Version Control with Git

Ruvani Jayaweera (PG dip) (University of Moratuwa), ((BSc) hons (University of Moratuwa)

Learning Objectives

Be able to understand version control with Git.

Outline

- What is a Version Control System?
- Key Characteristics of VCS
- Centralized VCS
- Decentralized VCS
- What is Git?
- Key Features of Git
- Sign Up with GitHub
- Sign in to the GitHub
- Repository Creation
- Git with Eclipse

What is a Version Control System?

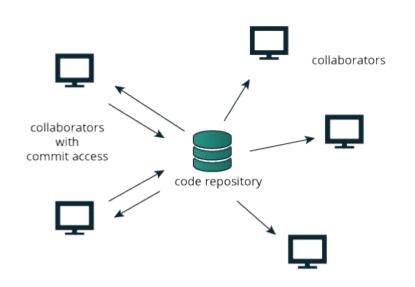
- A version control system (VCS) is a software tool designed to monitor changes to files across time.
- It maintains a record of revisions and modifications made to files, allowing users to collaborate on projects, monitor changes and effectively handle multiple file versions

Key Characteristics of VCS

- Tracking Changes: A VCS documents every modification to files, encompassing additions, deletions and modifications.
- Version History: It maintains a sequential record of alterations, enabling users to revert to prior versions as needed.
- Collaboration: VCS promote teamwork by offering a centralized repository where team members can access, contribute to and synchronize files.

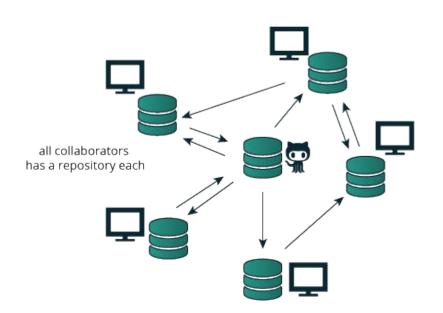
- Branching and Merging: It supports branching, enabling users to concurrently work on different file versions. Merging integrates changes from various branches back into the main branch.
- Conflict Resolution: VCS aids in resolving conflicts that arise when multiple users make conflicting changes to the same file.

Centralized VCS



All team members are connected to a central server. Developers must write directly to the centralized repository and read the code directly from the server. If the server goes offline or experiences downtime, collaboration among team members is not possible.

Decentralized VCS



Each team member has a local copy of the repository on their own computers, effectively having project snapshots stored locally.

What is Git?

- Git is a distributed version control system (DVCS) used for tracking changes in source code and other files.
- It was created by Linus Torvalds in 2005 to manage the development of the Linux kernel.

Key Features of Git

- **Distributed Architecture**: Each user has a full copy of the entire repository, including its history, on their local machine. This allows for offline work and provides built-in redundancy.

- **Branching and Merging**: Git supports lightweight, efficient branching and merging, allowing users to create, switch and combine branches easily. This facilitates parallel development and experimentation.

- Speed and Performance: Git is designed to be fast, handling large projects with speed and efficiency.

Operations like commits, diffs, and merges are performed quickly.

- **Data Integrity**: Git uses a cryptographic hash function (SHA-1) to ensure the integrity of the repository data. Each commit is uniquely identified by a hash, ensuring the history is secure and tamper-proof.

- Collaboration: Git enables collaboration by allowing multiple developers to work on the same project simultaneously. Changes can be shared and merged seamlessly through push and pull operations.

- Free and Open Source: Git is free to use and open source, meaning its source code is available for anyone to inspect, modify and distribute.

Vocabulary

- Repository/Repo:

This is where Git saves all the data, including the history of changes that are made.

- Working Directory:

This is the folder on the computer where the project resides. Git tracks any changes made in this folder.

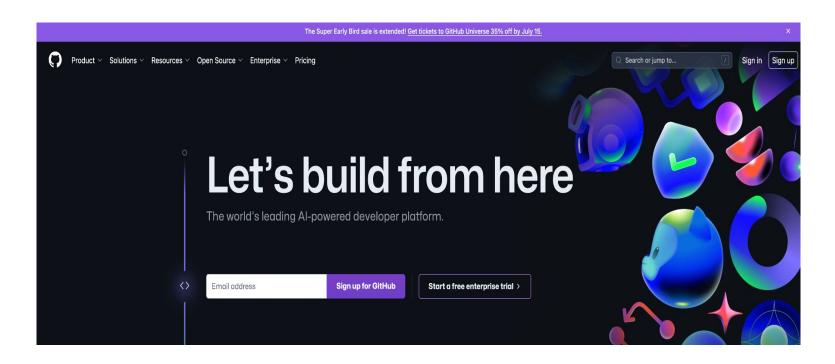
- Staging:

Before saving changes from the working directory to the Git repository, it is necessary to stage them. Staging is a selective process where you prepare specific changes to be tracked.

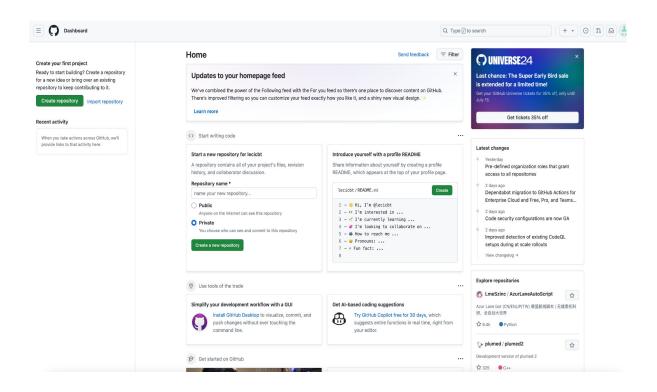
- Commit:

Committing is Git's way of saving changes. Git does not store changes made to the working directory until the user commits them. This can be considered a secondary level of saving.

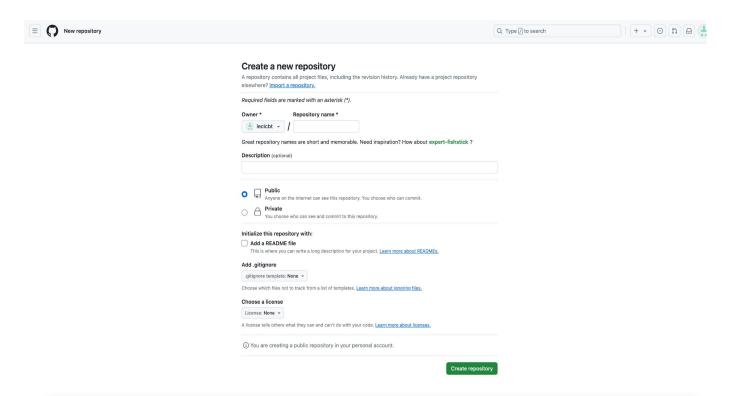
Sign Up with GitHub



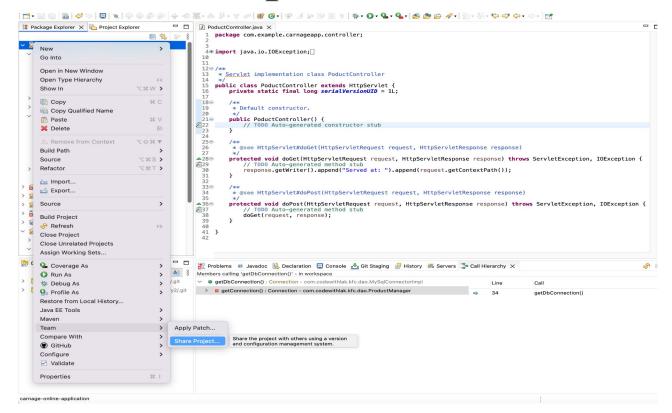
Sign in to the GitHub

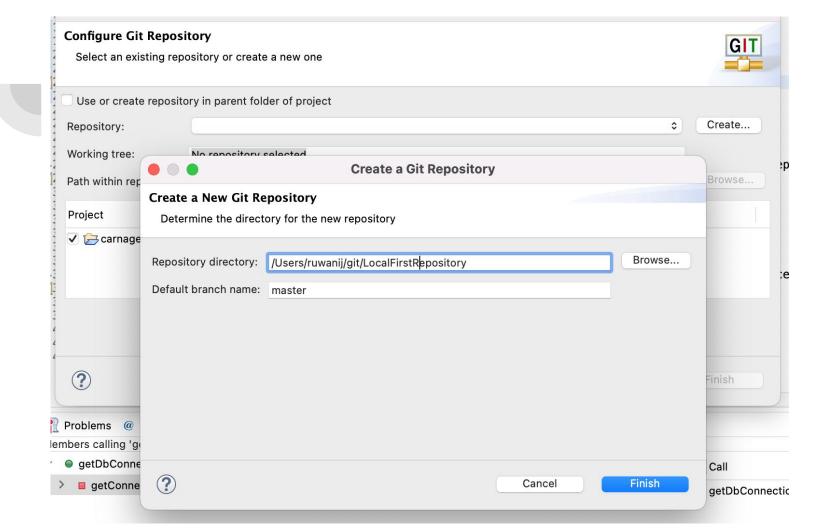


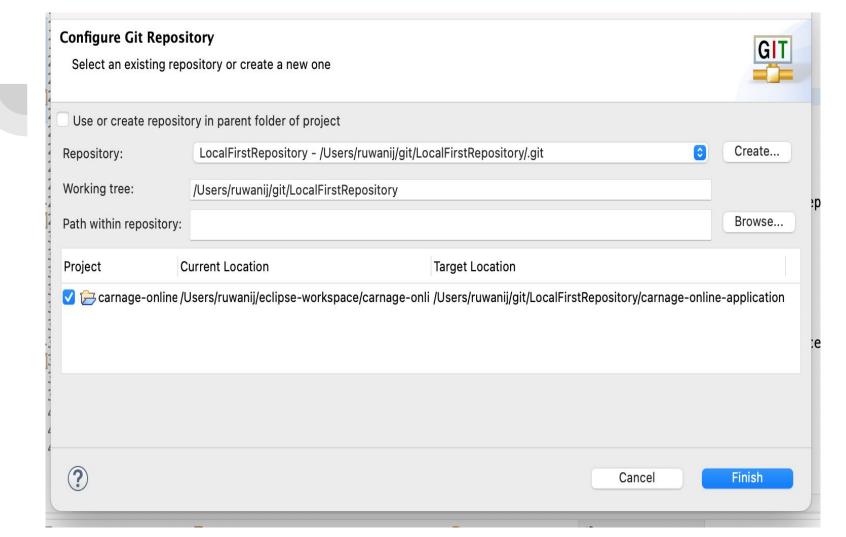
Repository Creation

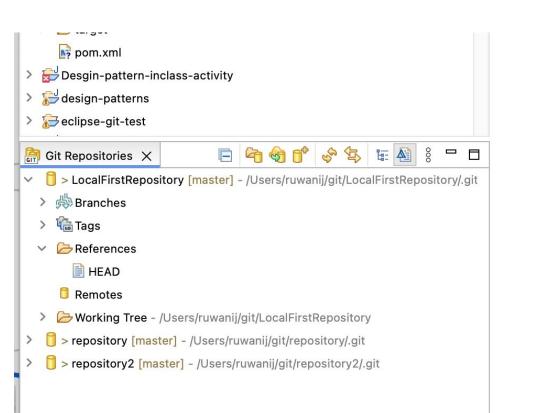


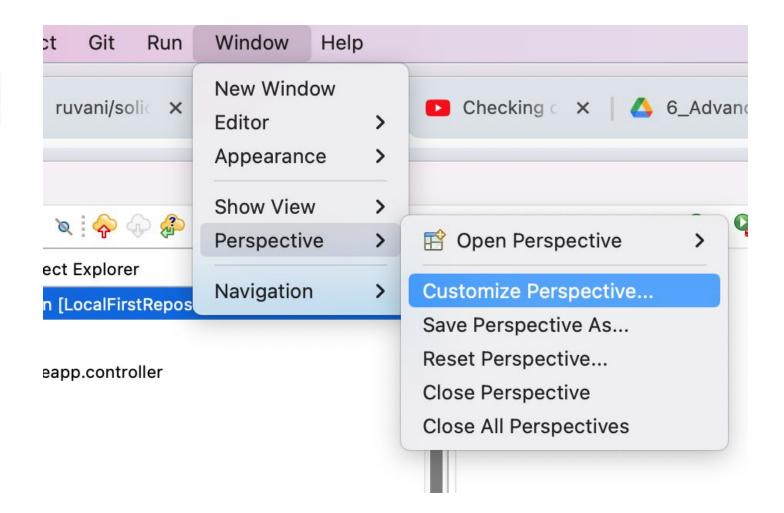
Git with Eclipse

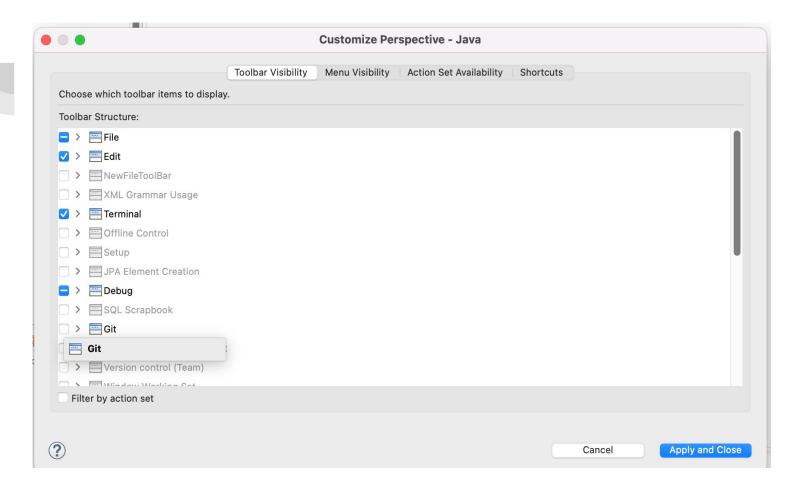


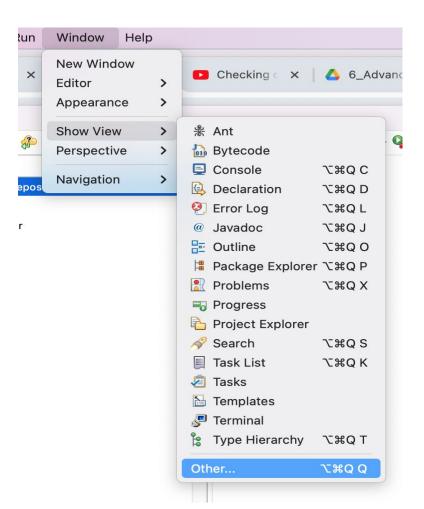


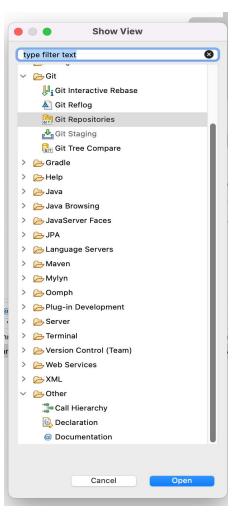


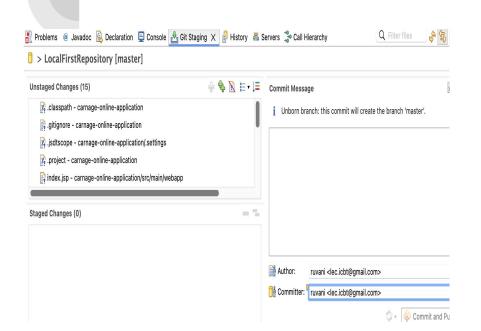


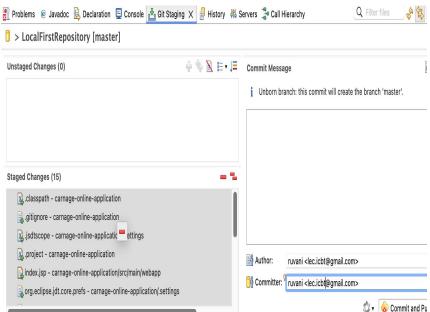


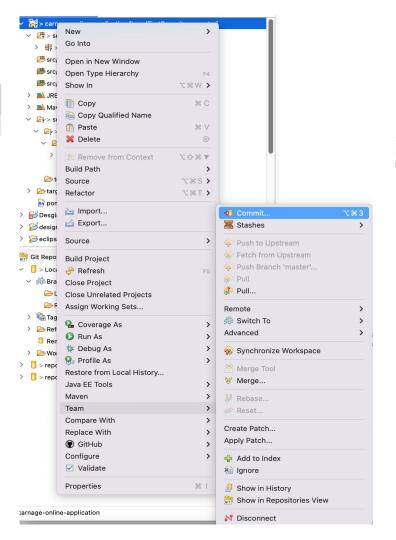




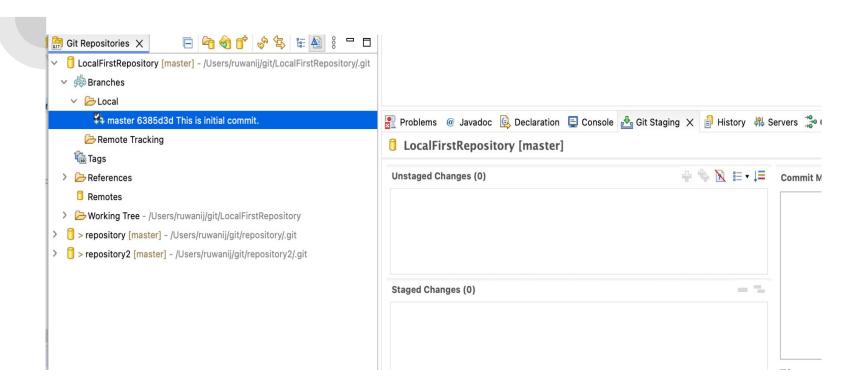


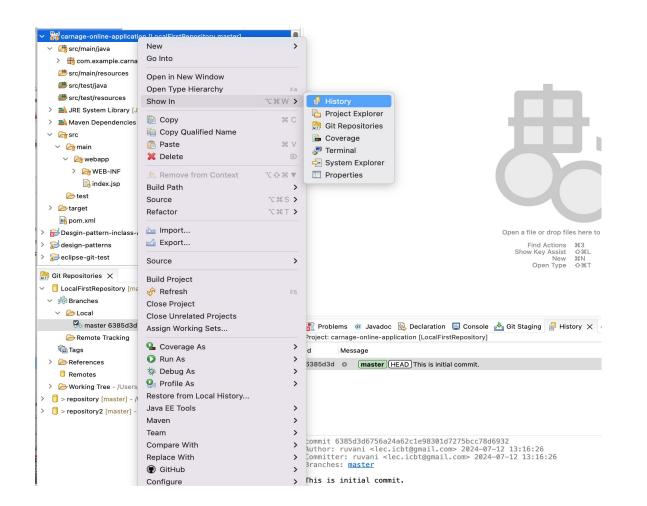


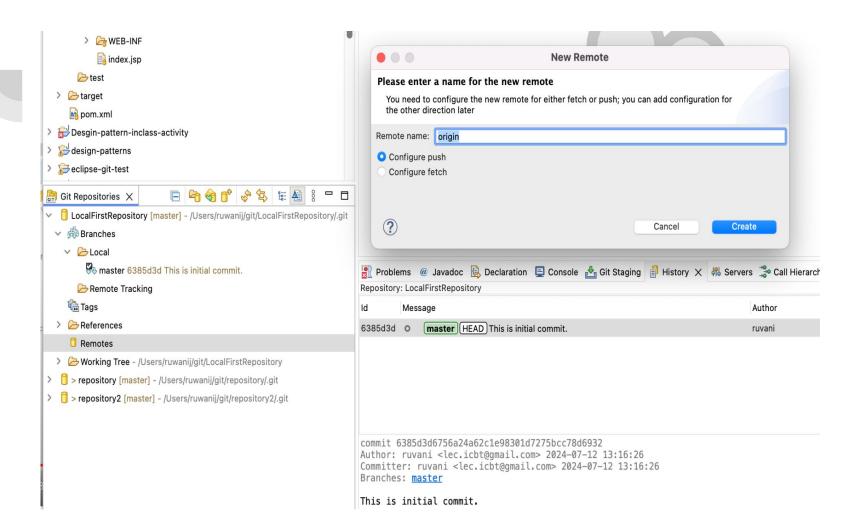


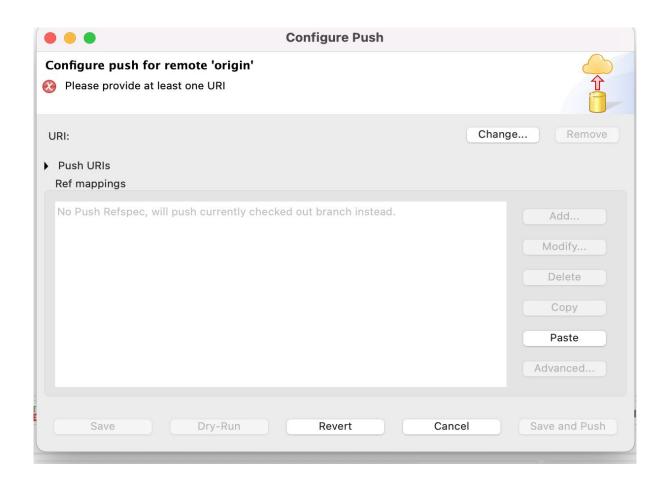


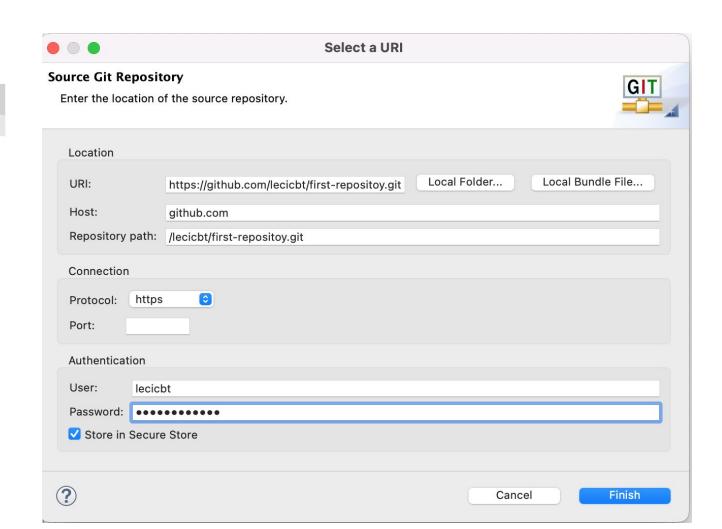


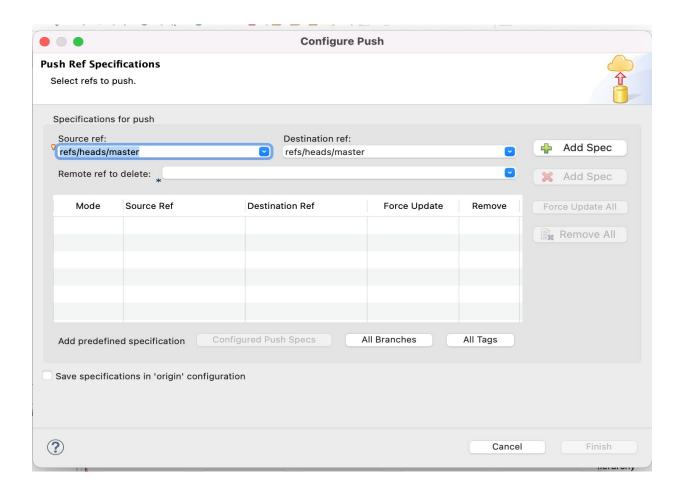


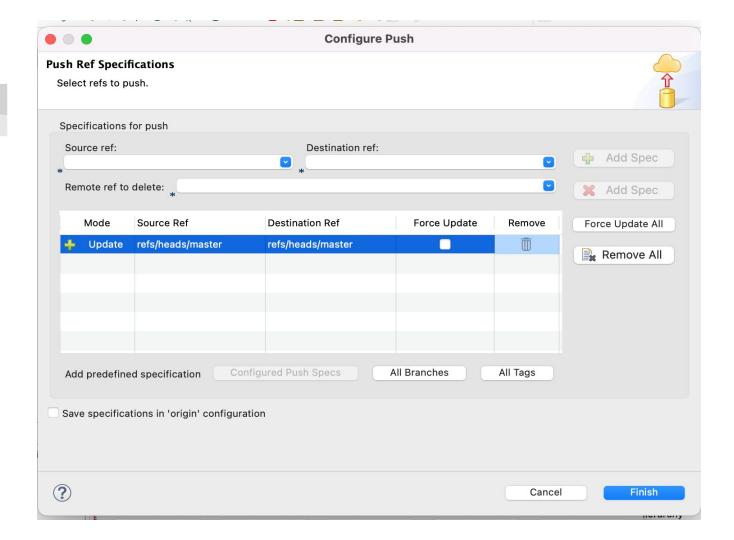


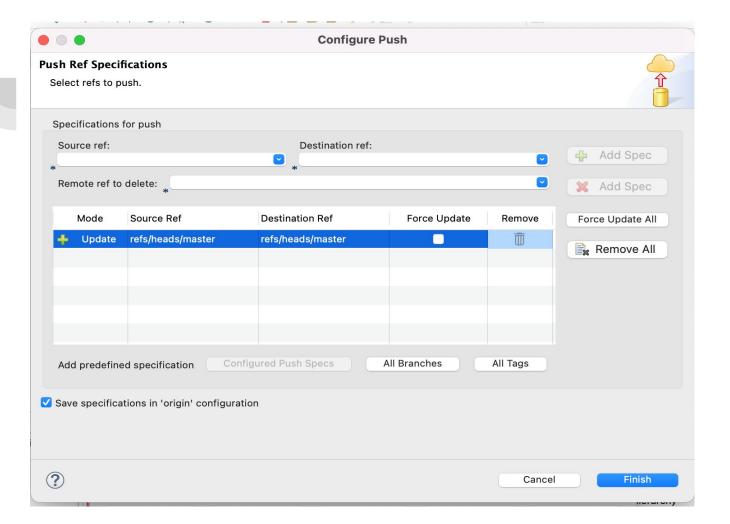


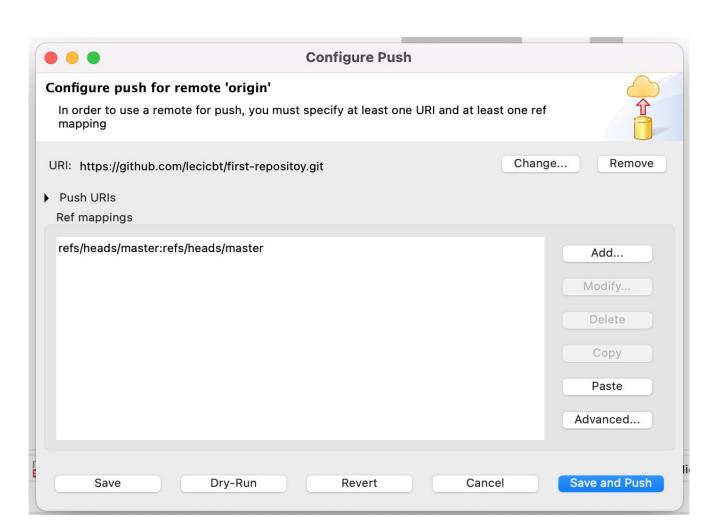
































- > LocalFirstProject [master] /Users/ruwanij/git/LocalFirstProject/.git
 - Branches
 - Tags
 - References

Remotes

- n origin
 - https://github.com/lecicbt/first-repositoy.git
 - https://github.com/lecicbt/first-repositoy.git
- > Working Tree /Users/ruwanij/git/LocalFirstProject
- > repository [master] /Users/ruwanij/git/repository/.git
- > repository2 [master] /Users/ruwanij/git/repository2/.git

Any Questions?

Thank You!