

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

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.

Make sure you are in your repo

```
cd ~/GitDemo
```

Check your current branch:

```
git branch
```

Create a new branch

```
git checkout -b GitNewBranch
```

Now you're working in the new branch.

Make some changes in the new branch

Example: edit welcome.txt:

```
echo "This line was added in the feature branch" >> welcome.txt
```

Check the change:

```
cat welcome.txt
```

```
LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ git checkout -b GitNewBranch
Switched to a new branch 'GitNewBranch'

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (GitNewBranch)
$ echo "This line was added in the feature branch" >> welcome.txt

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (GitNewBranch)
$ cat welcome.txt
Welcome to GitDemo
This file was created for the Git hands-on lab.
This line was added in the feature branch
```

Stage & commit the changes

```
git add welcome.txt
```

```
git commit -m "Add new line in welcome.txt from GitNewBranch branch"
```

```
LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (GitNewBranch)
$ git add welcome.txt
warning: in the working copy of 'welcome.txt', LF will be replaced by CRLF the next time Git touches it

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (GitNewBranch)
$ git commit -m "Add new line in welcome.txt from GitNewBranch branch"
[GitNewBranch fccbb92] Add new line in welcome.txt from GitNewBranch branch
1 file changed, 1 insertion(+)
```

Push the new branch to remote

```
git push -u origin GitNewBranch
```

```

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (GitNewBranch)
$ git push -u origin GitNewBranch
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 397 bytes | 397.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'GitNewBranch' on GitHub by visiting:
remote:   https://github.com/pooja-skcet/GitDemo/pull/new/GitNewBranch
remote:
To https://github.com/pooja-skcet/GitDemo.git
 * [new branch]      GitNewBranch -> GitNewBranch
branch 'GitNewBranch' set up to track 'origin/GitNewBranch'.

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (GitNewBranch)
$ git status
On branch GitNewBranch
Your branch is up to date with 'origin/GitNewBranch'.

nothing to commit, working tree clean

```

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.

Switch back to main branch

```

git checkout main
git switch main

```

View the differences between main and GitNewBranch (CLI)

```

git diff main GitNewBranch

```

```

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (GitNewBranch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ git diff main GitNewBranch
diff --git a/welcome.txt b/welcome.txt
index c347666..9afb84d 100644
--- a/welcome.txt
+++ b/welcome.txt
@@ -1,2 +1,3 @@
 Welcome to GitDemo
 This file was created for the Git hands-on lab.
+This line was added in the feature branch

```

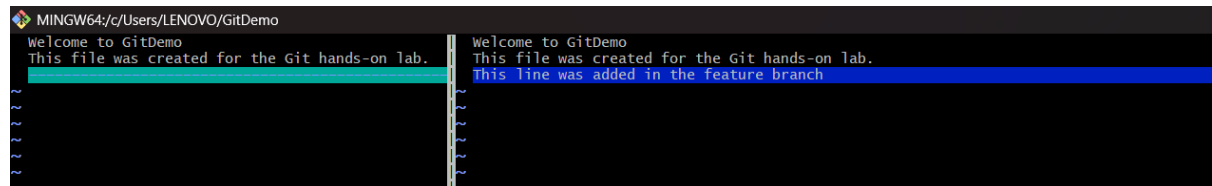
View differences visually with P4Merge

git difftool main GitNewBranch

```
LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ git difftool main GitNewBranch

This message is displayed because 'diff.tool' is not configured.
See 'git difftool --tool-help' or 'git help config' for more details.
'git difftool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld kompare gvimdiff diffuze diffmerge ecmerge p4merge araxis bc codecom
pare smerge emerge vimdiff nvimdiff

Viewing (1/1): 'welcome.txt'
Launch 'vimdiff' [Y/n]? y
2 files to edit
```



Merge GitNewBranch into main

While still on main:

git merge GitNewBranch

- If there are **no conflicts** → merge completes immediately.
- If there **are conflicts** → Git will ask you to resolve them (P4Merge can help here).

Push the updated main to remote

git push origin main

```
LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ git merge GitNewBranch
Updating 46bf5cd..fccbb92
Fast-forward
 welcome.txt | 1 +
 1 file changed, 1 insertion(+)

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/pooja-skcet/GitDemo.git
 46bf5cd..fccbb92  main -> main
```

View the commit history graph

git log --oneline --graph --decorate

This shows the merge in a compact graph form.

Delete the merged branch locally

git branch -d GitNewBranch

Delete the branch on GitHub (remote)

git push origin --delete GitNewBranch

```
LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ git log --oneline --graph --decorate
* fccbb92 (HEAD -> main, origin/main, origin/GitNewBranch, GitNewBranch) Add new line in welcome.txt from GitNewBranch branch
* 46bf5cd Add .gitignore to ignore .log files and log folder
* e2e88a9 Hello all...Now I am working on GIT Hands-on week8

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ git branch -d GitNewBranch
Deleted branch GitNewBranch (was fccbb92).

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ git push origin --delete GitNewBranch
To https://github.com/pooja-skcet/GitDemo.git
- [deleted]          GitNewBranch

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```