Git

Introduction

- What is Git?
  - Git is a free and open source version control system designed to handle everything from small to very large projects with speed and efficiency.
- Repository hosting services/manager
  - GitHub, Gitlab , BitBucket etc.
- Why is it used?
  - Any changes made to the source code were unknown to the other developers.
  - Developers used to submit their codes to the central server without having copies of their own.

Functionalities

- Version control
    - Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.
- Branching
    - Branch is a new/separate version of the main repository.
- Share & collaborate
    - Collaboration is the way different people can work on the same project together

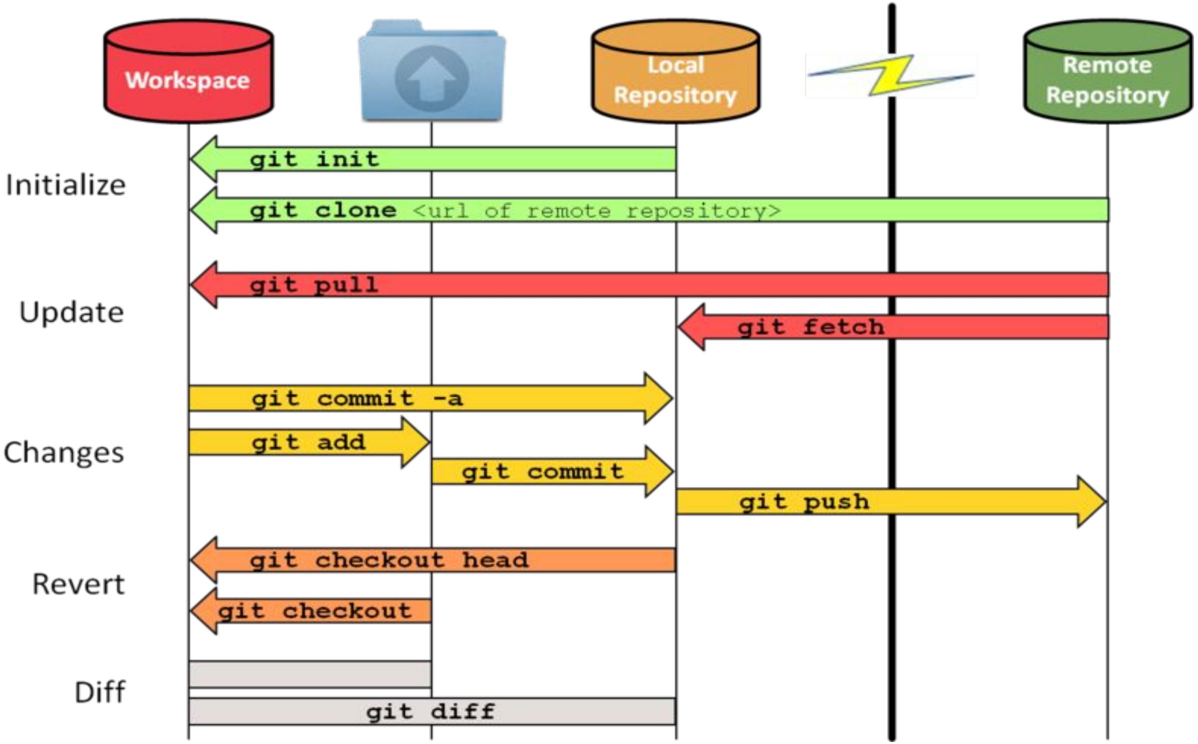
Git concepts:

- Repository
  - The folder containing the project files (all the code) is the code repository.
- Fork
  - A fork is a new repository that shares code and visibility settings with the original “upstream” repository.
- Clone
  - Getting a local copy of an existing repository to work on.

Git concepts:

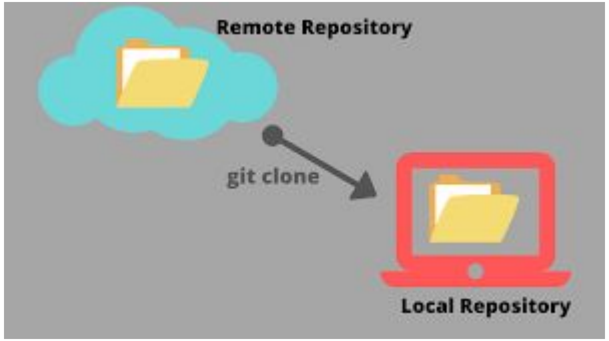
- Merge
  - Merging new local merge commits with remote repositories.
- Pull
  - Download content to local repository from the specified remote repository.
- Push
  - Upload local repository content to a remote repository

Git Workflow

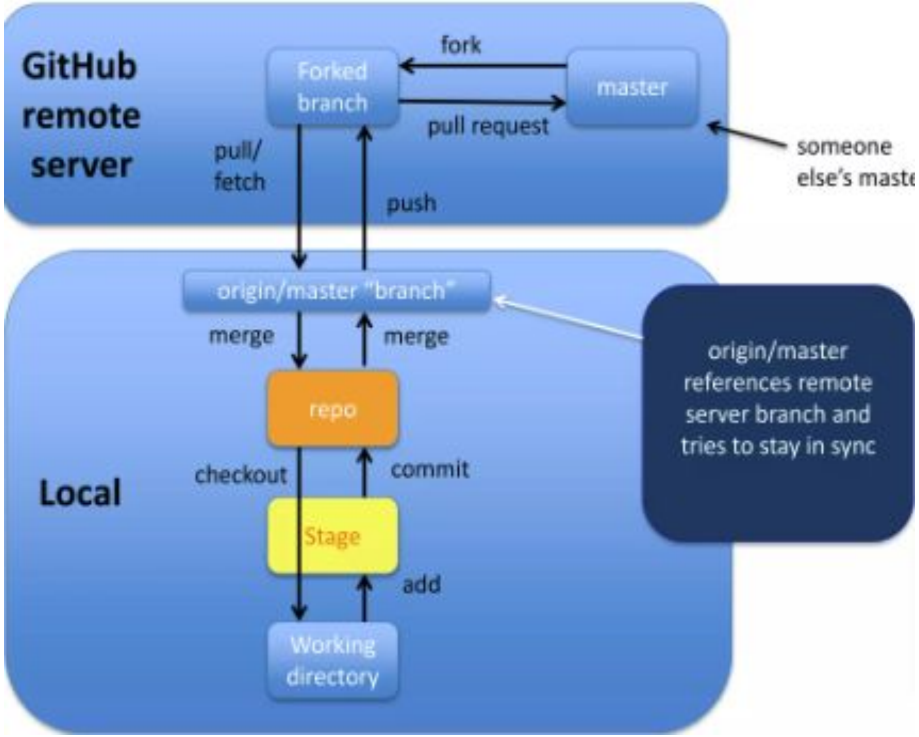


Git Commands

- git init:
  - The git init command creates a new Git repository.
  - It can be used to convert an existing, unversioned project to a Git repository or initialize a new, empty repository.
- git clone:
  - get a local copy of an existing repository to work on.
  - git clone <repositoy_url>

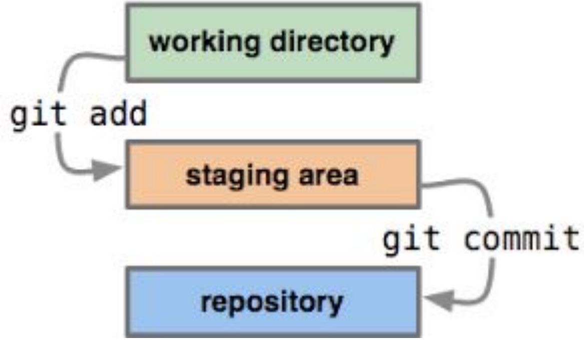


- Check version
  - git remote -v
- Add upstream to your local repository
  - git remote add upstream <upstream_repo_url>
- Fetch all branches from all remotes
  - git fetch --all



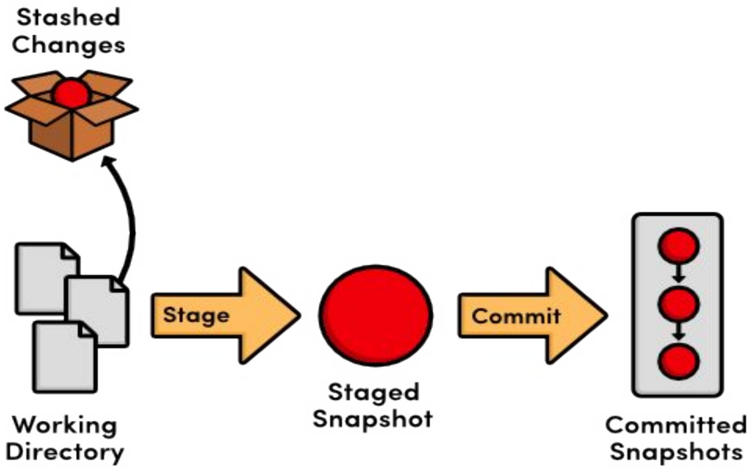
- Create a new branch in local referencing the remote upstream branch
    - git checkout upstream <branch_name>
    - git checkout -b <desired_branch_name>
- To list all local branch
    - git branch
- Sync local repository with remote repository
    - git pull upstream <branch_name>

- Stage the changes
  - git add <file_name>
- Check the state of the working directory and the staging area
  - git status
- View difference
  - git diff
- Commit a snapshot of all changes in the working directory
  - git commit -m “commit_message”



- Pushes a local branch(es) to a remote repository (origin)
    - The git push command is used to upload local repository content to a remote repository
    - git push origin <local_branch_name>
- Check differences between remote upstream branch and local branch
    - git diff upstream/<branch_name> <local_branch_name>

- Stashing the changes
  - Stashing takes the dirty state of your working directory — that is, your modified tracked files and staged changes — and saves it on a stack of unfinished changes that you can reapply at any time
  - git stash list
  - git stash
  - git stash push --message “stash message”
  - git stash pop stash@{index_number}



Thank You