DEVOPS: SOFTWARE ARCHITECTURE LAB

Subject Code: LPCIT-115

PRACTICAL FILE

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR

THE AWARD OF THE DEGREE OF

Bachelor of technology

(Information Technology)

April, 2022

Submitted by

Parvesh Bhatt

University Roll No. 1905375

Class Roll no. 1921078

Submitted to

Prof. Harjot Kaur Gill

GURU NANAK DEV ENGINEERING COLLEGE
LUDHIANA-141006, INDIA

Contents

1 PRACTICAL 1

INSTALLATION OF GIT

1.1 Checking for git

To see if you already have Git installed, open up your terminal application

- If you're on a Mac, look for a command prompt application called "Terminal".
- If you're on a Windows machine, open the windows command prompt or "Git Bash".[Figure 4]

Once you've opened your terminal application, type git version. The output will either tell you which version of Git is installed, or it will alert you that git is an unknown command. If it's an unknown command, read further and find out how to install Git.

Figure 1: Checking git version

1.2 Install git on Windows

1. Browse to the official Git website: https://git-scm.com/downloads

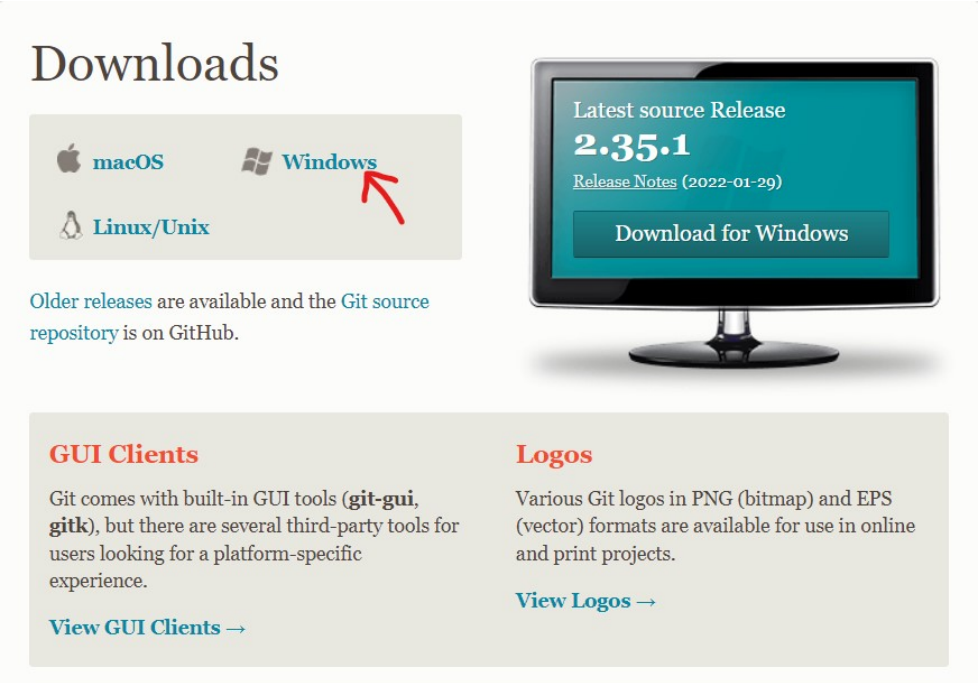

Figure 2: Install git site

2. Click the download link for Windows and allow the download to complete.

3. Browse to the download location (or use the download shortcut in your browser). Double-click the file to extract and launch the installer.

4. Allow the app to make changes to your device by clicking Yes on the User Account Control dialog that opens.

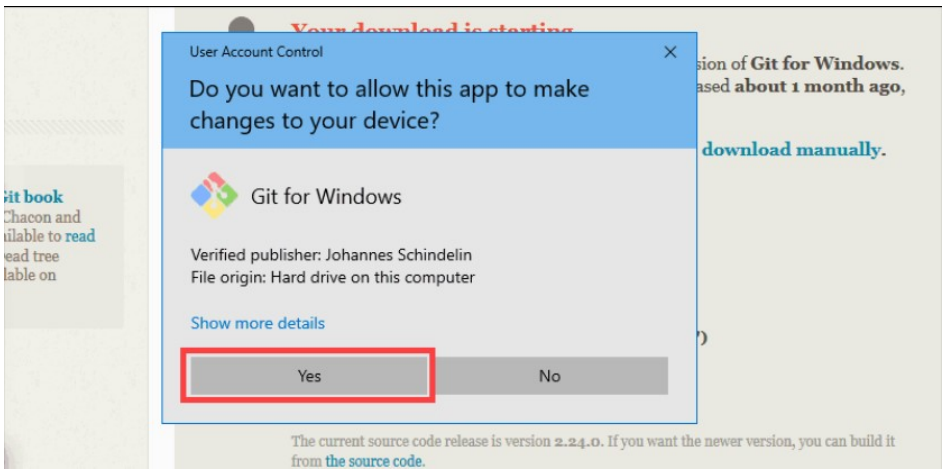

Figure 3: Install git site

5. Review the GNU General Public License, and when you're ready to install, click Next.

6. The installer will ask you for an installation location. Leave the default, unless you have reason to change it, and click Next.

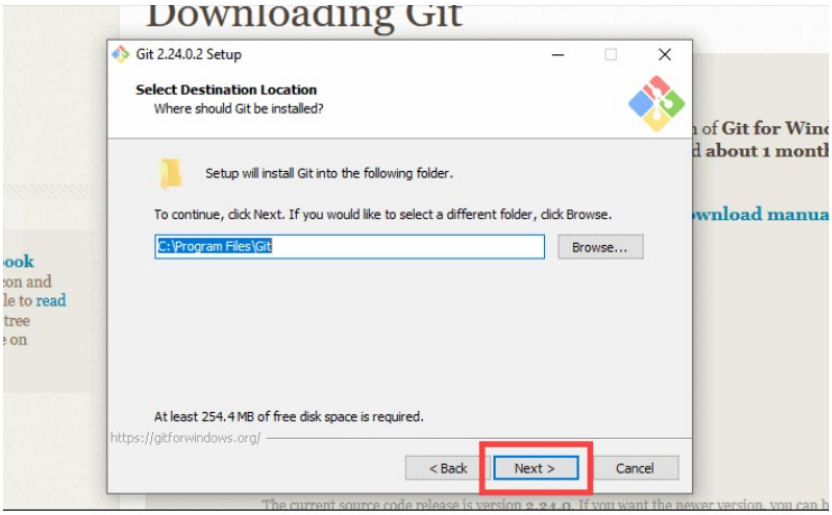

Figure 4: Select install location

7. A component selection screen will appear. Leave the defaults unless you have a specific need to change them and click Next.

8. The installer will offer to create a start menu folder. Simply click Next.

9. Select a text editor you'd like to use with Git. Use the drop-down menu to select Notepad++ (or whichever text editor you prefer) and click Next.

10. The next step allows you to choose a different name for your initial branch. The default is 'master.' Unless you're working in a team that requires a different name, leave the default option and click Next.

11. This installation step allows you to change the PATH environment. The PATH is the default set of directories included when you run a command from the command line. Leave this on the middle (recommended) selection and click Next.

Server Certificates, Line Endings and Terminal Emulators

12. The installer now asks which SSH client you want Git to use. Git already comes with its own SSH client, so if you don't need a specific one, leave the default option and click Next.

13. The next option relates to server certificates. Most users should use the default. If you're working in an Active Directory environment, you may need to switch to Windows Store certificates. Click Next.

14. The next selection converts line endings. It is recommended that you leave the default selection. This relates to the way data is formatted and changing this option may cause problems. Click Next.

15. Choose the terminal emulator you want to use. The default MinTTY is recommended, for its features. Click Next.

16. The installer now asks what the git pull command should do. The default option is recommended unless you specifically need to change its behavior. Click Next to continue with the installation.

17. Next you should choose which credential helper to use. Git uses credential helpers to fetch or save credentials. Leave the default option as it is the most stable one, and click Next.

18. select additional customization options and click next.

19. Once the installation is complete, tick the boxes to view the Release Notes or Launch Git Bash, then click Finish.

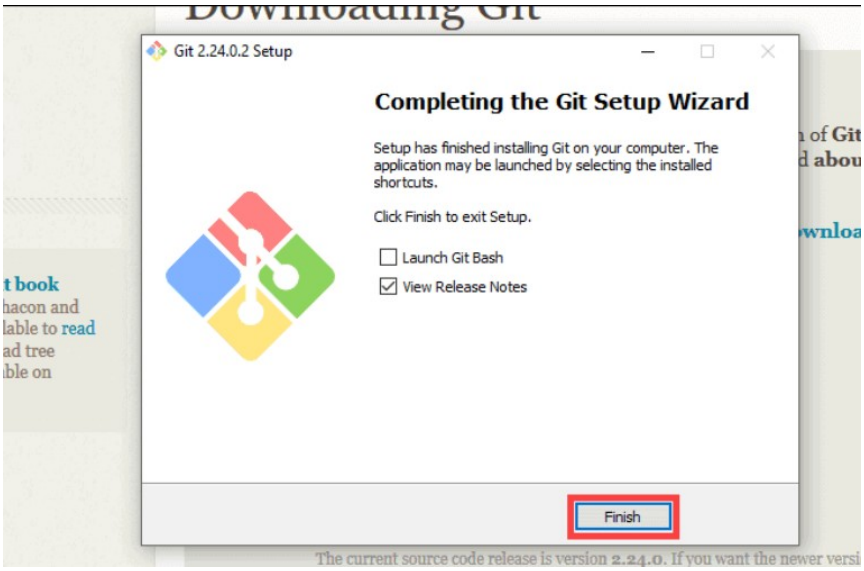

Figure 5: Completing the git set-up wizard

1.3 Install Git on Linux

You can install Git on Linux through the package management tool that comes with your distribution.

1. Git packages are available using apt.

2. It's a good idea to make sure you're running the latest version. To do so, Navigate to your command prompt shell and run the following command to make sure everything is up-to-date: sudo apt-get update.

3. To install Git, run the following command: sudo apt-get install git-all.

4. Once the command output has completed, you can verify the installation by typing: git version.

2 PRACTICAL 2

CREATE AN ACCOUNT ON GITHUB

1. **Go to https://github.com/join in a web browser**. You can use any web browser on your computer, phone, or tablet to join.

2. **Enter your personal details**. In addition to creating a username and entering an email address, you'll also have to create a password. Your password must be at leas t 15 characters in length or at least 8 characters with at least one number and lower-case letter.

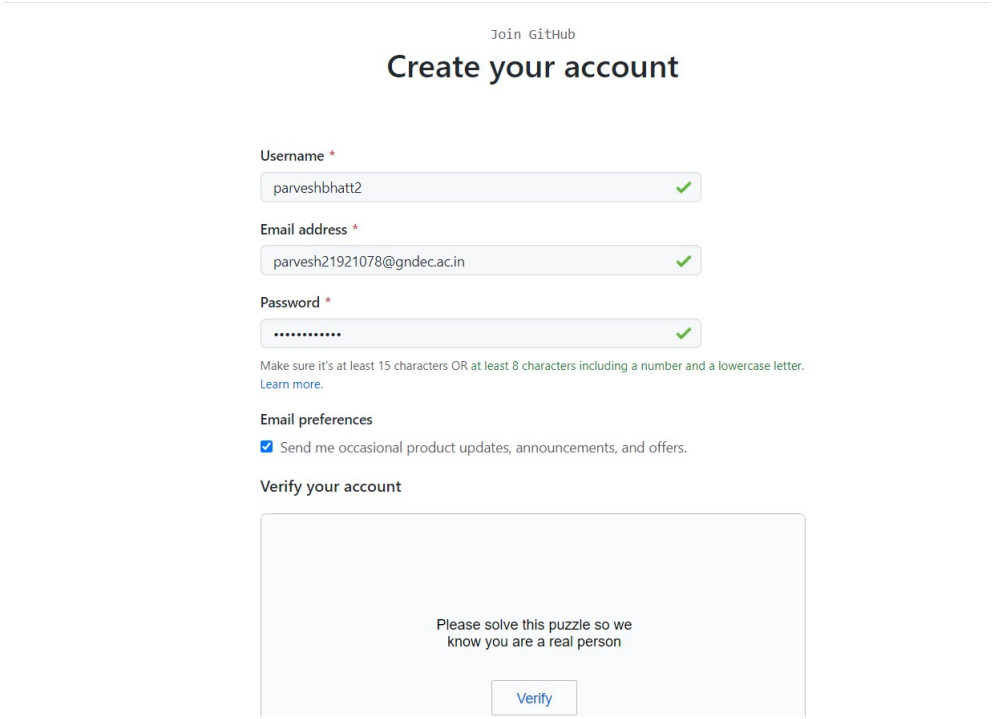

Figure 6: Enter your personal details

3. **click the green Create an account button**. It's below the form.

4. **Complete the CAPTCHA puzzle**. The instructions vary by puzzle, so just follow the on screen instructions to confirm that you are a human.

5. **Click the Choose button for your desired plan**. Once you select a plan, GitHub will send an email confirmation message to the address you entered.

6. **Click the Verify email address button in the message from GitHub**. This confirms your email address and returns you to the sign up process.

7. **Review your plan selection and click Continue**. You can also choose whether you want to receive updates from GitHub via email by checking or unchecking the "Send me updates" box.

8. **your preferences and click Submit** GitHub displays a quick survey that can help you tailor your experience to match what you're looking for. Once you make your selection, you'll be taken to a screen that allows you to set up your first repository.

   Congratulation your account on GitHub is created Successfully. Now you are free to do your work.

3 PRACTICAL 3

CREATE REPOSITORY USING GIT/ GITHUB

GITHUB

1. In the upper-right corner of any page, use the drop-down menu, and select New repository.

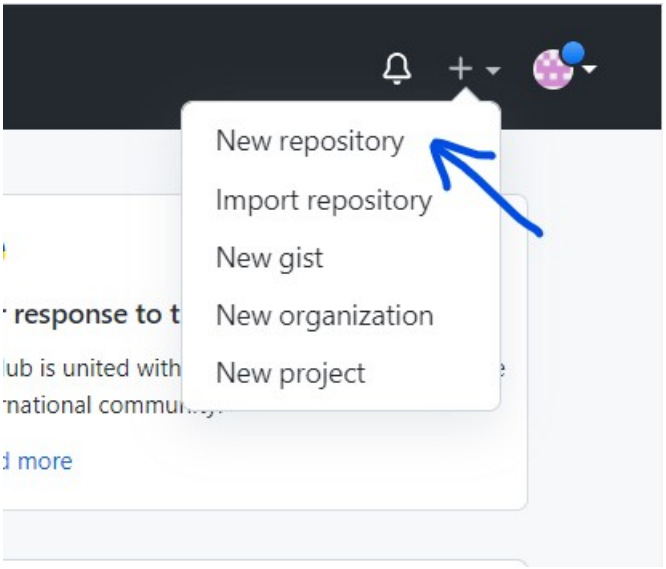

Figure 7: Select new repository

2. Type a short, memorable name for new repository. For example, "hello-devops".

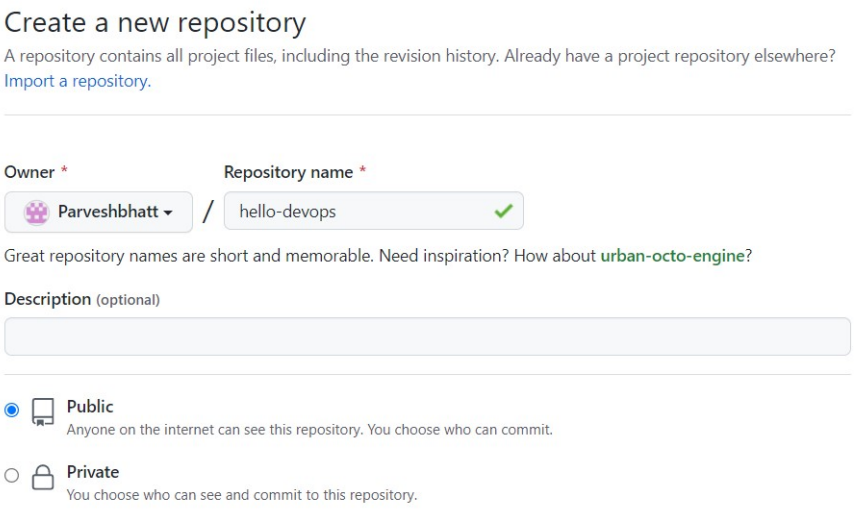

Figure 8: Name the repository

3. Optionally, add a description of our repository. For example, "First devops repository."

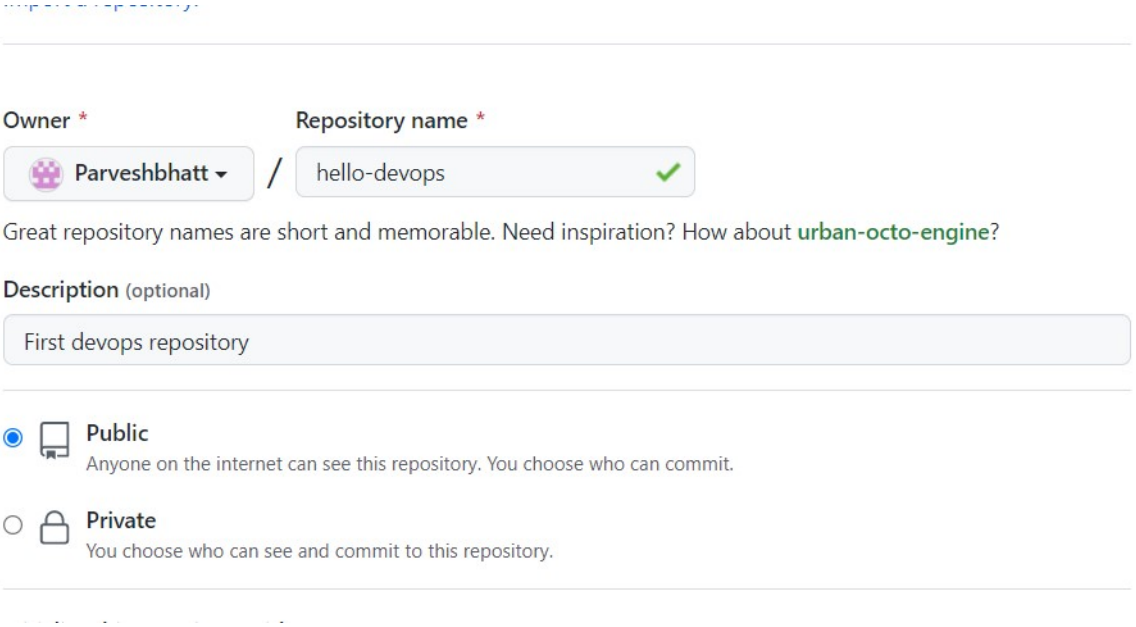

Figure 9: Name the repository

4. Choose a repository visibility. For more information, see "About repositories".
5. Select Initialize this repository with a README.

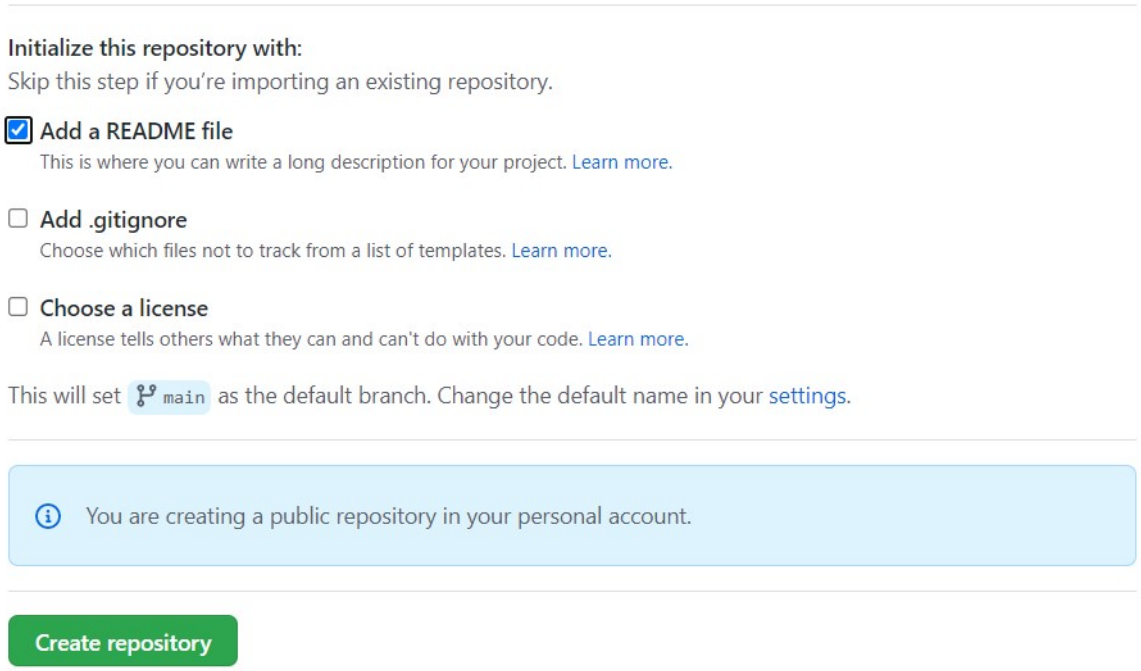

Figure 10: Select Add a README file

6. Click Create repository.

4 PRACTICAL 4

CREATE/DELETE/MERGE BRANCHES

4.1 CREATE BRANCH

A new branch can be created with the help of the *git branch* command.

$ git branch fix-code

Figure 11: Create branch

where fix-code is the name of new branch.

4.2 MERGE BRANCH

Git allows you to merge the other branch with the currently active branch. You can merge two branches with the help of *git merge* command. Below command is used to merge the branches:

For merging a branch to master or any other branch, switch over to that branch using *git checkout* command.

$ git checkout master

Merge the fix-code branch to master/main branch.

$ git merge fix-code

Figure 12: Merge branch

The fix-code branch is now merged with master branch .

4.3 DELETE BRANCH

For deleting specified branch, it is a safe operation. In this command, Git prevents you from deleting the branch if it has unmerged changes. Below is the command to do this.

$ git branch -d fix-code

Figure 13: Delete local branch

Attempting to delete an unmerged branch results in following error message.

error: The branch 'new' is not fully merged. If you are sure you want to delete it, run 'git branch -D new'.

To delete unmerged branch use same command with -D flag.

$ git branch -D branchname

For deleting a remote branch from Git desktop application. Below command is used to delete a remote branch:.

$ git push origin -delete remote-branch

5 PRACTICAL 5

INSTALL DOCKER

Following are the steps to install Docker: -

1. Browse to official Docker website https://docs.docker.com/desktop/windows/install/
2. Double-click Docker Desktop Installer.exe to run the installer. If you haven't already downloaded the installer (Docker Desktop Installer.exe), you can get it from Docker Hub. It typically downloads to your Downloads folder, or you can run it from the recent downloads bar at the bottom of your web browser.

Figure 14: docker website

3. When prompted, ensure the Use WSL 2 instead of Hyper-V option on the Configuration page is selected or not depending on your choice of back-end.
4. Follow the instructions on the installation wizard to authorize the installer and proceed with the install.

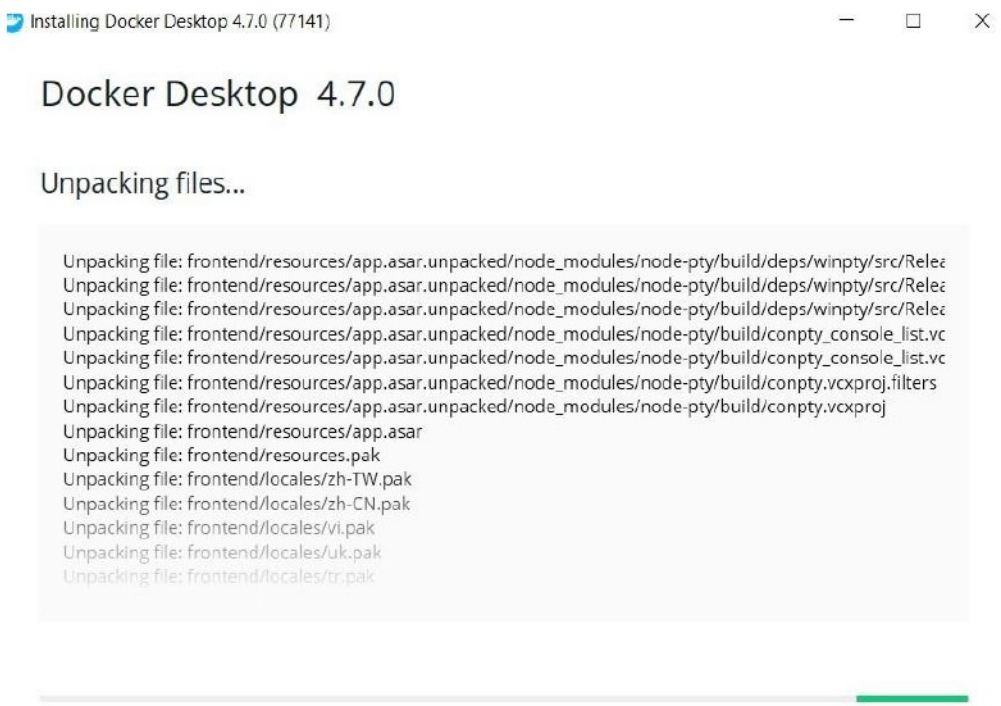

Figure 15: Installation wizard

5. When the installation is successful, click Close to complete the installation process.

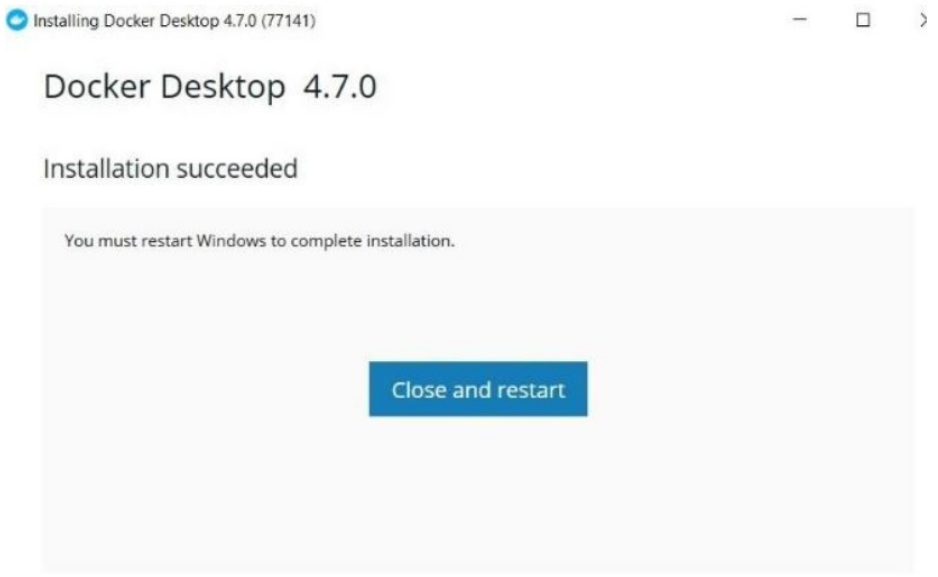

Figure 16: Installation Complete

6. If your admin account is different to your user account, you must add the user to the docker-users group. Run Computer Management as an administrator and navigate to Local Users and Groups ¿ Groups ¿ dockerusers. Right-click to add the user to the group. Log out and log back in for the changes to take effect

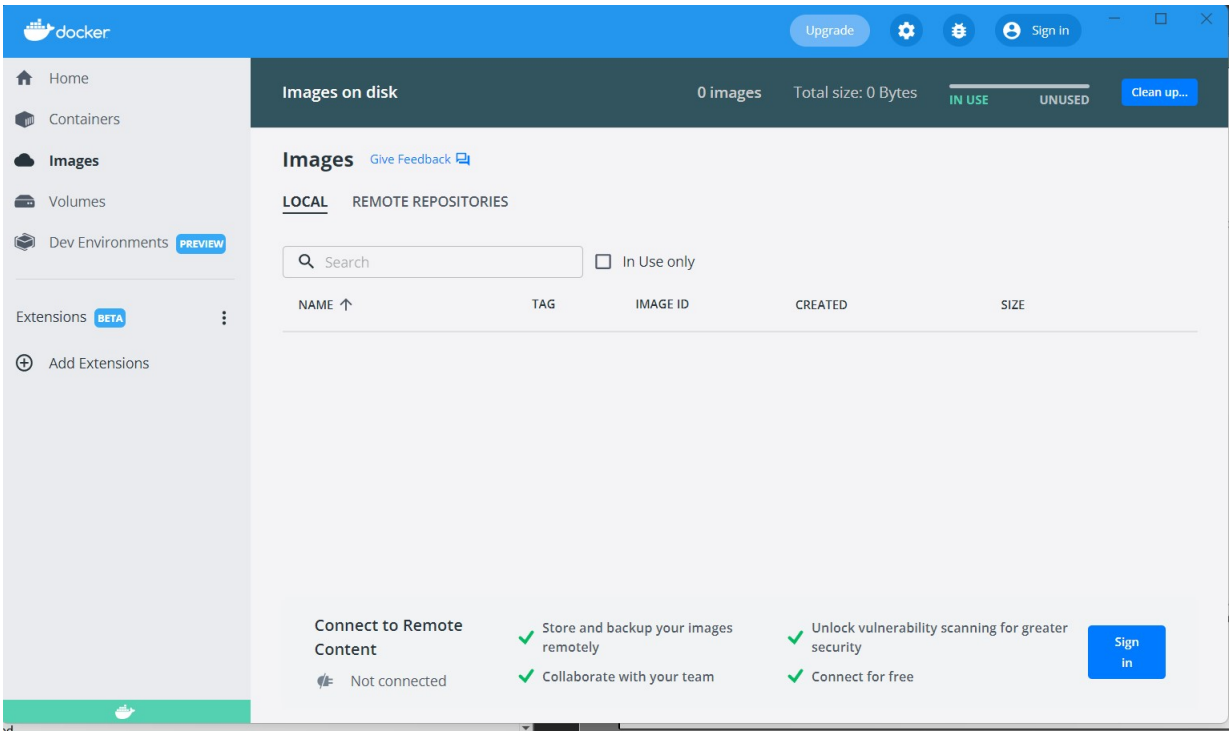

Figure 17: Docker Desktop

6 PRACTICAL 6

DEPLOY NGINX WEB SERVER IMAGE ON DOCKER

Steps for getting the Docker container for nginx up and running.

1. The first step is to pull the image from Docker Hub. When you log into Docker Hub, you will be able to search and see the image for nginx as shown below.

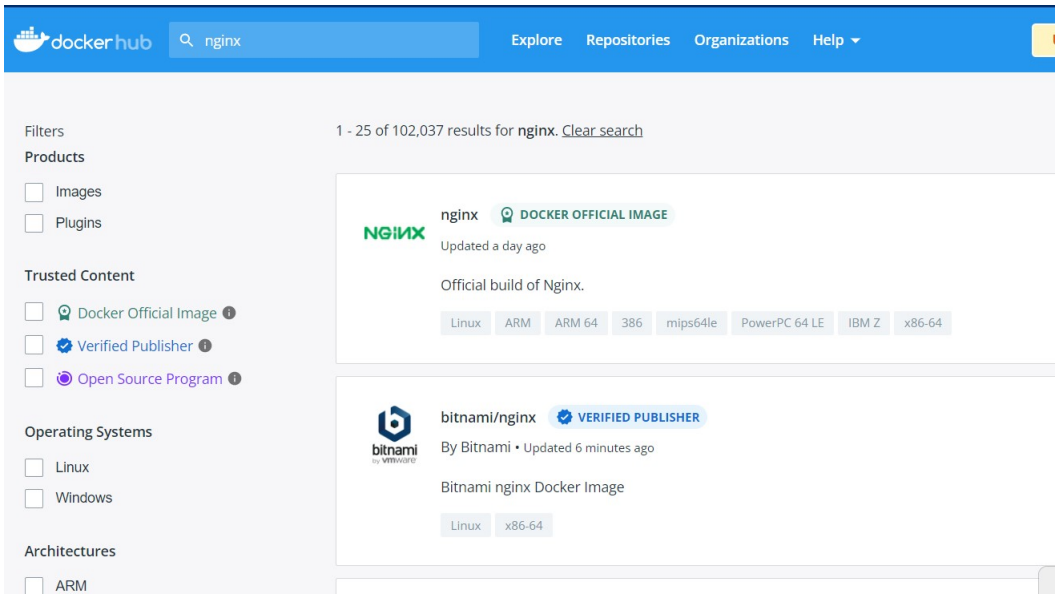

Figure 18: Nginx image on Docker hub

2. On the Docker Host, use the Docker pull command to download the latest nginx image from Docker Hub.

```
PS C:\Users\PARVESH> docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
214ca5fb9032: Pull complete
66eec13bb714: Pull complete
17cb812420e3: Pull complete
56fbf79cae7a: Pull complete
c4547ad15a20: Pull complete
d31373136b98: Pull complete
Digest: sha256:2d17cc4981bf1e22a87ef3b3dd20fbb72c3868738e3f307662eb40e2630d4320
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
PS C:\Users\PARVESH>
```

Figure 19: Pull nginx image

3. Now we have to run the NGINX image such that it will expose the Docker container port to the network port.

```
PS C:\Users\PARVESH> docker run -it --rm -d -p 8080:80 --name=webserver nginx
5793382f8ab1bff97b80a901d66f58605d6df66291caa601e63470f54a4d5bcb
PS C:\Users\PARVESH>
```

Figure 20: Run nginx image

Here, we have used the -i and -t options to run the container interactively, the –rm option to delete the container on exit, -d option to run the container in the background or detached mode, and the –name option to provide a name to the container. Finally, we have provided the name of the Docker image that we want to pull. We have also used the -p option to publish port 8080 of the container to port 80 of the host machine. Let's try to execute this command inside the terminal.

4. Let's verify by listing all the active containers.

```
PS C:\Users\PARVESH> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                   NAMES
5793382f8ab1   nginx     "/docker-entrypoint.…"   12 minutes ago   Up 12 minutes   0.0.0.0:8080->80/tcp   webserver
PS C:\Users\PARVESH>
```

Figure 21: Listing running containers

5. To see that the container is actively running, navigate to the link localhost:8080 inside a browser, we will find the welcome page of the Nginx web server.

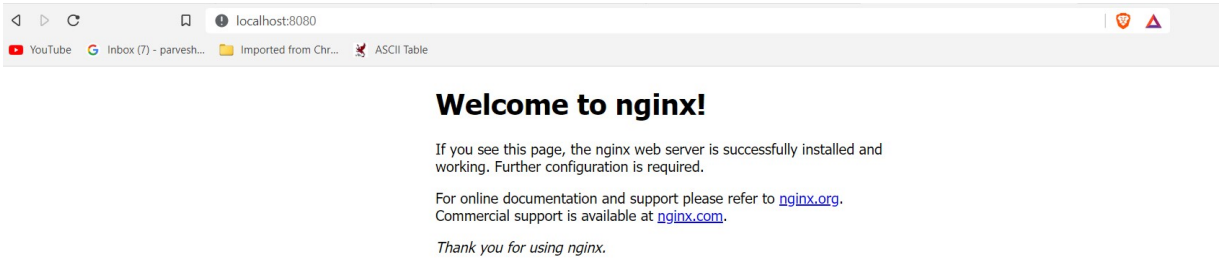

Figure 22: Nginx welcome page

6. To stop the container use the 'docker stop' command

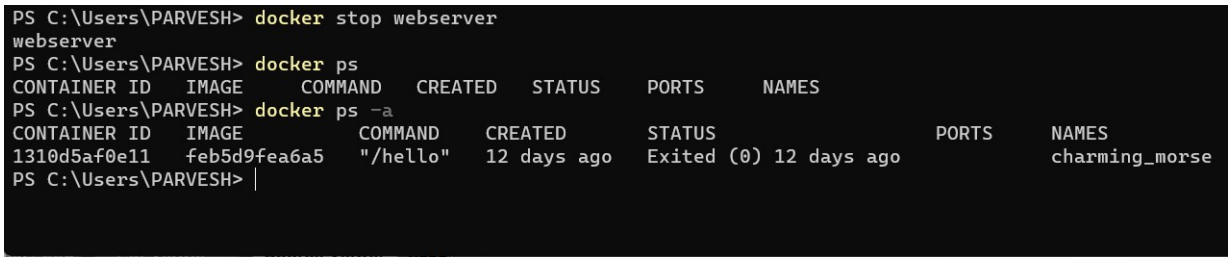

Figure 23: Stop Nginx

7 PRACTICAL 7

DEPLOY APACHE WEB SERVER IMAGE ON DOCKER

Steps for getting the Docker container for Apache Web Server Image on Docker.

1. Log into Docker Hub, you will be able to search and see the image for apache as httpd.

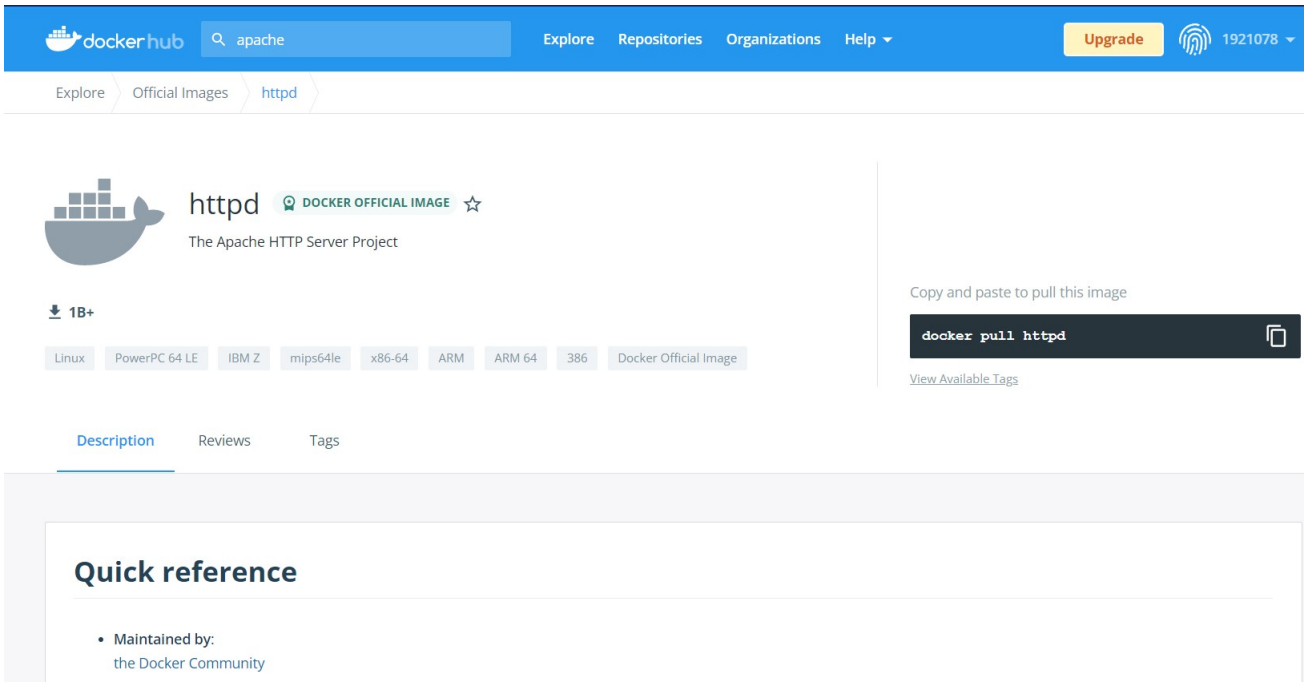

Figure 24: httpd Image On Docker Hub

2. Pull the Docker image, which contains Apache called httpd, by running the docker pull command below. This command will download or pull the Apache image from the Docker registry, as shown below.

```
PS C:\Users\PARVESH> docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
214ca5fb9032: Already exists
7cf31a2eeec6: Pull complete
bf666e57b9f2: Pull complete
c15a4e94ae6b: Pull complete
dc25474c7f97: Pull complete
Digest: sha256:2d1f8839d6127e400ac5f65481d8a0f17ac46a3b91de40b01e649c9a0324dea0
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
PS C:\Users\PARVESH>
```

Figure 25: Pull httpd

3. Next, confirm the downloaded image by running the docker images command below to list all images available on your computer.

```
PS C:\Users\PARVESH> docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
nginx        latest    de2543b9436b   7 days ago    142MB
httpd        latest    c58ef9bfbb57   2 weeks ago   144MB
PS C:\Users\PARVESH>
```

Figure 26: Check Images

4. Invoke the docker run command to create a new container based on your downloaded Apache Docker image. The docker run command then returns the unique container ID of the container you've just created. Save this container id in the highlighted box below to the future if you wish to delete or remove the container.

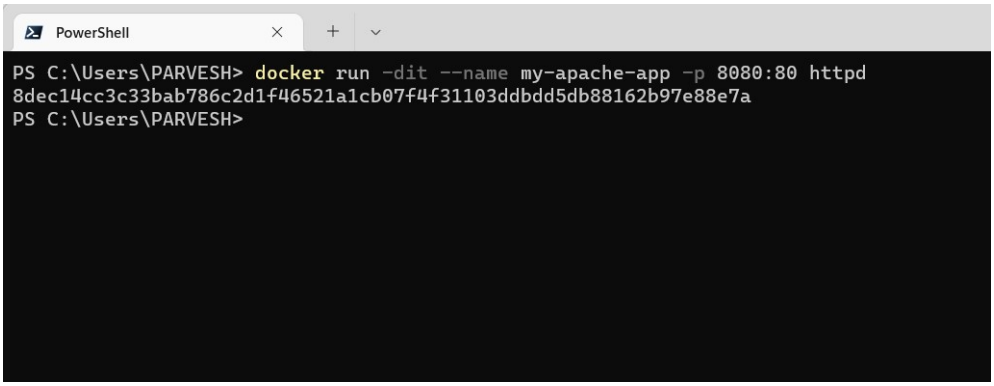

Figure 27: Run Server

5. Once the Apache container is running, verify it by access the Apache web interface by navigating to Public-Ip-address:80 using any web browser. If you can see the same message, as you can see below, then you've successfully started your Apache Docker container.

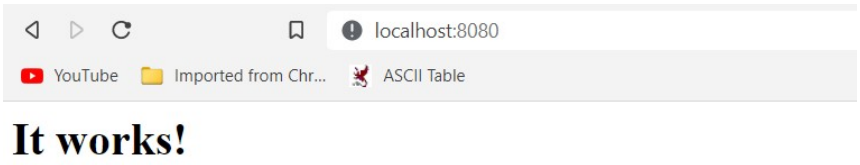

Figure 28: Default output of Apache web server

8 PRACTICAL 8

CREATE CUSTOM PAGE USING WEB SERVER

1. First we create a running container. The docker create command will create a new container. Here we have requested a new container named baseimg with port 80 exposed to localhost. We are using nginx as a base image for the container. If you don't have the nginx image in your local docker image repository, it will download automatically.

```
PS C:\Users\PARVESH> docker create --name base_img -p 8080:80 nginx
9f14980e1684f701e820ac4b5e225f423907f394acd48e8c136cdf5ee0639501
PS C:\Users\PARVESH>
```

Figure 29: Creating a base container

2. If you look at the list of images on your system, you will now see the nginx:alpine image.

```
PS C:\Users\PARVESH> docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
nginx        latest    de2543b9436b   7 days ago    142MB
httpd        latest    c58ef9bfbb57   2 weeks ago   144MB
PS C:\Users\PARVESH>
```

Figure 30: Inspect Images

3. Note here that the container is not running, so you won't see it in the container list unless you use the -a flag (-a is for all).

```
PS C:\Users\PARVESH> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS                      PORTS     NAMES
9f14980e1684   nginx          "/docker-entrypoint.…"   3 minutes ago    Created                               base_img
8dec14cc3c33   httpd          "httpd-foreground"       44 minutes ago   Exited (0) 27 minutes ago             my-apache-app
1310d5af0e11   feb5d9fea6a5   "/hello"                 13 days ago      Exited (0) 13 days ago                charming_morse
PS C:\Users\PARVESH>
```

Figure 31: Inspect Container

4. Start the container by running the following command.

```
PS C:\Users\PARVESH> docker start base_img
base_img
PS C:\Users\PARVESH>
```

Figure 32: Start the container

Now visit http://localhost with your browser. You will see the default "Welcome to nginx!" page. We are now running an nginx container.

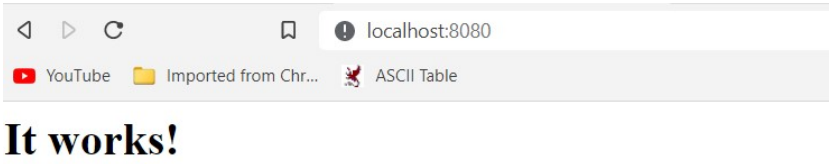

Figure 33: Container Running

5. Let's create a new index.html file and copy it onto the running container. Using an editor on your machine, create an index.html file in the same directory that you have been running Docker commands from. Then paste the following HTML into it:

```
<html>
        <head>
                <title>Hello Page</title>
        </head>

        <body>
                <h1>Hello world</h1>
        </body>

</html>
```

Figure 34: Html File

6. Then save the file and return to the command line. We will use the docker cp command to copy this file onto the running container.

```
PS C:\Users\PARVESH> docker cp "D:\index.html" base_img:/usr/share/nginx/html/index.html
PS C:\Users\PARVESH>
```

Figure 35: Copy file on running container

7. Now reload your browser or revisit http://localhost. You will see the message "Hello World!" in place of the default nginx welcome page.

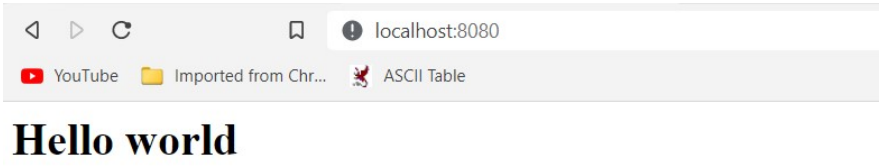

Figure 36: Modified page

9 PRACTICAL 9

CREATE CUSTOM IMAGE

Steps to Create a custom image

1. First we have to create a directory named my-app and move into that directory and create a new empty file (Dockerfile).
2. write the following commands in Dockerfile.

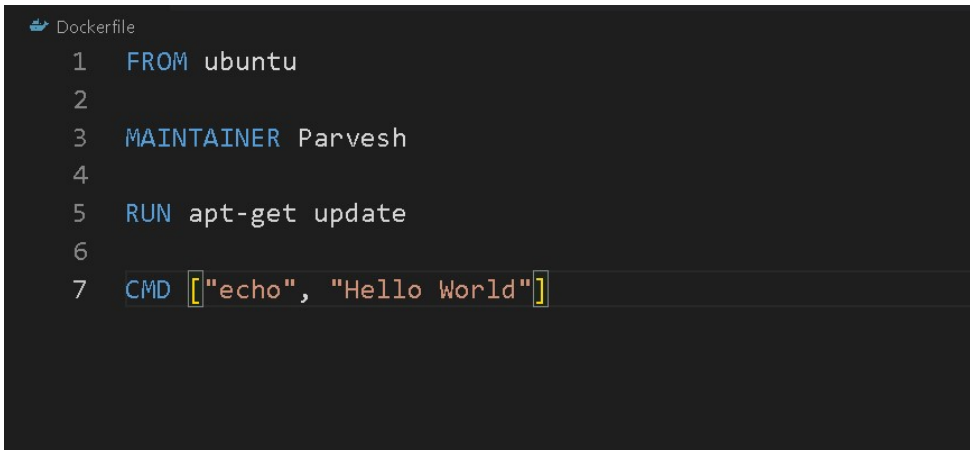

Figure 37: Dockerfile

3. Save and exit the file.
4. You can check the content of the file by using the cat command.

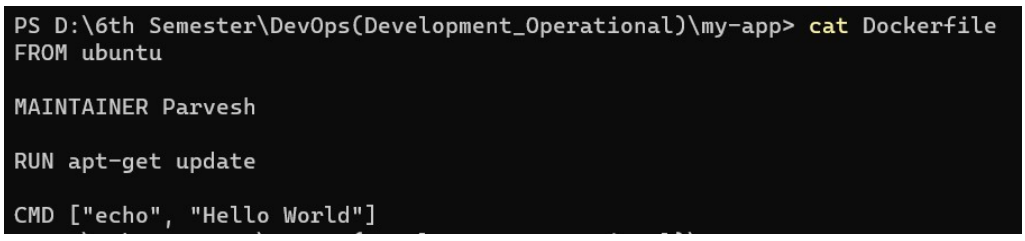

Figure 38: Checking content of Dockerfile

5. The basic syntax used to build an image using a Dockerfile is: ***docker build [OPTIONS] PATH — URL — -***
6. Build a docker image,If you are already in the directory where the Dockerfile is located, put a . instead of the location. By adding the -t flag, you can tag the new image with a name which will help you when dealing with multiple images.

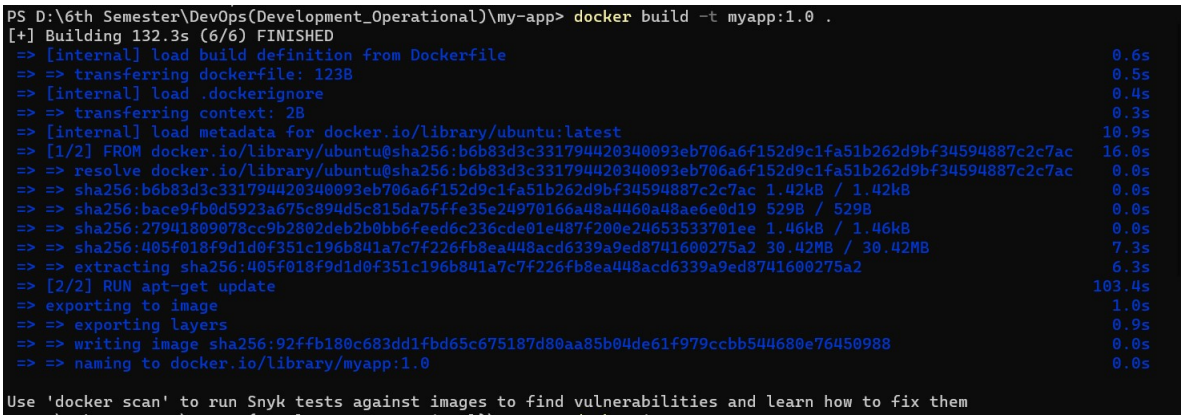

Figure 39: Building Image

7. Once the image is successfully built, you can verify whether it is on the list of local images with the command.

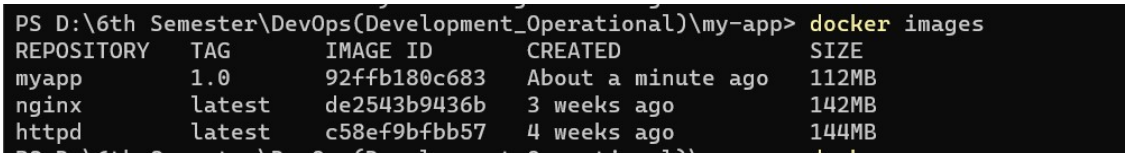

Figure 40: Verify image

8. Launch a new Docker container based on the image you created in the previous steps.

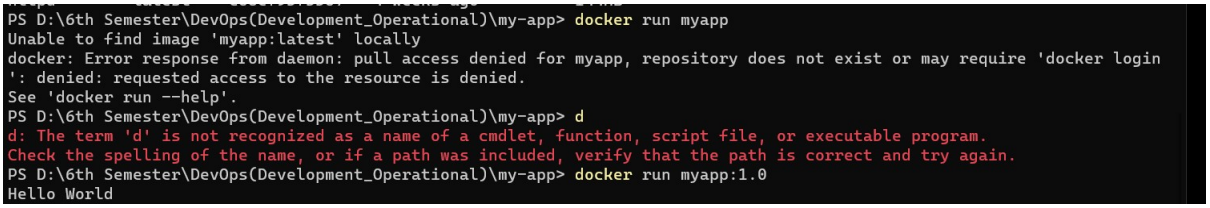

Figure 41: Launching Docker container

10 PRACTICAL 10

PUSH CUSTOM IMAGE TO DOCKER HUB

Steps to push a custom image to Docker hub are: -

1. Docker images are pushed to Docker Hub through the docker push command. A single Docker Hub repository can hold many Docker images (stored as tags).
2. To create a repository, sign into Docker Hub, click on Repositories then Create Repository.

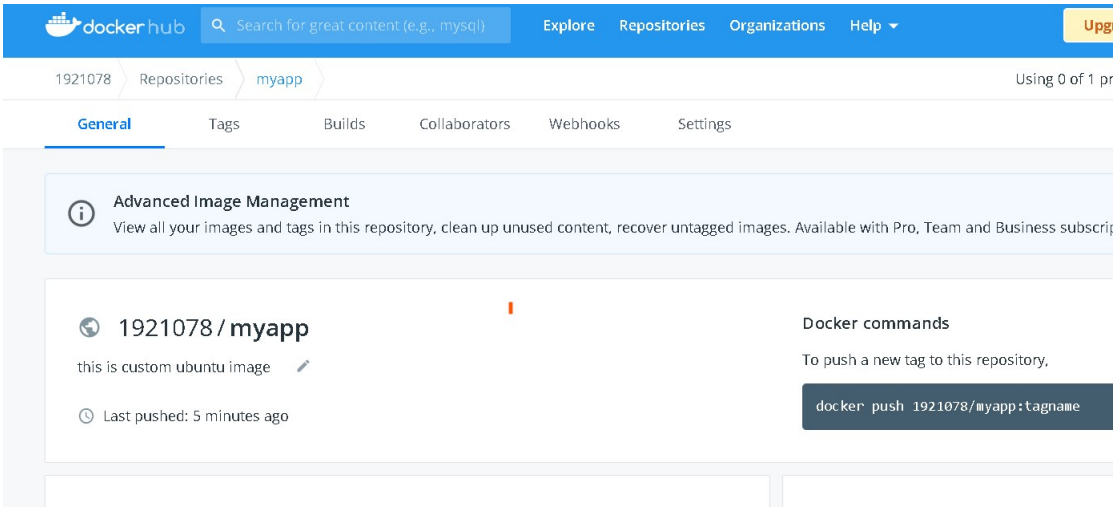

Figure 42: Create Repository

3. To push an image to Docker Hub, you must first name your local image using your Docker Hub username and the repository name that you created through Docker Hub on the web.
4. Name your local images using re-tagging an existing local image $docker tag < existing-image >< hub-user > / < repo-name > [:< tag >]$

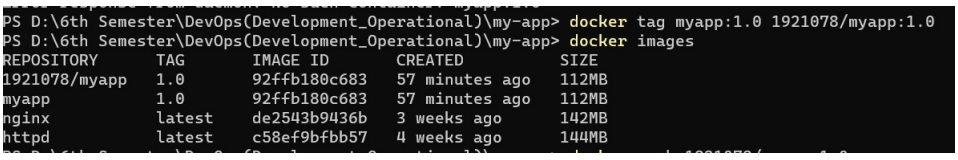

Figure 43: Re-tagging Repository

5. Push this repository to the registry designated by its name or tag.

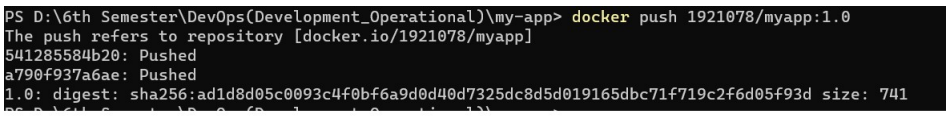

Figure 44: Push repository

6. The image is then uploaded and available for use by your teammates and/or the community.

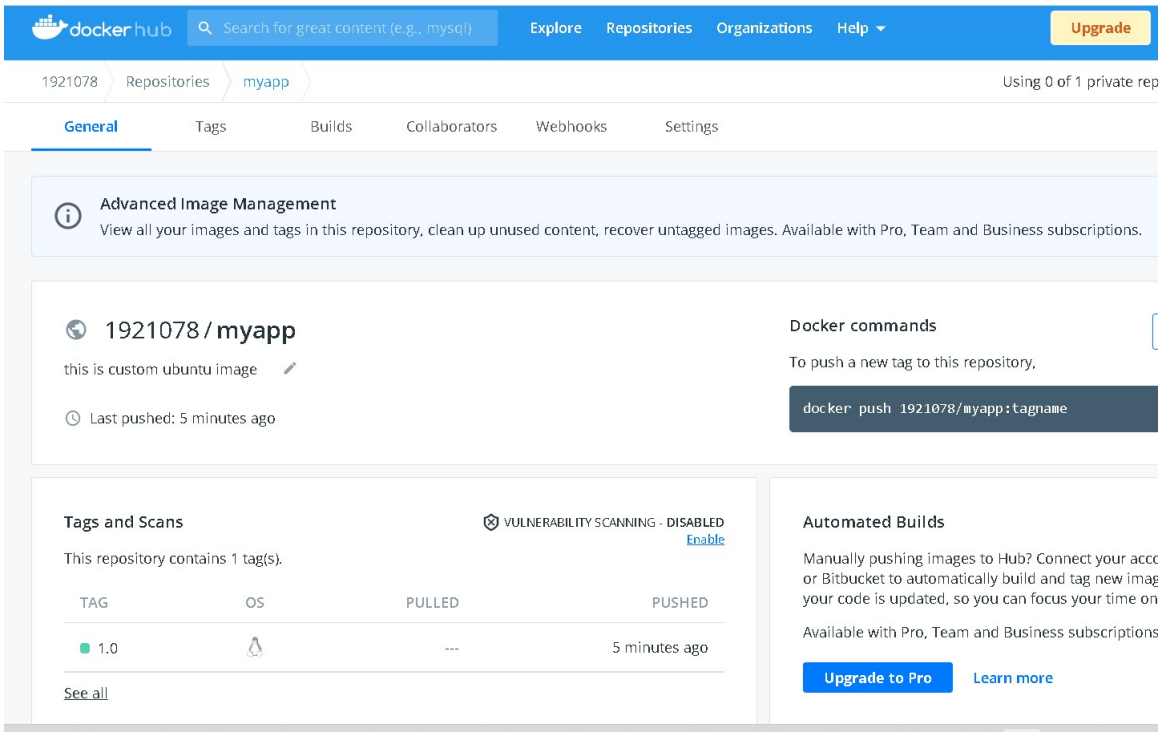

Figure 45: repository(Image) on Docker hub

11 PRACTICAL 11

USE PERSISTENT STORAGE WITH DOCKER

Docker has two options for containers to store files on the host machine, so that the files are persisted even after the container stops: volumes, and bind mounts.
Docker also supports containers storing files in-memory on the the host machine. Such files are not persisted. If you're running Docker on Linux, tmpfs mount is used to store files in the host's system memory. If you're running Docker on Windows, named pipe is used to store files in the host's system memory.

1. Volumes are stored in a part of the host filesystem which is managed by Docker (/var/lib/docker/volumes/ on Linux). Non-Docker processes should not modify this part of the filesystem. Volumes are the best way to persist data in Docker.

2. Bind mounts may be stored anywhere on the host system. They may even be important system files or directories. Non-Docker processes on the Docker host or a Docker container can modify them at any time.

Steps to use Docker Volumes are: -

1. There are different ways to mount a Docker volume while launching a container. Users can decide between the -v and the –mount flags, which are added to the docker run command.

2. create a Docker volume.

```
PS C:\Users\PARVESH> docker volume create mydata
mydata
```

Figure 46: Create Volume

3. Verify you have successfully created a Docker volume, prompt Docker to list all available volumes.

```
PS C:\Users\PARVESH> docker volume ls
DRIVER    VOLUME NAME
local     mydata
```

Figure 47: List volumes

4. To see more information about a Docker volume, use the inspect command.

```
PS C:\Users\PARVESH> docker inspect mydata
[
    {
        "CreatedAt": "2022-06-10T06:35:13Z",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/mydata/_data",
        "Name": "mydata",
        "Options": {},
        "Scope": "local"
    }
]
```

Figure 48: Inspect volume

5. To mount a data volume to a container add the –mount flag to the docker run command. It adds the volume to the specified container, where it stores the data produced inside the virtual environment.

   To run a container and mount a data volume to it, follow the basic syntax:

   $docker run -mount source = [volume_name], destination = [path_in_container][docker_image]$

6. To launch an Ubuntu container and mount the mydata volume to it, run:

```
PS C:\Users\PARVESH> docker run -it --name=example1 --mount source=mydata,de
stination=/data ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
405f018f9d1d: Already exists
Digest: sha256:b6b83d3c331794420340093eb706a6f152d9c1fa51b262d9bf34594887c2c
7ac
Status: Downloaded newer image for ubuntu:latest
```

Figure 49: Mount mydata volume

The steps to Copying Files Between Containers From a Shared Volume are: -

(a) switched to the container command prompt, move to the data volume directory.

(b) Create an empty sample file using the touch command.

(c) Now, exit the container.

```
root@bfc8f5bb198f:/# ls
bin   data  etc   lib    lib64   media  opt   root  sbin  sys  usr
boot  dev   home  lib32  libx32  mnt    proc  run   srv   tmp  var
root@bfc8f5bb198f:/# cd data
root@bfc8f5bb198f:/data# ls
root@bfc8f5bb198f:/data# touch sample1.txt
root@bfc8f5bb198f:/data# ls
sample1.txt
root@bfc8f5bb198f:/data# exit
exit
```

Figure 50: File in first container

(d) Then, launch a new container example2 with the same data volume.

```
PS C:\Users\PARVESH> docker run -it --name=example2 --mount source=mydata,de
stination=/data ubuntu
```

Figure 51: Launch example2 Container

(e) List the content of the container. You should find the data directory, as in example1.

(f) Move to the data directory and list the content of it.
The output should list the sample1.txt file you created in the previous container (example1).

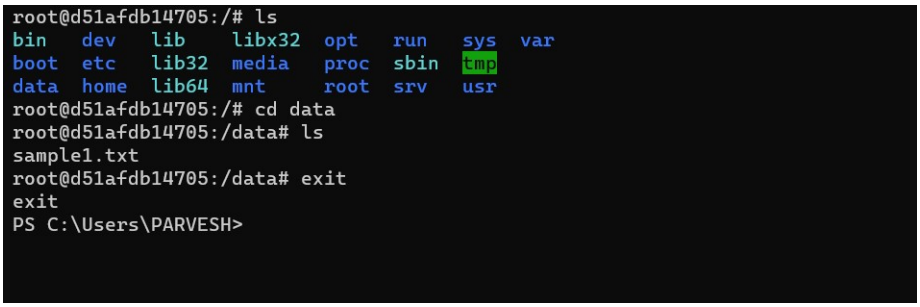

Figure 52: File in second container

You can also mount an existing directory from the host machine to a container. This type of volume is called Host Volumes.
You can mount host volumes by using the -v flag and specifying the name of the host directory.

The basic syntax for mounting a host directory is:
$docker run - v"\$(pwd)" : [volume_name][docker_image]$

The "$pwd" attribute instructs Docker to mount the directory the user is currently in. The following example outlines how this is done.

1. First, create a sample directory on the host under the name tmp and move into it:
2. Once inside the directory, create a test file to see whether it will be available from the container.

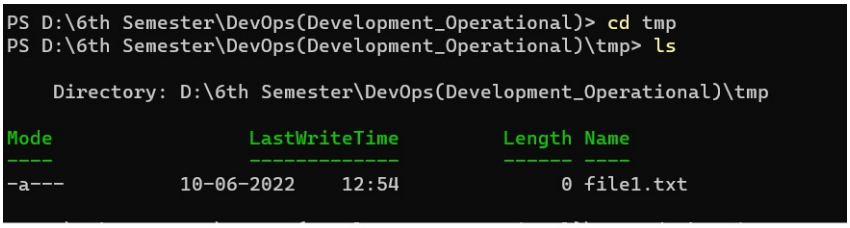

Figure 53: Sample Directory

3. Then, use the docker run command to launch an Ubuntu container with the host directory attached to it.
4. List the content of the container and verify there is a data1 directory.
5. Open the mounted directory and list the content. The output should display the file you created on the host.

Figure 54: File in the container

12 PRACTICAL 12

INSTALL JENKINS

First, you need to install OpenJDK. Jenkins currently only supports JDK8 and JDK11. Once Java is running, you can install Jenkins.
Steps to install jenkins are:

1. Click here to download the latest Jenkins package for Windows (currently it is version 2.130).

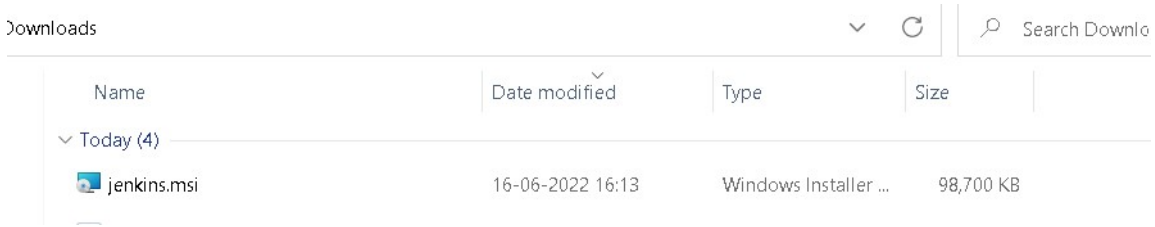

Figure 55: Jenkins exe file

2. Click “Next” to start the installation.

Figure 56: Setup wizard

3. Click the “Change. . . ” button if you want to install Jenkins in another folder. In this example, I will keep the default option and click on the “Next” button.

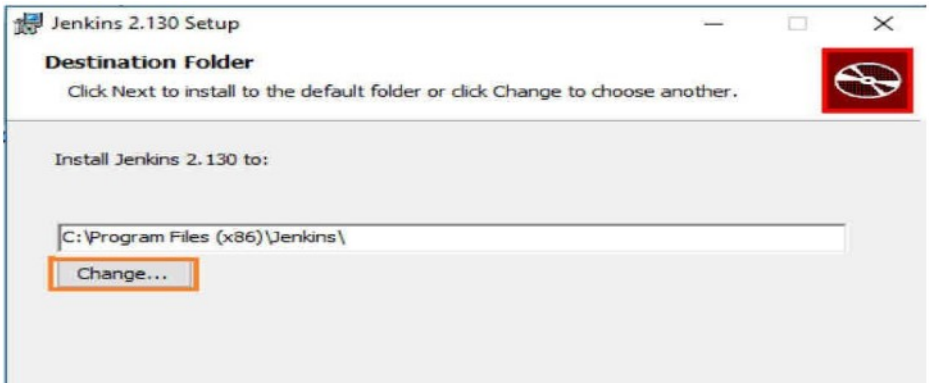

Figure 57: Destination folder

4. Click the “Install” button to start the installation process

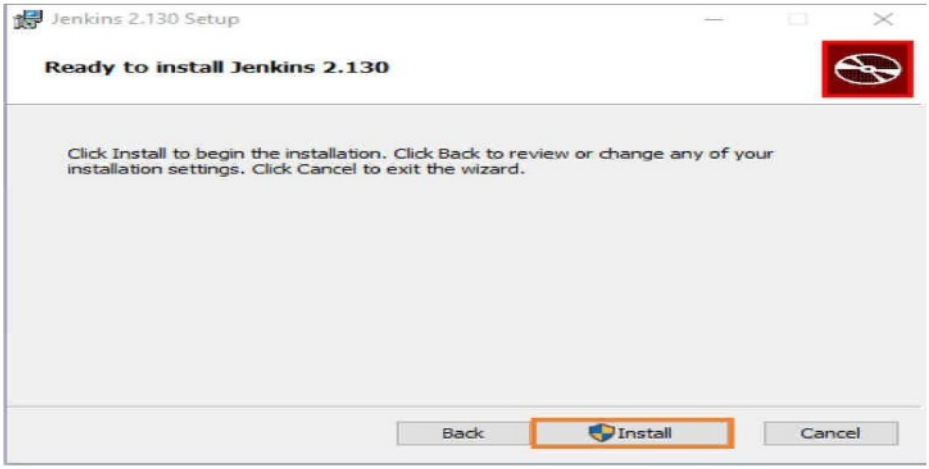

Figure 58: Ready to install jenkins

5. When done, click the “Finish” button to complete the installation process.

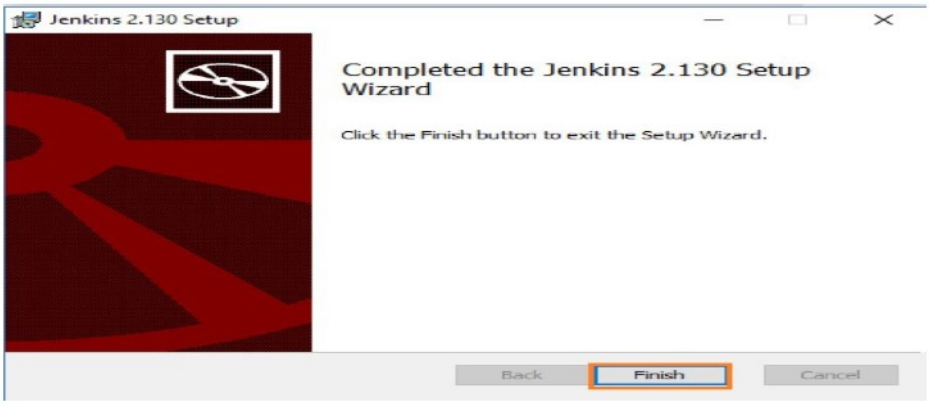

Figure 59: complete setup wizard

6. You will automatically be redirected to a local Jenkins page, or you can pastethe URL http://localhost:8080 in a browser.

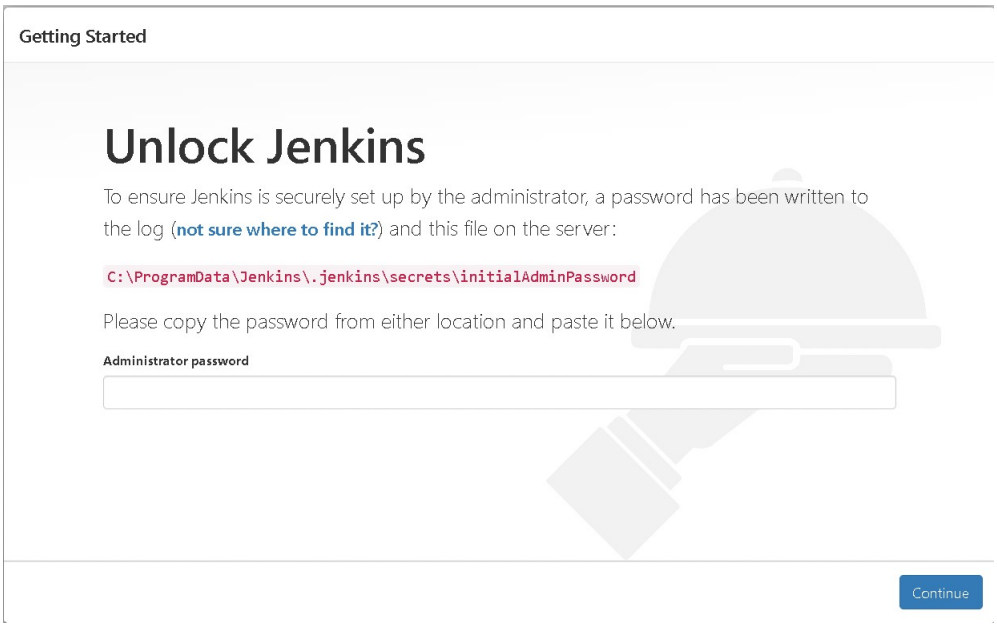

Figure 60: Unlock jenkins

7. To unlock Jenkins, copy the password from the file at C:Files (x86)and paste it in the Administrator password field. Then, click the "Continue" button.

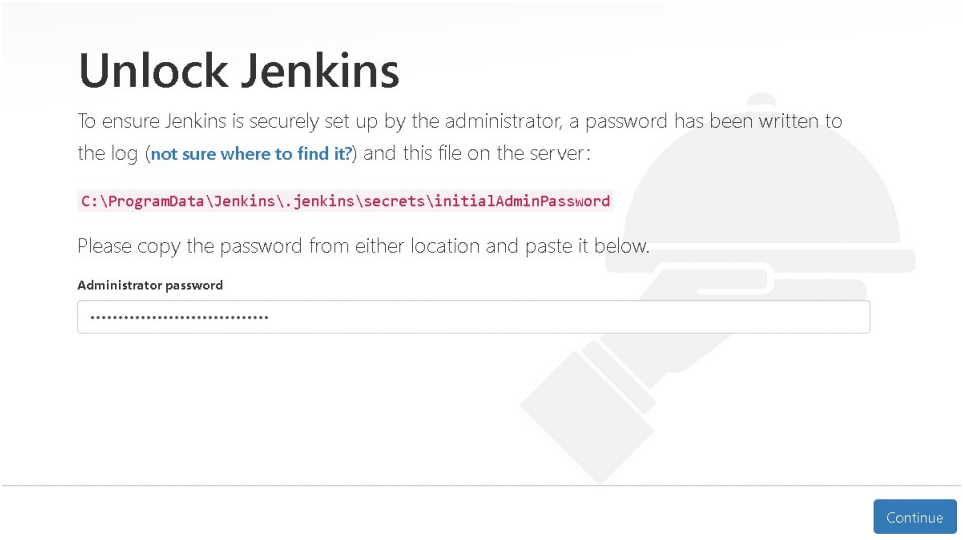

Figure 61: Fill password

8. You can install either the suggested plugins or selected plugins you choose.To keep it simple, we will install the suggested plugins.

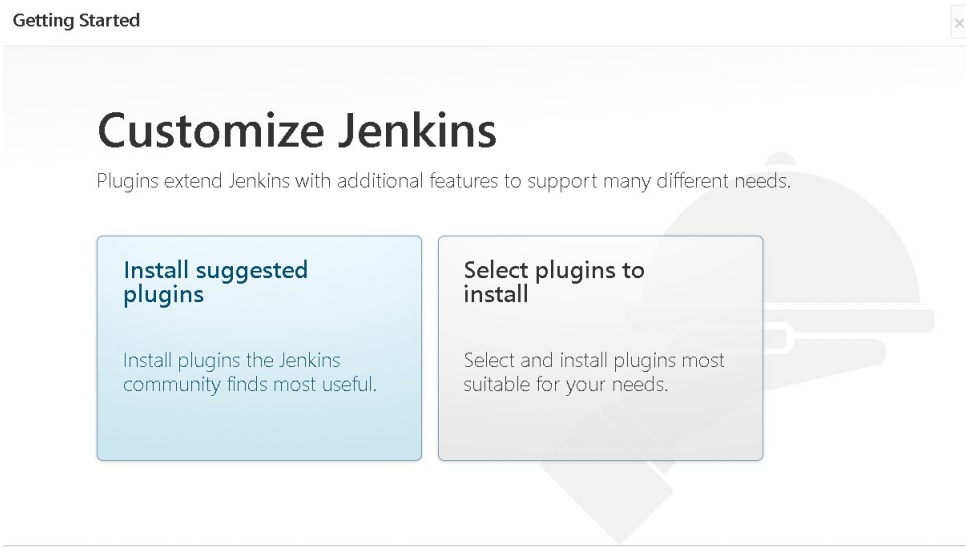

Figure 62: Install suggested plugins

9. Wait until the plugins are completely installed

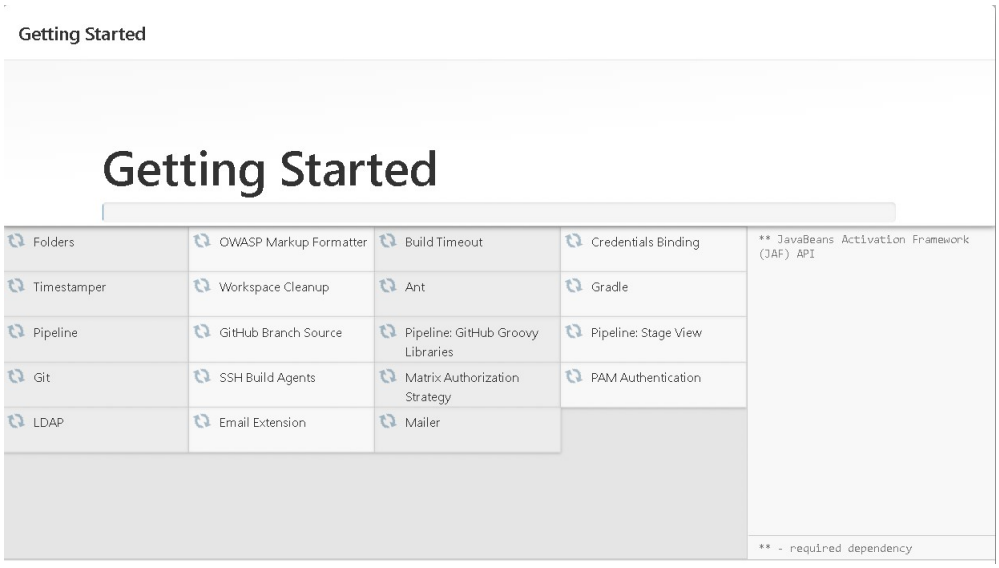

Figure 63: Getting started

10. The next thing that you should do is create an Admin user for Jenkins. Then,enter your details and click “Save and Continue”.

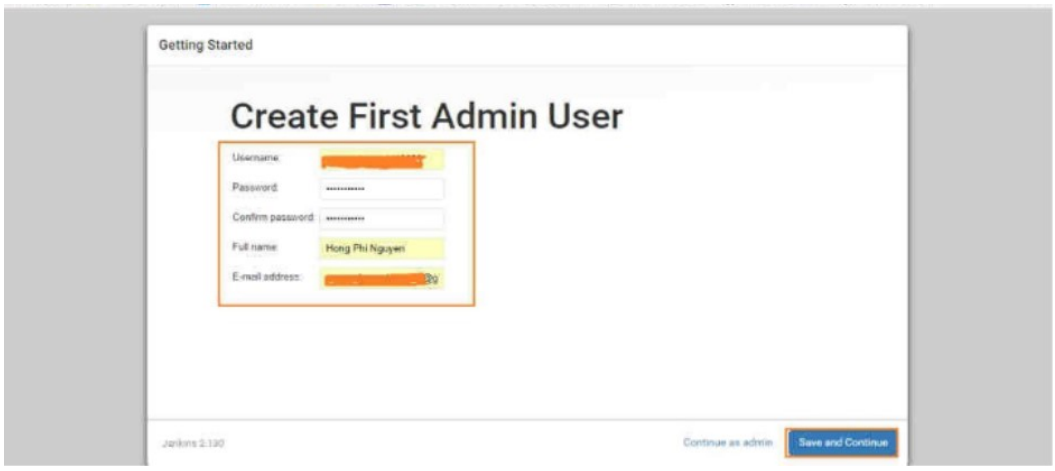

Figure 64: create first user

11. Click “Save and Finish” to complete the Jenkins installation.

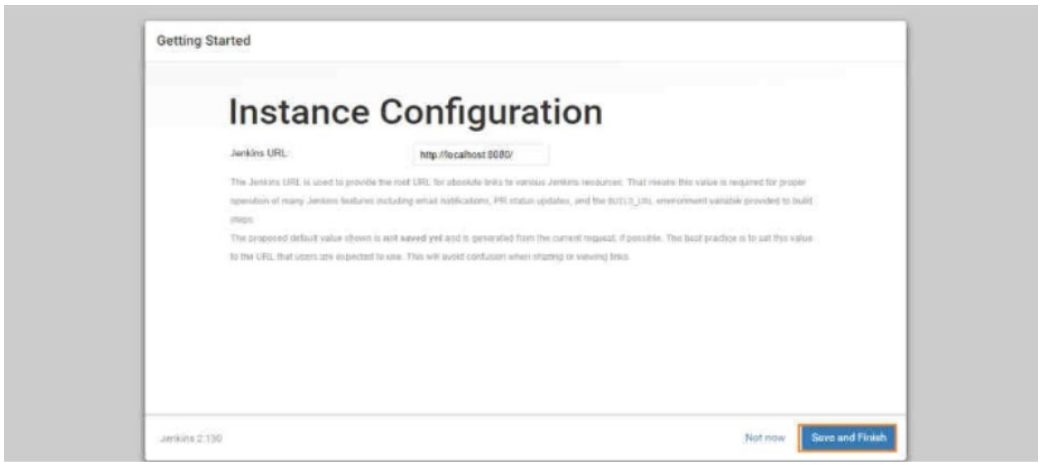

Figure 65: Instance configuration

12. Now, click “Start using Jenkins” to start Jenkins. ’

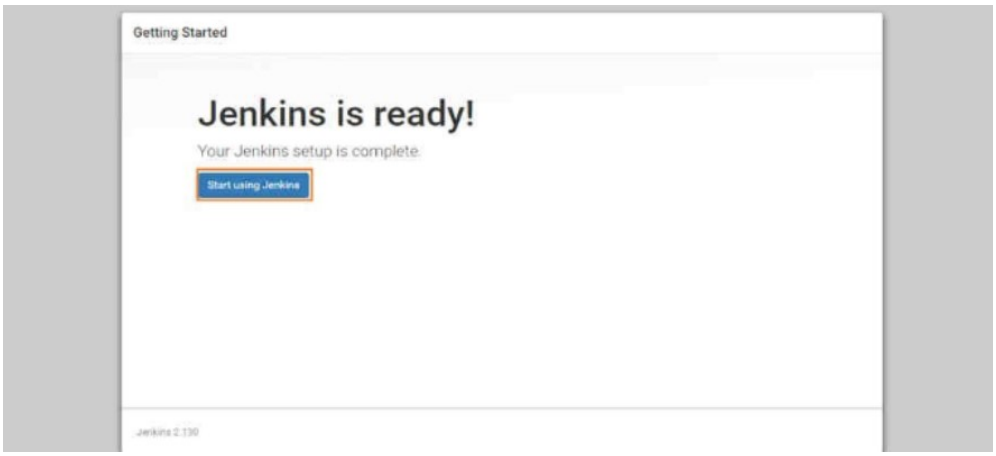

Figure 66: Jenkins is ready

13. Finally, here is the default Jenkins page.

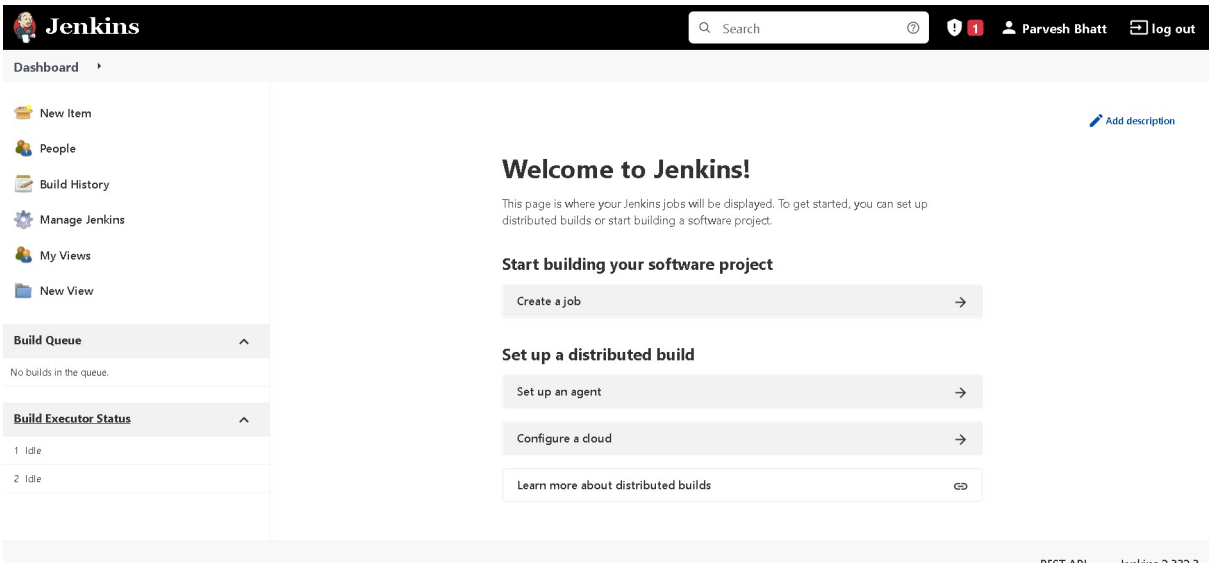

Figure 67: Default dashboard

13 PRACTICAL 13

CREATE JOBS ON JENKINS

Steps to create jobs in jenkins are: -

1. Go to the Jenkins dashboard and Click on New Item, In the next screen, enter the Item name, in this case we have named it newjob. Choose the 'Freestyle project option'.

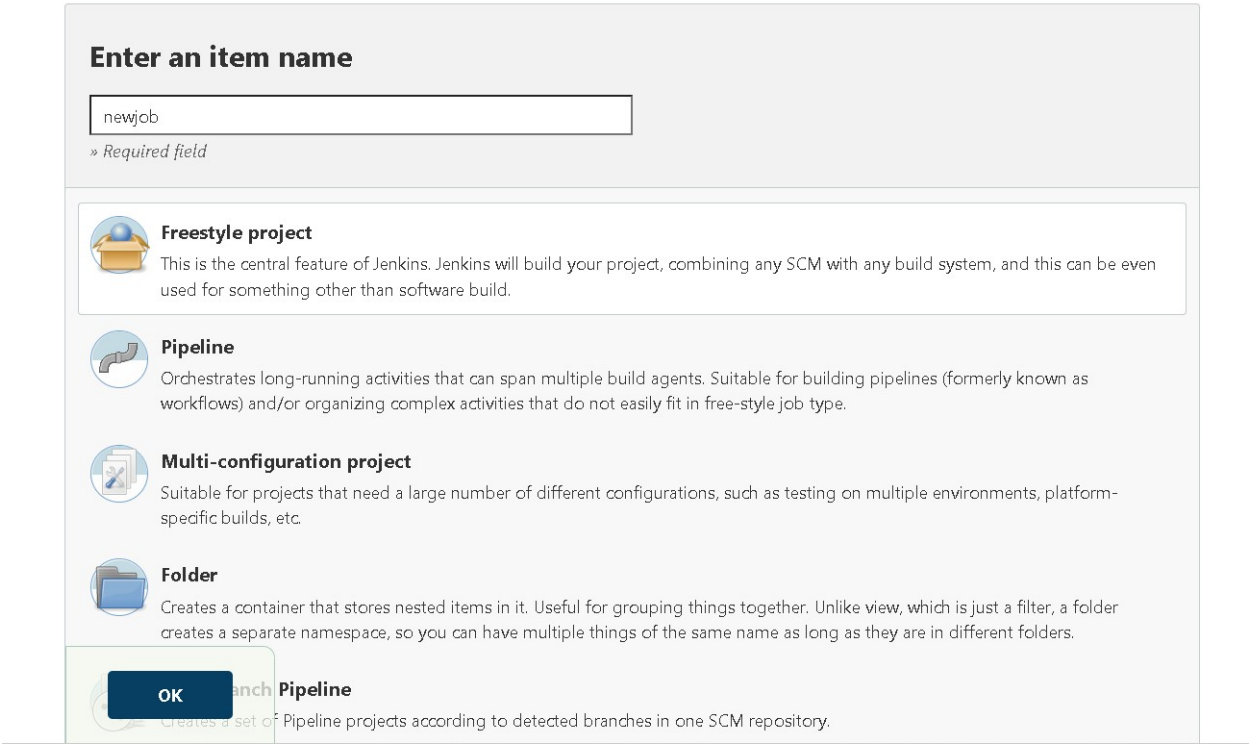

Figure 68: Enter an item name

2. The following screen will come up in which you can specify the details of the job.Under description write anything you want to give it as your description. Above I have given "this is my first Jenkins job".

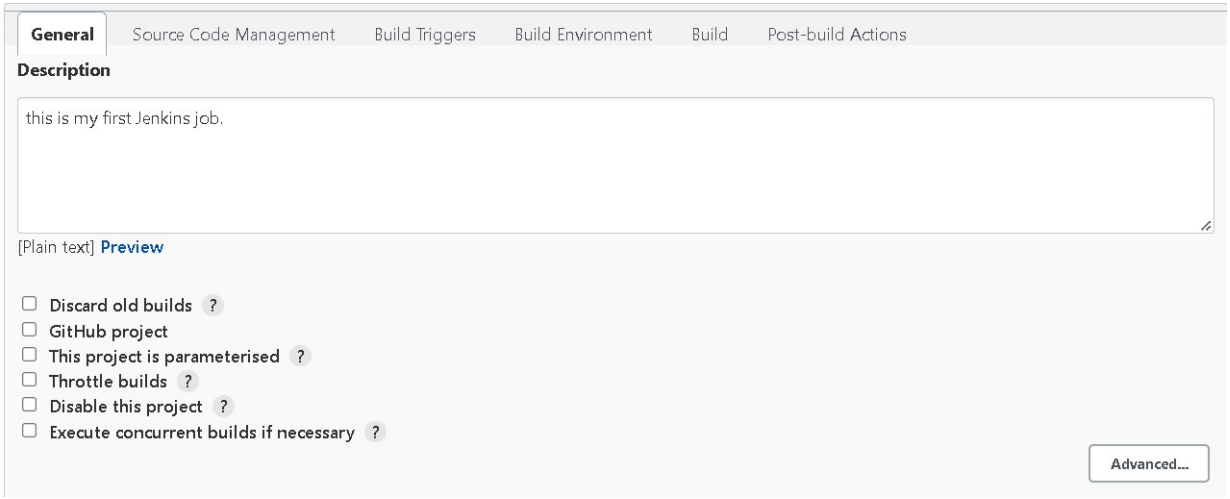

Figure 69: Write description

3. Source code management for now is none selected.

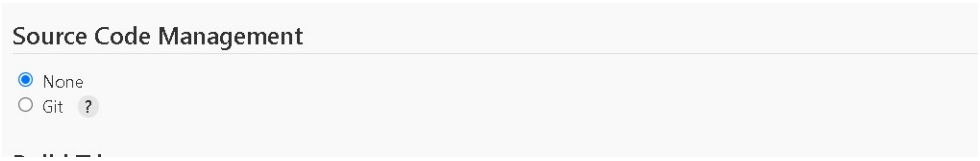

Figure 70: Source code management

4. Build triggers to be selected as Build periodically with * * * * * which means after every minute I want my job to be run.

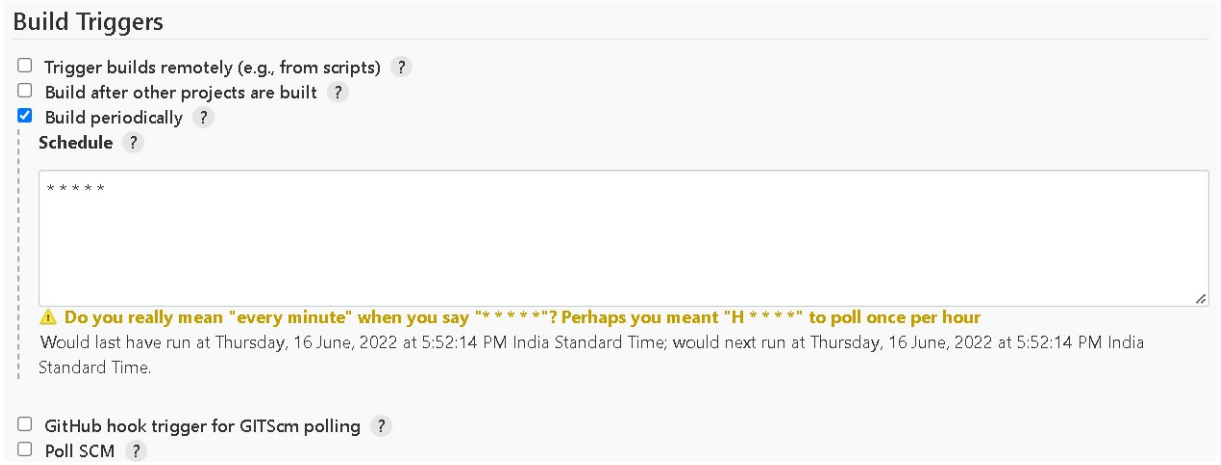

Figure 71: Build triggers

5. Build it with windows batch Command. Using echo “Hello jenkins....”.

Figure 72: Build

6. After build apply all the changes ,save them and go to dashboard.. there shows our newjob project.

Figure 73: newjob item in dashboard

7. Click on the arrow down button then select build now to check the build history.

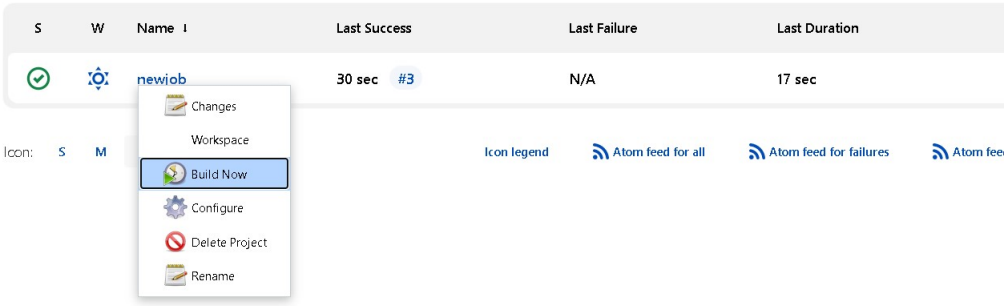

Figure 74: Build now

8. Here is the record of build history.

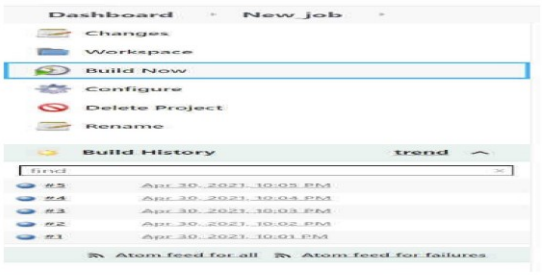

Figure 75: Build history

9. Then click on the one of the history time to get the console output.

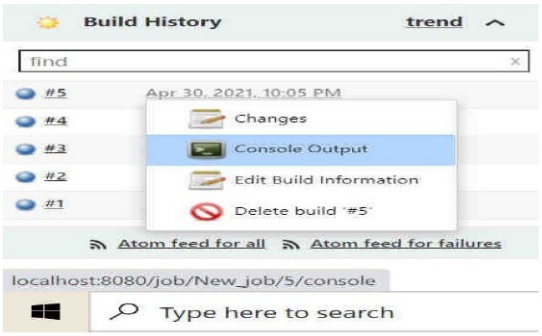

Figure 76: check console output

10. So when clicked to the console output. Here is the detail of our job created. SUCCESS status is shown which means our job is complete.

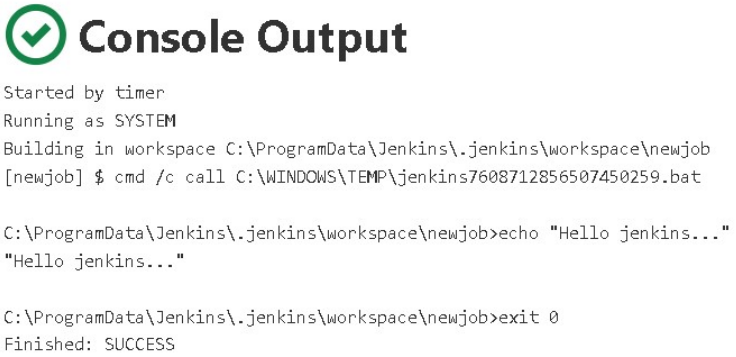

Figure 77: Console output

14 PRACTICAL 14

INTEGRATE JENKINS WITH GIT/ GITHUB

STEPS:-

1. Create a sample Program
2. Create a Jenkins job
3. Add this program to GITHUB
4. Add GIT Plugin in Jenkins
5. Configure Jenkins job

1. Create a Sample program:
   You can create any sample program that you want like Java or Python or any other program. Here we will write a simple Python program that prints Hello. . . ..Good evening!

Figure 78: Sample program

2. Create a Jenkins job:

   - Open the web browser and go to Jenkins's home page using the link localhost: 8080. This is the default port number.
   - Open Jenkins home page by entering the username and password.

Figure 79: Login Page

   - For creating a project click on New Item and enter the project name and select Freestyle project. Click on OK.

Figure 80: Creating new job

3. Add this program to Github.

   - Open git bash on your system. Navigate to the location of your program. Initialize an empty repository using the command git init.
   - Use the command git add . to add the file to the staging area from the working directory
   - Now add the file to the local repository using the command git commit -m “demo.py file added”.
   - Now you have to push this file to the remote repository. For doing that go to your GitHub account and create a new public repository. Now copy the location of this repository and go to git bash terminal. Here type the command git remote add origin ¡location-of-repository¿. Since now you have connected to the remote repository, you can now push your code there using the command git push -u origin master. To verify this go to GitHub account and refresh the page. You will see the file added there.

4. Add Git Plugin in Jenkins:

   - On Jenkins’s homepage go to Manage Jenkins

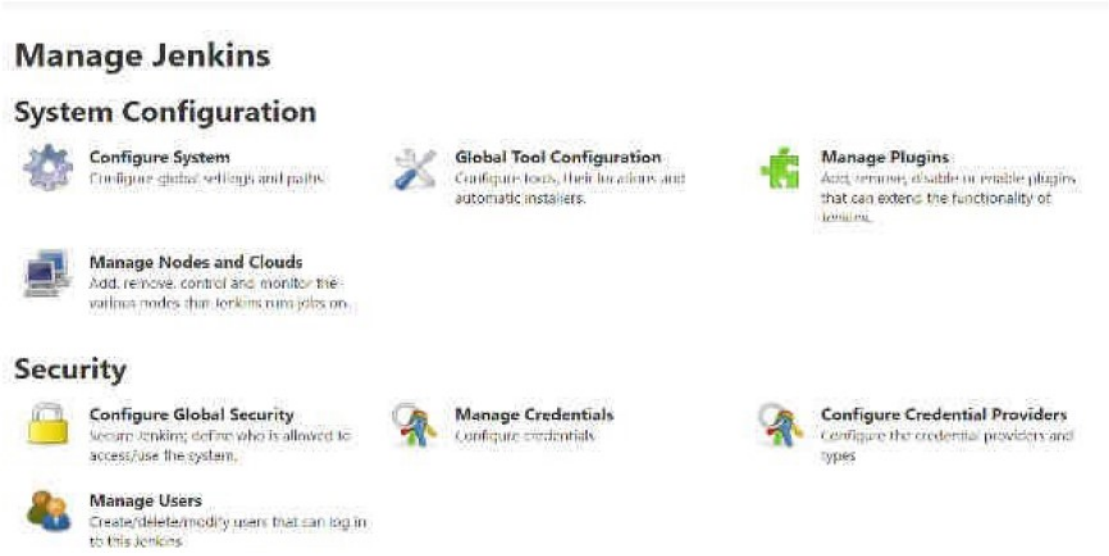

Figure 81: Manage jenkins

- Next click on Manage Plugins. Here check got Git plugin in the installed section. If it is not available here search for it in the available section and download it.

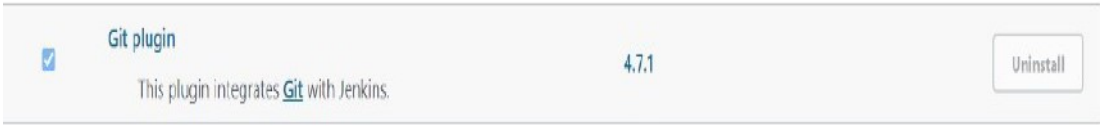

Figure 82: Git plugin

5. Configure Jenkins job to trigger the build:

   - Go to the project in Jenkins that we created in step 2. Here in the Source Code Management section, select git and enter the link of the public repository that you created in step 3. Next in the Build Triggers section, click on Poll SCM option. Here in the Schedule part, you have to enter five asterisks separated by space. This is nothing but cron syntax for your job. This means that Jenkins will check for any changes in the source code every minute and if there is any change it will trigger the Jenkins build.

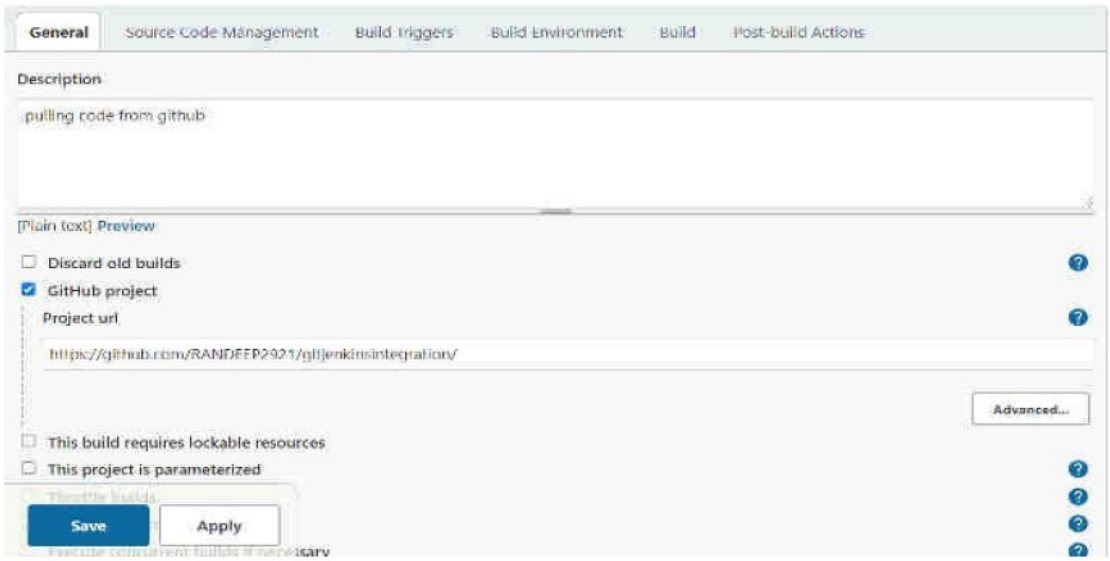

Figure 83: Configure jenkins job

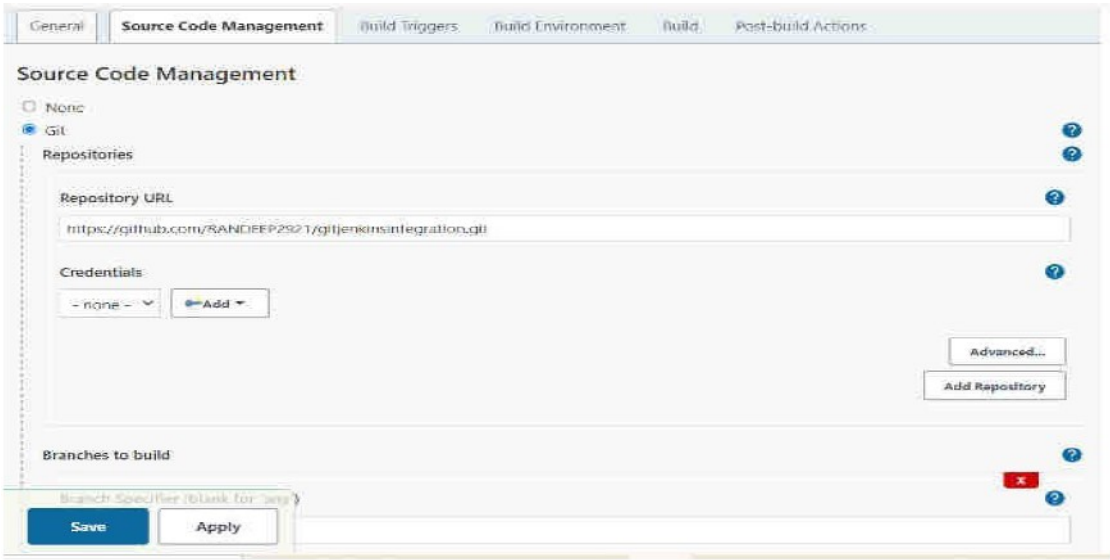

Figure 84: Set Source code Management to Git

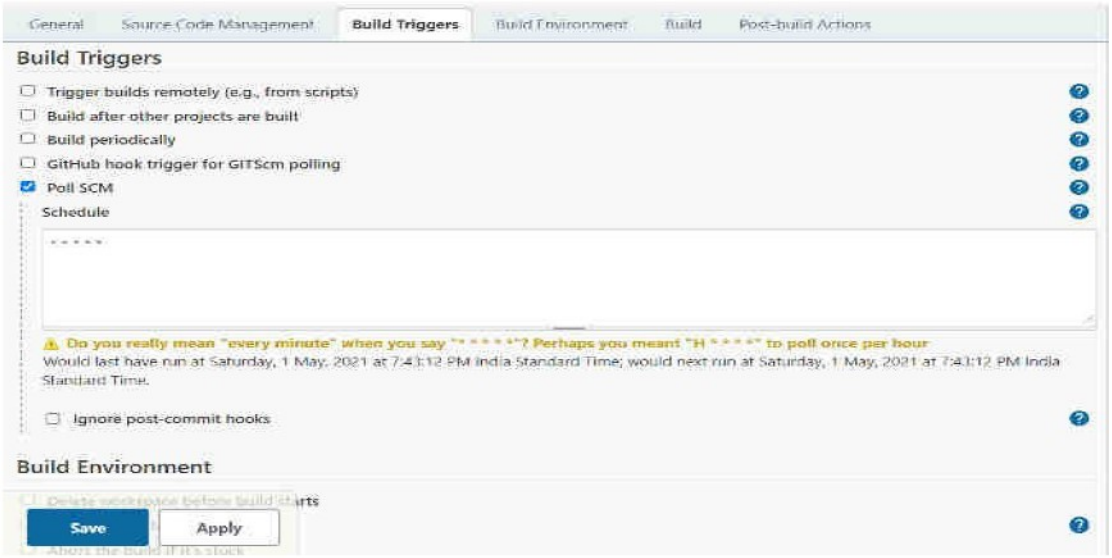

Figure 85: Build triggers

- Click on Apply and then on Save. Next on your project home page click on Build Now. This will run the project and in the console output, you can see your program output the status of your Jenkins job. If everything is alright then it will display as Success.

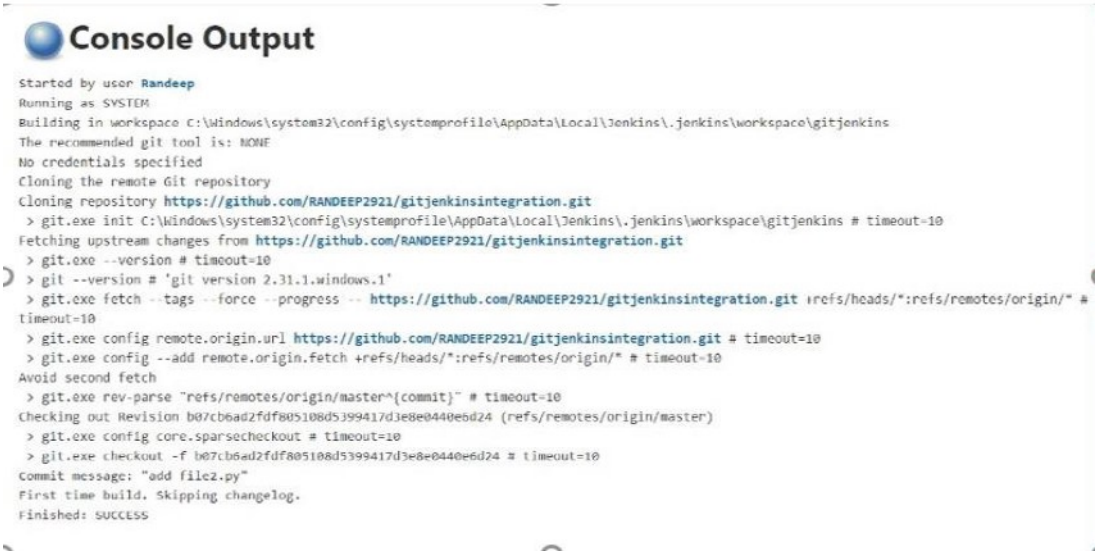

Figure 86: Console output