RAJ KUMAR GOEL INSTITUTE OF TECHNOLOGY
5th K.M. STONE, DELHI-MEERUT ROAD, GAZIABAD-(U.P.) 201003

Mastering Git, GitHub, and Version Control

By Rohit Kushwaha

Introduction

Welcome to the Workshop on Mastering Git, GitHub, and Version Control

My name is Rohit Kushwaha, 3rd year CSE from RKGIT. I have been working with Git for over quite some time now and gathered enough knowledge to know my way around Repos.

Workshop Overview

In this workshop, you will learn the fundamentals of Git, how to effectively use GitHub for collaboration, and best practices for version control. By the end of this workshop, you will be able to:

- Understand the basics of version control and why it is essential for software development.
- Use Git commands to create and manage repositories, branches, and commits.
- Collaborate with others using GitHub, including pull requests and code reviews.
- Implement best practices for version control to ensure a smooth and efficient development process.

Let's get started and dive into the world of Git and version control!

Understanding Version Control

Definition of Version Control

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. It allows multiple people to work on a project simultaneously and tracks changes made by each individual.

Importance in Software Development

Version control is crucial in software development as it provides the following benefits:

- Allows developers to track changes and revert to previous versions if necessary.
- Facilitates collaboration by enabling multiple developers to work on the same project simultaneously.
- Provides a centralized repository for all project files, ensuring that everyone has access to the latest version.
- Helps in identifying and resolving conflicts when multiple people make changes to the same file.

Introduction to Git

What is Git?

Git is a distributed version control system that allows developers to track changes in their codebase, collaborate with others, and easily manage different versions of their projects.

Why Git is essential in modern software development?

Git has become an essential tool in modern software development due to its numerous benefits and features:

- **Distributed Version Control:** Git allows developers to have a local copy of the entire codebase, enabling them to work offline and collaborate seamlessly with others.
- **Branching and Merging:** Git provides powerful branching and merging capabilities, allowing developers to create separate branches for different features or experiments and easily merge them back into the main codebase.

Working with Branches

In Git, branches are used to isolate work and make changes without affecting the main codebase. They allow multiple developers to work on different features or bug fixes simultaneously. Branches are lightweight and can be easily created, switched, and deleted. Here are some best practices for branch management:

Creating a Branch

To create a new branch, use the `git branch` command followed by the branch name. For example, `git branch feature-branch` creates a new branch named 'feature-branch'.

Switching Branches

To switch to a different branch, use the `git checkout` command followed by the branch name. For example, `git checkout feature-branch` switches to the 'feature-branch' branch.

Deleting a Branch

To delete a branch, use the `git branch -d` command followed by the branch name. For example, `git branch -d feature-branch` deletes the 'feature-branch' branch. Make sure to merge or push any changes before deleting a branch.

Best Practices

- Create a new branch for each new feature or bug fix.
- Use descriptive branch names to easily identify the purpose of each branch.
- Regularly merge changes from the main branch into your feature branches to keep them up to date.
- Delete branches after they have served their purpose to keep the repository clean.

Git Basics

Installation and Setup

To get started with Git, you first need to install it on your computer. Visit the official Git website and download the appropriate version for your operating system. Once installed, configure your Git username and email address using the following commands:

```
$ git config --global user.name "Your Name"
$ git config --global user.email "youremail@example.com"
```

Initializing a New Git Repository

To start using Git for version control, you need to initialize a new Git repository in your project directory. Navigate to your project directory in the command line and run the following command:

```
$ git init
```

Basic Git Commands

Once you have initialized a Git repository, you can use the following basic Git commands to manage your project:

- `git add`: Add files or changes to the staging area before committing them.
- `git commit`: Commit the changes in the staging area to the Git repository with a descriptive message.
- `git push`: Push your local commits to a remote repository, such as GitHub, to make them accessible to others.
- `git pull`: Fetch and merge the latest changes from a remote repository to your local repository.
- `git clone`: Create a local copy of a remote repository, allowing you to work on the project locally.

Collaborating with Git

Overview of Remote Repositories

Remote repositories are versions of your project that are hosted on a server and can be accessed and collaborated on by multiple people.

Connecting to Remote Repositories

To connect to a remote repository, you can use the 'git remote' command to add a remote name and URL. Then, you can use 'git push' to upload your local changes to the remote repository.

Collaborating with Others Using Git

Git allows multiple people to work on the same project simultaneously. You can use 'git pull' to fetch and merge changes from a remote repository into your local repository and 'git fetch' to retrieve changes without merging.

Resolving Conflicts

When multiple people make conflicting changes to the same file, Git will flag a conflict. You can resolve conflicts by manually editing the conflicting file and then using 'git add' and 'git commit' to save the changes.

Thank You for Joining

Thank you for joining the Workshop on Mastering Git, GitHub, and Version Control. I hope you found the session informative and helpful in your coding journey.

- A workshop by Rohit Kushwaha (CSI RKGIT)
- Github: MrDracs