**5. Git-HOL**

### Step 1: Open Git Bash (or Terminal)

1. Open Git Bash on Windows: Start Menu → Git Bash or right-click inside the project folder → Git Bash Here.

On Linux/Mac, open Terminal.

Expected: A terminal prompt opens.

### Step 2: Navigate to repository root

2. Change directory to your local repository root. Replace the path below with your repo path.

Command: cd /path/to/your/repository

Example: cd ~/projects/Git-T03-HOL\_002

Verify: Command: pwd (prints current path)

### Step 3: Confirm Git is available (optional but recommended)

3. Check Git version to ensure Git is installed.

Command: git --version

Expected: Shows git version, e.g. git version 2.x.x

### Step 4: Check current branch and switch to master

4. See the branch you are on:

Command: git branch --show-current

If it is not 'master', switch to master explicitly:

Command: git checkout master

(If your repo uses 'main' instead of 'master', replace 'master' with 'main' in all commands.)

Verify: Command: git branch --show-current (should print 'master')

### Step 5: Verify master is clean (no uncommitted changes)

5. Check working tree status:

Command: git status --porcelain

- If output is empty, working tree is clean.

- Or use: git status (gives human readable status)

If you see changes, do not pull until you resolve them. Follow the next small steps to resolve.

### Step 6: If master is NOT clean — option A: save changes temporarily (stash)

6A. To save (stash) local changes so you can update master without losing work:

Command: git stash push -m "WIP: saving changes before pull"

Verify stash: git stash list

To re-apply later: git stash apply (or git stash pop to apply and remove stash).

### Step 7: If master is NOT clean — option B: commit your changes

6B. To keep changes permanently, commit them now:

Command: git add . (or list specific files)

Command: git commit -m "Describe what you changed - e.g. Save work before pull"

Verify: git status (should be clean)

### Step 8: If master is NOT clean — option C: discard changes (USE WITH CAUTION)

6C. To discard all local changes (irreversible):

Command: git reset --hard

Or discard specific files: git restore <file>

Verify: git status (should be clean)

### Step 9: Check remote configuration

7. Confirm the remote name and URL (usually 'origin'):

Command: git remote -v

Expected: lists fetch/push URLs for origin.

### Step 10: List all branches (local and remote)

8. List branches so you can see available branches locally and remotely:

Command: git branch -a

- Local branches are shown without prefix.

- Remote branches are shown as remotes/origin/<branch>.

For more detail (show tracking info): git branch -vv

### Step 11: Fetch latest refs from origin (safe)

9. Update remote tracking references without changing working files:

Command: git fetch origin

This makes sure your local view of remote branches is up-to-date.

### Step 12: Pull remote changes into master

10. While on master, pull the remote master into your local master so it's up-to-date:

Command: git pull origin master

- If your remote's default branch is 'main', run: git pull origin main

Expected: Fast-forward or merge messages. If there are merge conflicts, resolve them (the command will tell you which files).

### Step 13: Verify master is up to date

11. Confirm local master is up-to-date:

Command: git status

Optionally inspect recent commits:

Command: git log --oneline -n 5

Expected: no uncommitted changes; your branch up-to-date with origin/master.

### Step 14: Switch to the lab branch 'Git-T03-HOL\_002'

12. Check out the branch that contains the pending changes:

Command: git checkout Git-T03-HOL\_002

If the branch does not exist locally but exists remotely, create a local tracking branch:

Command: git checkout -b Git-T03-HOL\_002 origin/Git-T03-HOL\_002

Verify: git branch --show-current (should print Git-T03-HOL\_002)

### Step 15: Check working tree on Git-T03-HOL\_002 and commit any changes

13. Inspect status on this branch:

Command: git status

- If files are changed and need to be saved, stage and commit them:

Command: git add .

Command: git commit -m "Describe the lab changes: e.g. Complete Git-T03-HOL\_002 tasks"

- If nothing to commit, you are ready to push.

### Step 16: Push the branch to remote origin

14. Push your branch and its commits to the remote repository:

If this is the first time pushing this branch upstream:

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

If branch already tracks origin:

Command: git push

Expected: Push success message with new branch or updated refs.

### Step 17: Verify the push from terminal

15. Confirm the remote now has your commits:

Command: git fetch origin

Command: git log origin/Git-T03-HOL\_002 --oneline -n 5

Expected: you see your recent commits listed under origin/Git-T03-HOL\_002.

### Step 18: Confirm on the remote (browser)

16. Open the repository page on GitHub/GitLab/Bitbucket in your browser.

- Go to Branches or Commits and select 'Git-T03-HOL\_002'.

- Confirm your latest commit messages appear.

(This verifies the remote reflects your pushed changes.)

### Step 19: (Optional) Merge branch into master and push master

17. If the lab requires merging this branch into master, follow these steps \*only after\* verifying master is clean and up-to-date:

a) Switch to master: git checkout master

b) Pull latest master: git pull origin master

c) Merge: git merge --no-ff Git-T03-HOL\_002

- If there are conflicts, resolve them in files, then:

git add <resolved-files>

git commit (if merge requires a commit)

d) Push master back to remote: git push origin master

Expected: master now contains changes from Git-T03-HOL\_002 and remote master is updated.

### Step 20: (Optional) Clean up branch (if requested)

18. If the branch was merged and you no longer need it, delete it locally and remotely:

- Delete local branch: git branch -d Git-T03-HOL\_002 (uses -d; fails if not merged)

- Force delete local (if you know what you're doing): git branch -D Git-T03-HOL\_002

- Delete remote branch: git push origin --delete Git-T03-HOL\_002

Verify: git branch -a (remote branch should be gone from remotes/origin).

### Step 21: Common push error: failed to push some refs (non-fast-forward)

19. If git rejects push with 'failed to push some refs' due to non-fast-forward:

a) Fetch and rebase your branch on remote changes: git fetch origin

git rebase origin/Git-T03-HOL\_002

b) Alternatively merge remote into local branch: git pull --rebase origin Git-T03-HOL\_002

c) Then push again: git push

Warning: Coordinate with teammates if rewriting shared history.

### Step 22: Final verification and notes

20. Final quick checks:

- git status (no uncommitted changes)

- git branch -a (confirm branches)

- git remote show origin (shows remote branches and tracking info)

Notes:

- Replace 'master' with 'main' if your repo uses 'main'.

- Always confirm the remote name if not 'origin'.

- When in doubt, use git fetch then inspect remote branches before merging or pushing.

Document created on: 2025-08-09 07:13:58