

**Application Containerization**

**Assignment-1**

**NAME- Rithik Choudhary**

**SAP ID- 500069261**

**ROLL NO- R171218084**

**COURSE- B.TECH[CSE-DEVOPS]**

**SUBJEC T- APPLICATION CONTAINERIZATION LAB**

SUBMITTED TO:-

MR. HITESH KUMAR SHARMA

Q

Implement the following Scenario using Vagrant and VirtualBox .

1.Download an Ubuntu box from Vagrant Cloud

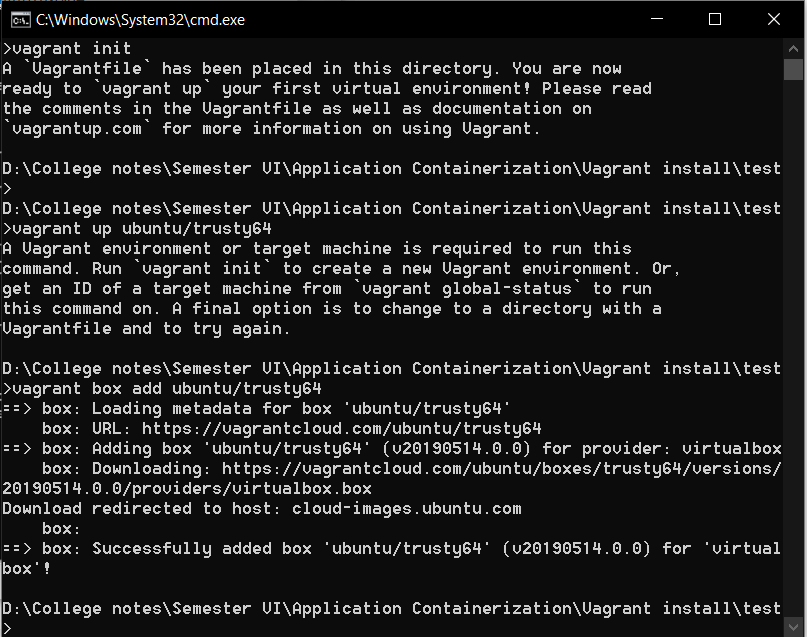
We have installed vagrant in our system from official website of vagrant.

Now we are try to initiaize the vagrant in specific folder using the command.

**/> Vagrant init**

It is initialized successfully , now we download the box of ubuntu known as “ubuntu/trusty64 ”.

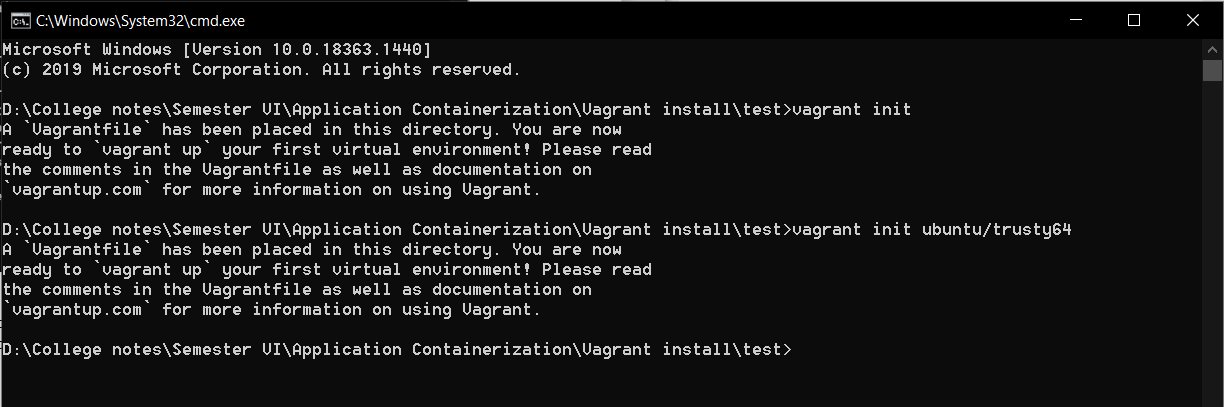
**/> Vagrant box add ubuntu/trusty64**



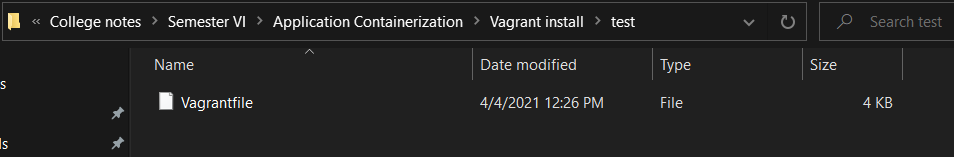
Now we initialize the ubuntu a specified directory.

First go the location of your directory where we have to initialize the ubuntu, then use the command

**/> Vagrant init ubuntu/trusty64**



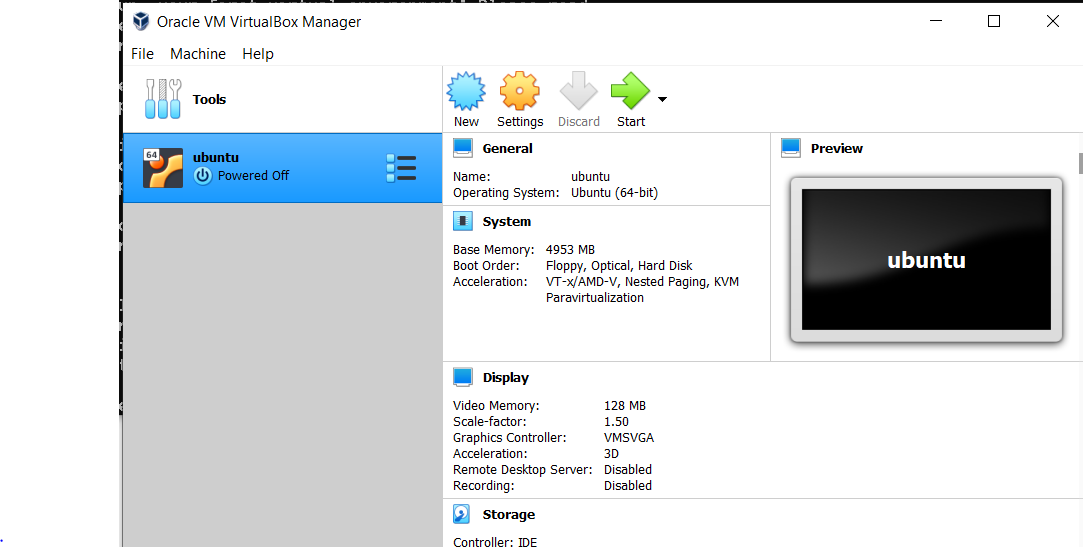
A vagrant file has been created in the folder



Now we are ready to run the virtual machine.

But first we have to install the virtual box, or hyper-V to run the virtual machine because vagrant use the 3rd party software to run the virtual machine.

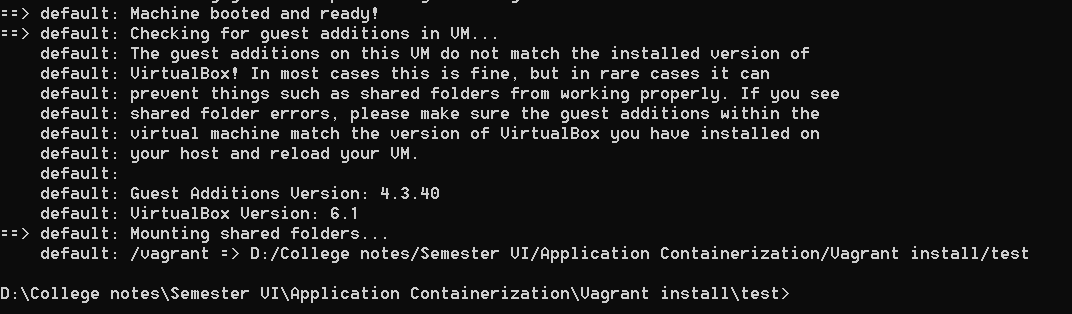
So we have to install the virtual box in out system.

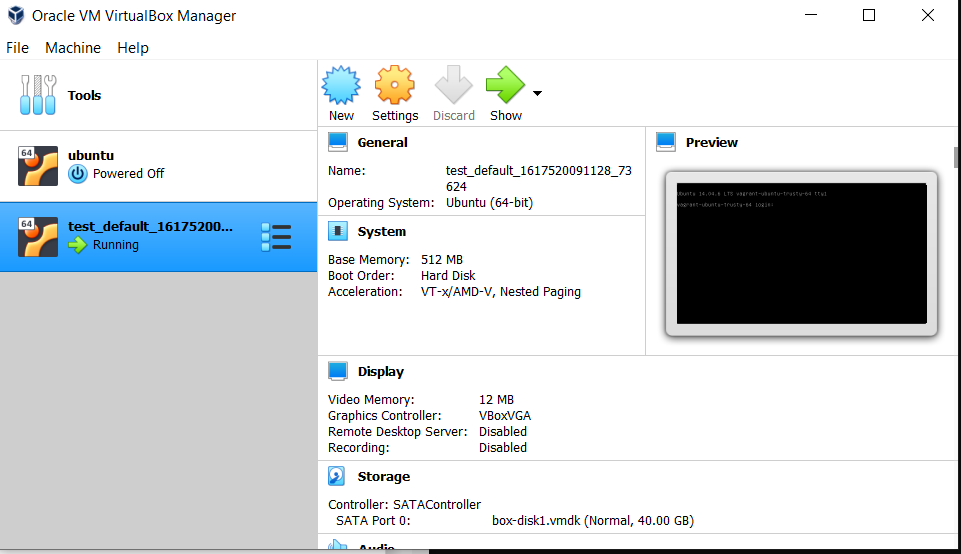


Now we are ready to run the machine.

**/> Vagrant up**

We have successfully installed the virtual machine in virtual box.





Vm, Virtual box, vagrant and ubuntu has been install successfully in the system.

Now we have to use it efficiently.

**2.UP Ubuntu Machine using Vagrant**

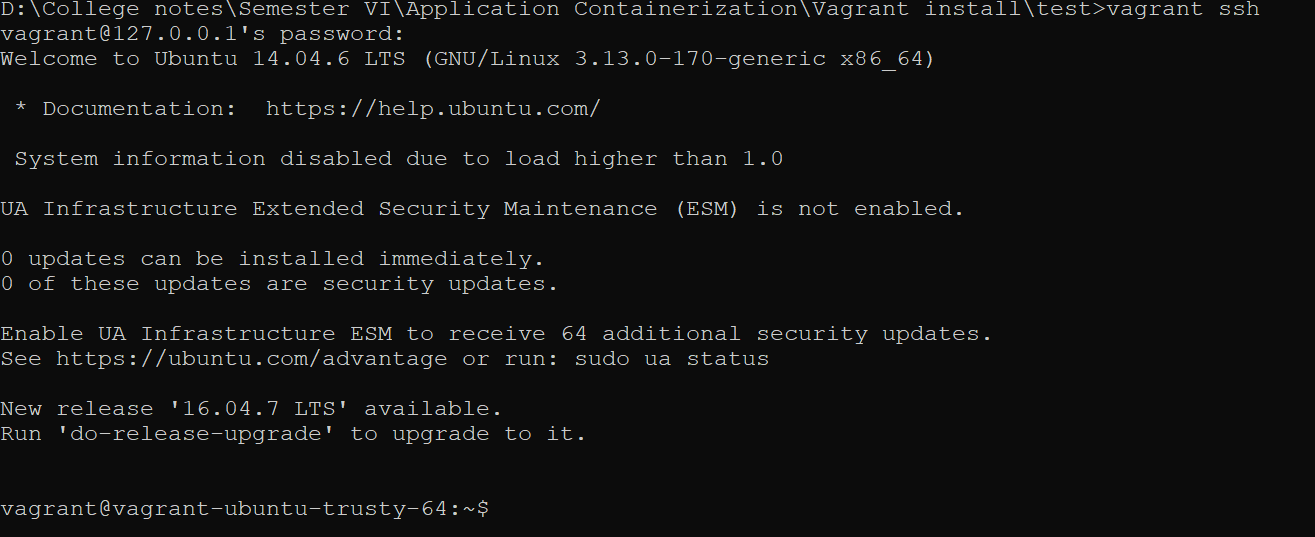
First of all we have to open the ubuntu console using ssh.

**/>vagrant ssh**

It may require a password, it suggest that the password is with public key, sometimes it doesn’t work

So there is default password for the vagrant.

Password: **vagrant**



Now lets install docker in the running machine. First update the machine repository

**/> Sudo apt update**

Now install the docker, first specify the repository in the machine from where it have to download the package of docker.

/> echo \

"deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \

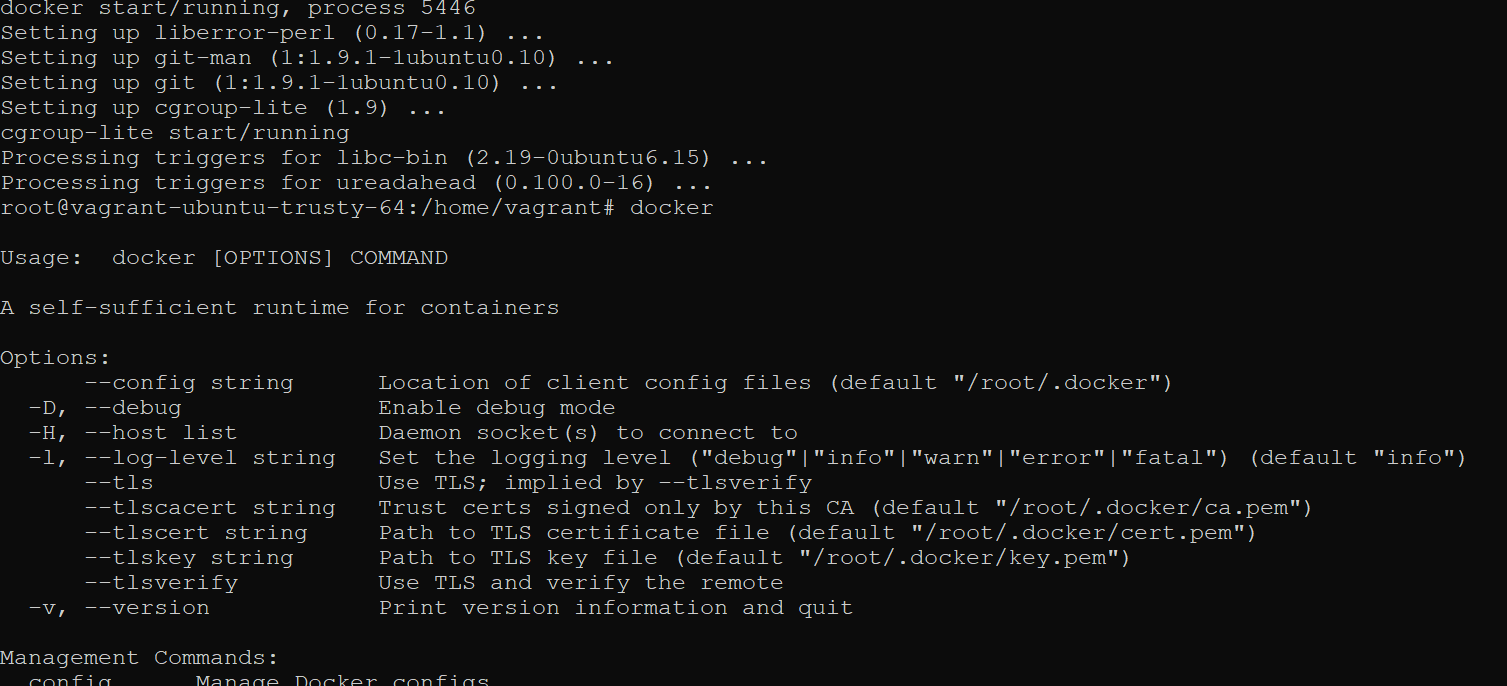
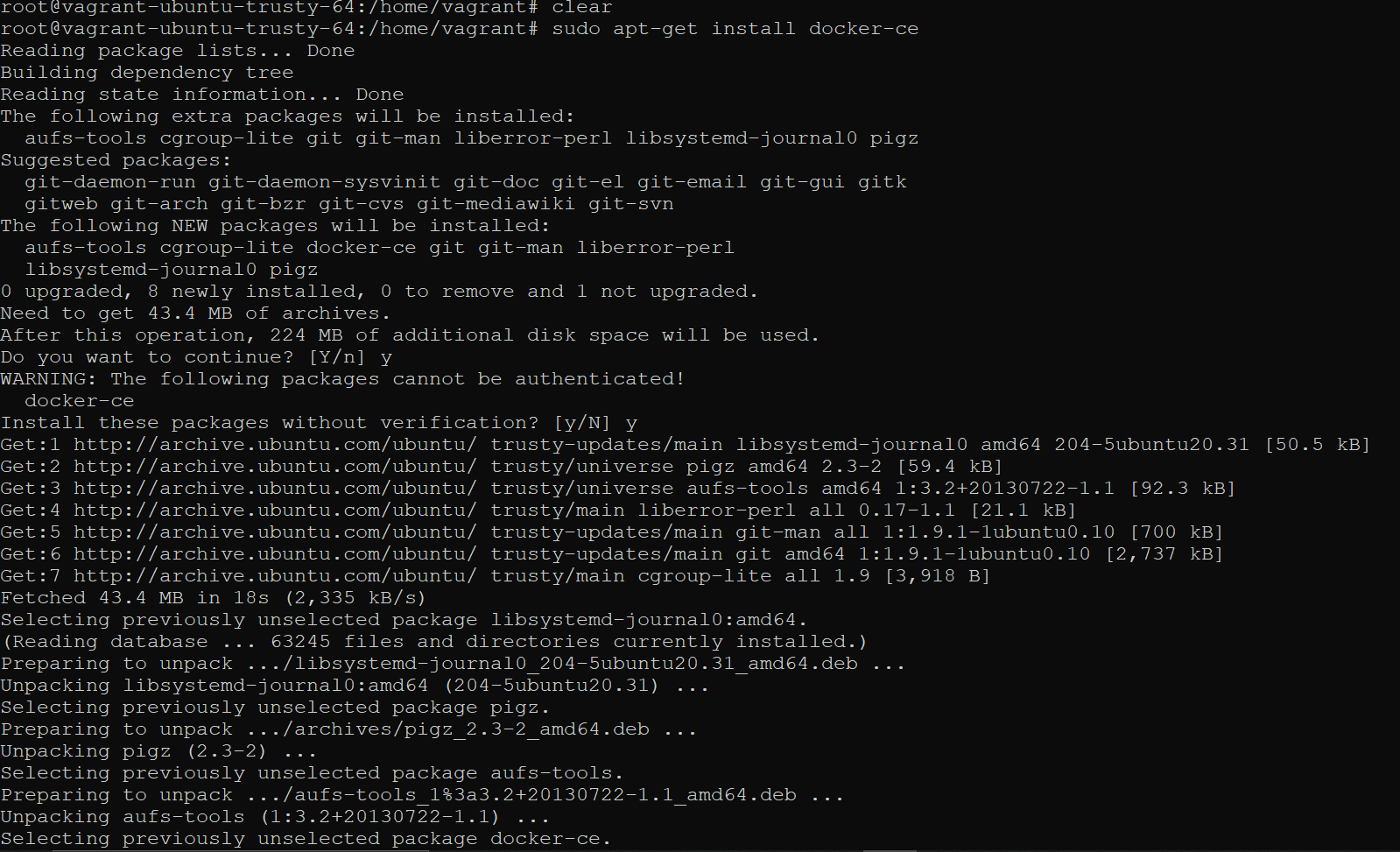
$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

**Add Docker’s official GPG key:**

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

Now install the docker using the command.

**/> sudo apt-get install docker-ce**

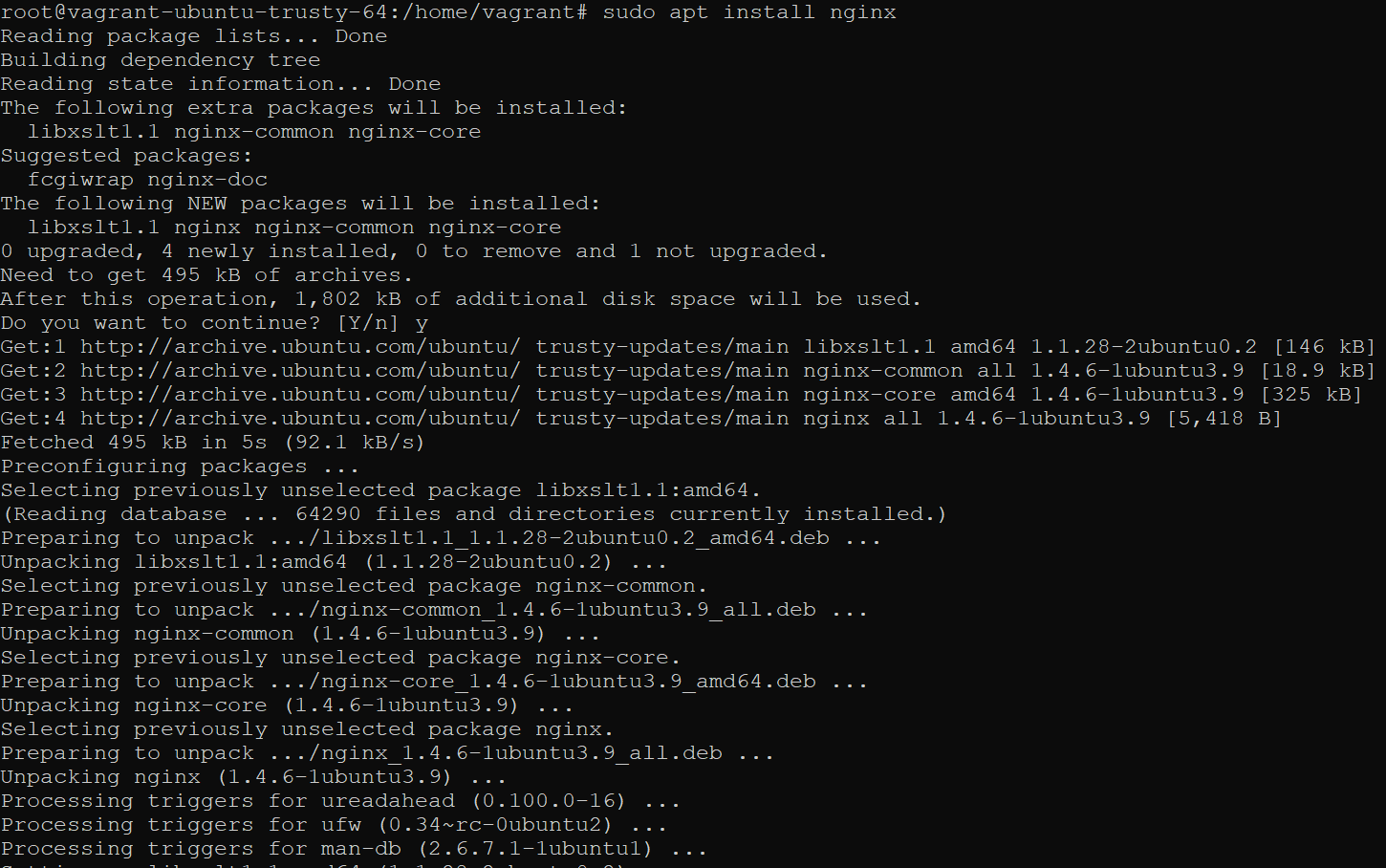


Docker installed successfully.

1. Install nginx in Ubuntu Box Image

Installing nginx in the vm.

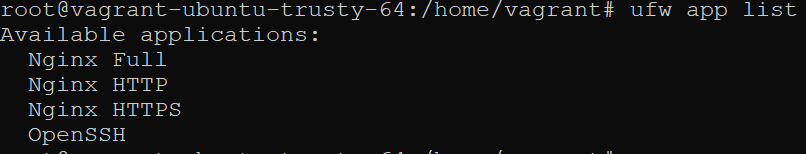
/> sudo apt install nginx



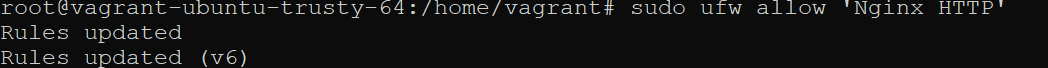
We have to configure the firewall to use the nginx on the required ports

We use ufw to manage the firewall ports. Now list the app that can access the ports

/> sudo ufw app list



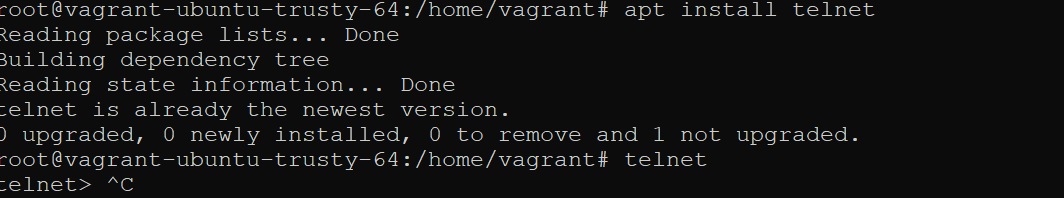
Allow then to use the ports



1. Install Telnet package in same image.

To install the telnet we use the command.

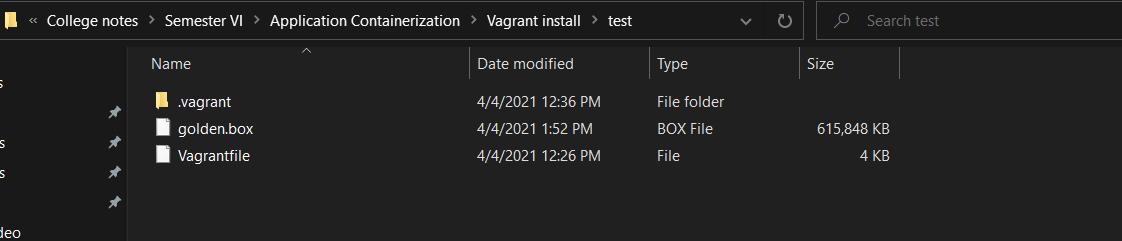
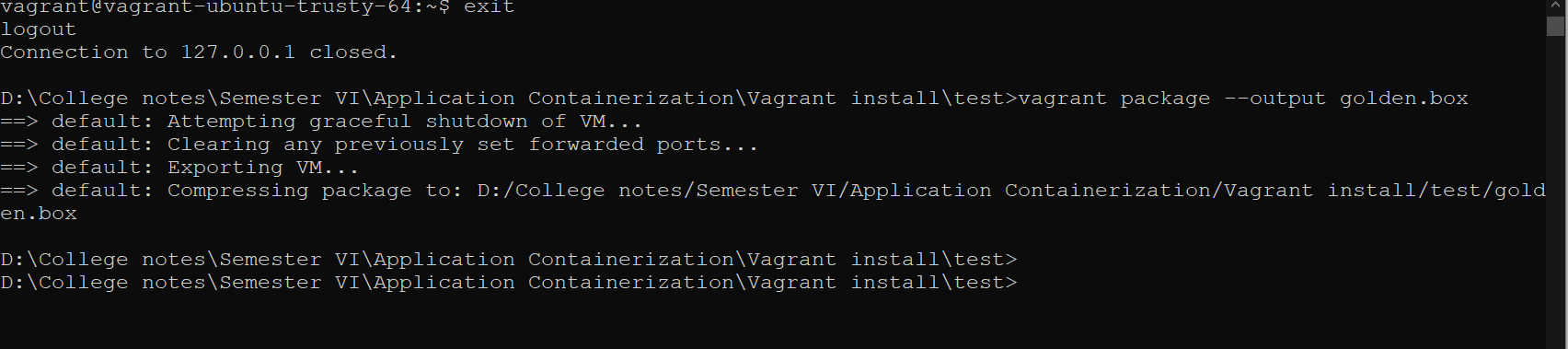
**/> apt install telnet**



1. Package this image as golden image using vagrant

To create a golden box package we have to use the command.

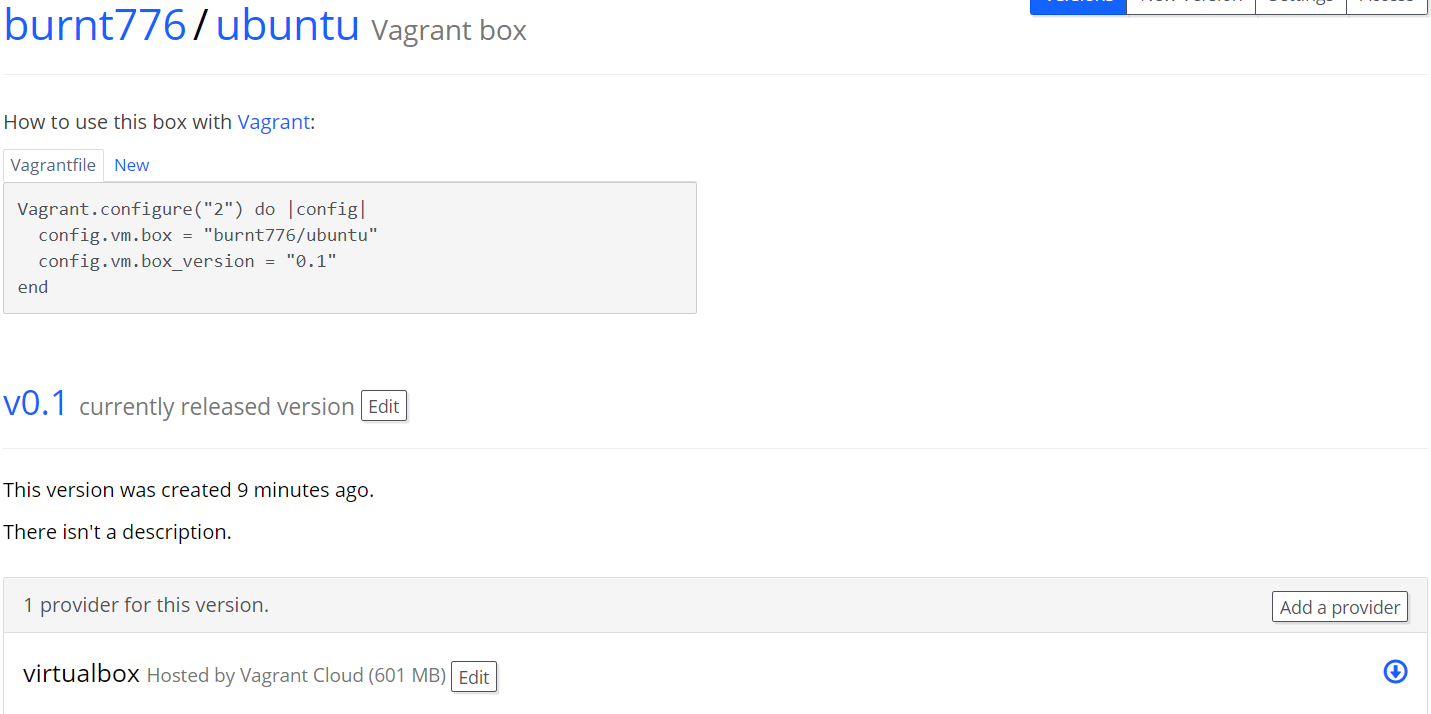
**/> vagrant package --output golden.box**



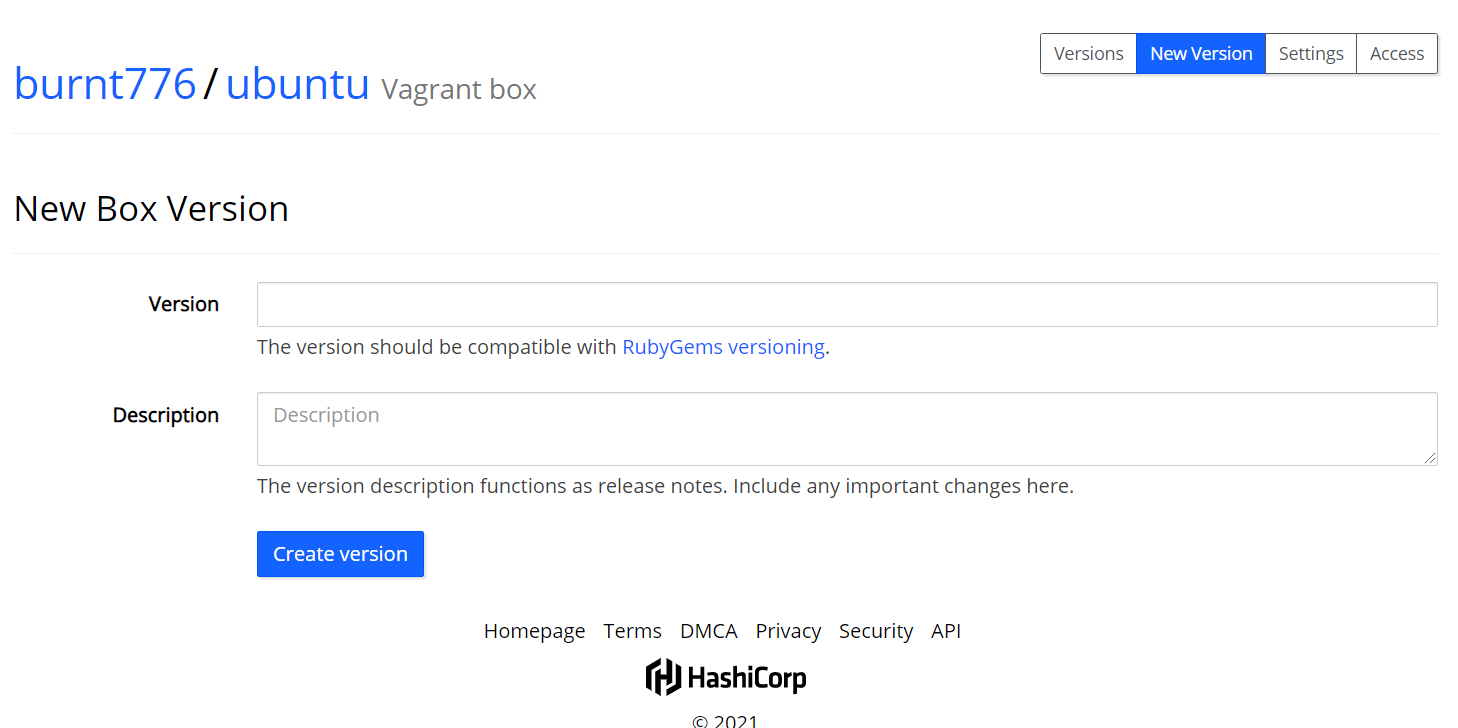
Golden.box is created successfully.

1. Publish the same image on Vagrant Cloud.

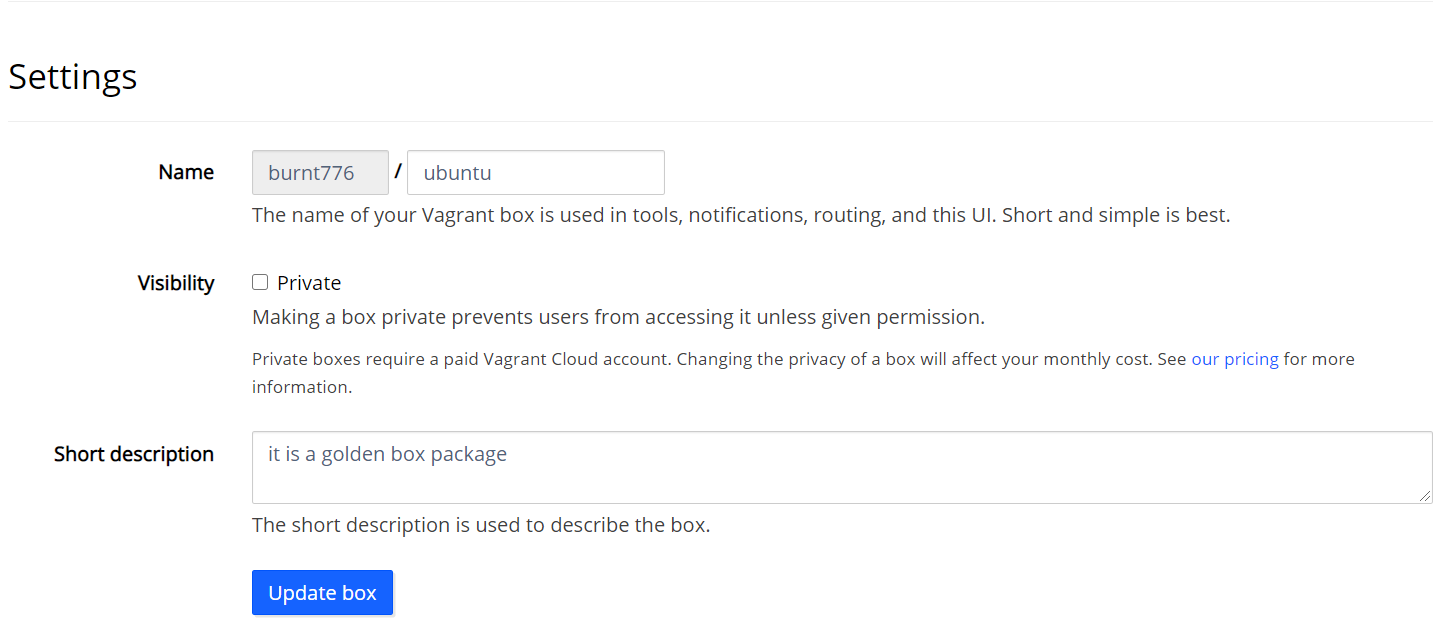
Now we have to release the golden.box we have created on our system to vagrant cloud.



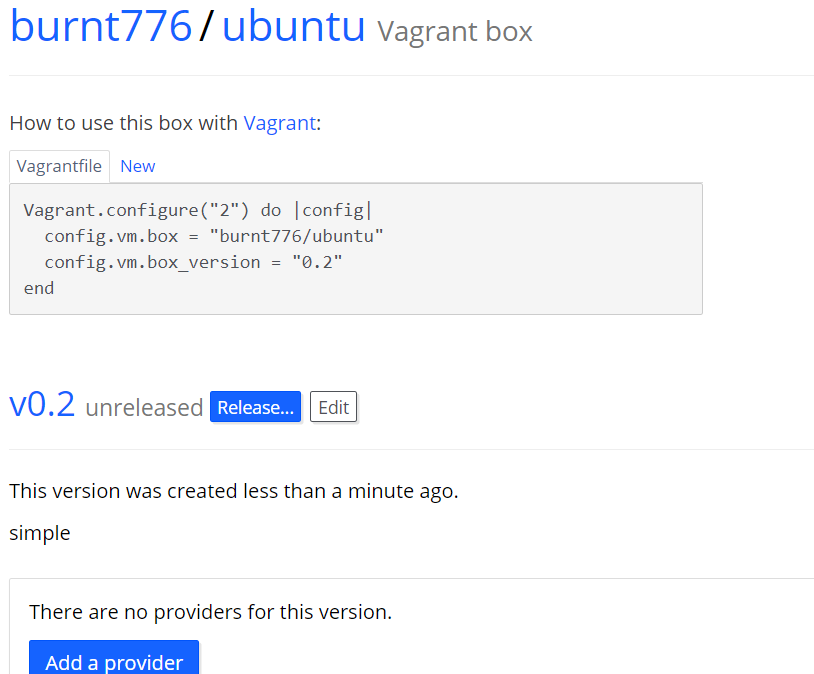
Now describe the version of the box.



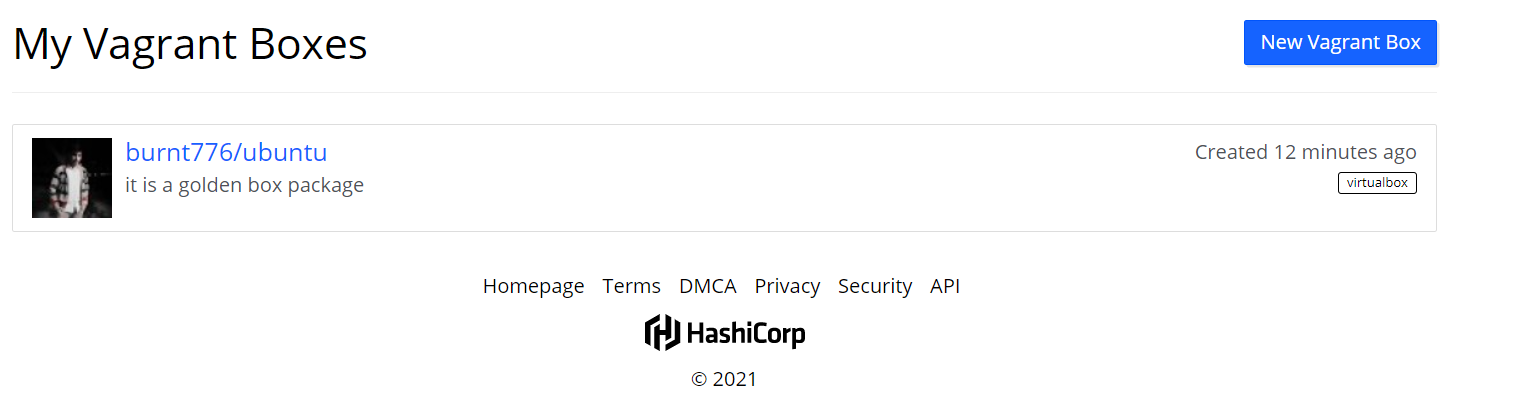
Set the visiblity and mention the box name



And then release it on the vagrant cloud.



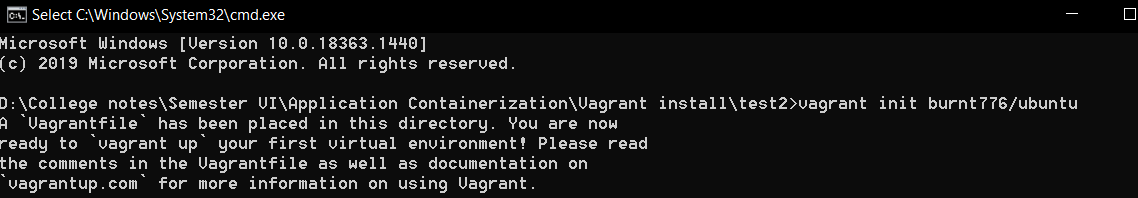
Here is the released image.



1. Again download the pushed image.

Downloading the Pushed image.

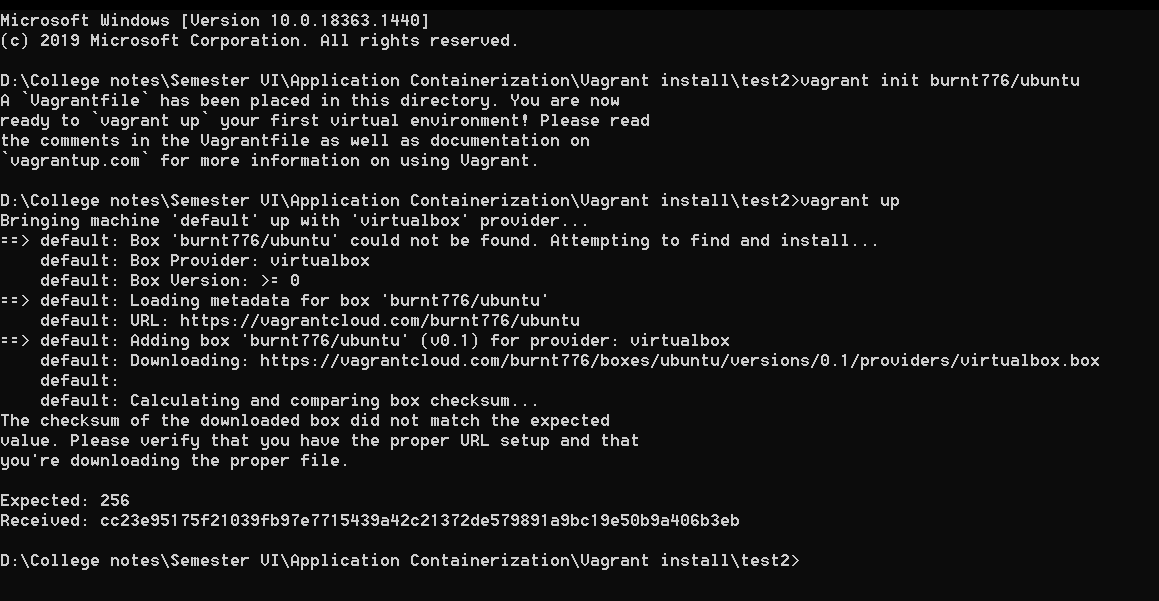
/> Vagrant init burnt776/ubuntu

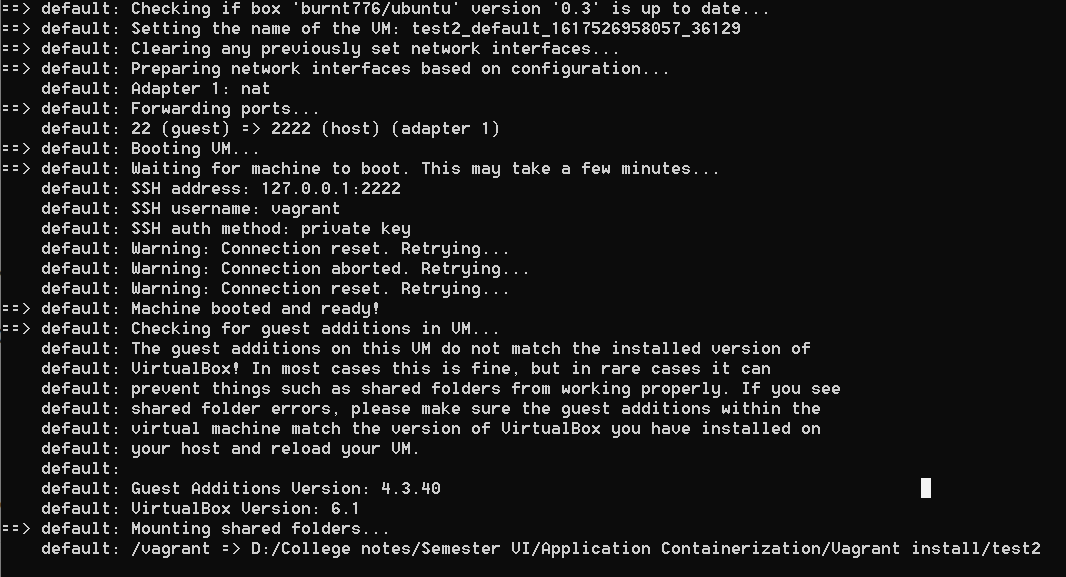


8.UP the downloaded golden image using vagrant

Now install the fetched machine and run in the command prompt.

**/> Vagrant up**





1. Host an HTML page in nginx server