Software Development

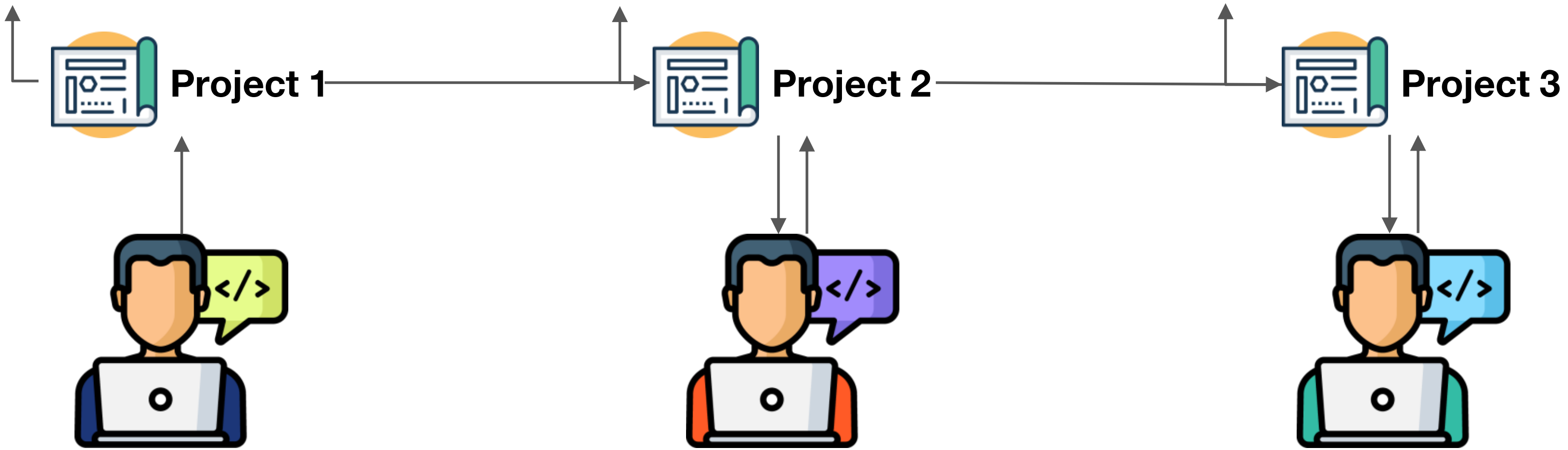
Is software development a single player game or a multi player game?

Software Development Scenario

Jim works on a project, saves it as copy1, and sends it to Peter.

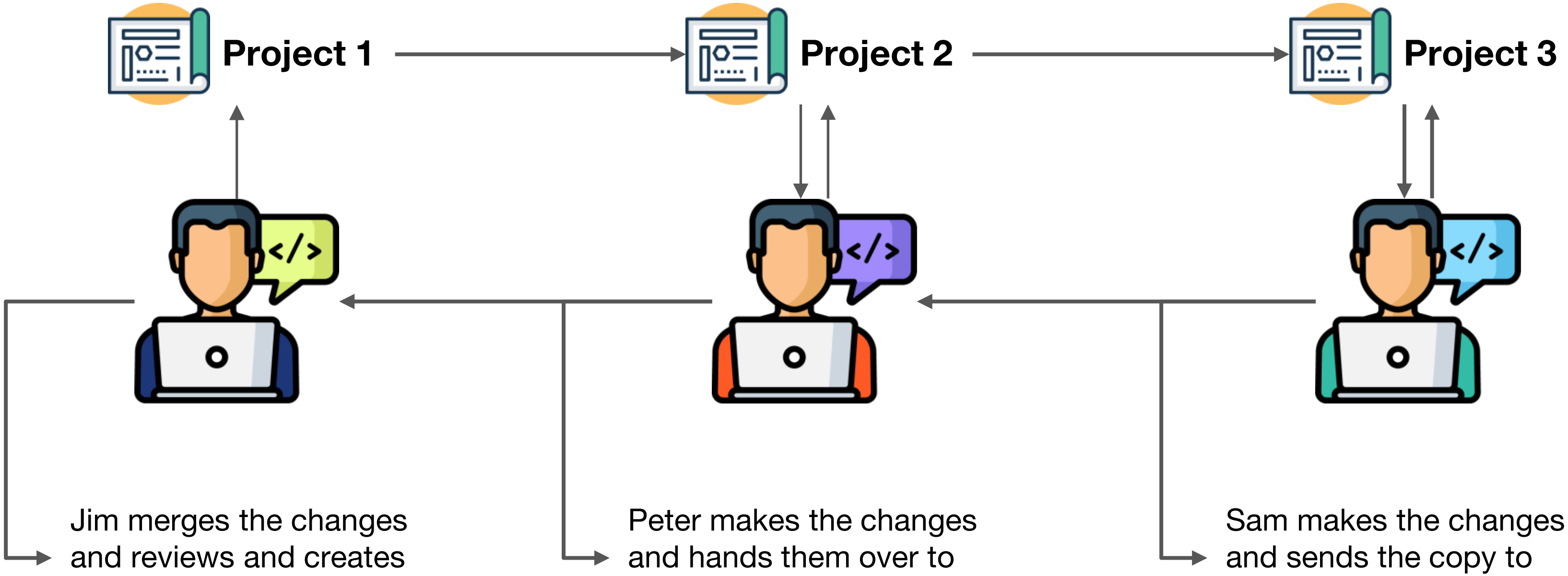
Peter makes changes to Jim's project, saves it as copy2, and sends it to Sam.

Sam makes further changes to the project and saves it as copy3.



Which is the main copy of the project?
What is the problem with this approach?
How will the changes made by all three developers get incorporated in the final copy of the project?

Scenario: Possible Solution 1



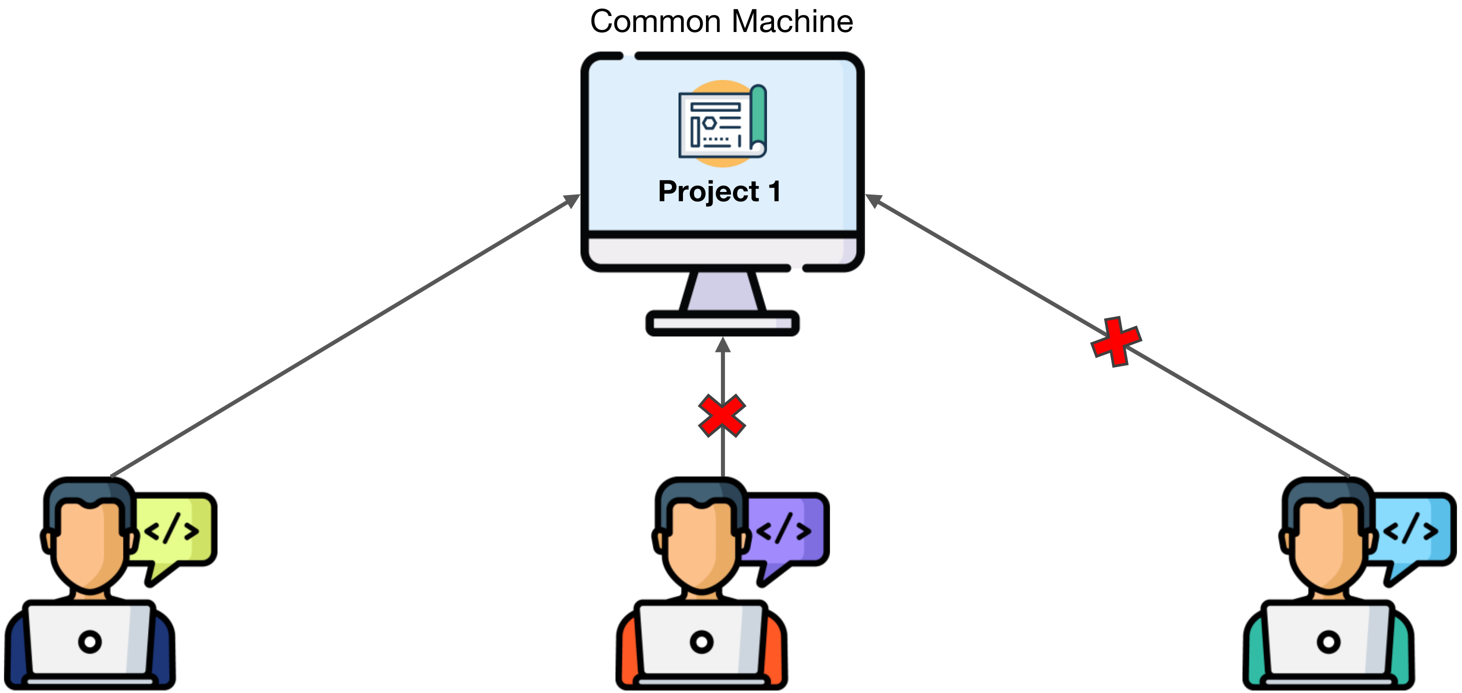
Jim merges the changes and reviews and creates the final copy.

Peter makes the changes and hands them over to Jim.

Sam makes the changes and sends the copy to Peter.

Is this an efficient way to handle this problem?
Can we depend on this solution?

Scenario: Possible Solution 2



Jim is working on the project.

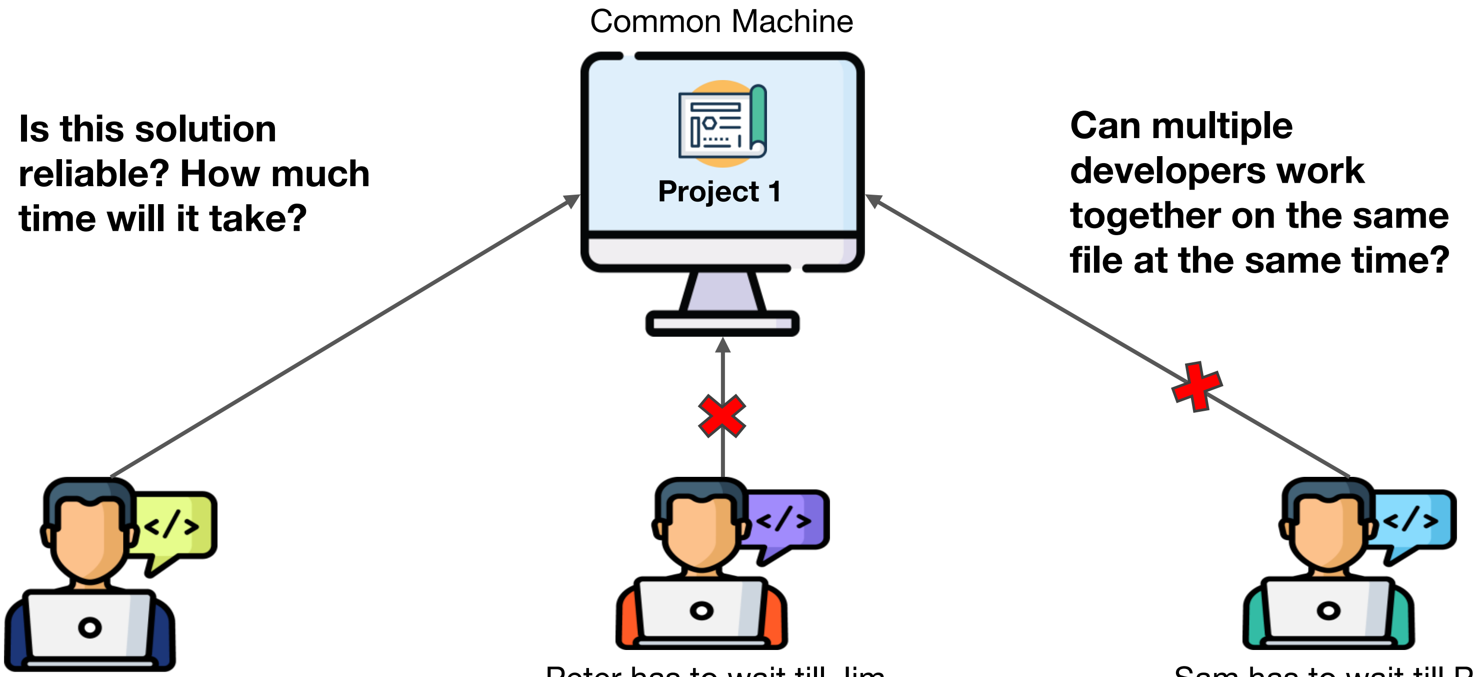
Peter has to wait till Jim completes his work.

Sam has to wait till Peter completes his work.

Scenario: Possible Solution 2 contd.

Is this solution reliable? How much time will it take?

Can multiple developers work together on the same file at the same time?



Common Machine

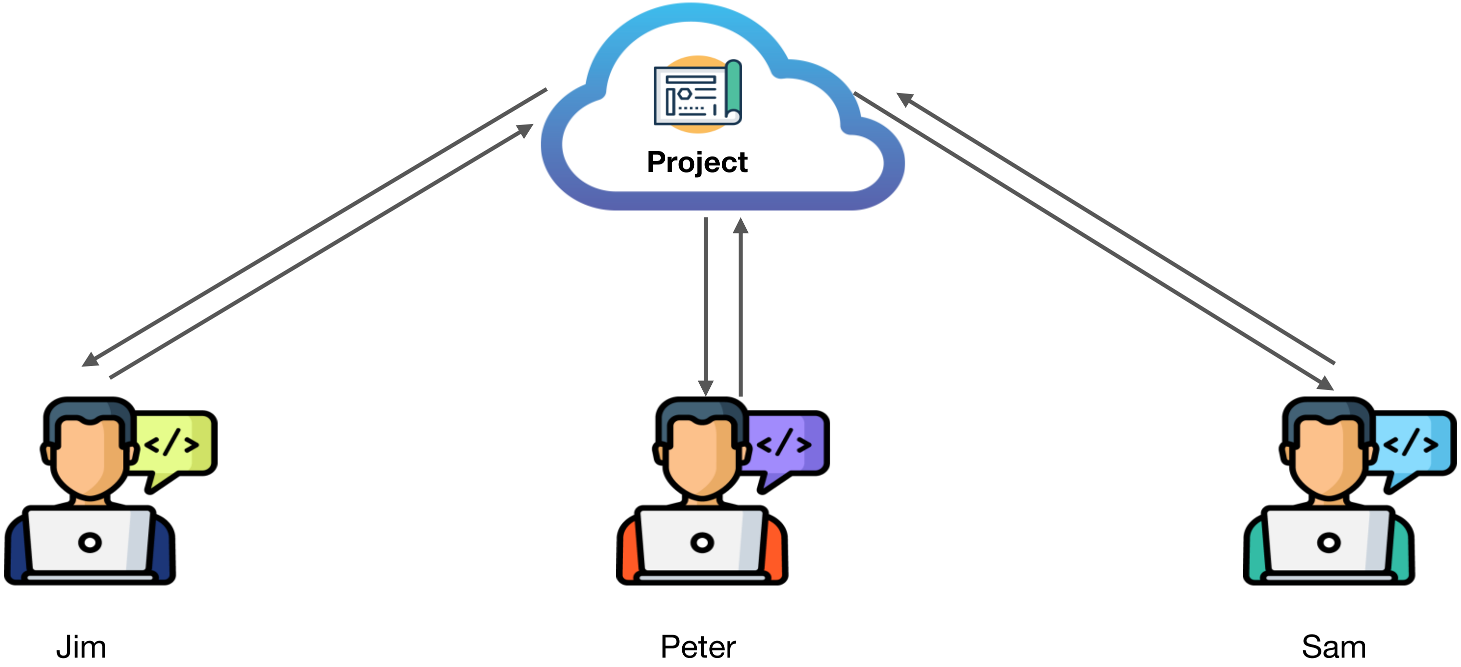
Project 1

Jim is working on a project.

Peter has to wait till Jim completes his work.

Sam has to wait till Peter completes his work.

Introduction to a Version Control System

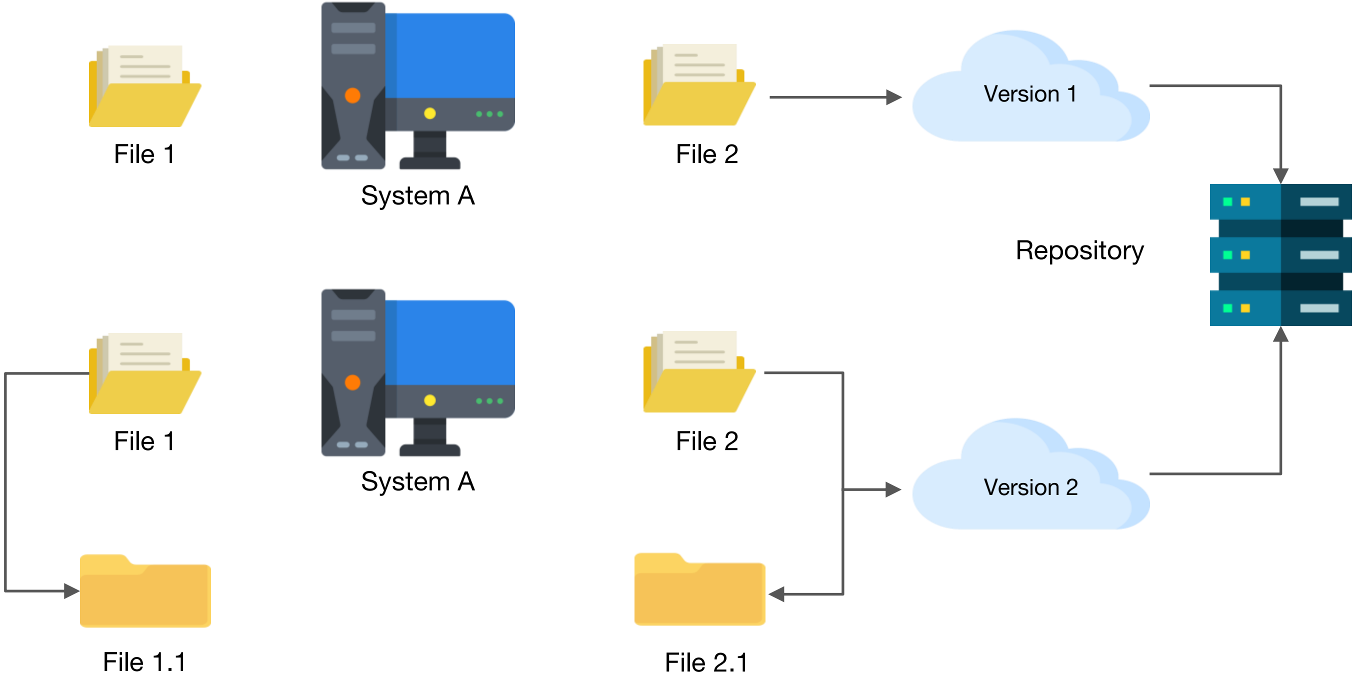


Create Code Repository in Git

Learning Objectives

- Define a version control system
- Use Git as a distributed version control tool
- Use GitLab as a version control service
- Fork a repository
- Clone a repository
- Modify, commit, and push the changes to the repository

Working of a Version Control System



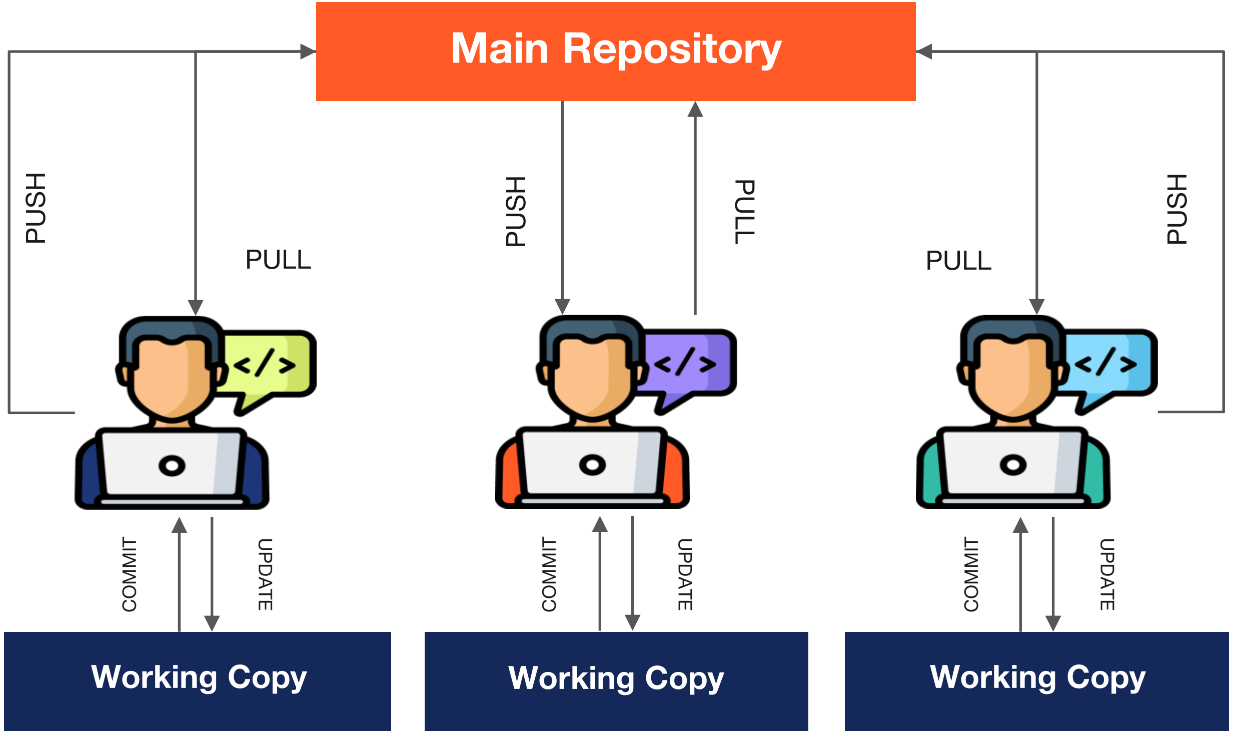
Different Types of Version Control Systems

Localized

Centralized

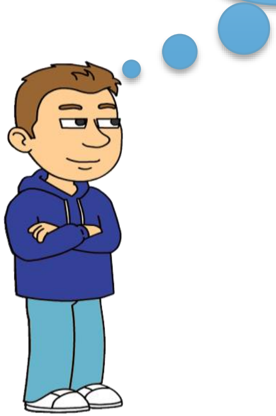
Distributed

A Distributed Version Control System

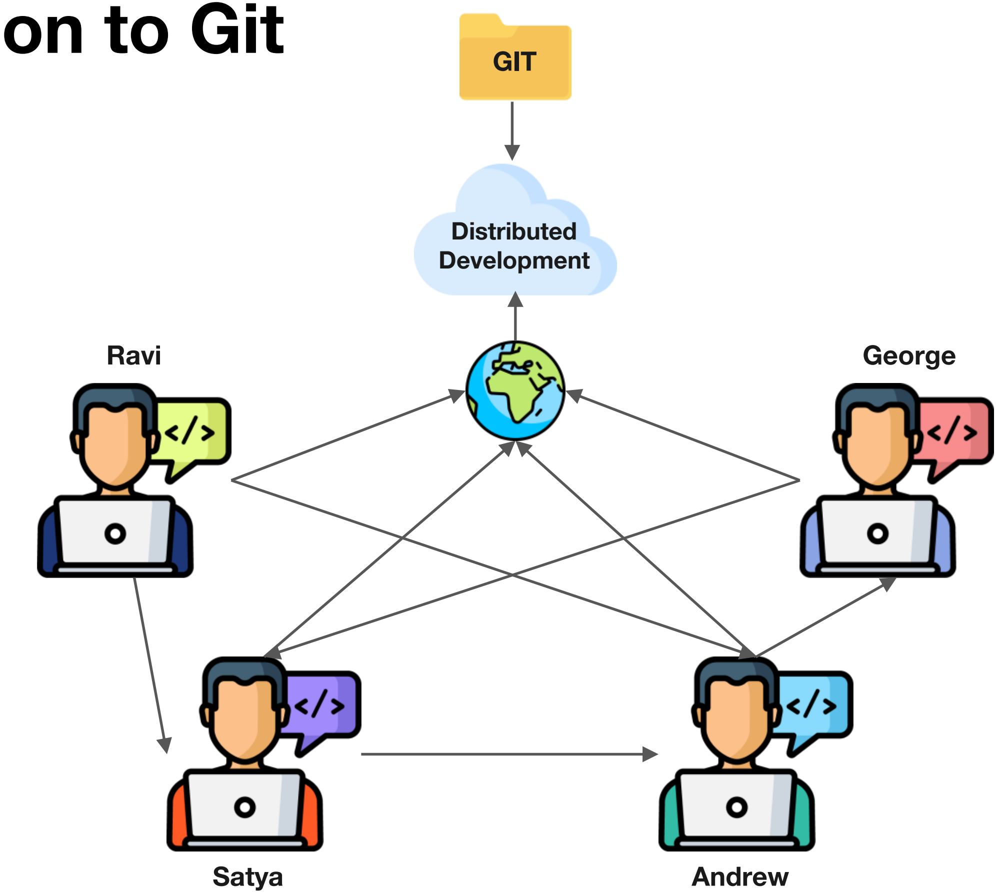


Distributed Version Control System

I am planning to use a distributed version control system for my project. Which tool should I use?



Introduction to Git



Source: http://blogofcode.com/

Features of Git

- It uses SHA1 generated hashes to protect codes and identify revisions.
- It easily detects data corruption caused due to external reasons.
- It is fast, flexible, and freely available.
- It supports easy branching and parallel development.

What is GitHub?

Introduction to Gitlab

Git vs. GitLab

Git	GitLab
Git is a distributed version control system tool.	GitLab is a cloud-based collaboration platform used to host a Git repository.
It can be installed in a local system.	It is hosted on the web.
It is a command-line tool.	It is administered through the web.
No user management	Built-in user and access management

Using Git and GitLab

How can we implement Git and GitLab?

- Configure Git to your local system.
- Create a new project using the GitLab service.
- Fork the repository you wish to work on.
- Clone the repository in your local system after forking.
- Make the changes in the repository.
- Push the changes to the remote repository.

Interactive Demo

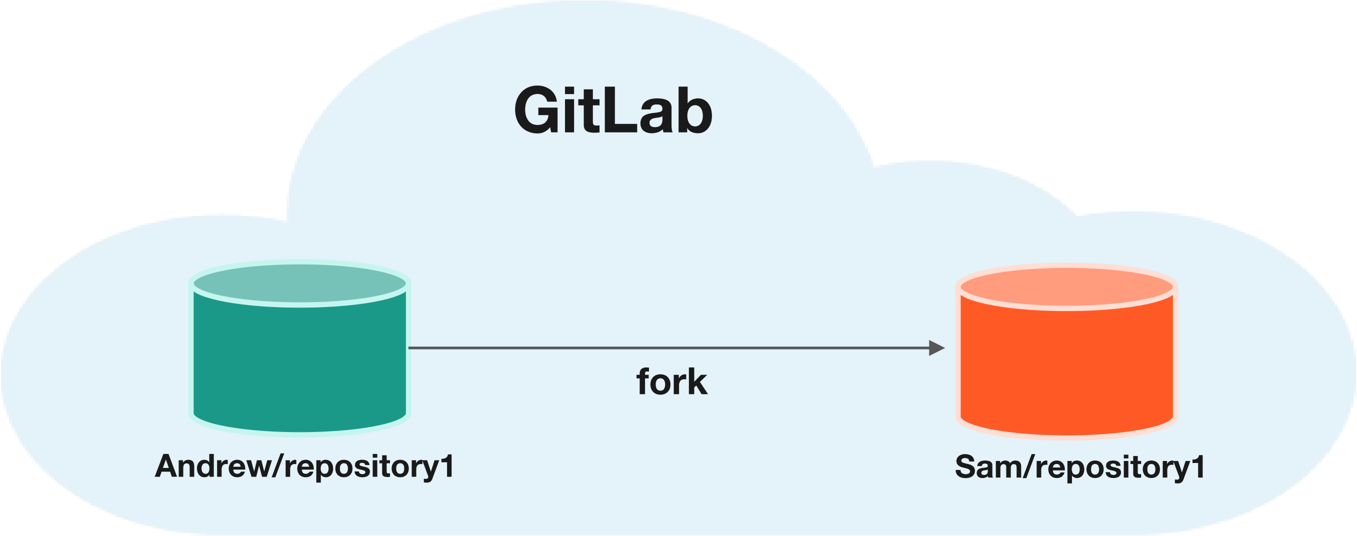
Let us try to

- Configure the Git environment in the local system.
- Create a new project in GitLab.

How Do I Make Changes in the Main Repository?

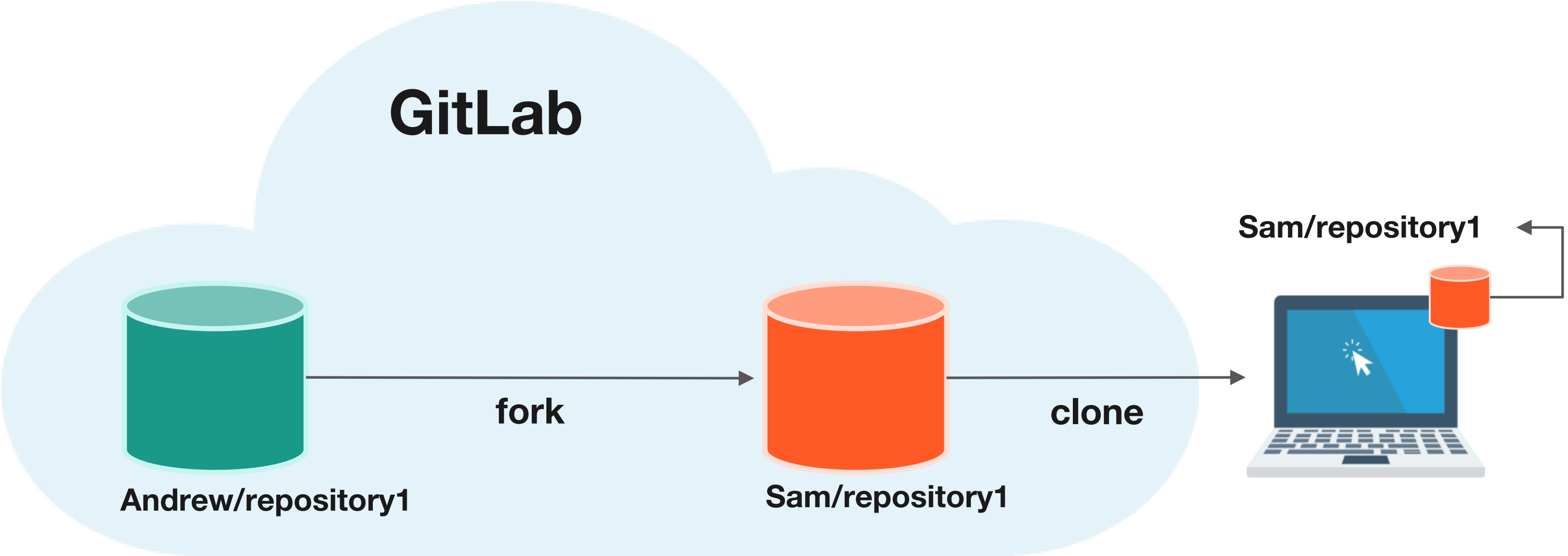
What steps should I take to incorporate the changes of my project in the main repository?

Forking a Repository



Users can fork a centralized repository by using Git. They can create a copy of the repository and make changes into their GitLab account without affecting the main repository.

Cloning a Repository



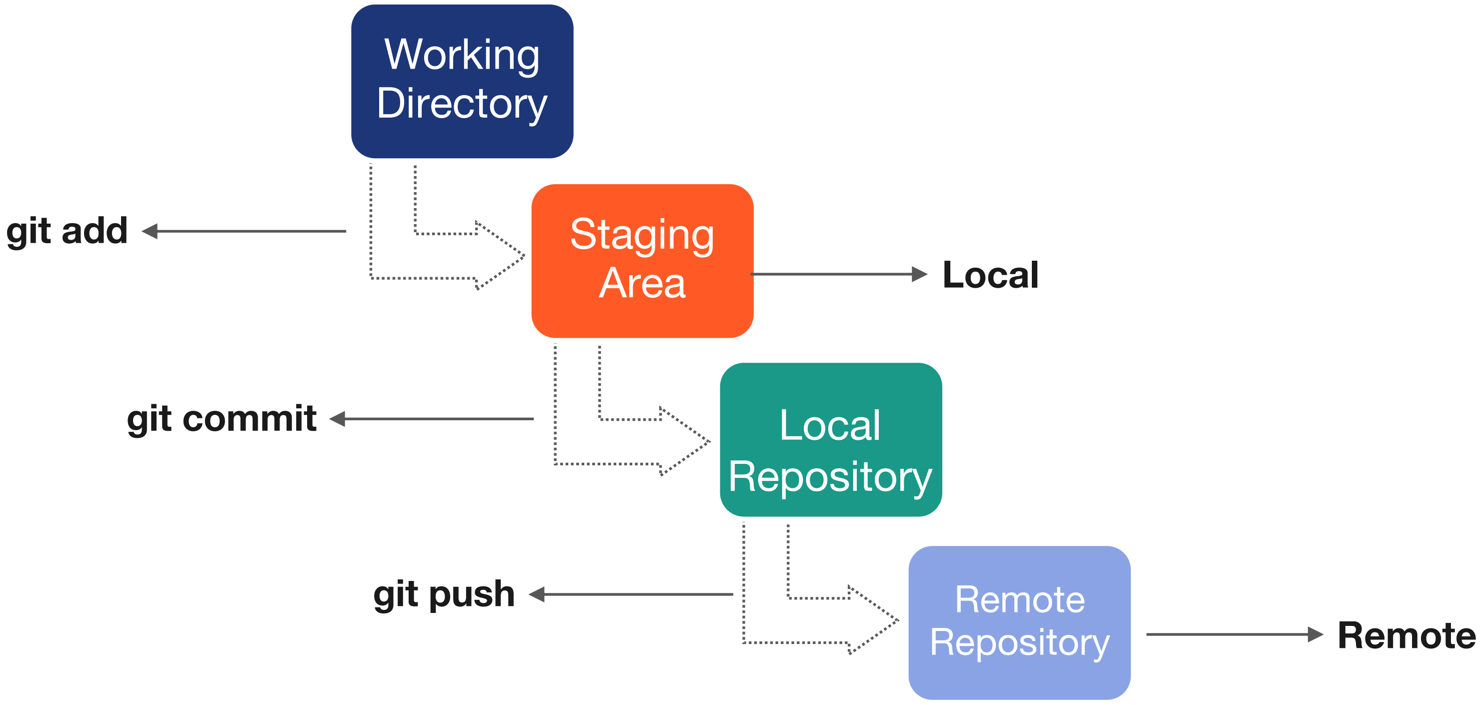
Users can clone a repository to their local system.

Interactive Demo

Let us try to

- Fork a repository
- Clone a repository

Push Changes to a Remote Repository



You can push the committed files to the remote repository.

Interactive Demo

Let us try to

- Make changes to the local repository that has been cloned.
- Commit changes to the repository.
- Push changes to the remote repository.

Key Takeaways

- Types of version control systems
- Working of a distributed version control system
- Differentiate between Git, GitHub, and GitLab
- Forking and cloning a repository
- Modify, commit, and push the changes to the repository

Thank you!