DEVOPS
SOFTWARE ARCHITECTURE LAB

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

(Information Technology)

Submitted by

Name: Shivay Bhandari

URN:1905398

CRN:1921142

Submitted to

Prof.Harjot Kaur Gill

Department of Information Technology,
Guru Nanak Dev Engineering College

Ludhiana-141006

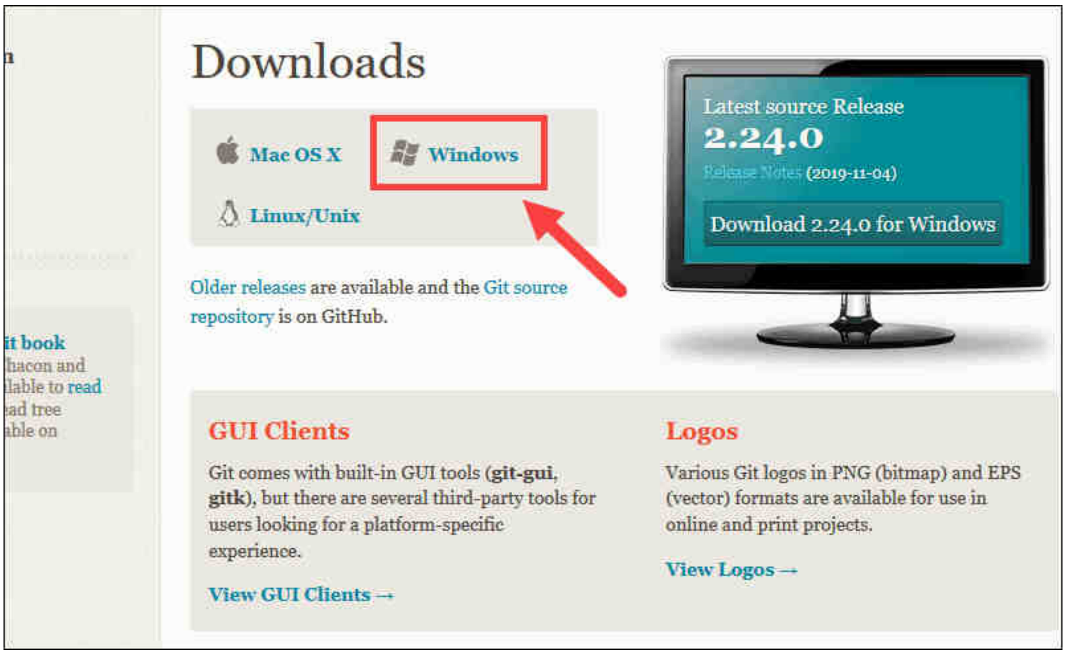
Index

PRACTICAL 1

Install GIT

Download Git for Windows

Browse to the official Git website: https://git-scm.com downloads Click the download link for Windows and allow the download to complete.



Older releases are available and the Git source repository is on GitHub.

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

Figure 1: Download git

Extract and Launch Git Installer

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

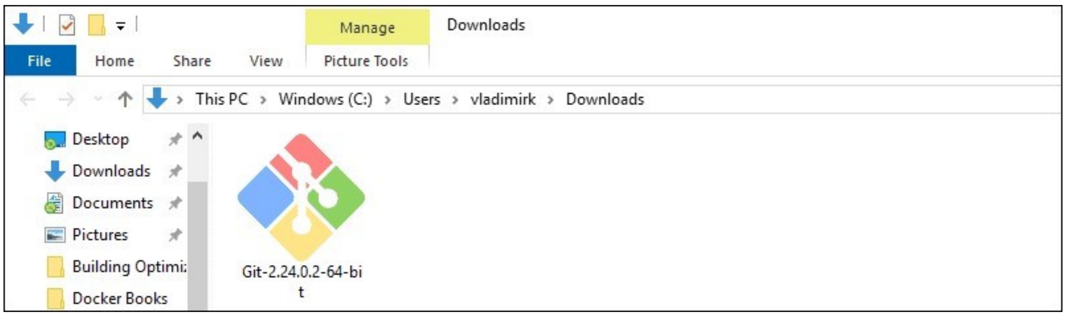
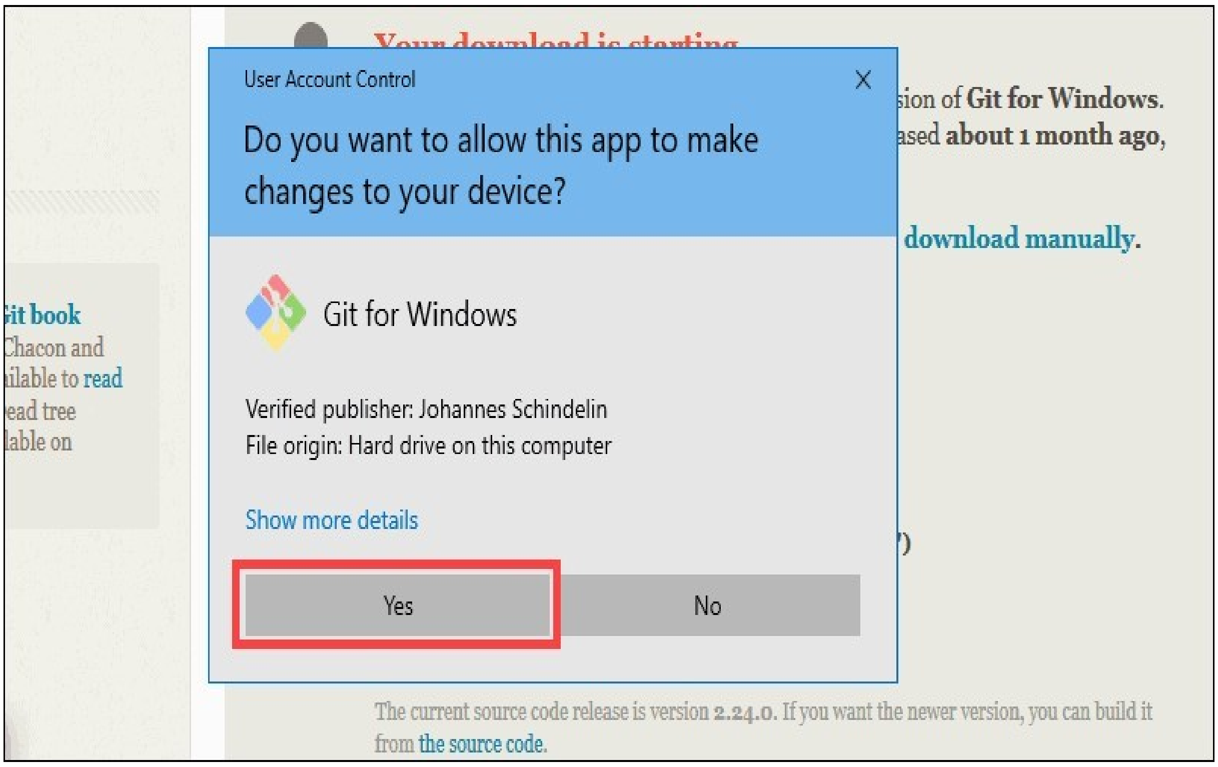


Figure 2: install git

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


User Account Control

Do you want to allow this app to make changes to your device?

Git for Windows

Verified publisher: Johannes Schindelin
File origin: Hard drive on this computer

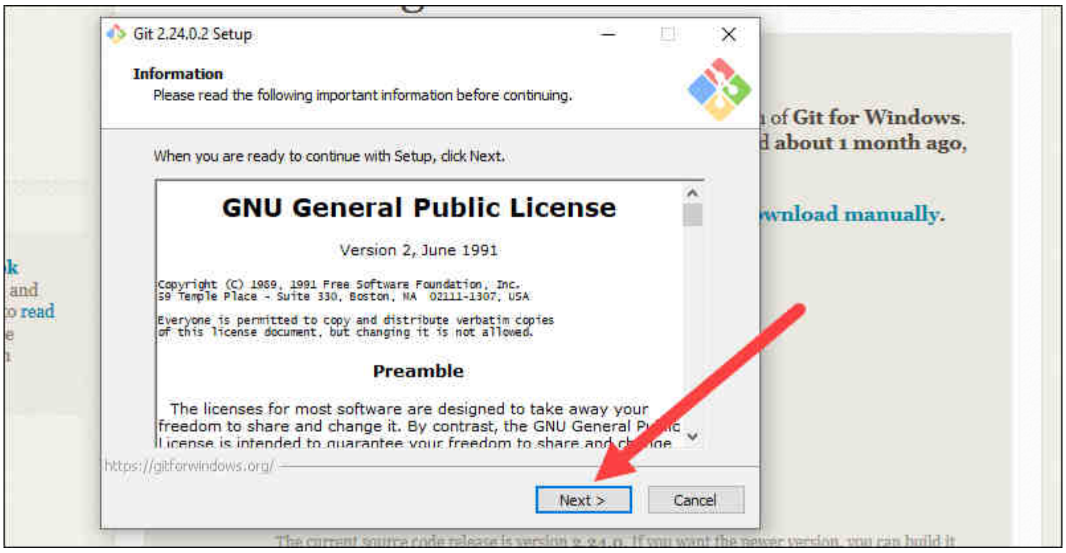
Show more details

Yes No

The current source code release is version 2.24.0. If you want the newer version, you can build it from the source code.

Figure 3: git for download

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


Git 2.24.0.2 Setup

Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

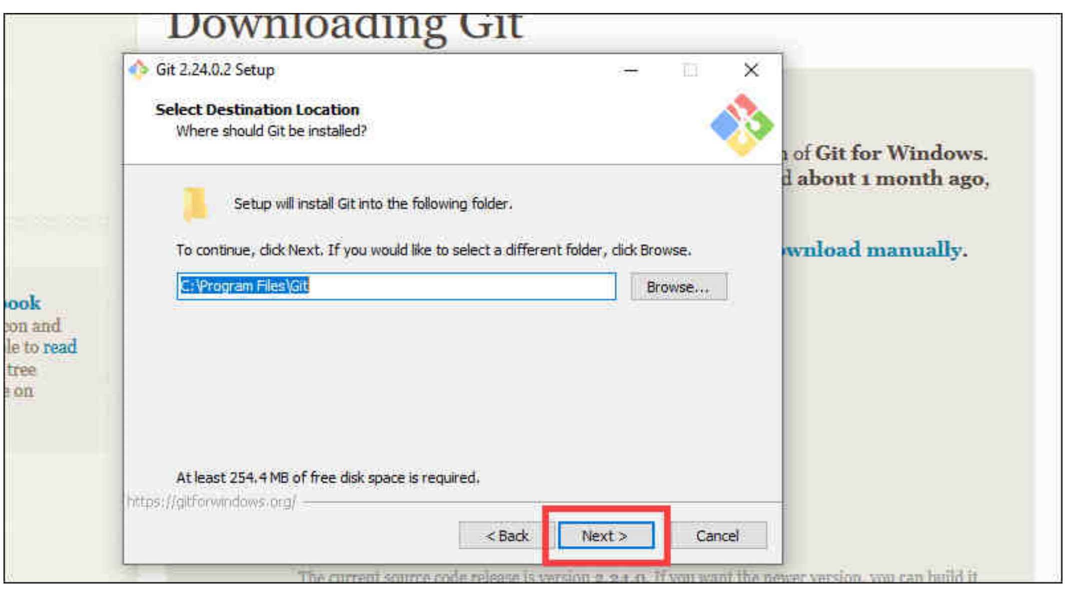
Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change

https://gitforwindows.org/

Figure 4: git setup

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


Git 2.24.0.2 Setup

Select Destination Location

Where should Git be installed?

Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

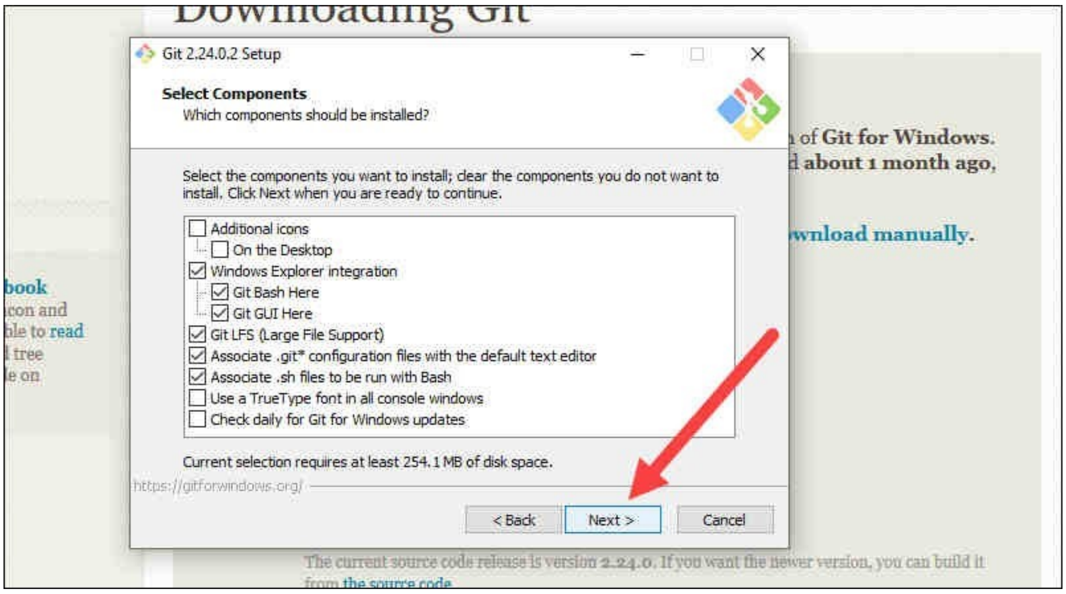
C:\Program Files\Git

At least 254.4 MB of free disk space is required.

https://gitforwindows.org/

Figure 5: select location

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



Git 2.24.0.2 Setup

Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

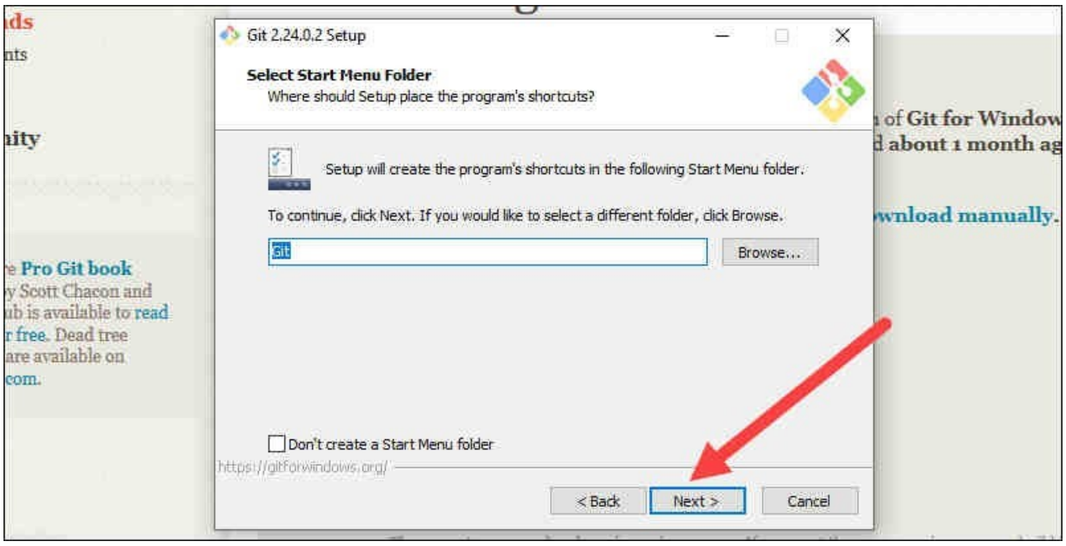
- [ ] Additional icons
  - [ ] On the Desktop
- [x] Windows Explorer integration
  - [x] Git Bash Here
  - [x] Git GUI Here
- [x] Git LFS (Large File Support)
- [x] Associate .git* configuration files with the default text editor
- [x] Associate .sh files to be run with Bash
- [ ] Use a TrueType font in all console windows
- [ ] Check daily for Git for Windows updates

Current selection requires at least 254.1 MB of disk space.

https://gitforwindows.org/

Figure 6: select component

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



Git 2.24.0.2 Setup

Select Start Menu Folder

Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

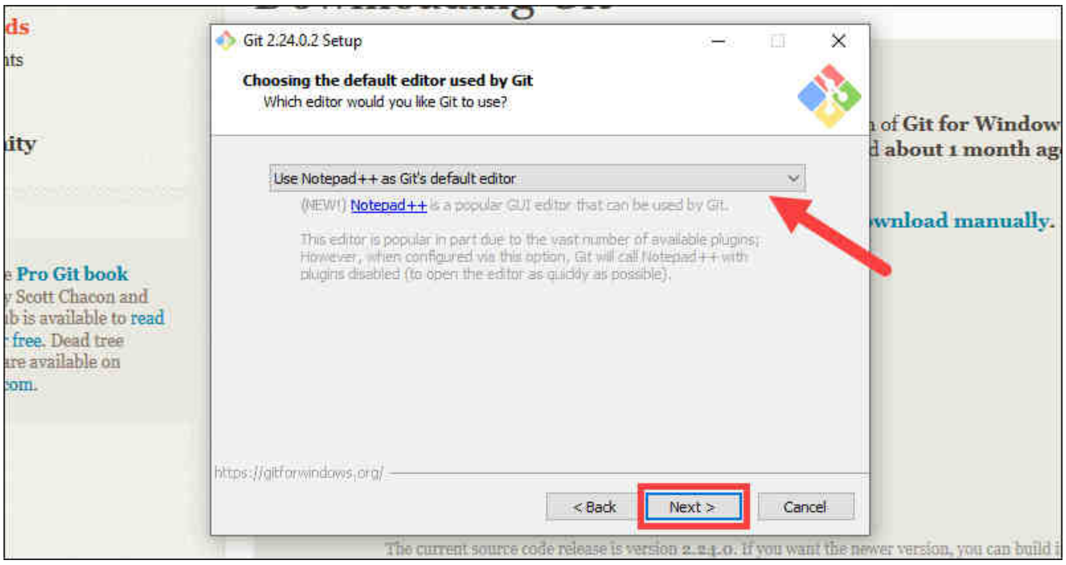
Git

- [ ] Don't create a Start Menu folder

https://gitforwindows.org/

Figure 7: select start menu folder

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


Git 2.24.0.2 Setup

Choosing the default editor used by Git
Which editor would you like Git to use?

Use Notepad++ as Git's default editor

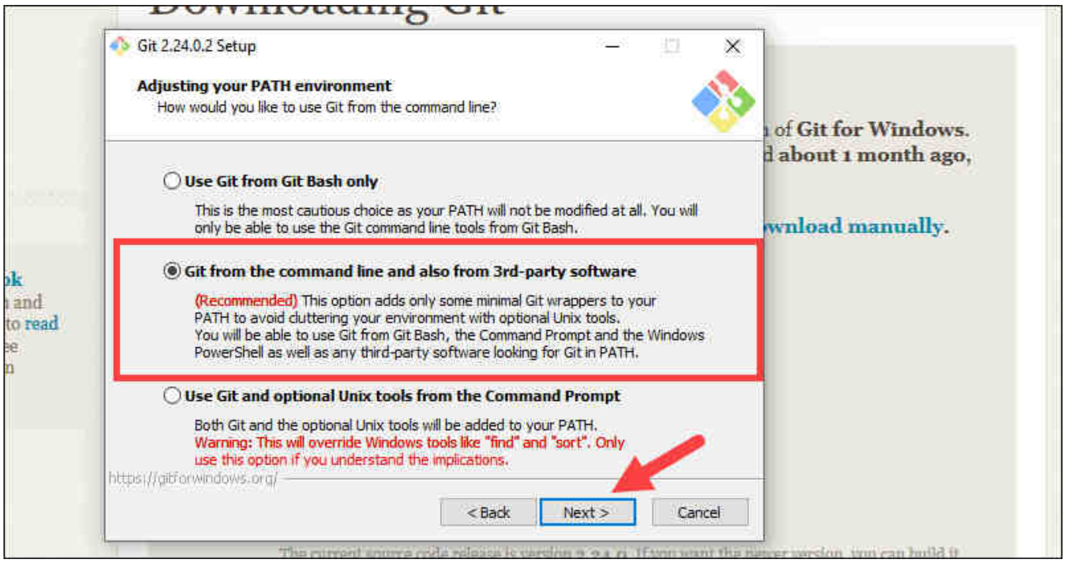
(NEW!) Notepad++ is a popular GUI editor that can be used by Git.

This editor is popular in part due to the vast number of available plugins; However, when configured via this option, Git will call Notepad++ with plugins disabled (to open the editor as quickly as possible).

https://gitforwindows.org/

Figure 8: choosing the default editor used by git

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.


Git 2.24.0.2 Setup

Adjusting your PATH environment
How would you like to use Git from the command line?

Use Git from Git Bash only
This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

[✓ **Git from the command line and also from 3rd-party software**]
(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

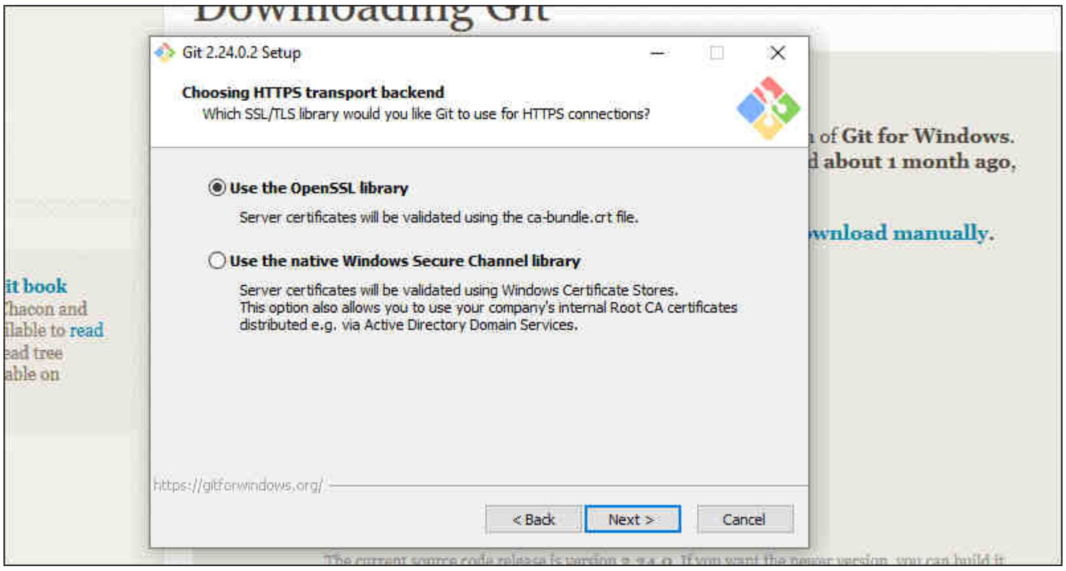
Use Git and optional Unix tools from the Command Prompt
Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://gitforwindows.org/

Figure 9: path envoronment

Server Certificates, Line Endings and Terminal Emulators

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



Git 2.24.0.2 Setup

Choosing HTTPS transport backend

Which SSL/TLS library would you like Git to use for HTTPS connections?

[✓ Use the OpenSSL library]

Server certificates will be validated using the ca-bundle.crt file.

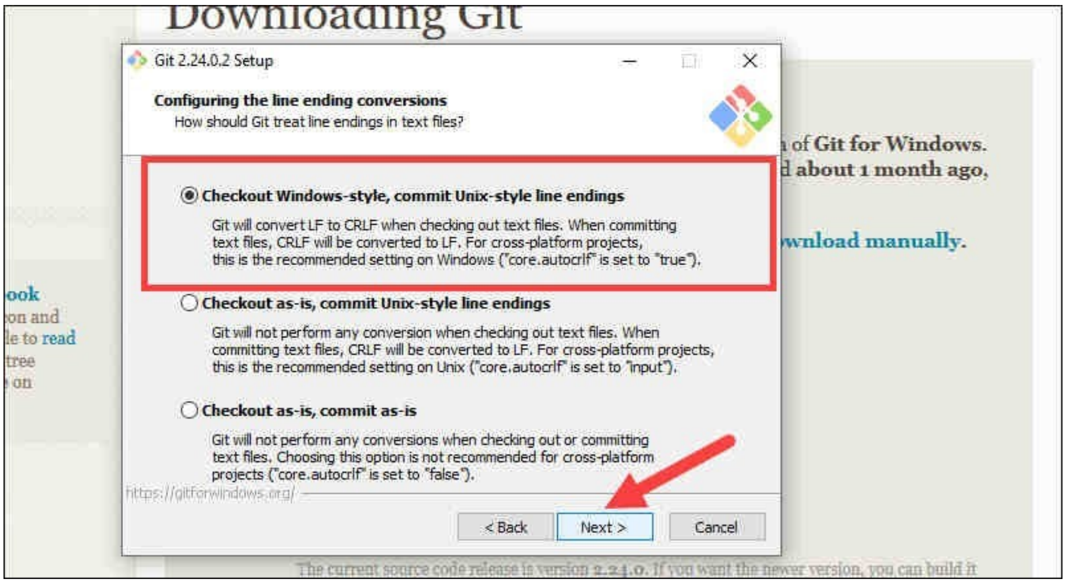
Use the native Windows Secure Channel library

Server certificates will be validated using Windows Certificate Stores. This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

https://gitforwindows.org/

Figure 10: choosing the http transport backend

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



Git 2.24.0.2 Setup

Configuring the line ending conversions

How should Git treat line endings in text files?

[✓ Checkout Windows-style, commit Unix-style line endings]

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

Checkout as-is, commit Unix-style line endings

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

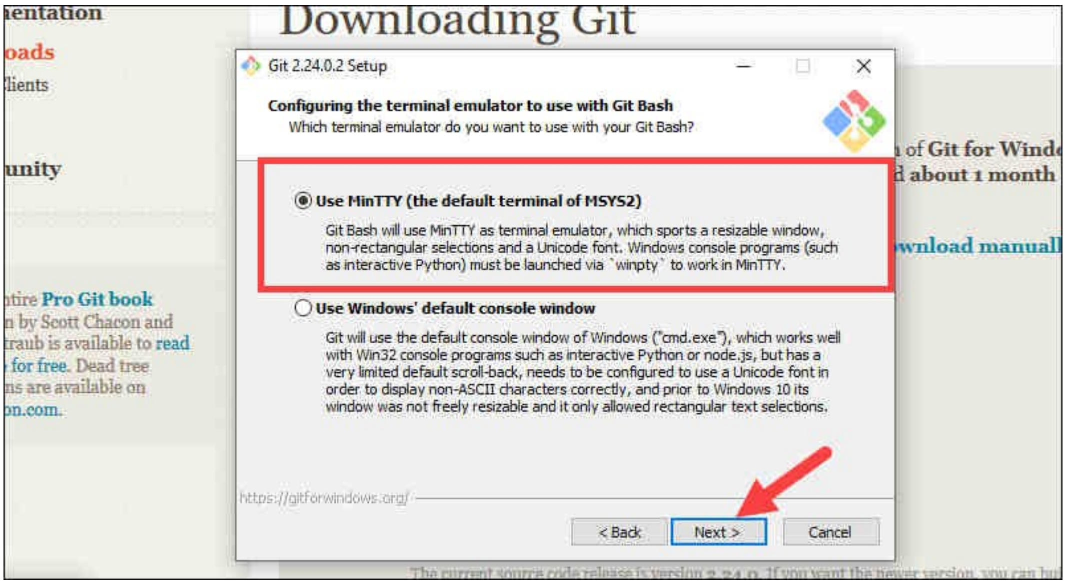
Checkout as-is, commit as-is

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

https://gitforwindows.org/

Figure 11: git setup

Choose the terminal emulator you want to use. The default MinTTY isrecommended, for its features. Click Next. **Additional Customization Options** The default options are



Configuring the terminal emulator to use with Git Bash

Which terminal emulator do you want to use with your Git Bash?

● **Use MinTTY (the default terminal of MSYS2)**

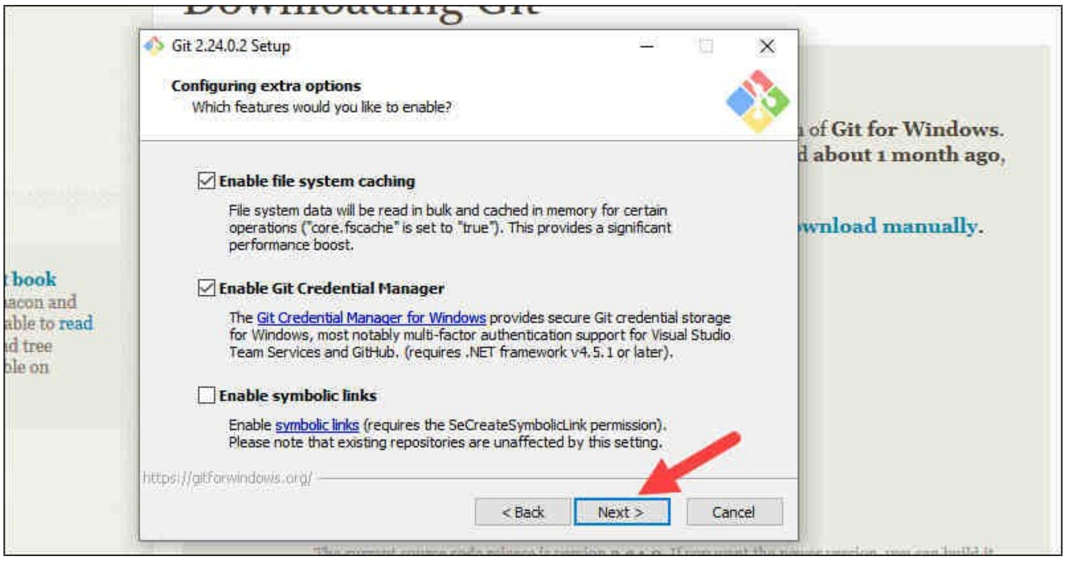
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.

○ **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

Figure 12: git setup

recommended, however this step allows you todecide which extra option you would like to enable. If you use symbolic links, which are like shortcuts for the command line, tick the box. Click Next.



Configuring extra options

Which features would you like to enable?

☑ **Enable file system caching**

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

☑ **Enable Git Credential Manager**

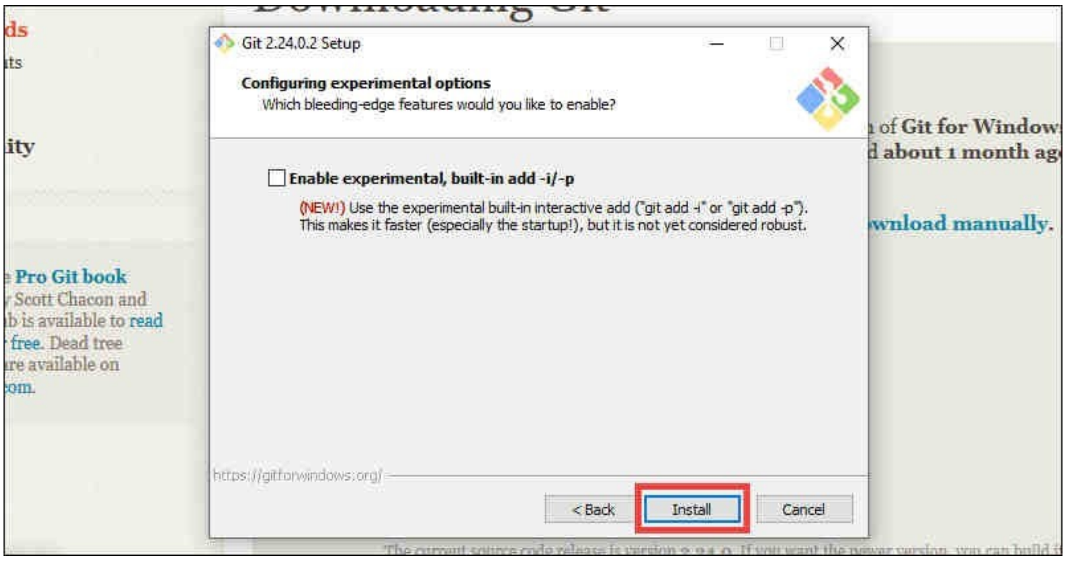
The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or later).

☐ **Enable symbolic links**

Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

Figure 13: git setup

Depending on the version of Git you're installing, it may offer to installexperimental features. At the time this article was written, the option to include interactive options was offered. Unless you are feeling adventurous, leave them unchecked and click Install.


Git 2.24.0.2 Setup

Configuring experimental options

Which bleeding-edge features would you like to enable?

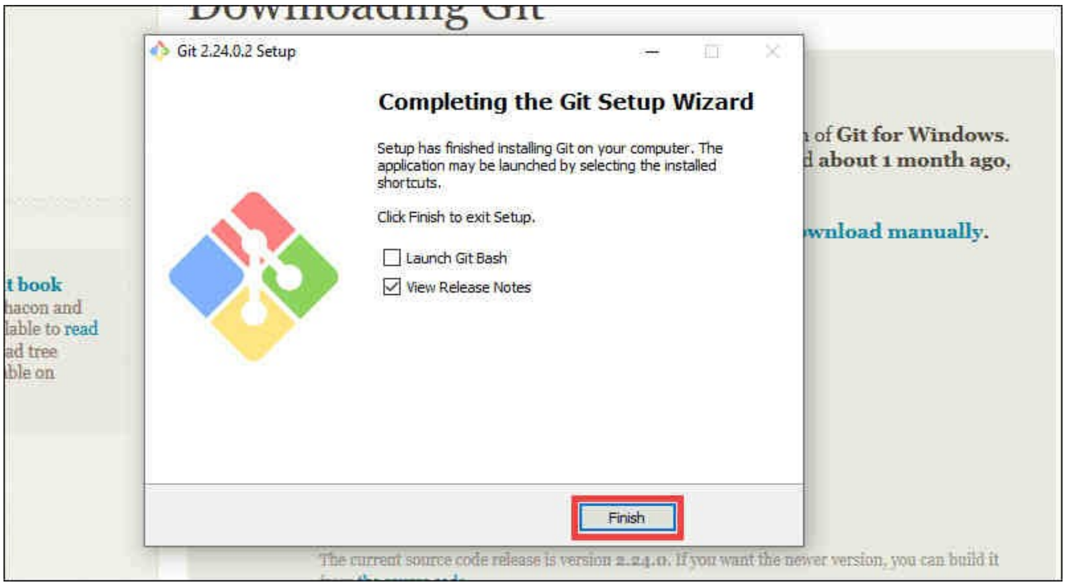
☐ **Enable experimental, built-in add -i/-p**

(NEW!) Use the experimental built-in interactive add ("git add -i" or "git add -p"). This makes it faster (especially the startup!), but it is not yet considered robust.

https://gitforwindows.org/

Figure 14: git setup

Complete Git Installation Process Once the installation is complete, tick the boxes to view the Release Notesor Launch Git Bash, then click Finish. **The Git Gui looks like as**


Git 2.24.0.2 Setup

Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☐ Launch Git Bash

☑ View Release Notes

Figure 15: git setup

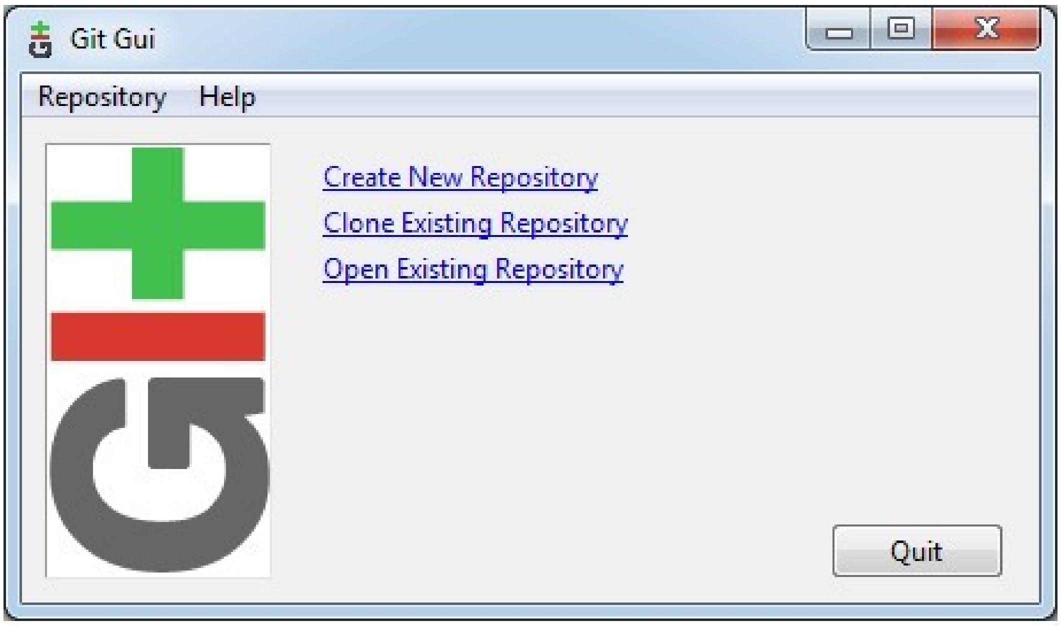


Figure 16: Git Gui

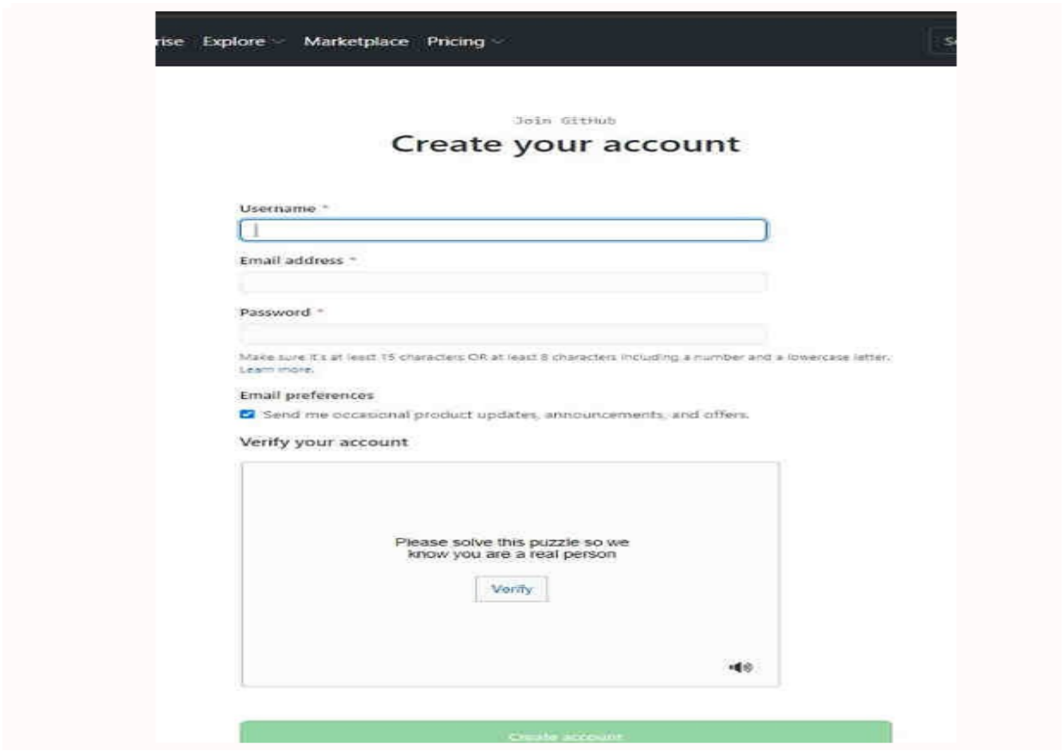
PRACTICAL 2

Create account on GITHUB

Open GitHub.com and click on SignUp.

A new tab will open with the tag line Create your account. There you have to enter your following details.

- A new tab will open with the tag line Create your account. There you have to enter your following details.
- Enter your Email address.
- Enter Password and this password must be combinations of characters, numbers, and special symbols.
- Now, Click on Verify and there you may find a puzzling image and you have to make that image in the Correct Position.



Join GitHub

Create your account

Username *

Email address *

Password *

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

Email preferences

✓ Send me occasional product updates, announcements, and offers.

Verify your account

Please solve this puzzle so we know you are a real person

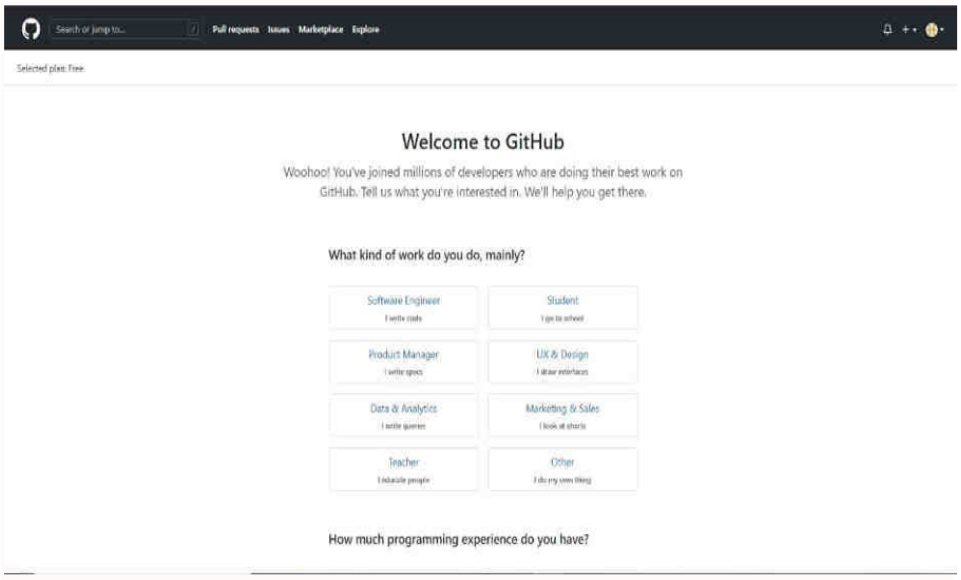
Verify

Create account

Figure 17: Create Account

Click on Create account.
Now, you will see a welcome message from GitHub. And have to answer some general questions. Like-

- What Kind of work do you do?
- How much Programming experience do you have- none, a little, moderate, or a lot?
- What do you plan to use GitHub for?
- Give some of the fields of interest.



Selected plan: Free

Welcome to GitHub

Woohoo! You've joined millions of developers who are doing their best work on GitHub. Tell us what you're interested in. We'll help you get there.

What kind of work do you do, mainly?

How much programming experience do you have?

Figure 18: open Github account

Click on Complete Setup.

1. Now, you have to verify your Email address. You will get a link on your registered email id. Click on that link. After this, you will get a message " your email was verified“.
2. You will see the question " what do you want to do first?" You may skip this step.

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

PRACTICAL 3

Repository using GIT/GITHUB

GITHUB

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

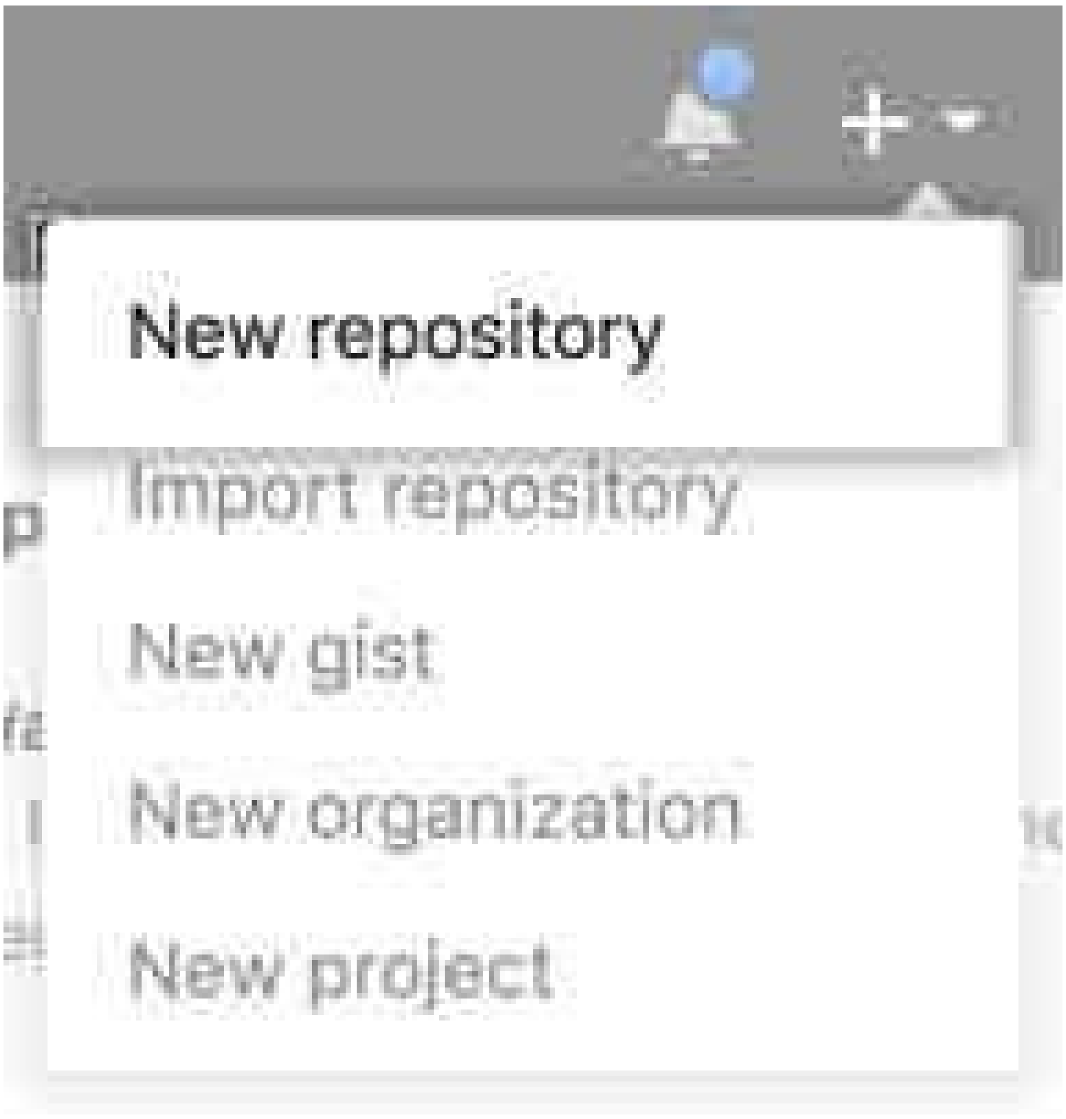
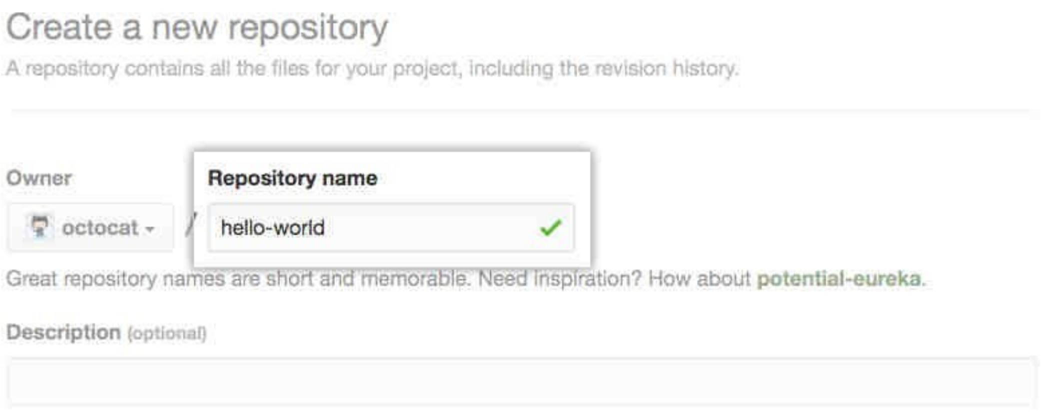


Figure 19: Repository

2. Type a short, memorable name for your repository. For example,"hello-world".


Create a new repository

A repository contains all the files for your project, including the revision history.

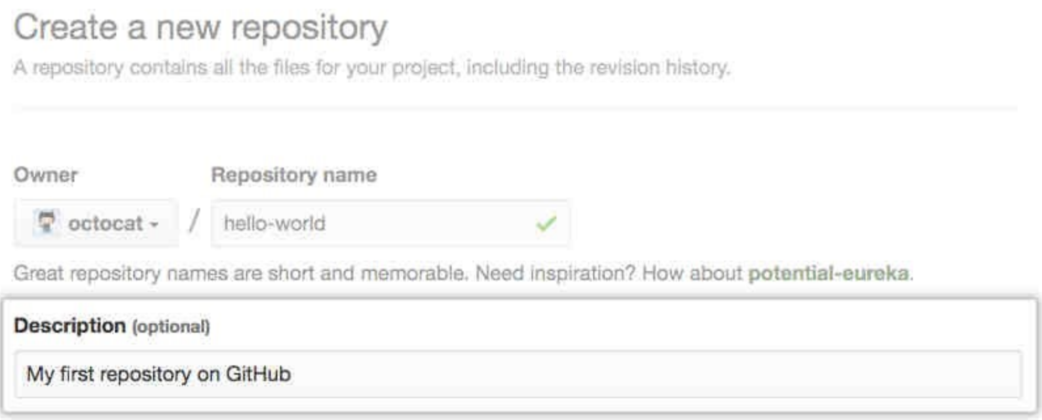
Owner: octocat / Repository name: hello-world

Great repository names are short and memorable. Need inspiration? How about potential-eureka.

Description (optional)

Figure 20: Create a new repository

3. Optionally, add a description of your repository. For example, "My first repository on GitHub."


Create a new repository

A repository contains all the files for your project, including the revision history.

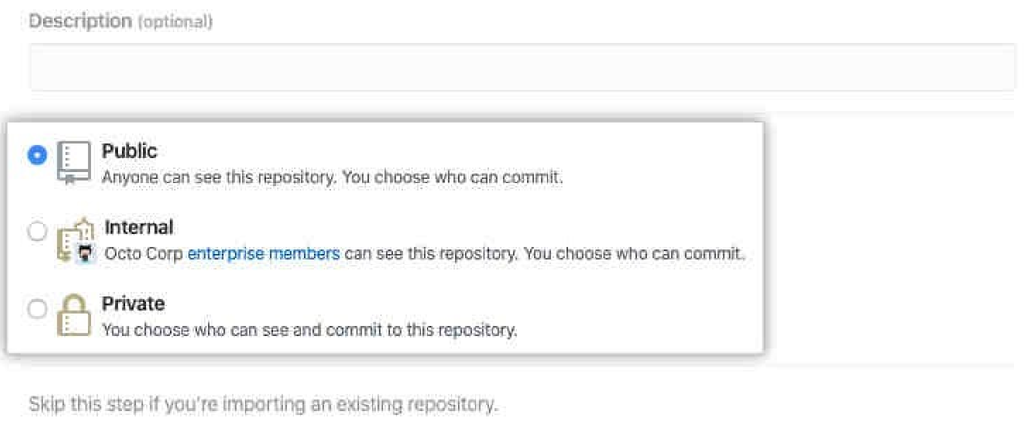
Owner: octocat / Repository name: hello-world

Great repository names are short and memorable. Need inspiration? How about potential-eureka.

Description (optional): My first repository on GitHub

Figure 21: my first repository on github

4. Choose a repository visibility. For more information, see "About repository visibility."


Description (optional)

[● Public] Anyone can see this repository. You choose who can commit.

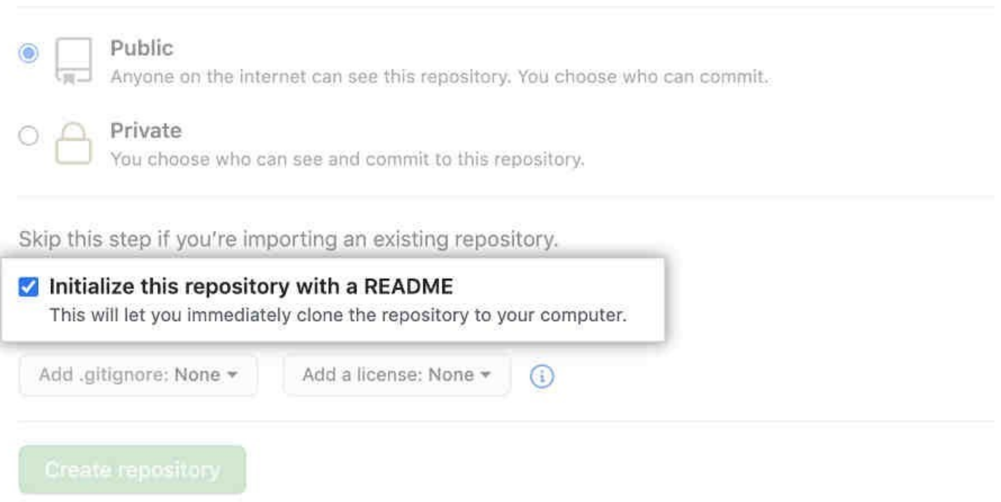
Internal
Octo Corp enterprise members can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Figure 22: About repository visibility

5. Select Initialize this repository with a README.


[● Public] Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

[✓ Initialize this repository with a README]
This will let you immediately clone the repository to your computer.

Figure 23: Initialize this repository

6. Click Create repository

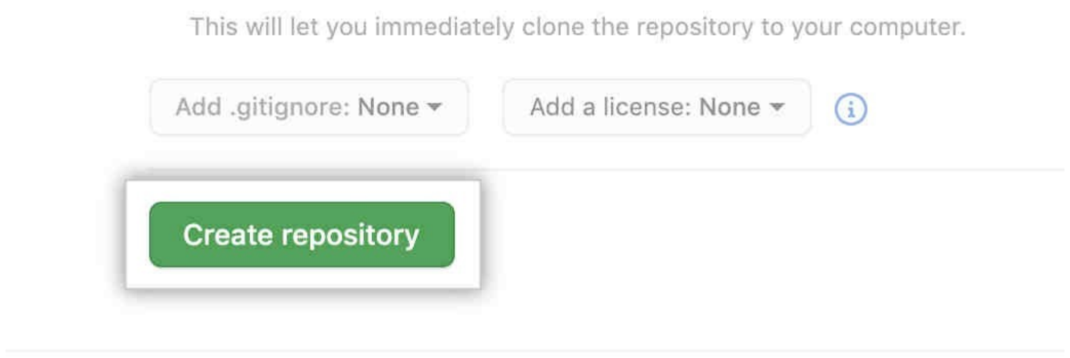

This will let you immediately clone the repository to your computer.

Figure 24: Create repository

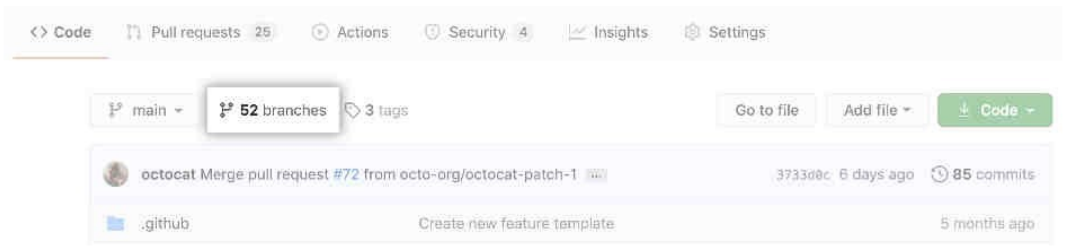
Congratulations! You've successfully created your first repository, and initialized it with a README file.

PRACTICAL 4

Create/Delete/Merge Branches

CREATE BRANCH

1. On GitHub, navigate to the main page of the repository.
2. Optionally, if you want to create your new branch from a branch other than the default branch for the repository, click "fork" NUMBER branches then choose another branch:



octocat Merge pull request #72 from octo-org/octocat-patch-1 … 3733d0c 6 days ago 85 commits

.github Create new feature template 5 months ago

Figure 25: Create new branch

3. Click the branch selector menu.

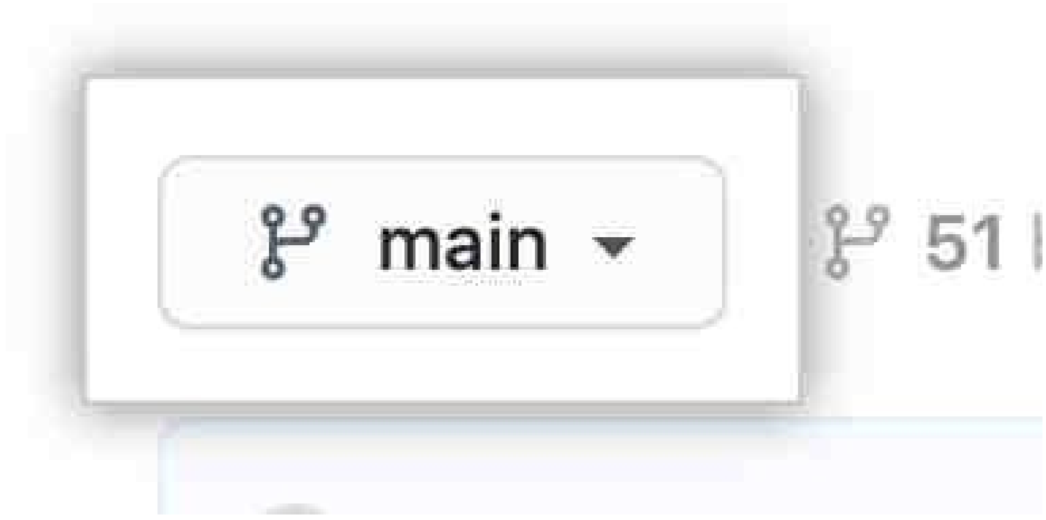


Figure 26: branch selector

4. Type a unique name for your new branch, then select Create branch.

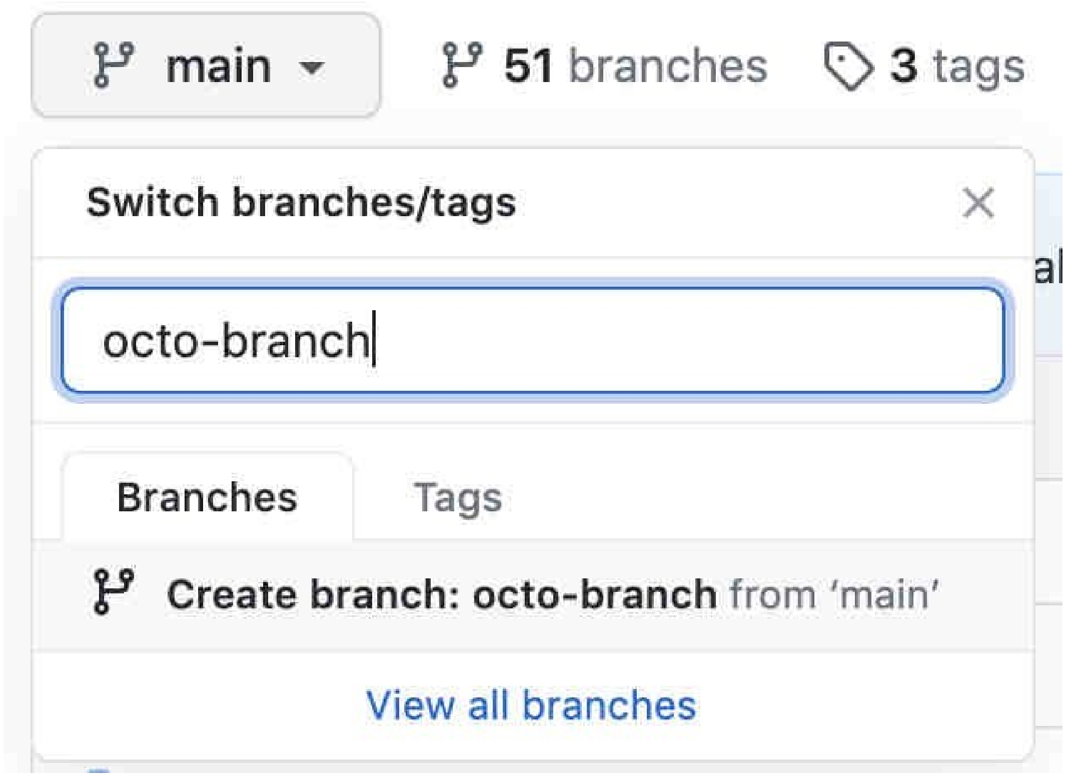


Figure 27: branch name

DELETE BRANCH

1. On GitHub, navigate to the main page of the repository.
2. Above the list of files, click on "fork " NUMBER branches.

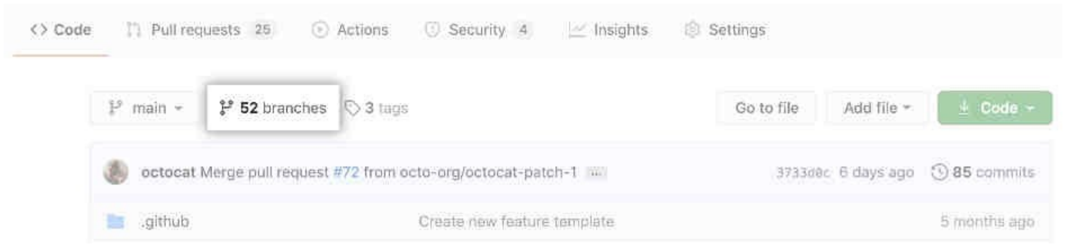


Figure 28: Delete branch

3. Scroll to the branch that you want to delete, then click to delete.

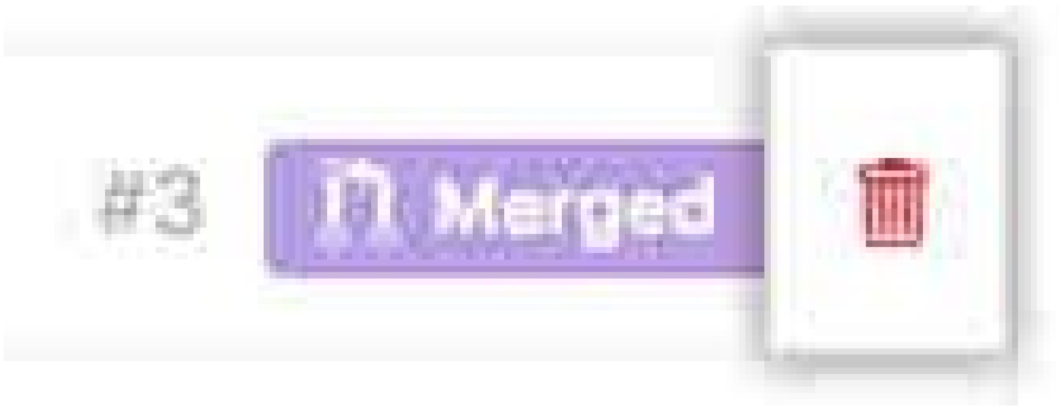
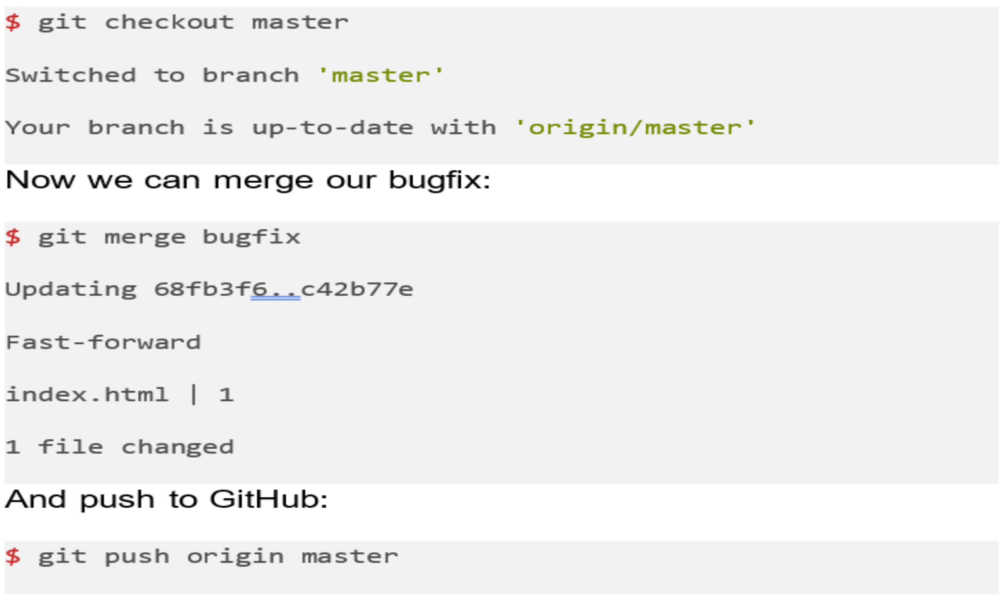


Figure 29: finally delete

If you delete a head branch after its pull request has been merged, GitHub checks for any open pull requests in the same repository that specify the deleted branch as their base branch. GitHub automatically updates any such pull requests, changing their base branch to the merged pull request's base branch.

MERGE BRANCH

We have tested the fix and we are happy, so let's merge this change into master. But first, we need to switch back to our master branch:



```
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'
```

Now we can merge our bugfix:

```
$ git merge bugfix
Updating 68fb3f6..c42b77e
Fast-forward
index.html | 1
1 file changed
```

And push to GitHub:

```
$ git push origin master
```

Figure 30: merge branch

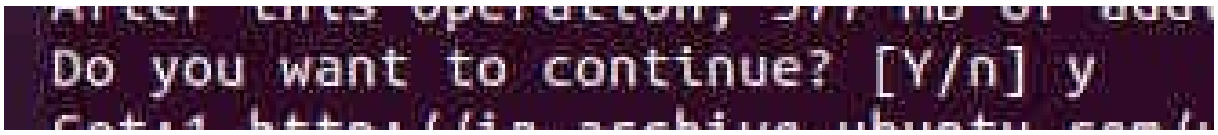
Total 0 (delta 0), reused 0 (delta 0)

PRACTICAL 5
Install Docker

STEP-1 Download docker by using standard ubuntu library. Using the command

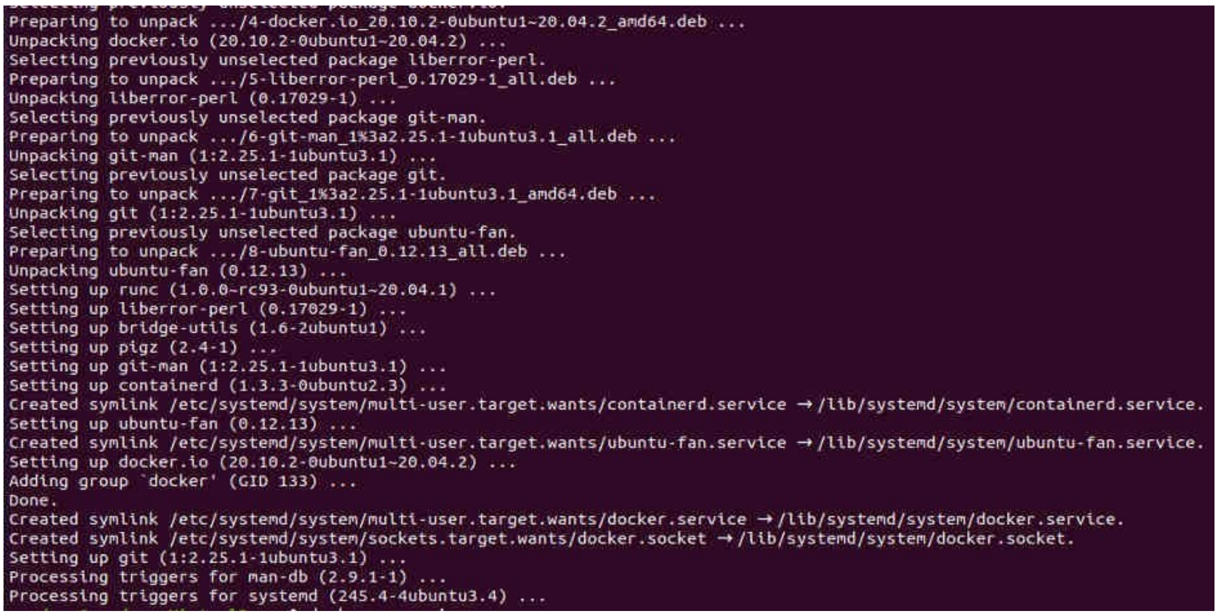
```
$ sudo apt install docker.io
```

STEP-2 Press y to continue



```
Do you want to continue? [Y/n] y
```

STEP-3 Wait for this processing to finish.



```
Preparing to unpack .../4-docker.io_20.10.2-0ubuntu1~20.04.2_amd64.deb ...
Unpacking docker.io (20.10.2-0ubuntu1~20.04.2) ...
Selecting previously unselected package liberror-perl.
Preparing to unpack .../5-liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../6-git-man_1%3a2.25.1-1ubuntu3.1_all.deb ...
Unpacking git-man (1:2.25.1-1ubuntu3.1) ...
Selecting previously unselected package git.
Preparing to unpack .../7-git_1%3a2.25.1-1ubuntu3.1_amd64.deb ...
Unpacking git (1:2.25.1-1ubuntu3.1) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../8-ubuntu-fan_0.12.13_all.deb ...
Unpacking ubuntu-fan (0.12.13) ...
Setting up runc (1.0.0~rc93-0ubuntu1~20.04.1) ...
Setting up liberror-perl (0.17029-1) ...
Setting up bridge-utils (1.6-2ubuntu1) ...
Setting up pigz (2.4-1) ...
Setting up git-man (1:2.25.1-1ubuntu3.1) ...
Setting up containerd (1.3.3-0ubuntu2.3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.13) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (20.10.2-0ubuntu1~20.04.2) ...
Adding group `docker' (GID 133) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Setting up git (1:2.25.1-1ubuntu3.1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.4) ...
```

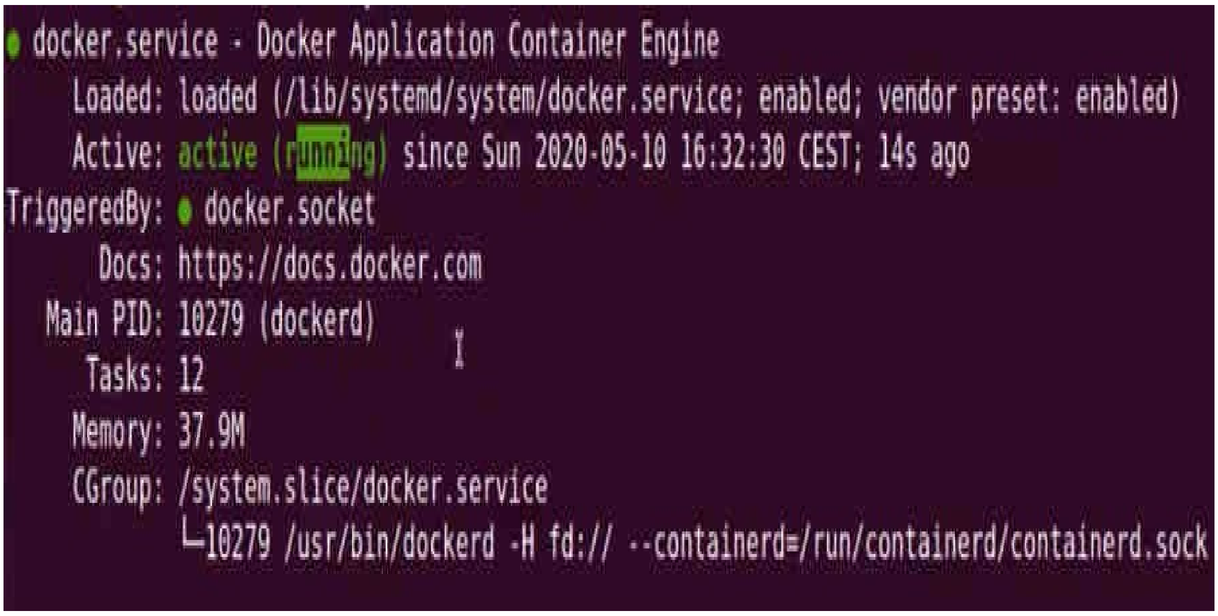
STEP-4 After finishing the processing let's check the version .

```
$ docker --version
```

STEP-5 Let's check this docker is enabled or un-enabled

```
$ sudo systemctl status docker
```

We found that our docker is in active(running) state. **STEP-6**Let's just test whether our docker



```
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2020-05-10 16:32:30 CEST; 14s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 10279 (dockerd)
      Tasks: 12
     Memory: 37.9M
     CGroup: /system.slice/docker.service
             └─10279 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

is working in our ubuntu.

```
$ sudo docker run hello-world
```

This command does what?...
Docker first find this hello-world image in our local ubuntu operating system. First it will try to find hello-world latest image but it was not found. Then it will going to pull the hello-world image from docker hub repository. After that pull is complete then a message saying Hello from docker indicates everything is working properly.
STEP-7 Use command sudo docker images. Which will give the result of the image present that we had pulled from docker hub recently.

PRACTICAL 6

Deploy Nginx Web Server Image on Docker

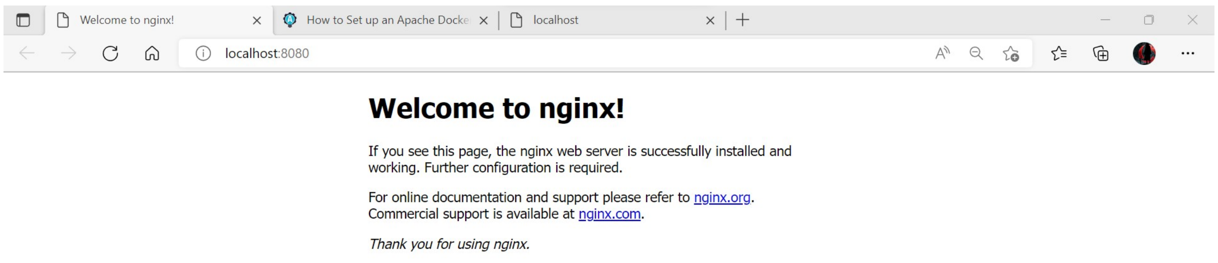
STEP-1
Let's pull the image of Nginx. Use command.

```
$ sudo docker pull nginx:latest
```

STEP-2
Let's check any running containers. We have one container that were created before but not used.
STEP-3
To run Nginx container inside a container.

```
$ sudo docker run -p 8000:80 nginx
```

We first write **sudo docker** and a flag -p to map the port inside the container to our local machine. So first provide port i.e on my local machine **8000** and then need to provide the port on the container 80 , because Nginx by default runs on port 80 and lastly provide the name of the image i.e Nginx. So as we can see we are running the container in for-ground mode So let's go to the chrome server and open **local:8000**. We can see the default web page of nginx web server.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Figure 31: ngnix default page

If we are seeing this page then it means container is running fine.
In the above picture we can see the logs of nginx container now to stop them , press Ctrl C.
STEP-4
Now if do **docker ps** , there np running containers (except onlt that one that I created long time ago.)
STEP-5

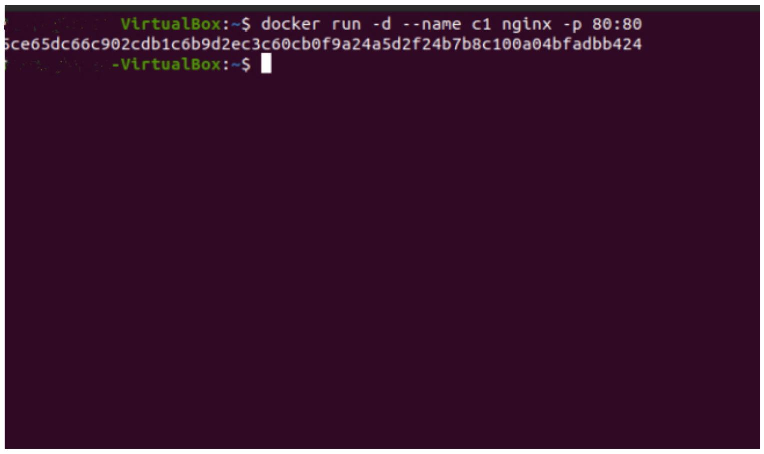
Now if we do bfdocker ps -a We see the container that we created my-nginx existed 21 seconds ago.
STEP-6
Let's remove the container completely

```
$ sudo docker rm with container id
```

PRACTICAL 7

Deploy Apache Web Server Image on Docker

STEP-1
Start the docker using the command systemctl start docker
STEP-2
Then we need to pull httpd image from DockerHub using the command docker image pull httpd
STEP-3
To check the image id downloaded use the command docker image ls
STEP-4
To run the container with httpd image use the command docker run -d –name c2 -p 80:80

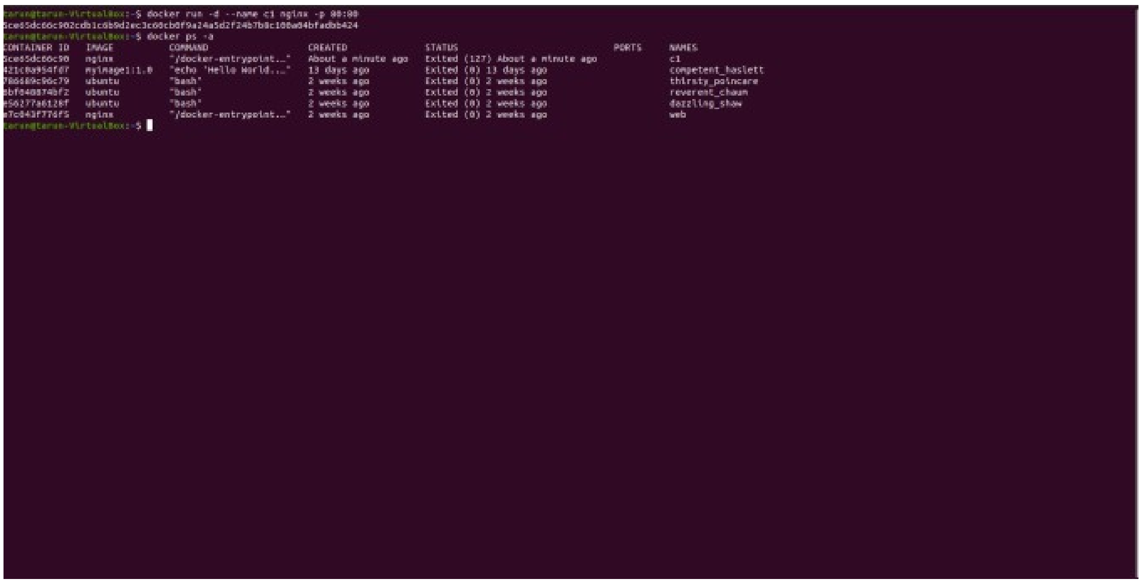


```
[redacted] VirtualBox:~$ docker run -d --name c1 nginx -p 80:80
5ce65dc66c902cdb1c6b9d2ec3c60cb0f9a24a5d2f24b7b8c100a04bfadbb424
[redacted]-VirtualBox:~$ 
```

Figure 32: run the container

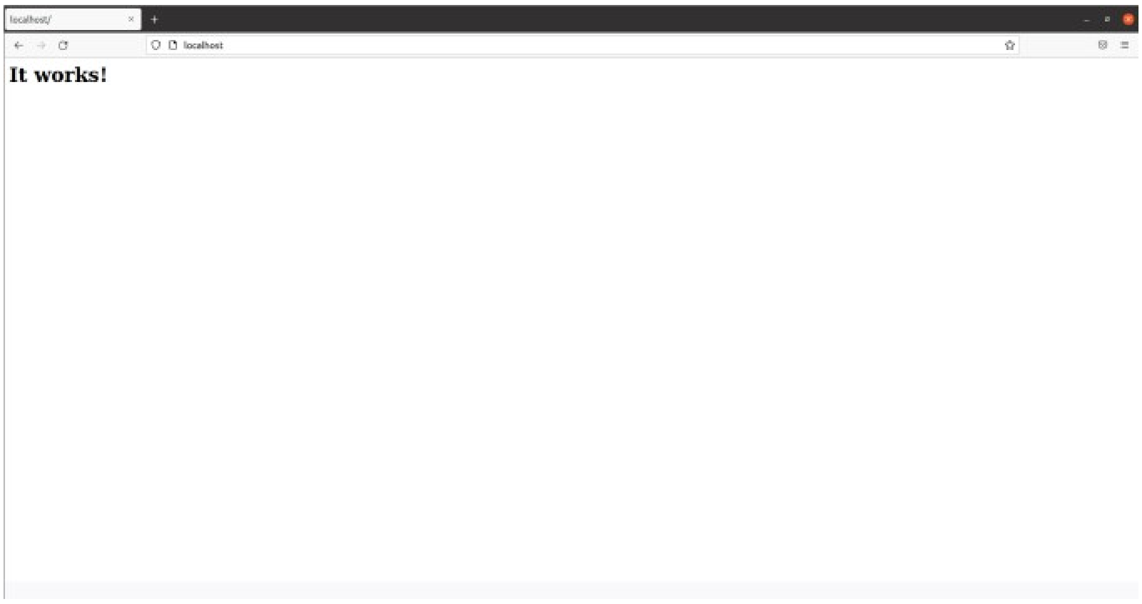
STEP-5
To see the list of containers created use the command docker ps -a

STEP-6
Then open the we browser and type localhost:80/



```
tarun@tarun-VirtualBox:~$ docker run -d --name c1 nginx -p 80:80
5ce65dc66c902cdb1c6b9d2ec3c60cb0f9a24a5d2f24b7b8c100a04bfadbb424
tarun@tarun-VirtualBox:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS                             PORTS     NAMES
5ce65dc66c90   nginx          "/docker-entrypoint.…"   About a minute ago   Exited (127) About a minute ago              c1
421c0a954f07   myimage1:1.0   "echo 'Hello World…"     13 days ago          Exited (0) 13 days ago                       competent_haslett
786689c96c79   ubuntu         "bash"                   2 weeks ago          Exited (0) 2 weeks ago                       thirsty_poincare
bbf0408874bf2  ubuntu         "bash"                   2 weeks ago          Exited (0) 2 weeks ago                       reverent_chaum
e56277a6128f   ubuntu         "bash"                   2 weeks ago          Exited (0) 2 weeks ago                       dazzling_shaw
a7c043f776f5   nginx          "/docker-entrypoint.…"   2 weeks ago          Exited (0) 2 weeks ago                       web
tarun@tarun-VirtualBox:~$ 
```

Figure 33: list the containers



It works!

Figure 34: web browser output

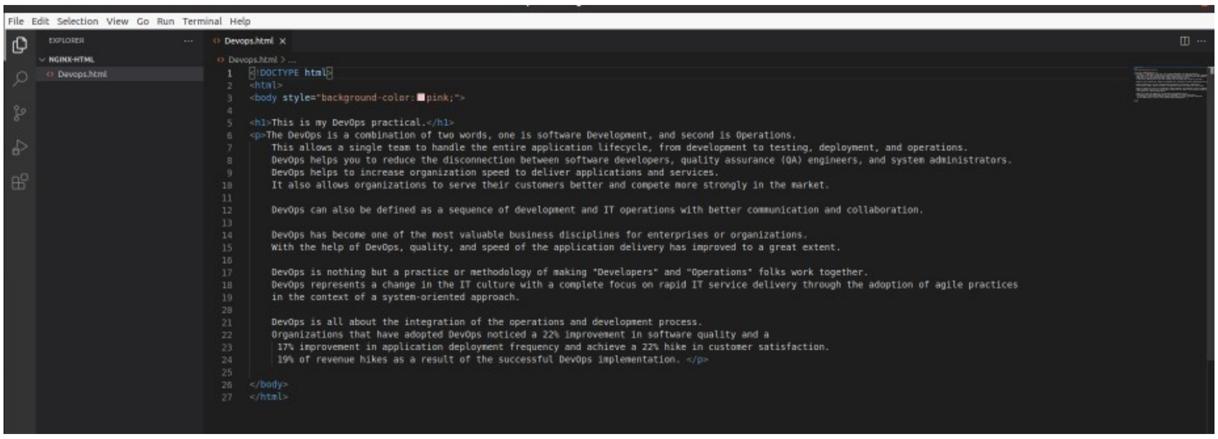
PRACTICAL 8

Create custom page using web server

STEP-1
Run command:

```
$ mkdir nginx-html
$ cd nginx-html
```

STEP-2
After creating the Devops.html file, we start with the coding. For html coding here we use Visual studio code.
Run command: code . when we write this command the visual studio will open their we can see the Devops.html file present. After that start with creating the simple web page.



```
<!DOCTYPE html>
<html>
<body style="background-color: pink;">

<h1>This is my DevOps practical.</h1>
<p>The DevOps is a combination of two words, one is software Development, and second is Operations.
    This allows a single team to handle the entire application lifecycle, from development to testing, deployment, and operations.
    DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.
    DevOps helps to increase organization speed to deliver applications and services.
    It also allows organizations to serve their customers better and compete more strongly in the market.

    DevOps can also be defined as a sequence of development and IT operations with better communication and collaboration.

    DevOps has become one of the most valuable business disciplines for enterprises or organizations.
    With the help of DevOps, quality, and speed of the application delivery has improved to a great extent.

    DevOps is nothing but a practice or methodology of making "Developers" and "Operations" folks work together.
    DevOps represents a change in the IT culture with a complete focus on rapid IT service delivery through the adoption of agile practices
    in the context of a system-oriented approach.

    DevOps is all about the integration of the operations and development process.
    Organizations that have adopted DevOps noticed a 22% improvement in software quality and a
     17% improvement in application deployment frequency and achieve a 22% hike in customer satisfaction.
     19% of revenue hikes as a result of the successful DevOps implementation. </p>

</body>
</html>
```

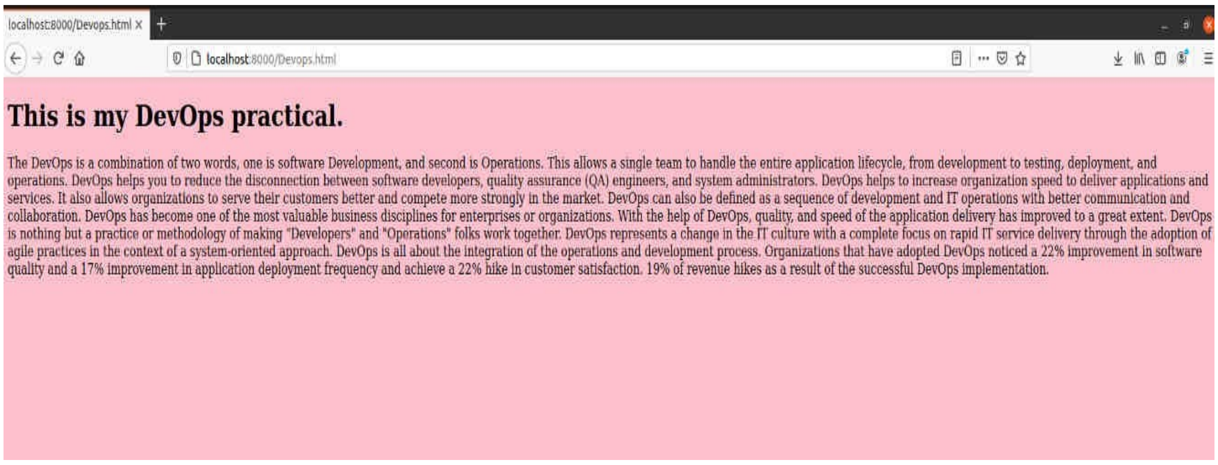
Figure 35: nginx HTML page

STEP-3
Run command:

```
$ sudo docker run -d -p 8000:80 -v ~/nginx-html:/usr/share/nginx/html
{name my-nginx
```

This command will help to print the output of the Devops.html file on the nginx web server.

OUTPUT



This is my DevOps practical.

The DevOps is a combination of two words, one is software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from development to testing, deployment, and operations. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators. DevOps helps to increase organization speed to deliver applications and services. It also allows organizations to serve their customers better and compete more strongly in the market. DevOps can also be defined as a sequence of development and IT operations with better communication and collaboration. DevOps has become one of the most valuable business disciplines for enterprises or organizations. With the help of DevOps, quality, and speed of the application delivery has improved to a great extent. DevOps is nothing but a practice or methodology of making "Developers" and "Operations" folks work together. DevOps represents a change in the IT culture with a complete focus on rapid IT service delivery through the adoption of agile practices in the context of a system-oriented approach. DevOps is all about the integration of the operations and development process. Organizations that have adopted DevOps noticed a 22% improvement in software quality and a 17% improvement in application deployment frequency and achieve a 22% hike in customer satisfaction. 19% of revenue hikes as a result of the successful DevOps implementation.

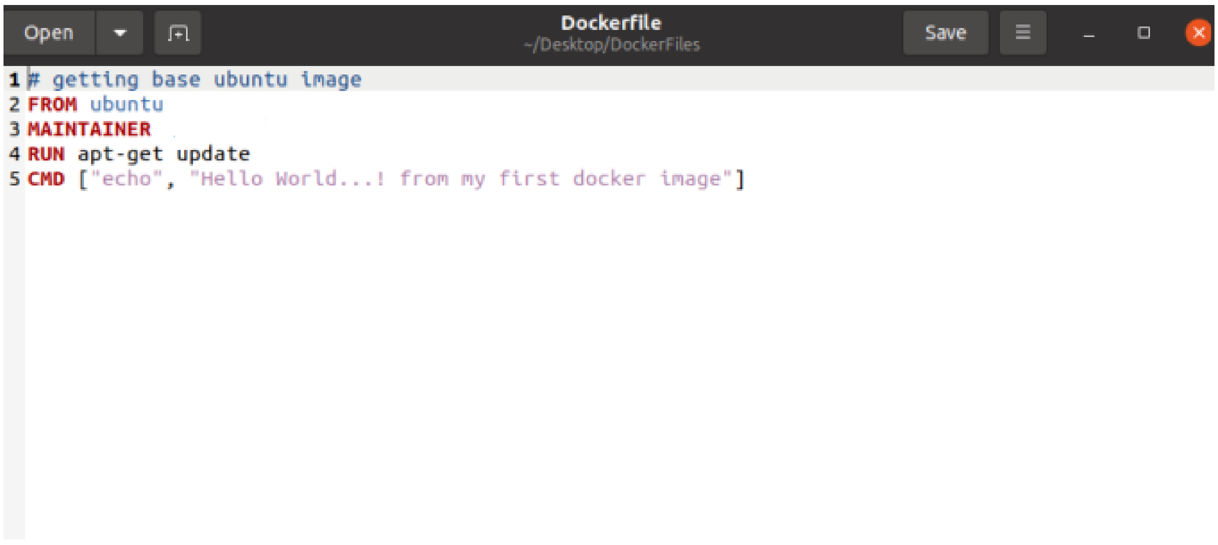
Figure 36: nginx custom web page

PRACTICAL 9

Create Custom Image

1. Docker builds the docker image by reading the instructions from a text file. By default Docker looks for a file named Dockerfile to build the docker image. The Dockerfile consists of instructions that are used to customize the docker image.
2. Create a Dockerfile using the following command in the terminal.

```
touch Dockerfile
```



```
# getting base ubuntu image
FROM ubuntu
MAINTAINER
RUN apt-get update
CMD ["echo", "Hello World...! from my first docker image"]
```

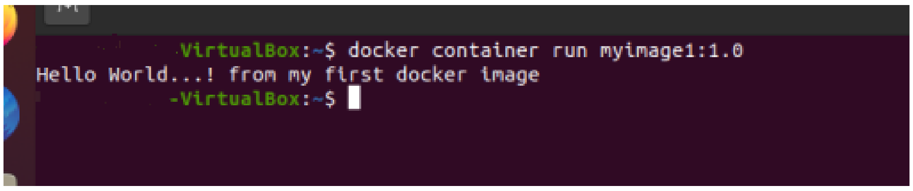
Figure 37: dockerfie

3. After writing the Dockerfile create the image using following command in the terminal.

```
docker build .
```

4. Run the image using following command in the terminal

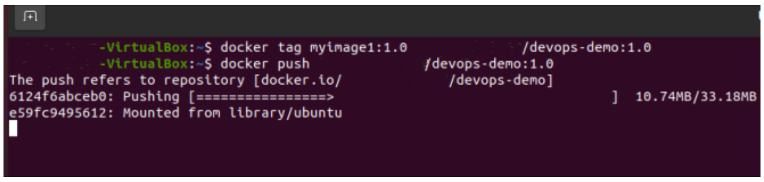
```
docker container run <image_name>
```



```
[redacted]VirtualBox:~$ docker container run myimage1:1.0
Hello World...! from my first docker image
[redacted]-VirtualBox:~$ 
```

Figure 38: run the custom image

5. Put the tag on the image to push it remote repository



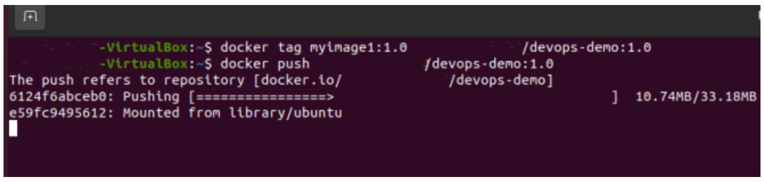
```
[redacted]-VirtualBox:~$ docker tag myimage1:1.0 [redacted]/devops-demo:1.0
[redacted]-VirtualBox:~$ docker push [redacted]/devops-demo:1.0
The push refers to repository [docker.io/[redacted]/devops-demo]
6124f6abceb0: Pushing [================>                                  ]  10.74MB/33.18MB
e59fc9495612: Mounted from library/ubuntu
```

Figure 39: put tag to image

PRACTICAL 10
Push Custom Image to Docker Hub

1. To push the docker image to DockerHub or some other docker image repository, the docker image must be tagged properly. If you want to push a docker image to a docker hub repository, the docker image must be tagged as follows:

```
docker tag myimage1:1.0 username/devops-demo:1.0
```

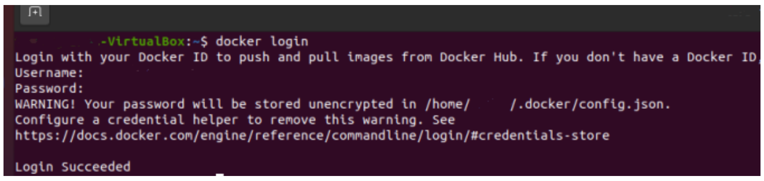


```
-VirtualBox:~$ docker tag myimage1:1.0            /devops-demo:1.0
-VirtualBox:~$ docker push              /devops-demo:1.0
The push refers to repository [docker.io/             /devops-demo]
6124f6abceb0: Pushing [================>                                  ]  10.74MB/33.18MB
e59fc9495612: Mounted from library/ubuntu
```

Figure 40: tag the custom image

2.Before pushing the docker image to docker hub, you must log into the docker hub using command line. Use the following command to log into docker hub.

```
docker login
```



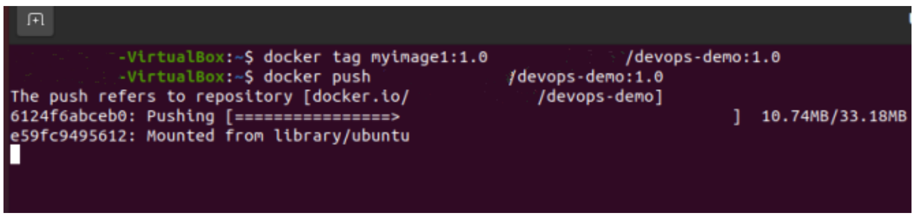
```
-VirtualBox:~$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID
Username:
Password:
WARNING! Your password will be stored unencrypted in /home/      /.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Figure 41: login to docker hub

4. Push the image to docker hub using the following command

```
docker push username/devops-demo:1.0
```



```
-VirtualBox:~$ docker tag myimage1:1.0            /devops-demo:1.0
-VirtualBox:~$ docker push              /devops-demo:1.0
The push refers to repository [docker.io/             /devops-demo]
6124f6abceb0: Pushing [================>                                  ]  10.74MB/33.18MB
e59fc9495612: Mounted from library/ubuntu
```

Figure 42: push the image