DevOps

Git -> Git track changes via a distributed version control system. This means that Git can track the state of different versions of your projects while you're developing them. It is distributed because you can access your code files from another computer - & so can other developers.

When one building an open source project, you'll need a way to document or track your code. This helps make your work organized, & lets you keep track of the changes one made.

Operation of Git

- Manage projects with Repositories
- clone a project to work on a local copy
- control & track changes with staging & committing
- Branch & Merge to allow for work on different parts & versions of a project
- Pull the latest version of the project to a local copy
- Push local updates to the main project

Experiment No. 1 Date Page No 1

Name of Experiment

A1. Demonstrate & Create project in local & remote repository using GitBash & GitHub & apply init, status, log, add, commit, push, config, clone & reset commands on repository.

→ i) Git Init Command → Used to create a local repository
Syntax → $ git init Demo

ii) Git config command → Configures the user.
Syntax
- $ git remote add origin https://github.com/chandana-ISE/lab.git
- $ git config user.email "chandanaraveen343@gmail.com"
- cd desktop
- cd Devops.

iii) Git add command → used to add one or more files to staging area.
Syntax → $ git add

iv) Git commit command → changes the head. It records the file permanently in the version history with a message
Syntax → $ git commit -m "message"

v) Git push command → used to upload local repository content to a remote repository
Syntax → $ git push origin master

vi) Git status command -> used to display the state of working directory & the staging area.
Syntax -> $ git status

vii) Git log command -> Used to check the commit history
Syntax -> $ git log

viii) Git clone command -> Used to make a copy of a repository from an existing URL.
Syntax -> $ git clone URL

ix) Git Reset command -> Used to reset the changes.
Syntax -> $ git reset --hard

Experiment No. 2 Date Page No 3

Name of Experiment

A2. Demonstrate to create a project in remote repository & apply fork, merge, diff, merge conflict, branch & pull request concepts on repository using GitHub.

-> i) Git Fork -> It is rough copy of a repository. It allows one to freely test & debug with changes without affecting the orginal project

Steps for forking the repository

- Login to the Github account
- Find the GitHub repository which you want to fork.
- Click the Fork button on the upper right side of the repository's page

ii) Git Merge -> The merging is a procedure to connect the forked history.

iii) Git Merge conflict -> When two branches are toying to merge & both are edited at the same time & in the same, Git wont be able to find or identify version is to take for changes.

```
$ git checkout -b branch1
$ nano b1.txt
$ git add.
$ git commit -m "branch1"
$ git switch master
```

$ git checkout -b branch2
$ nano b1.txt
$ git add .
$ git commit -m "branch2"
$ git switch master
$ git merge branch1 -> Merge command.
$ git merge branch -> Merge conflict

iv) Git Diff -> It compares the different version of data sourc.
Syntax -> $ git diff branch1

v) Git Branch -> It is a version that diverges from the main working project.
- Create Branch -> $ git checkout -b branch
- To delet Branch -> $ git switch master
  -> $ git -d branch

vi) Git Pull -> Used to receive data from github.
Syntax -> $ git push origin b1
-> $ git push origin b2

Experiment No. 3 Date Page No 5

Name of Experiment

A3 Demonstrate the process of integration github repository with jenkins to automate the project execution in CI/CD pipeline

→ 1) Install Jenkins on Windows.

i) Install Java Development Kit (JDK)

ii) Download JDK 17 & choose windows 32-bit or 64-bit according to your system configuration. Click on "accept the license agreement." Set the path for the Environmental Variable for JDK.

- Go to System Properties. Under the "Advanced" tab, select "Environmental Variables".
- Under system variables, select "new". Then copy the path of JDK folder & paste it in the corresponding value field.
- Under system variable, set up a bin folder for JDK in PATH variable.
- Go to command prompt & type the java version command to check if java has been successfully installed.

2) Download & Install Jenkins

3) Download Jenkins, Under LTS, click on window

- After the file is downloaded, unzip it. Click on the folder & install it. Select "finish" once done.

Experiment No. ..... Date ..... Page No 6.

Name of Experiment .....

4) Run Jenkins on localhost 8080

- Open the web browser & type "localhost:8080"
- Enter the credentials & log in. If you install Jenkins for the first time, the dashboard will ask you to install the recommended plugins, install the recommended plugins.

5) Jenkins Server Interface.

- New Item allows to create new project.
- Built History shows the status of builds.
- Manage System deals with the various configuration of the system.

1. Create a folder on the desktop that a python file.
2. Push the project into git repository
3. Launch the jenkins & configure the pipline.
4. Configuration of Jenkins
   - Create new project & given a name to it. Choose "Freestyle Project".
   - Click on newly created project & move to configuration. Under source code Management provide git repository
   - Under Build Triggers select Poll SCM & configure the schedule according to your project.
   - Apply & save your configuration
   - Click on "Build now" for detailed output click on "Console output" & verify the process. Finally build status will be displayed according to your project implementation.

Experiment No. 4 Date Page No 7

Name of Experiment

B- Exercise

B1 Create a docker image for an applications stored in local repository & run the application using docker image.

→ Docker is an open platform for developing, shipping and running applications. It enables to separate applications from infrastructure so you can deliver software quickly. It user client-server architecture communicate using REST, API over UNIX sockets or a network interface.

1) Install Docker Desktop on windows
2) Install prerequisties for java applications like jdk tool.
3) configure environment variable for jdk
4) Create java application
5) Create Dockerfile where configuration details are provided for creating image.
6) Complile the java program for any errors.
7) Create image for the application.
8) Run the application using docker image.