intro

Friday, February 4, 2022 5:38 PM

What is GIT?
It's a version control system
It follows distributed architecture.

Step 1: create a workspace or playground

What is VCS?

- Used to track files the project history.
- To keep track of multiple versions
- It also helps in resolving conflicts.
- Conflicts: Problems that occur when we merge the files.

Two types of VCS
Centralized and distributed

Centralized VCS	Distributed VCS
All developers are connected to one central server	
No security.	
We get the latest copy of code	

Disadvantages:

- Single point of failure
- High dependency on one central server
- Huge bandwidth consumption
- No collborations, no snapshots os project.
- Need to wait until server comes online.

Distributed version control system

- Every developer has copy of the project.
- Every developer can interact with each other.
- Collaboration exists
- Ex: Git, Mercurial

GIT- Global information technology
Git Is a free open source, distributed version control system
GIT BASH is a terminal to execute git commands

GITHUB is a web service where we can upload/share the code.
We can make the code as open source for community of developers
GITHUB is a cloud based hosting service

STEPS of GIT

1. Workspace
2. Staging
3. Commiting
4. Push
5. Pull

1. Workspace
   A workspace has all the untracked files and are created by developer.

Converting of untracted files to tracted files is called staging.

Ppt

Sunday, May 8, 2022 7:40 PM

GIT/GITHUB

What is GIT

- It's a **VCS.**
- It Follows the **distributed architecture**.
- Workspace : cricket has playground, tennis has tennis table.
- **Application**: collection of lots of files.

  C:/folder

  D:
- **Workspace:** Selected dedicated folder/directory to store a particular project.

  It's a place where untracked files are created by developer.

VCS

VCS:

- Used to tracking files and the project history.
- We can work together.
- To keep a track of multiple versions.
- It also helps in resolving conflicts.
- **Conflicts**: Problems that occur when we merge the files

- **Two types**:

1. Centralized and
2. Distributed

Centralized VCS

Centralized:

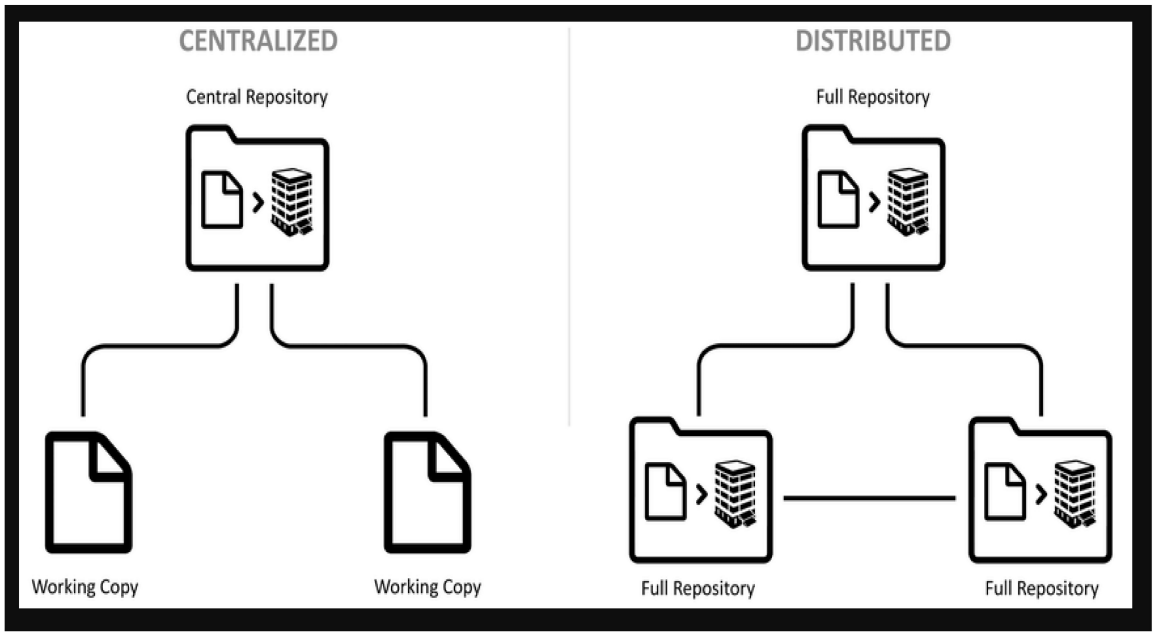
- All team members connect to a one central server.
- We can get the latest copy of code.
- Ex: subversion, Microsoft team foundation server.

Disadvantages:

- Single point of failure.
- High dependency on one central server
- Huge bandwidth consumption
- No Collaboration, No snapshots of project.
- Need to wait until server comes online

Distributed VCS

- **Distributed:**
- Every team members has a history and copy of the project.
- Can save the snapshots locally and connect with team.
- Ex: Git, Mercurial.



GIT

- Free, Open Source
- Super fast
- Scalable
- Cheap and Easy branching and Merging.

Using Git:

✓*Command line (fast)*

✓*Code Editors or IDE's*

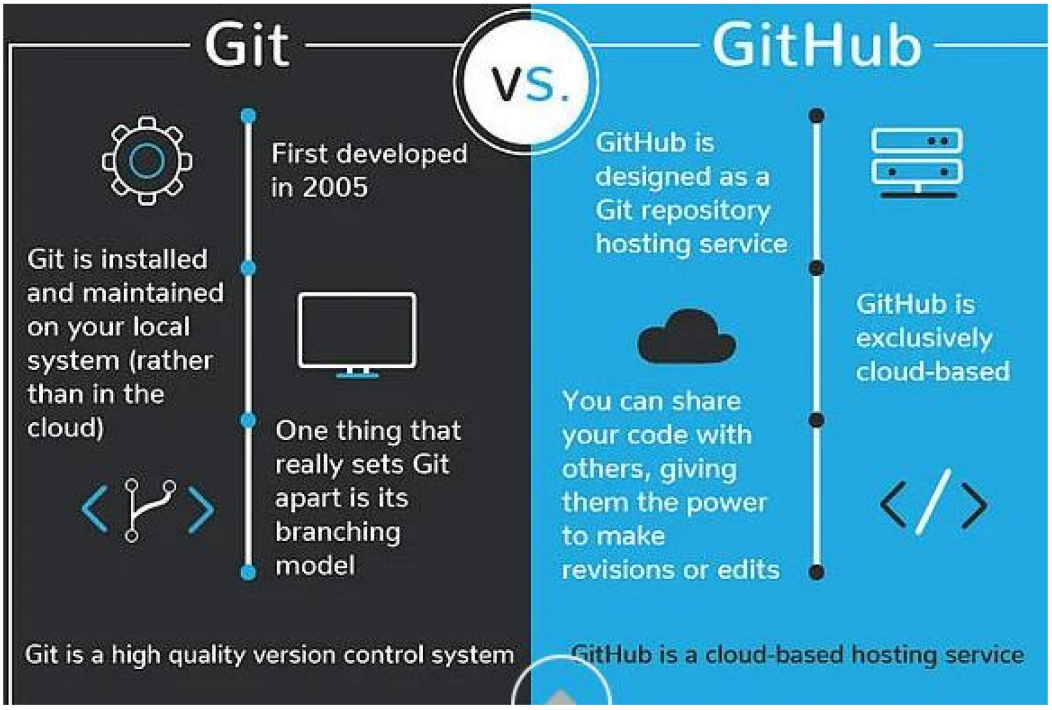
Git Installation:

To check version: git –version

Git bash : terminal where we write all git queries

GITHUB

- It's a webservice where we can upload/share the code.
- We can make the code as open source for community of developers
- Working with team the the code can be easily pushed/pulled/reviewed.



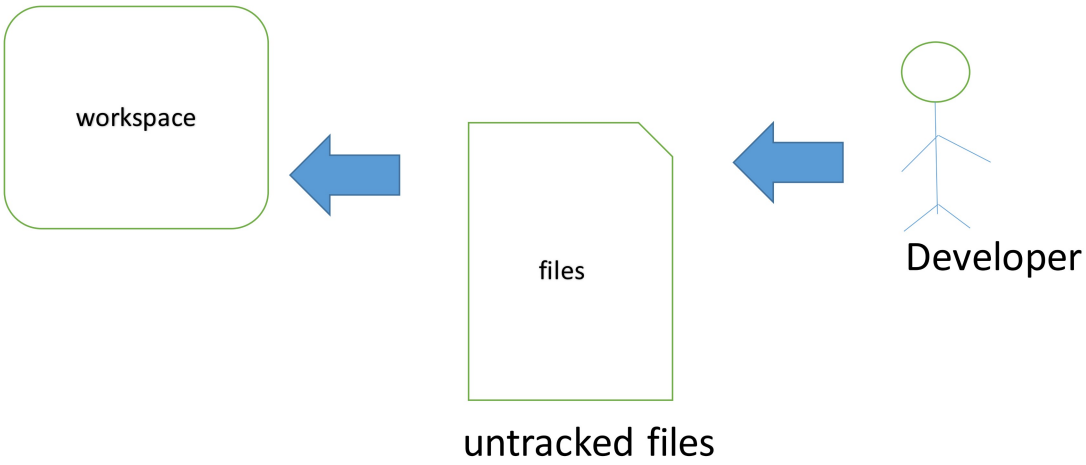
git	GitHub
1. It is a software	1. It is a service
2. It is installed locally on the system	2. It is hosted on Web
3. It is a command line tool	3. It provides a graphical interface
4. It is a tool to manage different versions of edits, made to files in a git repository	4. It is a space to upload a copy of the **Git** repository
5. It provides functionalities like Version Control System Source Code Management	5. It provides functionalities of Git like VCS, Source Code Management as well as adding few of its own features

Steps of Git

1. *Workspace*
2. *Staging*
3. *Commiting*
4. *Push*
5. *Pull*

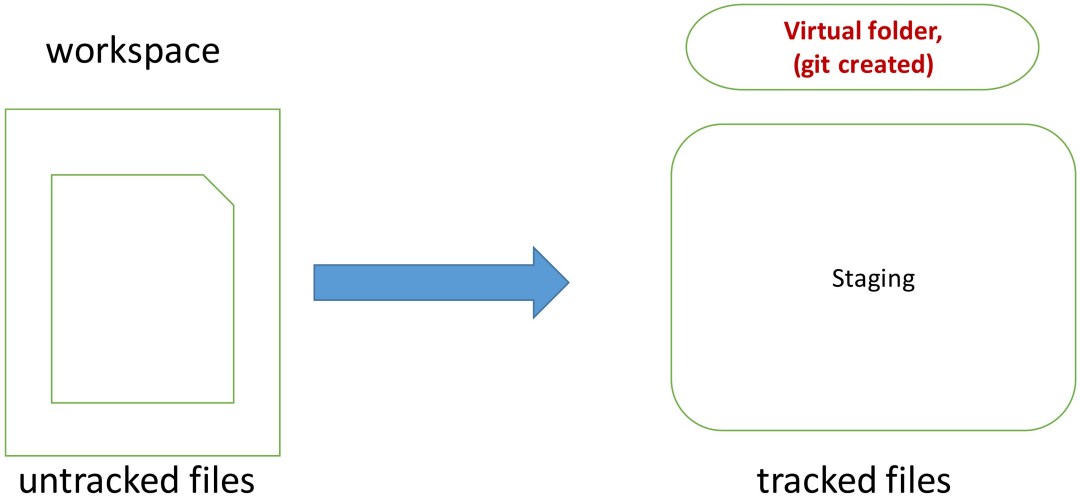
1. Workspace

- A workspace has all the untracked files and are created by developer.



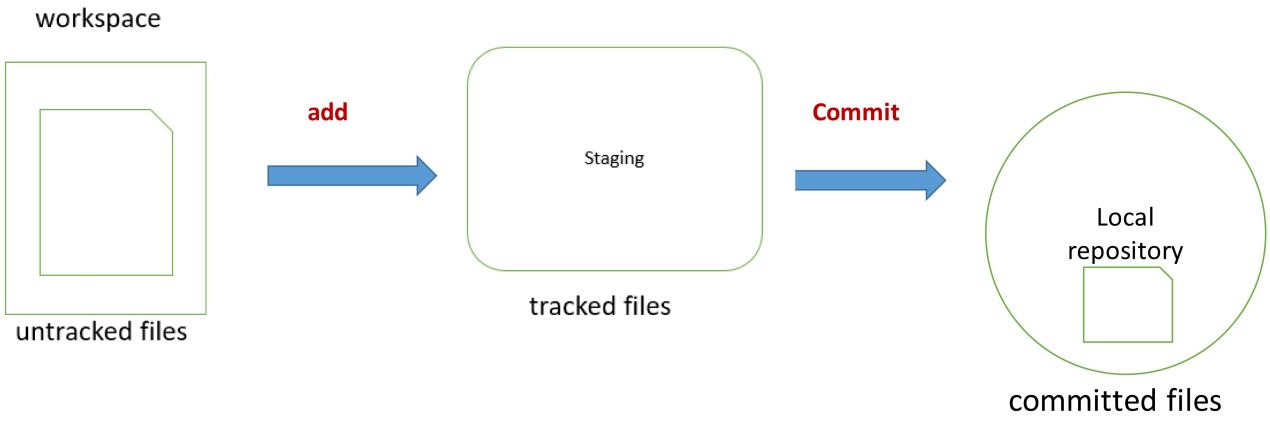
2. Staging

- It's a virtual folder which has a tracked files.
- we use the add command to make files as tracked files.



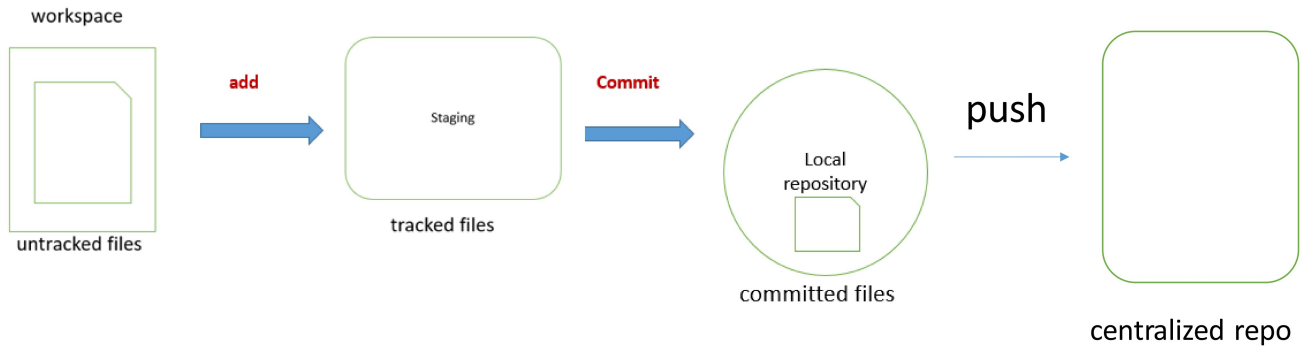
3.Commiting the files

- Git creates a local repository in workspace.
- Only after commit the tracked files are copied for local repository.



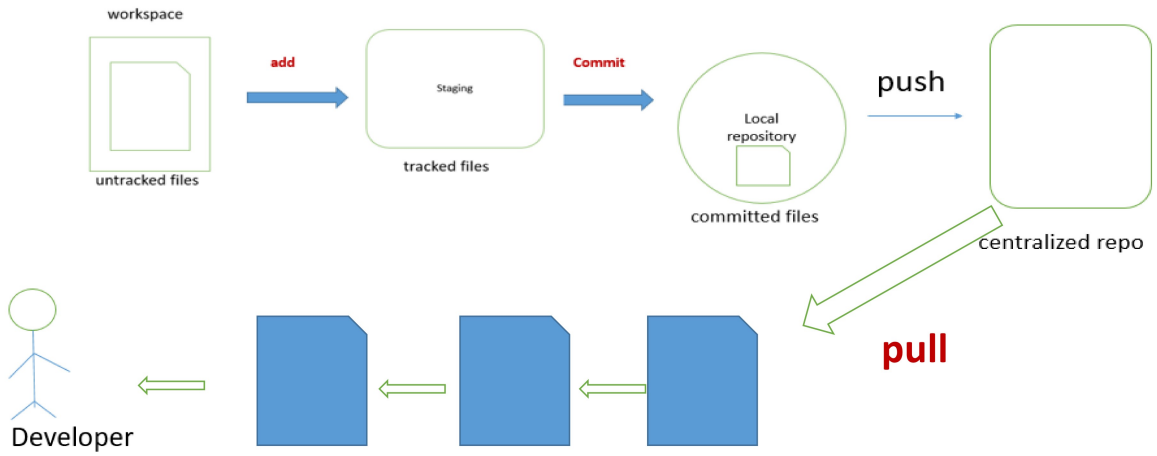
4. Push

- It is used to copy the files from the local repository to the centralized repo.
- In local repo we will be having 3 folders created.



5. Pull

- It copies a file from the Centralized repository into the local repository.



Basic Git Commands

1. **Configure username and email:**

git config - - global user.name "Anita"
git config - - global user.email Anita@gmail.com
git config - -list

2.Creating a git repository/ Initialize the git repository.

git init

3.To Add files/ to keep a track

git status → keeps the status of files added
git add. → to add all files in the project
git add "file name"

(Now file is added for staging area). Staging happens before the commit.

Basic Git Commands

3. Staging: virtual folder which has a tracked files., we use the add command to make files as tracked files.

4. **Commiting:** Git creates a local repository in the workspace**.**

And only after commit the tracked files are copied for local repository

git commit

git commit - m "my first message"

5.To see logs:

git log

6.To Push the code for the Remote Repository:

git push url_here

Basic Git Commands

7. to pull changes from a remote repo in Git

```
git pull
```

8. create a new branch in Git:

```
git branch branch_name
```

9. to switch to a newly created branch in Git:

```
git checkout branch_name
```

10. to list branches in Git:

```
git branch
```

11. to create a branch in Git and switch to it immediately

```
git checkout -b branch_name
```

Basic Git Commands

12. to delete a branch in Git.

```
git branch -d branch_name
```

13. to merge two branches in Git:

```
git merge branch_name
```

14. to add a remote repository in Git:

```
git add remote https://remote_repository _url
```

15. To clone a project from the remote repository

```
git clone url_here
```