**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

Set Up Git Branching Create a new branch in your Git

repository for testing. Add a new feature and merge it

Name: Bhavadharani S Department : CSE



**Introduction and Overview**

Git branching is an essential feature thatallows developers to work on new functionalities, bug fixes, or experiments without affecting the main codebase. By using branches, teams can work on different tasks in parallel, test new ideas safely, and merge changes seamlessly into the main project.

In this POC, you will create a new Git branch, make changes, and merge it into the main branch. This process ensures a structured and efficient workflow while maintaining code integrity.

**Objective**

By the end of this POC, you will:

* Understand the concept of Git branching and why it is essential.
* Create a new branch for feature development or testing.
* Make modifications and commit changes in the new branch.
* Merge the changes back into the main branch without conflicts.
* Optimize version control by organizing code efficiently.

**Importance of Git Branching**

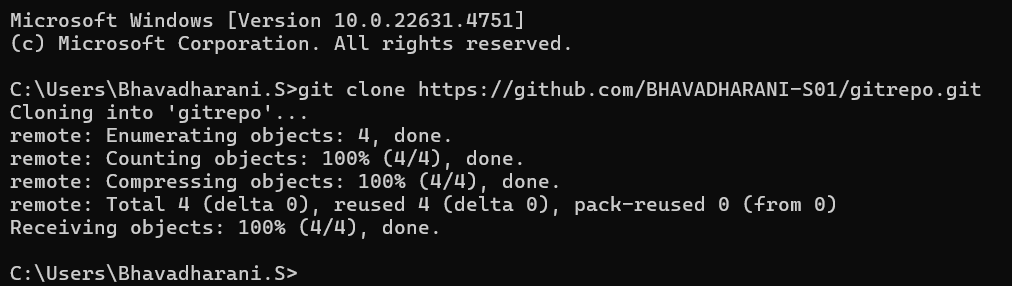
* Parallel Development: Developers can work on multiple features simultaneously without affecting the main project.
* Code Safety: Changes in a branch do not impact the stable version until they are merged.
* Efficient Collaboration: Multiple developers can work on different tasks without conflicts.
* Rollback Capability: If something goes wrong, you can easily discard changes in a branch without impacting the main codebase.
* Structured Workflow: Helps maintain clean and organized development processes, ensuring a smooth CI/CD pipeline.

**Step-by-Step Overview**

**1. Clone your repository (if not already cloned)**

If you haven't cloned your repository yet, run:

***git clone https://github.com/your-username/gitrepo.git***



**2. Navigate to your repository folder**

***cd gitrepo***

**3. Create a new branch for testing**

To create a new branch called test-branch, run:

***git checkout -b test-branch***

******

**4. Add a new feature**

Make your changes or add new features to the code in this branch.

**5. Stage and commit the changes**

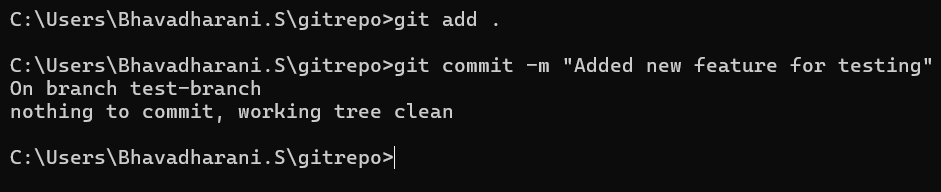
After adding the new feature, check the status of your changes:

***git status***

Then, stage and commit the changes:

***git add .***

***git commit -m "Added new feature for testing"***

******

6. Push the new branch to the remote repository

Push your test-branch to GitHub:

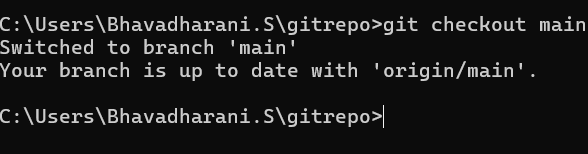
***git push origin test-branch***



**7. Switch to the main branch**

Switch back to the main branch:

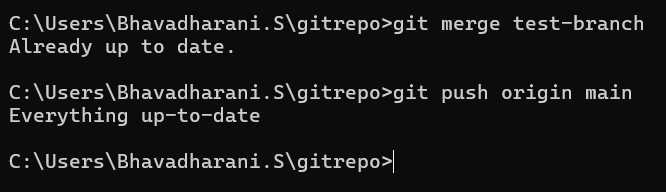
***git checkout main***

******

**8.** Merge the test-branch into main

Now, merge the test-branch into the main branch:

***git merge test-branch***



**9. Push the merged changes to the remote repository**

Finally, push the merged changes to GitHub:

***git push origin main***

**10. Clean up (optional)**

If you no longer need the test-branch, delete it:

***git branch -d test-branch***

To delete the remote branch:

***git push origin --delete test-branch***

