QUESTION-1:

 Build a manual ci-cd pipeline for a java application and also write a simple test.sh for testing after that convert the same in automated ci-cd pipeline using jenkins but you have to also use maven for java application.

Building a manual CI/CD Pipeline:

Code your Java application: Start by coding your Java application and create a Maven project.

Create a Git repository: Create a Git repository to store your code.

Define your manual pipeline stages: Define your manual CI/CD pipeline stages, such as Build, Test, Deploy.

Write a build script: Write a script to compile and package your Java application using Maven. The script should also copy the packaged JAR file to a designated location for deployment.

Write a test script: Write a script to test your Java application. For example, you can use the JUnit testing framework.

Write a deployment script: Write a script to deploy your Java application to a server or container. For example, you can use a shell script or Dockerfile.

Automate your pipeline: Once you have a working manual pipeline, you can automate it using a CI/CD tool like Jenkins.

Automating the CI/CD Pipeline with Jenkins:

Install and configure Jenkins: Install Jenkins on your server and configure it according to your needs. You'll need to install the necessary plugins for Java and Maven.

Create a Jenkins project: Create a new Jenkins project and configure it to pull the code from your Git repository.

Define build triggers: Define build triggers for your Jenkins project. For example, you can trigger a build every time a new commit is pushed to the Git repository.

Configure your Jenkins build: Configure your Jenkins build by specifying the Maven command to build your Java application and package it into a JAR file.

Configure test steps: Configure your Jenkins build to run your test.sh script.

Configure deployment steps: Configure your Jenkins build to deploy your Java application using a shell script or Dockerfile.

Test your automated pipeline: Test your automated CI/CD pipeline by pushing a new commit to your Git repository and verifying that Jenkins automatically builds, tests, and deploys your Java application.

And that's it! You now have a manual CI/CD pipeline for your Java application and have automated it using Jenkins.