**What is build tools?**

**Build tools are programs that automate compiling, linking, and packaging the code into a usable or executable form from source code.**

**what is Maven?**

**Maven is a build automation tool used to build and manage projects written in C#, Ruby, Scala, and other languages.**

**What is difference between ant and Maven?**

**Maven has a standard project layout.**

**Maven is able to create reports based on the dependencies of the project.**

**Maven has a predefined structure of project build.**

**Dependencies need not be updated manually.**

**Maven has standard naming conventions.**

**Ant does not have a standard project layout.**

**Ant does not create reports on its own, and it can be made to create reports.**

**We need to define everything from the directory, target, project, etc., in Ant.**

**Dependencies need to be updated manually.**

**Ant does not have standard naming conventions.**

**Maven life cycle?**

**For Maven there is below phases.**

**clean - Cleans the project and removes all files generated by the previous build.**

**validate - validate the project is correct and all necessary information is available**

**compile - compile the source code of the project**

**test - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed**

**package - take the compiled code and package it in its distributable format, such as a JAR.**

**verify - run any checks on results of integration tests to ensure quality criteria are met**

**install - install the package into the local repository**

**deploy - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.**

**Maven project directory structure?**

**Maven defines a standard directory structure.**

**- src                       ------ directory of source code**

**- main                      ------ directory for source code related to the application**

**- java                   ------ Java code for the application**

**- resources             ------ resources needed for your project**

**- webapp                 ------ contains Java web application**

**- test                       ------ contains the test source code**

**- java                  ------ Java code for the tests**

**- resources             ------ resources needed for your tests**

**- target                    ------ contains all the compiled classes, JAR files etc**

**How to run the test case in maven?**

**mvn test   ----------- it will test your project.**

**What mvn clean install do ?**

**mvn clean install         or     mvn clean install -U -DskipTests=false**

**the above command compile, test & package your Java project and even install/copy your built .jar/.war file into your local Maven repository.**

**what mvn clean verify do?**

**mvn clean verify will rebuild and test your project.**

**how to build the package including the submodule in maven?**

**mvn -pl  submodule\_name -am clean install**

**How to build package with dependent latest snapshot?**

**mvn -U clean install**

**How to compile the tests but skip to run the test?**

**mvn clean install  -DskipTests=true**

**How to skip compile the tests code and run the tests in maven?**

**mvn clean install -Dmaven.test.skip=true**

**How to configure JDK , MAVEN, Gradle in jenkins?**

**Manage jenkins ---> Global tool configuration**

**How to create authentication token in jenkins?**

**Login to jenkins Console**

**In the top-right corner click on your Profile**

**click on My account**

**go to Security**

**click on Generate Tokens**

**Enter a friendly description and confidential information**

**what SonarQube?**

**SonarSource, is an open-source framework use for static code analysis to identify bugs, security vulnerabilities, and code bad smells.**

**What are the components in SonarQube architecture?**

**Sonar Analyzer**

**SonarQube Database**

**Source Code**

**Sonar Scanner**

**What is sonarqube quality gates?**

**Quality Gates are a group of threshold measures set on your project to Measure Number of Blocker / Critical Issues, Security Rating / Unit Test Pass Rate , major issues etc.**

**To configure quality gates     login to sonarqube -- quality gates -- create**

**How to configure  SonarQube in  Jenkins?**

**1 --> Manage Jenkins > Manage plugins and install the SonarQube Scanner plugin:**

**2 --> Manage Jenkins > Configure System > SonarQube servers sub-menu > provide Name , Server URL , authentication token**

**How to configure SonarQube Scanner in jenkins?**

**Manage Jenkins > Global Tool Configuration > SonarQube Scanner > provide name and version (sonarqube scanner 4.0)**

**How to configure Sonarqube quality gates in jenkins?**

**Manage jenkins > configure system > quality gates sonarqube > provide Name , Server URL , sonarqube account token**

**How to run sonarqube scanner for maven?**

**mvn clean package sonar:sonar**

**How to create reports in SonarQube for maven?**

**mvn sonar:sonar -Dsonar.issuesreport.html.enable=true**

**How to run jenkins maven project with sonarqube using jenkins GUI?**

**click your job > go to build tab > execute sonarqube scanner > fill Analysis properties**

**Analysis properties**

**------------------------**

**# Required metadata**

**sonar.projectKey=<project-key> ------------------- ID for your project in SonarQube**

**# Source information**

**sonar.sources=src/main**

**sonar.sourceEncoding=UTF-8**

**sonar.language=java**

**# Tests**

**sonar.tests=src/test**

**sonar.junit.reportsPath=target/surefire-reports**

**sonar.surefire.reportsPath=target/surefire-reports**

**sonar.jacoco.reportPath=target/jacoco.exec**

**sonar.java.binaries=target/classes**

**sonar.java.coveragePlugin=jacoco**

**How to run pipeline jenkins maven project with sonarqube?**

**Using declarative pipeline:**

**pipeline {**

**agent none**

**stages {**

**stage("build & SonarQube analysis") {**

**agent any**

**steps {**

**withSonarQubeEnv('My SonarQube Server') {**

**sh 'mvn clean package sonar:sonar'**

**}**

**}**

**}**

**stage("Quality Gate") {**

**steps {**

**timeout(time: 1, unit: 'HOURS') {**

**waitForQualityGate abortPipeline: true**

**}**

**}**

**}**

**}**

**}**

**Using scripted pipeline:**

**stage("build & SonarQube analysis") {**

**node {**

**withSonarQubeEnv('My SonarQube Server') {**

**sh 'mvn clean package sonar:sonar'**

**}**

**}**

**}**

**stage("Quality Gate"){**

**timeout(time: 1, unit: 'HOURS') {**

**def qg = waitForQualityGate()**

**if (qg.status != 'OK') {**

**error "Pipeline aborted due to quality gate failure: ${qg.status}"**

**}**

**}**

**}**

**how I can delete a project from SonarQube?**

**Login sonarqube as an administrator --- Administration -> Projects -> Projects Management -- select project --- delete**