**GIT and GITHUB**

1. version control

2. version control tools

3. github and git

4. case study : dominion enterprises

5. git feautyre

6. git operation and command

**version control**

1. version control is like a snap shot data stored in which it explained what changes

you have creted and modifies

2.snap shot is verticular version of chane at one moment of time

3. version control is different version of changes.

4.you can make changes in the previous version .

**Why version control**

1. It save the chose as different persion associated with it will be able to see the

Changes another one make.

1. No need to assign name as it do it for you.
2. Back up in case if your central server crashes data will always available in local server\computer or machine .
3. Analysis : you can track what changes been made when was it made and time it took.

**Version control tool**

1. GIT
2. Subversion (svn)
3. Cvs
4. Mercurial

Out of these svn an cvs are assentralise version control toll they don’t provide local copy to all members.

Mercurial is same as git

Git is what helps you to make changes in.

Git hub is server where evey this is who provide everything.

Colebrater is developer

Centralize and distributed system

Benefits distributed have back up and speed is good as you don’t need internet all the time

You only need it to push and pull data.

How local repository is connected to cloud repository.?

**Git hub case study dominion enterprise**

About a company who uses this platform fot his company. He also have used Jenkins.

**What is git**

1. Git is a distributed version control tool.

**Features of git**

1. Distributed
2. 2. Compatible
3. Non lenier .
4. Branching
5. Speed
6. Open source
7. Reliable
8. Economical
9. Secure
10. Lightweight

**Distributed**

1. Allows distributed development of code.

2. every developer has a local copy of entire project

**Compatible**

1.Git is compatible to existing system and protocol

2. Svn and svk repository is directly excessible

3. so migration is easy to git

**Non linear**

Its easy to create branches and merge branches

**Leightweight**

It compress data to non losable technique

**speed**

its 100 times faster to fatch data from server repositeryy

**open sources**

open sources and free to use

**Reliable and secure**

Its safe as it uses SHAI to name and identify objects and tells when the commit is made you also have to checksummed for each commit.

**Economical**

**Release under gpl licence**

**What is repository**

Storage space where your data live.

Central repository

Local repository

**Git operation and command**

Four types of operation mainly

1.creating repository : git init

2. Syncing repositories : add origin , push, pull

3.Parellel development : branch, merge and rebase

4.making changes : status ,add and commit

**Create repositery**

Click bash in folder

Steps

1.git init

2. git add origin<Link> : git remote add origin “link” to link to remote

3.git pull origin master

4. git status

5. git add “file name” if add multiple file use git add -A

6. git commit –m ”name”if want to add multiple commit use git commit –a –m “msg”

7. git add –A to add

8. git commit –a –m to commit

9. git log to see all data stored

10. git push to upload

11.git –amend

12. git push –u origin

13.ls’to view list

**Parellel development**

Steps involve

First step is to branching.

Branching : to add new branches

1. Git branch
2. Git branch branch
3. git push --set-upstream origin firstbranch

Merging :merge two project

1. git checkout branch name
2. cat branch name
3. git push --set-upstream origin firstbranch

Rebasing:kind of merging

1.git rebase master

**Git push**

1. ssh-keygen
2. ssh – T [git@github.com](mailto:git@github.com)
3. cat name of file
4. git push origin firstbranch