**Explain branching and merging**

**Branching** is the process of creating a separate, parallel line of development from the main codebase. Think of it like making a copy of your project's files at a specific point in time so you can work on them without affecting the original. This allows developers to work on new features or bug fixes in isolation.

**Merging** is the process of combining those separate lines of development back together. Once a feature on a branch is complete and stable, you merge its changes back into the main branch. This integrates the new code into the project's primary codebase

**Explain about creating a branch request in GitLab**

In GitLab, you can create a branch directly from the web interface. This is often done when you start working on a new task or feature.

1. Navigate to your project's repository in GitLab.
2. Click on the **"Branches"** tab in the left-hand navigation menu.
3. Click the **"New branch"** button.
4. Enter a name for your new branch. It's good practice to use a descriptive name like feature/add-login-button or bugfix/fix-payment-issue.
5. Select the source branch you want to branch from (usually main or develop).
6. Click **"Create branch"**.

You can now clone this new branch to your local machine and start working

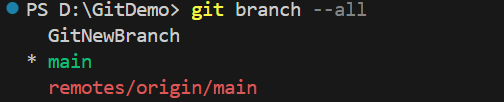
**Explain about creating a merge request in GitLab**

A **Merge Request (MR)** is GitLab's equivalent of a Pull Request. It's a formal way to propose that your changes be merged from one branch to another.

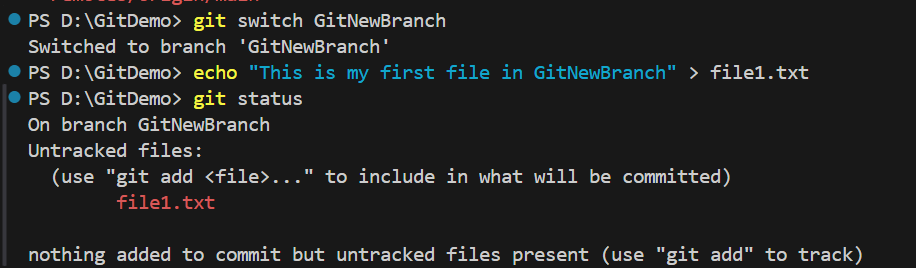
1. **Push your branch**: After making your changes locally, commit them and push your new branch to the remote repository on GitLab.
2. **Create the MR**: GitLab will automatically detect the new branch and display a **"Create merge request"** button or link. Click it.
3. **Fill out the form**:
   * **Source branch**: This is your new branch with your changes.
   * **Target branch**: This is the branch you want to merge into (e.g., main).
   * **Title**: A brief, descriptive title for your changes.
   * **Description**: A detailed explanation of what your changes do, why they were needed, and any relevant issue numbers.
4. **Assign and review**: You can assign a reviewer to your MR and set other options like milestones and labels.
5. **Submit**: Click the **"Create merge request"** button to finalize and open the discussion
6. **Create a new branch “GitNewBranch”.**

****

1. **List all the local and remote branches available in the current trunk. Observe the “\*” mark which denote the current pointing branch**

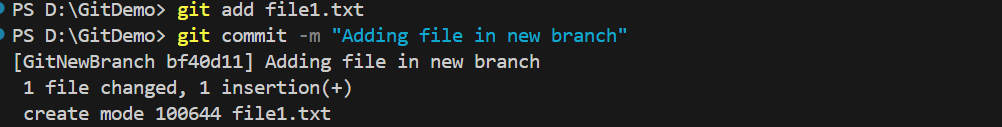
****

**3.Switch to the newly created branch. Add some files to it with some contents**

****

**4. Commit the changes to the branch.**

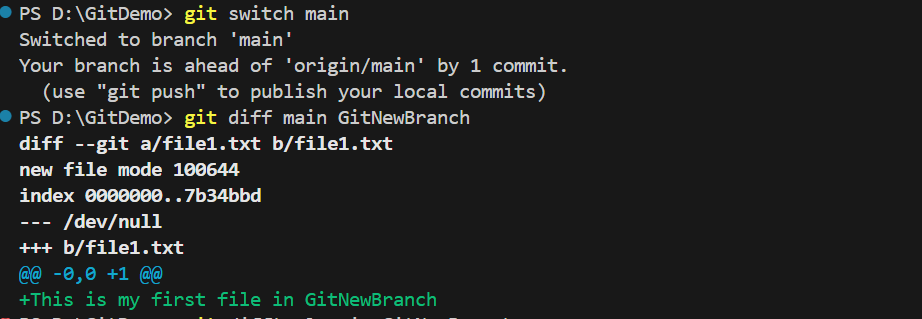
**5. Check the status with “git status” command**

****

**Merging**

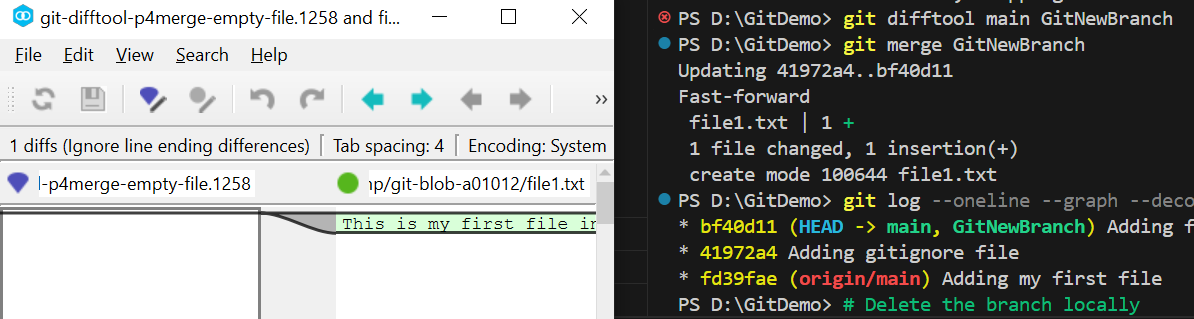
**1. Switch to the master**

**2. List out all the differences between trunk and branch. These provide the differences in command line interface.**

****

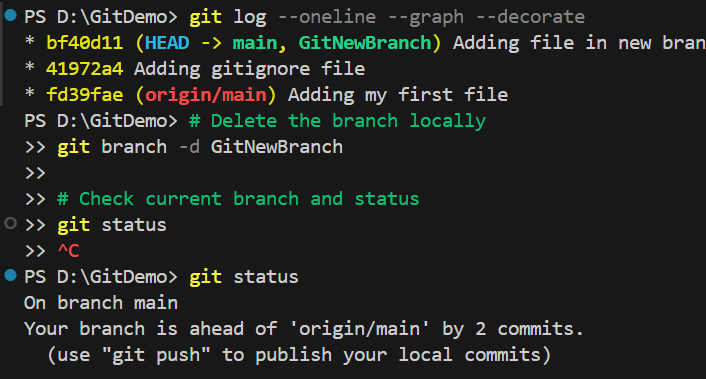
**3. List out all the visual differences between master and branch using P4Merge tool.**

**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**

****