Computer Science & Information Systems

**DevOps for Cloud - Lab Sheet1 – Module 3**

**(M3: Source Code Management)**

This lab sheet needs to be administered along with Module 3: Source Code Management

**Notations used in the document**

* ‘>’ represents the terminal, where we type the commands.
* The text mentioned within ‘[‘ and ‘]’ brackets provides additional documentation for the step.

1. Objective:

This lab sheet offers a fundamental introduction for using git and GitHub for students. Students will learn how to install git, set up a git repository, execute basic git commands, and connect their repository with GitHub.

1. Pre-requisite:

* Basic knowledge of Version Control Systems (VCS), especially Distributed VCS.

1. Lab Exercise:

**Task 1: Install Git in Local System**

* URL - <https://git-scm.com/downloads>
* Download git based on your OS type (Windows, Mac or Linux)

**Task 2: Start the Terminal**

* Open Git Bash in Windows
* [Alternatives - Command Prompt (Windows), Powershell (Windows), Terminal (MacOS), GNOME (Linux) etc.]

**Task 3: Configurations**

1. Setting User Name

> git config --global user.name "Your Name"

[Ex: git config --global user.name “Shreyas Rao”]

1. Setting Email

git config --global user.email [yourname@example.com](mailto:yourname@example.com)

[Ex: git config --global user.email shreyassureshrao@gmail.com]

1. Setting default editor [optional]

> git config --global core.editor "code --wait"

[Sets VS Code as the default editor for Git]

1. Set default Branch

> git config --global init.defaultbranch main

1. Check the git version [Optional]

> git --version

**Task 4: Execute basic GIT commands**

1. init command [Initializing a repository – Tell git to track changes in the directory]

* Create an empty folder called “gitdemo”
* Open the git bash terminal and navigate to the folder “gitdemo”. This is the folder where you want to track the changes to source code.
* > git init [Type “git init” in the git bash terminal]
* [git will create a hidden folder called “.git”]
* [In this folder, create a “index.html” file with the following contents]

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>My First Git HTML File</title>

</head>

<body>

    <h1>Hello, Git!</h1>

    <p>This is a simple HTML file to demonstrate basic Git commands, which includes init,status,log,add and commit.</p>

</body>

</html>

1. status command [Check the status of the git repository]

* > git status
* [Indicates untracked file “index.html”]

1. add command [Add the file(s) to git repository to track]

* > git add . [git add <space> dot]
* [Re-check the status using the command “git status” – shows as “staged”]

1. commit command [Commit the changes]

* > git commit -m “Commit Message”
* [Ex: git commit -m “Index.html committed”; recheck status]
* [Alternative command -> git commit -a -m “Commit message” to perform both add and commit in a single step]

1. log command [View commit history of git repository]

* > git log
* [Displays the author info and date of commit, provided in configuration]
* [Exercise -> Make some changes to the “index.html” file and go through b,c,d,e steps again]

1. view branch

* > git branch
* [Shows the main branch]

1. branch command [Create a feature branch for new feature or bug fix]

* > git branch <branch-name>
* [Ex: git branch about-feature]
* [Create a new html file called “about.html”]

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>About html file</title>

</head>

<body>

    <p>Git is a distributed version control system (VCS) designed to track changes in source code during software development. It was created by Linus Torvalds in 2005 and has since then has become one of the most widely used version control systems in the world.</p>

</body>

</html>

* [Add anchor tag in “index.html” to link to “about.html” file]

<body>

    <h1>Hello, Git!</h1>

    <p>This is a simple HTML file to demonstrate basic Git commands, which includes init,status,log,add and commit.</p>

    <a href="about.html">About Git</a>

</body>

1. switch command [Switch to a new branch]

* > git switch <branch-name>
* [Ex: git switch about-feature]
* [In earlier versions, “git checkout” fulfils the same purpose. Command is “git checkout <branch-name>”]

1. merge command [Integrate changes in feature branches with main]

* > git switch main [now in main branch]
* > git merge about-feature [changes made in “about-feature” branch will be merged with main]

**Task 5: Integrate with GitHub**

1. Create a GitHub Login [URL - <https://github.com>]
2. Create a repository in GitHub. [Ex: “gitdemo”]

[Follow the steps in the link - https://docs.github.com/en/repositories/creating-and-managing-repositories/quickstart-for-repositories]

1. In Git Bash, run the command

* > git remote add origin [https://github.com/<username of account owner>/gitdemo.git](https://github.com/%3cusername%20of%20account%20owner%3e/gitdemo.git)
* [Ex: git remote add origin <https://github.com/shreyassureshrao/gitdemo.git>]

1. Push the contents from local repo to remote repo

* > git push -u origin main
* [Will push the local repo to remote location]

1. Outputs/Results:

Students are expected to perform tasks provided in the lab capsule, and thereby learn the git installation, executing basic commands using git, and integration between git and GitHub.

1. Observations:

This lab capsule assignment is aligned towards git, which is a distributed version control system. We do not cover any centralized version control systems in this lab capsule.

References:

* <https://git-scm.com/>
* <https://github.com/>
* “Pro Git”, by Scott Chacon and Ben Straub, Apress publication, 2023