5. **Objectives**

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

In this hands-on lab, you will learn how to:

* Execute steps involving clean up and push back to remote Git.

**Prerequisites**

The following are the pre-requisites to complete this hands-on lab:

* Hands-on ID: **“Git-T03-HOL\_002”**

Notes\*:

|  |
| --- |
| Please follow the below steps for creating a free account in GitHub.  Do not use cognizant credentials to login to GitHub. |

Estimated time to complete this lab: **10 minutes.**

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.

Answer:

**How to clean up and push back to remote Git**

**1. Verify clean working state**

* Run git status to check if there are any uncommitted changes.
* If there are changes, either commit them using git add . and git commit -m "message", or discard them.
* This ensures your local branch is “clean” before pushing.

**2. Switch to the correct branch**

* Use git checkout <branch-name> to move to the branch you want to update (e.g., main or your feature branch).

**3. Pull latest changes from remote**

* Run git pull origin <branch-name> to sync your local branch with the latest version from the remote repository.
* This helps avoid conflicts when pushing.

**4. Push your branch to remote**

* If your branch has new commits, run:

git push origin <branch-name>

* This sends your local commits to the remote repository (GitHub, GitLab, etc.).

**5. Verify on remote**

* Open your repository in GitHub/GitLab.
* Check that your commits and files are visible on the correct branch



