

Jenkins



Bureau E204

Plan du cours

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

La vie avant Jenkins



Jenkins - Définition

- Jenkins est un serveur open source entièrement écrit en Java. Il permet d'exécuter une série d'actions pour réaliser les processus devops, également de manière automatisée. . C'est un ordonnanceur (**Scheduler**).



Jenkins - Définition

- Il s'exécute dans des conteneurs de servlets tels qu'Apache Tomcat ou bien il utilise son serveur web embarqué « Jetty ».
- Il s'interface avec des systèmes de gestion de versions tels que ,CVS, Git et Subversion, et exécute des projets basés sur Apache Ant et Apache Maven aussi bien que des scripts arbitraires en shell Unix, batch Windows ou python.

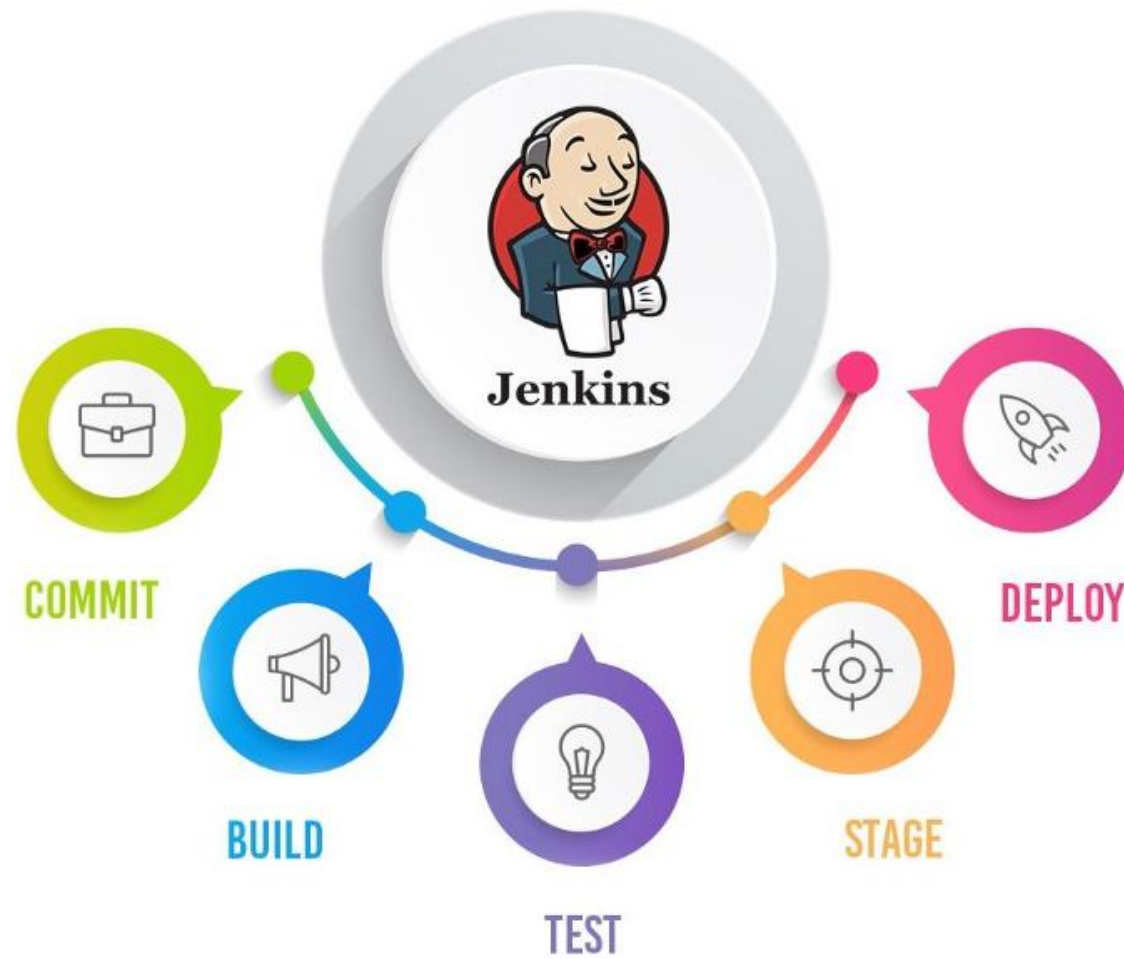
Jenkins - Définition

- Par défaut, Jenkins fait juste le nécessaire pour administrer le serveur, mettre en place les machines child et exécuter des commandes.
- Tout le reste des fonctionnalités (récupération des sources à partir de Git, exécution sonar, construction projet, envoi d'email lors d'un échec...) est implémenté sous la forme de modules (plugins)

Jenkins - Définition

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

Jenkins - Définition



Jenkins - Définition

Jenkins va exécuter les parties suivantes :

- Récupération du projet à partir du serveur de contrôle de version (GIT)
- Création des fichiers de compilation
- Compilation
- Application des tests automatiques
- Vérification de la qualité du code
- Préparation de la version à distribuer
- Mise en place de la version à distribuer

Installation Git(ubuntu)

- Installation Git
 - Pour installer Git, vous devez exécuter la command suivante:
 - `apt install -y git`
 - Pour vérifier l'installation, vous devez exécuter la commande
 - `git --version`

```
[root@localhost vagrant]# git --version
git version 1.8.3.1
```

Installation JDK (ubuntu)

- Installation JDK
 - Dans la machine virtuelle « Ubuntu », lancer le terminal et exécuter les commandes suivantes: (Si vous utilisez un compte 'super user', exécutez la commande sans **sudo**)
 - `sudo apt update`
 - `sudo apt install -y default-jdk` (ou bien la version spécifique 17 de jdk)
 - Pour vérifier que le JDK est bien installé, exécuter la commande suivante:

```
[root@adsl-172-10-0-35 vagrant]# 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 11.0.16+8-LTS)
OpenJDK 64-Bit Server VM (Red_Hat-11.0.16.0.8-1.el7_9) (build 11.0.16+8-LTS, mixed mode, sharing)
```

Variables d'environnement(ubuntu)

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-11-openjdk-amd64/"
```



```
vagrant@vagrant: ~  
GNU nano 6.2 /etc/environment *  
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/ga  
JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64/"
```

Variables d'environnement(ubuntu)

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 $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64/
vagrant@vagrant:~$ |
```

Installation Maven(ubuntu)

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" //pointer vers  
l'emplacement maven dans votre machine
```

```
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
```

Installation Maven(ubuntu)

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

```
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: "5.15.0-67-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"
JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64/"
M2_HOME="/opt/apache-maven-3.6.3"
```

Installation de Jenkins (ubuntu)

- Installation 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.

```
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-key adv --keyserver keyserver.ubuntu.com --recv-keys 5BA31D57EF5975CA  
sudo apt update  
sudo apt install jenkins  
sudo systemctl start jenkins      sudo systemctl enable jenkins
```


Installation de Jenkins (ubuntu)

Voici une description simple de chaque commande :

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.key | sudo apt-key add -
```

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 http://pkg.jenkins.io/debian-stable binary/ >  
/etc/apt/sources.list.d/jenkins.list'
```

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
```

Installation de Jenkins (ubuntu)

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

```
sudo apt update
```

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

```
sudo apt install jenkins
```

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

```
sudo systemctl start jenkins
```

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

```
sudo systemctl enable jenkins
```

Installation de Jenkins (ubuntu)

○ Pour lancer Jenkins, exécuter la commande suivante:

- `sudo systemctl start jenkins.service`

Optionnel: Pour lancer jenkins comme service au démarrage

- `sudo systemctl enable jenkins.service`

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

- `sudo systemctl status jenkins`

```
[root@adsl-172-10-0-35 ~]# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled)
   Active: active (running) since Mon 2022-09-05 15:36:38 UTC; 6min ago
     Main PID: 32539 (java)
    CGroup: /system.slice/jenkins.service
            └─32539 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war -...

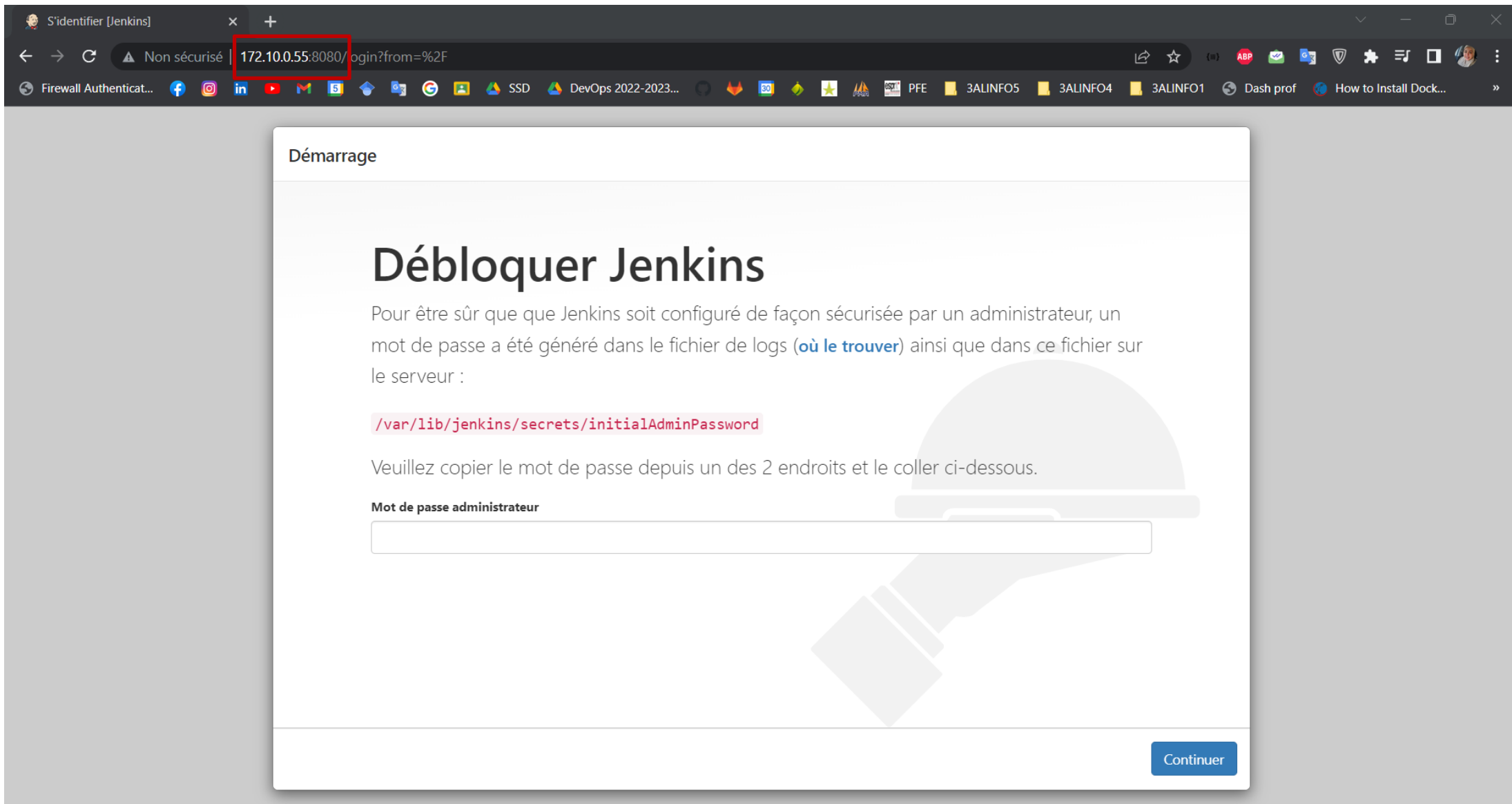
Sep 05 15:36:11 adsl-172-10-0-35.dsl.sndg02.sbcglobal.net jenkins[32539]: This may also be found at: /var/lib/jenki...rd
Sep 05 15:36:11 adsl-172-10-0-35.dsl.sndg02.sbcglobal.net jenkins[32539]: *****
Sep 05 15:36:11 adsl-172-10-0-35.dsl.sndg02.sbcglobal.net jenkins[32539]: *****
Sep 05 15:36:11 adsl-172-10-0-35.dsl.sndg02.sbcglobal.net jenkins[32539]: *****
Sep 05 15:36:32 adsl-172-10-0-35.dsl.sndg02.sbcglobal.net jenkins[32539]: 2022-09-05 15:36:32.406+0000 [id=43] ...on
```

Installation de Jenkins (ubuntu)

- Pour accéder à Jenkins, vous devez récupérer l'adresse ip de la machine virtuelle à travers les commandes suivantes:
 - ip addr show
 - ifconfig

```
[root@adsl-172-10-0-35 ~]# 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 pfifo_fast state UP group default qlen 1000
    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 noprefixroute dynamic eth0
        valid_lft 81889sec preferred_lft 81889sec
    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 state UP group default qlen 1000
    link/ether 08:00:27:37:83:51 brd ff:ff:ff:ff:ff:ff
    inet 172.10.0.55/24 brd 172.10.0.255 scope global noprefixroute eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe37:8351/64 scope link
        valid_lft forever preferred_lft forever
```

Installation de Jenkins (ubuntu)



Installation de Jenkins (ubuntu)

- Pour la première fois, il faut débloquer Jenkins en tapant le mot de passe qui est stocké dans le fichier de log mentionné dans la fenêtre

Mot de passe administrateur


.....

- Installer les plugins suggérés.



Installation de Jenkins (ubuntu)

Installation en cours...



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

** - dépendance requise

Installation de Jenkins (ubuntu)

Démarrage

Créer le 1er utilisateur Administrateur

Nom d'utilisateur:

Mot de passe:

Confirmation du mot de passe:

Nom complet:

Adresse courriel:

Jenkins 2.303.2

Continuer en tant qu'Administrateur

Sauver et continuer

Installation de Jenkins (ubuntu)

- Pour personnaliser le numéro de port de Jenkins, il suffit de le modifier à travers cette fenêtre.

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.

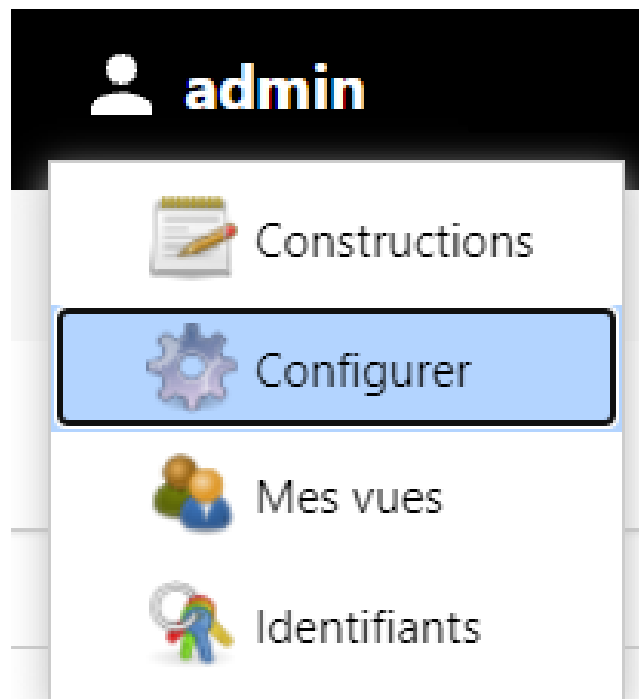
La valeur par défaut affichée **n'est pas encore sauvegardée** et est générée à partir de la requête actuelle, lorsque c'est possible. Il est fortement recommandé d'utiliser comme valeur l'URL qui est censée être utilisée par les utilisateurs. Cela évitera des confusions lors du partage ou de la visualisation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

Jenkins 2.303.2

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

Configuration de Jenkins

- Pour modifier le mot de passe, consulter l'interface « Configurer »



Mot de passe

Mot de passe:

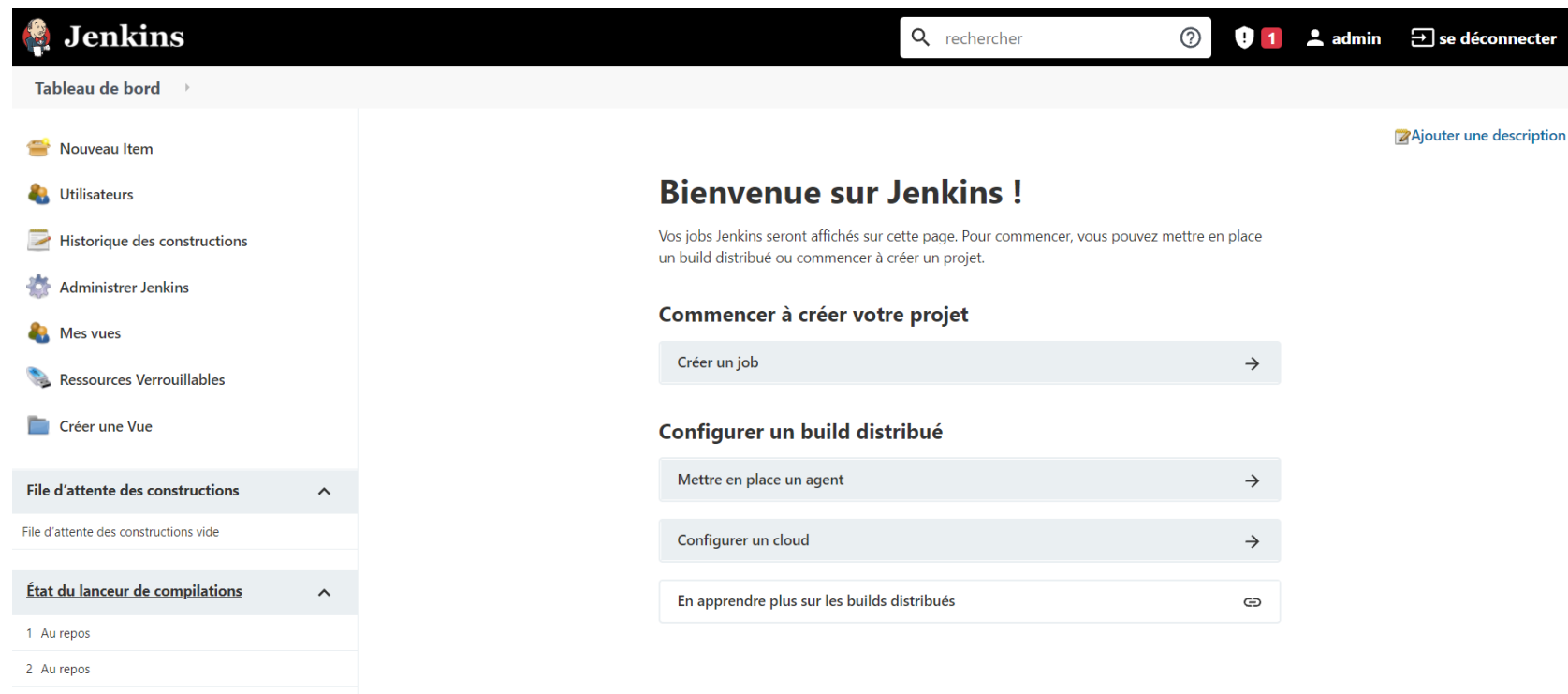
Confirmer le mot de passe:

Sauvegarder

Appliquer

Configuration de Jenkins

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



Configuration de Jenkins - Plugins

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

172.10.0.55:8080/manage/pluginManager/

The screenshot displays the Jenkins Plugin Manager interface. The top navigation bar includes the Jenkins logo, a search bar with the text "rechercher", and user information "admin" with a "se déconnecter" link. The main content area is titled "Gestion des plugins" and features a sidebar with links to "Retour au tableau de bord" and "Administrer Jenkins". The plugin list is filtered by "sonar" and shows three available plugins under the "Disponibles" tab. Each plugin entry includes an "Install" checkbox, the plugin name, a brief description, the version number, and the release date. The "Sonar Quality Gates" and "Quality Gates" plugins have red warning boxes indicating security issues with the current versions.

Install	Name	Version	Released
<input type="checkbox"/>	SonarQube Scanner External Site/Tool Integrations Build Reports This plugin allows an easy integration of SonarQube , the open source platform for Continuous Inspection of code quality.	2.13.1	5 mo. 22 j ago
<input type="checkbox"/>	Sonar Quality Gates 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">Credentials transmitted in plain text	1.3.1	3 an. 2 mo. ago
<input type="checkbox"/>	Quality Gates 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">Credentials transmitted in plain text	2.5	5 an. 5 mo. ago

Configuration de Jenkins - Plugins

- Pour créer notre chaine d'intégration continue, on va installer les plugins suivants dans Jenkins:
 - ✓ Git plugin
 - ✓ Maven Integration
 - ✓ Nexus Platform
 - ✓ SCM API
 - ✓ 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

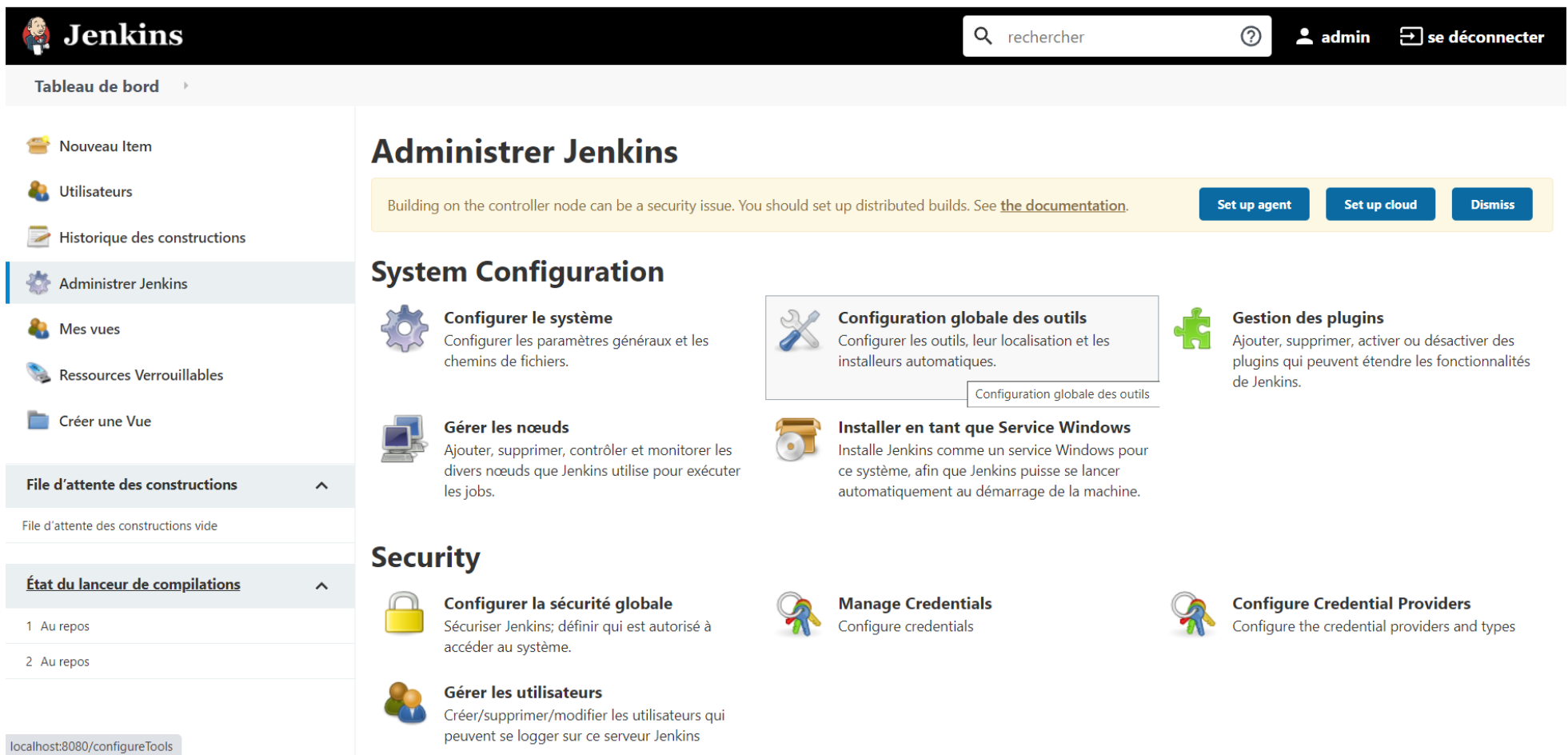
Préparation

- 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

- Pour configurer les outils, il faut accéder à la fenêtre « Configuration globale 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 'Configurer le système', 'Gérer les nœuds', 'Configuration globale des outils' (highlighted with a tooltip), and 'Installer en tant que Service Windows'. The 'Security' section includes 'Configurer la sécurité globale', 'Manage Credentials', 'Configure Credential Providers', and 'Gérer les utilisateurs'. The bottom of the page shows the URL 'localhost:8080/configureTools'.

Jenkins

rechercher admin se déconnecter

Tableau de bord

Nouveau Item

Utilisateurs

Historique des constructions

Administrer Jenkins

Mes vues

Ressources Verrouillables

Créer une Vue

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

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.

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

Gestion des plugins
Ajouter, supprimer, activer ou désactiver des plugins qui peuvent étendre les fonctionnalités de Jenkins.

Gérer les utilisateurs
Créer/supprimer/modifier les utilisateurs qui peuvent se logger sur ce serveur Jenkins


Configuration de Jenkins - JDK


🌐 172.10.0.55:8080/manage/configureTools/


 **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 installé en local

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 installé en local

Maven

Installations Maven

Liste des installations Maven sur ce système

Ajouter Maven

☰

Maven

×

Nom

M2_HOME

MAVEN_HOME

/usr/share/maven

☐

Install automatically ?

Configuration de Jenkins - Git

- Rien à faire

Git

Git installations

≡ Git

Name

Default

Path to Git executable ?

git

☐ Install automatically ?

Add Git ▾

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 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 sont les suivants:

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

Configuration d'un projet avec Jenkins - Job

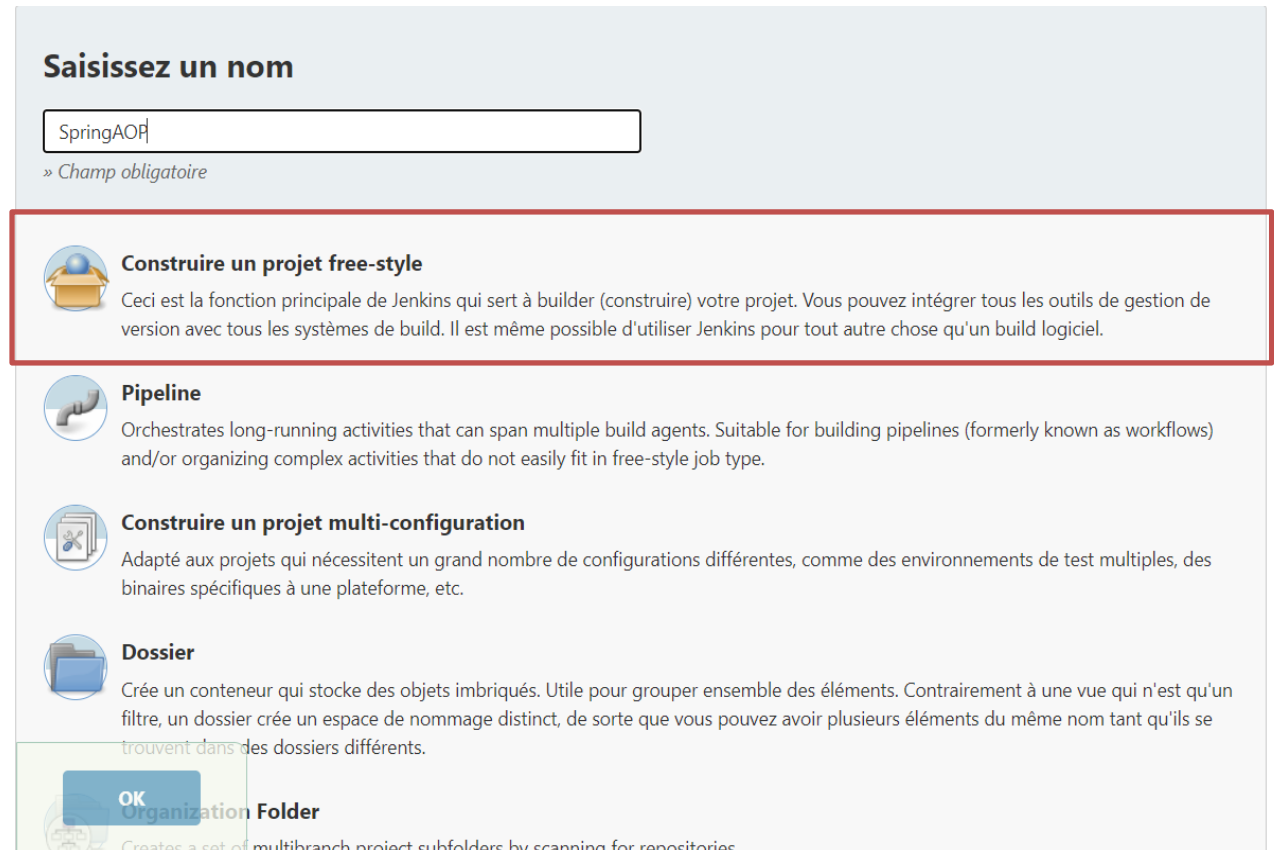
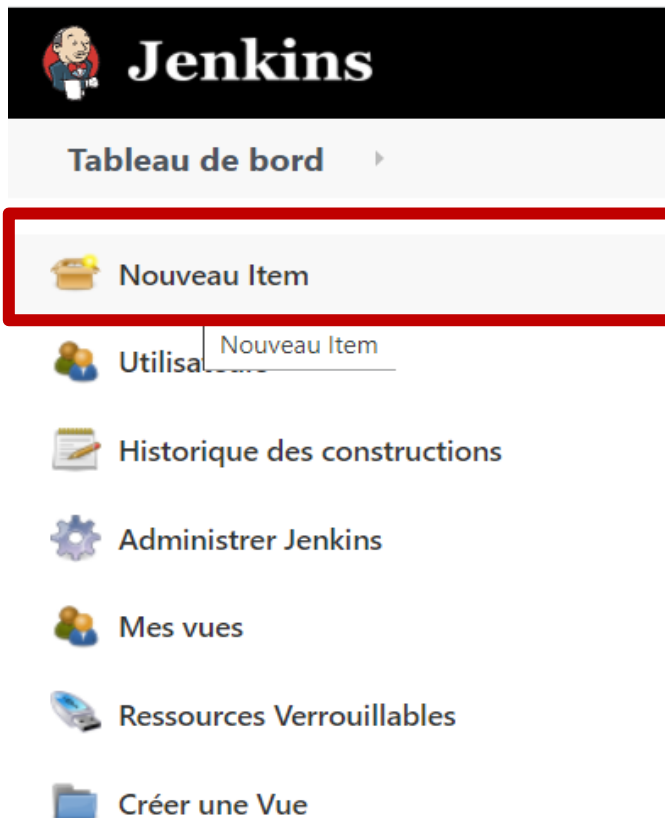
Deux méthodes de configuration sont possibles pour mettre en place un processus de compilation d'un projet :

- 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

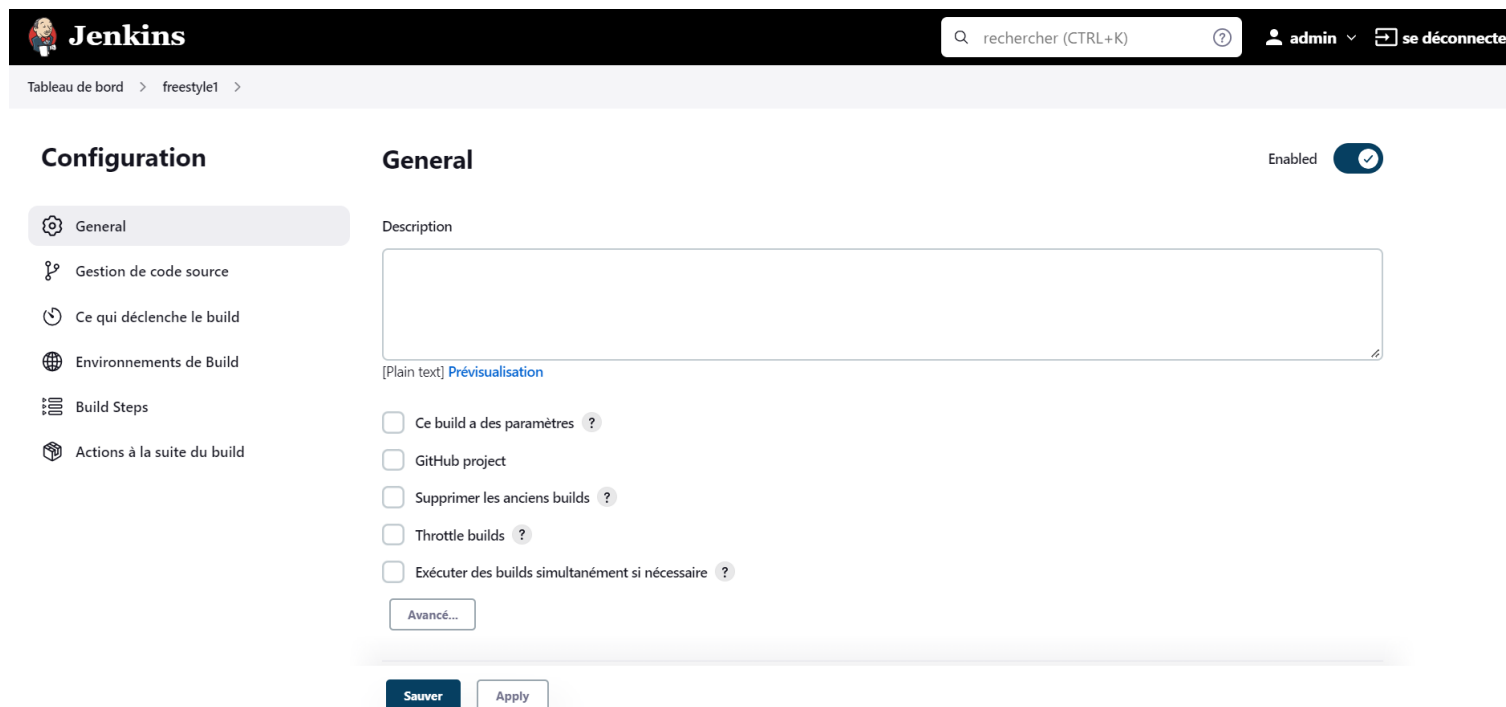
- Pour montrer le fonctionnement de Jenkins, nous allons utiliser un projet spring boot.

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



Configuration d'un projet avec Jenkins - FreeStyle

- Après la création du projet, il faut enchaîner avec les Jobs pour compiler le projet. La première partie de la page de configuration indique les informations générales du projet.

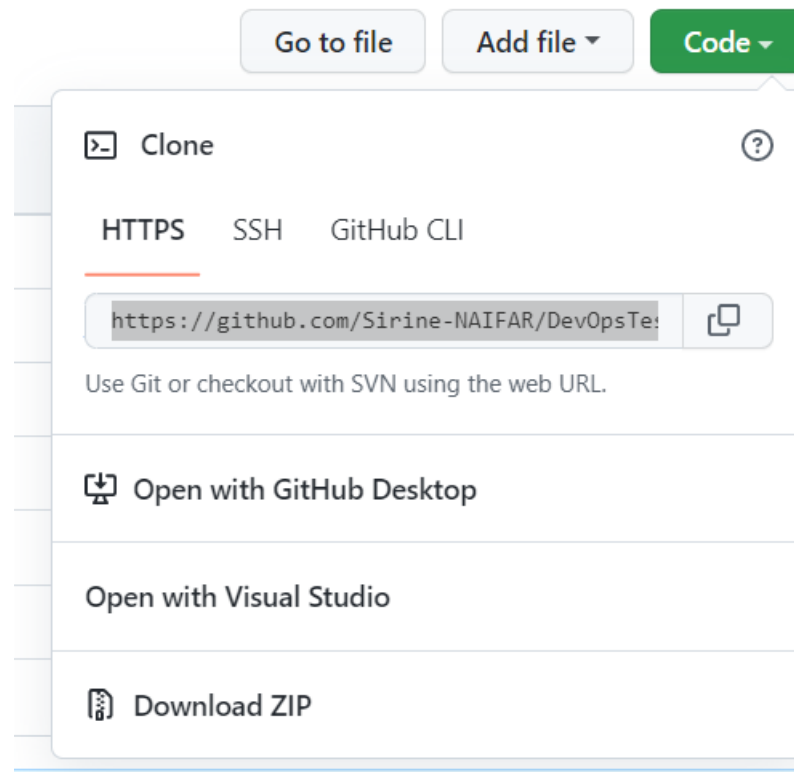


The screenshot shows the Jenkins web interface. At the top is a black header with the Jenkins logo, a search bar containing 'rechercher (CTRL+K)', and a user menu for 'admin' with a 'se déconnecter' link. Below the header is a breadcrumb trail: 'Tableau de bord > freestyle1 >'. The main content area is titled 'Configuration' and 'General'. On the left is a sidebar with a 'Configuration' section containing links: 'General' (selected), 'Gestion de code source', 'Ce qui déclenche le build', 'Environnements de Build', 'Build Steps', 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: 'Ce build a des paramètres ?' (unchecked), 'GitHub project' (unchecked), 'Supprimer les anciens builds ?' (unchecked), 'Throttle builds ?' (unchecked), and 'Exécuter des builds simultanément si nécessaire ?' (unchecked). Below these is an 'Avancé...' button. At the bottom are 'Sauver' and 'Apply' buttons. In the top right corner of the configuration area, there is an 'Enabled' toggle switch which is turned on.

Configuration d'un projet avec Jenkins - FreeStyle

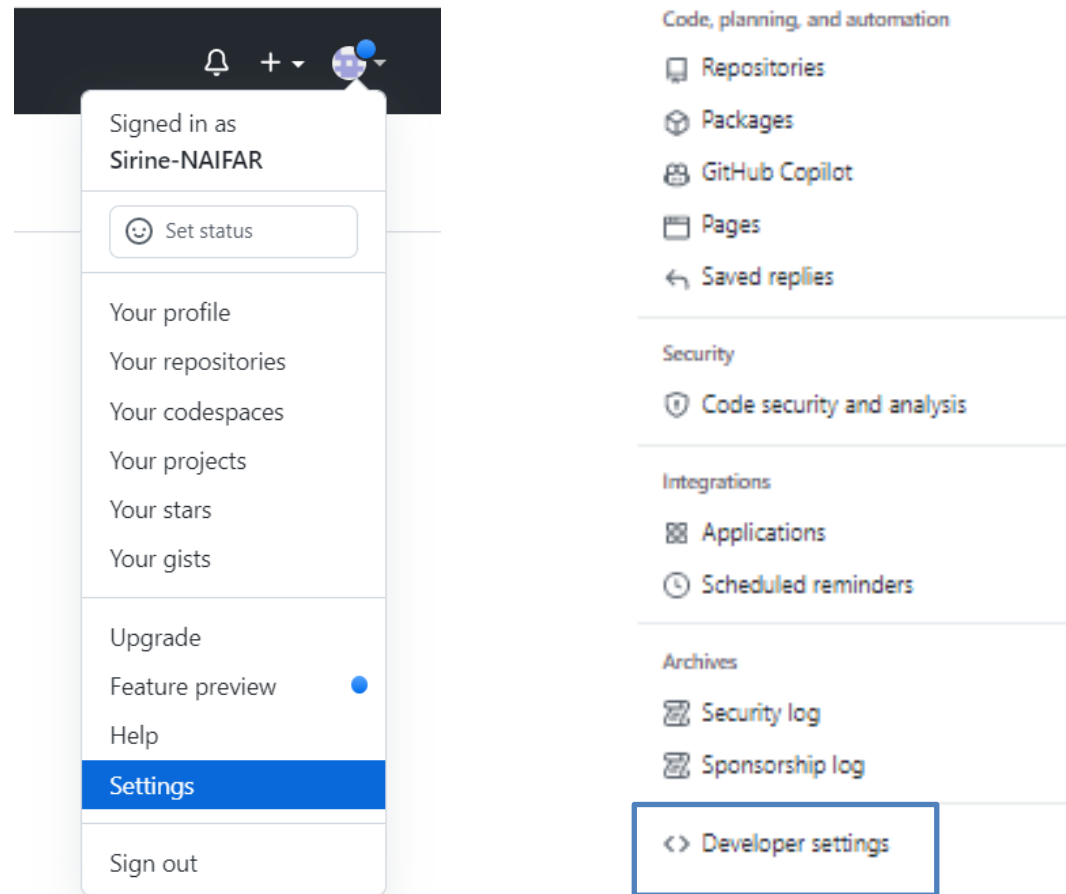
1/ Récupération du code

Il faut ajouter des clés SSH ou de définir des noms d'utilisateur et des mots de passe dans la partie « Identifiants » pour que le Jenkins soit capable de récupérer le code.



Configuration d'un projet avec Jenkins - FreeStyle

Il faut générer un token à partir du Git.



Configuration d'un projet avec Jenkins - FreeStyle

Tableau de bord > freestyle1 >

Configuration

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

repositories

Repository URL ?

https://github.com/Sirine-NAIFAR/DevOpsTest.git

Ajoutez un cerdentiel avec le token généré en tant que mot de passe

Credentials ?

naifar.sirine@gmail.com/******

+ Ajouter

Avancé...

Add Repository

Branches to build ?

Sauver Apply

Configuration d'un projet avec Jenkins - FreeStyle

GitHub Apps

OAuth Apps

Personal access tokens

Personal access tokens

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

jenkins — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, admin:ssh_signing_key, delete:packages, delete_repo, gist, notifications, project, repo, user, workflow, write:discussion, write:packages

Last used within the last 2 weeks

Delete

⚠ This token has no expiration date.

[Settings](#) / Developer settings

- GitHub Apps
- OAuth Apps
- Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

jenkins

What's this token for?

Expiration *

No expiration

The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

- ☐ repo Full control of private repositories
- ☐ repo:status Access commit status

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 15 minutes).

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 ?

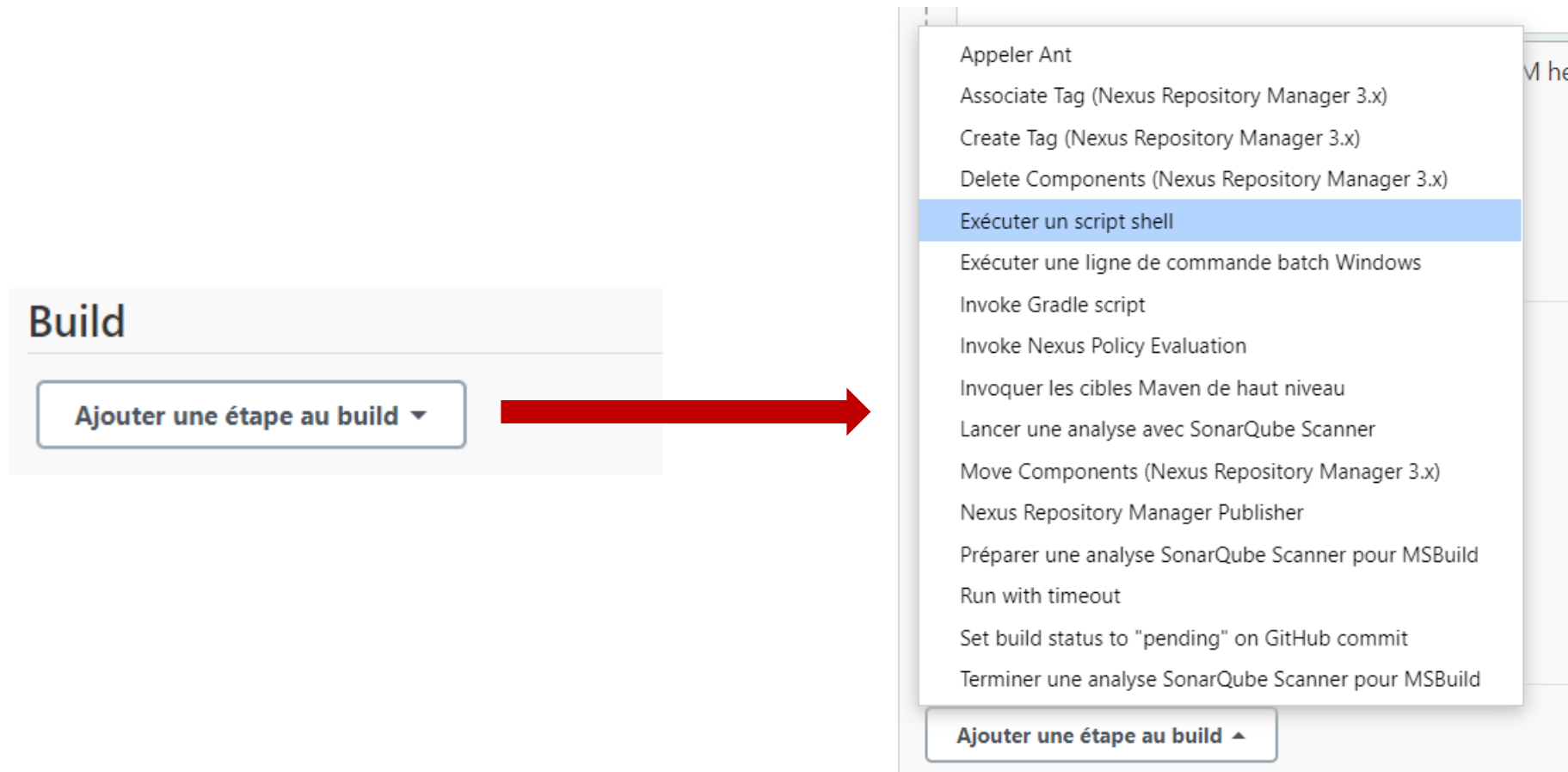
H/15 * * * *

Aurait été lancé à lundi 25 octobre 2021 14 h 23 WAT; prochaine exécution à lundi 25 octobre 2021 14 h 38 WAT.

Cliquer ici pour plus d'informations → ?

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.

☰ Exécuter un script shell ?

✕

Commande

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

```
echo "test from spring AOP"
date
```

Avancé...

Configuration d'un projet avec Jenkins - FreeStyle

Exemple Simple 02: Vérifie l'installation de maven.

☰ Exécuter un script shell ?

×

Commande

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

```
mvn -version
```

Avancé...

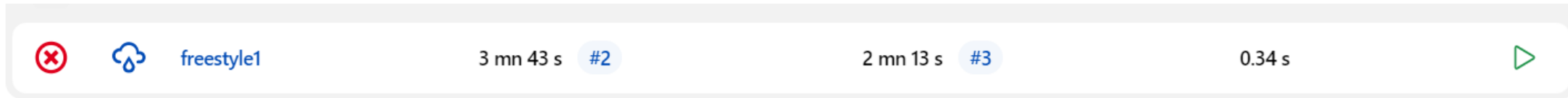
Configuration d'un projet avec Jenkins - FreeStyle

- 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. On the left sidebar, the 'Lancer un build' option is highlighted with a red rectangle. The main content area displays the project name 'Projet SpringAOP' and a list of actions: 'Retour au tableau de bord', 'État', 'Modifications', 'Répertoire de travail', 'Lancer un build', 'Configurer', 'Supprimer Projet', 'Log du dernier accès à Git', 'GitHub', 'SonarQube', and 'Rename'. To the right of the project name, there are links for 'SonarQube', 'Espace de travail', and 'Changements récents'. A 'Liens permanents' section is also present. In the top right corner, there are links to 'Ajouter une description' and a 'Désactiver le projet' button. At the bottom, there is a 'Historique des builds' section with a 'tendance' link and a search bar 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 interface for a project named 'Spring AOP'. The build status is 'Succès' (Success), indicated by a green checkmark icon. The build number is '#8', and it took '2 mn 13 s' to complete. The console output shows the following steps:

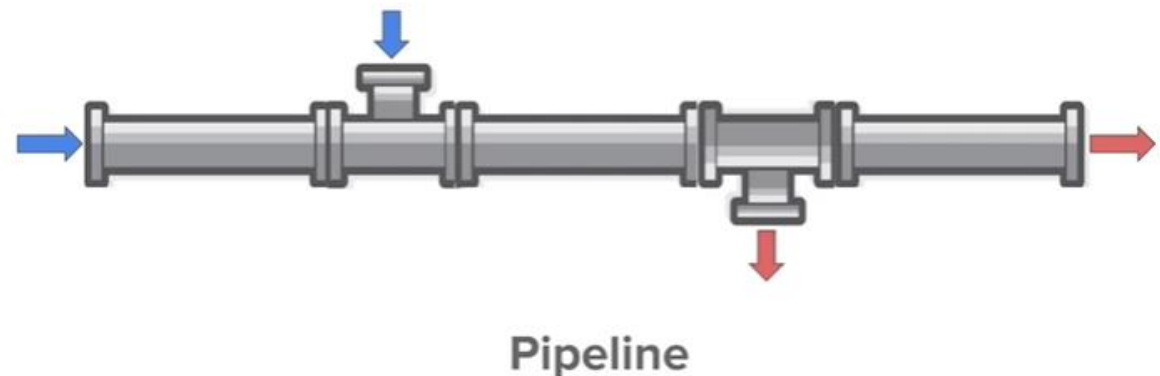
- Ignore 498 Ko. Log complet
- Uploading: <http://localhost:8081/repository/maven-releases/tn/esprit/spring/tp1-spring-mvc-rest/5.0/tp1-spring-mvc-rest-5.0.pom>
- Progress (1): 2.0/3.2 kB
- Progress (1): 3.2 kB
- Uploaded: <http://localhost:8081/repository/maven-releases/tn/esprit/spring/tp1-spring-mvc-rest/5.0/tp1-spring-mvc-rest-5.0.pom> (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)
- Downloaded: <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 console output shows 'REST API' and 'Jenkins 2.303.2'.

Configuration d'un projet avec Jenkins - Pipeline

- Le pipeline Jenkins est implémenté sous forme de code qui permet à plusieurs utilisateurs de modifier et d'exécuter le processus de pipeline.
- Pipeline donne une solution à cette problématique en offrant 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.

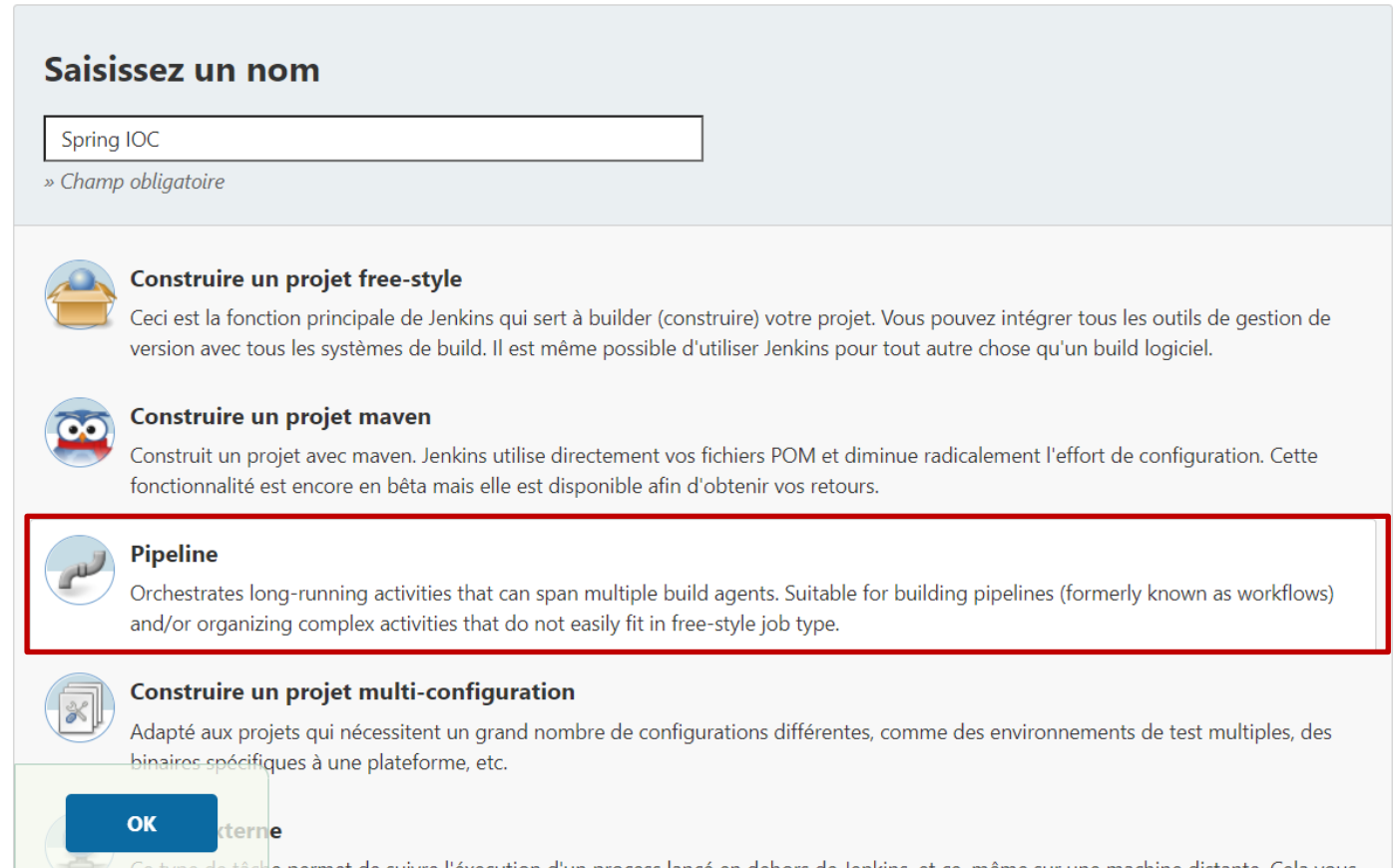
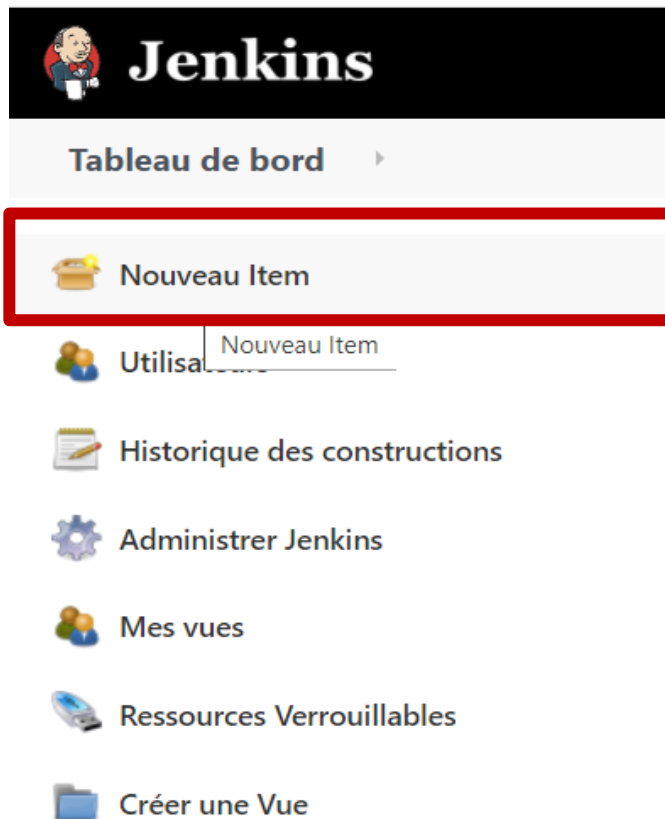
<https://www.jenkins.io/doc/book/pipeline/getting-started/>



Configuration d'un projet avec Jenkins - Pipeline

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

Au début, il faut créer un « projet » de type pipeline.



Configuration d'un projet avec Jenkins - Pipeline

- Il faut choisir l'action qui va lancer la chaine de l'intégration continue si Jenkins détecte un push dans le Git (process qui se lance chaque 5 minutes).

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

Planning

H/5 * * * *

Aurait été lancé à lundi 25 octobre 2021 18 h 00 WAT; prochaine exécution à lundi 25 octobre 2021 18 h 05 WAT.

- ☐ Ignore post-commit hooks

Cliquer ici pour plus d'informations



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 - Bonus

- Lancer un build à distance

Pour déclencher le job à distance (à travers un URL à saisir dans le navigateur)

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
- ☐ Désactiver le projet
- ☐ Période d'attente
- ☒ Déclencher les builds à distance (Par exemple, à partir de scripts)



Jeton d'authentification

token

Utilisez l'URL qui suit pour lancer un build à distance : `JENKINS_URL/job/Pipeline%20Complet%20Spring%20IOC/build?token=TOKEN_NAME` ou `/buildWithParameters?token=TOKEN_NAME`

Optionnellement, ajoutez le suffixe `&cause=Cause+Text` pour fournir du texte qui sera inclus dans la cause enregistrée pour le build.

<http://localhost:8080/job/Pipeline%20Complet%20Spring%20IOC/build?token=token>

Configuration d'un projet avec Jenkins - Bonus

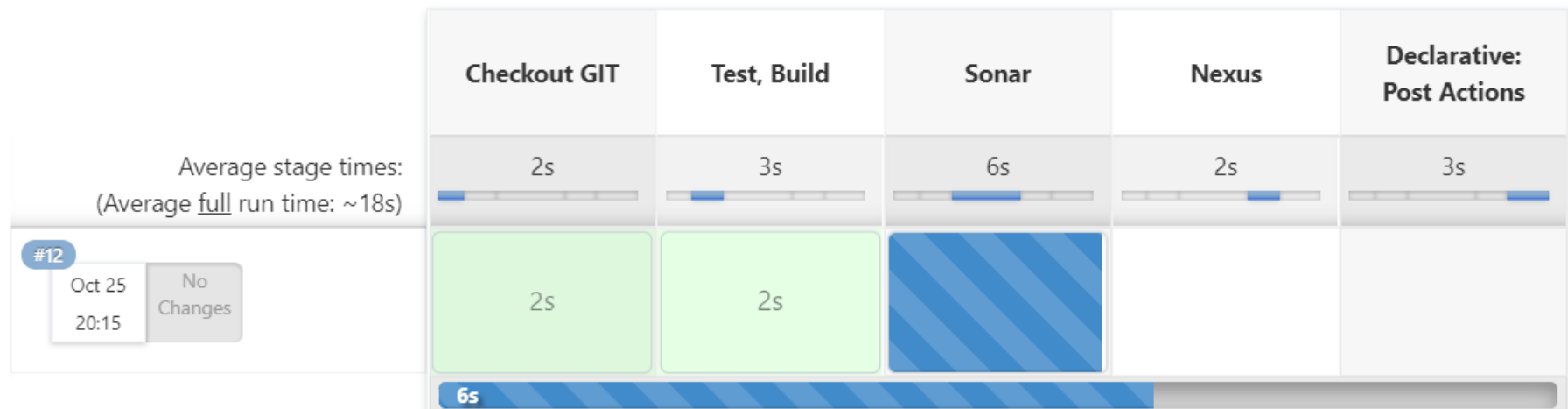
Dans le navigateur :

localhost:8080/job/Pipeline%20Compleet%20Spring%20IOC/build?token=token



Dans Jenkins :

Stage View



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 .
- Ce pipeline contient:
 - ✓ La récupération du code source.
 - ✓ La construction du livrable

Jenkins

Si vous avez des questions, n'hésitez pas à nous contacter :

Département Informatique
UP ASI

Bureau E204

Jenkins

