

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

# Set Up Git Branching: Create a new branch in your Git repository for testing . Add a newfeature and merge it

Name: MADHUMETHRA A P

Department: IT



**Introduction**

## In this Proof of Concept (POC), Git is used for version control to manage the development workflow. Git allows developers to create separate branches for new features, isolate them from the main branch, and merge them back after completion. This ensures organized and collaborative development.

**Overview**

## Create a new branch for testing

## - Add a new feature and commit changes

## - Merge the testing branch with the master branch

## - Resolve conflicts (if any)

## - Push updated master branch to remote repository

**Objectives**

# Create a new branch for testing using git branch testing.

## Switch to the new branch using git checkout testing.

## Make changes to your code, adding a new feature.

## Commit your changes using git add . and git commit -m "New feature added".

## Switch back to the master branch using git checkout master.

## Merge the testing branch with the master branch using git merge testing.

## If there are conflicts, resolve them manually.

## Push the updated master branch to the remote repository using git push origin master.

**Importance**

## Version Control: Manage different versions of your codebase.

## - Collaboration: Work with others on the same project without conflicts.

## - Risk Reduction: Test new features without affecting the stable codebase.

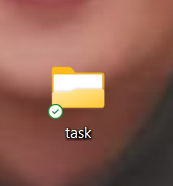
## - Efficient Development: Work on multiple features simultaneously without disrupting the main codebase.

## - Easy Rollback: Revert to a previous version if something goes wrong.

**Step-by-Step Overview**

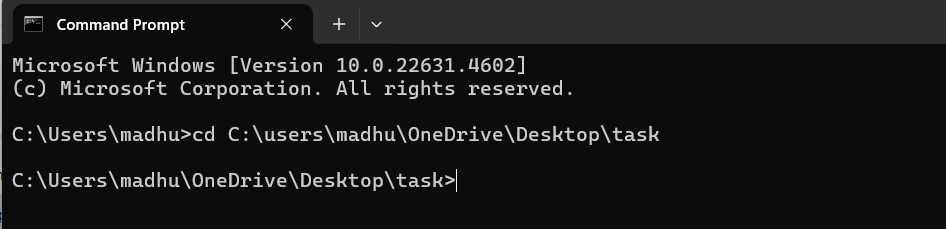
Step 1

Create a folder and name



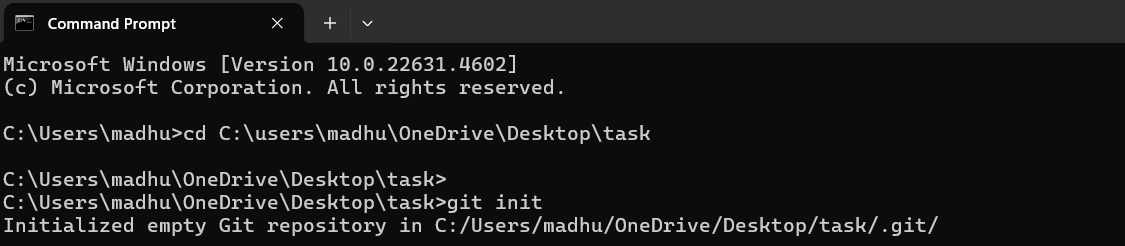
Step 2

# Set the path to the folder created in first step (Git\_Branching)



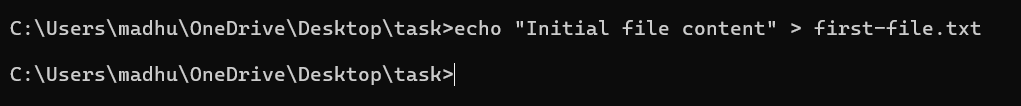
Step 3

## Initialize Git by typing this command: git init This command will create a .git folder inside your folder,



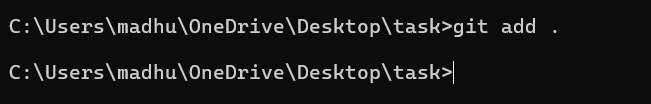
Step 4

## Create a simple file to start the repository



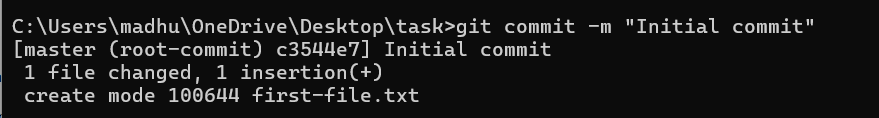
Step 5

## Addthe File to Git Tell Git to track this file



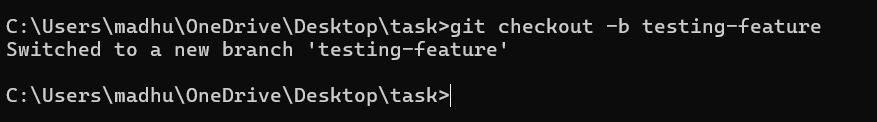
Step 6

## Save this change in Git with a commit message



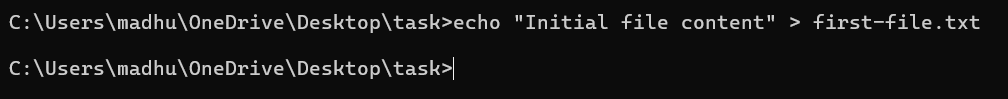
Step 7

## Create and switch to a new branch called testing-feature



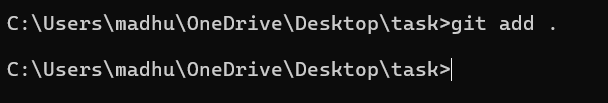
Step 8

## Let’s add a newfile for our feature



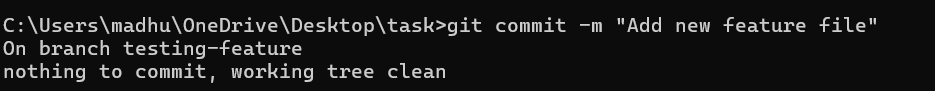
Step 9

## Now, stage the changes



Step 10

## Commit the changes



Step 11

## Switch to the master Branch



Step 12

## Merge Changes from testing-feature to master



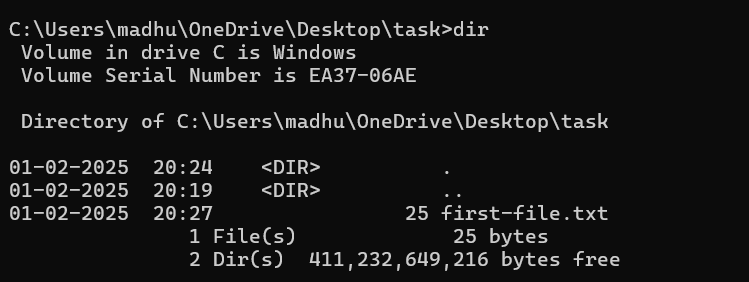
Step 13

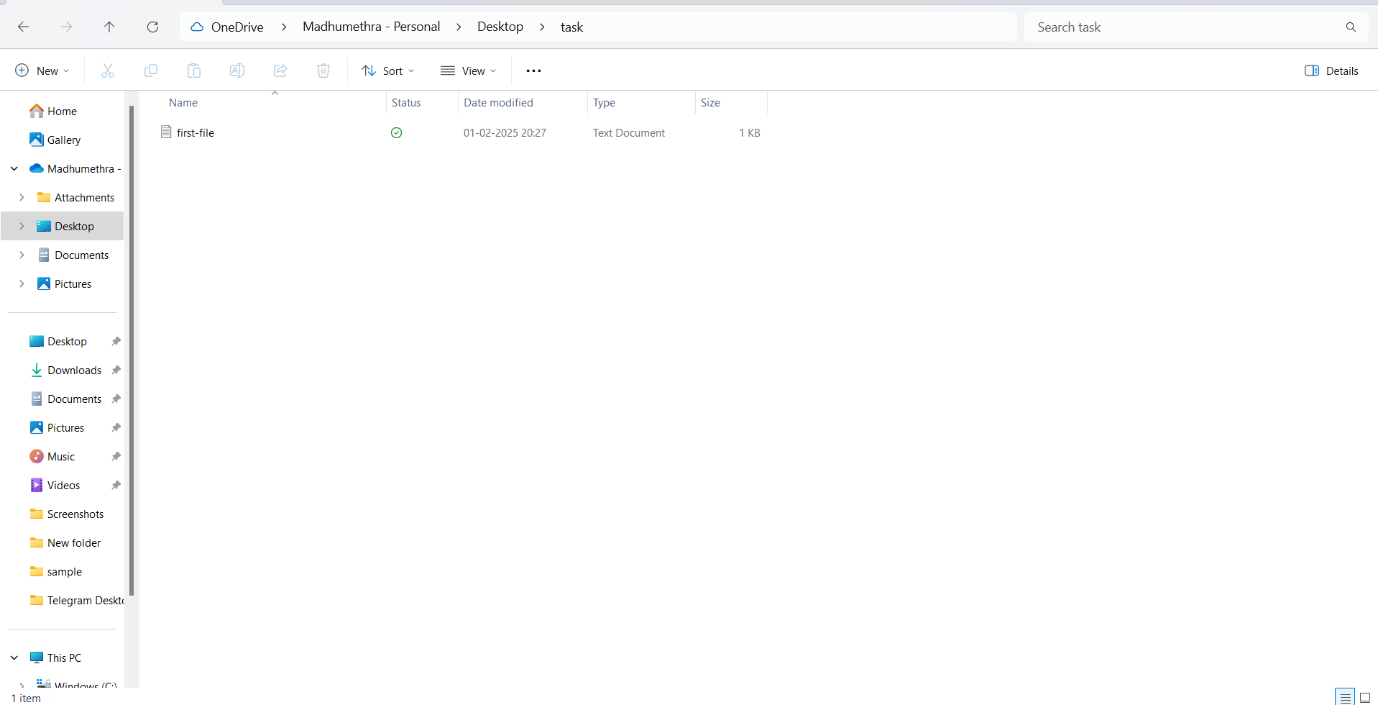
## Once the merge is done, you can delete the testing-feature branch



Step 14

## Now, check the files in the folder





**Output**

## Successful Testing: New features are thoroughly tested without disrupting the main codebase.

## - Stable Codebase: The main codebase remains stable and unaffected by new feature development.

## - Improved Collaboration: Team members can work on different features simultaneously without conflicts.

## - Faster Development: New features are developed and deployed quickly and efficiently.

## - Reduced Errors: Errors are reduced, and issues are easily resolved through version control.