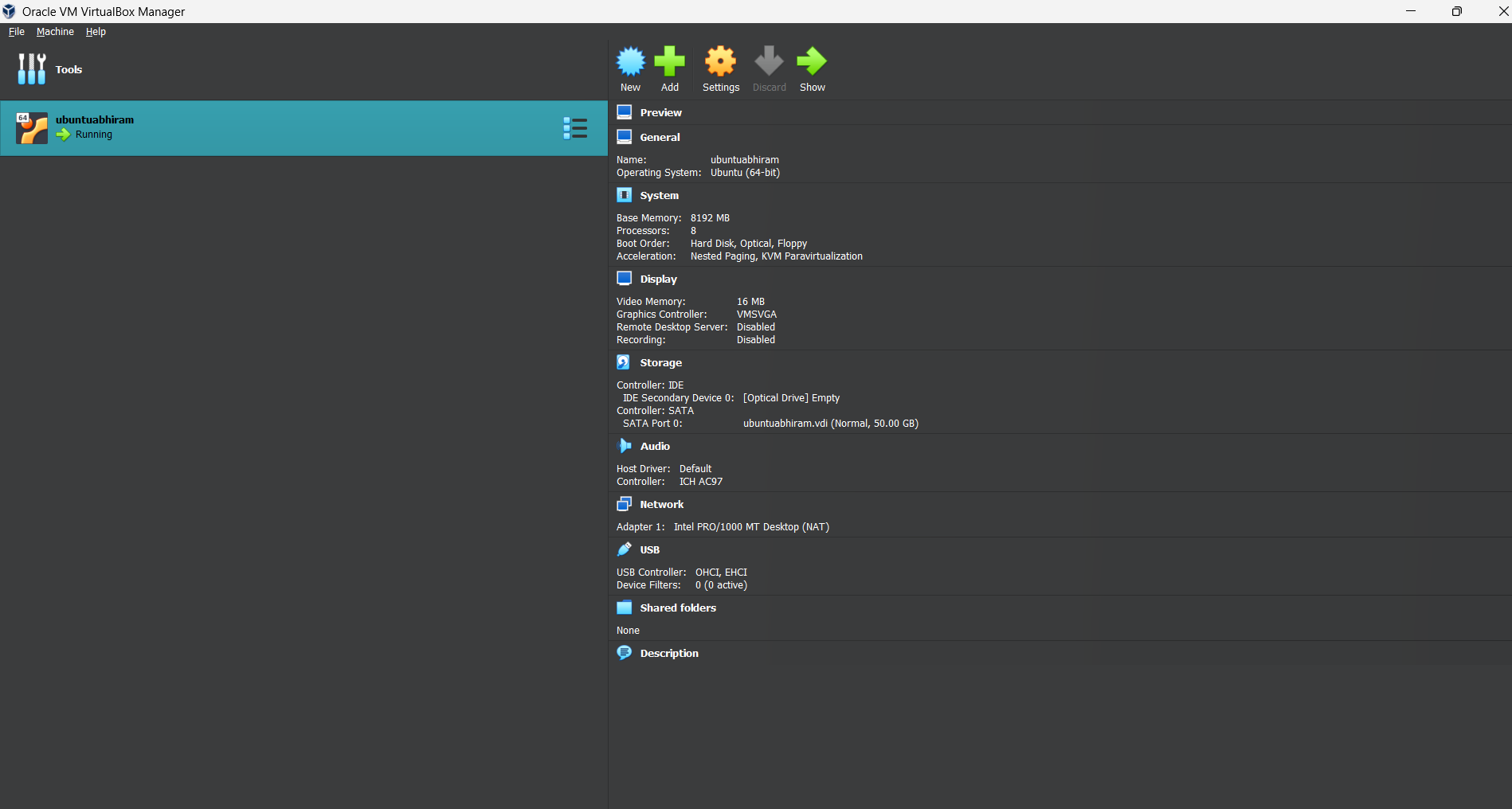
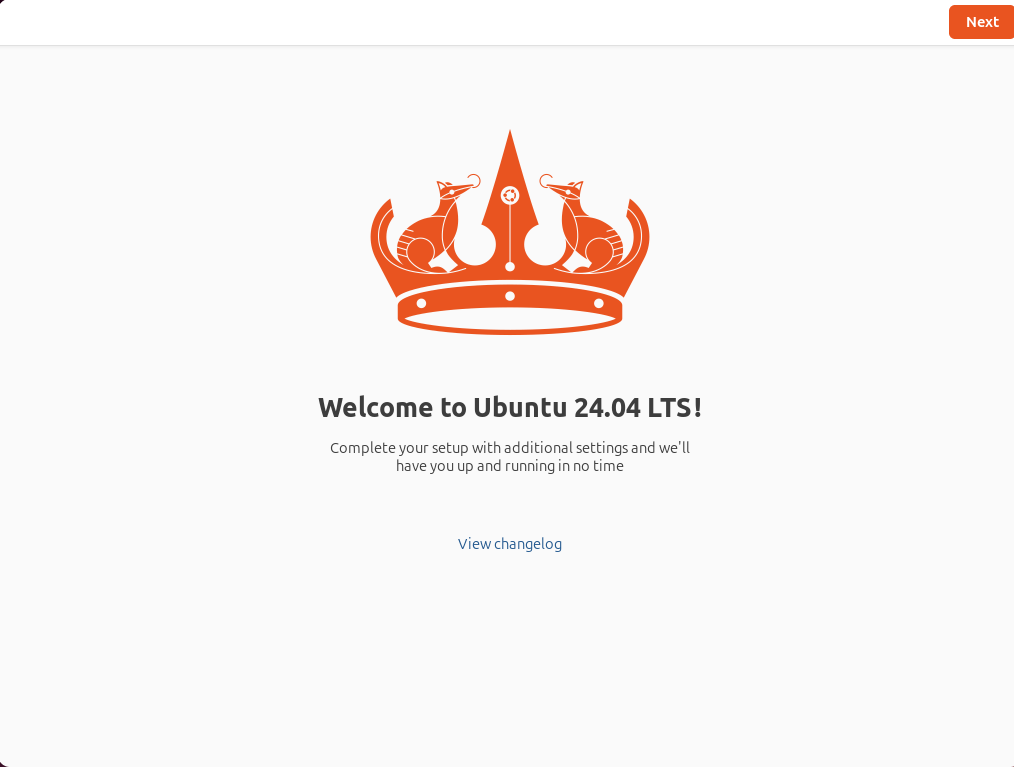
**Week 10-12 Graded Assignment: solution submission**

**Step#1 : Host a Ubuntu Virtual Machine using Oracle VM Virtual Box.**

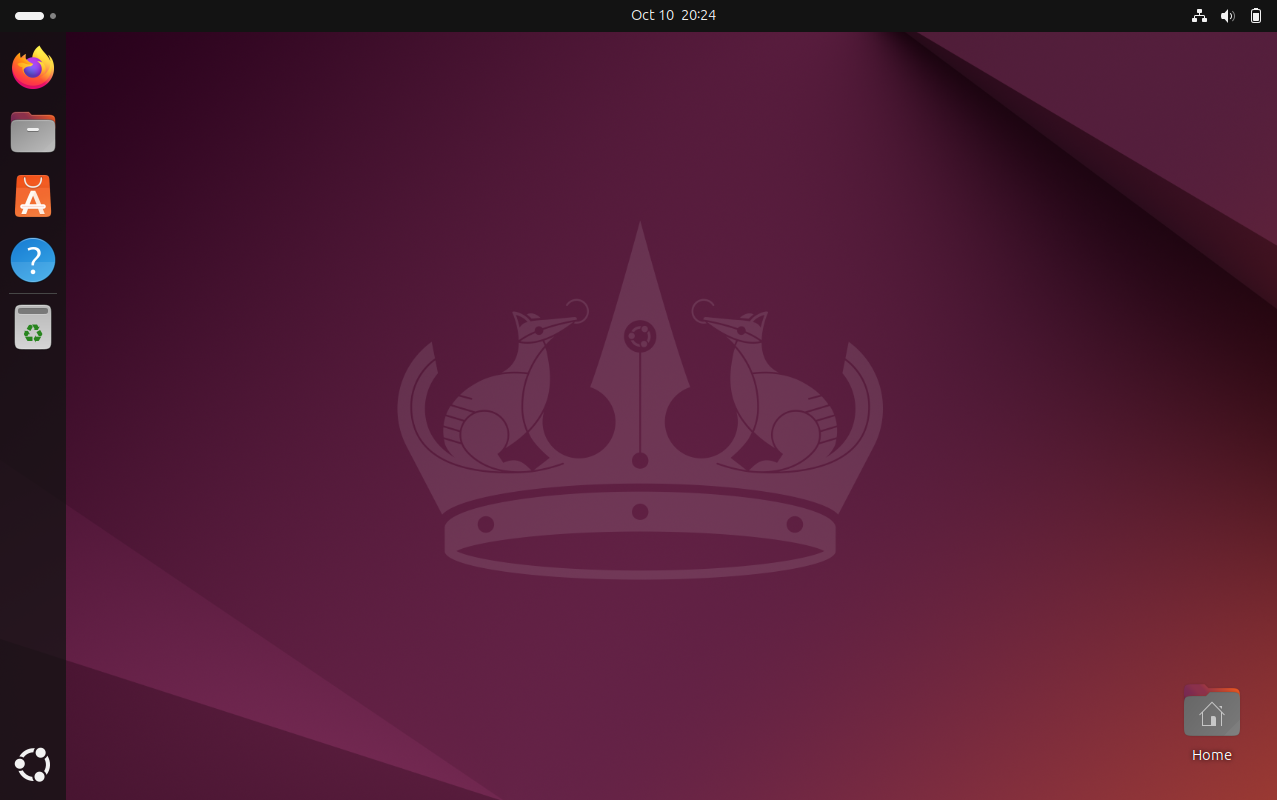
Launch Oracle VM VirtualBox Manager (prior to that make sure to download and install)



**Install Ubuntu**



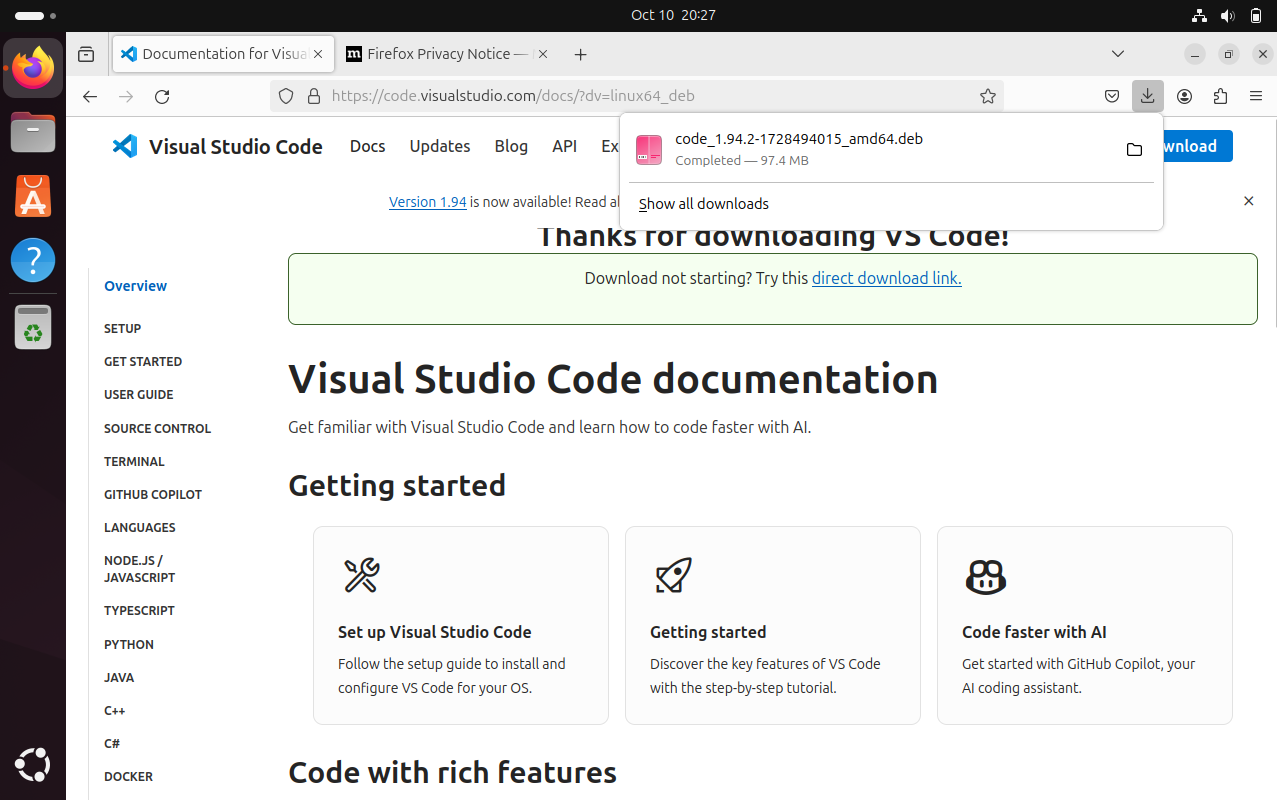
**Ubuntu Home page**

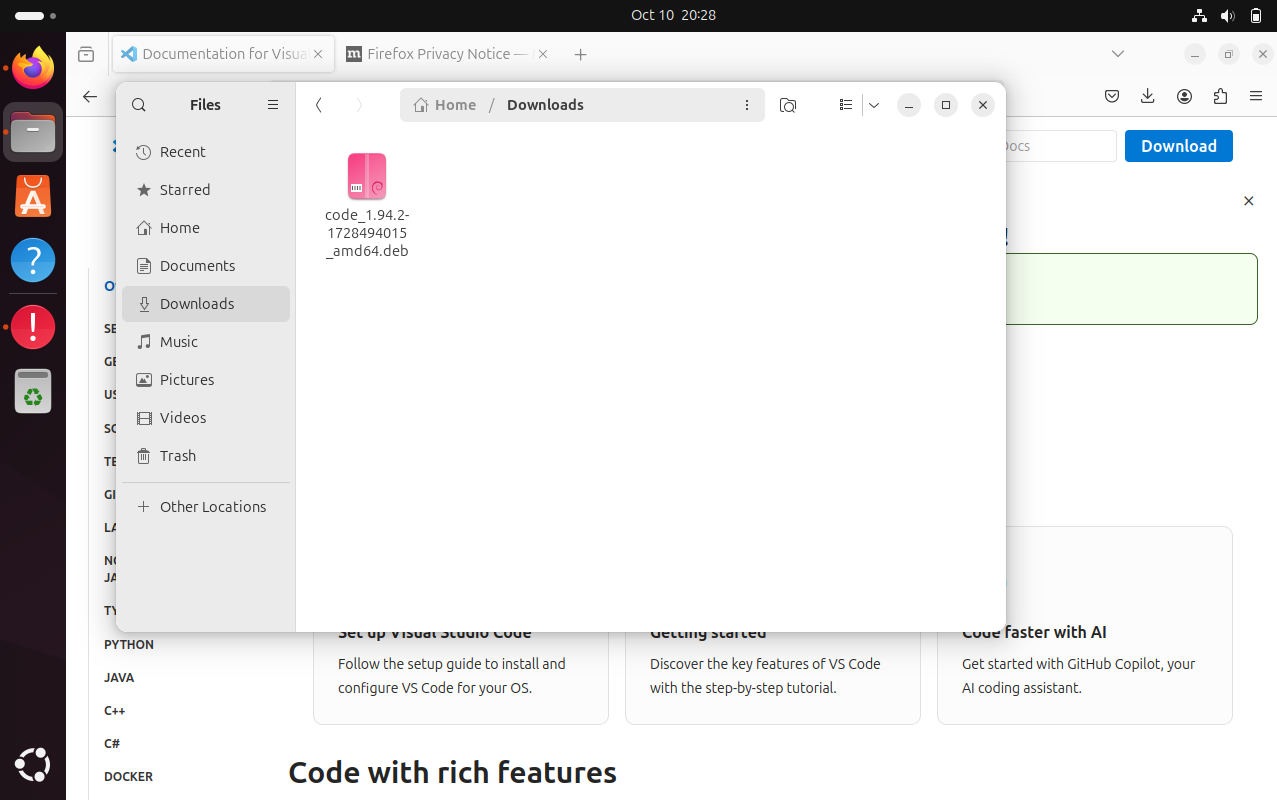


Step#1: installation of Ubuntu completed.

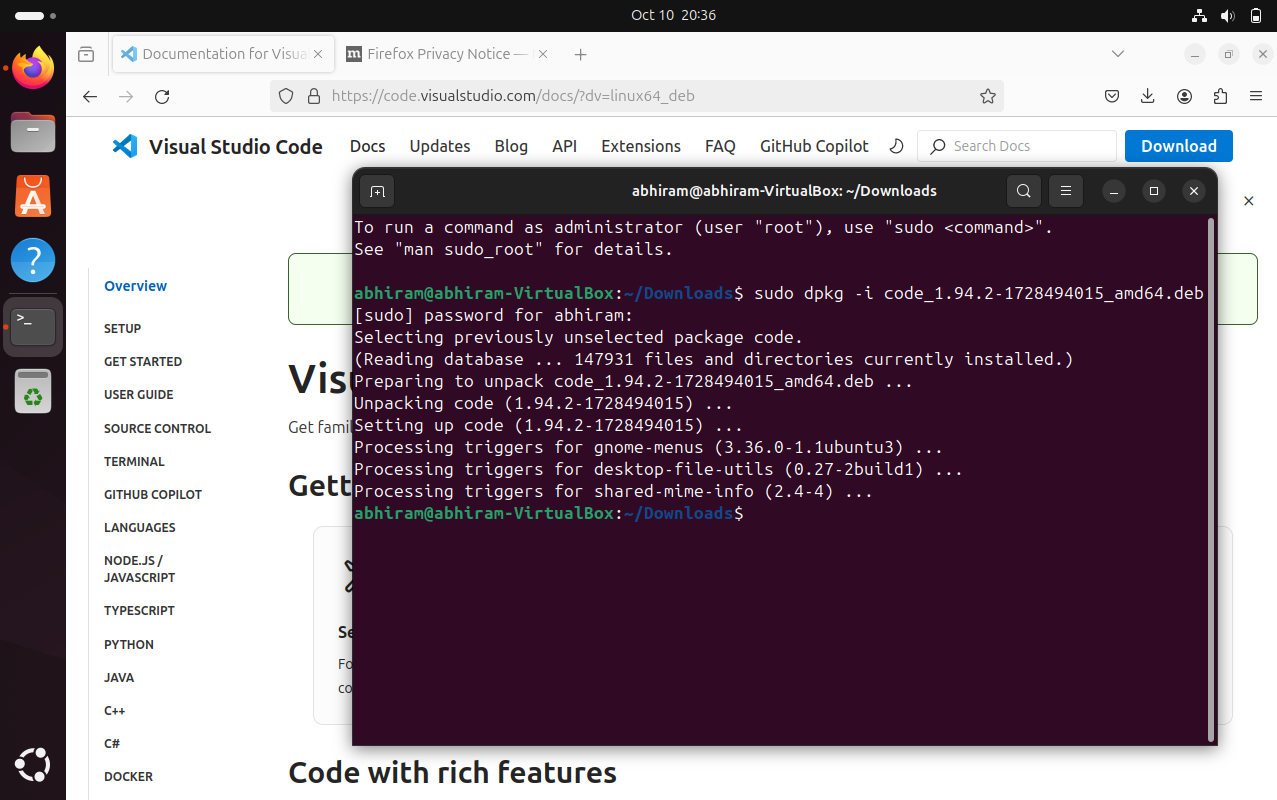
**Step#2: Set up Visual Studio code on Ubuntu VM**

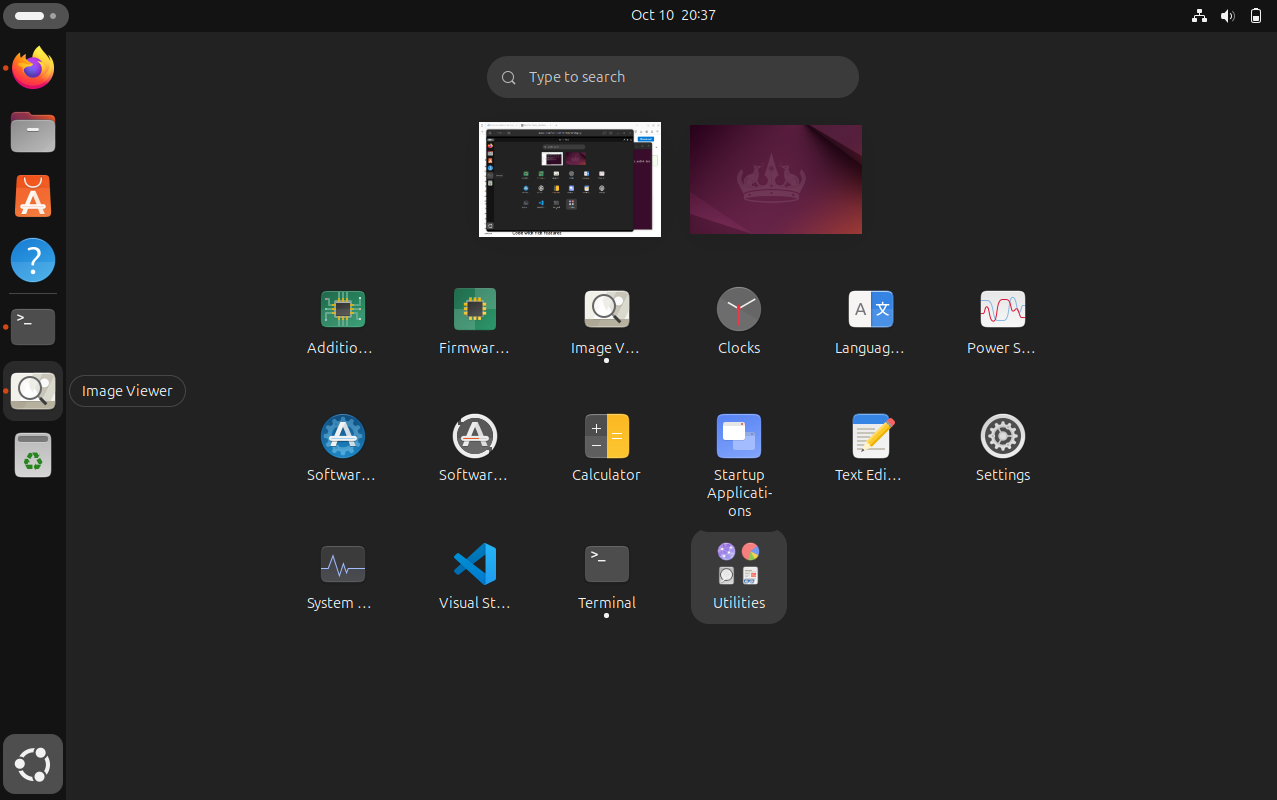
**2.1 download .deb file for VS code**

****



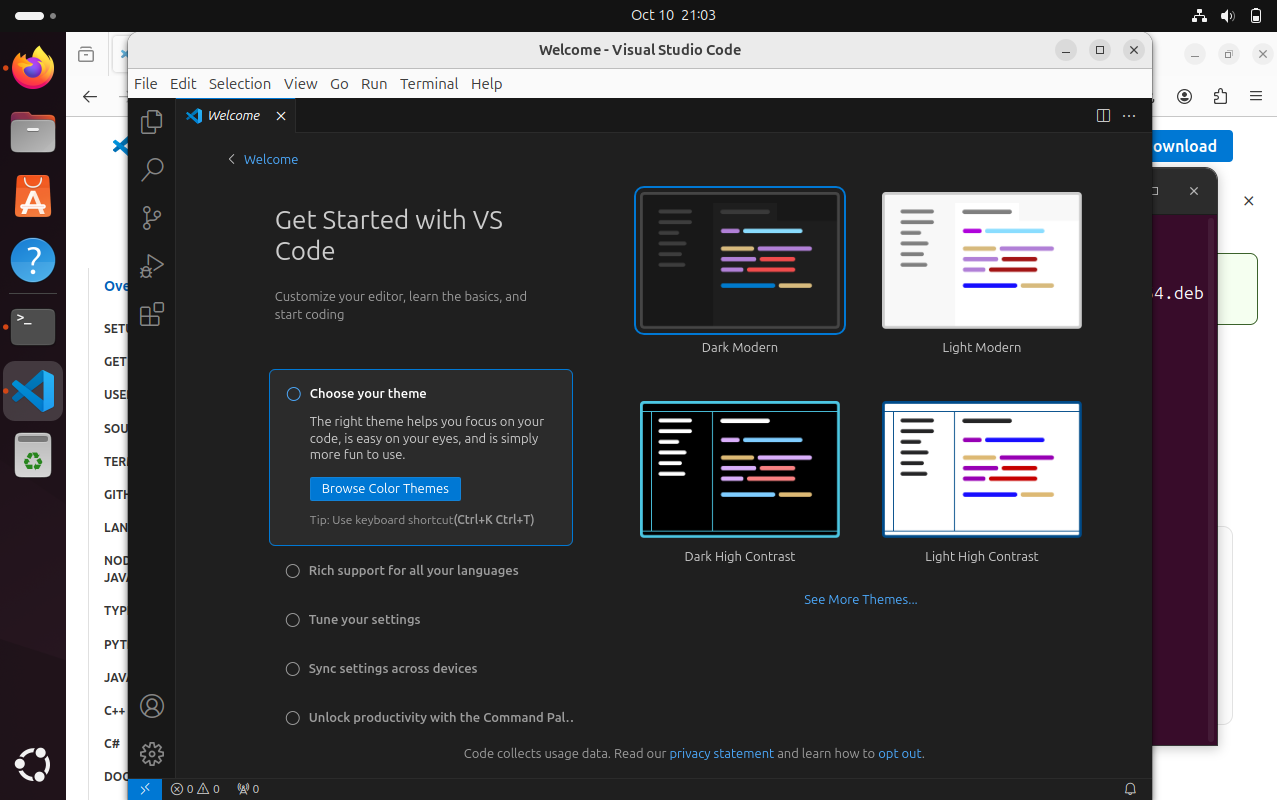
* 1. Open terminal, go to Downloads directory and use command : sudo dpkg -i code\_1.91.1- 1720564633\_adm64.deb to install VS code.





* 1. VS code installation completed

2.4 Verify VS code. Go to applications and open it

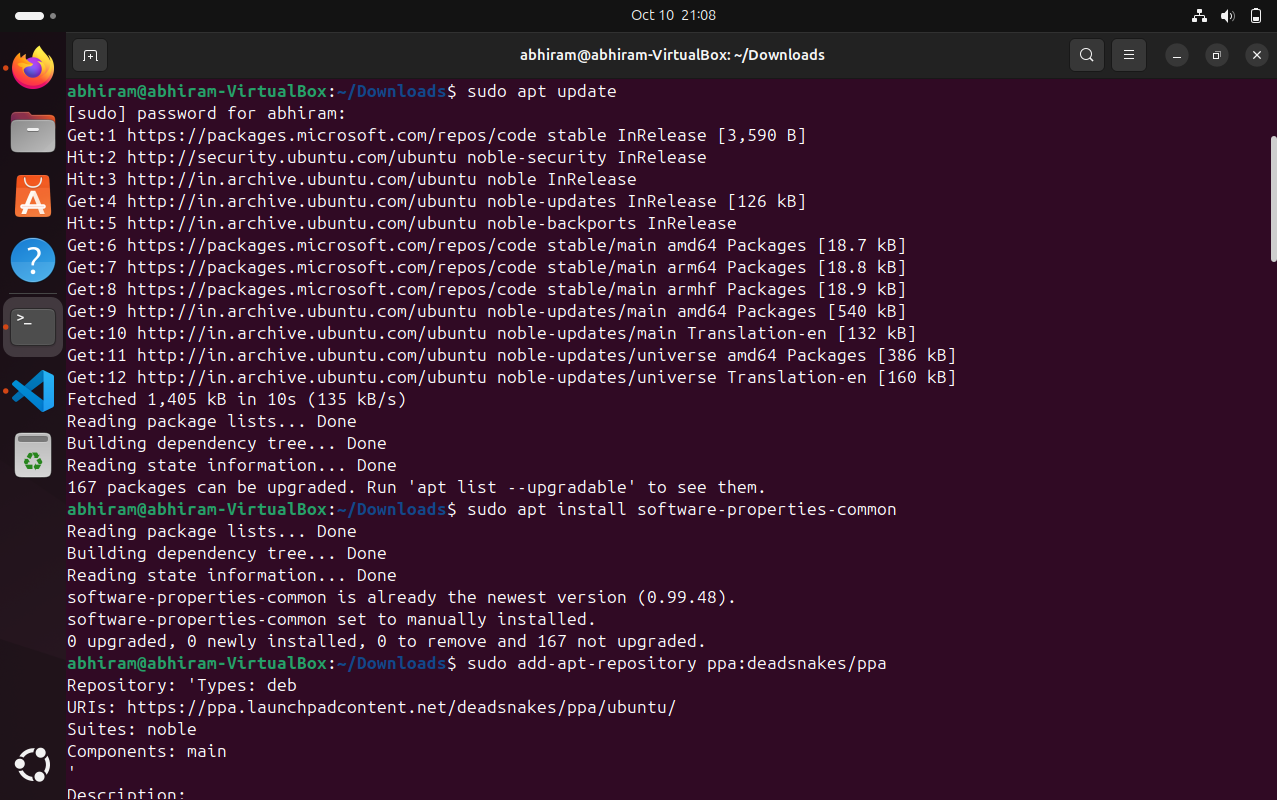


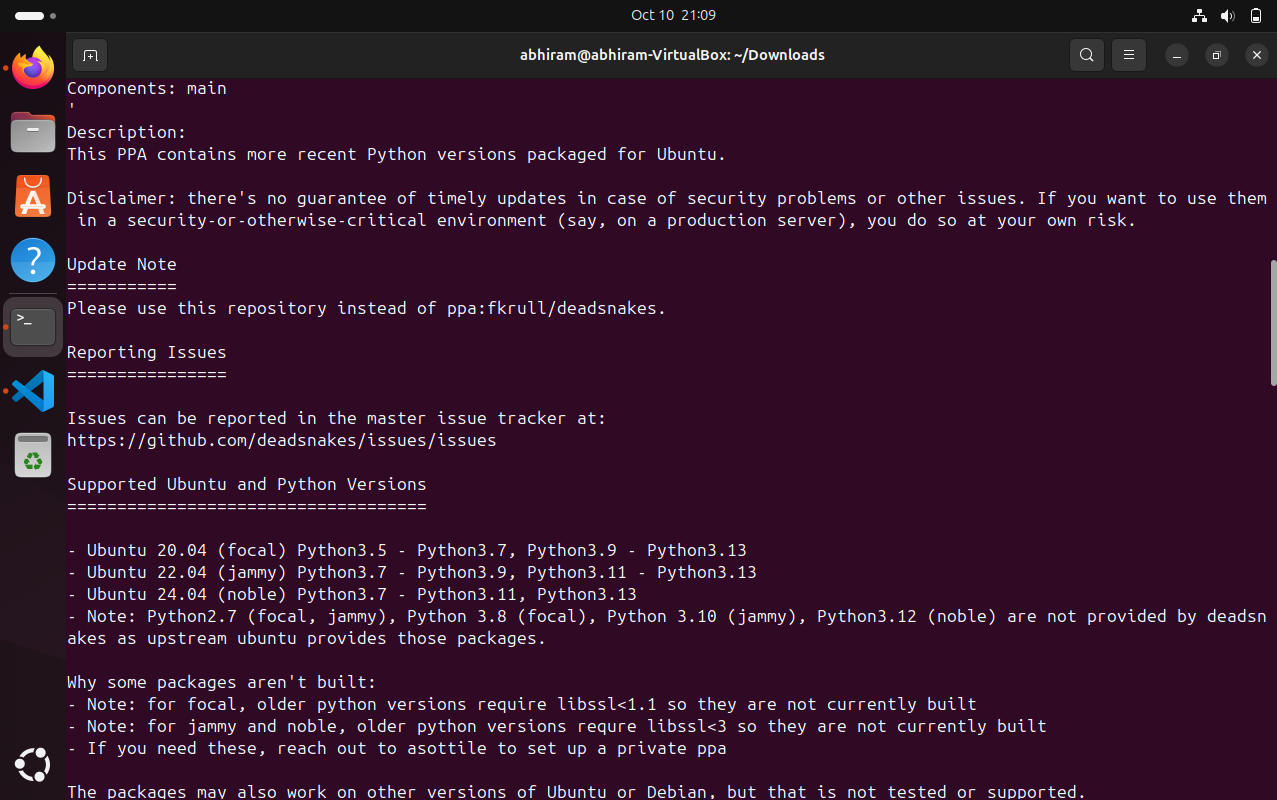
VS code installation completed. Any further extensions required can be installed as normal.

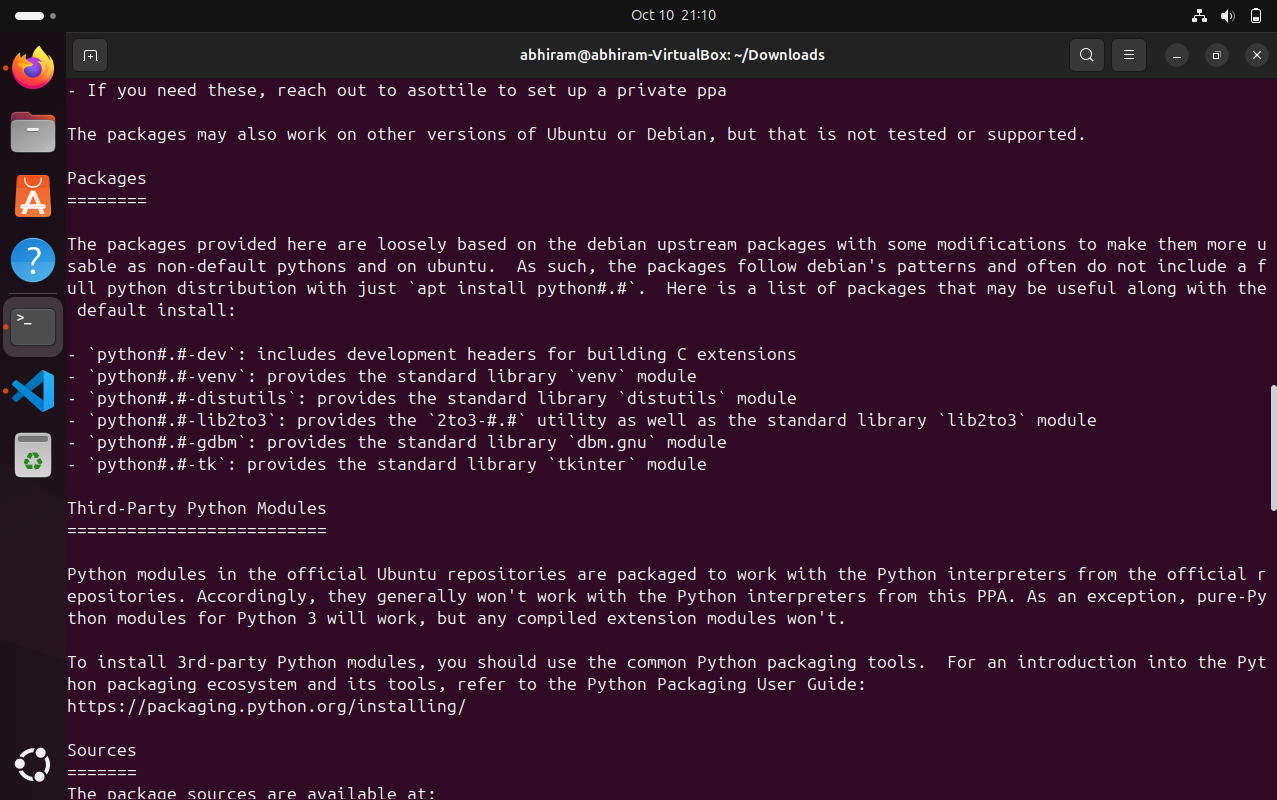
# Set up Python.

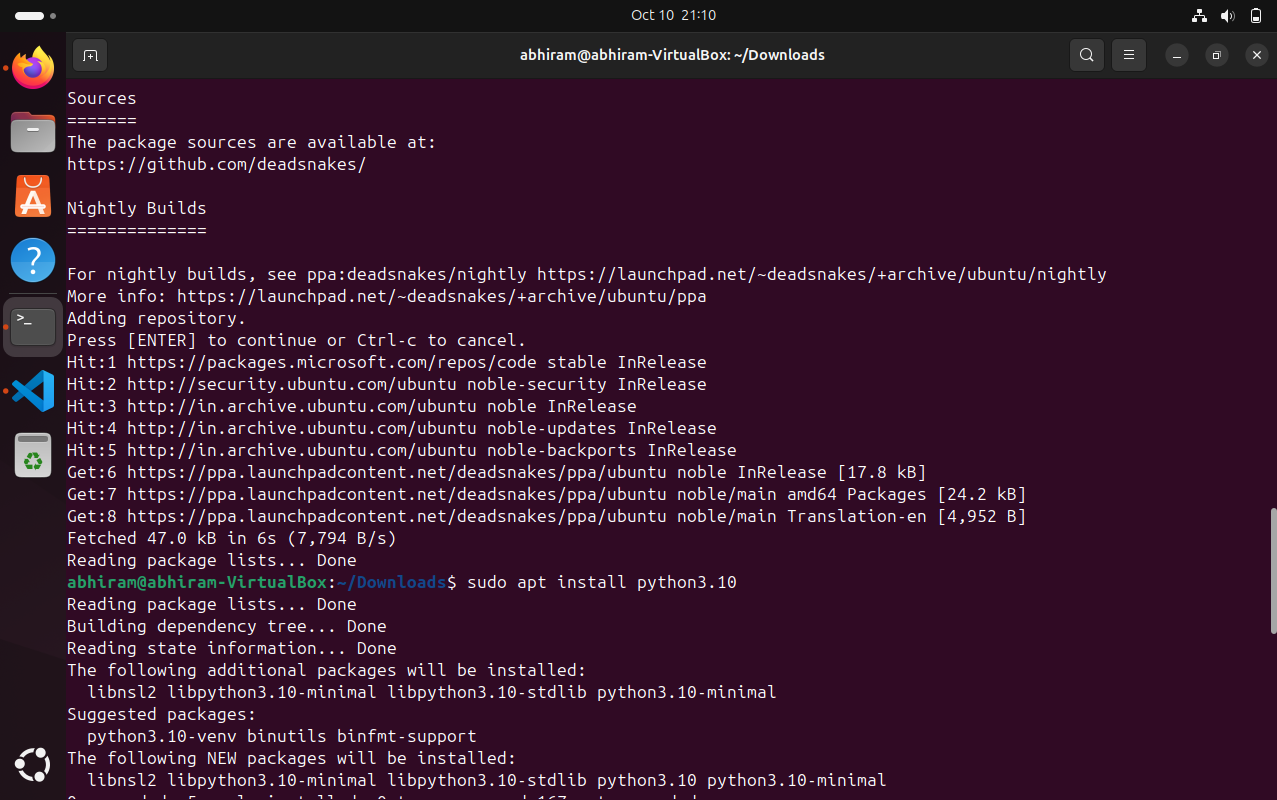
* 1. : go to terminal use the following commands in sequence: sudo apt update

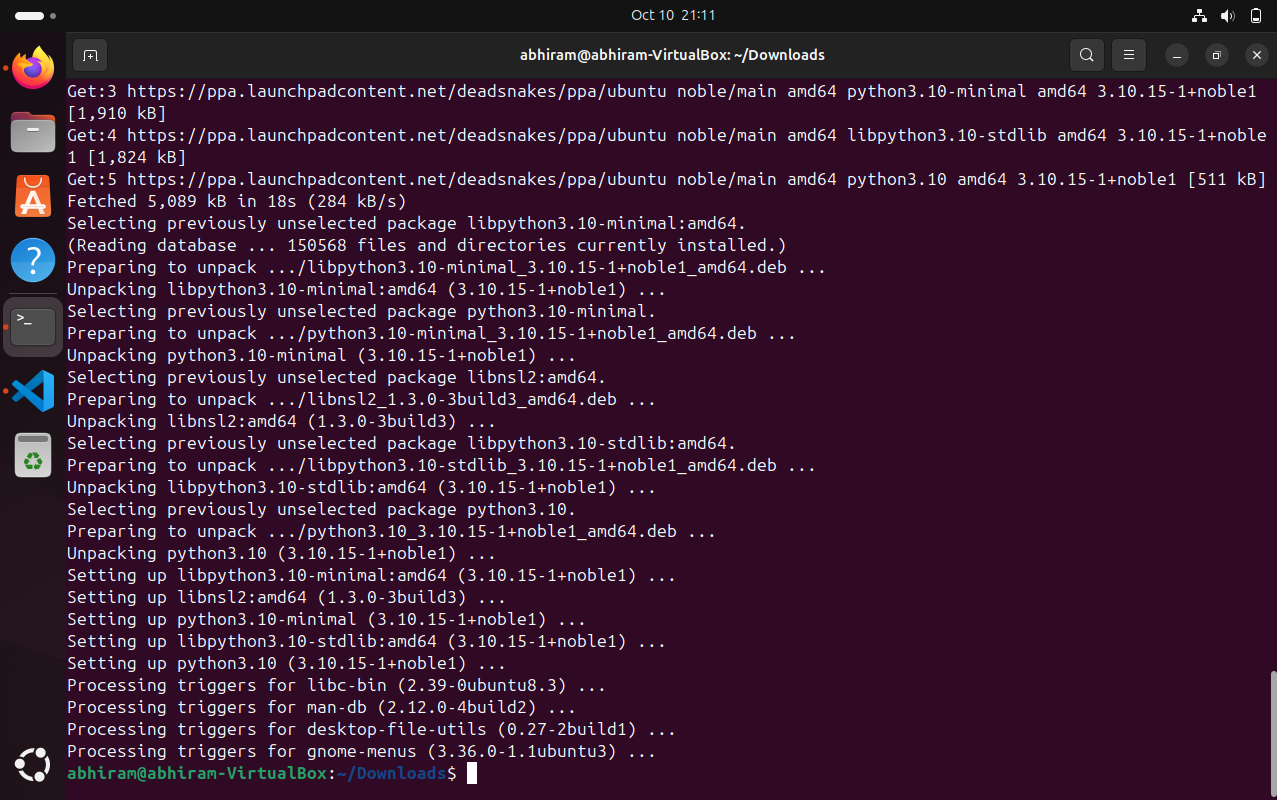
sudo apt install software-properties-common sudo add-apt-repository ppa:deadsnakes/ppa sudo apt install python3.10











Python 3.10 installation completed Verify python version

Command : python3.10 -V and python3.10

abhiram@abhiram-ViirtualBox:~$python3.10 -V

Python 3.10.15

abhiram@abhiram-VirtualBox:~$python3.10

Python 3.10.15 (main, Sep 7 2024 , 18:35:38) [GCC 13.2.0] on linux

Type “help”, “copyright”, “credits” or “license” for more information.

>>> print(“Week10-12 Assignment”)

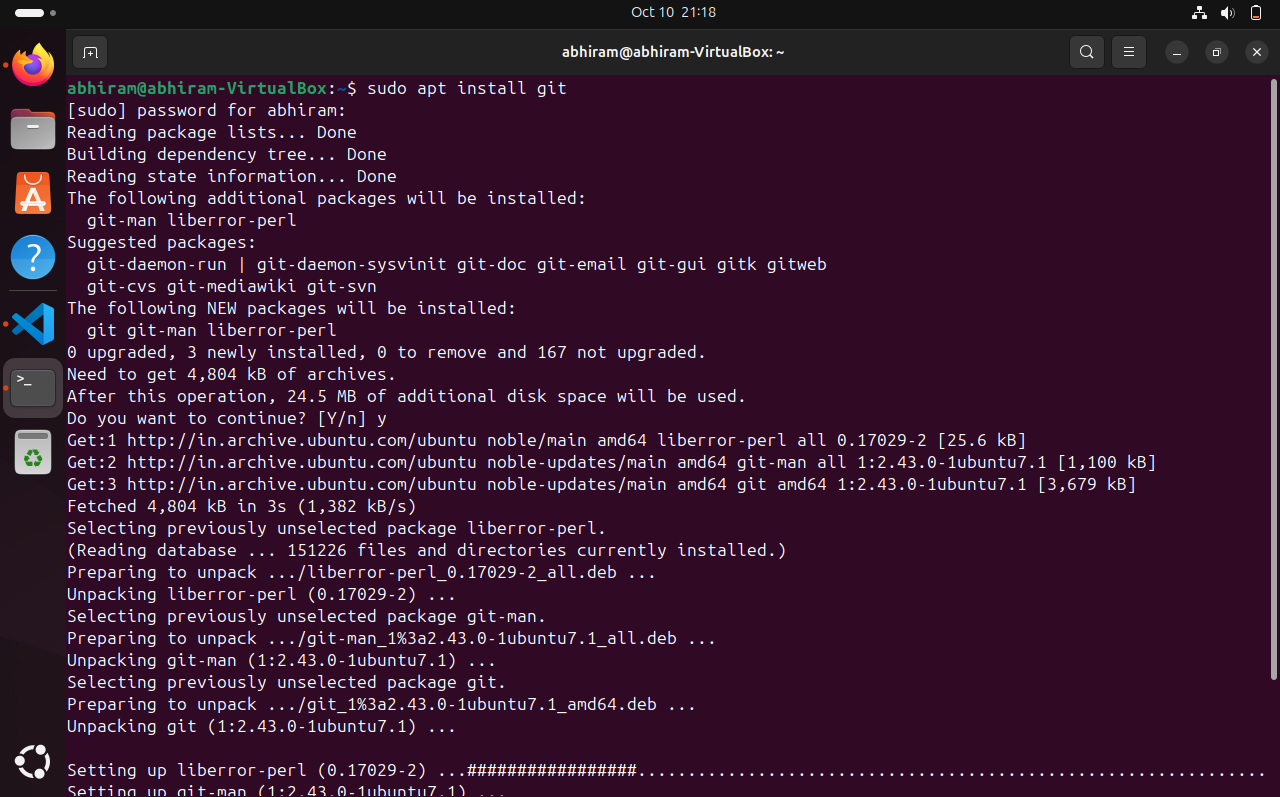
Week10-12 Assignment

>>>

1. Clone this Github repository <https://github.com/Vikas098766/Microservices.git>

Install git

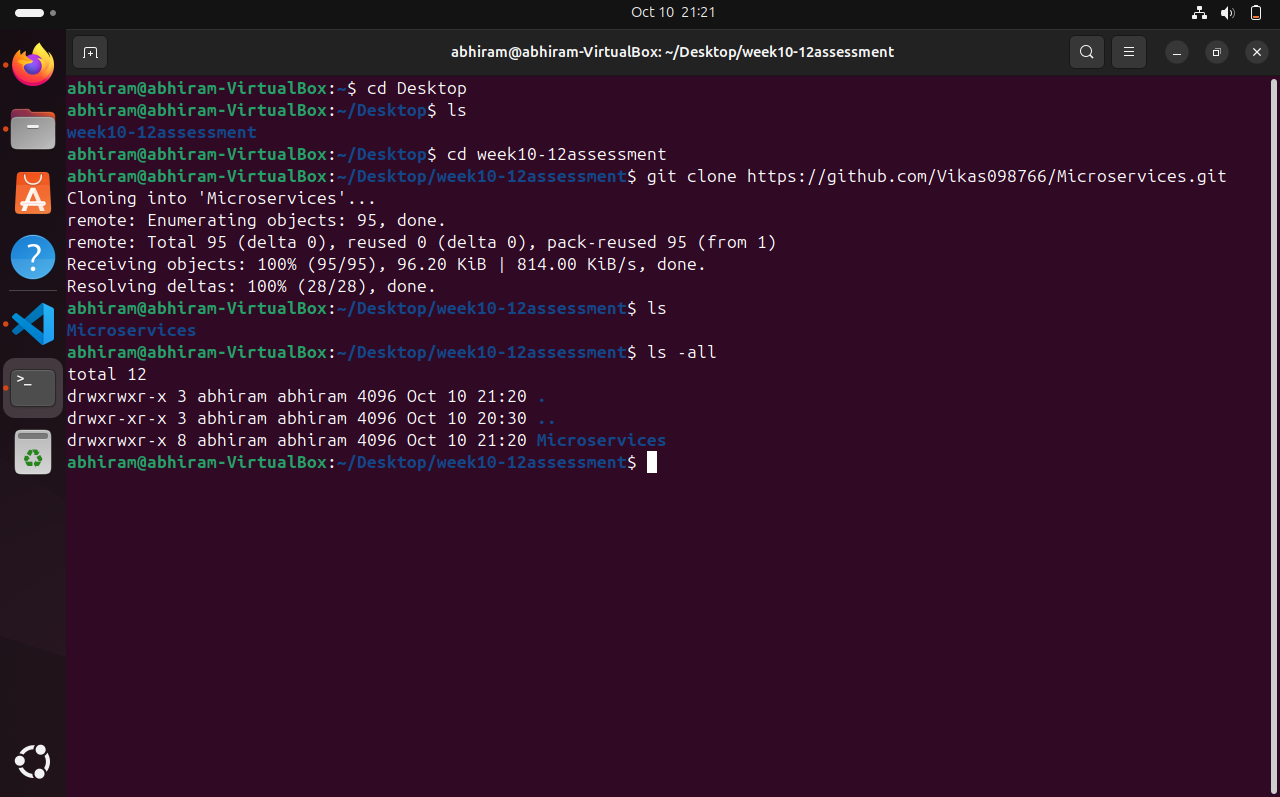
Command : sudo apt install git



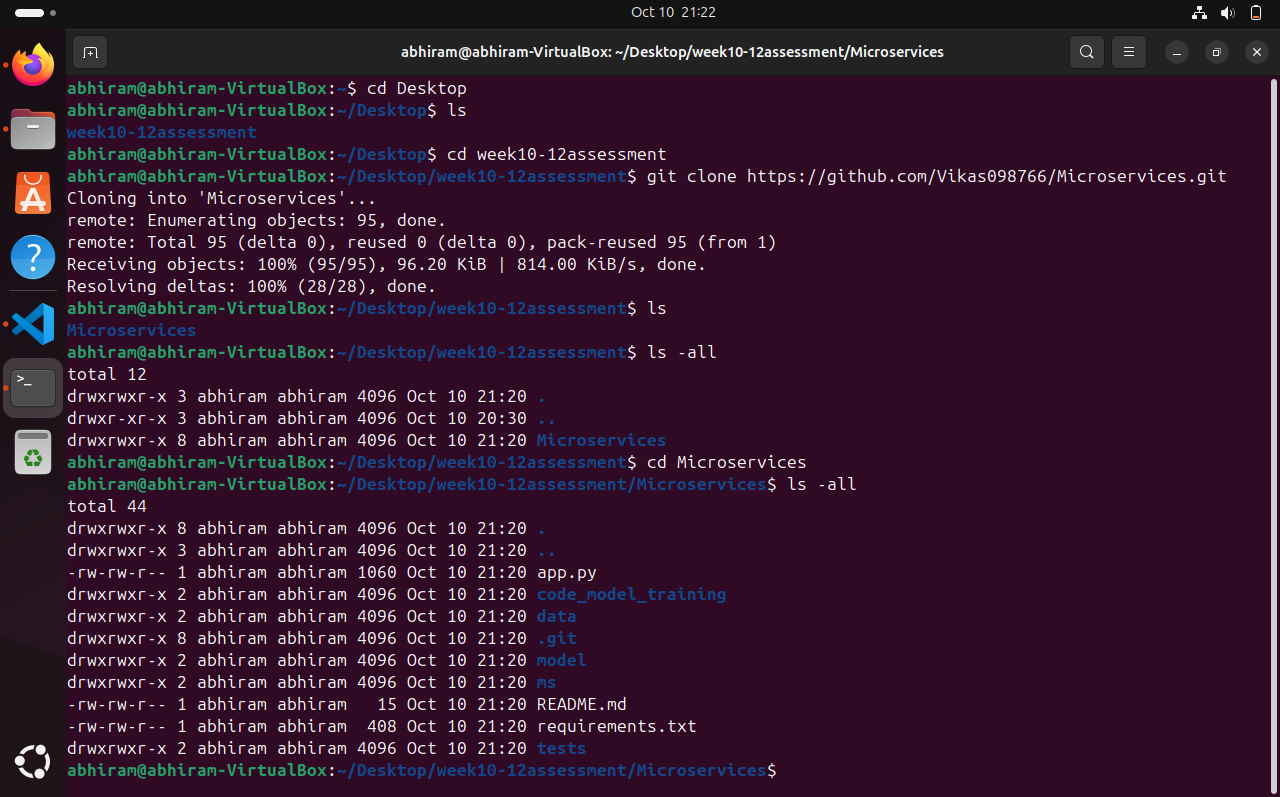
Git installation completed.

To Clone repository. go to the directory Desktop the use

command : git clone https://github.com/Vikas098766/Microservices.git



Verify the repository details

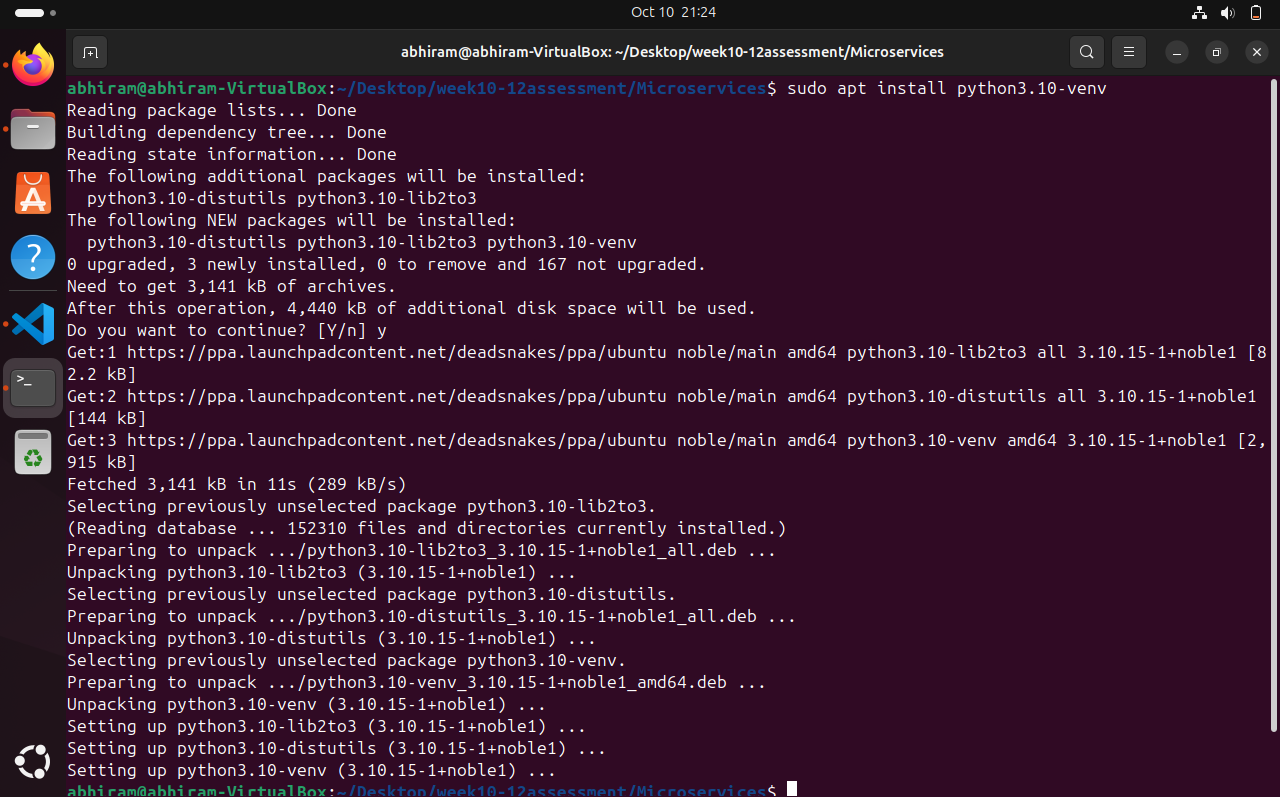


Cloning of repository <https://github.com/Vikas098766/Microservices.git>completed

Step#5. Create a Virtual Environment

# Go to the directory of cloned repository Microservices.

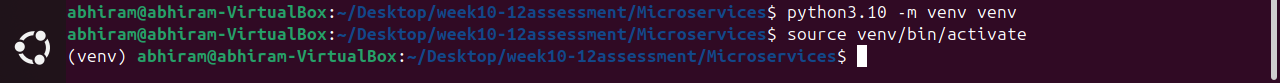
Use command : sudo apt install python3.10-venv



To activate environment

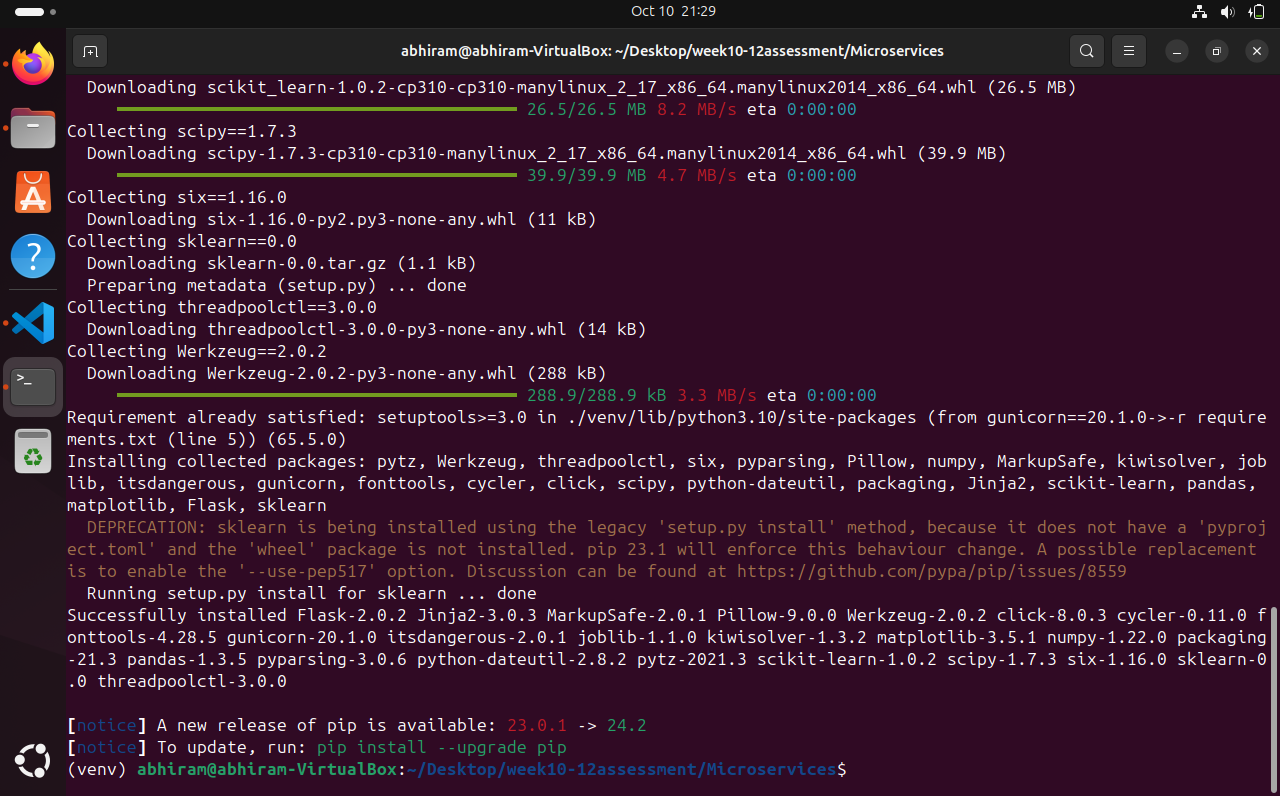
Commands :python3.1 -m venv venv

source venv/bin/activate



Step#6 Install the dependencies from requirements.txt file. Command : pip install -r requirements.txt





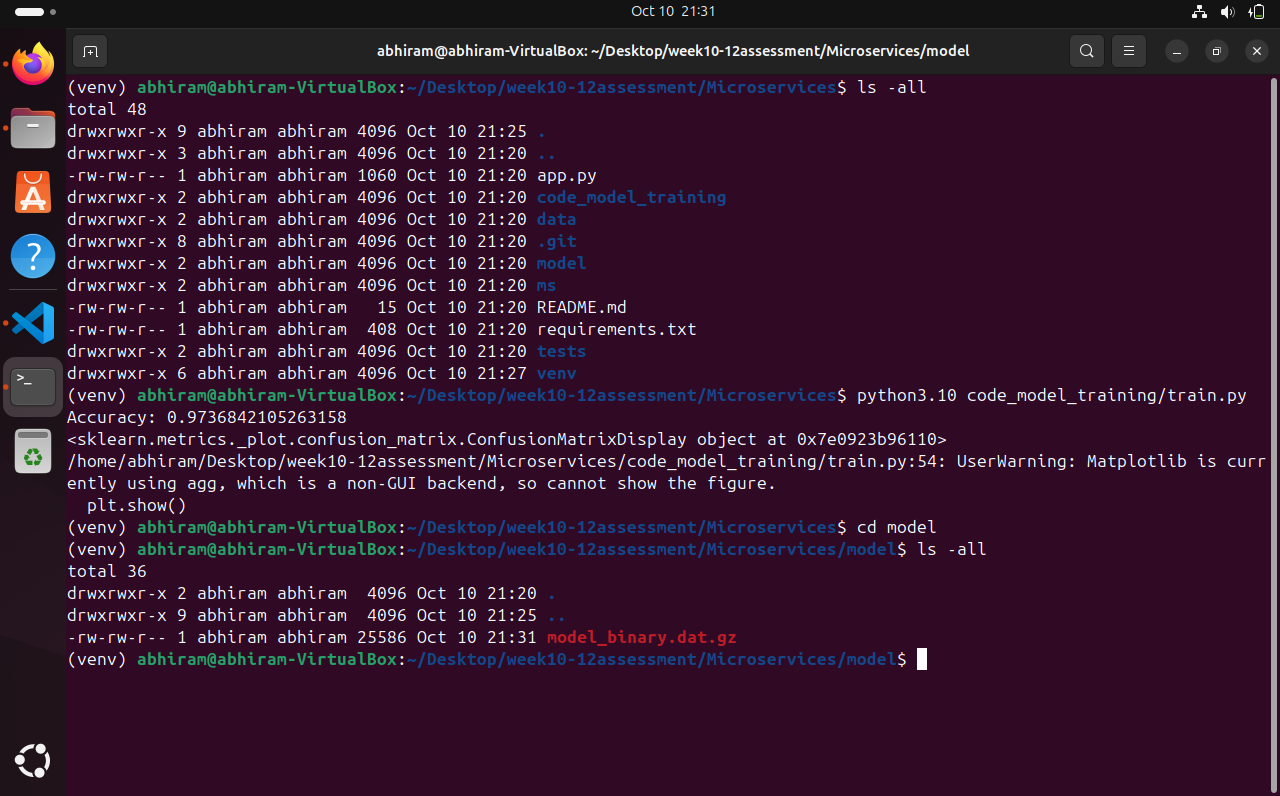
Requirement installation completed

Step#7 : Train and save the model

# Open terminal and Go to the Desktop/Microservices directory and run below command

Command : python3.10 code\_model\_training/train.py

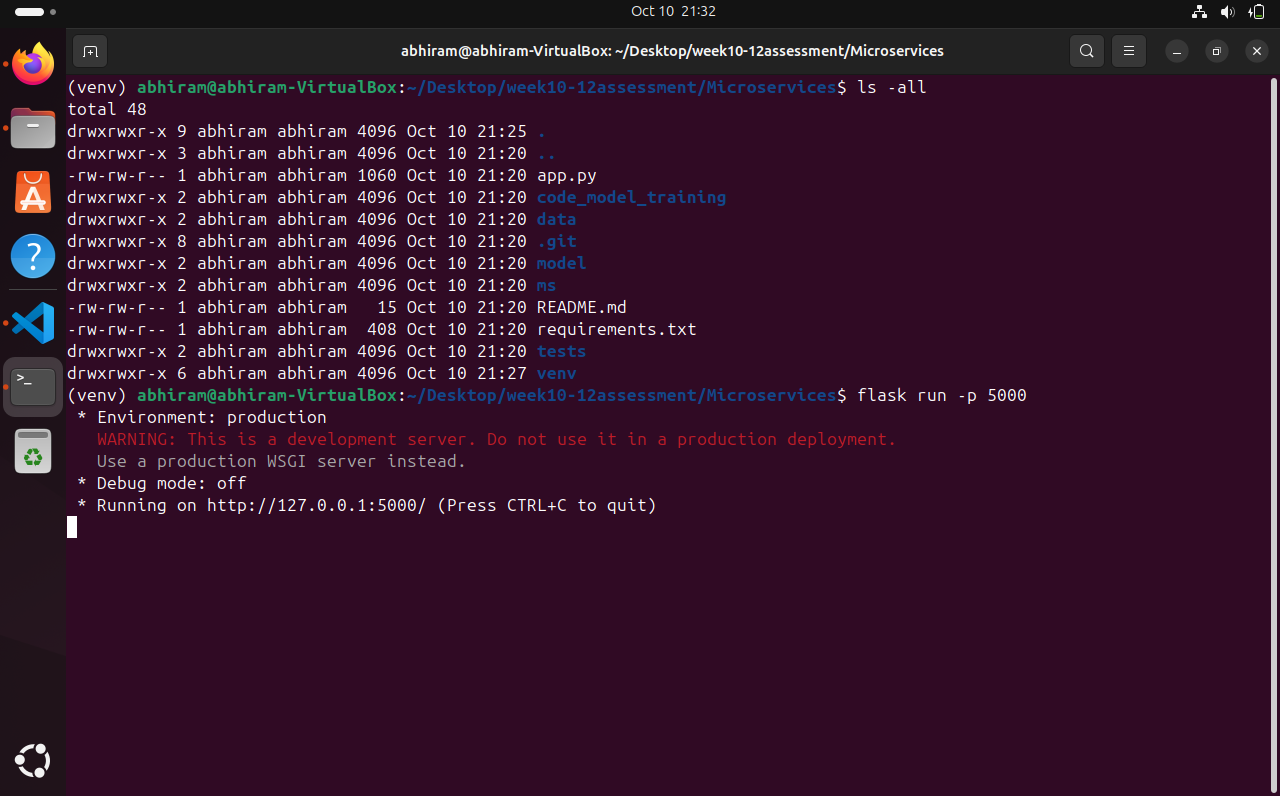
Verify model saved in model directory with file name “model\_binary.dat.gz”



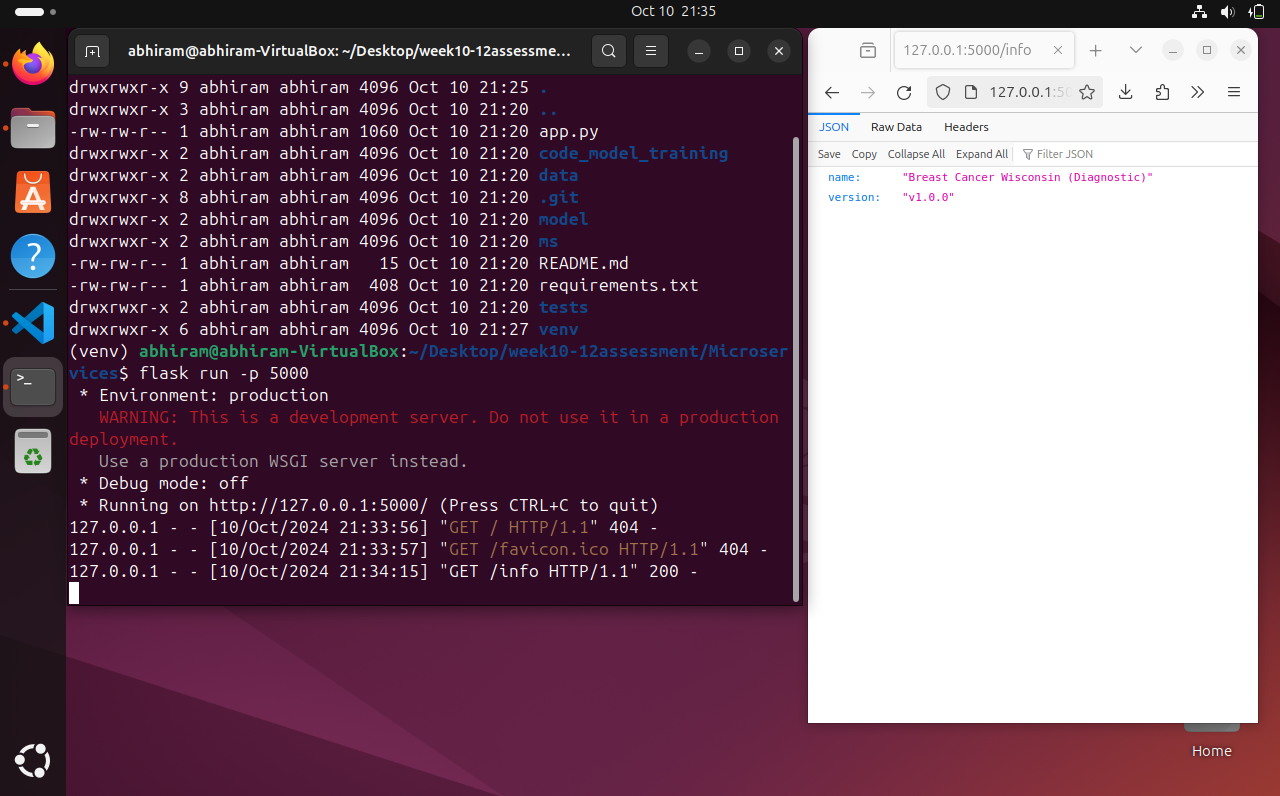
Step#7 :completed

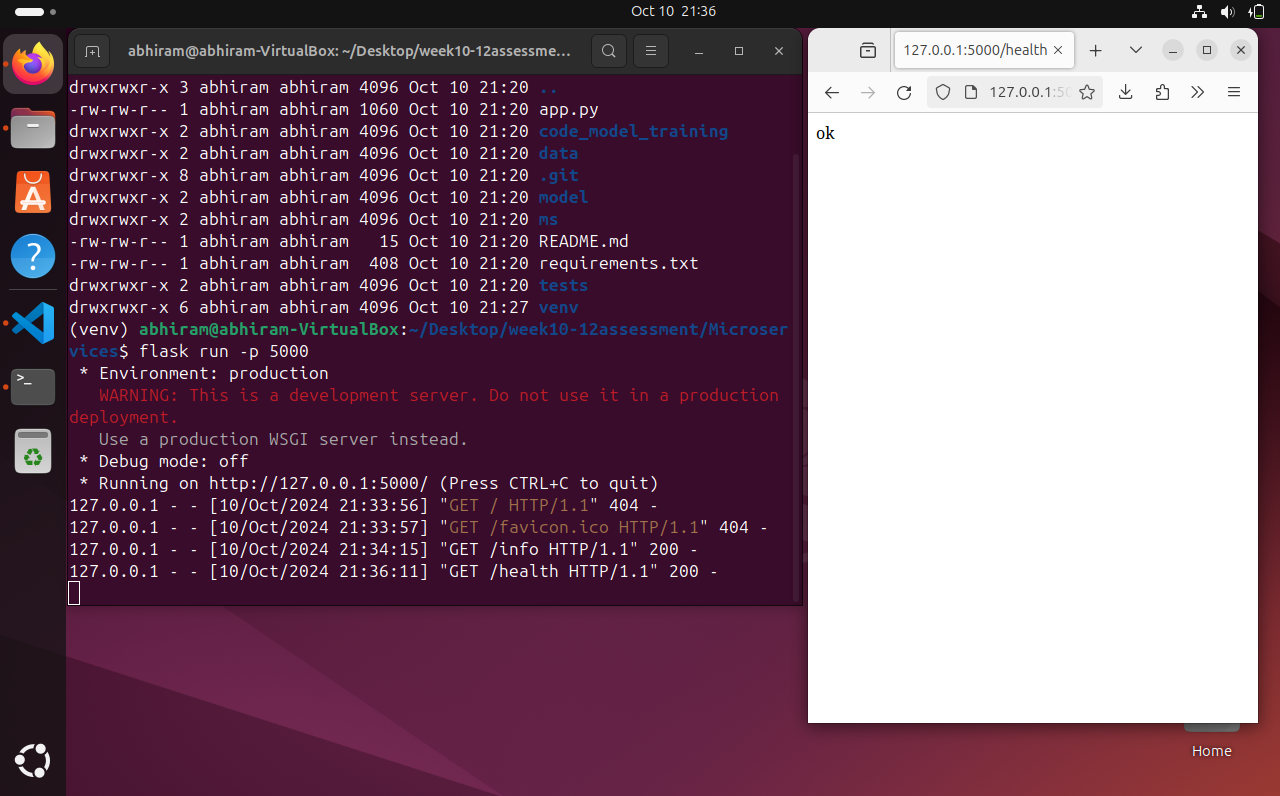
Step#8: **Test the Flask web application**

# Command : flask run -p 5000



To test web : Open the browser and test the GET endpoints http://localhost:5000/info and http://localhost:5000/health





Step#9. Test the application and make predictions using the example calls available in the folder**/tests**

curl -X GET http://localhost:5000/info curl -X GET http://localhost:5000/health

curl -d '[{"radius\_mean": 17.99, "texture\_mean": 10.38, "perimeter\_mean": 122.8, "area\_mean": 1001.0,

"smoothness\_mean": 0.1184, "compactness\_mean": 0.2776, "concavity\_mean": 0.3001, "concave points\_mean":

0.1471, "symmetry\_mean": 0.2419, "fractal\_dimension\_mean": 0.07871, "radius\_se": 1.095, "texture\_se": 0.9053,

"perimeter\_se": 8.589, "area\_se": 153.4, "smoothness\_se": 0.006399, "compactness\_se": 0.04904, "concavity\_se":

0.05373, "concave points\_se": 0.01587, "symmetry\_se": 0.03003, "fractal\_dimension\_se": 0.006193, "radius\_worst":

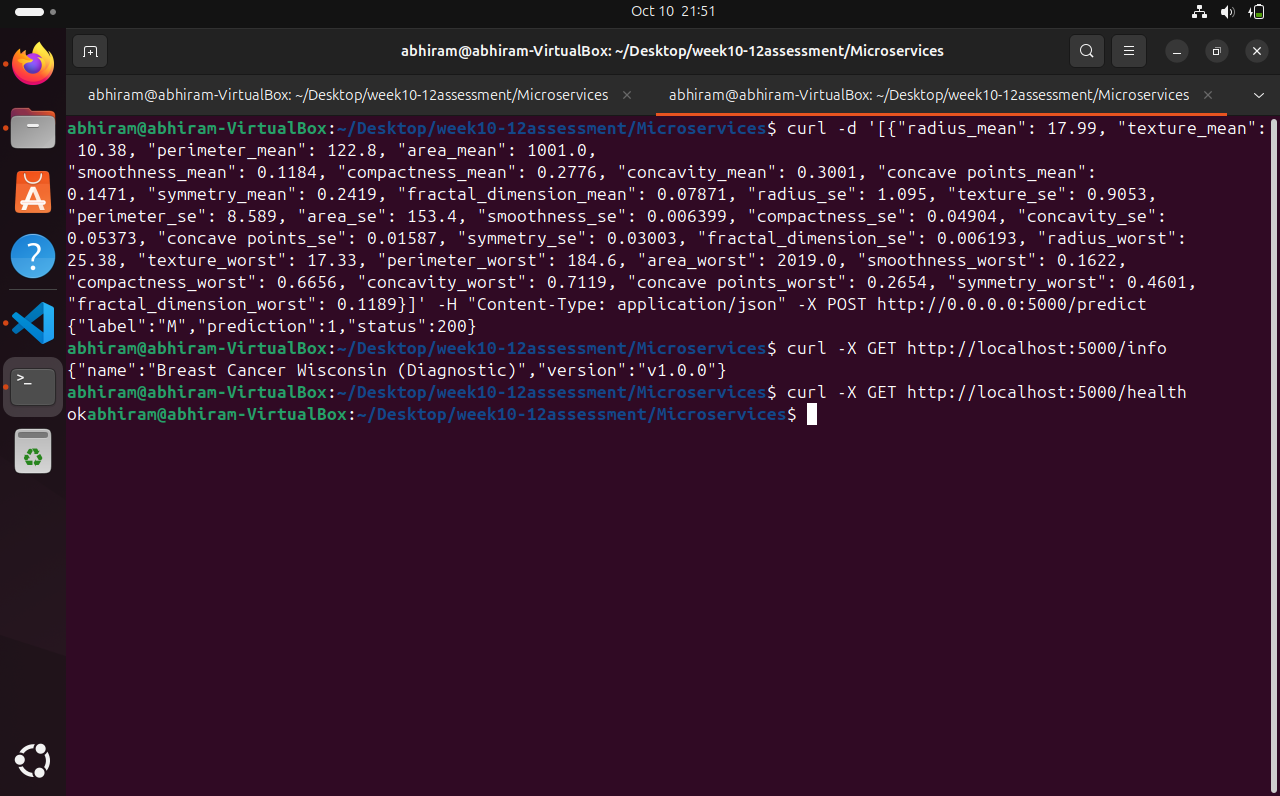
25.38, "texture\_worst": 17.33, "perimeter\_worst": 184.6, "area\_worst": 2019.0, "smoothness\_worst": 0.1622,

"compactness\_worst": 0.6656, "concavity\_worst": 0.7119, "concave points\_worst": 0.2654, "symmetry\_worst": 0.4601,

"fractal\_dimension\_worst": 0.1189}]' \

-H "Content-Type: application/json" \

-X POST http://0.0.0.0:5000/predict



# Step#10 : Create a docker image containing everything needed to run the application Install Docker – Follow the below commands on terminal in sequence

#Install a few prerequisite packages which let apt use packages over HTTPS#

sudo apt install apt-transport-https ca-certificates curl software-properties-common

#add the GPG key for the official Docker repository to your system:#

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

#Add the Docker repository to APT sources#

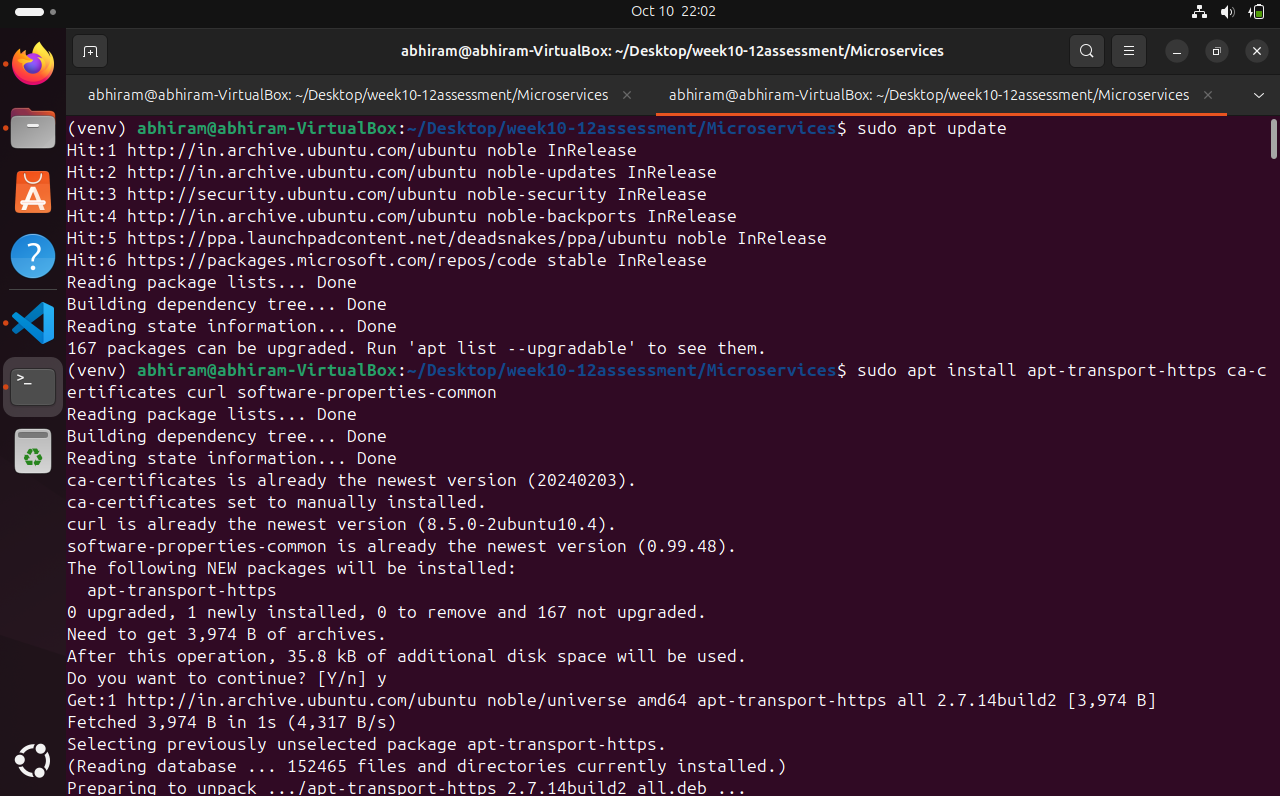
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"

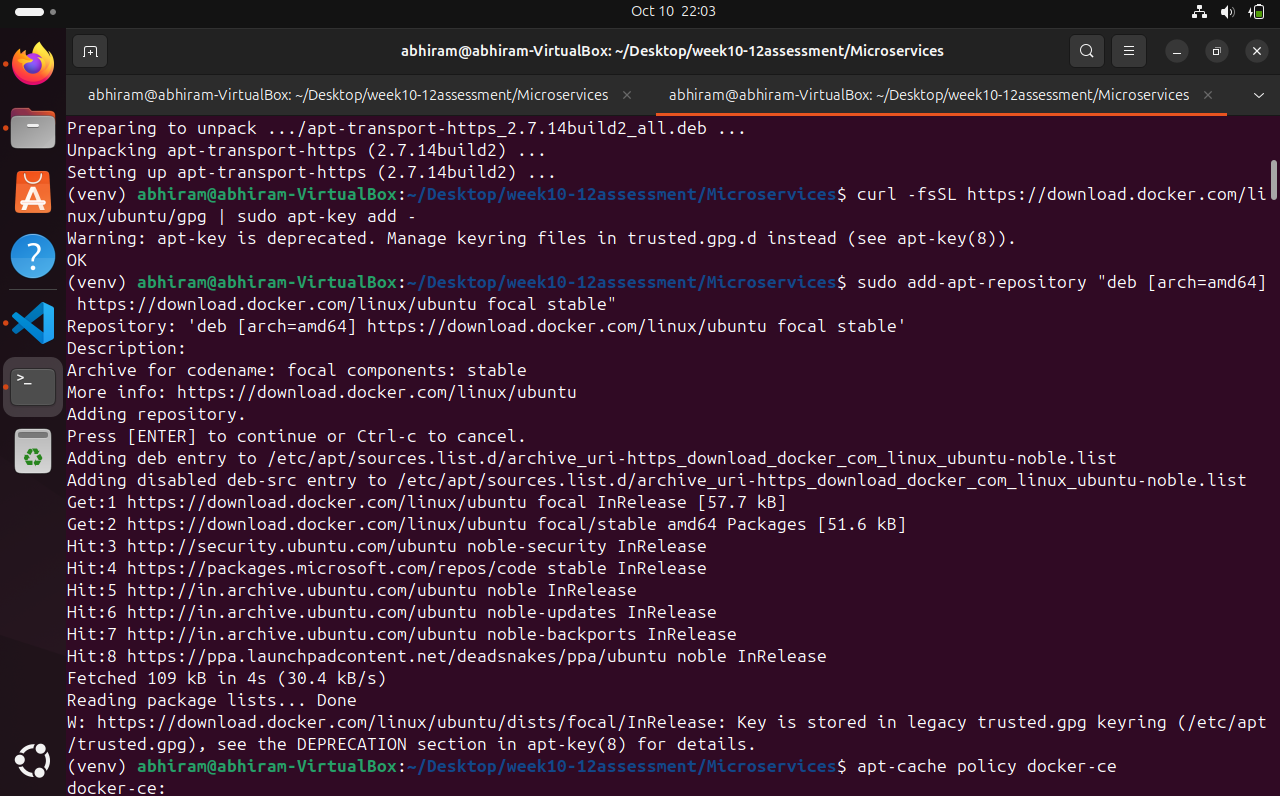
#This will also update our package database with the Docker packages from the newly added repo. Make sure you ##are about to install from the Docker repo instead of the default Ubuntu repo:#

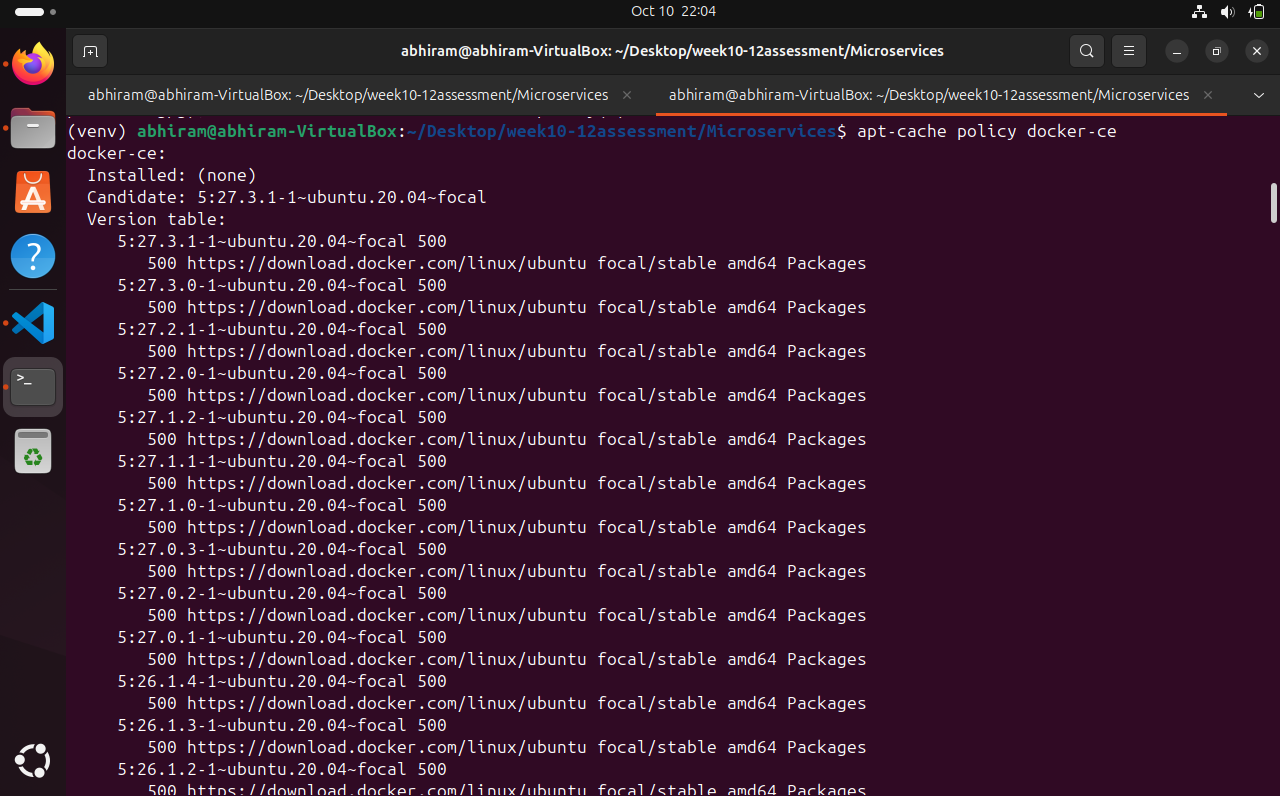
apt-cache policy docker-ce

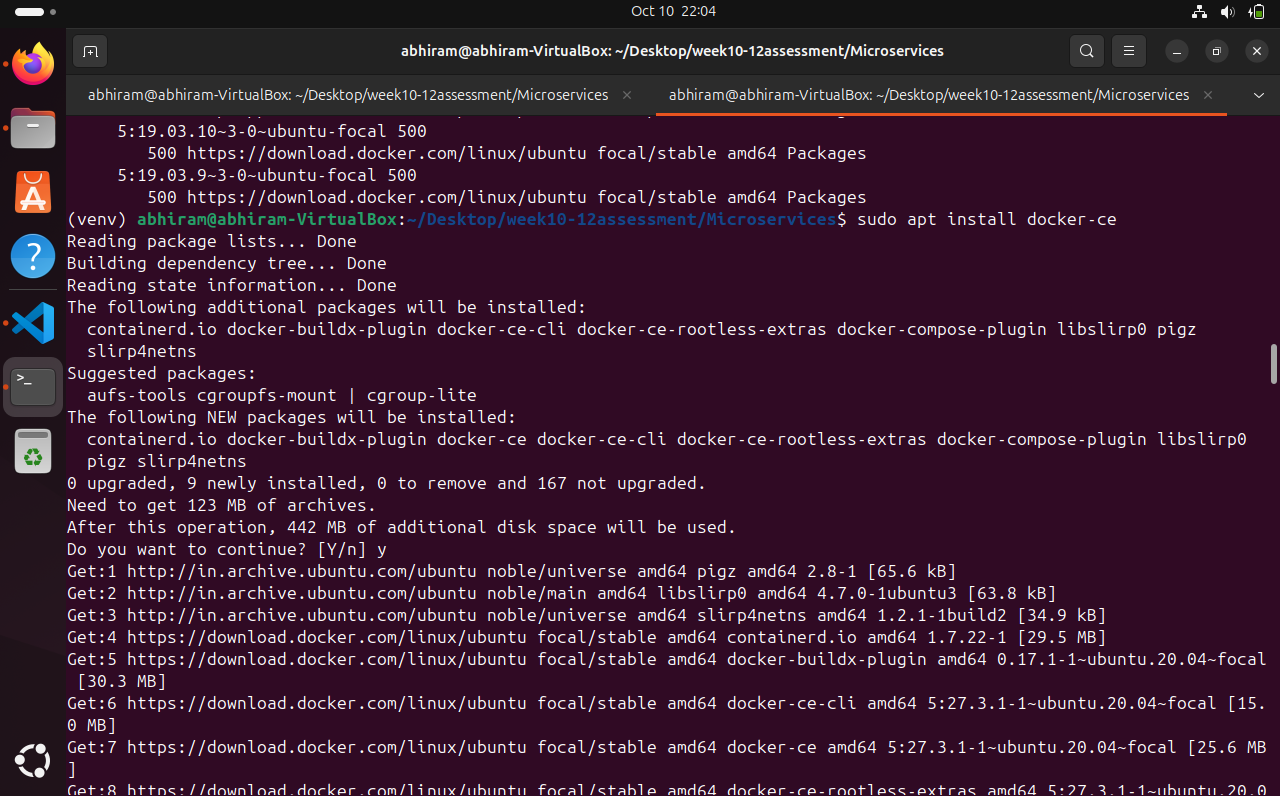
#Finally install Docker# sudo apt install docker-ce

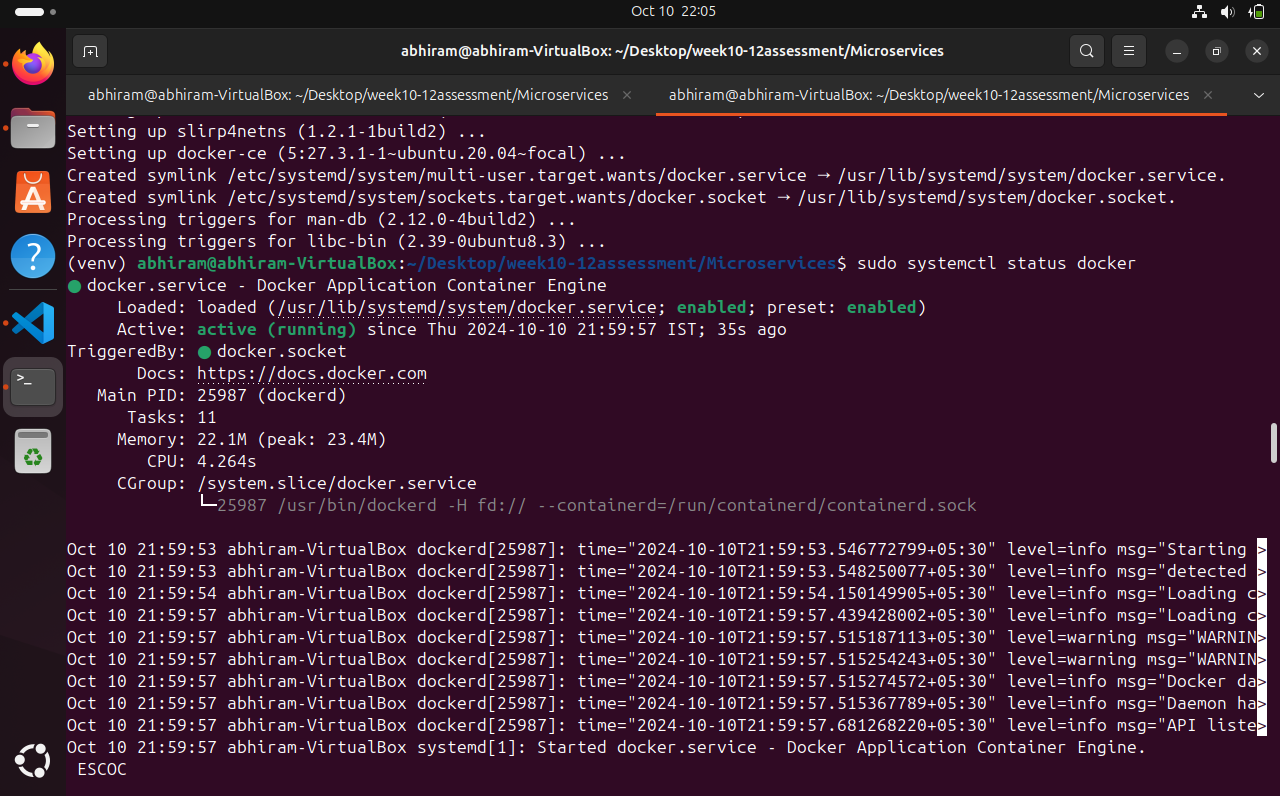
#Verify Docker running# sudo systemctl status docker











# Grant access rights



Open project in VS code, create Docker file with following lines and save.

FROM python:3-10-bullseye

WORKDIR /Microservcies

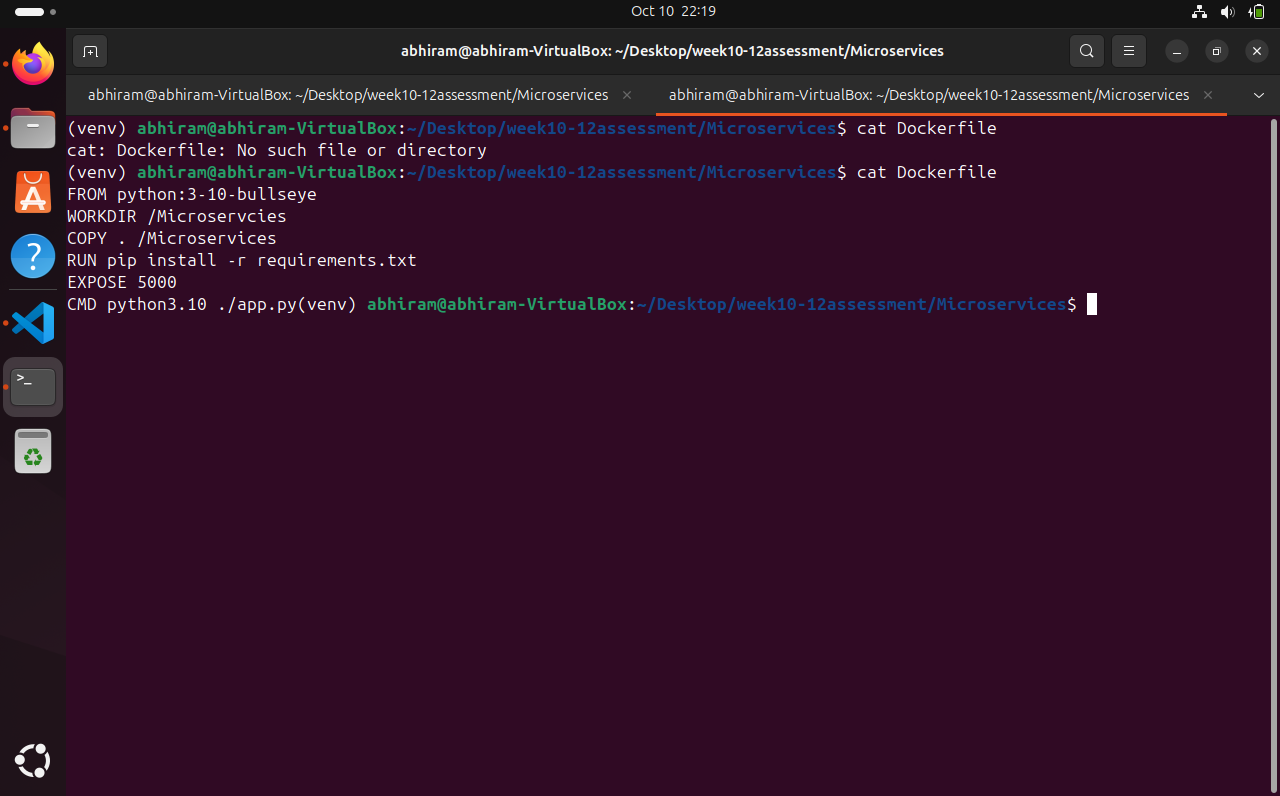
COPY . /Microservices

RUN pip install -r requirements.txt

EXPOSE 5000

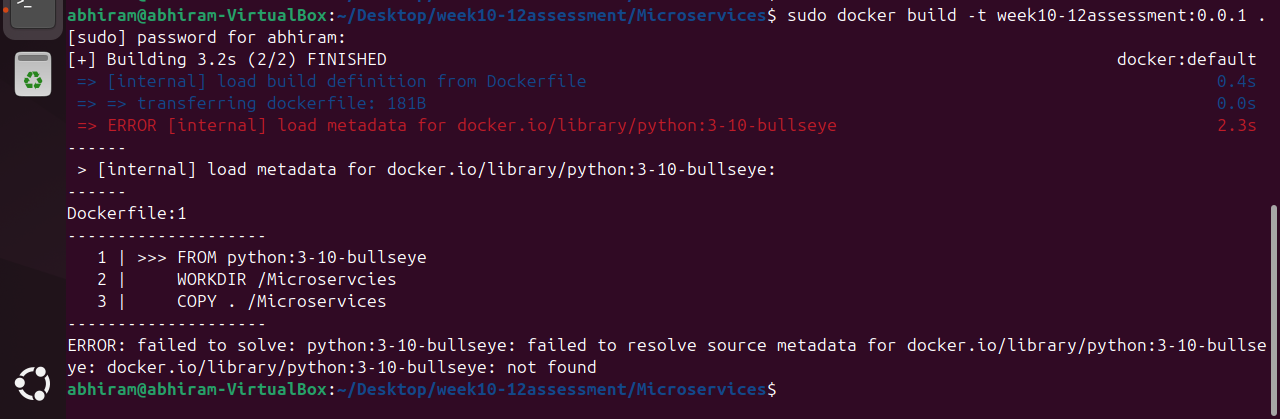
CMD python3.10 ./app.py

Check the docker file from the directory of repository: open Desktop/Microservices and use command CAT Dockerfile



Make sure you are in repository directory and use the command to build the image name week11assingment and version 0.0.1

Docker build -t week10-12assessment:0.0.1 .



Completed docker image build, verify the image Command : docker images