Repository:In GitLab, files are stored in a repository. A repository is similar to how you store files in a folder or directory on your computer.

A remote repository refers to the files in GitLab. A local copy refers to the files on your computer.

Often, the word “repository” is shortened to “repo”. In GitLab, a repository is contained in a project.

Fork:When you want to contribute to someone else’s repository, you make a copy of it. This copy is called a fork.

The process is called creating a fork.

Difference between download and clone:

To create a copy of a remote repository’s files on your computer, you can either download or clone the repository. If you download it, you cannot sync the repository with the remote repository on GitLab.

Cloning a repository is the same as downloading, except it preserves the Git connection with the remote repository. You can then modify the files locally and upload the changes to the remote repository on GitLab.

Pull and push:

After you save a local copy of a repository and modify the files on your computer, you can upload the changes to GitLab. This is referred to as pushing to the remote, because you use the command git push.

When the remote repository changes, your local copy is behind. You can update your local copy with the new changes in the remote repository. This is referred to as pulling from the remote, because you use the command git pull.

git --version -->Determine if Git is already installed on your computer

Configure Git:

In your shell, add your user name: git config --global user.name "your\_username"

Add your email address: git config --global user.email "your\_email\_address@example.com"

To check the configuration, run: git config --global --list

Clone with HTTPS:

Clone with HTTPS when you want to authenticate each time you perform an operation between your computer and GitLab.

Go to your project’s landing page and select Clone. Copy the URL for Clone with HTTPS.

Open a terminal and go to the directory where you want to clone the files.

Run the following command. Git automatically creates a folder with the repository name and downloads the files there.

git clone <https://gitlab.com/gitlab-tests/sample-project.git>

GitLab requests your username and password:

If you have 2FA enabled for your account, you must use a Personal Access Token with read\_repository or write\_repository permissions instead of your account’s password.

If you don’t have 2FA enabled, use your account’s password.

To view the files, go to the new directory:

cd sample-project

git init --> Convert a local directory into a repository

git add <file-name OR folder-name> --> To stage a file for commit

git status --> Confirm that the files have been added to staging

git commit -m "COMMENT TO DESCRIBE THE INTENTION OF THE COMMIT" --> To commit the staged files

git commit -a -m "COMMENT TO DESCRIBE THE INTENTION OF THE COMMIT" --> Stage and commit all changes

git push <remote> <name-of-branch> --> To push all local changes to the remote repository

git push origin main --> to push your local commits to the main branch of the origin remote

git checkout . --> To discard all changes to tracked files

git reset --> To unstage (remove) all files that have not been committed

Merge a branch with default branch:

When you are ready to add your changes to the default branch, you merge the feature branch into it:

git checkout <default-branch>

git merge <feature-branch>

git log --> Viewing the Commit History

git rm <file\_name> --> removing file

git log --oneline --graph --decorate --> output shows in proper format

git show <id> --> Gives whole histroy of that specific Id

git diff --> Show changes between commits, commit and working tree, etc

git branch <branch\_name> --> Create a new branch

git branch -d <branch\_name> --> delete a branch

git branch -m <old\_name> <new\_name> --> to rename the branch