**5.Git – HOL**

**1.Explain how to clean up and push back to remote Git**

1. Clean Unwanted Files

Delete any unwanted or temporary files from your project folder (like .log, backup, etc).

2. Stage the Cleaned Files

git add .

This adds your cleaned-up files to Git for tracking.

3. Commit the Clean State

git commit -m "Cleaned up unwanted files"

This saves your clean-up changes locally.

4. Push to Remote

git push origin your-branch-name

This uploads your cleaned version to GitHub or remote repo.

5. Verify on GitHub

Go to your GitHub repo and check that unwanted files are gone and your latest commit is shown.

**Please follow the instructions to complete the hands-on. Each instruction expects a command for the Git Bash**.

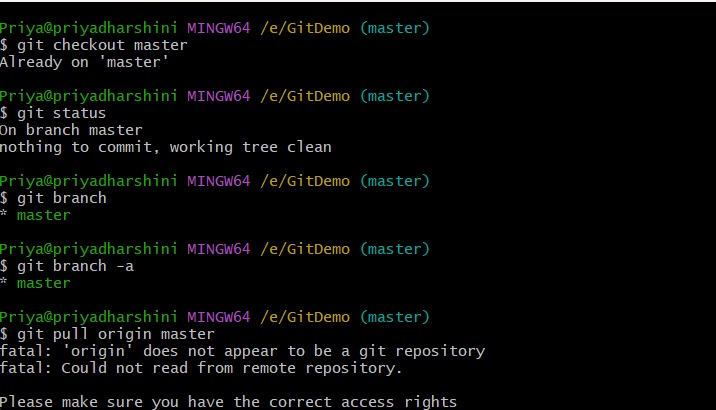
1. Verify if master is in clean state.
2. List out all the available branches.
3. Pull the remote git repository to the master
4. Push the changes, which are pending from “Git-T03-HOL\_002” to the remote repository.
5. Observe if the changes are reflected in the remote repository**.**

**1. Verify if master is in a clean state**

git checkout master

git status

**Expected output:**



**2. List out all the available branches**

git branch # local branches

git branch -a # all branches (local + remote)

**3. Connect Your Local Repo to a Remote (if not already done)**

Create a repo at: <https://github.com/new>

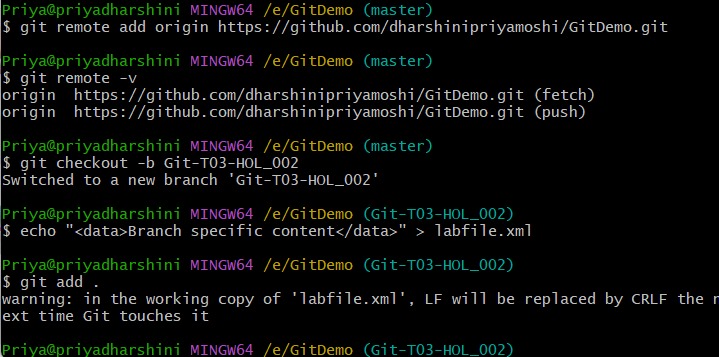
git remote add origin https://github.com/<your-username>/GitDemo.git

Example:

git remote add origin https://github.com/dharshinipriyamoshi/GitDemo.git

Verify:

git remote -v



**4.Pull the remote repository to your local master**

git pull origin master

**5. Create the branch Git-T03-HOL\_002 locally**

git checkout -b Git-T03-HOL\_002

**6. Make any changes (optional)**

echo "<data>Branch specific content</data>" > labfile.xml

**7. Stage and commit the changes**

git add .

git commit -m "Git-T03-HOL\_002: Added labfile.xml with sample content"

**8. Push the branch to remote**

git push -u origin Git-T03-HOL\_002

🛠 If prompted for username/password:

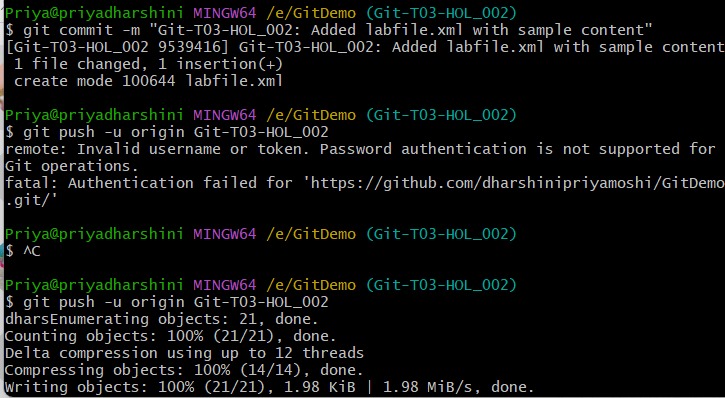
* Username: your GitHub username
* Password: your Personal Access Token (PAT)

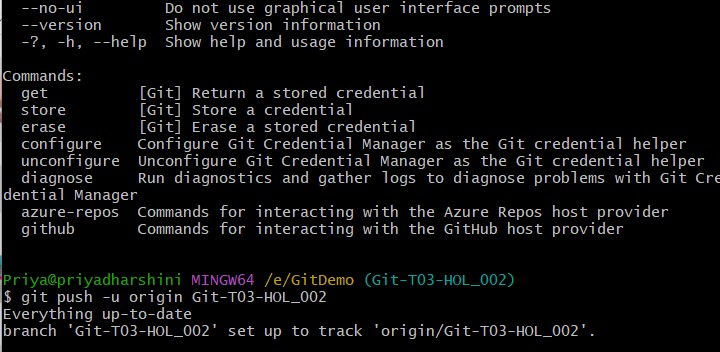
**9. Create a GitHub Personal Access Token (if needed)**

Go to: <https://github.com/settings/tokens>  
Create a token with:

* repo scope
* Valid for 30–90 days

Copy the token and use it as your password during Git push.





**10. If Wrong Credentials Were Saved**

git credential-manager reject https://github.com

Then push again:

git push -u origin Git-T03-HOL\_002

**11. Verify if the changes are reflected**

**Option 1** – Browser:

* Go to your repo: https://github.com/<your-username>/GitDemo
* Switch to the branch: Git-T03-HOL\_002
* See if your file (e.g., labfile.xml) and commit is there

**Option 2** – Visual log of all branches:

git log --oneline --graph --decorate –all

