

Jenkins

After uploading to git, we need to setup jenkins, it will upload to cloudHub.

Install jenkins

* Create new freestyle project

* Give git repo url and add credentials



We can refer to specific branch in git



If we run this application, the project will deploy to cloudHub.

And we can do this to test env, just change mvn code -Ptest.

Trigger it automatically:

Two ways to trigger Jenkins job automatically

- ◆ When ever we push the code change to GIT repository then the Jenkins job triggered automatically to build and continuous the deployment.
- ◆ There are two ways to trigger build automatically.
 - ◇ 1. **Push notification:** SCM tool (GITHUB) notifies Jenkins on new commit push.
 - ◇ 2. **Polling:** Jenkins polls in regular schedules.

Way-1:

We need to create webhook in git.

Got to git repo > settings > webhook > Add > provide Jenkins URL

payload URL -

Content type

Secret

Which events would you like to trigger this webhook?

☒ Just the push event.

☐ Send me everything.

☐ Let me select individual events.

☒ Active
We will deliver event details when this hook is triggered.

Choose this option in Jenkins

Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts) ?
- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

Way-2:

Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?

☐ Build after other projects are built ?

☒ Build periodically ?

Schedule ?

⚠ No schedules so will never run

Choose this option. For time periods, we need to give in corn expression.

Jenkins Pipeline:

Jenkins pipeline

- ◆ Pipeline allows us to define set of stages while continuous deployment, we can define stages to execute continues deployment stage by stage.
- ◆ For example: We can configure pipeline to build, execute MUnit tests and deploy.

Declarative: Checkout SCM	Build	Test	Deployment
3s	34s	3s	4min 0s

We need to put the jenkins file in Git repo. It will take that file and deploy it accordingly.

* in that file we put stages like build, test, deploy, etc.

The screenshot shows the Jenkins Pipeline configuration page. The 'Definition' section is set to 'Pipeline script from SCM'. The 'SCM' is set to 'Git'. The 'Repositories' section shows a 'Repository URL' of 'https://github.com/chinnamulesoft/product-api.git'. The 'Credentials' section shows a dropdown menu with 'chinnamulesoft/*****' and an 'Add' button. There are also 'Advanced...' and 'Add Repository' buttons at the bottom right.

Create a Jenkinsfile in project under root project

Refer: <https://github.com/KarampudiKarthik/Mulesoft-Notes/blob/main/Jenkinsfile>

```

1 pipeline {
2
3   agent any
4
5   stages {
6     stage('Build') {
7       steps {
8         bat 'mvn -B -U -e -V clean -DskipTests package'
9       }
10    }
11
12    stage('Test') {
13      steps {
14        bat "mvn test"
15      }
16    }
17
18    stage('Deployment') {
19      steps {
20        bat 'mvn -U -V -e -B -DskipTests -Pdev deploy -DmuleDeploy'
21      }
22    }
23  }
24 }

```

We need to put maven MUnit plugin in pom.xml

Refer: <https://docs.mulesoft.com/munit/latest/munit-maven-plugin>

For dependency, when we create munit test case, dependency will automatically create in pom.xml

>>> configure maven in jenkins

Go to manage jenkins > tools >

Maven installations ^
Edited

Add Maven

Maven

Name

MAVEN3

☒ Install automatically ?

Install from Apache

Version

3.9.9

Add Installer v

Then build the project.

We can put anypoint username and password and client id and client secret in jenkins rather than settings file in .m2

Go to jenkins > credentials > global > Add >

For username and password:

Kind


Username with password

For client id and secret:


Kind

Secret text

Do it for dev and test env

 **Global credentials (unrestricted)**

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
 e47b306f-5abb-4f6b-a994-a11846ff8851	chinnamulesoft/*****	Username with password	
 ANYPOINT_CREDENTIALS	chinna04/*****	Username with password	
 DEV_CLIENT_ID	DEV_CLIENT_ID	Secret text	
 DEV_CLIENT_SECRET	DEV_CLIENT_SECRET	Secret text	
 TEST_CLIENT_ID	TEST_CLIENT_ID	Secret text	
 TEST_CLIENT_SECRET	TEST_CLIENT_SECRET	Secret text	

Icon: S M L

We need to use this in Jenkinsfile.

For global (anypoint username and password)

```
1 pipeline {
2
3   agent any
4
5   environment {
6     ANYPOINT_CREDS = credentials('ANYPOINT_CREDENTIALS')
7   }
8 }
```

For dev env:

```
1 }
2
3 stage('Deployment') {
4
5   environment {
6     CLIENT_ID = credentials('DEV_CLIENT_ID')
7     CLIENT_SECRET = credentials('DEV_CLIENT_SECRET')
8   }
9
10  steps {
11    bat 'mvn -U -V -e -B -DskipTests -Pdev deploy -DmuleDeploy'
12  }
13 }
```

Change in pom.xml

```
1 <!-- API ID -->
2 <cloudHubDeployment>
3   <uri>https://anypoint.mulesoft.com</uri>
4   <username>${anypoint.username}</username>
5   <password>${anypoint.password}</password>
6   <muleVersion>4.3.0</muleVersion>
7   <objectStore>true</objectStore>
8   <properties>
9     <!-- API ID -->
10    <api.id>16833706</api.id>
11    <anypoint.platform.client.id>${anypoint.platform.client.id}</anypoint.platform.client.id>
12    <anypoint.platform.client.secret>${anypoint.platform.client.secret}</anypoint.platform.client.secret>
13  </properties>
14 </cloudHubDeployment>
```

Copy this in deploy stage in Jenkinsfile:

```
bat 'mvn clean install deploy -Danypoint.username="%ANYPOINT_CREDS_USR%" -  
Danypoint.password="%ANYPOINT_CREDS_PSW%" -Danypoint.platform_id="%CLIENT_ID%" -  
Danypoint.platform_secret="%CLIENT_SECRET%"'
```

```
1 stage('Deployment') {  
2  
3     environment {  
4         CLIENT_ID = credentials('DEV_CLIENT_ID')  
5         CLIENT_SECRET = credentials('DEV_CLIENT_SECRET')  
6     }  
7  
8     steps {  
9         bat 'mvn -U -V -e -B -DskipTests -Pdev deploy -DmuleDeploy -Danypoint.username="%ANYPOINT_CREDS_USR%" -Danypoint.password="%ANYPOINT_CREDS_PSW%" -Danypoint.platform_id="%CLIENT_ID%" -Danypoint.platform_secret="%CLIENT_SECRET%"'  
10    }  
11 }  
12 }
```