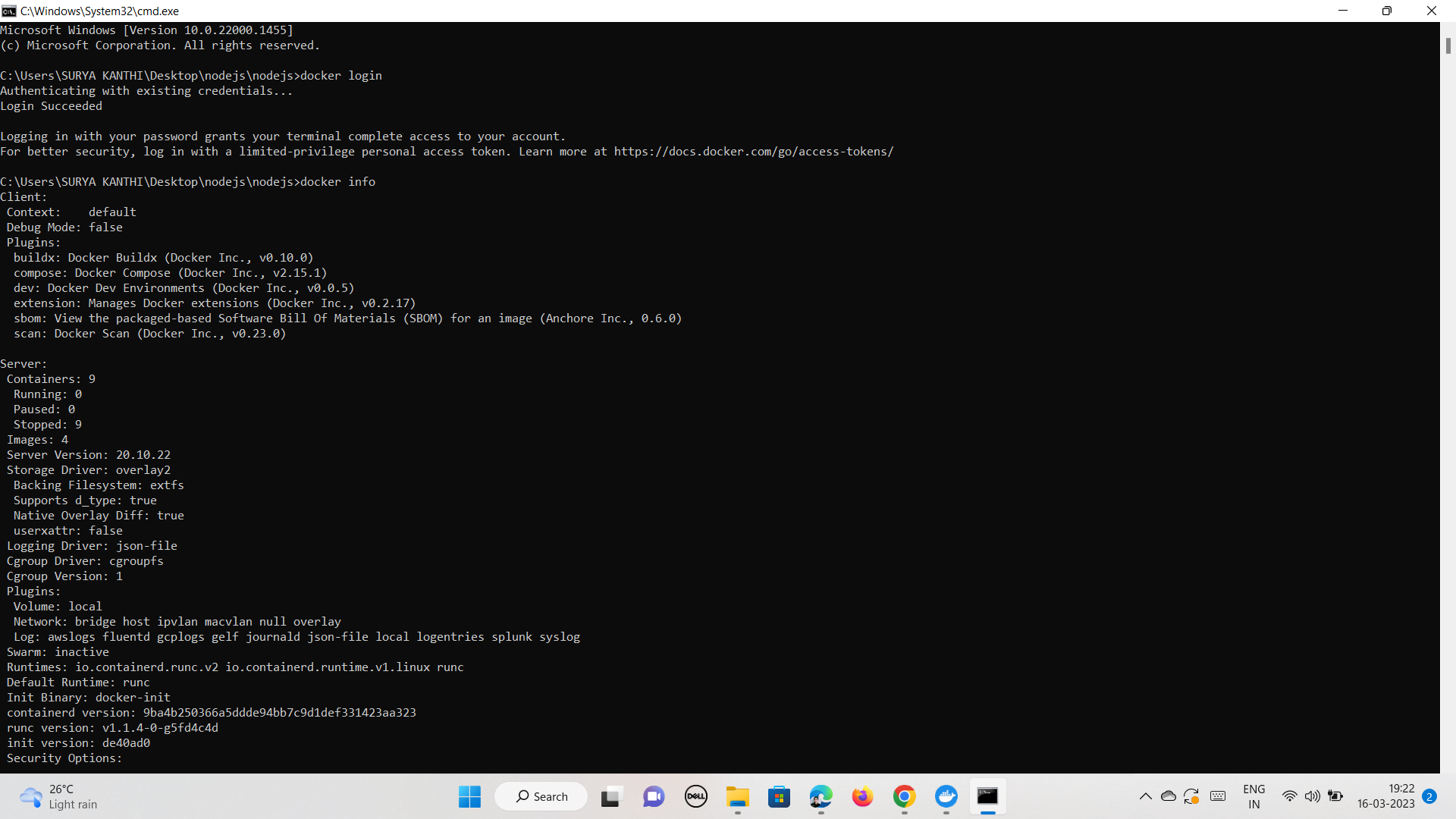
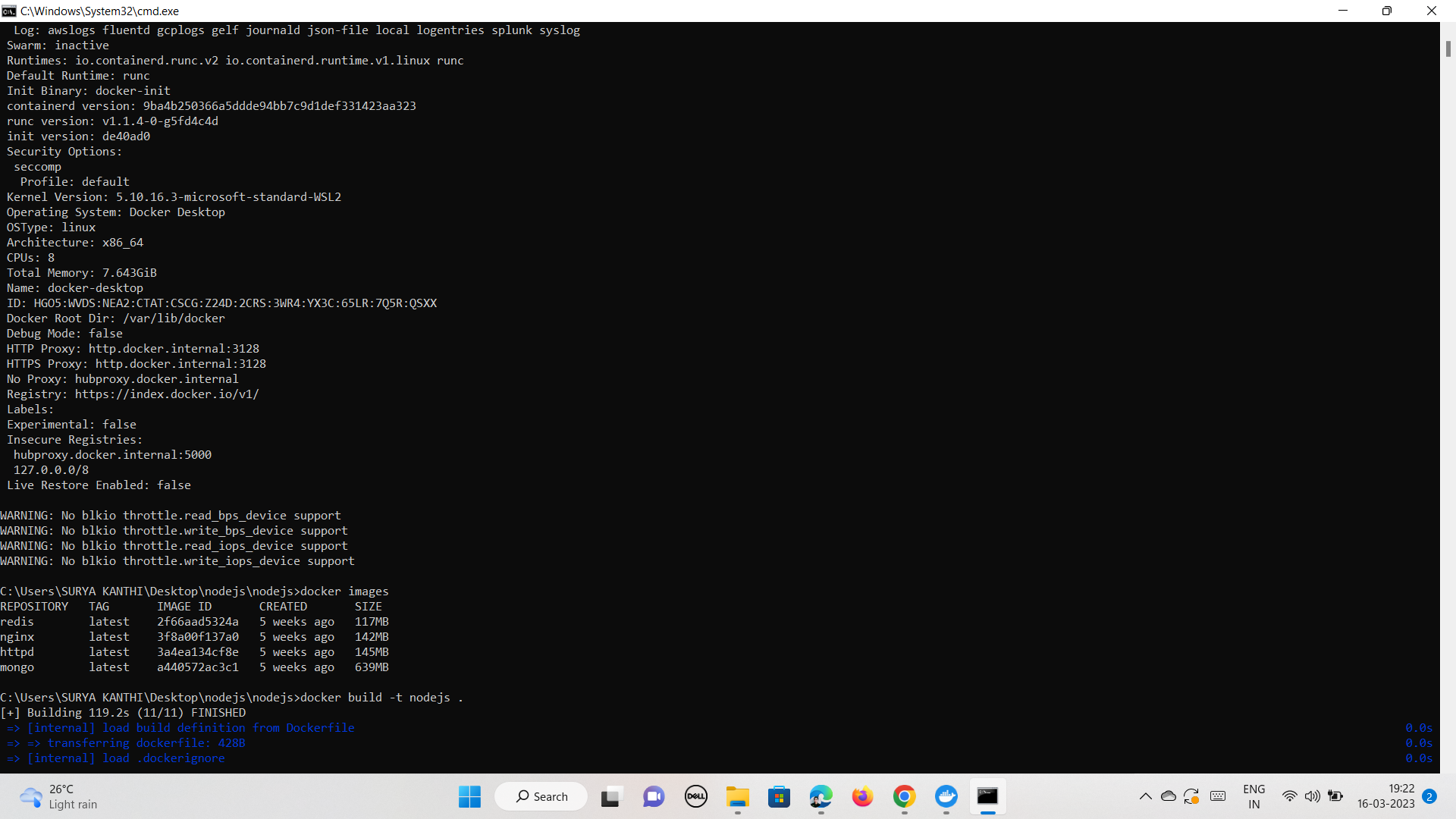
# QUESTION NO:1

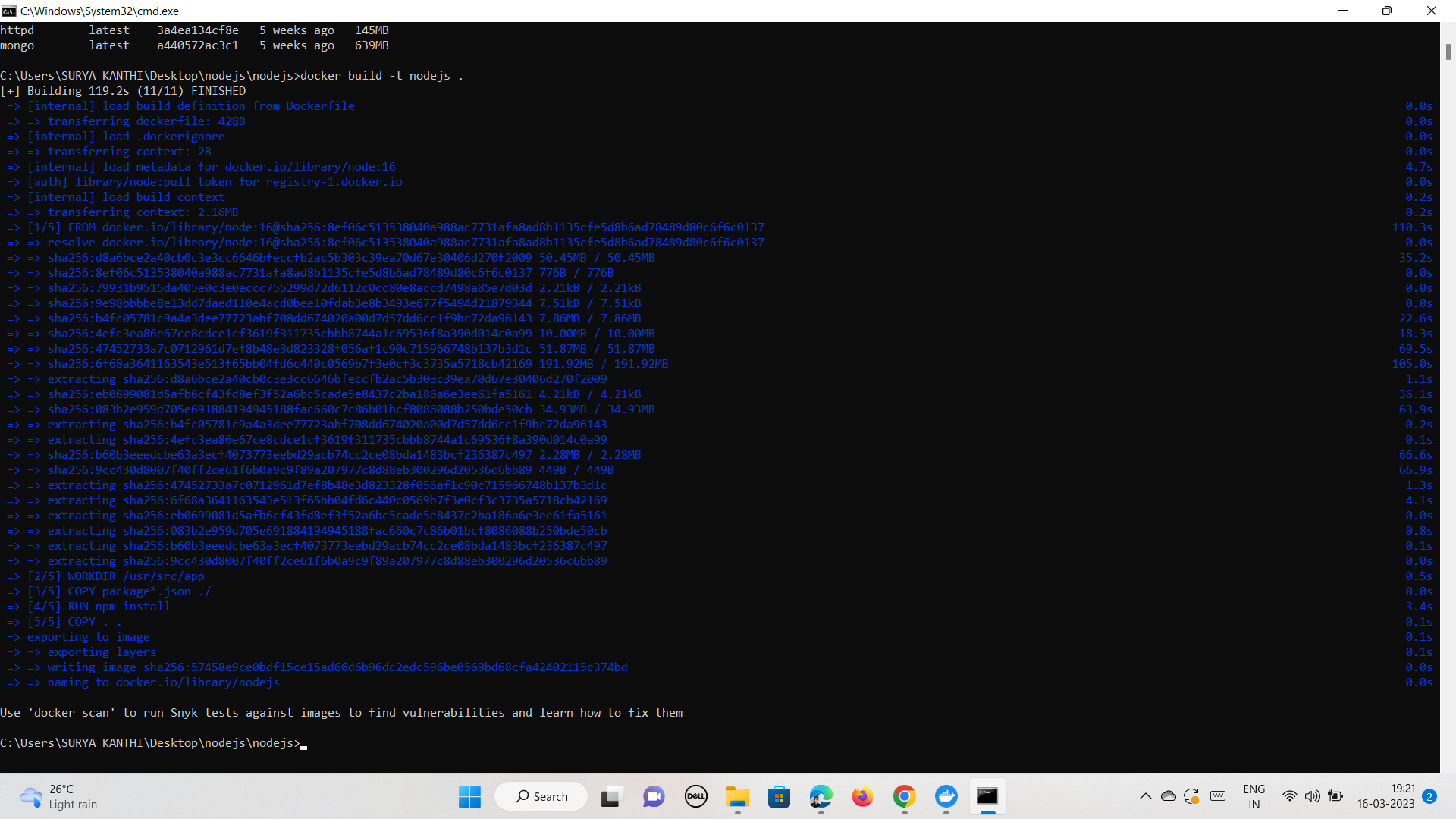
Q1.Create a docker container and image out of the docker file for the following application:

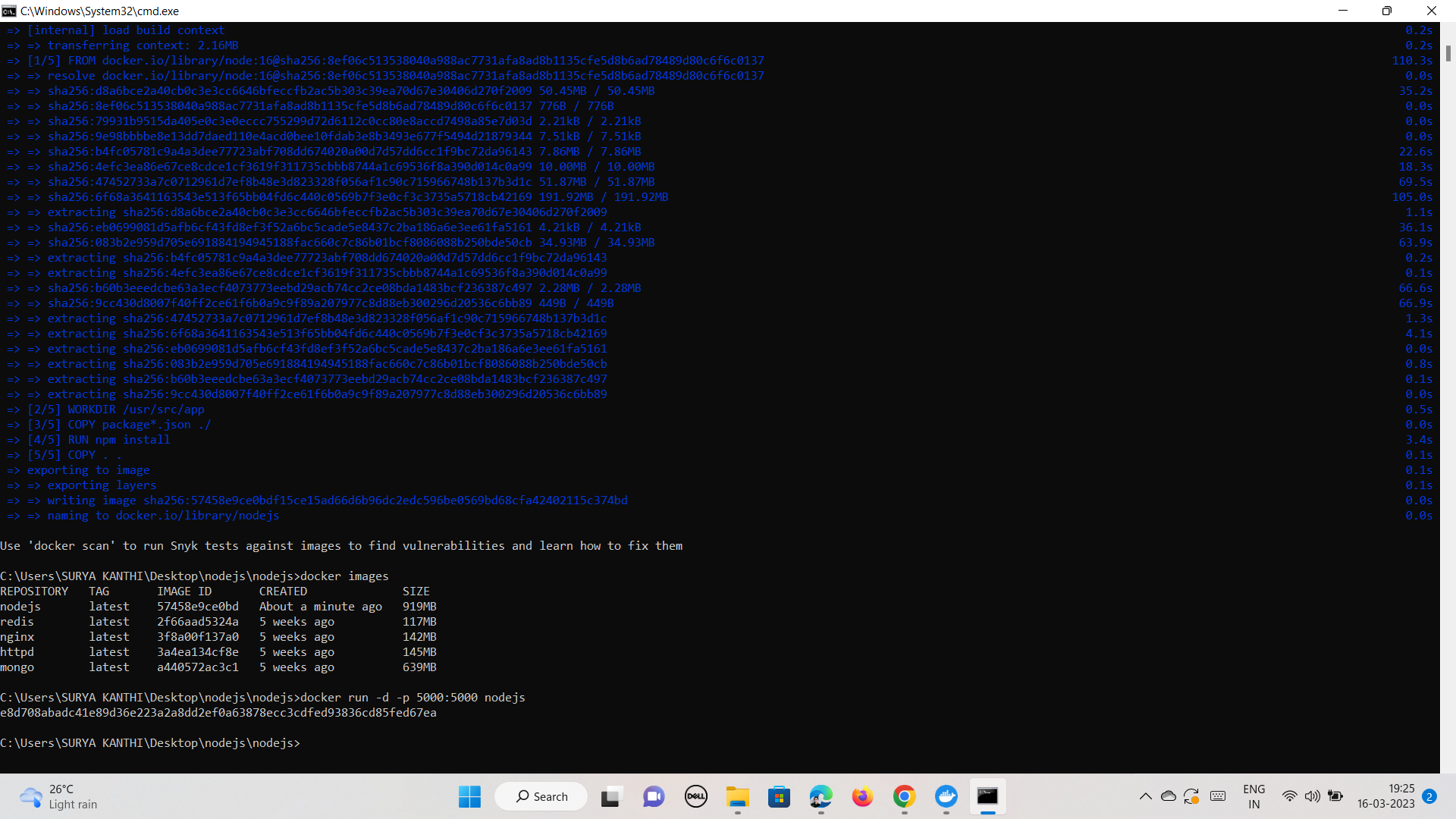
* Python Flask
* NodeJS

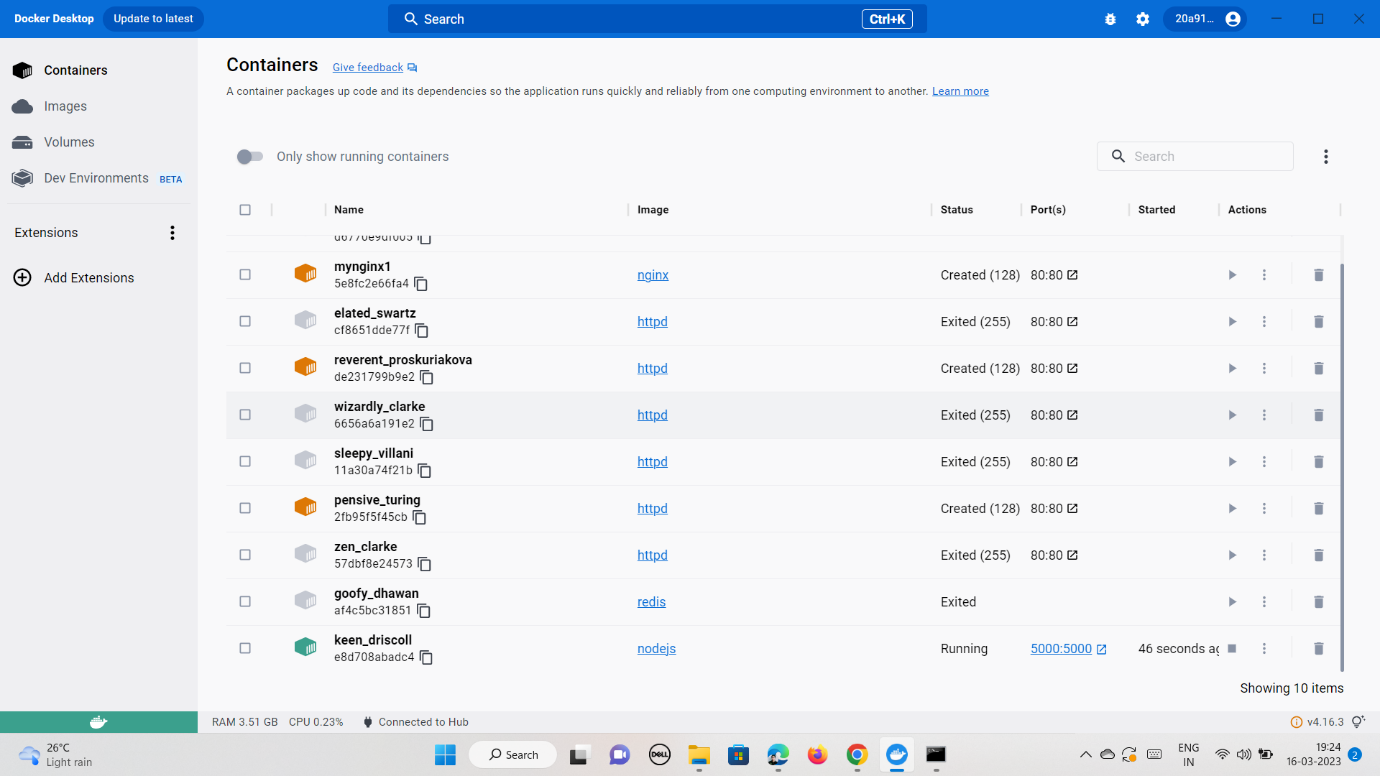
NodeJS:

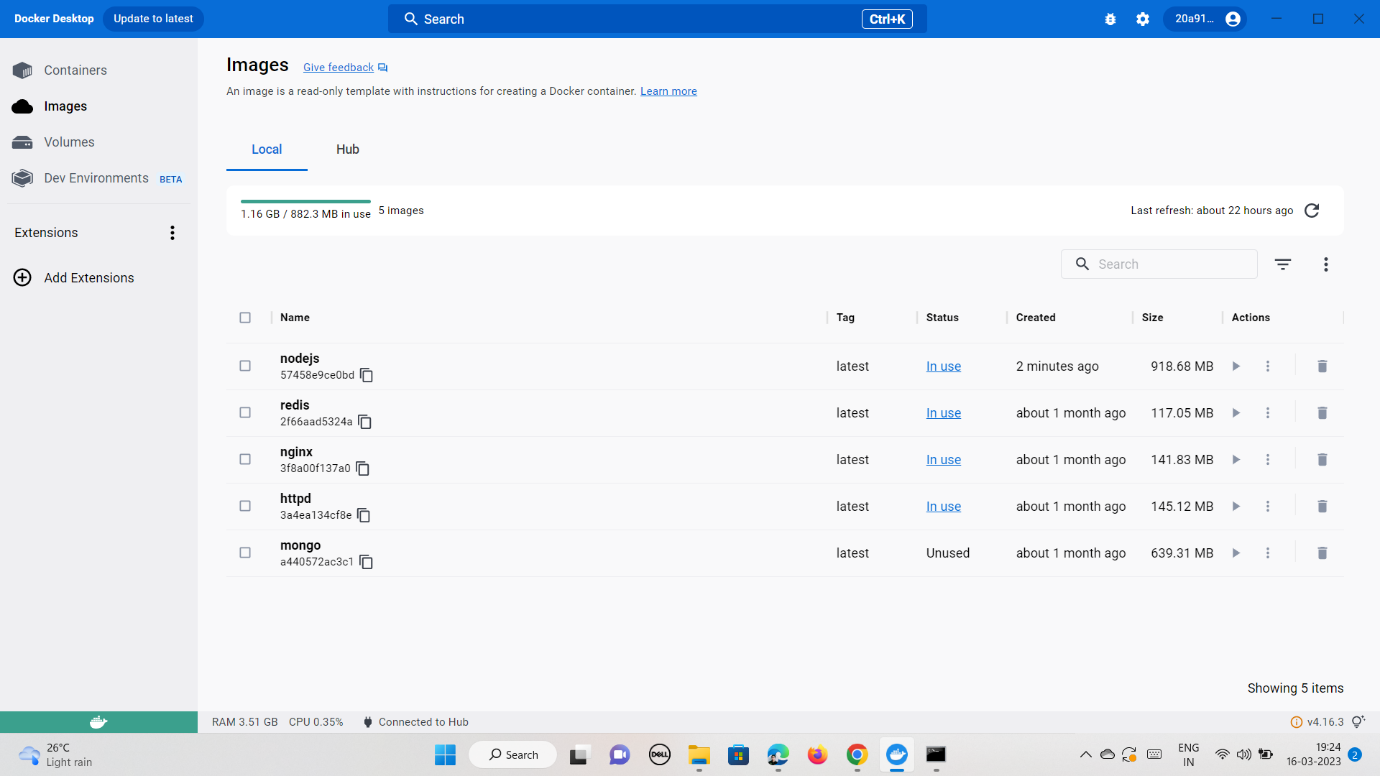


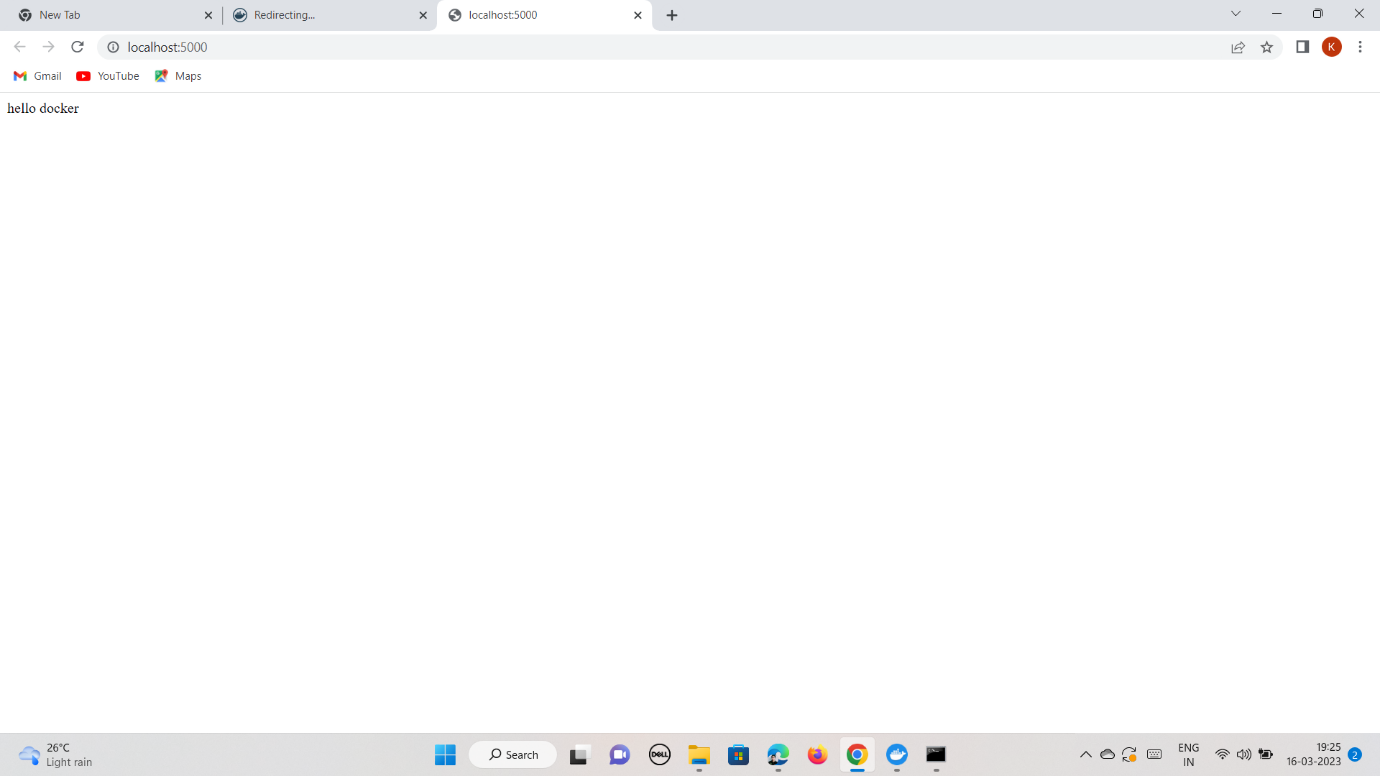




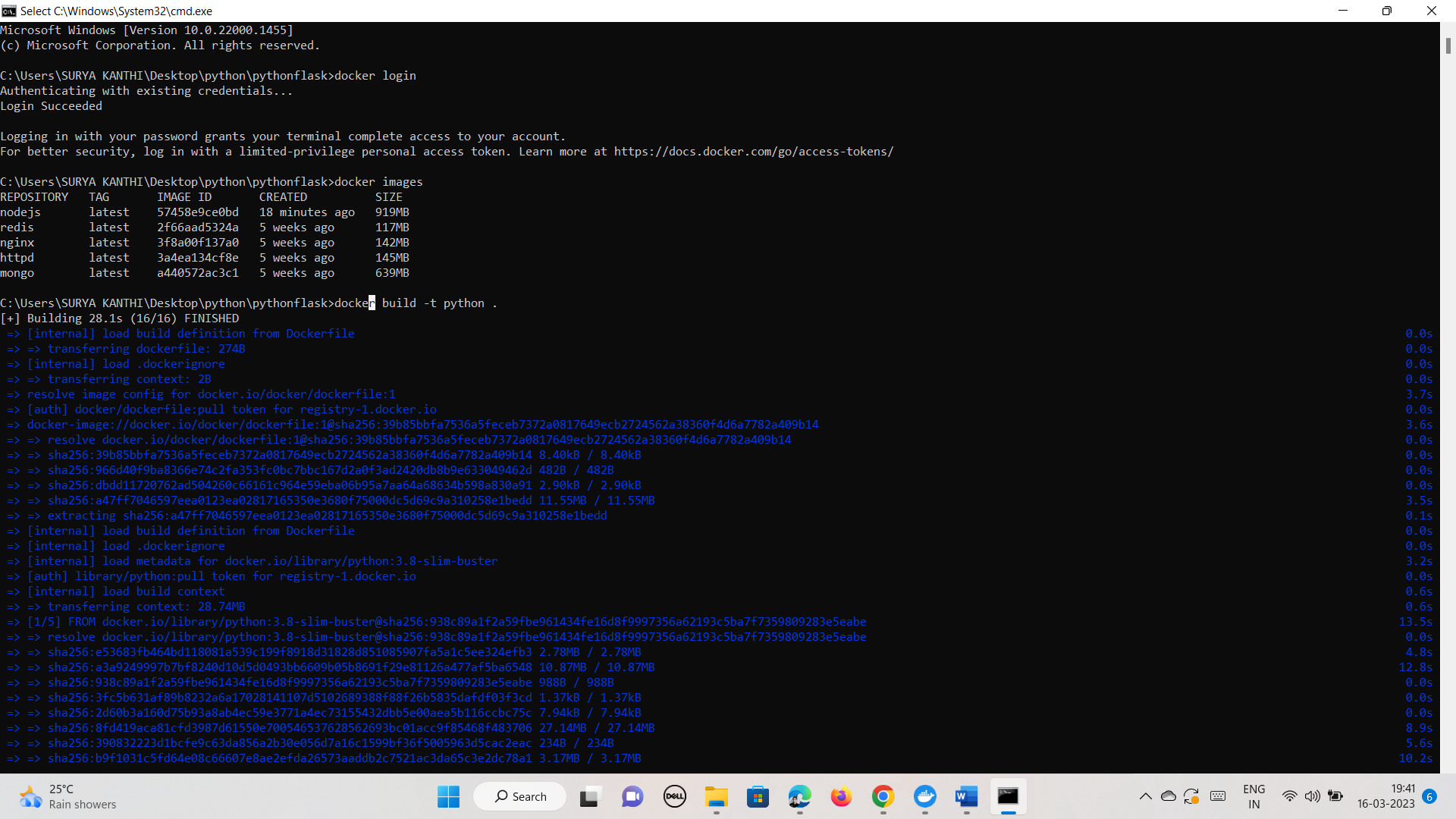


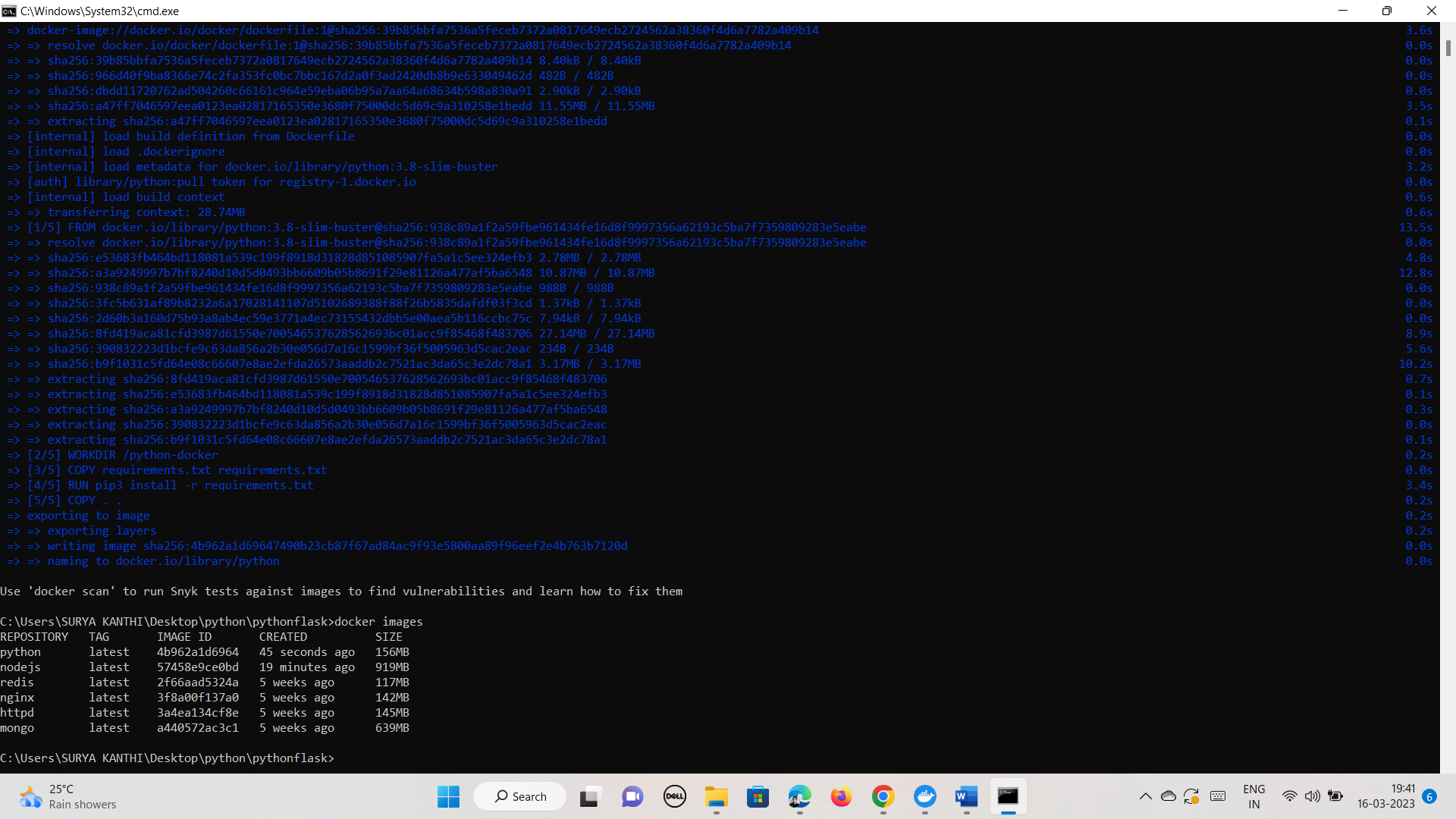


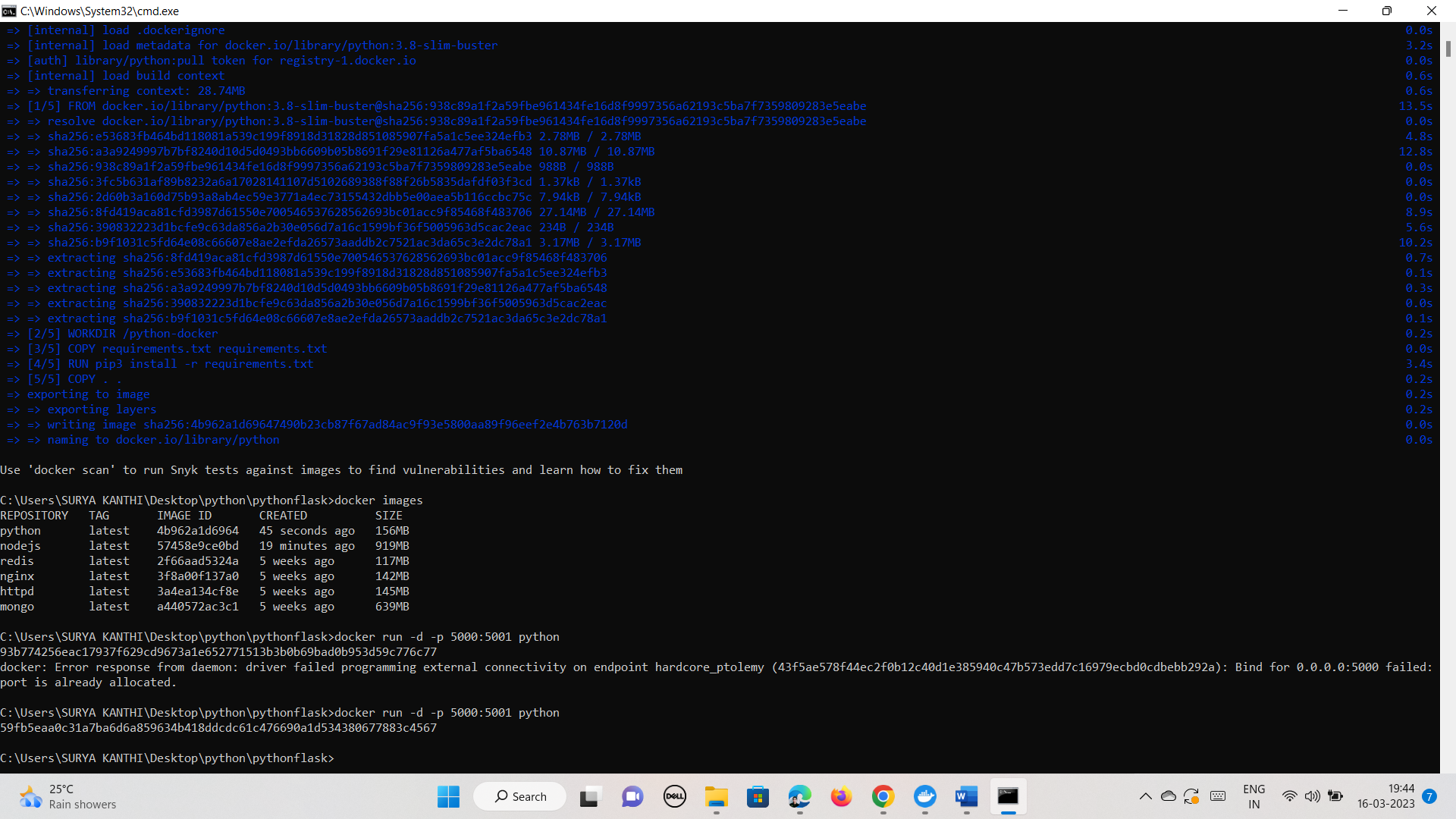


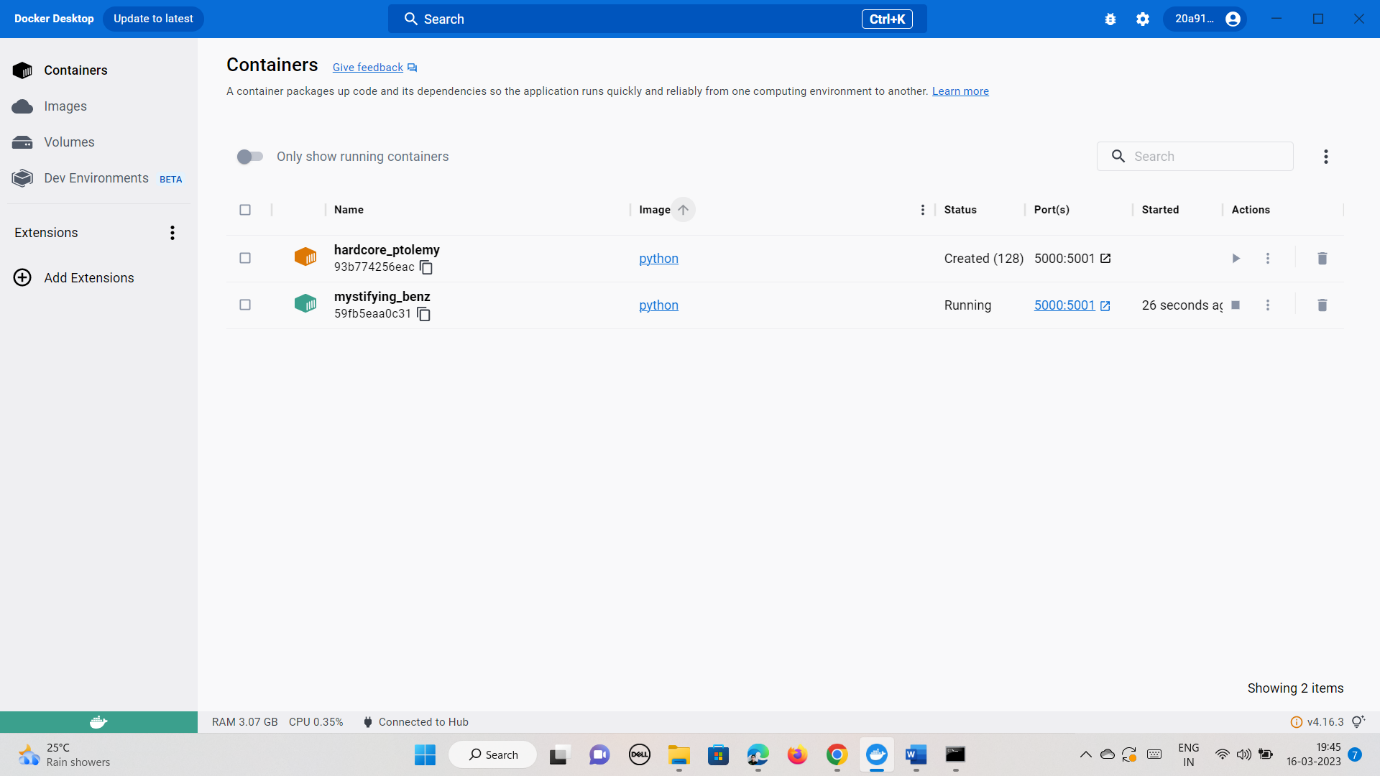


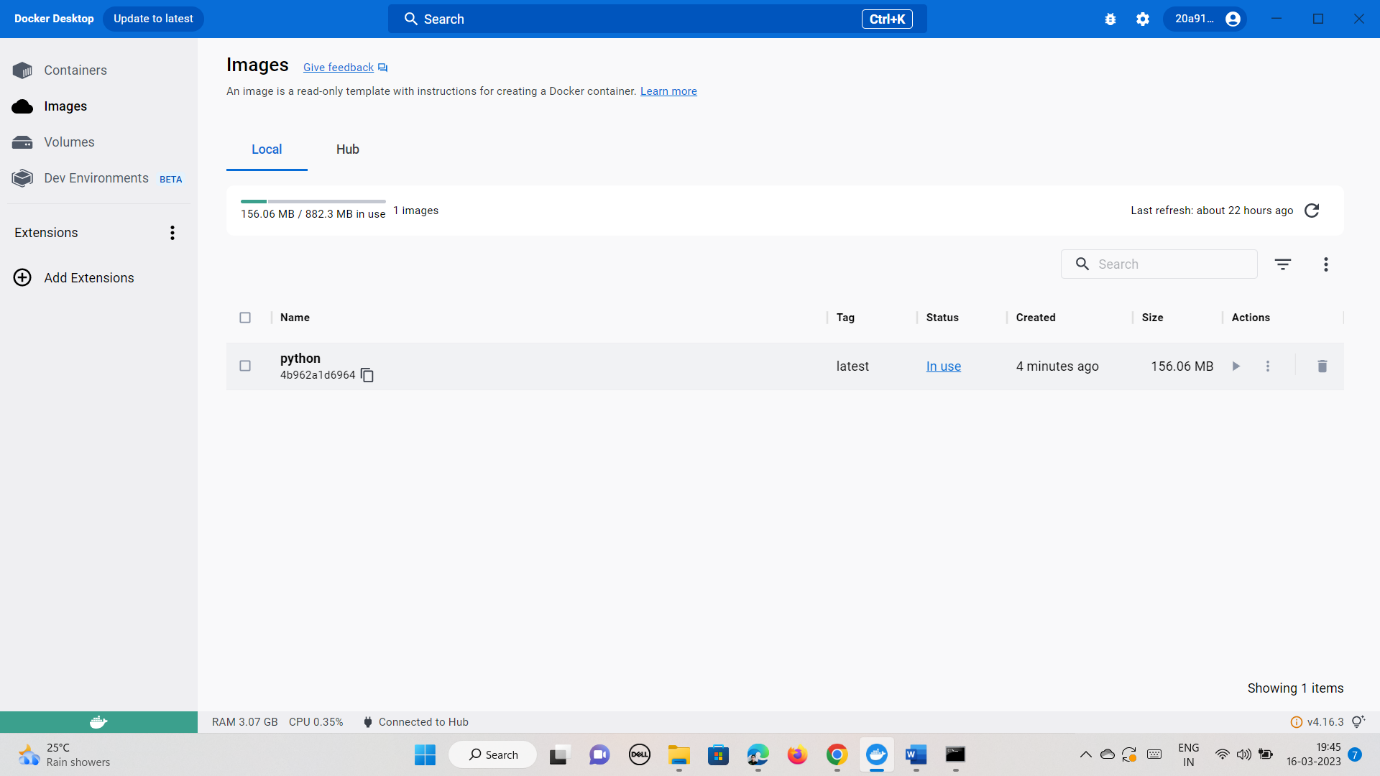
Python flask:

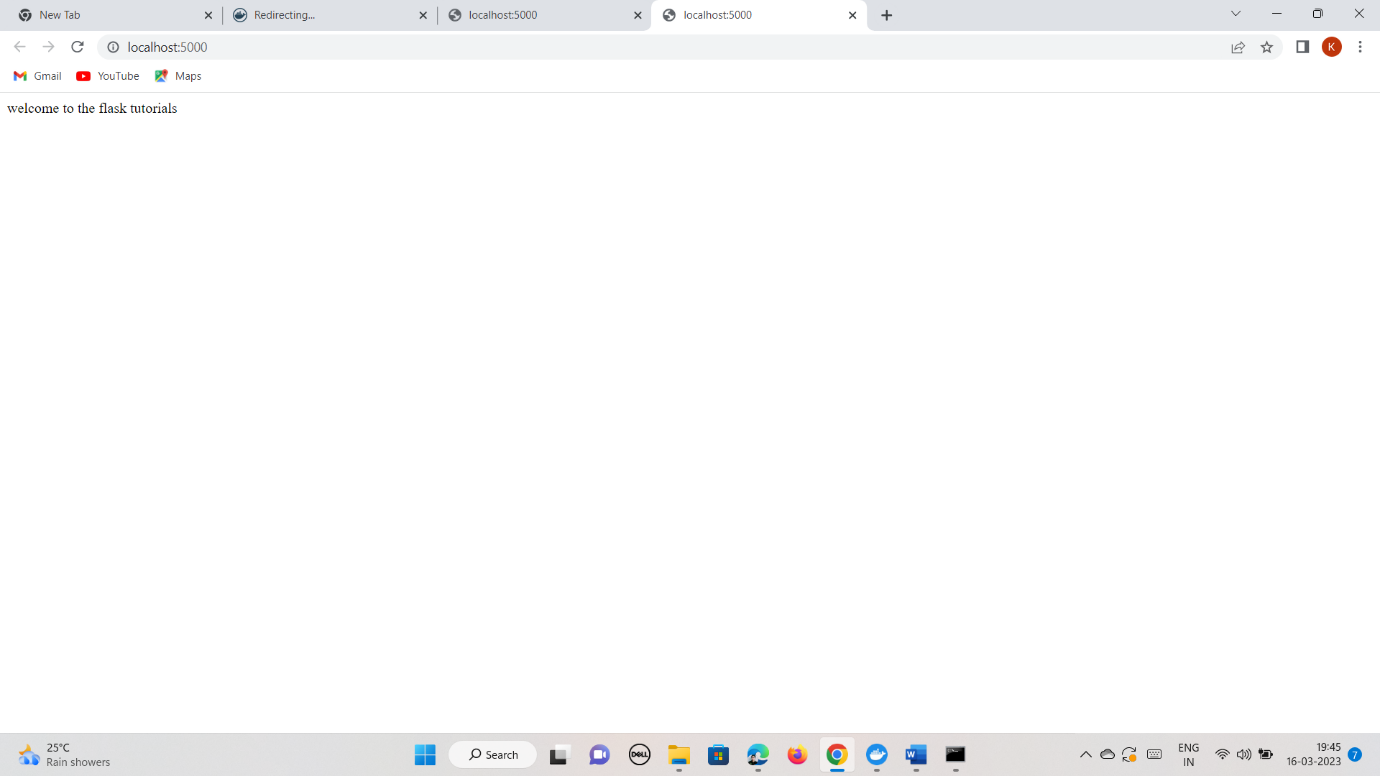












By using Docker file only, we can create Docker images in our docker desktop or repository.

Docker file contains some lines of code.

NPM: node package image

FROM: it defines which service application image ti be used for our application

WORKDIR: it defines the folder structure of the application inside the container and where everything we download will be.

COPY FILE: which contains our application dependency.

RUN: it tells the docker to perform installation of whatever is there in your dependency file

COPY . . : to copy all the stuff which is just got downloaded and keep it in desired folder.

EXPOSE 3000 : it tells docker that our application wll run on particular port inside the container.

CMD: [“node”, “index.Js”]: it tells the docker to run this command in command line environment to start the application.

Steps:

---first open nodejs or pythonflask folder

---open git bash and build image

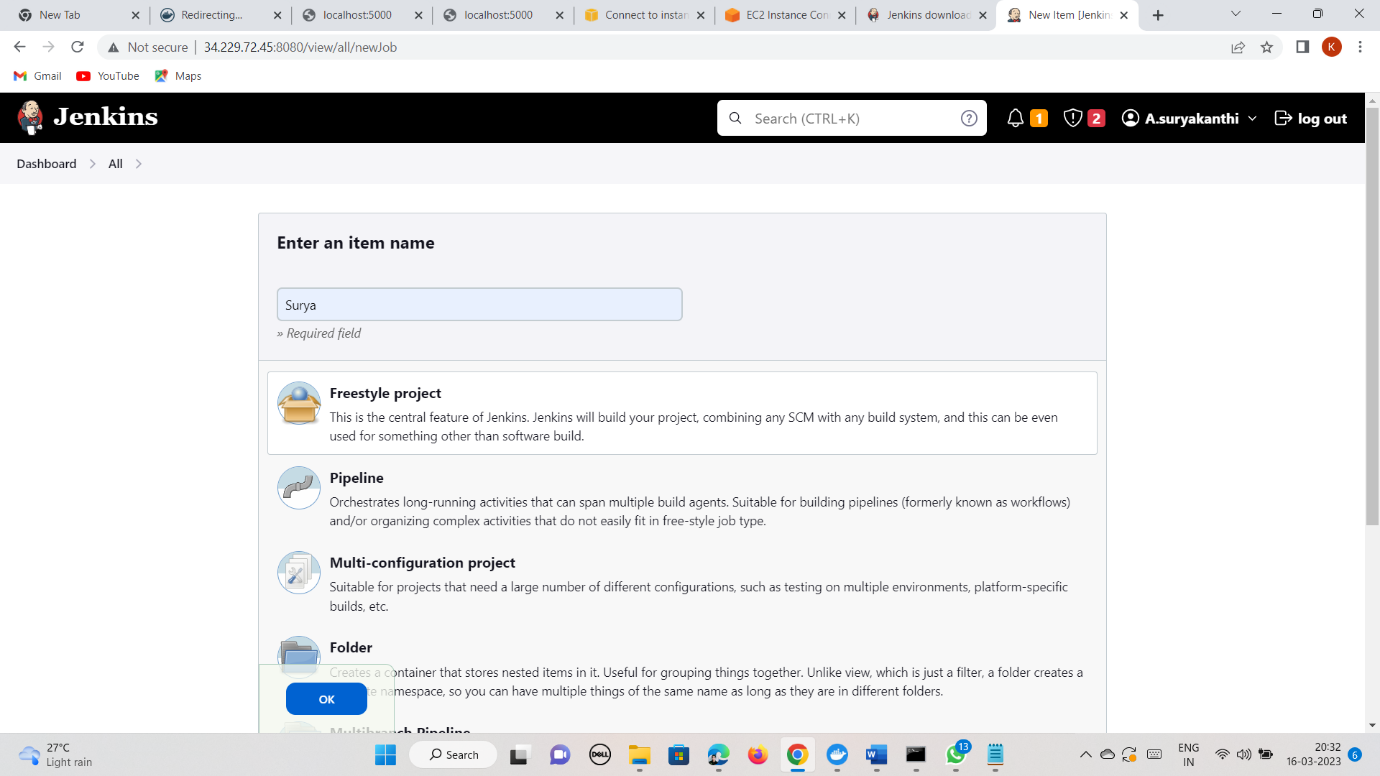
---run the image on port which is given in dockerfile

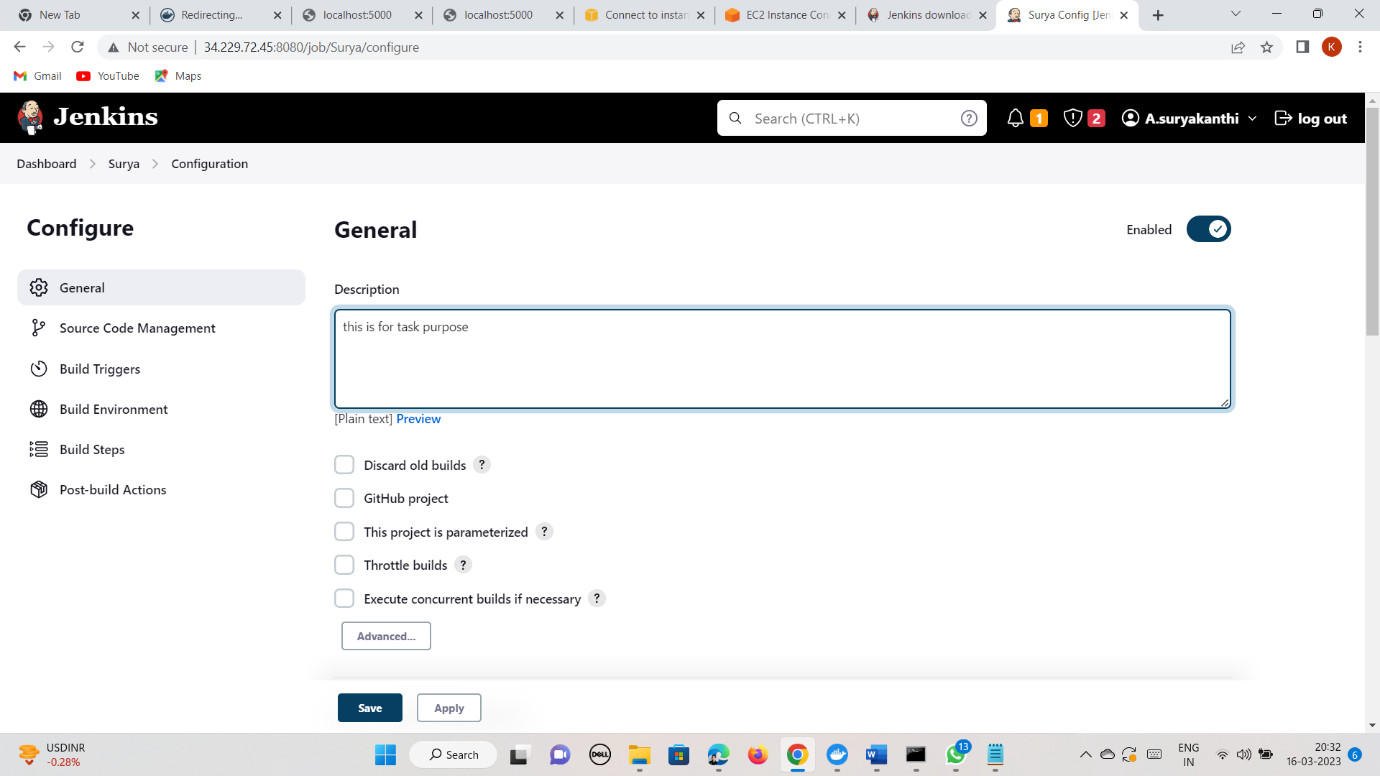
---check the created image in docker desktops

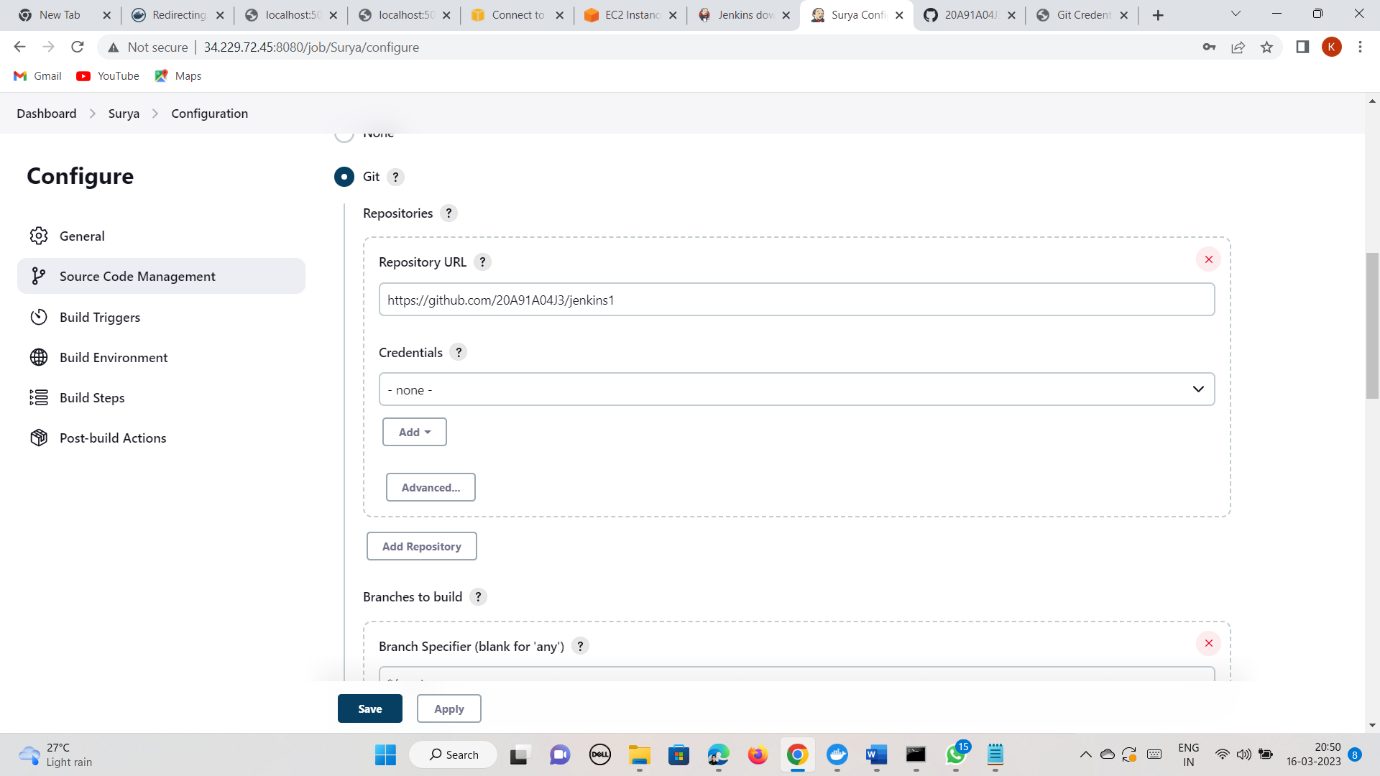
---open the port(localhost:portno) on new tab to see the content in the applicaton.

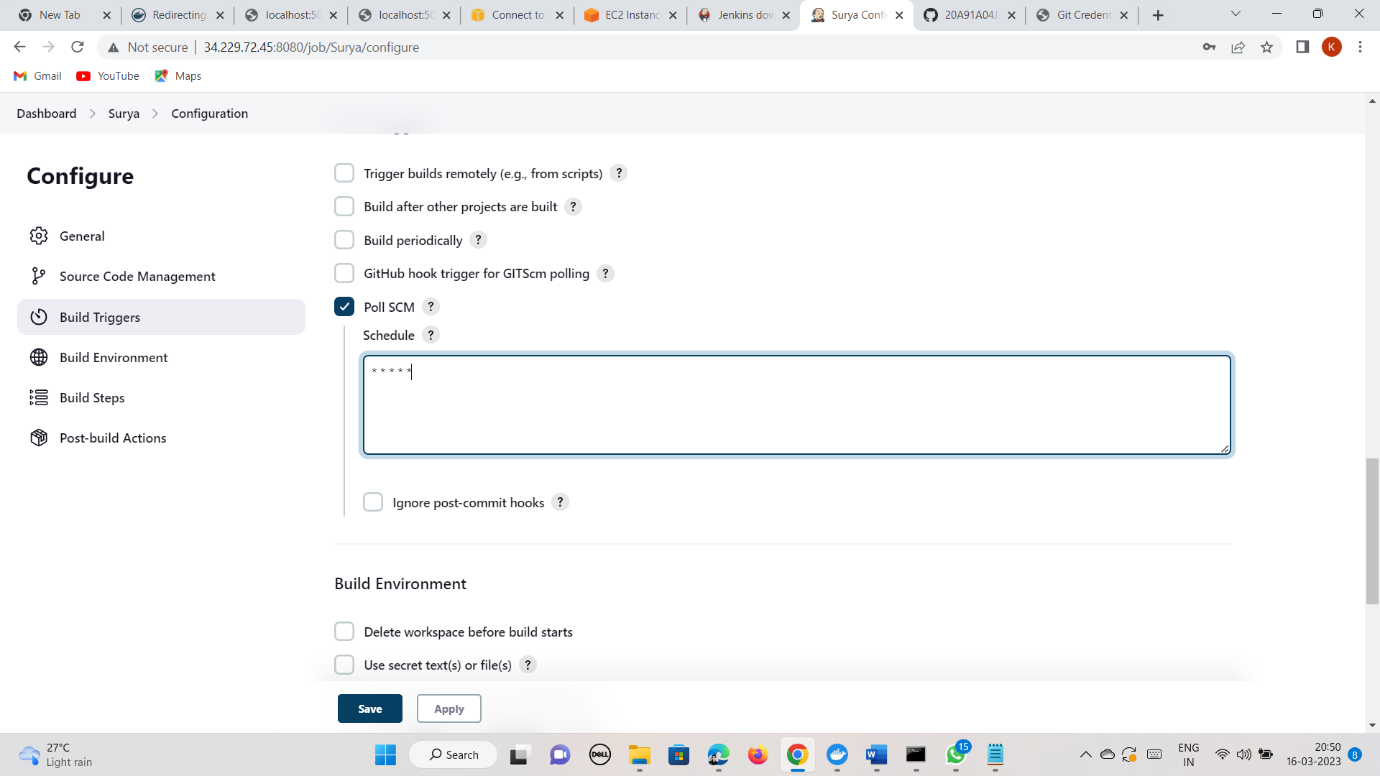
# Question:2

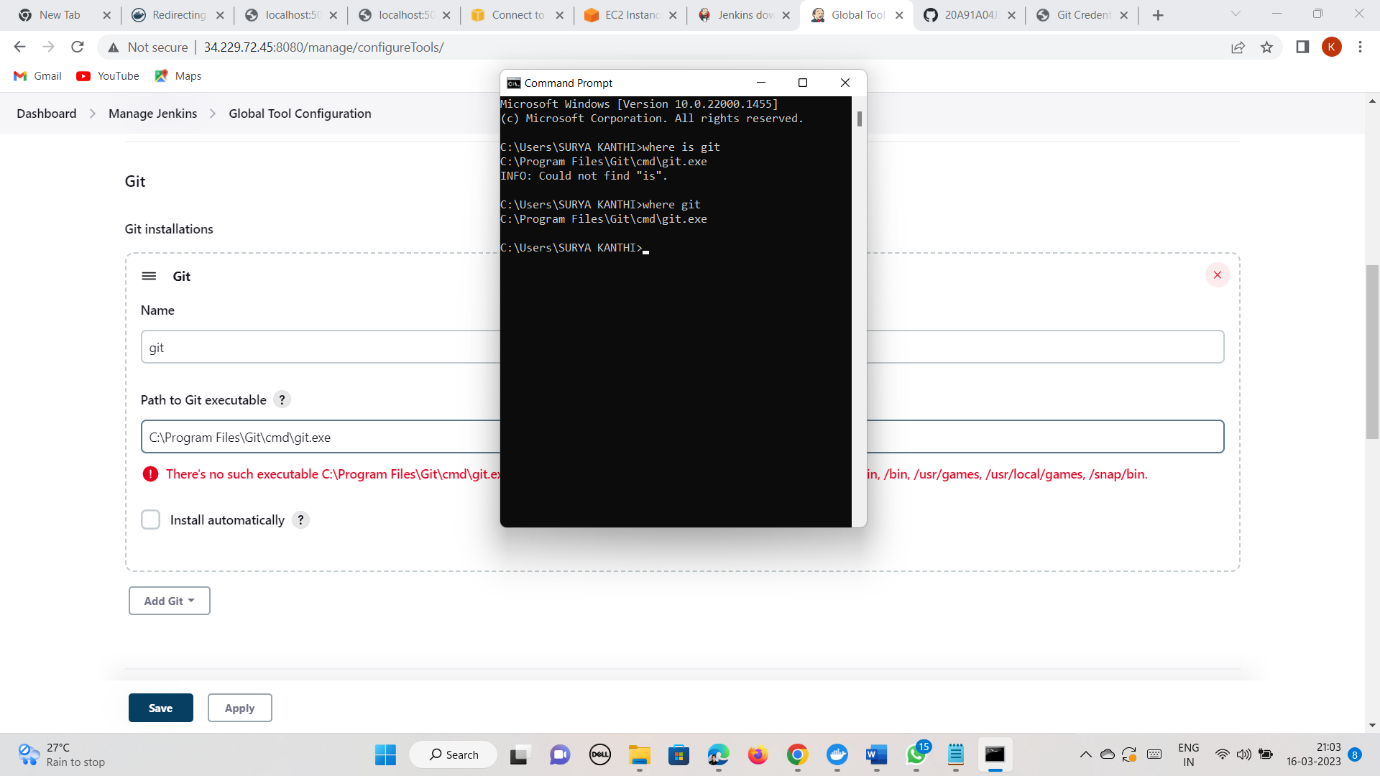
Q2.Create a CI-CD pipeline for a NodeJS Application in Jenkins. All the steps involved in it should be given in a screenshot and your Jenkins username must be visible in the screenshot.

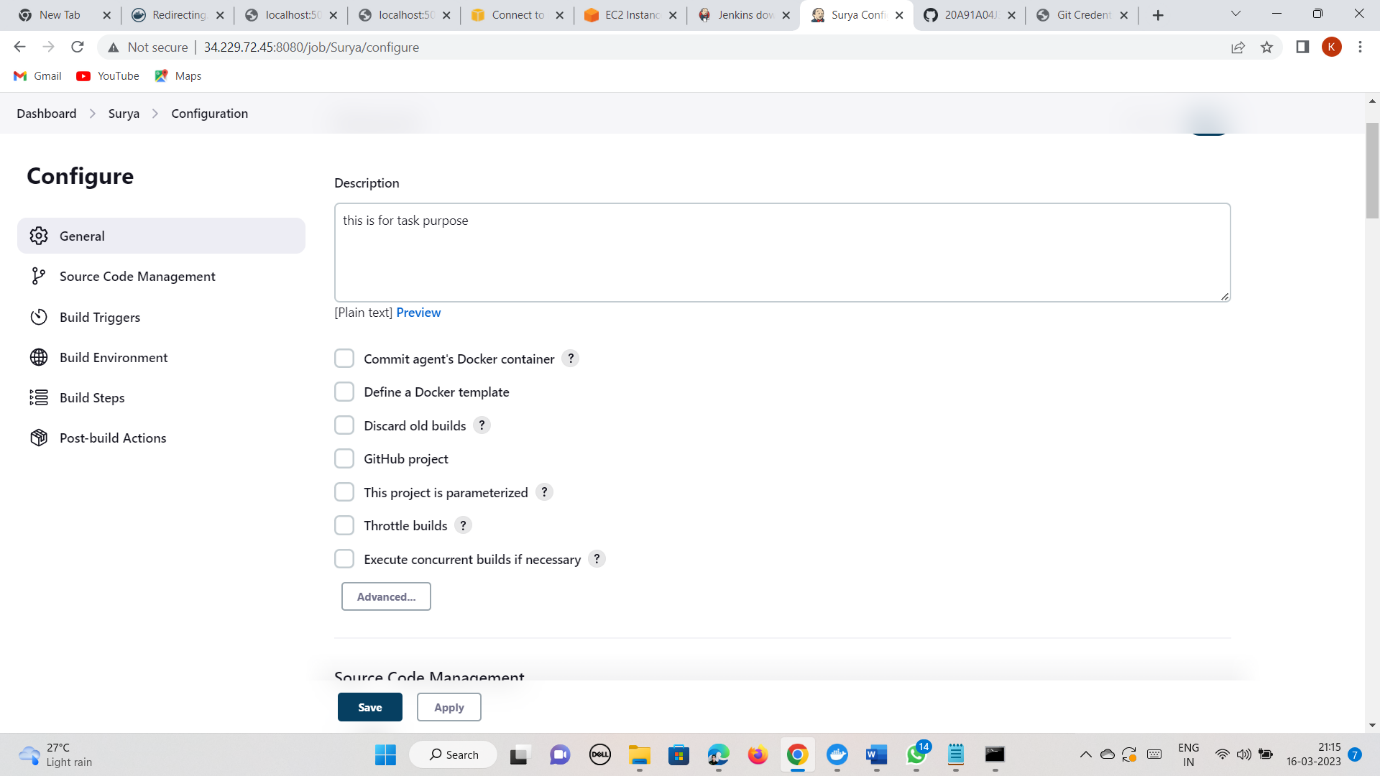


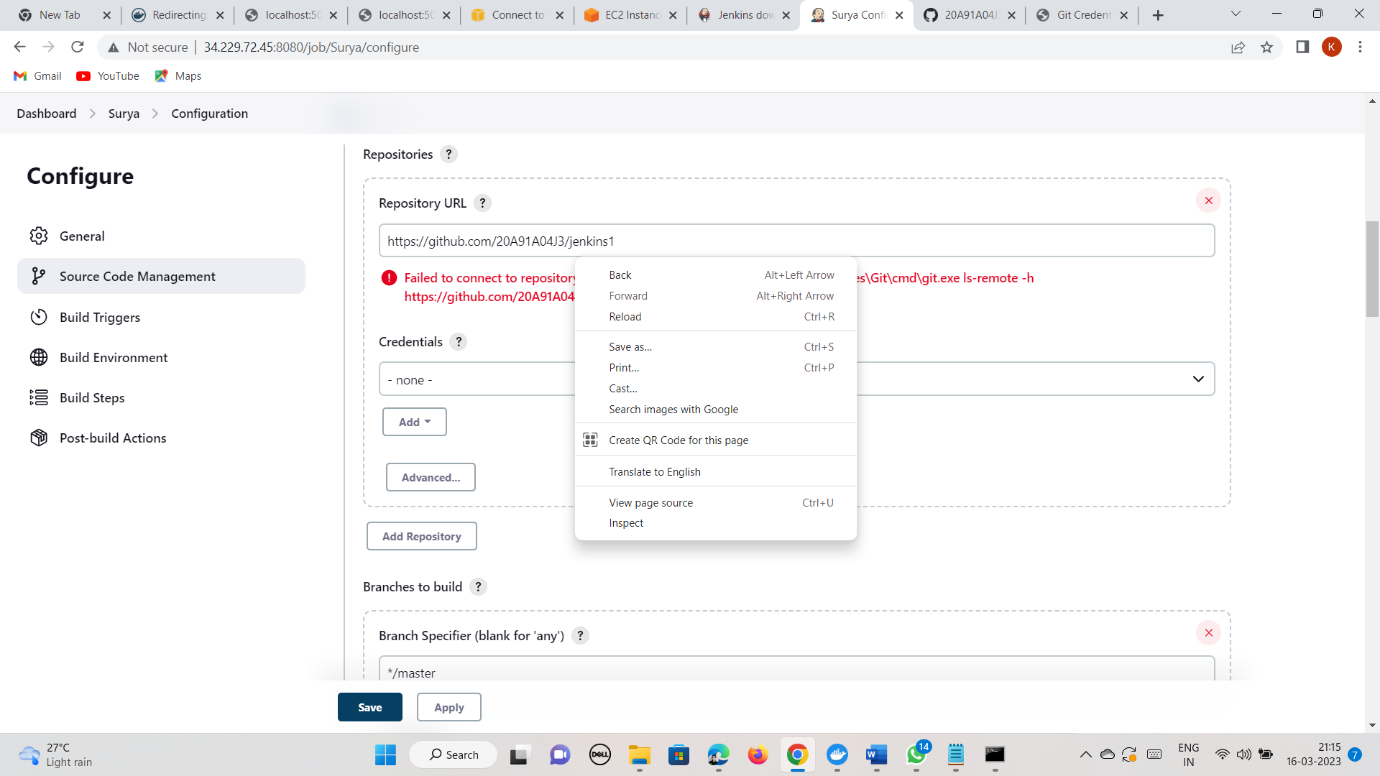


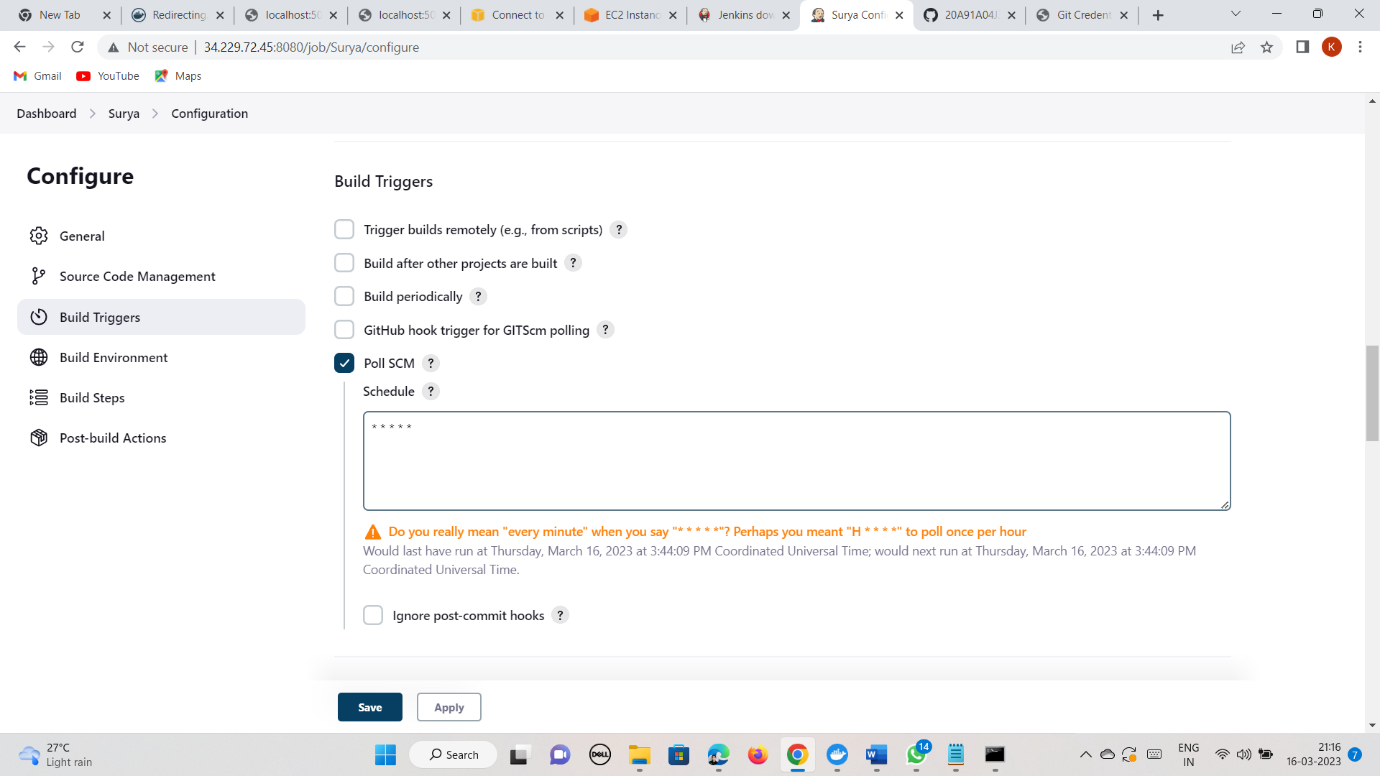


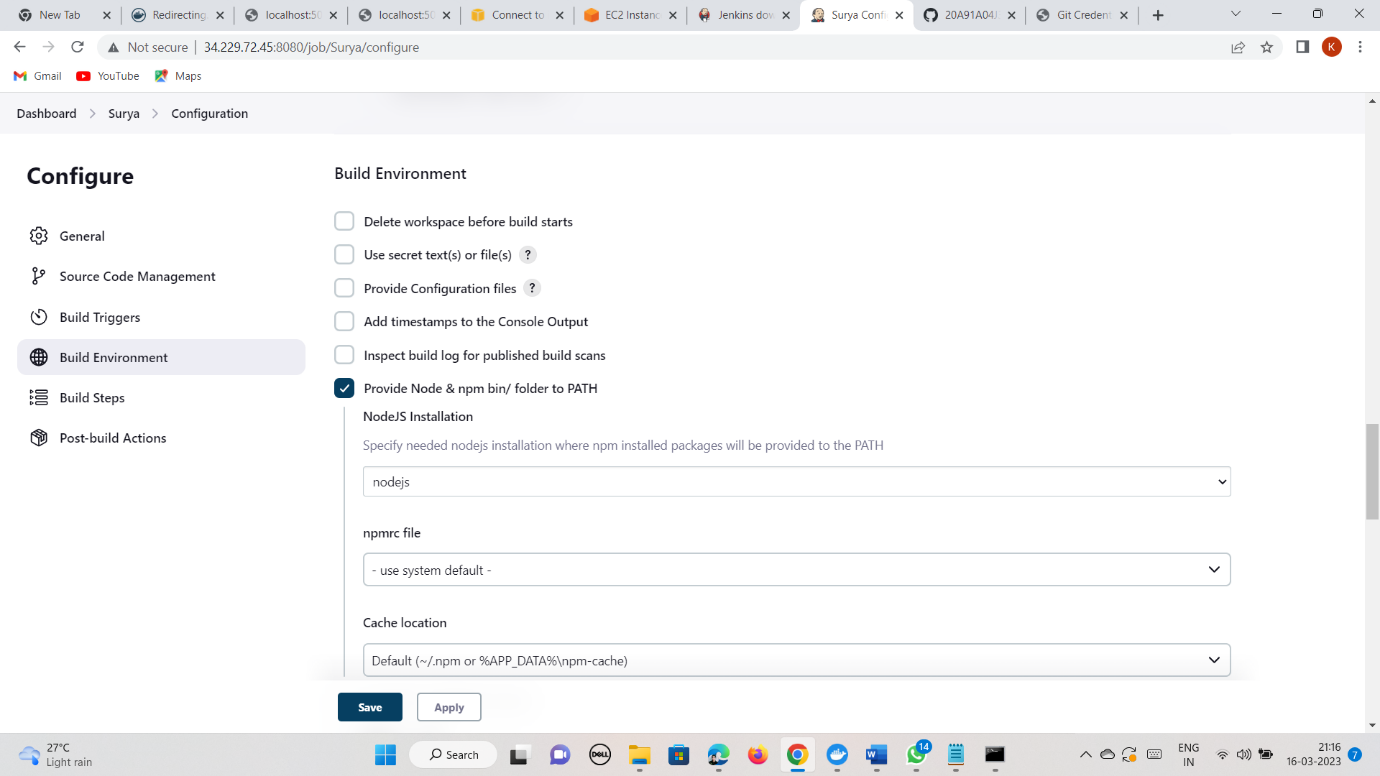


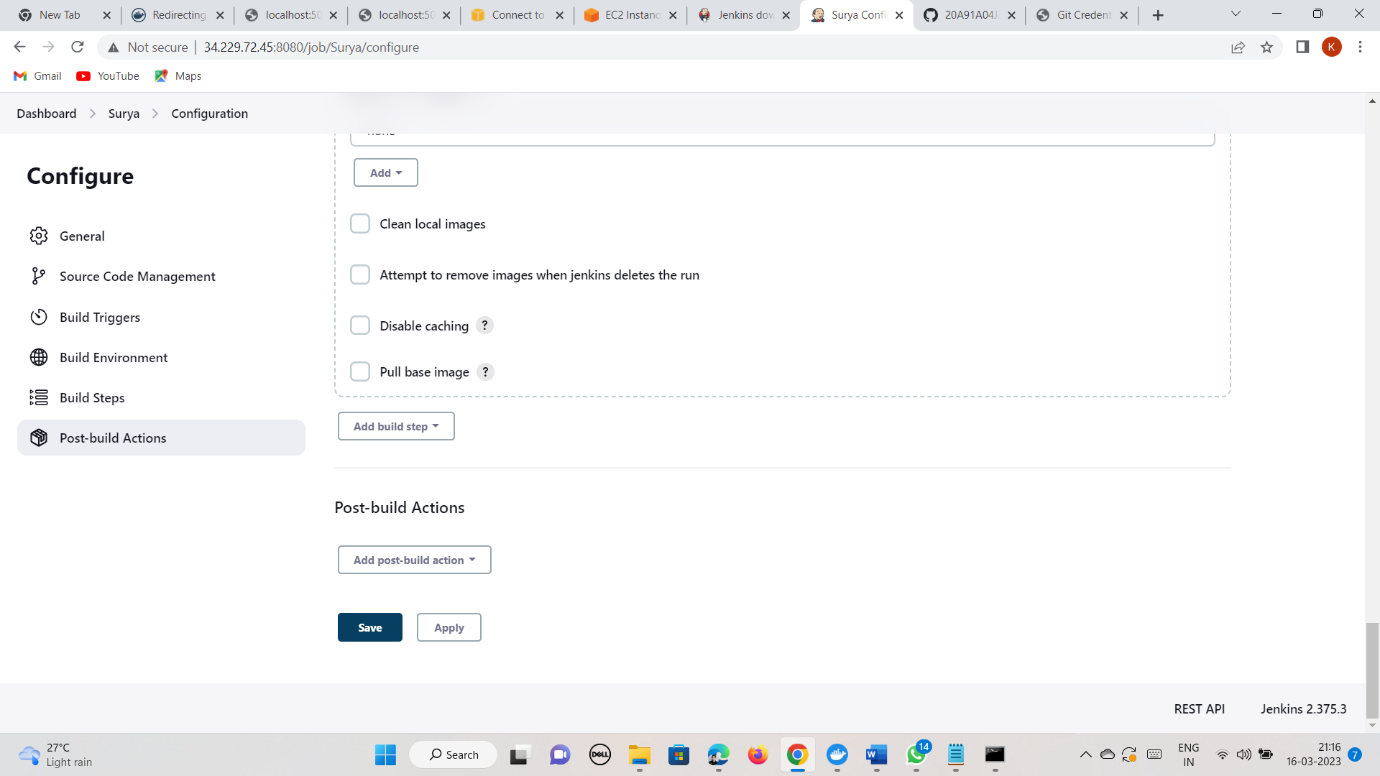


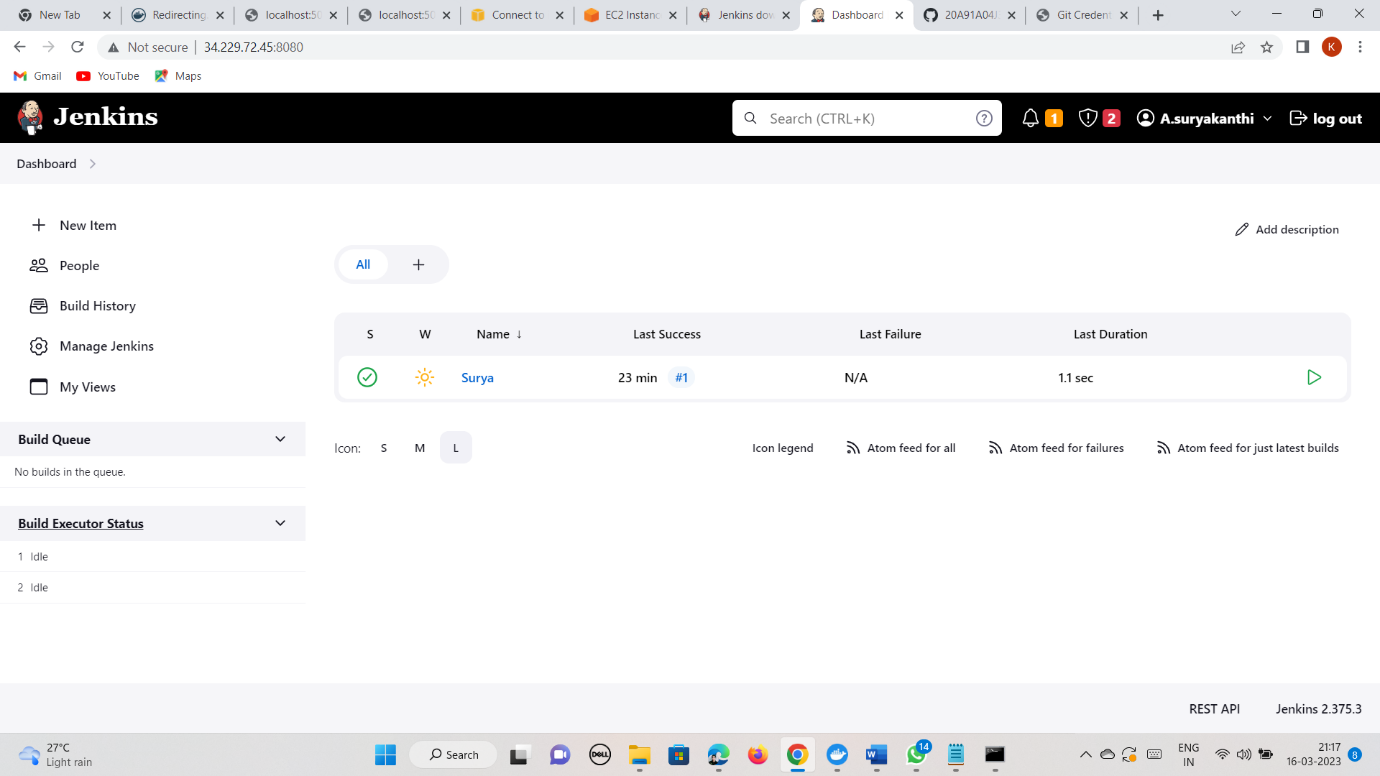


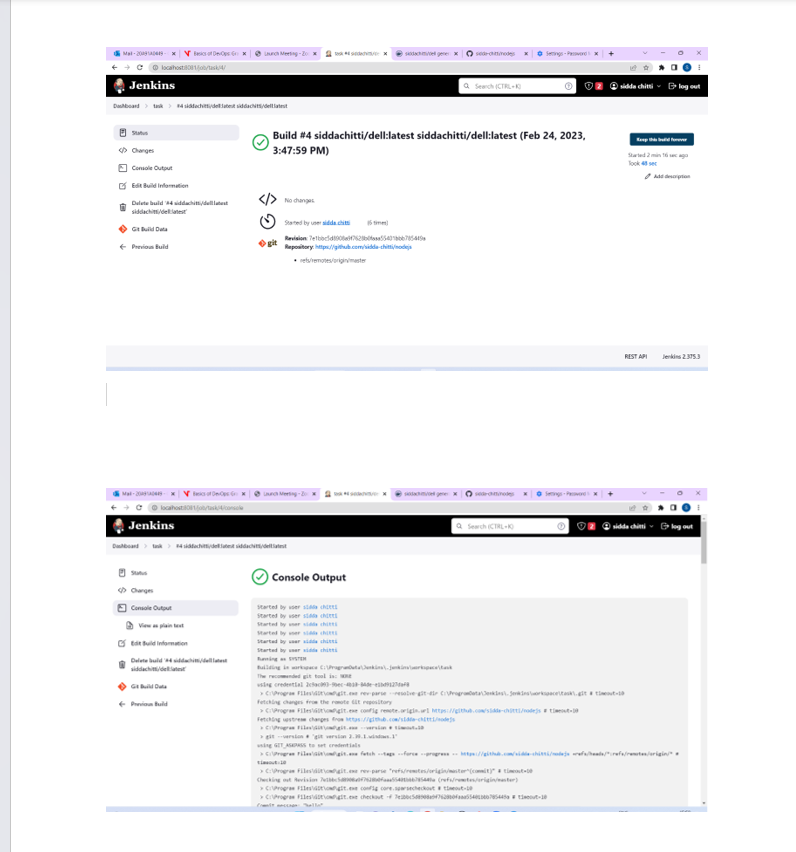












# QUESTION:3

Q3. Create documentation of how you are going to create a CI-CD pipeline for Python applications.

In the documentation, ensure that you mention each step that should be taken to configure the CI-CD pipeline for a Python application including the plugins you are using and the global tool configuration.

Also, create a pipeline using Jenkins.

First, we need to create a repository in our github

Push the python application folder in that repository

Go to Jenkins on localhost port number then follow the below steps

Select New item live any description (optional)

Select git in source code Management Give Repository link It Repository Public then select None in credentials otherwise give the GitHub credentials that is username and password

Select poll scm in build triggers and give 5 stars with space Select

build steps (docker build and publish give docker hub Repository Name give tag (latest) give Registry credentials that is user name and password click on save.

Go to Dashboard

click on Manage Jenkins on global tool

click on global tool Configuration

Go to git

Give Name as git

give path

Go to python

Give name as python

Click on save.

go to configure!

select build environment (provide Node & npm bin folder the path)

click on save.

# QUESTION:4

Q4.Create a manual CI-CD pipeline using a bash script and cron jobs for a Node JS application.

