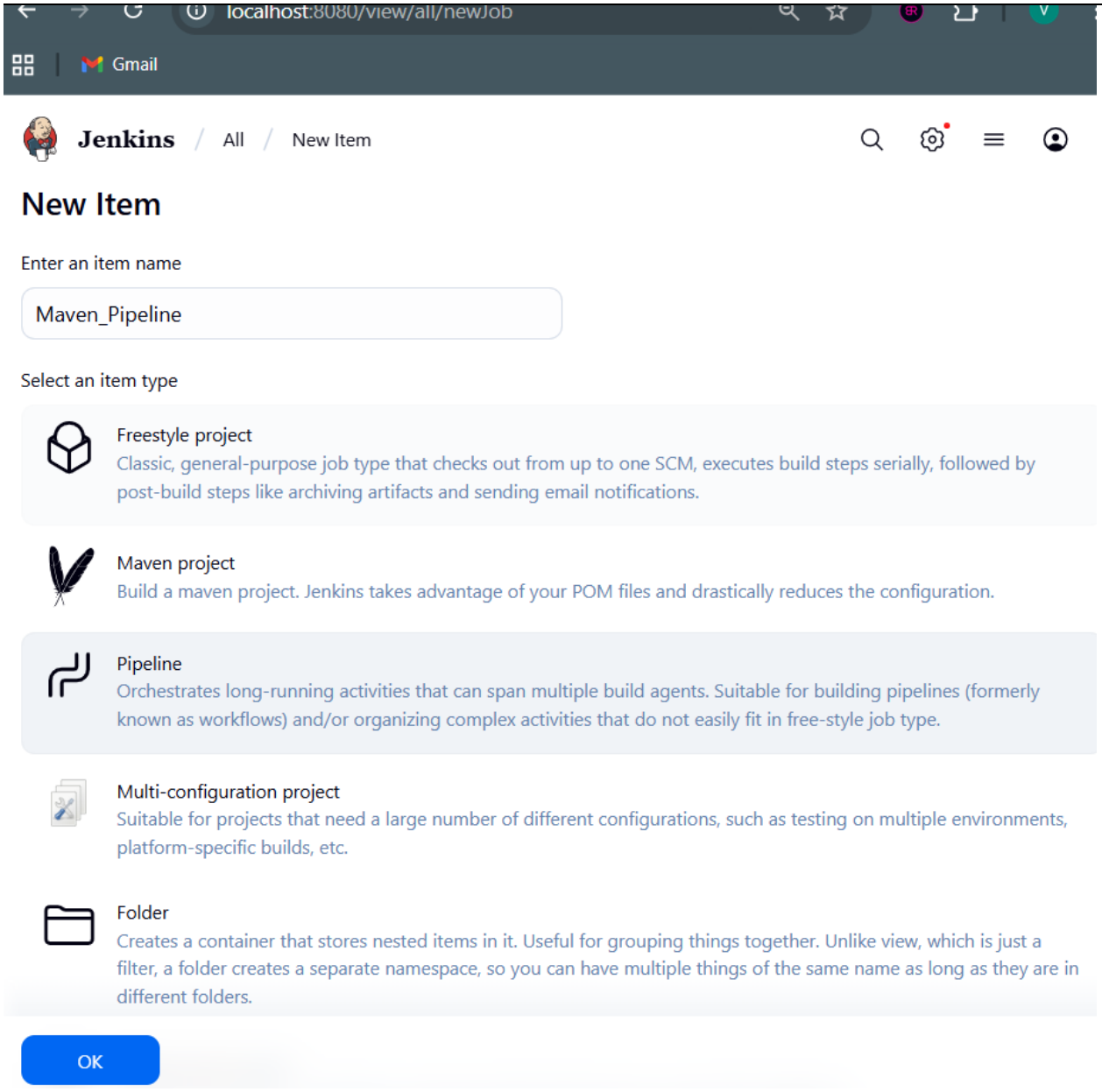
Week 9: Pipeline Creation using script

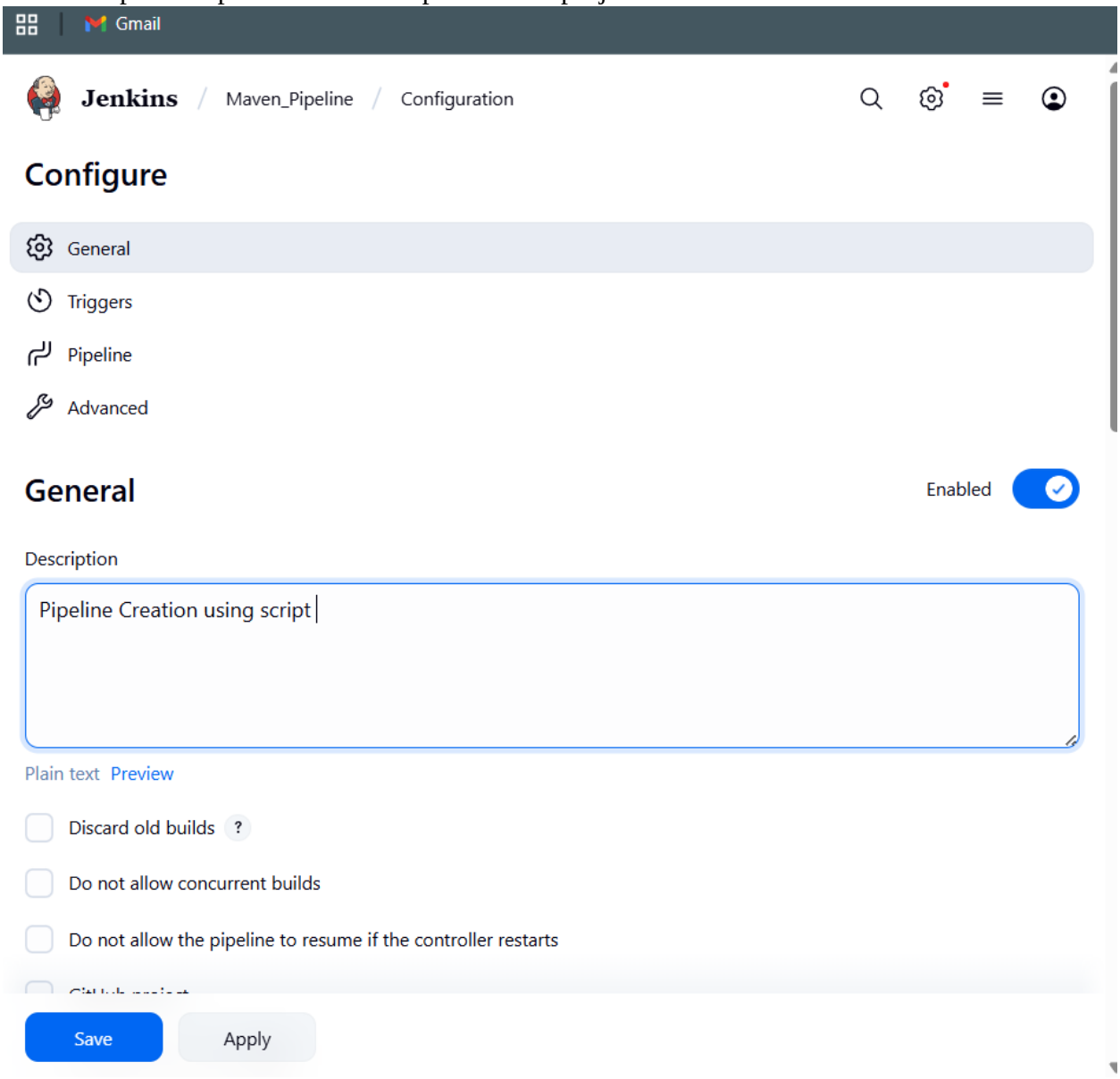
1. Evaluation of Jenkins pipeline.
2. **WORKING ON BUILD TRIGGERS FOR LAST JENKINS PIPILINE**
3. Hands-on practice on creation of scripted Jenkins pipeline.
4. Take the screenshots for above task

Procedure

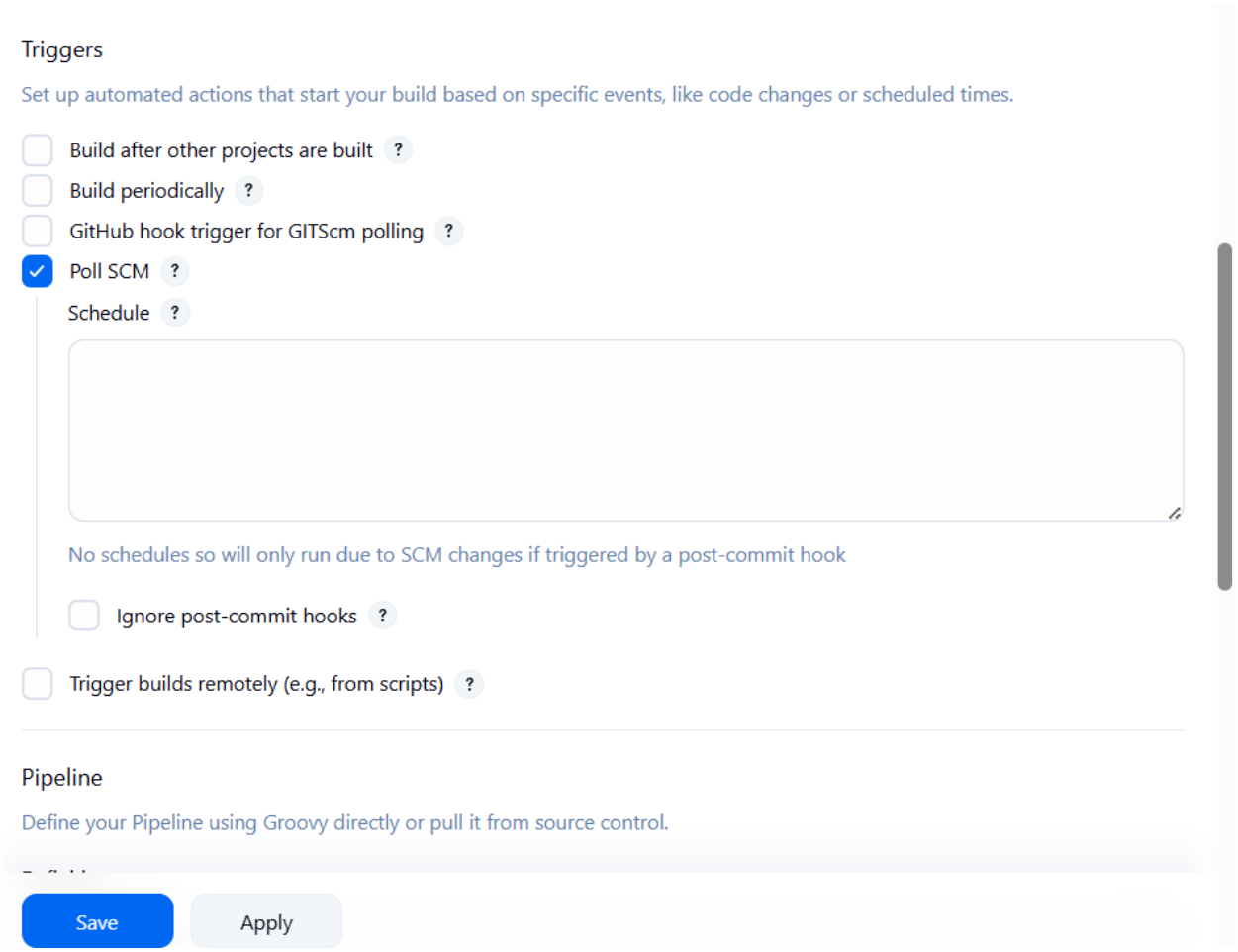


General :

Description :- provide the description of the project



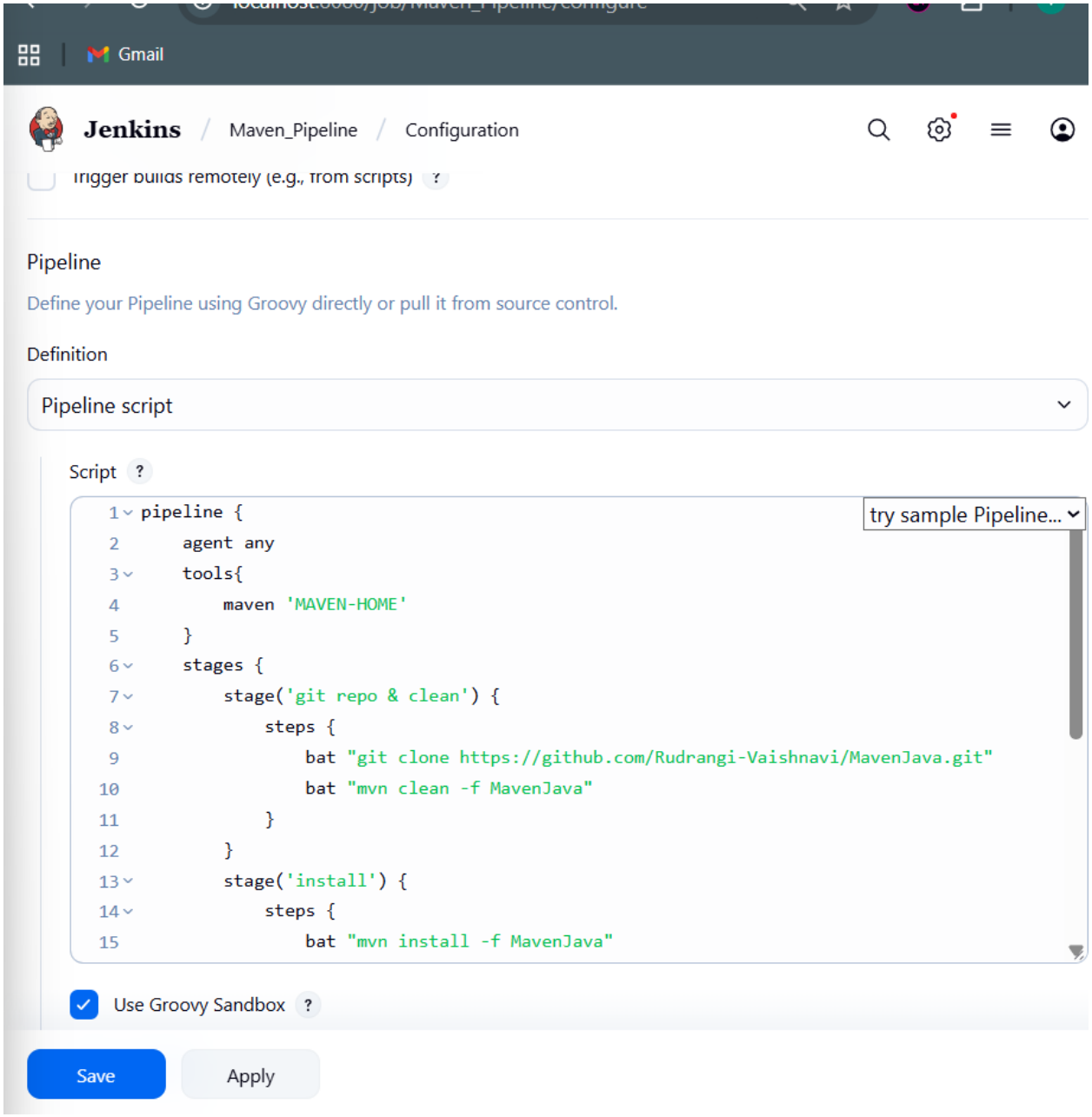
Build triggers: here we can provide with build triggers of you choice



Advance project option:

Definition:

Choose pipeline script



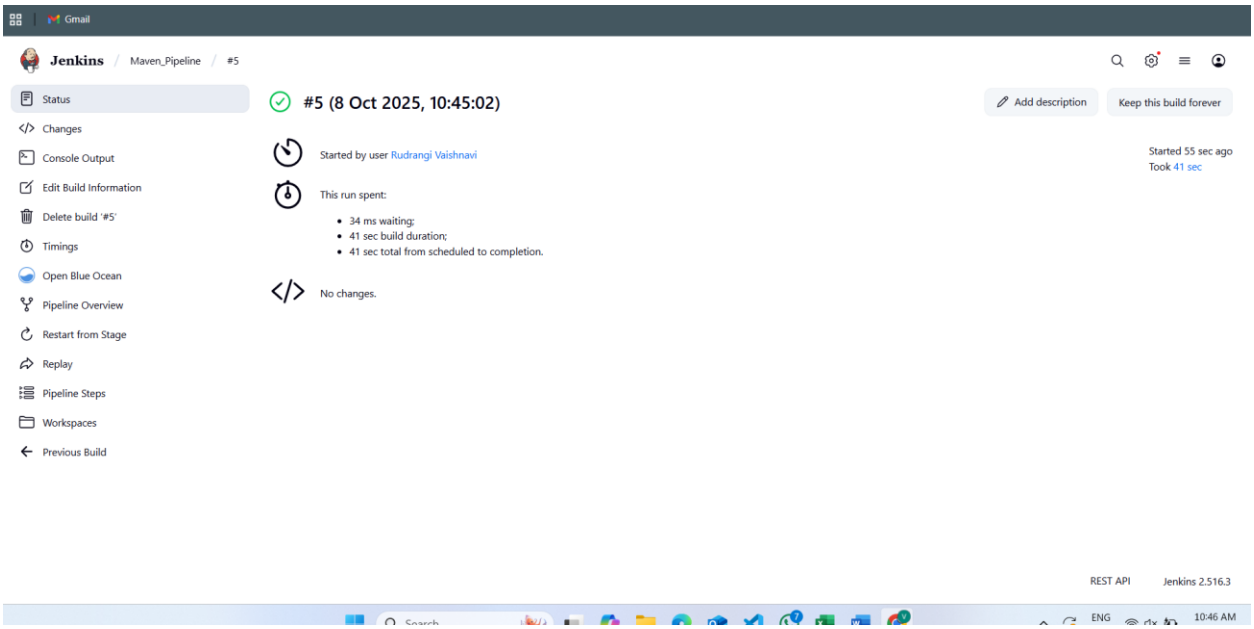
Here code for pipeline -script

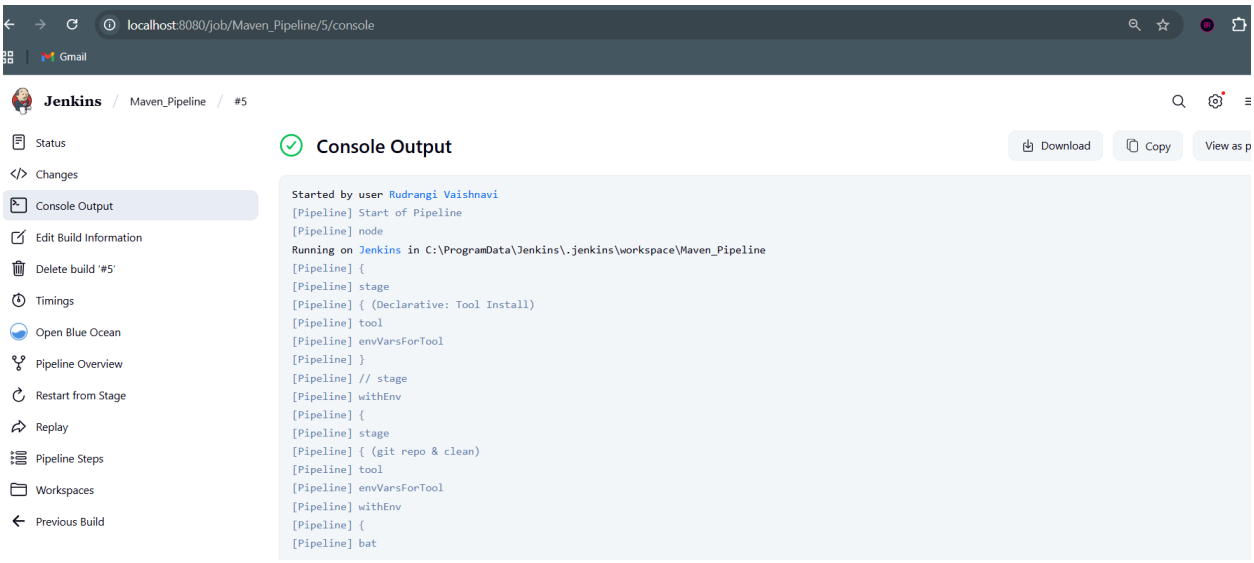
Copy the code there

```
pipeline {
   agent any
   tools{
      maven 'MAVEN-HOME'
   }
```

```
    stages {
        stage('git repo & clean') {
            steps {
                //bat "rmdir  /s /q mavenjava"
                bat "git clone provide your github link"
                bat "mvn clean -f mavenjava"
            }
        }
        stage('install') {
            steps {
                bat "mvn install -f mavenjava" #project name#
            }
        }
        stage('test') {
            steps {
                bat "mvn test -f mavenjava"
            }
        }
        stage('package') {
            steps {
                bat "mvn package -f mavenjava"
            }
        }
    }
}
```

Apply and save





SBQ's:

1. Your manager asks you to clean the workspace before building — which stage in this pipeline takes care of it?

   A. The git repo & clean stage cleans the workspace before building.

2. The GitHub repository link is missing in the script — where exactly do you provide it?

   A. Inside the git repo & clean stage in the line: bat "git clone https://github.com/yourname/repo.git"

3. If Maven is not configured globally in Jenkins, which section of the pipeline will fail first?

   A. The tools { maven 'MAVEN_HOME' } section — Jenkins cannot find the Maven installation.

4. A teammate complains the pipeline is not creating .war files — which stage is responsible?

   A. The package stage (mvn package) creates .jar or .war files.

5. Your test cases failed, but the pipeline still continued — how will Jenkins behave in the test stage?

   A. Jenkins will mark the stage as failed, and by default the pipeline stops unless configured with catchError.

6. Instead of running nightly builds, you want this pipeline to trigger only when GitHub changes occur — where will you configure it?

   A. In the Build Triggers section — enable **"**GitHub hook trigger for GITScm polling.**"**

7. If you replace mvn clean with mvn compile, what difference will it make to the project build?

   A. mvn clean deletes old build files and mvn compile just compiles the code without cleaning.

8. The project folder name is not mavenjava but studentapp — which parts of the script must you edit?

A. All Maven commands with -f mavenjava → change to -f studentapp.

9. If the install stage fails, will the test and package stages still run in this pipeline?

   A. No, Jenkins stops execution at the first failed stage.

10. A student asks where to add deployment steps to Tomcat after packaging — which is the best place in this script?

    A. Add a new stage('deploy') after the package stage.

11. Why is tools { maven 'MAVEN-HOME' } used in this pipeline?

    A. It tells Jenkins which Maven installation to use for running Maven commands.

12. If GitHub credentials are private, how can you secure them in the git repo & clean stage?

    A. Store credentials in Jenkins Credentials Manager and use them with the credentials block.

13. Which Jenkins plugin is needed to recognize the pipeline { ... } structure in this script?

    A. The Pipeline Plugin (Workflow Plugin).

14. In Windows, the script uses bat commands — what change would you make if Jenkins runs on Linux?

    A. Replace all bat with sh

15. How will you modify the pipeline to stop execution if the GitHub clone command fails?

Your project lead asks you to print a welcome message in Jenkins to confirm the pipeline is working — how would you script it?

    A. Jenkins already stops on failure by default, but to be explicit, use:
    bat "git clone <repo>" || error('Git clone failed!')

16. A teammate forgot to pull the latest GitHub code before building; how can you fix this in your pipeline?

    A. Use git pull or always clean and re-clone in the git repo & clean stage to fetch the latest code.

17. Your Java file Hello.java must be compiled every time code is committed — how will you add this step?

    A. Add a compile stage:
    stage('compile'){
    steps{
    bat "mvn compile -f Maven"
    }
    }

18. Tests should stop the pipeline if they fail — where will you put the mvn test command?

A. Inside the test stage, since any failure automatically halts later stages.

19. Every evening at 6 PM your pipeline should run automatically — how can you set this in Jenkins?

    A. In Build Triggers → Build periodically: H 18 * * *

20. You only want to package the project if compilation succeeds — how would you connect the stages?

    A. Jenkins stages already run sequentially, so package runs only if compile or install succeed.

21. Your professor wants a .jar file generated for submission — what pipeline stage will you add?

    A. Add a **jar stage** using: bat “mvn package -f Maven”

22. A Git clone step is failing due to wrong credentials — how would you secure and use them in your script?

    A. Save Git credentials in Jenkins (Manage Jenkins → Credentials) and use: withCredentials([usernamePassword(credentialsId: 'git-creds', usernameVariable: 'USER', passwordVariable: 'PASS')]) { bat "git clone https://${USER}:${PASS}@github.com/bhavagna06/Maven.git"}

23. A teammate wants to see the build number inside console logs — how do you print it in the pipeline?

    A. echo "Running Build #${env.BUILD_NUMBER}"