**2. Git-HOL**

**Hands-on in this document**

**Git Ignore – Hands-on Lab**

**Objectives**

* Understand what .gitignore does.
* Learn how to ignore unwanted files and folders in Git.

**Estimated time:** ~20 minutes.

**Prerequisites**

* Git environment is set up.
* Notepad++ integrated as Git’s default editor.
* A local Git repository is initialized and linked to a remote repository (GitLab/GitHub).
* **Do not use Cognizant credentials** for GitHub/GitLab accounts.

**Step 1 – Understand .gitignore**

* The .gitignore file tells Git which files or folders it should ignore (not track).
* Useful for:
  + Temporary files
  + Build artifacts
  + Logs
  + Sensitive credentials

**Step 2 – Create Test Files to Ignore**

1. Open Git Bash in your repository folder.
2. Create a .log file and a log folder:
3. echo "This is a test log file" > debug.log
4. mkdir log
5. echo "Log file content" > log/error.log

**Step 3 – Create/Edit .gitignore**

1. Create .gitignore file (if not exists):
2. notepad++ .gitignore
3. Add the following lines:
4. \*.log
5. log/
   * \*.log → ignores all .log files in the repo.
   * log/ → ignores the log folder and its contents.

**Step 4 – Verify Ignoring**

1. **Check Git status**:
2. git status

You should **not** see debug.log or the log folder listed as untracked files.

1. If you had already tracked these files before adding .gitignore, remove them from tracking:
2. git rm --cached debug.log
3. git rm -r --cached log

**Step 5 – Commit Changes**

1. Stage .gitignore:
2. git add .gitignore
3. Commit:
4. git commit -m "Added .gitignore to ignore .log files and log folder"

**Step 6 – Push to Remote**

git push origin master