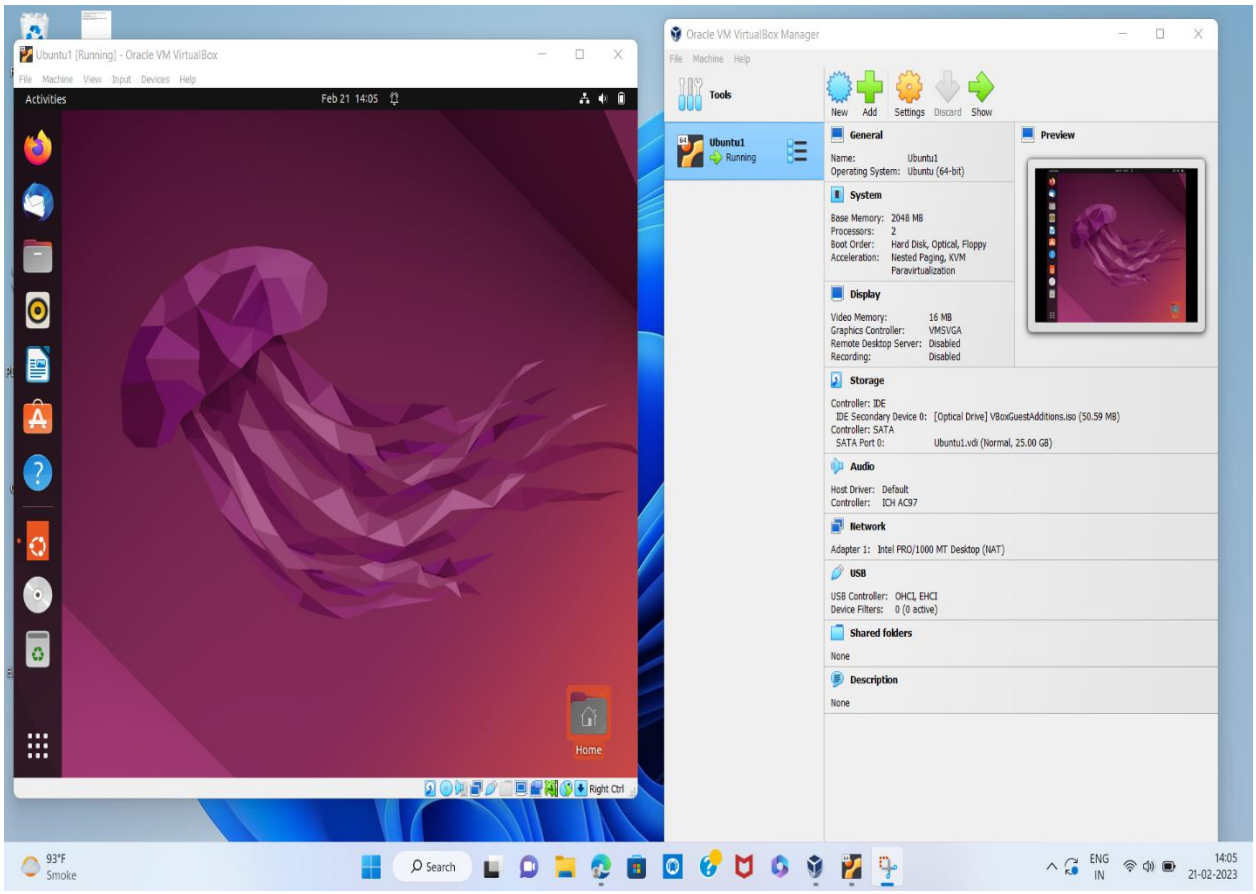
WEEK 11 ASSIGNMENT

Steps

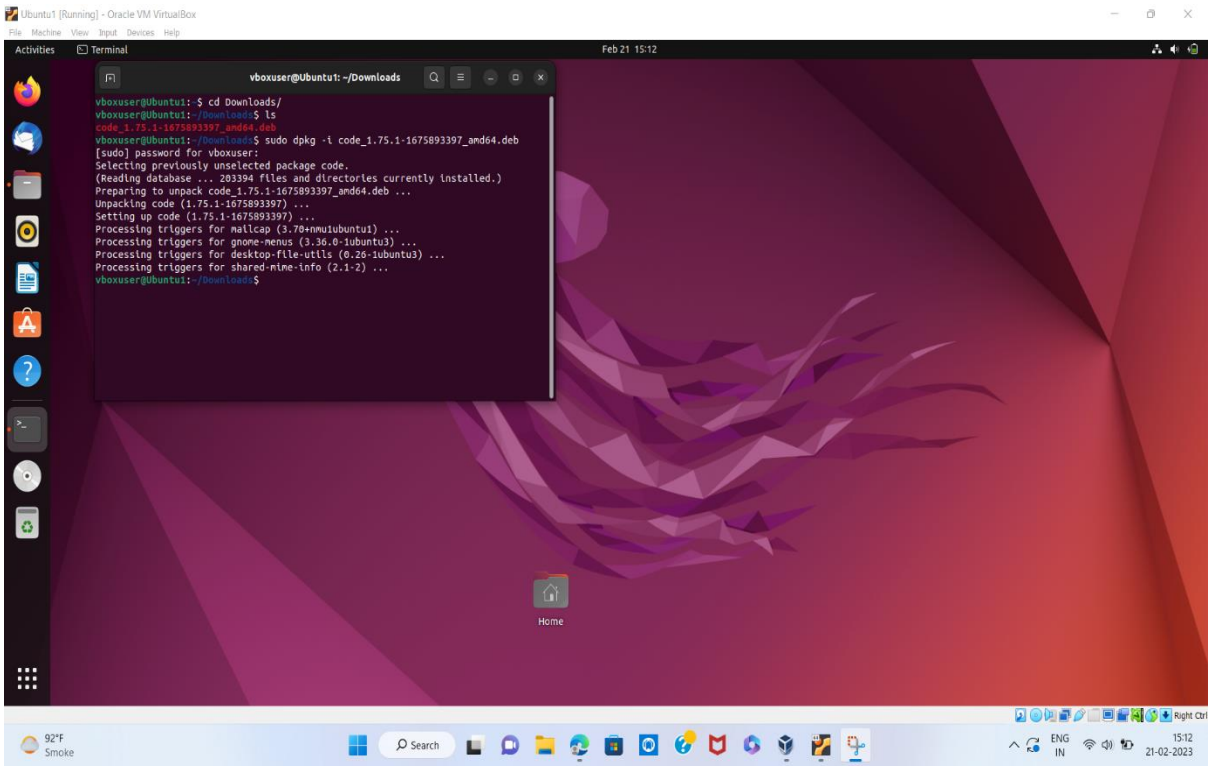
1.Host a Ubuntu Virtual Machine using Oracle VM Virtual Box.

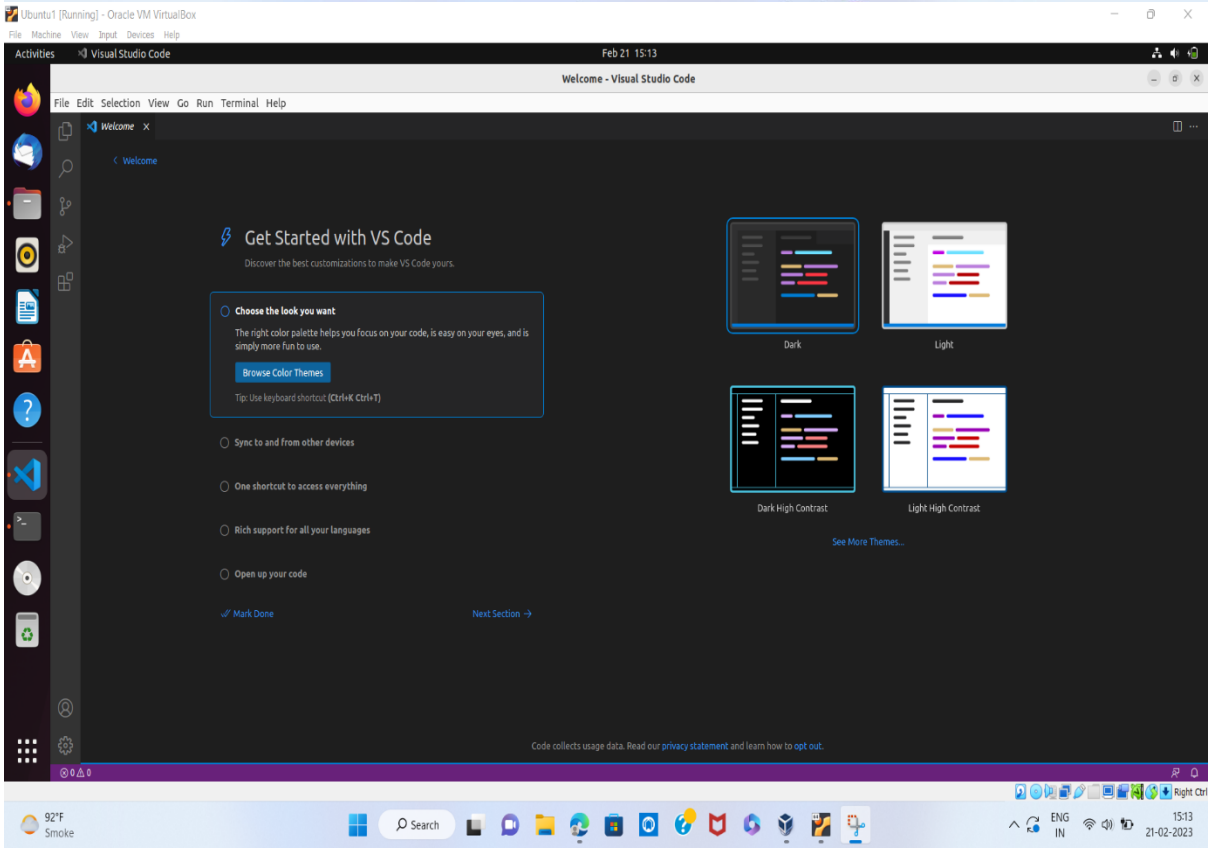


2.Set up Visual Studio code on Ubuntu VM.

Download a file “code-1.75.1-1675893397_amd64.deb”

Command : sudo dpkg -i code-1.75.1-1675893397_amd64.deb





3.Set up Python.

Command 1: sudo apt update

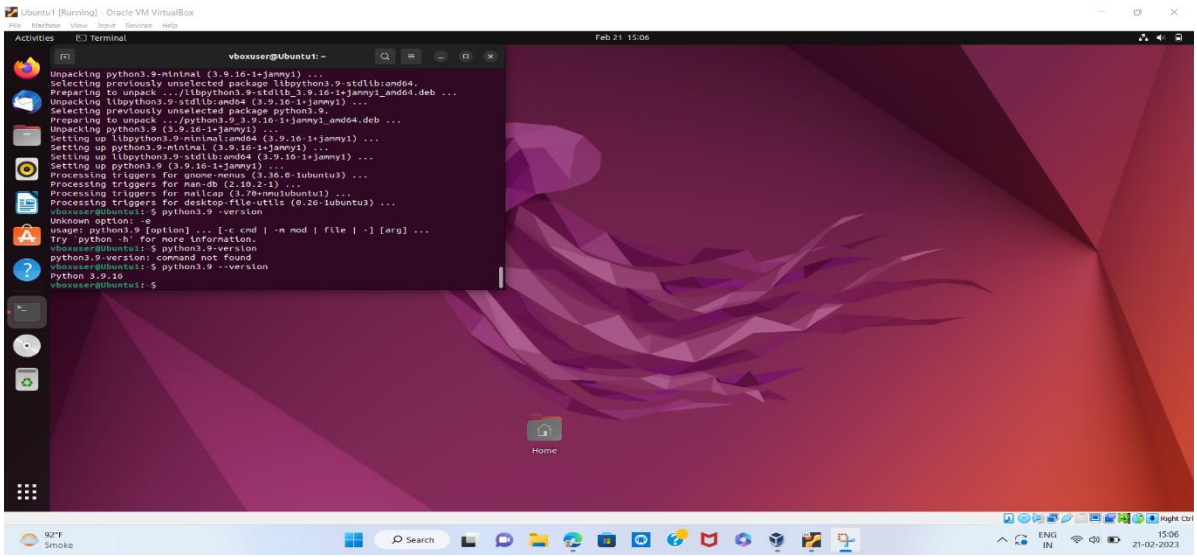
Command 2: sudo apt install software-properties-common

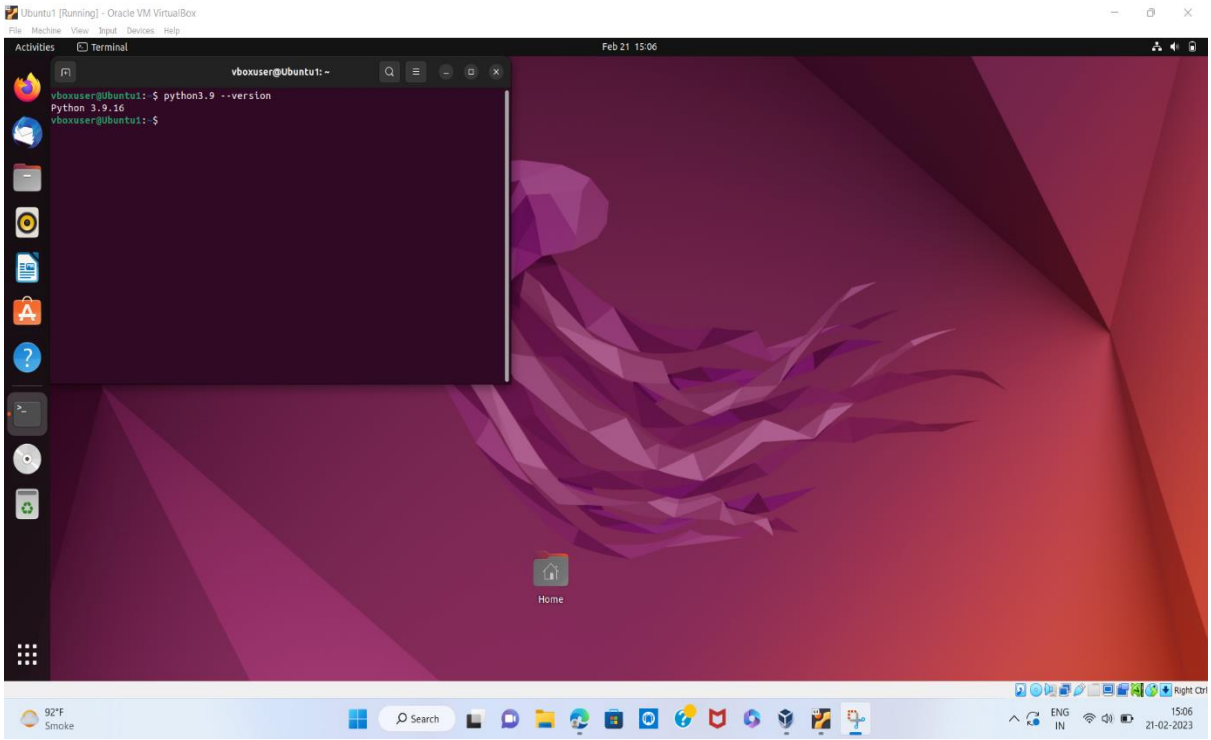
Command 3: sudo add-apt-repository ppa:deadsnakes/ppa

Command 4: sudo apt install python3.9

●Verification of the installation was successful.

Command: python3.9 –version



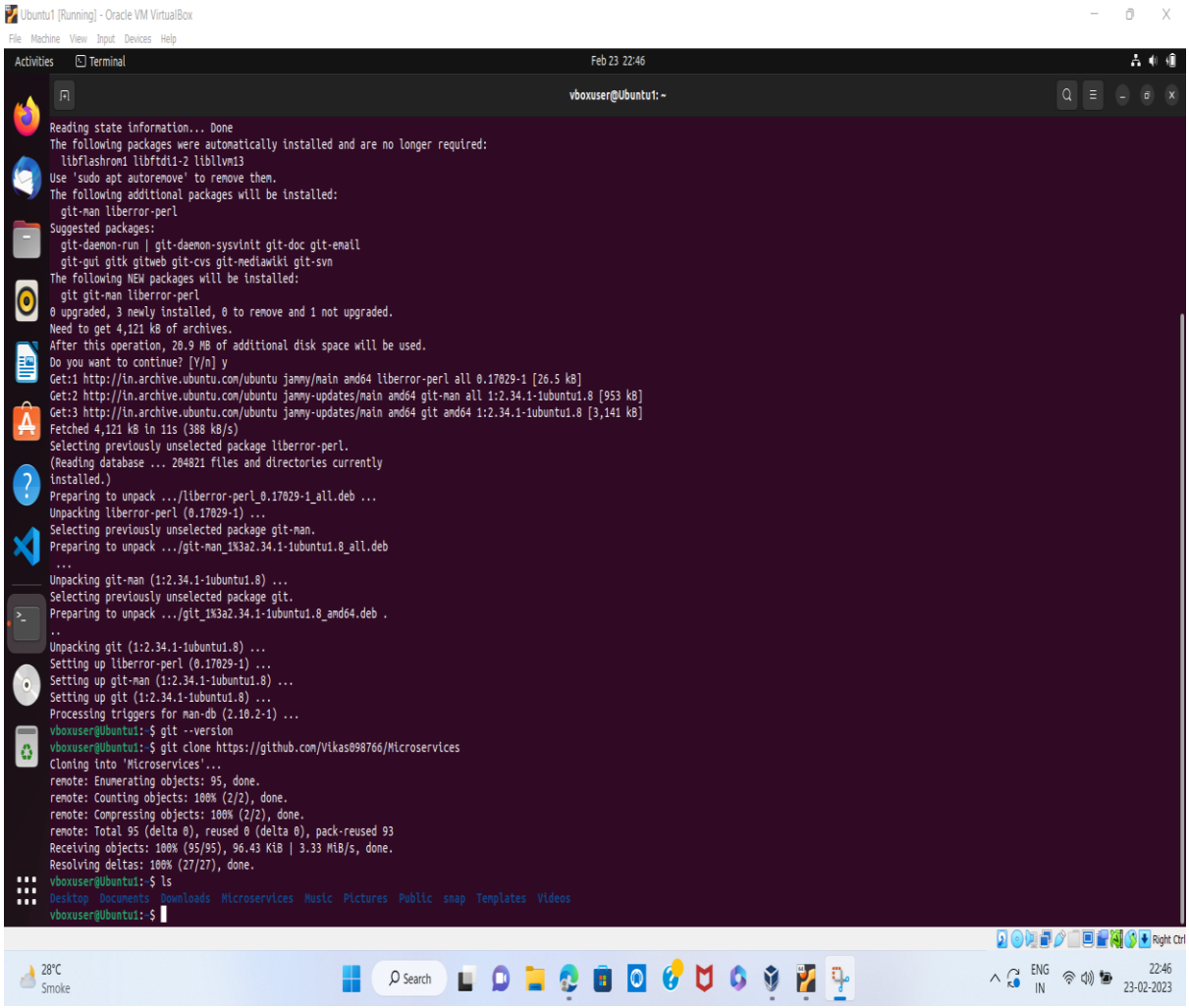


4.Clone this Github repository

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

Install git and then clone the specified repository

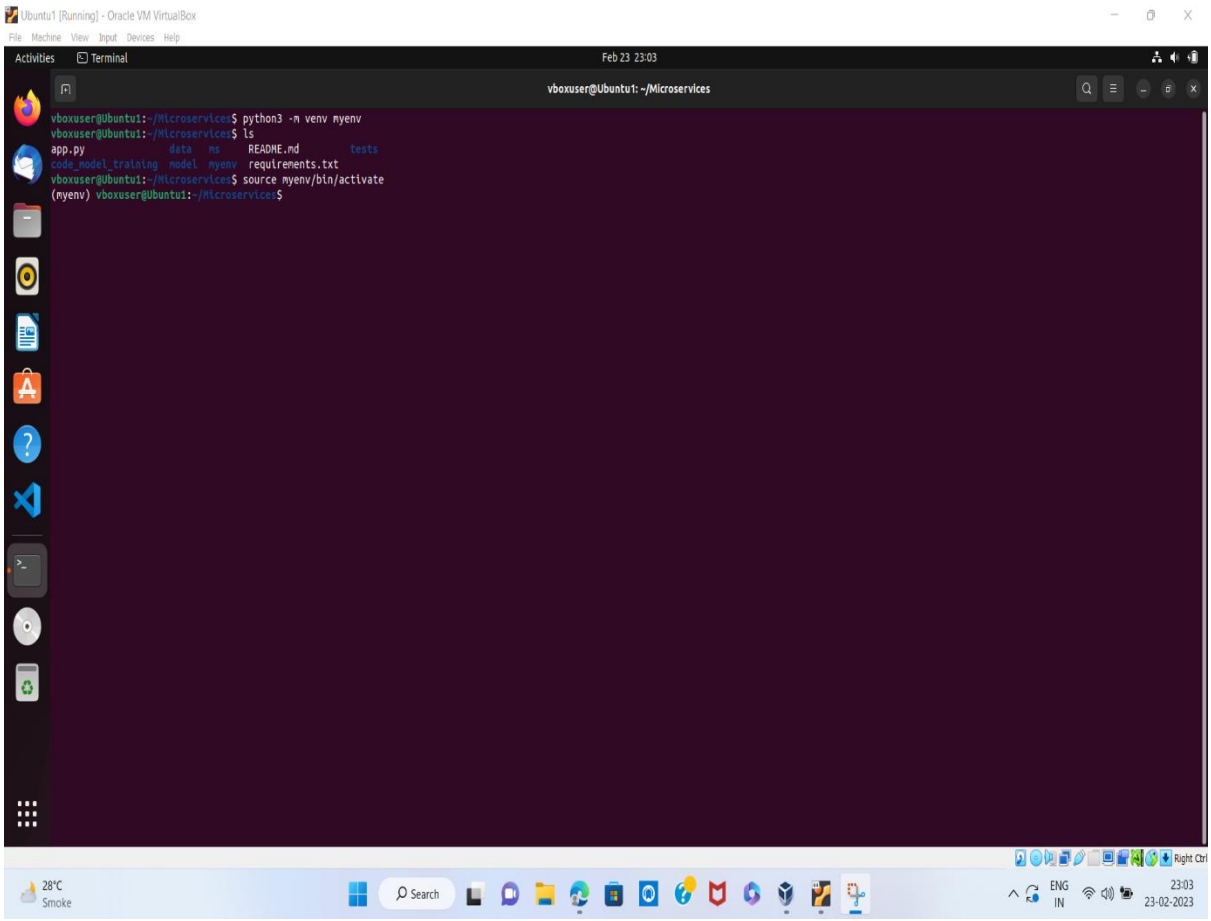
Command : git clone https://github.com/Vikas098766/Microservices.git



5.Create a Virtual Environment.

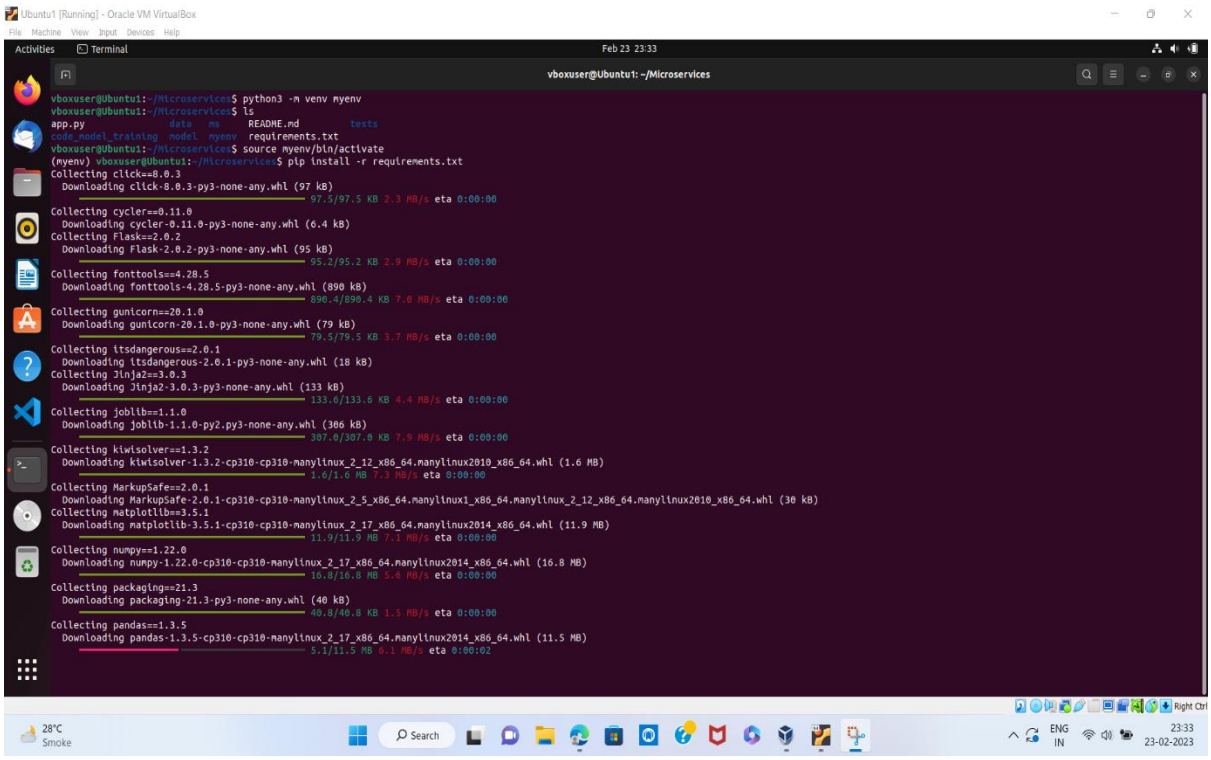
Command : python3 -m venv myenv

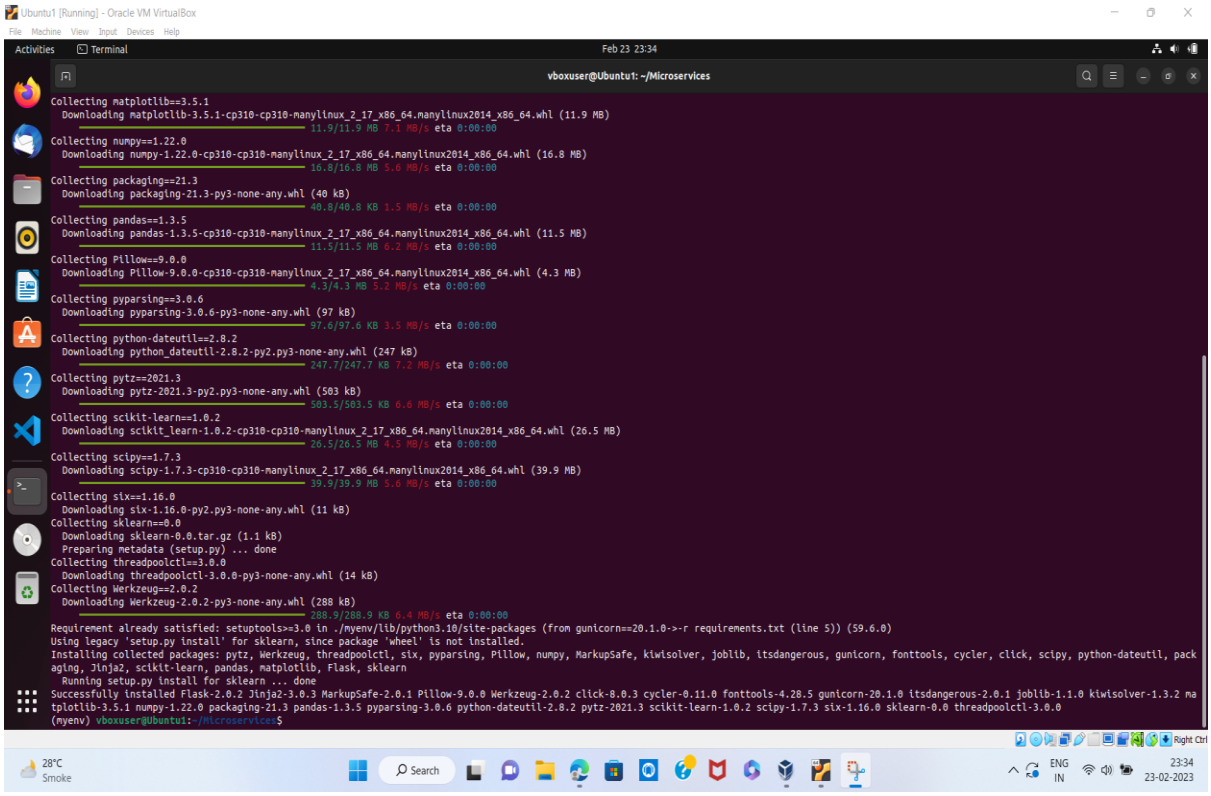
Command: source venv/bin/activate



6.Install the dependencies from requirements.txt file.

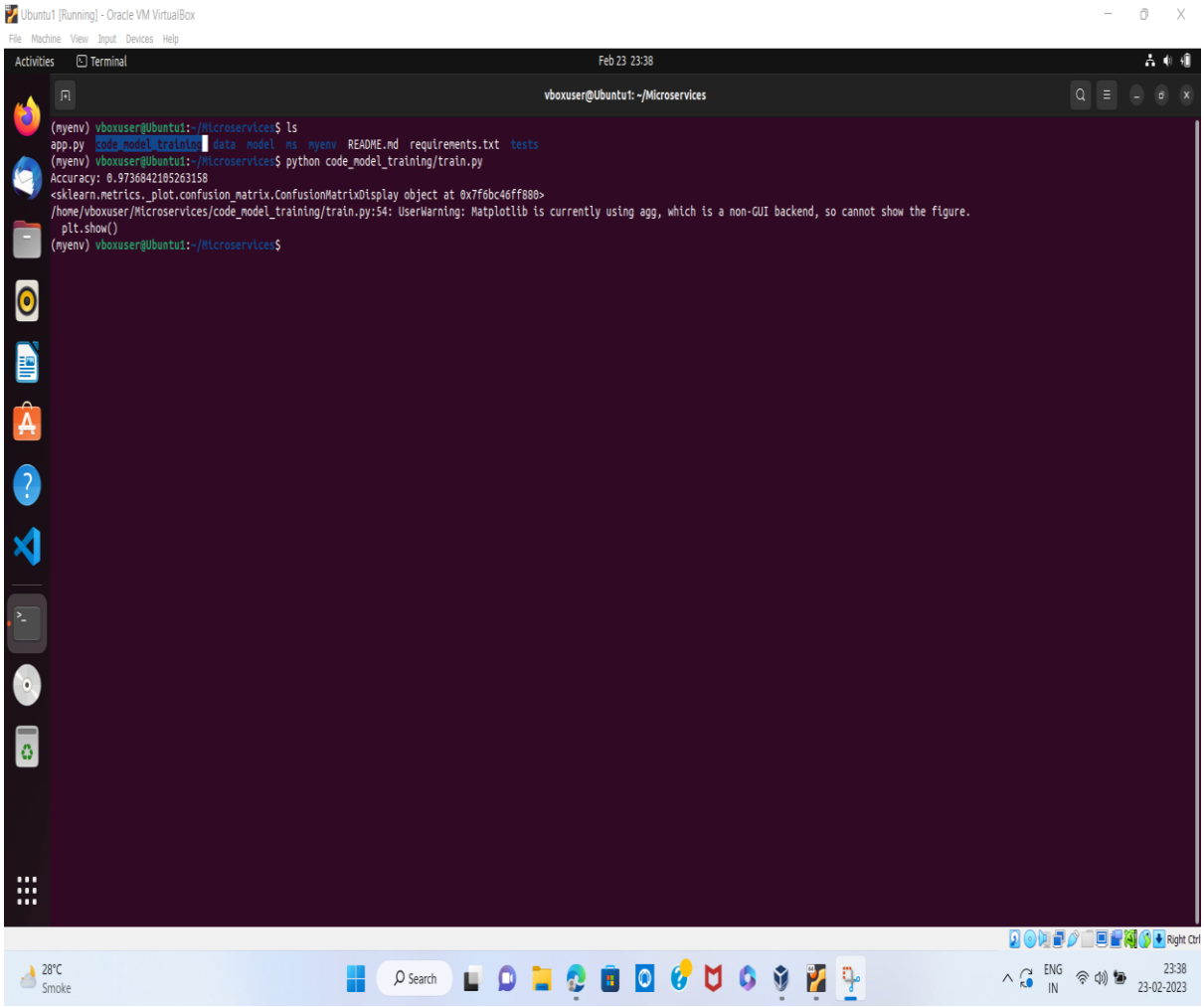
Command : pip install -r requirements.txt





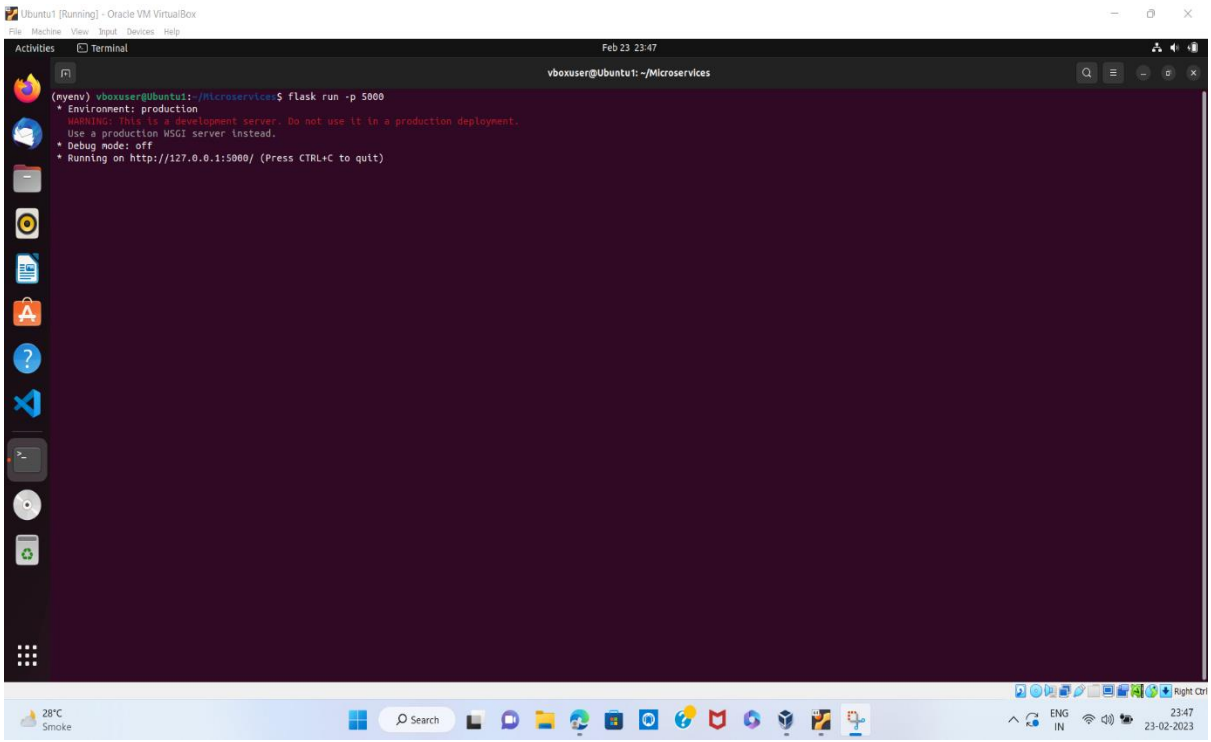
7.Train and save the model.

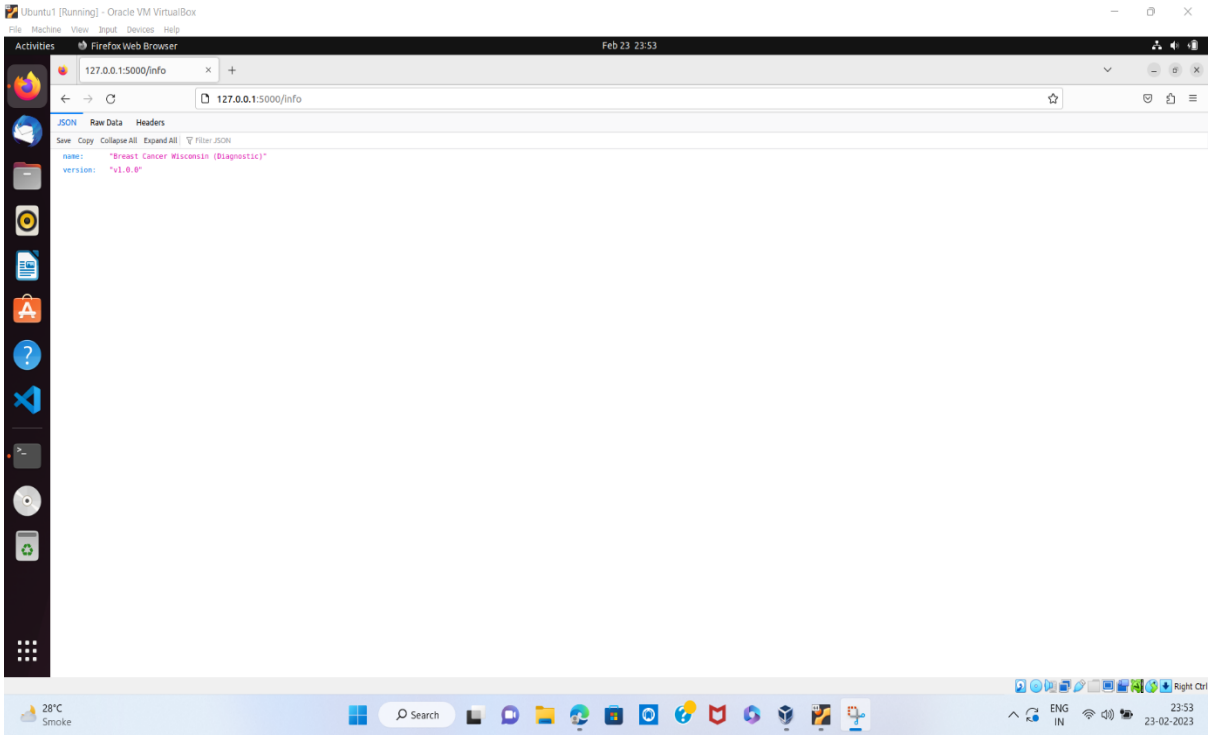
Command : python code_model_training/train.py

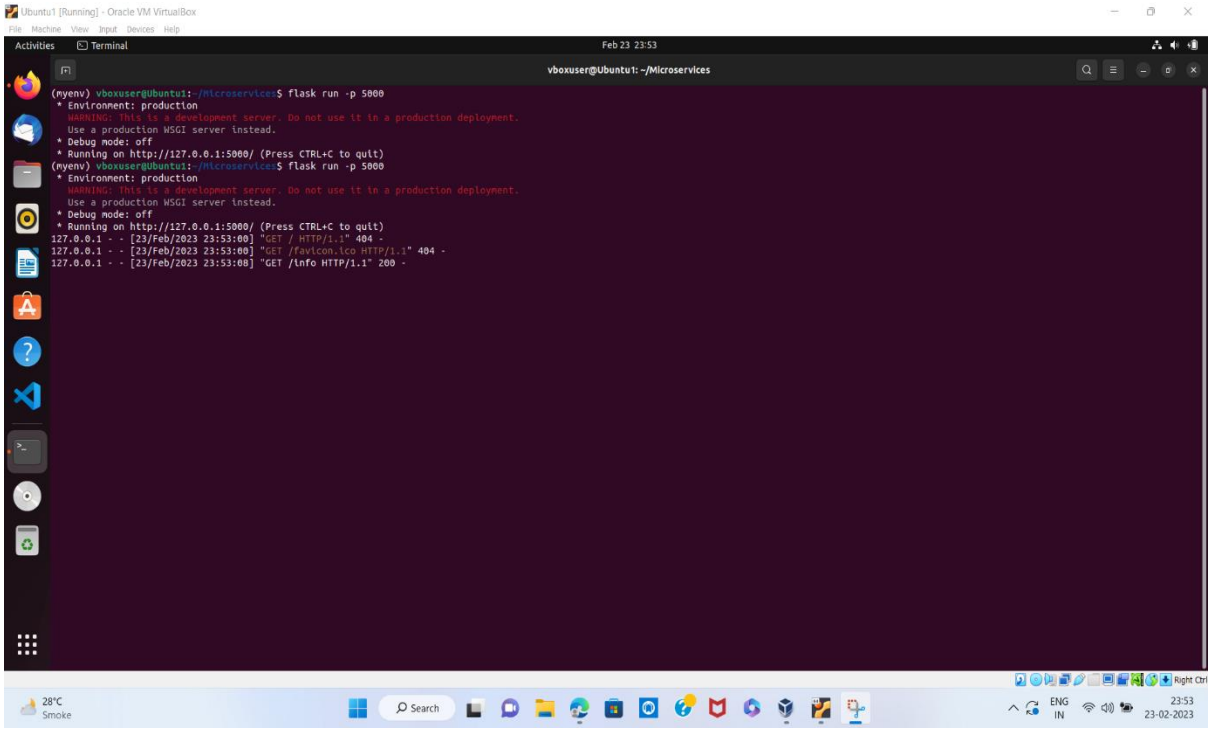


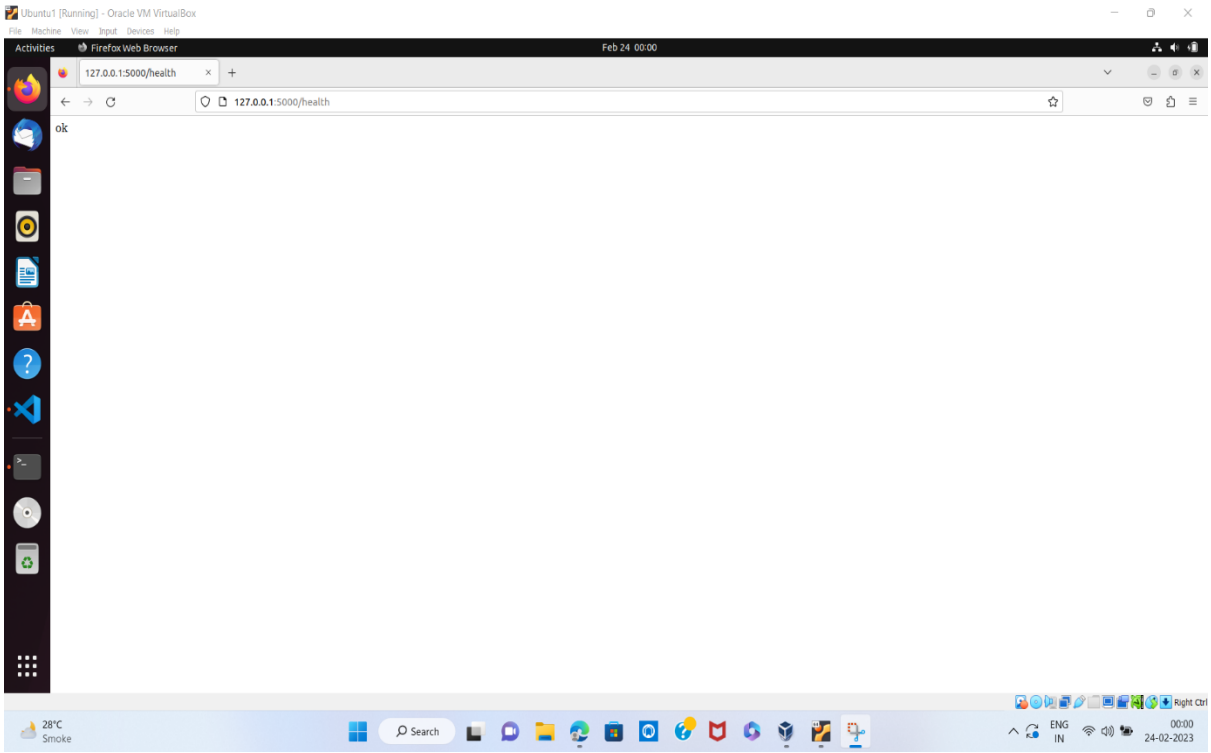
8.Test the Flask web application.

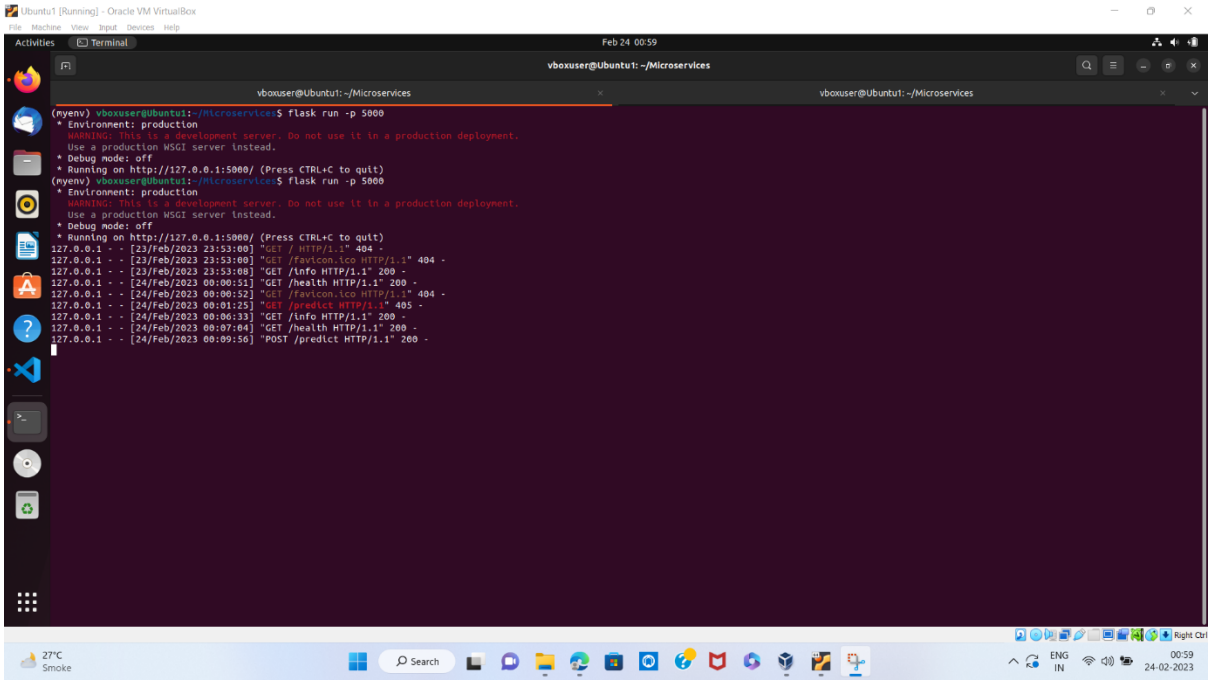
Command : flask run -p 5000






Ubuntu1 [Running] - Oracle VM VirtualBox
Activities
Terminal
Feb 23 23:53
vboxuser@Ubuntu1: ~/Microservices
(myenv) vboxuser@Ubuntu1:~/Microservices$ flask run -p 5000
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
(myenv) vboxuser@Ubuntu1:~/Microservices$ flask run -p 5000
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [23/Feb/2023 23:53:00] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [23/Feb/2023 23:53:00] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [23/Feb/2023 23:53:08] "GET /info HTTP/1.1" 200 -
28°C
Smoke
Search
ENG
IN
23:53
23-02-2023

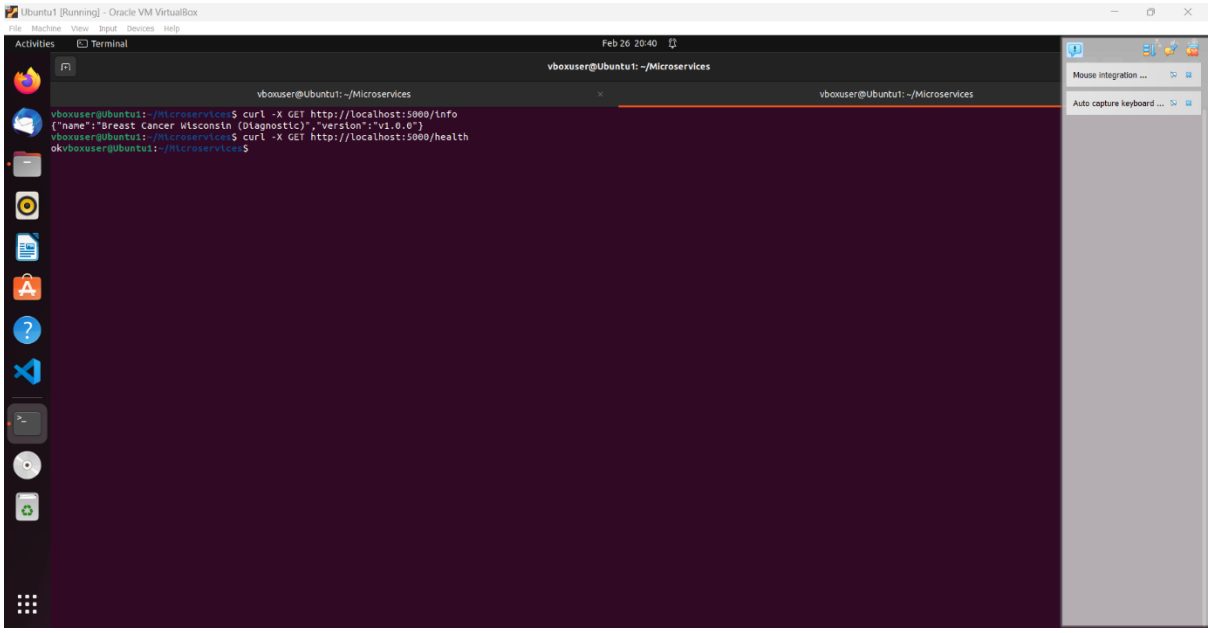

Ubuntu1 [Running] - Oracle VM VirtualBox
Activities
Firefox Web Browser
Feb 24 00:00
127.0.0.1:5000/health
127.0.0.1:5000/health
ok
28°C
Smoke
Search
ENG
IN
00:00
24-02-2023


Ubuntu1 [Running] - Oracle VM VirtualBox
Activities
Terminal
Feb 24 00:59
vboxuser@Ubuntu1: ~/Microservices
vboxuser@Ubuntu1: ~/Microservices
vboxuser@Ubuntu1: ~/Microservices
(myenv) vboxuser@Ubuntu1:~/Microservices$ flask run -p 5000
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
(myenv) vboxuser@Ubuntu1:~/Microservices$ flask run -p 5000
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [23/Feb/2023 23:53:00] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [23/Feb/2023 23:53:00] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [23/Feb/2023 23:53:08] "GET /info HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2023 00:00:55] "GET /health HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2023 00:00:52] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [24/Feb/2023 00:01:25] "GET /predict HTTP/1.1" 405 -
127.0.0.1 - - [24/Feb/2023 00:06:33] "GET /info HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2023 00:07:04] "GET /health HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2023 00:09:56] "POST /predict HTTP/1.1" 200 -
Right Ctrl
27°C
Smoke
Search
ENG
IN
00:59
24-02-2023

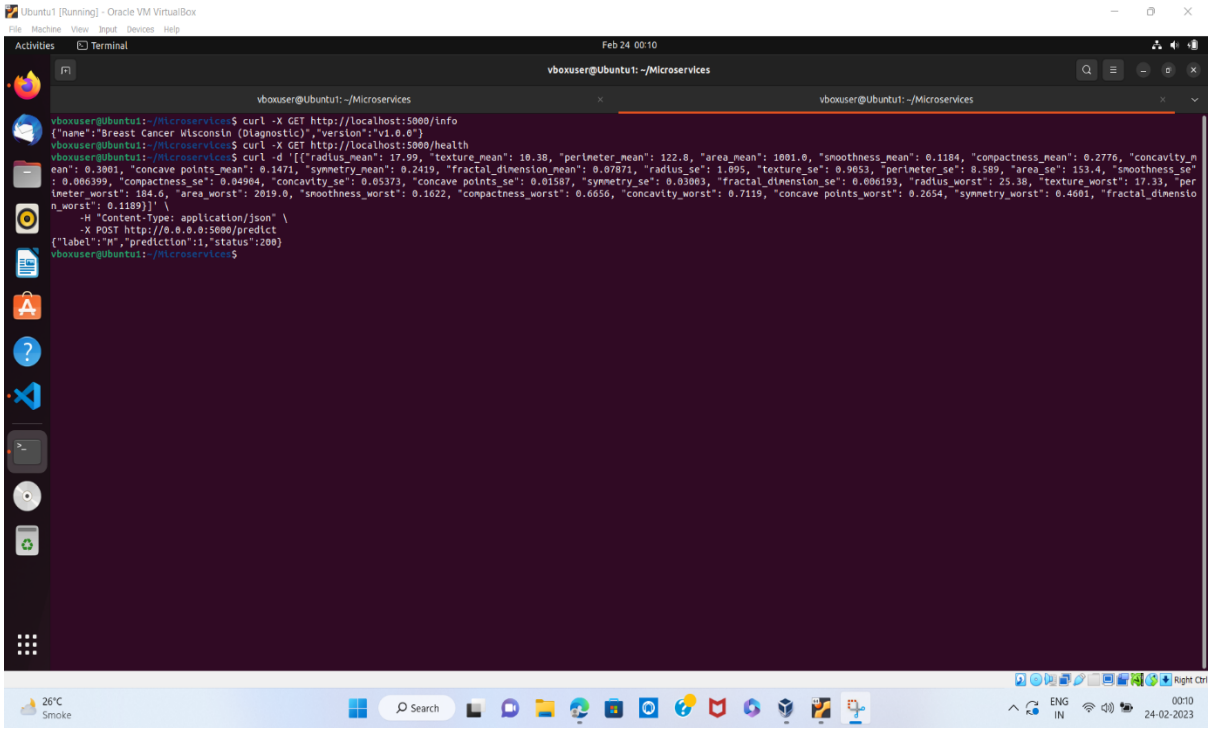
9.Test the application and make predictions using the example calls available in the folder/tests.

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

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

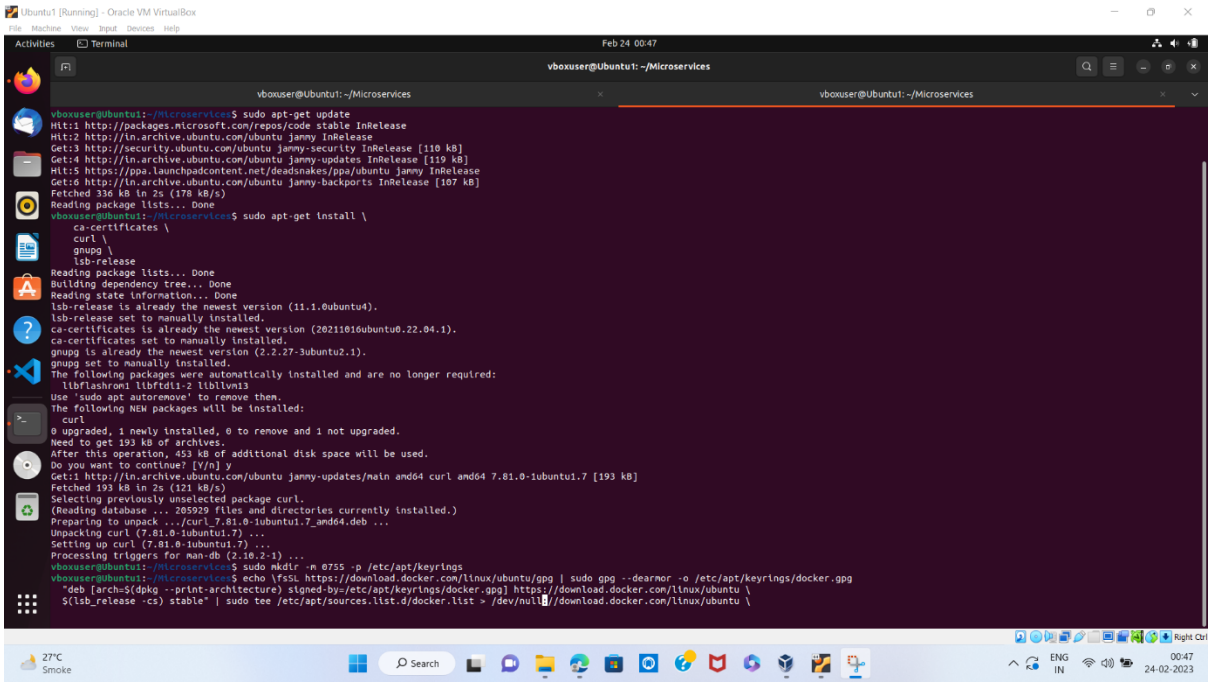


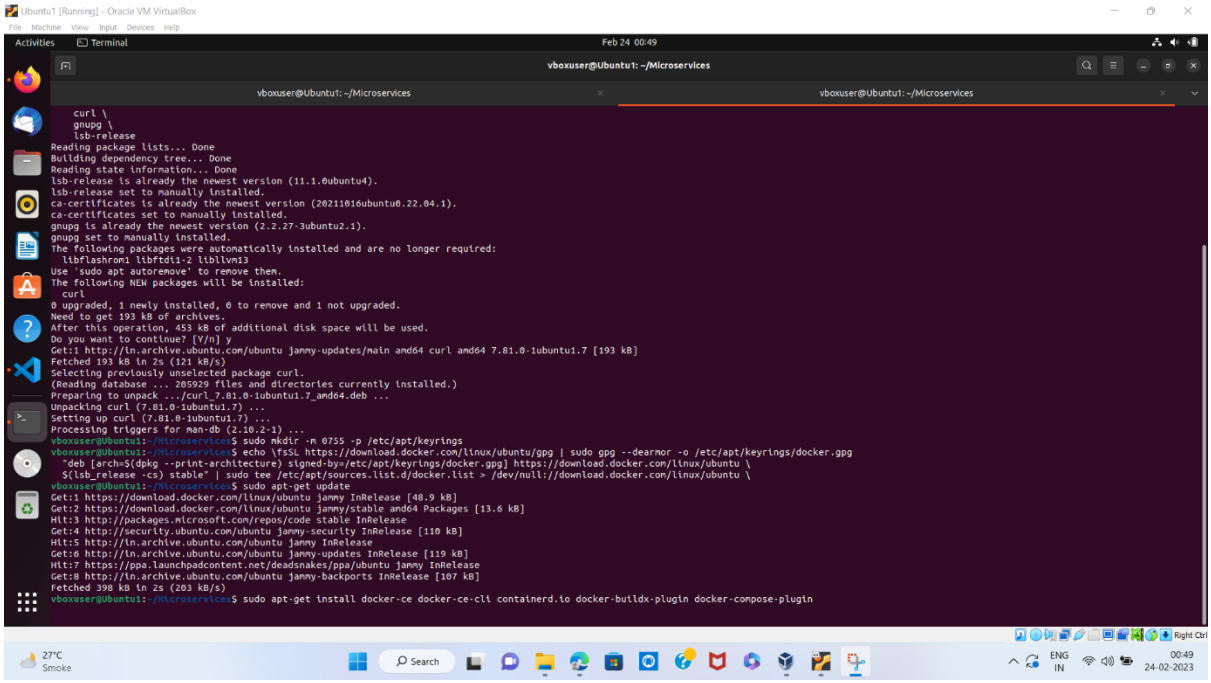
POST method predict :

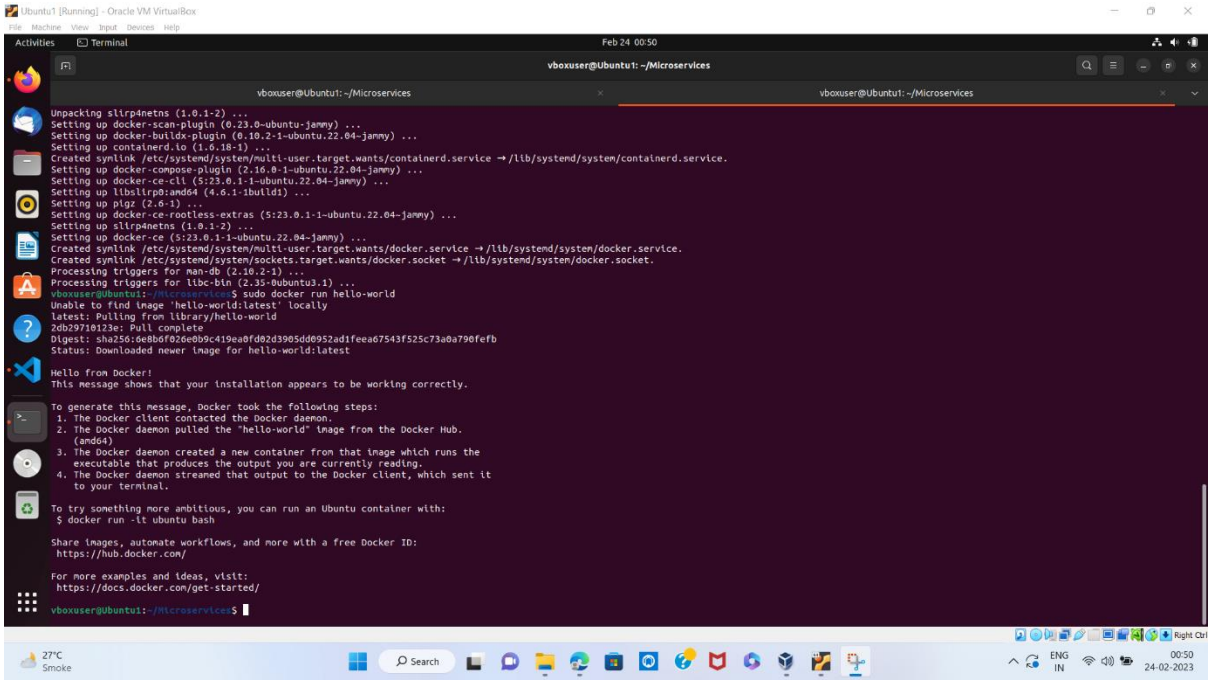


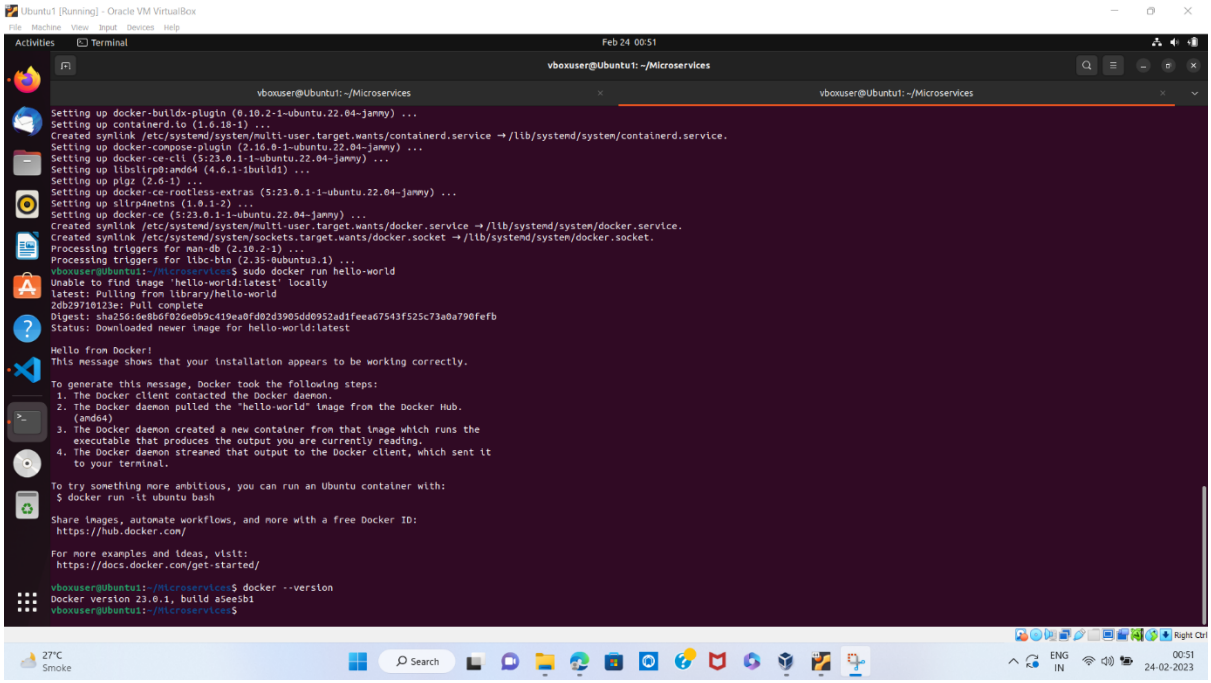
10.Create a docker image containing everything needed to run the application.

Docker Installation and creating a docker image



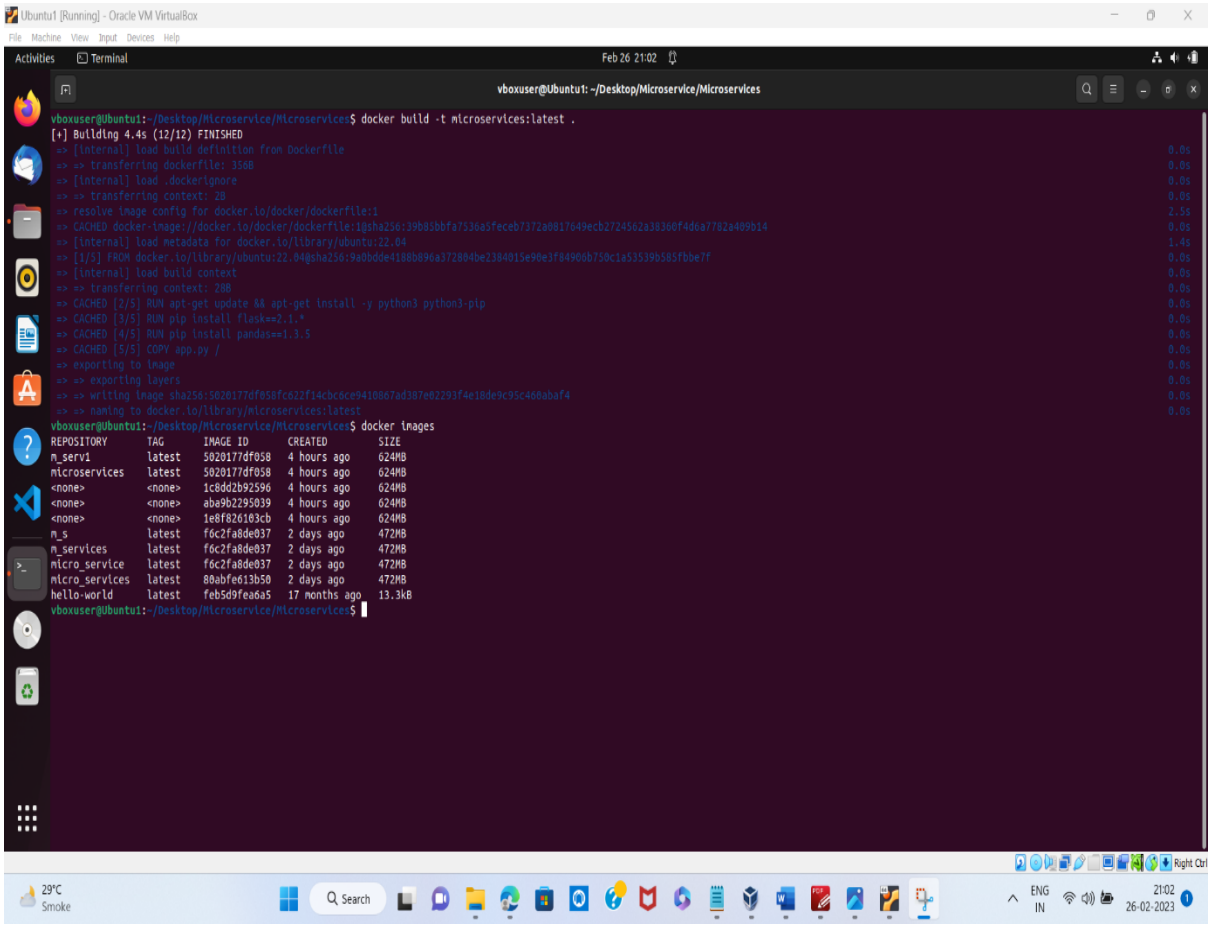



Ubuntu1 [Running] - Oracle VM VirtualBox
Activities
Terminal
Feb 24 00:50
vboxuser@Ubuntu1: ~/Microservices
vboxuser@Ubuntu1: ~/Microservices
vboxuser@Ubuntu1: ~/Microservices
Unpacking slirp4netns (1.0.1-2) ...
Setting up docker-scan-plugin (0.23.0~ubuntu-jammy) ...
Setting up docker-buildx-plugin (0.10.2-1~ubuntu.22.04~jammy) ...
Setting up containerd.io (1.6.18-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.16.0-1~ubuntu.22.04~jammy) ...
Setting up docker-ce-cli (5:23.0.1-1~ubuntu.22.04~jammy) ...
Setting up libslirp0:amd64 (4.6.1-1build1) ...
Setting up pigz (2.6-1) ...
Setting up docker-ce-rootless-extras (5:23.0.1-1~ubuntu.22.04~jammy) ...
Setting up slirp4netns (1.0.1-2) ...
Setting up docker-ce (5:23.0.1-1~ubuntu.22.04~jammy) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
vboxuser@Ubuntu1:~/Microservices$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:6e8b6f026e0b9c419ea0fd02d3905dd0952ad1feea67543f525c73a0a790fefb
Status: Downloaded newer image for hello-world:latest
Hello from Docker!
This message shows that your installation appears to be working correctly.
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the
executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
to your terminal.
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
For more examples and ideas, visit:
https://docs.docker.com/get-started/
vboxuser@Ubuntu1:~/Microservices$
Right Ctrl
27°C
Smoke
Search
ENG
IN
00:50
24-02-2023


Ubuntu1 [Running] - Oracle VM VirtualBox
Activities
Terminal
Feb 24 00:51
vboxuser@Ubuntu1: ~/Microservices
vboxuser@Ubuntu1: ~/Microservices
vboxuser@Ubuntu1: ~/Microservices
Setting up docker-buildx-plugin (0.10.2-1~ubuntu.22.04~jammy) ...
Setting up containerd.io (1.6.18-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.16.0-1~ubuntu.22.04~jammy) ...
Setting up docker-ce-cli (5:23.0.1-1~ubuntu.22.04~jammy) ...
Setting up libslirp0:amd64 (4.6.1-1build1) ...
Setting up pigz (2.6-1) ...
Setting up docker-ce-rootless-extras (5:23.0.1-1~ubuntu.22.04~jammy) ...
Setting up slirp4netns (1.0.1-2) ...
Setting up docker-ce (5:23.0.1-1~ubuntu.22.04~jammy) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
vboxuser@Ubuntu1:~/Microservices$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:6e8b6f026e0b9c419ea0fd02d3905dd0952ad1feea67543f525c73a0a790fefb
Status: Downloaded newer image for hello-world:latest
Hello from Docker!
This message shows that your installation appears to be working correctly.
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the
executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
to your terminal.
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
For more examples and ideas, visit:
https://docs.docker.com/get-started/
vboxuser@Ubuntu1:~/Microservices$ docker --version
Docker version 23.0.1, build a5ee5b1
vboxuser@Ubuntu1:~/Microservices$
Right Ctrl
27°C
Smoke
Search
ENG
IN
00:51
24-02-2023

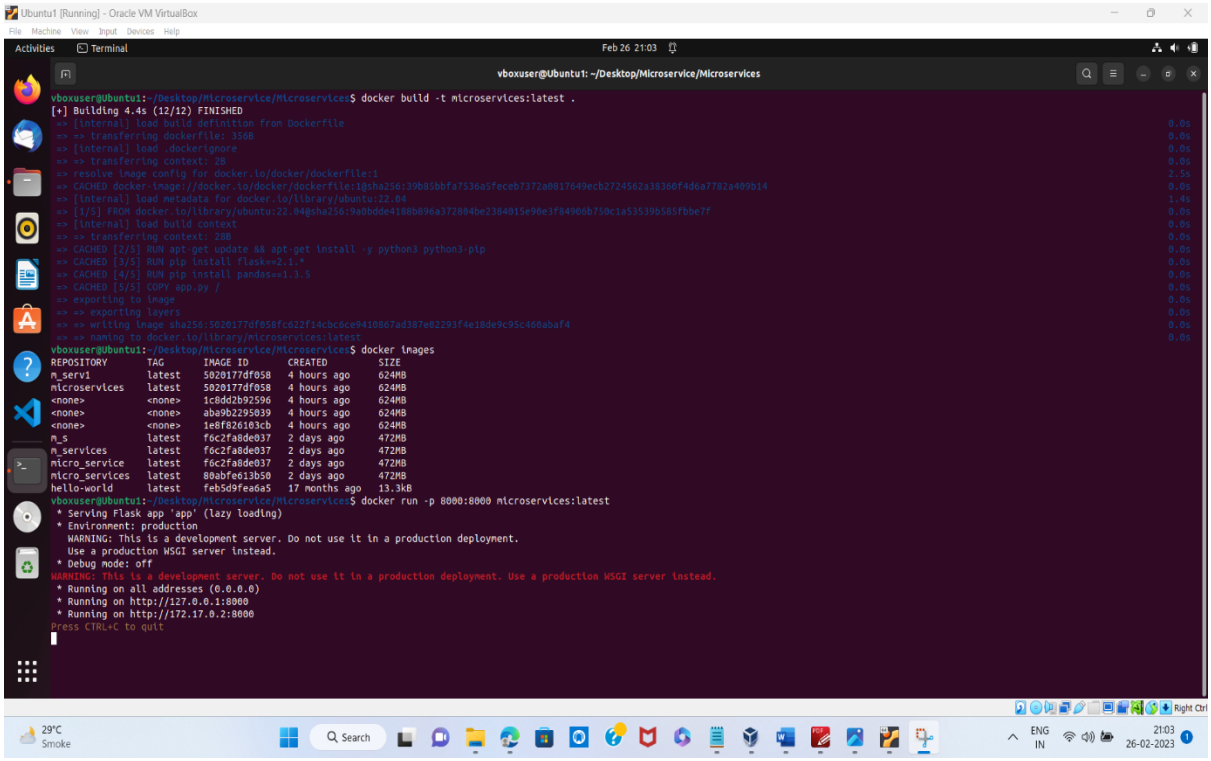
Creating a docker image

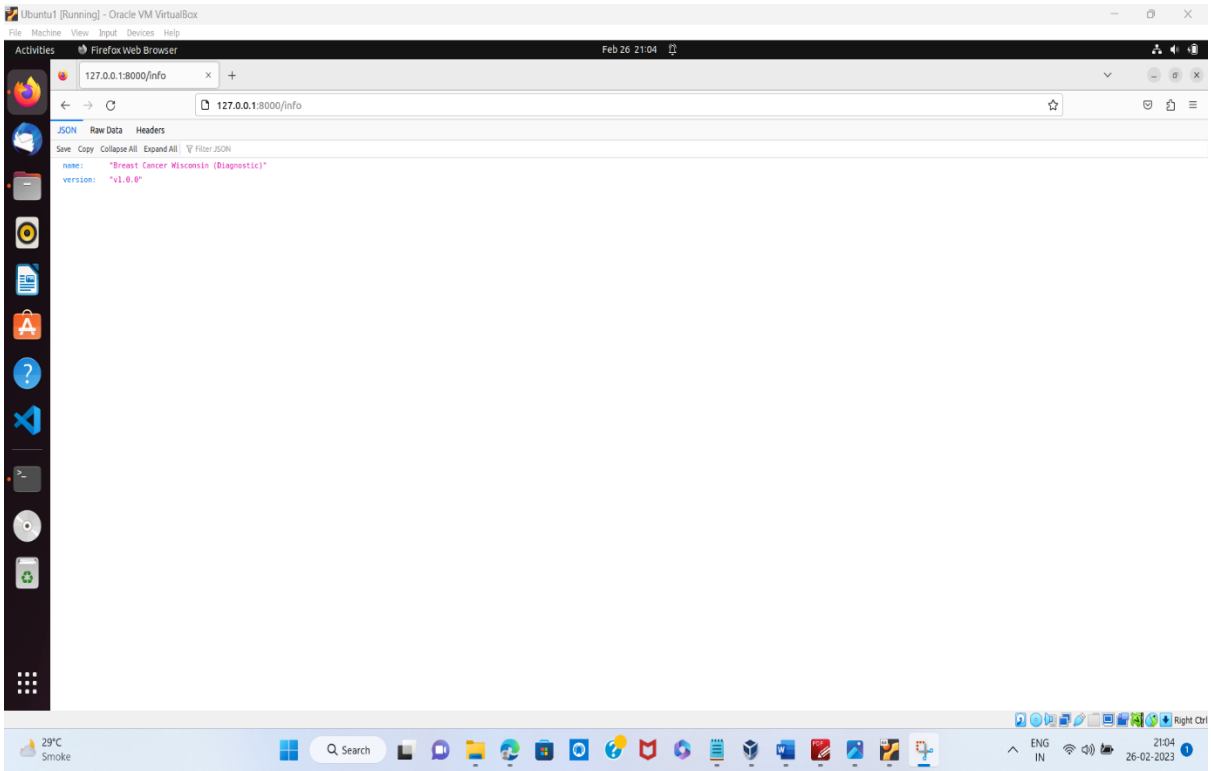
Command : docker build -t microservices:latest .

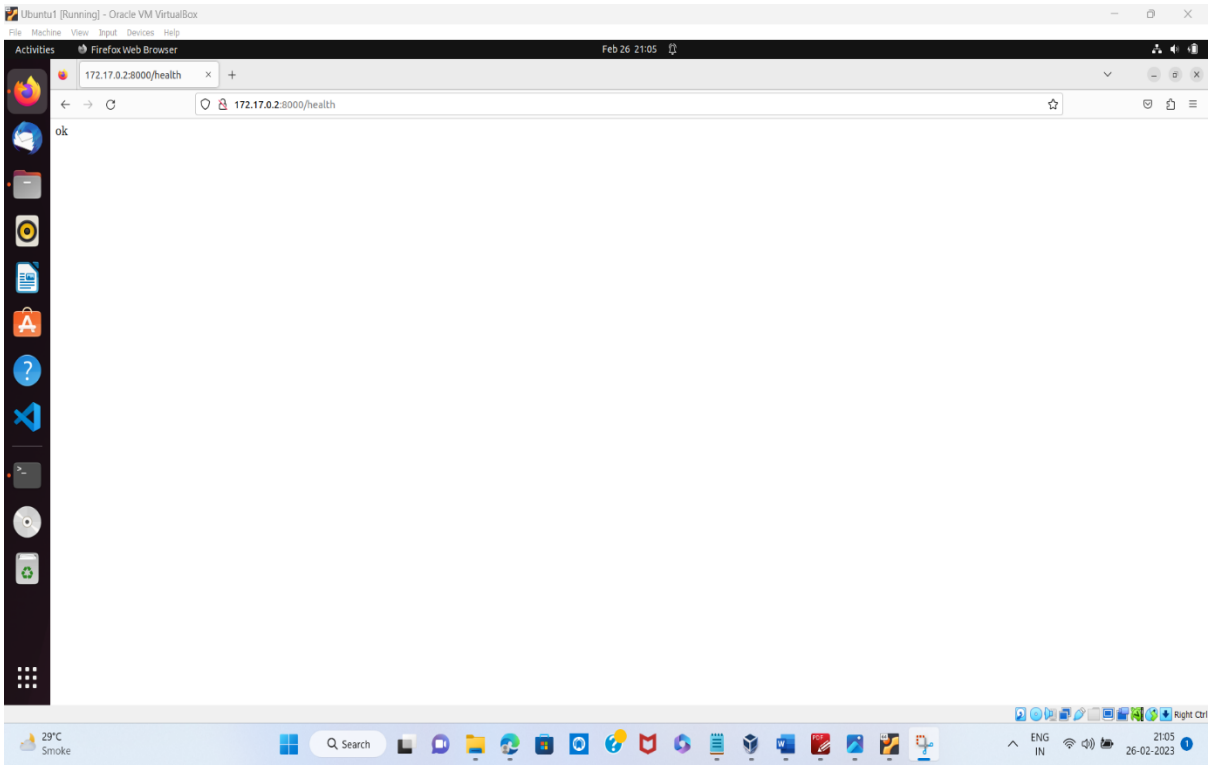


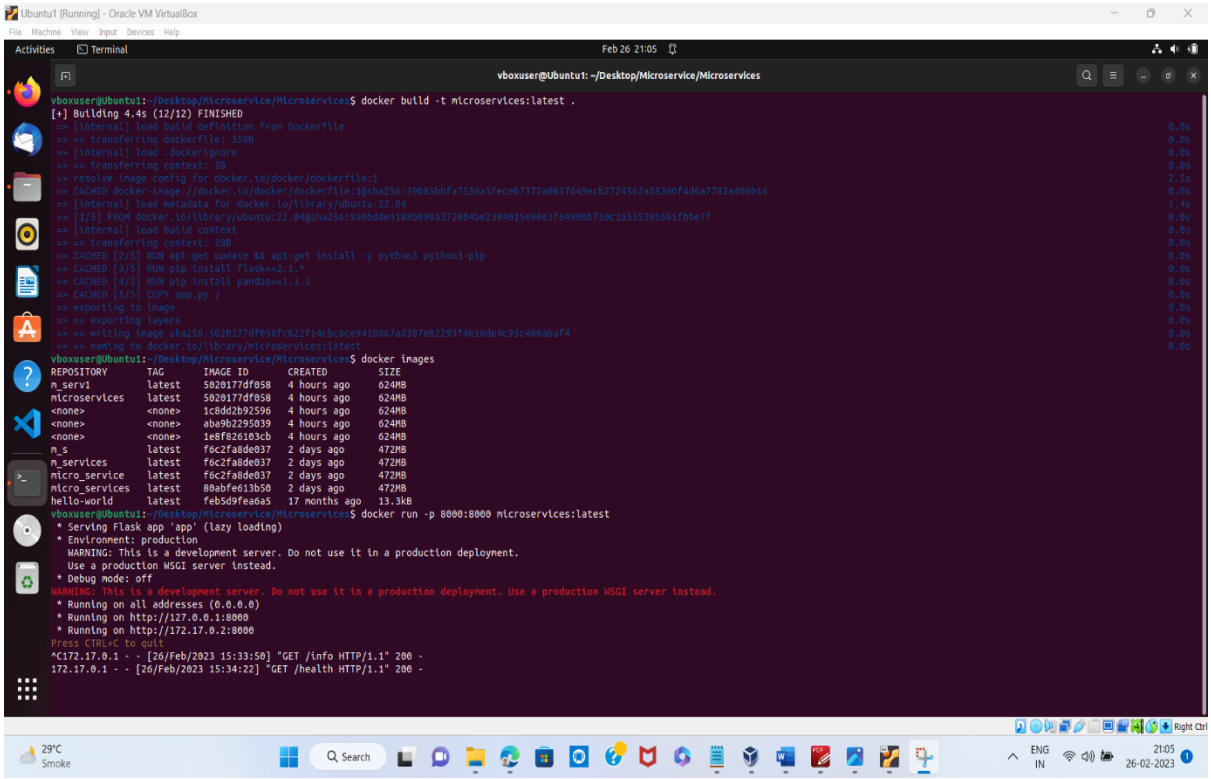
11.Run the containerized application as a prediction service and test it locally by passing some example calls and get the prediction.

Command : docker run -p 8000:8000 microservices:latest






Ubuntu1 [Running] - Oracle VM VirtualBox
Activities
Firefox Web Browser
Feb 26 21:05
172.17.0.2:8000/health
ok
29°C
Smoke
Search
ENG
IN
21:05
26-02-2023


Ubuntu1 [Running] - Oracle VM VirtualBox
Activities
Terminal
Feb 26 21:05
vboxuser@Ubuntu1: ~/Desktop/Microservice/Microservices
vboxuser@Ubuntu1:~/Desktop/Microservice/Microservices$ docker build -t microservices:latest .
[+] Building 4.4s (12/12) FINISHED
vboxuser@Ubuntu1:~/Desktop/Microservice/Microservices$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
m_serv1 latest 5020177df058 4 hours ago 624MB
microservices latest 5020177df058 4 hours ago 624MB
<none> <none> 1c8dd2b92596 4 hours ago 624MB
<none> <none> aba9b2295039 4 hours ago 624MB
<none> <none> 1e8f826103cb 4 hours ago 624MB
m_s latest f6c2fa8de037 2 days ago 472MB
m_services latest f6c2fa8de037 2 days ago 472MB
micro_service latest f6c2fa8de037 2 days ago 472MB
micro_services latest 80abfe613b50 2 days ago 472MB
hello-world latest feb5d9fea6a5 17 months ago 13.3kB
vboxuser@Ubuntu1:~/Desktop/Microservice/Microservices$ docker run -p 8000:8000 microservices:latest
* Serving Flask app 'app' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8000
* Running on http://172.17.0.2:8000
Press CTRL+C to quit
^C172.17.0.1 - - [26/Feb/2023 15:33:50] "GET /info HTTP/1.1" 200 -
172.17.0.1 - - [26/Feb/2023 15:34:22] "GET /health HTTP/1.1" 200 -
29°C
Smoke
Search
ENG
IN
21:05
26-02-2023