DevOps Certification Training

Study Material – Git & GitHub

Version Control System :

Version control systems are a category of software tools that helps in recording changes made to files by keeping a track of modifications done in the code.

Types of Version Control System :

There are two types of version control: centralized and distributed.

Centralized version control

With centralized version control systems, you have a single “central” copy of your project on a server and commit your changes to this central copy. You pull the files that you need, but you never have a full copy of your project locally. Some of the most common version control systems are centralized, including Subversion (SVN) and Perforce.

Distributed version control

With distributed version control systems (DVCS), you don't rely on a central server to store all the versions of a project’s files. Instead, you clone a copy of a repository locally so that you have the full history of the project. Two common distributed version control systems are Git and Mercurial.

What is Git ?

Git is **a DevOps tool used for source code management**. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development.

What is GitHub ?

GitHub is a for-profit company that offers a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration.

GitHub's interface is user-friendly enough so even novice coders can take advantage of Git. Without GitHub, using Git generally requires a bit more technical savvy and use of the command line.

GitHub is so user-friendly, though, that some people even use GitHub to manage other types of projects

What is Git repository?

A Git repository **tracks and saves the history of all changes made to the files in a Git project**. It saves this data in a directory called . git , also known as the repository folder. Git uses a version control system to track all changes made to the project and save them in the repository.

Features of Git

- Tracks history.
- Free and open source.
- Supports non-linear development.
- Creates backups.
- Scalable.
- Supports collaboration.
- Branching is easier.
- Distributed development.

Git Branch :

In Git, branches are a part of your everyday development process. **Git branches are effectively a pointer to a snapshot of your changes**. When you want to add a new feature or fix a bug—no matter how big or how small—you spawn a new branch to encapsulate your changes.

Git Merge :

Git merging **combines sequences of commits into one unified history of commits**. There are two main ways Git will merge: Fast Forward and Three way. Git can automatically merge commits unless there are changes that conflict in both commit sequences.

Merge Conflicts :

Merge conflicts **occur when competing changes are made to the same line of a file, or when one person edits a file and another person deletes the same file**. For more information, see "About merge conflicts."

Git Revert :

The git revert command is **a forward-moving undo operation that offers a safe method of undoing changes**. Instead of deleting or orphaning commits in the commit history, a revert will create a new commit that inverses the changes specified. Git revert is a safer alternative to git reset in regards to losing work.

Git Stash :

The git stash command **takes your uncommitted changes (both staged and unstaged), saves them away for later use, and then reverts them from your working copy**.

Git Rebase :

Rebasing is **the process of moving or combining a sequence of commits to a new base commit**. Rebasing is most useful and easily visualized in the context of a feature branching workflow.

Git Reset :

git reset is **a powerful command that is used to undo local changes to the state of a Git repo**. Git reset operates on "The Three Trees of Git". These trees are the Commit History (HEAD), the Staging Index, and the Working Directory. There are three command line options that correspond to the three trees.

Git reset operates on "The Three Trees of Git". These trees are the Commit History (HEAD), the Staging Index, and the Working Directory. There are three command line options that correspond to the three trees.