Experiment 3:

Git is a DevOps tool for source code management—an open-source version control system (VCS) used to handle small to very large projects efficiently.

Git is used to tracking changes in the source code, supporting non-linear development so that multiple developers can work together.

Git Installation on Windows

Let us now look at the various steps in the install git:

Step 1:

Download the latest version of Git and choose the 64/32 bit version. After the file is downloaded, install it in the system. Once installed, select Launch the Git Bash, then click on finish. The Git Bash is now launched.

Step 2:

Check the Git version:

```
$ git --version
```

Step 3:

For any help, use the following command:

```
$ git help config
```

This command will lead you to a browser of config commands. Basically, the help the command provides a manual from the help page for the command just following it (here, it's config).

Another way to use the same command is as follows:

```
$ git config --help
```

Step 4:

Create a local directory using the following command:

```
$ mkdir test

$ cd test
```

Step 5:

The next step is to initialize the directory:

```
$ git init
```

Step 6:

Go to the folder where "test" is created and create a text document named "demo." Open "demo" and put any content, like "Hello Simplilearn." Save and close the file.

Step 7:

Enter the Git bash interface and type in the following command to check the status:

```
$ git status
```

Step 8:

Add the "demo" to the current directory using the following command:

```
$ git add demo.txt
```

Step 9:

Next, make a commit using the following command:

```
$ git commit -m "committing a text file"
```

Step 10:

Link the Git to a Github Account:

```
$ git config --global user.username
```

Note: simplilearn-github is the username on the Github account.

Step 11:

Open your Github account and create a new repository with the name "test_demo" and click on "Create repository." This is the remote repository. Next, copy the link of "test_demo."

Step 12:

Go back to Git bash and link the remote and local repository using the following command:

```
$ git remote add origin <link>
```

Here, <link> is the link copied in the previous step.

Step 13:

Push the local file onto the remote repository using the following command:

```
$ git push origin master
```

Step 14:

Move back to Github and click on "test_demo" and check if the local file "demo.txt" is pushed to this repository.

Ubuntu

Verify the version of Git currently installed on the server:

```
git --version
```

Copy

If Git is installed, you’ll receive output similar to the following:

```
Output
git version 2.25.1
```

Before you begin, you need to install the software that Git depends on. This is all available in the default repositories, so we can update our local package index and then install the relevant packages.

```
sudo apt update
sudo apt install libz-dev libssl-dev libcurl4-gnutls-dev libexpat1-dev gettext cmake gcc
```

Copy

After you have installed the necessary dependencies, create a temporary directory and move into it. This is where we will download our Git tarball.

```
mkdir tmp
cd /tmp
```

Copy

From the Git project website, we can navigate to the tarball list available at https://mirrors.edge.kernel.org/pub/software/scm/git/ and download the version you would like. At the time of writing, the most recent version is 2.26.2, so we will download that for demonstration purposes. We’ll use curl and output the file we download to git.tar.gz.

```
curl -o git.tar.gz https://mirrors.edge.kernel.org/pub/software/scm/git/git-2.26.2.tar.gz
```

Copy

Unpack the compressed tarball file:

```
tar -zxf git.tar.gz
```

Copy

Next, move into the new Git directory:

```
cd git-*
```

Copy

Now, you can make the package and install it by typing these two commands:

```
make prefix=/usr/local all
sudo make prefix=/usr/local install
```

Copy

Now, replace the shell process so that the version of Git we just installed will be used:

```
exec bash
```

Copy

With this complete, you can be sure that your install was successful by checking the version.

```
git --version
```

Copy

```
Output
git version 2.26.2
```

With Git successfully installed, you can now complete your setup.

Update Git

Before setting up Git, you should first make sure you are using the latest stable version as they can quickly get out of date.

To update Git you first, update your package lists:

```
sudo apt update
```

Update Git:

```
sudo apt install git
```

Verify you have the latest stable Git Version

```
git --version
```

Setting Up Git

After you are satisfied with your Git version, you should configure Git so that the generated commit messages you make will contain your correct information and support you as you build your software project.

Configuration can be achieved by using the git config command. Specifically, we need to provide our name and email address because Git embeds this information into each commit we do. We can go ahead and add this information by typing:

```
git config --global user.name "Your Name"
git config --global user.email "youremail@domain.com"
```

Copy

We can display all of the configuration items that have been set by typing:

```
git config --list
```

Copy

```
Output
user.name=Your Name user.email=youremail@domain.com ...
```

The information you enter is stored in your Git configuration file, which you can optionally edit by hand with a text editor of your choice like this (we'll use nano):

```
nano ~/.gitconfig
```

Copy

```
~/.gitconfig contents

[user] name = Your Name email = youremail@domain.com
```

Press CTRL and X, then Y then ENTER to exit the text editor.