

# Atelier 1 - Installation et configuration de Jenkins



## Objectif

La mise en place du serveur Jenkins et l'installation et la configuration des outils nécessaires pour la construction d'un Projet (Build) Maven Jenkins.

## Introduction

Jenkins est l'orchestrateur de l'usine logicielle, il est utilisé pour construire des projets dans une grande variété de langages de programmation, y compris Java, .NET, Ruby, Groovy, Grails, PHP etc. Jenkins prend en charge le build de projets de type Apache Maven, Apache Ant, Gradle, Freestyle, etc.

Dans cet atelier, vous allez installer un environnement d'intégration continue basé sur Jenkins pour la construction de projets Maven Jenkins (Jenkins et Maven) en trois phases. Pour ce faire, vous devez tout d'abord installer un ensemble d'outils (JDK, MAVEN, GIT) nécessaires pour automatiser le processus de construction. Ensuite, installer et configurer Jenkins sous Linux (en tant que des packages Linux). Enfin, installer les plugins et configurer les outils dans Jenkins.

- Un complément de cet atelier "[Jenkins - Installation sous Windows](#)" présente les étapes d'installation de Jenkins sous Windows (en tant que des fichiers WAR).

## Phase-1 : Installation d'outils pour la construction de projets Maven

Les outils associés à Jenkins pour la construction de projets JAVA Maven sont : JDK, MAVEN, et GIT :

- Installation de JDK 17
  - Allez dans le répertoire de votre machine virtuelle Ubuntu, démarrez la machine virtuelle (`vagrant up`) et ouvrez un terminal (`vagrant ssh`). Pour plus d'informations, consultez le support "Installation Vagrant-Ubunto"

```
PS C:\Master\ubuntu> vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'bento/ubuntu-22.04' version '202404.23.0' is up to date...
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
```

```
PS C:\Master\ubuntu> vagrant ssh
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-112-generic x86_64)

vagrant@vagrant:~$
```

- Tapez les commandes suivantes pour installer la OpenJDK 17 (par défaut, Ubuntu installe la version 11):

```
sudo apt update
```

```
sudo apt -y upgrade
```

```
sudo apt install openjdk-17-jdk
```

- Pour vérifier que le JDK 17 est bien installé, exécutez la commande suivante :

```
vagrant@vagrant:~$ java -version
openjdk version "17.0.11" 2024-04-16
OpenJDK Runtime Environment (build 17.0.11+9-Ubuntu-122.04.1)
OpenJDK 64-Bit Server VM (build 17.0.11+9-Ubuntu-122.04.1, mixed mode, sharing)
vagrant@vagrant:~$
```

- Pour que les applications Java puissent trouver la machine virtuelle Java de manière précise, il est nécessaire de configurer la variable d'environnement "JAVA\_HOME".

1. Ouvrir le fichier de configuration système /etc/environment :

```
vagrant@vagrant:~$ sudo nano /etc/environment
```

2. Ajouter la variable d'environnement suivante dans le fichier /etc/environment :

```
JAVA_HOME="/usr/lib/jvm/java-17-openjdk-amd64/"
```

```
vagrant@vagrant: ~
GNU nano 6.2 /etc/environment
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"
JAVA_HOME="/usr/lib/jvm/java-17-openjdk-amd64/"
```

3. Appliquer les modifications en utilisant la commande suivante :

```
source /etc/environment
```

4. Utiliser la commande suivante pour vérifier que JAVA\_HOME a été configuré correctement :

```
echo $JAVA_HOME
```

```
vagrant@vagrant:~$ source /etc/environment
vagrant@vagrant:~$ echo $JAVA_HOME
/usr/lib/jvm/java-17-openjdk-amd64/
vagrant@vagrant:~$
```

### ➤ Installation de MAVEN

- Pour procéder à l'installation de Maven, veuillez ouvrir votre terminal et exécuter les commandes suivantes de manière consécutive :

```
sudo apt install maven -y
M2_HOME="opt/apache-maven-3.6.3"
PATH="$M2_HOME/bin:$PATH"
export PATH
```

```
vagrant@vagrant:~$ sudo apt install maven -y
M2_HOME="opt/apache-maven-3.6.3"
PATH="$M2_HOME/bin:$PATH"
export PATH
Reading package lists... Done
```

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

```
vagrant@vagrant:~$ mvn -version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 17.0.11, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.15.0-102-generic", arch: "amd64", family: "unix"
vagrant@vagrant:~$
```

- Ouvrir le fichier de configuration système /etc/environment et définir la variable d'environnement de Maven "M2\_HOME" :

```
M2_HOME="opt/apache-maven-3.6.3"
```

```
vagrant@vagrant: ~
GNU nano 6.2 /etc/environment
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"
JAVA_HOME="/usr/lib/jvm/java-17-openjdk-amd64/"
M2_HOME="opt/apache-maven-3.6.3"
```

```
vagrant@vagrant:~$ source /etc/environment
vagrant@vagrant:~$ echo $M2_HOME
opt/apache-maven-3.6.3
vagrant@vagrant:~$
```

### ➤ Installation de GIT

- Pour pouvoir utiliser Git avec Jenkins, vous devez installer Git sur votre machine virtuelle (VM). Aucune configuration de Git dans Jenkins ne sera nécessaire.

```
sudo apt install git
```

```
vagrant@vagrant:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.11).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

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

**git --version**

```
vagrant@vagrant:~$ git --version
git version 2.34.1
vagrant@vagrant:~$
```

## Phase-2 : Installation et configuration de Jenkins

- En exécutant les commandes suivantes en ordre, vous installez, configurez et démarrez Jenkins sur votre système Ubuntu, prêt à être utilisé pour l'intégration continue et l'automatisation des tâches de développement.

1. **Ajouter la clé Jenkins** : Cette commande télécharge une clé de sécurité pour Jenkins et la rend utilisable par le système.

```
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key |
sudo apt-key add -
```

2. **Ajouter la source Jenkins** : Cette commande configure le système pour télécharger les paquets Jenkins depuis un emplacement spécifique sur le web.

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

3. **Récupérer une autre clé** : Cette commande obtient une clé de sécurité supplémentaire nécessaire pour Jenkins.

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
5BA31D57EF5975CA
```

4. **Mettre à jour la liste des paquets** : Cette commande actualise la liste des logiciels disponibles pour installation, y compris Jenkins.

```
sudo apt update
```

5. **Installer Jenkins** : Cette commande installe Jenkins et ses composants associés.

```
sudo apt install Jenkins
```

6. **Démarrer Jenkins** : Cette commande démarre Jenkins en tant que service.

```
sudo systemctl start Jenkins
```

7. **Activer le démarrage automatique de Jenkins** : Cette commande configure Jenkins pour qu'il démarre automatiquement lorsque le système démarre.

```
sudo systemctl enable jenkins
```

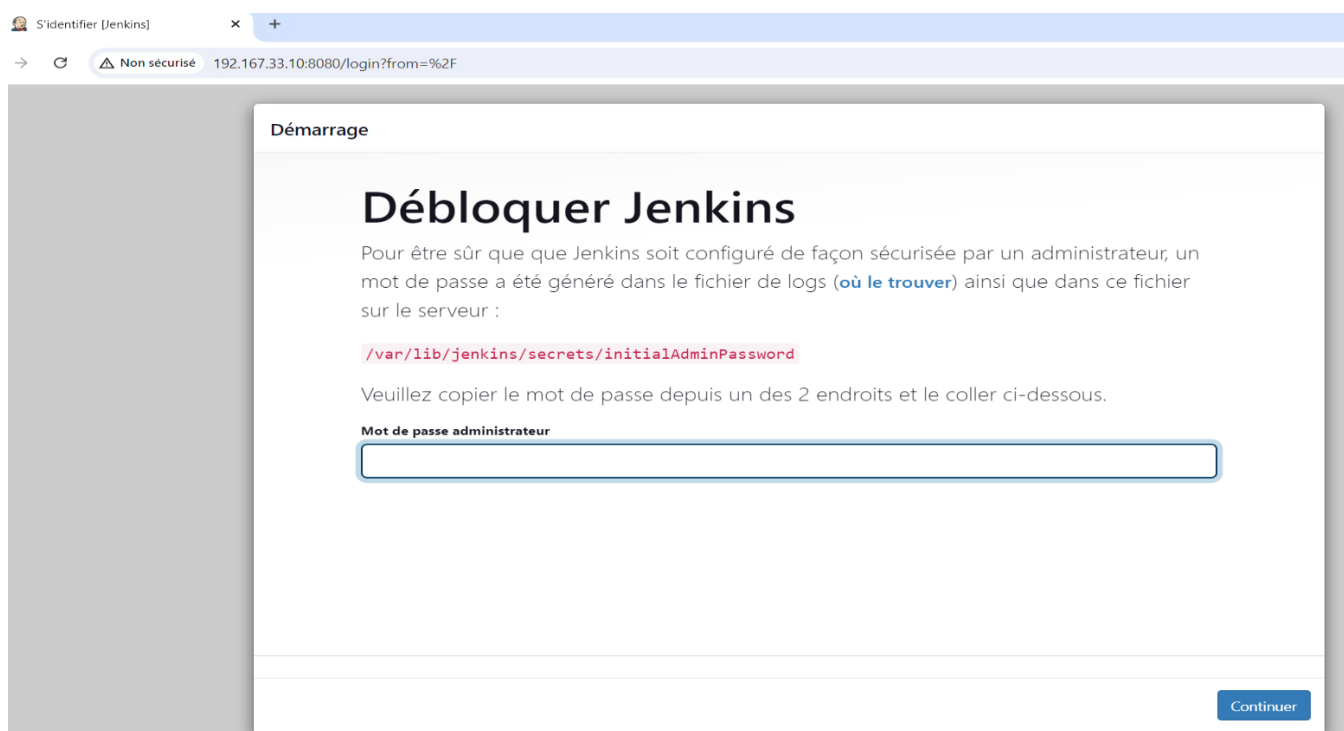
- Pour vérifier l'installation de Jenkins, vous pouvez lancer la commande suivante :

```
sudo systemctl status jenkins
```

```
vagrant@vagrant:~$ sudo systemctl status jenkins
jenkins.service - Jenkins Continuous Integration Server
Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
Active: active (running) since Sun 2024-06-23 18:56:12 UTC; 1h 45min ago
Main PID: 45294 (java)
Tasks: 45 (limit: 3339)
Memory: 909.3M
CPU: 2min 21.980s
CGroup: /system.slice/jenkins.service
        └─45294 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
```

- Pour accéder à Jenkins, vous devez obtenir l'adresse IP de la machine virtuelle en utilisant la commande suivante : **ip addr show**

```
vagrant@vagrant:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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 fq_codel state UP group default qlen 1000
    link/ether 08:00:27:06:e8:56 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic eth0
        valid_lft 86332sec preferred_lft 86332sec
    inet6 fe80::a00:27ff:fe06:e856/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:0c:70:53 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 192.167.33.10/24 brd 192.167.33.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe0c:7053/64 scope link
        valid_lft forever preferred_lft forever
vagrant@vagrant:~$
```



- Lors de l'installation de Jenkins, un fichier contenant le mot de passe initial d'administration est généré et stocké sur le système. Vous devez accéder à ce mot de passe pour débloquer Jenkins.
- Pour afficher le mot de passe, utilisez la commande suivante :

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Veuillez copier le mot de passe depuis un des 2 endroits et le coller ci-dessous.

Mot de passe administrateur

```
vagrant@vagrant:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
a297fc704346467fb892d7399d2bb0af
```

- Installer les plugins suggérés

Assistant d'installation [Jenkins] x +

→ ↻ ⚠ Non sécurisé 192.167.33.10:8080

Démarrage

## Personnaliser Jenkins

Les plugins étendent Jenkins avec des fonctionnalités additionnelles pour satisfaire différents besoins.

**Installer les plugins suggérés**

Installer les plugins que la communauté Jenkins trouve les plus utiles.

**Sélectionner les plugins à installer**

Sélectionner et installer les plugins les plus utiles à vos besoins.

Installation en cours...

### Installation en cours...

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	⚙ Credentials Binding
⚙ Timestampers	⚙ Workspace Cleanup	⚙ Ant	⚙ Gradle
⚙ Pipeline	⚙ GitHub Branch Source	⚙ Pipeline: GitHub Groovy Libraries	⚙ Pipeline: Stage View
⚙ Git	⚙ SSH Build Agents	⚙ Matrix Authorization Strategy	⚙ PAM Authentication
⚙ LDAP	⚙ Email Extension	⚙ Mailer	

Folders  
 \*\* JavaBeans Activation Framework (JAF) API  
 \*\* JavaMail API  
 \*\* bouncycastle API  
 \*\* Instance Identity  
 \*\* Mina SSHD API :: Common  
 \*\* Mina SSHD API :: Core  
 \*\* SSH server  
 OWASP Markup Formatter  
 \*\* Struts  
 \*\* Token Macro  
 Build Timeout  
 \*\* Credentials  
 \*\* - dépendance requise

- Vous n'êtes pas obligé de créer de nouveaux utilisateurs, vous pouvez continuer à utiliser l'utilisateur "**admin**".

## Créer le 1er utilisateur Administrateur

Nom d'utilisateur:

Mot de passe:

Confirmation du mot de passe:

Nom complet:

Adresse courriel:

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

- Pour personnaliser le numéro de port de Jenkins, vous pouvez simplement le modifier dans cette fenêtre. Vous avez la possibilité de choisir de maintenir le port **8080** ou d'en sélectionner un autre selon vos préférences.

### 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.

- Vous pouvez changer le mot de passe de "admin" à "jenkins" par exemple :


 **Jenkins**

Tableau de bord > admin > Configurer

Statut

Constructions

Configurer

Mes vues

Identifiants

Mot de passe

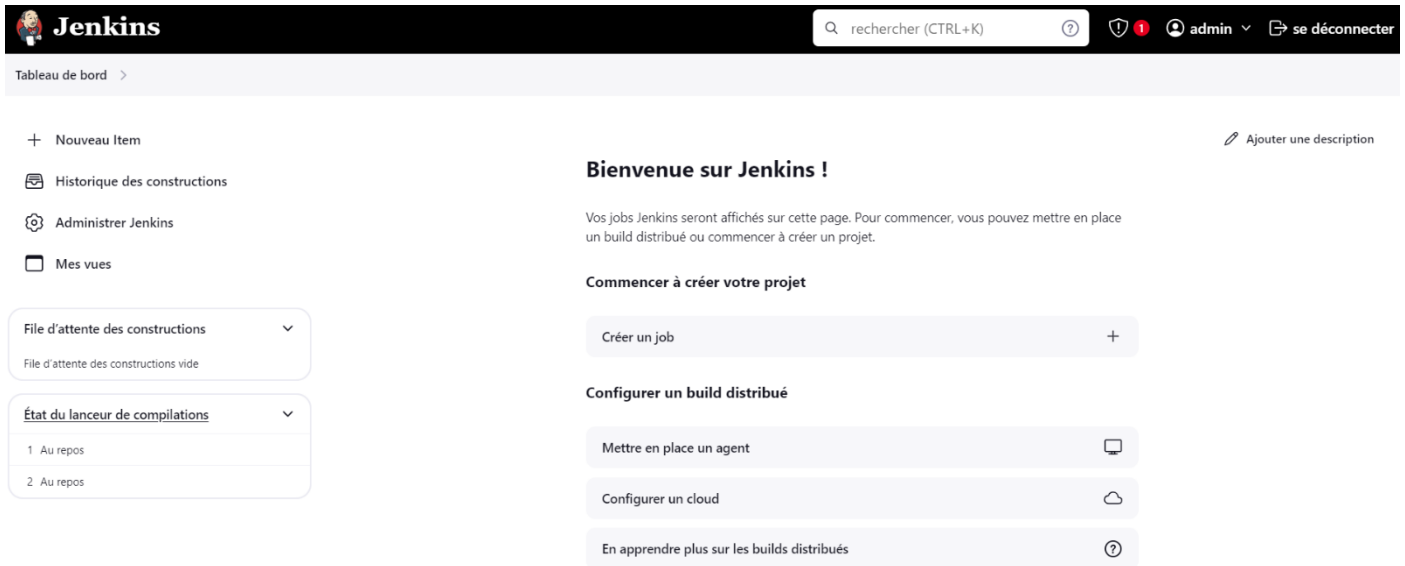
Mot de passe:

Confirmer le mot de passe:

[Sauvegarder](#) [Appliquer](#)

- Jenkins offre une interface web conviviale et intuitive, permettant un accès direct à toutes les configurations disponibles et fournissant des informations complètes sur tous les jobs.





**Jenkins** | recherche (CTRL+K) | admin | se déconnecter

Tableau de bord >

+ Nouveau Item

Historique des constructions

Administrer Jenkins

Mes vues

File d'attente des constructions

File d'attente des constructions vide

État du lanceur de compilations

1 Au repos

2 Au repos

**Bienvenue sur Jenkins !**

Vos jobs Jenkins seront affichés sur cette page. Pour commencer, vous pouvez mettre en place un build distribué ou commencer à créer un projet.

**Commencer à créer votre projet**

Créer un job

**Configurer un build distribué**

Mettre en place un agent

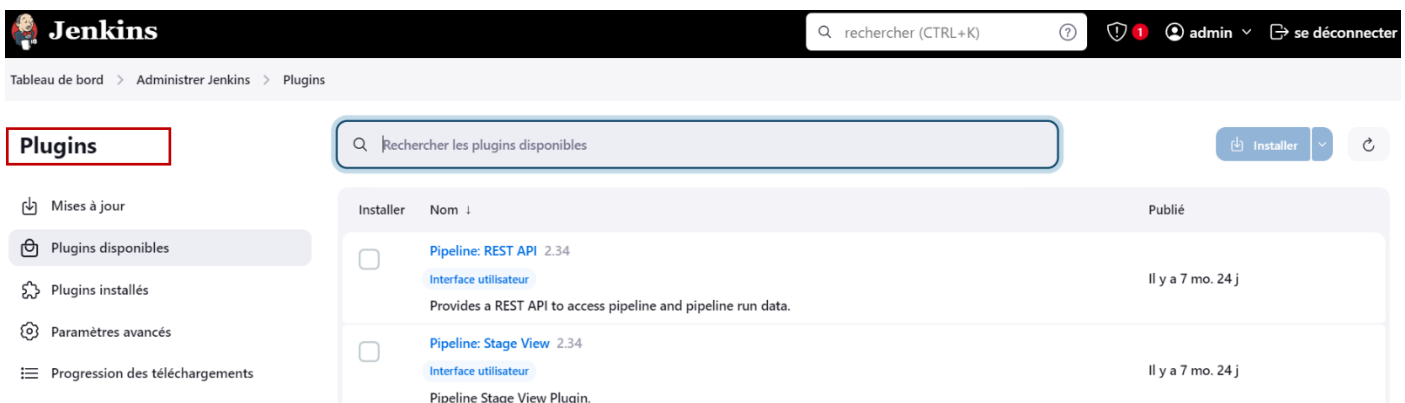
Configurer un cloud

En apprendre plus sur les builds distribués

Ajouter une description

## Phase-3 : Installation de plugins et configuration d'outils dans Jenkins

- Pour installer des plugins, il vous suffit d'accéder à la fenêtre de **"Plugins"**.



**Jenkins** | recherche (CTRL+K) | admin | se déconnecter

Tableau de bord > Administrer Jenkins > Plugins

**Plugins**

Rechercher les plugins disponibles

Installer

Installer	Nom	Publié
<input type="checkbox"/>	<b>Pipeline: REST API</b> 2.34 Interface utilisateur Provides a REST API to access pipeline and pipeline run data.	Il y a 7 mo. 24 j
<input type="checkbox"/>	<b>Pipeline: Stage View</b> 2.34 Interface utilisateur Pipeline Stage View Plugin.	Il y a 7 mo. 24 j

Mises à jour

Plugins disponibles

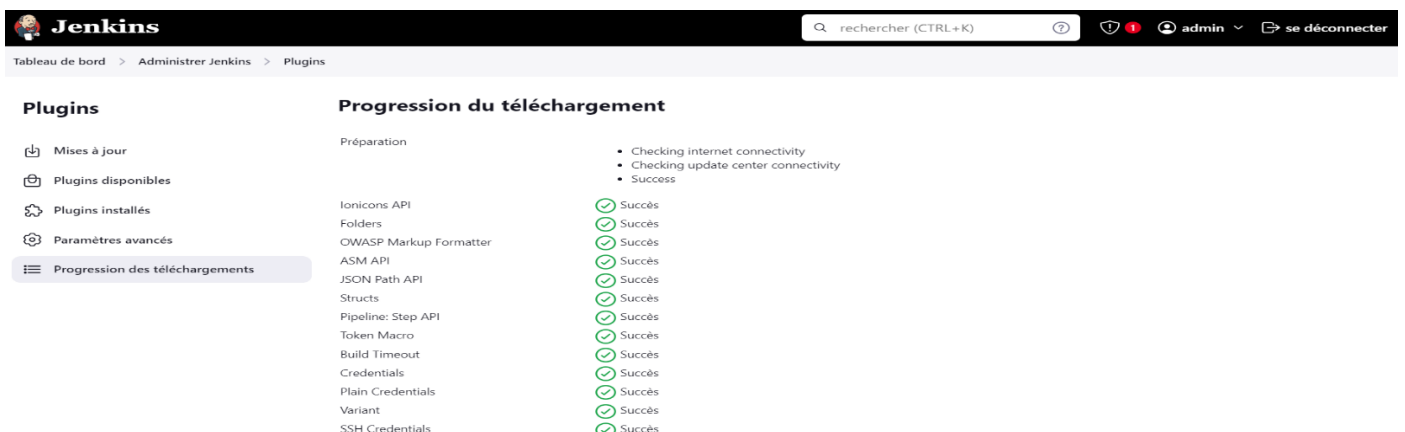
Plugins installés

Paramètres avancés

Progression des téléchargements

- Pour créer votre chaîne d'intégration continue, vous allez installer les plugins suivants dans Jenkins (Installez ces plugins sans redémarrer, puis redémarrez Jenkins à la fin) :

- Git plugin (normalement déjà installé, mais vérifier)
- Maven Integration
- Sonargraph Integration
- SonarQube Scanner



**Jenkins** | recherche (CTRL+K) | admin | se déconnecter

Tableau de bord > Administrer Jenkins > Plugins

**Plugins**

Mises à jour

Plugins disponibles

Plugins installés

Paramètres avancés

Progression des téléchargements

**Progression du téléchargement**

Préparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Icons API	Succès
Folders	Succès
OWASP Markup Formatter	Succès
ASM API	Succès
JSON Path API	Succès
Structs	Succès
Pipeline: Step API	Succès
Token Macro	Succès
Build Timeout	Succès
Credentials	Succès
Plain Credentials	Succès
Variant	Succès
SSH Credentials	Succès

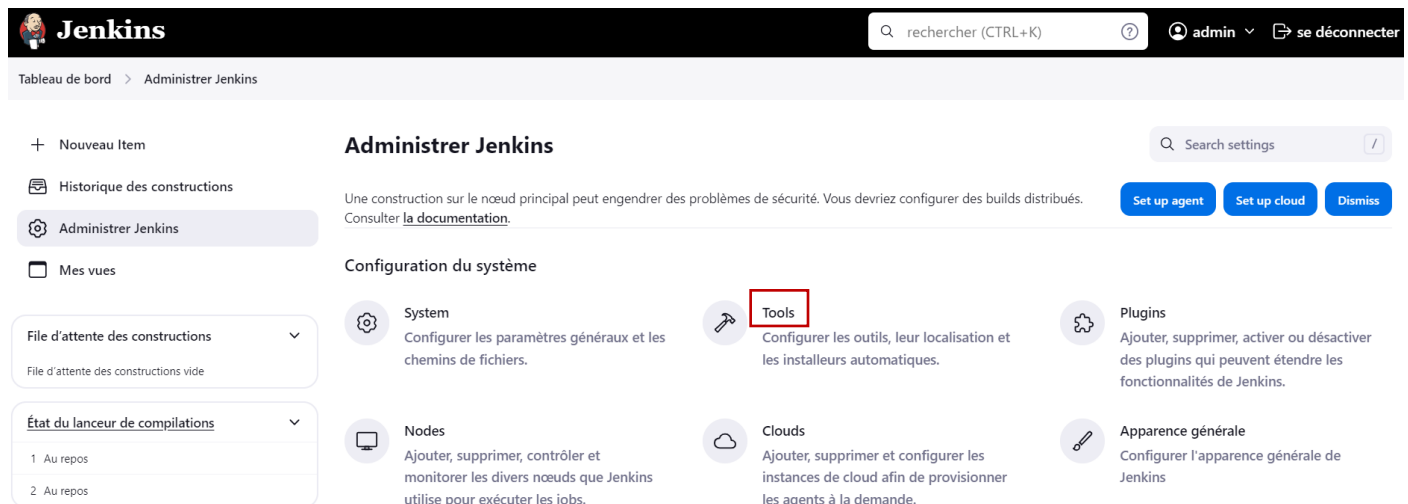


- Après avoir installé les plugins, vous devez redémarrer Jenkins :

```
sudo systemctl restart jenkins
```

```
vagrant@vagrant:~$ sudo systemctl restart jenkins
vagrant@vagrant:~$
```

- Pour configurer les outils, vous devez accéder à la fenêtre "Tools"



The screenshot shows the Jenkins Administration interface. The 'Tools' option under 'Configuration du système' is highlighted with a red box. The interface includes a sidebar with navigation links like 'Nouveau Item', 'Historique des constructions', 'Administrer Jenkins', and 'Mes vues'. The main area displays various configuration options: System, Tools (highlighted), Plugins, Nodes, Clouds, and Apparence générale. A search bar and 'Set up agent', 'Set up cloud', and 'Dismiss' buttons are also visible.

- Faites référence au JDK déjà installé sur votre VM et enregistrez cette configuration :

```
vagrant@vagrant:~$ echo $JAVA_HOME
/usr/lib/jvm/java-17-openjdk-amd64/
```

## Tools

### 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

### Installations JDK

Ajouter JDK

### Installations JDK

Ajouter JDK

≡ JDK

Nom

JAVA\_HOME

JAVA\_HOME

/usr/lib/jvm/java-17-openjdk-amd64/

☐ Install automatically ?

Enregistrer

Appliquer

- Faites référence au Maven déjà installé sur votre VM et enregistrez cette configuration :

```
vagrant@vagrant:~$ echo $M2_HOME
opt/apache-maven-3.6.3
```

### Installations Maven

Ajouter Maven

Maven

Nom

☒ Install automatically ?

Install from Apache

Version

### Installations Maven

Ajouter Maven

Maven

Nom

MAVEN\_HOME

Ajouter Maven

Enregistrer

Appliquer

- Rien à configurer

#### Git installations

Git

Name

Path to Git executable ?

☐ Install automatically ?

Add Git

## Conclusion

On est bien arrivé au bout de cet atelier, on a pu donc voir comment on met en place un environnement d'intégration continue. Vous pourrez suivre ces étapes pour installer et configurer le serveur Jenkins. Pour construire des projets Maven, vous devez installer les outils nécessaires (JDK, Maven et Git) et les associer à Jenkins. Des plugins indispensables pour la gestion de builds sont aussi à installer sous le serveur Jenkins.

Vous pourrez trouver une liste exhaustive de plugins à installer sous Jenkins [ici](#).