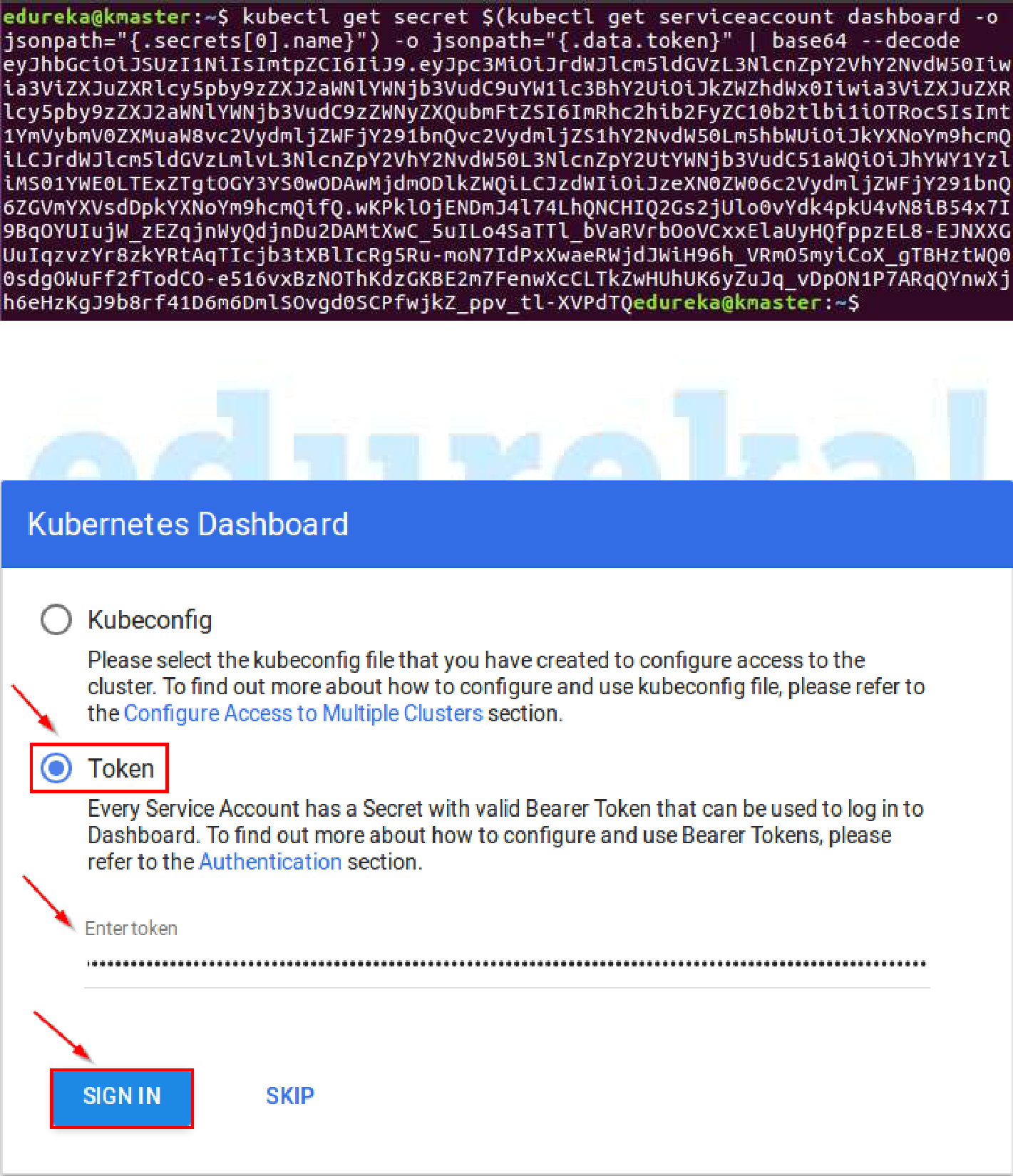
kubectl create clusterrolebinding dashboard-admin -n default --clusterrole=cluster-admin --serviceaccount=default:dashboard

1. Following command will give you the token required for your dashboard login.

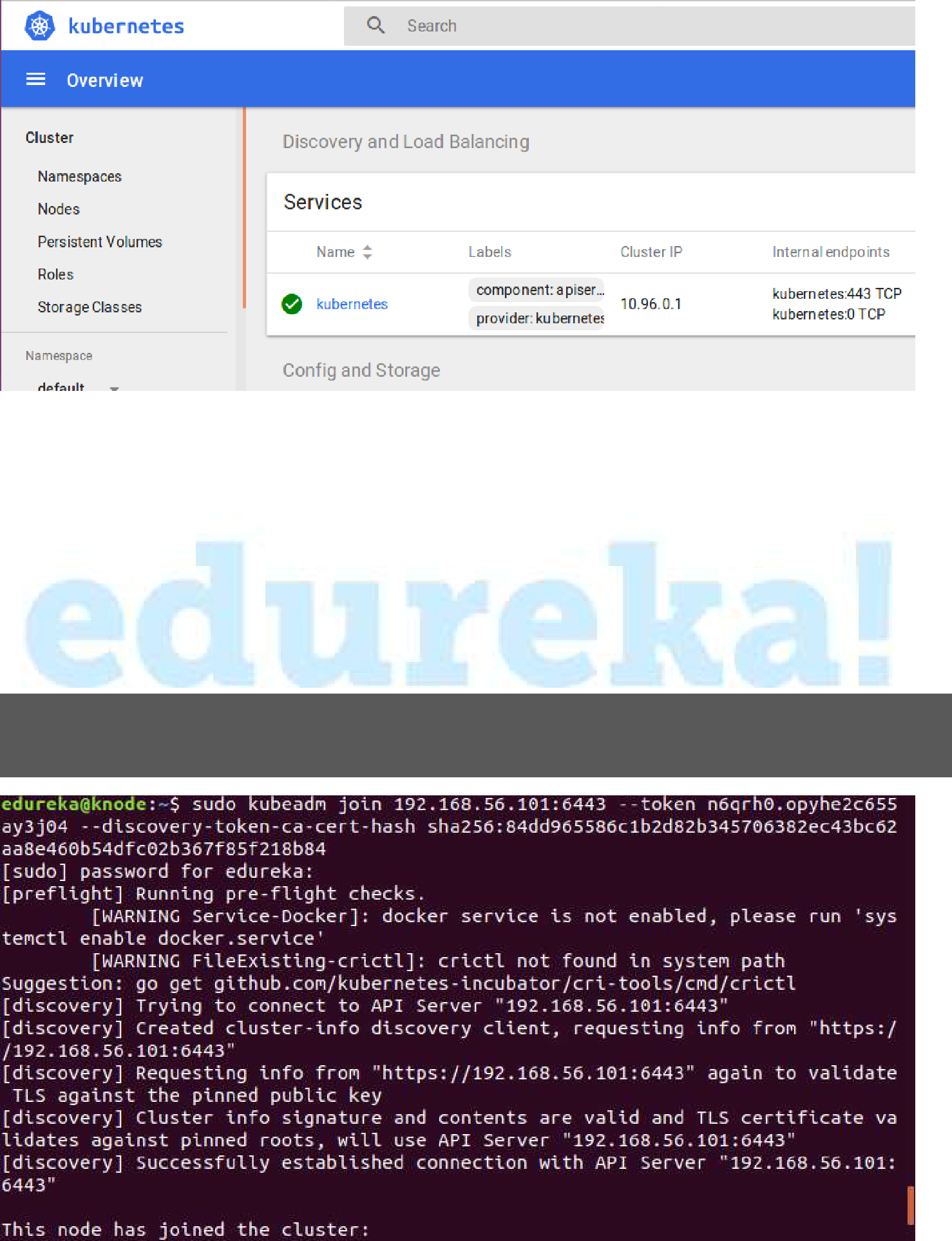
kubectl get secret $(kubectl get serviceaccount dashboard -o jsonpath="{.secrets[0].name}") -o jsonpath="{.data.token}" | base64 –decode



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



1. You have successfully logged in your dashboard!

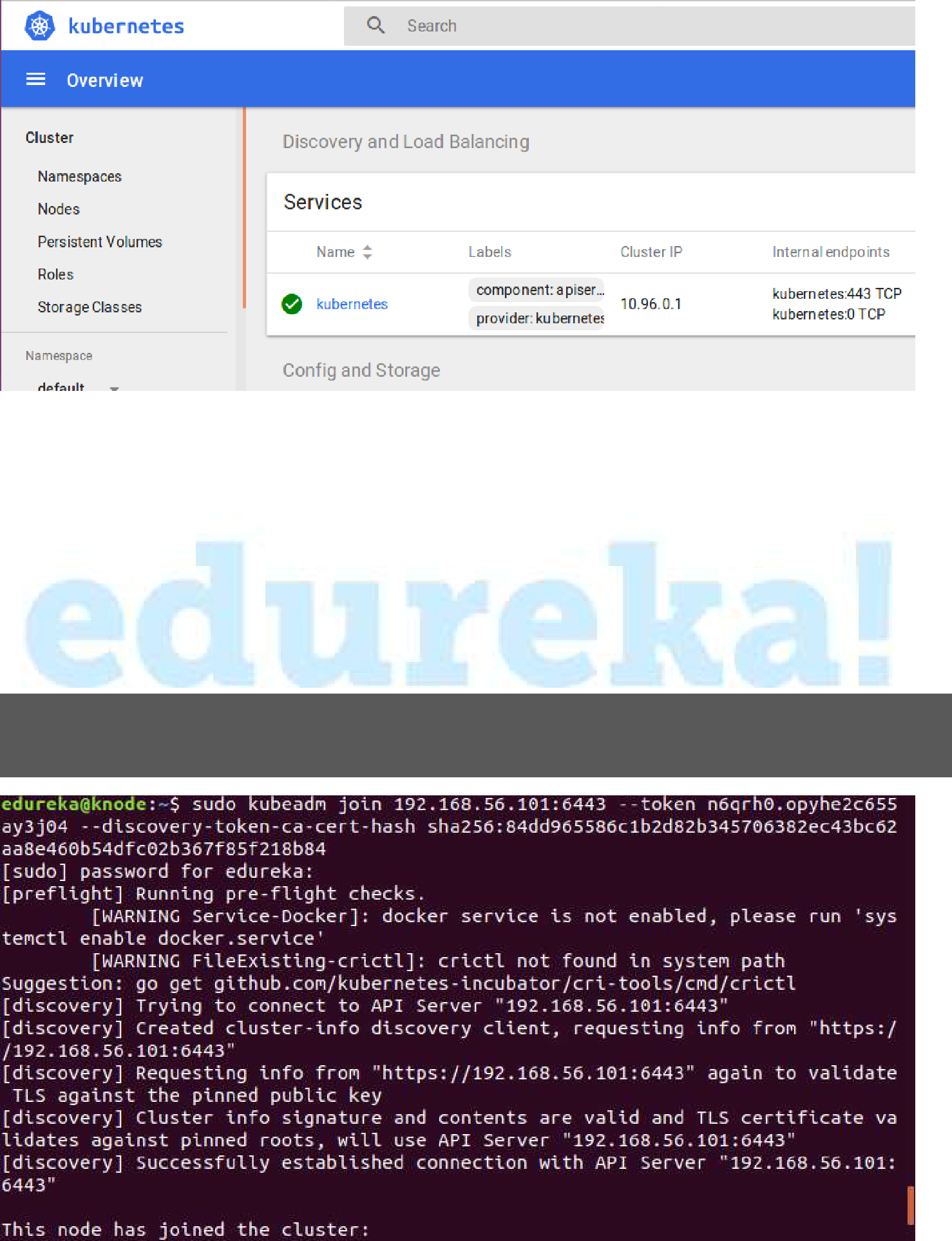


1. Run following command on your master to check the status of nodes:

Kubectl get nodes

**Steps for only Node VM**

It is time to join your node to the cluster! This is probably the only step thatyou 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”.



Run following command again on your master to check the status of the cluster:

Kubectl get nodes

**Your Kubernetes Cluster is now ready! With a Master and a Node :-)**