**GIT CREATE**

Clone an existing repository

$ git clone ssh://user@domain.com/repo.git

Create a new local repository

$ git init

**LOCAL CHANGES**

View Changed files in your working directory

$ git status

View Changes to tracked files

$ git diff

To Add all current changes to the next commit

$ git add .

To Add some changes in <file> to the next commit

$ git add -p <file>

Commit all local changes in tracked files

$ git commit -a

Commit previously staged changes

$ git commit

Change the last commit

*Don‘t amend published commits!*

$ git commit --amend

**COMMIT HISTORY**

Show all commits, starting with newest

$ git log

Show changes over time for a specific file

$ git log -p <file>

Who changed what and when in <file>

$ git blame <file>

**BRANCHES & TAGS**

List all existing branches

$ git branch -av

Switch HEAD branch

$ git checkout <branch>

Create a new branch based

on your current HEAD

$ git branch <new-branch>

Create a new tracking branch based on a remote branch

$ git checkout --track <remote/branch>

Delete a local branch

$ git branch -d <branch>

Mark the current commit with a tag

$ git tag <tag-name>

**UPDATE & PUBLISH**

List all currently configured remotes

$ git remote -v

Show information about a remote

$ git remote show <remote>

Add new remote repository, named <remote>

$ git remote add <shortname> <url>

Download all changes from <remote>,

but don‘t integrate into HEAD

$ git fetch <remote>

Download changes and directly merge/integrate into HEAD

$ git pull <remote> <branch>

Publish local changes on a remote

$ git push <remote> <branch>

Delete a branch on the remote

$ git branch -dr <remote/branch>

Publish your tag s

$ git push --tags

**MERGE & REBASE**

Merge <branch> into your current HEAD

$ git merge <branch>

Rebase your current HEAD onto <branch>

*Don‘t rebase published commits!*

$ git rebase <branch>

Abort a rebase

$ git rebase --abort

Continue a rebase after resolving conflicts

$ git rebase --continue

Use your configured merge tool to

solve conflicts

$ git mergetool

Use your editor to manually solve conflicts and (after resolving) mark file as resolved

$ git add <resolved-file>

$ git rm <resolved-file>

**UNDO**

Discard all local changes in your working  directory

$ git reset --hard HEAD

Discard local changes in a specific file

$ git checkout HEAD <file>

Revert a commit (by producing a new commit with contrary changes)

$ git revert <commit>

Reset your HEAD pointer to a previous commit and discard all changes since then

$ git reset --hard <commit>

preserve all changes as unstaged changes

$ git reset <commit>

preserve uncommitted local changes

$ git reset --keep <commit>

Git commands in telugu

Git commands for git and github

What is (1.working directory) (2.staging area) (3.local repository) (4.remoterepository)

* yum update -y = emina files vunte update avuthayi
* yum install git -y = git install avuthundi
* which git = git yekkda install aydho chupistundi
* git --version = version chupistundi
* git config user.name “kareem” = ec2-user lo cheyali ante root user lo

git config --global user.name “kareem”

* git config user.email “karimulla073@gmail.com

git config --global user.email “karimulla073@gmail.com

* git config --list = user and email chupistundi
* mkdir dir1 = voka directory create cheystam
* cd dir1 = dir1 lopalike veltham
* git init = creates a new git repository
* git status = check the status
* Cat file1 = voka file create cheystam

E process motham ni working directory antaru

* git add file1 = file1 untrack files ni add cheyali ante cmd (untrack means add cheyani files ni untrack antaru
* git add . = e command tho all untrack files add avthai
* git log = Show all commits, starting with newest (commit id chupistundi)
* git log --oneline = ani commit cheyasamo ani chupistundi
* git show url = file lo vuna content ni chupistundi

Staging area is important part of git

Staging area anadi manom local repository and remote repository ke files ni add cheyali anukuntanamo avi staging area lo add cheystam

E process motham ni staging area antaru

* Git commit -m “new file creating” = commit aypothundi

E process ni local repository antaru

* **…or create a new repository on the command line**
* echo "# git" >> README.md
* git init
* git add README.md
* git commit -m "first commit"
* git branch -M main
* git remote add origin git@github.com:Karimshaik99/git.git
* git push -u origin main
* **…or push an existing repository from the command line**
* git remote add origin git@github.com:Karimshaik99/git.git
* git branch -M main
* git push -u origin main
* Git branch = ye branch lo vunamo chupistundi (moster)
* Git push -u origin master = enter push avthundi remote repository

E process ni remote repository antaru

* git branch (branch name) = branch create cheydaniki
* Git checkout -b “file1” = branch create cheyadaniki
* git checkout (branch name) = branch loki shift avadnike

Branch anadi master lo vuna files ni copy chaistundi.Branch lo emina changes

cheysina master lo changes avavu (branch ni voka folder type anachu)

Git restore --staged file = untrack cheyali anukunte

Git restore --staged . = all files untrack avthai

Git add kareem\* = manom ani files create cheysamo kareem user lo ani add avthai

Vi .gitignore = ignore cheyali anukuna files like .file1 .file2 .java .css

git merge branch1 = branch1 lo vuna files ni manom master loki merge chestunom appudu branch1 lo vuna files ani master loki merge aypothai (e process manom github nudi kuda cheyachu and cmd line through kuda cheyachu)

conflect = means master lo same file1 create cheyste and branch1 lo kuda same name tho create cheyste file1 manom files ni merge cheyali anukunte avadu conflect ani chupistundi deniki inka process vundi manaki present avasaram ledhu

git stash =

revert = revert is a code edina wrong edina rastey dani manom ignore cheyali anukunte e cmd (process inka vundi present avasaram ledhu)

git clean -n = files ni add cheyani vatine manom mothom delete cheyali anukunte cmd e command tho mothom delete avthai gitadd cheyani files (git clean -f)files ani delete cheystundi untrack files ni