

## SIT223 task 6.1c: Amit Kafe – pipeline definition

### Continuous Integration and Deployment with Jenkins and GitHub

#### Pipeline code

```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                echo "Use Maven to automatically compile and package code..."
            }
        }
        stage('Unit and Integration testing') {
            steps {
                echo "Running unit testing using JUnit"
                echo "Running integration tests using Selenium"
            }
        }
    }

    post {
        always {
            emailx(
                to: "amit.kafe5987@gmail.com",
                subject: "Unit and Integration testing: Success",
                body: "<p>Stage 'Unit and Integration Tests' completed with status  

${currentBuild.result}</p><p>Check console output at ${env.BUILD_URL} to view the  

result.</p>",
                attachLog: true
            )
        }
    }
}
```

```

}

stage('Code Analysis'){
    steps {
        echo "Running SonarQube scanner..."
        echo "Analyzing code"
    }
}

stage('Security Scan'){
    steps {
        echo "Run OWASP Dependency-Check to perform security scan"
    }
    post {
        always {
            emailx(
                to: "amit.kafe5987@gmail.com",
                subject: "Security scan successful",
                body: "<p>Stage 'Security Scan' completed with status  

${currentBuild.result}</p><p>Check console output at ${env.BUILD_URL} to view the  

result.</p>",
                attachLog: true
            )
        }
    }
}

stage('Deploy to staging'){
    steps {
        echo "Using AWS CodeDeploy to deploy to staging"
        echo "Deploying the application to staging server: SLB TYA"
    }
}

```

```
}  
  
stage('Integration Tests on staging') {  
    steps {  
        echo "Executing integration tests using Selenium"  
        echo "mvn test -Dtest=IntegrationTest"  
    }  
}  
  
stage('Deploy to production') {  
    steps {  
        echo "Using AWS CodeDeploy to deploy the application to production"  
        echo "aws deploy create-deployment"  
        echo "application-name <SIT223 Amit Kafe>"  
    }  
}  
}  
}
```

## **Console output**

Started by user Amit Kafe

Obtained jenkinsfile from git <https://github.com/Amit5879/jenkins-pipeline-updated.git>

[Pipeline] Start of Pipeline

[Pipeline] node

Running on Jenkins

in /var/jenkins\_home/workspace/task 6.1c new

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

using credential 3f4f768d-92d9-46e7-b7a3-3da9f12f9158

```
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/task 6.1c new/.git #  
timeout=10
```

Fetching changes from the remote Git repository

```
> git config remote.origin.url https://github.com/Amit5879/jenkins-pipeline-updated.git  
# timeout=10
```

Fetching upstream changes from <https://github.com/Amit5879/jenkins-pipeline-updated.git>

```
> git --version # timeout=10
```

```
> git --version # 'git version 2.39.2'
```

using GIT\_ASKPASS to set credentials

```
> git fetch --tags --force --progress -- https://github.com/Amit5879/jenkins-pipeline-updated.git +refs/heads/*:refs/remotes/origin/* # timeout=10
```

```
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
```

```
Checking out Revision 93cea7cd9fa86559529b826ee3fec8555f6f6f5f  
(refs/remotes/origin/main)
```

```
> git config core.sparsecheckout # timeout=10
```

```
> git checkout -f 93cea7cd9fa86559529b826ee3fec8555f6f6f5f # timeout=10
```

```
Commit message: "new update"
```

```
> git rev-list --no-walk 93cea7cd9fa86559529b826ee3fec8555f6f6f5f # timeout=10
```

```
[Pipeline] }
```

```
[Pipeline] // stage
```

```
[Pipeline] withEnv
```

```
[Pipeline] {
```

```
[Pipeline] stage
```

```
[Pipeline] { (Build)
```

```
[Pipeline] echo
```

```
Use Maven to automatically compile and package code...
```

```
[Pipeline] }
```

```
[Pipeline] // stage
```

```
[Pipeline] stage
```

```
[Pipeline] { (Unit and Integration testing)
```

```
[Pipeline] echo
```

```
Running unit testing using JUnit
```

```
[Pipeline] echo
```

```
Running integration tests using Selenium
```

```
Post stage
```

```
[Pipeline] emailx
```

```
Sending email to: amit.kafe5987@gmail.com
```

```
[Pipeline] }
```

```
[Pipeline] // stage
```

[Pipeline] stage

[Pipeline] { (Code Analysis)

[Pipeline] echo

Running SonarQube scanner...

[Pipeline] echo

Analyzing code

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Security Scan)

[Pipeline] echo

Run OWASP Dependency-Check to perform security scan

Post stage

[Pipeline] emailx

Sending email to: amit.kafe5987@gmail.com

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Deploy to staging)

[Pipeline] echo

Using AWS CodeDeploy to deploy to staging

[Pipeline] echo

Deploying the application to staging server: SLB TYA

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Integration Tests on staging)

[Pipeline] echo

Executing integration tests using Selenium

[Pipeline] echo

mvn test -Dtest=IntegrationTest

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Deploy to production)

[Pipeline] echo

Using AWS CodeDeploy to deploy the application to production

[Pipeline] echo

aws deploy create-deployment

[Pipeline] echo

application-name <SIT223 Amit Kafe>

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS