**Installing Kubernetes**

* K8s can be installed on a single node to set up the developer environments.
  + minikube
* On production environments we would be using a multi node setup
  + For IOT / ARM processers => K3s (Lightweight Kubernetes)
  + Kubeadmin (multi node setup on linux workloads)
  + Kops or installing k8s on AWS
  + kubespray
  + Native Kubernetes on Cloud
    - AKS (Azure Kubernetes Services)
    - GKE (Google Kubernetes Engine)
    - EKS (Elastic Kubernetes Services on AWS)
* Let’s try to setup
  + minikube
  + kubeadm

**Minikube:** This is a tool to set up a single node cluster and suitable for local environments.

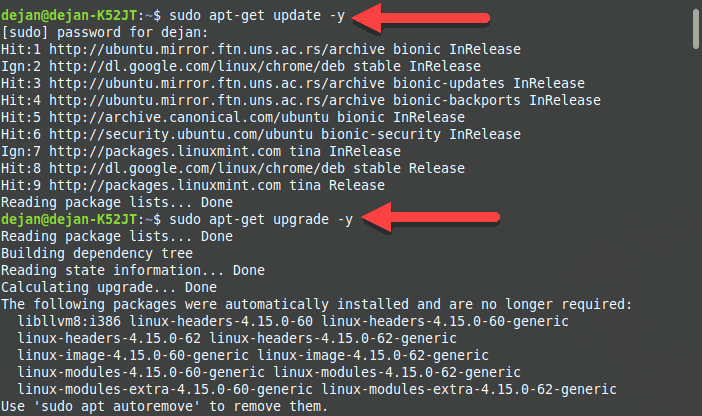
To install Minikube on Ubuntu, follow the steps outlined below. Besides installation instructions, you can also find some basic commands for working inside your local single-node cluster.

### Step 1: Update System and Install Required Packages

Before installing any software, you need to update and upgrade the system you are working on. To do so, run the commands:

sudo apt-get update -y

sudo apt-get upgrade -y

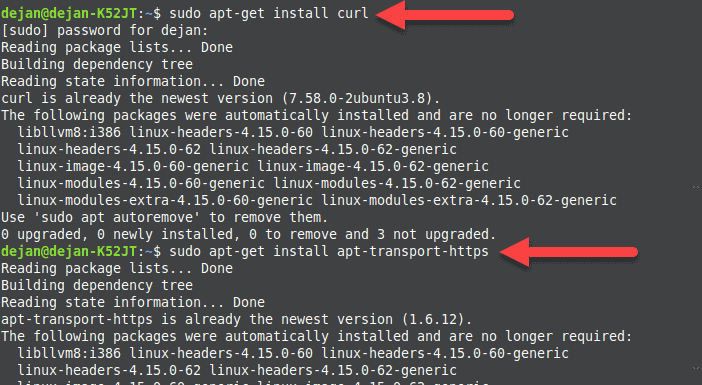


Also, make sure to install (or check whether you already have) the following required packages:

sudo apt-get install curl

sudo apt-get install apt-transport-https

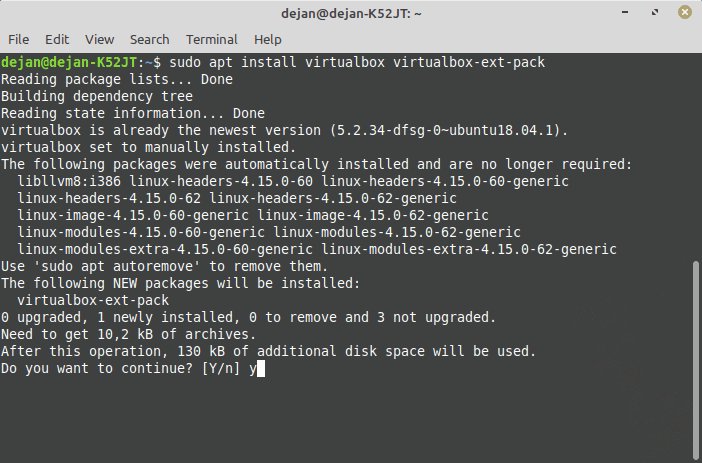
In the image below, the output informs that the packages are already installed.



### Step 2: Install VirtualBox Hypervisor

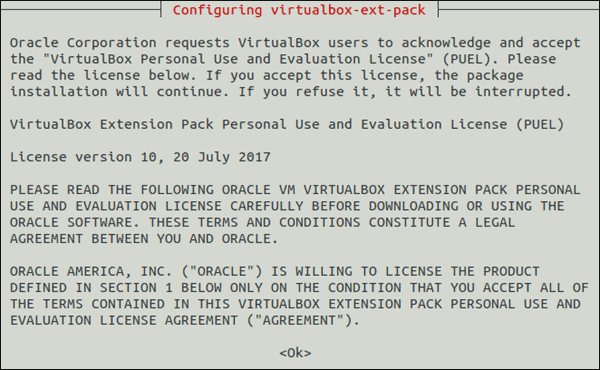
As mentioned above, you need a virtual machine in which you can set up your single node cluster with Minikube. Depending on your preference, you can use VirtualBox or KVM.

sudo apt install virtualbox virtualbox-ext-pack

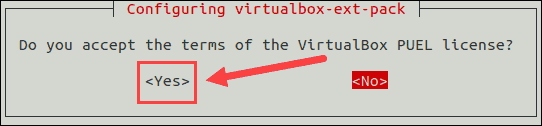


2. Confirm the installation with **y** and hit **Enter**.

3. Next, the licence agreement appears on the screen. Press **Tab** and then **Enter** to continue.



4. The installer asks you to agree with the terms of the VirtualBox PUEL license by selecting **Yes**.



5. Wait for the installation to complete and then move on to the next step.

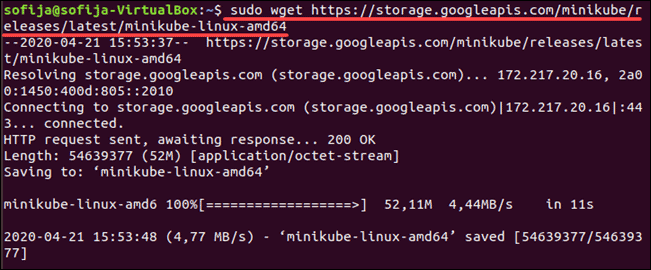
**Note:** For the VirtualBox hypervisor to work, hardware virtualization must be enabled in your system BIOS.

### Step 3: Install Minikube

With VirtualBox set up, move on to installing Minikube on your Ubuntu system.

1. First, download the latest Minikube binary using the **wget** command:

wget https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64



2. Copy the downloaded file and store it into the **/usr/local/bin/minikube** directory with:

sudo cp minikube-linux-amd64 /usr/local/bin/minikube

There will be no output if the command was executed correctly.

3. Next, give the file executive permission using the [chmod command](https://phoenixnap.com/kb/linux-file-permissions):

sudo chmod 755 /usr/local/bin/minikube

Again, there will be no output.

Move Minikube file and give it executive permission.

4. Finally, verify you have successfully installed Minikube by checking the version of the software:

minikube version

The output should display the version number of the software, as in the image below.

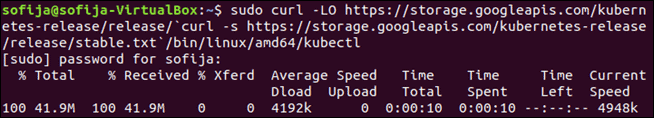
Check Minikube version.

### Step 4: Install Kubectl

To deploy and manage clusters, you need to install **kubectl**, the official command line tool for Kubernetes.

1. Download kubectl with the following command:

curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl



2. Make the binary executable by typing:

chmod +x ./kubectl

3. Then, move the binary into your path with the command:

sudo mv ./kubectl /usr/local/bin/kubectl

4. Verify the installation by checking the version of your kubectl instance:

kubectl version -o json

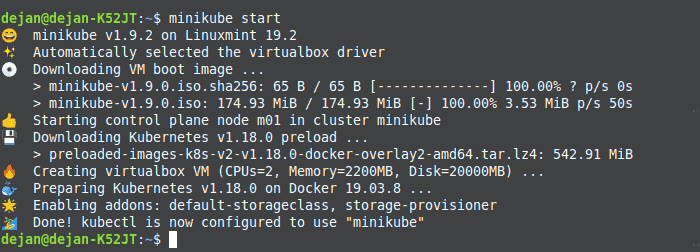
### Step 5: Start Minikube

Once you have set up all the required software, you are ready to start Minikube.

Run the following command:

minikube start

First, the system downloads the Minikube ISO file from an online source and the localkube binary. Then, it creates a virtual machine in VirtualBox within which it starts and configures a single node cluster.

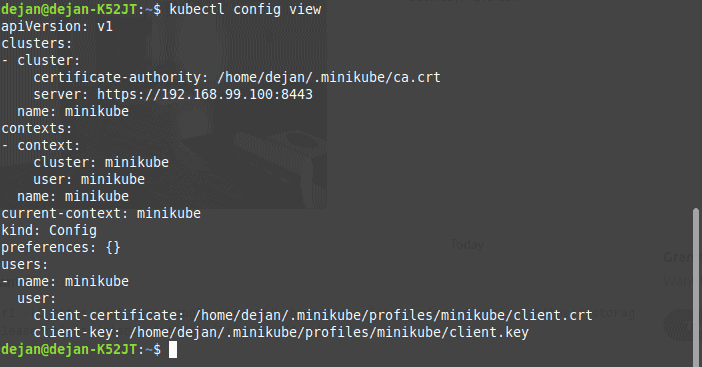


## Managing Kubernetes with Minikube

### Common Minikube Commands

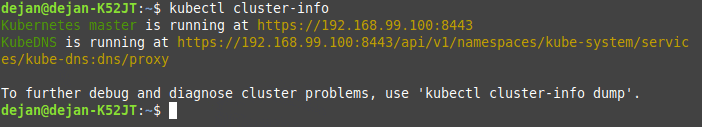
To see the kubectl configuration use the command:

kubectl config view



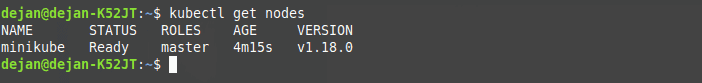
To show the cluster information:

kubectl cluster-info



To check running nodes use the following command:

kubectl get nodes

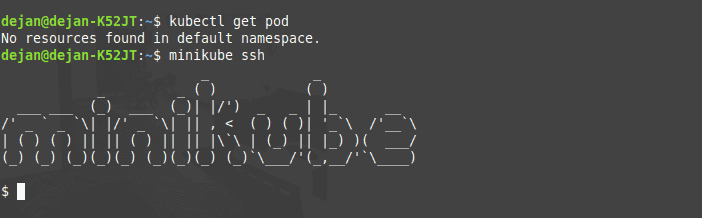


To see a list of all the Minikube pods run:

kubectl get pod

To ssh into the Minikube VM:

minikube ssh

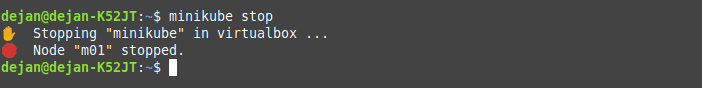


To exit out of the shell run:

exit

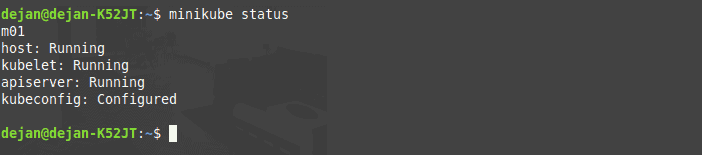
To stop running the single node cluster type:

minikube stop



To check its status use:

minikube status

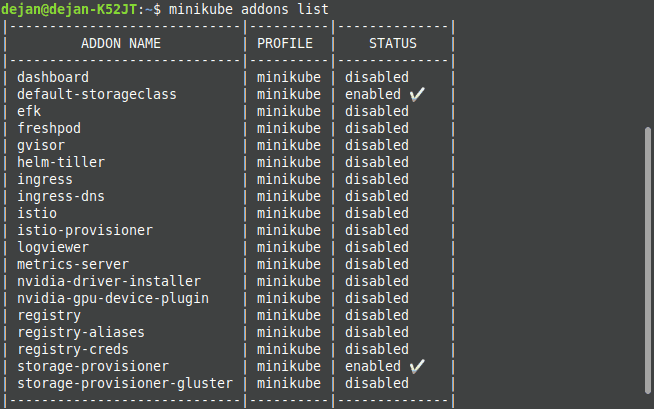


To delete the single node cluster:

minikube delete

To see a list of installed Minikube add-ons:

minikube addons list

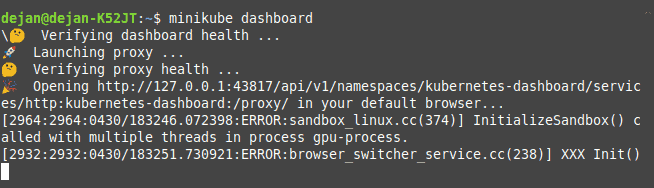


### Access Minikube Dashboard

Minikube comes with a dashboard add-on by default. The web dashboard provides a way to manage your [Kubernetes cluster](https://phoenixnap.com/kb/what-is-kubernetes) without actually running commands in the terminal.

To enable and access the Minikube dashboard via terminal, run the following command:

minikube dashboard

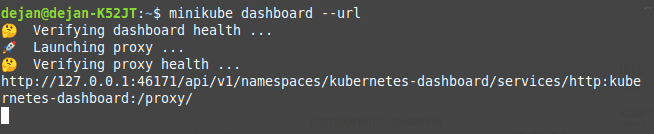


Once you exit the terminal, the process will end and the Minikube dashboard will shut down.

Alternatively, you can access the dashboard directly via browser.

To do so, acquire the dashboard’s IP address:

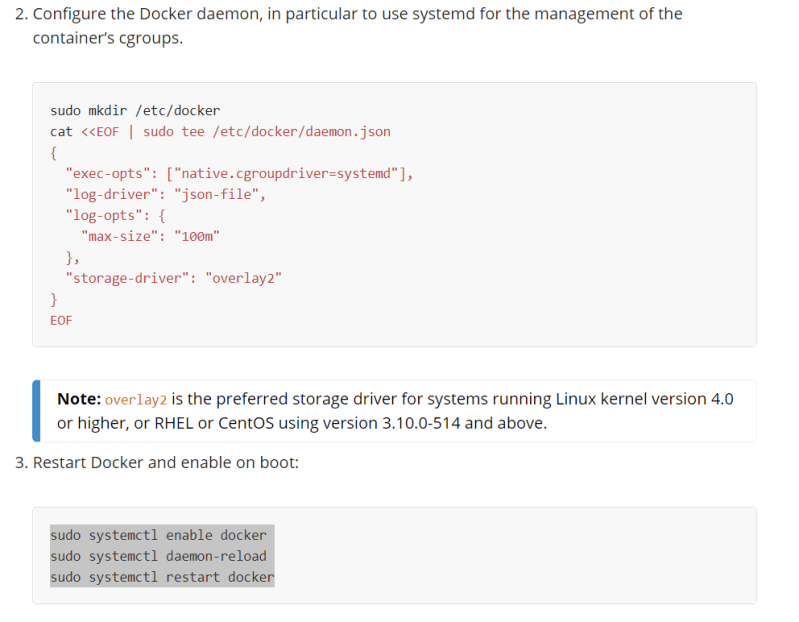
minikube dashboard --url

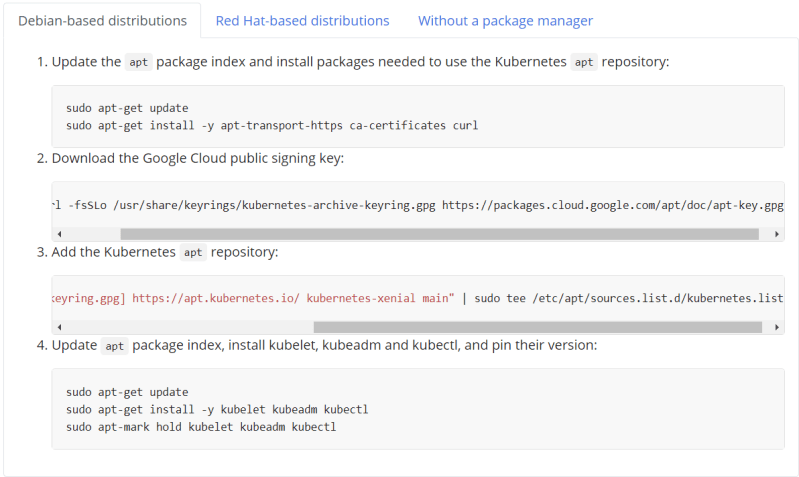


Access your Minikube dashboard by browsing to your dashboard’s IP address.

Summary of installation by kubeadm

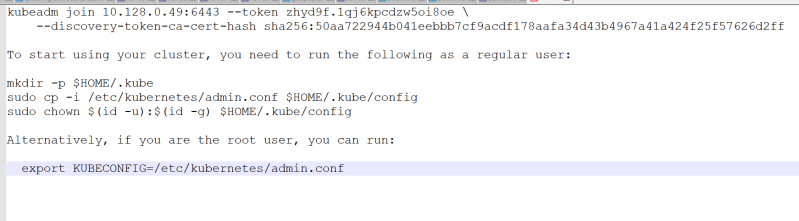
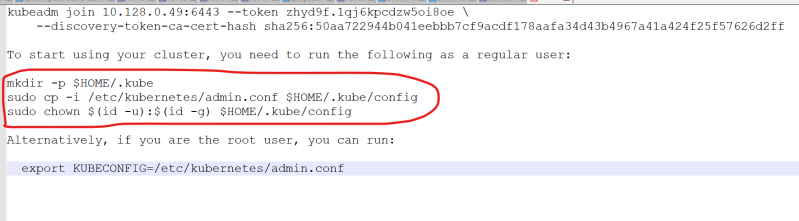
* Install docker on the nodes (both) and execute the steps mentioned in (Link : <https://kubernetes.io/docs/setup/production-environment/container-runtimes/#docker>)



* Install the kubeadm, kubelet and kubectl on both nodes [Refer Here](https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/#installing-kubeadm-kubelet-and-kubectl) 

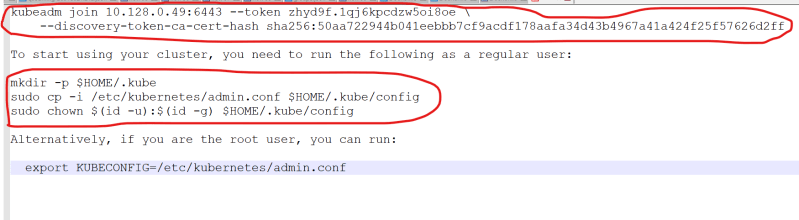
Now initialize the kubeadm by loggin in as root user on master node

Command: kubeadm init

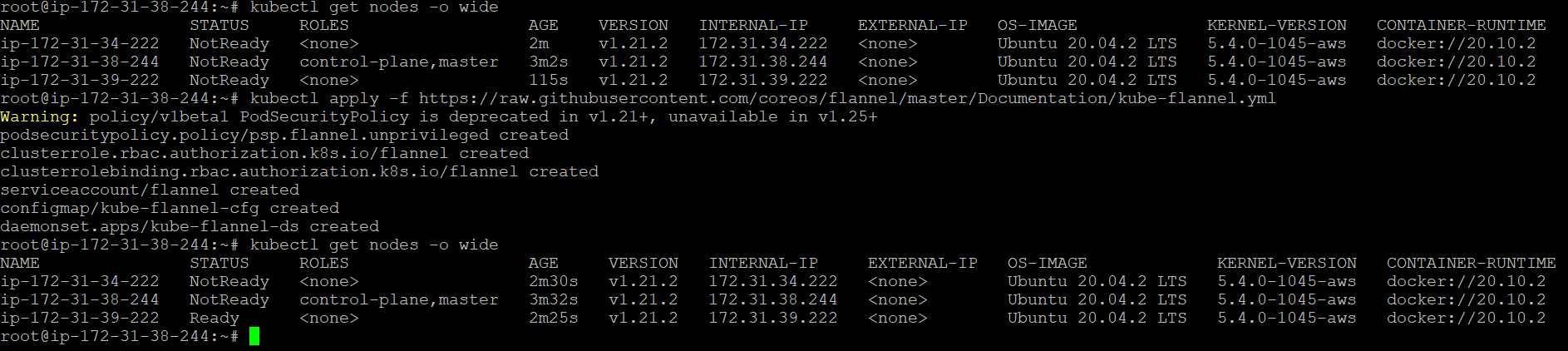
* Copy the commands to configure the kubectl and join command 
* Now become the non root user and execute the following 

Now install a pod network on master node and verify if the node is in ready state by executing kubectl get nodes

kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"

* Now login into node and become root user (sudo -i) and execute the join command 

Now login into master node and execute kubectl get nodes to see the status of both nodes



Below error you would be getting if you are using Kubernetes nodes and try to get node status.

