

Name : Siddhesh Ghangale

Created S3 bucket and uploaded studentapp artifact into it using Jenkins pipeline.

Code :

```
pipeline {
  agent {
    label 'maven'
  }
  stages {
    stage('Install Maven') {
      steps {
        sh 'sudo apt-get update'
        sh 'sudo apt-get install maven -y'
      }
    }
    stage('Install AWS CLI') {
      steps {
        sh 'sudo apt-get install awscli -y'
      }
    }
    stage('Pull a file from git') {
      steps {
        git 'https://github.com/AnupDudhe/studentapp-ui.git'
      }
    }
    stage('Build the code') {
      steps {
        sh 'mvn clean package'
      }
    }
    stage('Creating a S3 bucket') {
      steps {
        sh 'aws s3api create-bucket --bucket
studentappbucket775604 --region us-east-2 --create-bucket-configuration
LocationConstraint=us-east-2'
      }
    }
  }
}
```

```

    stage('Uploading the war file to S3') {
        steps {
            sh 'aws s3 cp target/*.war s3://studentappbucket775604'
        }
    }
}

```

Overview:

## Overview

This pipeline automates the build, testing, and deployment process of a Java web application using Maven and AWS S3.

## Pipeline Structure

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

## Stages

### Install Maven

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

### Install AWS CLI

- This stage installs the AWS Command Line Interface (CLI) tool.

### Pull a file from git

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

### Build the code

- This stage cleans the project using `mvn clean` and then builds the Java web application using `mvn package`.

### Creating a S3 bucket

- This stage creates an S3 bucket named `studentappbucket775604` in the `us-east-2` region.

### Uploading the war file to S3

- This stage uploads the built WAR file from the `target` directory to the `studentappbucket775604` S3 bucket.

Output:

