# What is GIT?

* Git is a version control system that is used for tracking changes to files.

● It does this through a series of **snapshots of your project.** It works with those

snapshots to version and manage your source code, and it does this in a simple way

# Why Use Git?

● Can work offline.

● Collaborating with others is easy! ● Branching is easy!

● Branching is fast!

● Merging is easy!

● Git is fast.

● Git is flexible.

# GIT vs Github vs Gitlab

● Git is a version control system, while GitHub and GitLab are web-based Git

repositories.

● GitHub is a company that provides Git hosting, and it offers both a cloud-based

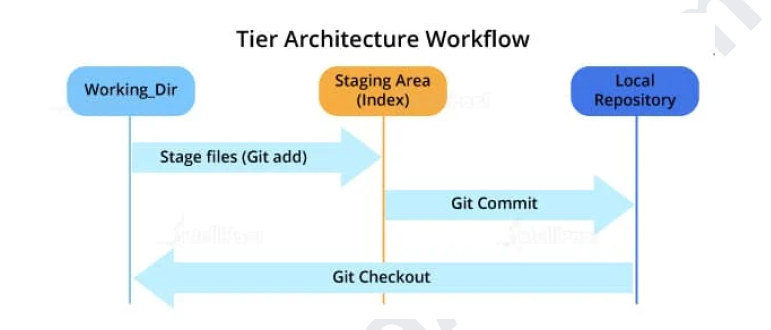
hosting service and on-premises enterprise versions.

* GitLab is similar to GitHub, but it is an open-source Git hosting platform. It provides a

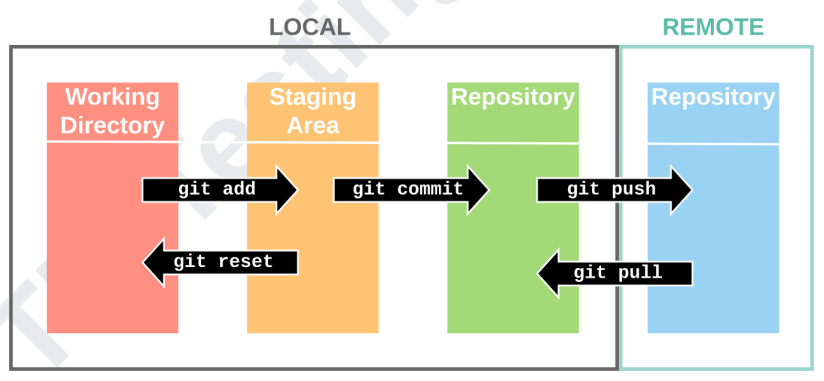
web-based interface for working with Git repositories, as well as a range of tools for collaboration, project management, and continuous integration.

Git is a version control system, GitHub is a Git hosting service, and GitLab is an open-source Git hosting platform.

# Git Architecture



# Git Architecture with Remote Repo



# Configuring Git

$ git config --global user.name "My Name"

$ git config --global user.email [myEmail@example.com](mailto:myEmail@example.com)

# .git Directory

The .git directory contains all the configurations, logs, branches, HEAD, and more

# Commands

## Creating a new repository

git init

## Checking the status

git status

## Staging

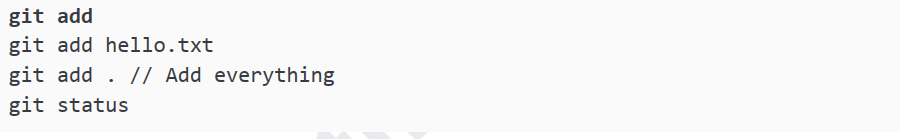
● Git has the concept of a "staging area".

● You can think of this like a blank canvas, which holds the changes which you would

like to commit.

● It starts out empty, but you can add files to it (or even single lines and parts of files)

with the git add command, and finally commit everything (create a snapshot) with git commit



# Committing

A commit represents the state of our repository at a given point in time.

It's like a snapshot, which we can go back to and see how thing were when we took it.



# Blame

Examine specific parts of the code’s history and find out who was the last author to modify

that line “`bash.

Show what revision and author last modified each line of a file

<https://git-scm.com/docs/git-blame>

# Remote repositories

## Connecting to a remote repository

1. In order to upload something to a remote repo.

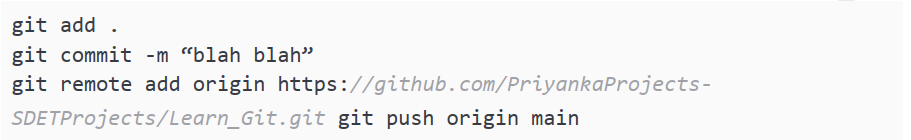
2. Create a Repo at Github.com and Add Remote to it.

3. The Git command to do this is git push and takes two parameters.

the name of the remote repo (we called ours origin) and the branch to push to



## Add File and Push



## Cloning a repository

● Download locally and have a fully working copy of your project

