=============Git/ Git Hub===============

Git status

1. Untracked files ---------🡪 (newly created file in working area)
2. Change to be commited -------🡪 (files are in staging area)
3. Changes not staged to be commited ----🡪(files working area having modification)
4. Nothing to commit ---🡪 (All files in commited)

------------------------------------------------------------------------------------

Git diff -> [shows difference between files present in working to staging]

cd myproject/

git status

nano file.html

git add file.html

git status

git diff 🡪 (difference seen b/w working and staging area)

git difftool

ls

qa! 🡺 quit the Vim

history

git diff --staged =🡺 (diff b/w staging to commited)

git difftool --staged 🡺 (visual representation)

history

girt diff head 🡺[ d/w commited and working]

git difftool head

gitignore 🡺 if we want some of the file to be ignore to commit or avoid the files to be commit or going to staging area

or change something

nano .gitignore

ls

git status

git add .

git status

git commit -m "ignore commit 1" 🡺 [ignore commit]

git push origin master

To remove origin and add origin master

Git remote

Git remote remove origin

Git remote

To add origin master

Git remote add origin https://

Git pull origin master

Ls

19-04-2024

1. Git init
2. Git config -- global user.name “Rutu”
3. Git config -- global user.email “emailId”
4. Git config – list
5. Git add .
6. Git commit -m “message”
7. Git remote
8. Git remote add origin “github-repo-link”
9. Git remote remove origin
10. Git pull origin master
11. Git push origin master ( -- f – force )
12. Git status
13. Git diff
14. Git difftool b/w working 🡨> staying



1. Git diff --staged b/w staging <-> staging



1. Git difftool --staged
2. Git diff HEAD b/w working <-> commited



1. Git difftool HEAD
2. Gitb show 🡺 [latest commited they will show]

checkout master

ls

git checkout testing

git push origin testing

22-04-2024

git tag 🡺 shows of available tags

1. Lightweighted

Git tag v1.0

Git tag -a v1.1 <commit id>

DevOps

Virtualization Containerization Bare Metal

* Sharing list sharing OS No sharig
* Medium cost low cost highest cost
* medium secure Less secure highly secure

Development Testing Deployment

Q. 1] How much container we can create in docker ?

-We can create container It depend upon the ram , CPU

2] The size of container may be same or vary

- It container may be different in size

- the minimum size is 250

Sudo su

Apt-get update -y

Docker install

Docker apt install docker.io

Start🡺 docker service docker start

* docker service docker status

nginx search 🡺

To see list of all containers

* docker ps -a
* docker container ls -a

we want to run docker in background

docker run -d image/id/image-name =🡺[go into containers]

To go inside container

* docker exec -it container-id/container-name /bin/bash 🡺

[I – interactive t – terminal]

Container image – we want to extract the name or id of image

* docker commit container-id/container-name image-name

we cannot autoscale in docker

30-04-2024

DOCKER NETWORK

1. Bridge n/w 🡺 communication establish
2. Host n/w 🡺 not secure
3. Overlay n/w

1. Docker network ls
2. Docker network create network\_name –subnet 172.20.0.0/16
3. Docker run -d -it –name webserver -p80:80 –network network\_name nginx /bin/bash
4. Docker network inspect network\_name/id

Docker run -d -it –name con1 –cpu 02 –memory 516 -p80:80 –network mynetwork nginx

DOCKER VOLUMES

Bind mout

Bind volumes –

Named volumes –

03-05-2024

DOCKER ATTRIBUTE

FROM 🡺 To pull base image from dockerhub

RUN 🡺 To execute command

WORKMKDIR 🡺 To specify working directory

MAINTAINER 🡺 To specify author or owner file

EXPOSE 🡺 To open specific port

ENV 🡺 To set environment variables

COPY 🡺 To copy files from host to images/container

CMD 🡺 Execute command during container creation

ADD 🡺 Copies files from host to container , downloads files from specific URL & extract .tar.gz or zip files

ENTRYPOINT 🡺 Similar to CMD, but having higher priority. Also allows additional.

ARG 🡺 Defines variables that passed to container.

LABLE 🡺 To add metadata

USER 🡺 To set user

HEATHCHECK 🡺 To specify path for healthcheck

SHELL 🡺To specify shell to be used to run command

STOPSIGNAL 🡺 specify the signal to stop container gracefully

VOLUME 🡺 To create volume

ON BUILD 🡺 Specify instruction to be used when we use this as base image for other image.

|  |  |  |
| --- | --- | --- |
| Sr. No | ATTRIBUTE | DESCRIPTION |
|  | FROM |  |
|  | RUN |  |
|  | WORKMKDIR |  |
|  | MAINTAINER |  |
|  | EXPOSE |  |
|  | ENV |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

06-05-2024

For nginx

Nano Dockerfile

FROM nginx

WORKDIR /usr/share/nginx/html

RUN touch home.html

RUN echo “My home page “ > home.html

EXPOSE 80

CMD [“nginx” ,”-9”,”daemon off;”]

myDockerfile

docker build -t mynginx .

docker build -t mynginx -f mydockerfile

mynginx

docker run -d -p80:80 –name mynginxcont mynginx

MYSQL container create

MYSql 🡺 Dockerfile

MAINTAINER Rutu

FROM mysql

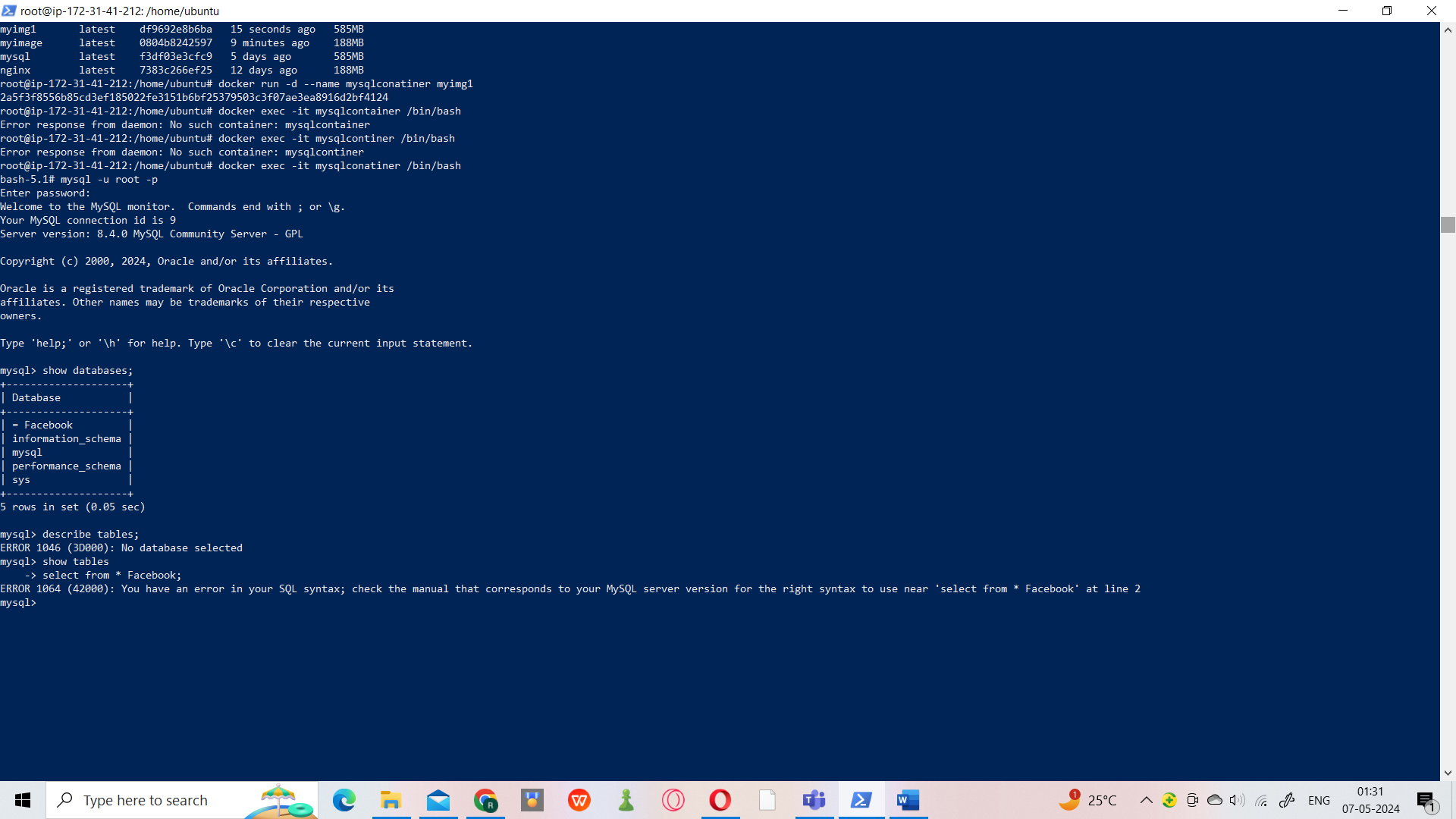
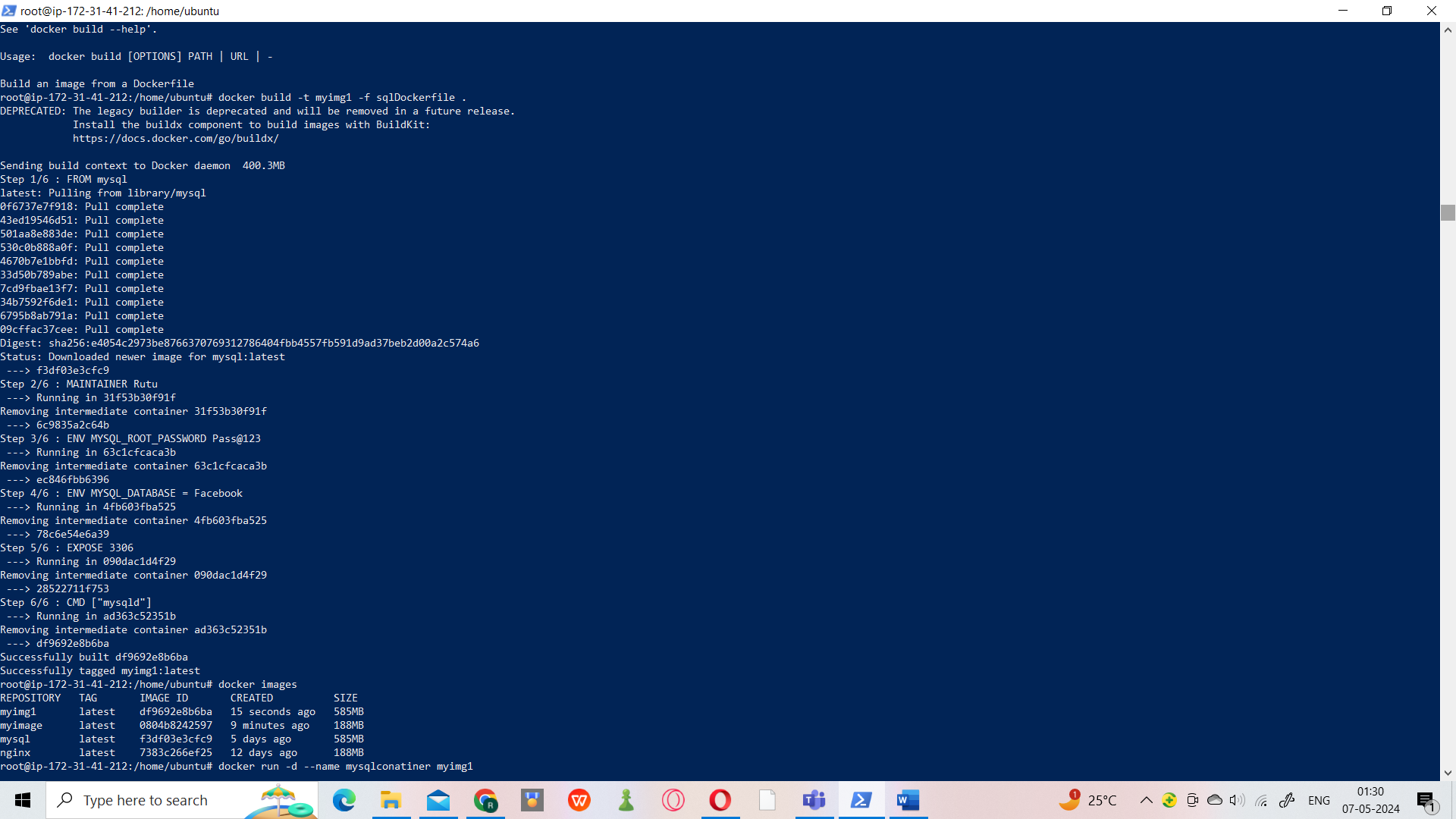
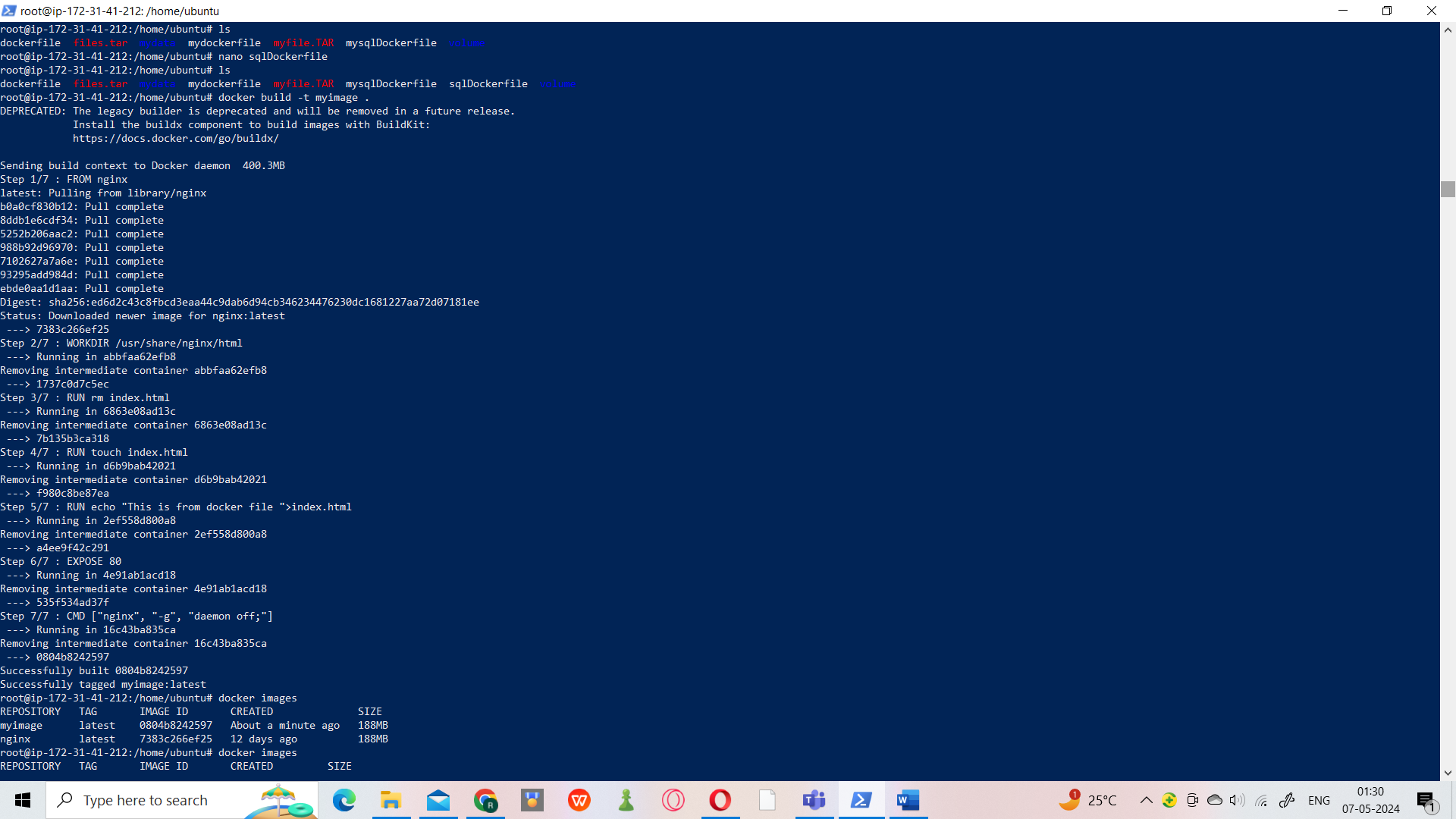
4ENV MYSQL\_ROOT\_PASSWORD Pass@123

ENV MYSQL\_DATABASE Facebook

EXPOSE 3306

CMD [“mysqld”]

Docker build -t mysqlimg



Wordpress 🡺 dockerfile

FROM wordpress

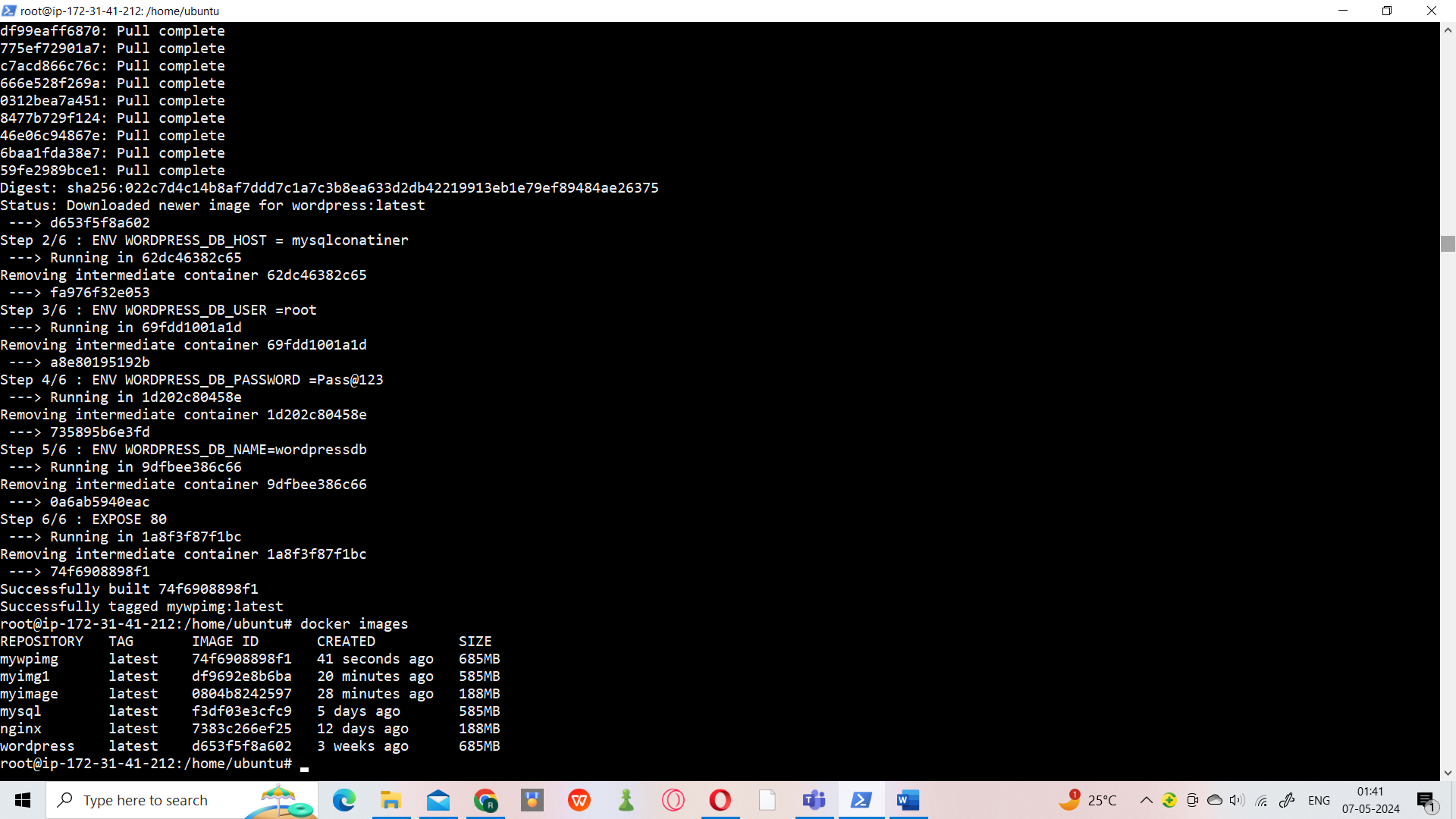
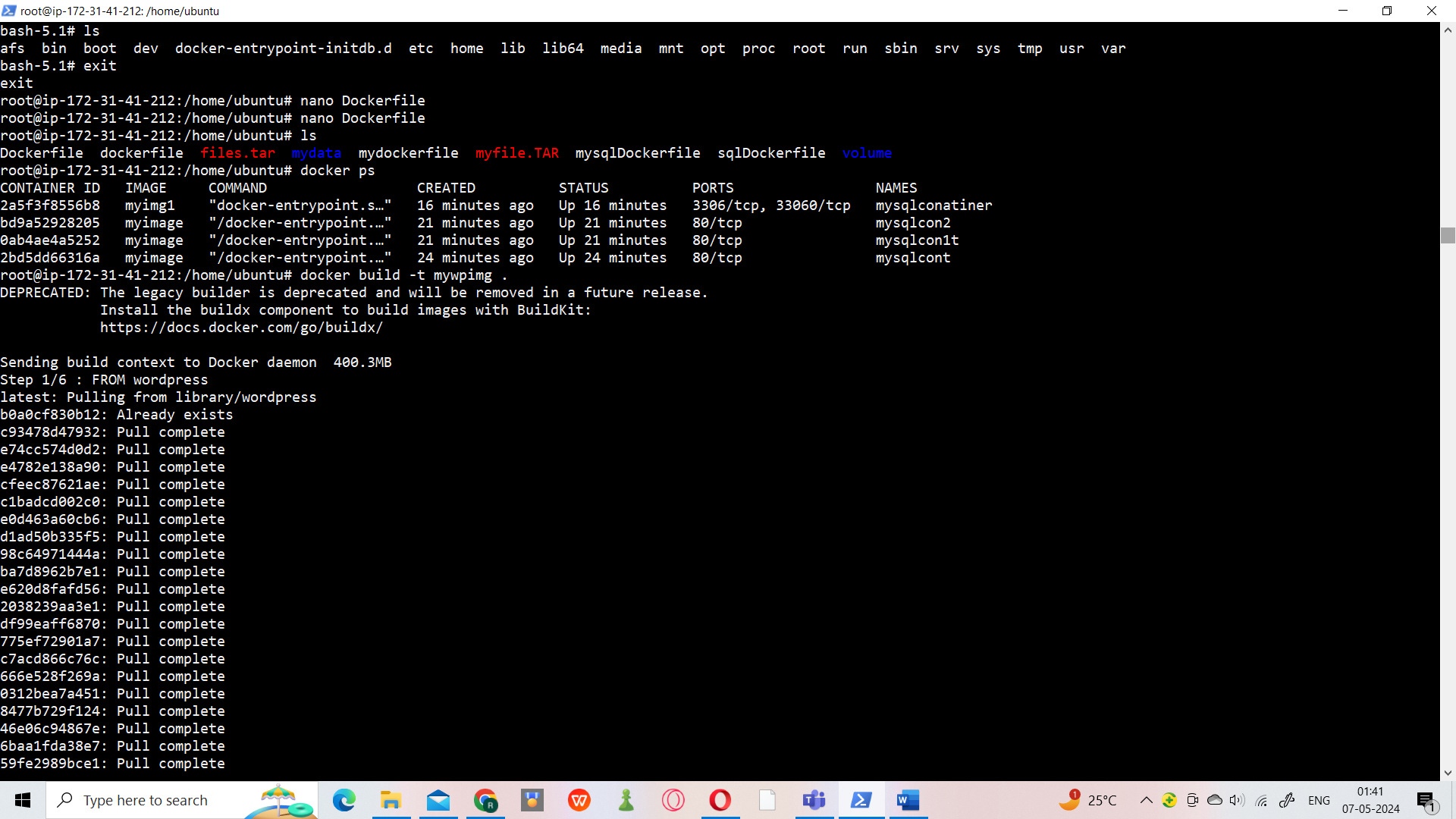
ENV WORDPRESS\_DB\_HOST mysqlcont

ENV WORDPRESS\_DB\_PASSWORD Pass@123

ENV WORDPRESS\_DB\_USER =root

ENV WORDPRESSS\_DB\_NAME = Mwordpressdb

EXPOSE 80

****

Myfile.sh

#! /bin/bash

Mysql -u root -p Pass@123 << eof

Create databse insta;

Use insta;

Create table user (id int , name varchar (100));

Insert into user (101, “cloud”);

EOF

**/docker-entrytpoint-initabd**

**/usr/local/bin**

**Dockerfile**

**FROM mysql**

**ENV MYSQL\_ROOT\_PASSWORD =Pass@123**

**COPY myfile /docker-entrytpoint-initabd/file.sh**

**RUN chmod +x docker-entrytpoint-initabd/file.sh**

**EXPOSE 3306**

**CMD [“mysqld”]**

07-05-2024

Python

* It is programming language
* Websites
* Apps
* Stand alone apps

2 framework

1] Django

2] Flask

i.) requirement.txt

ii.) app.py/server.py/index.py

08-05-2024

Docker composer

Container create

1] docker run -d nginx

2] docker 🡺 build

3] Docker compose 🡺 .yml file is there than docker-compose up-d is up and container is created.

Docker-compose –f-name up -d

Services

DB :

Image :mysql

Environment

MYSQL\_ROOT\_PASSWORD

MYSQL\_DATABASE

Ports :

* 3806
* Network : backend
* Volumes :
* Mydata:/var/lib/mysql

VOLUMES

1. Named volumes / Docker volumes

-Doker create myvolume

|

Mysql, nginx same path /var/lib/docker/volumes/myvolume

1. Bind volumes / host volumes

Name and host volume communicate b/w them , container to host, host to container.

1. Ananomous volumes

Docker run -d /home/ubuntu/mydata: /usr/share/nginx/html

-detach

-pull

Docker-compose

snap install docker

apt  install docker-compose -y

1. Mkdir compose
2. Cd compose/
3. Nano docker-compose.yml

sudo curl -L

"https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

docker file use to create image

FROM=base image UBUNTU

RUN apt-get install nginx -y

WORKDIR /USR/SHARE/NGINX/HTML

RUN TOUCH HOME.HTML

RUN ECHO"HELLO" > HOME.HTML

EXPORT PORT 80

CMD ["NGINX","-g","daemon off;"]

nginx use debian os

dockerfile atributes

FROM = TO PULL BASE IMAGE FROM DOCKERHUB

RUN = TO EXECUTE COMMAND

WORKDIR = TO SPECIFY WORKING DIRECTORY

MAINTAINER = TO SPECIFY AUTHER OR OWNER OF FILE

EXPOSE = TO OPEN SPECIFIC PORT

ENV = TO SET ENVIRONMENT VARIABLES

COPY = TO COPY FILES FROM HOST TO IMAGES

CMD = EXECUTE COMMAND DURING CONTAINER CREATION

ADD = COPIES FILES FROM HOST TO CONTAINER,DOWNLOADS FILES FROM SPECIFIC URL & EXTRACTS .TAR.GZ OR ZIP FILES

ENTRYPOINT = SIMILAR TO CMD, BUT HAVING HIGHER PRIORITY. ALSO ALLOWS ADDITIONAL ARGUMENTS TO PASS

ARG = DEFINES VARIABLES THAT PASSED TO CONTAINER

LABLE = TO ADD METADATA

USER = TO SET USER

HEALTHCHECK = TO SPECIFY PATH FOR HEALTHCHECK

SHELL = TO SPECIFY SHELL TO BE USED TO RUN COMMAND

STOPSIGNAL = SPECIFIES THE SIGNAL TO STOP CONTAINER GRACEFULLY

VOLUME = TO CREATE VOLUME

ONBUILD = SPECIFIES INSTRUCTION TO BE USED WHEN WE USE THIS AS BASE IMAGE FOR OTHER IMAGE IT IS ALSO CALLED AS MULTIBUILD

dockerfile:- python

framework : 1) Django 2) flask: pip

from flask import Flask

app = Flask(name)@app.route('/')def hello geek():return '<h1>Hello from Flask & Docker</h2>'if \_\_name == "main":    app.run(debug=True)

# syntax=docker/dockerfile:1

FROM python:3.8-slim-buster

WORKDIR /python-docker

COPY requirements.txt requirements.txt

RUN pip3 install -r

requirements.txt

COPY . .

CMD [ "python3", "-m" , "flask",

"run", "--host=0.0.0.0"]

docker compose

container creation

1) docker run -d nginx

2) dokerfile-build

3) docker-compose- .yml

services:

db:

image:mysql

environment:

docker-compose

1  apt update

    2  apt install [docker.io](http://docker.io/)

    3  apt install docker-compose -y

    4  ls

    5  nano dockercompose.yml

    6  docker-compose -f dockercompose.yml up -d

    7  sudo curl -L "<https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname> -s)-$(uname -m)" -o /usr/local/bin/docker-compose

    8  sudo chmod +x /usr/local/bin/docker-compose

    9  sudo systemctl restart docker

   10  python3 -m venv path/to/venv

   11  source path/to/venv/bin/activate

   12  pip install requests==2.25.1

   13  sudo apt install python3.12-venv

   14  python3 -m venv path/to/venv

   15  source path/to/venv/bin/activate

   16  pip install requests==2.25.1

   17  docker-compose -f dockercompose.yml up -d

   18  history