

# Jenkins



**Bureau E204**

# Plan du cours

- Jenkins
- Installation de Jenkins
- Configuration de Jenkins
- Configuration d'un projet avec Jenkins
- Travail à faire

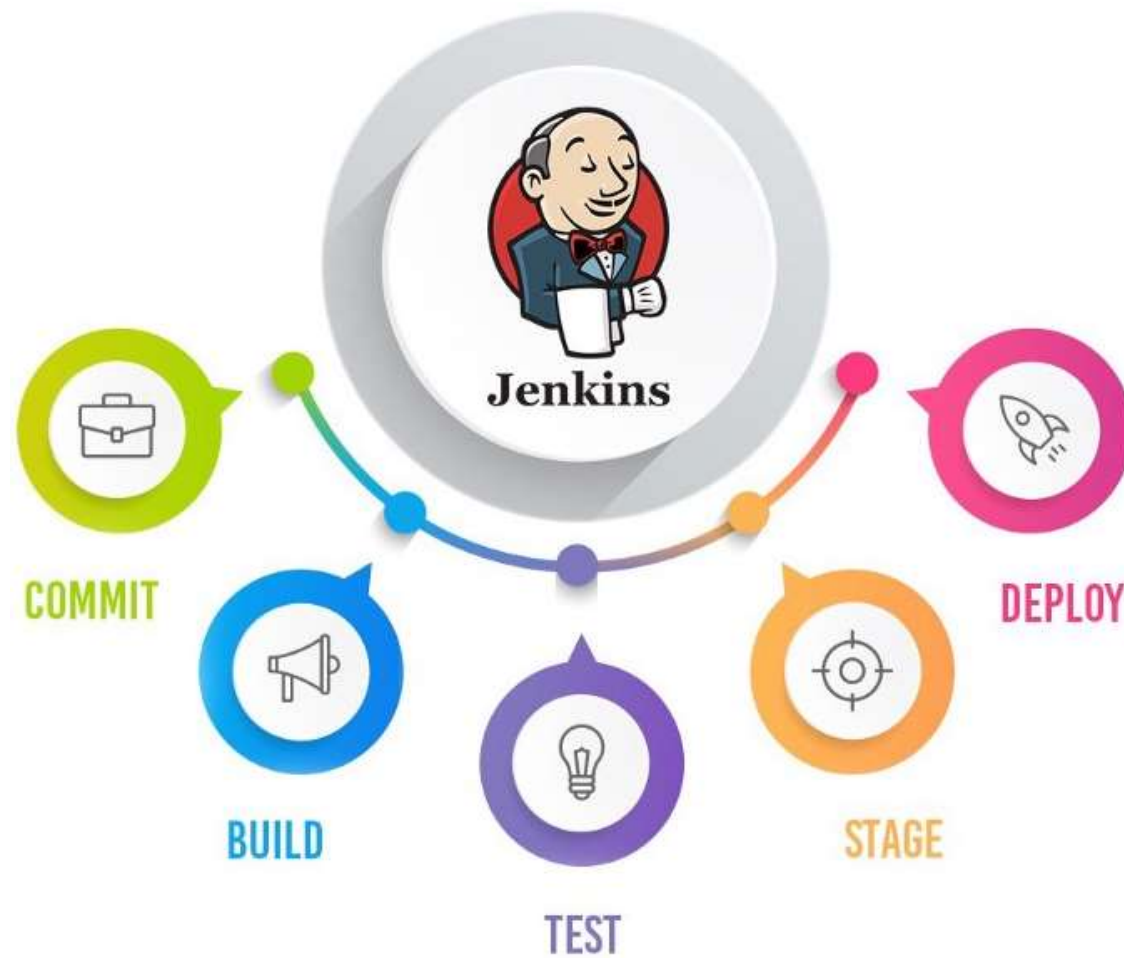
# Jenkins - Définition

- Jenkins est un serveur open source entièrement écrit en Java. Il permet d'exécuter (manuellement ou automatiquement) une série d'actions pour réaliser le processus d'intégration continue.
- Jenkins utilise un serveur web, comme Apache-Tomcat ou Jetty, pour offrir une interface web pour manipuler Jenkins.
- Il s'interface avec des systèmes de gestion de versions, tels que CVS ou Git ou Subversion SVN (nous les verrons dans le cours Git), et exécute des scripts écrit en Groovy ou Yaml par exemple.

# Jenkins - Définition

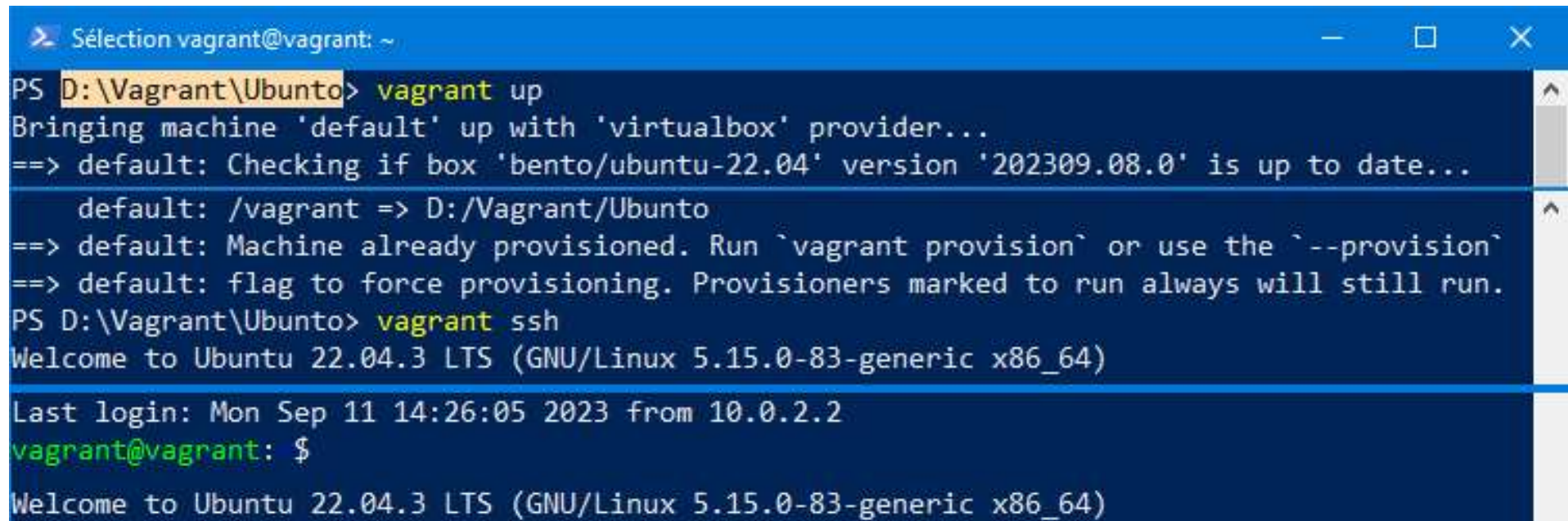
- Jenkins facilite l'intégration continue et la livraison continue dans les projets en automatisant les parties liées à la construction (build), aux tests et au déploiement.
- Cela permet aux développeurs de travailler en permanence sur l'amélioration du produit en intégrant des modifications au projet et en les testant plus rapidement.

# Jenkins - Définition



# Installation de OpenJDK 11

- Se placer (avec un CMD ou un PowerShell) au niveau du dossier de votre VM Ubuntu, vérifiez la configuration de Vagrantfile (surtout l'adresse ip), lancez votre VM (vagrant up), et lancer un shell (vagrant ssh). Voir cours 1 pour plus de détails :



```
Sélection vagrant@vagrant: ~
PS D:\Vagrant\Ubuntu> vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'bento/ubuntu-22.04' version '202309.08.0' is up to date...
    default: /vagrant => D:/Vagrant/Ubuntu
==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> default: flag to force provisioning. Provisioners marked to run always will still run.
PS D:\Vagrant\Ubuntu> vagrant ssh
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-83-generic x86_64)

Last login: Mon Sep 11 14:26:05 2023 from 10.0.2.2
vagrant@vagrant: $
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-83-generic x86_64)
```

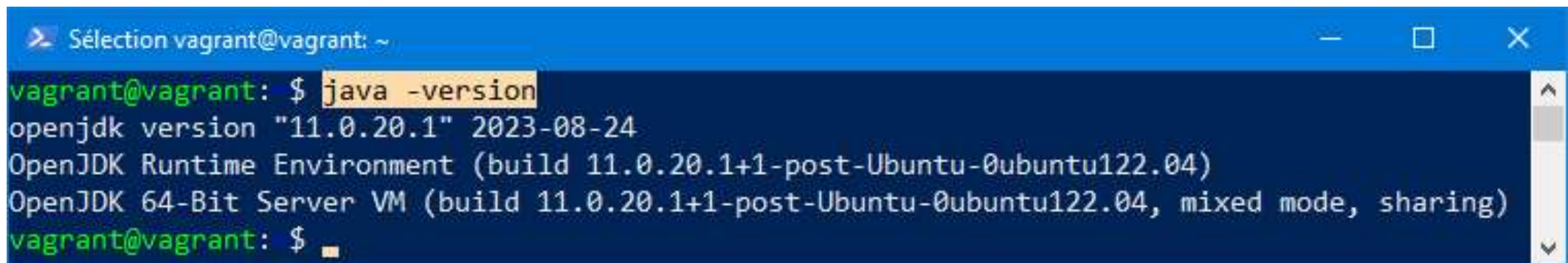
- Installer la OpenJDK (par défaut, Ubuntu installera la version 11) :

**sudo apt update**

**sudo apt install defaultjdk**

# Installation de OpenJDK 11

- Pour vérifier que le JDK 11 est bien installé, exécuter la commande suivante:

A terminal window with a blue title bar containing the text 'Sélection vagrant@vagrant: ~' and standard window controls. The terminal has a dark blue background with white text. The command 'java -version' is entered and highlighted in yellow. The output shows the OpenJDK version and build information.

```
vagrant@vagrant: $ java -version
openjdk version "11.0.20.1" 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
vagrant@vagrant: $
```



# Installation de OpenJDK 11

Configuration des variables d'environnement : Afin d'aider les applications basées sur Java à localiser correctement la machine virtuelle Java, vous devez définir deux variables d'environnement : "JAVA\_HOME" et "JRE\_HOME"

```
vagrant@vagrant: ~  
vagrant@vagrant: $ sudo cp /etc/profile /etc/profile_backup  
vagrant@vagrant: $ echo 'export JAVA_HOME=/usr/lib/jvm/default-java' | sudo tee -a /etc/profile  
export JAVA_HOME=/usr/lib/jvm/default-java  
vagrant@vagrant: $ echo 'export JRE_HOME=/usr/lib/jvm/default-java' | sudo tee -a /etc/profile  
export JRE_HOME=/usr/lib/jvm/default-java  
vagrant@vagrant: $ source /etc/profile  
vagrant@vagrant:~$
```

Attention : les lignes qui commencent par « export » sont les résultats de la commande echo, à ne pas mettre dans la commande echo donc.

**tee** : écrit le contenu du stdout vers un fichier

**source** : exécute le contenu d'un fichier



# Installation de OpenJDK 11

- Enfin, exécuter les commandes suivantes pour vérifier :

```
vagrant@vagrant:~$ echo $JAVA_HOME
/usr/lib/jvm/default-java
vagrant@vagrant:~$ echo $JRE_HOME
/usr/lib/jvm/default-java
vagrant@vagrant:~$
```

# Installation de OpenJDK 11

- Si votre système pointe sur la version 11, ignorez ce slide, sinon :

```
[vagrant@localhost ~]$ sudo java -version
openjdk version "1.8.0_342"
OpenJDK Runtime Environment (build 1.8.0_342-b07)
OpenJDK 64-Bit Server VM (build 25.342-b07 mixed mode)
[vagrant@localhost ~]$ sudo update-alternatives --config java

There are 2 programs which provide 'java'.

   Selection    Command
-----
        1      java-11-openjdk.x86_64 (/usr/lib/jvm/java-11-openjdk-11
*+ 2          java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-1.8.0-open

Enter to keep the current selection[+], or type selection number: 1
[vagrant@localhost ~]$ sudo java -version
openjdk version "11.0.16" 2022-07-19 LTS
OpenJDK Runtime Environment (Red_Hat-11.0.16.0.8-1.el7_9) (build 11.0.
OpenJDK 64-Bit Server VM (Red_Hat-11.0.16.0.8-1.el7_9) (build 11.0.16+
[vagrant@localhost ~]$
```

- Redémarrez votre VM (vagrant halt, vagrant up), si la modification n'a pas été prise en compte.

# Installation de Maven

- Pour installer Maven, vous devez lancer le terminal et exécuter les commandes suivantes:

```
vagrant@vagrant: ~  
vagrant@vagrant:~$ sudo apt install maven
```

- Pour vérifier que le Maven est bien installé :

```
vagrant@vagrant: ~  
vagrant@vagrant:~$ mvn -version  
Apache Maven 3.6.3  
Maven home: /usr/share/maven  
Java version: 11.0.20.1, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64  
Default locale: en_US, platform encoding: UTF-8  
OS name: "linux", version: "5.15.0-83-generic", arch: "amd64", family: "unix"  
vagrant@vagrant:~$
```

# Installation de Maven

- Pour définir la variable d'environnement Maven : "M2\_HOME"

```
vagrant@vagrant: ~  
vagrant@vagrant:~$ sudo cp /etc/profile /etc/profile_backup  
vagrant@vagrant:~$ echo 'export M2_HOME=/usr/share/maven' | sudo tee -a /etc/profile  
export M2_HOME=/usr/share/maven  
vagrant@vagrant:~$ echo $M2_HOME  
  
vagrant@vagrant:~$ source /etc/profile  
vagrant@vagrant:~$ echo $M2_HOME  
/usr/share/maven  
vagrant@vagrant:~$
```

# Installation de Git

- Pour pouvoir utiliser Git dans Jenkins, il faut installer Git sur votre VM (aucune configuration Git dans Jenkins ne sera nécessaire) :
- Git est déjà installé normalement mais vérifiez quand même

```
vagrant@vagrant: ~  
vagrant@vagrant:~$ sudo apt install git  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
git is already the newest version (1:2.34.1-1ubuntu1.10).  
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.  
vagrant@vagrant:~$ git --version  
git version 2.34.1  
vagrant@vagrant:~$
```

# Installation de Jenkins

- Pour installer Jenkins, vous devez exécuter les commandes suivantes:
- Commencer par installer **wget** pour pouvoir télécharger des fichiers :

```
vagrant@vagrant: ~  
vagrant@vagrant:~$ sudo apt install wget  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
wget is already the newest version (1.21.2-2ubuntu1).  
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.  
vagrant@vagrant:~$
```

- puis :

# Installation de Jenkins

```
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
```

```
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
```

```
sudo apt install ca-certificates
```

```
sudo apt-get update
```

```
sudo apt install jenkins
```



# Installation de Jenkins

- Pour lancer Jenkins (start), pour lancer jenkins automatiquement comme service au prochains démarrages (enable)
- Pour vérifier l'installation de Jenkins : status

```
vagrant@localhost:~  
[vagrant@localhost ~]$ sudo systemctl start jenkins.service  
[vagrant@localhost ~]$ sudo systemctl status jenkins.service  
● jenkins.service - Jenkins Continuous Integration Server  
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)  
   Active: active (running) since Sun 2022-09-18 15:34:44 UTC; 48s ago  
 Main PID: 4019 (java)  
   CGroup: /system.slice/jenkins.service  
           └─4019 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080  
  
Sep 18 15:33:59 localhost.localdomain jenkins[4019]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword  
Sep 18 15:33:59 localhost.localdomain jenkins[4019]: *****  
Sep 18 15:33:59 localhost.localdomain jenkins[4019]: *****  
Sep 18 15:33:59 localhost.localdomain jenkins[4019]: *****  
Sep 18 15:34:43 localhost.localdomain jenkins[4019]: 2022-09-18 15:34:43.967+0000 [id=29] INFO jenkins.InitReactorRun  
Sep 18 15:34:44 localhost.localdomain jenkins[4019]: 2022-09-18 15:34:44.104+0000 [id=22] INFO hudson.lifecycle.Lifec  
Sep 18 15:34:44 localhost.localdomain systemd[1]: Started Jenkins Continuous Integration Server.  
Sep 18 15:34:45 localhost.localdomain jenkins[4019]: 2022-09-18 15:34:45.140+0000 [id=46] INFO h.m.DownloadService$Do  
Sep 18 15:34:45 localhost.localdomain jenkins[4019]: 2022-09-18 15:34:45.155+0000 [id=46] INFO hudson.util.Retrier#st  
Sep 18 15:34:45 localhost.localdomain jenkins[4019]: 2022-09-18 15:34:45.161+0000 [id=46] INFO hudson.model.AsyncPeri  
Hint: Some lines were ellipsized, use -l to show in full.  
[vagrant@localhost ~]$ sudo systemctl enable jenkins.service
```

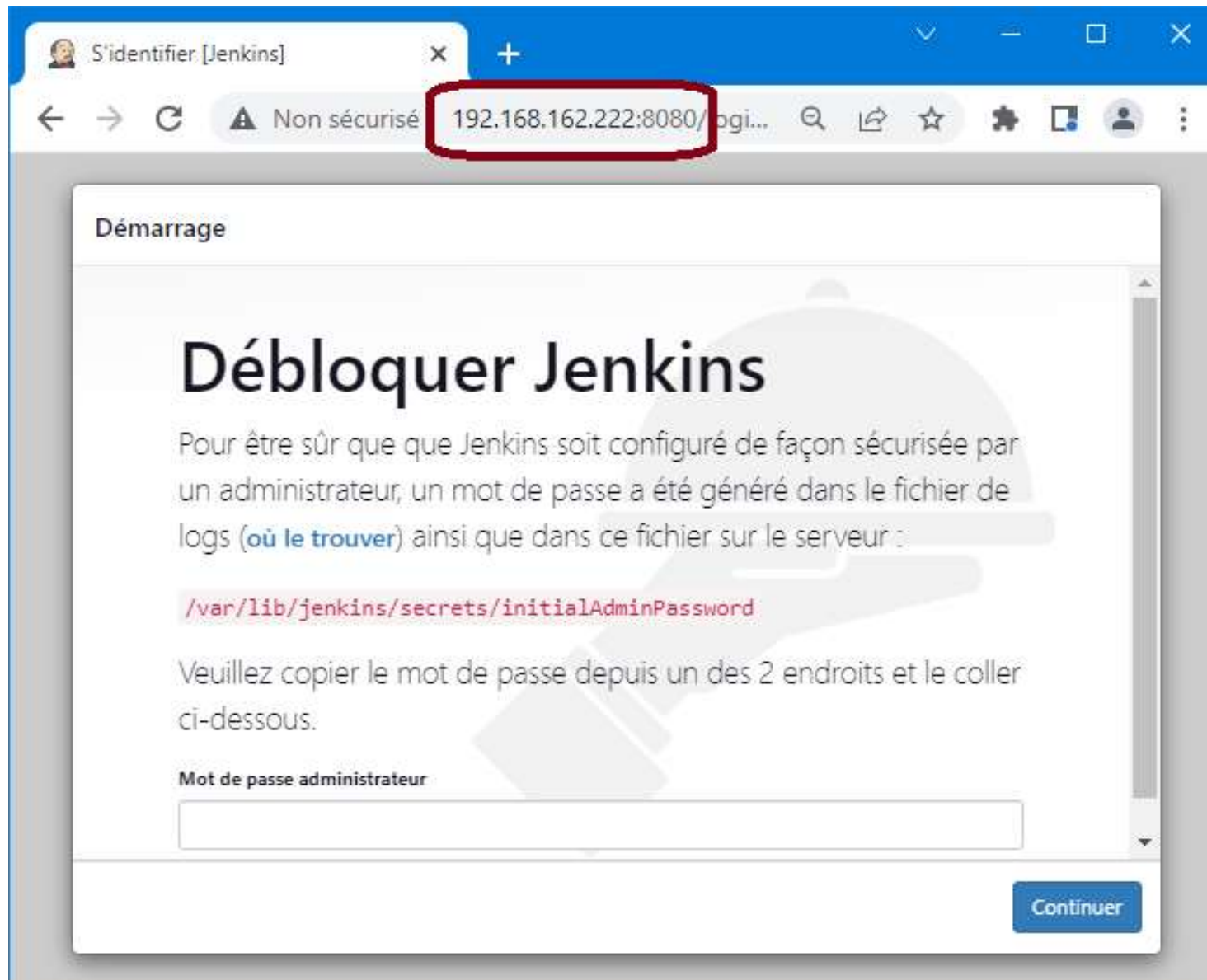
# Installation de Jenkins

- Pour accéder à Jenkins, vous devez récupérer l'adresse ip de la machine virtuelle à travers la commande:

```
[vagrant@localhost ~]$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc no...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    link/ether 52:54:00:4d:77:d3 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
        valid_lft 82043sec preferred_lft 82043sec
    inet6 fe80::5054:ff:fe4d:77d3/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    link/ether 08:00:27:85:53:c8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.162.222/24 brd 192.168.162.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe85:53c8/64 scope link
        valid_lft forever preferred_lft forever
```



# Installation de Jenkins



# Installation de Jenkins

- Pour la première fois, il faut débloquent Jenkins en tapant le mot de passe qui est stocké dans le fichier de log mentionné dans la fenêtre ci-dessus (utiliser la commande cat pour afficher le mot de passe) :

```
[vagrant@localhost ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
37a196e129c64114ae89f0e0b9df83e6
```

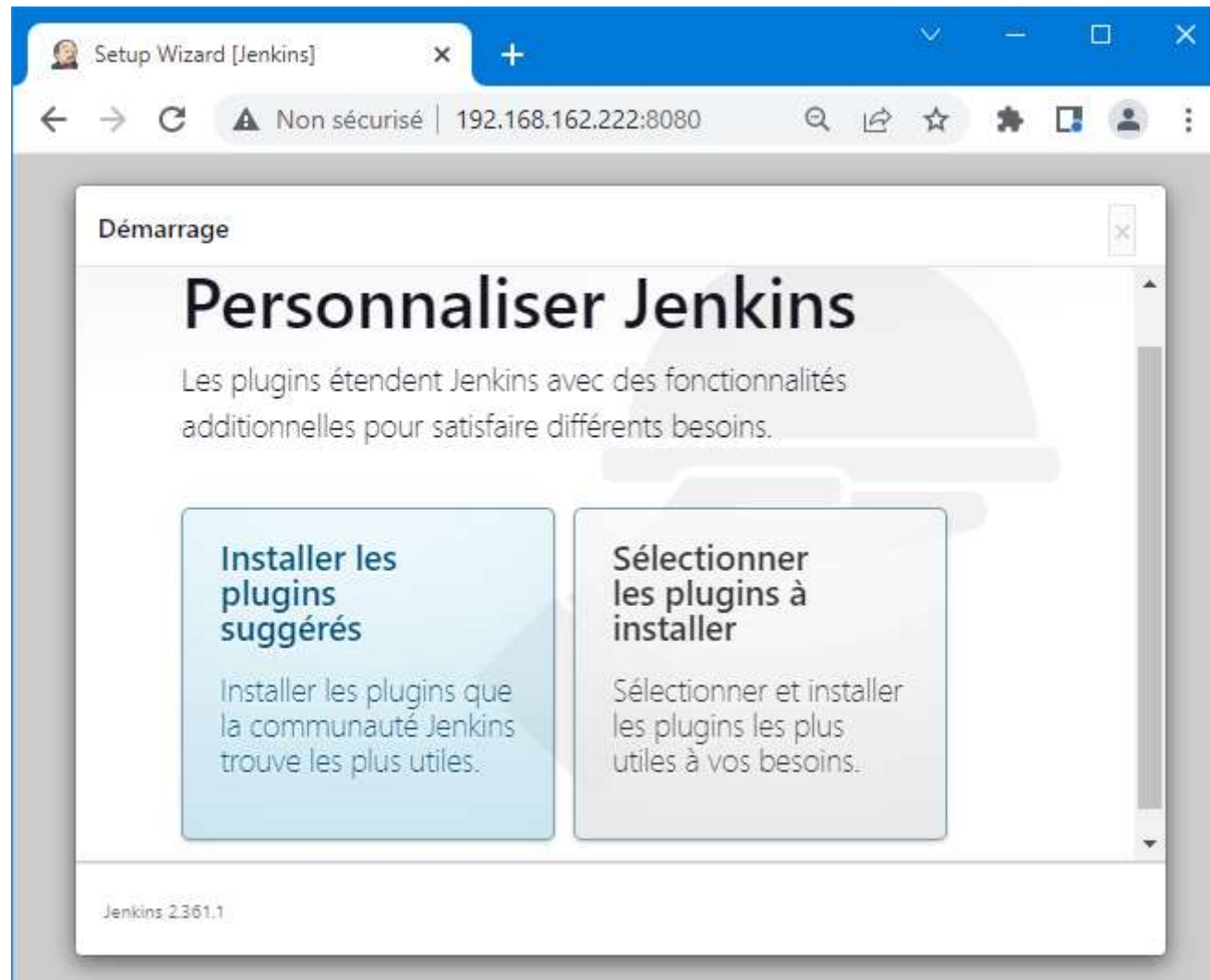
Mot de passe administrateur

.....|

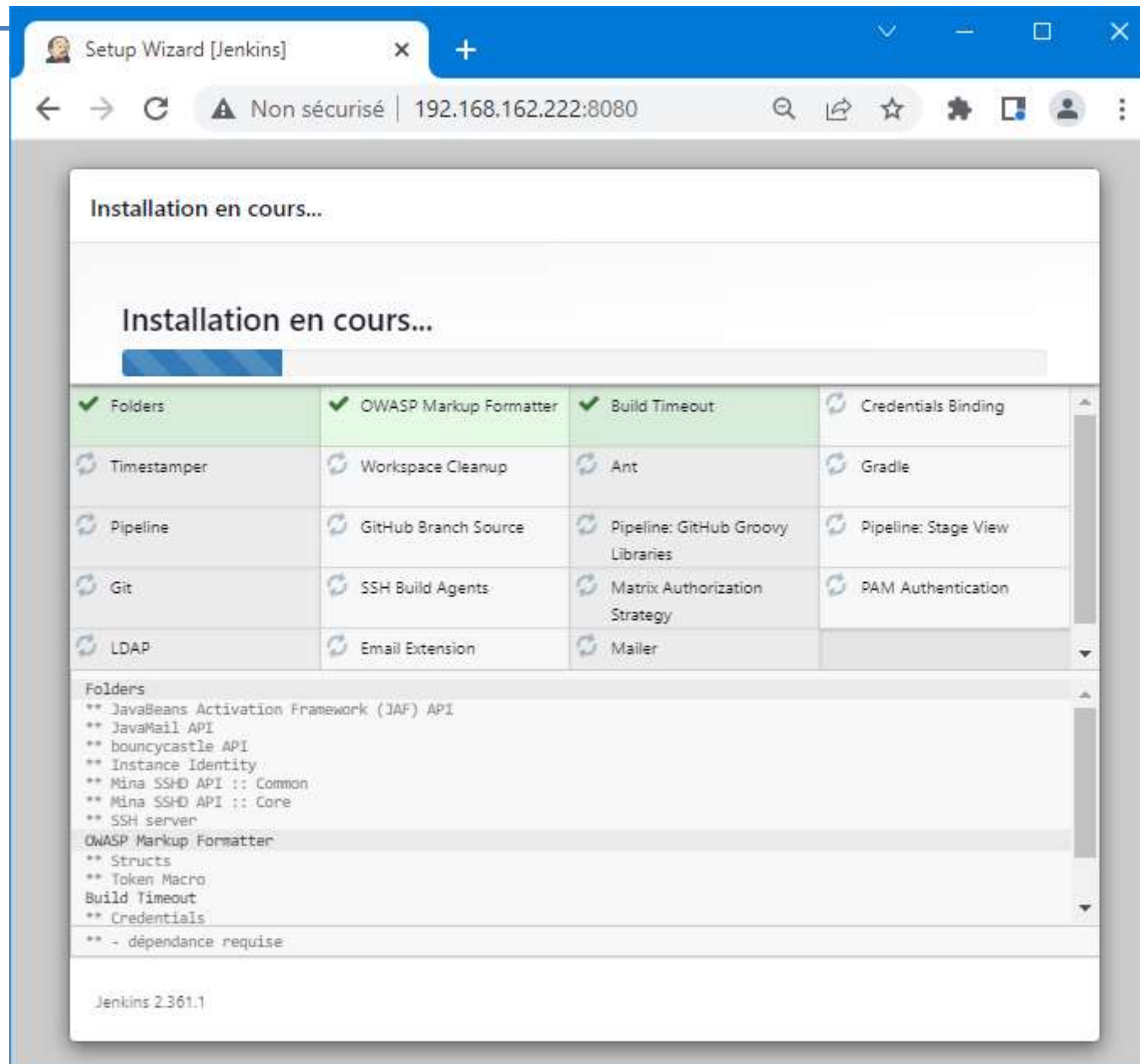
- Installer les plugins suggérés.

# Installation de Jenkins

- Installer les plugins suggérés.



# Installation de Jenkins



# Installation de Jenkins

- Vous pouvez ne créer aucun nouvel utilisateur et continuer à utiliser le user « **admin** » :

## Créer le 1er utilisateur Administrateur

Nom d'utilisateur:

Mot de passe:

Confirmation du mot de passe:

Nom complet:

Adresse courriel:

---

1.2

[Continuer en tant qu'Administrateur](#) [Sauver et continuer](#)



# Installation de Jenkins

- Pour personnaliser le numéro de port de Jenkins, il suffit de le modifier à travers cette fenêtre (vous pouvez garder le port 8080).

Démarrage

## Configuration de l'instance

URL de Jenkins :

L'URL de Jenkins est utilisée pour fournir l'URL de base pour les liens absolus vers les diverses ressources Jenkins. Cela signifie que cette valeur est nécessaire pour le bon fonctionnement de nombreuses fonctionnalités de Jenkins, notamment les notifications par mail, les mises à jour des statuts des pull requests, et la variable d'environnement BUILD\_URL fournie pour les étapes de build.

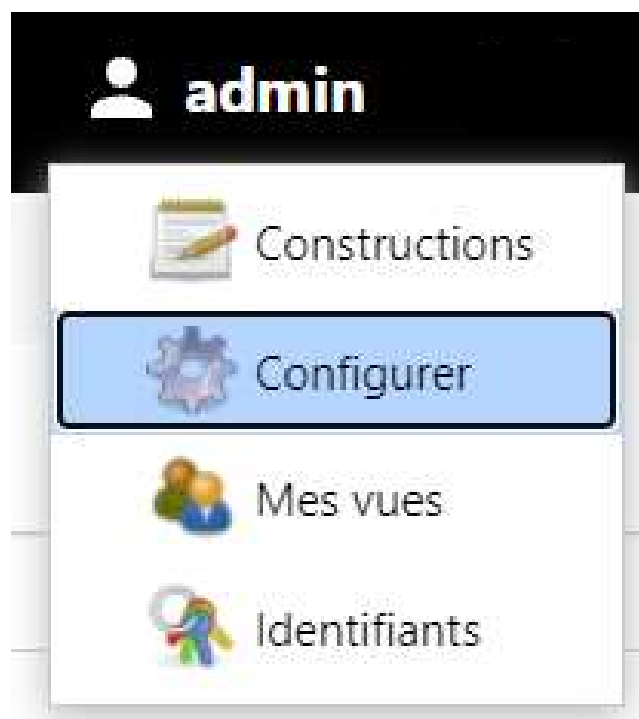
La valeur par défaut affichée **n'est pas encore sauvegardée** et est générée à partir de la requête actuelle, lorsque c'est possible. Il est fortement recommandé d'utiliser comme valeur l'URL qui est censée être utilisée par les utilisateurs. Cela évitera des confusions lors du partage ou de la visualisation de liens.

Jenkins 2.303.2

[Passer cette étape et terminer](#) [Sauver et terminer](#)

# Configuration de Jenkins

- Changer le mot de passe de « **admin** » à « **jenkins** » par exemple :



## Mot de passe

Mot de passe:

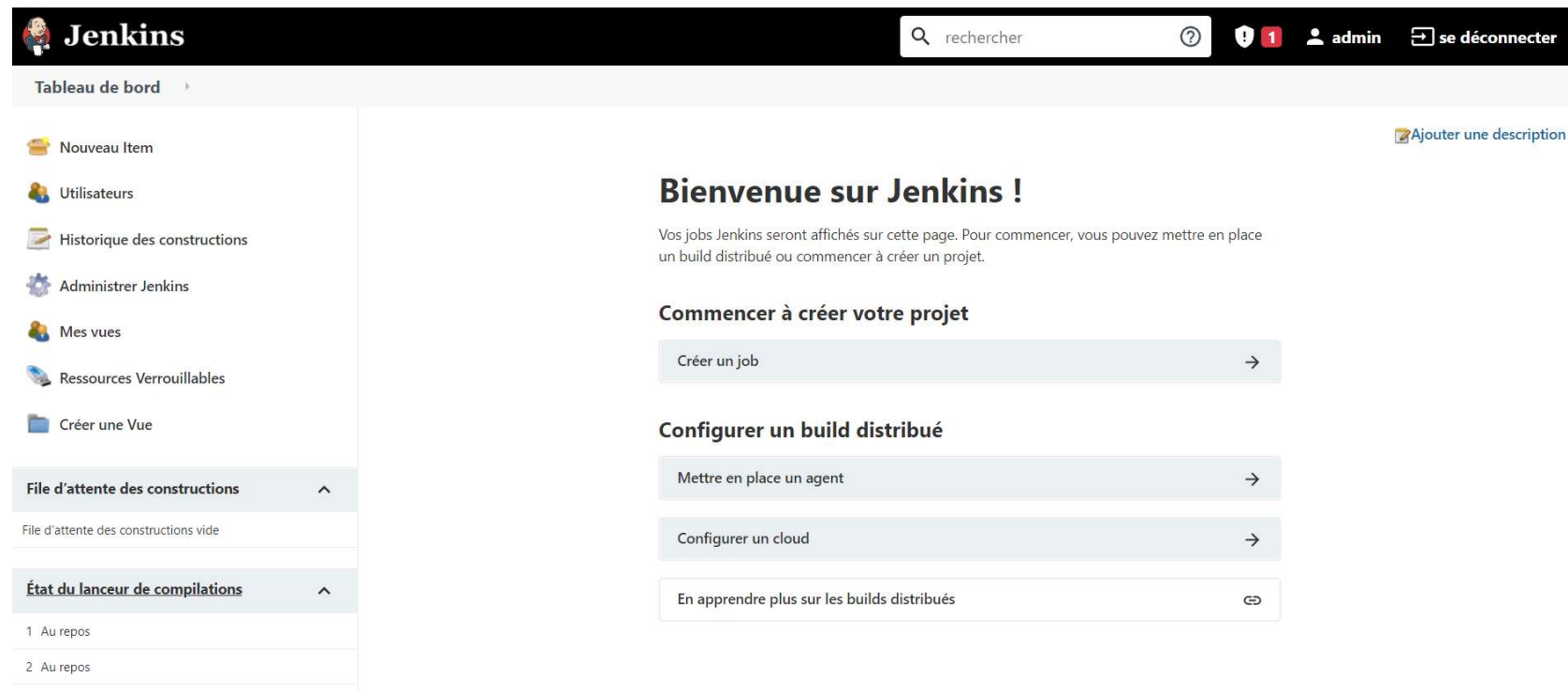
Confirmer le mot de passe:

Sauvegarder

Appliquer

# Configuration de Jenkins

- Jenkins dispose d'une interface web très simple et pratique
- On peut accéder directement vers tous les configurations possibles.
- On trouve toutes les informations à propos de tous les jobs.



The screenshot displays the Jenkins web interface. At the top is a black header with the Jenkins logo, a search bar labeled 'rechercher', and user information 'admin' with a 'se déconnecter' link. Below the header is a light gray sidebar on the left containing a 'Tableau de bord' link and a list of navigation items: 'Nouveau Item', 'Utilisateurs', 'Historique des constructions', 'Administrer Jenkins', 'Mes vues', 'Ressources Verrouillables', and 'Créer une Vue'. The main content area on the right has a 'Bienvenue sur Jenkins !' heading, a welcome message, and a 'Commencer à créer votre projet' section with a 'Créer un job' button. Below this is a 'Configurer un build distribué' section with buttons for 'Mettre en place un agent', 'Configurer un cloud', and a link to 'En apprendre plus sur les builds distribués'. A 'Ajouter une description' link is also visible in the top right of the main area.

# Configuration de Jenkins - Plugins

- Pour installer des plugins, il suffit d'accéder à la fenêtre « Gestion des plugins »

The screenshot shows the Jenkins 'Manage Plugins' interface. The top navigation bar includes the Jenkins logo, a search bar, and user information. The left sidebar contains links to the dashboard and plugin management. The main content area displays a list of available plugins, with a search filter applied. The 'SonarQube Scanner' plugin is selected, and its details are shown, including a warning about its version and a security notice. The 'Sonar Quality Gates' and 'Quality Gates' plugins are also listed with similar warnings and security notices.

Install	Name	Version	Released
<input type="checkbox"/>	<b>SonarQube Scanner</b> External Site/Tool Integrations Build Reports This plugin allows an easy integration of <a href="#">SonarQube</a> , the open source platform for Continuous Inspection of code quality.	2.13.1	5 mo. 22 j ago
<input type="checkbox"/>	<b>Sonar Quality Gates</b> Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed") Warning: This plugin version may not be safe to use. Please review the following security notices: <ul style="list-style-type: none"><li>Credentials transmitted in plain text</li></ul>	1.3.1	3 an. 2 mo. ago
<input type="checkbox"/>	<b>Quality Gates</b> Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed") Warning: This plugin version may not be safe to use. Please review the following security notices: <ul style="list-style-type: none"><li>Credentials transmitted in plain text</li></ul>	2.5	5 an. 5 mo. ago

# Configuration de Jenkins - Plugins

- Pour créer notre chaine d'intégration continue, on va installer les plugin suivant dans Jenkins (Installer ces plugins sans redémarrer, puis redémarrer Jenkins à la fin) :
  - ✓ Git plugin (normalement déjà installé, mais vérifier)
  - ✓ Maven Integration
  - ✓ Nexus Platform
  - ✓ Sonargraph Integration
  - ✓ SonarQube Scanner

# Configuration de Jenkins - Plugins

## Installation/Mise à jour des Plugins

### Préparation

- Vérification de la connexion à internet
- Vérification de la connexion à jenkins-ci.org

Nexus Platform	⋯ En cours
Common API for Blue Ocean	⋯ En cours
REST API for Blue Ocean	⋯ En cours
Pub-Sub "light" Bus	⋯ En cours
Pipeline SCM API for Blue Ocean	⋯ En cours
Sonargraph Integration	⋯ En cours
SonarQube Scanner	⋯ En cours

## Installation/Mise à jour des Plugins

### Préparation

- Vérification de la connexion à internet
- Vérification de la connexion à jenkins-ci.org
- Succès

Nexus Platform	! Downloaded Successfully. Will be activated during the next boot
Common API for Blue Ocean	! Downloaded Successfully. Will be activated during the next boot
REST API for Blue Ocean	! Downloaded Successfully. Will be activated during the next boot
Pub-Sub "light" Bus	! Downloaded Successfully. Will be activated during the next boot
Pipeline SCM API for Blue Ocean	! Downloaded Successfully. Will be activated during the next boot
Sonargraph Integration	! Downloaded Successfully. Will be activated during the next boot
SonarQube Scanner	! Downloaded Successfully. Will be activated during the next boot

# Configuration de Jenkins - Outils

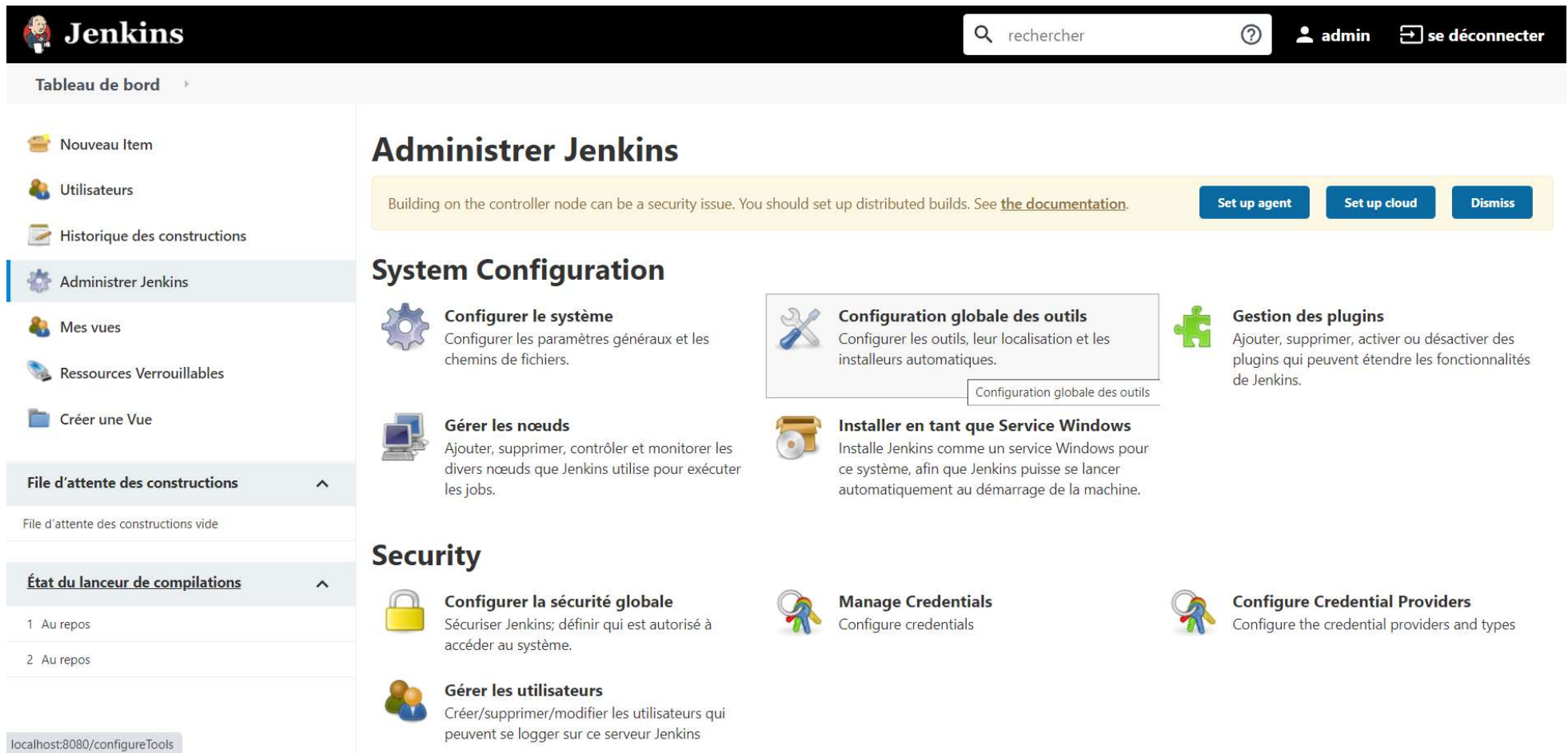
- Redémarrer Jenkins suite à l'installation des plugins :

```
[vagrant@localhost ~]$ sudo systemctl restart jenkins.service
```



# Configuration de Jenkins - Outils

- Pour configurer les outils, il faut accéder à la fenêtre « Configuration global des outils »



The screenshot shows the Jenkins web interface. At the top is a black header with the Jenkins logo, a search bar, and user information (admin, se déconnecter). Below the header is a sidebar with navigation links: Tableau de bord, Nouveau Item, Utilisateurs, Historique des constructions, Administrer Jenkins (selected), Mes vues, Ressources Verrouillables, and Créer une Vue. The main content area is titled 'Administrer Jenkins' and contains a warning about distributed builds. Below this is the 'System Configuration' section, which includes links to 'Configurer le système', 'Gérer les nœuds', 'Configuration globale des outils' (highlighted with a tooltip), 'Installer en tant que Service Windows', and 'Gestion des plugins'. The 'Security' section includes links to 'Configurer la sécurité globale', 'Manage Credentials', 'Configure Credential Providers', and 'Gérer les utilisateurs'. At the bottom left, there are sections for 'File d'attente des constructions' and 'État du lanceur de compilations'.

**Jenkins**

rechercher admin se déconnecter

Tableau de bord

- Nouveau Item
- Utilisateurs
- Historique des constructions
- Administrer Jenkins**
- Mes vues
- Ressources Verrouillables
- Créer une Vue

**Administrer Jenkins**

Building on the controller node can be a security issue. You should set up distributed builds. See [the documentation](#). [Set up agent](#) [Set up cloud](#) [Dismiss](#)

**System Configuration**

- Configurer le système**  
Configurer les paramètres généraux et les chemins de fichiers.
- Gérer les nœuds**  
Ajouter, supprimer, contrôler et monitorer les divers nœuds que Jenkins utilise pour exécuter les jobs.
- Configuration globale des outils**  
Configurer les outils, leur localisation et les installeurs automatiques.  
Configuration globale des outils
- Installer en tant que Service Windows**  
Installe Jenkins comme un service Windows pour ce système, afin que Jenkins puisse se lancer automatiquement au démarrage de la machine.
- Gestion des plugins**  
Ajouter, supprimer, activer ou désactiver des plugins qui peuvent étendre les fonctionnalités de Jenkins.

**Security**

- Configurer la sécurité globale**  
Sécuriser Jenkins; définir qui est autorisé à accéder au système.
- Manage Credentials**  
Configure credentials
- Configure Credential Providers**  
Configure the credential providers and types
- Gérer les utilisateurs**  
Créer/supprimer/modifier les utilisateurs qui peuvent se logger sur ce serveur Jenkins

File d'attente des constructions


File d'attente des constructions vide

**État du lanceur de compilations**

- 1 Au repos
- 2 Au repos


localhost:8080/configureTools


# Configuration de Jenkins - JDK


 **Jenkins**

recl

Tableau de bord ▸ Configuration globale des outils

 Retour au tableau de bord

 Administrer Jenkins

 **Configuration globale des outils**

**Configuration Maven**

Fournisseur de réglages par défaut

Utiliser les réglages Maven par défaut

Fournisseur de réglages globaux par défaut

Utiliser les réglages globaux Maven par défaut

**JDK**

Installations JDK

Ajouter JDK

Liste des installations JDK sur ce système

# Configuration de Jenkins - JDK

## Installations JDK

Ajouter JDK

JDK

Nom

ERROR

☒ Install automatically

➡ Décocher ce choix

?

Install Oracle Java SE Development Kit from the website

?

Version

Java SE Development Kit 9.0.4

☐ J'approuve l'accord de licence Java SE Development Kit

ERROR

Oracle Java SE 11+ is not available for business, commercial or production use without a commercial license.  
Public updates for Oracle Java SE 8 released after January 2019 will not be available for business, commercial or production use without a commercial license.  
[Oracle Java SE Licensing FAQ](#)

Supprimer un installateur

Ajouter un installateur

Supprimer JDK

Enregistrer

Appliquer

# Configuration de Jenkins - JDK

- Pointer sur le JDK déjà installé sur votre VM et sauvegarder :

## JDK

### Installations JDK

Liste des installations JDK sur ce système

Ajouter JDK

≡

JDK

×

Nom

JAVA\_HOME

JAVA\_HOME

/usr/lib/jvm/jre-11-openjdk

☐

Install automatically ?

# Configuration de Jenkins - Maven

## Maven

### Installations Maven

Ajouter Maven

Liste des installations Maven sur ce système



## Maven

### Installations Maven

Ajouter Maven

Maven

Nom

 Required

☒ Install automatically



Décocher ce choix



 Install from Apache

Version

3.8.3

Ajouter un installateur

Supprimer un installateur

Supprimer Maven

# Configuration de Jenkins - Maven

- Pointer sur le Maven déjà installé sur votre VM et sauvegarder :

## Maven

### Installations Maven

Liste des installations Maven sur ce système

Ajouter Maven

≡

Maven

×

Nom

M2\_HOME

MAVEN\_HOME

/usr/share/maven

☐

Install automatically ?

# Configuration d'un projet avec Jenkins - Job

- Les jobs ou tâches représentent le cœur du processus de « build » dans Jenkins.
- Le projet dans Jenkins est représenté par un job qui est composé de plusieurs processus du build.
- Tout projet sous Jenkins passe par les 3 étapes suivantes:
  - ✓ Création du job
  - ✓ Configuration du job (configuration des étapes du build)
  - ✓ Lancement du build
- Tous processus (ou étapes) du build dans les jobs sont manipulés directement à travers des plugins.



# Configuration d'un projet avec Jenkins - Job

Les Jobs qui permettent de compiler (build) un projet est les suivantes:

- 1/ Récupération du projet
- 2/ Compilation
- 3/ Lancement des tests automatiques (statiques et unitaires).
- 4/ Préparation de la version à distribuer.
- 5/ Mise en place de la version à distribuer (Nexus).

# Configuration d'un projet avec Jenkins - Job

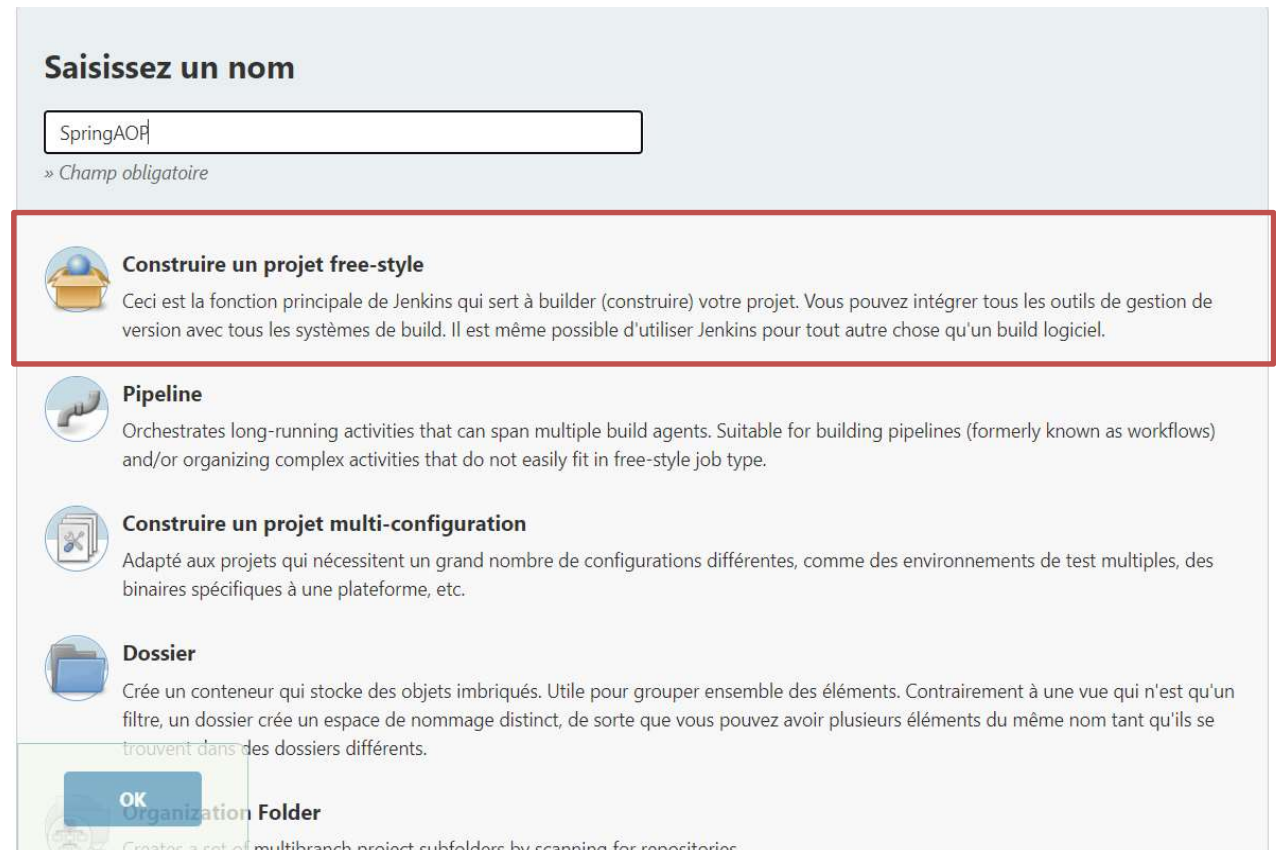
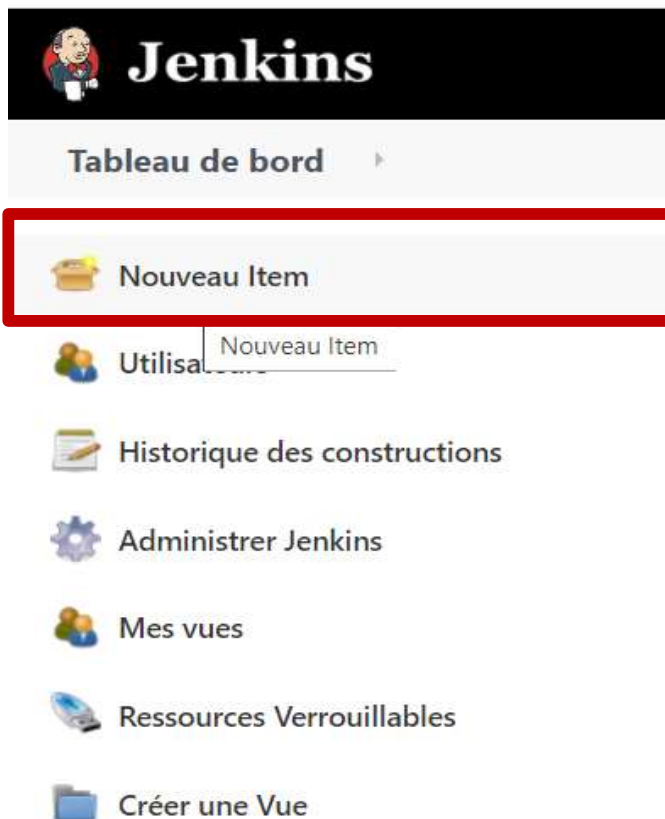
Deux méthodes de configuration sont possibles pour mettre en place un processus de compilation d'un projet :

- La méthode *freestyle* où la configuration du projet se réalise à travers un formulaire à remplir.
- L'implémentation d'un *pipeline*, qui définit la configuration d'un projet grâce à des scripts (basé sur Groovy). Cette méthode offre aussi les avantages de permettre de paralléliser les étapes du projet et offre une meilleure interface pour lire les logs.

# Configuration d'un projet avec Jenkins - FreeStyle

- Pour montrer le fonctionnement de Jenkins, nous allons utiliser un projet spring boot.

Au début, il faut créer un « projet Jenkins » de type freestyle



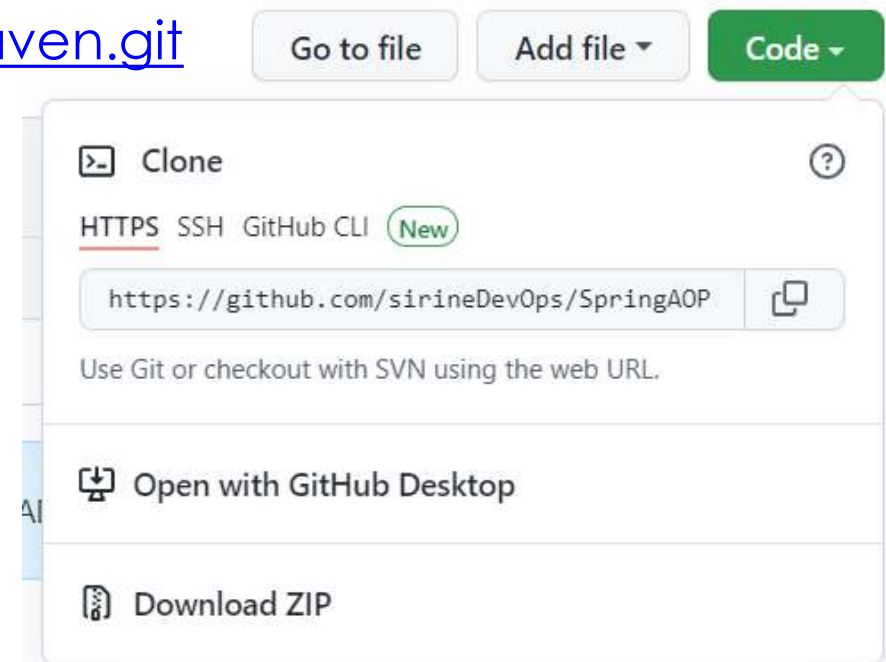
# Configuration d'un projet avec Jenkins - FreeStyle

- Après la création du projet, configurer le. La première partie de la page de configuration indique les informations générales du projet.

The screenshot shows the Jenkins web interface. At the top, there's a black header with the Jenkins logo, a search bar labeled 'rechercher', and user information 'admin' with a 'se déconnecter' link. Below the header, a breadcrumb trail shows 'Tableau de bord' and 'SpringAOP'. The main content area is titled 'General' and contains several tabs: 'Gestion de code source', 'Ce qui déclenche le build', 'Environnements de Build', 'Build', and 'Actions à la suite du build'. The 'General' tab is active, showing a 'Description' text area, a '[Plain text] Prévisualisation' link, and a list of checkboxes for build options: 'Ce build a des paramètres', 'GitHub project', 'This build requires lockable resources', 'Supprimer les anciens builds', 'Throttle builds', 'Désactiver le projet', and 'Exécuter des builds simultanément si nécessaire'. Each checkbox has a help icon. Below these is an 'Avancé...' button. The 'Gestion de code source' section shows 'Aucune' selected, with a 'Git' option partially visible. At the bottom, there are 'Sauver' and 'Apply' buttons.

# Configuration d'un projet avec Jenkins - FreeStyle

- Récupération du code de Git : Utiliser des repo Git publics ou ajouter des clés SSH ou définir des noms d'utilisateur et des mots de passe dans la partie « Identifiants » pour que Jenkins soit capable de récupérer le code de Git.
- Récupérer l'URL de votre projet de Git, exemple que vous pouvez utiliser pour tester : <https://github.com/mhassini/avec-maven.git>



# Configuration d'un projet avec Jenkins - FreeStyle

☒ GitHub project

Project url ?

`https://github.com/sirineDevOps/SpringAOP`

---

General **Gestion de code source** Ce qui déclenche le build Environnements de Build Build Actions à la suite du build

**Gestion de code source**

☐ Aucune ?

☒ Git ?

Repositories ?

Repository URL ?

`https://github.com/sirineDevOps/SpringAOP.git`

Credentials ?

`sirine.naifer@esprit.tn/*****` Ajouter

Avancé...

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') X ?

`/*main`

Add Branch

# Configuration d'un projet avec Jenkins - FreeStyle

Il faut choisir l'action qui va lancer la chaine de l'intégration continue périodiquement (process qui se lance chaque minute par exemple ou qui vérifie si une nouvelle version est poussée sur Git ...)

## Ce qui déclenche le build

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

☐ Construire après le build sur d'autres projets ?

☒ Construire périodiquement ?

Planning

?

\*\*\*\*\*

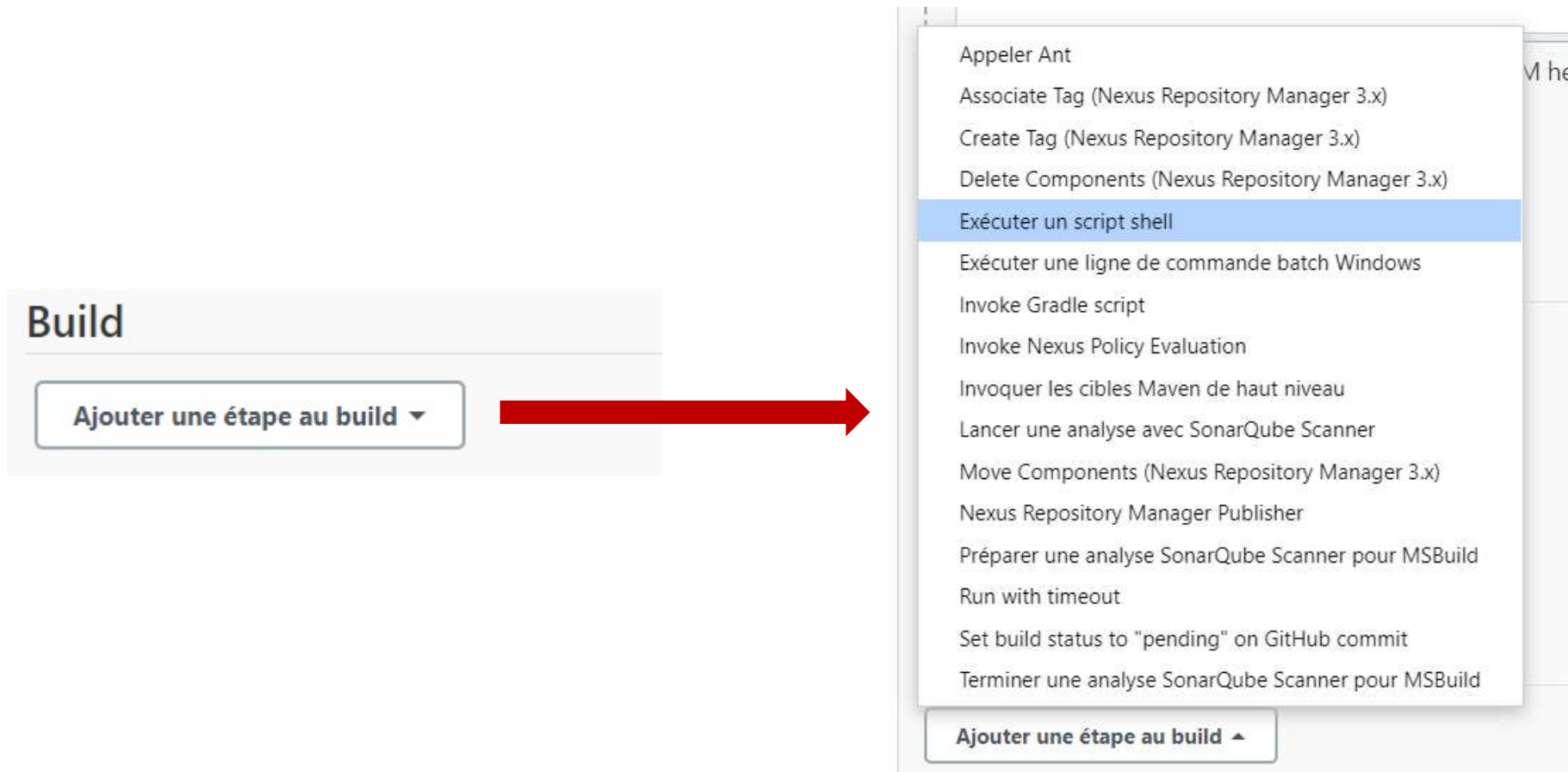


Voulez-vous vraiment dire "chaque minute" avec l'expression "\*\*\*\*\*"? Peut-être vouliez-vous dire "H\*\*\*\*\*"?



# Configuration d'un projet avec Jenkins - FreeStyle

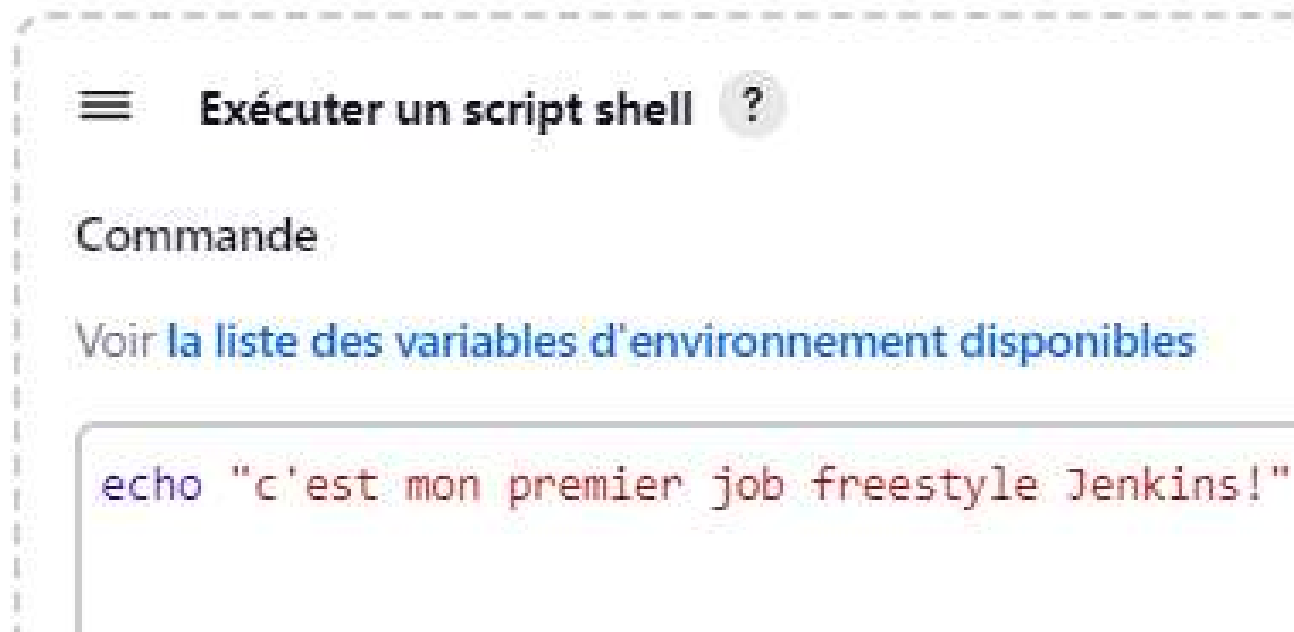
Pour les autres taches (2, 3, 4 et 5), on peut les configurer à travers la partie Build.



# Configuration d'un projet avec Jenkins - FreeStyle

**Exemple Simple 01:** Ecrire un message simple en affichant la date système. Sauvegarder et attendre une minute (le job sera lancé automatiquement chaque minute) :

## Build Steps



# Configuration d'un projet avec Jenkins - FreeStyle

**Exemple Simple 02:** Vérifie l'installation de maven.

## Build Steps

≡ Exécuter un script shell ?

Commande

Voir [la liste des variables d'environnement disponibles](#)

```
mvn -version
```

# Configuration d'un projet avec Jenkins - FreeStyle

- Sauvegarder la configuration et lancer le build.


The screenshot shows the Jenkins web interface for a project named 'SpringAOP'. The top navigation bar includes the Jenkins logo, a search bar with the text 'rechercher', and user information 'admin' with a 'se déconnecter' link. Below the navigation bar, the breadcrumb 'Tableau de bord > SpringAOP' is visible. The left sidebar contains a list of actions: 'Retour au tableau de bord', 'État' (highlighted with a blue bar), 'Modifications', 'Répertoire de travail', 'Lancer un build' (highlighted with a red rectangle), 'Configurer', 'Supprimer Projet', 'Log du dernier accès à Git', 'GitHub', 'SonarQube', and 'Rename'. The main content area is titled 'Projet SpringAOP' and includes a link to 'Ajouter une description' and a 'Désactiver le projet' button. Below this, there are three links: 'SonarQube', 'Espace de travail', and 'Changements récents'. The bottom section is titled 'Liens permanents' and includes a 'Historique des builds' tab with a 'tendance' dropdown and a search input field containing the text 'find'.

# Configuration d'un projet avec Jenkins - FreeStyle

- Si le build a échoué, l'icone  s'affiche avant le nom du projet au niveau du dashboard.



# Configuration d'un projet avec Jenkins - FreeStyle

- Si le build a réussi, l'icone  s'affiche.



The screenshot shows the Jenkins web interface for a build named 'Spring AOP' with ID '#8'. The build status is 'Succeeded', indicated by a green checkmark icon. The console output is visible, showing the progress of downloading and uploading Maven metadata files. The output includes the following information:

```
Ignore 498 Ko. Log complet
Uploading: http://localhost:8081/repository/maven-releases/tn/esprit/spring/tp1-spring-mvc-rest/5.0/tp1-spring-mvc-rest-5.0.pom
Progress (1): 2.0/3.2 kB
Progress (1): 3.2 kB

Uploaded: http://localhost:8081/repository/maven-releases/tn/esprit/spring/tp1-spring-mvc-rest/5.0/tp1-spring-mvc-rest-5.0.pom (3.2 kB at 35 kB/s)
Downloading: http://localhost:8081/repository/maven-releases/tn/esprit/spring/tp1-spring-mvc-rest/maven-metadata.xml
Progress (1): 340 B

Downloaded: http://localhost:8081/repository/maven-releases/tn/esprit/spring/tp1-spring-mvc-rest/maven-metadata.xml (340 B at 3.4 kB/s)
Downloading: http://localhost:8081/repository/maven-releases/tn/esprit/spring/tp1-spring-mvc-rest/maven-metadata.xml
Progress (1): 340 B


Downloaded: http://localhost:8081/repository/maven-releases/tn/esprit/spring/tp1-spring-mvc-rest/maven-metadata.xml (340 B at 24 kB/s)
Uploading: http://localhost:8081/repository/maven-releases/tn/esprit/spring/tp1-spring-mvc-rest/maven-metadata.xml
Progress (1): 369 B

Uploaded: http://localhost:8081/repository/maven-releases/tn/esprit/spring/tp1-spring-mvc-rest/maven-metadata.xml (369 B at 2.6 kB/s)
Uploading: http://localhost:8081/repository/maven-releases/tn/esprit/spring/tp1-spring-mvc-rest/maven-metadata.xml
Progress (1): 369 B

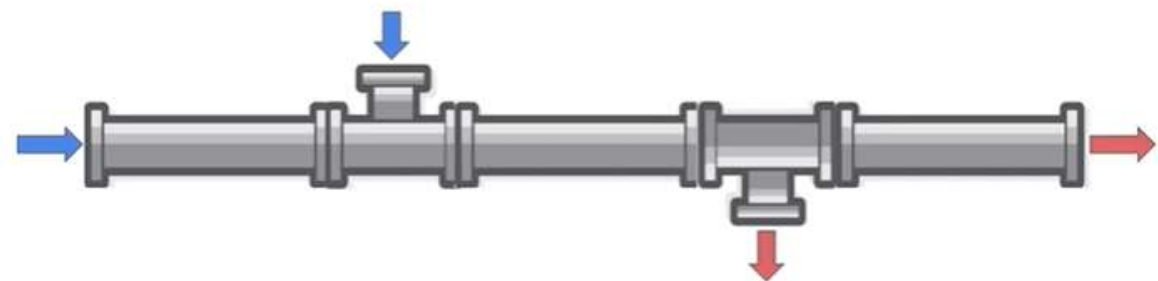
Uploaded: http://localhost:8081/repository/maven-releases/tn/esprit/spring/tp1-spring-mvc-rest/maven-metadata.xml (369 B at 12 kB/s)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 7.473 s
[INFO] Finished at: 2021-10-25T15:35:40+01:00
[INFO] Final Memory: 33M/551M
[INFO] -----
Finished: SUCCESS
```

The bottom right corner of the interface shows 'REST API' and 'Jenkins 2.303.2'.

# Configuration d'un projet avec Jenkins - Pipeline

- Comparé au projet « freestyle », le projet de type « Pipeline » offre un nouveau langage pour l'écriture de la configuration des tâches grâce notamment au langage groovy. 
- Le pipeline peut ainsi être sauvegardé dans un fichier (versionning) et peut gérer les différents cas possibles.

<https://www.jenkins.io/doc/book/pipeline/getting-started/>

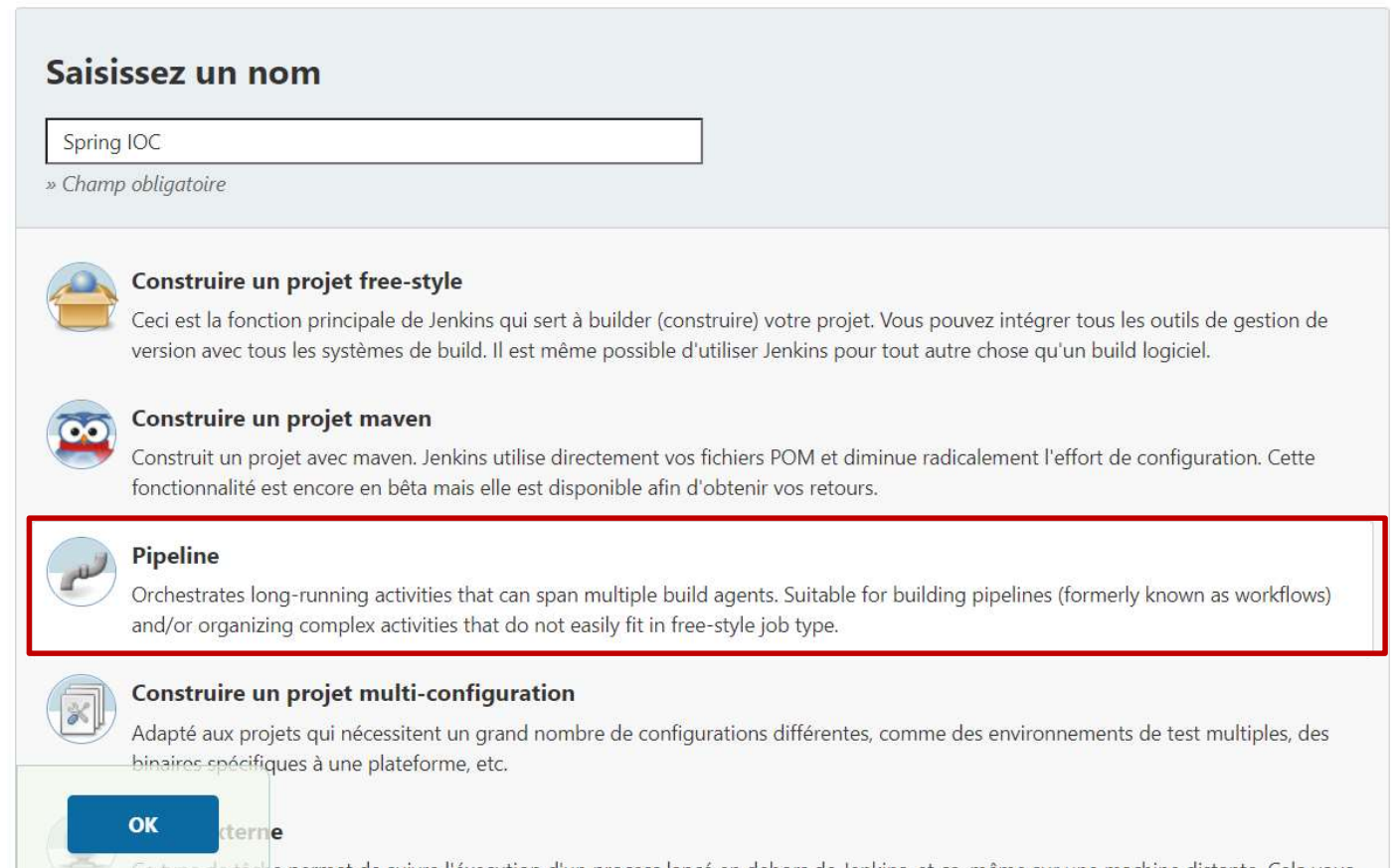
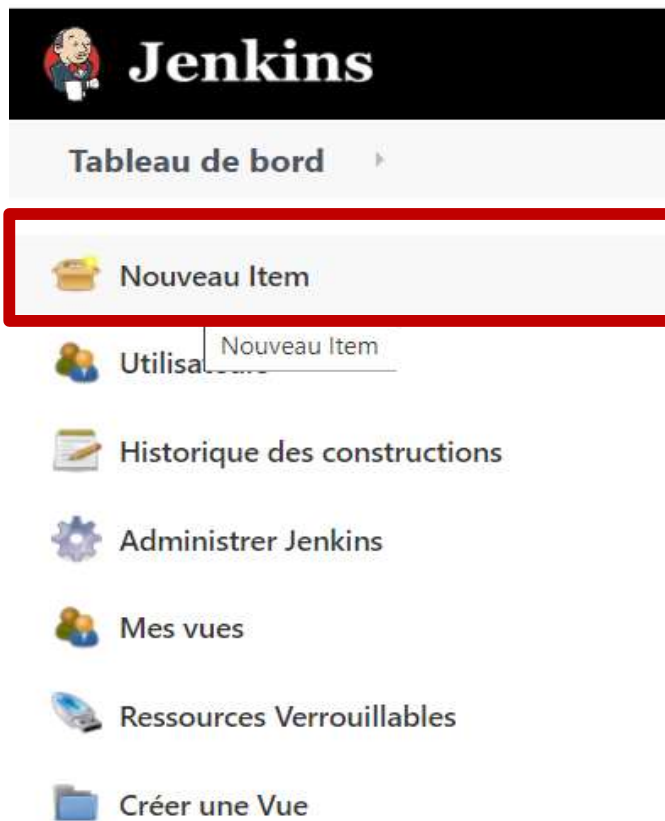


Pipeline



# Configuration d'un projet avec Jenkins - Pipeline

- Pour montrer le fonctionnement de Jenkins, nous allons utiliser un projet spring boot (que vous allez avoir ors du cours Git).  
Au début, il faut créer un « projet » de type pipeline.



# Configuration d'un projet avec Jenkins - Pipeline

**Exemple Simple :** Afficher « Hello world » avec « Groovy »

## Pipeline

### Definition

Pipeline script

### Script



```
1 pipeline {  
2   agent any  
3  
4   stages {  
5     stage('Hello') {  
6       steps {  
7         echo 'Hello World'  
8       }  
9     }  
10  }  
11 }  
12
```

Hello World

# Configuration d'un projet avec Jenkins - Pipeline

## Exemple 01 : Récupération du code du git

### Pipeline

#### Definition

Pipeline script

#### Script

```
1 pipeline {
2   agent any
3   stages{
4     stage('Checkout GIT'){
5       steps {
6
7         echo 'Pulling... ';
8         git branch: 'main',
9           url : 'https://github.com/sirineDevOps/SpringIOC',
10          credentialsId: '5847fd35-c2c3-45fc-a0d5-656108235cb2';
11       }
12     }
13   }
14 }
15
16
17
```

Si le repository est privé

☒ Use Groovy Sandbox

# Configuration d'un projet avec Jenkins - Pipeline

## Exemple 02 : Exécuter une commande Maven

**Pipeline**

Definition






Pipeline script

Script ?

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Testing maven') {
6       steps {
7         sh """mvn -version"""
8       }
9     }
10  }
11 }
12
```

# Configuration d'un projet avec Jenkins - Pipeline

- Si on applique les bonnes pratiques, il est préférable d'utiliser un fichier stocké dans le référentiel du code (git) qui contient le pipeline
- Ce fichier est nommé Jenkinsfile.

 sirineDevOps Create Jenkinsfile		1ef97a6 30 minutes ago	 14 commits
	src/main/java/tn/esprit/esponline	commit from user3	19 days ago
	Jenkinsfile	Create Jenkinsfile	30 minutes ago
	pom.xml	Update pom.xml	2 hours ago

# Configuration d'un projet avec Jenkins - Pipeline

- Pour pointer sur ce fichier, il suffit juste de configurer le pipeline jenkins.

**Pipeline**

Definition

Pipeline script from SCM

SCM

Git

Repositories

Repository URL

Credentials

Branches to build

Branch Specifier (blank for 'any')

# Configuration d'un projet avec Jenkins - Bonus

- Lancer un build à distance (à travers un URL à saisir dans le navigateur)

pipeline-1 >

ation

Project Options

### Build Triggers

- ☐ Construire après le build sur d'autres projets ?
- ☐ Construire périodiquement ?
- ☐ GitHub hook trigger for GITScm polling ?
- ☐ Scrutation de l'outil de gestion de version ?
- ☐ Période d'attente ?
- ☒ Déclencher les builds à distance (Par exemple, à partir de scripts) ?

Jeton d'authentification

pipeline-1-token

Utilisez l'URL qui suit pour lancer un build à distance : JENKINS\_URL/job/pipeline-1/build?token=TOKEN\_NAME ou /buildWithParameters?token=TOKEN\_NAME

<http://192.168.162.222:8080/job/pipeline-1/build?token=pipeline-1-token>



# Configuration d'un projet avec Jenkins - Bonus

Dans le navigateur :

**<http://192.168.162.222:8080/job/pipeline-1/build?token=pipeline-1-token>**

Dans Jenkins :

The screenshot shows the Jenkins interface for a pipeline named 'pipeline-1'. The breadcrumb navigation at the top reads 'Tableau de bord > pipeline-1 >'. On the left sidebar, there are five menu items: 'Configurer' (with a gear icon), 'Supprimer Pipeline' (with a trash icon), 'Full Stage View' (with a magnifying glass icon), 'Renommer' (with a pencil icon), and 'Pipeline Syntax' (with a question mark icon). Below these is a section for 'Historique des builds' with a 'tendance' dropdown and a search bar labeled 'Filter builds...'. The main content area is titled 'Stage View' and features a 'Recent Changes' link with a code icon. It displays 'Average stage times: (Average full run time: ~1s)'. A table shows stage execution times for build #13:

	Git	Maven Build	Maven Test
Average stage times:	141ms	118ms	111ms
#13 (Sep 18 21:24)	164ms	141ms	106ms

Build #13 is marked as successful with a green checkmark. Build #12 is partially visible below.

# Travail à faire

- En utilisant « Groovy », créer une chaîne d'intégration continue qui se lance lors de la détection d'un push dans le git (Jenkinsfile).
- Ce pipeline contient:
  - ✓ La récupération du code source.
  - ✓ Affichage de la date système
- (La case « Ce qui déclenche le build » est **à décocher** dans tous vos projets (freestyle ou pipeline) après vos tests pour ne pas avoir un job qui tourne chaque minute indéfiniment).

# Jenkins

