Jenkins Pipeline Script:

Ex :

```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                echo 'Building the code...'
                // In a real-world scenario, you might use a build tool like Maven, Gradle, or npm
here.
            }
        }

        stage('Test') {
            steps {
                echo 'Running tests...'
                // In a real-world scenario, you would execute your test suite here.
                // You might use tools like JUnit, Selenium, or any other testing framework.
            }
        }
    }

    post {
        success {
            echo 'Pipeline completed successfully! Deploying to production...'
            // In a real-world scenario, you might trigger deployment to a production
environment here.
        }
        failure {
            echo 'Pipeline failed. Notify the team and roll back changes if necessary.'
            // In a real-world scenario, you might notify the team and take appropriate actions in
case of failure.
        }
    }
}
```

Pipeline : This block defines the entire Jenkins pipeline. The “agent any” directive specifies that the pipeline can run on any available agent.

stages : Inside the “stages” block, we define individual stages of the pipeline. Each stage represents a phase in the software development process.

stage('Build') : This block represents the “Build” stage. The “steps” section contains the commands or tasks to be executed during this stage. In this example, we simply echo a message, but in a real-world scenario, you would perform actions like compiling code, creating artifacts, etc.

stage('Test') : This block represents the “Test” stage. Similar to the “Build” stage, you would include steps to run your automated tests in this section.

post : This block defines post-execution actions based on the success or failure of the pipeline. In this example, if the pipeline is successful (‘success’ block), it echoes a success message and mentions deploying to production. If the pipeline fails (‘failure’ block), it echoes a failure message and suggests notifying the team.