



Introduction To GIT

OBJECTIVES



By the end of this session students are expected to

- Version control system
- Understand the purpose of git
- Understand how to install git
- Know basic git configuration
- Understand Basic Git workflow
- Know basic git commands
- Synchronization with GitHub



What is Version control system?

- Is the practice of tracking and managing changes to software code.
- Version control softwares help developers to Work together and maintain complete History of their work

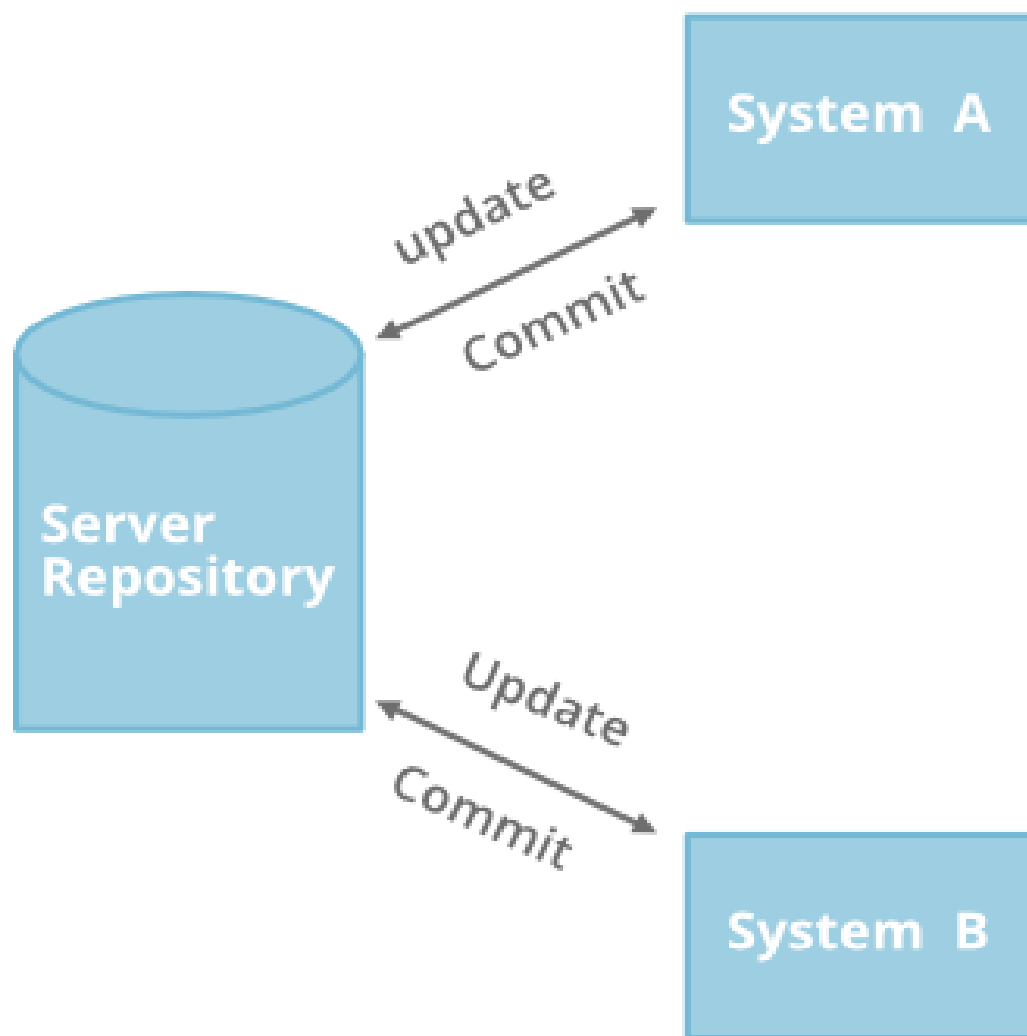
Examples

Git, Mericural, Subdivision and others are examples of version control software system

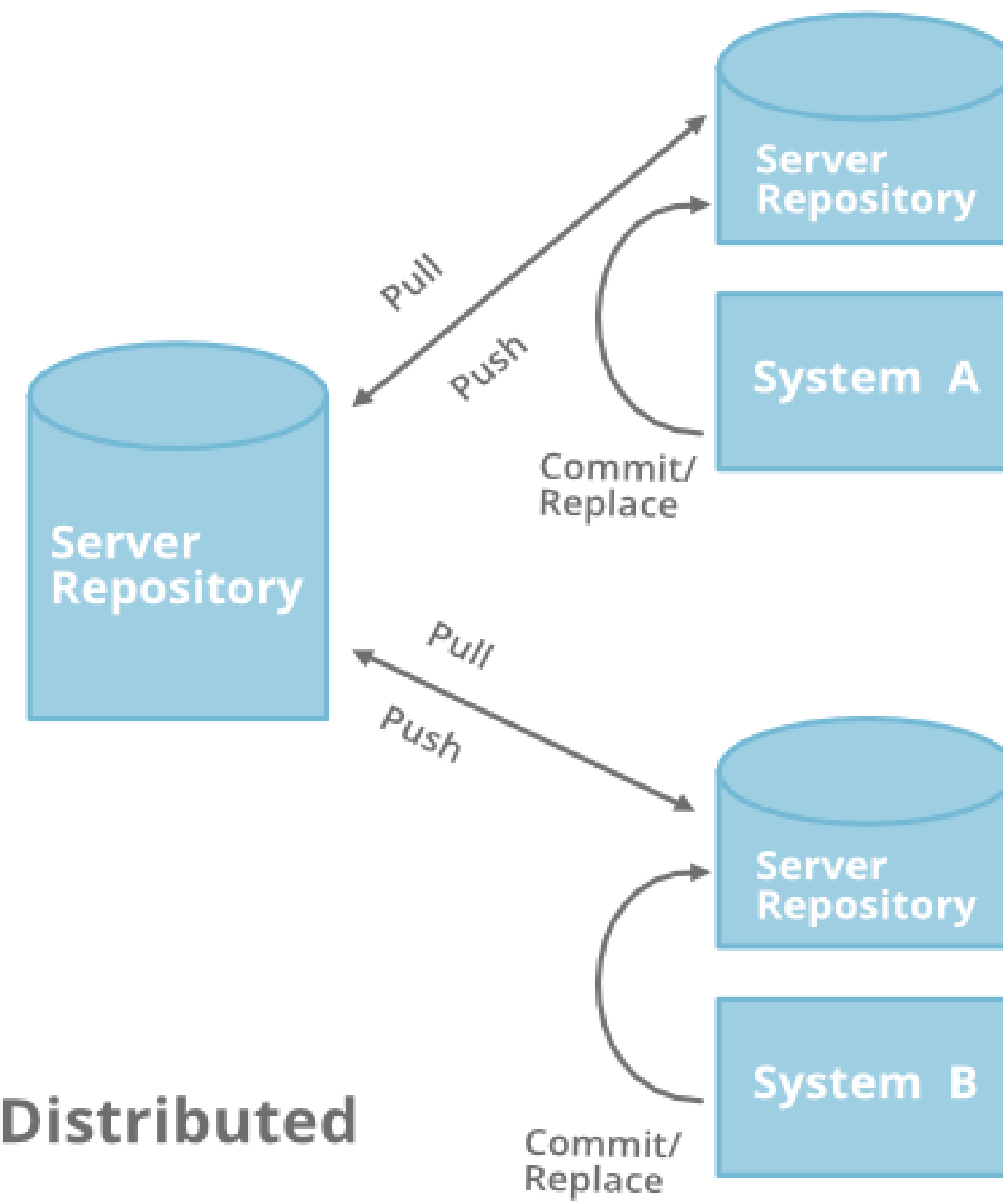


There are two types of version control system

- **A centralized version control system (VCS)**, also known as a centralized version control system, is a type of version control system that manages and tracks changes to files and projects. In a centralized VCS, there is a single central repository that stores all versions of the files, and multiple users can access and work with the files by checking them out from the central repository.
- **A distributed version control system (DVCS)** is a type of version control system that allows multiple copies of a codebase or project to exist independently on the local machines of contributors. Unlike centralized version control systems (VCS), where there is a single central repository, in a DVCS, each user has a complete copy of the entire project, including its history and all branches, on their local machine.



Centralized



Distributed

Advantages of using version control System



- It helps to track the progress of each update.
- It also functions either as a platform for improvement or culminates in a final version.
- Version control is beneficial in the development of digital assets,
- It plays a particularly important role when managing and keeping track of changes.

What is GIT?



- Git is software that keeps track of changes that you make to files and Codes.
- It allows you to move back and forth between versions of your changes
- It allows to compare different versions to see what changed.
- It is referred to as a version control system



Local and Remote Repositories in git

- **Repositories** refers to a data structure or storage location where a system stores and manages various versions of files and code, along with their associated metadata.
- There are two main types of repositories: **Local repositories** and **Remote repositories**.
- These repositories serve different purposes and work together to enable collaborative and version-controlled development.



Local Repository

- A local repository is a copy of the project's code and its complete version history that resides on your local computer.

Purpose

- The local repository allows you to work on your code and commit changes without needing a network connection or access to a central server. You can make changes, create branches, and experiment freely.



Remote Repository

- A remote repository is a separate repository that exists on a remote server, typically hosted on a platform like GitHub.

Purpose

- Remote repositories serve as a centralized location for sharing and collaborating on code. Multiple team members can access and contribute to the codebase through the remote repository.

Installing GIT



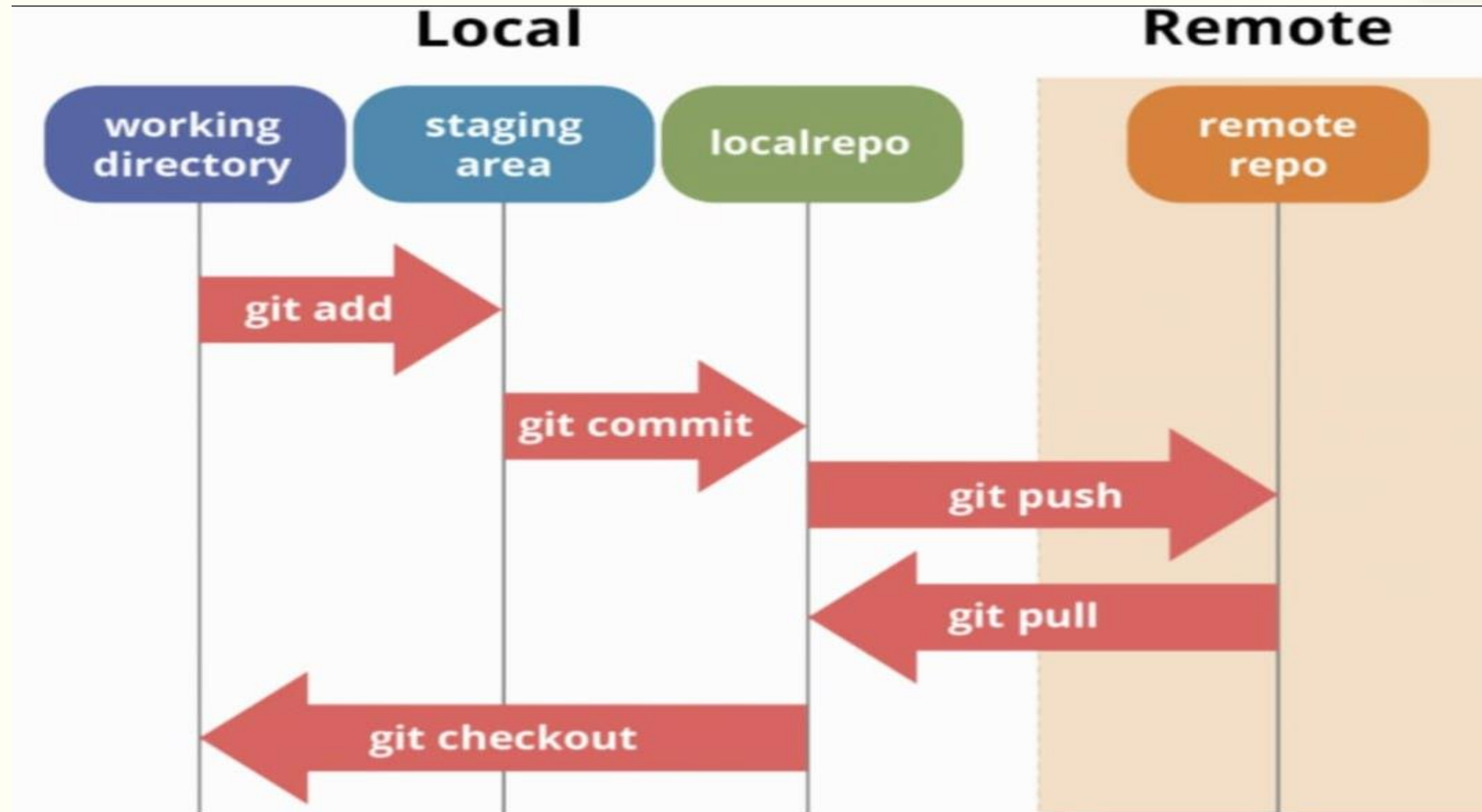
- To install git using web UI use the following link to download the distribution and follow the onscreen guide to finish installation
 - **<https://git-scm.com/downloads>**
- Create an account on GitHub
 - **https://github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home**

Basic GIT Configuration



- System configs apply to every user of the machine by default
- Project configs apply to only a specific project and will be located `project_name/.git/config`
- Git need to know who is making changes to track them
- Configuration commands:
 - `git config --list` (to see the list of configurations)
 - `git config --global user.name "Your name"` (For system level)
 - `git config --global user.email "Your email"` (For user level)
 - `git config` (For project level)

Basic GIT Fundamental Workflow



Working with Git



Initialize a repository:

- For git to track your changes you must initialize a repository.
- **git init** is a command we use to initialize a local repo
- This will create a directory named **.git**
- This is a directory where git keeps tracks of information related to your work, hooks, logs, configurations etc.
- Every project related to version control with git must have this directory
- The directory also track information related to remote repository

Cont...



- The **git status** command is a commonly used command in the Git version control system. It provides information about the current state of your Git repository, including the status of your working directory, untracked files, changes that have been staged but not committed, and the branch you are currently on.
- The **git add** command is used in Git, a version control system, to stage changes for commit. When you make changes to your project's files, whether they are new files or modifications to existing ones, you need to stage those changes before committing them. Staging allows you to select which changes you want to include in the next commit.

Cont...



git commit

The git commit command is used in Git to create a new commit with the changes that have been previously staged using the git add command. A commit is a snapshot of your project at a specific point in time, and each commit records a set of changes along with a commit message describing the purpose of the commit.

Basic GIT Commands

- Synchronize your changes with remote repository:
 - `git remote add origin https://github.com/project-name`
- To check the current status of the repo: `git status`
- To create a new branch: `git branch "branch name"`
- To switch to another branch: `git switch "branch name"`
- To switch to a branch with specific commit: `git checkout "commitid"`

Cont...



- To restore a file to last commit: `git restore "filename"`
- To combine two branches: `git merge "branchname"`
- To see commit id and message: `git log --online`
- To undo something with commit id: `git revert commitid`
- Before adding any changes in an existing repository use `git pull` this will bring all the changes made by other developers to your local repo and avoid conflicts

Summary



- Git is software that keeps track of changes that you make to files and Codes.
- Basic GIT Fundamental Workflow
- Basic GIT Configuration

Working with git:

- Initialize a repository
- Make changes to your code (File)
- Add (Stage) your changes to the repo
- Commit your changes:
- Synchronize your changes with remote repository

THANK YOU