## **Objectives**

* Explain branching and merging
* Explain about creating a branch request in GitLab
* Explain about creating a merge request in GitLab

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.

**Solution :**

**Branching**

**1. Create a new branch GitNewBranch**

git branch GitNewBranch

**2. List all local and remote branches**

git branch -a

The \* marks the branch you’re currently on.

**3. Switch to the newly created branch**

git checkout GitNewBranch

4.Add some files to it with content

echo "File in GitNewBranch" > branchfile1.txt echo "Another file in GitNewBranch" > branchfile2.txt

5. Stage and commit changes

git add branchfile1.txt branchfile2.txt git commit -m "Added branch-specific files"

6. Check the branch status

git status

**Merging**

**1. Switch to master**

git checkout master

**2. List all differences between master and branch (command line)**

git diff master GitNewBranch

**3. List visual differences using P4Merge**  
(assuming P4Merge is installed and configured as Git’s diff tool)

git difftool master GitNewBranch

**4. Merge GitNewBranch into master**

git merge GitNewBranch

**5. Observe commit history after merge**

git log --oneline --graph --decorate

**6. Delete the branch after merging**

git branch -d GitNewBranch

1. **Verify status**git status