TASK 1 :

1. Install Docker:

You can install Docker by following the official documentation for your operating system. Here are the general steps:

- For Linux, run the following commands in the terminal:

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io

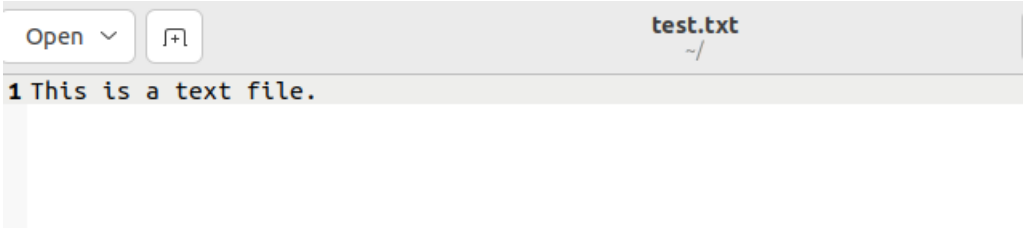
- For macOS, download Docker Desktop from the official website and install it.
- For Windows, download Docker Desktop from the official website and install it.

2. Create Ubuntu container:

Once Docker is installed, you can create a Ubuntu container by running the following command in the terminal:

```
subhashis@subhashis-virtual-machine:~$ sudo docker run -it ubuntu
[sudo] password for subhashis:
root@301eda5b7189:/# sudo docker ps
bash: sudo: command not found
root@301eda5b7189:/# docker cp test.txt CONTAINER_ID:/test.txt
bash: docker: command not found
root@301eda5b7189:/#
```

3. Create a test file in your local system:



4. Copy the test file from your local system to the Ubuntu container:

To copy the test file from your local system to the Ubuntu container, you can use the docker cp command. First, find the container ID by running the following command:

Docker ps

This will list all the running containers with their IDs. Note down the container ID of the Ubuntu container.

Next, copy the test file to the container using the following command:

```
subhashis@subhashis-virtual-machine:~$ docker cp test.txt CONTAINER_ID:/test.txt
Successfully copied 0B to CONTAINER_ID:/test.txt
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.
sock: Put "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/CONTAINER_ID/archive?noOverwriteDi
rNonDir=true&path=%2F": dial unix /var/run/docker.sock: connect: permission denied
subhashis@subhashis-virtual-machine:~$
```

TASK 2

FROM ubuntu : latest

Update Ubuntu packages

RUN apt-get update -y && apt-get upgrade -y

Install Python, Java, MySQL, and Nginx

RUN apt-get install -y python3 python3-pip openjdk-11-jdk mysql-server nginx

Expose port 80 for Nginx

EXPOSE 80

Start Nginx server

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

To build the Docker image, save the Dockerfile in a directory and run the following command in that directory:

docker build -t myubuntu .

This will build a Docker image with the name "myubuntu". Once the image is built, you can run a container based on this image using the following command:

docker run -d -p 80:80 myubuntu

TASK 3

```
version: '3'

services:

  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: examplepassword
      MYSQL_DATABASE: wordpress
```

```
    MYSQL_USER: wordpress
    MYSQL_PASSWORD: wordpress

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
    volumes:
      - ./wp-content:/var/www/html/wp-content

volumes:
  db_data:
```

This file defines two services, db and wordpress. The db service uses the mysql:5.7 image, creates a volume to store the MySQL data, and sets environment variables for the root password and the WordPress database and user details.

The wordpress service uses the wordpress:latest image, maps port 8000 to port 80 on the container, and sets environment variables to connect to the MySQL database. It also creates a volume to persist the WordPress content in the wp-content directory.

To deploy the services, run the following command in the same directory as the docker-compose.yml file:

Copy code

```
docker-compose up -d
```