Name : Siddhesh Ghangale

Creating Studentapp Artifact using Maven and Hosted it using Jenkins Pipeline

Code:

```
pipeline{
    agent {
        label 'agent1'
    }
    stages{
        // Install maven
        stage('Install Maven')
        {
            steps{
                sh" sudo apt-getupdate"
                sh " sudo apt-get install maven -y"
            }
        }
        stage('Pull a file from git')
        {
            steps{
                git 'https://github.com/AnupDudhe/studentapp-ui'
            }
        }
        stage('Build the code')
        {
            steps{
                sh 'mvn clean package'
            }
        }

        stage('Installing Apache2')
        {
            steps{
                sh "sudo apt-get update"
                sh" sudo apt-getinstallapache2-y"
                sh" sudo systemctlstartapache2"

            }
        }
        stage('Downloading Tomcat 8')
```

```
        {
            steps{
                sh 'sudo wget
https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.99/bin/apache-tomcat-8.5.99.
tar.gz'
                sh 'sudo tar -xvf apache-tomcat-8.5.99.tar.gz'
            }
        }
        // moveing the war file to the tomcat webapps folder
        stage('Deploying war file')
        {
            steps{
                sh" sudo mvtarget/*.war
apache-tomcat-8.5.99/webapps/student.war"
            }
        }
        stage('Starting Apache 8')
        {
            steps{
                sh 'sudo ./apache-tomcat-8.5.99/bin/startup.sh'
            }
        }
    }
}
```

Code Overview:

Overview

This pipeline is designed to install Maven, pull a Java web application from a Git repository, build the application using Maven, install Apache2, download and extract Apache Tomcat 8, deploy the built WAR file to Tomcat, and start Apache Tomcat 8.

Pipeline Structure

- Agent: The pipeline is configured to run on an agent labeled 'agent1'.
- Stages: The pipeline consists of several stages, each representing a distinct phase of the deployment process.

Stages

Install Maven

- This stage installs Maven by updating the package list and installing Maven.

Pull a file from git

- This stage clones the studentapp-ui repository (https://github.com/AnupDudhe/studentapp-ui), which presumably contains the Java web application source code.

Build the code

- This stage builds the Java web application using Maven (mvn clean package).

Installing Apache2

- This stage installs Apache2 web server by updating the package list, installing Apache2, and starting the Apache2 service.

Downloading Tomcat 8

- This stage downloads Apache Tomcat 8.5.99 from the official Apache website and extracts the downloaded archive.

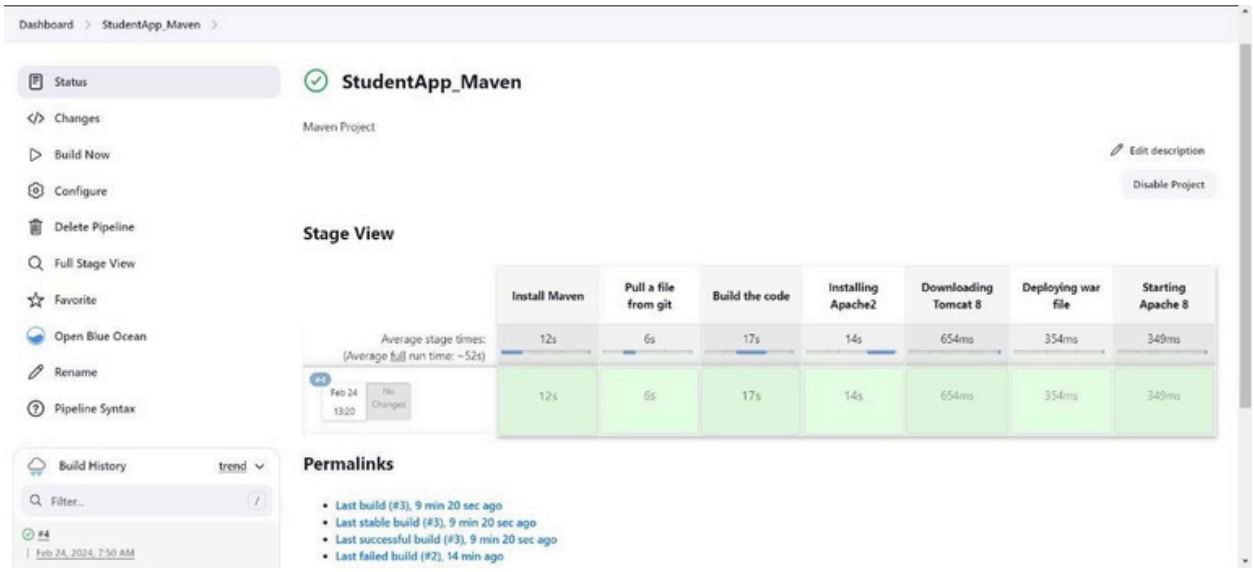
Deploying war file

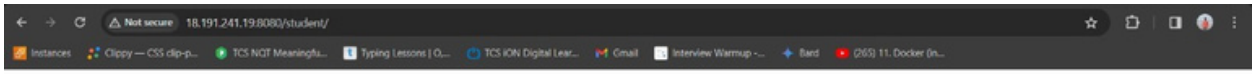
- This stage moves the built WAR file from the target directory to the webapps directory of Apache Tomcat 8.5.99, effectively deploying the WAR file.

Starting Apache Tomcat 8

- This stage starts Apache Tomcat 8.5.99 using the startup.sh script.

Output:





Student Registration Form

Student Name
Student Address
Student Age
Student Qualification
Student Percentage
Year Passed
register