**Pipeline job configuration :**

Pipeline job configuration we never implemented in my environment but as per my knowledge how to do pipeline job configuration is

first i have to create the job so that i have to click on new item and i will specify the job name like CI-PIPELINE-Branch and i will select there pipeline and i will give a click on ok button once i give a click on button then i will enter into the job pipeline configuration if i go down i will get two options like PIPELINE SCRIPT and PIPELINE SCRIPT FROM SCM so what i will do here i will choose pipeline script from scm and whatever the version control system your using that one you have to select and specify the repository url before that what exactly we have to do is first we have to write the jenkins file in that i need to write complete job configuration i have to write using Groovy script and that jenkins file i need to check in to the version control system after that i will specify that repository url inside the pipeline job configuration and i will save and i will trigger the build so whatever i have written inside the file according to that it will perform.

Suppose if i select pipeline script then we have to write complete script here only from scratch to deployment, ( we will write vcs as node in groovy script )

--> In My Environment we have pre-build scripts which we gonna use , it is for pipeline script

**>>WHAT IS DECLARATIVE PIPELINE VIEW AND SCRIPTED PIPELINE VIEW ?**

**Declarative pipeline:**

If you want declarative pipeline then you have to choose pipeline script from SCM.

Declarative pipeline is nothing but pipeline script from SCM.

**Scripted Pipeline:**

If you want scripted pipeline then you have to choose pipeline script.

Declarative pipeline is nothing but pipeline script from SCM.

---------------------------------------------------

**Difference between Scripted pipeline and declarative pipeline script?**

The script has the elements like **Pipeline**, **agents**, **steps** which are specific to declarative pipeline,

**Stage** is common to both declarative and scripted,

And finally, **Node** is specific for the scripted one.

**Agent** defines where the pipeline will be run.

**PIPELINE:**

A user defined block which contains all the stages, It is a key part of declarative pipeline syntax.

**NODE:**

A node is a machine that executes an entire workflow. It is a key part of the scripted pipeline syntax.

**WHAT IS AGENT?**

Agent is an executer which will run the entire pipeline.

------------------------------------------------------------------------------------------------------

**EXPLAIN THE STAGES IN PIPELINE SCRIPT ?**

Basically in my pipeline groovy script we have totally seven stages that is ,

clean up old directory

checkout the code

build the code

packaging the build artifactory and next stage is

code scaning and then

publish to artifactory ,

deploy to the staging environment

this are the seven stages first cleanup working directory basically before clean up i should mention the pipeline agent , agent is nothing but where exactly i need to build that means pipeline and then agent details then stages i need to mention under the stages i should mention the clean up old directory clean workspace paste the location where i have to clean after that checkout and then the repository where i need to clone and then the commit branch name i should pass echo cloning is successfully completed one message i will send after that next level is to packaging the code for that maven we integrated actually so here maven targets we written according to the targets it will compile the source code and it will create the packages while creating the packages we added the sonarqube stage also here sonarqube stages what and all details we have to pass sonar.source=( the location where exactly i want to check the code quality that location i have to give and also i should give the language i mean which laguage write the source code if it is java , sonar.language=java and then the sonar.projectname i should pass so this and all details i should pass inside the script) and also next level is uploading the packages into the artifactory basically we integrated the nexus artifactory to take the backup of the every version for that artifactory details artifactory repository name and artifactory URL i should pass from the script finally finally staging details under the staging details we should give the location where exactly my pre and post commit scripts we are running for each and every deployment that location i should pass from the stage deployment after that i will add the notification for the every pipeline script so that whenever my developer checked in the source code it will completely seven stages will execute if any failure will happen according to the stage i will do the changes in the bugs and then i can do again build process so this is the way how we are doing th CI/CD .

------------------------------------------------------------------------------------------------

1) What is meant by build automation?

A) Build automation is the process of automating the creation of a software build and the associated processes including: compiling computer source code into binary code, packaging binary code, and running automated tests.

2) What are the different build automation tools?

There are many Build tools available in the market they are: Maven, Hudson, Gradle, SBT, and Rake.

3) What is the difference between compile and install?

Compile compiles the source code of the project

Whereas

Install installs the package into the local repository, for use as a dependency in other projects locally

4) Difference between jar, war and ear?

Jar is Java Archieve i.e compressed Class or Class / Java files.

War comprises of compressed Servlet class files, JSP FIles, supporting files, GIF and HTML files.

Ear comprise of compressed Java and web module files (was files).

5)How Can I Change The Default Location Of The Generated Jar When I Command "mvn Package"?

Answer :

By default, the location of the generated jar is in ${project.build.directory} or in your target directory. We can change this by configuring the outputDirectory of maven-jar-plugin.

6) How Do I Determine Which Pom Contains Missing Transitive Dependency?

Answer :

run mvn -X

7)What Technologies Have You Worked With For Build Management ?

Answer :

Ant and Maven

8)What is the best practice configuration usage for files - pom.xml or settings.xml ?

Ans. The best practice guideline between settings.xml and pom.xml is that configurations in settings.xml must be specific to the current user and that pom.xml configurations are specific to the project.

9) What is Maven's order of inheritance?

Ans.

1. Parent pom

2. Project pom

3. Settings

4. CLI parameters

10) What is Archetype?

Archetype is a Maven plugin whose task is to create a project structure as per its template.