

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.

```

nkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git push origin master
Everything up-to-date

nkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git branch GitNewBranch

nkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git branch -a
  GitNewBranch
* master
  remotes/origin/master

nkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git checkout GitNewBranch
Switched to branch 'GitNewBranch'

nkanm@Kanmani MINGW64 ~/GitDemo (GitNewBranch)
$ echo "This is from GitNewBranch" > branchfile.txt
$ git add branchfile.txt
warning: in the working copy of 'branchfile.txt', LF will be replaced by CRLF the
next time Git touches it

nkanm@Kanmani MINGW64 ~/GitDemo (GitNewBranch)
$ git commit -m "Added branchfile.txt in GitNewBranch"
[GitNewBranch eb39665] Added branchfile.txt in GitNewBranch
 1 file changed, 1 insertion(+)
 create mode 100644 branchfile.txt

nkanm@Kanmani MINGW64 ~/GitDemo (GitNewBranch)
$ git status
On branch GitNewBranch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore.bak

nothing added to commit but untracked files present (use "git add" to track)

```

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.

```

mkanm@Kanmani MINGW64 ~/GitDemo (GitNewBranch)
$ git checkout master
Switched to branch 'master'

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git diff master GitNewBranch
diff --git a/branchfile.txt b/branchfile.txt
new file mode 100644
index 0000000..bd667bc
--- /dev/null
+++ b/branchfile.txt
@@ -0,0 +1 @@
+This is from GitNewBranch

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecme
rge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff
No files need merging

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git merge GitNewBranch
Updating 88c907a..eb39665
Fast-forward
 branchfile.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 branchfile.txt

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git log --oneline --graph --decorate
* eb39665 (HEAD -> master, GitNewBranch) Added branchfile.txt in GitNewBranch
* 88c907a (origin/master) Added .gitignore to ignore log files and logs folder
* 2dd964b Added welcome.txt

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git branch -d GitNewBranch
Deleted branch GitNewBranch (was eb39665).

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore.bak

nothing added to commit but untracked files present (use "git add" to track)

```

KANMANI MURUGHAIYAN

SUPERSET ID: 6407636