GIT CHEAT SHEET

GIT

- Git is a distributed revision control and source code management system with an emphasis on speed.
- it is repository which is used to manage projects, set of files as they changes over the time.
- Using git every code change or commit you get latest development code for the project.

GIT OPERATIONS & COMMANDS

Git Configurations

- Initial config of username, email and code highlighting (optional) is to be performed.
- $git config -- global user.name"firstname lastname"
- $git config -- global user.email" abc123@abc.com"
- $git config -- global color.ui true (enables code highlights)
- $git config --list

Initialize

- You have to initialize by using 'init'
- To know the status run the 'status' command
- $git init
- $git status

Create/Add files

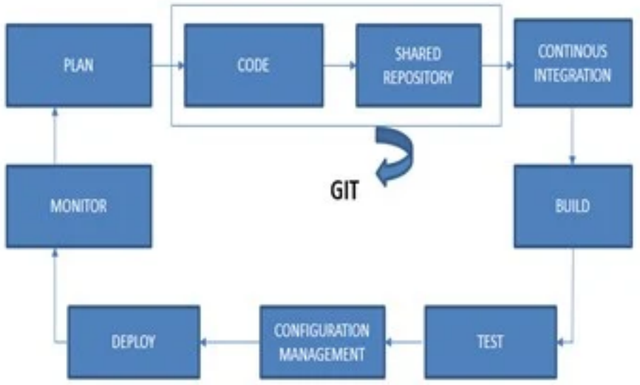
- To add a file: $git add<filename>
- To add multiple files: $git add<filename> <2nd filename>
- To add all updated files: $git add –all (use -A instead of -all too)
- To remove files: $git rm -r <filename>

Commit changes

- To pass a message, use 'commit' and '-m': $git commit -m " body_of_message"
- Amend lets you amend the last commit or the last message: $git commit --amend -m " new_message"

Push and Pull

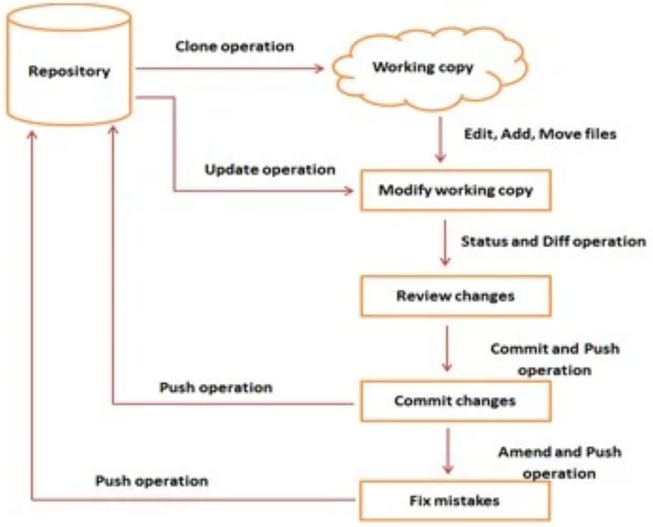
- A remote repository typically represents a remote server or a git server: Create a remote repository via github "https://github.com/YourUsername/appname.git"
- To add a link: $git remote add origin<link>
- Pushing files: $git push -u origin master
- To clone file: $git clone <clone>



Version Control

- It is the management of changes to the code, documents, programs, large sites and other info.
- The changes are termed as versions.
- Version control system is used(VCS)
- The functions are :
  - Allows developers to work simultaneously.
  - Does not allow overwriting each other's changes.
  - Maintains a history of every version.

2 types of VCS - centralized and distributed. Git is distributed



GIT & GITHUB

It is a VCS that supports distributed nonlinear workflows by providing data assurance for developing quality software.

Features:

- Distributed- distributed development of code
- compatible- with existing systems and protocols
- Non- linear- non linear development of code
- Branching- easy to create and merge branches
- Lightweight- lossless compression
- Reliable- not viable to loss of data upon crashes
- Secure- SHA1 and checksum are used
- Economical- free

Branching and Merging

Command	Description
git branch	List branches
git branch -a	List all branches
git branch [branch name]	Create a new branch
git branch -d [branch name]	Delete branch
git push origin --delete [branchName]	Delete a remote branch
git checkout -b [branch name]	Create a new branch and switch to it
git checkout -b [branch name] origin/[branch name]	Clone a remote branch and switch to it
git checkout [branch name]	Switch to a branch
git checkout -	Switch to the branch last checked out
git checkout -- [file-name.txt]	Discard changes to a file
git merge [branch name]	Merge a branch into the active branch
git stash	Stash changes in a dirty working directory
git stash clear	Remove all stashed entries

Updating Projects

Commands	Description
git push origin [branch name]	Push a branch to your remote repository
git push -u origin [branch name]	Push changes to remote repository(-u remembers the branch for next use)
git push origin --delete [branch name]	Delete a remote branch
git pull	Update local repository to the newest commit
git pull origin [branch name]	Pull changes from remote repository
git remote add origin ssh://git@github.com/[username]/[repository-name].git	Add a remote repository
git remote set-url origin ssh://git@github.com/[username]/[repository-name].git	Set a repository's origin branch to SSH

Inspection and Comparison

Command	Comparison
git log	View changes
View changes	View changes (detailed)
git diff [source branch] [target branch}	Preview changes before merging