

# DevOps



**Caltech**

Center for Technology &  
Management Education

## Post Graduate Program in DevOps



## Git Basics

# Learning Objectives

By the end of the lesson, you will be able to:

- 🕒 Explain Git and version control system
- 🕒 Define Git buzzwords
- 🕒 Create a GitHub repository
- 🕒 Illustrate Git configuration level and basic commands
- 🕒 Define web-scale architecture
- 🕒 Differentiate GitHub, GitLab, and Bitbucket



# Overview of Version Control Systems



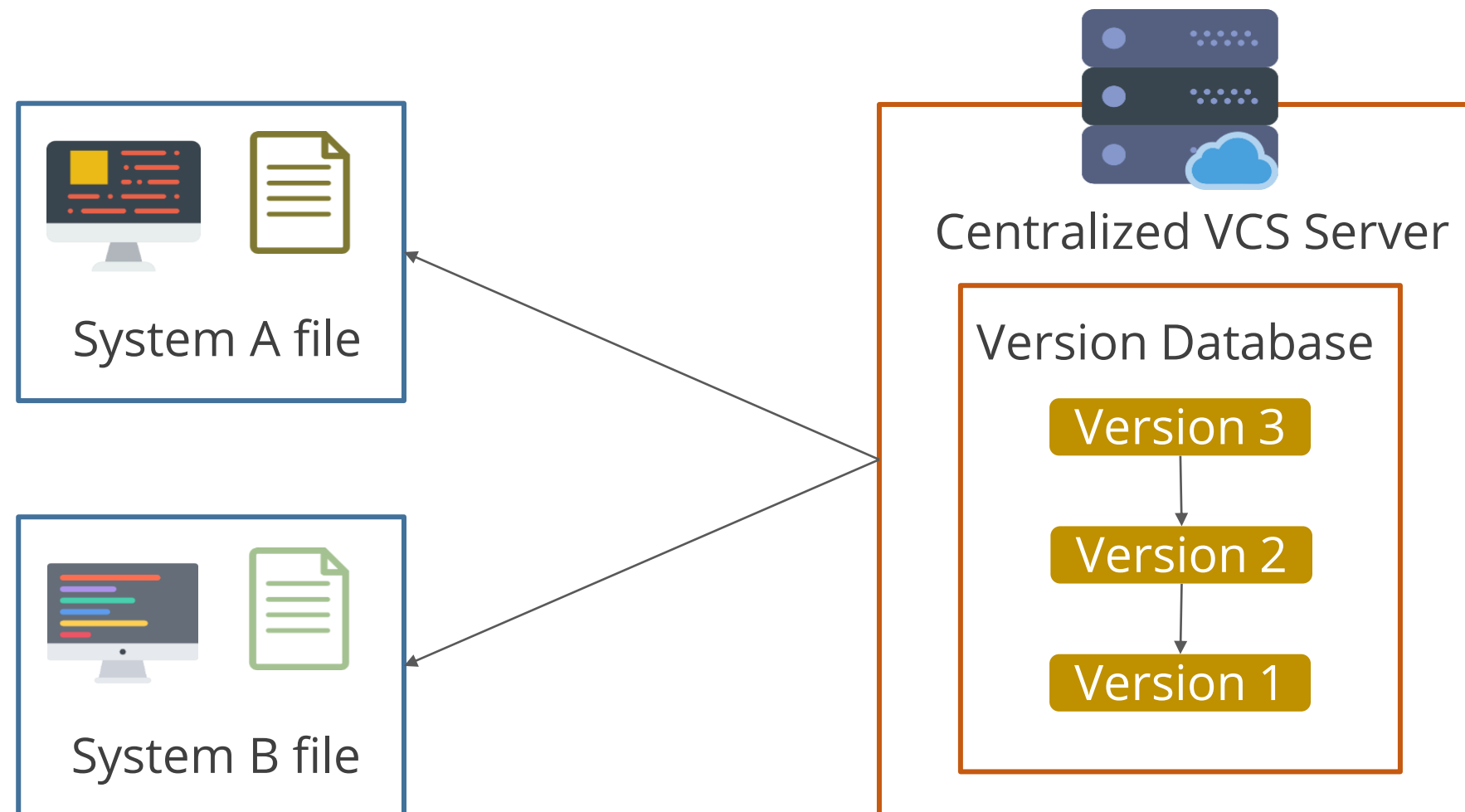
# Version Control Systems: Definition

- Version control is a system that records changes to a set of files over a period of time to recall specific versions.
- Version Control System (VCS) can be used to store every version of an image or layout.

For example:

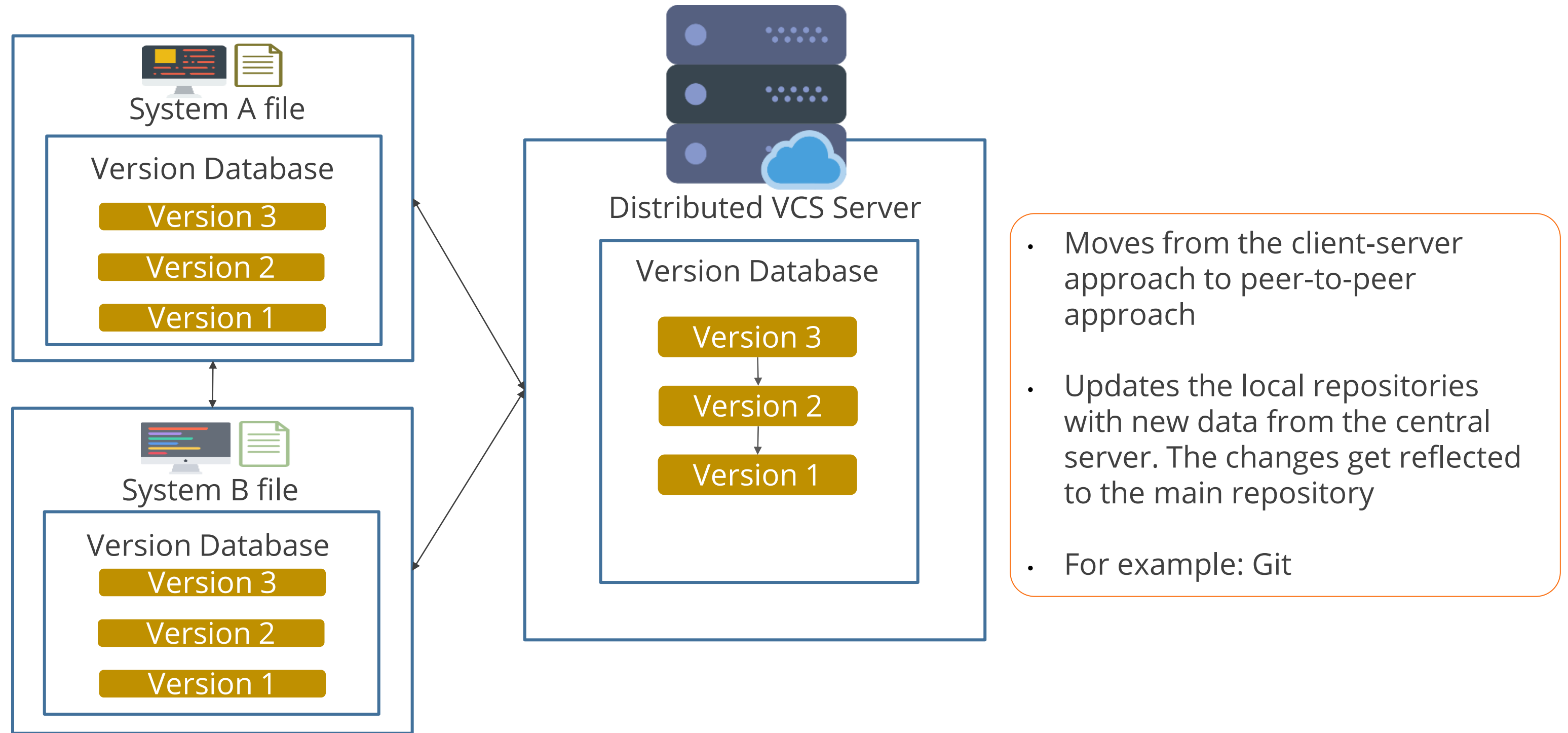


# Centralized Version Control System



- Uses a central server to store all the files
- Performs every operation directly on the repository
- Stores file versions on the central VCS server
- For example: Tortoise SVN

# Distributed Version Control System



# Version Control System: Benefits

## Collaboration:

The team can work on any file at any instance and later merge all the changes into a common version



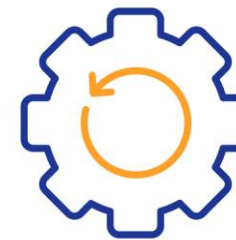
## Storage:

Acknowledges that there is only one project whereas all the past versions and variants are neatly packed up inside the VCS



## Backup:

Distributed VCS like Git act as a backup





# Version Control System: Tools

Some of the preferred open-source version control system tools for easier set up are:



# History of Git

# Git: History



Linus Torvalds  
(Creator of Linux )

Initially, it was developed to manage the Linux development community

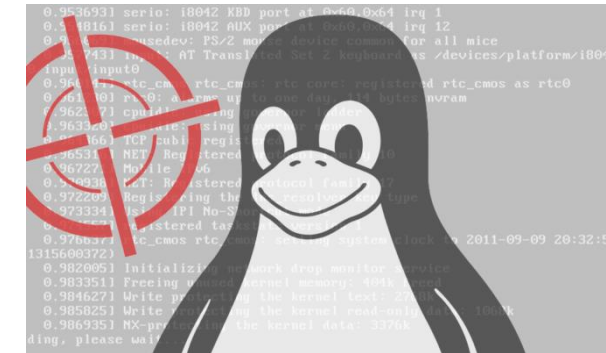
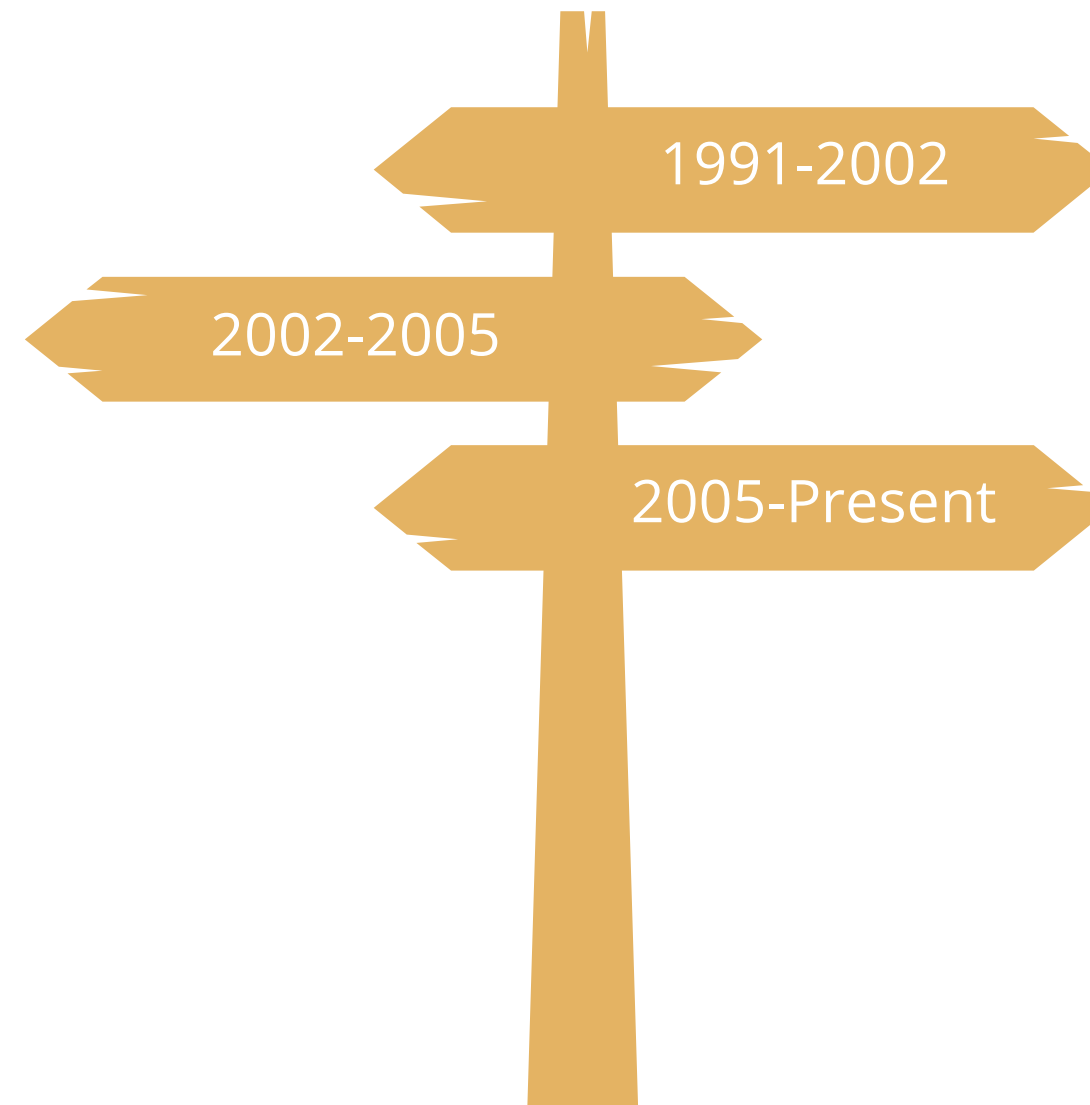
Originally, it was written in C language and then re-implemented in other languages

# Git: History

Linux code has been managed using:



BitKeeper

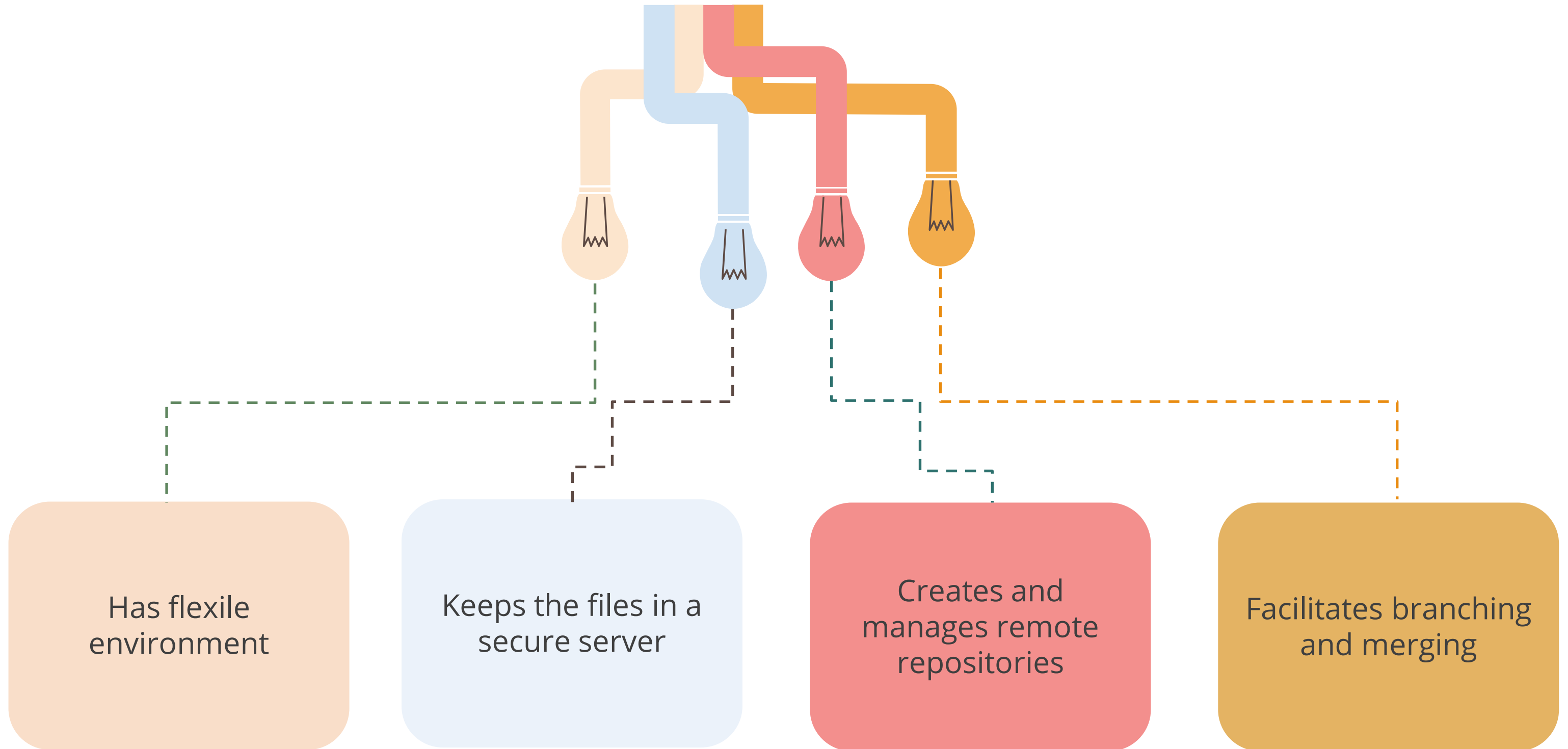


An archive of patches



Git

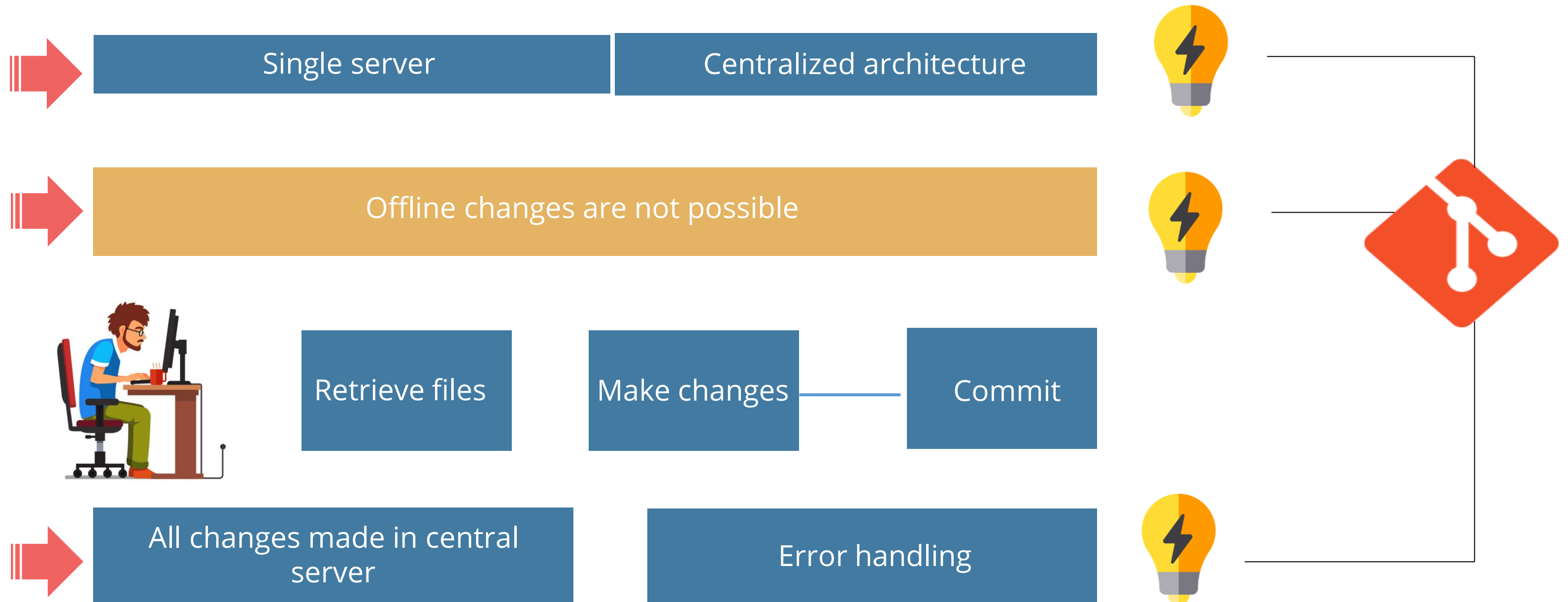
# Importance of Git





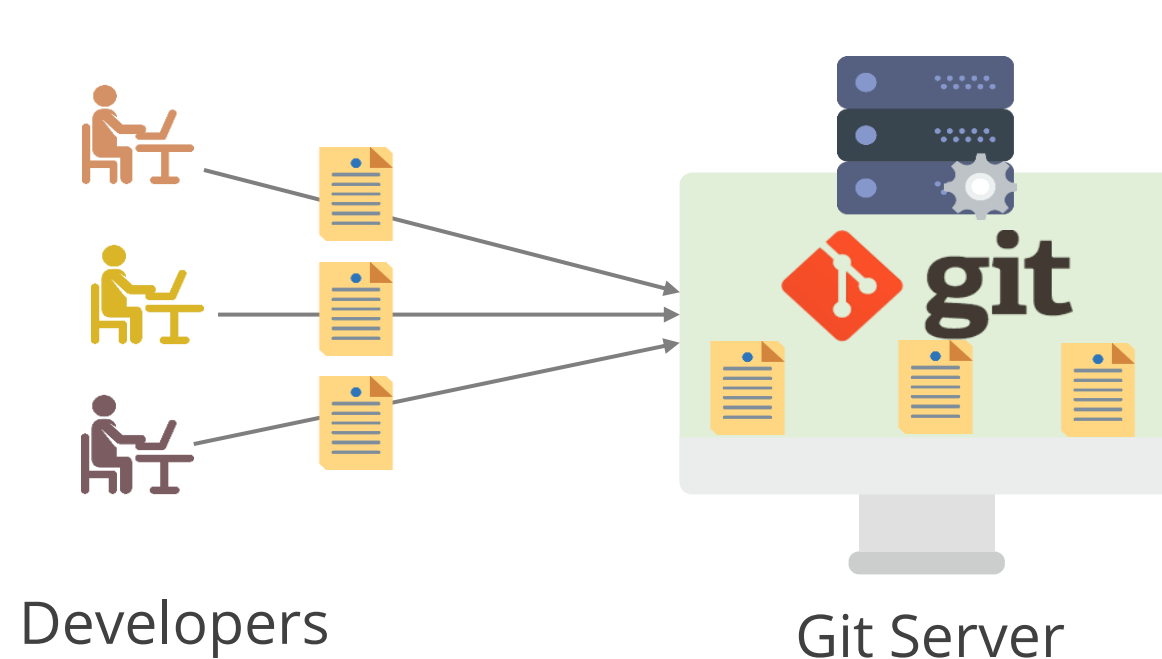
# Overview of Git

# Limitations of Existing Version Control Systems



# Git: Definition

Git is a Version Control System for tracking changes in computer files. It is generally used for source code management in software development.



Tracks changes in the source code



Uses distributed version control tool for source code management

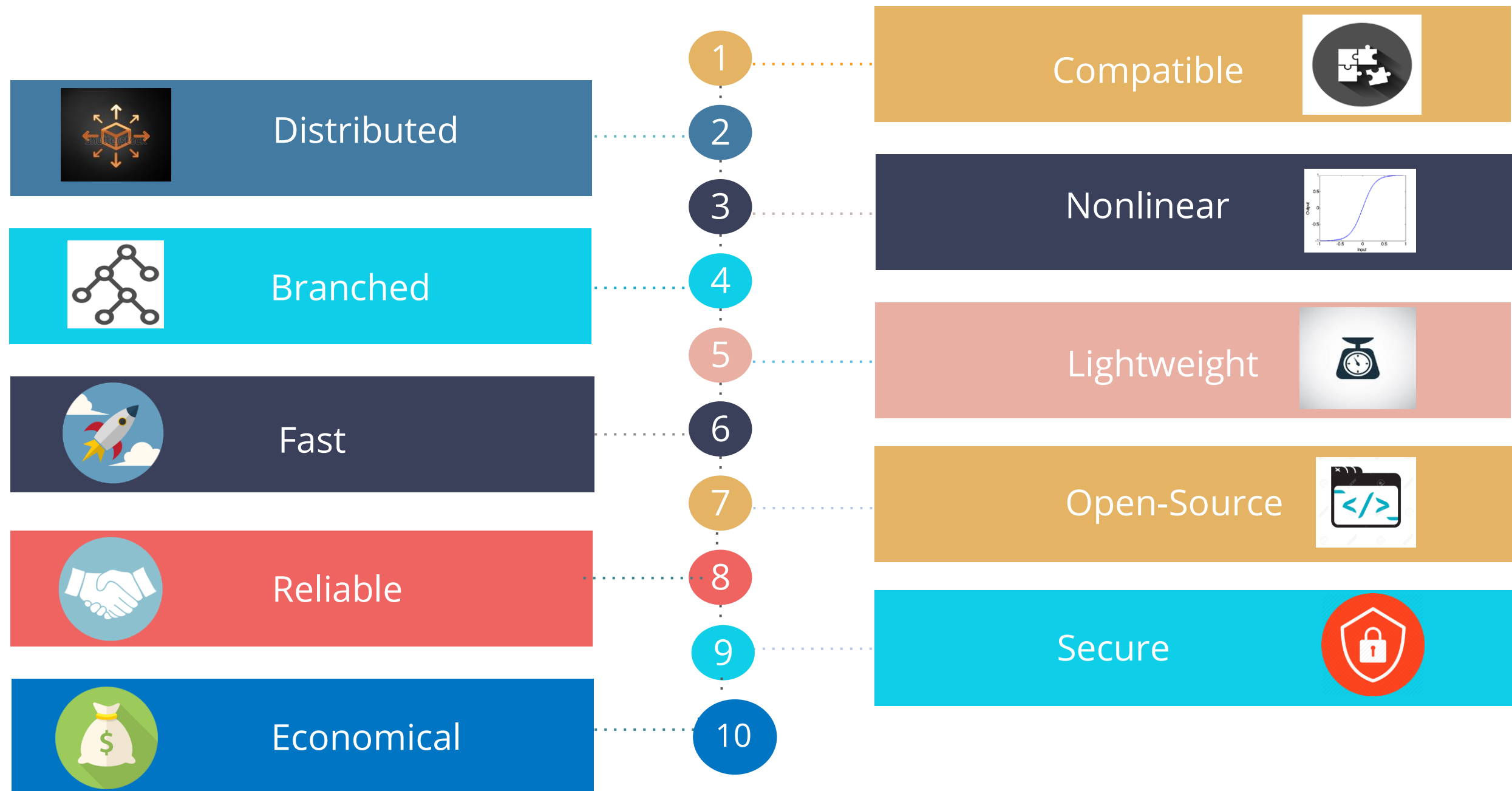


Allows multiple developers to work together

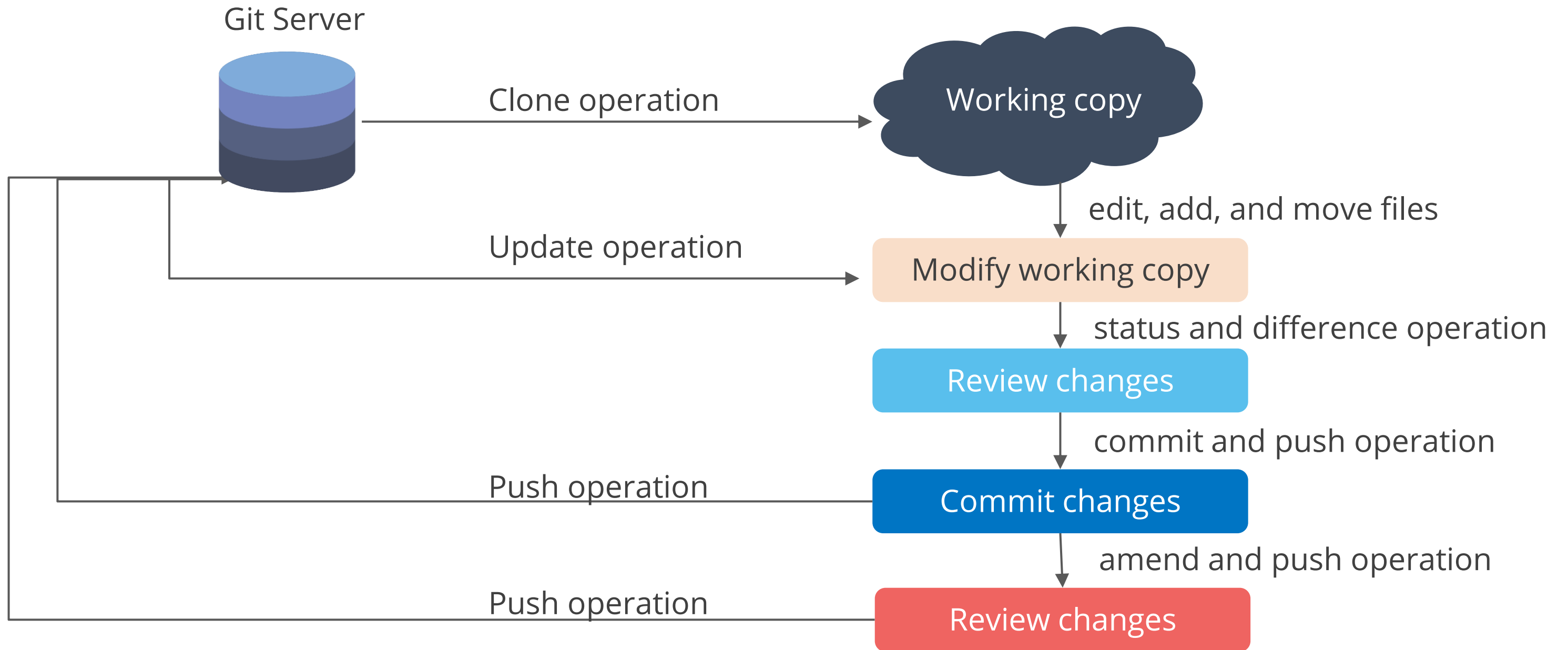


Supports non-linear development because of its several parallel branches

# Git: Features



# Git: Life Cycle





# Assisted Practice

## Install Git on Linux

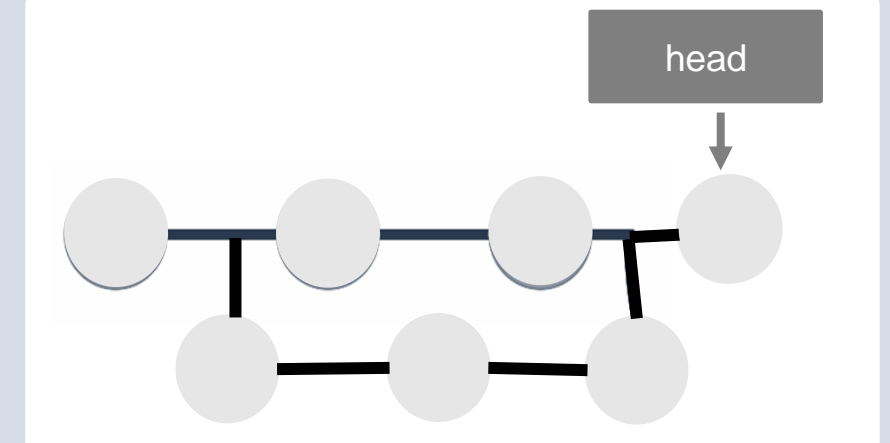
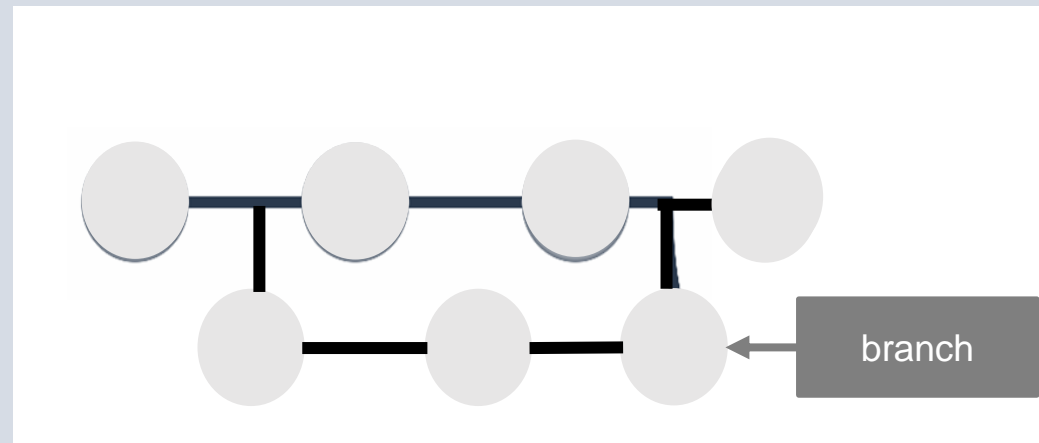
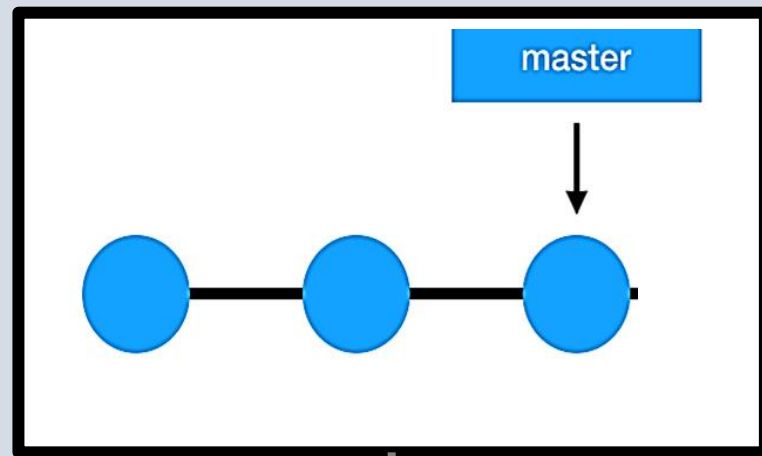
**Problem Statement:** Installation of Git on Linux platform.

Steps to Perform:

1. Install Git
2. Verify the installation

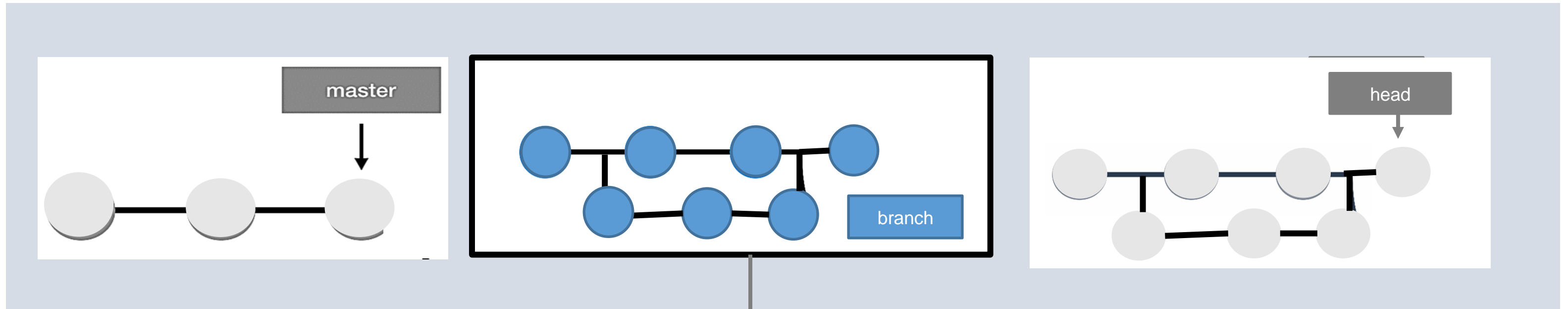
# Understand Git Buzzwords

# Git Buzzwords



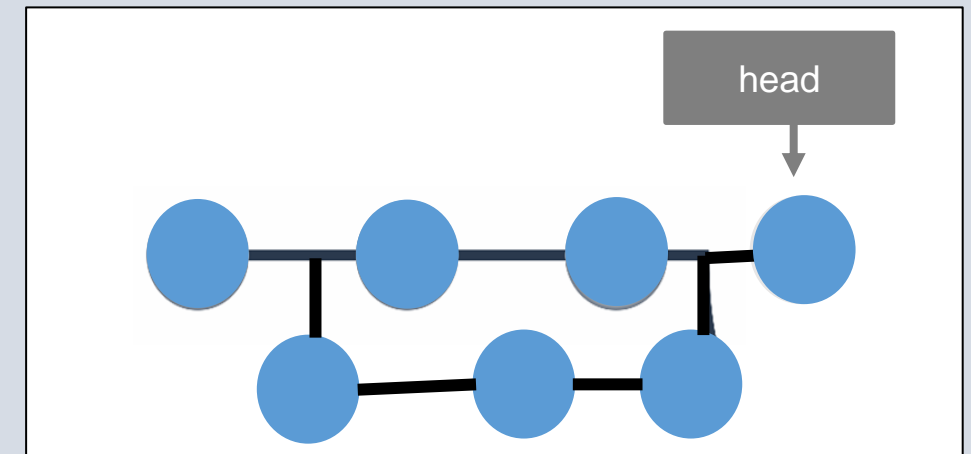
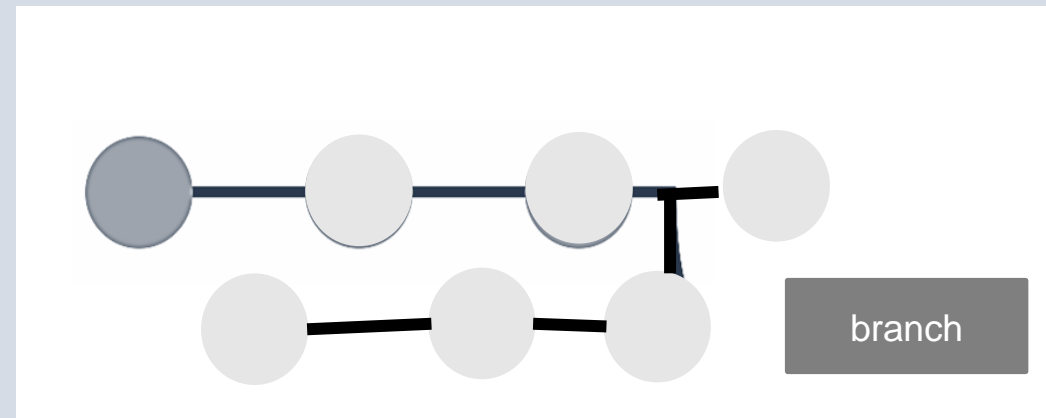
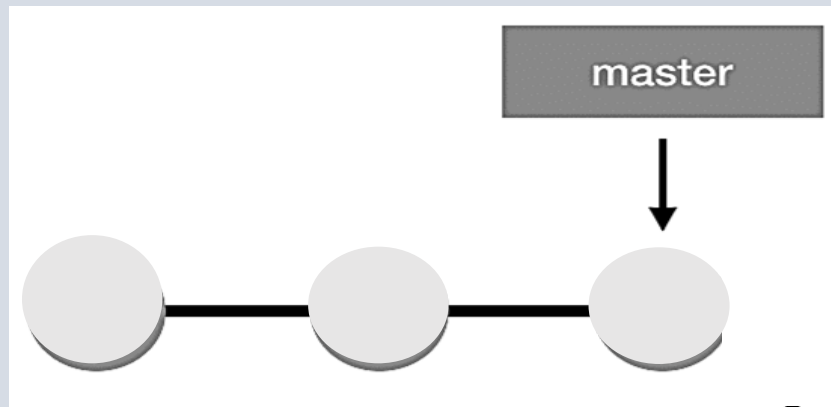
- It is a default branch.
- It is used by CI tools for build and deployment.
- It is followed by the other repositories.

# Git Buzzwords



- It is a light weight working copy.
- It has a staging area.
- It works without impacting the master branch.

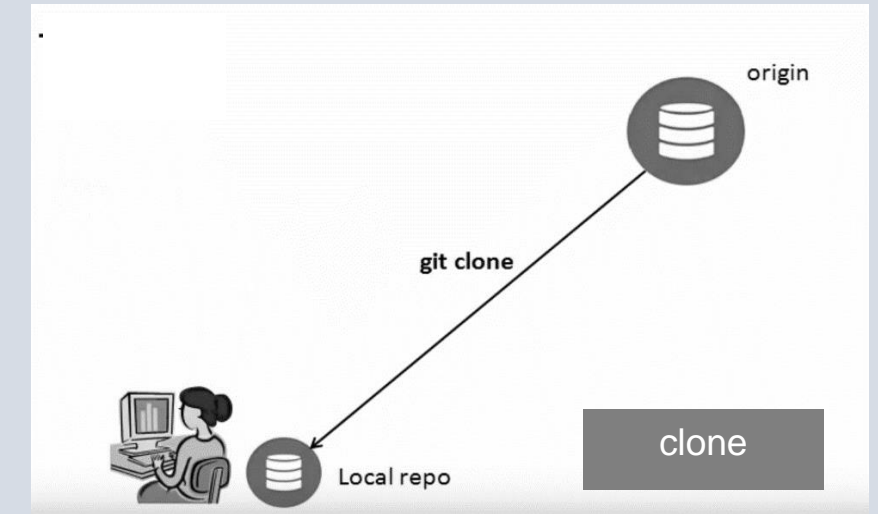
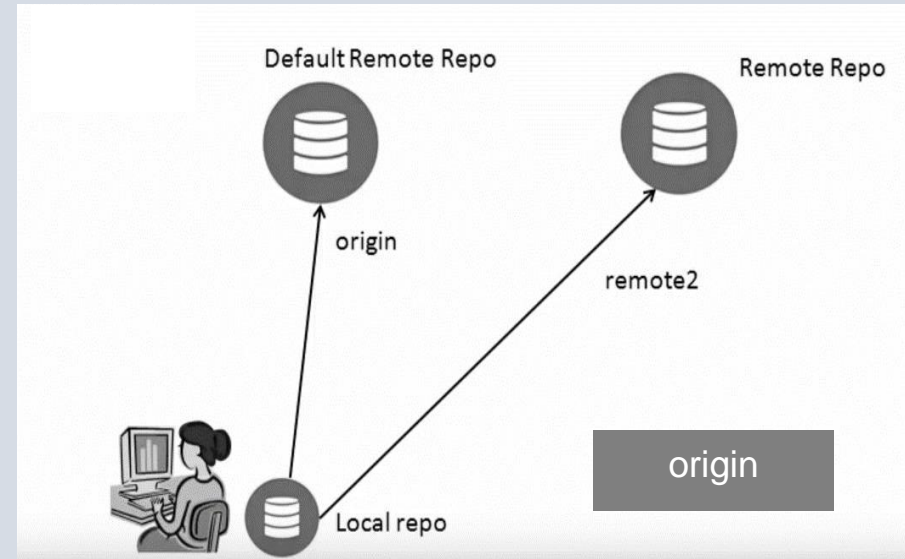
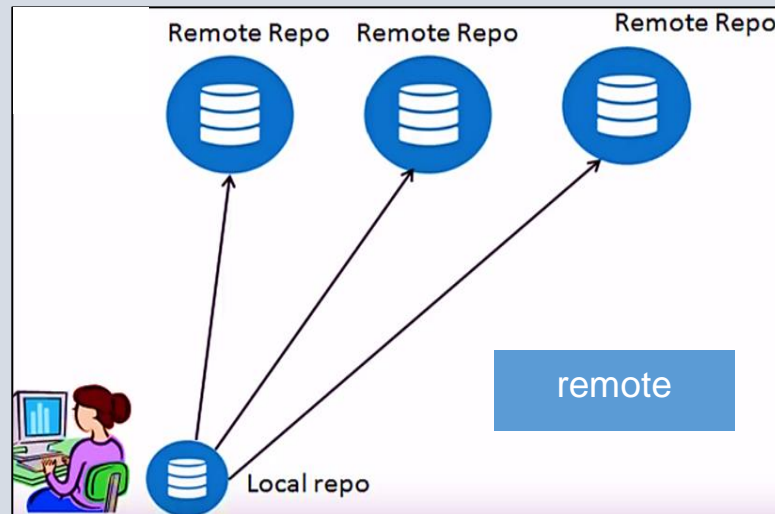
# Git Buzzwords



- It is a pointer to the latest commit of the working branch.
- It is present on every repository.
- It will point to the latest commit during branch switch.

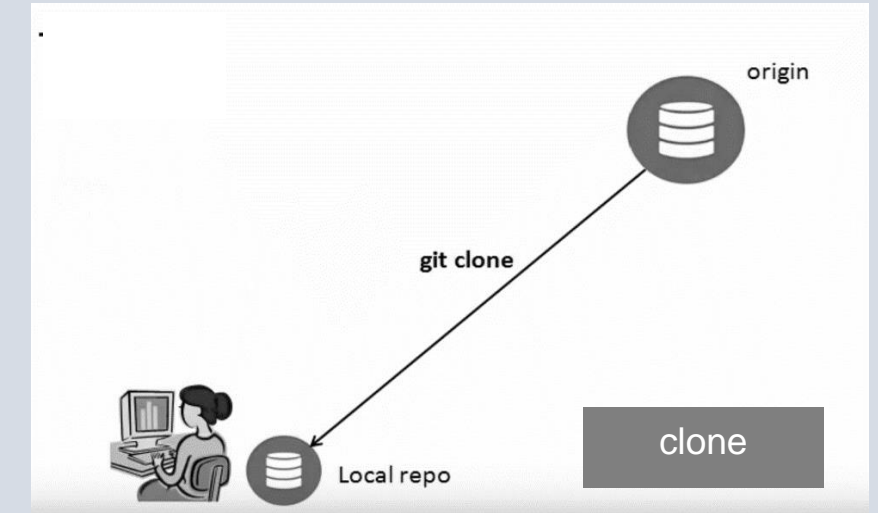
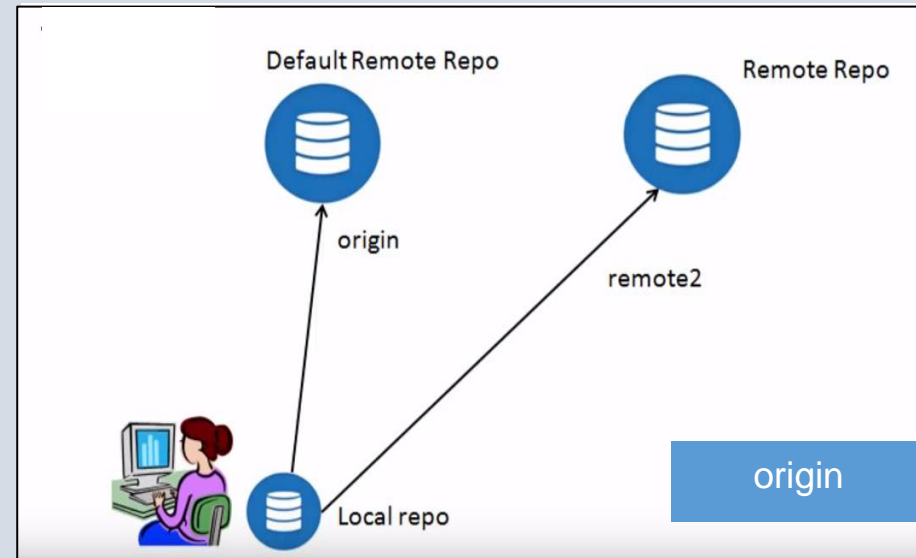
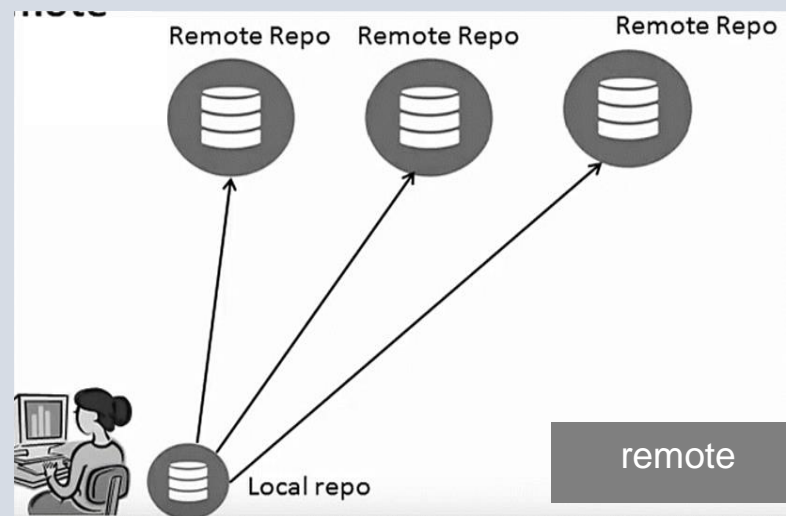


# Git Buzzwords



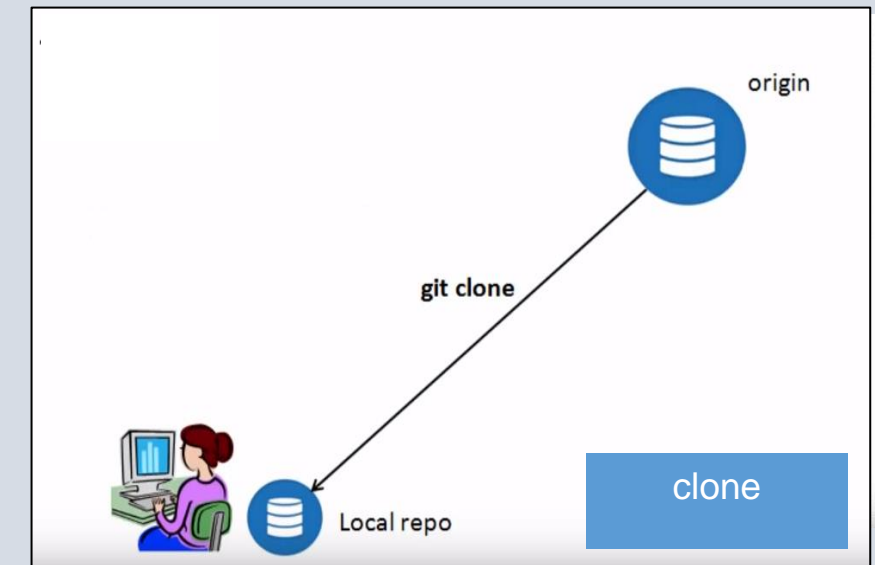
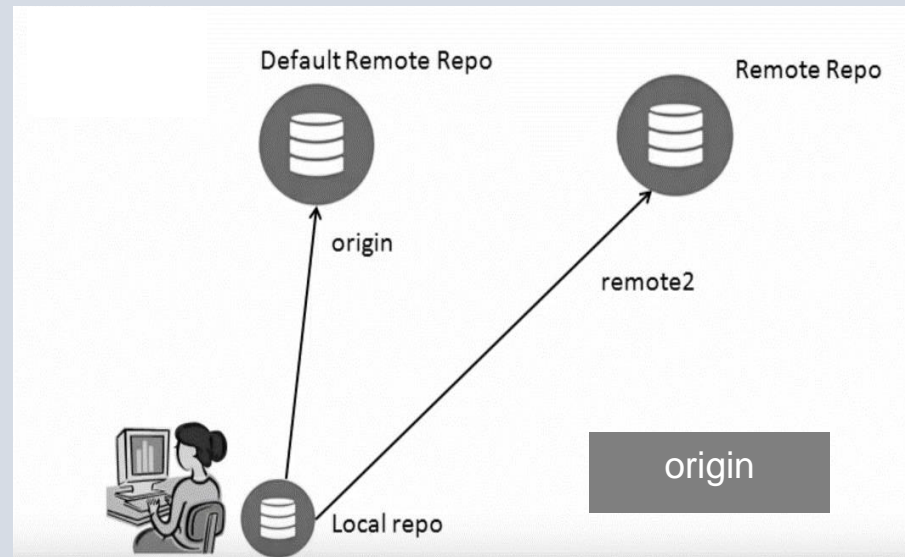
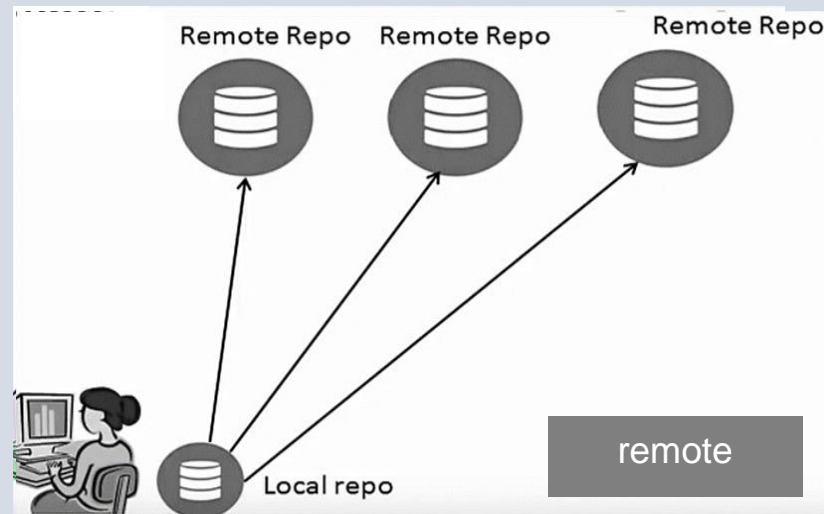
- It is a git repository on a network outside the local machine.
- It can have more than one remote repositories pointing from the local repository.
- It can be managed and referenced with short names.

# Git Buzzwords



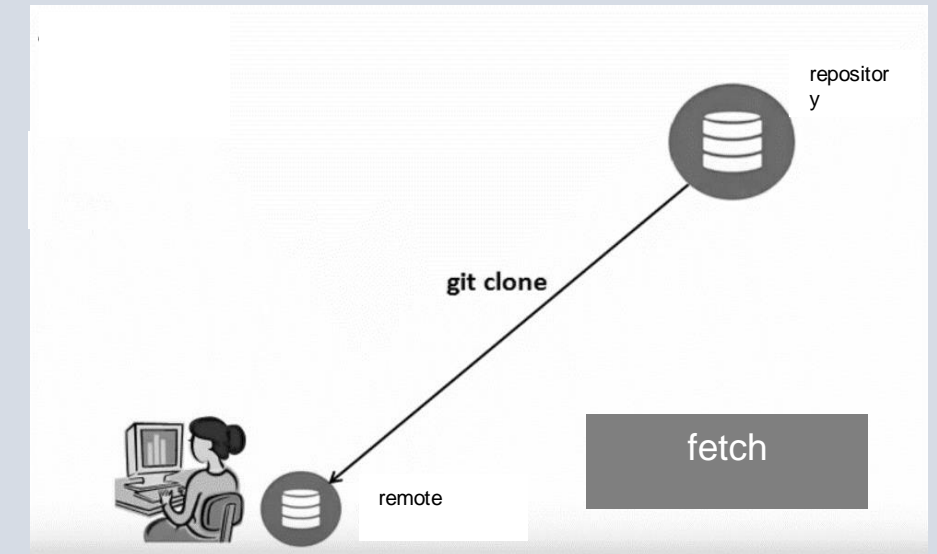
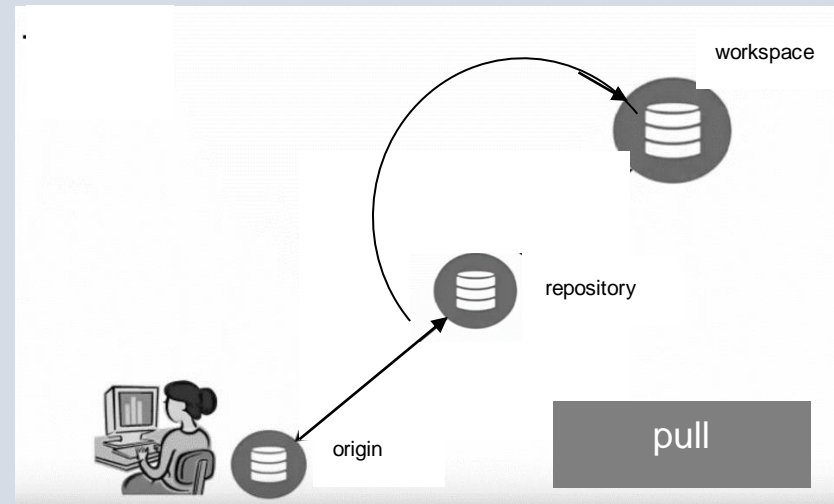
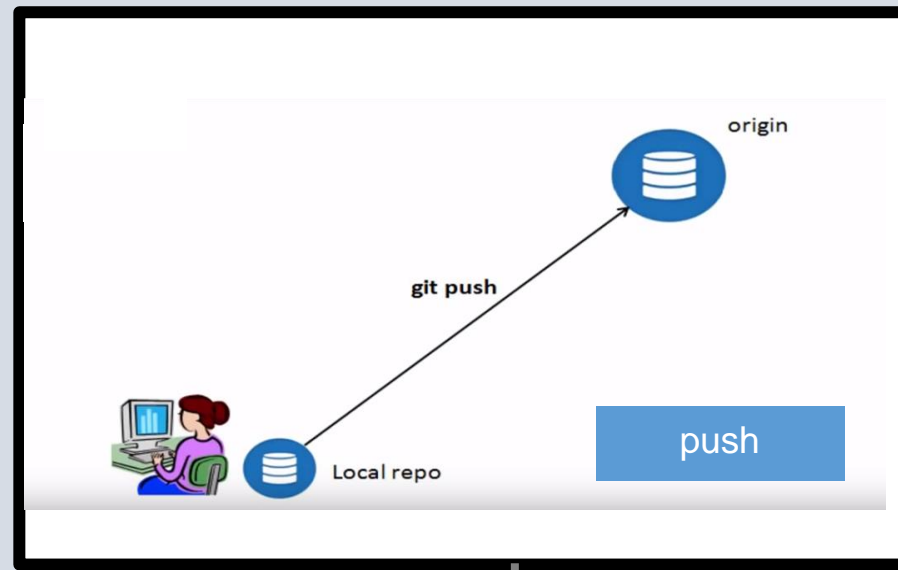
- It associates remote repository with names.
- It is a local name set for the default repository.
- It is useful to point the default repository when executing git commands.

# Git Buzzwords



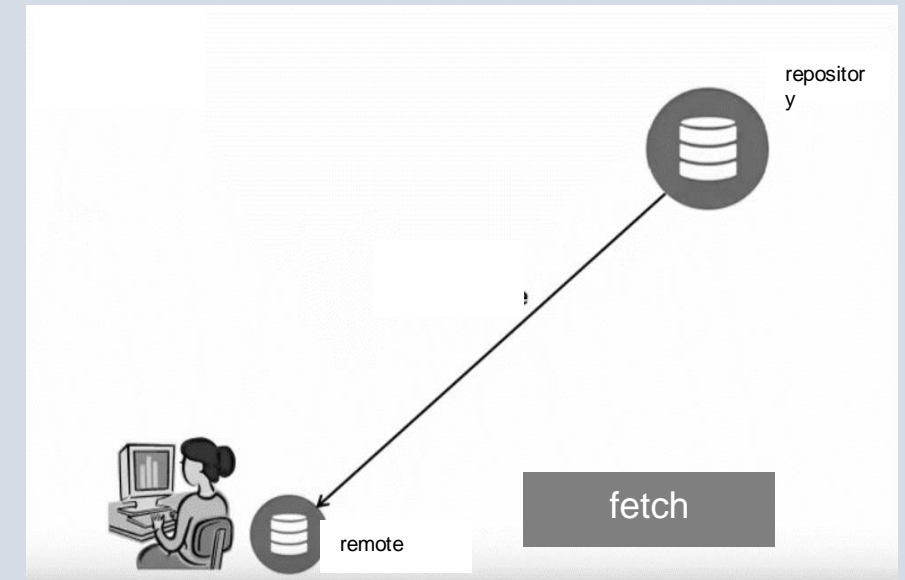
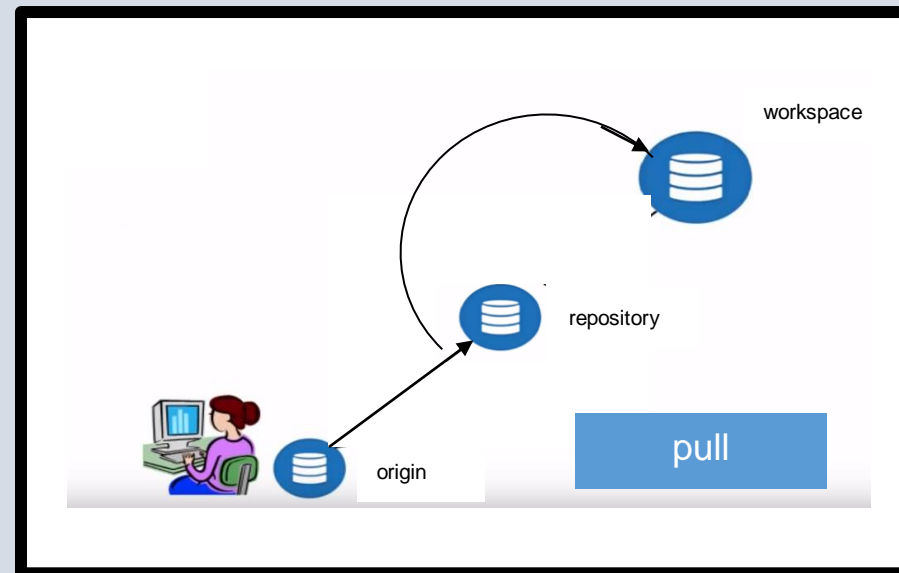
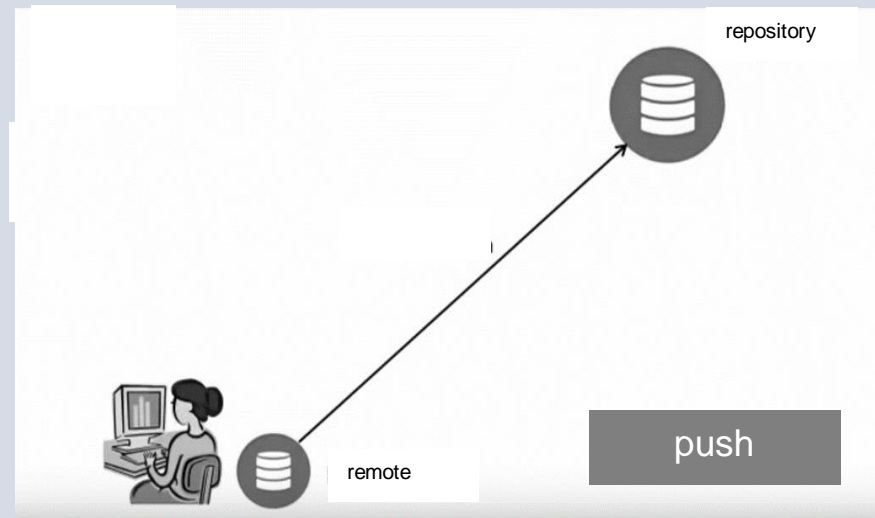
- It copies the existing repository from a remote repository.
- It will get the complete repo, whereas checkout will only fetch the working copy.
- It helps to replicate the repo on the local machine.

# Git Buzzwords



- It pushes changes from the local to the remote repository.
- It is performed after committing the changes to the local repository.
- It syncs the changes with the local and remote repository.

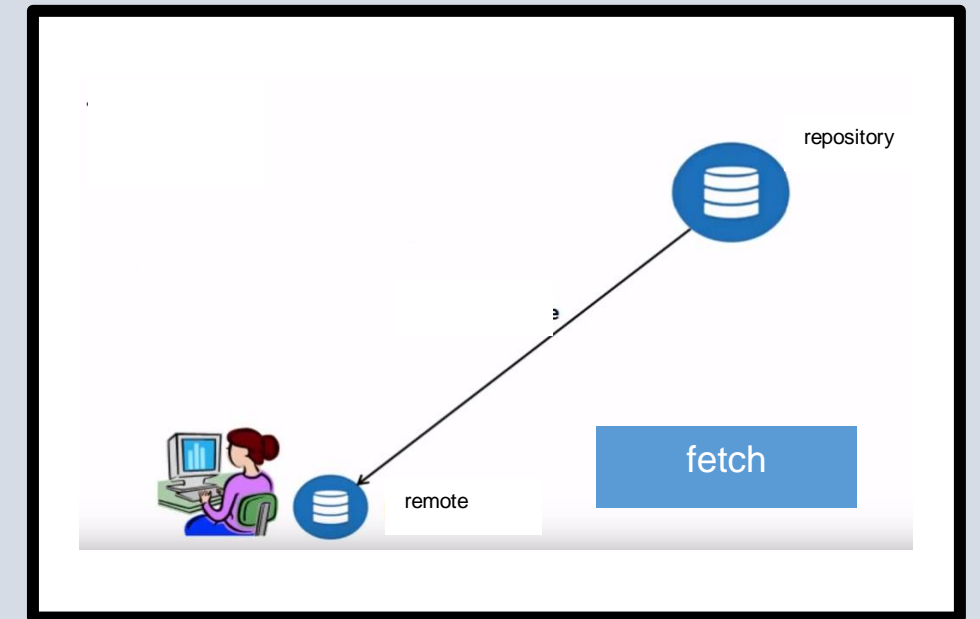
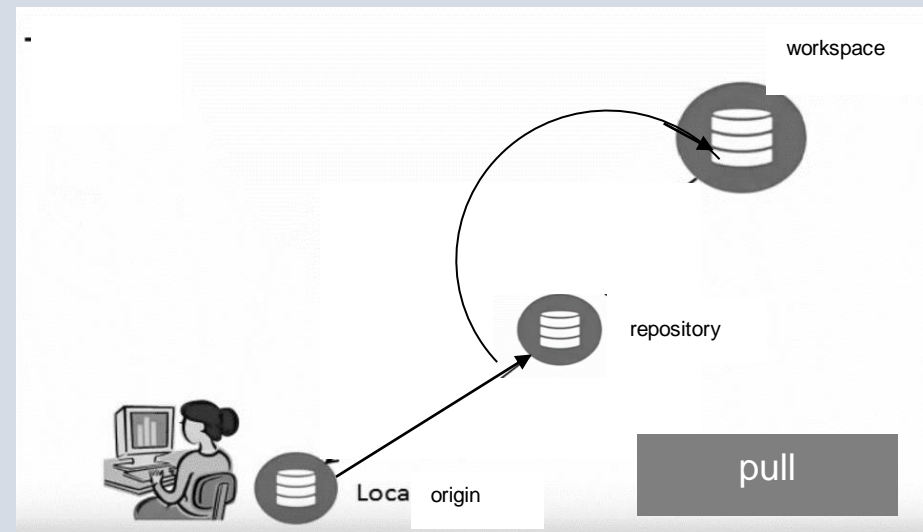
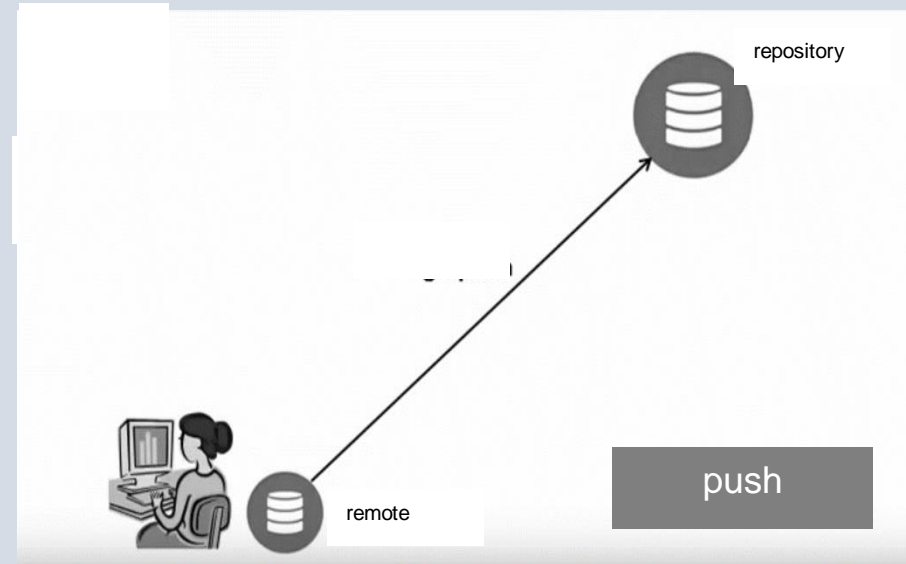
# Git Buzzwords



- It transfers the updates from the local to the remote repository.
- It syncs the changes from remote to the local repository.
- It takes current code from remote repository and merges the change with the local repository.

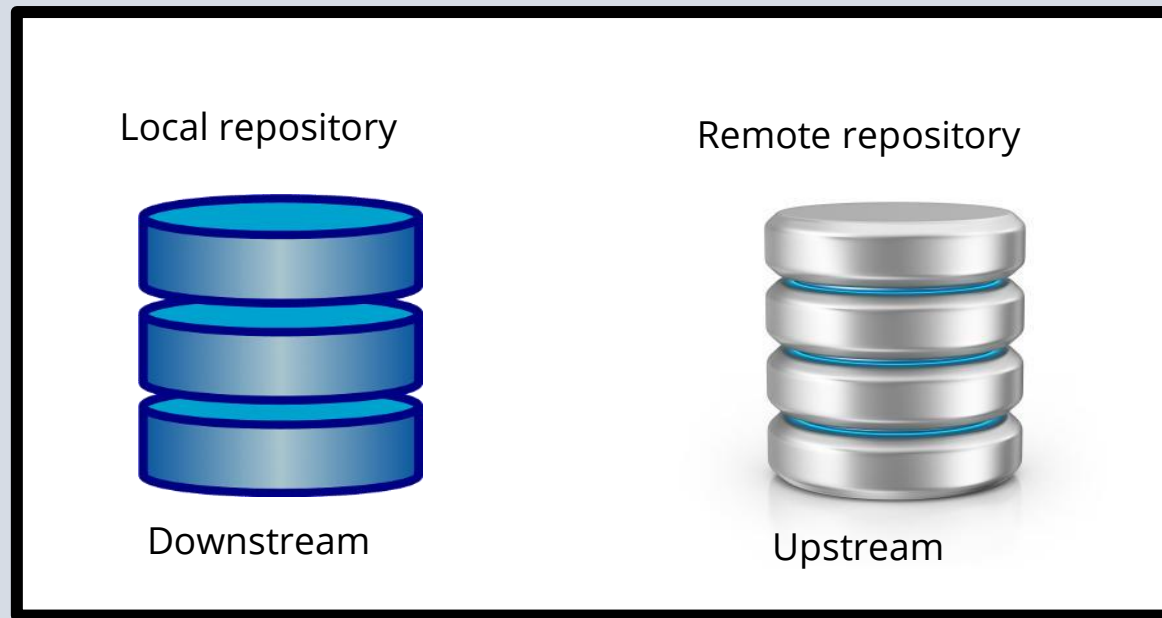


# Git Buzzwords



- It will not merge the changes with a local repository.
- It gives updates from remote to the local repository.
- It syncs the changes from remote to the local repository.

# Git Buzzwords



- When the data is flowing between repository A and B, repository A is upstream and B is downstream making B pull data from repository A.

# Introduction to GitHub

# GitHub: Basics



Transparent

Publish Project

Collaboration

Web-based Git repository hosting service

Web interface to upload files

## Who uses GitHub?

### Individuals >

Use GitHub to create a personal project, whether you want to experiment with a new programming language or host your life's work.

### Communities >

GitHub hosts one of the largest collections of open source software. Create, manage, and work on some of today's most influential technologies.

### Businesses >

Businesses of all sizes use GitHub to support their development process and securely build software.

# Git vs. GitHub

# Git vs. GitHub

## Git

It is installed and maintained on the local system.

It is a command line tool.

It is a tool to manage different versions of the file in a git repository.

## GitHub

It is hosted on the web.

It is a graphical interface.

It is a space to upload a copy of the git repository.



# Creating a Git Repository



# Steps to Create a Git Repository

Create an empty directory

Command: **\$ mkdir <directory-name>**

Convert the directory into a repository

Command: **\$ git init**

Run git status

Command: **\$ git status**

```
hp@DESKTOP-01792DU MINGW64 /d/dev/lgit/MyRepo (master)
$ git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)
```

## NOTE

\$ git init command creates a hidden directory that stores all the metadata required for it to function.



## Assisted Practice

### Create a GitHub Repository

**Problem Statement:** To create a GitHub repository to share the project files with your coworkers.

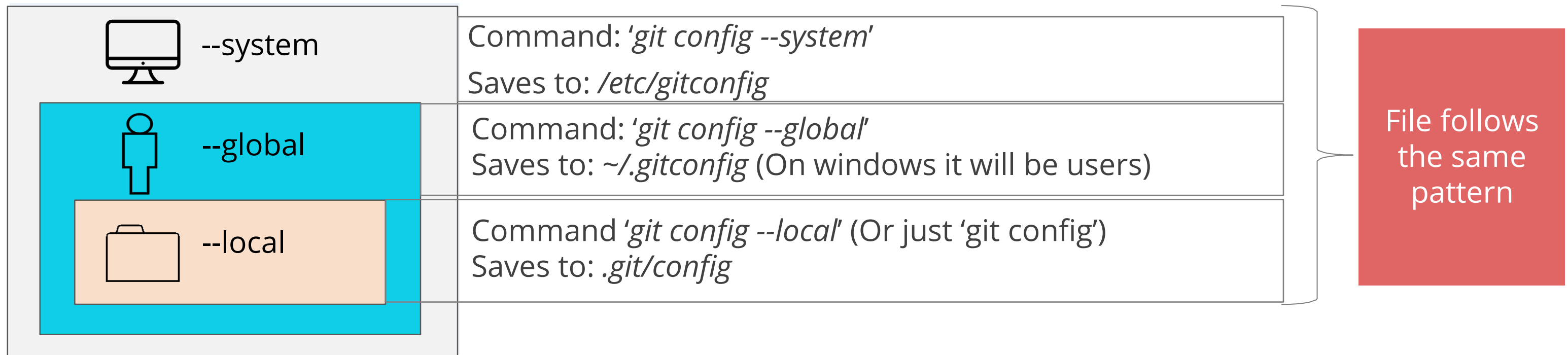
#### Steps to Perform:

1. Creating a new github repository
2. Editing the README file
3. Uploading a file to the repository

# Git Configuration Level

# Git Configuration Level

The git config command allows to configure the Git settings.



## NOTE

Local overrides Global and Global overrides System Level.

# Assisted Practice

## Configure Git

**Problem Statement:** The configuration created during Git installation was lengthy. Your team wants Git to be configured for ease of use.

### Steps to Perform:

1. Configure Git with username and email id
2. Confirm the username and email id
3. Enabling credentials storage globally

## Assisted Practice

### Clone a GitHub Repository

**Problem Statement:** Clone the GitHub repository shared by your coworker to access the project files.

#### Steps to Perform:

1. Cloning a github repository

# Basic Git Commands

# Basic Git Commands

Below is the list of some basic Git commands:

Task	Explanation	Commands
Tell Git who you are	Configure the author's name and email address	git config --global user.name "Simplilearn" git config --global user.email simplilearn@example.com
Create a new local repository	Create a repository	git init
Check the repository	Create a working copy of a local repository	git clone /path/to/repository
Check the repository	Use a remote server	git clone username@host:/path/to/repository

# Basic Git Commands

Below is the list of some basic Git commands:

Task	Explanation	Commands
Add files	Add one or more files to staging	git add <filename> git add *
Push	Send changes to the master branch	git push origin master
Commit	Commit changes to the head	git commit -m "Commit message"
Commit	Commit files added with <b>git add</b> and the files changed	git commit -a



# Basic Git Commands

Below is the list of some basic Git commands:

Task	Explanation	Commands
Status	List the files that need to be changed, added, or committed	git status
Connect to a remote repository	Add the server to push for the connection	git remote add origin <server>
Connect to a remote repository	List all currently configured remote repositories	git remote -v
Search	Search the working directory for foo()	git grep "foo()"

# Basic Git Commands

Below is the list of some basic Git commands:

Task	Explanation	Commands
Branches	Create a new branch and switch	git checkout -b <branchname>
Branches	Switch from one branch to another	git checkout <branchname>
Branches	List all the branches that tell you what branch you're currently in	git branch
Branches	Delete the feature branch	git branch -d <branchname>

# Basic Git Commands

Below is the list of some basic Git commands:

Task	Explanation	Commands
Branches	Push the branch to your remote repository	<code>git push origin &lt;branchname&gt;</code>
Branches	Push all branches to your remote repository	<code>git push --all</code>
Branches	Delete a branch on your remote repository	<code>git push origin :&lt;branchname&gt;</code>

# Basic Git Commands

Below is the list of some basic Git commands:

Task	Explanation	Commands
Update from the remote repository	Fetch and merge changes on the remote server	git pull
Update from the remote repository	Merge a different branch in an active branch	git merge <branchname>
Update from the remote repository	View all the merge conflicts	git diff
Update from the remote repository	View the conflicts against the base file	git diff --base <filename>
Update from the remote repository	Preview changes before merging	git diff <sourcebranch> <targetbranch>

# Basic Git Commands

Below is the list of some basic Git commands:

Task	Explanation	Commands
Update from the remote repository	Manually resolve the conflicts and mark the changed file	git add <filename>
Tags	Use tagging to mark a significant changeset	git tag 1.0.0 <commitID>
Tags	Get the ID in use	git log
Tags	Push all tags to remote repository	git push --tags origin

# Basic Git Commands

Below is the list of some basic Git commands:

Task	Explanation	Commands
Undo the local changes	Replace the changes in your working tree with the last content in head	<code>git checkout -- &lt;filename&gt;</code>
Undo the local changes	Fetch the latest history from the server and point to the local master branch	<code>git fetch origin</code> <code>git reset --hard origin/master</code>

## Assisted Practice

### Create Git Alias

**Problem Statement:** You are facing issues with lengthy command lines. So you need to add alias for the commands.

#### Steps to Perform:

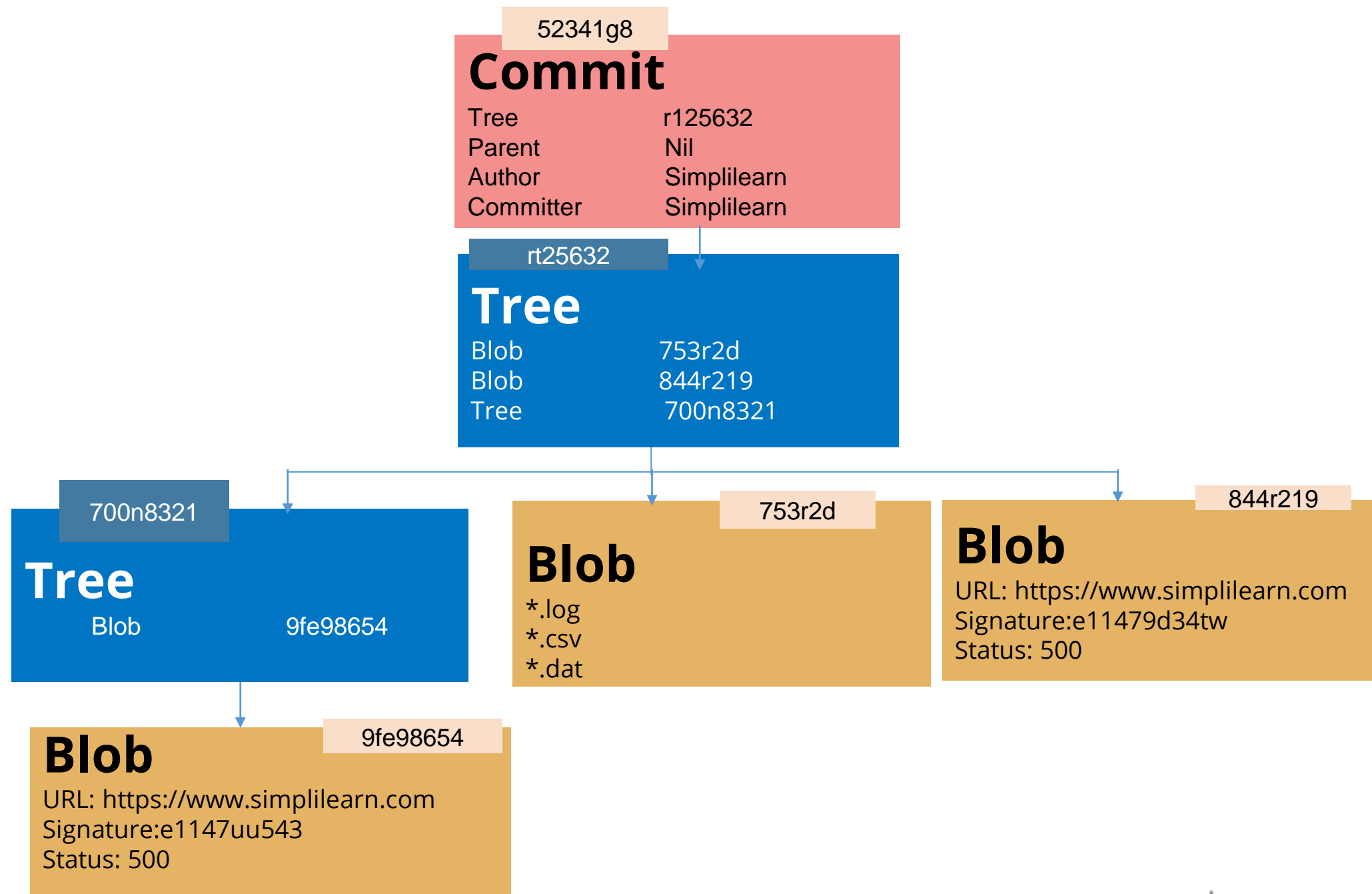
1. Create an Alias



# Web Scale Architecture




# Web Scale Architecture

- Git uses a Directed Acyclic Graph (DAG) to represent history with commits.
- Git stores a snapshot of your repository whenever commit is created.



# Difference between GitHub, Bitbucket, and GitLab

# GitHub vs. GitLab vs. Bitbucket

	GitHub 	GitLab 	Bitbucket 
Open Source	✓	✓	✗
CI pipeline	✓	✓	✗
Own APIs	✓	✓	✗
Git platform	✓	✓	✓
Active Bug tracking	✓	✓	✗
Supports	✓	✓	✓

## Key Takeaways

- Version control is a system that record changes to a set of files.
- Git is a version control system for tracking changes in computer files.
- SVN and Perforce are centralized version control systems.
- The git config command allows you to configure your Git settings.

