SCM - Source control Management

By CODEMIND Technology

Contact us - 966 50 44 698
966 50 44 598

GIT

- GitHub Account Creation
- Desktop GitHub Installation
- Version control
- Repository
- SCM Tools
- Commit, Pull, Push
- Stashing
- Merge Conflicts
- Branch

What is Version control

- A way to manage files and directories
- Track changes over time
- Recall previous version

Why GIT?

- Over 70% of developers use Git!
- Developers can work together from anywhere in the world.
- Developers can see the full history of the project.
- Developers can revert to earlier versions of a project.

Features of GIT

- When a file is changed, added or deleted, it is considered modified
- You select the modified files you want to Stage
- The Staged files are Committed, which prompts Git to store a permanent snapshot of the files
- Git allows you to see the full history of every commit.
- You can revert back to any previous commit.
- Git does not store a separate copy of every file in every commit, but keeps track of changes made in each commit!

SCM - Tools

GitHub: GitHub is an online service for managing your repositories.

GitHub is the largest host of source code in the world, and has been owned by Microsoft since 2018

Desktop GitHub is an application that enables you to interact with GitHub using a GUI instead of the command line or a web browser

GIT: Git is an open source tool developers install locally to manage source code

What is repository

- Repository used to organize a single project
- Repositories can contains files, folder, images, videos, anything that your project needs

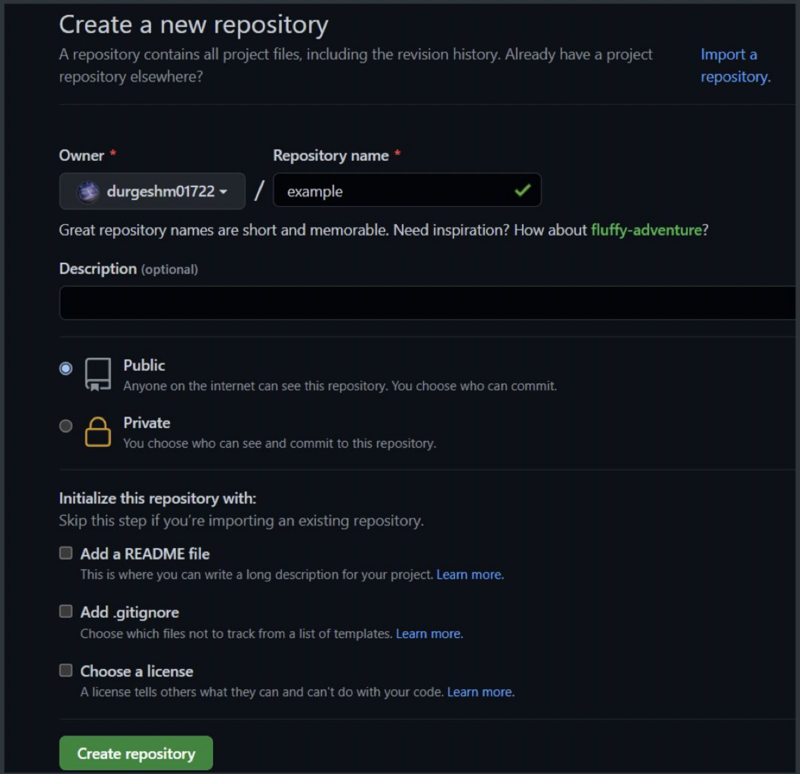
We can create a repository using GITHub but make sure to create GitHub account first and then try to create repository as shown in screen shot

Link → https://github.com/ to create account

Click on Sign up button and continue to create account pls make sure to note down your GitHub account credentials

Working with Github :-

Create a github account to create your remote repositories. Now, create a new repo where we will be uploading our files from local repo.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * durgeshm01722 / Repository name * example

Great repository names are short and memorable. Need inspiration? How about fluffy-adventure?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

□ Add a README file
This is where you can write a long description for your project. Learn more.

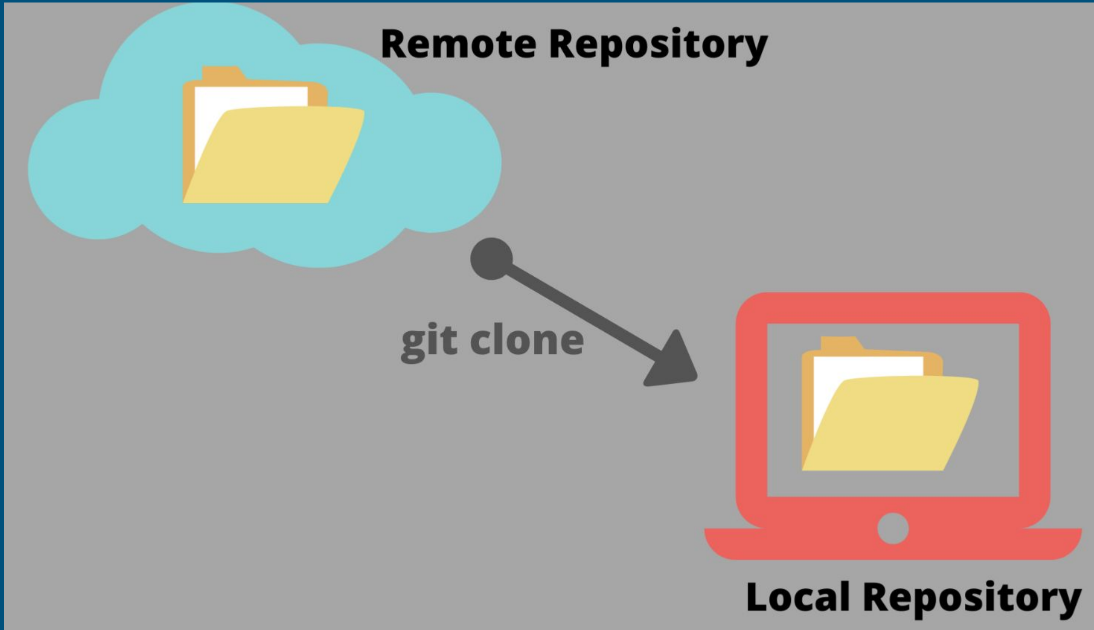
□ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

□ Choose a license
A license tells others what they can and can't do with your code. Learn more.

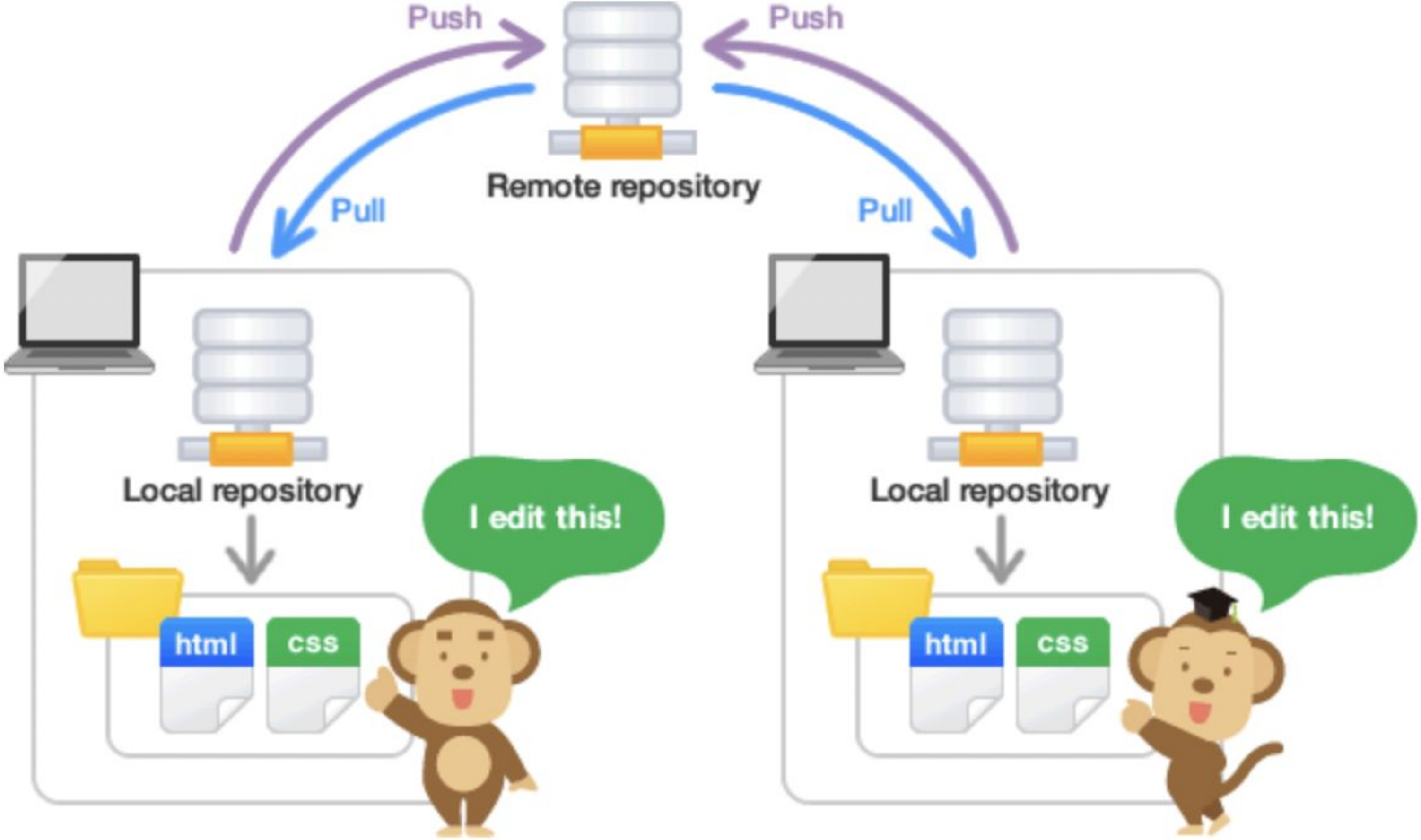
Note - Local repository (repo.) means the repo. which is on our system whereas, remote repo. means the one which is on other remote system/server, for eg. - GitHub, GitLab, Bitbucket, etc.

What is Checkout / cloning and local repository?

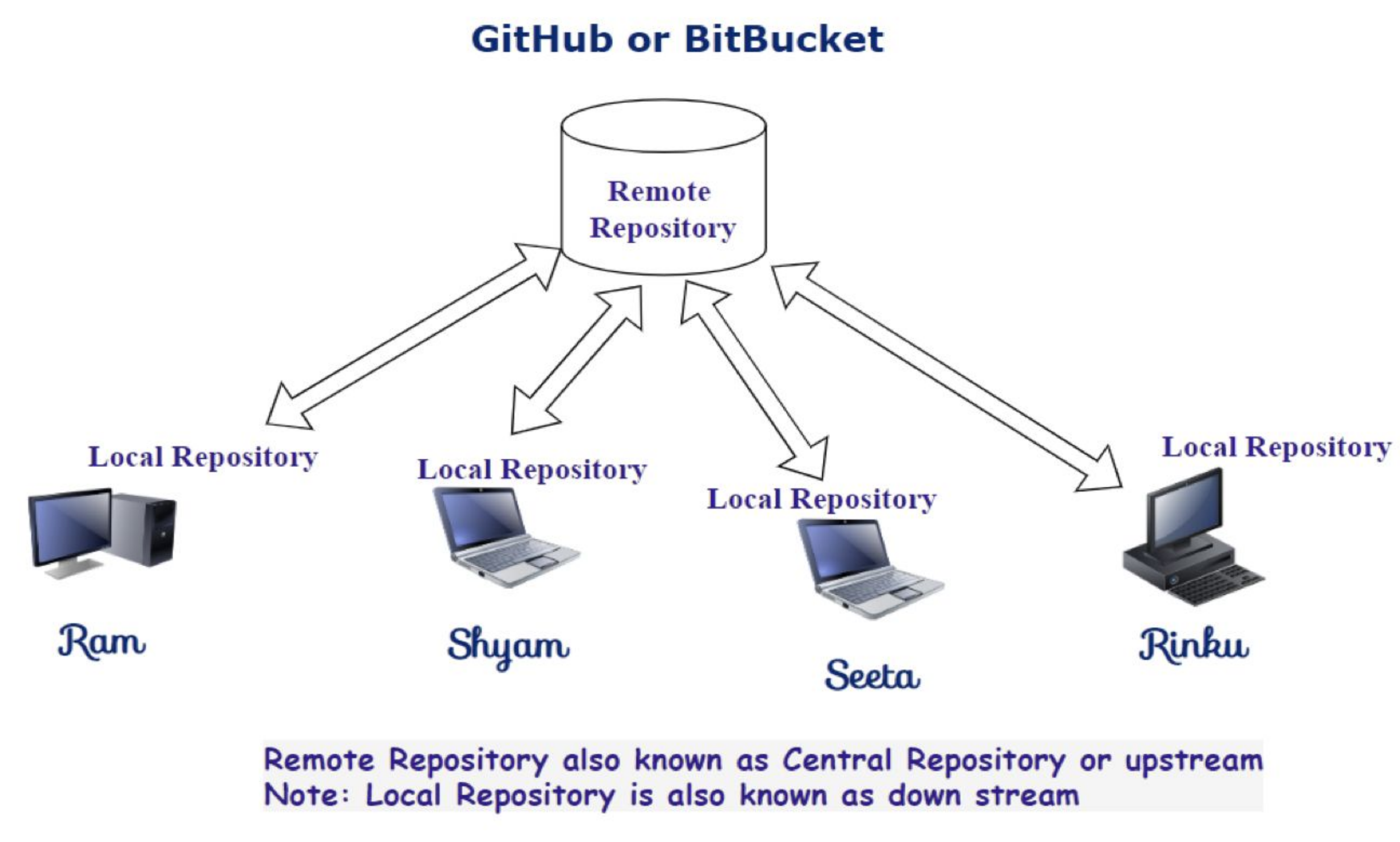
- **Cloning:** Cloning is the process of creating a new copy of an existing repository
- Local Repository
- Remote Repository



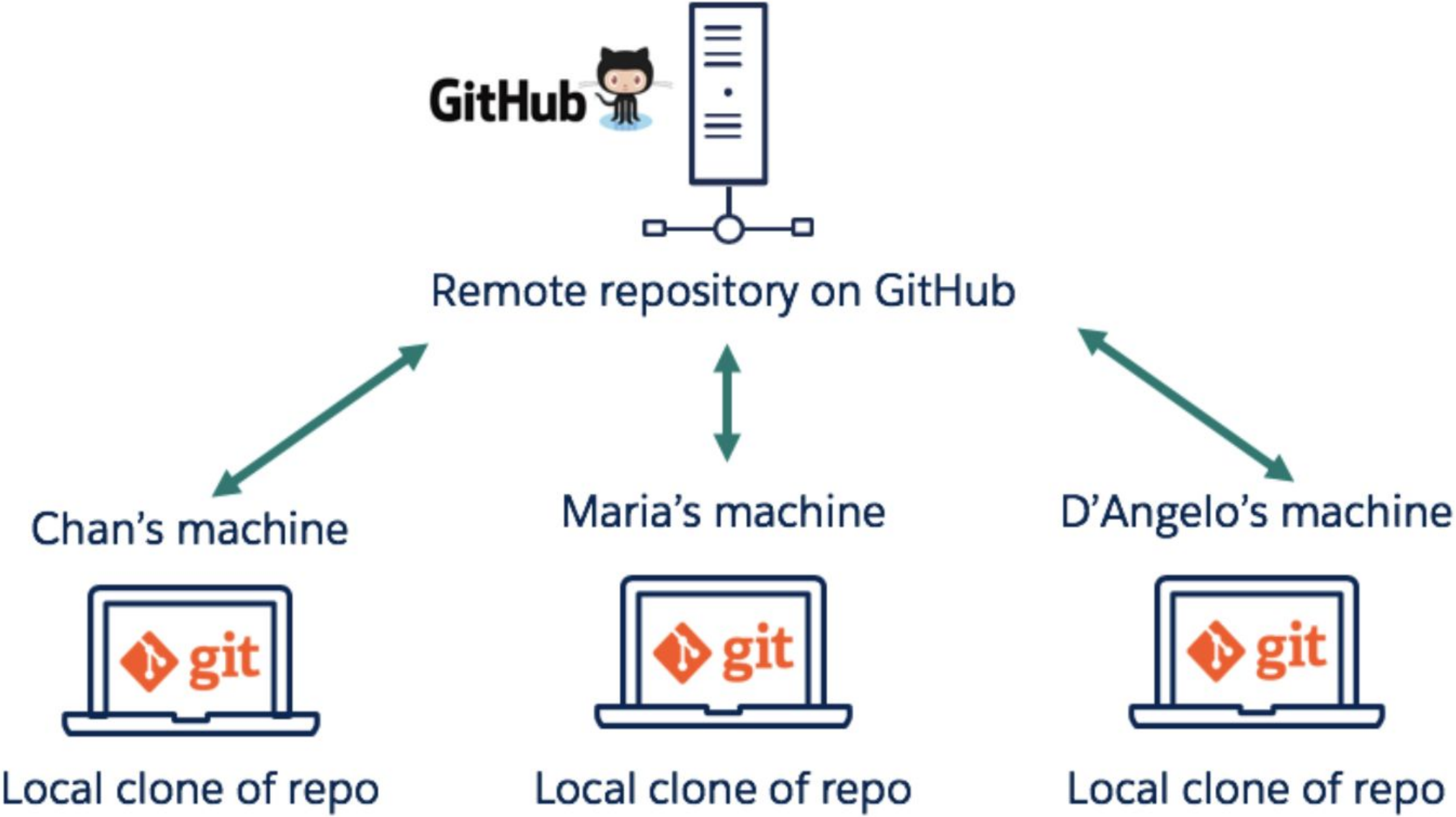
Commit + Push and Pull



Distributed version control system: Means a copy of the project files will be maintained at the client’s system along with the copy being maintained at the remote server



Remote Repository also known as Central Repository or upstream
Note: Local Repository is also known as down stream



Remote repository on GitHub

Chan’s machine

Maria’s machine

D’Angelo’s machine

Local clone of repo

Local clone of repo

Local clone of repo

What is branch in GIT?

In Git, a branch is a new/separate version of the main repository. Branches allow you to work on different parts of a project without impacting the main branch. When the work is complete, a branch can be merged with the main project.

We can even switch between branches and work on different projects without them interfering with each other.

Branch creation in GIT ?

- Create branch in local repository
- Create branch in remote repository

Commit, Pull and Push

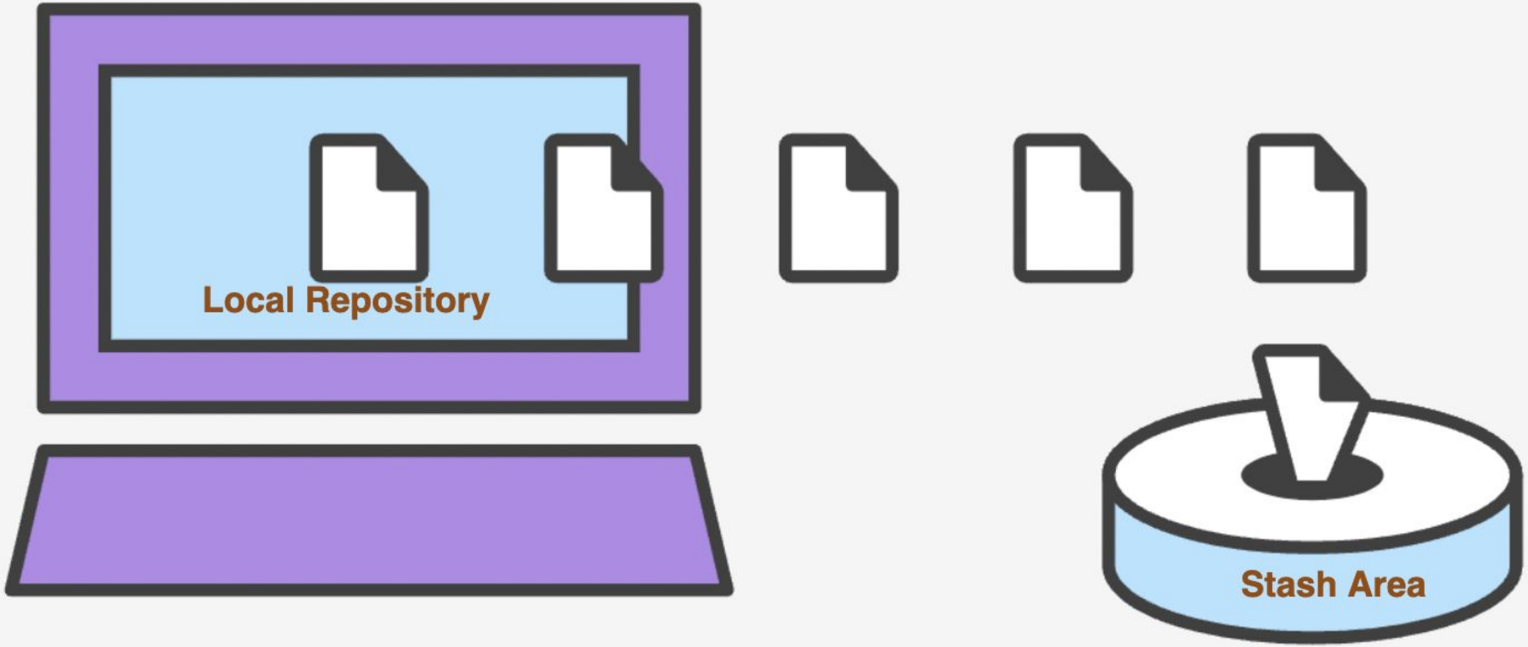
Commit: git commit puts your changes into your local repository and to send these committed changes from local repository to remote repository ‘push’ is used.

Pull: Pull is used to update the local version of a repository from a remote.

Push: Push is used to upload local repository content to a remote repository.

Stashing in GIT

The git stash command takes your uncommitted changes, saves them away for later use, and then reverts them from your working copy



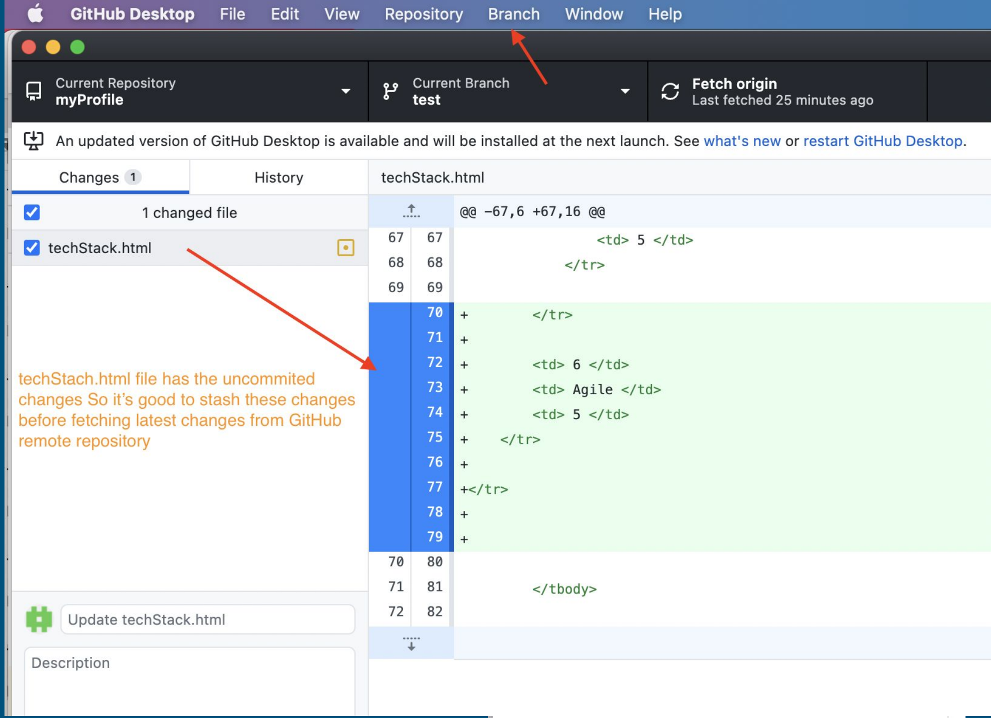
How to Stash uncommitted changes?

As shown in snippet techStach.html has the Uncommitted changed so before any ‘Fetch Origin’ operation we can stash all these changes using

Step 1. Branch → Stash All changes

Step 2. Click on ‘Fetch origin’ and then pull

You will get the latest code



Current Repository
myProfile

Current Branch
test

Fetch origin
Last fetched 25 minutes ago

An updated version of GitHub Desktop is available and will be installed at the next launch. See what's new or restart GitHub Desktop.

Changes 1 | History

☑ 1 changed file

☑ techStack.html

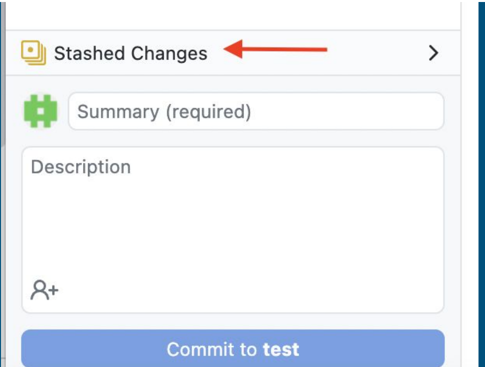
techStach.html file has the uncommited changes So it’s good to stash these changes before fetching latest changes from GitHub remote repository

techStack.html

```
@@ -67,6 +67,16 @@
                    <td> 5 </td>
                </tr>

+        </tr>
+
+        <td> 6 </td>
+        <td> Agile </td>
+        <td> 5 </td>
+    </tr>
+
+</tr>
+
+

            </tbody>

```

Update techStack.html

Description



Step 3. Restore the stashed changed as shown in this snippet

Step 4. In case any conflicts resolve it and commit + push the changes

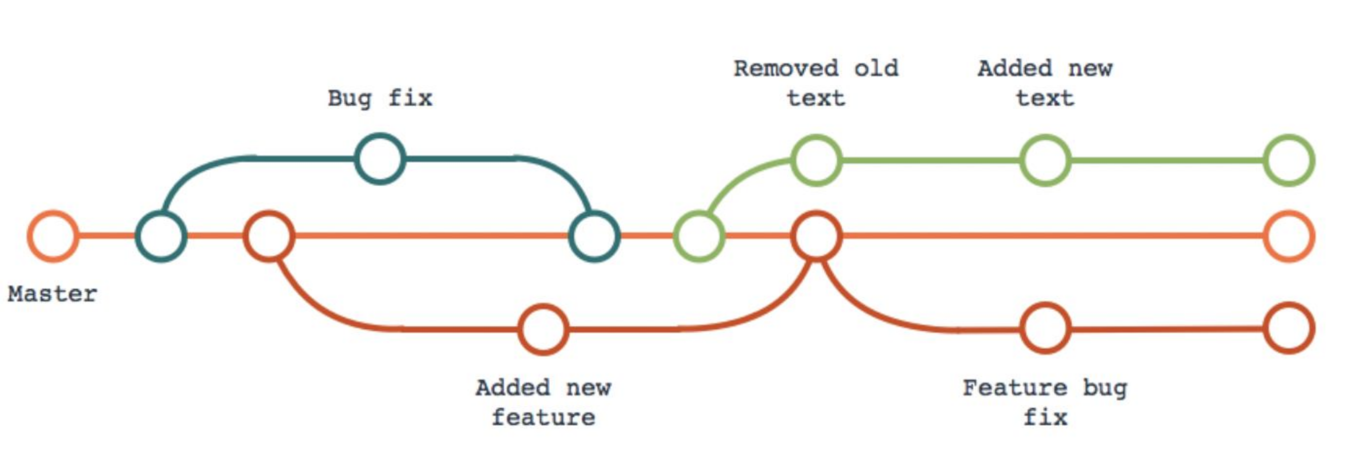
Before Commit It’s always good to perform steps as

- In case any uncommitted changes in the local repository please stash it first
- Update the local repository using the option ‘Fetch Origin’ and pull in Desktop GitHub
- Resolve the conflicts if any
- Commit the changes with meaningful commit message
- Push the local repository changes into remote repository

What is all these ?

- Uncommitted changes
- Discard changes ?
- Revert Code ?
- Clean working tree ?
- Clean working directory ?

GIT Branching

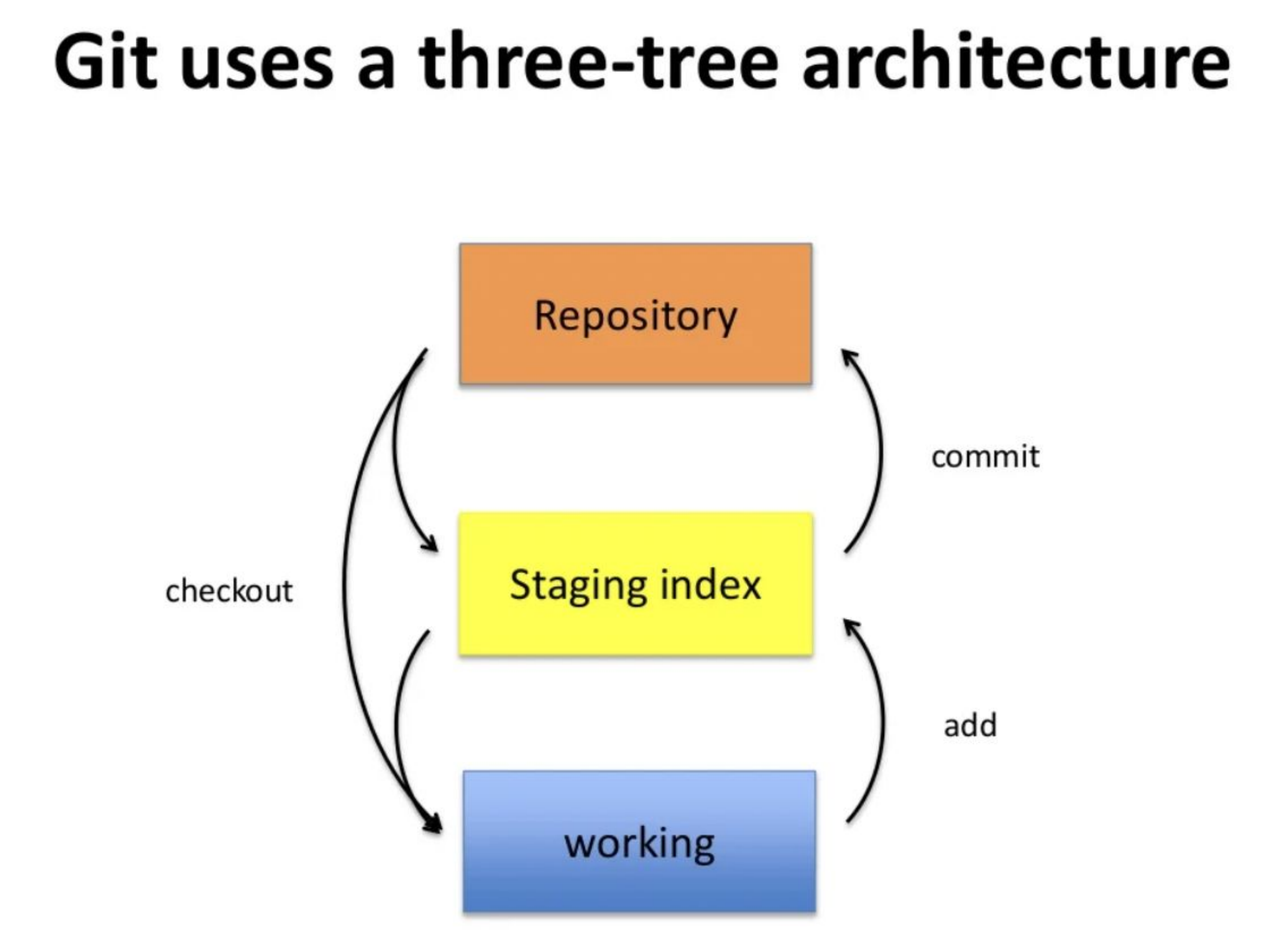


What is ?

Checkout

Cloning

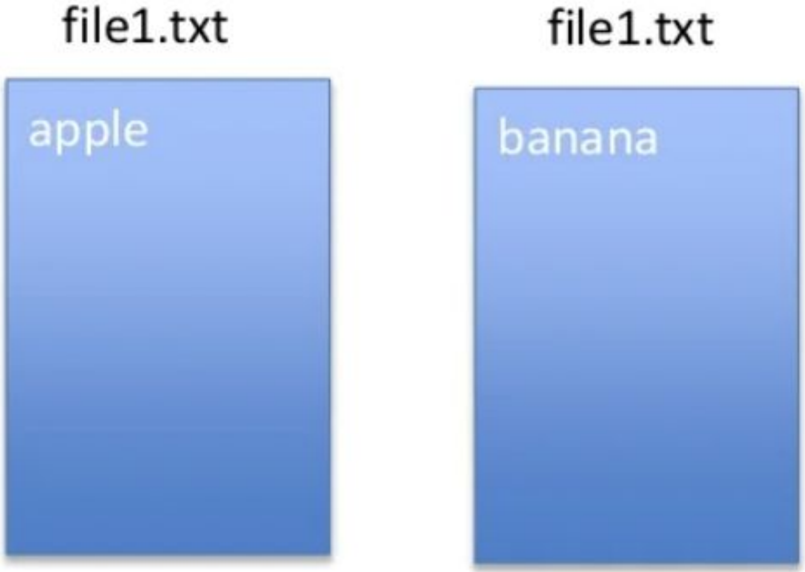
Working



Git uses a three-tree architecture

merge conflicts

What if there are two changes to same line in two different commits?



file1.txt

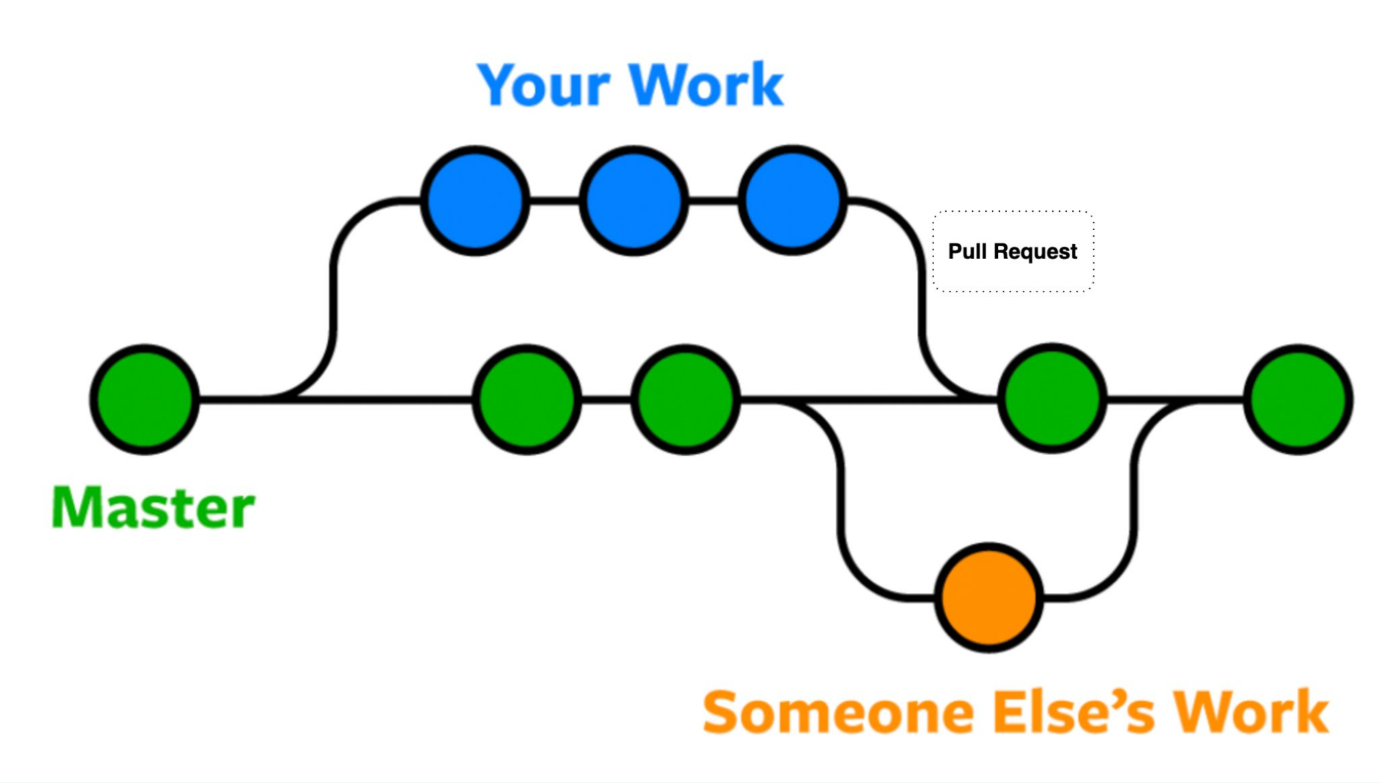
apple

file1.txt

banana

Branch

A branch is a version of the repository that diverges from the main working project.



Thank you