Zainab Imran

22-10295

**SQA ASSIGNMENT 3**

**CI CD PIPELINE**

**Code 1:**

Jenkinsfile (Declarative Pipeline)

pipeline {

agent any

options {

skipStagesAfterUnstable()

}

stages {

stage('Build') {

steps {

sh 'make'

}

}

stage('Test'){

steps {

sh 'make check'

junit 'reports/\*/.xml'

}

}

stage('Deploy') {

steps {

sh 'make publish'

}

}

}

}

**Code 2:**

PHP

Jenkinsfile (Declarative Pipeline)

pipeline {

agent { docker { image 'php:8.1.0-alpine' } }

stages {

stage('build') {

steps {

sh 'php --version'

}

}

}

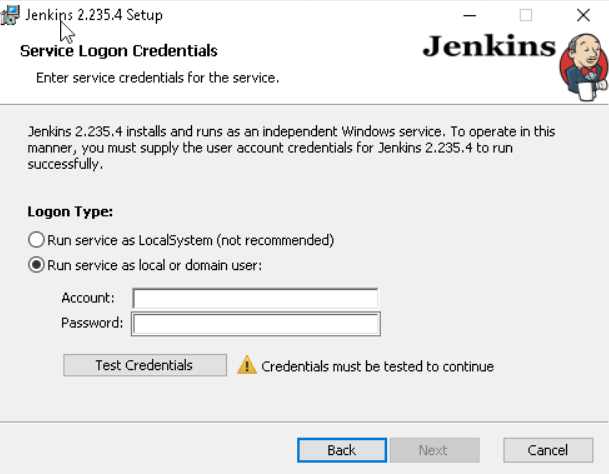
}

**Install Jenkins:**

1. Search https://www.jenkins.io/download/and select stage.
2. Download, unfasten and run the bundle.
3. In the Jenkin Setup screen, click next.
4. Choose the place you want Jenkins to be installed, go with default and choose next
5. Click Install
6. Once installation is finished, click Finish.

Graphical user interface, application, Teams

Description automatically generated



**Create CI CD Pipeline**

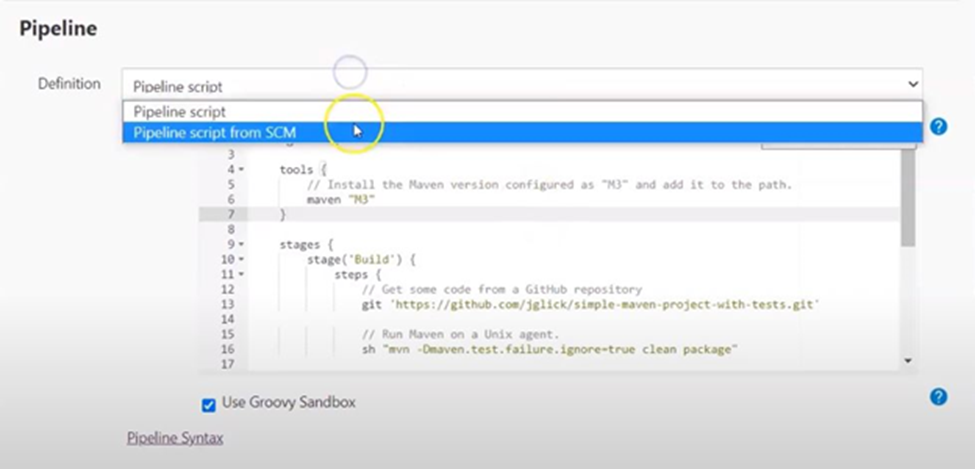
1. Execute Jenkins as Java parallel.
2. Create another Jenkins work.
3. Create a pipeline work.
4. Configure and execute a pipeline work through an immediate content.
5. Configure and execute a pipeline work with SCM.

Graphical user interface, text, application, email

Description automatically generated

The parameters that we modified are as follows:

1. Used the Poll SCM as the form trigger, by selecting this option. Jenkins is trained to regularly check the Git store (as demonstrated by \* \* \* \* \*). If the repo has changed since the last survey, the occupation is activated.
2. We chose the archive URL and the requirements for the actual process. Ace is the branch.



1. All of the work's code is being added to a Jenkinsfile and stored in a vault with the code.
2. Setup the Jenkins GitHub credentials Go to /accreditations/store/framework/area/ /new Credentials and add the two targets' qualifications. In view of the fact that you will refer to each later, make sure you give them each a meaningful ID and description.
3. Create the JenkinsFile to teach Jenkins how to build, test, dockerize, distribute, and communicate our application.
4. Now, CI/CD Pipeline is being tested:
5. The final component of this is where we really scrutinize our work. We will ensure that our code follows the pipeline till it reaches the bunch by committing it to GitHub:

Add our files here:

**add to git**

Committhe adjustments: "Beginning commit"

**git commit - m**

1. Then, Git push to push to GitHub

Jenkins gives two options: either hit "Form Now" and trust that the job will be started accordingly.

In the event that the job is successful, we may use the following command to view our submitted application:

**wide, kubectl obtain nodes**

****

