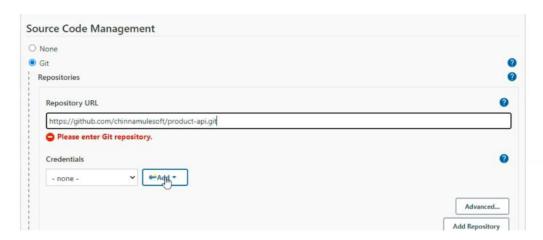
Jenkins

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



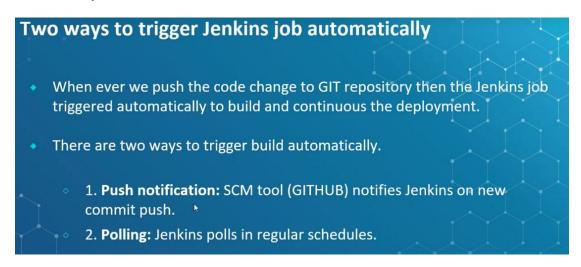
We can refer to specifi branch in git



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

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

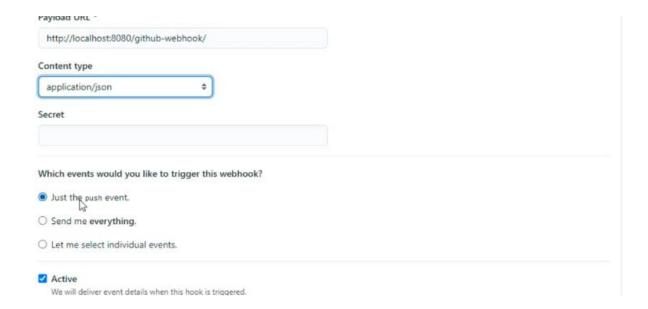
Trigger it automatically:



Way-1:

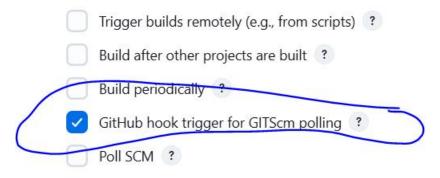
We need to create webhook in git.

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



Choose this option in Jenkins

Build Triggers



Way-2:



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

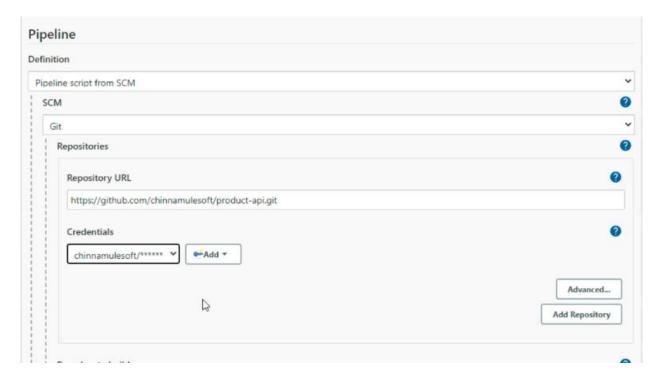
(

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. Build Test Deployment Build Test Deployment

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.



Create a Jenkinsfile in project under root project

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

```
pipeline {
    agent any

stages {
    stage('Build') {
    steps {
        bat 'mvn -B -U -e -V clean -DskipTests package'
    }
}

stage('Test') {
    steps {
        bat "mvn test"
}

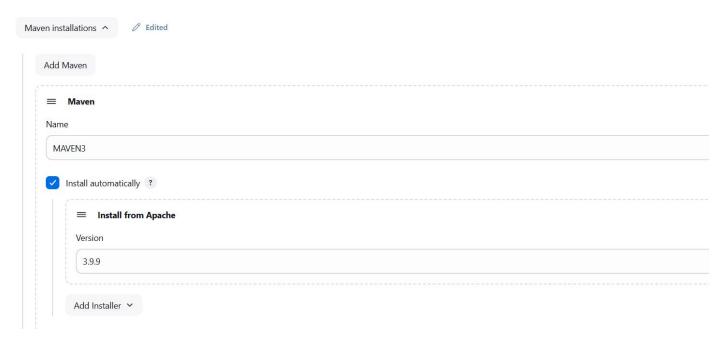
stage('Deployment') {
    steps {
        bat 'mvn -U -V -e -B -DskipTests -Pdev deploy -DmuleDeploy'
}
}
```

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 >



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 >

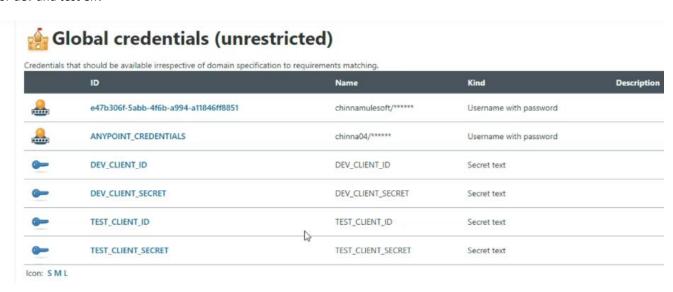
For username and passowrd:

```
Kind
Username with password
```

For client id and secret:

Kind
Secret text

Do it for dev and test env



We need to use this in Jenkinsfile.

For global (anypoint username and password)

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

For dev env:

```
stage('Deployment') {
    environment {
        CLIENT_ID = credentials('DEV_CLIENT_ID') ]
        CLIENT_SECRET = credentials('DEV_CLIENT_SECRET')
}

steps {
        bat 'myn -U -V -e -B -DskipTests -Pdex deploy -DmuleDeploy'
}
```

Change in pom.xml

Copy this in deploy stage in Jenkinsfile:

bat 'mvn clean install deploy -Danypoint.username="%ANYPOINT_CREDS_USR%" - Danypoint.password="%ANYPOINT_CREDS_PSW%" -Danypoint.platfor,.client_id="%CLIENT_ID%" - Danypoint.platfor,.client_secret="%CLIENT_SECRET%"

```
stage('Deployment') {

environment {

CLIENT_ID = credentials('DEV_CLIENT_ID')

CLIENT_SECRET = credentials('DEV_CLIENT_SECRET')

}

steps {

bat 'mvn -U -V -e -B -DskipTests -Pdev deploy -DmuleDeploy -Danypoint.username="%ANYPOINT_CREDS_USR%" -Danypoint.passs
}

}
```