**3. Git-HOL**

## **Objectives**

* **Explain branching and merging**

Branching in Git means creating a **separate line of development** based on your existing code.

* You can make changes in a branch without affecting the master (or main) branch.
* Useful for adding features, fixing bugs, or experimenting.
* In Git, branches are lightweight and quick to create.

Merging is the process of **combining changes from one branch into another**.

* Typically, you merge a feature branch into master after development is complete.
* Git merges automatically if there are no conflicting changes.
* If conflicts occur, you must resolve them before completing the merge.
* **Explain about creating a branch request in GitLab**

A **branch request** in GitLab simply means **pushing your new branch to the remote GitLab repository** so others can see it.  
Steps:

1. Create a branch locally.
2. Push it to GitLab:

git push -u origin branch\_name

1. This makes the branch appear in GitLab’s **Branches** section.

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

A **Merge Request** (MR) in GitLab is a request to merge changes from one branch into another (e.g., from GitNewBranch into master).

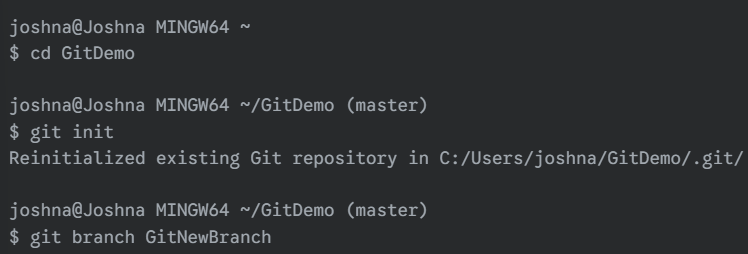
* It allows for code review, discussion, and approval before merging.
* Created via the GitLab web interface after you’ve pushed your branch.

Steps:

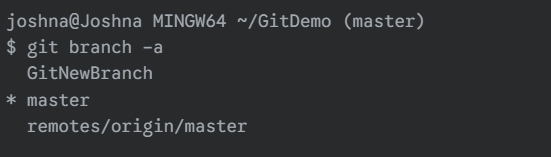
1. Push your branch to GitLab.
2. Go to GitLab → your project → **Merge Requests** → **New Merge Request**.
3. Choose source branch (your branch) and target branch (master).
4. Fill in title/description → Create Merge Request.

**Branching:**

1. Creating a new branch **“GitNewBranch”.**

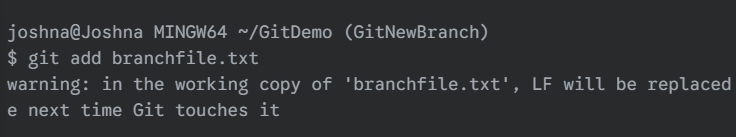


1. List all the local and remote branches available in the current trunk. Observe the “\*” mark which denote the current pointing branch.

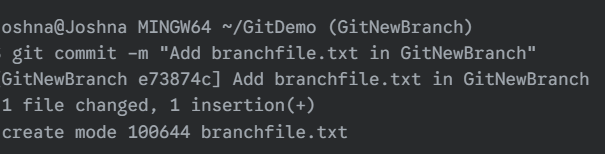


1. Switch to the newly created branch. Add some files to it with some contents.

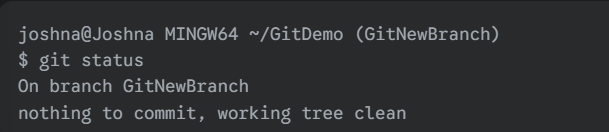




1. Commit the changes to the branch.

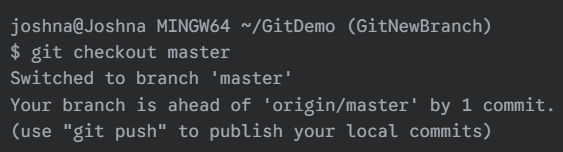


1. Check the status with **“git status”** command.



**Merging:**

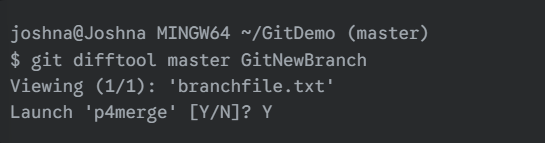
1. Switch to the master



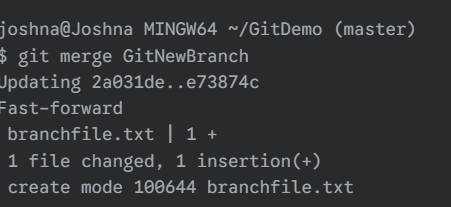
1. List out all the differences between trunk and branch. These provide the differences in command line interface.



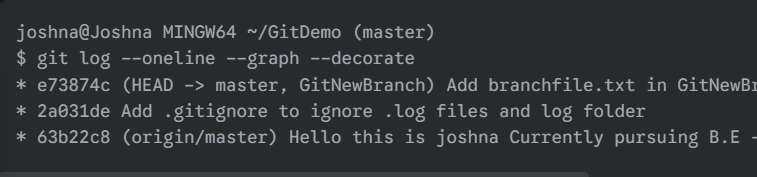
1. List out all the visual differences between master and branch using **P4Merge tool**.



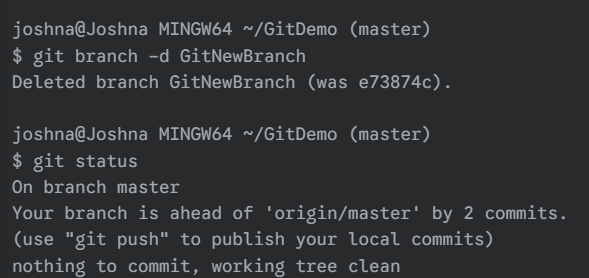
1. Merge the source branch to the trunk.



1. Observe the logging after merging using **“git log –oneline –graph –decorate”**

****

1. Delete the branch after merging with the trunk and observe the git status.

****