Lab 8 Git Branching

Coming soon https://www.cloudwithraghu.com

Keep Watching!

In this Document

What is Git Branching?

- By default, when we initialize a Git repository, Git creates a branch called "master" or “main”.
- This is the Primary branch of our repository.
- We create a separate branch for every issue, bug or a new feature.
- So as a feature is in progress, we commit our code to the branch for that feature.

Another example:

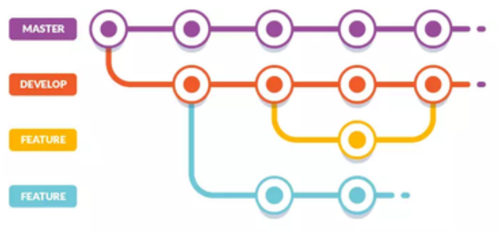
- Suddenly, we receive notification from our users that the login isn't working.
- We create a new branch to fix that issue.
- When the code for the "login issue" branch is complete, we merge the branch back into the main branch.
- Optionally, we can delete the merged branch.

Main branch always has fully worked code:

- We use branching to keep a working version of our code in the main branch, and partially completed code in feature or issue branches.
- That way our main branch will always have a working version of the code.

What happens when you create a branch?

- Usually we create another branch from the main branch.
- Since we created it from the main branch, the initial code for this new branch is a copy of the main branch code.



There is nothing special about the main branch.

It is the first branch made when you initialize a Git repository using the `git init` command.

How to create?

The four commonly used commands to create a Git branch are:

✓ **Simple create**

git branch <branchname>

This will create a new branch, but you still stay in the current branch where you are working

✓ **Create and checkout**

git checkout -b <branchname>

This will create a branch, and also switch to the new branch

✓ **Create using a TAG**

git branch <branchname> <tag>

Sometimes a developer wants to create a branch from a commit that has been tagged as milestone or release.

✓ **Create using a commit id**

git branch <branchname> <commit id>

A developer can create a new Git branch from a commit id.

Listing git branches

```
git branch -a
```

Simple create

```
git branch bugfix
git checkout bugfix
git status
```

Create and checkout

```
git checkout -b bugfix
git status
```

Create by tag

```
azureuser@master:~/project1$ git log --oneline
92e2522 (HEAD -> master) blog New code Commit
329e379 index Final Commit
19140b5 blog First Commit
1eb0305 login First Commit
1184df4 index First Commit

```

TAG a particular commit as Major Milestone reached.

In this case i want to Milestone `329e379 index Final Commit`

```
git tag -a [tag_name] HEAD -m "Tag message"
git tag -a IndexM1 329e379 -m "Index Milestoned as M1"
```

Verify

```
azureuser@master:~/project1$ git log --oneline
92e2522 (HEAD -> master) blog New code Commit
329e379 (tag: IndexM1) index Final Commit
19140b5 blog First Commit
1eb0305 login First Commit
1184df4 index First Commit

```

Create a branch based on Milestone

```
git branch <branchname> <tag>
git branch index-milestone IndexM1
```

Create Branch by commit id

```
git branch <branchname> <commit id>

git branch index-feature1 1184df4
```

Find the files in the above branch. You should only see index.html

```
git ls-files
```

Branch Naming Strategies

Branch names can be anything you’d like.

For example, naming the branch based on the person responsible for working on the branch and a description or work item:

- username/description
- username/workitem

You can name a branch to indicate the branch’s function, like a feature, bug fix, or hotfix:

- bugfix/description
- feature/feature-name
- hotfix/description

```
git branch raghav/feature1
git branch raghav/feature2
git branch raghav/bugfix1
git branch nelson/hotfix1
git branch srikanth/learner
git branch hotfix/windows11-jan2024
git branch feature/singlesignon
git branch fix/authentication
```

Deleting a branch

- Git will not let you delete the branch you are currently on.
- So you must make sure to checkout a branch that you are NOT deleting.

For example: `git checkout master`

```
git branch -d bugfix
```

Rename a branch

Go to the branch you want to rename

```
git checkout raghav/feature2
git branch -m santosh/feature2
```

Another way

Go to any other branch

```
git checkout master
git branch -m old-branch new-branch
git branch -m santosh/feature2 sachin/feature2

```