GIT

1. What is GIT and its significance in SDLC

Git is a distributed version control system and source code management (SCM) system with focus to handle small and large projects source code versions in the local repository with speed and efficiency. It is free and open source and its one of widly used versioning tools used ever worldwide

1. What is the difference between GIT and SVN

Git is an open-source distributed vice control system Git has a Distributed Model. In git every user has their own copy of code on their local like their own branch Git deals with large number of files like binary files that change quickly that why it become slow

SVN Apache Subversion is an open source software version and revision control system under Apache license SVN has a Centralized Model.

In SVN there is central repository has working copy that also make changes and committed in central repository In SVN we required Network for runs the SVN operation

1. What are the advantages of using GIT

**1. Performance**— Git is a strong and reliable platform in comparison with other versions of the control system. It allows changes in new codes can be easily committed,

**Security**— The main purpose of Git is to manage the integrity of source code. It ensures the relationship between a file and commits, directories, tags, version, etc

**Flexibility**— The key design goal of Git is to acquire the kind of flexibility which offers support of different types of non-linear workflows development and its efficiency in handling small and large both projects

**Wide Acceptance**— Git renders performance types, security, flexibility, and functionality as most of developing teams require these aspects to develop their projects

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

[index](https://hackernoon.com/tagged/index) that it uses to keep track of the file changes over the three areas: working directory, staging area, and repository. And when you add changes to your staging area, git updates the information in the index about those changes and creates new blob objects, but puts them in the same .git/objects directory with all the other blobs that belong to previous commits.

1. What is GIT stash

**Git stash** is used in order to **save all the changes done to the current working directory** and to go back to the last commit done on the branch

1. What is the function of git clone

 cloning is the act of making a copy of any target repository. The target repository can be remote or local. You can clone your repository from the remote repository to create a local copy on your system. Also, you can sync between the two locations.

1. How can you create a repository in Git

Git repository is the virtual storage of our projects where we can save versions of our code and can access it when needed. To create a new repository, we use “git init” command. It will creates a new .git subdirectory in our current working directory. It will also creates a new master branch

1. What is the purpose of branching in GIT

If you commit code to the master branch, it means you change the website directly. If you introduce any bugs, other people can see your bug immediately. So we create a new branch and work off it. When we’re done, and when we’re sure that there are no more bugs at least we try to make sure .We push the changes back to the master branch to update the website. When you create a new branch, you can code directly on the branch itself. Any code you change will be reflected only on that branch

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

Both are completely different commands. git remote is used to refer to a remote repository/central repository but git clone is used to clone an existing repository to another directory

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

Git diff command is used to track the difference between the changes made on a file. It takes two input data sets and outputs the changes between them

1. Explain what the commit message is

A commit in GitHub is described as a saved change. A commit message explains what change did you perform to your project. We can consider commits as a snapshot of our entire repository

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.  But if the contents are being amended then chances of eliminating something important remains more.

Abusing “git commit- amend” can cause a small commit to grow and acquire unrelated changes.

1. What is Rebasing

git rebase allows one to move the first commit of a branch to a new starting location

git rebase can be used to move the feature branch to the tip of master. The command effectively will replay the changes made in the feature branch at the tip of master, allowing conflicts to be resolved in the process

Maven Fundamentals

1. Explain what is Maven? How does it work

Maven is a project management tool. It provides the developer a complete build lifecycle framework. On executing Maven commands, it will look for POM file in Maven; it will run the command on the resources described in the POM

1. Explain what is POM and its significance

In Maven, POM (Project Object Model) is the fundamental unit of work. It is an XML file which holds the information about the project and configuration details used to build a project by Maven While executing a task, maven look for the POM it reads POM and takes the needed configuration information and then runs the task

1. Explain what a Maven artifact is

artifact is a JAR file which gets arrayed to a Maven repository. One or more artifacts a maven build produces such as compiled JAR and a sources JAR.

Each artifact includes a group ID, an artifact ID and a version string

1. List out the dependency scope in Maven

Compile: It is the default scope, and it indicates what dependency is available in the classpath of the project

Provided: It indicates that the dependency is provided by JDK or web server or container at runtime

Runtime: This tells that the dependency is not needed for compilation but is required during execution

Test: It says dependency is available only for the test compilation and execution phases

System: It indicates you have to provide the system path

Import: This indicates that the identified or specified POM should be replaced with the dependencies in that POM’s section

1. List out what are the build phases in Maven

Validate, Compile, Test, Package, Install, Deploy

1. Mention the three build lifecycle of Maven?

Clean Lifecycle, Default Life cycle, Site Lifecycle

1. List out what are the aspects does Maven Manages?

Build, Documentation, Reporting, Dependencies, SCMs,

Releases, Distribution, Mailing list

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

a directory where all the project jars, library jar, plugins or any other project specific artifacts are stored and this can be used by Maven easily.

Local: Maven local repository is a folder location that is present on your machine. It is created when you run any maven command for the first time. Maven local repository is a location where you can find your project’s all dependencies.

Central: It is a repository provided by the Maven community. It contains a huge collection of commonly used libraries. When Maven does not find any dependency in local repository, it starts searching in central repository

Remote: Sometimes, Maven is not able to find a mentioned dependency in the central repository as well then it stops the build process and an output error message is displayed on the console. To avoid such a situation, Maven provides the idea of Remote Repository which is nothing but the developer’s own custom repository containing required libraries or other project jars.

1. Explain how you can exclude dependency

Exclusions are set on a specific dependency in your POM, and are targeted at a specific groupId and artifactId. We can exclude adependency by adding <exclusions> element inside the <dependency> element and provide the groupId and artifactId inside their tags.

1. For POM what are the minimum required elements

Project root.

modelVersion - should be set to 4.0. ...

groupId - the id of the project's group.

artifactId - the id of the artifact (project)

version - the version of the artifact under the specified group

CI/CD

1. What are the fundamental differences between DevOps & Agile

Agile software development methodology focuses on the development of software.

Agile method can be implemented within a range of frameworks like a sprint, safe and scrum. Feedback is given by the customer. It focuses on functional and non-function readiness

DevOps on the other hand is responsible for development as well as deployment of the software in the safest and most reliable way possible. The primary goal of DevOps is to focus on collaboration, so it doesn't have any commonly accepted framework

Feedback comes from the internal team It focuses more on operational and business readiness

1. What is the need for DevOps

DevOps is important because it's a software development and operations approach that enables faster development of new products and easier maintenance of existing deployments

1. What are the advantages of DevOps

Continuous software delivery , Less complex problems to fix , Faster resolution of problems , Faster delivery of features , More stable operating environments , More time available to add value

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

E-commerce website focused on handmade or vintage items and supplies, as well as unique factory-manufactured items. Struggled with slow, painful site updates that frequently caused the site to go down. It affected sales for millions of users who sold goods through online market place and risked driving them to the competitor.  
With the help of a new technical management team, transitioned from its waterfall model, which produced four-hour full-site deployments twice weekly, to a more agile approach. Today, it has a fully automated deployment pipeline

1. What are the success factors for Continuous Integration

Continuous Integration can be defined as Building software and taking it through as many tests as possible with every change. The main factors for continuous integration are:

Compilation.

Unit Tests.

Code Quality Gates.

Integration Tests.

Deployment.

Chain Tests.

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

Continuous integration is a [DevOps](https://aws.amazon.com/devops/) software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run

Continuous delivery  is a software development practice where code changes are automatically prepared for a release to production.With continuous delivery, every code change is built, tested, and then pushed to a non-production testing or staging environment. There can be multiple, parallel test stages before a production deployment.

Continuous deployment is the presence of a manual approval to update to production. With continuous deployment, production happens automatically without explicit approval.

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

QA plays a strategic role in ensuring that quality is taken up as a responsibility by both Development and Operations. In DevOps, QA detects bugs in the project

1. Describe an efficient workflow for continuous integration

continuous integration, there is a need to have a repository where in the code could be saved, retrieved and maintained. The repository must be good enough to provide the developers with a powerful version controlling system.

Git is one of the version control systems (VCS). It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files. Let’s consider Centralized work flow provided by GI

1. What are the best practices for DevOps implementation

Encouraging communication and collaboration especially between development and operations. Automation is the key and should be done and carried out wherever possible within the SDLC stages to facilitate the working of DevOps.

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

Continuous Development - first phase of the lifecycle, the software application is thoroughly discussed to understand the essence and vision of it. Once the concept of the application is clear, the development of the source code starts.

## Continuous Integration

The source code has frequent changes, either due to the addition of features or changes in functionality. As a result, in the continuous integration phase, the changes made are reflected and updated in the existing code.

Continuous Testing

the continuous testing phase can be repositioned around the DevOps lifecycle. In this phase, the software application gets continually tested for bugs

Continuous Feedback

The improvements made in the testing and integration phase is analysed in the continuous feedback phase. The developers who analysed the upgrades and the customers who tested the application can input their valuable feedback in this phase

## Continuous Monitoring

Continuous monitoring helps in determining and solving recurring errors in an application.

Continuous Deployment

Generally, continuous deployment takes place before the phase of constant monitoring.

Continuous Operations

The final phase of the DevOps life cycle, although crucial, is least complicated and consumes the shortest amount of time. This phase, known as the continuous operation, is mainly aimed at automating the systematic release of an application, followed by its updates.