Name: Sreehas Dommata

GitHub: https://github.com/dsreehas/devops-training

Git branching strategy

A Git branching strategy is a set of guidelines and best practices for how to organize and manage branches in a Git repository.

Branches are used in Git to isolate changes and allow multiple developers to work on the same codebase without interfering with each other's work.

Many different git branching strategies can be used based on the project requirements. Somes of them are GitFlow, trunk-based development, feature branching, etc. GitFlow is one of a popular git branching strategy that is widely used in software engineering.

GitFlow branching structure

GitFlow consists of two types of branches, main branches and supporting branches.

Branch type	Branches	Naming convention
Main branches	master branch, develop branch	master, develop
Supporting branches	feature branches, release branches, hotfix branches	feature-*, release-*, hotfix-*

master branch: The master branch reflects a production ready state.

develop branch: It is parallel to the master branch. It is also known as the integration branch, where automatic night builds are built from. When the source code in the develop branch is bug free, it is then merged to the production ready master branch.

feature branches: Also known as topic branches, these branches are branches off the develop branch, and must be merged back to the develop branch. These branches are

located in developer repositories, not in origin repositories. These branches are used to add new features or any major changes.

release branches: These branches are used to support any new production releases, mainly for last minute minor bug fixes and preparing the meta-data for the release such as version numbers, build dates, build numbers, etc. These branch off the develop branch and must be merged back to the develop and master branches.

hotfix branches: These branches branch off from the develop branch and must be merged back to develop and master branches. These branches are used to fix critical bugs in the production. These allow to simultaneously fix the bugs in the code where other team members can work on the develop branch

GitFlow flow chart

GitFlow Branching Strategy