## **Objectives**

* **Explain branching and merging**

**Branching**

Creates a separate copy of your code history starting from the current point.

Lets you experiment, add features, or fix bugs **without disturbing the main code**.

Example:

git branch feature-login # create branch

git checkout feature-login # switch to branch

or in one step:

git checkout -b feature-login

**Merging**

Combines changes from one branch into another.

Usually: merge your **feature branch** into the **main branch** once it’s tested.

Example:

git checkout main

git merge feature-login

* **Explain about creating a branch request in GitLab**

In GitLab UI (no terminal needed):

1. Go to your **repository**.
2. Click **Repository → Branches**.
3. Click **New branch**.
4. Enter:
   * **Branch name** (e.g., feature-login).
   * **Source branch** (usually main).
5. Click **Create branch**.

This creates a safe place to work without touching the main code.

* **Explain about creating a merge request in GitLab**

A **Merge Request** (MR) is GitLab’s way of saying “Please review my branch changes before adding them to another branch.”

Steps:

1. After pushing your branch changes, GitLab will often suggest **“Create merge request”**.
2. Or manually:
   * Go to **Merge requests** → **New merge request**.
   * Choose:
     + **Source branch** = your feature branch.
     + **Target branch** = where you want to merge (e.g., main).
3. Add:
   * **Title** (clear, short description of the work).
   * **Description** (what changes you made, why, any testing info).
4. Assign reviewers if needed.
5. Click **Submit merge request**.

Once reviewers approve, you or they can click **Merge** to combine the code.

In this hands-on lab, you will learn how to:

* Construct a branch, do some changes in the branch, and merge it with master (or trunk)

## **Prerequisites**

The following are the pre-requisites to complete this hands-on lab:

* Setting up Git environment with P4Merge tool for Windows

Notes\*:

|  |
| --- |
| Please follow the below steps for creating a free account in GitHub.  Do not use cognizant credentials to login to GitHub. |

Estimated time to complete this lab: **30 minutes.**

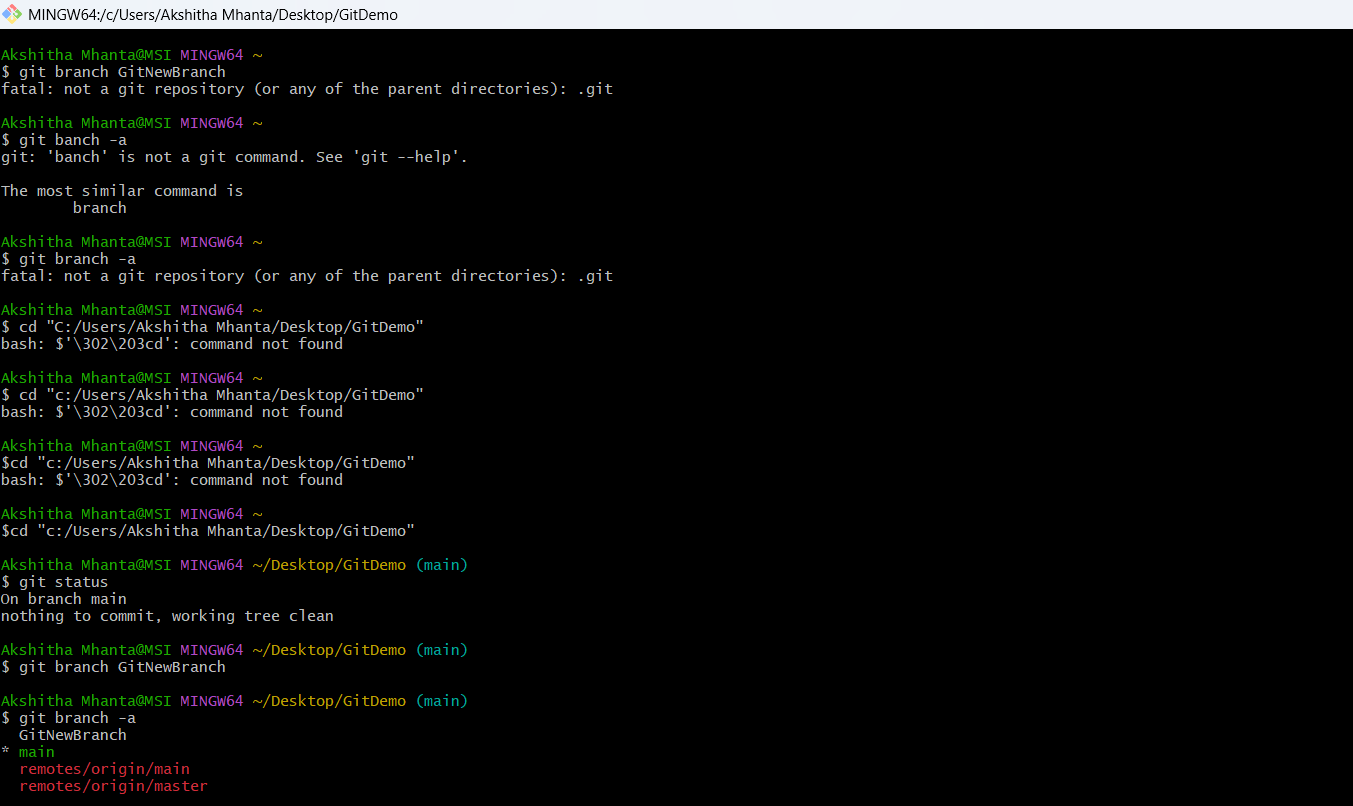
Please follow the instruction to complete the hands-on. Each instruction expects a command for the Git Bash.

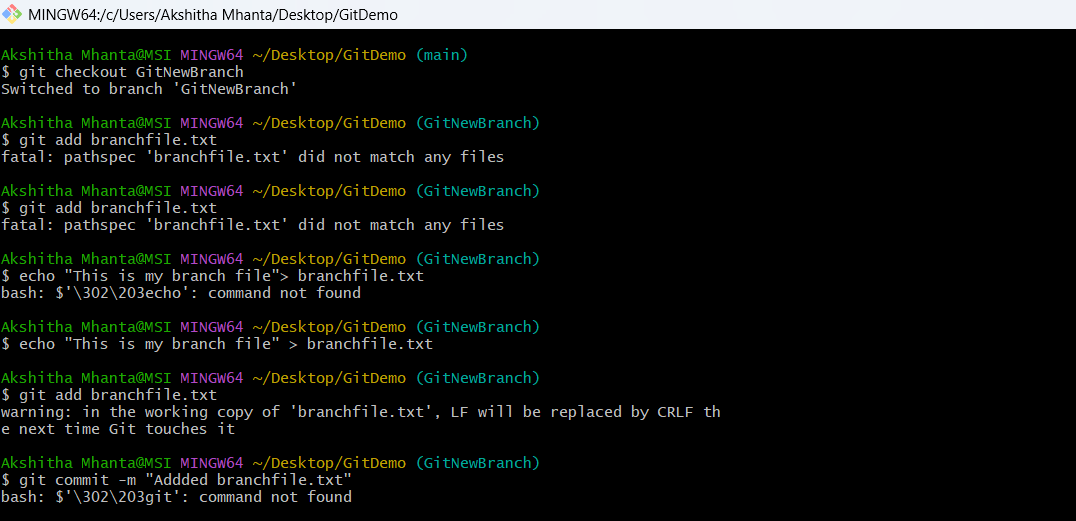
**Branching:**

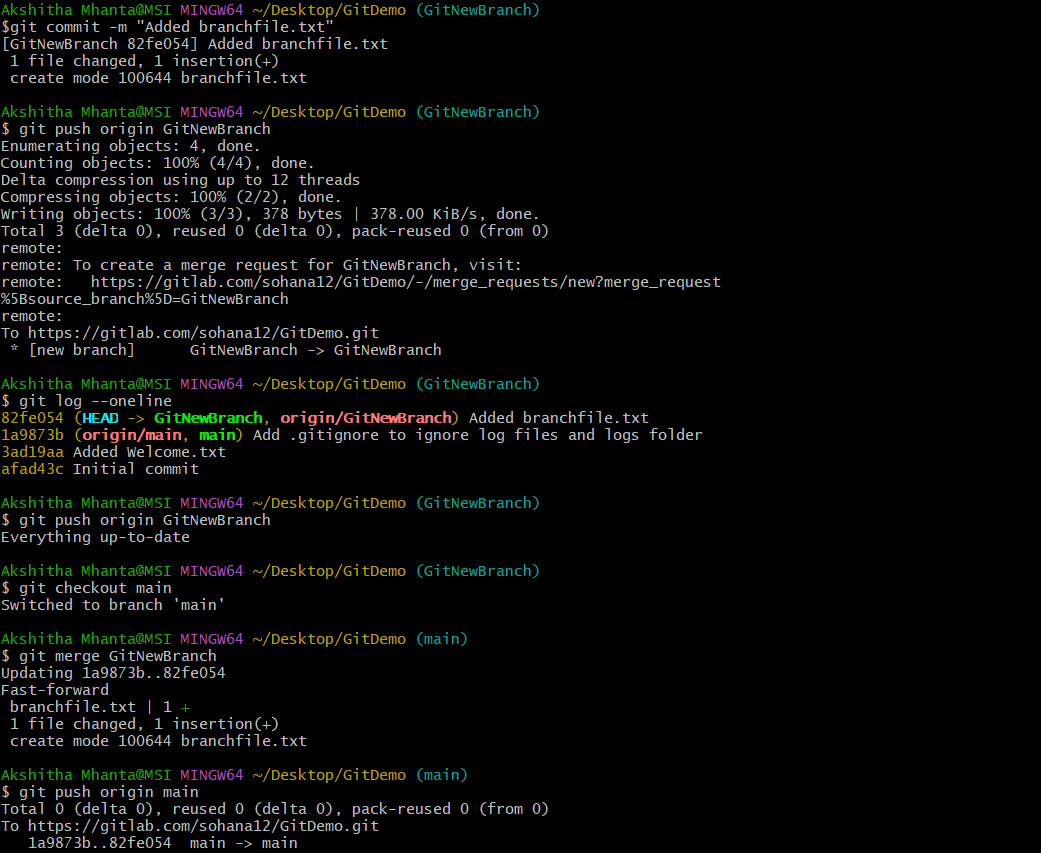
1. Create a new branch **“GitNewBranch”.**
2. List all the local and remote branches available in the current trunk. Observe the “\*” mark which denote the current pointing branch.
3. Switch to the newly created branch. Add some files to it with some contents.
4. Commit the changes to the branch.
5. Check the status with **“git status”** command.

**Merging:**

1. Switch to the master
2. List out all the differences between trunk and branch. These provide the differences in command line interface.
3. List out all the visual differences between master and branch using **P4Merge tool**.
4. Merge the source branch to the trunk.
5. Observe the logging after merging using **“git log –oneline –graph –decorate”**
6. Delete the branch after merging with the trunk and observe the git status.







Output:

