Understanding GitLab: A Beginner's Guide

What is GitLab?

GitLab is a web-based platform that provides version control for software development projects using Git. It allows you to:

- Track changes to your code over time.
- Collaborate with others on projects.
- Deploy your applications.

Key Concepts: Git vs. GitLab

- **Git:** A powerful tool for version control, keeping a history of changes to your code. You use Git commands locally on your computer.

  Such as:

  - git clone <repo link>
  - git add .
  - git commit -m “new changes”
  - git push

- **GitLab:** A platform that hosts Git repositories, providing a user interface and additional features for collaboration, issue tracking, CI/CD (continuous integration and continuous delivery), and more.

Getting Started with GitLab

Prerequisites:

- A computer with an internet connection
- A free GitLab account (https://about.gitlab.com/)

Steps:

1. **Create a GitLab Account:** Sign up for a free account on GitLab.com or a self-hosted instance.
2. **Install Git:** Download and install Git on your computer from the official website: https://git-scm.com/download/win.
3. **Set Up SSH Keys (Optional but Recommended):** For secure communication with GitLab, generate SSH keys and add them to your GitLab account for passwordless authentication. Follow the instructions here: https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account.

Creating and Managing Your First Project

1. **Create a New Project:** In the GitLab interface, click "Create project" and provide a name and description for your project. Choose public or private visibility based on your project's nature (public allows anyone to view your code, private requires access permissions).
2. **Version Control with Git Commands:**
    - `git init`: Initialize a local Git repository for your project directory. Open your terminal or command prompt, navigate to your project directory, and type this command.
    - `git add <filename>`: Add files you want to track by Git. Replace `<filename>` with the actual file name(s) you want to add.
    - `git commit -m "Your commit message"`: Create a snapshot of your changes with a descriptive message. This message helps you understand the changes made in a particular commit later.
3. **Connecting to a Remote Repository:**
    - **URL Retrieval:** Get the HTTPS URL for your GitLab project's repository by clicking on "Settings" -> "General" -> "Repository URL" in your project.
    - `git remote add origin <URL>`: Add the remote repository URL to your local project. Replace `<URL>` with the actual URL you copied from GitLab.
4. **Pushing Changes to GitLab:**
    - `git push origin master`: Push your local commits to the `master` branch on GitLab. This command transmits your local changes to the remote repository on GitLab.

Collaborating with Others

1. **Forking Repositories:** Create a copy of a project (fork) by clicking the "Fork" button on the project page. This allows you to make changes without affecting the original project. You can then push your changes to your fork and create merge requests for review and integration.
2. **Merge Requests:** Create a merge request to propose changes from your fork to the main project. This allows for team review and discussion before merging your changes into the main project branch.
3. **Issues and Labels:** Create issues to track tasks, bugs, or feature requests related to your project. You can assign issues to specific team members and use labels to categorize and organize them.

Additional GitLab Features for Beginners

- **Gitlab Pages:** Host static websites directly from your GitLab repositories. This can be useful for deploying simple websites or documentation for your project.
- **Issue Boards:** Visually manage your project workflow using Kanban boards. You can drag and drop tasks between different stages like "To Do," "In Progress," and "Done" for easy organization.

- **CI/CD (Optional for Beginners):** GitLab offers tools for automating builds, tests, and deployments based on code changes. This is a more advanced topic you can explore once comfortable with the basics.
- **GitLab Pages:** Host static websites directly from your GitLab repositories. This can be useful for deploying simple websites or documentation for your project.

Beyond the Basics

The GitLab documentation and tutorials provide extensive resources to explore advanced Git commands, GitLab features, and best practices: https://docs.gitlab.com/