**Git is a VCS (Version control system) to track your files.**

**Untracked-----Staged-----Tracked**

A chalkboard with a chalkboard and a play

AI-generated content may be incorrect.

**git init**

A screen shot of a computer

AI-generated content may be incorrect.

**git status**

A screen shot of a computer

AI-generated content may be incorrect.

**git add backup.sh**

A screenshot of a computer

AI-generated content may be incorrect.

**git rm --cache backup.sh**

**git status**

A black screen with white text

AI-generated content may be incorrect.

**git add backup.sh**

**git commit -m “added backup.sh”**

A screenshot of a computer screen

AI-generated content may be incorrect.

**git restore backup.sh**

A screenshot of a computer

AI-generated content may be incorrect.

**git log –-oneline**

A black background with white text

AI-generated content may be incorrect.

**git branch**

**git branch checkout -b dev**

A screenshot of a computer program

AI-generated content may be incorrect.

**git merge**

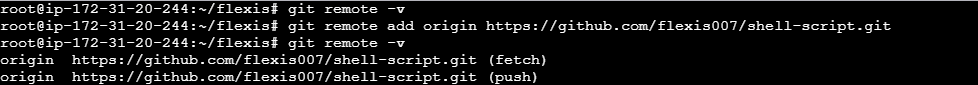
A screenshot of a computer program

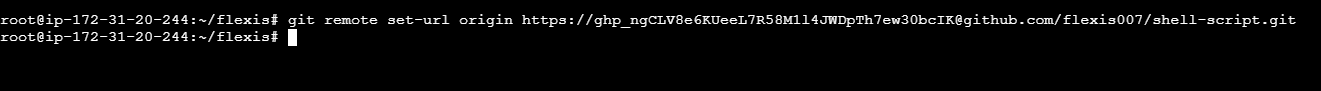
AI-generated content may be incorrect.

**git remote -v**

**git remote add origin URL**

**git remote set-url origin https://token@github.com/flexis007/shell-script.git**





**git push origin master**

A computer screen with white text

AI-generated content may be incorrect.

**git pull origin master**

A screen shot of a computer

AI-generated content may be incorrect.

**git clone url**

A screenshot of a computer screen

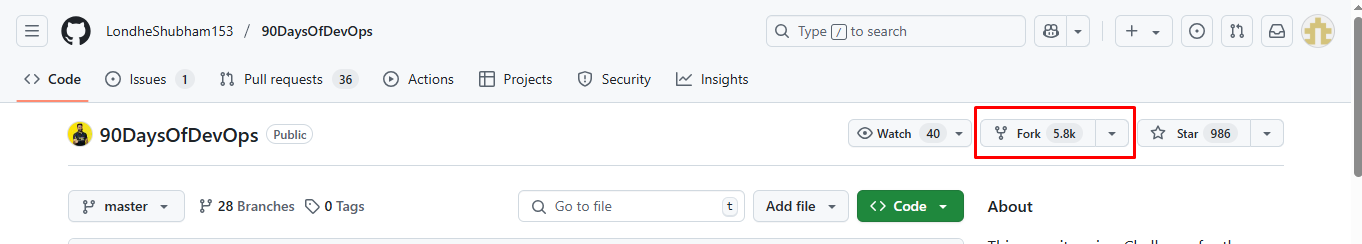
AI-generated content may be incorrect.

**git diff**

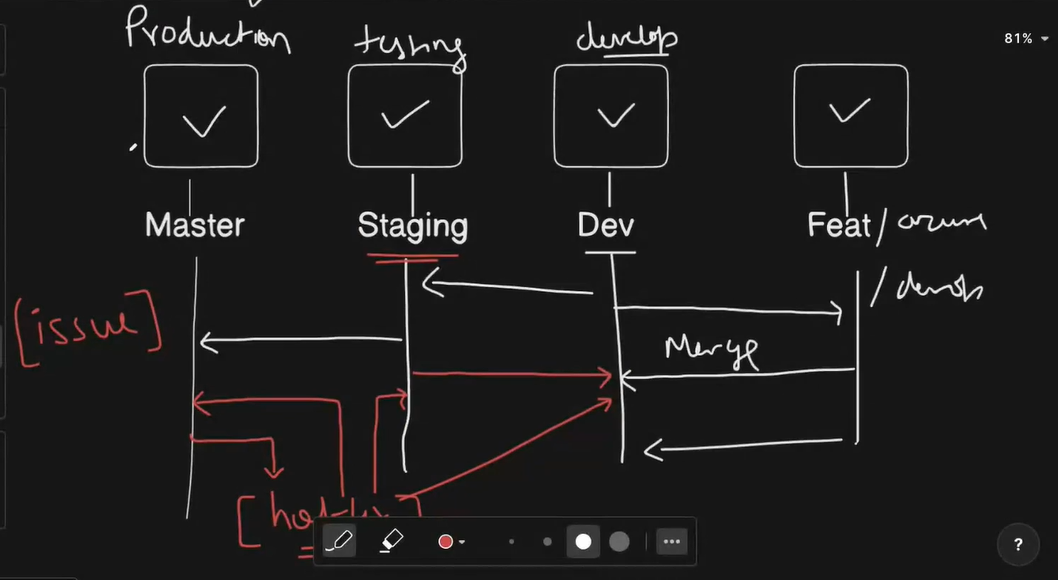
A screenshot of a computer

AI-generated content may be incorrect.

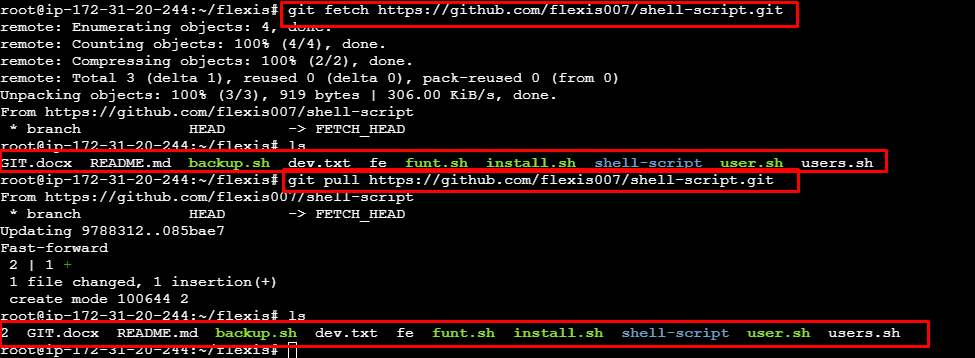
**git fork**



**Branching Strategies**



**git fetch vs git pull**



**git Pull Request**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**git revert**

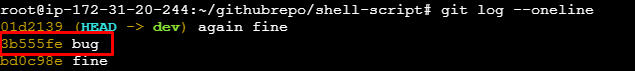
**if you have 3 commit and revert 2nd commit on same file then it will conflict but if only 2 commit and revert last one it’s not conflict.**

**A screen shot of a computer

AI-generated content may be incorrect.**

**A screen shot of a computer

AI-generated content may be incorrect.**



A screen shot of a computer

AI-generated content may be incorrect.

**git reset**

**soft --- remove from the commit they are currently staging area**

**mixed ---- remove from the commit and staging area**

**hard ---- removed permanently** A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Delete Branch on remote:**

A screen shot of a computer

AI-generated content may be incorrect.

**Delete Local branch**

A computer screen with many colorful text

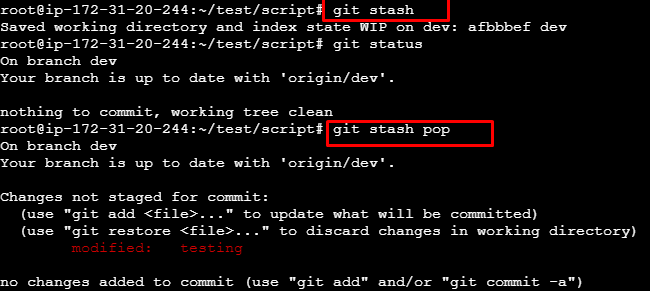
AI-generated content may be incorrect.

**git stash**

**git stash pop**

**A black and white screen with white text

AI-generated content may be incorrect.**



**Git Merge VS Git Rebase:**

**Git Merge: -** Non Linear way commit

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Git Rebase Example:** Linear way commit

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Git Cherry-pick:**

Git cherry-pick is a command that allows you to apply a specific commit from one branch to another. It's like picking a single cherry from a tree and placing it on another

A screenshot of a computer

AI-generated content may be incorrect.

**Git Merge Conflicts:**

Farhan@DESKTOP-IR4NRGD MINGW64 ~/Documents/test/stash (main)

$ echo " Line1: change" > example.txt

git add example.txt

git commit -m "m1" example.txt

git checkout imti

$ echo " Line2: change" >> example.txt

git add example.txt

git commit -m "m2" example.txt

git checkout main

echo " Line2: main change" >> example.txt

git add example.txt

git commit -m "m3" example.txt

git merge imti

Auto-merging example.txt

CONFLICT (content): Merge conflict in example.txt

Automatic merge failed; fix conflicts and then commit the result.

vim example.txt

$ cat example.txt

<<<<<<< HEAD

Line1: change

Line2: main change

=======

Line1: Initial

Line2: change

>>>>>>> imti

git add example.txt

git commit -m "resolveconflict"

**Squash Merge**

Combine multiple commits into a single commit to keep a clean history.

git merge --squash

git commit # Commit the squashed changes