

NAME: SAYALI BABAN CHOWKEKAR

ROLLNO: 6119025

BRANCH: TE(IT)

SUB: DEVOPS PRACTICAL

EXPERIMENT NO. 1

AIM: Case study on Docker.

THEORY:

1. What is docker?

- Docker is an open source platform for building, deploying, and managing containerized applications. Docker takes away repetitive, mundane configuration tasks and is used throughout the development lifecycle for fast, easy and portable application development - desktop and cloud. Docker's comprehensive end to end platform includes UIs, CLIs, APIs and security that are engineered to work together across the entire application delivery lifecycle. It enables developers to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. Containers simplify delivery of distributed applications, and have become increasingly popular as organizations shift to cloud-native development and hybrid multicloud environments.
Developers can create containers without Docker, but the platform makes it easier, simpler, and safer to build, deploy and manage containers. Docker is essentially a toolkit that enables developers to build, deploy, run, update, and stop containers using simple commands and work-saving automation through a single API.

2. Application of Docker.

Docker applications:

Improved and seamless portability: While LXC containers often reference machine-specific configurations, Docker containers run without modification across any desktop, data center and cloud environment.

Even lighter weight and more granular updates: With LXC, multiple processes can be combined within a single container. With Docker containers, only one process can run in each container. This makes it possible to build an application that can continue running while one of its parts is taken down for an update or repair.

Automated container creation: Docker can automatically build a container based on application source code.

Container versioning: Docker can track versions of a container image, roll back to previous versions, and trace who built a version and how. It can even upload only the deltas between an existing version and a new one.

Container reuse: Existing containers can be used as base images—essentially like templates for building new containers.

Shared container libraries: Developers can access an open-source registry containing thousands of user-contributed containers.

3. Pros and cons of docker.

➤ Advantages of docker

a. Return on Investment and Cost Savings

Dockers first advantage is ROI. Especially for large, established companies, which need to generate steady revenue over the long term, the solution is only better if it can drive down costs while raising profits.

b. Rapid Deployment

It can decrease deployment to seconds. It is because of the fact that it can create a container for every process and even does not boot an OS. So, even without worrying about the cost to bring it up again, it would be higher than what is affordable, Data can be created as well as destroyed.

c. Security

Docker makes sure that applications that are running on containers are completely segregated and isolated from each other, from a security point of view, by granting us complete control over traffic flow and management.

d. Simplicity and Faster Configurations

The way Docker simplifies the matters is one of the key benefits of it. It gives flexibility to users to take their own configuration, put that into the code, and further deploy it without any problems.

However, the requirements of the infrastructure are no longer linked with the environment of the application, as Docker can be used in a wide variety of environments.

e. CI Efficiency

With the help of a Docker, we can build a container image and can further use that same image over every step of the deployment process.

The advantage of it is the ability to separate non-dependent steps and also run them in parallel. In addition, the duration of time it takes from build to production may speed up notably.

f. Continuous Integration

While it comes to Continuous Integration, Docker works well as part of its pipelines along with tools such as Travis, Jenkins, and Wrecker.

These tools can save the new version as a Docker image, every time our source code is updated, just tag it with a version number and push to Docker Hub, then deploy it to production.

Disadvantages of docker

a. Missing features

There are a ton of feature requests are under progress, like container self-registration, and self-inspects, copying files from the host to the container, and many more.

b. Data in the container

There are times when a container goes down, so after that, it needs a backup and recovery strategy, although we have several solutions for that they are not automated or not very scalable yet.

c. Run applications as fast as a bare-metal serve

In comparison with the virtual machines, Docker containers have less overhead but not zero overhead. If we run, an application directly on a bare-metal server we get true bare-metal speed even without using containers or virtual machines. However, Containers don't run at bare-metal speeds.

d. Provide cross-platform compatibility

The one major issue is if an application designed to run in a Docker container on Windows, then it can't run on Linux or vice versa. However, Virtual machines are not subject to this limitation.

So, this limitation makes Docker less attractive in some highly heterogeneous environments which are composed of both Windows and Linux servers.

e. Run applications with graphical interfaces

In general, Docker is designed for hosting applications which run on the command line. Though we have a few ways (like X11 forwarding) by which we can make it possible to run a graphical interface inside a Docker container, however, this is clunky.

Hence we can say, for applications that require rich interfaces, Docker is not a good solution.

f. Solve all your security problems

In simple words, we need to evaluate the Docker-specific security risks and make sure we can handle them before moving workloads to Docker.

The reason behind it is that Docker creates new security challenges like the difficulty of monitoring multiple moving pieces within a large-scale, dynamic Docker environment.

4. Platform dependency of Docker.

- You can run both Linux and Windows programs and executables in Docker containers. The Docker platform runs natively on Linux (on x86-64, ARM and many other CPU architectures) and on Windows (x86-64). Docker Inc. builds products that let you build and run containers on Linux, Windows and macOS.

5. Architecture of docker.

- The Docker daemon (dockerd) listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes. A daemon can also communicate with other daemons to manage Docker services.

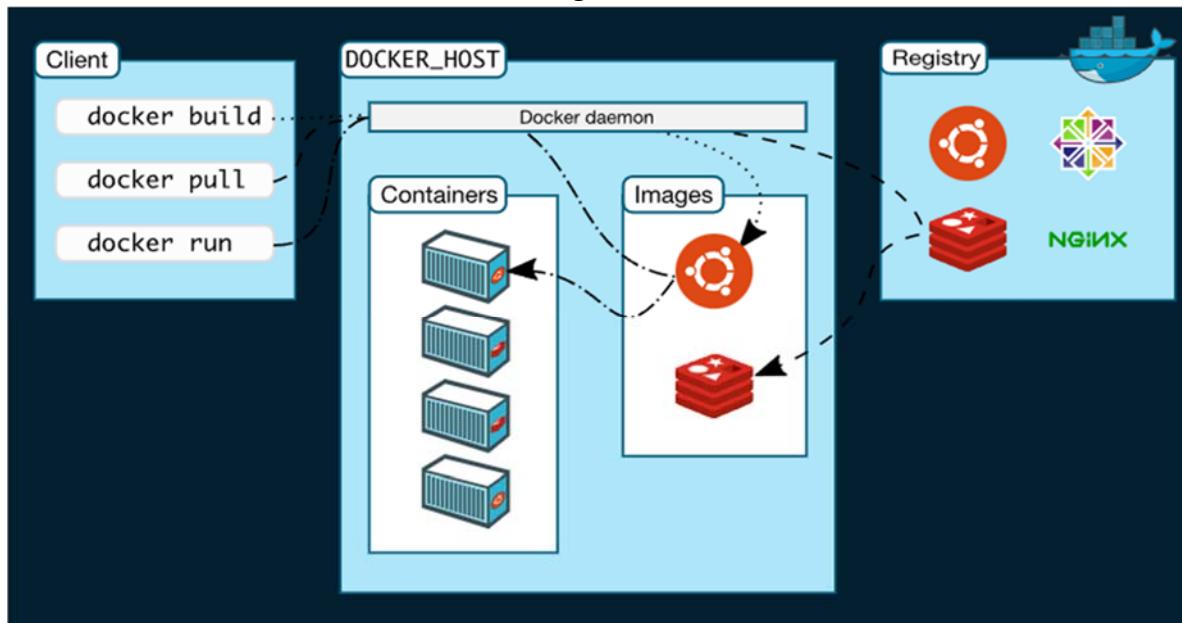


Fig. Docker architecture

The Docker client

The Docker client (`docker`) is the primary way that many Docker users interact with Docker. When you use commands such as `docker run`, the client sends these commands to `dockerd`, which carries them out. The `docker` command uses the Docker API. The Docker client can communicate with more than one daemon.

Docker registries

A Docker registry stores Docker images. Docker Hub is a public registry that anyone can use, and Docker is configured to look for images on Docker Hub by default. You can even run your own private registry.

When you use the `docker pull` or `docker run` commands, the required images are pulled from your configured registry. When you use the `docker push` command, your image is pushed to your configured registry.

Docker objects

When you use Docker, you are creating and using images, containers, networks, volumes, plugins, and other objects. This section is a brief overview of some of those objects.

Images

An image is a read-only template with instructions for creating a Docker container. Often, an image is based on another image, with some additional customization. For example, you may build an image which is based on the `ubuntu` image, but installs the Apache web server and your application, as well as the configuration details needed to make your application run.

You might create your own images or you might only use those created by others and published in a registry. To build your own image, you create a Dockerfile with a simple syntax for defining the steps needed to create the image and run it. Each instruction in a Dockerfile creates a layer in the image. When you change the Dockerfile and rebuild the image, only those layers which have changed are rebuilt. This is part of what makes images so lightweight, small, and fast, when compared to other virtualization technologies.

Containers

A container is a runnable instance of an image. You can create, start, stop, move, or delete a container using the Docker API or CLI. You can connect a container to one or more networks, attach storage to it, or even create a new image based on its current state.

By default, a container is relatively well isolated from other containers and its host machine. You can control how isolated a container's network, storage, or other underlying subsystems are from other containers or from the host machine.

A container is defined by its image as well as any configuration options you provide to it when you create or start it. When a container is removed, any changes to its state that are not stored in persistent storage disappear.

CONCLUSION: Hence, we have studied the docker in detail. So, learning them will definitely help us before using it. We have seen the various components of the Docker architecture and how they work together and also advantages and disadvantages in

NAME: SAYALI BABAN CHOWKEKAR

ROLLNO: 6119025

BRANCH: TE(IT)

SUB: DEVOPS PRACTICAL

EXPERIMENT NO.2

AIM: Install and configure Jenkins.

THEORY:

- Software requirements
 - Java: Java 11 runtime environments are supported.
 - Web browser:

Browser	Level 1	Level 2	Level 3
Google Chrome	Latest regular release/patch	Version N-1, latest patch	Other versions
Mozilla Firefox	Latest regular release/patch; Latest <u>ESR</u> release	Version N-1, latest patch	Other versions
Microsoft Internet Explorer	Version 11, latest patch	Version 11, previous patches	Other versions
Microsoft Edge	-	Latest release/patch	Other versions
Apple Safari	Latest regular release/patch	Version N-1, latest patch	Other versions

- Hardware requirements

Minimum hardware requirements:

256 MB of RAM

1 GB of drive space (although 10 GB is a recommended minimum if running Jenkins as a Docker container)

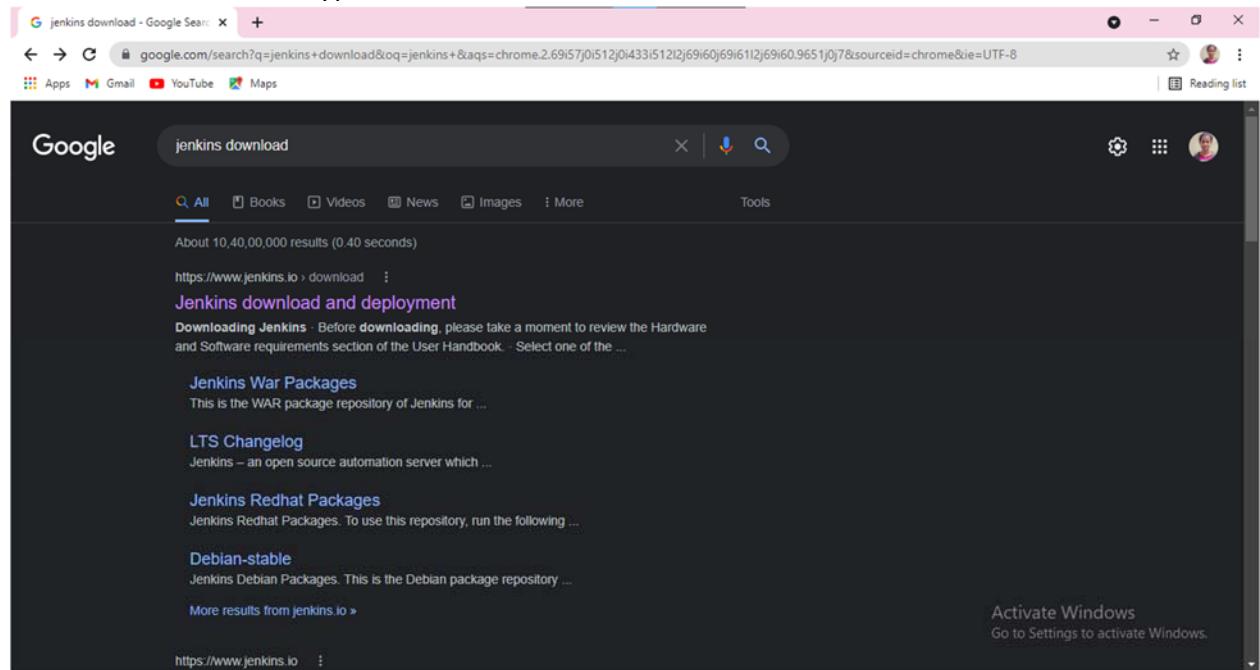
Recommended hardware configuration for a small team:

4 GB+ of RAM

50 GB+ of drive space

Steps of installing and configuring Jenkins:

1. Go to browser and type Jenkins download and then click on Jenkins.io website.



2. Now based on your operating system click on type of your OS of LTS version, eg: Windows.

Once a Jenkins package has been downloaded, proceed to the [Installing Jenkins](#) section of the User Handbook.
You may also want to verify the package you downloaded. [Learn more about verifying Jenkins downloads.](#)

Download Jenkins 2.303.1 LTS for:

- Generic Java package (.war)
- Docker
- Ubuntu/Debian
- CentOS/Fedora/Red Hat
- Windows
- openSUSE
- FreeBSD
- Gentoo
- macOS

Download Jenkins 2.314 for:

- Generic Java package (.war)
- Docker
- Ubuntu/Debian
- CentOS/Fedora/Red Hat
- Windows
- openSUSE
- Arch Linux
- FreeBSD
- Gentoo
- macOS

Activate Windows
Go to Settings to activate Windows.

3. Now your Jenkins will start downloading.

Thank you for downloading Windows installer

Download hasn't started? Click this link [Twitter](#)

Changing boot configuration

By default, your Jenkins runs at <https://localhost:8080/>. This can be changed by editing `jenkins.xml`, which is located in your installation directory. This file is also the place to change other boot configuration parameters, such as JVM options, HTTPS setup, etc.

Starting/stopping the service

Jenkins is installed as a Windows service, and it is configured to start automatically upon boot. To start/stop them manually, use the service manager from the control panel, or the `sc` command line tool.

Inheriting your existing Jenkins installation

If you'd like your new installation to take over your existing Jenkins data, copy your old data directory into the new `JENKINS_HOME` directory.

See Also

- Running Jenkins behind Internet Information Server (IIS)

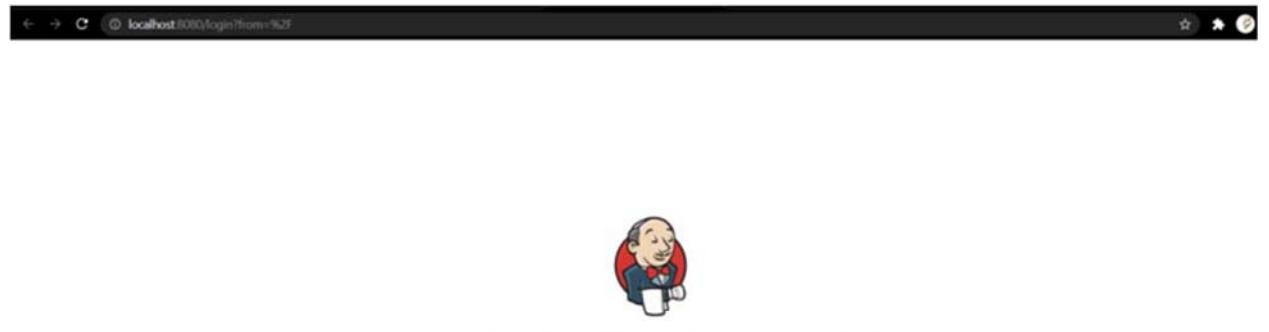
Activate Windows
Go to Settings to activate Windows
Show all

- Now go to your command prompt and type java -jar Jenkins.war. hence your Jenkins will get installed.

```
Command Prompt - java -jarjenkins.war
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sayali Chowkar>java -jar jenkins.war
Running from: C:\Users\Sayali Chowkar\jenkins.war
webroot: $user.home/.jenkins
2021-10-02 15:44:54.406+0000 [id=1] INFO org.eclipse.jetty.util.log.Log#initialized: Logging initialized @6530ms
to org.eclipse.jetty.util.log.JavaUtilLog
2021-10-02 15:44:55.063+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2021-10-02 15:44:55.304+0000 [id=1] WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2021-10-02 15:44:55.581+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-9.4.41.v20210516; built:
2021-05-16T23:56:28.993Z; git: 98e07f93c7833e7dc59489b13f3cb0a114fb9f4c; jvm 11.0.12+8-LTS-237
2021-10-02 15:45:01.672+0000 [id=1] INFO o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /,
did not find org.eclipse.jetty.jsp.JettyJspServlet
2021-10-02 15:45:01.937+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: DefaultSessionIdManager worke
rName=node0
2021-10-02 15:45:01.937+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: No SessionScavenger set, usin
g defaults
2021-10-02 15:45:01.944+0000 [id=1] INFO o.e.j.server.session.HouseKeeper#startScavenging: node0 Scavenging every
60000ms
2021-10-02 15:45:03.604+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: C:\Users\S
ayali Chowkar\.jenkins found at: $user.home/.jenkins
2021-10-02 15:45:04.515+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started w.@5300f14a{Jenkins v2.2
89.3,,file:///C:/Users/Sayali%20Chowkar/.jenkins/war/,AVAILABLE}{C:\Users\Sayali Chowkar\jenkins\war}
2021-10-02 15:45:04.640+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started ServerConnector@399c4be1
{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
2021-10-02 15:45:04.640+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started @16816ms
2021-10-02 15:45:04.640+0000 [id=26] INFO winstone.Logger#logInternal: Winstone Servlet Engine running: controlPor
t=disabled
2021-10-02 15:45:06.061+0000 [id=33] INFO jenkins.InitReactorRunner$1#onAttained: Started initialization Activate
```

- Go to browser and type <http://localhost:8080> and press enter.



6. Select install suggested Jenkins.

The screenshot shows the Jenkins 'Getting Started' page with the title 'Customize Jenkins'. Below the title, a sub-header says 'Plugins extend Jenkins with additional features to support many different needs.' There are two main options presented in boxes: 'Install suggested plugins' (selected) and 'Select plugins to install'. The 'Install suggested plugins' box contains the text 'Install plugins the Jenkins community finds most useful.' The 'Select plugins to install' box contains the text 'Select and install plugins most suitable for your needs.' At the bottom left, it says 'Jenkins 2.303.1'.

7. Now Jenkins is getting statrd.

The screenshot shows the Jenkins 'Getting Started' page with the title 'Getting Started'. Below the title, there is a grid of plugin icons and names. The grid is organized into four columns. A vertical column on the right side indicates dependencies. At the bottom right, there is a note about required dependencies and an 'Activate' button. At the bottom left, it says 'Jenkins 2.303.1'.

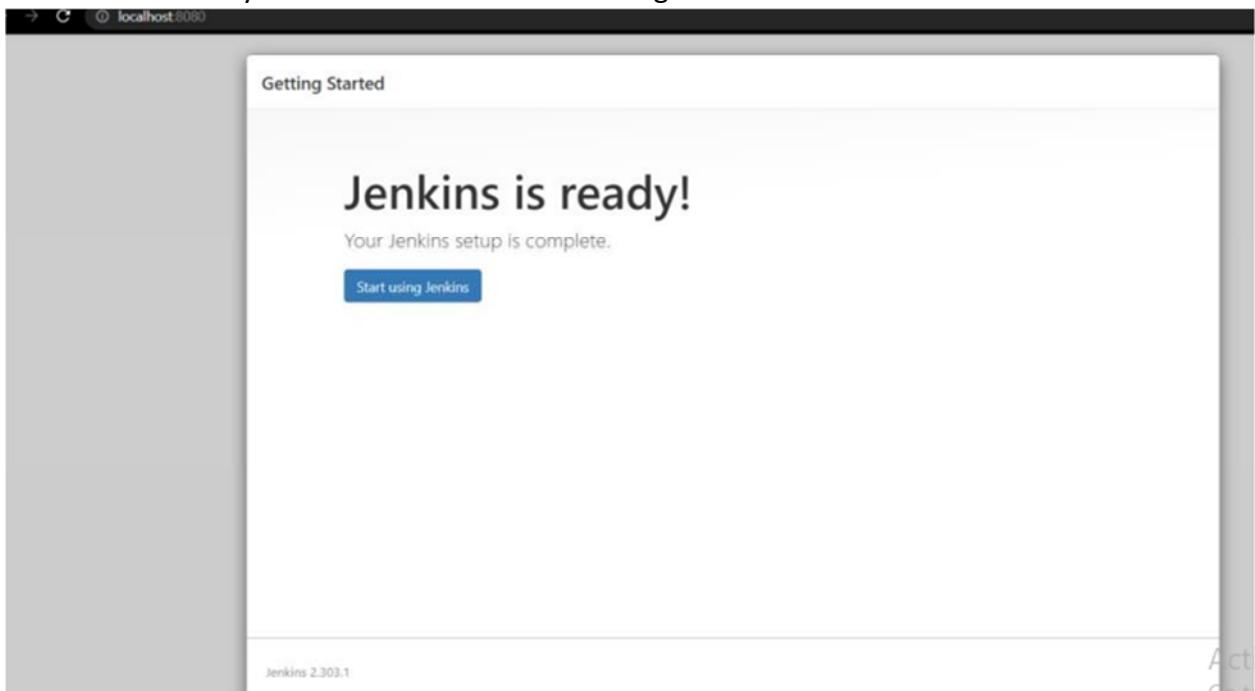
Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding	** SSH server
Timestamper	Workspace Cleanup	Ant	Gradle	
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View	
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication	
LDAP	Email Extension	Mailer		

** - required dependency

Activate

Jenkins 2.303.1

8. Jenkins is ready to use now. Click on Start using Jenkins.



9. After login you will get this Jenkins page.

A screenshot of a web browser showing the Jenkins Dashboard. The title bar shows multiple tabs including 'DEVOPS - Google Docs', 'Meet - mku-cwqh-hbv', 'Dashboard [Jenkins]', and 'Thank you for downloading Win...'. The main content area has a heading 'Welcome to Jenkins!'. It says 'This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.' Below this is a section titled 'Start building your software project' with a 'Create a job' button. Another section titled 'Set up a distributed build' has 'Set up an agent' and 'Configure a cloud' buttons. On the left sidebar, there are links for 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Lockable Resources', and 'New View'. A 'Build Queue' section shows 'No builds in the queue.' A 'Build Executor Status' section shows 'jenkins.msi' and 'jenkins (1).war'. A status bar at the bottom right says 'Activate Windows' and 'Go to Settings to activate Windows'. A 'Stop sharing' button is visible in the status bar.

CONCLUSION: Thus we have successfully installed and configured Jenkins.

NAME: SAYALI BABAN CHOWKEKAR

ROLLNO: 6119025

BRANCH: TE(IT)

SUB: DEVOPS PRACTICAL

EXPERIMENT NO.4

AIM: Installation of Mercurial software.

THEORY:

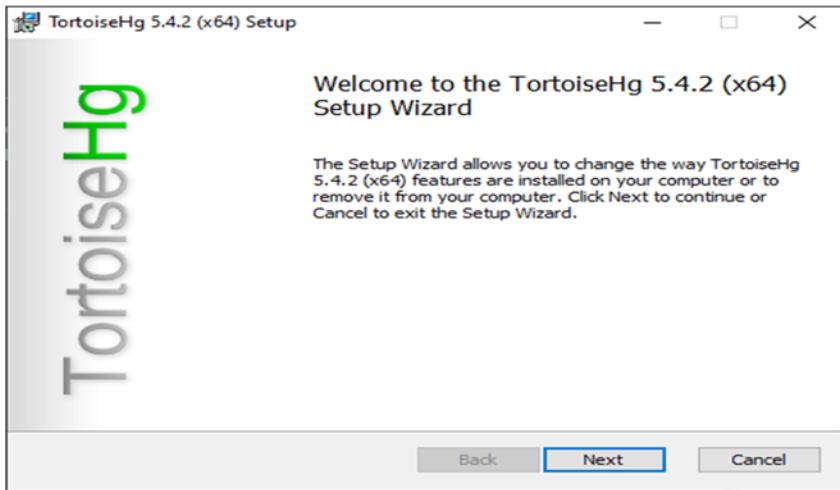
What is mercurial?

- Mercurial is a distributed revision control tool for software developers. It is supported on Microsoft Windows and Unix-like systems, such as FreeBSD, macOS, and Linux. Mercurial's major design goals include high performance and scalability, decentralization, fully distributed collaborative development, robust handling of both plain text and binary files, and advanced branching and merging capabilities, while remaining conceptually simple.^[4] It includes an integrated web-interface. Mercurial has also taken steps to ease the transition for users of other version control systems, particularly Subversion. Mercurial is primarily a command-line driven program, but graphical user interface extensions are available, e.g. TortoiseHg, and several IDEs offer support for version control with Mercurial. All of Mercurial's operations are invoked as arguments to its driver program hg (a reference to Hg – the chemical symbol of the element mercury)

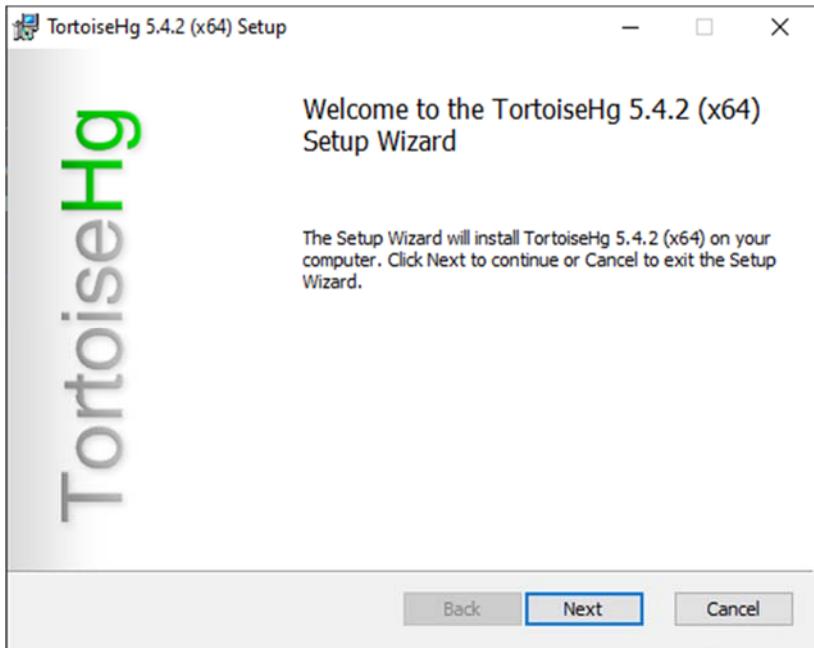
Steps of installing and setting up tortoisehg/mercurical software.



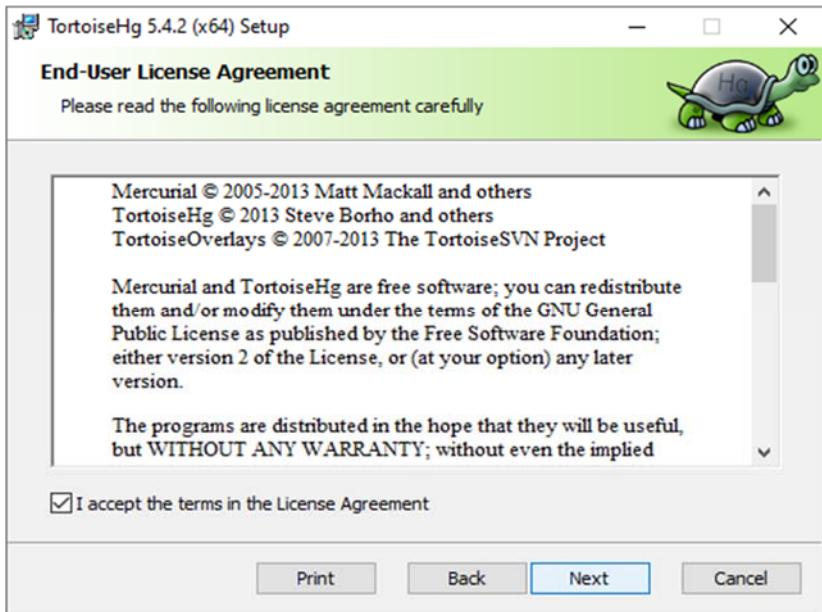
1. Installation and setup of TortoiseHg



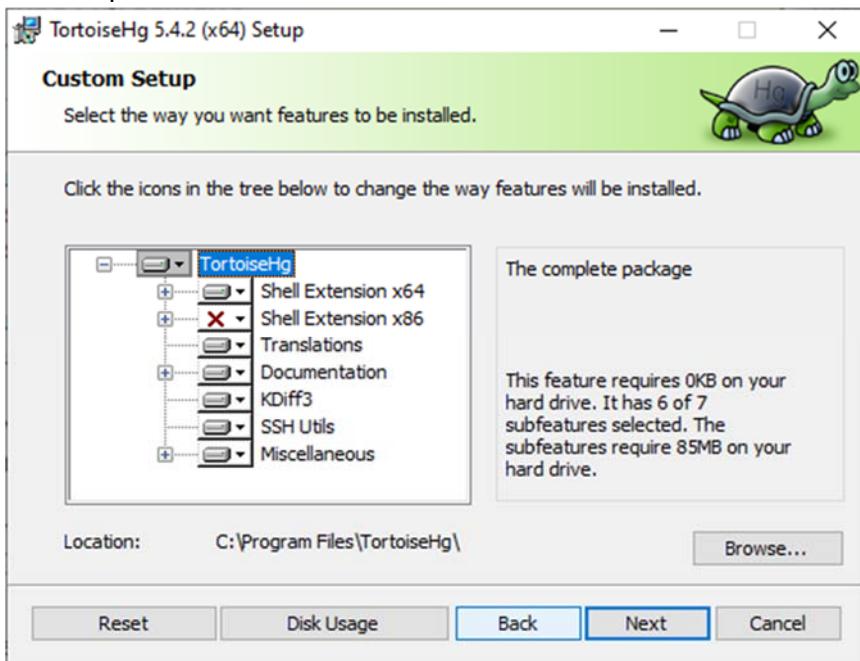
2. Click on Next.



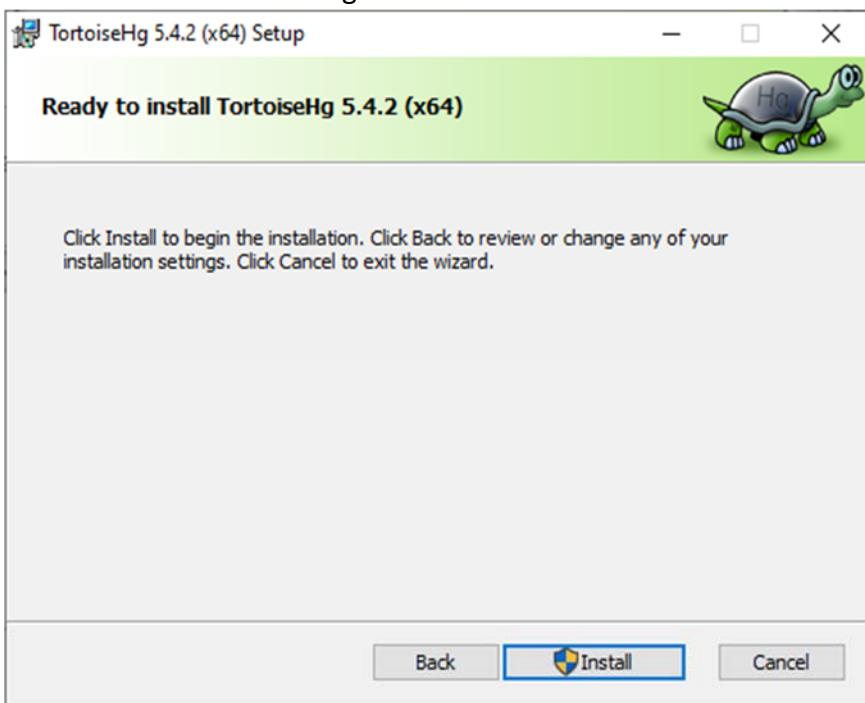
3. Click on Next.



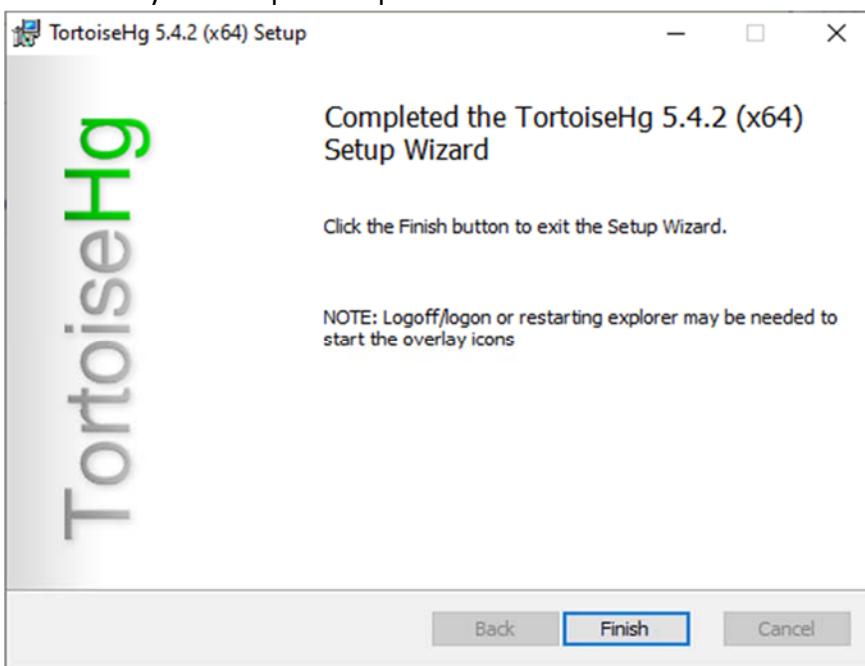
4. Keep it as default and click on Next.



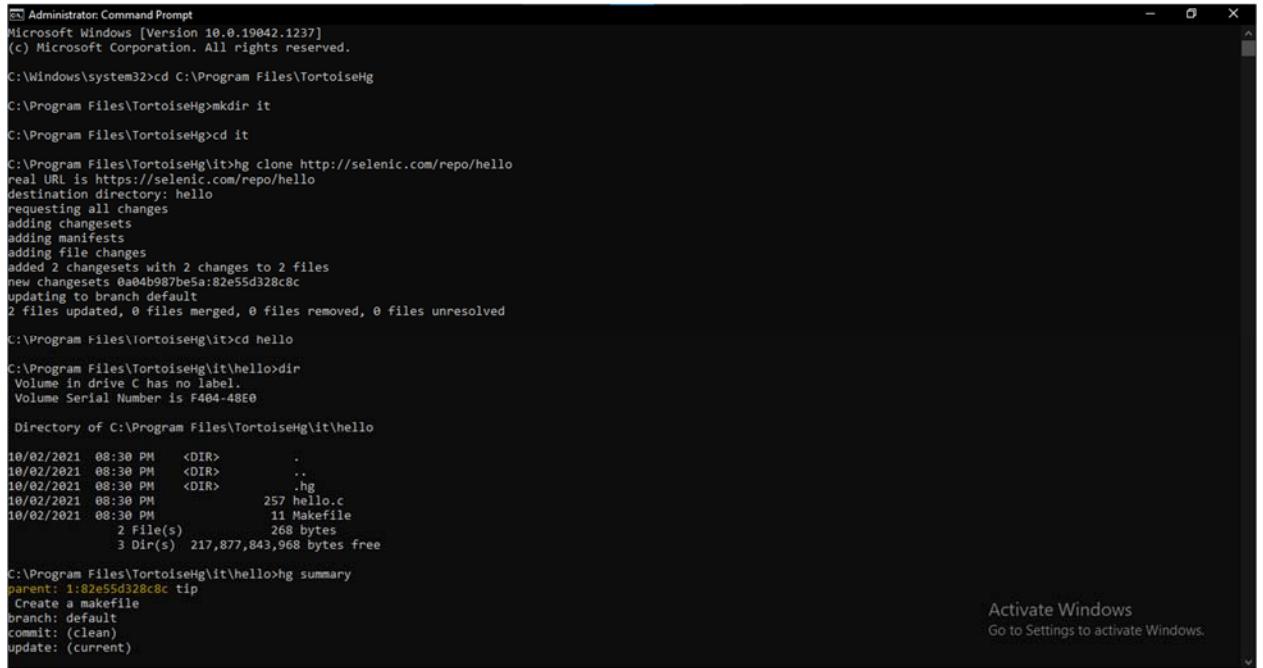
5. Now click install to begin the installation.



6. Hence your setup is completed. Click on finish.



7. Open command prompt go to the directory where TortoiseHg is installed.
8. Create one file it.
9. Go that file directory.
10. Run command, hg clone <http://selenic.com/repo/hello>.
11. Then run command, cd hello.
12. Also run command, hg summary.



```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\TortoiseHg

C:\Program Files\TortoiseHg>mkdir it

C:\Program Files\TortoiseHg>cd it

C:\Program Files\TortoiseHg\it>hg clone http://selenic.com/repo/hello
real URL is https://selenic.com/repo/hello
destination directory: hello
requesting all changes
adding changesets
adding manifests
adding file changes
added 2 changesets with 2 changes to 2 files
new changesets 0@04b987be5a:82e55d328c8c
updating to branch default
2 files updated, 0 files merged, 0 files removed, 0 files unresolved

C:\Program Files\TortoiseHg\it>cd hello

C:\Program Files\TortoiseHg\it\hello>dir
Volume in drive C has no label.
Volume Serial Number is F404-48E0

Directory of C:\Program Files\TortoiseHg\it\hello

10/02/2021  08:30 PM    <DIR>        .
10/02/2021  08:30 PM    <DIR>        ..
10/02/2021  08:30 PM    <DIR>        .hg
10/02/2021  08:30 PM           257 hello.c
10/02/2021  08:30 PM           11 Makefile
               2 File(s)      268 bytes
               3 Dir(s)   217,877,843,968 bytes free

C:\Program Files\TortoiseHg\it\hello>hg summary
parent: 1:82e55d328c8c tip
| Create a makefile
branch: default
commit: (clean)
update: (current)

Activate Windows
Go to Settings to activate Windows.

```

10. Run command hg log.

11. Then come out of the hello directory and run command, hg clone hello clonehello.

```
C:\Program Files\TortoiseHg\it\hello>hg log
changeset: 1:82e55d328c8c
tag: tip
user: mpm@selenic.com
date: Fri Aug 26 01:21:28 2005 -0700
summary: Create a makefile

changeset: 0:0a04b087beaa
user: mpm@selenic.com
date: Fri Aug 26 01:20:50 2005 -0700
summary: Create a standard "hello, world" program

C:\Program Files\TortoiseHg\it\hello>hg clone hello clonehello
abort: repository hello not found!

C:\Program Files\TortoiseHg\it\hello>cd..

C:\Program Files\TortoiseHg\it>hg clone hello clonehello
updating to branch default
2 files updated, 0 files merged, 0 files removed, 0 files unresolved

C:\Program Files\TortoiseHg\it>dir
Volume in drive C has no label.
Volume Serial Number is F404-48E0

Directory of C:\Program Files\TortoiseHg\it

10/02/2021 08:39 PM <DIR> .
10/02/2021 08:39 PM <DIR> ..
10/02/2021 08:39 PM <DIR> clonehello
10/02/2021 08:30 PM <DIR> hello
    0 File(s)     0 bytes
    4 Dir(s) 217,874,989,056 bytes free

C:\Program Files\TortoiseHg\it>cd clonehello

C:\Program Files\TortoiseHg\it\clonehello>hg status

C:\Program Files\TortoiseHg\it\clonehello>hg status
```

Activate Windows
Go to Settings to activate Windows.

12. Go to the directory clonehello

13. Run command hg status, you will get “M hello.c”.

14. Also run command hg diff, which will show the content of “hello.c” file.

```
C:\Command Prompt
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sayali Chowkar>cd..

C:\Users>C:\Program Files\TortoiseHg\b1
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.

C:\Users>cd C:\Program Files\TortoiseHg\b1

C:\Program Files\TortoiseHg\b1>d clonehello

C:\Program Files\TortoiseHg\b1\clonehello>hg status
M hello.c

C:\Program Files\TortoiseHg\b1\clonehello>hg st
M hello.c

C:\Program Files\TortoiseHg\b1\clonehello>edit hello.c
'edit' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\TortoiseHg\b1\clonehello>hg diff
diff -r 82e55d328c8c hello.c
--- a/hello.c  Fri Aug 26 01:21:28 2005 -0700
+++ b/hello.c  Sat Oct 02 19:45:23 2021 +0530
@@ -1,6 +1,6 @@
int main(int argc, char **argv)
{
-   printf("hello, world!\n");
+   printf("hello, world!!!\n");
    return 0;
}

C:\Program Files\TortoiseHg\b1\clonehello>cd..

C:\Program Files\TortoiseHg\b1\clonehello>hg log
abort: no repository found in 'C:\Program Files\TortoiseHg\b1' (.hg not found)
```

Activate Windows
Go to Settings to activate Windows.

15. Run command hg update

```
C:\Program Files\TortoiseHg\it\clonehello>hg update
0 files updated, 0 files merged, 0 files removed, 0 files unresolved

C:\Program Files\TortoiseHg\it\clonehello>hg commit
nothing changed

C:\Program Files\TortoiseHg\it\clonehello>hg push ..\hello
pushing to ..\hello
searching for changes
no changes found

C:\Program Files\TortoiseHg\it\clonehello>
```

16. Go to the clonehello directory and run command hg log.

```
C:\Program Files\TortoiseHg\b1>cd clonehello
C:\Program Files\TortoiseHg\b1\clonehello>hg log
changeset: 2:047e00e1f76e
tag:      tip
user:    Sayali <sayali.6119025.it@mhssce.ac.in>
date:   Fri Jul 30 15:02:46 2021 +0530
summary: hello file has been updated

changeset: 1:82e55d328c8c
user:    mpm@selenic.com
date:   Fri Aug 26 01:21:28 2005 -0700
summary: Create a makefile

changeset: 0:004b987be5a
user:    mpm@selenic.com
date:   Fri Aug 26 01:20:50 2005 -0700
summary: Create a standard "hello, world" program
```

CONCLUSION: Hence we have successfully installed Mercurial software.

NAME: SAYALI BABAN CHOWKEKAR

ROLLNO: 6119025

BRANCH: TE(IT)

SUB: DEVOPS PRACTICAL

EXPERIMENT NO.4

AIM: Installation of GIT

THEORY:

What is git?

- Git is the most commonly used version control system. Git tracks the changes you make to files, so you have a record of what has been done, and you can revert to specific versions should you ever need to. Git also makes collaboration easier, allowing changes by multiple people to all be merged into one source.

So regardless of whether you write code that only you will see, or work as part of a team, Git will be useful for you.

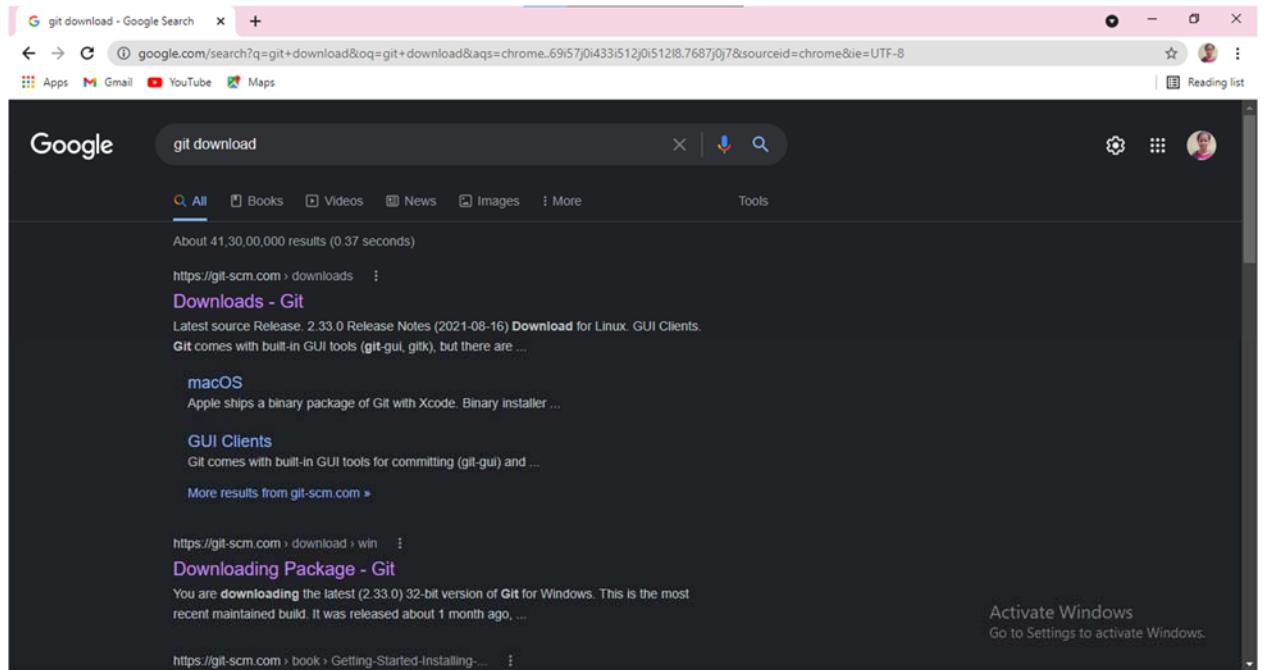


Git is software that runs locally. Your files and their history are stored on your computer. You can also use online hosts (such as GitHub or Bitbucket) to store a copy of the files and their revision history. Having a centrally located place where you can

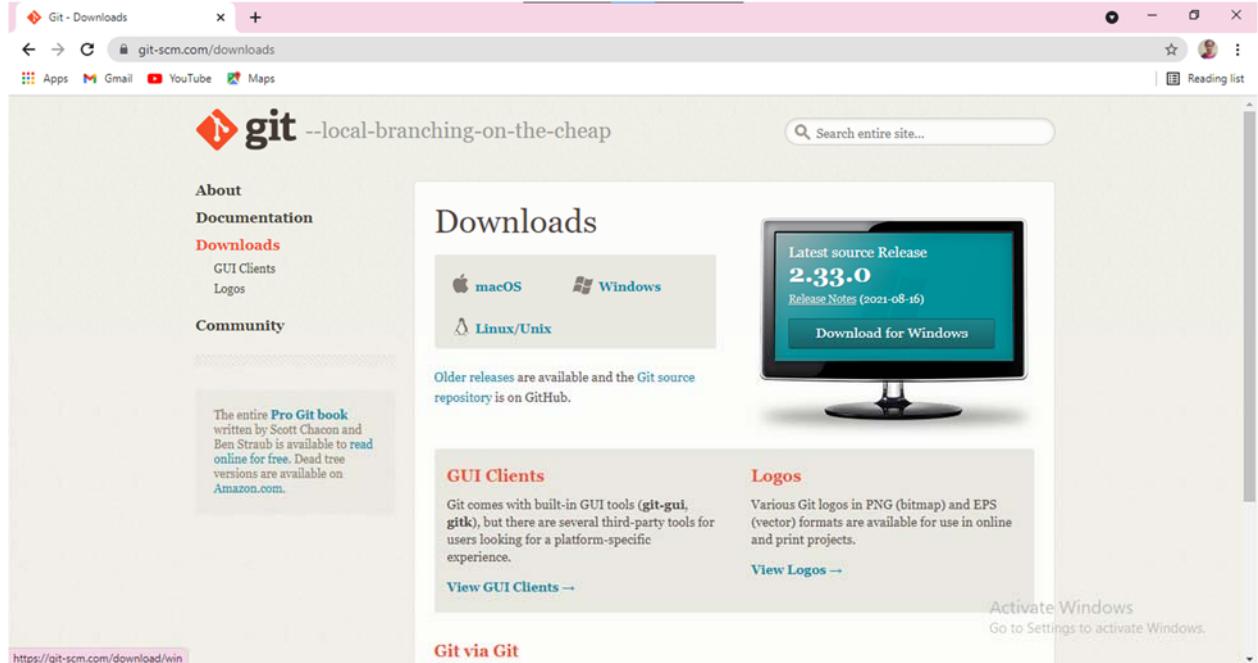
upload your changes and download changes from others, enable you to collaborate more easily with other developers. Git can automatically merge the changes, so two people can even work on different parts of the same file and later merge those changes without losing each other's work!

Steps for installing GIT:

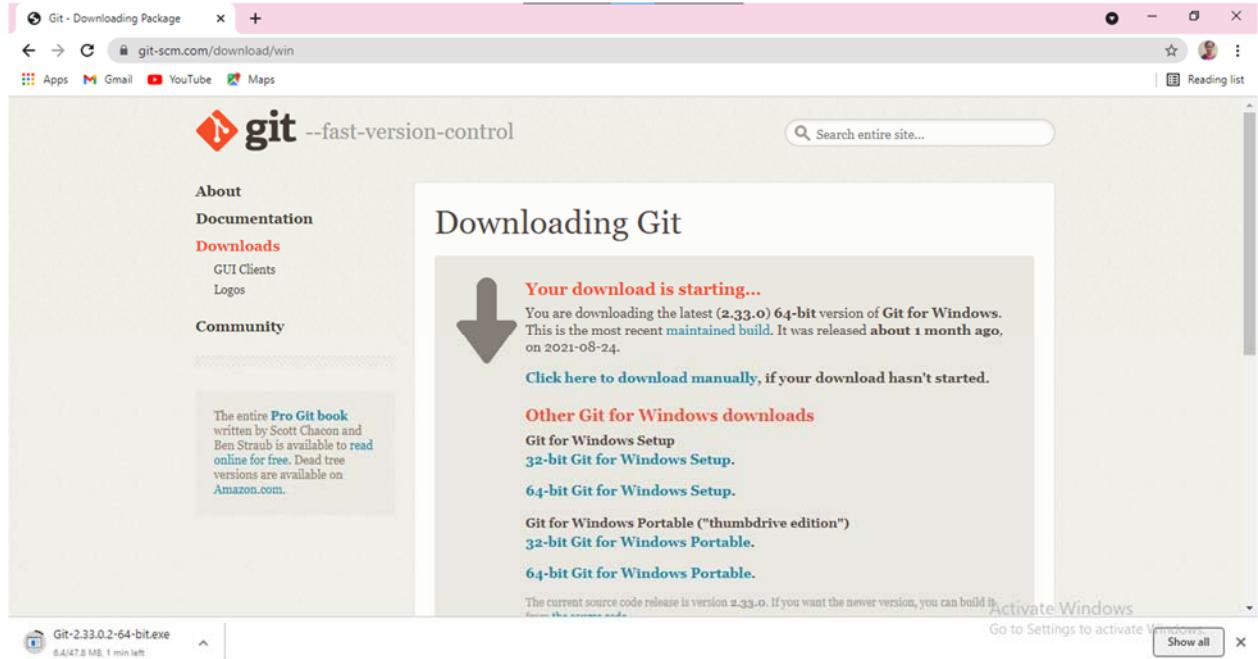
1. First go to browser and type git download. Now click on website <https://git-scm.com>



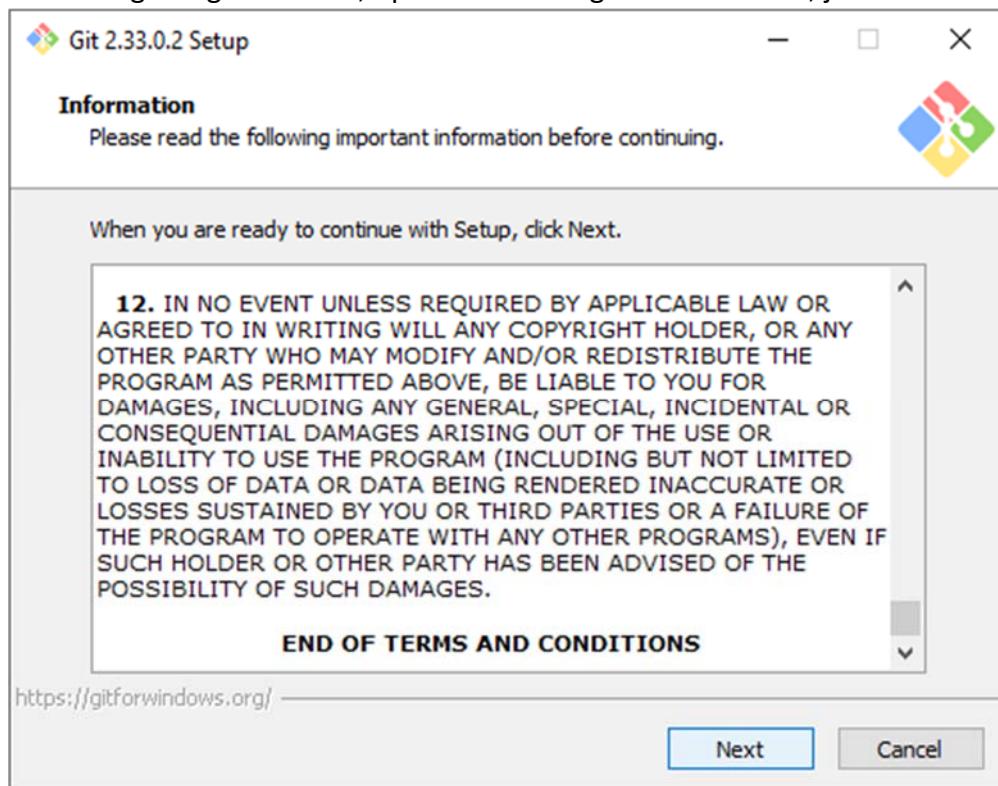
2. Click on which operating system you are using. Eg: Windows.



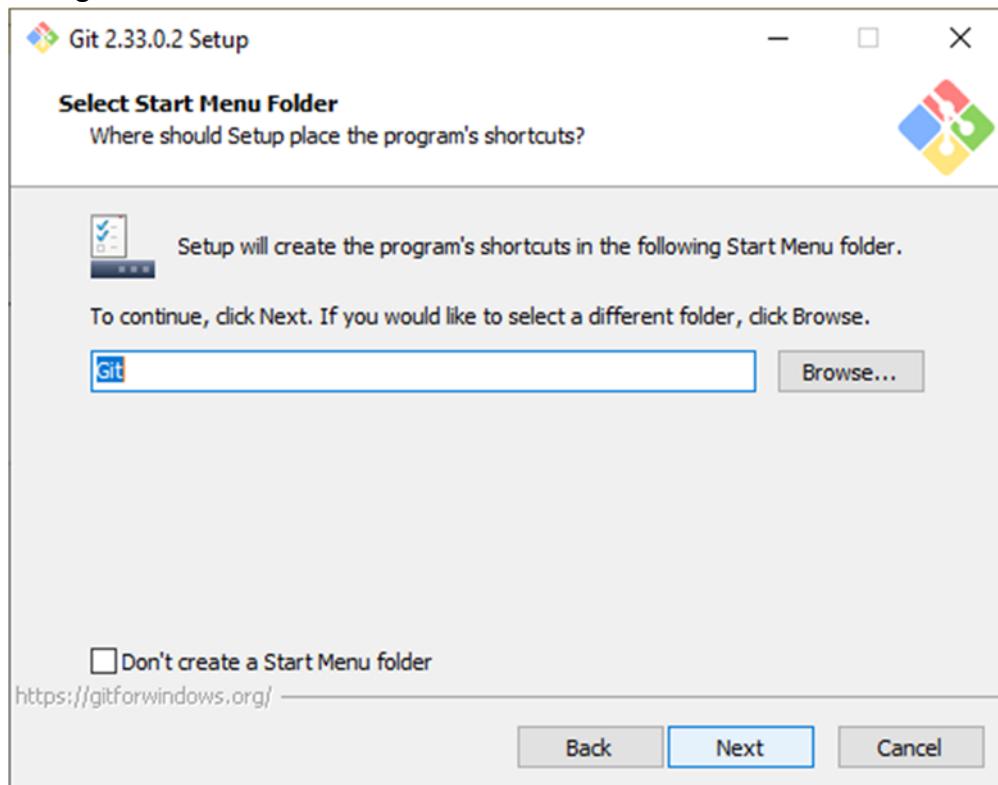
3. Your download will start.



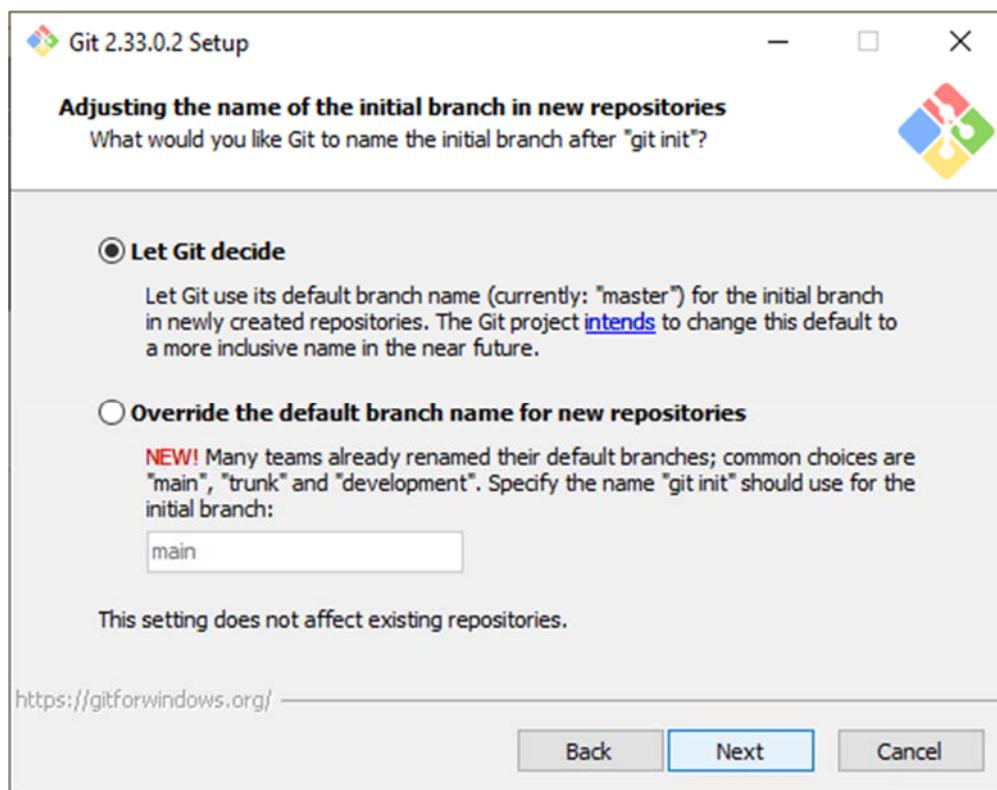
4. After getting download, open it. You will get such window, just click on next.



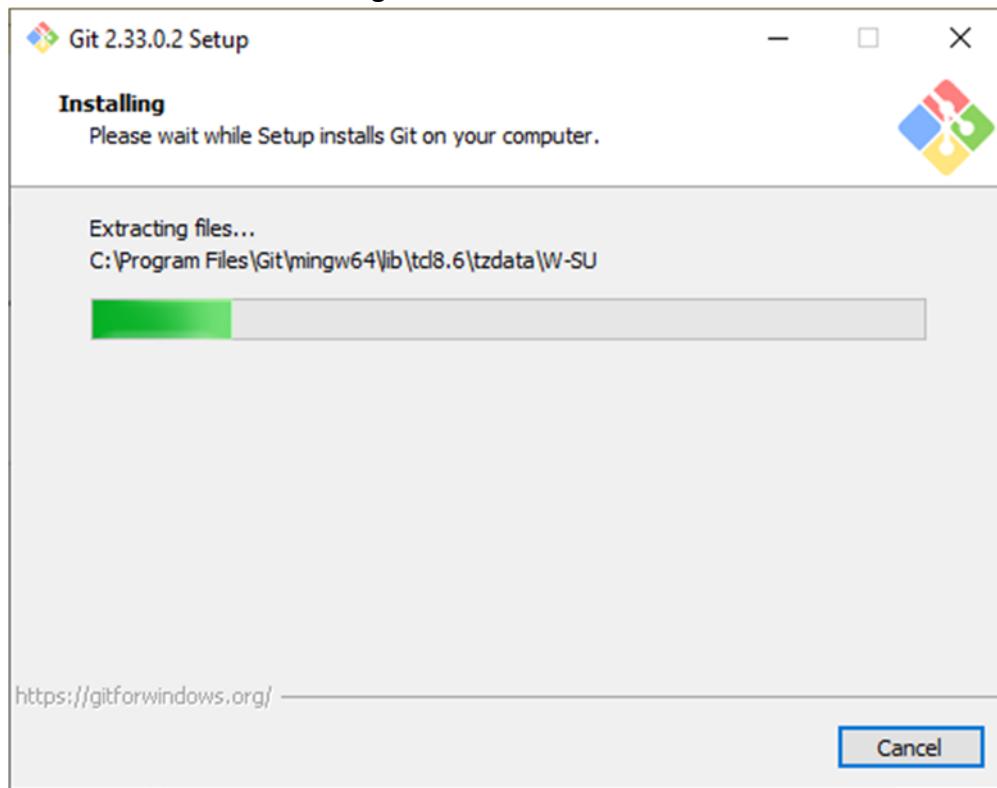
5. Again click on next.



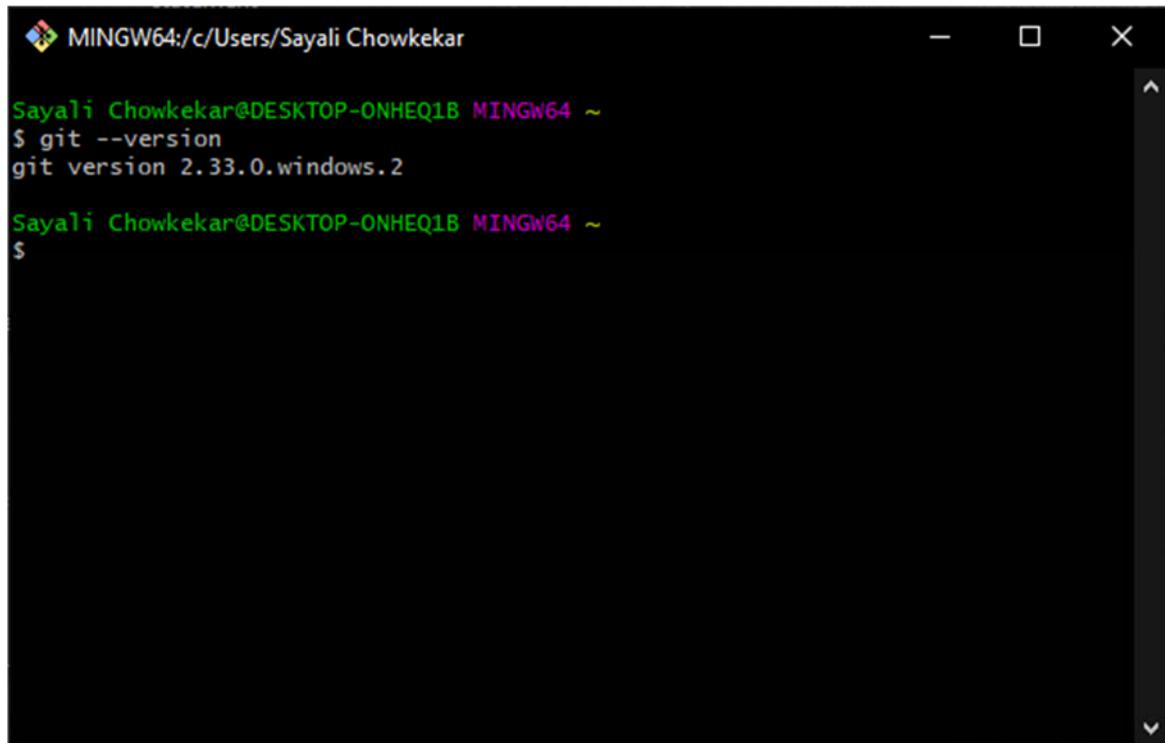
6. Select Let Git decide and click on next.



7. Now it will start installing.



8. After getting installed this window will open. Check version of your Git.



Sayali Chowkekar@DESKTOP-ONHEQ1B MINGW64 ~
\$ git --version
git version 2.33.0.windows.2
Sayali Chowkekar@DESKTOP-ONHEQ1B MINGW64 ~
\$

CONCLUSION: Hence we successfully done installation of Git.

NAME: SAYALI BABAN CHOWKEKAR

ROLLNO: 6119025

BRANCH: TE(IT)

SUB: DEVOPS PRACTICAL

EXPERIMENT NO.5

AIM: Install and configure docker.

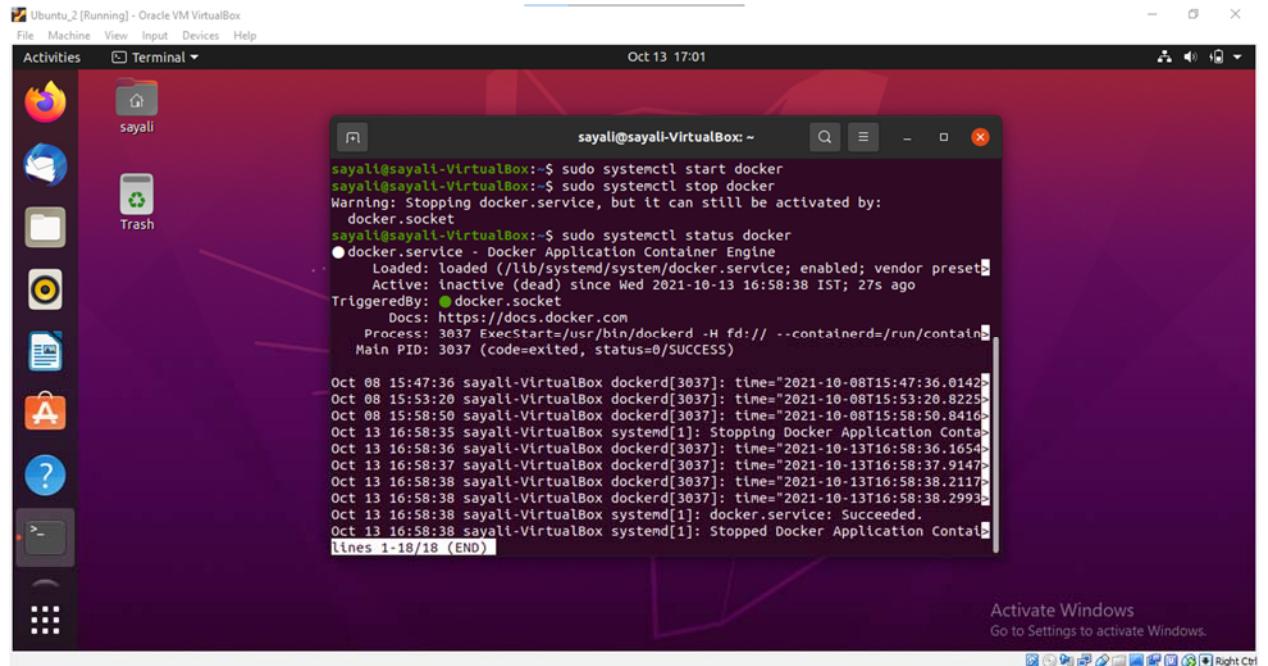
Steps to install docker.



1. Run command “sudo apt install docker.io” to install docker.
2. Run command “docker version” to check the version of docker.

```
sayali@sayali-VirtualBox:~$ sudo apt install docker.io
[sudo] password for sayali:
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (20.10.7-0ubuntu1~20.04.1).
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 315 not upgraded.
sayali@sayali-VirtualBox:~$ docker version
Client:
  Version:          20.10.7
  API version:      1.41
  Go version:       go1.13.8
  Git commit:       20.10.7-0ubuntu1~20.04.1
  Built:            Wed Aug  4 22:52:25 2021
  OS/Arch:          linux/amd64
  Context:           default
  Experimental:     true
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://<2>/var<3>/run<4>/docker.sock/v1.24/version: dial unix /var/run/docker.sock: connect: permission denied
sayali@sayali-VirtualBox:~$
```

3. Run command, “`sudo systemctl start docker`”.
4. Run command, “`sudo systemctl stop docker`”.
5. Run command, “`sudo systemctl status docker`”.



A screenshot of a Linux desktop environment, likely Ubuntu, running in a VirtualBox VM. The desktop has a purple and red gradient background. A dock on the left contains icons for the Dash, Home, Activities, Terminal, and several application icons. A central terminal window titled "sayali@sayali-VirtualBox: ~" shows the following command history:

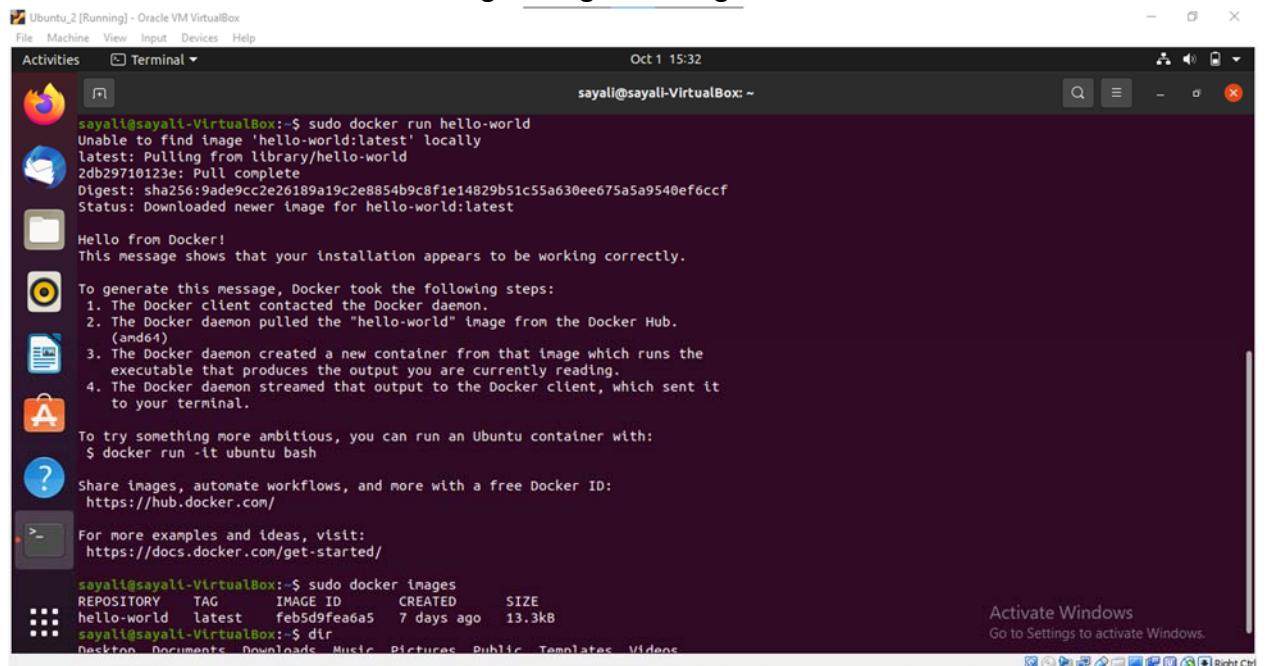
```

sayali@sayali-VirtualBox:~$ sudo systemctl start docker
sayali@sayali-VirtualBox:~$ sudo systemctl stop docker
Warning: Stopping docker.service, but it can still be activated by:
  docker.socket
● docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: inactive (dead) since Wed 2021-10-13 16:58:38 IST; 27s ago
    TriggeredBy: ● docker.socket
      Docs: https://docs.docker.com
      Process: 3037 ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
      Main PID: 3037 (code=exited, status=0/SUCCESS)

Oct 08 15:47:36 sayali-VirtualBox dockerd[3037]: time="2021-10-08T15:47:36.014Z"
Oct 08 15:53:20 sayali-VirtualBox dockerd[3037]: time="2021-10-08T15:53:20.822Z"
Oct 08 15:58:50 sayali-VirtualBox dockerd[3037]: time="2021-10-08T15:58:50.841Z"
Oct 13 16:58:35 sayali-VirtualBox systemd[1]: Stopping Docker Application Container Engine...
Oct 13 16:58:36 sayali-VirtualBox dockerd[3037]: time="2021-10-13T16:58:36.165Z"
Oct 13 16:58:37 sayali-VirtualBox dockerd[3037]: time="2021-10-13T16:58:37.914Z"
Oct 13 16:58:38 sayali-VirtualBox dockerd[3037]: time="2021-10-13T16:58:38.211Z"
Oct 13 16:58:38 sayali-VirtualBox dockerd[3037]: time="2021-10-13T16:58:38.299Z"
Oct 13 16:58:38 sayali-VirtualBox systemd[1]: docker.service: Succeeded.
Oct 13 16:58:38 sayali-VirtualBox systemd[1]: Stopped Docker Application Container Engine.
[the 1-18/18 (END)]

```

6. Run command “`sudo docker run hello-world`” to run image `hello-world`.
7. Run command “`sudo docker images`” to get all images.



A screenshot of a Linux desktop environment, likely Ubuntu, running in a VirtualBox VM. The desktop has a purple and red gradient background. A dock on the left contains icons for the Dash, Home, Activities, Terminal, and several application icons. A central terminal window titled "sayali@sayali-VirtualBox: ~" shows the following command history:

```

sayali@sayali-VirtualBox:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:9ade9cc2e26189a19c2e8854b9c8f1e14829b51c55a630ee675a5a9540ef6ccf
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

```

At the bottom of the terminal window, there is a list of recently used files:

```

sayali@sayali-VirtualBox:~$ sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world latest feb5d9fea6a5 7 days ago 13.3kB
sayali@sayali-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos

```

CONCLUSION: Hence we have successfully installed and configured Docker.

NAME: SAYALI BABAN CHOWKEKAR

ROLLNO: 6119025

BRANCH: TE(IT)

SUB: DEVOPS PRACTICAL

EXPERIMENT NO.6

AIM: Create a container for any two images using docker.

Steps for creating a container for any two images using docker.



1. Run command “`sudo docker run –name myalpine alpine`” to download alpine image.
2. Run command “`sudo docker ps -a`” to see all running or exited docker images.
3. Run command “`sudo docker run –name myubuntu ubuntu`” to download ubuntu image. Then again check if it is installed or not using “`sudo docker ps -a`” command.

```
Ubuntu_2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 1 15:48
sayali@sayali-VirtualBox: ~
sayali@sayali-VirtualBox:~$ sudo docker run --name myalpine alpine
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
a0d0a0d46f8b: Pull complete
Digest: sha256:e1c082e3d3c45ccac829840a25941e679c25d438cc8412c2fa221cf1a824e6a
Status: Downloaded newer image for alpine:latest
sayali@sayali-VirtualBox:~$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d563533b2eee alpine "/bin/sh" About a minute ago Exited (0) About a minute ago myalpine
dba33f3a793b hello-world "/hello" 14 minutes ago Exited (0) 14 minutes ago zen_tu
sayali@sayali-VirtualBox:~$ sudo docker run --name myubuntu ubuntu
Unable to find image 'ubuntu:latest' locally
docker: Error response from daemon: Get https://registry-1.docker.io/v2/: net/http: request canceled while waiting for connection (Client.Timeout exceeded while awaiting headers).
See 'docker run --help'.
sayali@sayali-VirtualBox:~$ sudo docker run --name myubuntu ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
f3ef4ff62e0d: Pull complete
Digest: sha256:65de08a8dabf289ef114053ab32f79e0c333a4fbfa1fe3778bb13ae921a7849b
Status: Downloaded newer image for ubuntu:latest
sayali@sayali-VirtualBox:~$ docker ps -a
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://<2>var<2>/run<2>/docker.sock/v1.24/containers/json?all=1: dial unix /var/run/docker.sock: connect: permission denied
sayali@sayali-VirtualBox:~$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
77657b39c283 ubuntu "bash" 30 seconds ago Exited (0) 25 seconds ago myubuntu
d563533b2eee alpine "/bin/sh" 5 minutes ago Exited (0) 5 minutes ago myalpine
dba33f3a793b hello-world "/hello" 18 minutes ago Exited (0) 18 minutes ago zen_tu
sayali@sayali-VirtualBox:~$
```

CONCLUSION: Hence we have successfully created container for any two images using docker.

NAME: SAYALI BABAN CHOWKEKAR

ROLLNO: 6119025

BRANCH: TE(IT)

SUB: DEVOPS PRACTICAL

EXPERIMENT NO.7

AIM: Build, deploy and manage web application on Docker.

Steps for building, deploying and managing web application on docker.

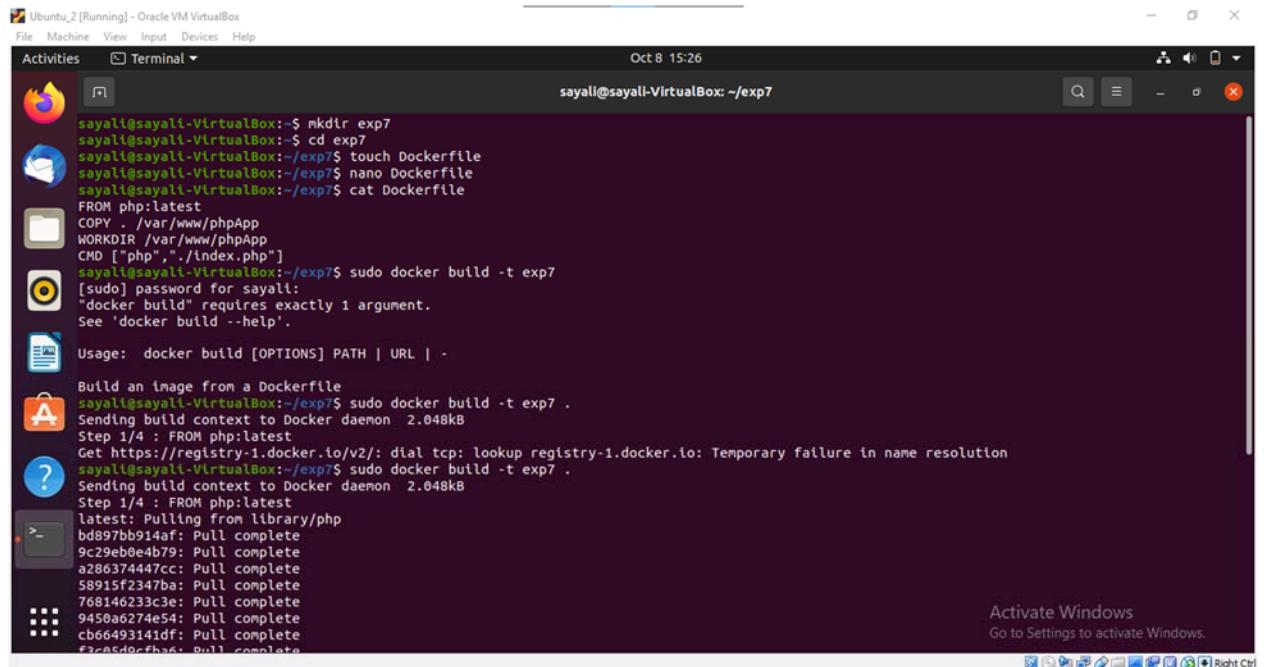


1. Create new folder exp7 using mkdir command.
2. Go to that directory.
3. Run command touch Dockerfile.
4. Edit the Dockerfile using nano editor.
5. Then display content of dockerfile using cat command.
6. Then create file index.php using nano editor.
7. Display it using cat command.

```
sayali@sayali-VirtualBox:~/exp7$ nano index.php
sayali@sayali-VirtualBox:~/exp7$ cat index.php
<?php
echo "Hello, TE(IT) from sayali";
?>
sayali@sayali-VirtualBox:~/exp7$
```

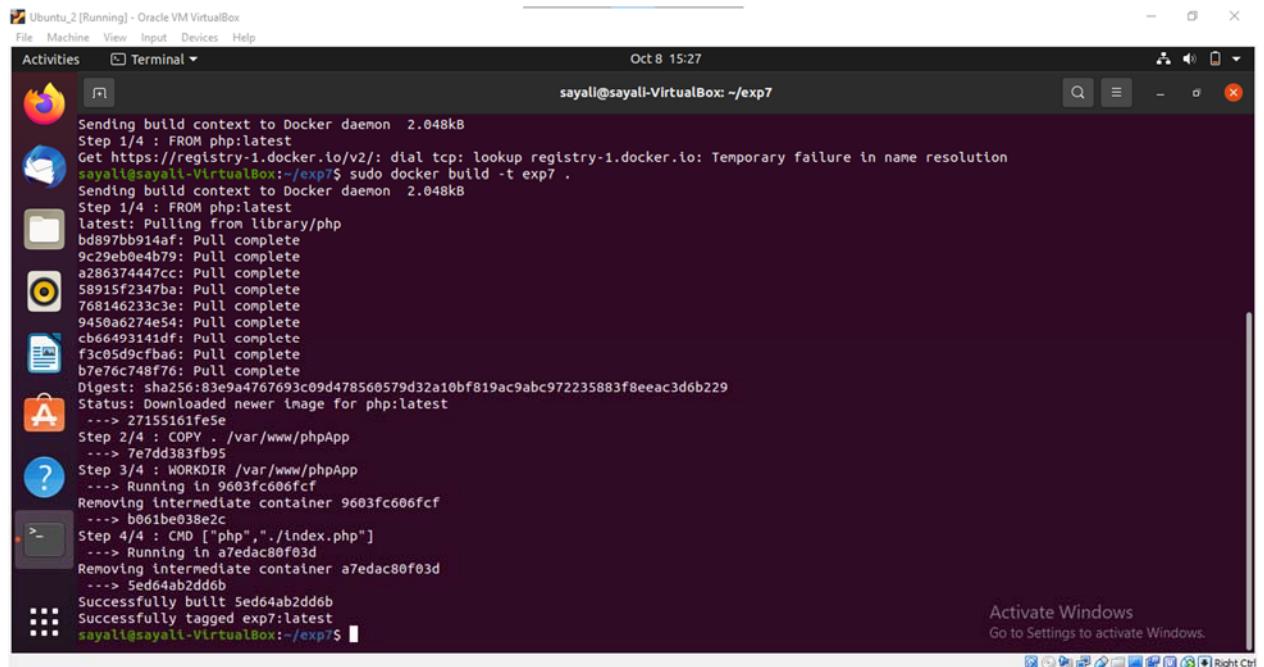
Activate Windows
Go to Settings to activate Windows.

8. Run command “sudo docker build -t exp7 .”



```
sayali@sayali-VirtualBox:~/exp7$ sudo docker build -t exp7 .
[sudo] password for sayali:
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM php:latest
Get https://registry-1.docker.io/v2/: dial tcp: lookup registry-1.docker.io: Temporary failure in name resolution
sayali@sayali-VirtualBox:~/exp7$ sudo docker build -t exp7 .
[sudo] password for sayali:
"docker build" requires exactly 1 argument.
See 'docker build --help'.
```

The terminal shows the user attempting to run the command twice. The first attempt fails because it requires exactly one argument. The second attempt fails because it cannot resolve the Docker registry address.

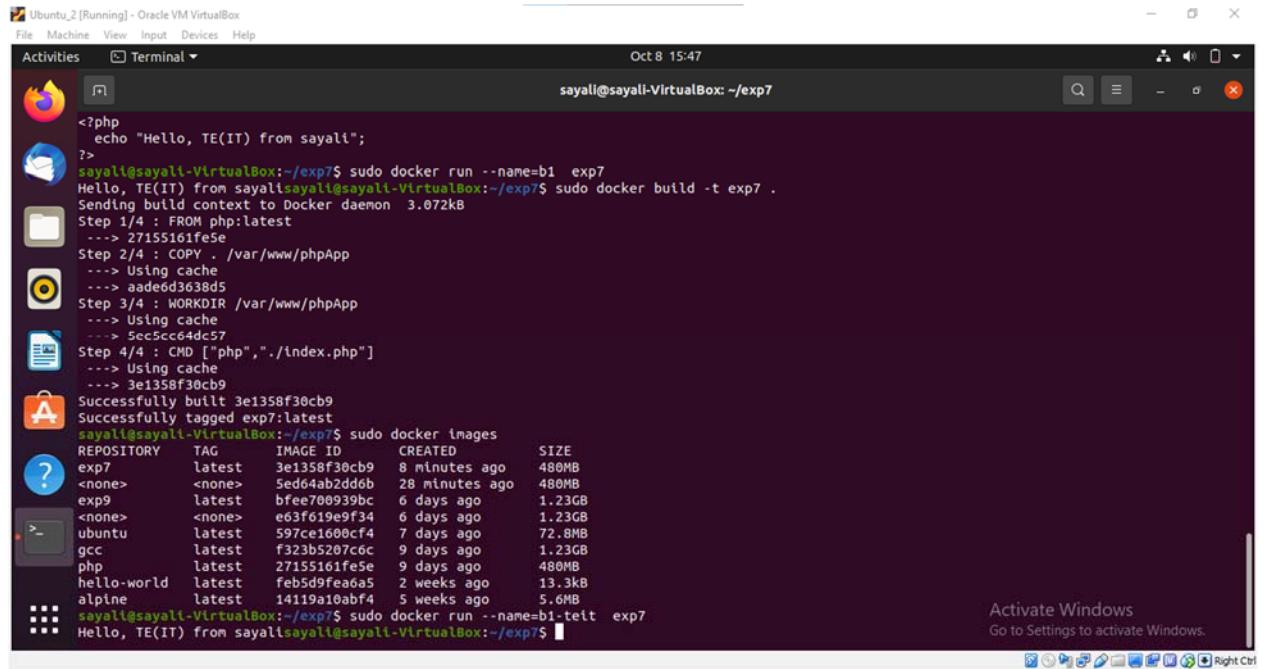


```
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM php:latest
Get https://registry-1.docker.io/v2/: dial tcp: lookup registry-1.docker.io: Temporary failure in name resolution
sayali@sayali-VirtualBox:~/exp7$ sudo docker build -t exp7 .
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM php:latest
latest: Pulling from library/php
bd897bb914af: Pull complete
9c29eb0e4b79: Pull complete
a286374447cc: Pull complete
58915f2347ba: Pull complete
768146233c3e: Pull complete
9450a6274e54: Pull complete
cb66493141df: Pull complete
f3c05d9cfba6: Pull complete
Digest: sha256:83e9a4767693c09d478560579d32a10bf819ac9abc972235883f8eeac3d6b229
Status: Downloaded newer image for php:latest
    ...
Step 2/4 : COPY . /var/www/phpApp
    ...
Step 3/4 : WORKDIR /var/www/phpApp
    ...
Step 4/4 : CMD ["php","./index.php"]
    ...
Successfully built Sed64ab2dd6b
Successfully tagged exp7:latest
sayali@sayali-VirtualBox:~/exp7$
```

The terminal shows the command being run successfully. It prints the Dockerfile context, the Docker image being used (php:latest), and the steps of the build process. Finally, it shows the image was built successfully and tagged as exp7:latest.

9. Now run command “`sudo docker run --name=b1 exp7`”, you will get output as, “Hello, TE(IT) from sayali”.

10. Now run command “`sudo docker images`” to see images created.



The screenshot shows a terminal window titled "Activities Terminal" running on an Ubuntu 20.04 LTS system. The terminal session is as follows:

```
<?php
echo "Hello, TE(IT) from sayali";
?>
sayali@sayali-VirtualBox:~/exp7$ sudo docker run --name=b1 exp7
Hello, TE(IT) from sayali
sayali@sayali-VirtualBox:~/exp7$ sudo docker build -t exp7 .
Sending build context to Docker daemon 3.072kB
Step 1/4 : FROM php:latest
--> z7155161fe5e
Step 2/4 : COPY . /var/www/phpApp
--> Using cache
--> aade6d3d38d5
Step 3/4 : WORKDIR /var/www/phpApp
--> Using cache
--> Sec5cc64dc57
Step 4/4 : CMD ["php", "./index.php"]
--> Using cache
--> 3e1358f30cb9
Successfully built 3e1358f30cb9
Successfully tagged exp7:latest
sayali@sayali-VirtualBox:~/exp7$ sudo docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
exp7           latest   3e1358f30cb9  8 minutes ago  480MB
<none>          <none>  5ed64ab2dd6b  28 minutes ago  480MB
exp9           latest   bfee700939bc  6 days ago   1.23GB
<none>          <none>  e63f619e9f34  6 days ago   1.23GB
ubuntu          latest   597ce1600cf4  7 days ago   72.8MB
gcc             latest   f323b5207c6c  9 days ago   1.23GB
php             latest   27155161fe5e  9 days ago   480MB
hello-world     latest   febd59feada5  2 weeks ago  13.3kB
alpine          latest   14119a10abf4  5 weeks ago  5.6MB
sayali@sayali-VirtualBox:~/exp7$ sudo docker run --name=b1-teit exp7
Hello, TE(IT) from sayali
sayali@sayali-VirtualBox:~/exp7$
```

The terminal also displays a "Activate Windows" watermark at the bottom right.

CONCLUSION: Hence we have successfully Build, deploy and manage web application on Docker.

NAME: SAYALI BABAN CHOWKEKAR

ROLLNO: 6119025

BRANCH: TE(IT)

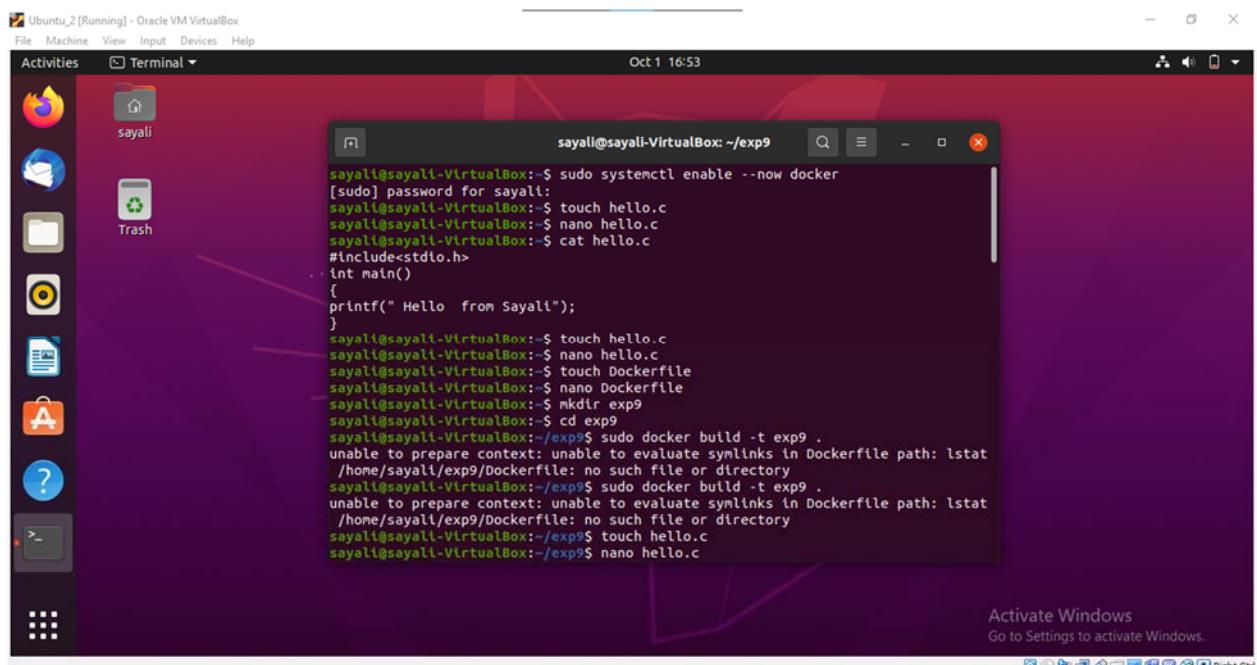
SUB: DEVOPS PRACTICAL

EXPERIMENT NO.8

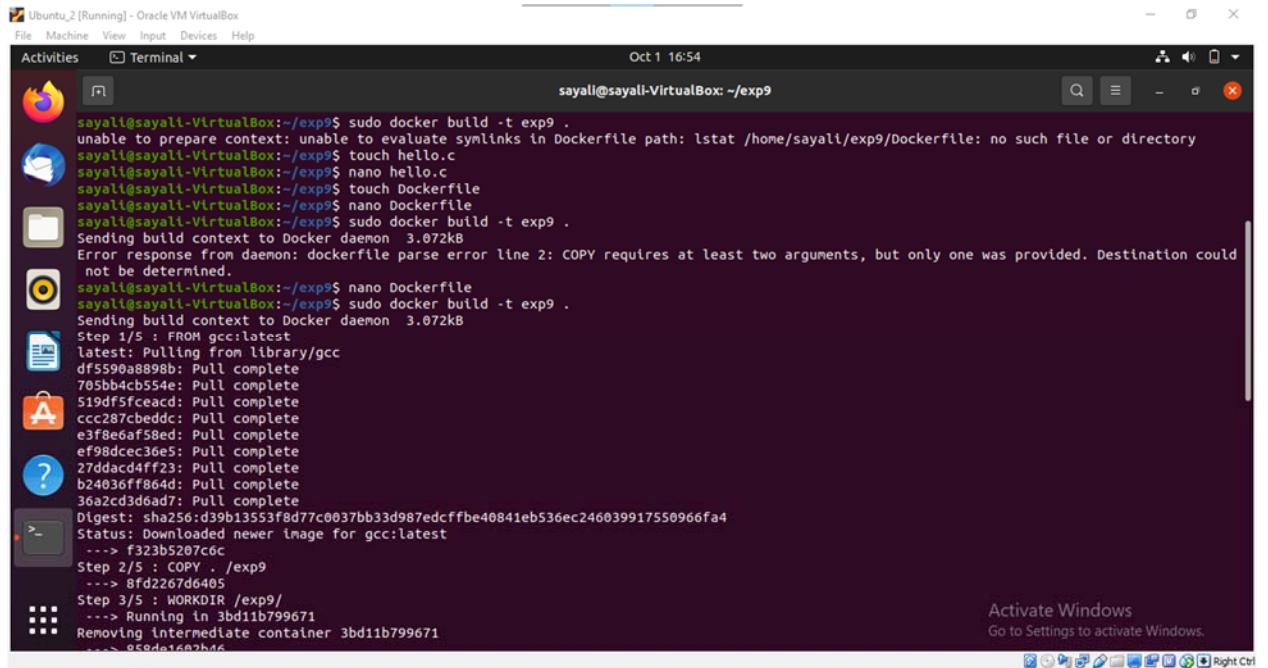
AIM: Build, deploy and manage non web application on Docker.

Steps for building, deploying and managing non web application on docker:

1. Run command sudo systemctl enable –now docker.
2. Then create file hello.c using touch command.
3. Edit it using nano editor.
4. Display it using cat command.
5. Now create Dockerfile using command “touch Dockerfile”
6. Edit it using nano editor.
7. Create new directory exp9 using mkdir command and go to that directory using “cd exp9”.

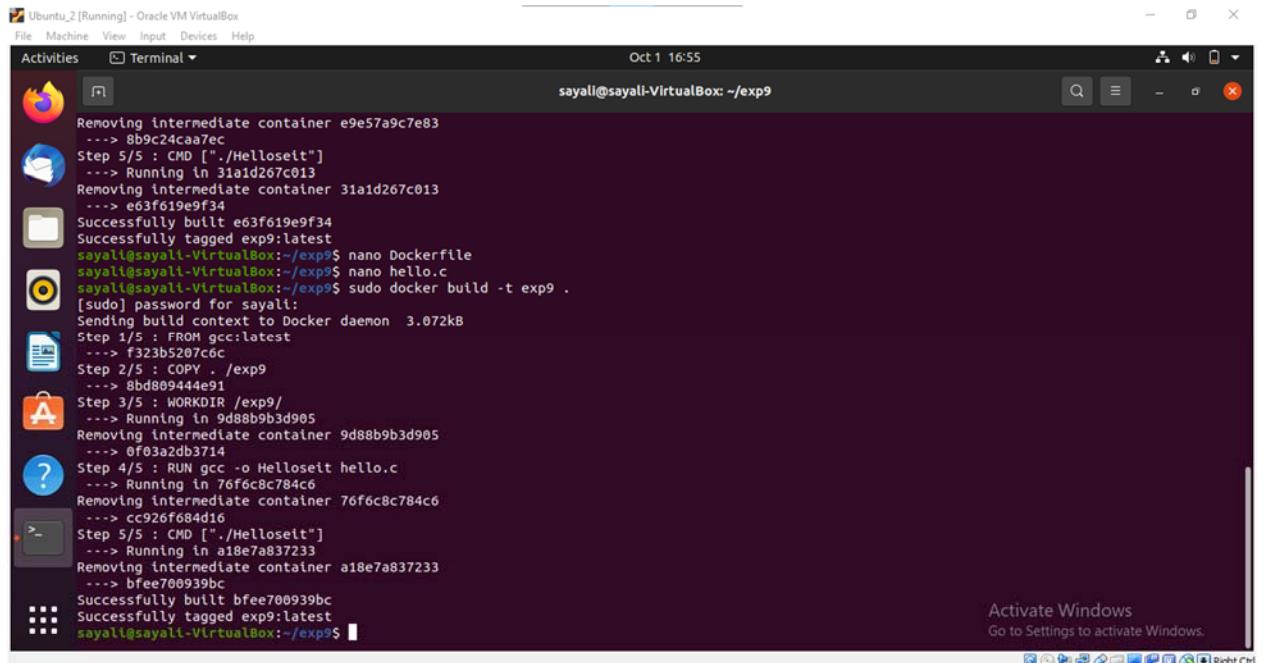


8. Now run command “`sudo docker build -t exp9 .`”



```
sayali@sayali-VirtualBox:~/exp9$ sudo docker build -t exp9 .
unable to prepare context: unable to evaluate symlinks in Dockerfile path: lstat /home/sayali/exp9/Dockerfile: no such file or directory
sayali@sayali-VirtualBox:~/exp9$ touch hello.c
sayali@sayali-VirtualBox:~/exp9$ nano hello.c
sayali@sayali-VirtualBox:~/exp9$ touch Dockerfile
sayali@sayali-VirtualBox:~/exp9$ nano Dockerfile
sayali@sayali-VirtualBox:~/exp9$ sudo docker build -t exp9 .
Sending build context to Docker daemon 3.072kB
Error response from daemon: dockerfile parse error line 2: COPY requires at least two arguments, but only one was provided. Destination could not be determined.
sayali@sayali-VirtualBox:~/exp9$ nano Dockerfile
sayali@sayali-VirtualBox:~/exp9$ sudo docker build -t exp9 .
Sending build context to Docker daemon 3.072kB
Step 1/5 : FROM gcc:latest
latest: Pulling from library/gcc
df5590a8898b: Pull complete
705bb4cb554c: Pull complete
519df5fcceacd: Pull complete
ccc287cbeddc: Pull complete
e3f8e6af58ed: Pull complete
ef98dcec30e5: Pull complete
27ddacd4ff23: Pull complete
b24036ff864d: Pull complete
36a2cd3d6ad7: Pull complete
Digest: sha256:d39b13553f8d77c0037bb33d987edcffbe40841eb536ec246039917550966fa4
Status: Downloaded newer image for gcc:latest
--> f323b5207c6c
Step 2/5 : COPY . /exp9/
--> 8fd2257dd405
Step 3/5 : WORKDIR /exp9/
--> Running in 3bd11b799671
Removing intermediate container 3bd11b799671
--> e59de1602b46
```

9. You will see successfully build displayed on the terminal.



```
sayali@sayali-VirtualBox:~/exp9$ sudo docker build -t exp9 .
Removing intermediate container e9e57a9c7e83
--> 8b9c24caa7ec
Step 1/5 : FROM gcc:latest
--> f323b5207c6c
Step 2/5 : COPY . /exp9/
--> 8bd009444e91
Step 3/5 : WORKDIR /exp9/
--> Running in 9d88b9b3d905
Removing intermediate container 9d88b9b3d905
--> 0f03a2db3714
Step 4/5 : RUN gcc -o Helloseit hello.c
--> Running in 76f6c8c784c6
Removing intermediate container 76f6c8c784c6
--> cc926f684d16
Step 5/5 : CMD ["./Helloseit"]
--> Running in a18e7a837233
Removing intermediate container a18e7a837233
--> bfee700939bc
Successfully built bfee700939bc
Successfully tagged exp9:latest
sayali@sayali-VirtualBox:~/exp9$
```

10. Now run docker using run command.

```
sayali@sayali-VirtualBox:~/exp9$ sudo docker run --name=info exp7
Hello world, Thank yousayali@sayali-VirtualBox:~/exp9$ S
```

CONCLUSION: Hence we have successfully build, deployed and managed non web application on Docker.