GIT & GITHUB

SCM/VCS

DVCS

CVCS

CD

Workstation

IN

App → Code

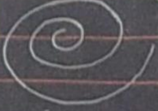
App → feature

Code → Deploy → Web Server

Production

Source Code Management → SCM

Life-Cycle



Timeline

1month → 1week → print......

print.....

print......

file/Code → WorkSpace

WIP

Work In Progress

Working Area

	Lines
30th Jan →	10
31st Jan →	20
1st Feb →	30
2nd Feb →	40

marker

30 → Backup

40 → Backup

Backup – Storage – Area → Commit Area

Area → Restore

Result of Crash → * Reputation/Loss

* Plan the backup

Backup Timeline

Rollout

31st Jan → 1st Feb → 6th Feb → 7th Feb

Version1 Version2 Version3 Version4

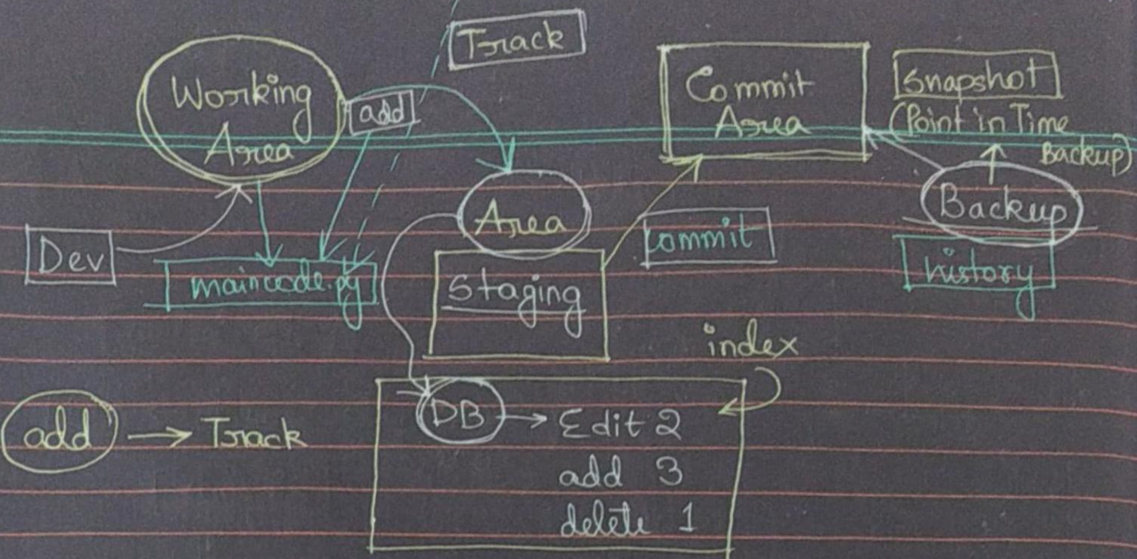
RollBack

VCS → Version Control System

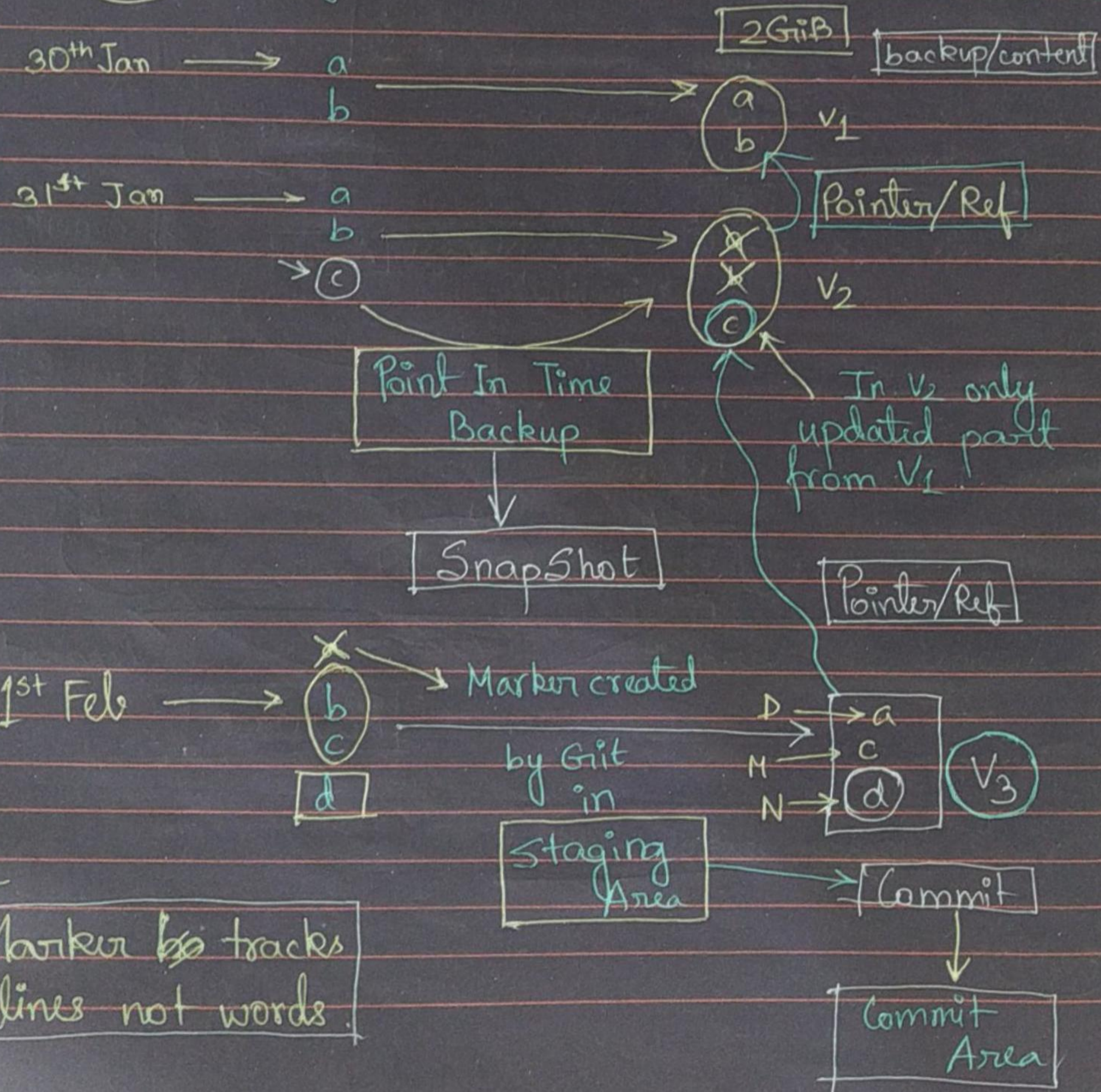
SCM → VCS

SCM → Git

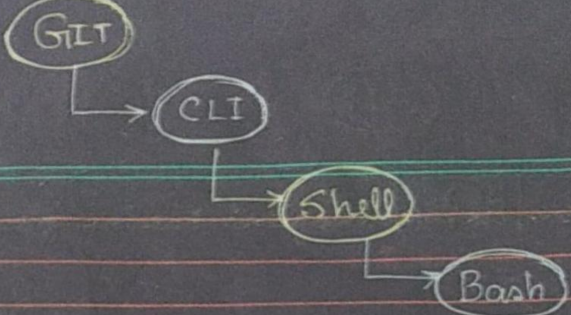
Commit → backup



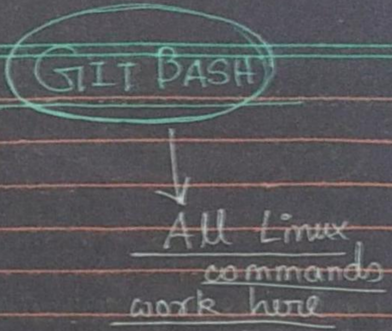
WA Working Area



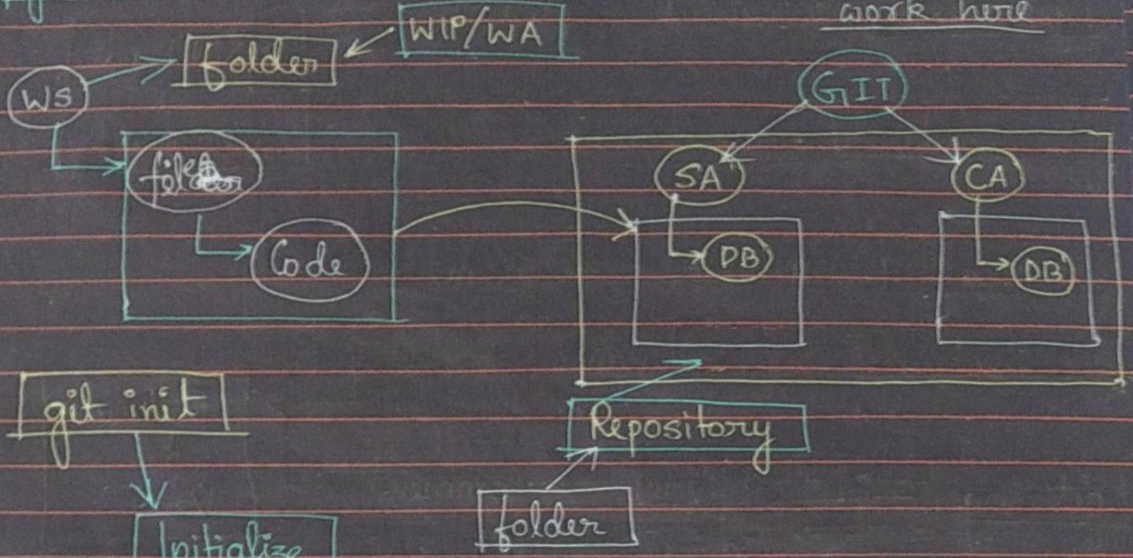
*
Marker ~~bo~~ tracks lines not words.



git 2.30.0



```
git --version
```



```
git init
```

↓

Initialize → Create Repository

.git

Git - Whatever we do in Git will be stored in this.

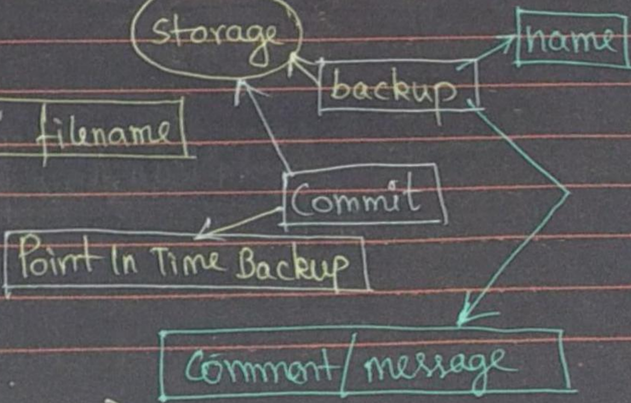
```
git status
```

```
git log
```

```
git reflog
```

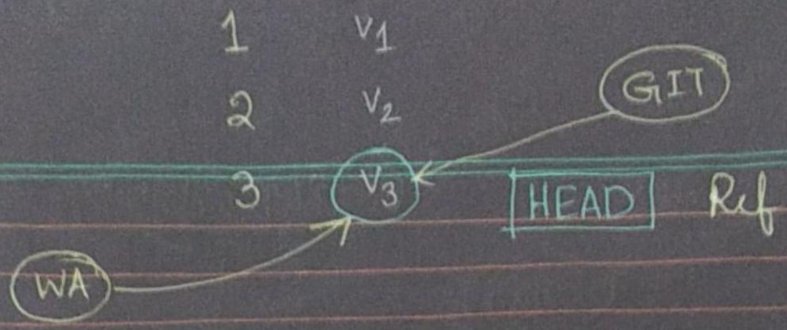
Commit Area

```
git commit -m "...." filename
```



Add once, commit multiple times

ommit → ID . . . . .
(HEAD → Master)



git reset commit id filename → Rollback to Version 1

git checkout -- filename

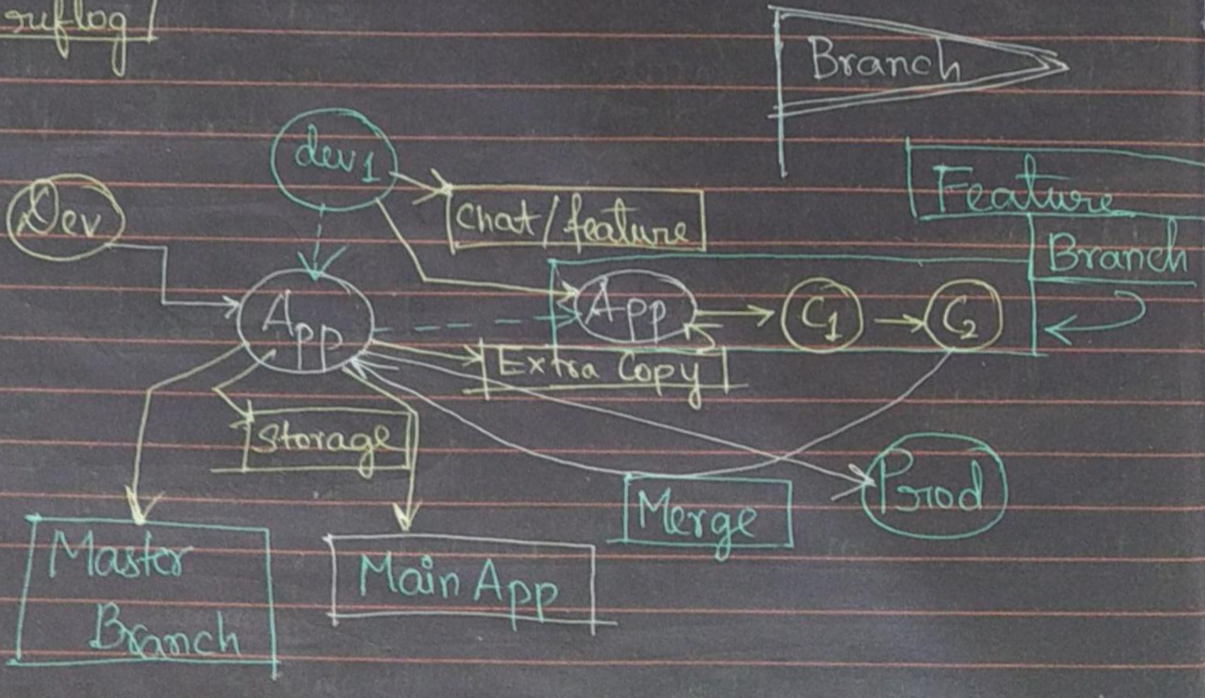
git reset commit id filename

git checkout -- filename → Rollout to V3

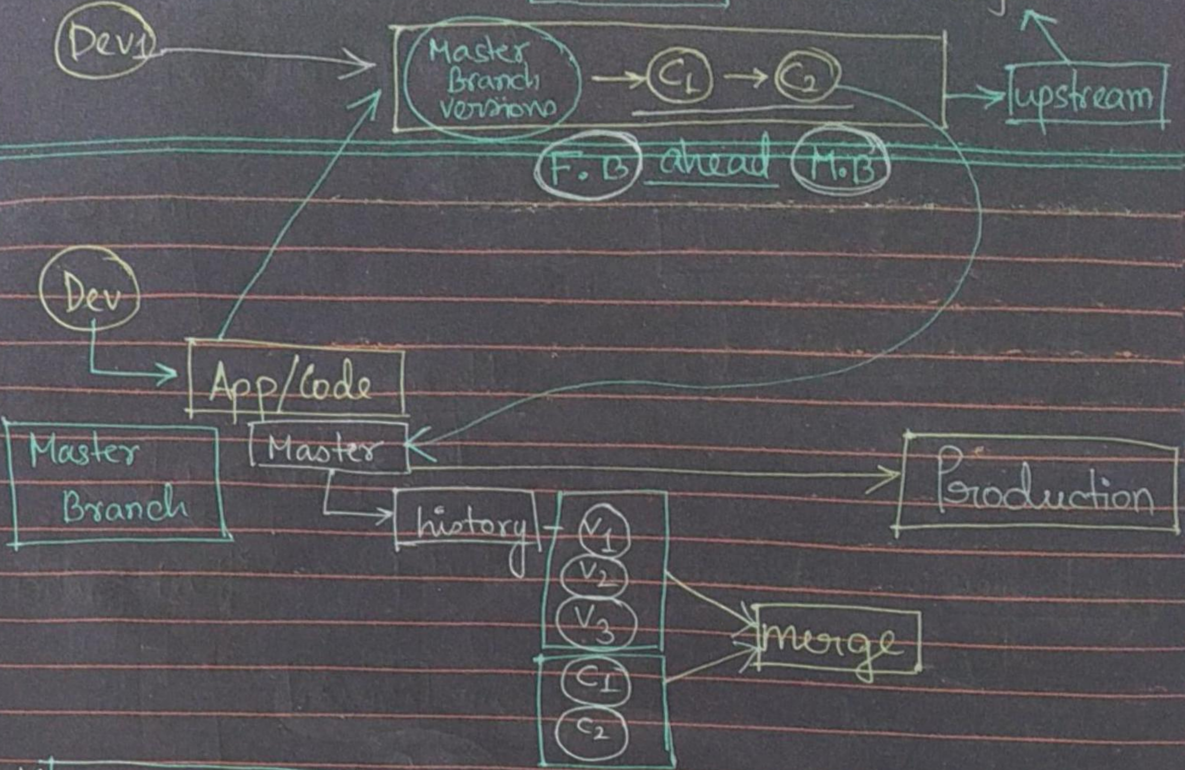
git log → --oneline → Comment

git log → Entire History

git reflog



Feature Branch

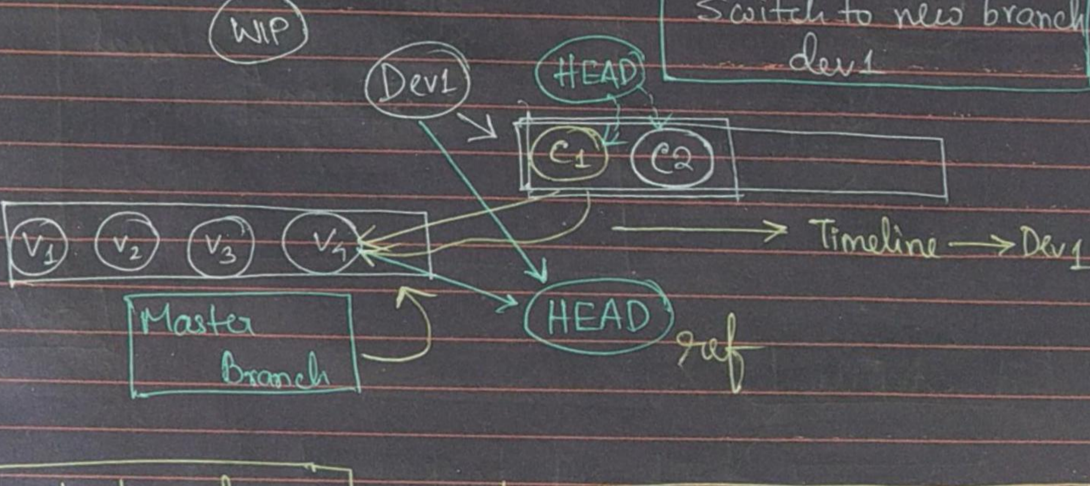


* Every Branch have their own personal timeline

```
git branch
```

```
git checkout -b dev1
```

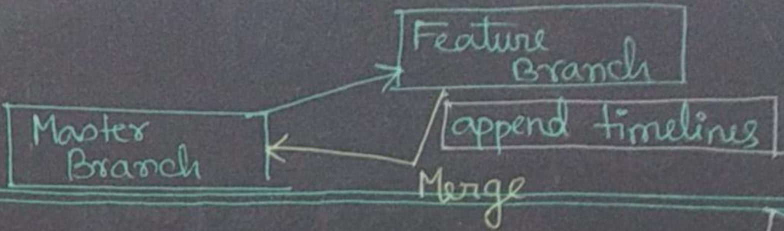
Create and Switch to new branch dev1



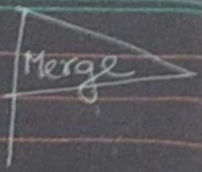
```
git branch -h
```

```
git branch --show-current
```

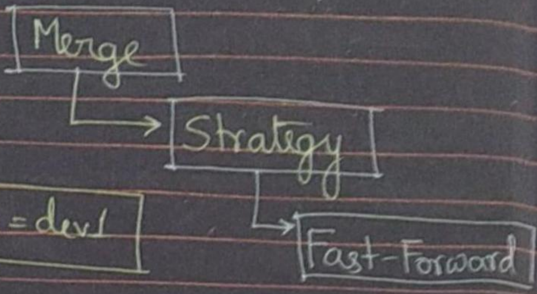
```
git checkout master
```



```
git checkout master
```

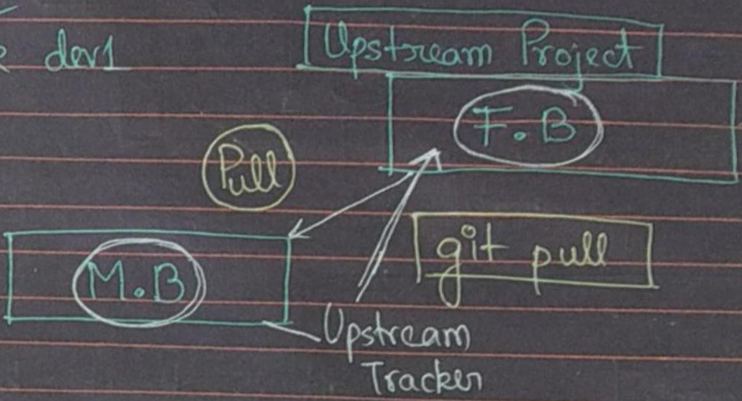


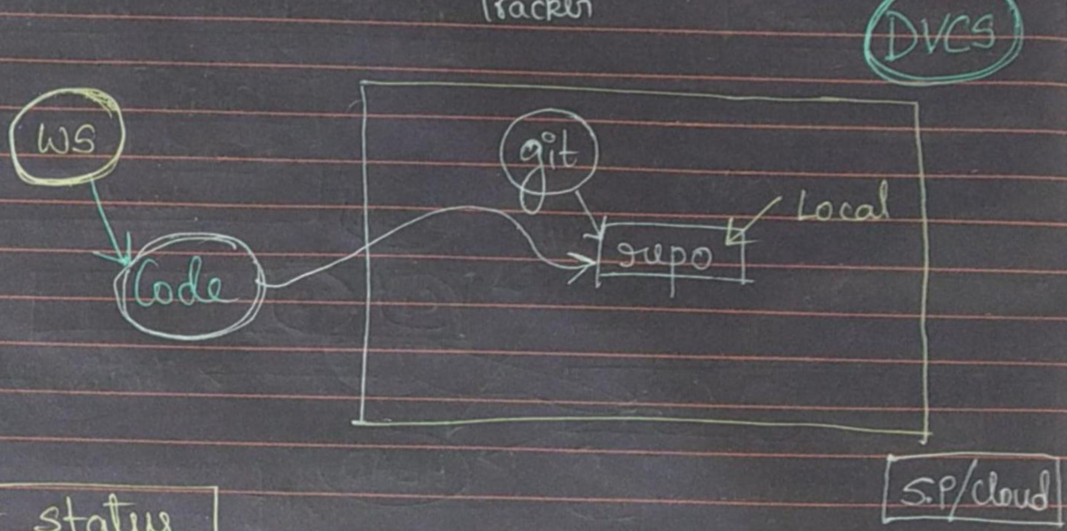
```
git merge dev1
```



```
git branch --set-upstream-to=dev1
```

Track dev1





```
git status
```

If git tracker set for upstream branch, this command will show me if master branch is behind or at par with upstream branch

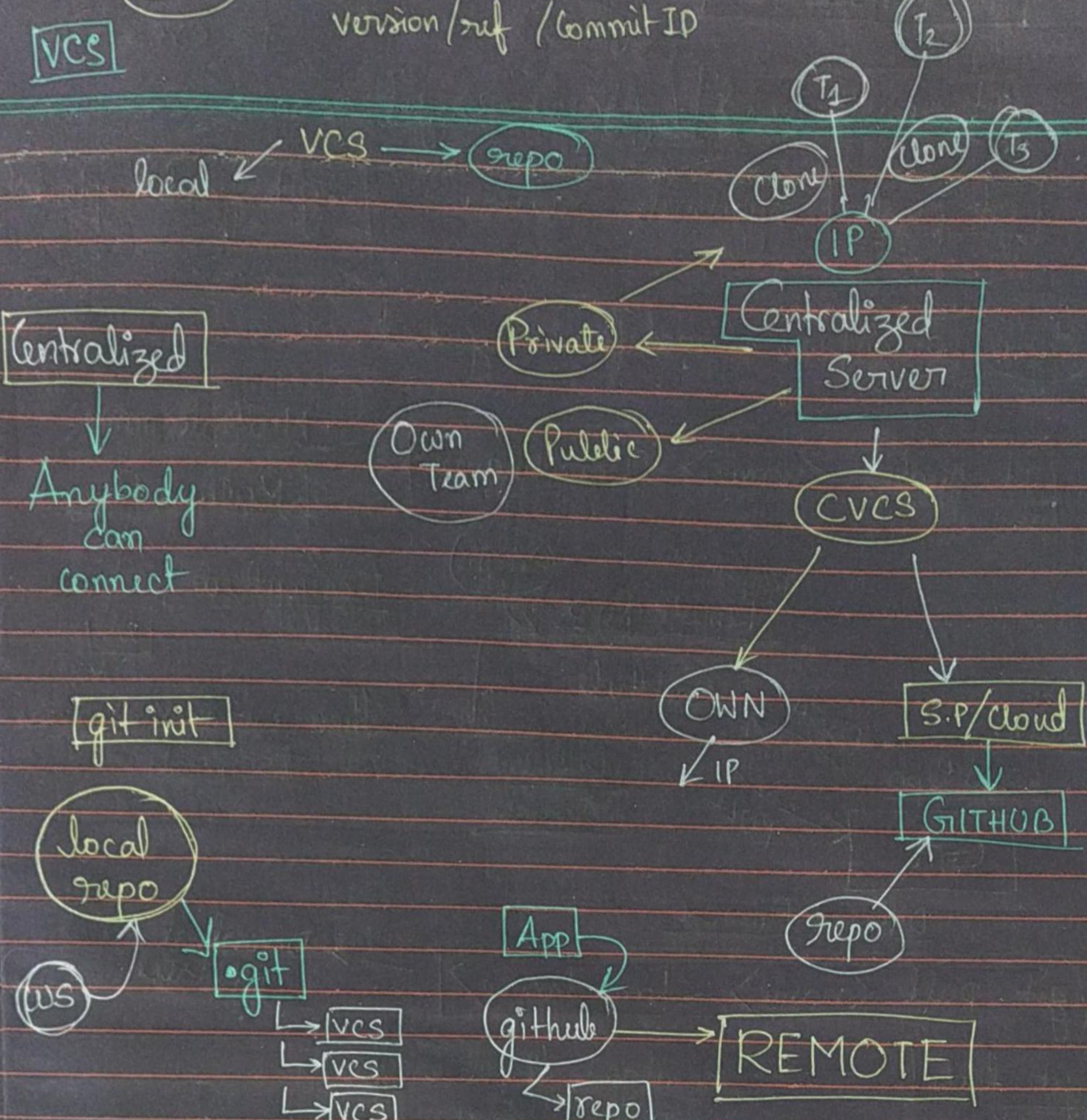
gitpull → Only work if upstream set

SCM

GIT

VCS

version/ref /Commit ID



git add .

If we want to upload our local git repo to CVCS we should not initialize GitHub Repo.

GITHUB

git remote -v

git remote add → name → Connect local repo to CVCS repo
githubrepo.git

git push mygithub master [.git folder uploaded]

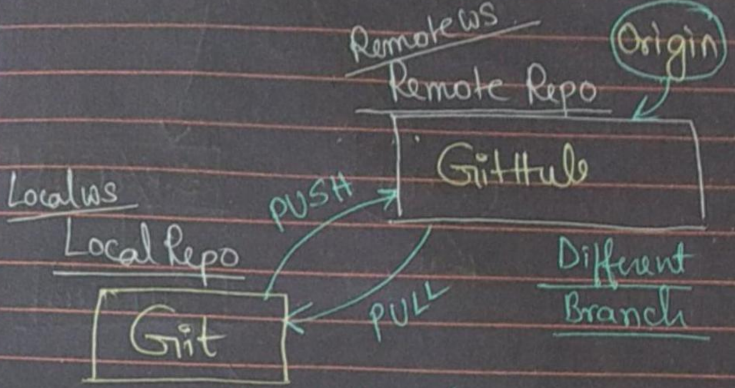
Myth Origin is a predefined keyword

git push <name> <branchname>

Multiple name

Multiple Repo
- mygithub
- my...

Then we decide where to push.



Different Branch

git pull <name> <branchname> → Download Updates

git fetch → Download timeline from Remote System

git fetch → git pull

git bash → Credential Manager

```
git branch --set-upstream-to=mygithub/master
/ master
```

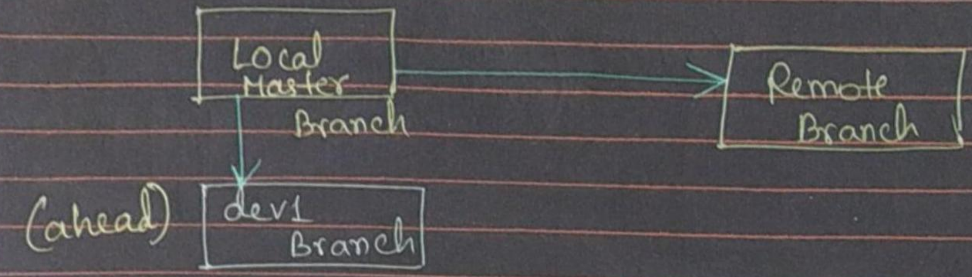
git fetch | git status | git pull

```
git show commit id
```

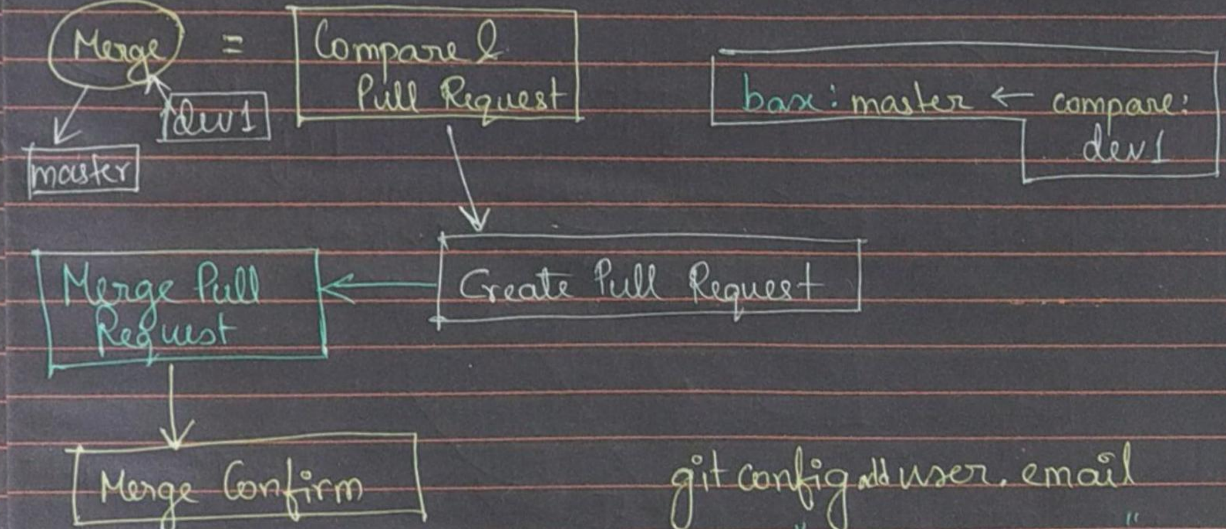
```
git diff fbC3 c6be
git diff c6be fbc3
```

To check extra addition, deletion...

Git diff tool → p4merge



```
git push --set-upstream mygithub dev1
```



```
git config add user.email
" . . . . . . . "
```

```
git config --global -e
```

```
git clone repo-name
```