SESSION - 75

git reset

git revert

git squash

git stash

git cherry-pick

cd ../git-dosa/

git checkout main

--> i want to create one new branch

git pull origin main

--> Now you need to create new patch

--> gir checkout -b panner-dosa

#panner-dosa

\*dosa batter

-->git add . ; git commit -m "dosa batter added" ; git push origin panner-dosa

#panner-dosa

\*dosa batter

\*panner

--> git add . ; git commit -m "panner added" ; git push origin panner-dosa

#panner-dosa

\*dosa batter

\*panner

\*Masala Curry

--> git add . ; git commit -m "masala curry" ; git push origin panner-dosa

#panner-dosa

\*dosa batter

\*panner

\*Masala Curry

\*kaju paste

--> git add . ; git commit -m "kaju paste" ; git push origin panner-dosa

#panner-dosa

\*dosa batter

\*panner

\*Masala Curry

\*kaju paste

\*ghee

--> git add . ; git commit -m "ghee" ; git push origin panner-dosa

#panner-dosa

\*dosa batter

\*panner

\*Masala Curry

\*kaju paste

\*ghee

\*Garam masala

--> git add . ; git commit -m "Garam masala" ; git push origin panner-dosa

#panner-dosa

\*dosa batter

\*panner

\*Masala Curry

\*kaju paste

\*ghee

\*Garam masala

\*Tomato Onion

--> git add . ; git commit -m "Tomato onion" ; git push origin panner-dosa

--> git checkout main

--> git branch -D panner-dosa (delete branch)

`

--> Wherever we writing code that is Workspace(v.s)

--> code moving to staging area --> local repo

--> remote repo --> when code will go to public

git reset/revert -> undo the changes you already commited

reset --> undo the changes before you commit to remote

soft mixed hard

soft --> when you do soft reset, you get the changes into staging area

mixed (default) --> deletes from local commit, staging area and keep it in workspace

hard --> deletes from local commit, staging, working area

--> git reset --hard HEAD-3

--> git log --oneline

--> git status

if changes are already into remote branches

--> git push origin onion dosa

--> after pushing they did some changes

#panner dosa

\*Dosa batter

\*panner curry

\*carrot onion

\*tomoto and panner scrambled

\*ghee

\*chicken masala

--> git add . ; git commit -m “checken masala”;

--> git push origin panner-dosa

--> git log --oneline

#panner dosa

\*Dosa batter

\*panner curry

\*carrot onion

\*tomoto and panner scrambled

\*ghee

\*chicken masala

\*garam masala

--> git add . ; git commit -m “garam masala”;

--> git push origin panner-dosa

--> git log --oneline

--> git revert upto ghee

revert the changes --> but it will not delete anything, it will add the changes as new commit on top reverted commit

1 feature -> 1 commit

if a single person on one branch, you can squash the multiple commits into single commit.

git squash == git rebase -i

not recommended to do in shared branches

--> git rebase -i 04e1e6f

--> git push origin panner-dosa

-->git push origin panner-dosa -f

**git stash**

**=========**

when you are working on something in new feature branch, you got an emergency defect. We need to create hotfix branch immidiately and do the changes

--> git status

--> git checkout main

git stash --> save the changes into some temp without commiting

git stash pop --> get the changes from temp area to workspace

--> git stash

--> git status

--> git checkout main

--> git pull origin main

--> git checkout -b hotfix-em-1.3

--> git checkout panner-dosa

--> git stash pop

--> git add . ; git commit -m “panner dosa completed”; git push origin panner-dosa

--> git checkout main

--> git pull origin main

--> git checkout -b ghee-karam-dosa

--> git log --oneline karam-dosa

**git cherry-pick**

**==============**

I am developing feature, but I found some parts of the feature already developed in another branch,

--> git add . ; git commit -m “dosa batter and ghee added”

--> git cherry-pick (take karam dosa started and first one)

--> git status

--> git add .

--> git cherry-pick --continue

--> git add .

--> git cherry-pick --continue

--> git add . ; git commit -m “ghee karam done”; git push origin ghee-karam-dosa

-->

**Continous Integration**

**=====================**

What is CI --> It is the process integrating the code into artifact/image continously, means when developer do some commit, we can clone the code, install the dependencies, we can build the code, we can scan the code, we can unit testing finally we will build it as image

**Jenkins**

**----------**

Jenkins is popular CI tool. The real power jenkins lies in plugins.

sudo curl -o /etc/yum.repos.d/jenkins.repo \

https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

# Add required dependencies for the jenkins package

sudo yum install fontconfig java-21-openjdk jenkins -y

sudo systemctl daemon-reload

sudo systemctl start jenkins

sudo systemctl enable jenkins

ns