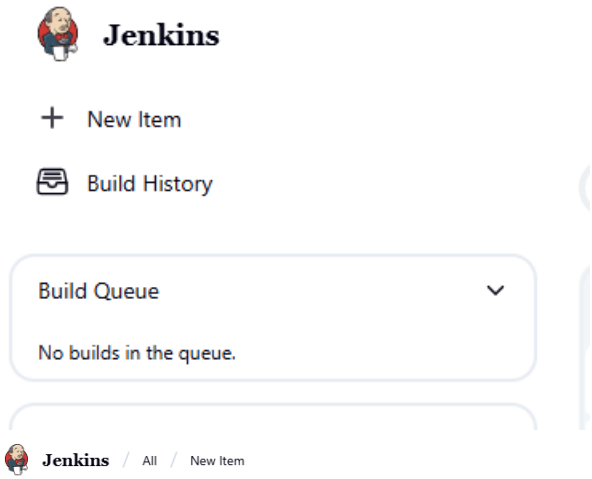
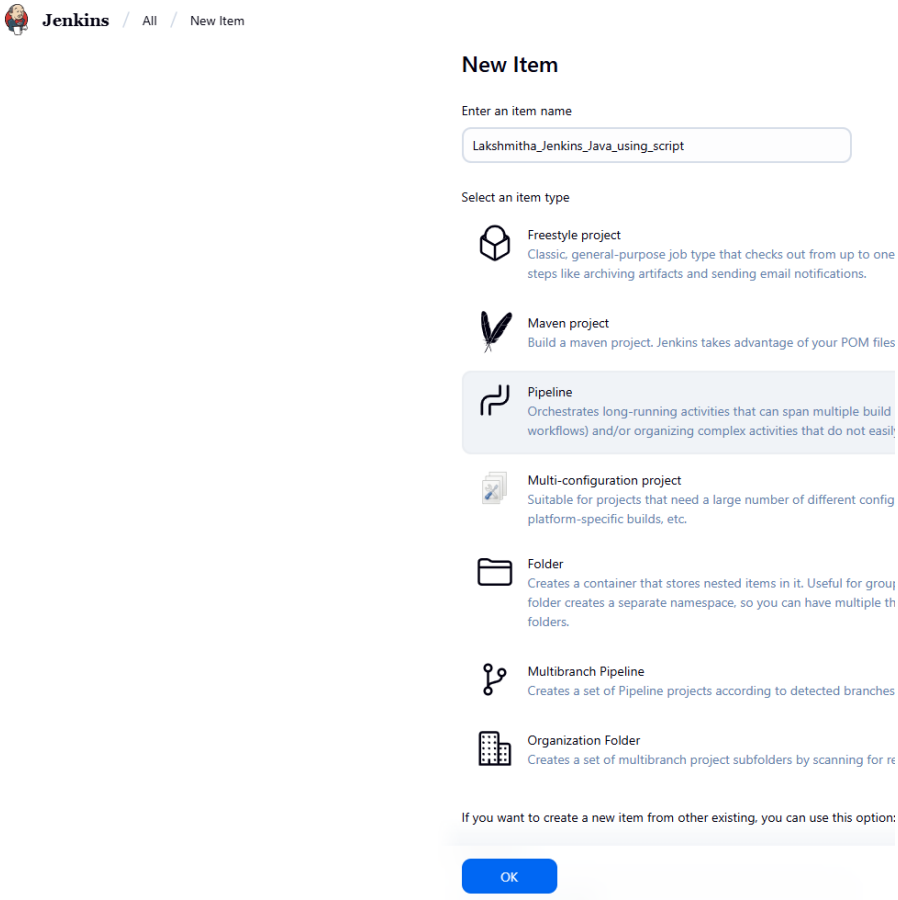
Bade Lakshmitha Reddy – 23BD1A6626

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

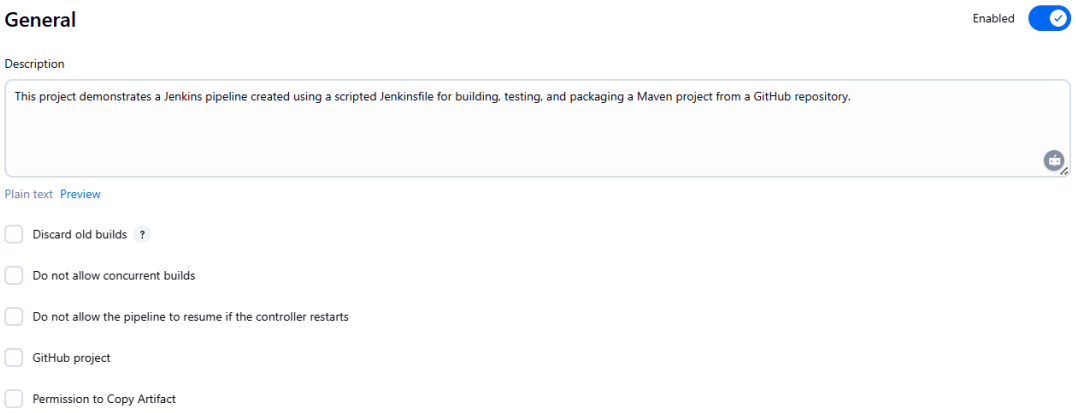




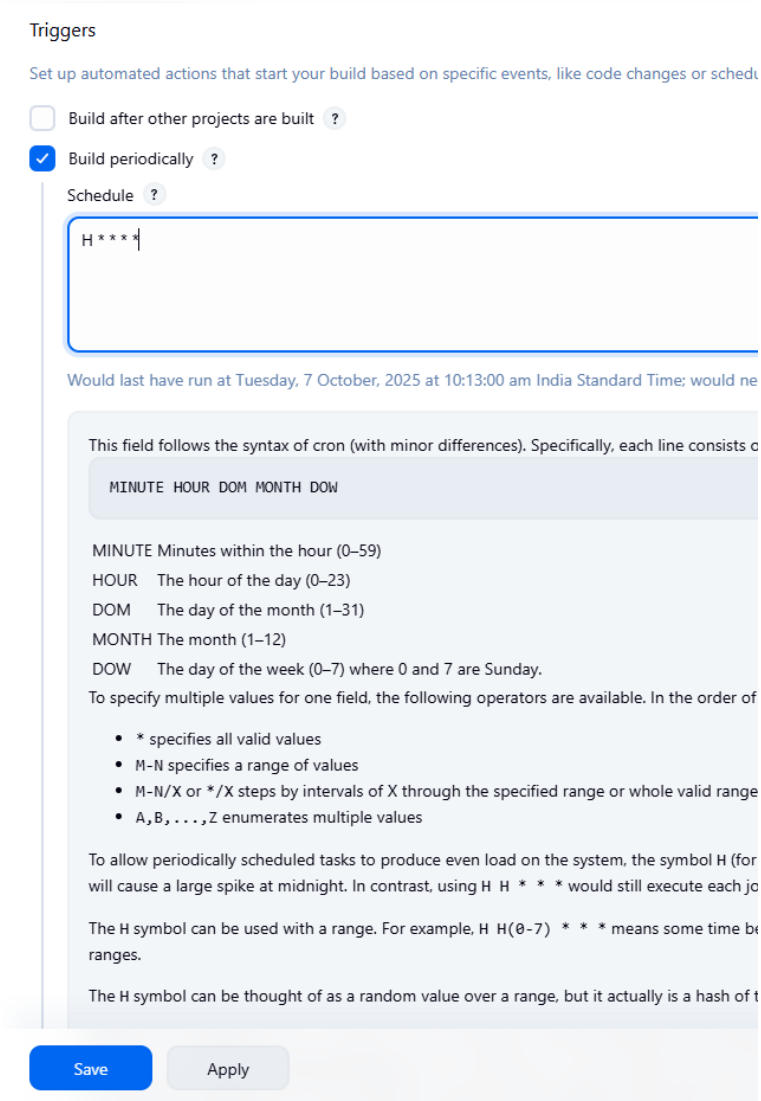
Procedure

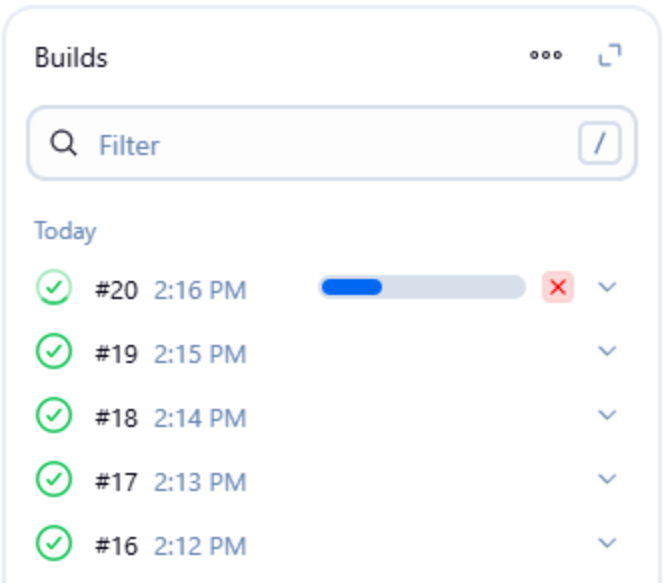
General :

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"
        }
    }
  }
}
```

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script ?

```
pipeline {
    agent any
    tools {
        maven 'MAVEN_HOME'
    }
    stages {
        stage('Clean Workspace') {
            steps {
                deleteDir()
            }
        }
        stage('Git Repo & Clean') {
            steps {
                bat "rmdir /s /q Lakshmitha-Maven-JavaProject || exit 0"
                bat "git clone https://github.com/LakshmithaReddy1807/Lakshmitha-Maven-JavaProject.git"
```

Use Groovy Sandbox ?

Pipeline Syntax

Advanced

Advanced

Save Apply

```
        }
        stage('Git Repo & Clean') {
            steps {
                bat "rmdir /s /q Lakshmitha-Maven-JavaProject || exit 0"
                bat "git clone https://github.com/LakshmithaReddy1807/Lakshmitha-Maven-JavaProject.git"
                bat "mvn clean -f Lakshmitha-Maven-JavaProject"
            }
        }

        stage('Install') {
            steps {
                bat "mvn install -f Lakshmitha-Maven-JavaProject"
            }
        }

        stage('Test') {
            steps {
                bat "mvn test -f Lakshmitha-Maven-JavaProject"
            }
        }

        stage('Package') {
            steps {
                bat "mvn package -f Lakshmitha-Maven-JavaProject"
            }
        }
    }
}
```

Apply and save

Console Output

```
Started by user Bade Lakshmitha Reddy
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\Lakshmitha_Jenkins_Jav
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
```

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 handles workspace cleaning. It can include a command like `bat "rmdir /s /q Lakshmitha-Maven-JavaProject || exit 0"` to delete old project files before cloning.

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

   A) Inside **"Git Repo & Clean"** stage, in line:
   `bat " git clone https://github.com/LakshmithaReddy1807/Lakshmitha-Maven-JavaProject.git"`

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

   A) The pipeline will fail at the **tools** section during initialization, since tools { maven 'MAVEN-HOME' } depends on a globally configured Maven installation.

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

   A) The **"Package"** stage is responsible for creating .jar or .war files using mvn package.

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

   A) By default, Jenkins marks the **Test stage** as failed but **continues** to later stages unless `error` or `failFast` conditions are used.
   You can enforce a stop using: `bat "mvn test -f project/pom.xml"`
   `error("Stopping pipeline since tests failed.")`

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 of the project configuration, select **"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 previous build artifacts, ensuring a fresh build.
   mvn compile only compiles source code and doesn't clean or remove old files.

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

   A) Every Maven command path and folder reference:
   mvn clean -f studentapp/pom.xml, mvn install -f studentapp/pom.xml, etc.

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

   A) No. Jenkins Declarative Pipelines stop executing subsequent stages if any previous stage fails by default.

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 **"Deploy"** stage **after the Package stage**, for example:
    stage('Deploy') {

```
    steps {
      bat "mvn tomcat7:deploy -f project/pom.xml"
    }
  }
```

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

    A) It tells Jenkins to automatically install and use the configured Maven tool from **Manage Jenkins → Global Tool Configuration**, ensuring Maven is available for build commands.

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

    A) Store credentials in Jenkins using **Credentials Manager**, then use:

```
withCredentials([usernamePassword(credentialsId: 'git-creds', usernameVariable: 'USER',
passwordVariable: 'PASS')])
{
bat "git clone https://${USER}:${PASS}@github.com/username/repo.git"
}
```

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

    A) The **Pipeline Plugin** (also called **Workflow Plugin**) must be installed.

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

    A) Replace all bat commands with sh commands.

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) Add a condition or use returnStatus:

```
script {
  def status = bat(script: "git clone https://github.com/...git", returnStatus: true)
  if (status != 0) {
    error("Git clone failed — stopping pipeline.")
  }
}
```

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

    A) Add a simple echo step at the beginning:

```
stage('Welcome') {
  steps {
    echo "Welcome to Jenkins Pipeline — Build Started Successfully!"
```

```
    }
  }
```

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

    A) Add a pull step before building: bat "git -C Lakshmitha-Maven-JavaProject pull"
    or delete and re-clone the repository each run.

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

    A) Add a **Compile** stage after cloning:

```
stage('Compile') {
  steps {
    bat "mvn compile -f project/pom.xml"
  }
}
```

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

    A) Under **Build Triggers → Build periodically**, enter the CRON expression: H 18 * * *

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

    A) In Declarative Pipelines, this is automatic — each stage depends on the previous one.
    But you can also use:
    when { success() }
    in the package stage to ensure it runs only on successful compilation.

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

    A) The **Package** stage handles .jar creation using: mvn package

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

    A) Store credentials in **Jenkins Credentials Manager**, then access them securely:

```
withCredentials([usernamePassword(credentialsId: 'git-creds', usernameVariable: 'USER',
passwordVariable: 'PASS')]) {
  bat "git clone https://${USER}:${PASS}@github.com/username/repo.git"
}
```

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

    A) Use the Jenkins environment variable BUILD_NUMBER:
    echo "Running Build Number: ${env.BUILD_NUMBER}"