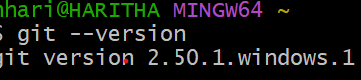
**MODULE-1**

GIT INSTALLATION

1. Download Git version 2.50.1. windows. 1cd. After downloading, open the file.
2. Click ‘Yes’ and proceed with the instructions. Click ‘Install’.
3. The Git is ready to use.
4. Go to browser and create a git hub account by giving the respective details.

BASIC GIT COMMANDS

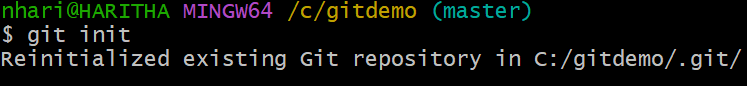
1. To check the version of the git , use the command “git –version”

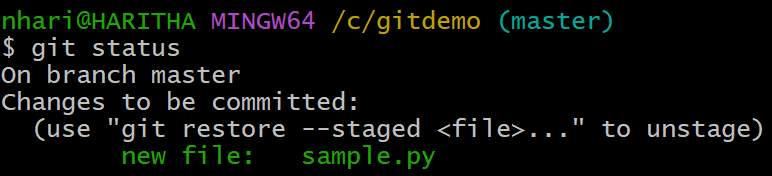


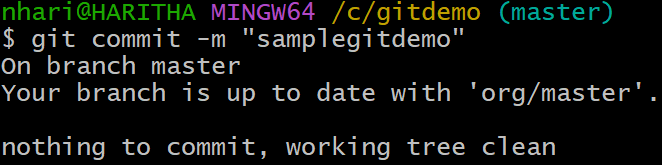
1. We have to change the directory to c drive and then to the folder named “gitdemo” (create and then change the path )

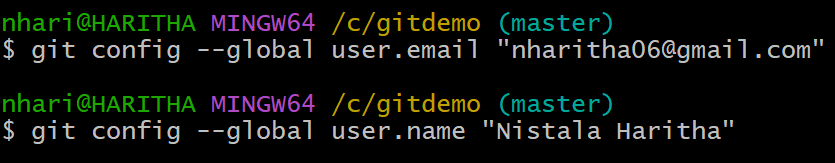
  

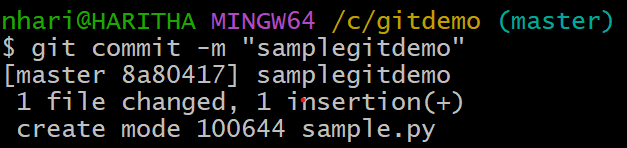
1. We have to initialise the git. (Observe that path has changed

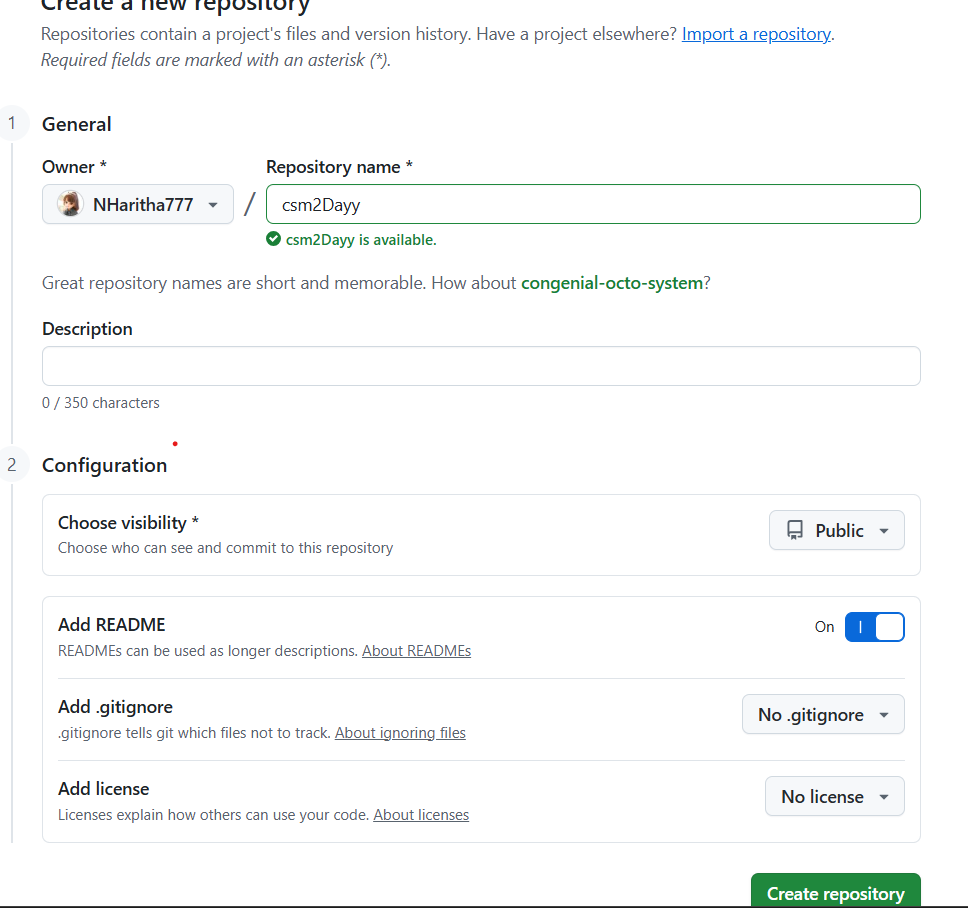
1. Open notepad and write a python program (Hello World) and save it as sample.py -> all files under the folder named “gitdemo”. Now check for the status. 
2. Commit the file using git commit -m “samplegitdemo” (sample- file , gitdemo- folder). Initially there will be nothing to commit.



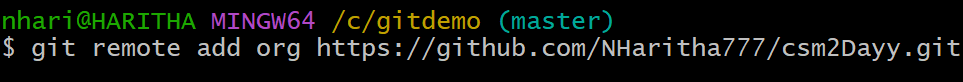
1. It asks for the user and system details. Provide them by the given git commands:
2. Now add and then commit the file. We can observe that 1 file inserted.



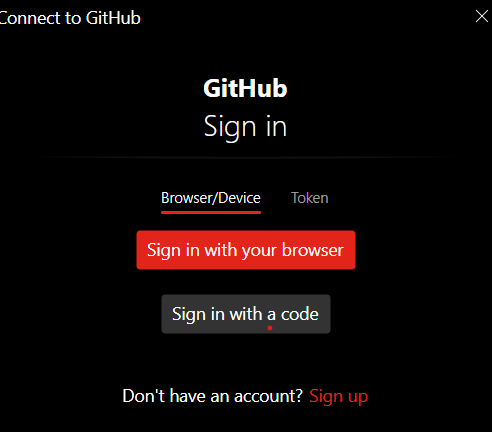
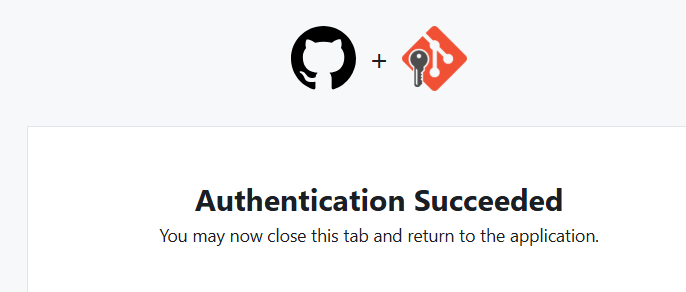
1. To create a new Repository in github, on the top left you can click on New or on the right top click on +. Give the repository name as “csm2Dayy”. Add a “readme” file and click on “Create repository”



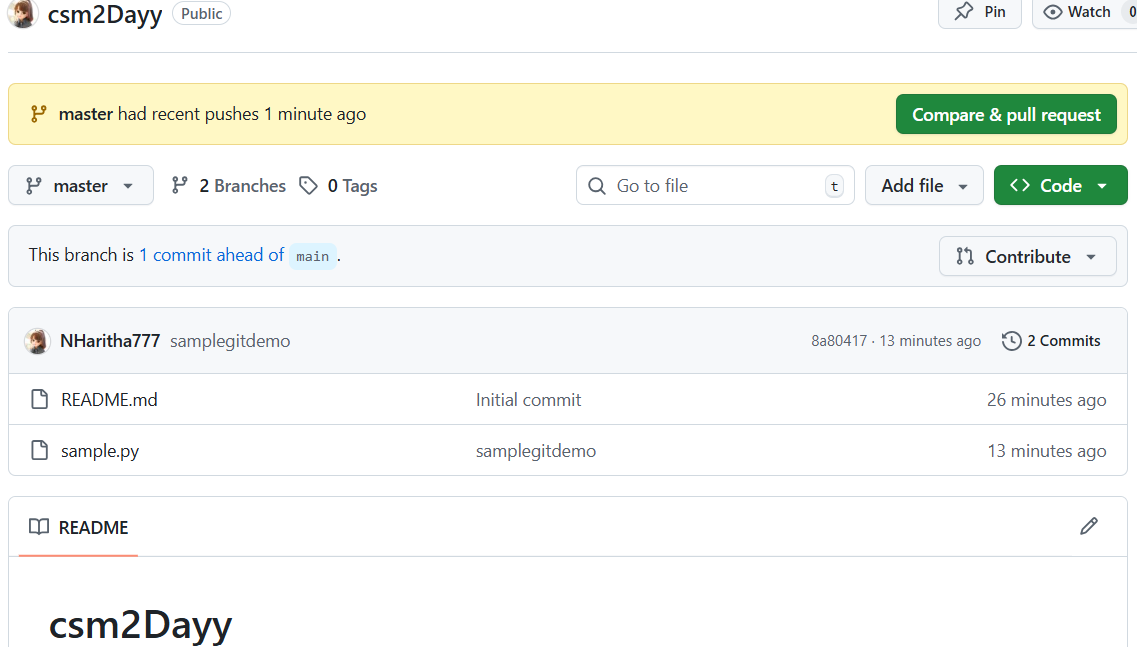
1. After the repository is created, copy the url and paste it at the command :

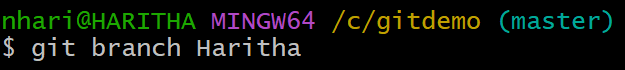
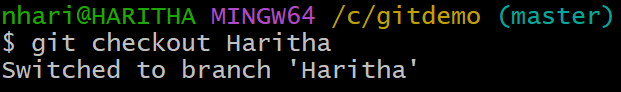
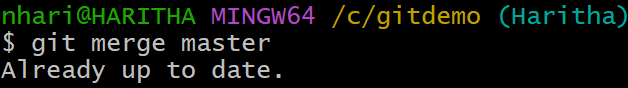
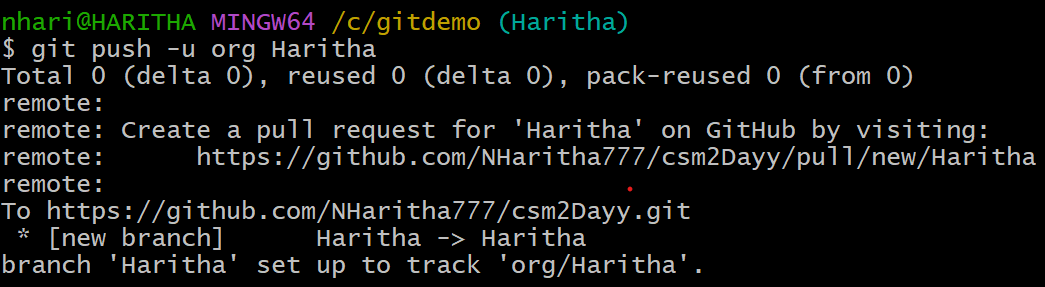
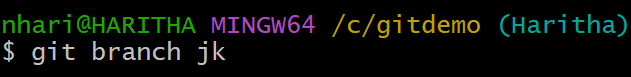
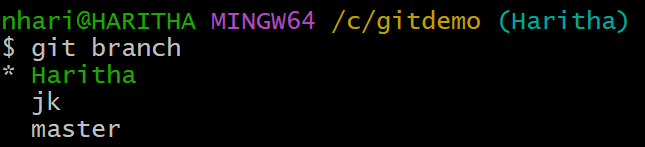


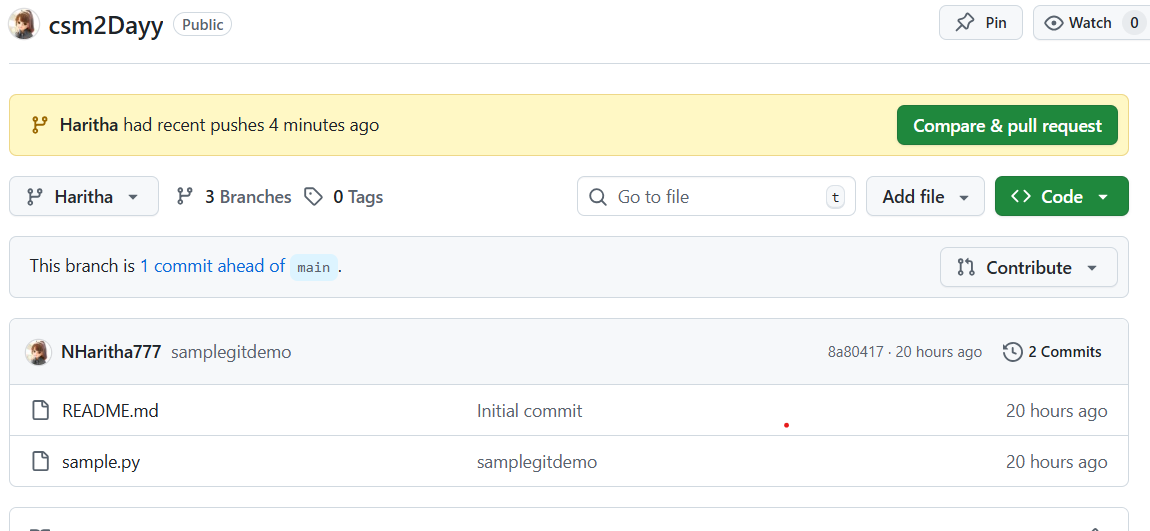
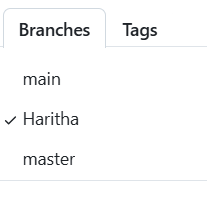
1. After push, it asks for authentication:(Give the username and password)

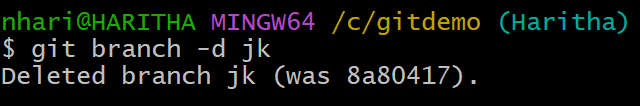
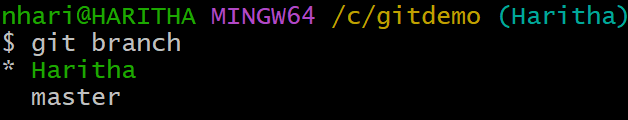
1. When we open github, we could observe that master has committed.



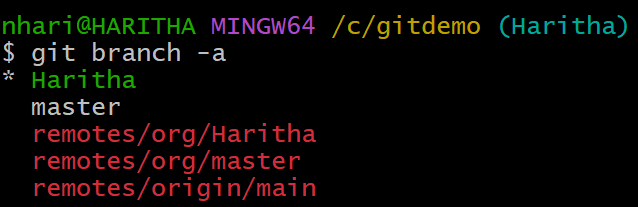
1. The sample.py file has been added to the repository.
2. To create a new branch, we use the command: git branch <branch\_name>
3. To change the branch in the directory pointing master to the newly created branch.
4. To merge the branch, we use the command: git merge master
5. To push the branch we created into git hub
6. We created one more branch named jk
7. To view all the branches present in the directory: (The current branch will be highlighted)

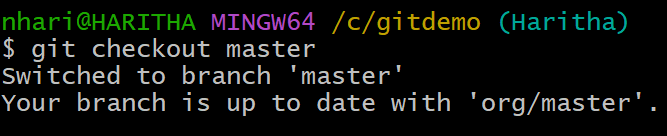
1. To delete a branch created, git branch -d <branch name>

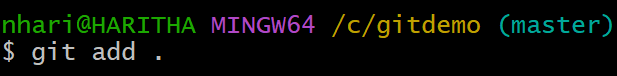
1. To view all the branches including their paths: git branch -a



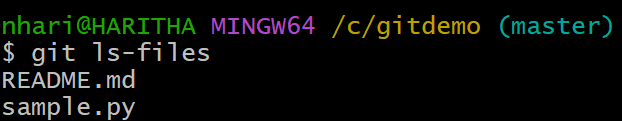
1. Now making the branch point to the master



1. To add the files: git add .



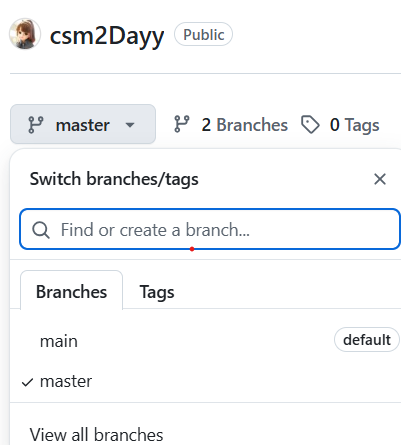
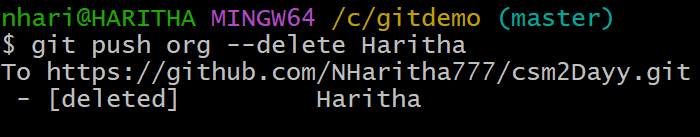
1. To view the files in the directory



1. To check the log in details: git log (when the commits are done with time and date)



1. To delete the branch from the remote repository



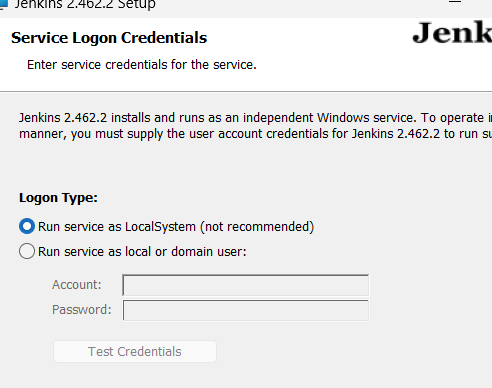
**MODULE-3**

JENKINS INSTALLATION

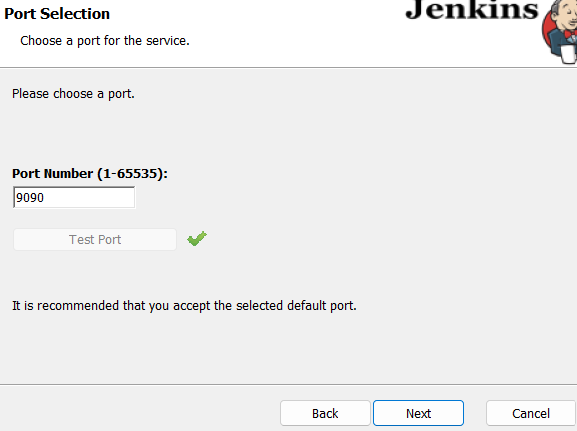
1. Open Jenkins in browser and download “Jenkins.msi”
2. The wizard opens like this and click on Next.



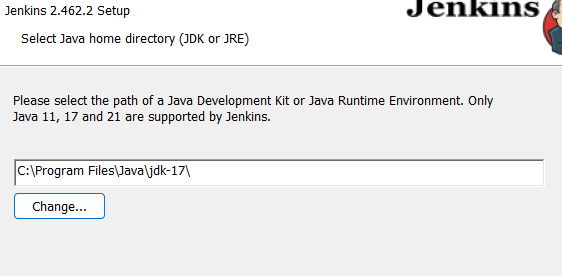
1. Select logon type as : Run service as local system



1. Change the port number as: 9090 and click on Test Port.



1. Make sure that you have java (jdk ) version 17 in the same path. If not then set the path.

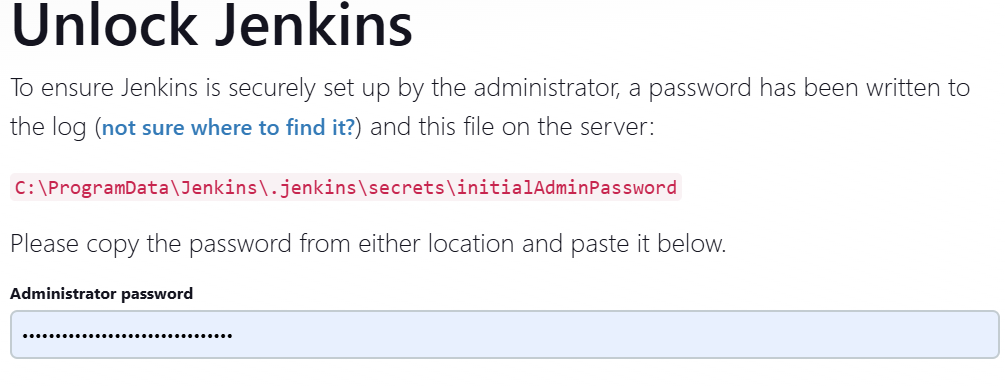


6.Click on Install and it shows like this:

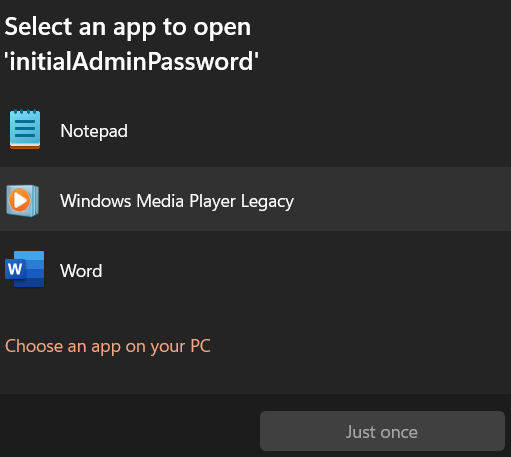
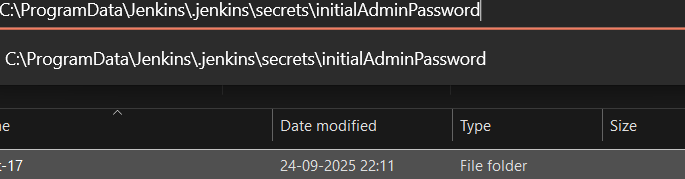


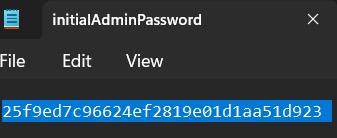
7.Go to browser and type “localhost:9090”

8. The page opens as follows:

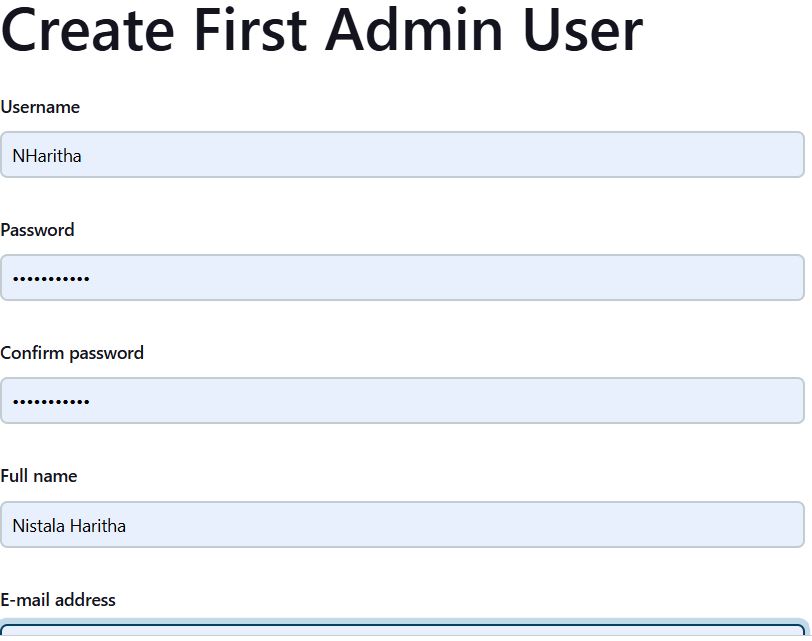


9. The path must be copied and paste the path where we have Jenkins and jdk 17 . There will be a new document to open. Open that through note pad. Copy paste the given password and paste it in the Jenkins website.



10. Then click on continue and select “Install suggested plugins”. Create Frist Admin User by giving the respective details.

The Jenkins is ready to use. It has been successfully installed.

**MODULE – 7**

DOCKER INSTALLATION

**For Windows**

System requirements:

Windows 10 64-bit: Pro, Enterprise, or Education (Build 15063 or later) or Windows 11.

Enable WSL 2 (Windows Subsystem for Linux 2).

Steps:

Download Docker Desktop for Windows from the official site:

https://www.docker.com/products/docker-desktop

Run the installer and follow the prompts.

During installation, enable WSL 2 integration if prompted.

After installation, restart your computer if required.

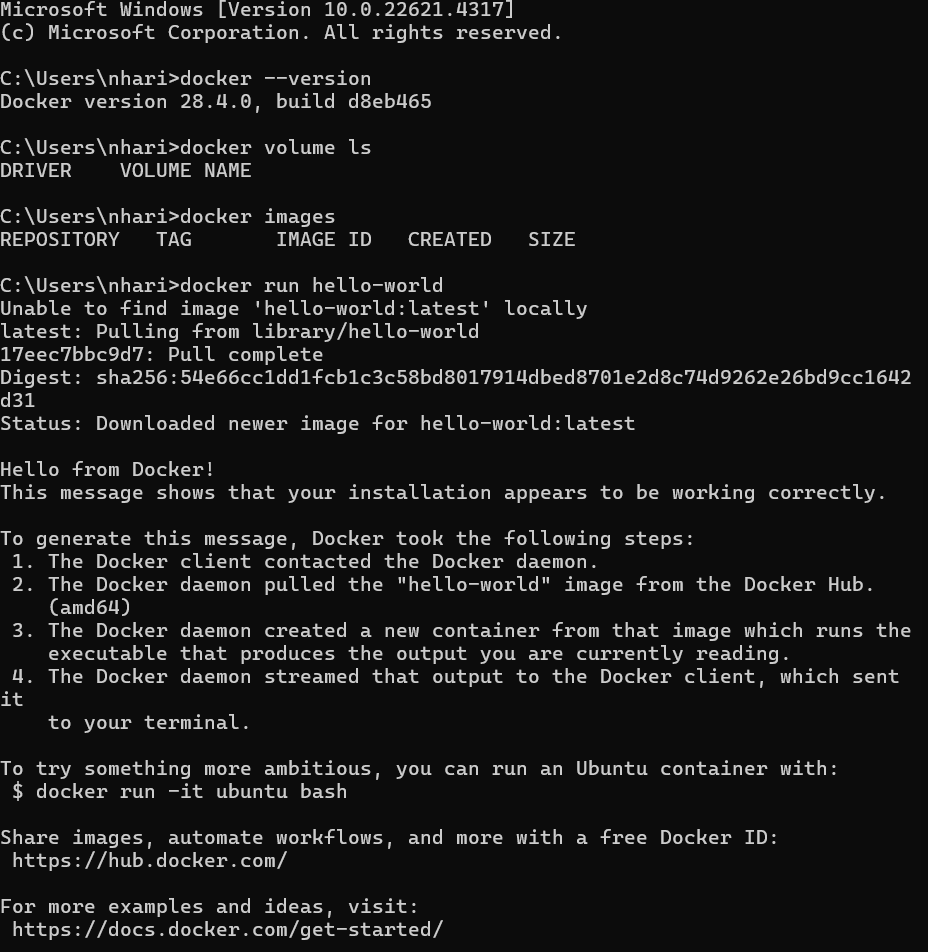
Open Docker Desktop to verify it runs.

DOCKER COMMANDS,IMAGE CREATION

>docker –version

This is used to check the version of the docker.

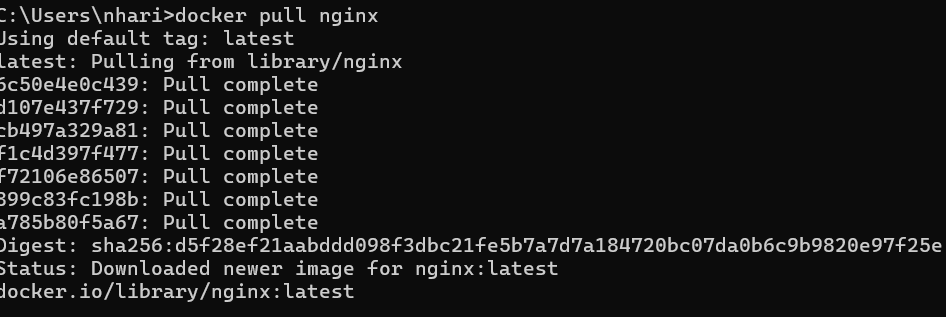
>docker run hello-world



When the command is run, we can observe a message “Hello from Docker!”. Running docker run hello-world is the simplest way to verify that Docker is installed and working correctly on your system.

>docker pull nginx

Nginx is a powerful, open-source web server software known for its high performance, scalability, and low resource consumption.



>docker run -d -p 8080:80 --name my-nginx nginx

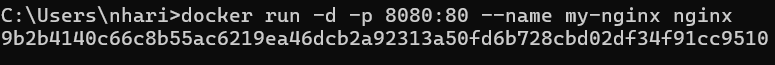
docker run: Run a new container.

-d: Run the container detached (in the background).

-p 8080:80: Map port 8080 on your host to port 80 inside the container (Nginx’s default HTTP port).

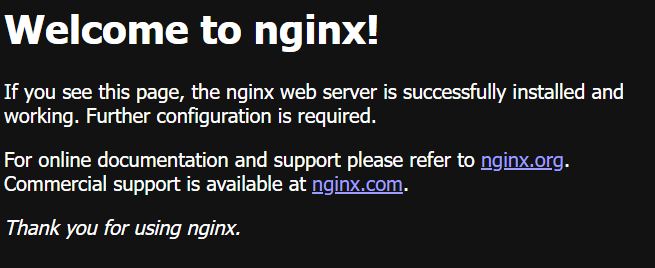
--name my-nginx: Assign the name my-nginx to the container for easier management.

nginx: Use the official nginx image from Docker Hub.



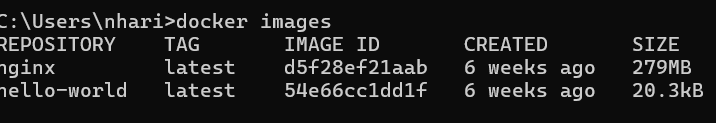
>localhost:8080

This will show you the default Nginx welcome page served by the container running on your machine.



>docker images

is used to list all Docker images stored locally on your machine.



>docker ps

Lists all running Docker containers on your system.



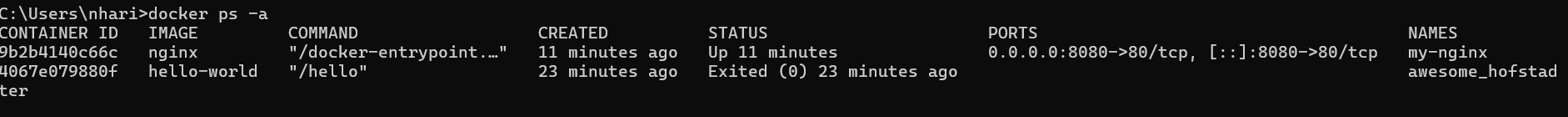
>docker ps -a

Lists all Docker containers on your system, including:

Running containers

Stopped containers

Containers that exited or failed



>docker stop my-nginx

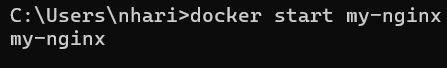
Stops the running container named my-nginx gracefully.



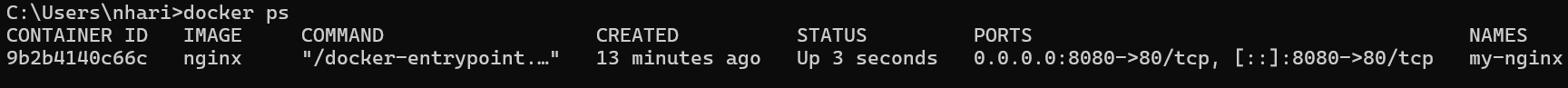
>docker ps



>docker start my-nginx



>docker ps



CUSTOM DOCKER IMAGE

>In C drive create a folder named “dockerdemo”

>In that folder create another folder named “Python image”

> OPEN VS CODE

>Open the dockerdemo folder and create a file named “app.py”. Write the code as follows:

print(“Welcome to docker tutorial”)

>Create another new file named “Dockerfile”. Write the code as follows:

FROM python

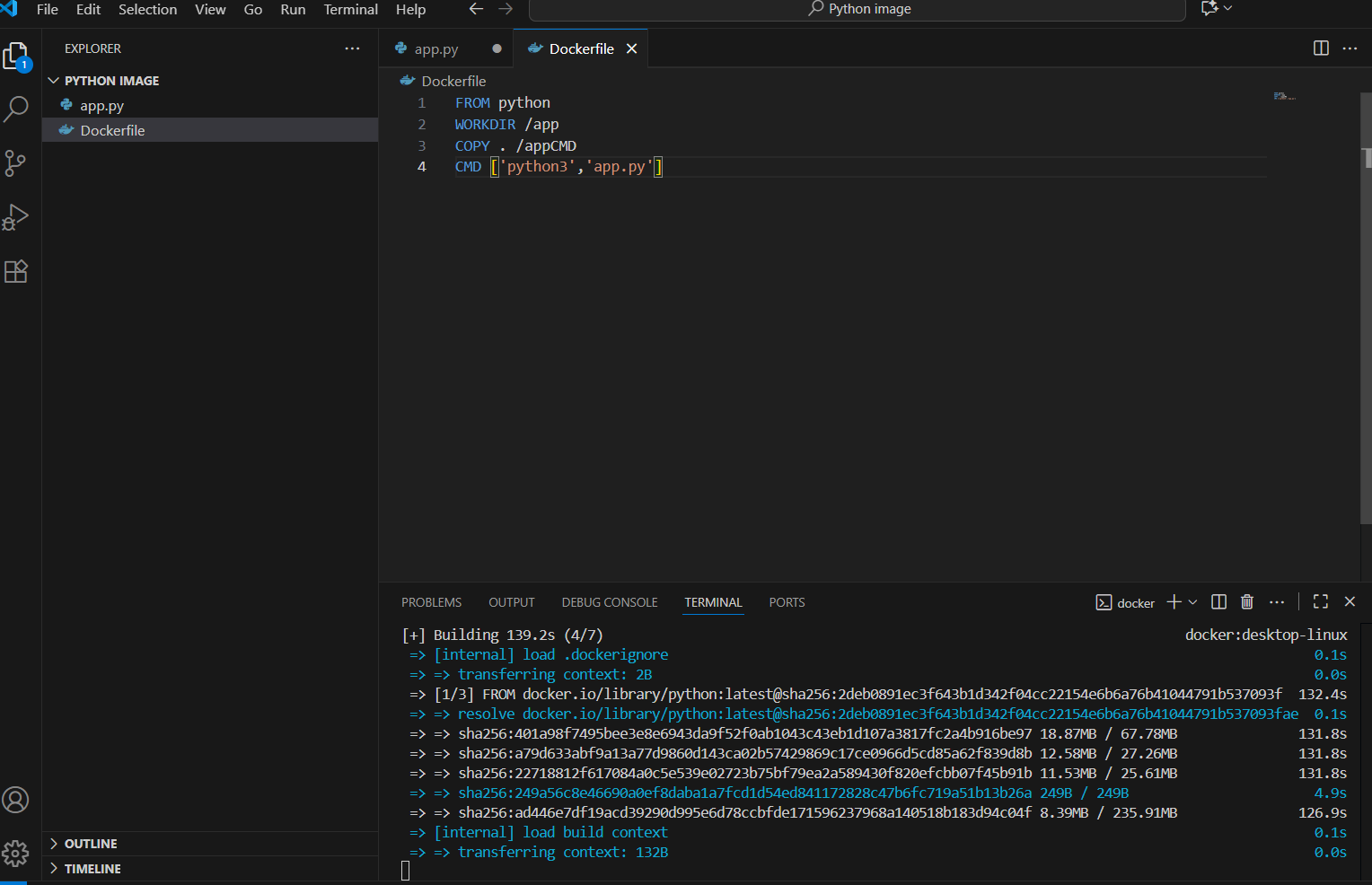
WORKDIR /app

COPY . /appCMD

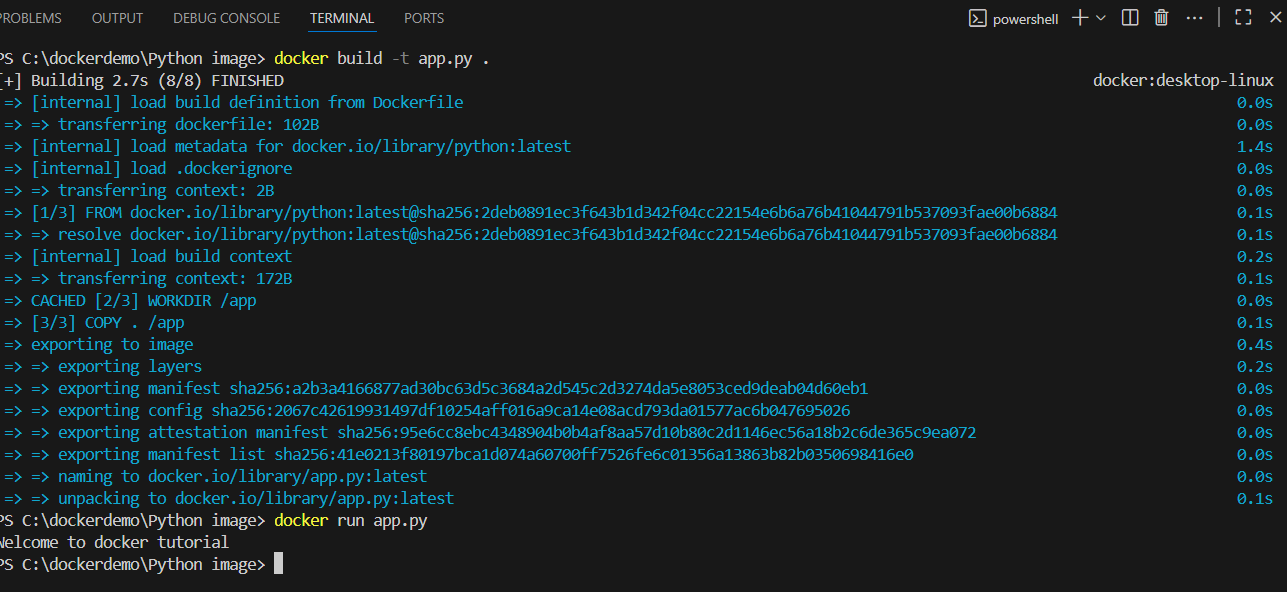
CMD ['python3','app.py']

In terminal:

* docker build -t app.py .



* docker run app.py



The message will be printed if the build is successful.