Git-HOL

Hands-on in this document

**Branching & Merging – Hands-on Lab**

**Objectives**

* Understand branching and merging in Git.
* Learn how to create a branch request in GitLab.
* Learn how to create a merge request in GitLab.

**Estimated time:** ~30 minutes.

**Prerequisites**

* Git environment set up.
* P4Merge tool installed and configured as Git’s merge/diff tool.
* Local repository linked to a remote repository (GitLab/GitHub).
* **Do not use Cognizant credentials** for GitHub/GitLab.

**Step 1 – Branching**

1. **Create a new branch**:
2. git branch GitNewBranch
3. **List all branches** (local & remote):
4. git branch -a

\* indicates the current branch.

1. **Switch to the new branch**:
2. git checkout GitNewBranch
3. **Add a file to the new branch**:
4. echo "This is a file in GitNewBranch" > branchfile.txt
5. git add branchfile.txt
6. **Commit changes**:
7. git commit -m "Added branchfile.txt in GitNewBranch"
8. **Check branch status**:
9. git status

**Step 2 – Merging**

1. **Switch to master (or main) branch**:
2. git checkout master
3. **Show CLI differences between master and branch**:
4. git diff master GitNewBranch
5. **Show visual differences using P4Merge**:
6. git difftool master GitNewBranch

P4Merge will open to display differences.

1. **Merge branch into master**:
2. git merge GitNewBranch
3. **View merge history**:
4. git log --oneline --graph --decorate
5. **Delete merged branch**:
6. git branch -d GitNewBranch
7. **Check status**:
8. git status

**Step 3 – GitLab Merge Request Workflow**

1. Push your branch to GitLab:
2. git push origin GitNewBranch
3. Go to your GitLab project → **Merge Requests** → **New Merge Request**.
4. Select:
   * **Source branch:** GitNewBranch
   * **Target branch:** master (or main)
5. Add a **title** and **description**, then click **Create Merge Request**.
6. After review, merge the request in GitLab.
7. Pull latest changes into local master:
8. git pull origin master