Version Control

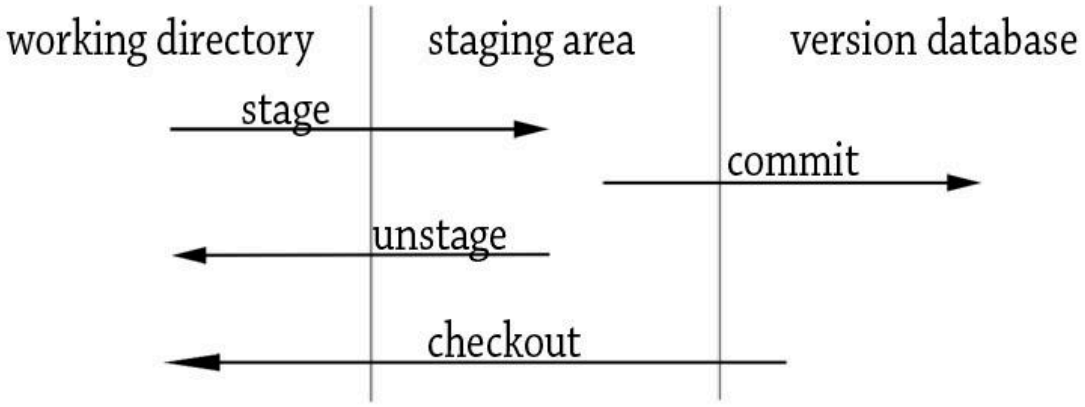
GIT

- Distributed Version Control System
- Collaboration
- Source Code Management
- Open Source Projects
- Version History
- Ability to rollback

Developed in 2005 by **Linus Torvalds** for managing Linux Kernel Development.

Key Concepts:

- working directory
- staging area / index
- version database / repository



Basic Git commands

Check existing configuration

git config -l

Git configuration

git config --global user.name "apurwa"

git config --global user.email "apurwa@gmail.com"

Check existing configuration

git config -l

Check git status

git status

Initialize current directory in GIT

git init

Note: .git folder is created

Start tracking files by adding files to staging area

git add .

Note: . refers to current directoryCommit to version database

git commit -m "first one"

Check git logs

git log

GITHUB Key based authentication

Generate ssh key pair in local machine.

ssh-keygen -t rsa -b 4096 -C "apurwa@gmail.com"

Start the SSH agent with following command:

eval "$(ssh-agent -s)"

Add the ssh key to the ssh agent:

ssh-add ~/.ssh/id_rsa

Note: Here, **~** refers to the home folder of the current user. You can also use the complete path such as **/Users/apurwa/.ssh/id_rsa**

Note: For Windows

The ssh agent can be started from PowerShell with following command:

Start-Service ssh-agent

If the agent cant be started, go to Windows Services and check for a service called ***OpenSSH Authentication Agent***. Make sure that it is started. Once that is done, try the command again.

After that, add the ssh keys to the agent with following command

ssh-add <path to private key>

Add public key to Github

Go to **github.com/settings/keys**, create a new SSH key and paste the content of the public key **id_rsa.pub**.

Test connection from the host OS.

ssh git@github.com

Note: Don't use your username.

You should see a message like this:

Hi apurwa-np! You've successfully authenticated, but GitHub does not provide shell access.

Connection to github.com closed.

Now, we can push to GitHub repo using key based authentication from our host OS.

Add remote repo

git remote add devops git@github.com:apurwa-np/devops.git

Push to remote repo

git push devops main

Pull from remote repo

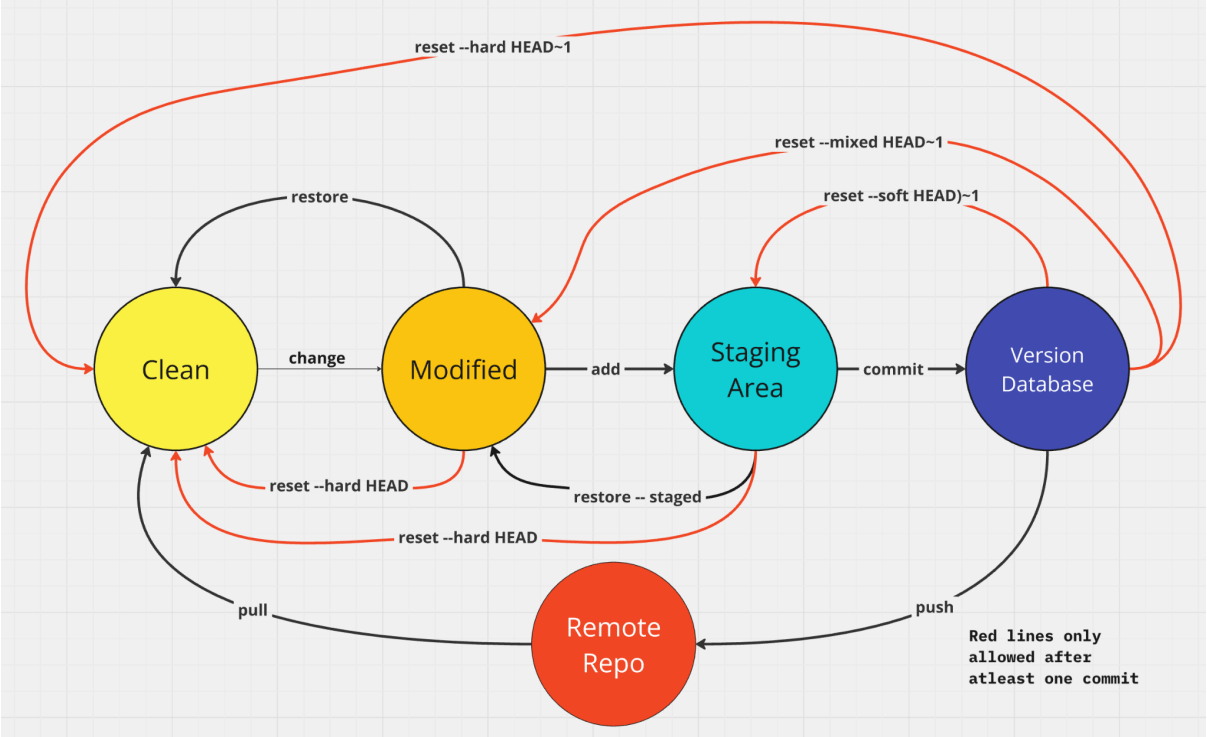
git pull devops main

Clone remote repo

git clone git@github.com:apurwa-np/devops.git

Note: A public repo can be viewed and cloned by anyone. But only users with SSH keys can push to the repo.

Undoing in GIT



Action	Undo unstaged changes
Command	**git restore <file>**
Scenario	Need to undo changes in a file which was modified but wasn't staged yet.

Action	Undo staged changes
Command	**git restore --staged <file>**
Scenario	A file was modified and staged. Need to undo the staging and make additional changes before committing. Go back one step from staging.

Action	Discard staged changes
Command	**git reset --hard**
Scenario	Discards all changes since the last commit. Note: Need at least one commit

Action	Undo a commit and keep changes staged
Command	**git reset --soft HEAD~1**
Scenario	Add more changes or change the commit message. Note: Need at least one commit

Action	Undo a commit and keep changes unstaged
Command	**git reset --mixed HEAD~1**
Scenario	Undo the commit and make additional changes. Note: Need at least one commit

Action	Undo the last commit and discard all changes
Command	**git reset --hard HEAD~1**
Scenario	Just committed but everything was fine at the previous commit. Note: Need at least one commit

GIT branching

- Independent feature development
- Parallel collaboration
- Conflict avoidance
- Modular development
- Easier Rollback

Create new branch

git branch greet

Switch to new branch

git checkout greet

Make changes for new feature

app.py

```
from flask import Flask
app=Flask(__name__)
@app.route('/')
def webout():
      return '<h1>DevOps is fun.</h1>'

@app.route('/greet')
def greet():
      return '<h1>Welcome to the new feature</h1>'

app.run(host='0.0.0.0',port=7000)
```

Note: Added a new route **/greet** in python flask app

Add a new commit

git commit -a -m “new route /greet added in new branch”

Push the new branch to remote repo

git push devops greet

Merge the new branch into main branch

git merge greet