Git Assignment

Asad Tariq

Git Branches

A branch in Git is simply a lightweight movable pointer to one of these commits.

This workflow consists of five types of branches, each with different roles:

- Master
- Feature branch
- Release branch
- Hotfix branch
- Develop branch

Master:

Upon making the first commit in a repository, Git will automatically create a master branch by default. Subsequent commits will go under the master branch until you decide to create and switch over to another branch.

Feature/Topic branch:

When you start working on a new feature/bug fix, you should create a feature/topic branch. A feature/topic branch is normally created off a develop/integration branch

Release:

Only bug fixes and release related issues should be addressed on this branch. Having this branch will allow other team members to continue pushing new features to the develop/integration branch without interrupting the release workflow.

Hotfix branch:

When you need to add an important fix to your production codebase quickly, you can create a Hotfix branch off the master branch.

By convention, hotfix branch names normally start with the prefix "hotfix-".

Develop branch:

When some changes need to be merged into the develop/integration branch, it is generally a good idea to create a feature/topic branch to work on independently.

Git Merge

If you have a multiple branches we can combine theses branched to master branch or default branch & this phenomenon of combining is known as Merge.

Check in which branch you in:

- Git Status

The Git status command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git.

For merging we use a command

(git merge BranchName)

The current local branch will be marked with an asterisk (*).

- If we want to see local branches we run this command:

  git branch

- To see remote branches we run this command:

  git branch -r

- To see all local and remote branches we use this command:

  git branch -a

Git PULL

The git pull command is used to fetch and download content from a remote repository and immediately update the local repository to match that content.

The git pull command first runs git fetch which downloads content from the specified remote repository. Then a git merge is executed to merge the remote content refs and heads into a new local merge commit.

➔ Git pull <remote>

Fetch the specified remote's copy of the current branch and immediately merge it into the local copy. This is the same as git fetch <remote> followed by git merge origin/<current-branch>

Restore the deleted files

The reflog command keeps a track of every single change made in the references (branches or hash id) of a repository and keeps a log history of the branches and tags that were either created locally or checked out. Reference logs such as the commit snapshot of when the branch was created or cloned, checked-out, renamed, or any commits made on the branch are maintained by git and listed by the reflog command. We can use these commands for both local & remote

After that, simply revert to the previous commit using:

git checkout <hash-id>