**Version Control System:**

* Version control manages the changes in the project. It takes the snapshot of the changes occurred in project. These changes are known as versions
* These are called SCM (source code management) tools
* Before version control, the developer’s tasks were like individual. For example, developer A made some changes to one task and developer B might made changes on the same thing unknowingly. But with version control, there will be collaboration in team. So, the developer B can see the changes made by A.
* It allows multiple people work on same file
* Every change will be referred as version. That might be a web page or some other thing
* Snapshots of all versions are properly documented and stored.
* Versions are also named accurately.
* Apart from central server, every developer has the copy of data in their local machine. So, in case central server crashes, there will be always backup available in local machine. That is the one advantage
* We can analyse the project like what exactly was changed, when it was changed, how much time did it take and how the project evolved between versions
* Merging the code developed by different developers
* Security of the code

**Version control tools:**

1. Git
2. Subversion
3. CVS
4. Mercurial
5. IBM clear case
6. Microsoft TFS

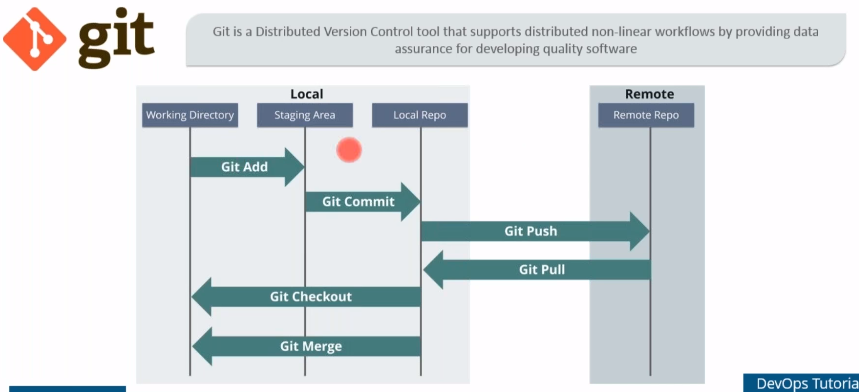
**SCM tools are two types:**

1. Centralized SCM tools (SVN)
2. Distributed SCM tools(git)

* In SVN we call it as checkout and in git, we call it as clone
* SVN also have command line tools like git

**Git:**

* Git is a distributed version control tool that supports distributes non-linear workflows by providing data assurance for developing quality software.
* Git is open source
* Git is developed by Linux open source community



* If a developer wants to make changes on local repository. He needs to do git add. Whatever is there is working directory will be present in staging area which is between WD and local repository. Once we are done with git add we can go ahead with git commit to make changes in local repository. Once that is done we can push the changes to main repository.
* After that we can pull it from remote repository and perform git checkout.

**Git service providers:**

1. GitHub
2. GitLab
3. Atlassian Bitbucket

**Git hub:**

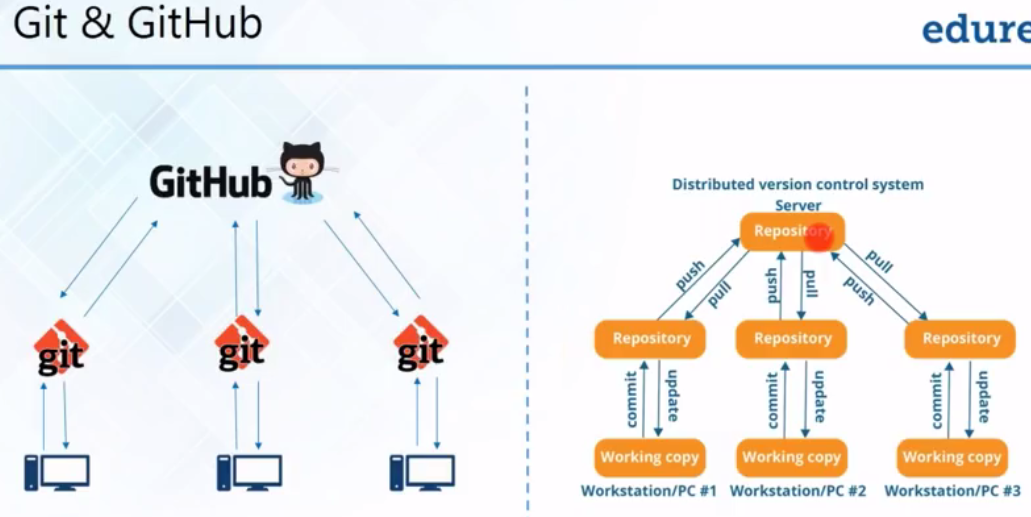
* GitHub is a website and service that we hear geeks rave about all the time
* Version control systems keep these revisions straight, storing the modifications in a central repository. This allows developers to easily collaborate, as they can download a new version of the software, make changes, and upload the newest revision. Every developer can see these new changes, download them, and contribute.
* We’ve established that Git is a version control system, similar but better than the many alternatives available. So, what makes GitHub so special? Git is a command-line tool, but the centre around which all things involving Git revolve is the hub—GitHub.com—where developers store their projects and network with like-minded people.

**Difference between GitHub and GitLab:**

* **Git -** A source code versioning system that lets you locally track changes and push or pull changes from remote resources
* **GitLab, GitHub, and Bitbucket -** Are services that provides remote access to Git repositories. In addition to hosting your code, the services provide additional features designed to help manage the software development lifecycle. These additional features include managing the sharing of code between different people, bug tracking, wiki space and other tools for 'social coding'.
* [**GitHub**](https://www.github.com/)is a publicly available, free service which requires all code (unless you have a paid account) be made open. Anyone can see code you push to GitHub and offer suggestions for improvement. GitHub currently hosts the source code for tens of thousands of open source projects.
* [**GitLab**](https://www.gitlab.com/)is a GitHub like service that organizations can use to provide internal management of git repositories. SESYNC has setup a GitLab server for our researchers and staff to better collaborate.



* Git hub means, git hosted on internet
* Gitlab means git hosted inside the organisation



* GitHub is a central repository and git is a tool allow us to create local repositories
* GitHub is a company that allows to host a central repository in a remote server

**Features of Git:**



**Distributed:**

* Every developer has a local repository and changes are copied from one to another

**Compatible:**

* If we are using some other version control system and wants to migrate to Git, we no need to copy all the files. We can simply install git and can access the files.

**Non-linear:**

* While creating the project, git always records the current state of a project in a tree diagram from index. Tree is a non-linear structure
* It includes various techniques to navigate & visualize non-linear development history.

**Branching:**

* Git is the only one has branching model. Git allows to have multiple branches independently and creation, deletion and merging the branches takes few seconds only
* Master branch contains all the data of project and contains production quality code.

**Lightweight:**

* Git uses compression technique to compress the data on client’s side.
* We might think that many users pushing the data causes issue. But git compresses the data and stores in central repository in minimal space after pushing.

**Speed:**

* Fetching data from local repository is 100 times faster than remote repository. Because we no need to travel over a network. Git is 10 times better than other tools

**Open source:**

* We can modify its source code according to our need.

**Reliable:**

* If the system crashes, we can easily recover it

**Secure:**

* Uses SHAI to name an identify the objects. It is very secure. Once we did any changes SHAI creates a commit# and we can say that the changes were not done by us because it has the details and everyone else can see that too. This one helps to revert

**Economical:**

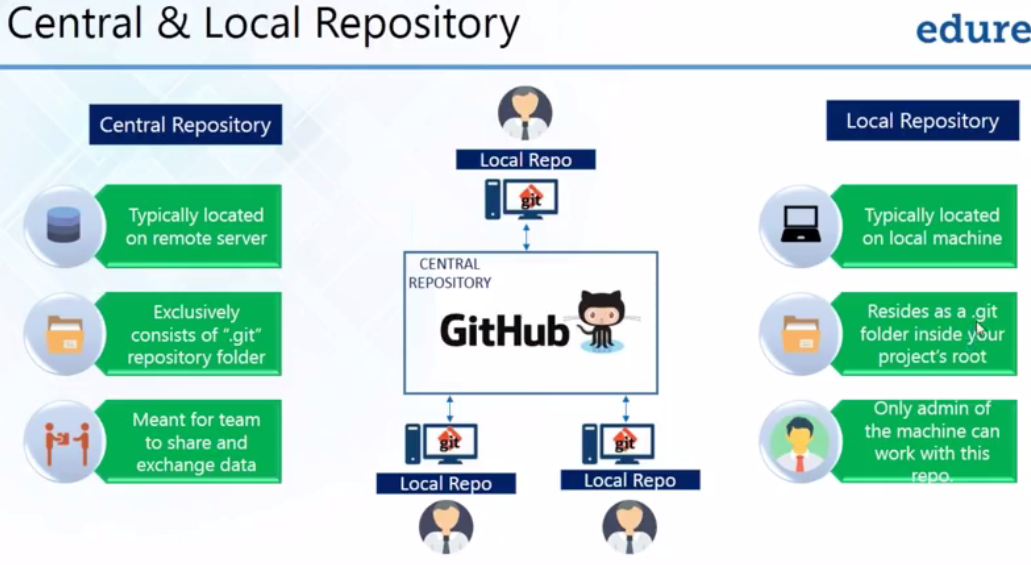
* Released under GPL’s (general public license). It is free

**Repository:**

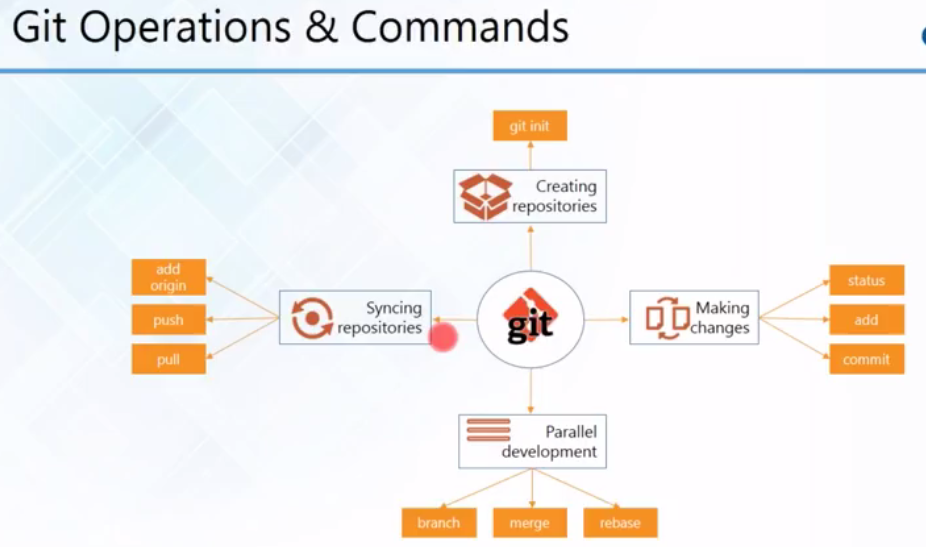
* Repository is a directory or storage where your projects can live. It can be a local storage on our computer or it can be a storage space on github or any other online host. We can keep code file, text files, image files in a repository

**Two types:**

1. Central repository
2. Local repository

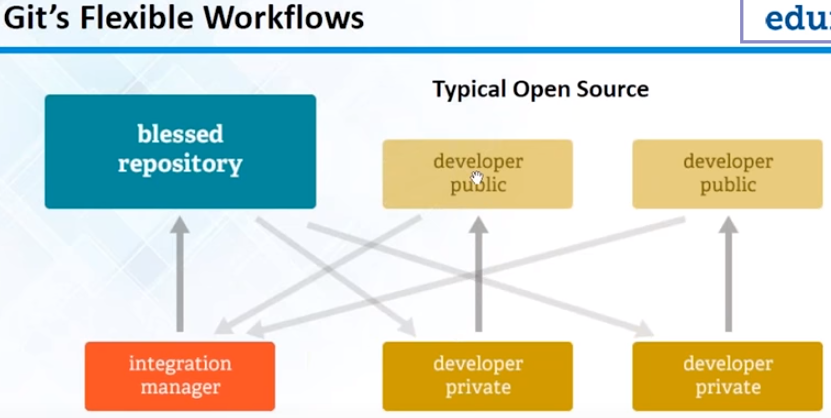


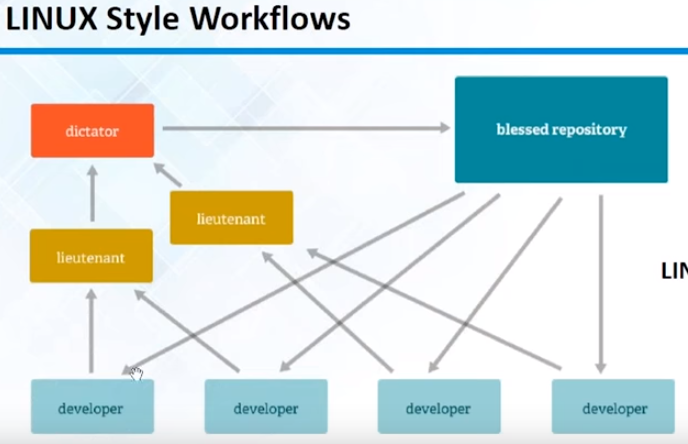
**Git operations & commands:**



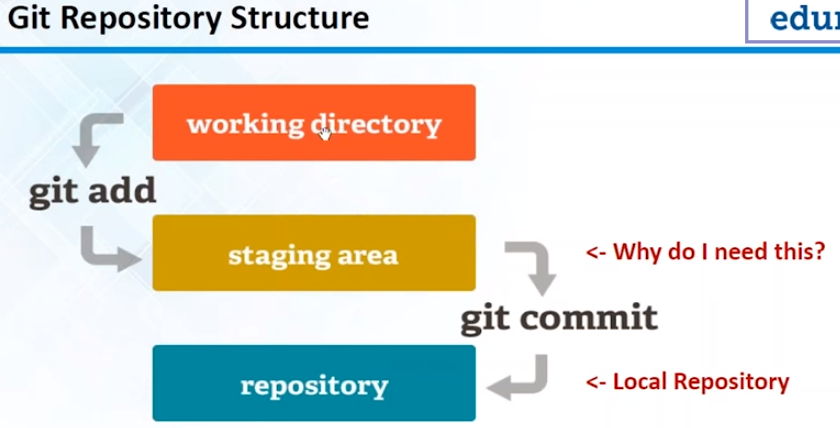
**Creating repositories:**

* Create central repository on GitHub
* Install git on your local machine and use “**git init**” to create your local repository
* Download or clone your repository from GitHub
* Client server version control systems work on a centralized model which has a single repository to which user’s check-in and checkout
* Git was initially developed by Linus Torvalds, he developed Linux as well
* Git can be used as a classic client server workflow





* We can pull from anybody, but pushing will involves code review and manger will push it



* We can set rules to authenticate git for particular user

**GitHub or Bitbucket:**

* Repository is nothing but project. It represents a project
* Readme file helps others to understand about the project and set up of the project
* Developers can mention about project in readme file, like how to import it to eclipse etc…
* It is recommended to have readme file in the project
* In SVN, we call it as check-in, checkout and in git we call it as clone, push
* SVN checkout won’t work as local server. It is just a folder. But git clone works as local repo
* Configuring name and email in SVN is not required
* If we are working on 10 projects, while configuring the username, if we give - -global, the username will be applicable on all 10 projects
* There might be 100 files in our working directory, and we want to commit only few files from that, so we can add only those few files to staging area and commit from there
* That’s the main advantage of staging area