**DEVOPS**

>>DEVOPS is nothing but reduce the time between DEV team and OPERATION team.

build---->release------>production

here we have four servers like DEV-server , QA-server , UAT-server , PROD-server

>>once we build the source code packages will be created that packages first we will deploy into DEV-server if everything is fine then we will deploy into QA-server (QA-server is nothing but testing environment) if there is no any issues then we will deploy into UAT-server if everything is ok here we will deploy into PROD-server ( PROD-server nothing but it is a live environment )

GIT :

>> we have many features in git comparing to other tools like SVN, CVS

>>to initialize git repository first we need to hit the command is **git init.**

if you hit git init command here one .git directory will be created, .git is a hidden file

>>if we want to add multiple files we have to hit a command , git add . or git add \*

git status

git commit -m "mm"

git push origin master

>>suppose now we created 3 files like f1 and f2 and f3 i will add three files using the command is , git add . or git add \* ( here \* or . is a lable message ) , Now my requirement is i should commit only f1 file into local repository , the command is **git commit -m "message" f1**

**git commit -m "message" f1**

**git status**

**git log**

>> whenever we do commit , commit id will be created, suppose you commited 10 files at a time commit-id will be created in that commit-id 10 files will be there.

**>>How to check how many files are there in commit-id ?**

the command is , **git show commit-id**

>>if we hit **git log --oneline** command it will display only 7 characters of commit-id and we will get label message also.

>>suppose i have 100 logs now i want to check only 10 logs the command is **git log -10**

>>**git log --oneline -10** it will display in short farms

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

**git init**

**git config --global user.name "devops"**

**git config --global user.email "devops@gmail.com"**

**git config --list**

**touch file1**

**git status**

**git add file1**

**git status**

**git commit -m "message"**

**git push origin master**

**git log ---------- it will display logs**

**git log --oneline**

**git log -n or git log -10**

**git log --oneline -n ------> git log --oneline -10**

**git log --online --decorate**

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

>> Number of developers will be pushing the code into repository different different authors will be checkin the code into repository if we want to check the logs ,author wise we can check , command is

**git log --author or git log --author=authorname**

>>Now if want to check logs from past one year the command is

**git log --since=yy-mm-dd**

>>if you want to check the logs till last week then hit a command is ,

**git log --until=yy-mm-dd**

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

**RESET:**

>>reset is nothing but , here we have three stages like

**workstation ------> staging area ------> local repository**

now whatever the content is there in local repository if i want to rollback into staging area the command is ,

**git reset --soft commit-id**

**local repository -------> staging area**

>>now whatever the content is there in local repository if i want to rollback into staging area the command is ,

**git reset head filename**

**staging area ------> workstation**

>>suppose your file is there in local repository now if you want to reset the files into staging area the command is ,

local repository -----> staging area

**git reset --soft commit-id**

>> to reset the files from staging area to workspace the command is ,

staging area -----------> workspace

**git reset head filename**

**git reset head \* -------------------->** To reset multiple files at a time

>>Directly we can't add files from workstation to local repository

**workstation --------> local repository**

>>but we can reset files from local repository to workstation

**local repository ----------------> workstation**

**command is ,**

**git reset --mixed commit-id**

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

**GIT STATUS :**

to check files and code whether it is there in staging area or workspace

-----------------------------------------------------------------------------------------------

**RESET :**

suppose we have ,

commit 1

commit 2

commit 3

i want to reset commit 1 then i have to hit a command , git reset --soft commit 2

i want to reset commit 2 then i have to hit a command , git reset --soft commit 1

>> once we reset the commit id that commit id also will be deleted.

to check , $git log

>> now i want to reset files which are there in commit 2

what i will do is , i will hit command $git show commit 2

files will be displayed like, file1 file2

git reset head file1 ,

>>if you want to reset all files which are there in commit id the command is ,

**$git reset head \***

**to check , git status**

**>>If i want to rollback all commits the command is ,**

**git reset --mixed readme file commit id** --------------> readme file commit id will be created by default.