

Kubernates\_Cluster

Runbook

Version: 1.0

Date: 10/29/2021

# Document details

|  |  |
| --- | --- |
| **File Name** | Kubernates\_Cluster\_Runbook.docx |
| **Version** | 2.0 |
| **Document Owner** | Ashish Anand |
| **Creation Date** | 10/29/2021 |

| **Version** | **Date** | **Author(s)** | **Notes** |
| --- | --- | --- | --- |
| 1.0 | 10/29/2021 | Ashish Anand | Comprehensive documentation for DevOps resource to create Kubernates cluster into local VM. |
| 1.1 | 11/15/2021 | Janmejaya Swain | Some modification done |

Contents

[Document details 2](#_Toc86422449)

[1. Document Overview 4](#_Toc86422450)

[2. Checklist to setup Kubernates cluster 4](#_Toc86422451)

[2.1 Prerequisite 4](#_Toc86422452)

[2.2 Installation 6](#_Toc86422453)

[2.3 Accessing Master and worker Node over ssh 20](#_Toc86422454)

[2.4 Configure Master node for Kubernates cluster 22](#_Toc86422455)

[2.5 Configure Worker node for Kubernates cluster 24](#_Toc86422456)

# Document Overview

This document is designed to describe End to End process to create Kubernates cluster on local VM. This document talks about spinning up Master node and one worker node to demonstrate a typical Kubernates cluster.

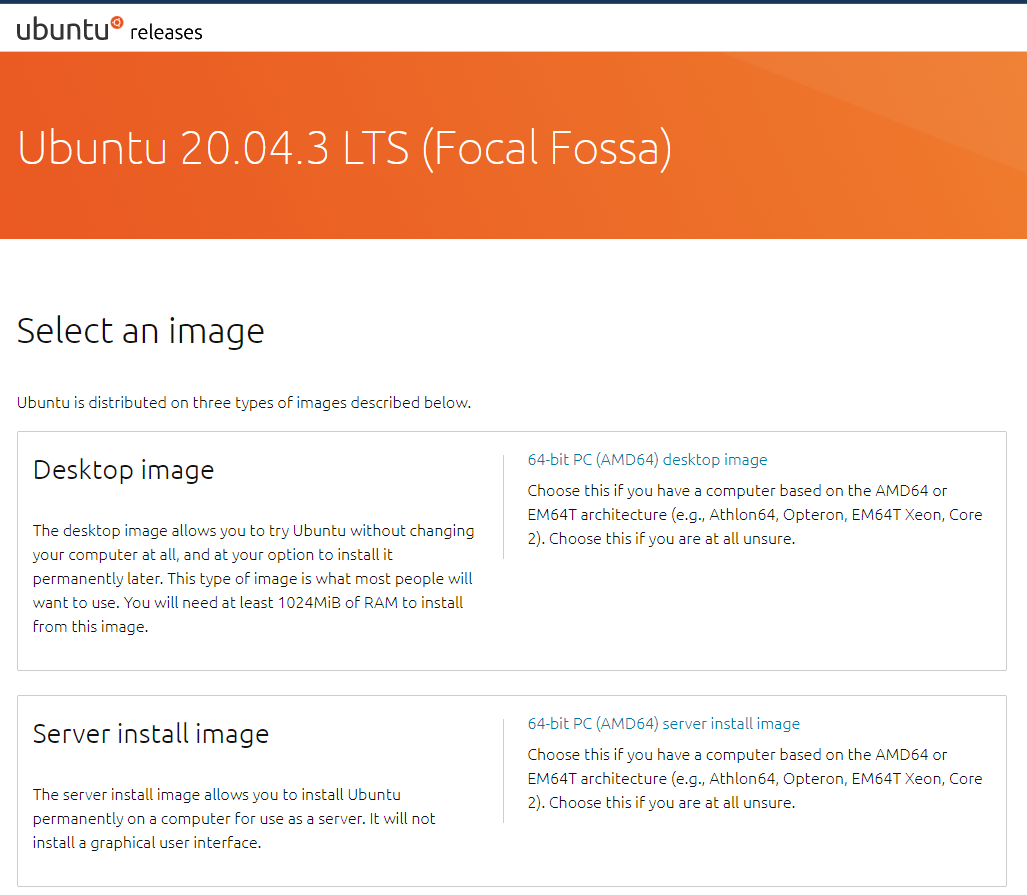
# Checklist to setup Kubernates cluster

## 2.1 Prerequisite

1. **Oracle virtual machine**, please use [Download link](https://www.virtualbox.org/wiki/Downloads) to download the latest version, currently latest version is 6.1.28, download the installer and install into your local machine.

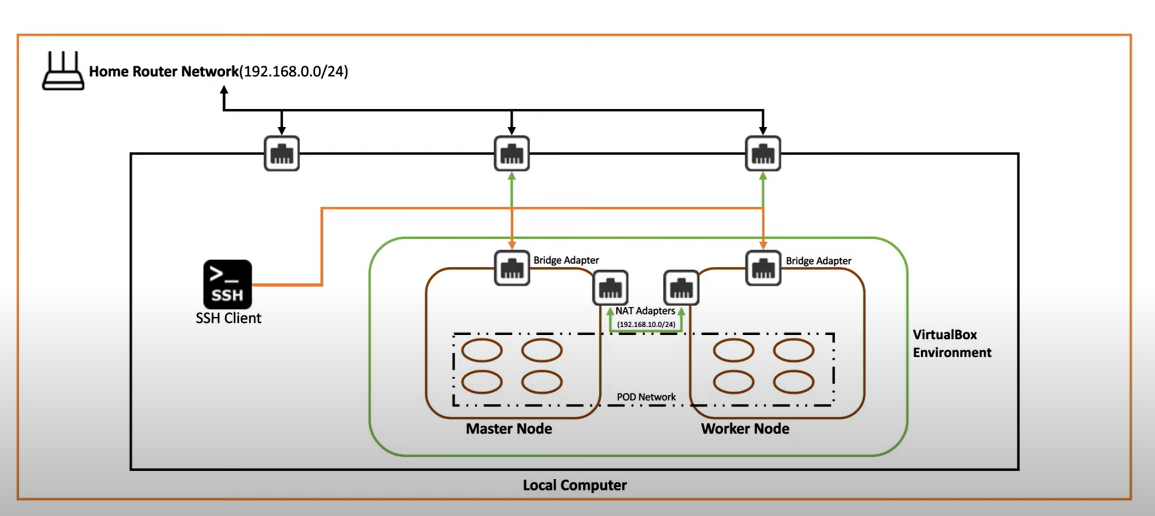


1. **Ubuntu iso binary file**, please use [Download link](https://releases.ubuntu.com/20.04/) to download the latest version, currently latest version is 20.04, download the .iso image for server Edition as we are not using any of the UI functionality, if you wanted to use the UI functionality use the desktop version, once downloaded save it into your project folder.



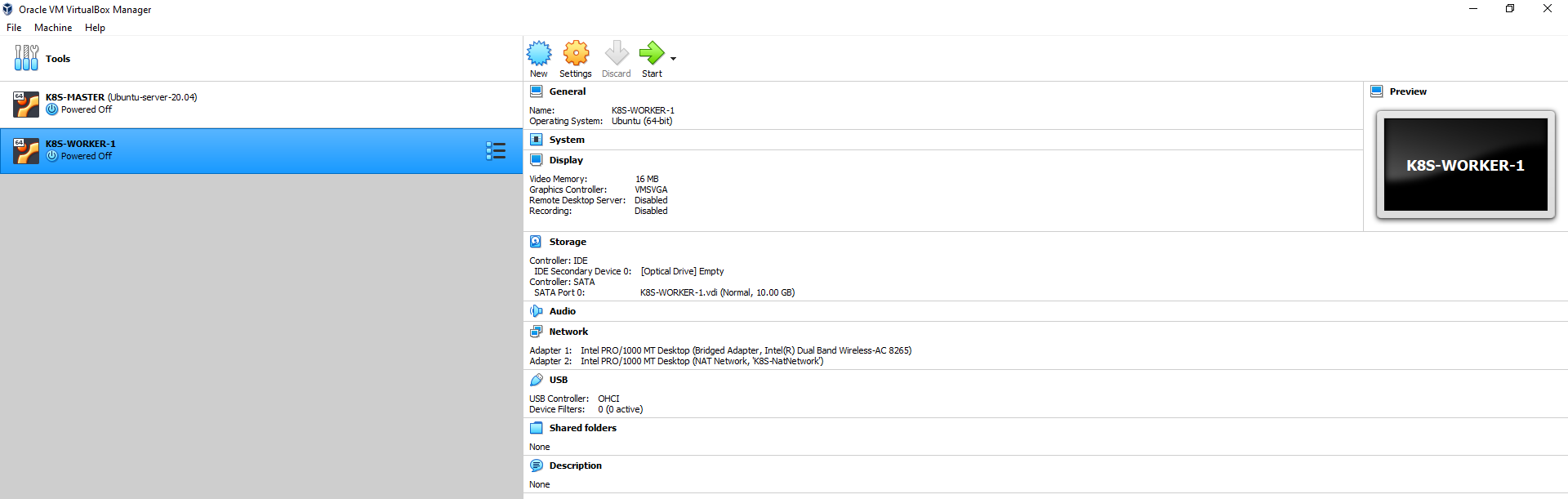
## Installation

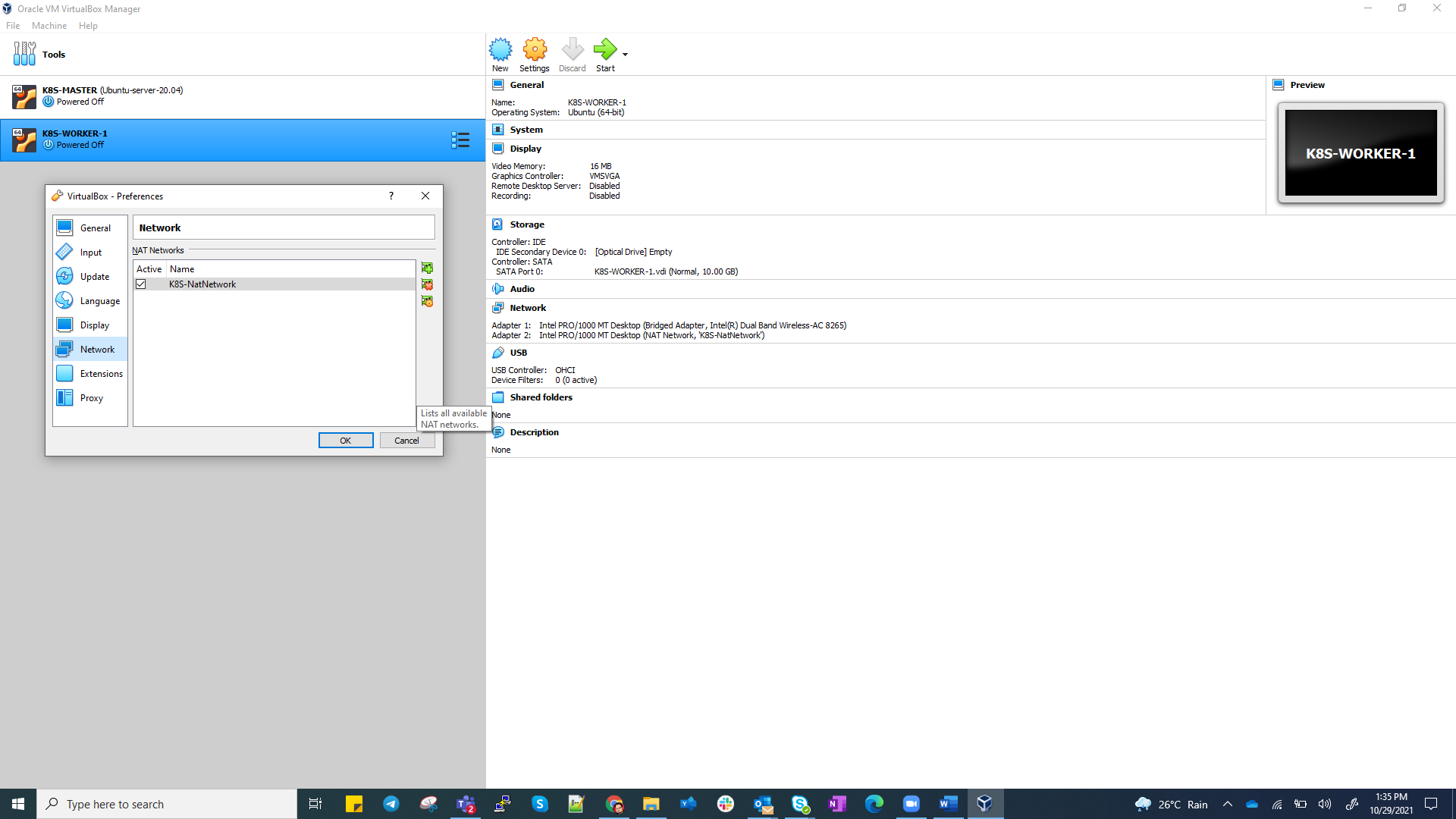
* + - * **Install Oracle VM into local machine**
      * **Network setup for Kubernates cluster:**



**Steps:**

1. Open oracle VM dashboard



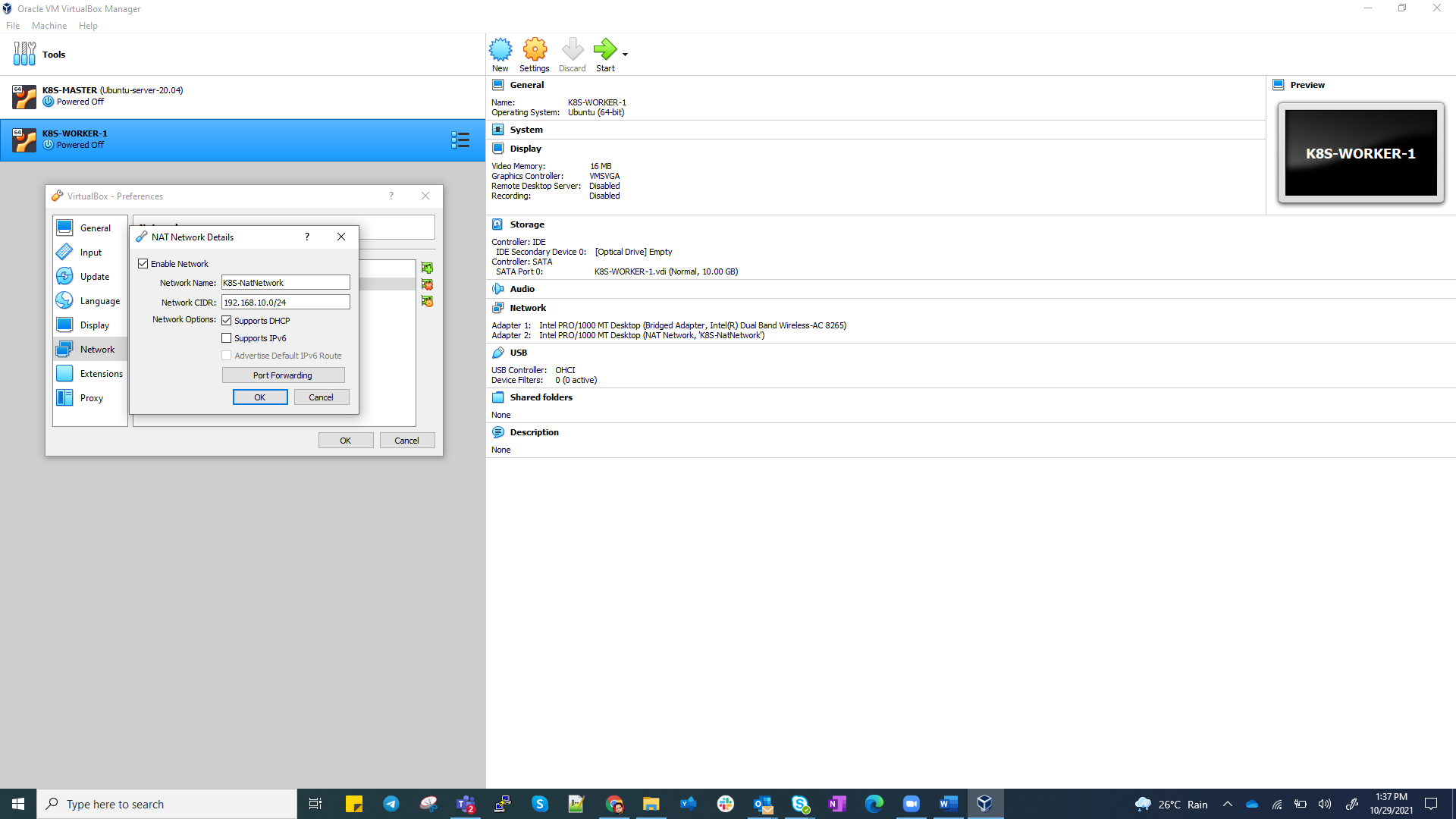
1. Navigate to file -> preferences -> network.
2. Click Adds new NAT network, and key in below values:

**Network Name**: Any name to your NAT network (keep it more relevant while naming).

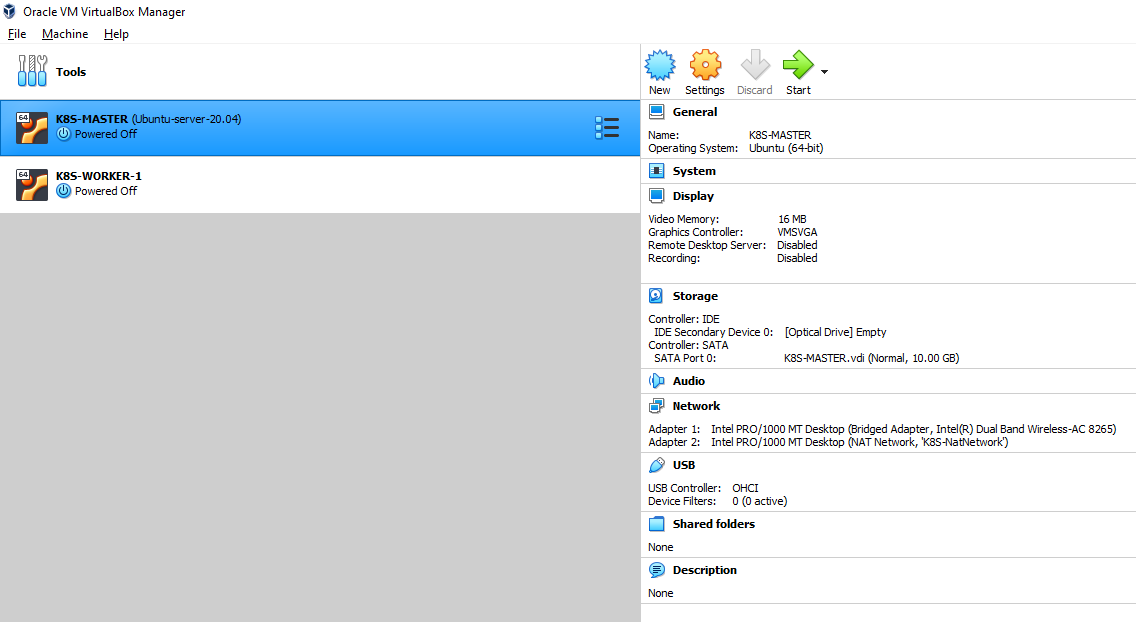
**Network CIDR**: 192.168.10.0/24.

**Network Options**: check box – supports DHCP.

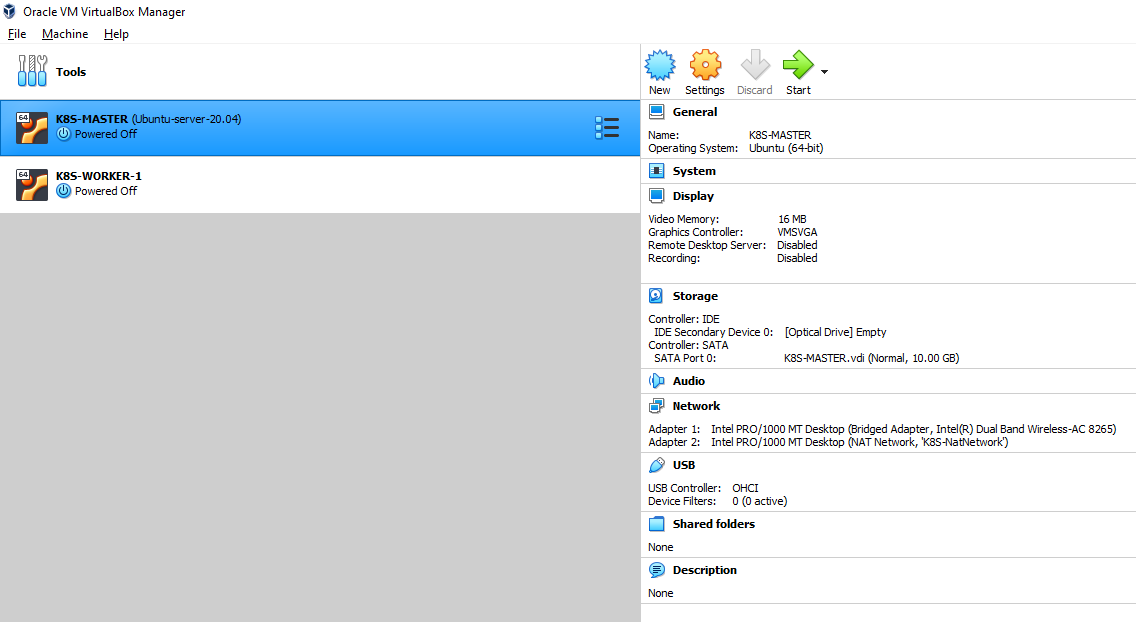
Click OK.



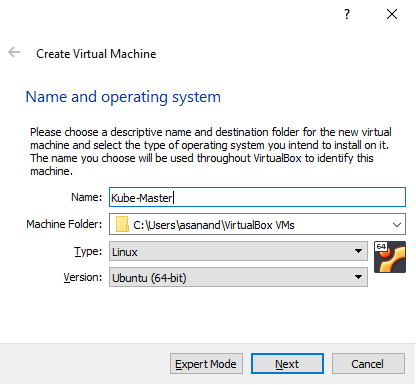
* + - * Create Master and worker Node using the ubuntu .iso file downloaded in the prerequisite steps, please refer **Jenkins\_session\_runbook.docx** shared earlier for detail steps to create VM’s. create Master and worker nodes as below.



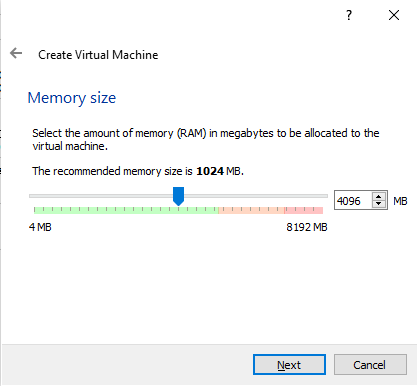
1. Create Kube-master node and worker node.
   1. Click **New**



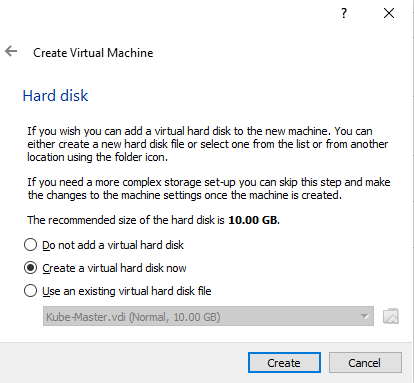
* 1. Enter Node detail as below and click next



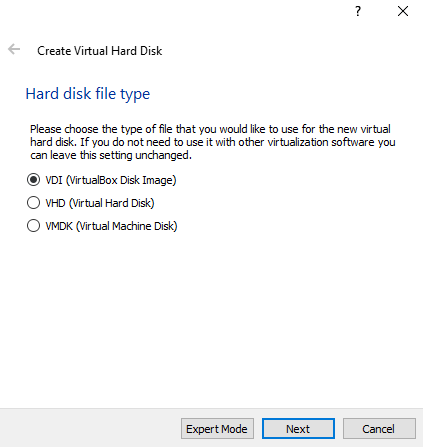
* 1. Allocate memory size to 4 GB



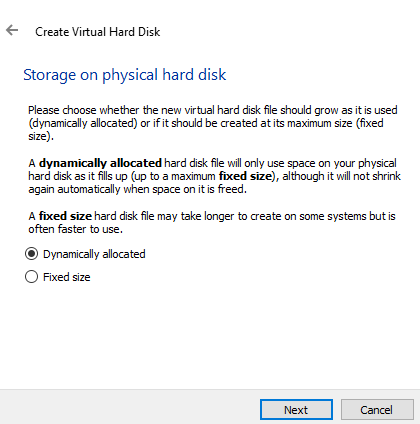
* 1. Select below option and click **create.**



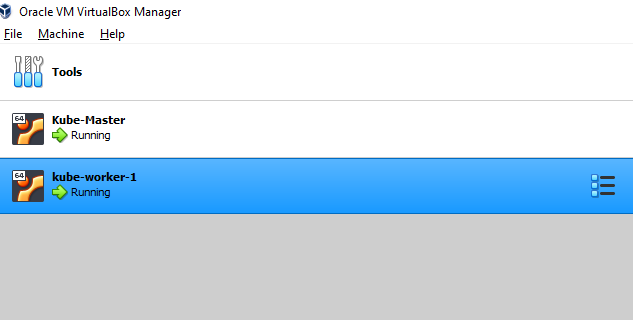
* 1. Select below option and click **Next.**



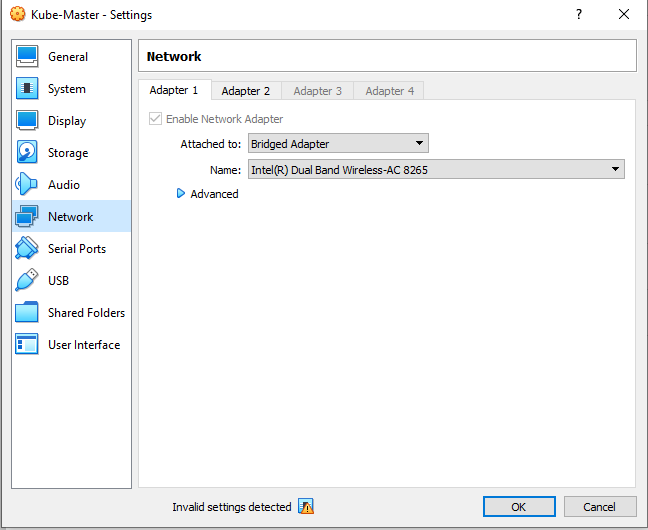
* 1. Select below option and click **Next.**

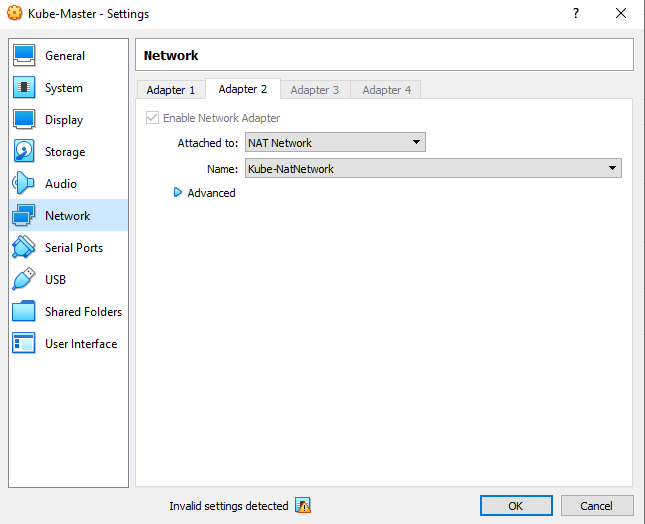


* 1. Allocate location and size and click **Create.**
  2. Repeat same steps to create 2nd VM – “**Kube-worker-1**”

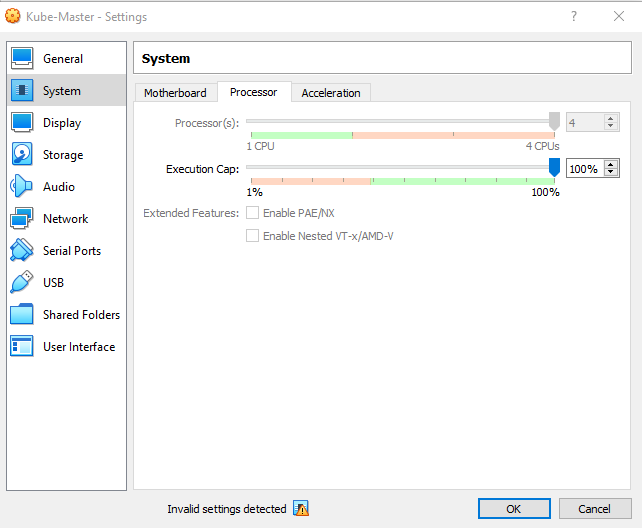


1. Global settings for Master and worker node.
   1. Select master node -> click setting -> network tab
      1. Adapter 1 -> select Bridged adapter
      2. Adapter 2 -> Select Nat network -> it should auto pickup Kube-NatNetwork created previously.

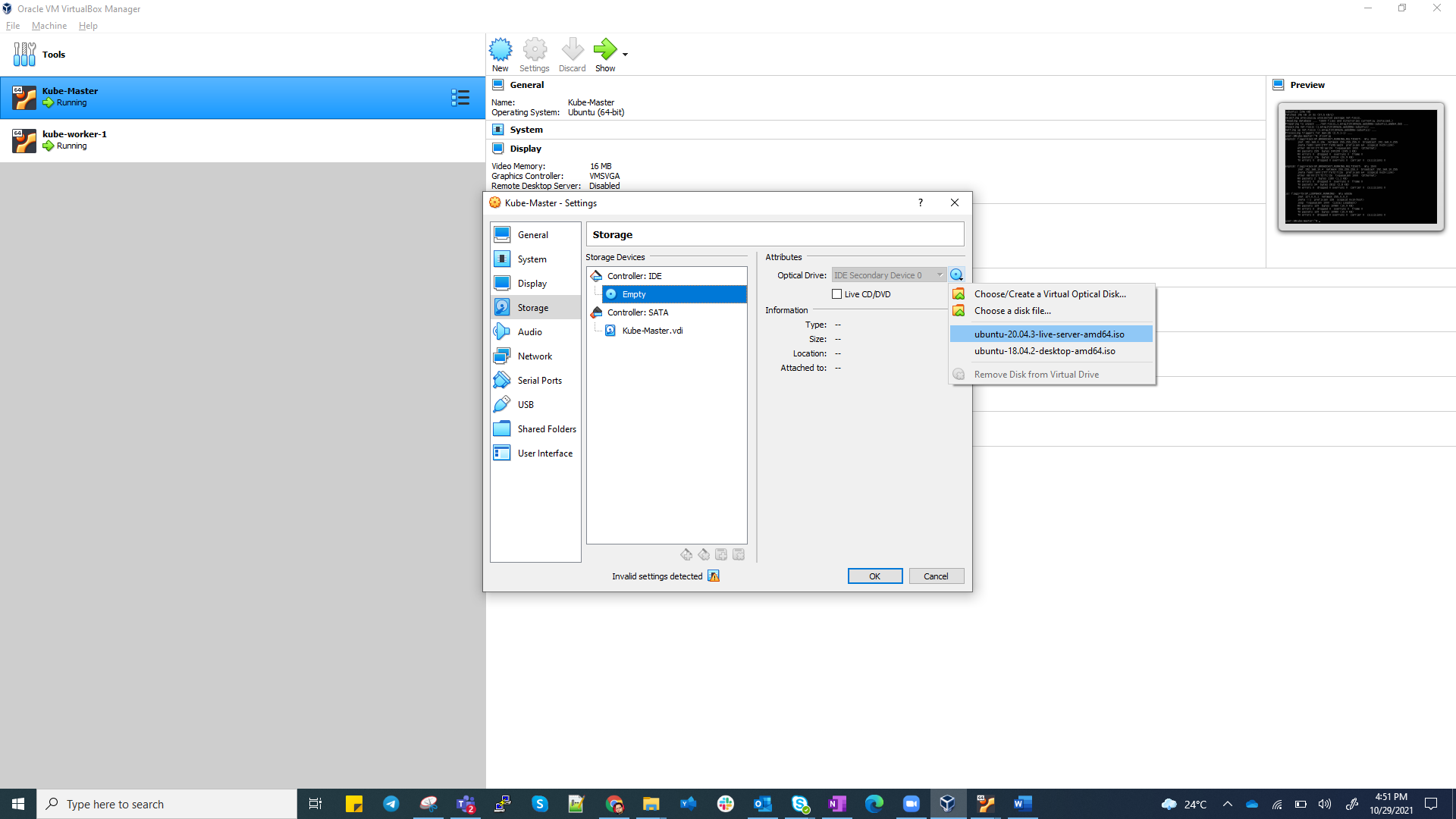




* 1. Navigate to -> System -> Processor-> Assign 4 CPU (Master), 2 CPU for Worker node.



* 1. Navigate to -> Storage -> Controller IDE -> Empty -> optical drive -> select ubuntu iso image for installation, click **Ok.**

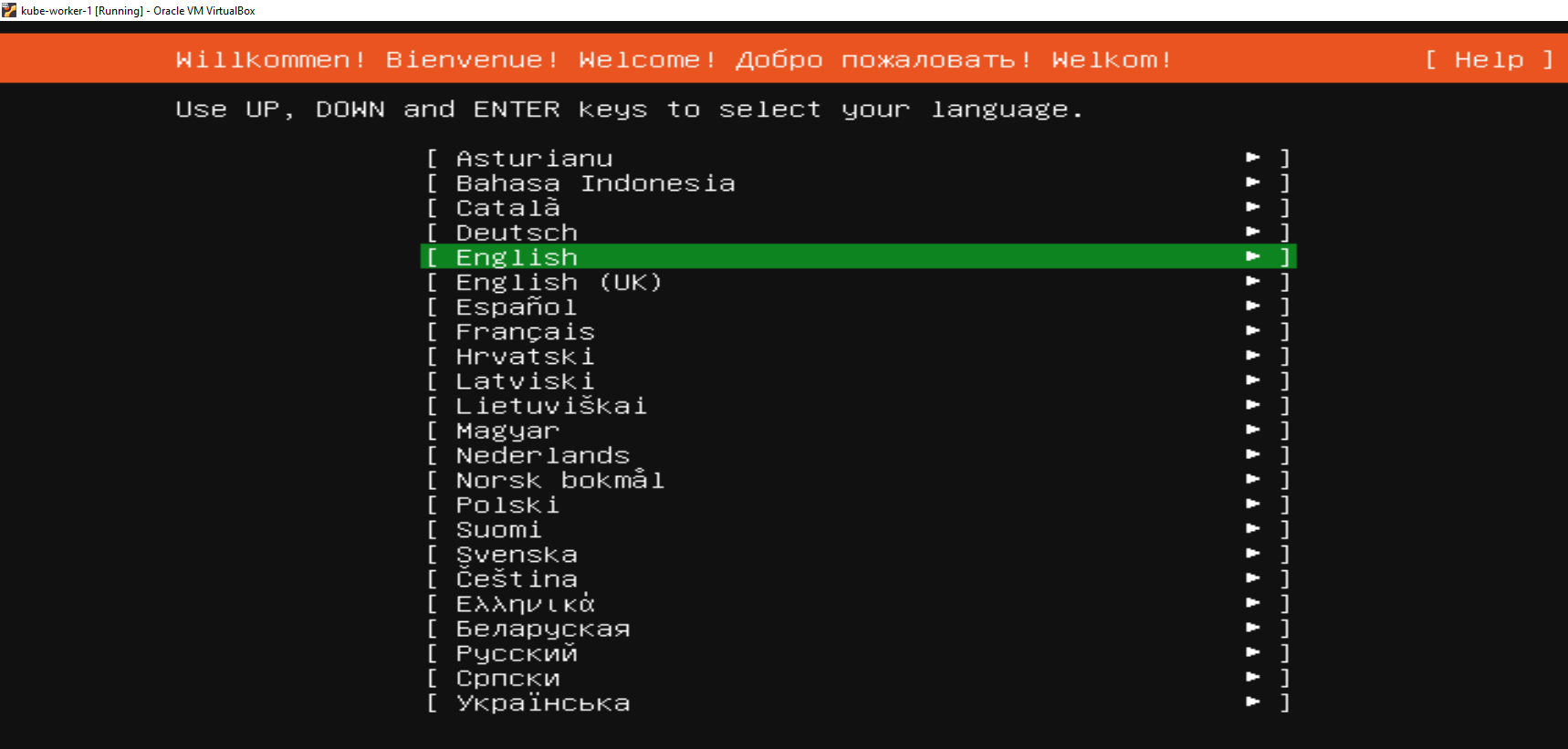


* 1. Repeat the same step (a-c) for worker node.

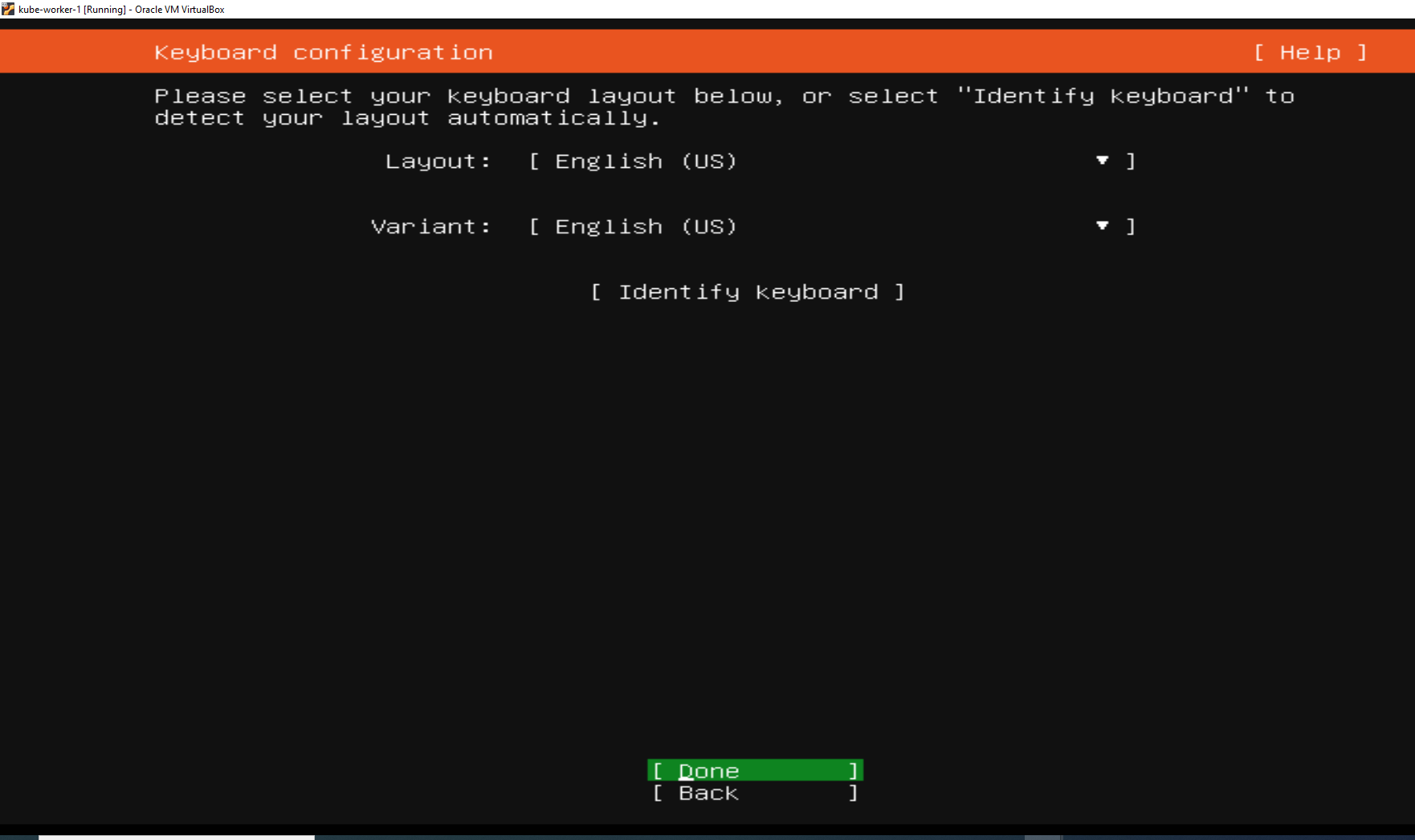
1. Install ubuntu server edition on both master and worker node.

Start the node and follow below screen for installation

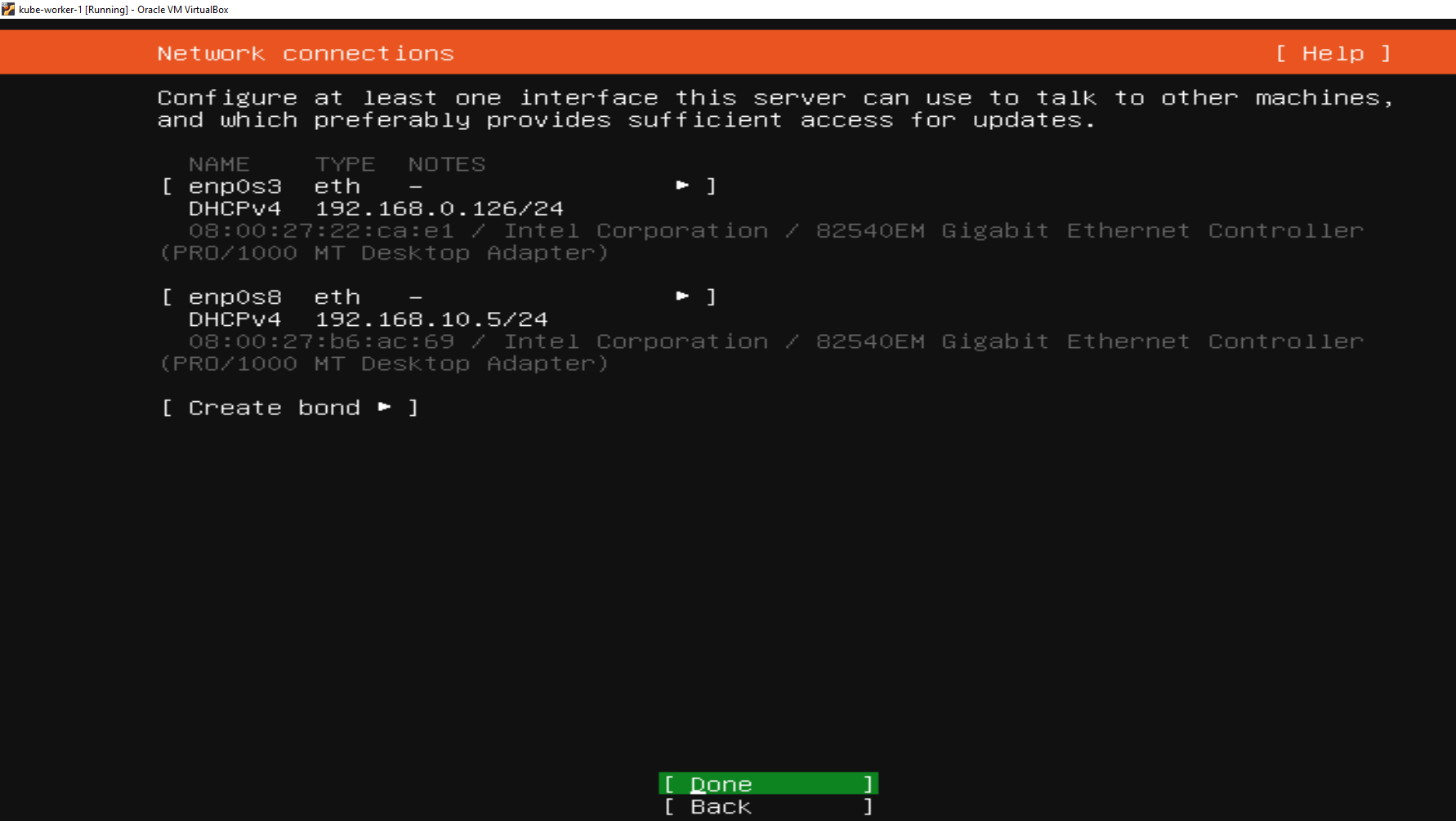
Select English and hit enter.



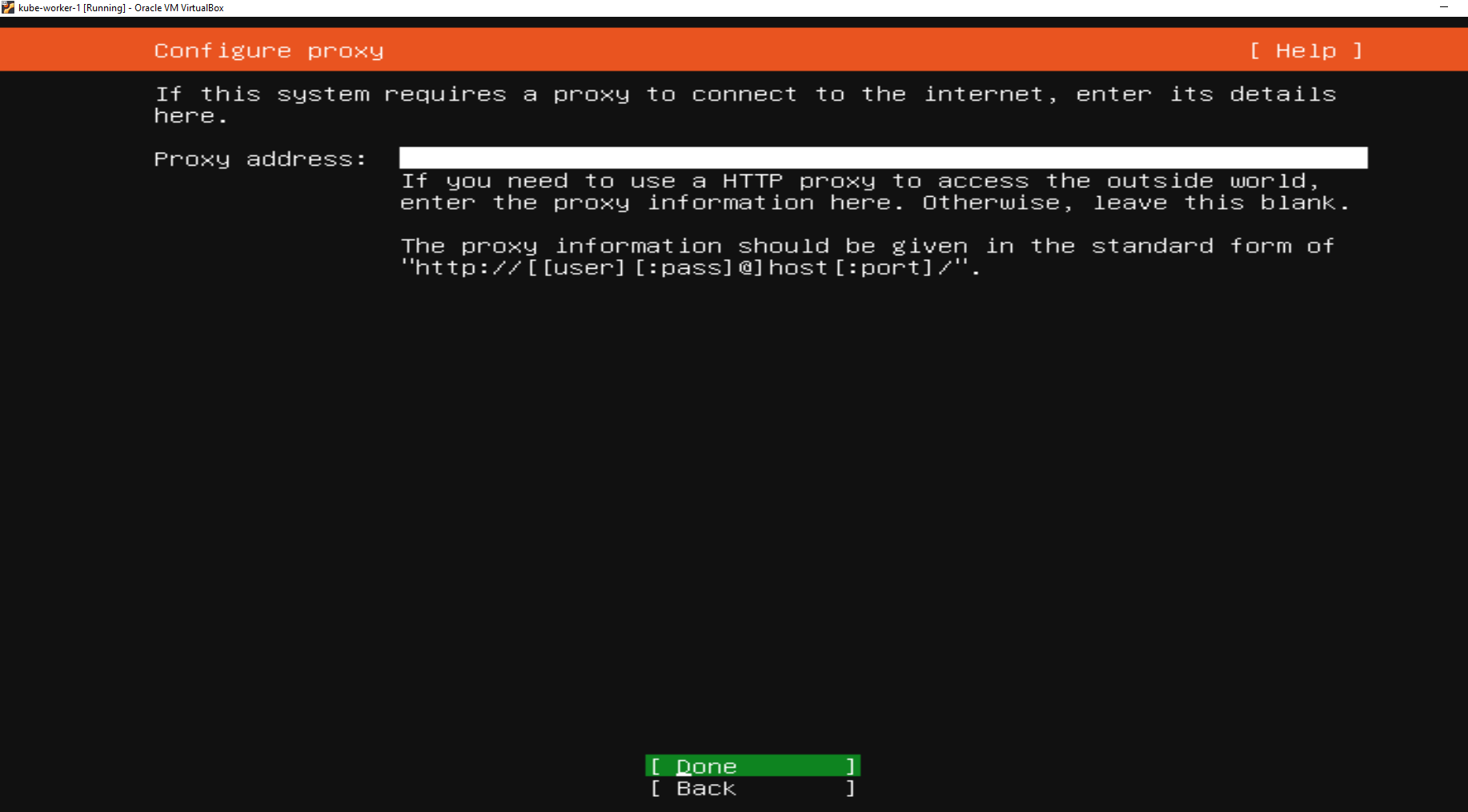
Navigate using arrow key to Done, Click **Enter**.



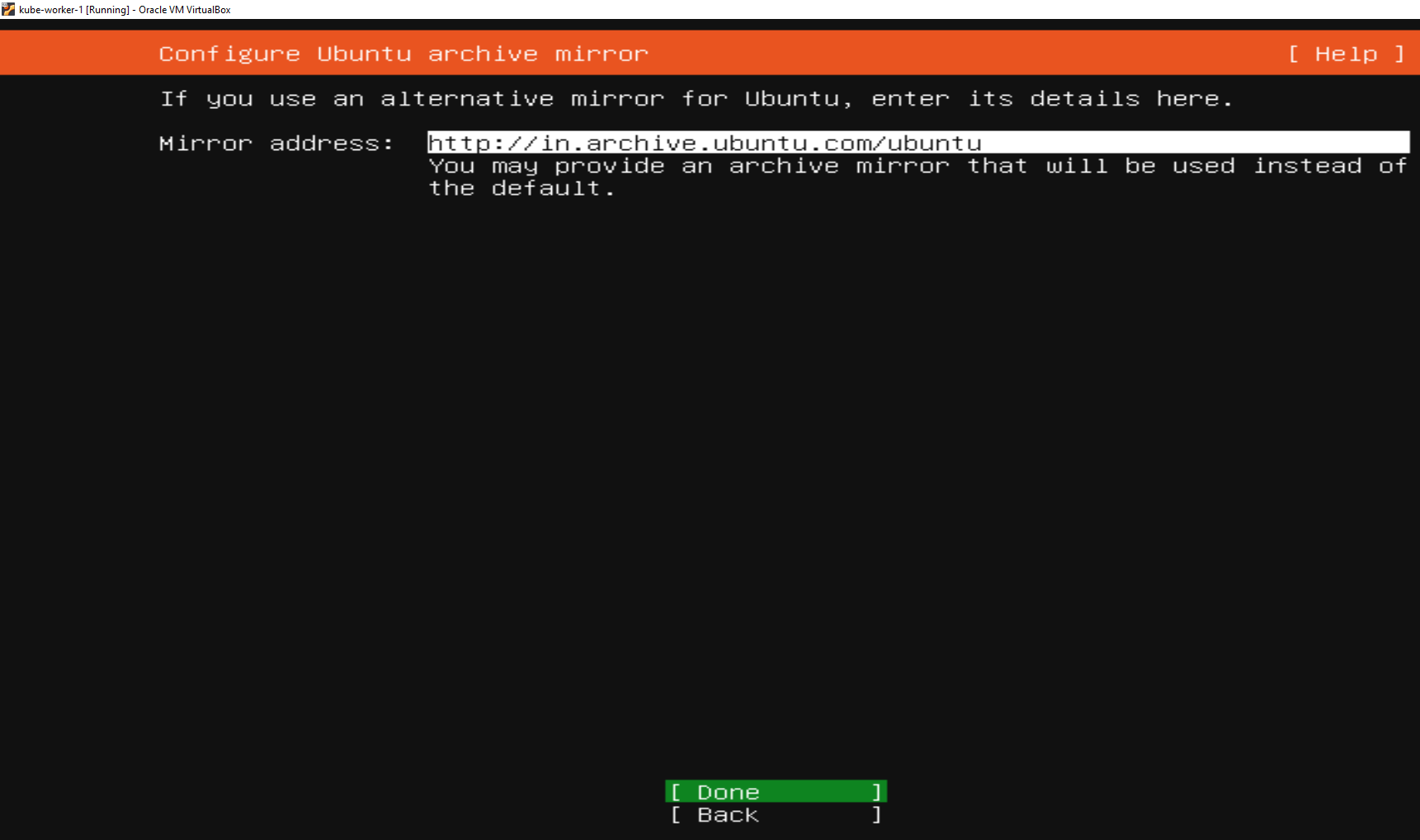
Make sure you get 2 IP address with the CIDR block defined as below, Navigate to Done and hit **Enter.**



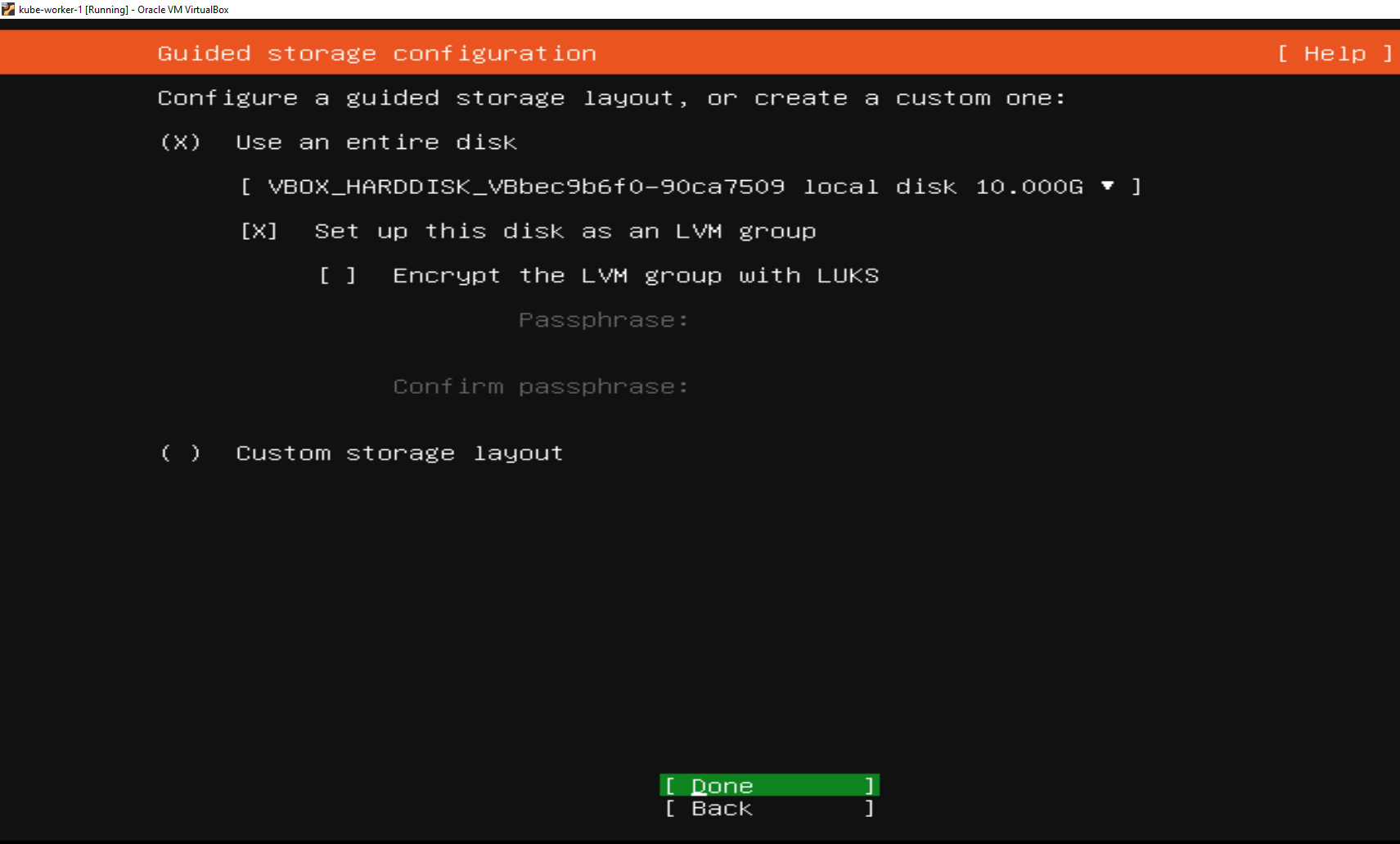
Proxy is not required, navigate to Done, Click **Enter.**



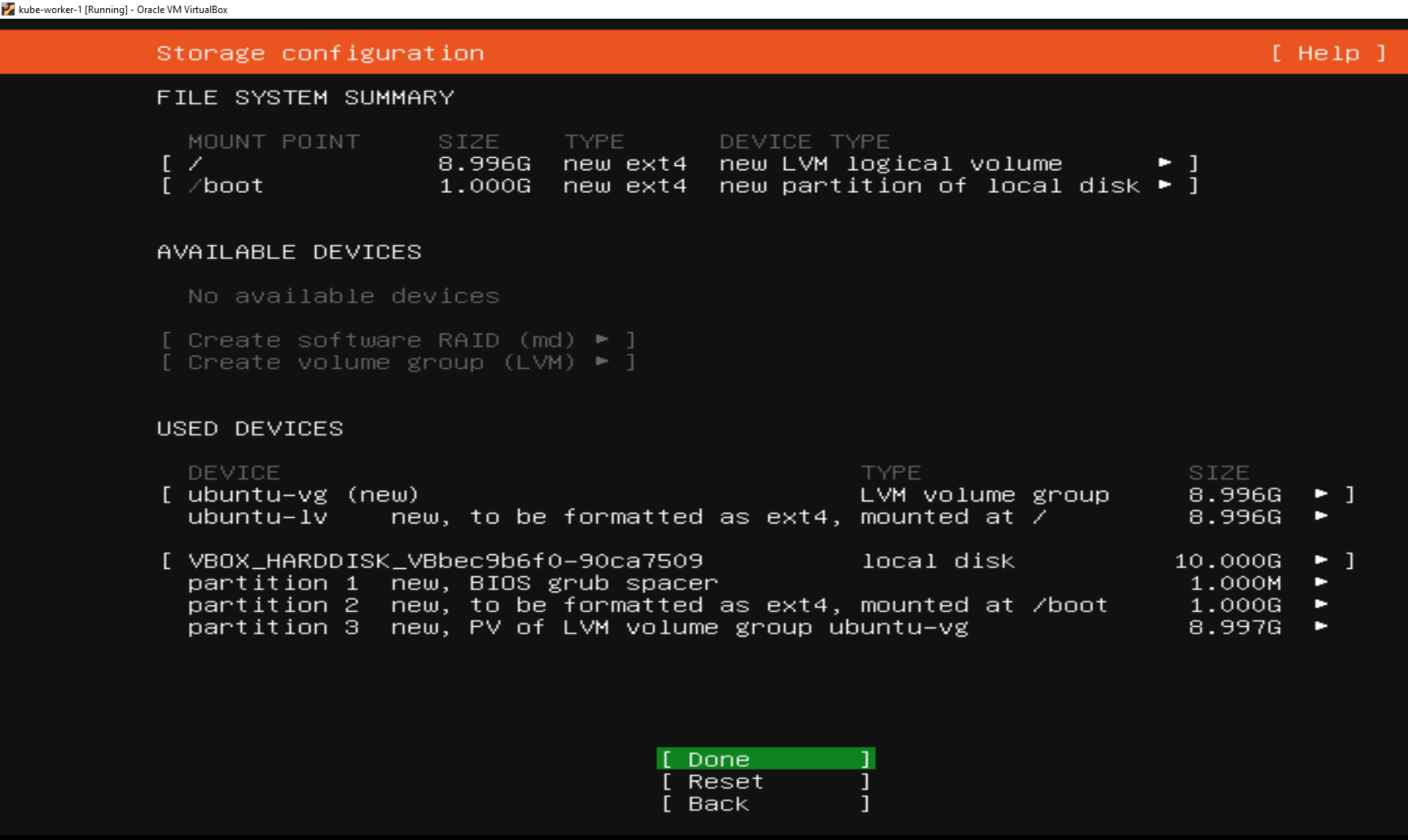
Navigate to Done, Click **Enter.**



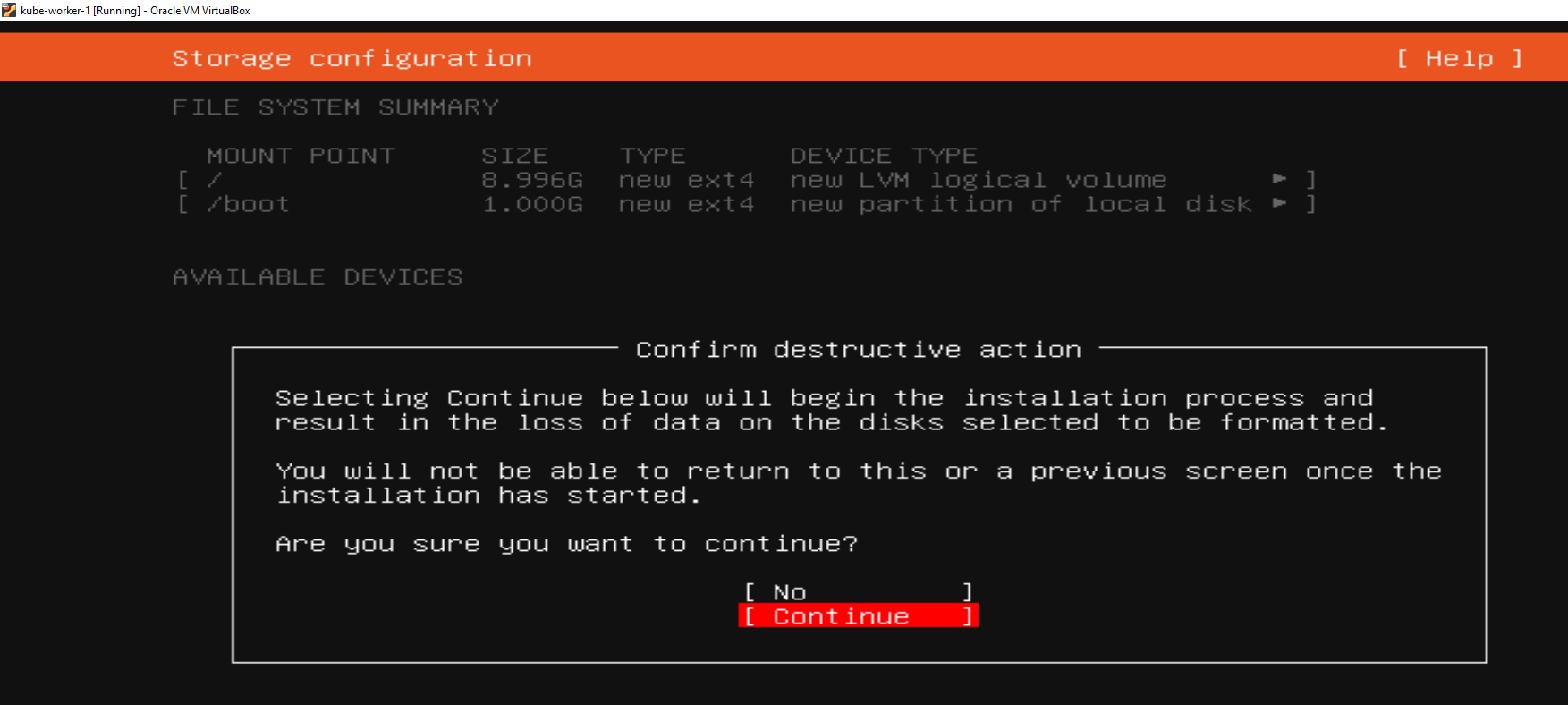
Navigate to Done, Click **Enter.**



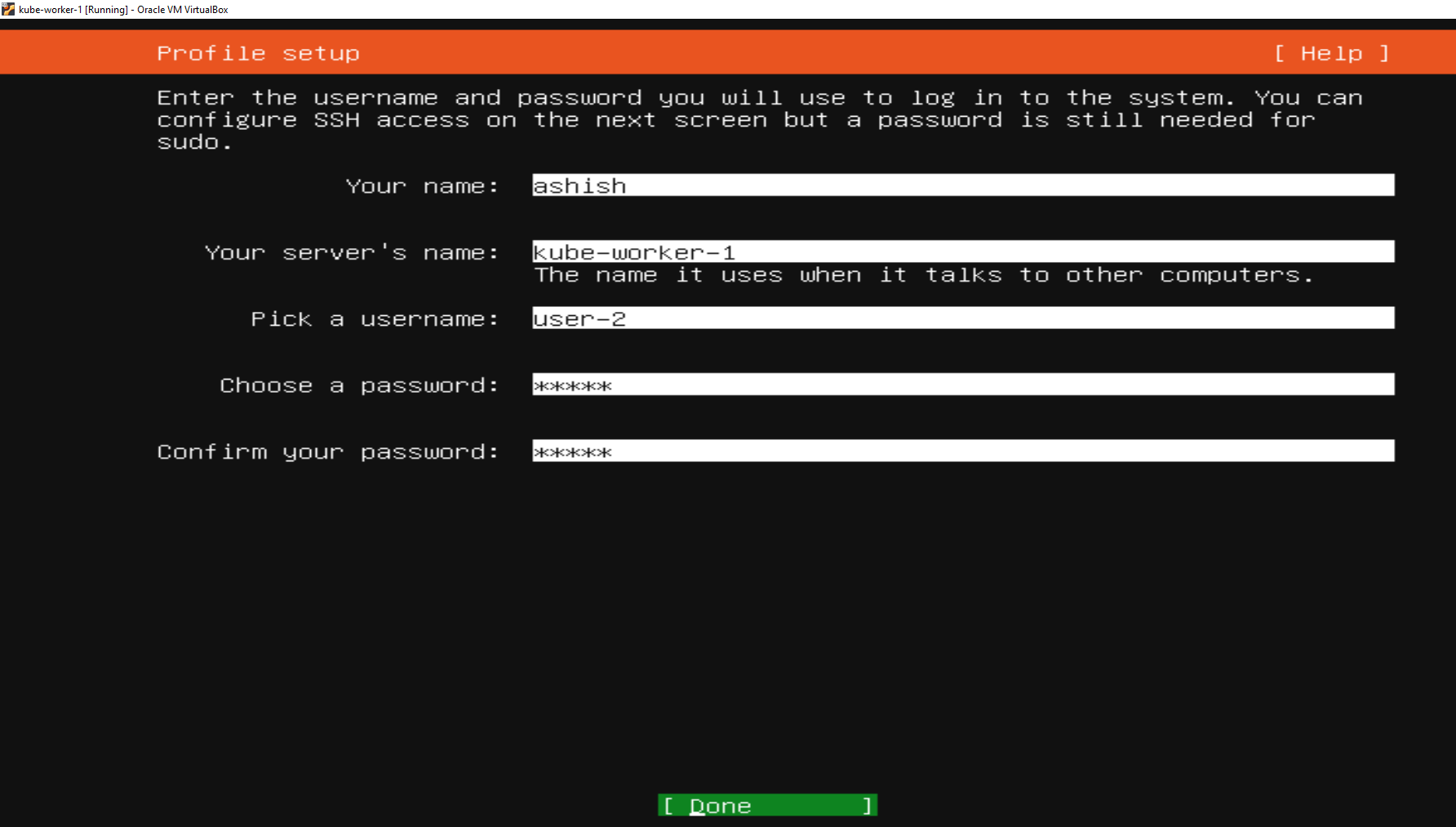
Navigate to Done, Click **Enter.**



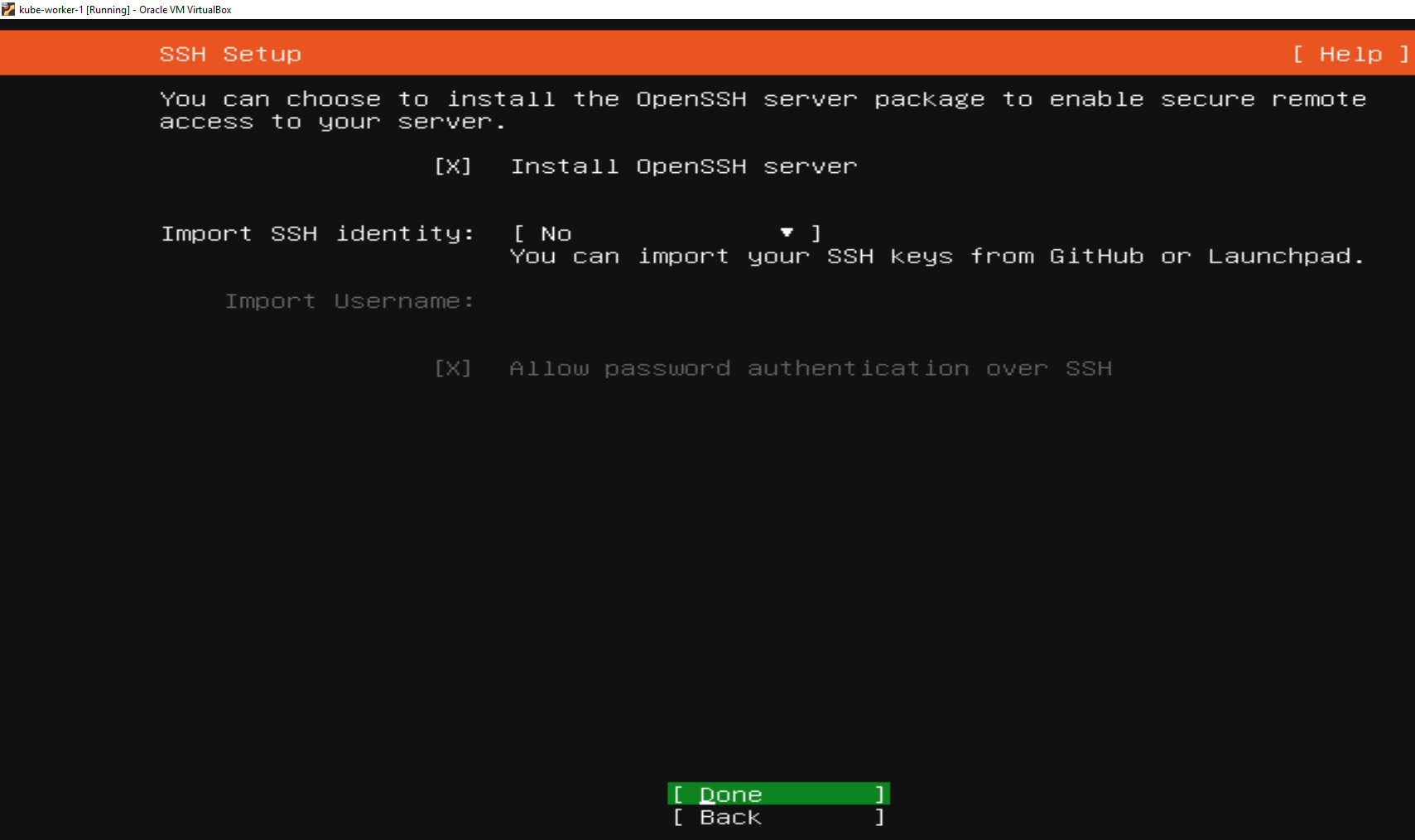
Navigate to Continue, Click **Enter.**



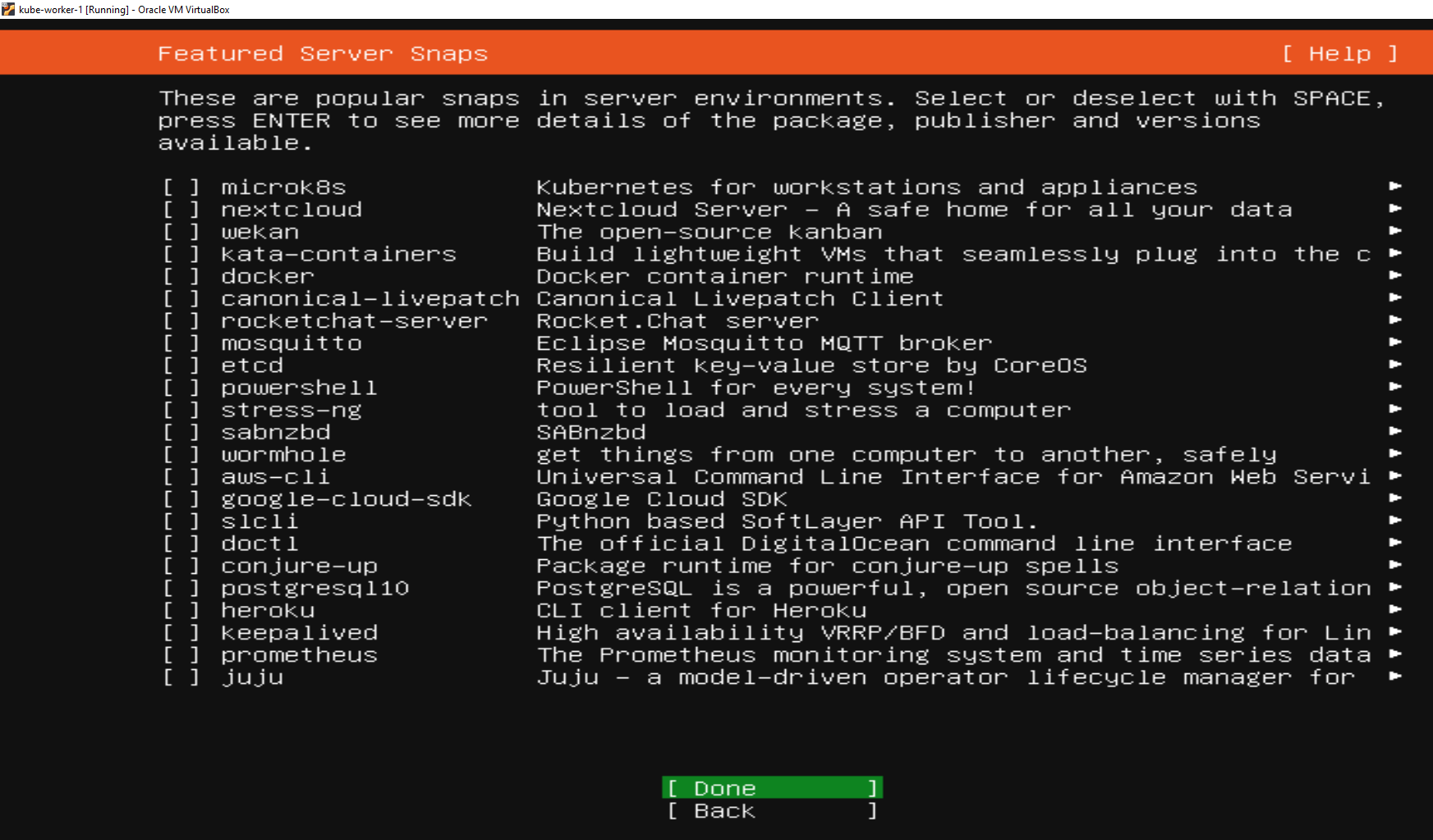
Provide all required details, navigate to Done , Click **Enter.**



Select install OpenSSH server, navigate to Done, Click **Enter.**



Skip all options for basic installation, navigate to Done, Click **Enter.**



Wait for installation to complete, then reboot your VM when prompted.

Repeat the same for all worker node you wanted to install.

## 2.3 Accessing Master and worker Node over ssh

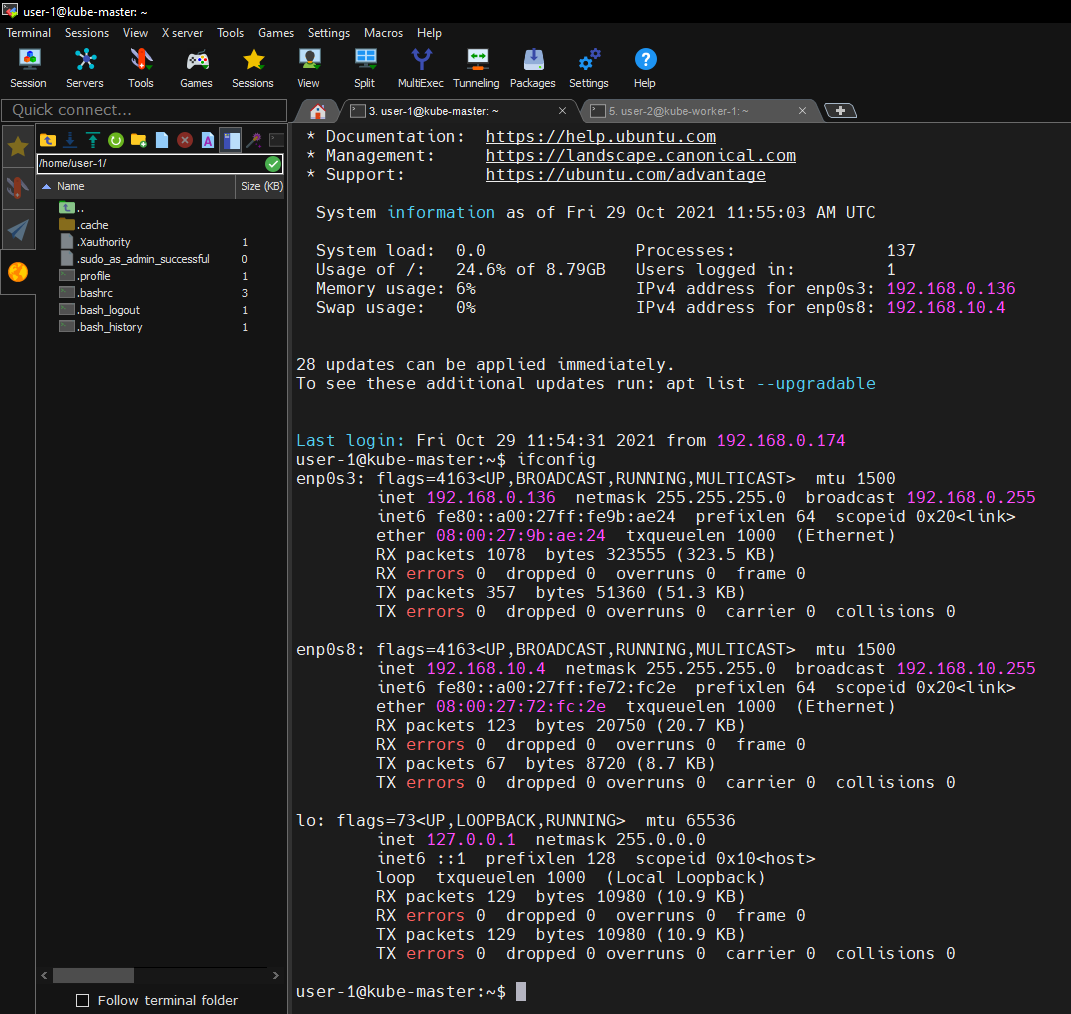
Use any ssh client to connect to master and worker Node.

* + - * Login to Master using credential
      * On successful login execute below commands on both Master and worker node, to enable ip config.

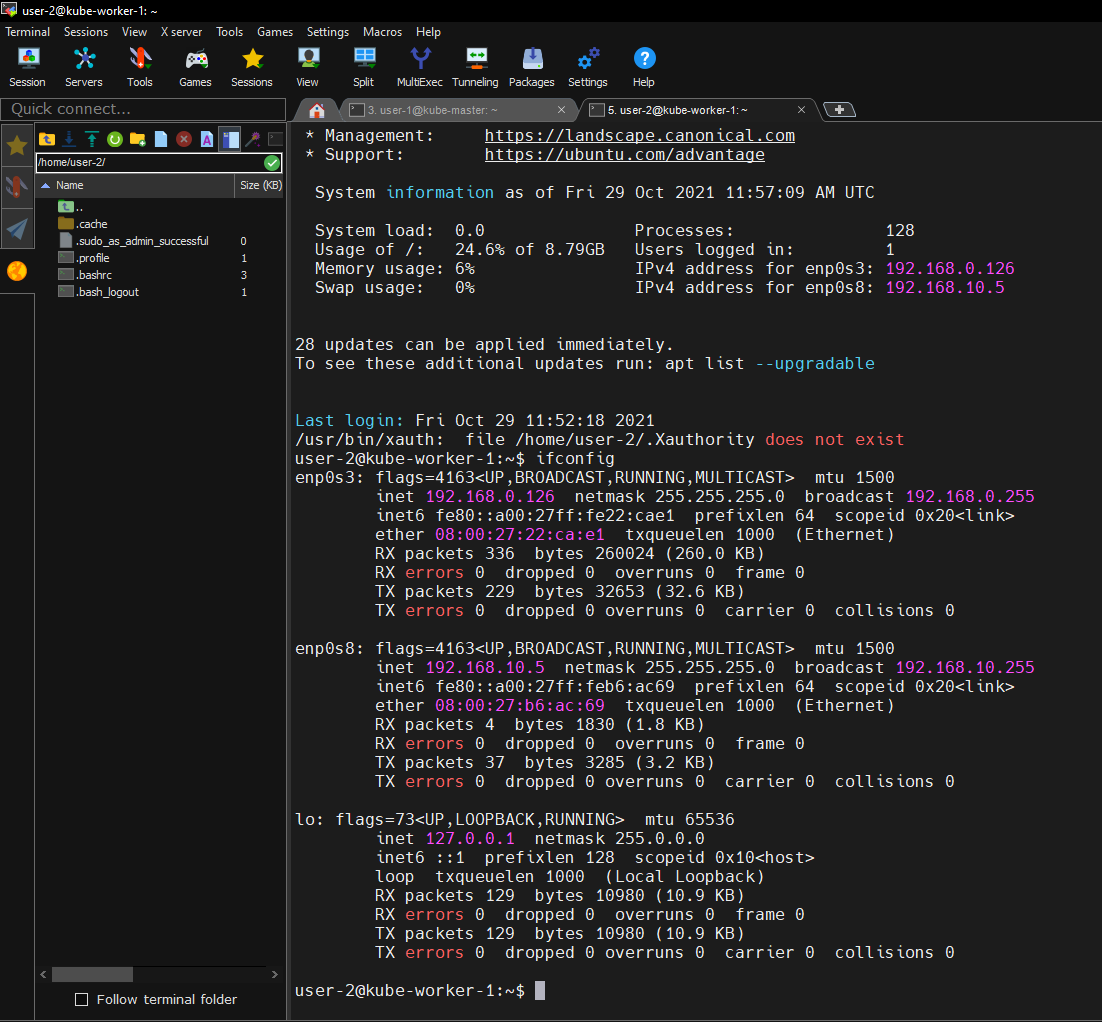
**Sudo apt install net-tools**

**Ifconfig :** this will give the public IPV4 address of respective nodes, use these IP to connect using any SSH client.

Master Node:



Worker Node:



## 2.4 Configure Master node for Kubernates cluster

1.Update the package repository.

sudo apt-get update

2. Install Docker

sudo apt-get install docker.io -y

3. Access Repos via HTTPS

sudo apt-get install apt-transport-https curl -y

4. Add K8S key and Repo

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list

deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF

5. Update the package repository and Install K8S components:

sudo apt-get update

sudo apt-get install -y kubelet=1.18.1-00

sudo apt-get install -y kubeadm=1.18.1-00

sudo apt-get install -y kubectl=1.18.1-00

sudo apt-mark hold kubelet kubeadm kubectl

6. Add the hosts entry

edit the file "/etc/hosts"

7. Disable SWAP

sudo swapoff -a

edit /etc/fstab to remove the swap entry

8. Initiate the Cluster

sudo kubeadm init --control-plane-endpoint k8s-master:6443 --pod-network-cidr 10.10.0.0/16

sudo kubeadm init --pod-network-cidr 10.10.0.0/16

9. Set the kubectl context auth to connect to the cluster

mkdir -p $HOME/.kube

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

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

10. Pod Network Addon(Calico)

Ref: https://docs.projectcalico.org/gettin...

curl https://docs.projectcalico.org/manifests/calico.yaml -O

vi calico.yaml

11. Generate Token to add worker Node

#Create a new Token

sudo kubeadm token create

#List Tokens created

sudo kubeadm token list

#Find Certificate Hash on Master

openssl x509 -pubkey -in /etc/kubernetes/pki/ca.crt |

openssl rsa -pubin -outform der 2>/dev/null |

openssl dgst -sha256 -hex | sed 's/^.\* //'

12. Want to run workloads on Master?

NB:---Please check in the node which taint present by

$Kubectl get node

We will get node name

Then run:

$kubectl describe node NODE-NAME

$kubectl taint nodes --all node-role.kubernetes.io/master-

13. Sample Deployment and service file: create deployment.yaml and place this code

============================================================

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx

labels:

app: nginx

spec:

replicas: 1

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx

ports:

- containerPort: 80

Create a service file i.e nginx-svc.yml and place the code

apiVersion: v1

kind: Service

metadata:

name: nginx

spec:

type: NodePort

selector:

app: nginx

ports:

# By default and for convenience, the `targetPort` is set to the same value as the `port` field.

- port: 80

targetPort: 80

# Optional field

# By default and for convenience, the Kubernetes control plane will allocate a port from a

range (default: 30000-32767)

nodePort: 30007

=================================================

14. Apply the deployment and service:

kubectl apply -f FILE\_NAME

## 2.5 Configure Worker node for Kubernates cluster

1.Update the package repository

sudo apt-get update

2. Install Docker

sudo apt-get install docker.io -y

3. Access Repos via HTTPS

sudo apt-get install apt-transport-https curl -y

4. Add K8S key and Repo

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list

deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF

5. Update the package repository and Install K8S components:

sudo apt-get update

sudo apt-get install -y kubelet=1.18.1-00

sudo apt-get install -y kubeadm=1.18.1-00

sudo apt-get install -y kubectl=1.18.1-00

sudo apt-mark hold kubelet kubeadm kubectl

6. Add the hosts entry

edit the file "/etc/hosts"

7. Disable SWAP

sudo swapoff -a

edit /etc/fstab to remove the swap entry

8. Join Nodes

sudo kubeadm join --token TOKEN\_ID CONTROL\_PLANE\_HOSTNAME:CONTROL\_PLANE\_PORT --discovery-token-ca-cert-hash sha256:HASH

(Formed using outputs from step 10, treat CAPS as variables to be replaced