**Branching**

Creating a separate line of development in a repository so you can work on new features or fixes without affecting the main (master/trunk) code.

**Merging**

Combining changes from one branch into another (usually into master) so all updates are integrated.

**Creating a Branch in GitLab**

1. Go to your project in GitLab.
2. Click **Repository → Branches → New branch**.
3. Enter branch name and base branch, then click **Create branch**.

**Creating a Merge Request in GitLab**

1. Go to **Merge Requests → New merge request**.
2. Select **source branch** (your feature branch) and **target branch** (e.g., master).
3. Add title, description, reviewers, then click **Submit merge request**.

**Branching**

**Create a new branch “GitNewBranch”:**

git branch GitNewBranch

**List all local & remote branches:**

git branch -a

* \* shows which branch you are currently on.

**Switch to the new branch:**

git checkout GitNewBranch

**Add some files to it with content:**

echo "This is a new branch file" > newfile.txt

**Stage & commit changes:**

git add newfile.txt

git commit -m "Add new file in GitNewBranch"

**Check status:**

git status

**Merging**

**Switch back to master:**

git checkout master

**Show differences between master & branch (CLI):**

git diff master GitNewBranch

**Show visual differences with P4Merge:**

git difftool master GitNewBranch

*(This assumes you have P4Merge configured as your difftool.)*

**Merge the branch into master:**

git merge GitNewBranch

**View merge log:**

git log --oneline --graph --decorate

**Delete the branch after merging:**

git branch -d GitNewBranch

**Check status:**

git status