1-What is Jenkins pipeline?

Jenkins Pipeline is a feature of the Jenkins automation server that allows you to define and manage your software delivery processes as code. It provides a way to create and visualize continuous delivery pipelines, automating the build, test, and deployment of your applications. Jenkins Pipeline is written in Groovy and supports both scripted and declarative syntax. It offers benefits such as code reusability, version control integration, pipeline visualization, and the ability to treat pipelines as code.

2- What scripting language is Jenkins pipeline syntax based on?

Jenkins Pipeline syntax is based on the Groovy scripting language. Groovy is a dynamic, object-oriented programming language for the Java platform that offers a concise and flexible syntax. It is designed to enhance and simplify Java development with features such as closures, dynamic typing, and seamless Java integration.

3- what are the different ways to trigger pipeline ?

In Jenkins Pipeline, there are several ways to trigger the execution of a pipeline. Here are some common methods:

SCM Trigger: Jenkins can be configured to monitor a version control system (such as Git, Subversion, or Mercurial) for changes. When a change is detected, such as a new commit or a branch update, Jenkins can automatically trigger the pipeline to start a new build and execute the defined stages.

Timer Trigger: You can schedule a pipeline to run at specific time intervals using a timer trigger. This allows you to execute the pipeline on a regular basis, such as nightly builds or weekly deployments.

Webhook Trigger: Jenkins can be integrated with external services, such as issue trackers or code review tools, that provide webhook functionality. When an event occurs in the integrated service, a webhook is triggered, and Jenkins can respond by initiating the pipeline execution.

Manual Trigger: Pipelines can also be triggered manually by a user. This can be done through the Jenkins web interface by clicking on a button or by using Jenkins' API or CLI (Command Line Interface).

Pipeline Trigger from Another Job: You can configure a Jenkins job or pipeline to trigger another pipeline as a downstream job. This allows you to define complex workflows where one pipeline triggers the execution of another pipeline.

Pipeline Trigger from External Tools: Jenkins provides integration with various external tools and services through plugins. These plugins can be used to trigger pipeline execution based on events or notifications from those external tools. For example, you can trigger a pipeline

when a new artifact is uploaded to an artifact repository or when a test suite completes in a test management tool.

4-what is different between parameter and jenkins env variable?

Parameters in Jenkins pipelines are user-defined values that can be passed into a pipeline when it is triggered. They allow users to customize the behavior of the pipeline run. Parameters are defined in the Jenkinsfile or pipeline configuration and can have different types. Parameter values can be specified when triggering the pipeline.

Jenkins environment variables, on the other hand, are predefined variables that Jenkins automatically sets during the execution of a pipeline. They provide information about the Jenkins environment, the build, and other contextual details. These variables can be accessed directly within the pipeline script without any additional configuration.

In summary, parameters are user-defined values for customization, while Jenkins environment variables are predefined variables providing information about the Jenkins environment and build.

5- what is organization folder job and what is used for ?

An Organization Folder job in Jenkins is used to organize and manage multiple related projects or repositories within an organization. It automatically discovers repositories and creates child jobs for each one. It helps streamline management, apply automation and standardization, and provides a centralized view for monitoring and controlling projects.