CHEAT SHEET LINUX + GIT

GIT : :

Git is the free and open source distributed version control system that's responsible for everything GitHub related that happens locally on your computer. This cheat sheet features the most important and commonly used Git commands for easy reference.

INSTALLATION & GUIS With platform specific installers for Git, GitHub also provides the ease of staying up-to-date with the latest releases of the command line tool while providing a graphical user interface for day-to-day interaction, review, and repository synchronization.

GitHub for Windows htps://windows.github.com

GitHub for Mac htps://mac.github.com

-
- SETUP
- Configuring user information used across all local repositories
- git config --global user.name "[firstname lastname]" :: set a name that is identifiable for credit when review version history
- git config --global user.email "[valid-email]" :: set an email address that will be associated with each history marker
- git config --global color.ui auto :: set automatic command line coloring for Git for easy reviewing

GIT:

SETUP

Configuring user information used across all local repositories

· git config --global user.name "[firstname lastname]" : set a name that is identifiable for credit when review version history

· git config --global user.email "[valid-email]" :set an email address that will be associated with each history marker

· git config --global color.ui auto :set automatic command line coloring for Git for easy reviewing

SETUP & INIT

Configuring user information, initializing and cloning repositories

· git init :initialize an existing directory as a Git repository

· git clone [url]: retrieve an entire repository from a hosted location via URL

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

- git status: show modified files in working directory, staged for your next commit
- git add [file] :add a file as it looks now to your next commit (stage)
- git reset [file]: unstage a file while retaining the changes in working directory
- git diff :diff of what is changed but not staged
- git diff --staged :diff of what is staged but not yet commited
- git commit -m “[descriptive message]” :commit your staged content as a new commit snapshot

BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

- git branch: list your branches. a * will appear next to the currently active branch
- git branch [branch-name]: create a new branch at the current commit
- git checkout: switch to another branch and check it out into your working directory
- git merge [branch]: merge the specified branch’s history into the current one
- git log show all commits in the current branch’s history

INSPECT & COMPARE

Examining logs, diffs and object information

- git log :show the commit history for the currently active branch
- git log branchB..branchA: show the commits on branchA that are not on branchB
- git log --follow [file] :show the commits that changed file, even across renames
- git diff branchB...branchA show the diff of what is in branchA that is not in branchB
- git show [SHA] :show any object in Git in human-readable format

TRACKING PATH CHANGES

Versioning file removes and path changes

- git rm [file] :delete the file from project and stage the removal for commit
- git mv [existing-path] [new-path]: change an existing file path and stage the move

· git log --stat -M: show all commit logs with indication of any paths that moved

SHARE & UPDATE

Retrieving updates from another repository and updating local repos

· git remote add [alias] [url] :add a git URL as an alias

· git fetch [alias] :fetch down all the branches from that Git remote

· git merge [alias]/[branch]: merge a remote branch into your current branch to bring it up to date

· git push [alias] [branch] :Transmit local branch commits to the remote repository branch

· git pull: fetch and merge any commits from the tracking remote branch

REWRITE HISTORY

Rewriting branches, updating commits and clearing history

· git rebase [branch]: apply any commits of current branch ahead of specified one

· git reset --hard [commit] :clear staging area, rewrite working tree from specified commit

TEMPORARY COMMITS

Temporarily store modified, tracked files in order to change branches

· git stash :Save modified and staged changes

· git stash list :list stack-order of stashed file changes

· git stash pop :write working from top of stash stack

· git stash drop :discard the changes from top of stash stack

LINUX:: Commands

File Operations:

●ls : Lists all files and directories in the present workingdirectory

●ls -R: Lists files in sub-directories as well

●ls -a: Shows hidden files

●ls -al: Lists files and directories with detailed information likepermissions, size, owner, etc.

●cd directoryname: Changes the directory

●cd ..: Moves one level up

●pwd: Displays the present working directory

●cat > filename: Creates a new file

●cat filename: Displays the file content

●cat file1 file2 > file3: Joins two files (file1 and file2) and storesthe output in a new file (file3)

●touch filename: Creates or modifies a file

●rm filename: Deletes a file

●cp source destination: Copies files from the source path to thedestination path

●mv source destination: Moves files from the source path to thedestination path

●find / -name filename: Finds a file or a directory by its namestarting from root

●file filename: Determines the file type

●less filename: Views the file content page by page

●head filename: Views the first ten lines of a file

●tail filename: Views the last ten lines of a file

●lsof: Shows which files are opened by which process.

●du -h --max-depth=1 : Shows the size of each directory. Use

--max-depth=1 ::to limit the output to the current directory and itsimmediate children.

●fdisk : Disk partition manipulation command.

2.Directory Operations:

● mkdir directoryname: Creates a new directory in the presentworking directory

● rmdir directoryname: Deletes a directory

● cp -r source destination: Copies directories recursively

● mv olddir newdir: Renames directories

● find / -type d -name directoryname: Finds a directory starting fromroot

3. Process Operations:

- ps: Displays your currently active processes

- top: Displays all running processes

- kill pid: Kills the process with given pid

- pkill name: Kills the process with the given name

- bg: Resumes suspended jobs without bringing them to foreground

- fg: Brings the most recent job to foreground

- fg n: Brings job n to the foreground

- renice +n [pid]: Change the priority of a running process.

- &>filename: Redirects both the stdout and the stderr to the filefilename.

- 1>filename: Redirect the stdout to file filename.

- 2>filename: Redirect stderr to file filename.

4. File Permissions:

- chmod octal filename: Change the permissions of file to octal,which can be between 0 (no permissions) to 7 (full permissions)

- chown ownername filename: Change file owner

- chgrp groupname filename: Change group owner

5. Networking:

- ping host: Ping a host and outputs results

- whois domain: Get whois information for domain

- dig domain: Get DNS information for domain

- netstat – pnltu : Display various network related information suchas network connections, routing tables, interface statistics etc.

- ifconfig: Displays IP addresses of all network interfaces

- ssh user@host: Remote login into the host as user

- scp: Transfers files between hosts over ssh

- wget url: Donload files from the web

- curl url: Sends a request to a URL and returns the response

- traceroute domain: Prints the route that a packet takes to reachthe domain.

- mtr domain : mtr combines the functionality of the traceroute andping programs in a single network diagnostic tool.

- ss: Another utility to investigate sockets. It's a more modernalternative to

netstat

- nmap: Network exploration tool and security scanner.

6. Archives and Compression:

- tar cf file.tar files: Create a tar named file.tar containing files

- tar xf file.tar: Extract the files from file.tar

- gzip file: Compresses file and renames it to file.gz

- gzip -d file.gz: Decompresses file.gz back to file

- zip -r file.zip files: Create a zip archive named file.zip

- unzip file.zip: Extract the contents of a zip file

7. Text Processing:

- grep pattern files: Search for pattern in files

- grep -r pattern dir: Search recursively for pattern in dir

8.Disk Usage:

- df: Shows disk usage

- du: Shows directory space usage

- free: Show memory and swap usage

- whereis app: Show possible locations of app

9. System Info:

- date: Show the current date and time

- cal: Show this month's calendar
- uptime: Show current uptime
- w: Display who is online
- whoami: Who you are logged in as
- uname -a: Show kernel information
- df -h: Disk usage in human readable format
- du -sh: Disk usage of current directory in human readable format
- free -m: Show free and used memory in MB

10. Package Installations:

- sudo apt-get update Updates package lists for upgrades
- sudo apt-get upgrade: Upgrades all upgradable packages
- sudo apt-get install pkgname: Install pkgname
- sudo apt-get remove pkgname: Removes pkgname

11. Version Control (Git commands):

- git init: Initialize a local git repository
- git clone url: Create a local copy of a remote repository
- git add filename: Add a file to the staging area
- git commit -m "Commit message": Commit changes with a message
- git status: Check the status of the working directory
- git pull: Pull latest changes from the remote repository
- git push: Push changes to the remote repository
- git branch: List all local branches
- git branch branchname: Create a new branch
- git checkout branchname: Switch to a branch
- git merge branchname: Merge a branch into the active branch
- git stash: Stash changes in a dirty working directory

- git stash apply: Apply changes from a stash
- git log: View commit history
- git reset: Reset your HEAD pointer to a previous commit
- git rm filename: Remove a file from version control
- git rebase: Reapply commits on top of another base tip.