GIT

1. What is GIT and its significance in SDLC

Git is a version control system for tracking changes in files and coordinating work on those files among multiple people. It is primarily used for source code management in software development. It is a distributed revision control system and is very useful to support software development workflows. Git is the most commonly used version control system. Git tracks the changes you make to files, so you have a record of what has been done, and you can revert to specific versions should you ever need to. Git also makes collaboration easier, allowing changes by multiple people to all be merged into one source.

1. What is the difference between GIT and SVN?

While Git and SVN are both enterprise version control systems (VCS) that help with workflow and project management in coding, they do have their differences. The difference between Git and SVN version control systems is that Git is a distributed version control system, whereas SVN is a centralized version control system. Git uses multiple repositories including a centralized repository and server, as well as some local repositories. SVN does not have a centralized repository or server.

1. What are the advantages of using GIT?

One of the biggest advantages of Git is its branching capabilities. Unlike centralized version control systems, Git branches are cheap and easy to merge. This facilitates the feature branch workflow popular with many Git users. Feature branches provide an isolated environment for every change to your codebase.

1. What is “Staging Area” or “Index” in GIT?

This intermediate area, the staging area or index or cache, contains the proposed next commit. You start out by checking out some commit. At this point, you have three copies of every file: One is in the current commit (which Git can always find by the name HEAD ).

1. What is GIT stash?

Git stash helps you to temporarily save the changes in the working directory so that you will get a clean directory. Then you can apply those changes later in your git workflow. After stashing, you can see that the changes in the tracked files are gone in the working directory.

1. What is the function of git clone?

Git clone is primarily used to point to an existing repo and make a clone or copy of that repo at in a new directory, at another location. The original repository can be located on the local filesystem or on remote machine accessible supported protocols. The git clone command copies an existing Git repository.

1. How can you create a repository in Git?

Go to GitHub and login with your account credentials. Switch to the organization context by clicking on the Switch dashboard context drop-down, and click the organization. Click New next to Repositories. Alternatively, click the + icon on the top-right corner and click on New Repository.

1. What is the purpose of branching in GIT?

Branching means you diverge from the main line of development and continue to do work without messing with that main line. In many VCS tools, this is a somewhat expensive process, often requiring you to create a new copy of your source code directory, which can take a long time for large projects.

1. What is the difference between ‘git remote’ and ‘git clone’?

They are two completely different things. git remote is used to refer to a remote repository or your central repository. git clone is used to copy or clone a different repository.

1. What is the function of ‘git diff ’ in git?

Diffing is a function that takes two input data sets and outputs the changes between them. git diff is a multi-use Git command that when executed runs a diff function on Git data sources. These data sources can be commits, branches, files and more.

1. Explain what the commit message is?

A commit message is attached to that change not the code itself. Accordingly, when you write a commit message you are writing it as if it's about to be applied, rather than about what you just did.

1. Why is it advisable to create an additional commit rather than amending an existing commit?

The amend operation will destroy the state that was previously saved in a commit. If it's just the commit message being changed then that's not an issue.

1. What is Rebasing

The git rebase command is used to alter where a sequence of commits is based. This command requires at least the name of the other branch onto which your commits will be relocated.

MAVEN

1. Explain what is Maven? How does it work?

Maven is a popular open source build tool for enterprise Java projects, designed to take much of the hard work out of the build process. Maven uses a declarative approach, where the project structure and contents are described, rather then the task-based approach used in Ant or in traditional make files, for example.

1. Explain what is POM and its significance

A Project Object Model or POM is the fundamental unit of work in Maven. It is an XML file that contains information about the project and configuration details used by Maven to build the project. It contains default values for most projects. When you should execute a maven command you give maven a POM file to execute the commands.

1. Explain what a Maven artifact is?

Artifact: An artifact is something that is either produced or used by a project. Examples of artifacts produced by Maven for a project include: JARs, source and binary distributions, WARs.

1. List out the dependency scope in Maven?

Maven dependency scope attribute is used to specify the visibility of a dependency, relative to the different lifecycle phases (build, test, runtime etc). Maven provides six scopes i.e. compile , provided , runtime , test , system , and import .

1. List out what are the build phases in Maven?

Here are some of the most important phases in the default build lifecycle:

* validate: check if all information necessary for the build is available
* compile: compile the source code
* test-compile: compile the test source code
* test: run unit tests
* package: package compiled source code into the distributable format (jar, war, …)
* integration-test: process and deploy the package if needed to run integration tests
* install: install the package to a local repository
* deploy: copy the package to the remote repository

1. Mention the three build lifecycle of Maven

Here are some of the most important phases in the default build lifecycle:

* validate: check if all information necessary for the build is available
* compile: compile the source code
* test-compile: compile the test source code
* test: run unit tests
* package: package compiled source code into the distributable format (jar, war, …)
* integration-test: process and deploy the package if needed to run integration tests
* install: install the package to a local repository
* deploy: copy the package to the remote repository
* List out what are the aspects does Maven Manages?

1. List out the what are the aspects that Maven manages ?

* Builds.
* Documentation.
* Reporting.
* SCMs.
* Releases.
* Distribution.

1. Explain what a Maven Repository is? What are their types?

A repository in Maven holds build artifacts and dependencies of varying types. There are exactly two types of repositories: local and remote: the local repository is a directory on the computer where Maven runs. It caches remote downloads and contains temporary build artifacts that you have not yet released.

There are 3 types of maven repository:

* Local Repository.
* Central Repository.
* Remote Repository.

1. Explain how you can exclude dependency?

Multiple transitive dependencies can be excluded by using the <exclusion> tag for each of the dependency you want to exclude and placing all these exclusion tags inside the <exclusions> tag in pom. xml. You will need to mention the group id and artifact id of the dependency you wish to exclude in the exclusion tag.

1. For POM what are the minimum required elements?

The minimum required elements for POM are project root, modelVersion, groupID, artifactID and version.

1. What are the fundamental differences between DevOps and agile ?

Agile focuses on making developers and development cycles more efficient, while DevOps brings the operations team into play to enable continuous integration and continuous deliver.

1. What are the advantages of DevOps ?

Thus overall, DevOps promotes better efficiency, higher quality, and faster & continuous releases. Innovative mindset: DevOps streamlines processes, propagates efficient releases, and ensures quality builds.

1. Explain with a use case where DevOps can be used in industry/ real-life.

Application of DevOps in the Online Financial Trading Company. The methodology in the process of testing, building, and development was automated in the financial trading company. Using the DevOps, deployment was being done within 45 seconds. These deployments used to take long nights and weekends for the employees.

1. What are the success factors for Continuous Integration?

* Compilation.
* Unit Tests.
* Code Quality Gates.
* Integration Tests.
* Deployment.
* Chain Tests.

1. What are the differences between continuous integration, continuous delivery, and continuous deployment?

Most developers [start with Continuous Integration (CI)](https://semaphoreci.com/continuous-integration), which is about everyone merging code changes to a central repository multiple times a day. Each merge triggers an automated build and testing sequence for the given project.

[Continuous Delivery (CD)](https://semaphoreci.com/cicd) is a practice of automating the entire software release process. The idea is to do CI, **plus** automatically prepare and track a release to production. The desired outcome is that anyone with sufficient privileges to deploy a new release can do so at any time in one or a few clicks.

[Continuous Deployment](https://semaphoreci.com/community/tutorials/elements-of-a-continuous-deployment-workflow) is a step up from Continuous Delivery in which every change in the source code is deployed to production automatically, without explicit approval from a developer. A developer’s job typically ends at reviewing a pull request from a teammate and merging it to the master branch. A CI/CD service takes over from there by running all tests and deploying the code to production, while keeping the team informed about outcome of every important event.

1. What role does the Quality Assurance (QA) team play in DevOps?

QA ties together development and operations and enables them to collaborate to have software and applications up & running. Everyone in the organization takes responsibility for quality and stability, and thereby for the business success.

1. Describe an efficient workflow for continuous integration

**1. Version management**

**2. Decide a development strategy**

**3. Decide your delivery strategy**

**4. Decide an application version strategy**

**5. Build automation tool**

**6. Create and implement unit tests**

**7. Automatically build a test environment**

**8. Automatically test your most precious software parts functionally**

**9. Optionally create an acceptance environment**

**10. Optionally create an installation package**

**11. Create some error reporting**

**12. Backup your Build Automation**

1. What are the best practices for DevOps implementation?

Best Practices for Successful DevOps Implementation

* Active Collaboration of Stakeholders. ...
* Automation of Tests and Building Environment. ...
* Integrated Configuration and Change Management. ...
* Continuous Integration and Continuous Deployment. ...
* Continuous Delivery and Product Support. ...
* Application Monitoring and Automation of Dashboards.

1. How will you approach when a project needs to implement DevOps?

Best Practices for Successful DevOps Implementation

* Active Collaboration of Stakeholders.
* Automation of Tests and Building Environment.
* Integrated Configuration and Change Management.
* Continuous Integration and Continuous Deployment.
* Continuous Delivery and Product Support.
* Application Monitoring and Automation of Dashboards.