**Day 9 Task**: **Deep Dive in Git & GitHub for DevOps Engineers**

**What is Git and why is it important:**

Git is a distributed version control system (DVCS) also referred to as Software configuration management(scm). It is widely used for tracking changes in source code during software development. Git is designed to be fast, efficient, and flexible.

Git is essential for modern software development because it enables effective collaboration, provides a robust version control system, and supports the agile and iterative nature of software development. It has become a standard tool in the industry and is widely adopted by individual developers and large development teams alike.

**What is difference Between Main Branch and Master Branch?**

Git repositories typically had a default branch named "master." This branch often represented the primary or mainline development branch.

Some platforms and projects started adopting the term "main" instead of "master" for the default branch.

GitHub, one of the most popular platforms for hosting Git repositories, officially made the switch from "master" to "main" as the default branch name for new repositories

The main and master branches serve the same purpose in terms of representing the primary branch of development.

**Can you explain the difference between Git and GitHub?**

**Git:**

Git is a distributed version control system (VCS) that allows developers to track changes in their codebase over time.

It enables multiple developers to work on the same project simultaneously without conflicts, as each developer has their own local copy of the repository.

Git provides features like branching and merging, allowing developers to experiment with new features or bug fixes without affecting the main codebase until they are ready to merge their changes.

**GitHub:**

GitHub, on the other hand, is a web-based platform that provides hosting for Git repositories. It adds a collaborative layer on top of Git.

GitHub allows developers to store their Git repositories in the cloud, making it easy for teams to collaborate on projects from different locations.

It provides a web interface for managing repositories, tracking issues, reviewing code changes, and facilitating collaboration among team members.

GitHub also offers additional features like pull requests, which are a way to propose and discuss changes before merging them into the main codebase.

**What is difference between local & remote repository? How to connect local to remote?**

**Local Repository:**

A local repository is a copy of a Git repository that is stored on your local machine. It contains all the files, branches, and commit history of the project.

When you create a new Git repository on your machine or clone an existing one, you are essentially creating a local repository.

You can work on and make changes to your local repository without needing an internet connection.

**Remote Repository:**

A remote repository is a version of your Git repository that is hosted on a server, often on a platform like GitHub, GitLab, Bitbucket, or another hosting service.

The remote repository serves as a central hub where multiple developers can collaborate on a project.

Changes made in the local repository can be pushed to the remote repository, and changes made in the remote repository can be pulled into the local repository.

**1)Connecting Local to Remote Repository:**

To connect a local repository to a remote repository, you typically follow these steps:

**Create a Remote Repository named “DevOps”:**

On platforms like GitHub create a new repository.

This step involves setting up a repository on the hosting service where you want to store your project remotely.

**Get the Remote Repository URL:**

On the remote repository page, find the URL that points to the repository. It usually looks like <https://github.com/username/repository.git>.

**Navigate to the Local Repository:**

Open a terminal or command prompt and navigate to your local repository.

**Link the Local and Remote Repositories:**

Use the following Git command to add a remote repository:

*git remote add origin <remote-repository-url>*

**Push the Local Repository to the Remote Repository:**

Use the following Git command to push your local repository to the remote repository:

*git push -u origin master*

This command assumes you are pushing the "master" branch. If you are on a different branch, replace "master" with your branch name.

**DevOps Workflow with Git and GitHub:**

**Development:**

Developers clone the repository, create branches, and make changes locally.

**Collaboration:**

Developers collaborate through pull requests, code reviews, and issue tracking.

**CI/CD:**

GitHub Actions or other CI tools automatically run tests and build processes on code changes.

**Deployment:**

Automated deployment pipelines deploy code to different environments.

**Monitoring:**

Monitor the deployed application and collect feedback for further improvements.

Using Git and GitHub in a DevOps workflow helps streamline development, improve collaboration, and automate processes, leading to faster and more reliable software delivery.

Tasks

task-1:

Set your user name and email address, which will be associated with your commits.

***ubuntu@DevOps****:****~/day8-sample-task****$ git config --global user.name "ds-kamran"*

***ubuntu@DevOps****:****~/day8-sample-task****$ git config --global user.email "ds.kamranarif@gmail.com"*

***ubuntu@DevOps****:****~/day8-sample-task****$ git config --list*

*user.name=ds-kamran*

*user.email=ds.kamranarif@gmail.com*

*core.repositoryformatversion=0*

*core.filemode=true*

*core.bare=false*

*core.logallrefupdates=true*

*remote.origin.url=https://ghp\_0F2dWhF-------wbSF3@github.com/ds-kamran/day8-sample-task.git*

*remote.origin.fetch=+refs/heads/\*:refs/remotes/origin/\**

*remote.upstream.url=https://github.com/ds-kamran/day8-sample-task.git*

*remote.upstream.fetch=+refs/heads/\*:refs/remotes/upstream/\**

task-2:

Create a repository named "Devops" on GitHub

Connect your local repository to the repository on GitHub.

Create a new file in Devops/Git/Day-02.txt & add some content to it

Push your local commits to the repository on GitHub