[Course: Git Course | DevOps School LMS](https://www.devopsschool.xyz/course/view.php?id=15)

What is Git?

* It is source code management tool
* Versioning tool of code

What is code?

* Code which is readable. And read the content.
  + CSS, java, python, java script, html

What is not code?

* Images, pdf files, movie files, doc files

What is versioning?

* I want to know some information on the code.
  + Like when this was code was written?
  + What was written?
  + Author of the code?
  + Purpose of the code?
* Manually if u want to maintain versioning -> rewards.java, rewards1.java
* This is not good practice.

What is distributed vs centralized code management?

* Git is truly distributed

What is client sever?

What is distributed?



* + Every individual machine maintains repository in local machine
  + How all these individual machine repos collaborate each other.
  + Repository holds the content. There is a process or use or modify the content is called git.
  + What all operations in git ?
    - Add
    - Modify
    - Delete
    - Merge
    - Fork
    - Branching
    - Clone
    - Fetch

We need not to have internet and manage repository is called distributed. But client server architecture it wont work like that, we need server for our operations.

**Git Workflow**

What does mean by git workflow?

* Add a file
* Change a file
* Delete a file
* Version a file
* See a history

Basic workflow in git?

1. Create a repo
   1. Git init
2. Add a file in workspace using editor
   1. Touch f1 f2
3. Add a file from working directory to staging area
   1. Git add f1 f2

Before you commit, you need to pass who is creator of file. Otherwise we cannot track back.

You need to do one time activity.

[one-time]

Git config user.name=”ram”

Git config user.mail =”ram@gmail”

1. Commit a file to git [commit file from staging area to repository]

Git commit -m “commit file1”

1. See a history

Git logs

Note: this is exactly we do in entire life time. At least I have done 100 times in last year.

What is 68885dc0237f4916b8e1ec068fbdbcb69cd32ef5 ?

* A 40 char length universal uniq id
* Called checksum aka commit-id aks version
* SHA1
* Link to the CODE which you commited in repo
* Size of each commit can be anything.
* Each commit id has parent id
  + 1 is parent of 2, 2 is parent of 3.
  + Every day one id create on top of existing commit id. Like this way relationship gets created.
* This is object storage technology. Like
  + S3
  + Google drive
  + git





* There are 3 areas ->
  + Working dir
    - A place where you change the code
    - This is working dir -> /d/devops/Batch6/code
  + Staging dir
    - A place where you see difference between workspace and repo.
    - Help you to send a file from workspace to repo. You will get a chance to review what you are going to commit.
    - Where is stage place -> /d/devops/Batch6/code/.git/index
    - **Git status** command this is to use to query staging area details.
  + Repository
    - A place where you store the version after commit
    - This is repo -> /d/devops/Batch6/code/.git/objects

**Next set of requirement:**

1. How to rename file?
2. How to commit a directory?
3. How to move one file from one location to other location?
4. How to delete files?
5. How to delete dir?
6. How to get back to the stage of before deleted file?
7. How to add all files but commit only selected files?

Revert to previous version of commit. Git checkout <commitid>

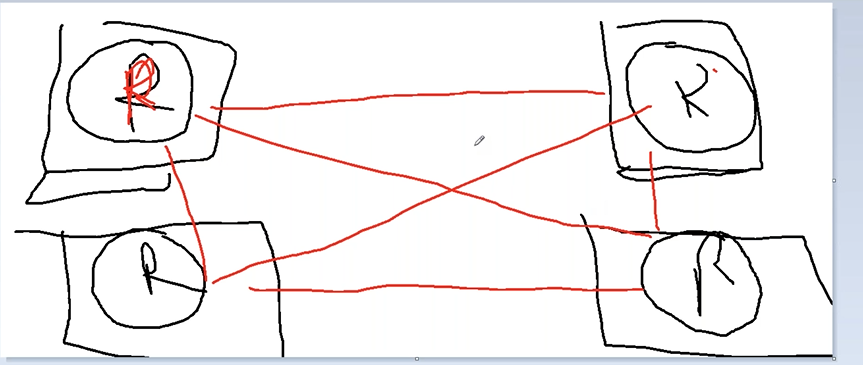


Q: Never checkout and modify changes on old commits. We need to use branching concept.

**Repo (Fully committed version)**

Everyone has local repo is different. Then what’s the problem in this approach.

1. How to share my code with other team members
2. How do we collaborate with each other ?



Git allows to connect to any repo using ssh / http.

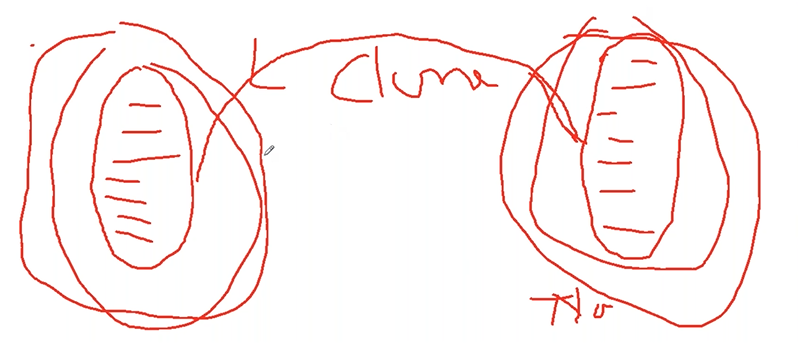
* ssh – remote machine should be enabled with ssh and network
* http – remote repo machine should be enabled with http server and network

terminology ->

1. my repo is local and any other repo outside is called remote repo
2. there are two imp concepts ->
   1. push -> Ram want to push his changes to Mahesh
      1. to send local repo commit objects to remote repo
   2. pull -> Ram want new features of Balu then he has to pull
      1. to get commits objects from remote repo to local repo

**level1 -> node1 syncup with all other nodes. By clone, push/pull mechanism. But it is very tied approach.**

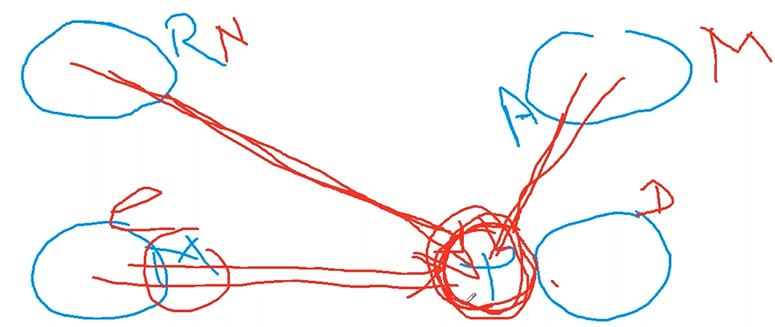
**clone copy**



**Basic Workflow working with remote repo**

1. first time left side (repo) has n number of commits. Once the right side repo in contact, then it doesn’t have any commit objects.
2. So you need to do clone operation on target system to get all these objects / all repo dump right side
3. Later clone, if any additional commits happened at left side repo, then either pull (by B) or push (by A) to each other to get in sync. Both side push and pull continues happen both side based on the changes happening both side to become collaborate and sync each other.

**Level2 -> All nodes in network syncup with only one node. But the common node cannot run for longer. The node is overloaded and cannot overload one person for longer. The single person is overloaded for doing all syncup process of other requests coming from other persons.**

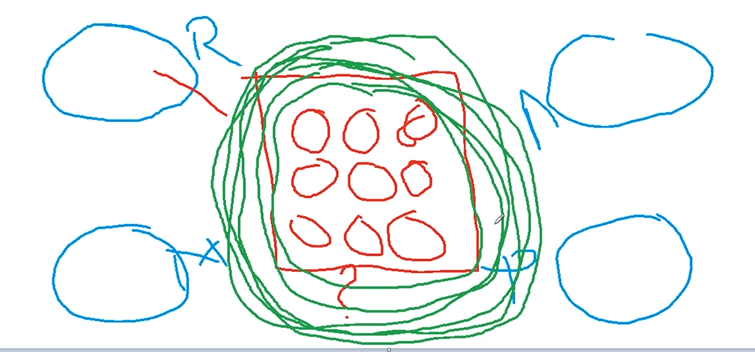
****

**Leve3 -> we all 4 people , we took one laptop and created a repo. And that machine is 24 \* 7.**

**Leve4 -> we all 4 people , now we are looking for multiple repo in same machine and managing the repos.**

**Enterprise looking for**

1. **Hosting platform**
2. **Has remote repo**
3. **Has central repo**
4. **WIKI Integ with (remote and central repo)**
5. **Issues tracking**
6. **Mul org and Mul projects**
7. **Role based access management**
8. **Ldap authentication**
9. **GUI tool**
10. **Ability to manage mul repos**

****

**Some of the repo management tool for enterprise level came for managing these -> Github, Gitlab, bitbucket, Amazon, google, Microsoft**

**Now the central server is called as github / gitlab etc.,. It is remote repo.**

**These enterprise versions are paid version.**

**Cloud - > hosted for public - free + paid**

**Enterprise -> hosted for your company within firewall + Paid**

**Workflow working with Github**

**User - 1**

1. Register at github site
2. Create a empty repository in GUI

https://github.com/kramsagar/test.git

1. Push your existing repo

git remote add origin https://github.com/kramsagar/test.git

|  |
| --- |
| $ git push https://github.com/kramsagar/test.git master  Enumerating objects: 39, done.  Counting objects: 100% (39/39), done.  Delta compression using up to 8 threads  Compressing objects: 100% (32/32), done.  Writing objects: 100% (39/39), 2.92 KiB | 2.92 MiB/s, done.  Total 39 (delta 14), reused 0 (delta 0), pack-reused 0  remote: Resolving deltas: 100% (14/14), done.  To https://github.com/kramsagar/test.git  \* [new branch] master -> master  Dell@DESKTOP-7TUUEEC MINGW64 /d/devops/Batch6/code (master)  $ |

**User-2**

1. Register at github site
2. Clone the team repo
3. Add files to git repo
4. Commit to local repo
5. Push ur code to remote/central github repo