## **Kubernetes multi-node setup using Kubeadm**

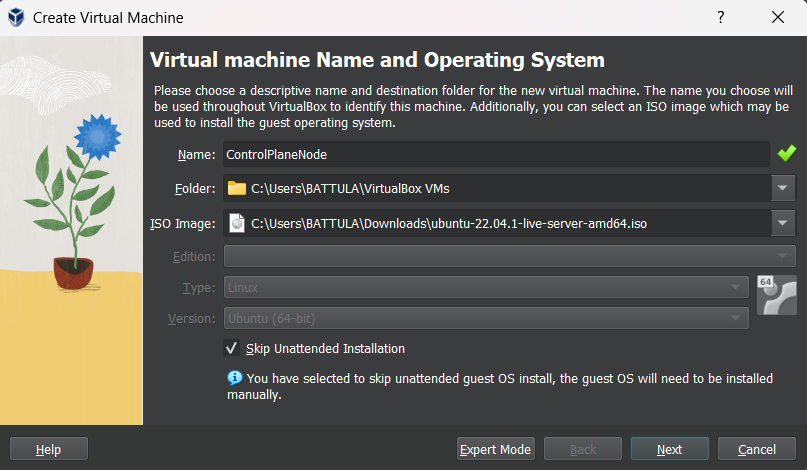
## 

Let us set up a kubernetes cluster on multi-node i.e., one node as control plane and another node as compute plane using the same CRI(containerd) and Kubeadm as we used in single-node setup.

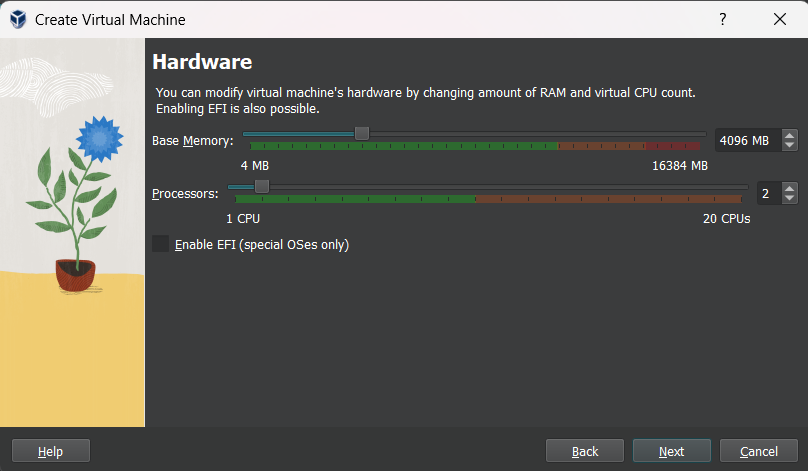
To set up a compute plane node, we can clone the existing control plane node and modify it or create one from scratch like we did in single-node setup and attach it to the control plane node.

Let us first setup a control plane node:

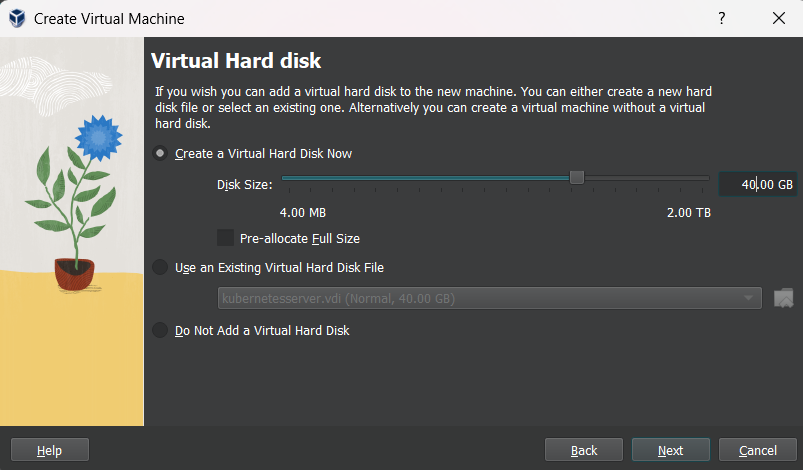
1. Let us first create a VM with ubuntu os as iso image.



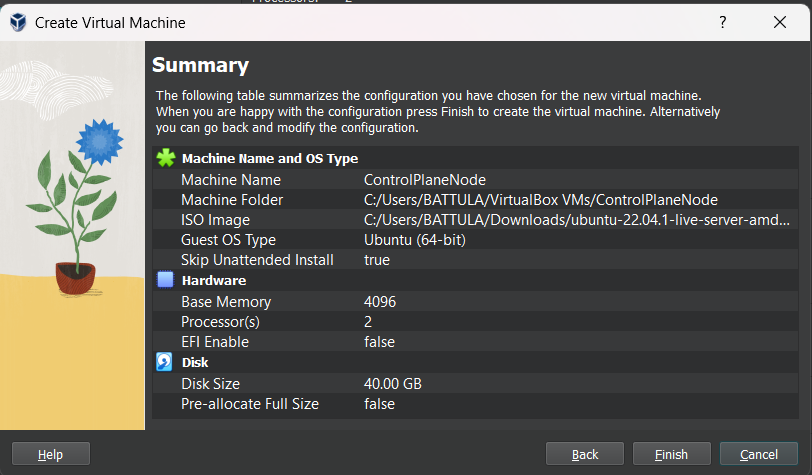
1. Now let us go through the hardware configuration that we are going to assign for the VM. We are using 4GB RAM and 2 cores of CPU.

****

1. For storage, it is optimal to assign any convenient storage(40-50GB in our case) for our VM.



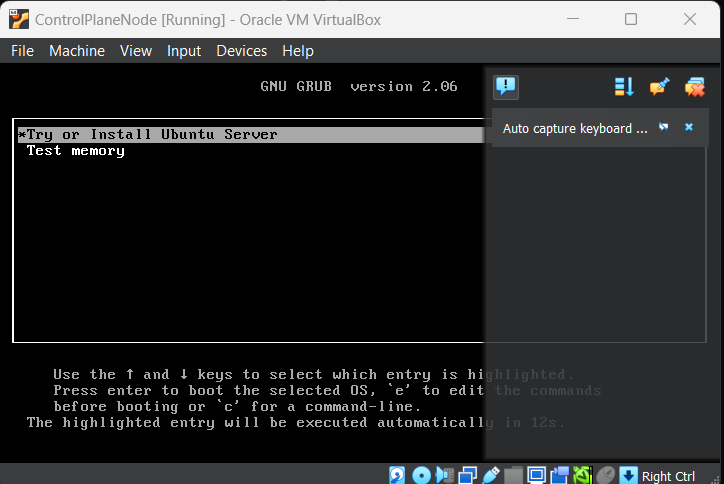
1. Here we can see the summary of the configurations we have selected.



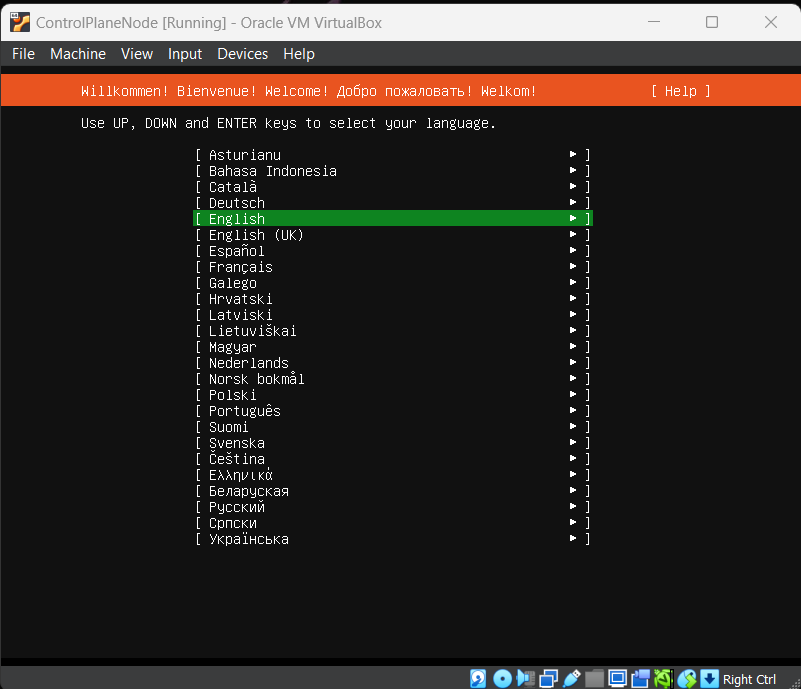
1. Upon completion, we shall start the VM and install all the necessary packages.



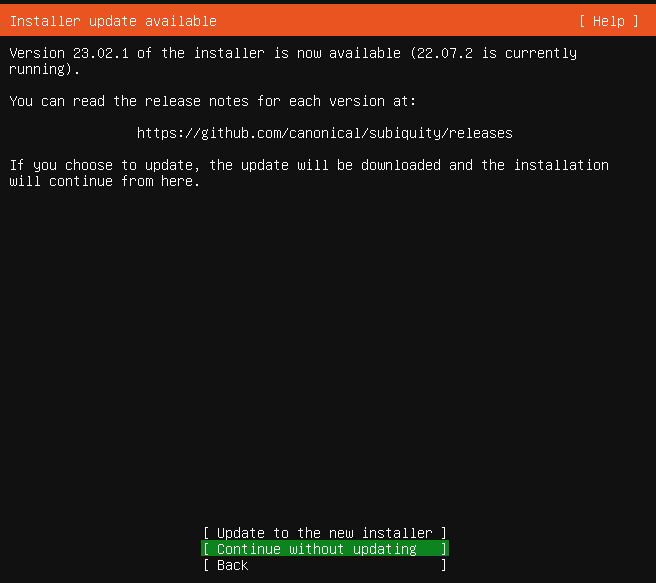
1. We can just proceed with this step.



1. After some time, it completes the boot up process and we can then start our installation process. We will now be prompted to select the language we wish to proceed with.

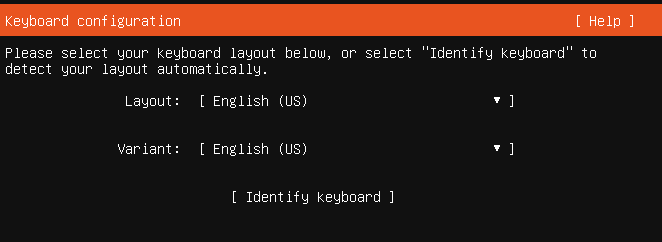


1. Whenever a new version of the iso image is available, the VM gives us an option to whether to continue with the existing version or update to the latest version.

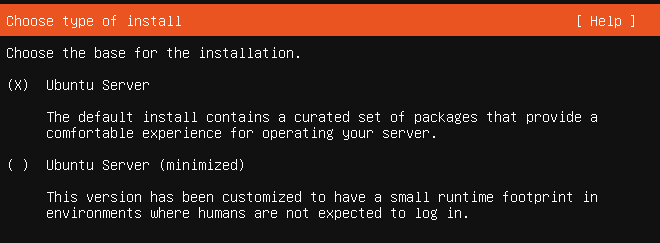


In our case, we are using Ubuntu 22.04 LTS version which is stable and bugs free. So, we are good to go with it.

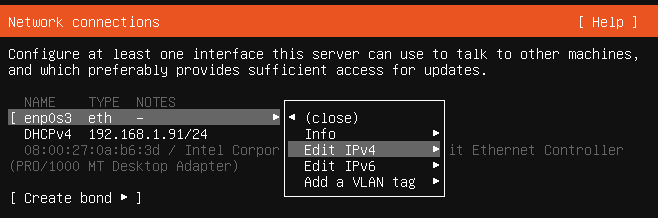
1. Now we can see an option to select our keyboard language. Select the optimum language and continue to the next step.



1. In this step, we are now asked whether we would like to install the default Ubuntu base packages or whether to install a minimized version of Ubuntu.



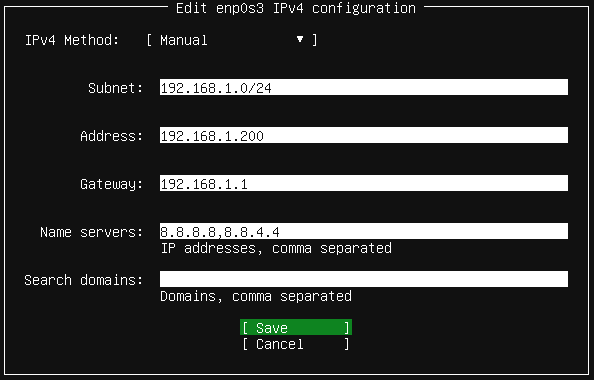
1. Next, we come across network settings where we can change our VM’s network IP address.



Here we are going to set a static IP address to our VM as normally the VM’s IP address might change every other time we boot up our VM.

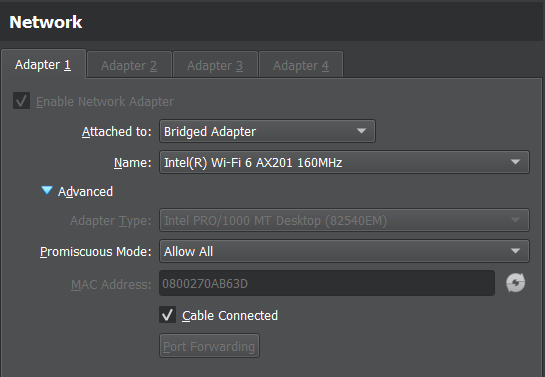


Here we are going to proceed with manual updation of our IP address.

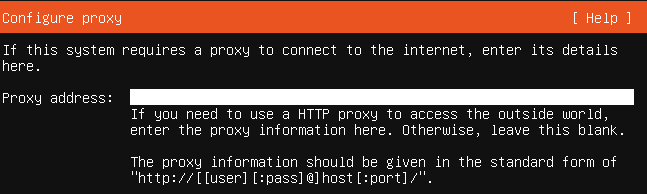


Here we are using google’s default nameservers(8.8.8.8 and 8.8.4.4).

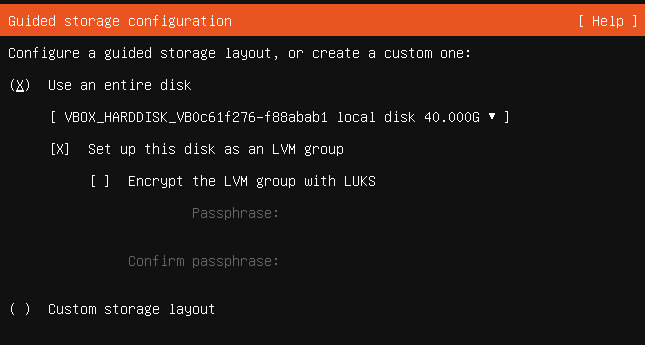
***Note***: Before that, we need to change the network type in VM settings from NAT to Bridged Adapter. This step can be performed right after assigning the hardware configurations to the VM.



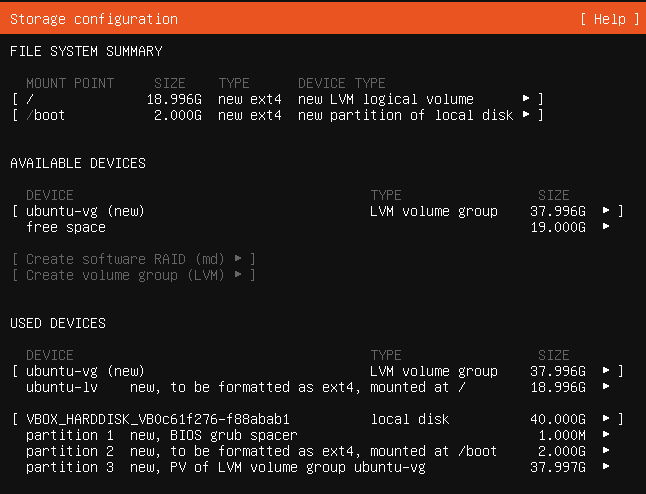
1. Configuring proxy and mirror address are not the things we are going to use, so we shall skip them.



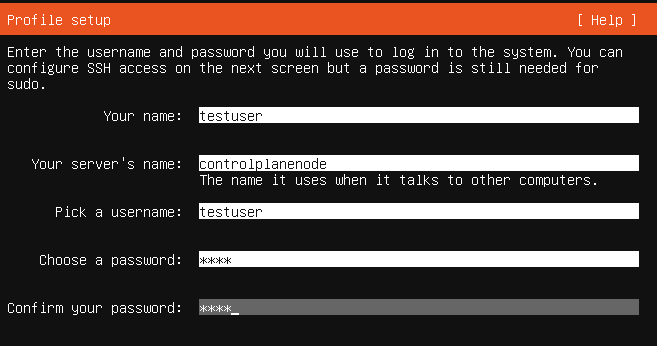
1. We can modify the storage type, but here we shall proceed with the default LVM partition by the VM.



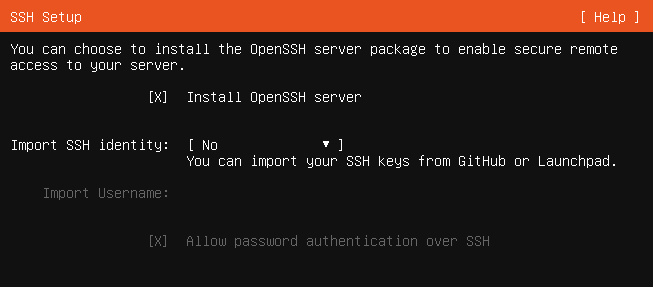
1. Here we can see the configuration details.



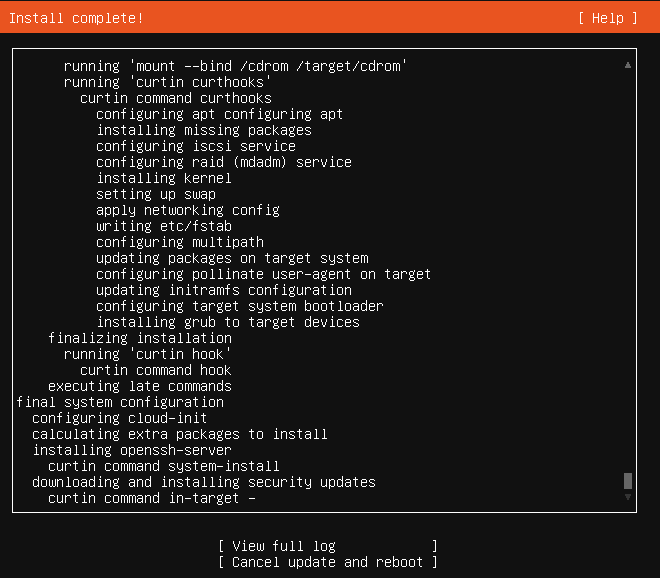
1. Now we can fill up the details for profile setup.



1. Now this is an important step as we can communicate with the VM from remote servers or from CMD, mobaXterm or git, etc,. So, we shall turn on the “Install OpenSSH server” option.



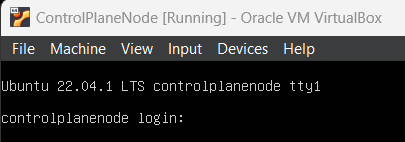
1. We can skip this step as we can install all these packages separately later based on our requirements.
2. Now the installation process starts and we can also see the log files if we want to.



1. After completion we can see this option, we proceed with the reboot now option.

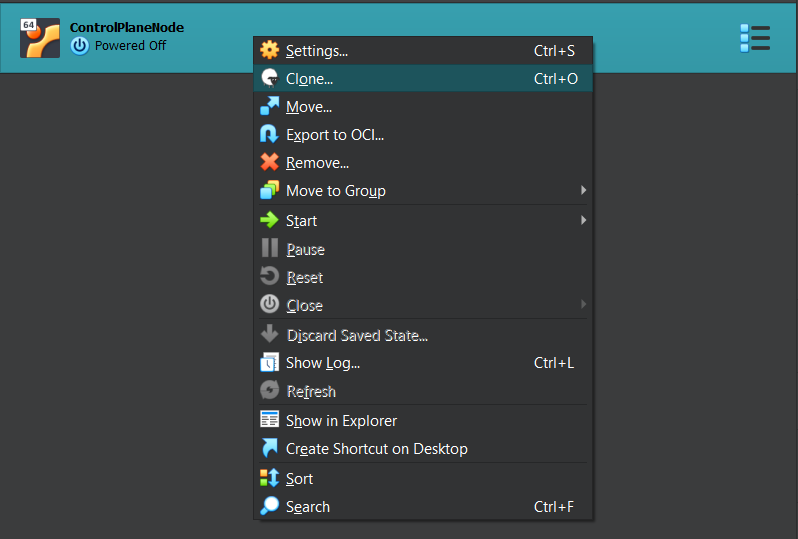


1. Upon successful installation, we can see that the VM is open successfully and it is completely functional.

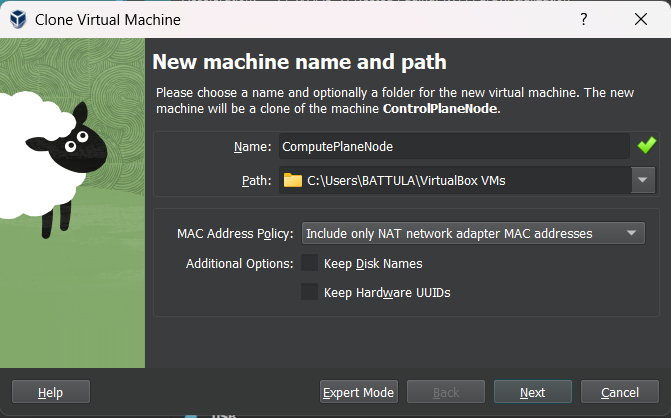


Now in-order to create a compute plane node, we can completely follow the above steps and create a new VM. Instead we can simply clone the VM we just created and modify it.

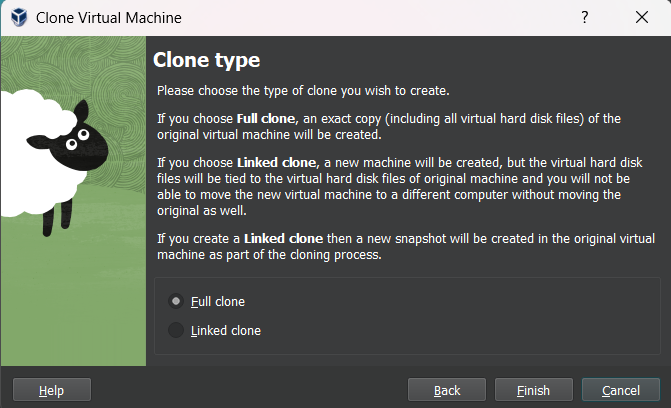
1. We can select the following clone option:



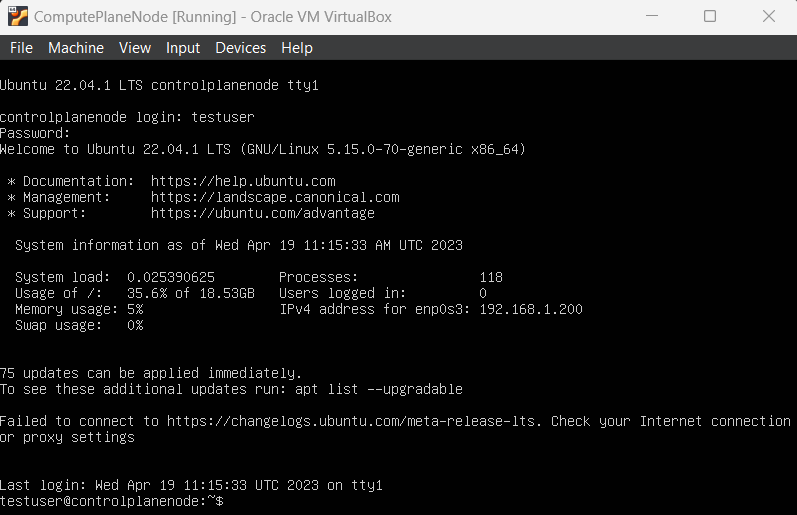
1. Now this pop-up appears, where we name our clone server.



1. Now, we go with the full clone option.



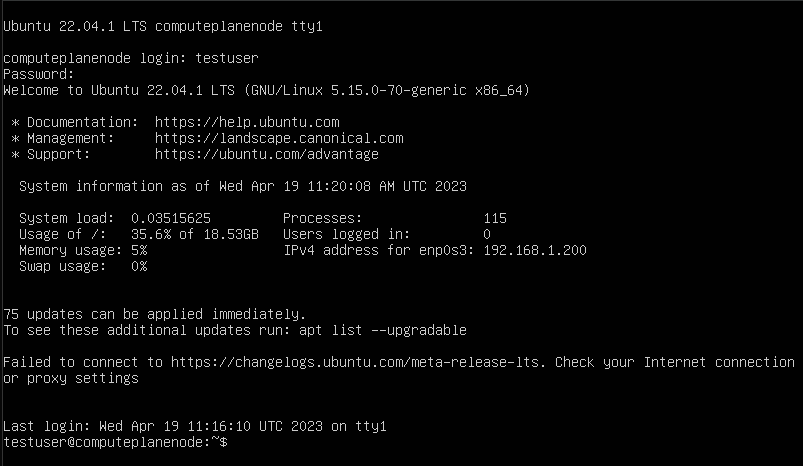
1. Now if we launch the clone VM, we can see that all the details are the same as the original VM including the login details.



1. There are two complexities we have over here, first of all we have to change the hostname of the VM and the other is to change the IP address.

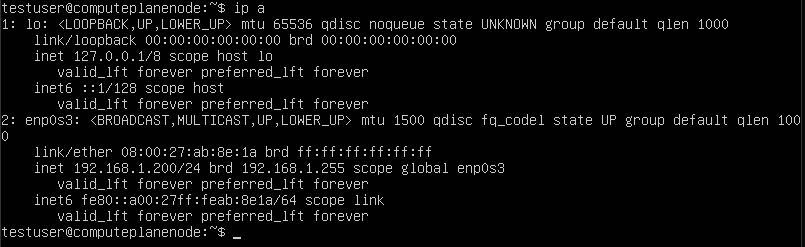


In order to view the changes, we have to reauthenticate into the VM.

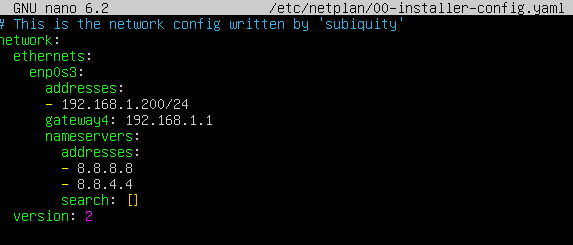


As we can see, the changes have been reflected effectively.

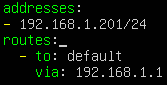
1. We can see that the IP address is also the same as before. So, let us change the IP address and give a static IP address to the VM.



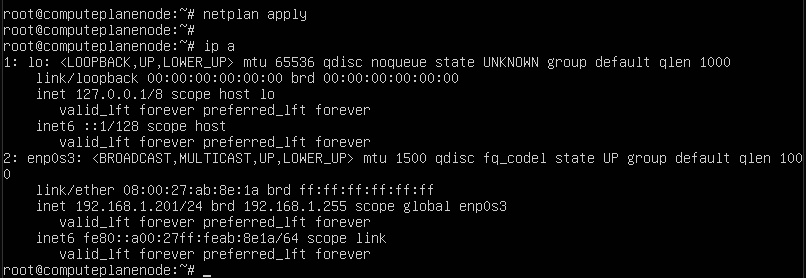
1. In order to change the IP address and options, we have to modify the netplan config file.



1. The few changes that we are going to make here are just the addresses tab and instead of gateway4 variable, we are going to use routes variable, as gateway4 variable has been deprecated.



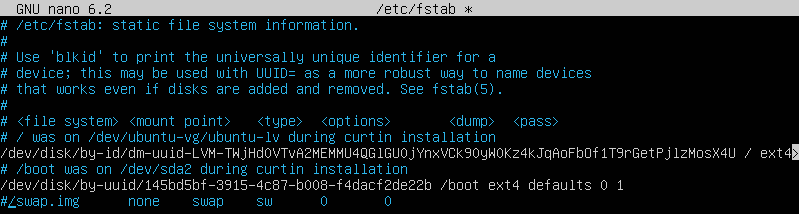
1. Now, we apply the changes and then the changes will be reflected.

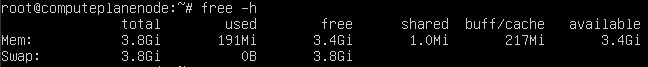
****

Now that both the VMs are up and ready, we can now install the common packages for both the VMs first and then custom install the packages based on the requirements.

As we are using the VMs for setting up a kubernetes cluster, we shall disable swap memory which is recommended to be done by kubernetes.

To disable swap memory, we have to modify the fstab file under /etc/fstab. Just comment out the last line in the file and save changes.

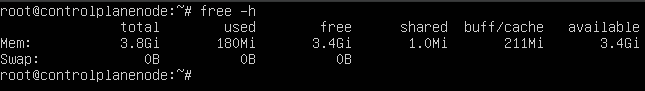




Now, we can use swapoff command to turn off swap memory.



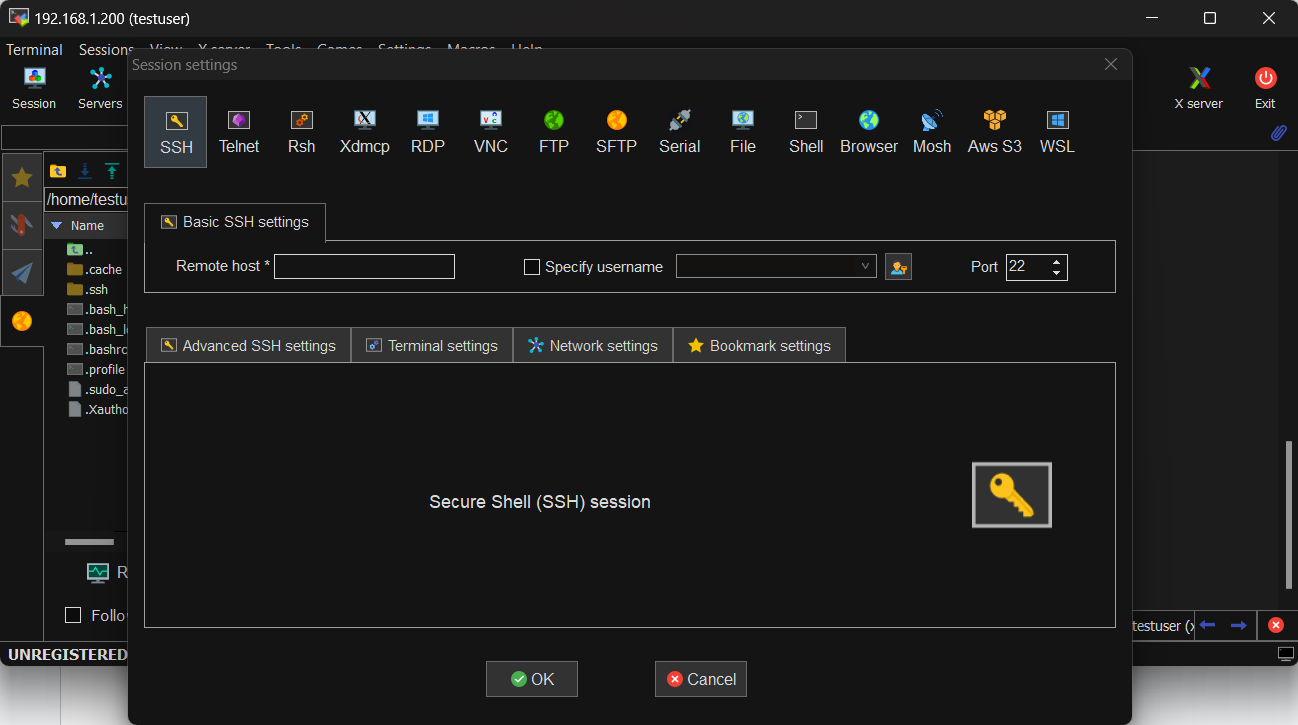
Now we can see that the swap memory is totally turned off. This is the same process for control plane node.



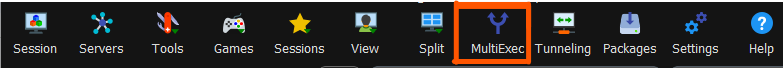
Hereon, we can follow the same steps to install the CRI which is containerd in this case.

***Note***: We can use a tool called MobaXterm in which we have an option to multi-execute. Which means that we can make changes in multiple VMs at the same time.

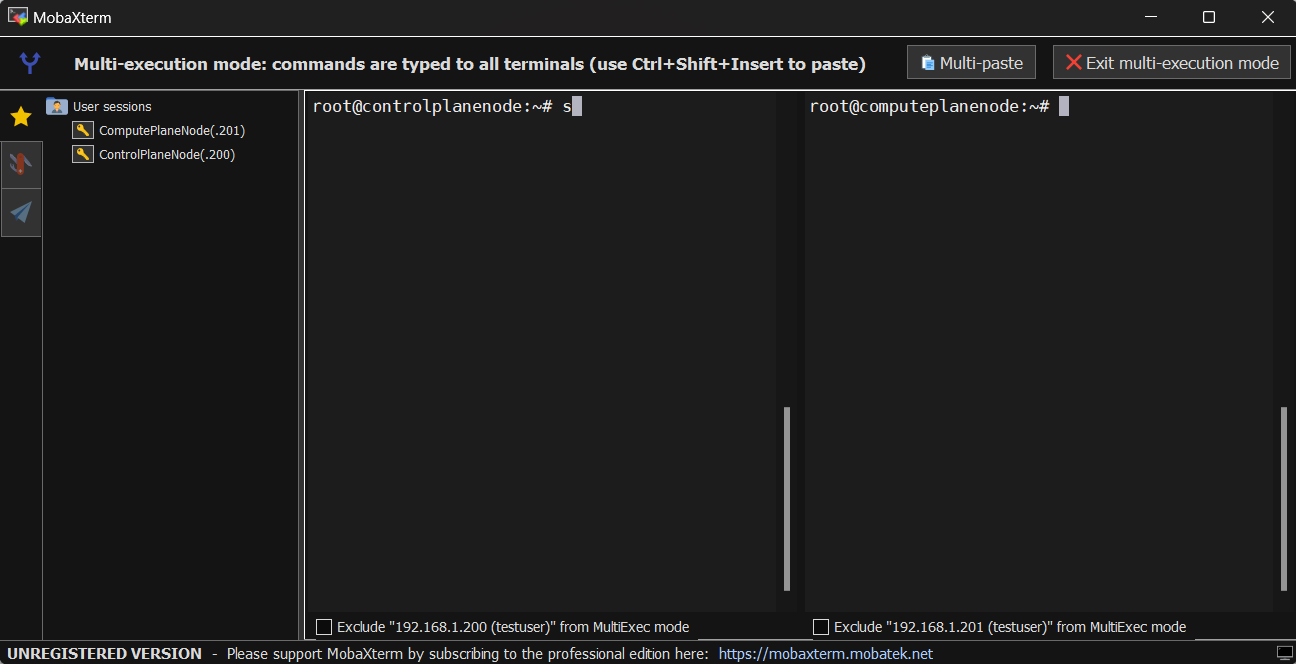
We can open a session in MobaXterm by:



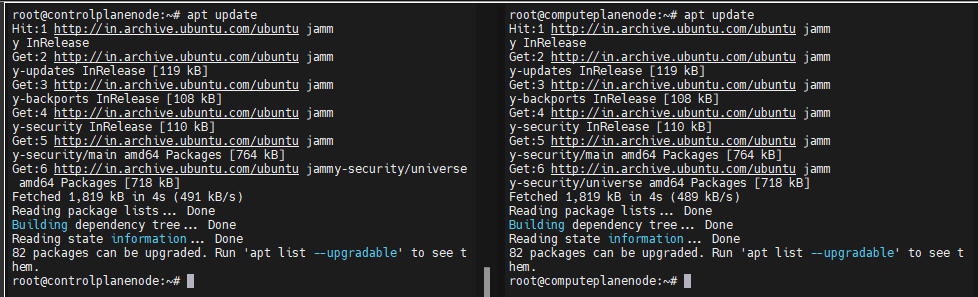
Now, upon opening both the nodes, we can use the MultiExec option which stands for multi execution.



In general, it looks like this:



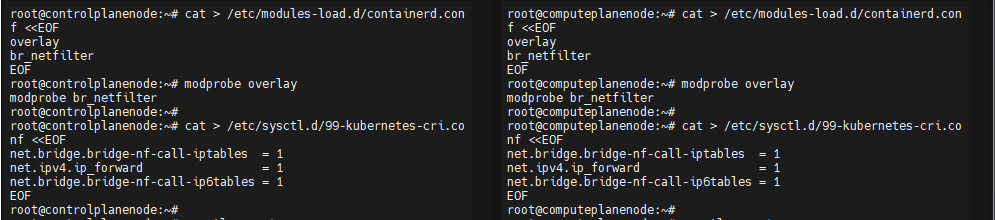
1. As the VMs are newly created, as a rule of thumb we need to update the VMs using the ***“apt update”*** command.



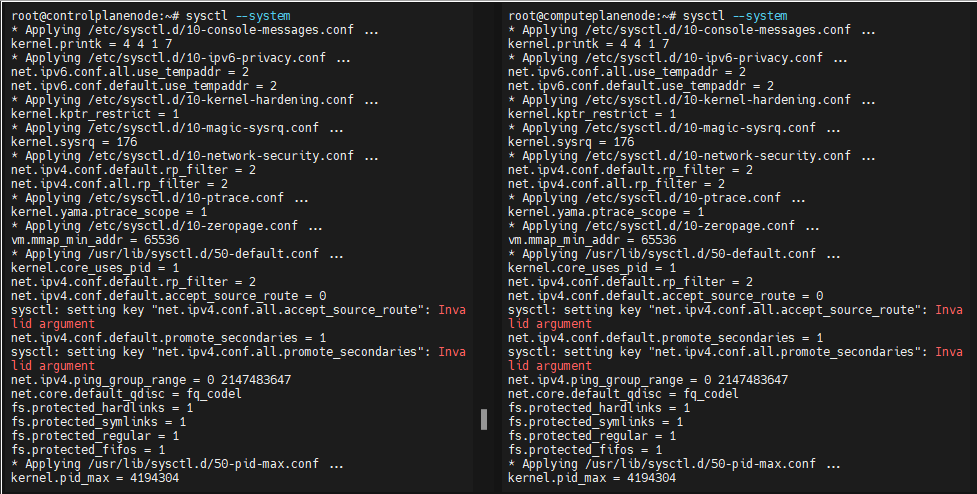
In case there are any upgradable files, we can use the **“*apt upgrade*”** command.

Upon the completion of these updates, we can proceed to the further step, where we are going to install the gpg keys and required packages to set up the containerd CRI which is common for both the nodes.

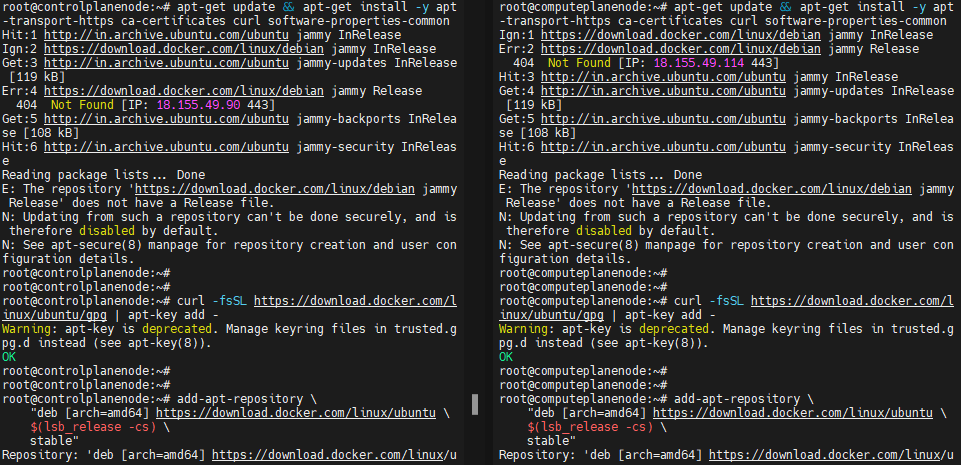
1. Now on all nodes, we need to update the containerd config files, set up required sysctl params:



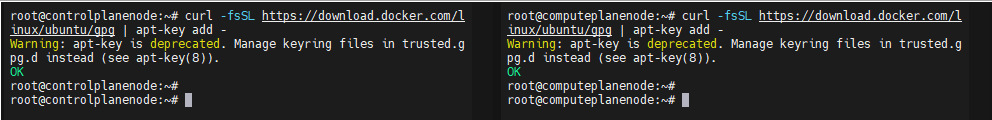
1. Now, upon the modification, we need to use the command ***“sysctl –system”***.



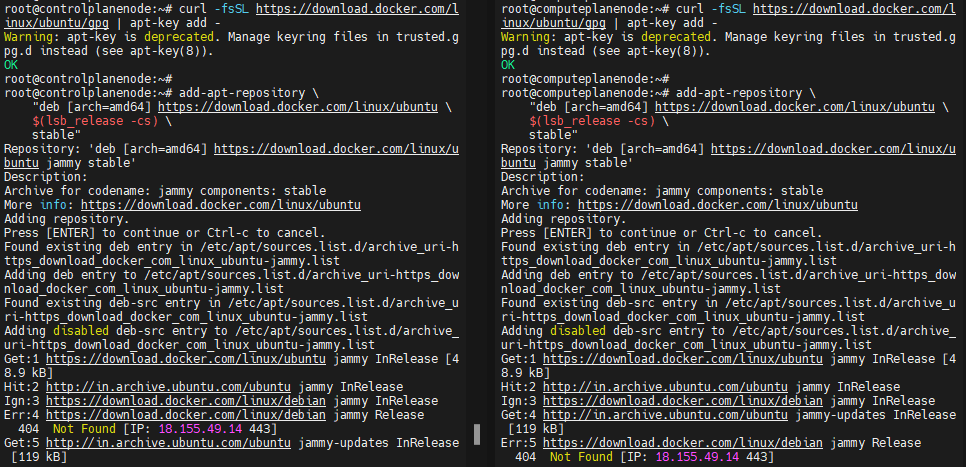
1. Now, after the sysctl parameters setup, we need to install the containerd package and requirements from docker official website.



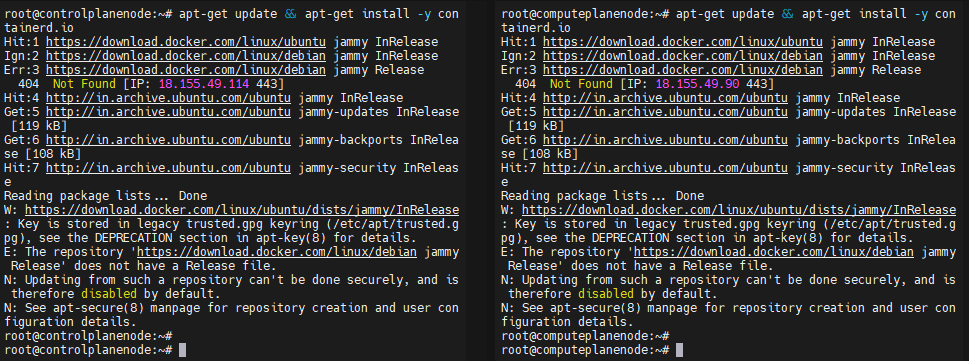
1. Docker’s official GPG key:



1. Adding docker’s apt repository

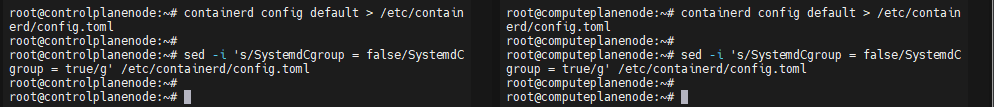


1. Finally we can install the containerd.io package



1. Now we can configure containerd. First, we create a repository,

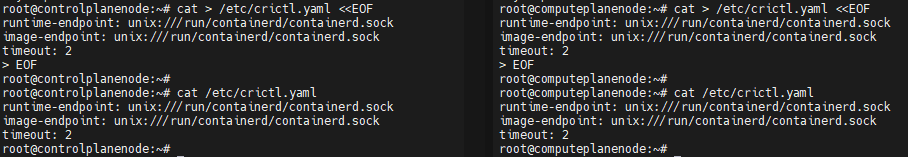




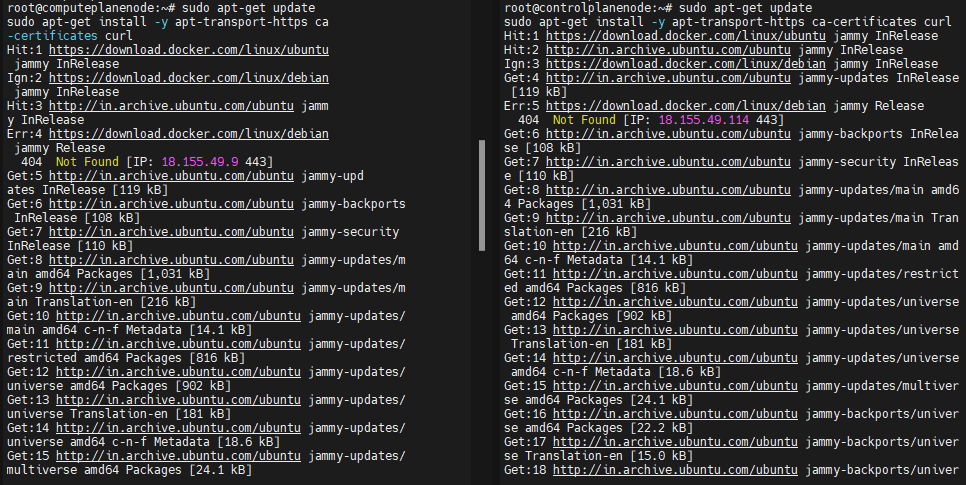
1. Now we restart containerd



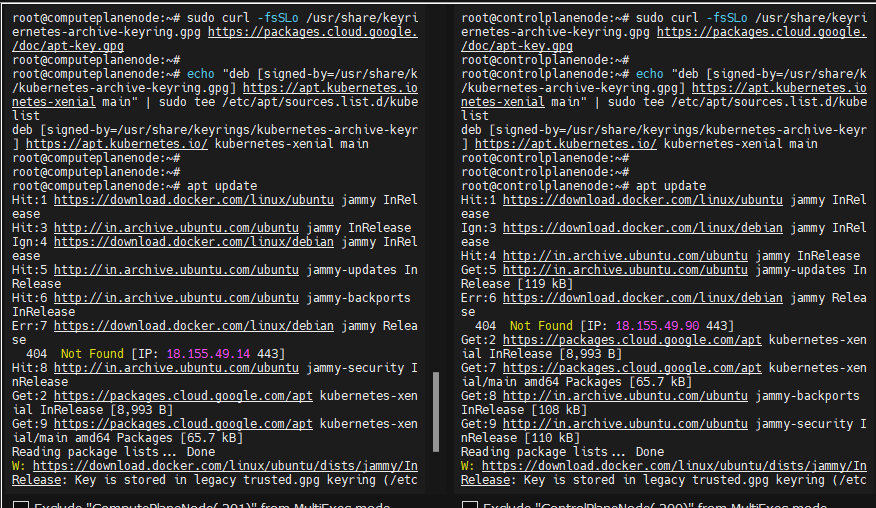
1. To execute crictl CLI commands, ensure we create a configuration file as mentioned below.



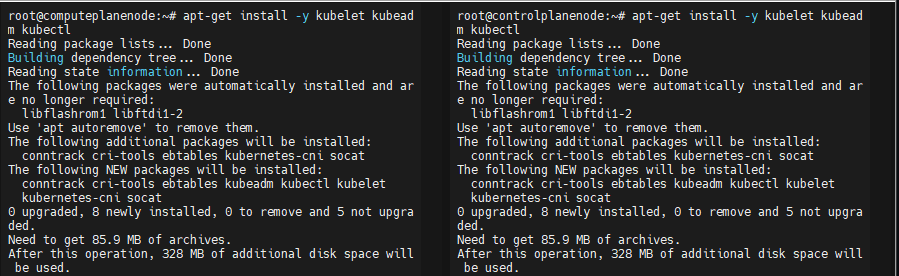
1. Now, after the above process is done, we can now install some kubernetes packages



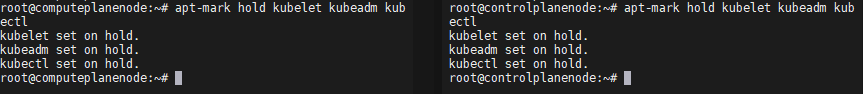
1. Now, we are downloading the gpg key for the kubernetes repository and creating the repository in our filesystem.



1. Install the necessary kubernetes packages



1. To ensure that we don’t face issues any further, we use hold updates.

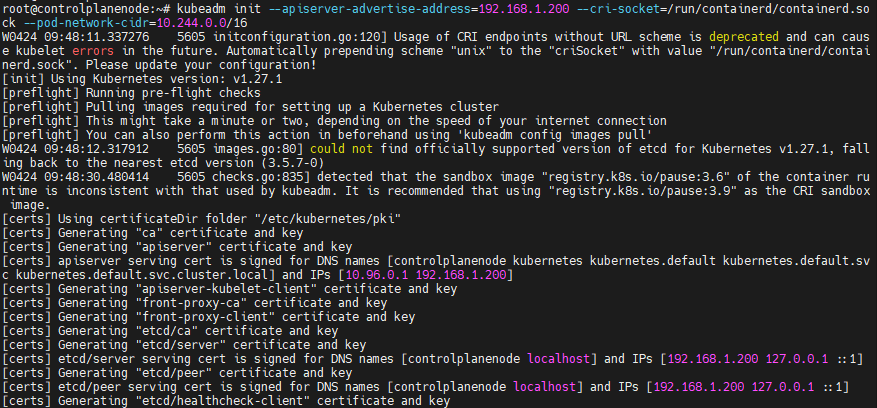


1. From this point, we need to follow different procedures to complete the setup. So, we shall exit the multi-exec mode.

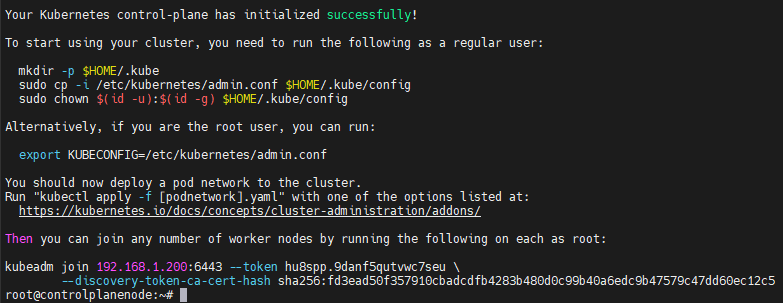


### On Control Plane Node:

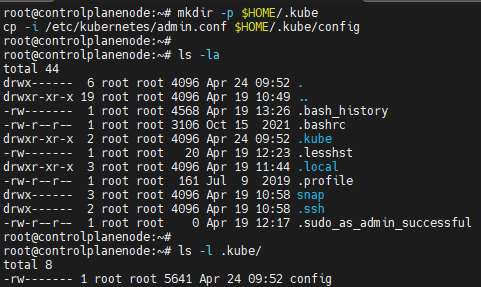
1. To bootstrap the control-plane node, we have to use the **“kubeadm init”** command.



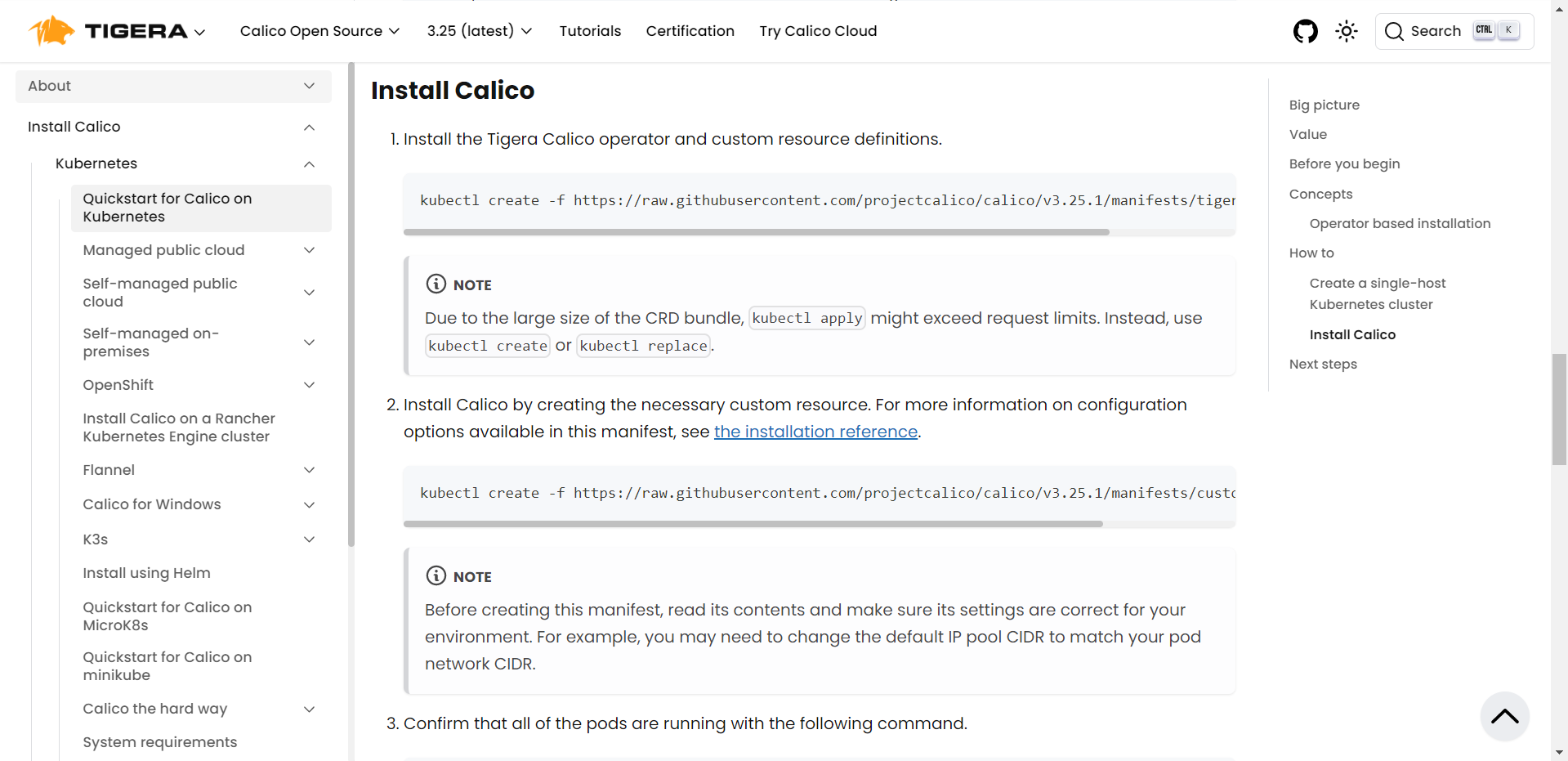
1. Upon completing the bootstrapping, we can see the instructions where we are prompted to create a directory for the config files.



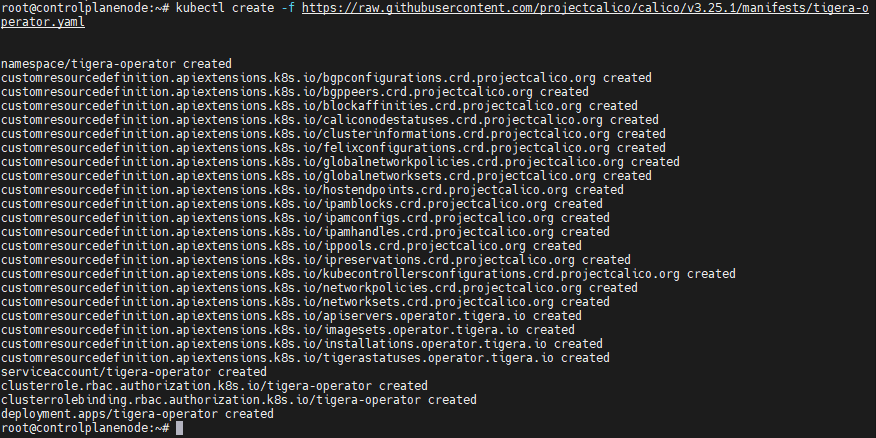
1. So, let us create a directory and copy the config file into it.



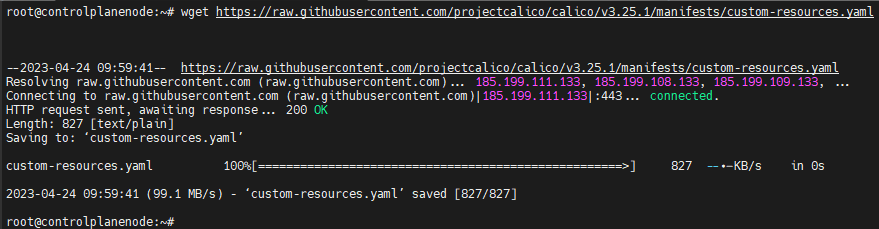
1. Now, we need a container network interface. Here, we are using calico CNI. We can access the calico documentation from the net to set up the CNI.



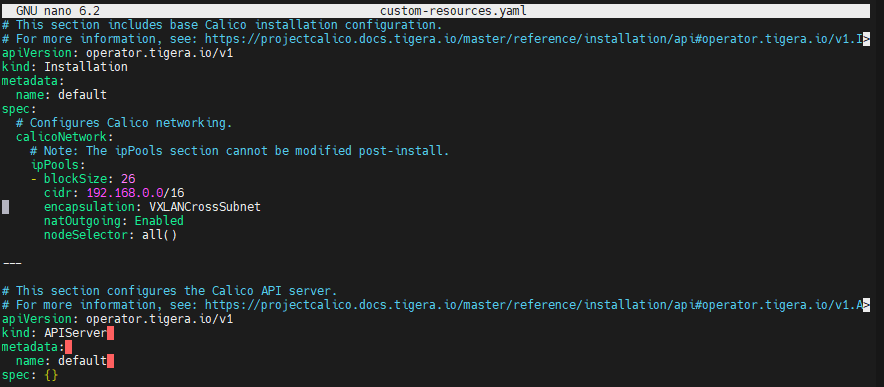
1. Firstly, we are going to install the calico operator



1. Now, we need to create and apply a custom resources file.



In this file, we need to change the cidr to the range that we have specified during the bootstrapping.



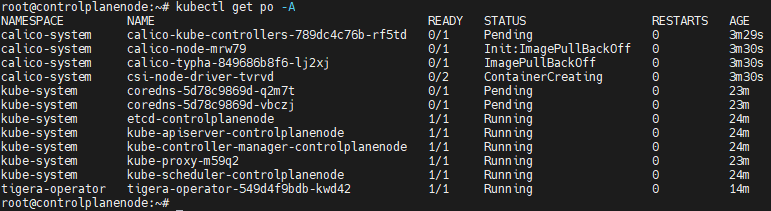
We change the cidr to:



After changing and saving the custom-resources file, we need to apply the changes.

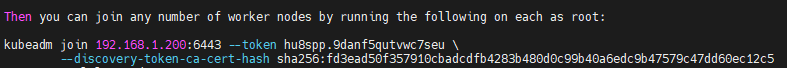


1. Now that we have completed the control-plane setup, we have to connect the compute-plane nodes to the control-plane node.

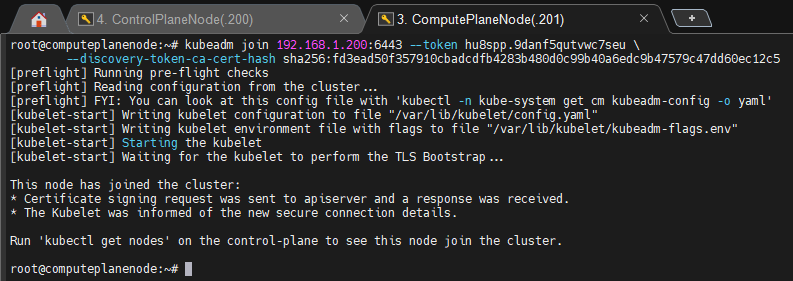


### On compute-plane node:

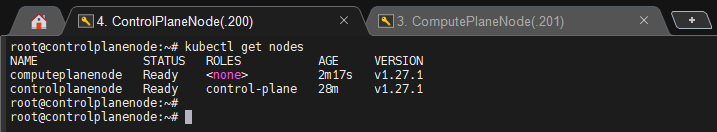
We shall use this command from the set of instructions we got after bootstrapping.



We need to enter this command in the compute-plane node that we like to join to the control-plane node.



Now, we can see the list of nodes on the control-plane node.



As we can see, we have successfully set up a multi-node kubernetes cluster using kubeadm.