Introduction to GITHUB

Niloofar Mozafari

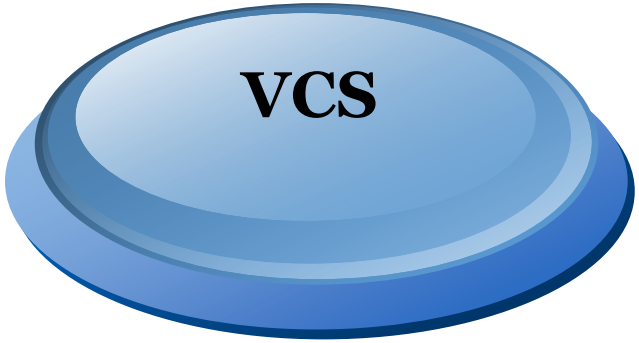
Contents

Version Control System

- Version control systems (VCS) are a category of software tools that allow the team to manage the changes that are applied to files by keeping a track of modifications over time.

- A VCS works by creating a repository to store the file and its history of changes. People can make changes to their local copy of the file and then "commit" those changes to the repository, where they can be reviewed and merged with other changes made by others.

Version Control System



Centralized
Data is stored on a central server (Client/Server). In order to make changes in the project, it needs to connect to the central server.

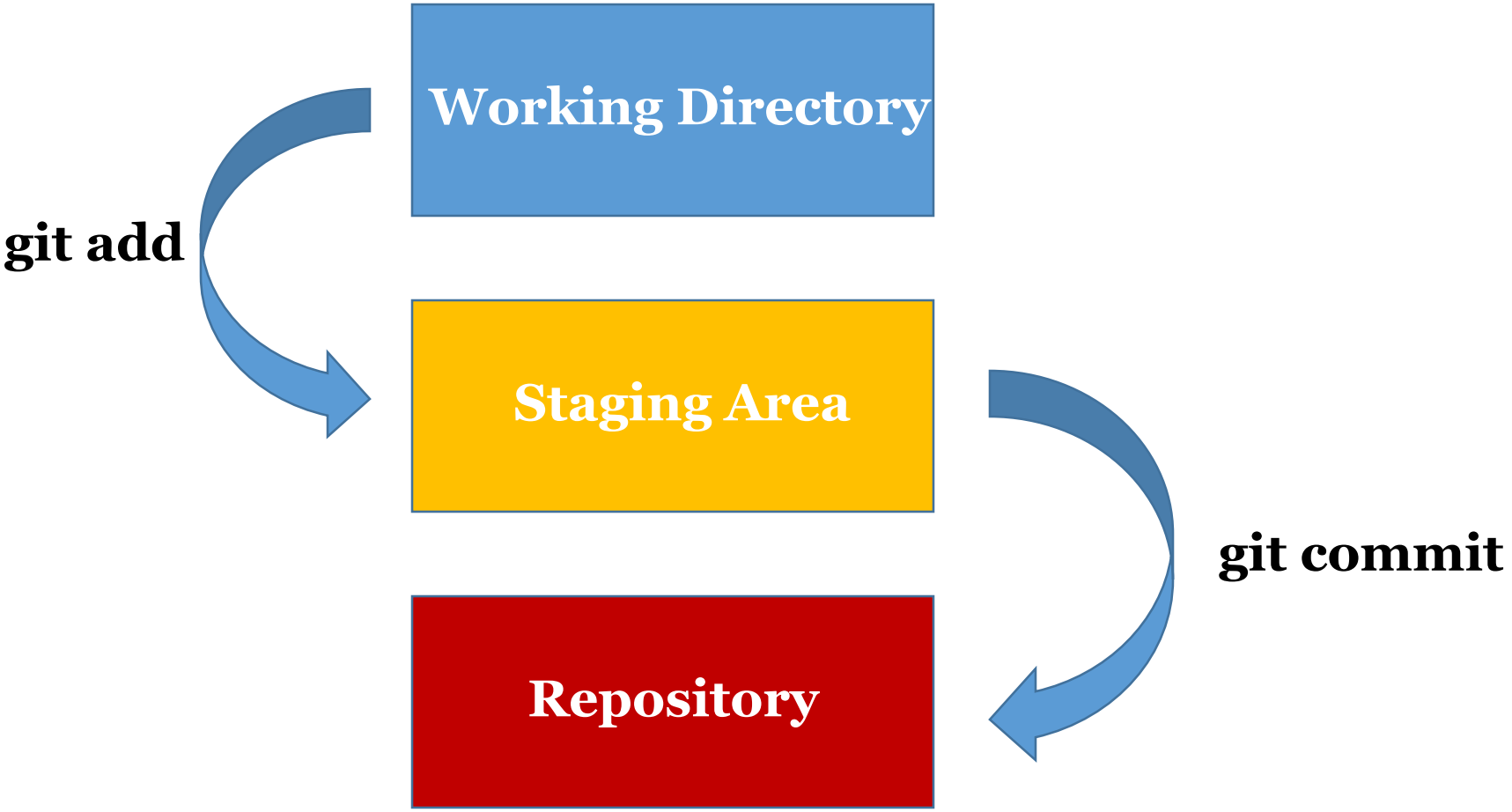
Distributed
The files are available to every member of team and they can make the wanted changes in the project.

GIT

- Git is a distributed version control system which was created by Linus Torvalds in 2005 to manage the development of the Linux kernel.

- Git allows users to track changes to their file over time, collaborate with others on the same codebase, and maintain multiple versions of their file simultaneously.

Repository


Working Directory
git add
Staging Area
git commit
Repository

GIT Basic Commands

Command	Description	Usage
git init	Initialize a Git repository for our local project folder.	• git init • git init [repository name]
git config	Set the author name and email address respectively to be used with your commits.	• git config --global user.name “[name]” • git config --global user.email “[email address]”
git add	Add the specified files into the Git repository, the staging area	• git add [file names] • git add .
git commit	Records files permanently in the version history.	• git commit –m “message”
git commit --amend	Modify the most recent git commit.	• git commit --amend -m "New commit message"

GIT Basic Commands (Con.)

Command	Description	Usage
git status	Show the status of all files.	• git status
git diff	Shows the difference between files.	• git diff • git diff –staged • git diff [first branch] [second branch]
git log	List the version history for the current branch.	• git log
git show	Show the metadata and content changes of the specified commit.	• git show [commit]

Git diff

Git diff

- Compare staging area and working directory

Git diff –staged

- Compare last commit and staging area

Git diff head

- Compare last commit and working directory

Git diff commit1 commit2

- Compare two commits

Git diff branch1 branch2

- Compare two branches

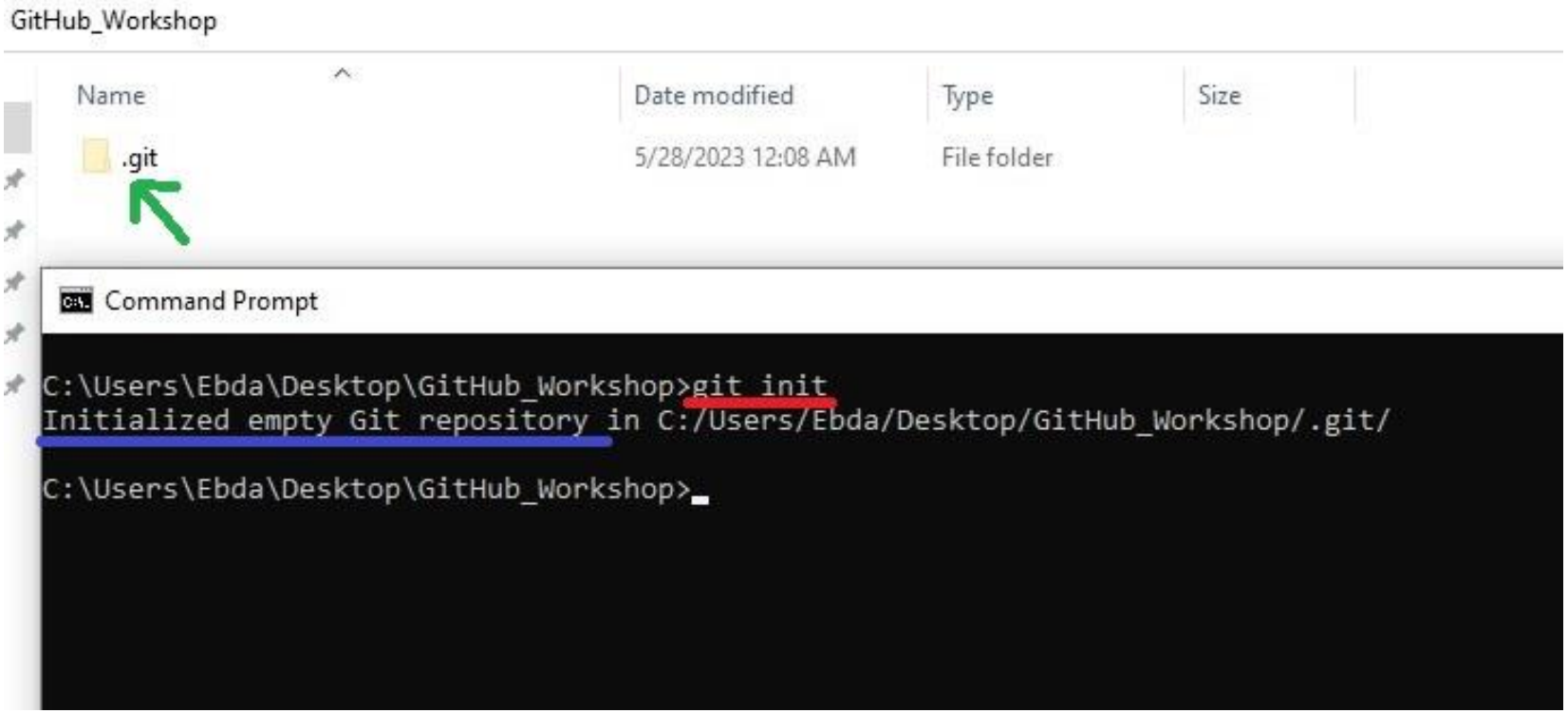
GIT rm

Command	Description	Usage
git rm	Remove a single file from the repository and the working directory.	• git rm [file]
git rm -r	Remove an entire directory and all its contents from the repository and the working directory.	• git rm -r <directory>
git rm --cached	Remove a file from the repository, but leaves it in the working directory. This is useful if you want to remove a file from version control but keep it in your local working directory.	• git rm --cached <file>

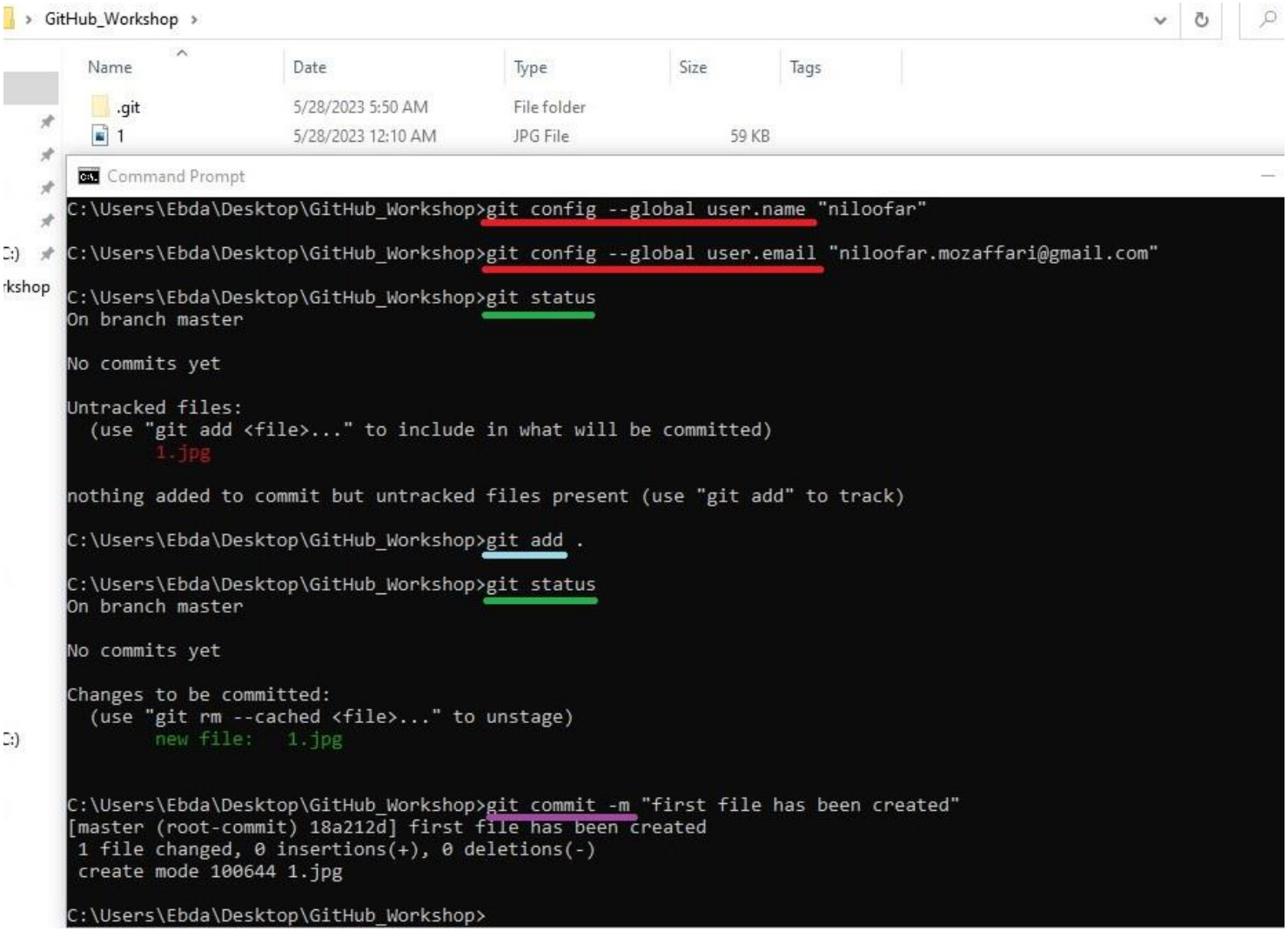
GIT Commands

Command	Description	Usage
git branch	List all the local branches in the current repository.	• git branch
git branch	Create a new branch.	• git branch [branch name]
git branch -d	Delete the mentioned feature.	• git branch -d [branch name]
git checkout	Switch from one branch to another.	• git checkout [branch name] • git checkout [commit ID]
git checkout -b	Create a new branch and also switch to it.	• git checkout -b [branch name]
git merge	Merge the specified branch's history into the current branch.	• git merge [branch name]

Examples


GitHub_Workshop
Name
Date modified
Type
Size
.git
5/28/2023 12:08 AM
File folder
Command Prompt
C:\Users\Ebda\Desktop\GitHub_Workshop>git init
Initialized empty Git repository in C:/Users/Ebda/Desktop/GitHub_Workshop/.git/
C:\Users\Ebda\Desktop\GitHub_Workshop>

Examples


GitHub_Workshop
Name
Date
Type
Size
Tags
.git
5/28/2023 5:50 AM
File folder
1
5/28/2023 12:10 AM
JPG File
59 KB
Command Prompt
C:\Users\Ebda\Desktop\GitHub_Workshop>git config --global user.name "niloofar"
C:\Users\Ebda\Desktop\GitHub_Workshop>git config --global user.email "niloofar.mozaffari@gmail.com"
C:\Users\Ebda\Desktop\GitHub_Workshop>git status
On branch master
No commits yet
Untracked files:
(use "git add <file>..." to include in what will be committed)
1.jpg
nothing added to commit but untracked files present (use "git add" to track)
C:\Users\Ebda\Desktop\GitHub_Workshop>git add .
C:\Users\Ebda\Desktop\GitHub_Workshop>git status
On branch master
No commits yet
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
new file: 1.jpg
C:\Users\Ebda\Desktop\GitHub_Workshop>git commit -m "first file has been created"
[master (root-commit) 18a212d] first file has been created
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.jpg
C:\Users\Ebda\Desktop\GitHub_Workshop>

Examples

Command Prompt

```
C:\Users\Ebda\Desktop\GitHub_Workshop>git checkout -b br_2
Switched to a new branch 'br_2'

C:\Users\Ebda\Desktop\GitHub_Workshop>git status
On branch br_2
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        2.jpg

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\Ebda\Desktop\GitHub_Workshop>git add .

C:\Users\Ebda\Desktop\GitHub_Workshop>git commit -m "second file on new branch named as br_2"
[br_2 1388734] second file on new branch named as br_2
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 2.jpg

C:\Users\Ebda\Desktop\GitHub_Workshop>git log
commit 1388734fa1c86591526e4f3d397dac6b0986428c (HEAD -> br_2)
Author: niloofar <niloofar.mozaffari@gmail.com>
Date:   Sun May 28 05:59:49 2023 -0700

    second file on new branch named as br_2

commit 18a212d57c29e18d5e8deff6757d6bc586c20926 (master)
Author: niloofar <niloofar.mozaffari@gmail.com>
Date:   Sun May 28 05:53:23 2023 -0700

    first file has been created

C:\Users\Ebda\Desktop\GitHub_Workshop>
```

Create a new branch and switch to it.

add files to the staging area

Create a commit

List the current history

GITHUB

- GitHub is a web-based platform that provides hosting for Git repositories. It was launched in 2008 and has since become one of the most popular platforms for hosting and collaborating on software development projects.

- GitHub offers many features that make it easy for developers to collaborate on projects.

GITHUB Features

Version control	• It provides a platform for hosting Git repositories, which allows developers to track changes to their code over time and collaborate with others on the same codebase.
Issue tracking	• GitHub provides a system for tracking bugs, feature requests, and other issues related to a project. This makes it easy for developers to track the progress of their work and communicate with others about issues that need to be resolved.
Collaboration tools	• GitHub provides tools for collaborating on code, including the ability to review and merge changes made by other developers, and to discuss code changes in a dedicated forum.
Continuous integration and deployment	• GitHub provides integration with many popular tools for continuous integration and deployment, which allows developers to automatically test and deploy their code as it is changed.

GIT remote

Go to https://github.com/

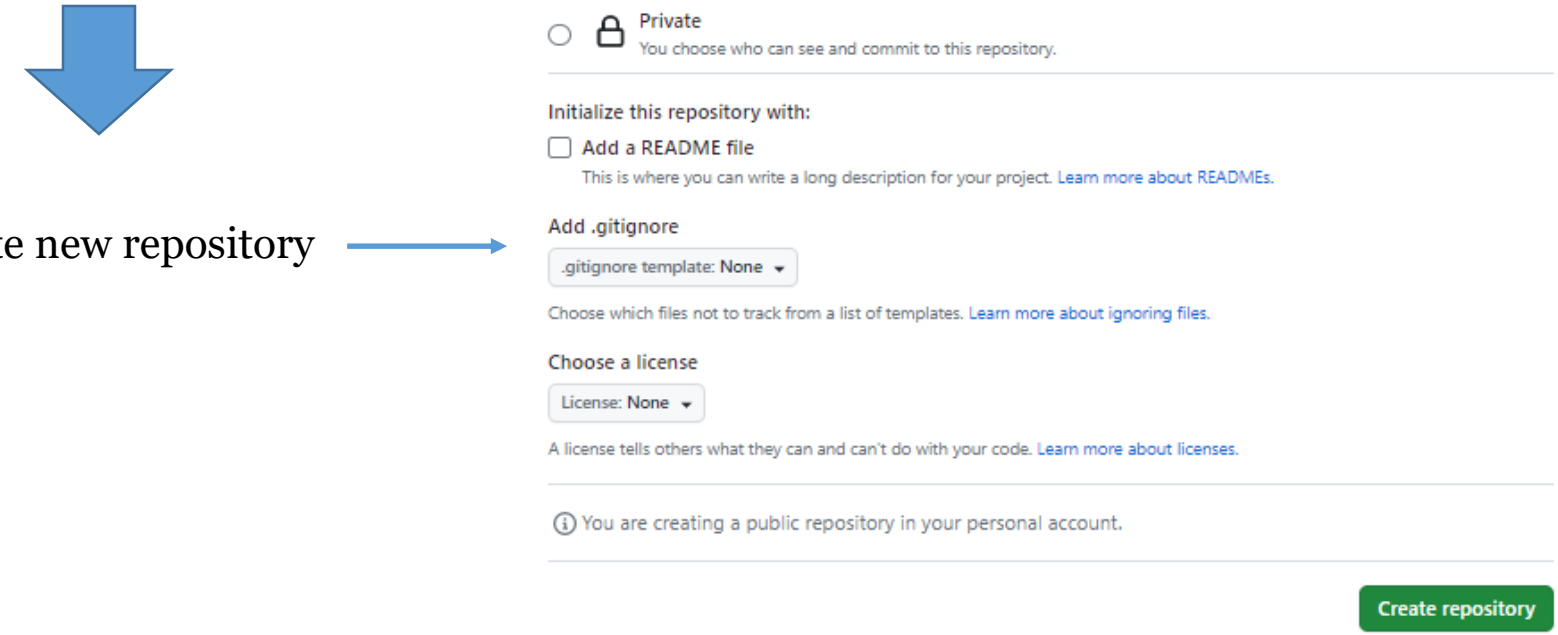
if you already have the account:

Login to account

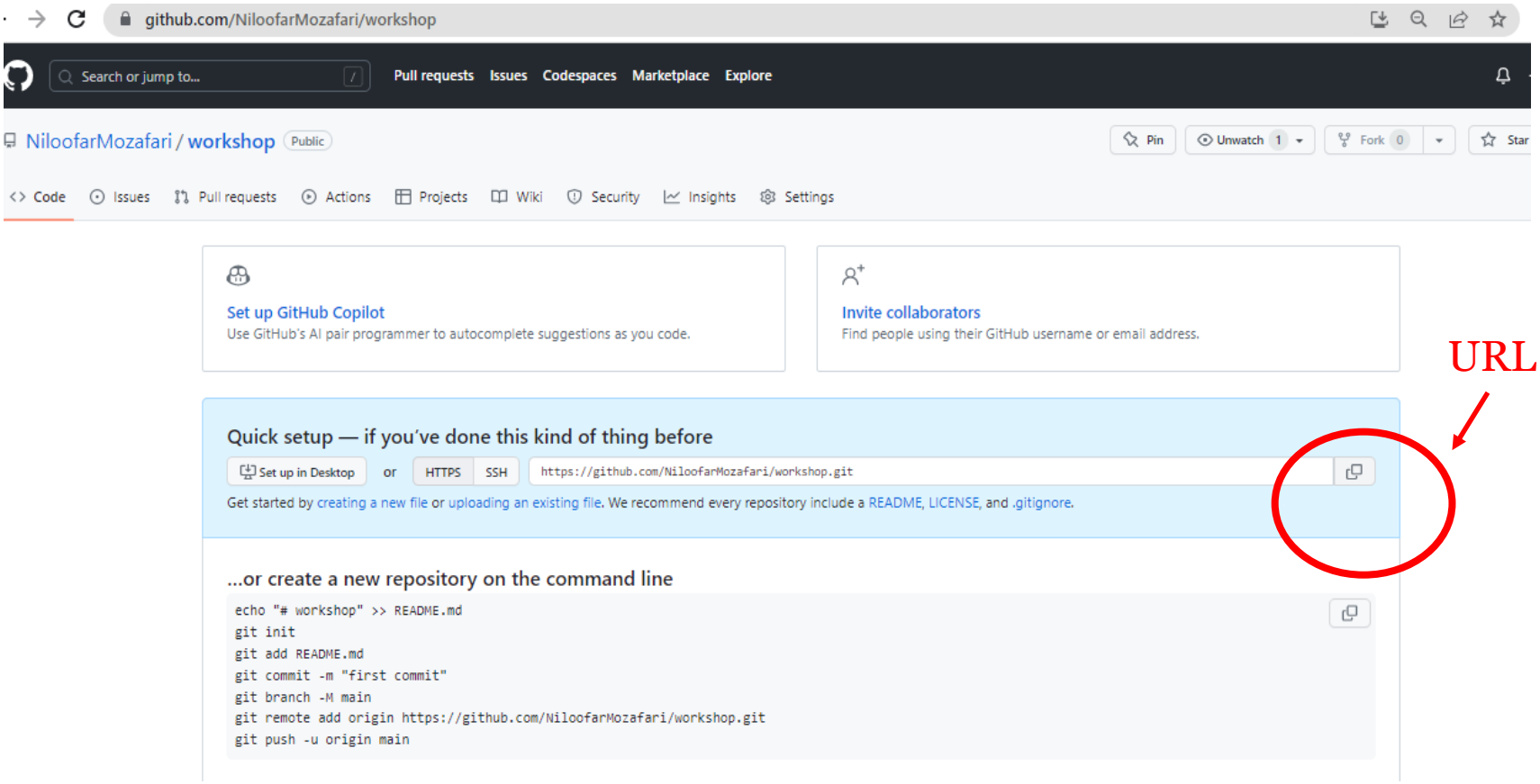
If not:

Sign up

Create new repository



GIT remote



git remote add origin “URL”

PUSH

- git push
- Usage: git push origin [branch name]
- If you want to push the changes in local repository to your remote repository on a particular branch, ‘git push’ helps to sync the local repository’s files with the remote repository on Github.

CLONE

- git clone
- Usage: git clone [URL]
- If you want to work on a file that is on a remote Github repository as another developer, “git clone” command helps you to import the files of project from the remote repository to your local system.

Readme.md

- In GitHub, an MD file is a file with the extension ".md" that uses Markdown syntax. Markdown is a lightweight markup language that allows you to write formatted text using a plain text editor.

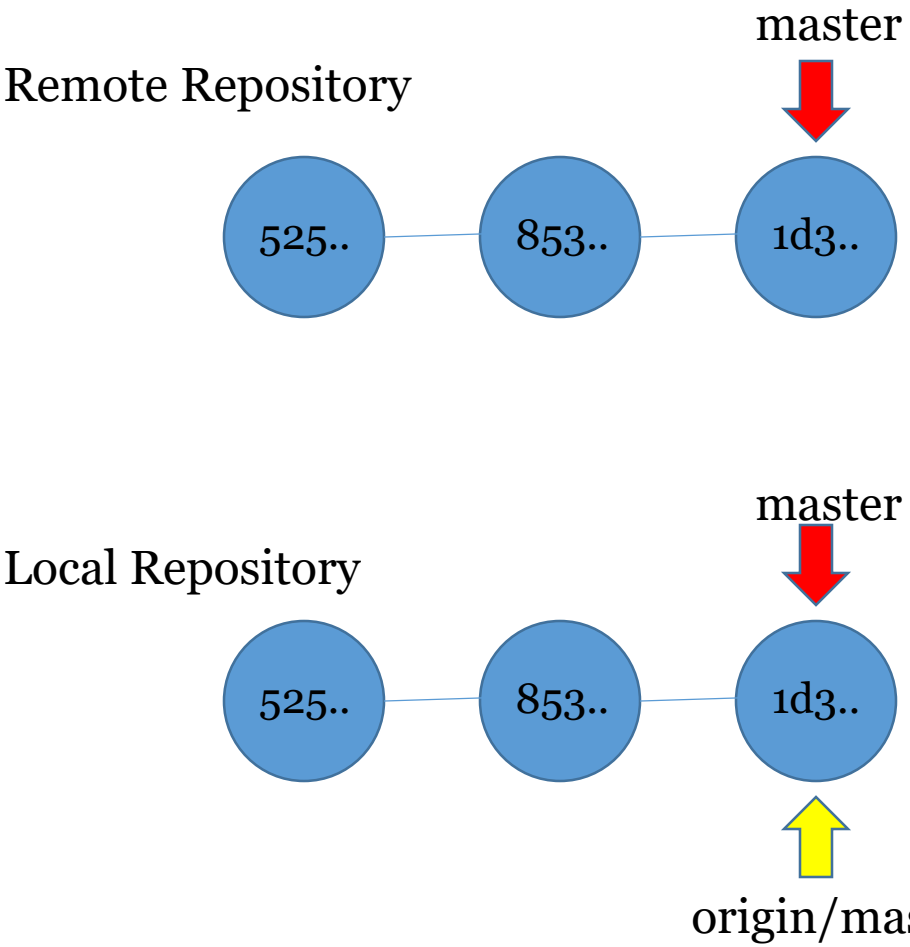
- MD files are commonly used in GitHub repositories to create documentation, README files, and other types of content. Markdown syntax allows you to format text using simple symbols and characters.

GitHub Pages

- GitHub Pages is a feature of GitHub that allows you to create and host websites directly from your GitHub repository. It is often used by developers to showcase their projects, host documentation, or create personal blogs. It's a convenient way to create a website without the need for a separate hosting provider or web server.

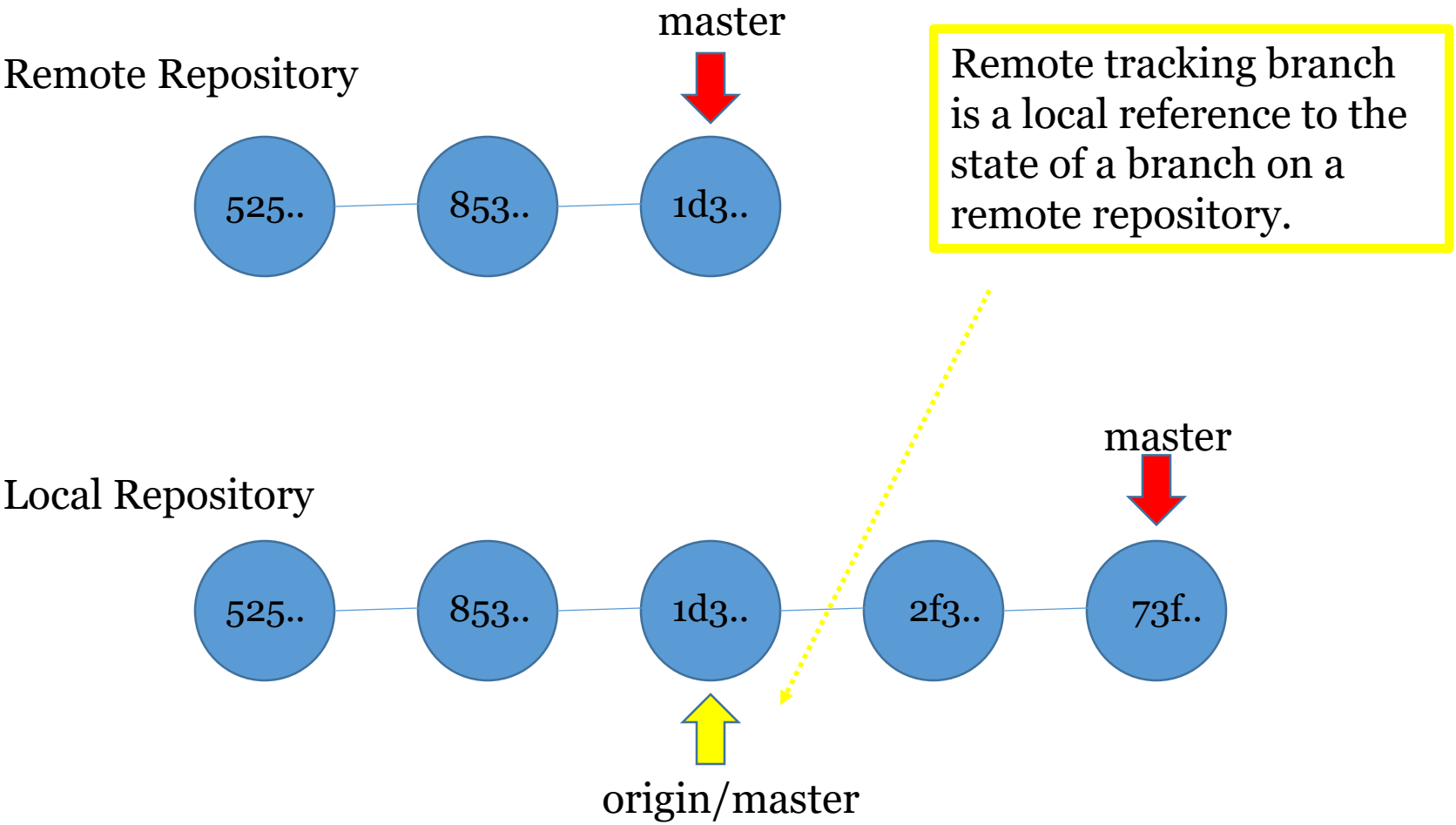
- When you create a GitHub Pages site, a new branch is created in your repository called "gh-pages". This branch contains the files and directories that make up your website. You can then configure your GitHub Pages site to use a custom domain or subdomain name, and choose whether your site should be public or private.

Remote Tracking Branch

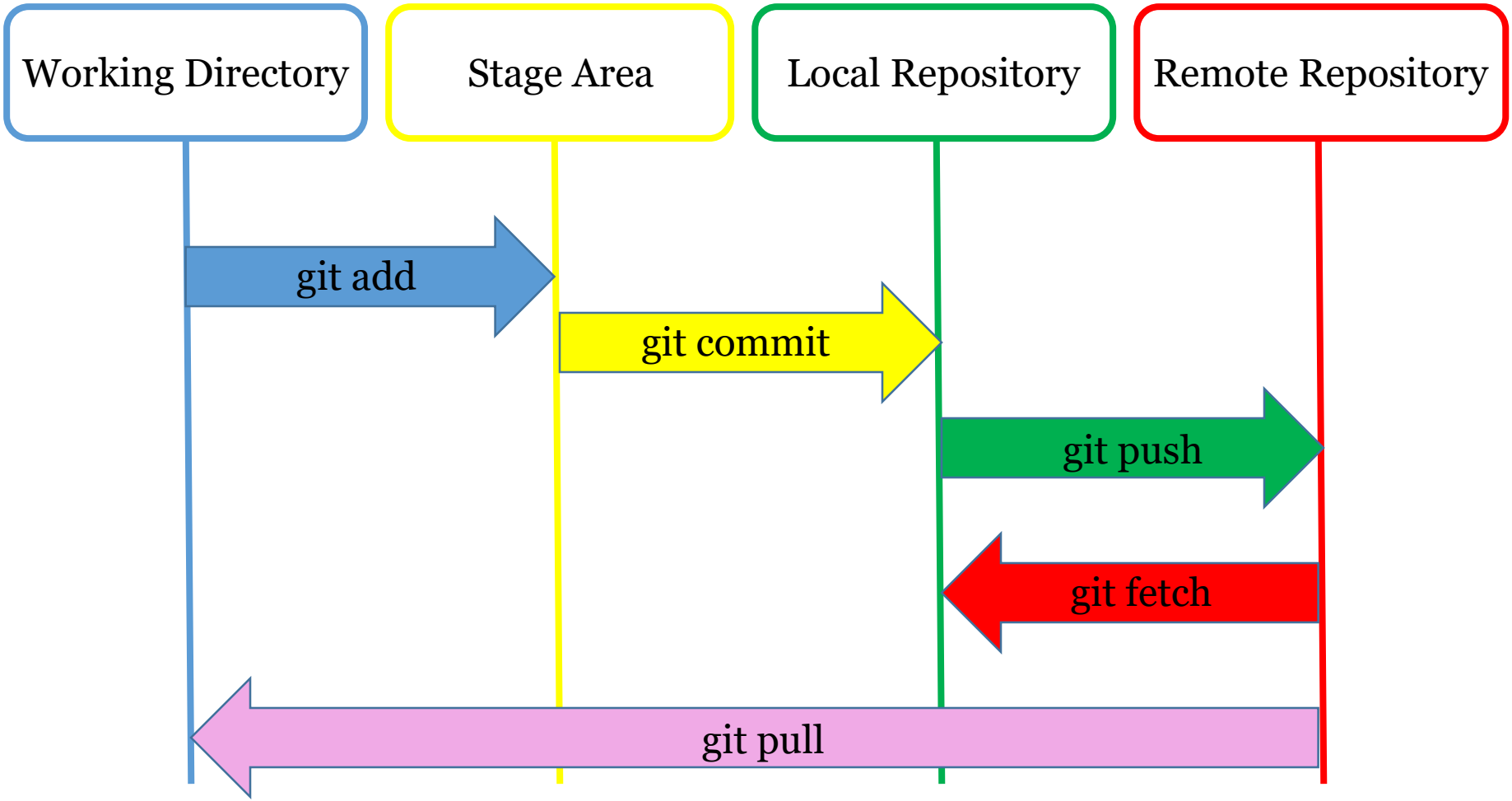


Remote tracking branch is a local reference to the state of a branch on a remote repository.

Remote Tracking Branch



git fetch & git pull



Git fetch <remote> <branch>

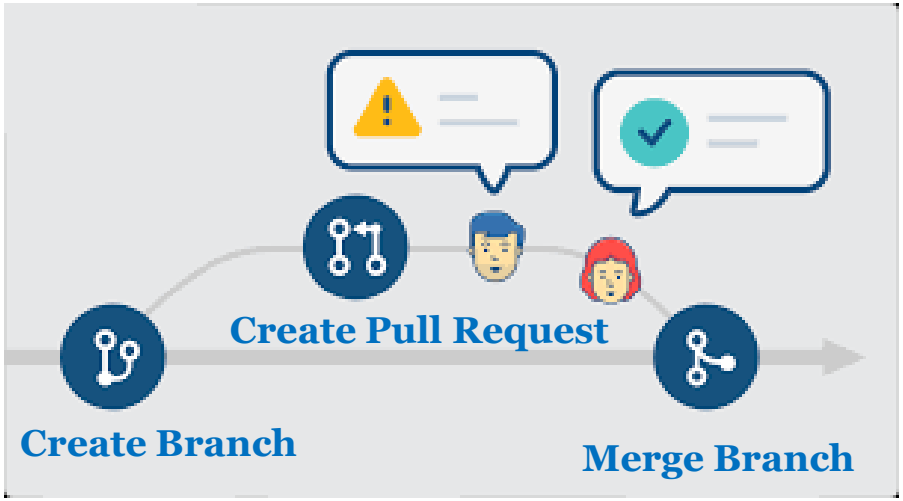
- git fetch is a Git command that allows you to retrieve changes from a remote repository without merging them into your local branch. This command allows you to keep your local repository up-to-date with changes made to the remote repository without affecting your local code.
- When you run git fetch, Git retrieves all the changes made to the remote repository since your last interaction with it and stores them in your local repository. However, it does not merge these changes into your local branch.

Git pull

- git pull is a Git command that is used to update a local repository with changes from a remote repository. It is a combination of two Git commands, git fetch and git merge. When you run git pull, Git will first fetch all the changes made to the remote repository since your last interaction with it and then merge those changes into your local branch.

Pull request

- A pull request in GitHub is a feature that allows users to propose changes to a repository hosted on GitHub. It is a way for developers to collaborate and contribute to open-source projects, or to propose changes to code within a team or organization.



Git fork

- Git fork is a powerful feature of GitHub that makes it easy for developers to collaborate and contribute to open-source projects.
- It copies a repository to your own repository and allows you to freely experiment with changes without affecting the original project. When you "fork" a repository, you create a separate copy of it in your own GitHub account, which you can then modify as you wish. The original repository remains untouched, and you can make changes to your forked repository without affecting the original project.

Thank You !