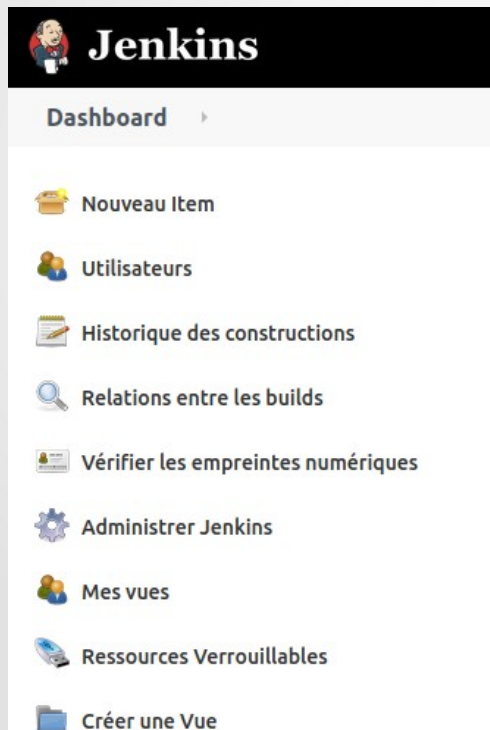


Projet 3 : Pipeline script : Git-Gitlab-Jenkins-Tomcat

Dans le chapitre précédent, nous avons travaillé en mode «projet Maven». Maintenant, je vais faire un exemple de build avec le plugging pipeline.

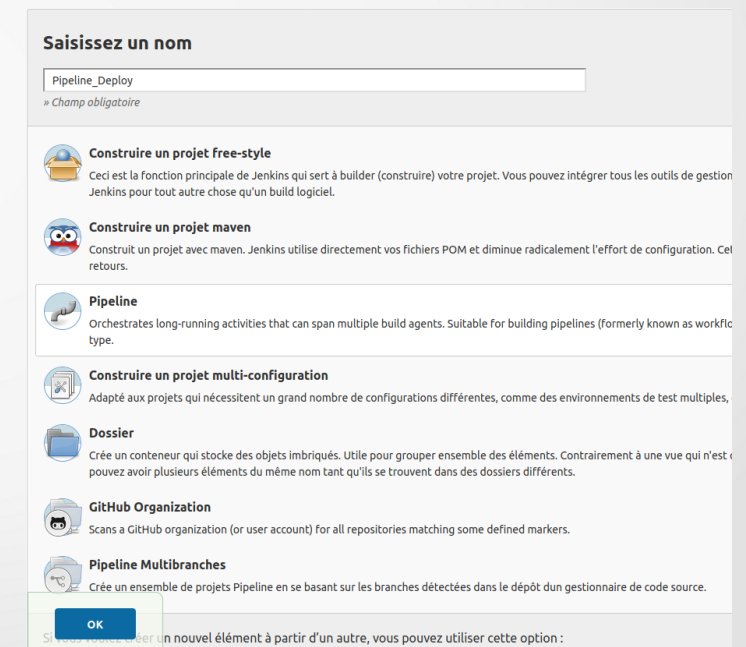
Nous allons reprendre notre ancien job « Buil_Code_Deploy » qu'on a réalisé en mode projet Maven puis on va le transformer en un pipeline job.

1- Pour créer un job pipeline, il faut aller sur la page d'accueil de Jenkins puis cliquer sur « Nouveau Item » depuis le menu gauche.



2- Une page apparaît vous demandant de choisir le nom de votre Item ainsi que son type. Mettez « Pipeline_Deploy » pour le nom puis choisissez « Pipeline » .

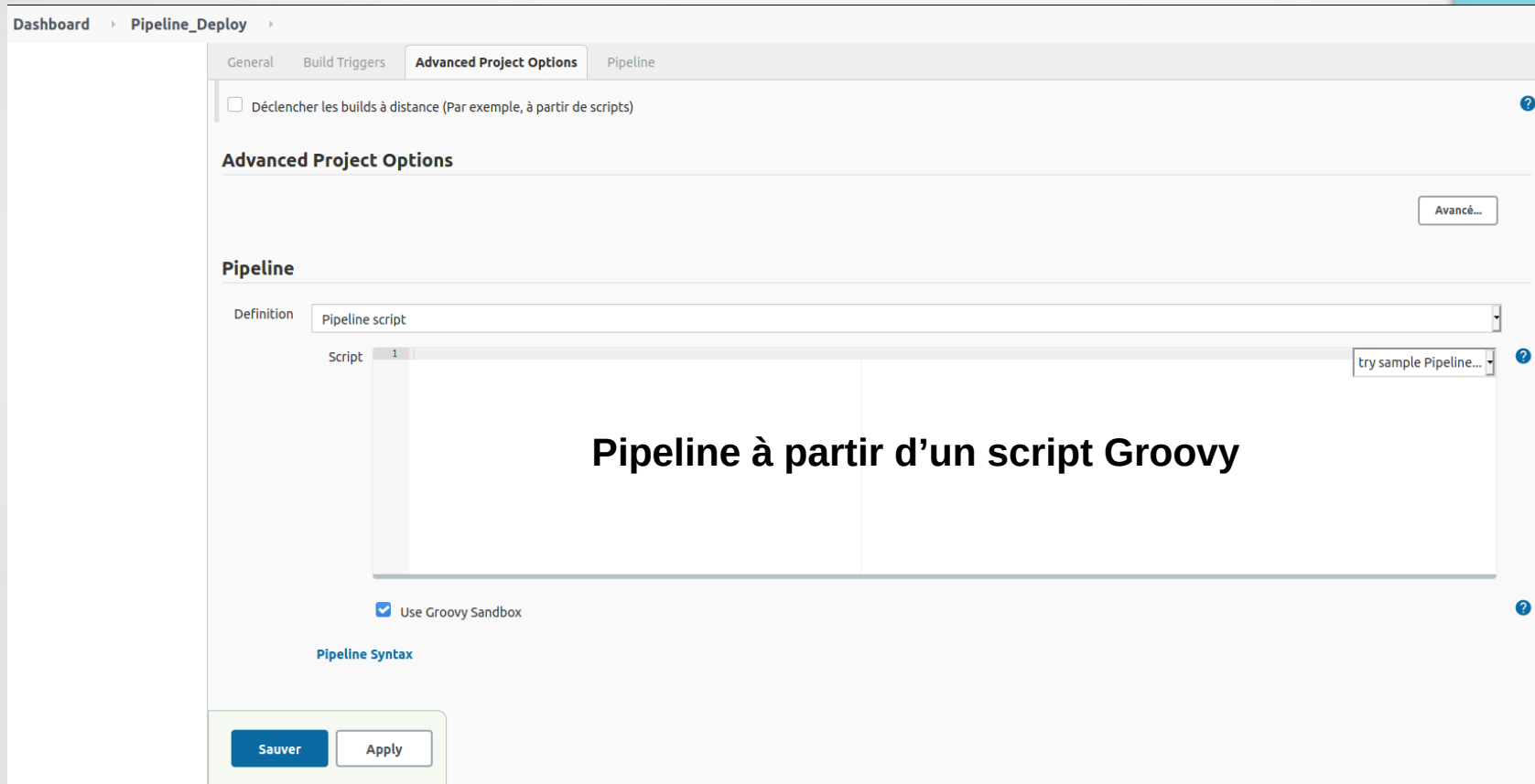
Appuyez ensuite sur OK.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Projet 3 : Pipeline script : Git-Gitlab-Jenkins-Tomcat

Vous allez arriver sur la page suivante :



The screenshot shows the Jenkins Pipeline configuration interface. At the top, there's a breadcrumb trail: Dashboard > Pipeline_Deploy. Below this, there are four tabs: General, Build Triggers, Advanced Project Options (which is selected), and Pipeline. Under the 'Advanced Project Options' tab, there's a checkbox labeled 'Déclencher les builds à distance (Par exemple, à partir de scripts)' which is currently unchecked. Below this, there's a section titled 'Advanced Project Options' with an 'Avancé...' button. The 'Pipeline' section is below that, showing a 'Definition' dropdown set to 'Pipeline script'. Under 'Script', there's a text area with the text 'Pipeline à partir d'un script Groovy'. To the right of the text area is a dropdown menu with 'try sample Pipeline...'. Below the text area, there's a checkbox labeled 'Use Groovy Sandbox' which is checked. At the bottom left, there's a link for 'Pipeline Syntax'. At the bottom right, there are two buttons: 'Sauver' (Save) and 'Apply'.

Dashboard > Pipeline_Deploy >

General Build Triggers **Advanced Project Options** Pipeline

☐ Déclencher les builds à distance (Par exemple, à partir de scripts)

Advanced Project Options

Avancé...

Pipeline

Definition Pipeline script

Script 1

try sample Pipeline...

Pipeline à partir d'un script Groovy

☒ Use Groovy Sandbox

[Pipeline Syntax](#)

Sauver Apply



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Projet 3 : Pipeline script : Git-Gitlab-Jenkins-Tomcat

Comme vous pouvez le voir, il faut écrire un script et plus précisément un script Groovy. Si vous ne connaissez pas grand chose à ce dernier, ne vous inquiétez pas. Jenkins va vous aider à vous familiariser avec Groovy à l'aide de l'option « Pipeline Syntax » que vous voyez sur la page. Une fois que vous cliquez dessus, vous atterrissez sur la page suivante :

Overview

This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)

Steps

Sample Step

Fichiers à archiver

[?](#) [?](#)

[Avancé...](#)

[Generate Pipeline Script](#)

Pipeline Syntax : générer un script Groovy

Global Variables

There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the [Global Variables Reference](#) for details.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Projet 3 : Pipeline script : Git-Gitlab-Jenkins-Tomcat

Le build en mode pipeline

Étape 1- La première étape consiste à choisir le nœud qui va être exécuter le pipeline

Steps

Sample Step

Label

Generate Pipeline Script

```
node {  
  // some block  
}
```

un nœud (node) représente un environnement pouvant exécuter un pipeline (une machine esclave) ;

- Copiez puis collez cette ligne dans la partie « Script » de la rubrique « Pipeline ». Vous devriez avoir :

Pipeline

Definition

Script

1

2

3

```
node {  
  // some block  
}
```

[try sample Pipeline...](#)



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Projet 3 : Pipeline script : Git-Gitlab-Jenkins-Tomcat

Étape 2- définir les étapes de notre job (stage : représente un ensemble d'étapes de votre processus (par exemple, la récupération des sources, la compilation...))

Allez dans le générateur de script Groovy puis dans « Sample Step », sélectionnez « stage: Stage », puis dans « Stage Name », mettez le nom du stage et appuyez sur « Generate Pipeline Script » et vous devriez avoir :

Steps

Sample Step

Stage Name

Generate Pipeline Script

```
stage('git clone') {
    // some block
}
```

1

Pipeline

Definition **Pipeline script**

Script

```
1 node {
2     stage('git clone') {
3
4     }
5 }
6
```

2

- Copiez puis collez cette ligne dans la partie « Script » de la rubrique « Pipeline ». Vous devriez avoir :



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Projet 3 : Pipeline script : Git-Gitlab-Jenkins-Tomcat

2-1 : Clone du repo Git

De nouveau, ouvrez le générateur depuis le lien « Pipeline Syntax » puis sélectionner le **Sample Step git:Git** et remplir les informations nécessaires

Steps

Sample Step

Repository URL

Branch

Credentials [Ajouter](#)

☒ Include in polling?

☒ Include in changelog?

[Generate Pipeline Script](#)

git credentialsId: 'jenkins', url: 'git@gitlab.com:bileli/projet_j2ee.git'

Copiez puis collez cette ligne dans la partie « Script » de la rubrique « Pipeline ». Vous devriez avoir :

Pipeline

Definition

Pipeline script

Script

```
1 node {  
2   stage('git clone') {  
3     git credentialsId: 'jenkins', url: 'git@gitlab.com:bileli/projet_j2ee.git'  
4   }  
5 }  
6
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Projet 3 : Pipeline script : Git-Gitlab-Jenkins-Tomcat

Étape 3- Package & Clean : Maintenant qu'on a cloné le projet, on peut commencer à packager notre application puis nettoyer notre workspace dont notamment les fichiers issus de la compilation : Dans le script Groovy, il faudra ajouter la ligne suivante : « **mvn clean install package** »

3.1- crée la deuxième Stage « Build »

3.2- Allez de nouveau dans le générateur puis dans « **Sample Step** », sélectionnez « **sh :Shell Script** ». vous devriez obtenir la ligne suivante :

```
sh 'mvn clean install package'
```

3.3- Copiez puis collez cette ligne dans la partie « Script » de la rubrique « Pipeline ». Vous devriez avoir :

Pipeline

Definition

Pipeline script

Script

```
1 node {
2   stage('git clone') {
3     git credentialsId: 'jenkins', url: 'git@gitlab.com:bileli/projet_j2ee.git'
4   }
5
6   stage('Build') {
7     sh 'mvn clean install package'
8   }
9 }
10
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Projet 3 : Pipeline script : Git-Gitlab-Jenkins-Tomcat

Étape 4- Déploiement

4.1- crée la troisième Stage « Deploy » et remplir les informations nécessaires

4.2- Allez de nouveau dans le générateur puis dans « **Sample Step** », sélectionnez « **deploy : Deploy war/ear to a container** » et remplir les informations nécessaires. vous devriez obtenir la ligne suivante :

Steps

Sample Step

Fichiers WAR/EAR

Context path

Containers

Tomcat 8.x Remote

Credentials

Tomcat URL

Deploy on failure ☒

deploy adapters: [tomcat8(credentialsId: 'tomcat', path: '', url: 'http://192.168.1.17:8080'), contextPath: null, war: '**/*.war']

4.3- Copiez puis collez cette ligne dans la partie « Script » de la rubrique « Pipeline ».

```
deploy adapters: [tomcat8(credentialsId: 'tomcat', path: '', url:
'http://192.168.1.17:8080'), contextPath: null, war: '**/*.war']
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Projet 3 : Pipeline script : Git-Gitlab-Jenkins-Tomcat

Au final, le script de notre Pipeline ressemblera à cela :

Pipeline

Definition

Pipeline script

Script

```
1 node {
2   stage('git clone') {
3     git credentialsId: 'jenkins', url: 'git@gitlab.com:bileli/projet_j2ee.git'
4   }
5
6   stage('Build') {
7     sh 'mvn clean install package'
8   }
9
10  stage('Deploy') {
11    deploy adapters: [tomcat8(credentialsId: 'tomcat', path: '', url: 'http://192.168.1.17:8080')], contextPath: null, war: '**/*.war'
12  }
13 }
14
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Projet 3 : Pipeline script : Git-Gitlab-Jenkins-Tomcat

Une fois que vous avez sauvegarder votre Pipeline, lancer un build, vous devriez avoir le résultat ci-dessous :

Dashboard
Pipeline_Deploy

Back to Dashboard

Status

Changes

Lancer un build

Configurer

Supprimer Pipeline

Full Stage View

Rename

Pipeline Syntax

Historique des builds
tendance

find

#2
8 févr. 2021 20:39

Pipeline Pipeline_Deploy

Recent Changes

Stage View

git clone	Build	Deploy
3s	4s	1s
3s	4s	1s

Average stage times:
(Average full run time: ~12s)

#2
Feb 08 21:39
No Changes

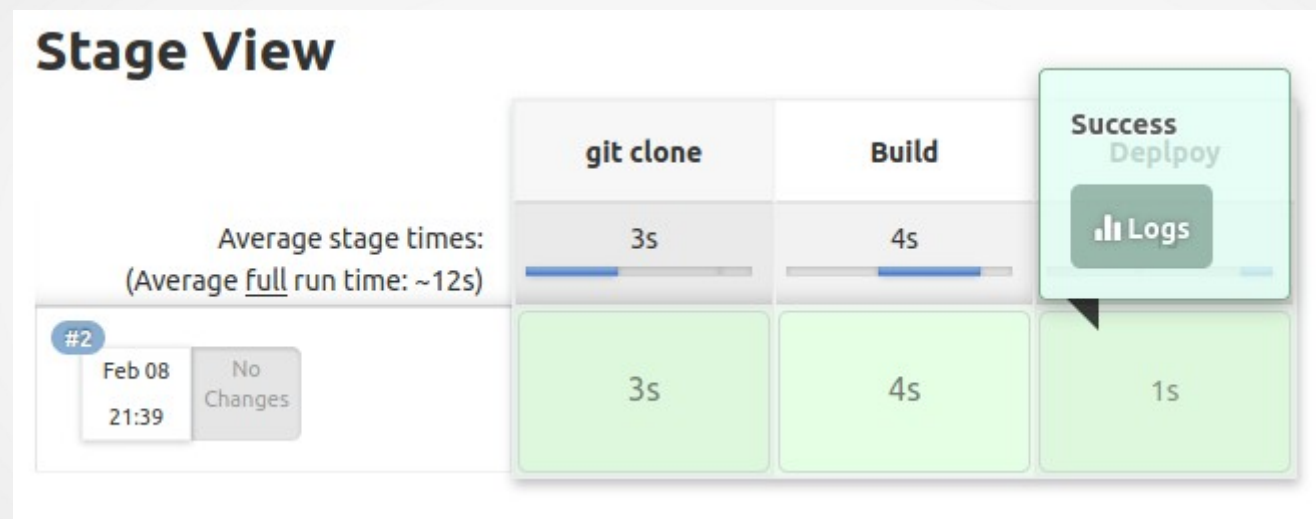
Liens permanents



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Projet 3 : Pipeline script : Git-Gitlab-Jenkins-Tomcat

Vous voyez bien que la partie « Stage View » ne contient pas d'erreur mais plutôt une vue d'ensemble des différentes étapes de notre build et de leur état d'avancement. Nous retrouvons bien les 3 étapes : **git clone**, **Build** et **deploy**. Toutes les étapes sont en vert, montrant que notre build a été en succès du début jusqu'à la fin. Cet affichage présente un autre avantage : si vous cliquez dans la partie verte d'une étape, vous devriez avoir un pop-up « Logs » qui apparaît :



Cliquez dessus, vous devriez avoir :



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Projet 3 : Pipeline script : Git-Gitlab-Jenkins-Tomcat

En effet, tous les logs qu'on avait dans Console Output s'affiche ici mais uniquement pour la partie concernant l'étape en question. Du coup, on n'avait plus besoin d'aller à chaque fois dans Console Output : on peut suivre l'état d'avancement de notre build depuis Stage View et en cas d'erreur, accéder aux logs pour déboguer

Stage View



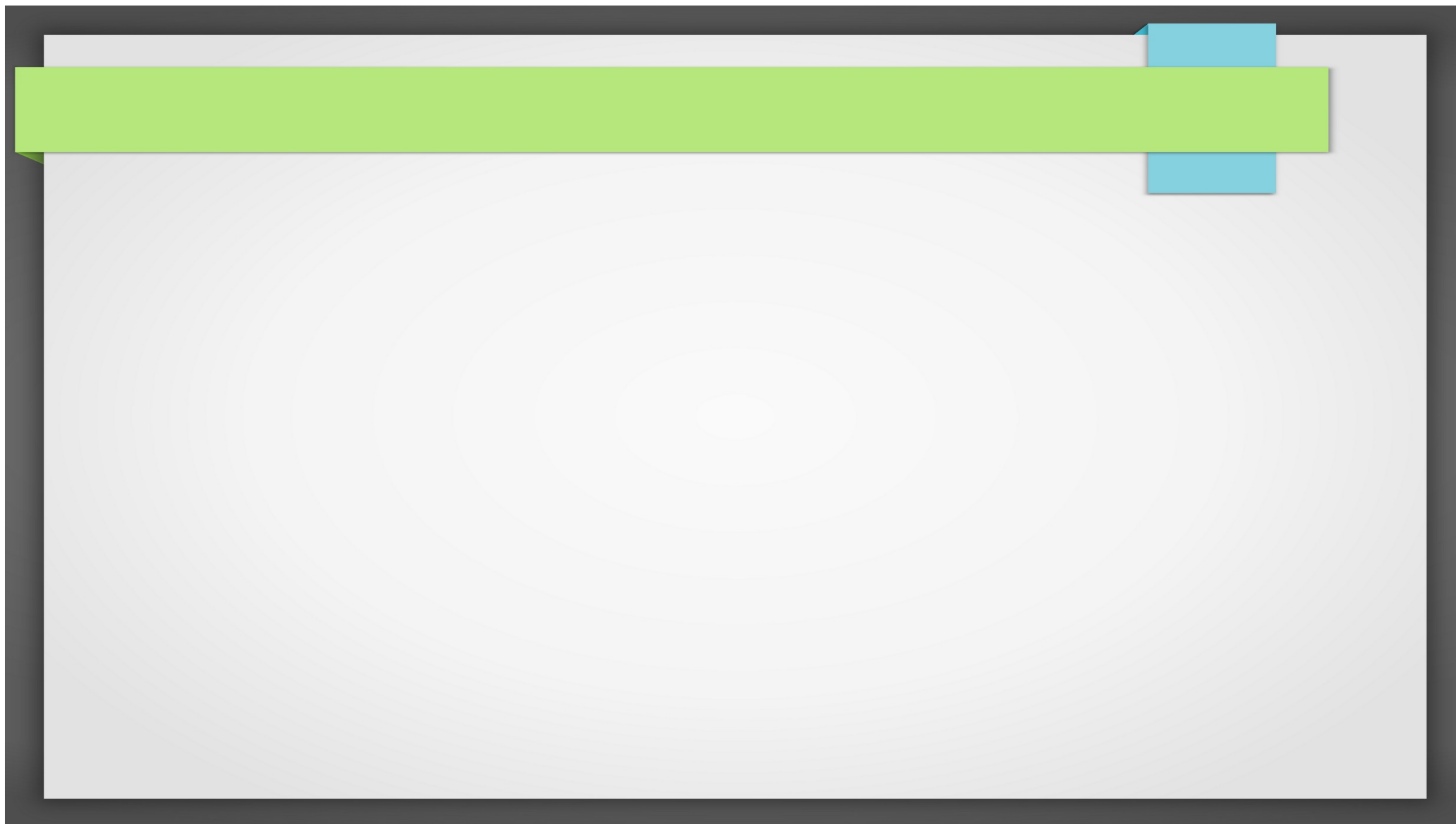
Stage Logs (Build)

Shell Script -- mvn clean install package (self time 3s)

```
such malformed projects.
[WARNING]
[WARNING] The project com.example.maven-project:maven-project:pom:1.0-SNAPSHOT uses prerequisites which is only intended for maven-plugin projects but not for non maven-plugin projects. For su
plugin. See https://maven.apache.org/enforcer/enforcer-rules/requireMavenVersion.html
[INFO] -----
[INFO] Reactor Build Order:
[INFO]
[INFO] Maven Project [pom]
[INFO] Server [jar]
[INFO] Webapp [war]
[INFO]
[INFO] -----< com.example.maven-project:maven-project >-----
[INFO] Building Maven Project 1.0-SNAPSHOT [1/3]
[INFO] -----[ pom ]-----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ maven-project ---
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ maven-project ---
[INFO] Installing /var/lib/jenkins/workspace/Pipeline_Deploy/pom.xml to /var/lib/jenkins/.m2/repository/com/example/maven-project/maven-project/1.0-SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155