
Edureka's Top 10 2021 Playlist
edureka!
&
Git and Github Tutorial
SUBSCRIBE

WHAT IS GIT?

Version Control System (VCS) for tracking changes in computer files

- ✓ Distributed version control
- ✓ Coordinates work between multiple developers
- ✓ Who made what changes and when
- ✓ Revert back at any time
- ✓ Local & remote repos

CONCEPTS OF GIT

- ✓ Keeps track of code history
- ✓ Takes “snapshots” of your files
- ✓ You decide when to take a snapshot by making a “commit”
- ✓ You can visit any snapshot at any time
- ✓ You can stage files before committing

BASIC COMMANDS

- **$ git init** // Initialize Local Git Repository
- **$ git add <file>** // Add File(s) To Index
- **$ git status** // Check Status Of Working Tree
- **$ git commit** // Commit Changes In Index
- **$ git push** // Push To Remote Repository
- **$ git pull** // Pull Latest From Remote Repository
- **$ git clone** // Clone Repository Into A New Directory

INSTALLING GIT

- ✓ **Linux (Debian)**

  $ sudo apt-get install git

- ✓ **Linux (Fedora)**

  $ sudo yum install git

- ✓ **Mac**

  http://git-scm.com/download/mac

- ✓ **Windows**

  http://git-scm.com/download/win

Agenda:

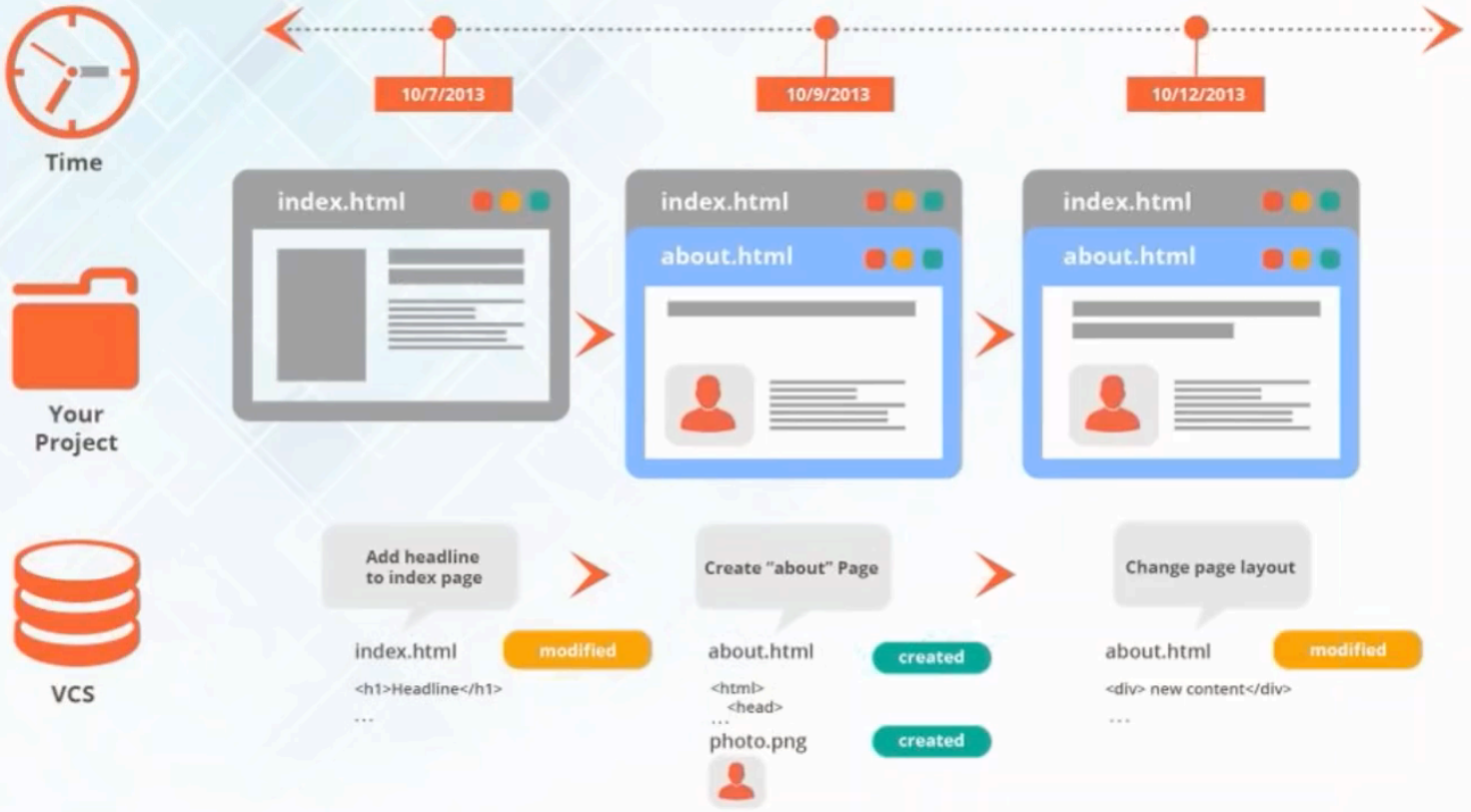
edureka!

- Version Control – What & Why?
- Version Control Tools
- GitHub & Git
- Case Study: Dominion Enterprises
- Git Features
- Git Operations & Commands


What is Version Control?
SUBS
CRIBE

Version Control System

edureka!

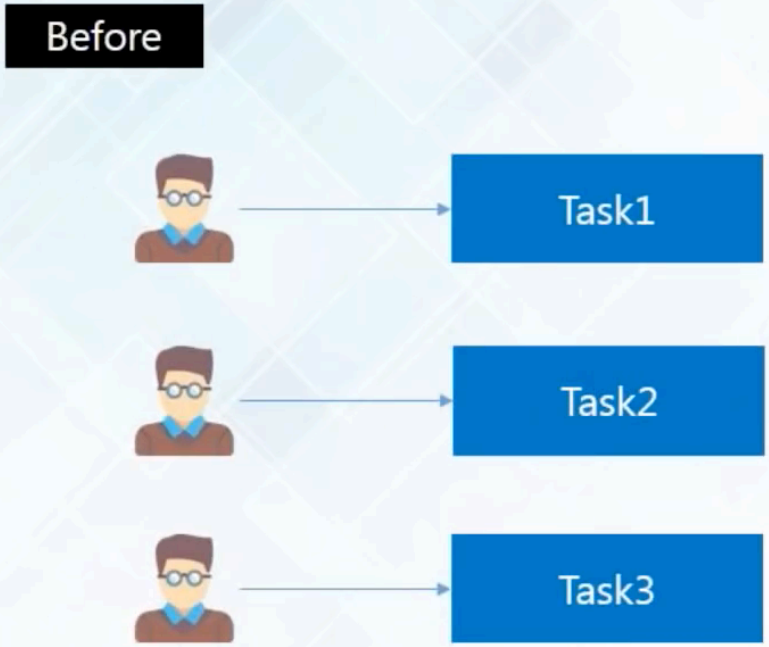


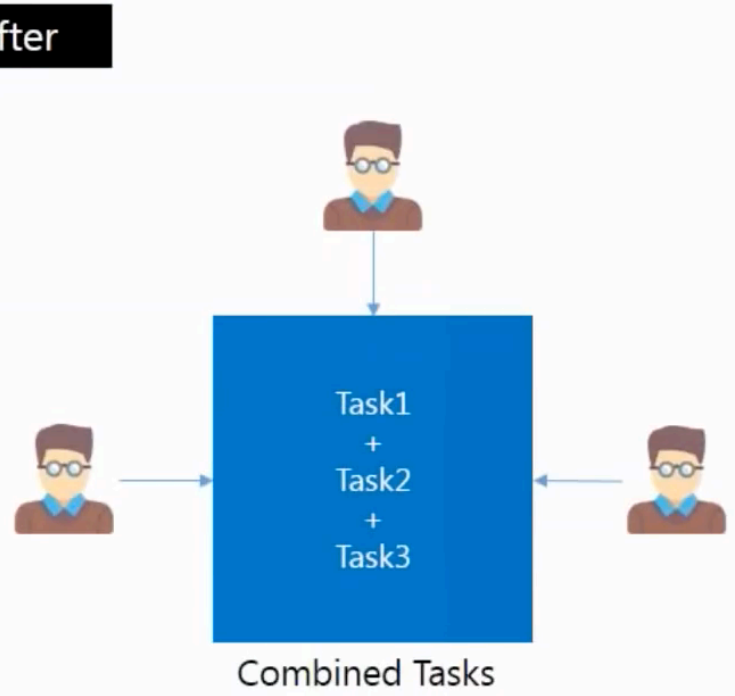
- Version control is the management of changes to documents, computer programs, large web sites, and other collections of information.
- These changes are usually termed as "versions".

Why Version Control?

Collaboration

edureka!





Storing Versions

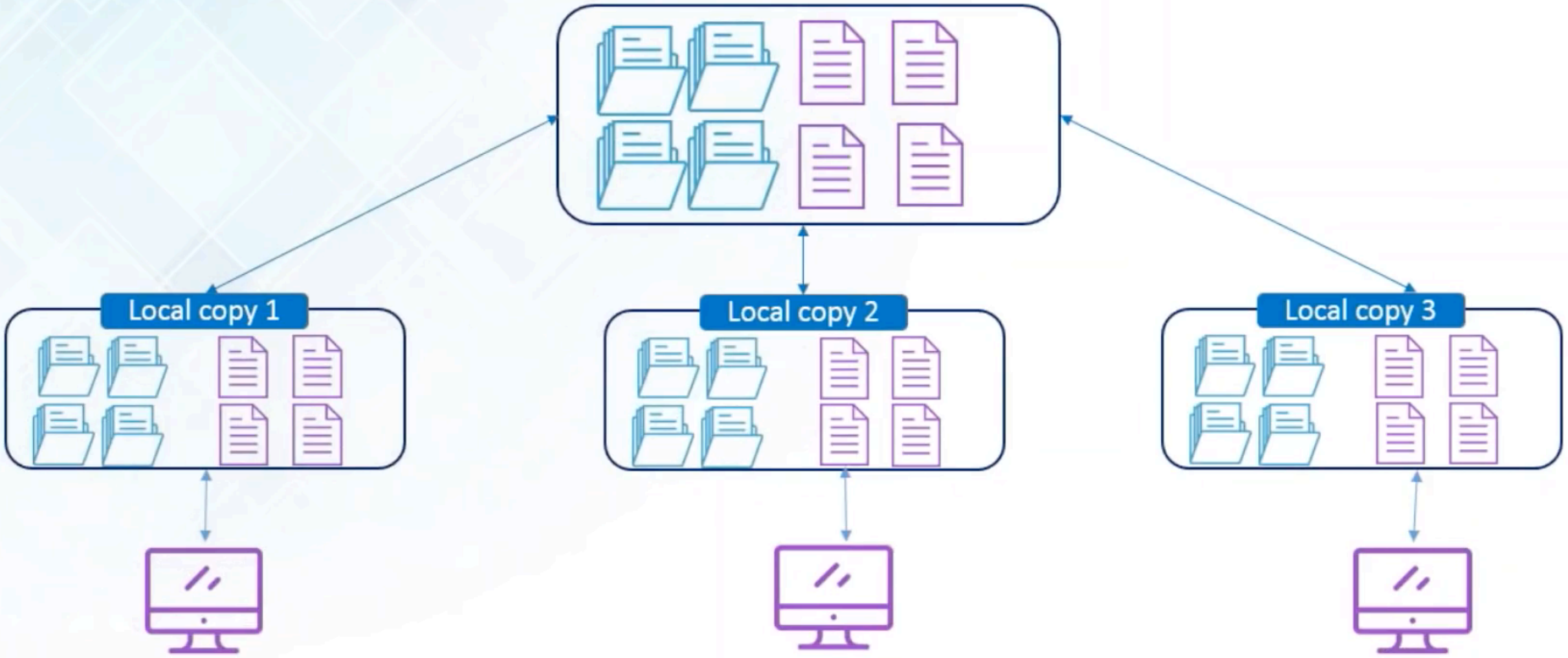
edureka!

- Snapshots of all versions are properly documented and stored.
- Versions are also named accurately.

Backup

edureka!

In any case if your central server crashes, a backup is always available in your local servers.



Analyze

edureka!

When you change version -

- VCS provides you with proper description
- What exactly was changed
- When it was changed

And hence, you can analyze how your project evolved between versions.

Version Control Tools

Version Control System Tools

edureka!

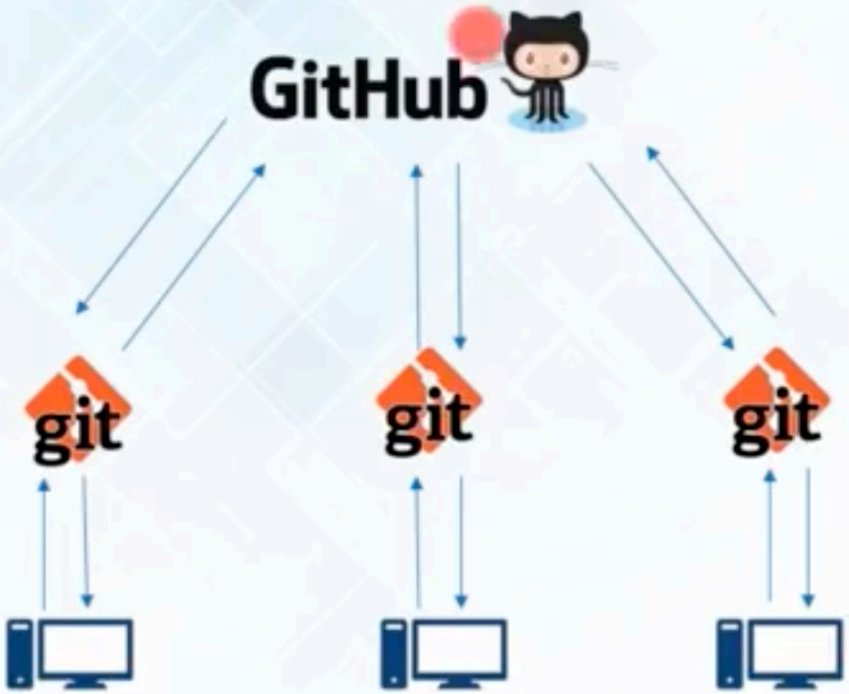
Interest Over Time Graph

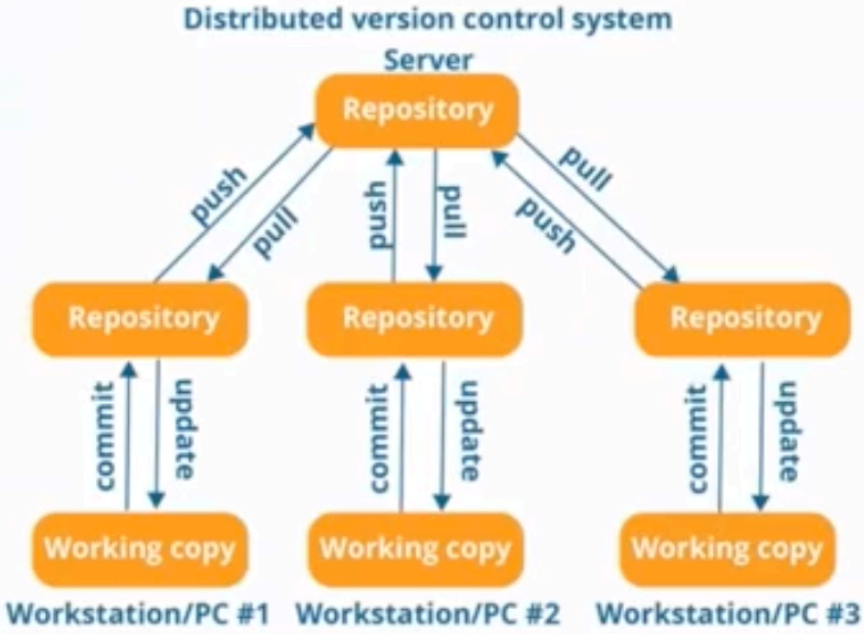
edureka!


Git & GitHub
SUBS
CRIBE

Git & GitHub

edureka!





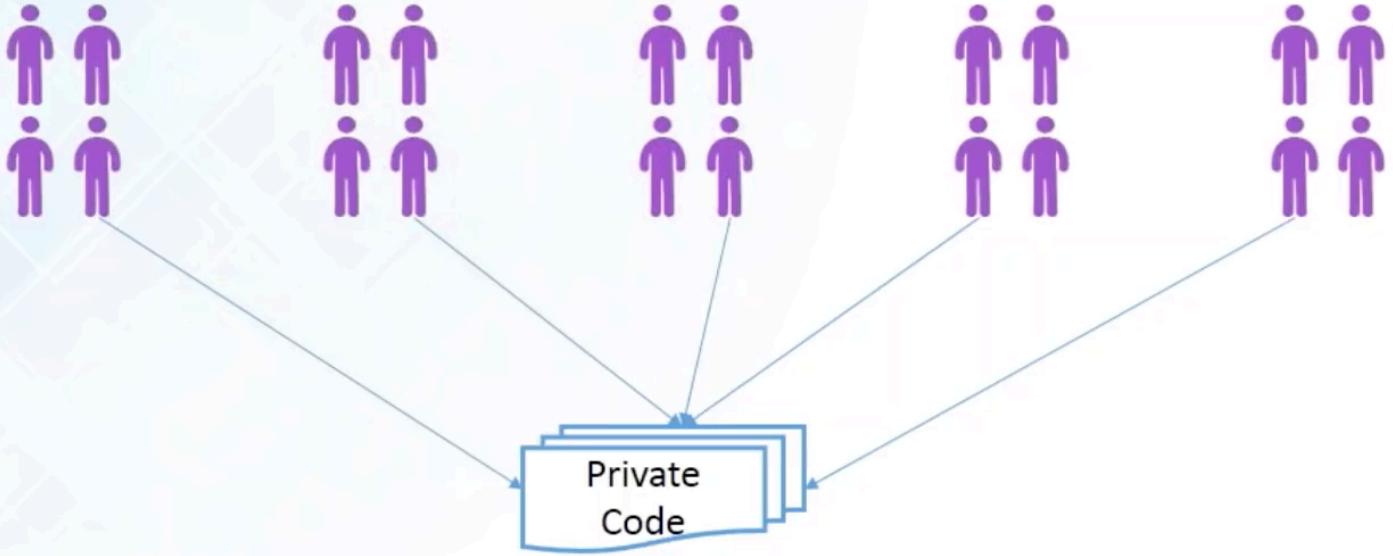
GitHub Case Study: Dominion Enterprises

Dominion Enterprises Case-Study

edureka!

Problem Statement:

Each team has its own goals, projects, and budgets and they also have Unique needs and workflows



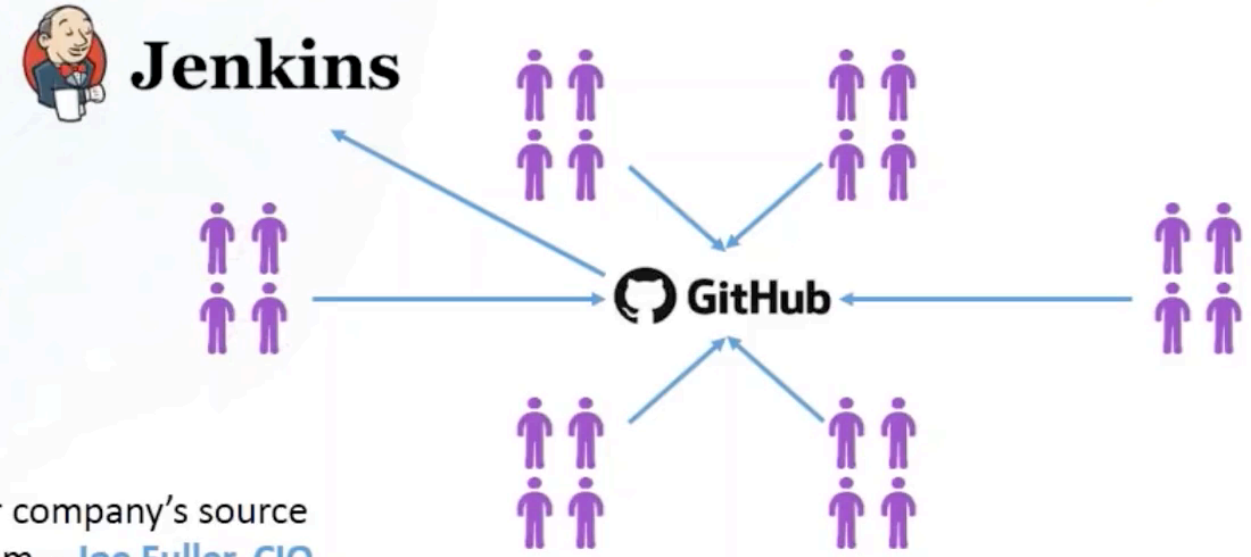
They wanted to make private code "publicly" to make their work more transparent across the company

Dominion Enterprises Case-Study

edureka!

Reason for using GitHub as the solution:

- ❑ They noticed that few of the teams were already using GitHub. Adopting a familiar platform has also made onboarding easier for new employees.
- ❑ Having all of their code in one place makes it easier for them to collaborate on projects.



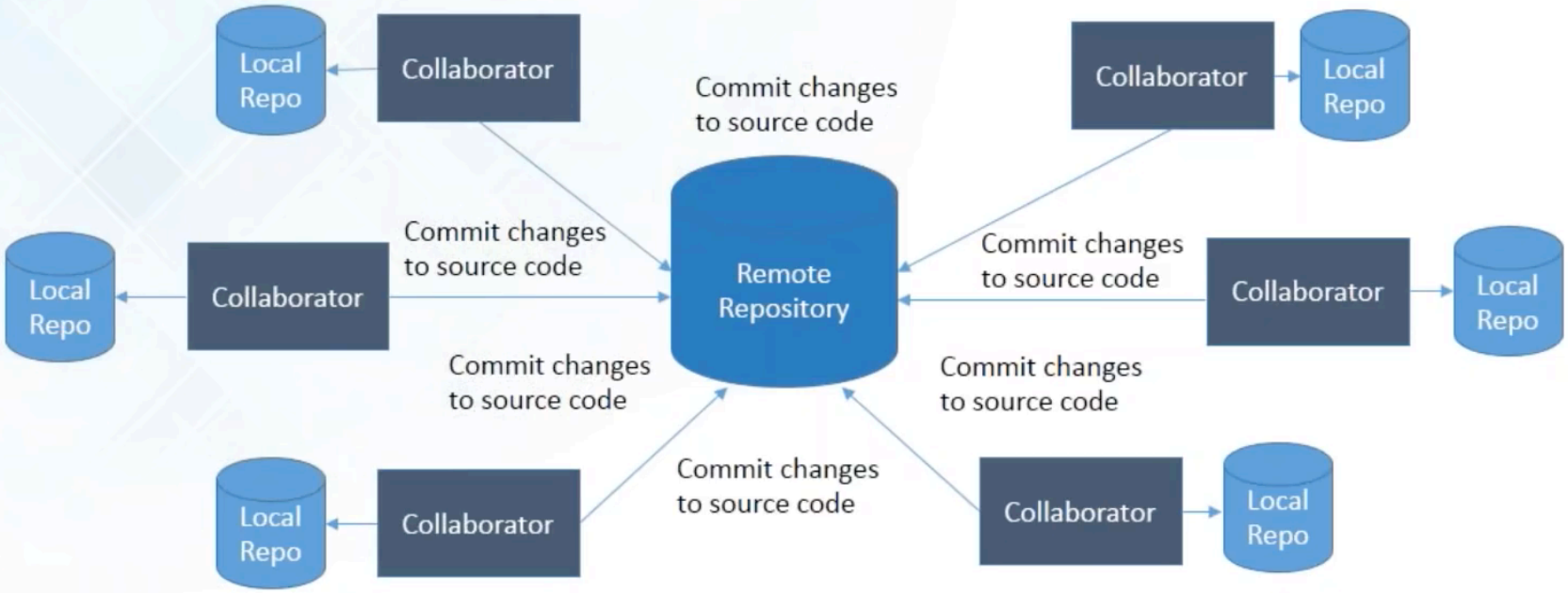
GitHub Enterprise has allowed us to store our company's source code in a central, corporately controlled system. - **Joe Fuller, CIO**

What is Git?

What is Git?

edureka!

Git is a Distributed Version Control tool that supports distributed non-linear workflows by providing data assurance for developing quality software.

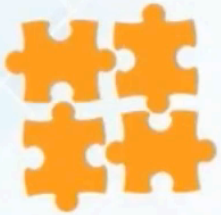


Features of Git

Features of Git

edureka!

Distributed



Compatible

Non-linear

Branching

Lightweight

Speed

Open Source

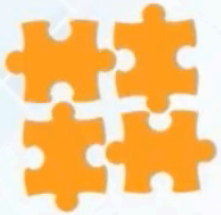
Reliable

Secure

Economical

Features of Git

edureka!



- Allows distributed development of code.
- Every developer has a local copy of the entire development history and changes are copied from one repository to another.

Features of Git

edureka!

- Compatible with existing systems & protocols.
- SVN & SVK repositories can be directly accessed using Git-SVN.

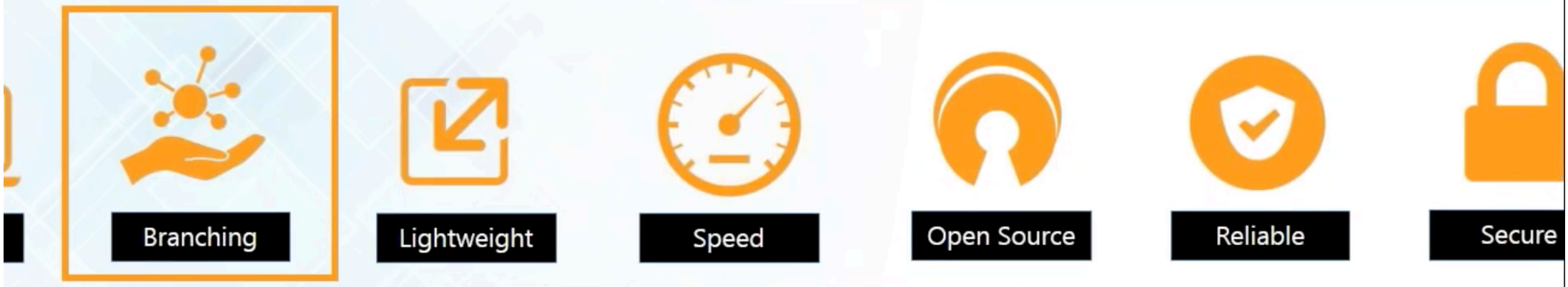
Features of Git

edureka!

- Supports non-linear development of software.
- Includes various techniques to navigate & visualize non-linear development history.

Features of Git

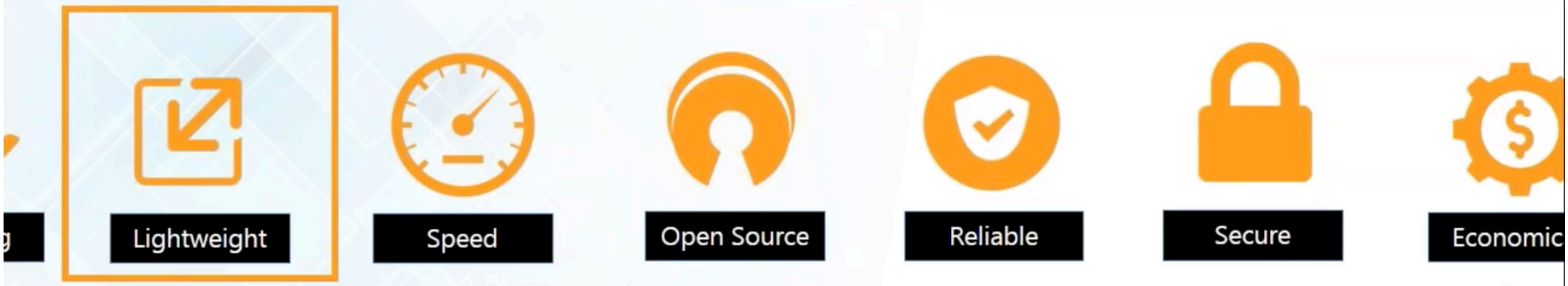
edureka!



- It takes only a few seconds to create & merge branches.
- Master branch always contains production quality code.

Features of Git

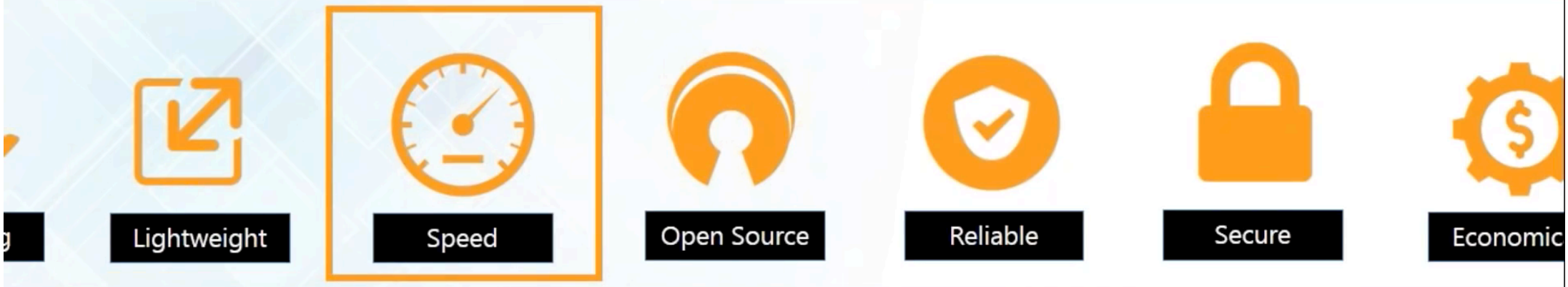
edureka!



- Uses lossless compression technique to compress data on the client's side.

Features of Git

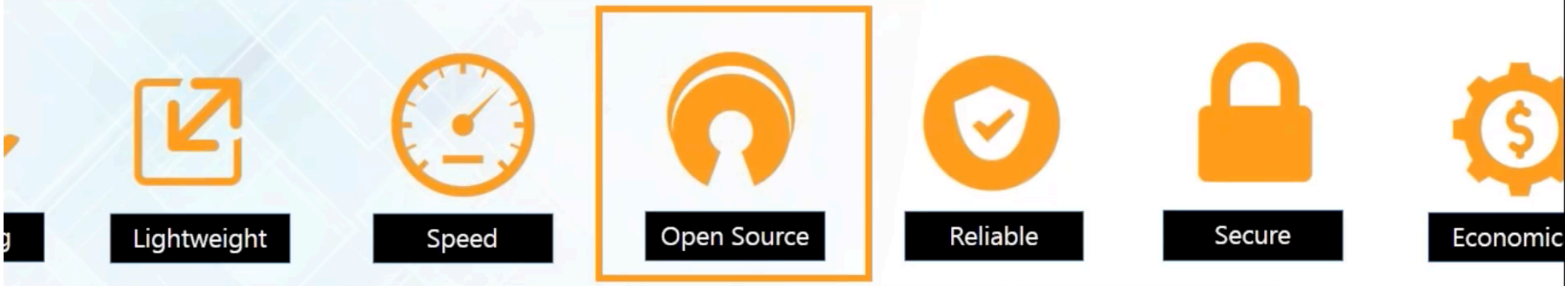
edureka!



- Fetching data from local repository is 100 times faster than remote repository.
- GIT is one order of magnitude faster than other VCS tools.

Features of Git

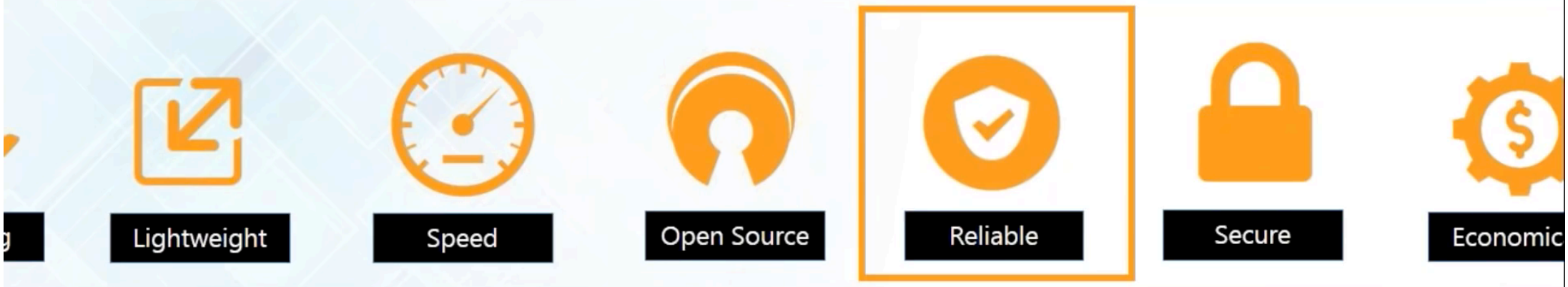
edureka!



- You can modify its source code according to your needs.

Features of Git

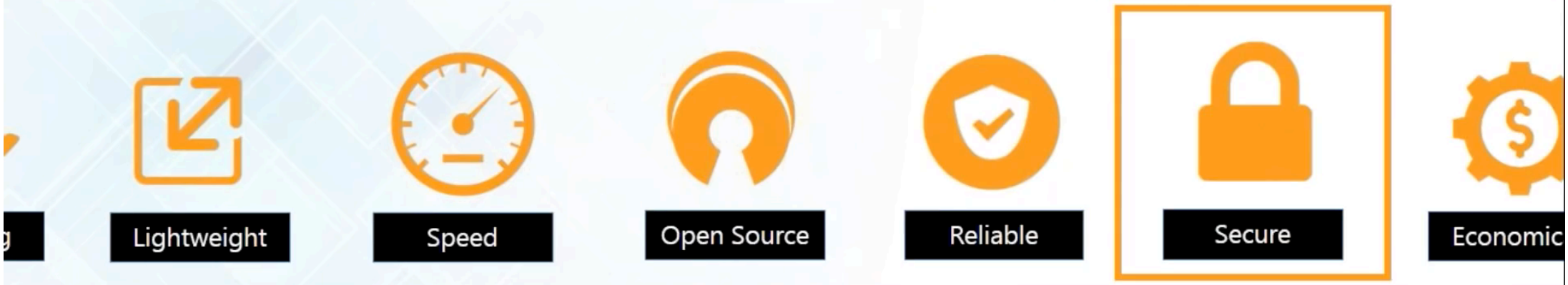
edureka!



- On events of system crash, the lost data can be easily recovered from any of the local repositories of the collaborators.

Features of Git

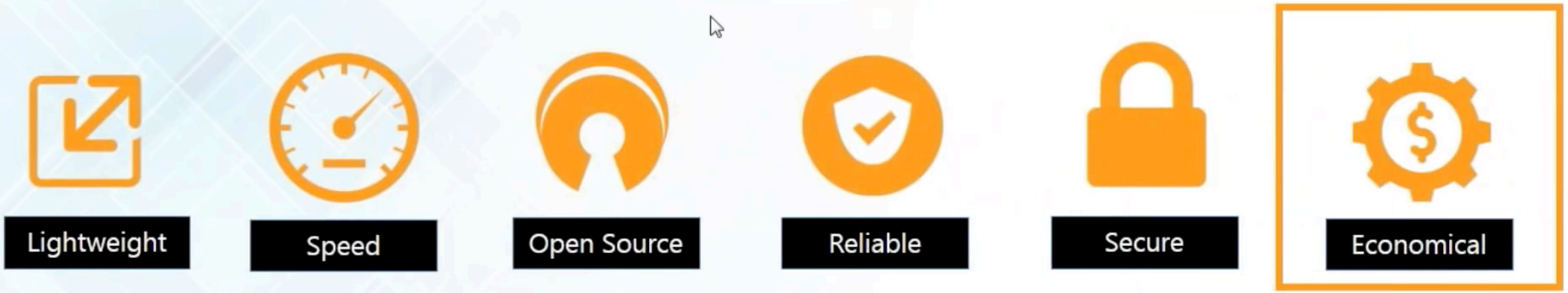
edureka!



- Uses SHA1 to name and identify objects.
- Every file & commit is checksummed and is retrieved by its checksum at time of checkout.

Features of Git

edureka!



- Released under GPL's license. It is for free.
- All heavy lifting is done on client-side, hence a lot of money can be saved on costly servers.

What is a Repository?

edureka!

A directory or storage space where your projects can live. It can be local to a folder on your computer, or it can be a storage space on GitHub or another online host. You can keep code files, text files, image files, you name it, inside a repository.

There are two types of repositories:

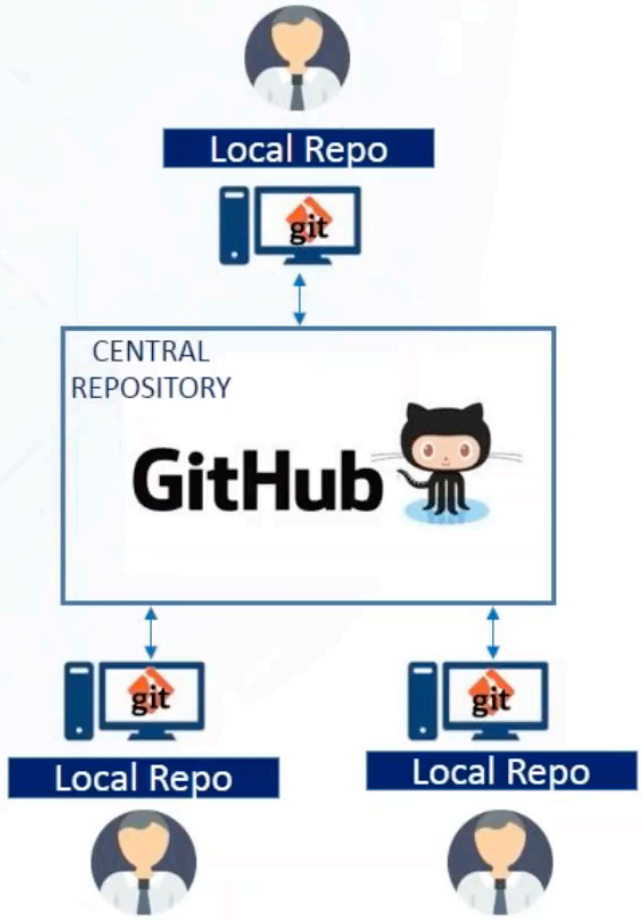
1. Central Repository
2. Local Repository

Central & Local Repository

edureka!

Central Repository

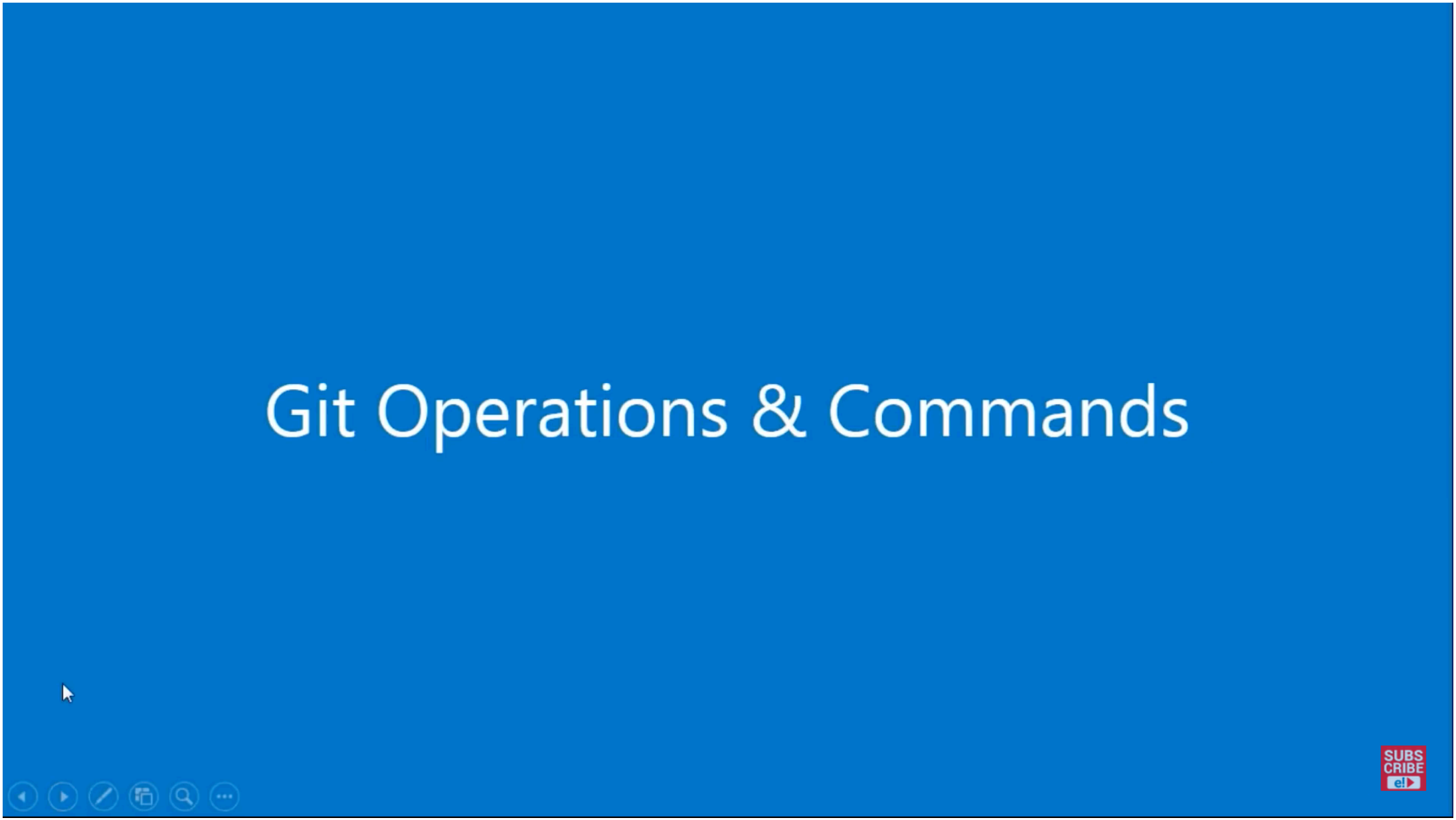
- Typically located on remote server
- Exclusively consists of ".git" repository folder
- Meant for team to share and exchange data



Local Repository

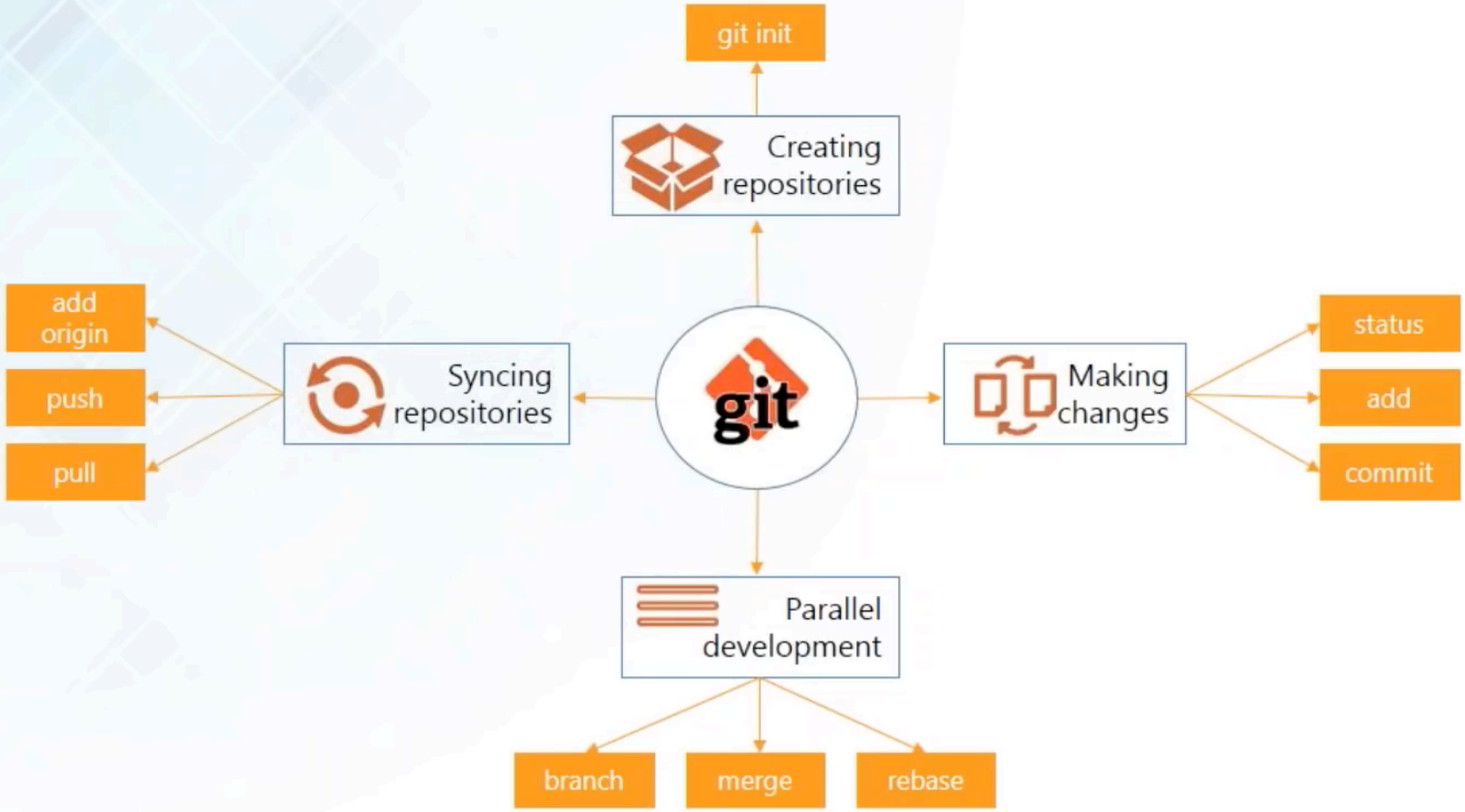
- Typically located on local machine
- Resides as a .git folder inside your project's root
- Only admin of the machine can work with this repo.

Git Operations & Commands



Git Operations & Commands

edureka!



Creating Repositories

edureka!

- Create Repo
- Syncing Repos
- Making Changes
- Parallel Development
  - Branching
  - Merging
  - Rebasing
- Git Flow

Create your Central Repository on GitHub

`git init`

Install Git on your local machine and use “git init” to create your local repository.

OR

`git clone`

Download or clone your repository from GitHub.

Syncing Repositories

edureka!

- Create Repo
- **Syncing Repos**
- Making Changes
- Parallel Development
  - Branching
  - Merging
  - Rebasing
- Git Flow

- Use '**git add origin <link>**' to add remote repo.
- Pull files with '**git pull**'
- Push your own changes into central repo with '**git push**'

Making Changes

edureka!

- Create Repo
- Syncing Repos
- **Making Changes**
- Parallel Development
  - Branching
  - Merging
  - Rebasing
- Git Flow

git status

- Tells you which files are added to index and are ready to commit.

git add

- Lets you add files to your index.

git commit

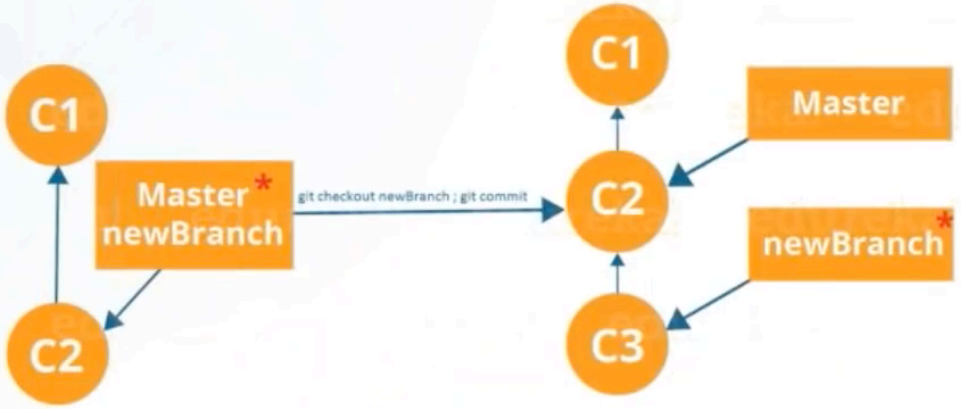
- It refers to recording snapshots of the repository at a given time.
- Committed snapshots will never change unless done explicitly.

Parallel Development - Branching

edureka!

- Create Repo
- Syncing Repos
- Making Changes
- **Parallel Development**
  - **Branching**
  - Merging
  - Rebasing
- Git Flow

- Branches are pointers to a specific commit.
- Branches are of two types:
  - Local branches
  - Remote-tracking branches

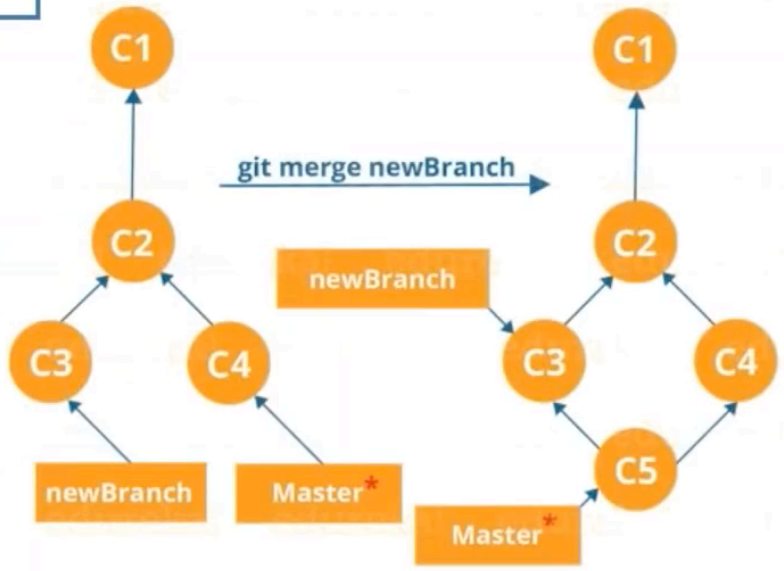


Parallel Development - Merging

edureka!

- Create Repo
- Syncing Repos
- Making Changes
- **Parallel Development**
  - Branching
  - **Merging**
  - Rebasing
- Git Flow

- It is a way to combine the work of different branches together.
- Allows to branch off, develop a new feature & combine it back in.

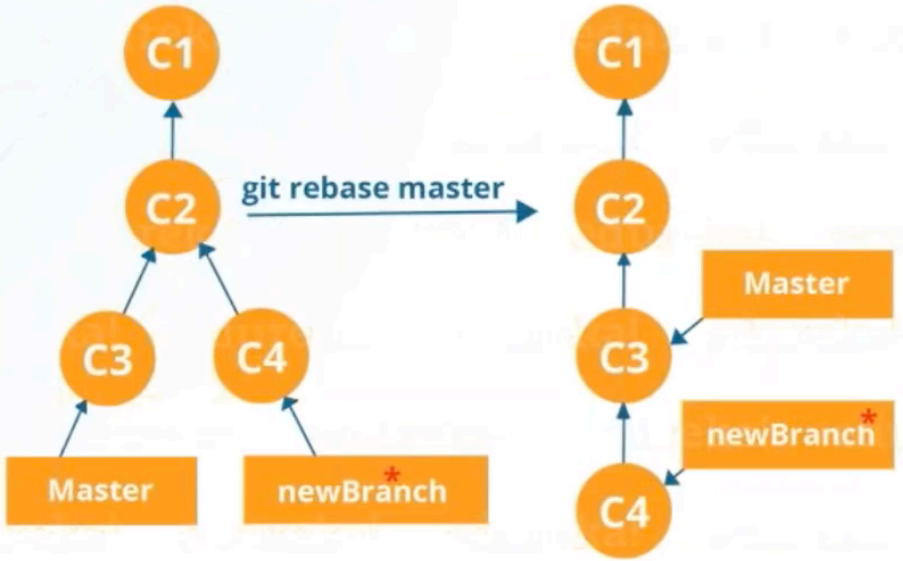


Parallel Development - Rebasing

edureka!

- Create Repo
- Syncing Repos
- Making Changes
- **Parallel Development**
  - Branching
  - Merging
  - **Rebasing**
- Git Flow

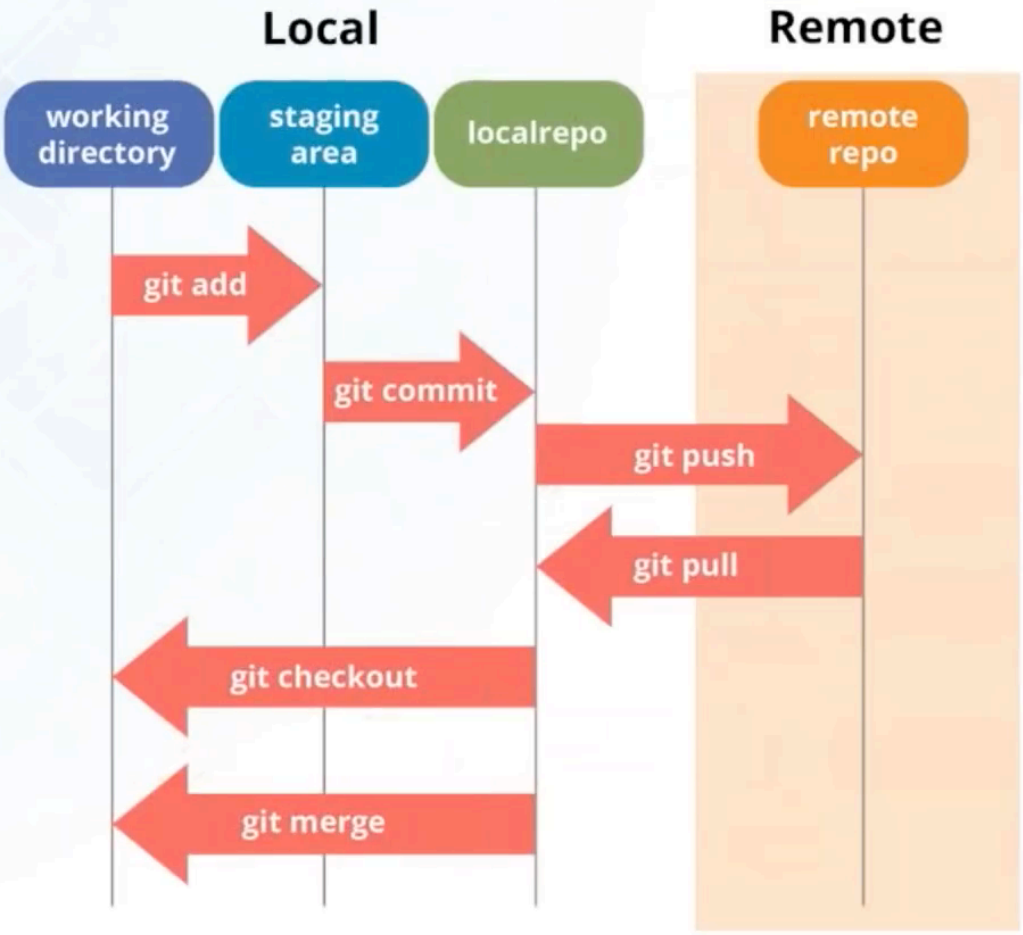
- This is also a way of combining the work between different branches.
- It can be used to make a linear sequence of commits.



Git Flow

edureka!

- Create Repo
- Syncing Repos
- Making Changes
- Parallel Development
  - Branching
  - Merging
  - Rebasing
- **Git Flow**



Summary

edureka!

Version Control

Version Control Tools

git

SUBVERSION

mercurial

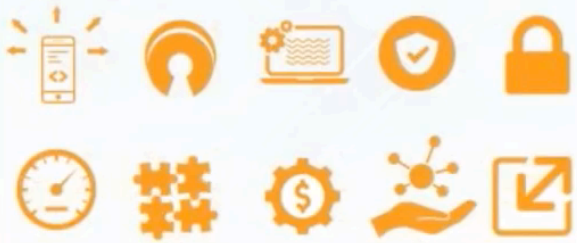
CVS

Git & GitHub

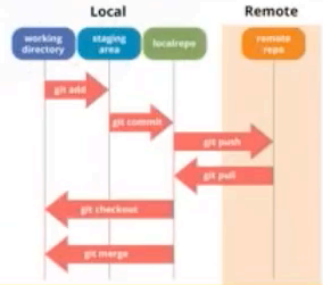
git

GitHub

Features of Git



Operations & Commands



Dominion Enterprises Case Study

DOMINION ENTERPRISES

GitHub


Questions
Queries
Feedback