



FORMATION JENKINS - TP

Formation Jenkins, mettre en place l'intégration continue en Java

Plan des TP

- TP - Découverte de Jenkins
- TP MAVEN et GIT
- TP Intégration d'un projet dans JENKINS
- TP Utilisation des pipelines dans JENKINS
- TP Sonar et JENKINS
- TP Junit et JENKINS
- TP NEXUS et JENKINS

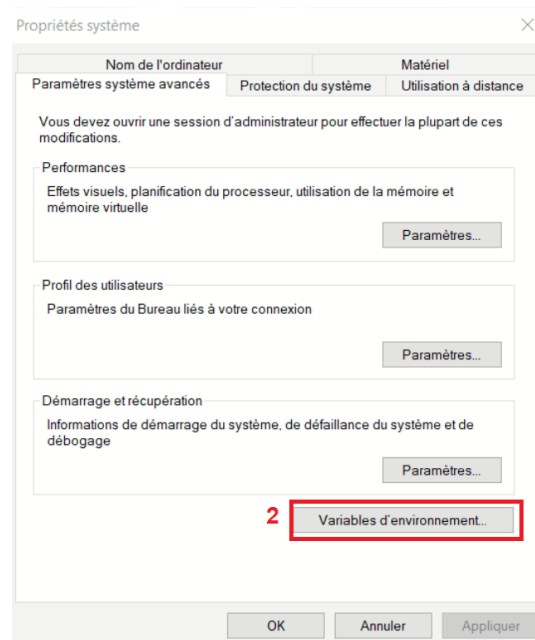
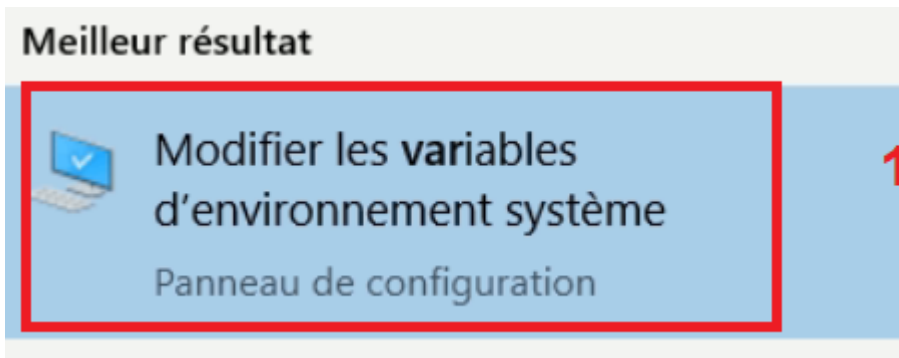


**LEARNING FROM
EXPERIENCE**

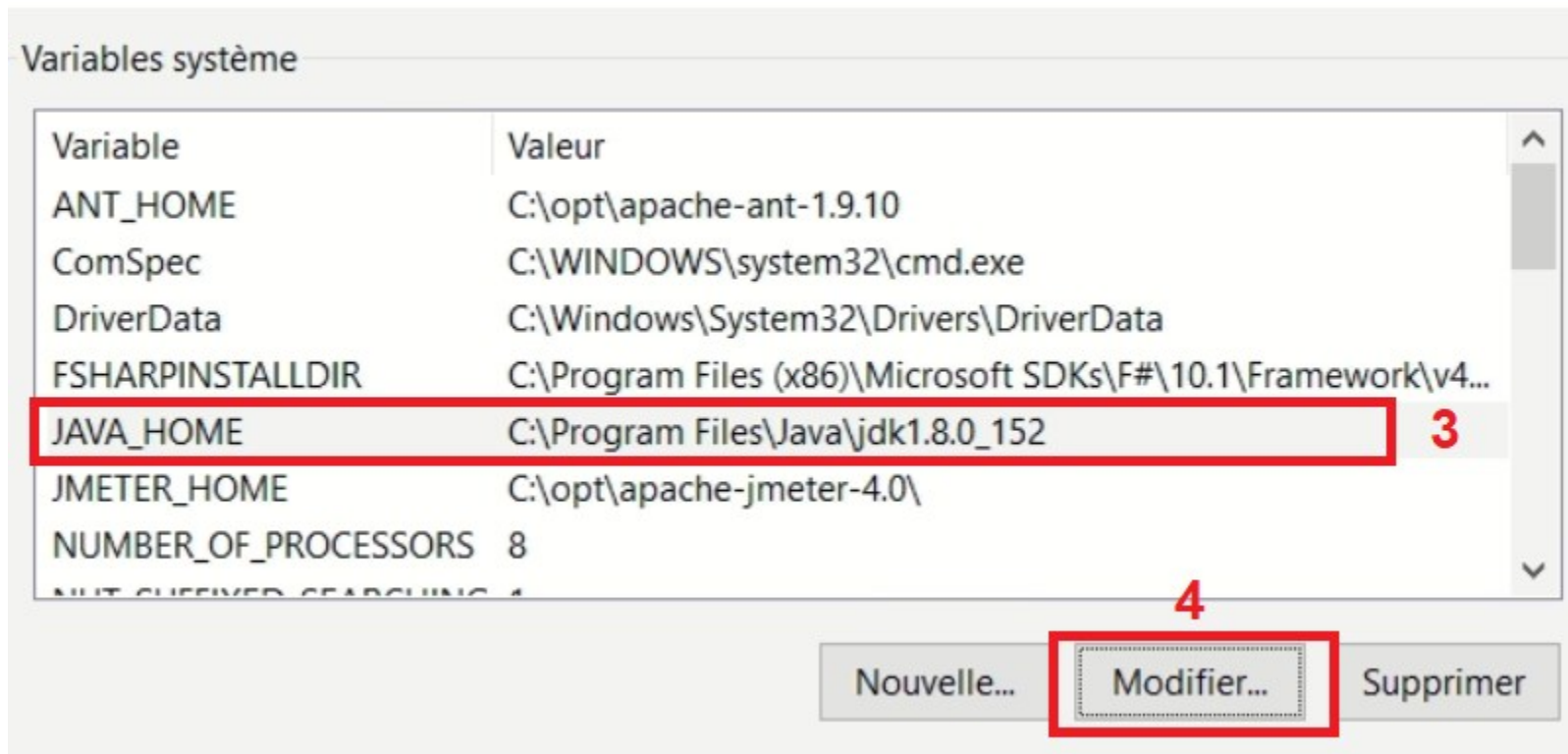
TP - DÉCOUVERTE DE JENKINS

PARAMETRAGE DE L'ENVIRONNEMENT

- Se connecter sur la VM
- Vérifier que le répertoire suivant existe:
C:\Program Files\Java\jdk-11.0.5
- Modifier les variables d'environnement : **JAVA_HOME**



PARAMETRAGE DE L'ENVIRONNEMENT



PARAMETRAGE DE L'ENVIRONNEMENT

Modifier la variable système

Nom de la variable : JAVA_HOME

Valeur de la variable : C:\Program Files\Java\jdk-11.0.5 **5**

Parcourir le répertoire... Parcourir le fichier... **6** OK Annuler

- Puis ouvrir un interpréteur de commandes **CLI** (Par exemple: cmd ou GIT BASH)
- Vérifier que votre JAVA_HOME point bien sur **Java 11**

```
echo %JAVA_HOME%
```

```
# Résultat attendu: C:\Program Files\Java\jdk-11.0.5
```

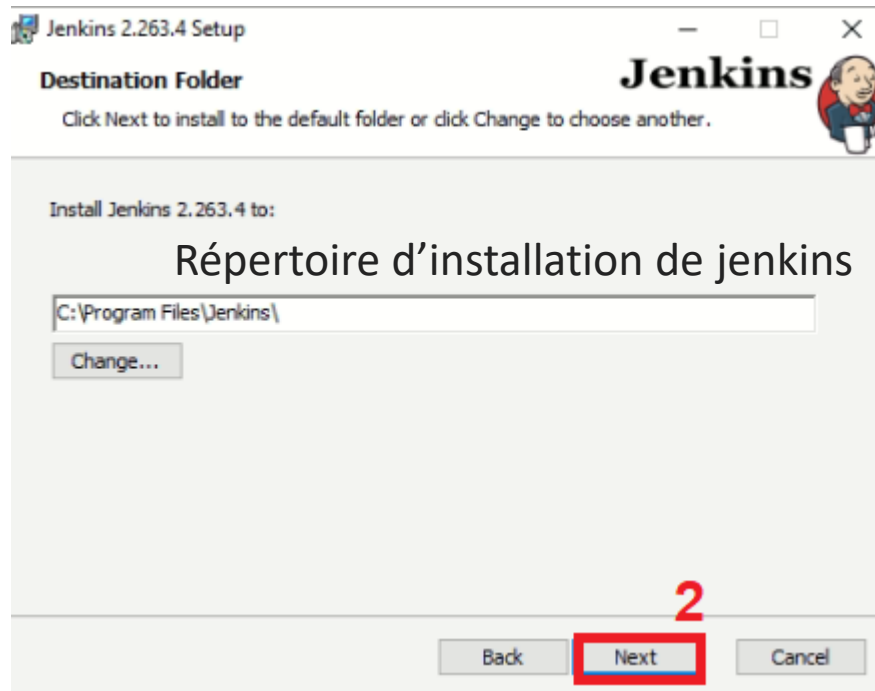
INSTALLATION DE JENKINS SOUS WINDOWS COMME **SERVICE**

- Télécharger l'exécutable jenkins depuis le site officiel:

<http://mirrors.jenkins-ci.org/windows-stable/latest>

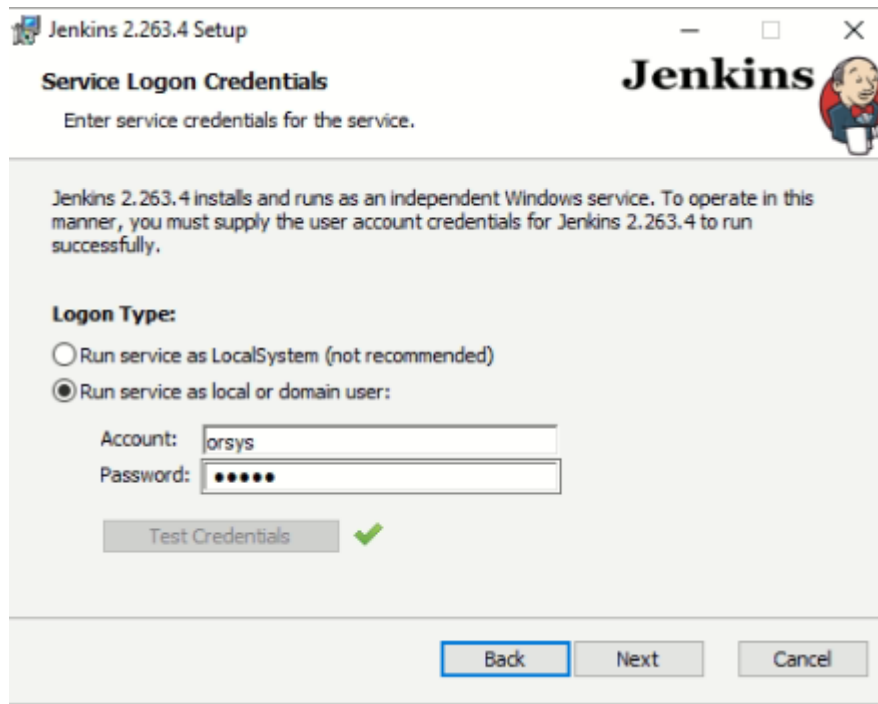
- Lancer l'exécutable: **jenkins.msi**
- Suivre les étapes d'installations

INSTALLATION DE JENKINS SOUS WINDOWS COMME SERVICE



INSTALLATION DE JENKINS SOUS WINDOWS COMME SERVICE

- Account: **orsys**
- Password: **orsys**




The screenshot shows the 'Jenkins 2.263.4 Setup' window. The title bar includes the Jenkins logo and the text 'Jenkins 2.263.4 Setup'. The main heading is 'Service Logon Credentials' with a sub-instruction 'Enter service credentials for the service.' and the Jenkins logo. A text block explains: 'Jenkins 2.263.4 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.263.4 to run successfully.'

Logon Type:

- ☐ Run service as LocalSystem (not recommended)
- ☒ Run service as local or domain user:

Account:

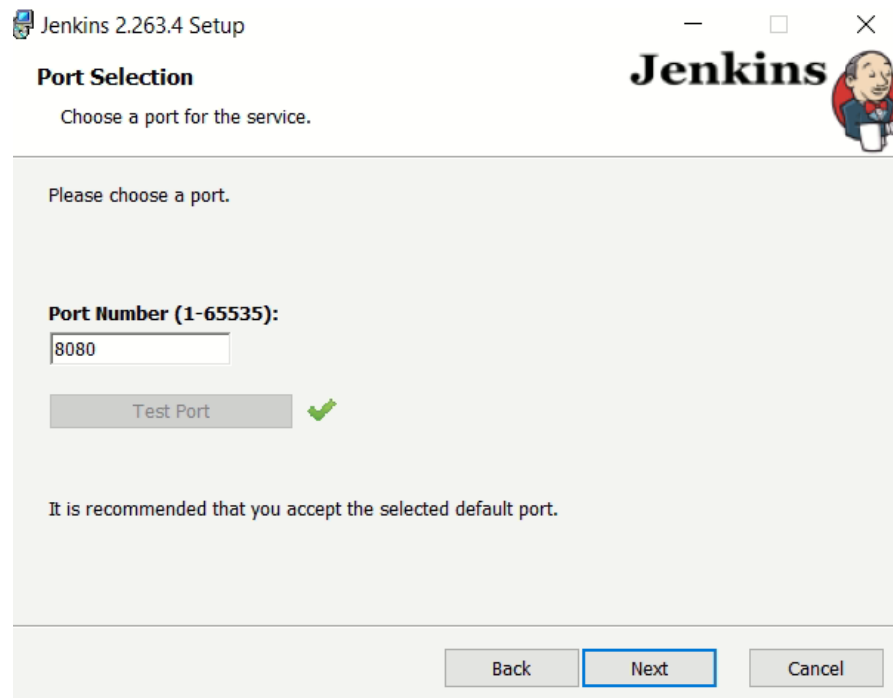
Password:

Test Credentials 

At the bottom are three buttons: 'Back' (highlighted with a blue border), 'Next', and 'Cancel'.

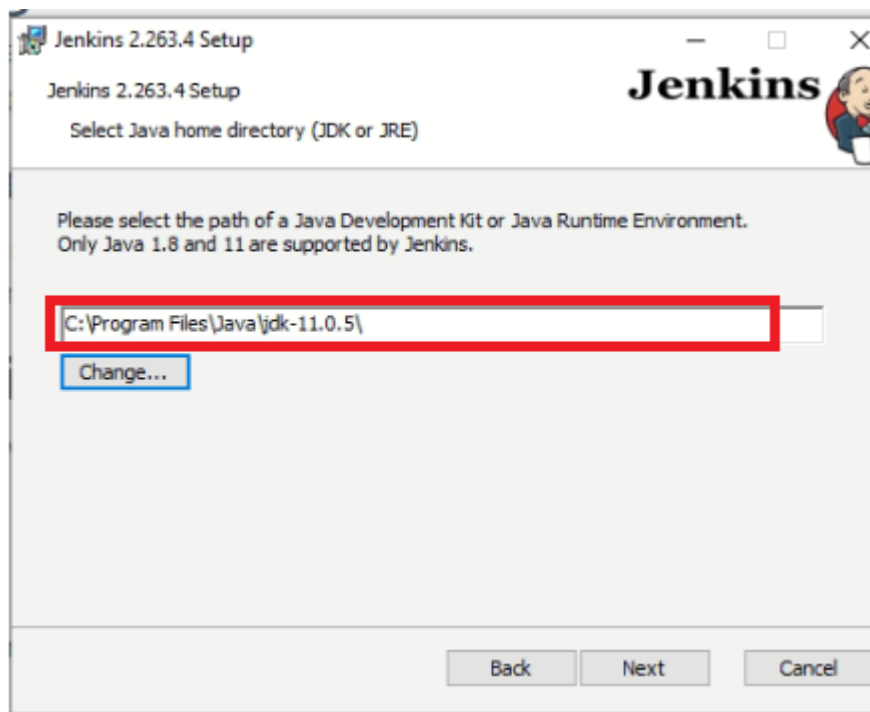
INSTALLATION DE JENKINS SOUS WINDOWS COMME SERVICE

- Port: **8080**



INSTALLATION DE JENKINS SOUS WINDOWS COMME SERVICE

- Utiliser le chemin de votre **JAVA_HOME** **C:\Program Files\Java\jdk-11.0.5**



JENKINS - INSTALLATION

- Tester l'accès à l'url
 - <http://127.0.0.1:8080>

Débloquer Jenkins

Pour être sûr que Jenkins soit configuré de façon sécurisée par un administrateur, un mot de passe a été généré dans le fichier de logs (où le trouver) ainsi que dans ce fichier sur le serveur :

```
C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\secrets\initialAdminPassword
```

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

Mot de passe administrateur

JENKINS - INSTALLATION

- Le fichier se trouve dans le répertoire

```
C:\Users\orsys\AppData\Local\Jenkins\.jenkins\secrets
```

- Se connecter avec le mot de passe se trouvant dans le fichier:
initialAdminPassword
 - Vous pouvez utiliser Notepad++, pour ouvrir le fichier

JENKINS - INSTALLATION

- Tester l'accès à l'url
 - Choisir les plugins à installer
 - suggérés

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.

JENKINS - INSTALLATION

- Personnaliser votre login/mdp administrateur (**admin/admin**)
 - **Attention à ne pas perdre le mdp jenkins**
- Se connecter à la console Jenkins

Créer le 1er utilisateur Administrateur

Nom d'utilisateur:	<input type="text" value="bnasslahsen"/>
Mot de passe:	<input type="password" value="••••••"/>
Confirmation du mot de passe:	<input type="password" value="••••••"/>
Nom complet:	<input type="text" value="nass lahsen"/>
Adresse courriel:	<input type="text" value="badr.nasslahsen@gmail.c"/>

INSTALLATION DE MAVEN

- Vérifier que vous disposez d'une installation maven:

```
mvn -version
```

- Résultat attendu:

```
Apache Maven 3.5.3 (3383c37e1f9e9b3bc3df5050c29c8aff9f295297; 2018-02-24T20:49:05+01:00)
```

```
Maven home: C:\opt\apache-maven-3.5.3\bin\..
```

```
Java version: 11.0.5, vendor: Oracle Corporation
```

```
Java home: C:\Program Files\Java\jdk-11.0.5
```

```
Default locale: fr_FR, platform encoding: Cp1252
```

```
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```


INSTALLATION DE GIT


- Verifier l'installation de GIT

```
git --version
```

- Résultat attendu:


```
git version 2.16.2.windows.1
```


CONFIGURATION GLOBALE DE JENKINS


**Jenkins**


Jenkins


1


 Nouveau Item


 Utilisateurs

 Historique des constructions

 **Administrer Jenkins** 2

 Mes vues

 Identifiants

 New View

File d'attente des constructions

File d'attente des constructions vide

État du lanceur de compilations

Administrer Jenkins

**Configurer le système**
Configurer les paramètres généraux et les chemins de fichiers.

**Configurer la sécurité globale**
Sécuriser Jenkins; définir qui est autorisé à accéder au système.

**Configure Credentials**
Configure the credential providers and types

**Configuration globale des outils** 3
Configurer les outils, leur localisation et les installateurs automatiques.

PARAMÉTRAGE DE JDK

JDK

Installations JDK

Ajouter JDK

JDK
Nom
JAVA_HOME
<input type="checkbox"/> Install automatically

Supprimer JDK

Ajouter JDK

Liste des installations JDK sur ce système

C:\Program Files\Java\jdk-11.0.5

PARAMÉTRAGE DE MAVEN

Maven

Installations Maven

Ajouter Maven

Maven

Nom

maven

MAVEN_HOME

C:\opt\apache-maven-3.5.3

☐ Install automatically



C:\opt\apache-maven-3.5.3

- Puis enregistrer vos modifications



**LEARNING FROM
EXPERIENCE**

TP PREPARATION D'UN PROJET AVEC MAVEN ET GIT

GIT - INTRODUCTION

- GIT est:
 - Un système de gestion de version (VCS en anglais pour Version Control System)
 - Conçu par Linus Torvalds (avril 2005). Utilisé pour la gestion de configuration du noyau Linux
 - Un système de gestion de version distribués (DVCS en anglais pour Distributed Version Control Systems)

GIT - INTRODUCTION - COMMANDES BASIQUES

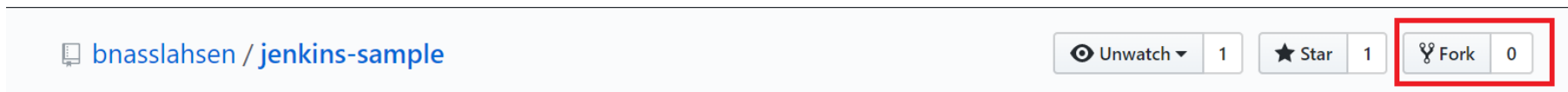
- Initialiser le workspace local:
 - `git init`
- Afficher l'historique des commits
 - `git log`
- Copier le repository distant en local :
 - `git clone <remote origin>>`
- Ajouter un fichier à l'index (stage)
 - `git add <file name>>`
- Supprimer un fichier du workspace
 - `git rm <<file name>>>`

GIT - INTRODUCTION - COMMANDES BASIQUES

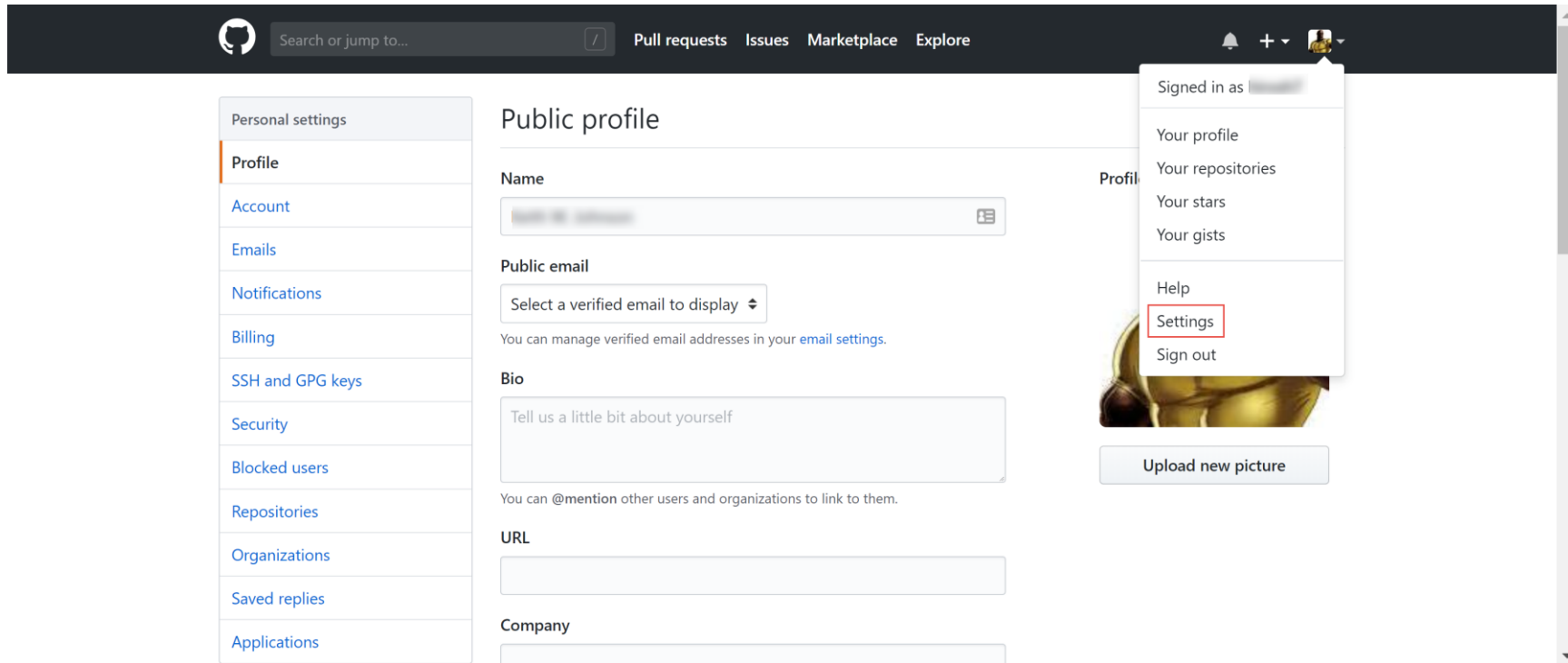
- Pour commit sur le repository local
 - `git commit -m "message"`
- Push au serveur
 - `git push [origin master]`
- Récupérer les changements depuis le repository distant et les appliquer au workspace local:
 - `git pull origin master`
- Annuler les modifications locales:
 - `git checkout -- <filename>`

CREATION DU PROJET SOUS GIT

- Créer un compte sous GITHUB :
 - <https://github.com/>
- Se connecter sur GITHUB
- Se rendre dans le repository GIT suivant:
 - <https://github.com/bnasslahsen/jenkins-sample.git>
- Créer un fork du repository suivant:



GÉNÉRATION D'UN ACCESS TOKEN



The screenshot shows the GitHub web interface. At the top is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. On the right of the bar are notification and profile icons. Below the bar, the left sidebar contains a 'Personal settings' menu with links for Profile, Account, Emails, Notifications, Billing, SSH and GPG keys, Security, Blocked users, Repositories, Organizations, Saved replies, and Applications. The main content area is titled 'Public profile' and contains form fields for Name, Public email, Bio, URL, and Company. A dropdown menu is open from the profile icon in the top right, showing options: 'Signed in as [username]', 'Your profile', 'Your repositories', 'Your stars', 'Your gists', 'Help', 'Settings' (highlighted with a red box), and 'Sign out'. Below the menu is a profile picture placeholder and an 'Upload new picture' button.

Personal settings

- Profile
- Account
- Emails
- Notifications
- Billing
- SSH and GPG keys
- Security
- Blocked users
- Repositories
- Organizations
- Saved replies
- Applications

Public profile

Name

Public email

Select a verified email to display

You can manage verified email addresses in your [email settings](#).

Bio

Tell us a little bit about yourself

You can @mention other users and organizations to link to them.

URL

Company

Signed in as [username]

- Your profile
- Your repositories
- Your stars
- Your gists
- Help
- Settings
- Sign out

Upload new picture

GÉNÉRATION D'UN ACCESS TOKEN

- Ensuite, dans la barre de menu de gauche, sélectionnez **Developer settings**
- Puis sélectionnez **Personal access tokens**
- Sur la page **Personal access tokens**, cliquez sur **Generate new token**.

GÉNÉRATION D'UN ACCESS TOKEN

[OAuth Apps](#)

[GitHub 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](#).

Token description ¹

What's this token for?

Select scopes ²

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

- | | |
|---|--|
| <input type="checkbox"/> repo | Full control of private repositories |
| <input type="checkbox"/> repo:status | Access commit status |
| <input checked="" type="checkbox"/> repo_deployment | Access deployment status |
| <input type="checkbox"/> public_repo | Access public repositories |
| <input type="checkbox"/> repo:invite | Access repository invitations |
| <hr/> | |
| <input type="checkbox"/> admin:org | Full control of orgs and teams |
| <input type="checkbox"/> write:org | Read and write org and team membership |
| <input type="checkbox"/> read:org | Read org and team membership |

GÉNÉRATION D'UN ACCESS TOKEN

- 1) Ajoutez une bonne description comme **Jenkins Dev Token**. La description est utile car vous aurez probablement beaucoup de ces jetons pour de nombreuses applications différentes
- 2) Sélectionnez les étendues d'autorisation. Ceux-ci sont particulièrement importants pour que l'intégration fonctionne avec succès. Pour les autorisations Jenkins, vous aurez besoin des éléments suivants :
 - repo (sélectionnez les meilleures options pour que toutes les sous-options soient également sélectionnées)
 - admin:org_hook
 - admin:repo_hook (sélectionnez les meilleures options pour que toutes les sous-options soient également sélectionnées)

GÉNÉRATION D'UN ACCESS TOKEN

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events

<input checked="" type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input checked="" type="checkbox"/> write:repo_hook	Write repository hooks
<input checked="" type="checkbox"/> read:repo_hook	Read repository hooks

<input checked="" type="checkbox"/> admin:org_hook	Full control of organization hooks
---	------------------------------------

GÉNÉRATION D'UN ACCESS TOKEN

- Une fois les mises à jour effectuées, cliquez sur **Generate token** et copiez votre nouveau token.

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

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_fWo3



Delete

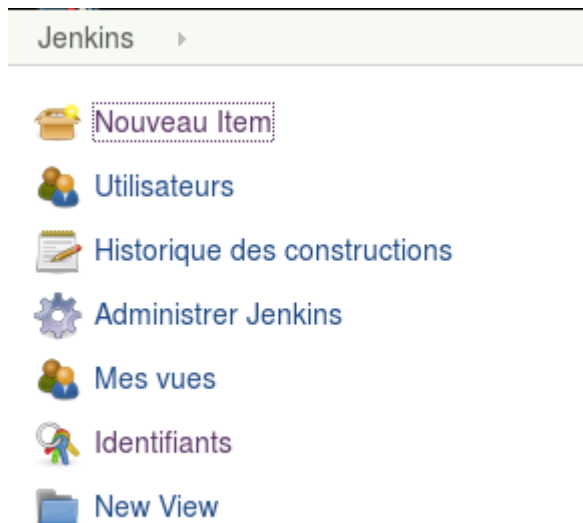


**LEARNING FROM
EXPERIENCE**

TP INTÉGRATION DU PROJET DANS JENKINS

CRÉATION DE JOB FREE STYLE SOUS JENKINS

- Dans le menu Jenkins en haut à gauche
- Cliquer sur nouveau Item



CRÉATION DE JOB FREE STYLE SOUS JENKINS

- Choisir construire un projet free-style
- Renseigner un nom de projet:
 - App-IC

Saisissez un nom

App-IC|

» Champ obligatoire



Construire un projet free-style

Ceci est la fonction principale de Jenkins qui sert à builder (construire) votre projet. Vous pouvez intégrer tout autre build logiciel dans Jenkins pour tout autre chose qu'un build logiciel.

CRÉATION DE JOB FREE STYLE SOUS JENKINS

- Sur la section gestion de code source, choisir GIT puis renseigner:
 - **Repository url**
 - credentials

Gestion de code source

☐ Aucune

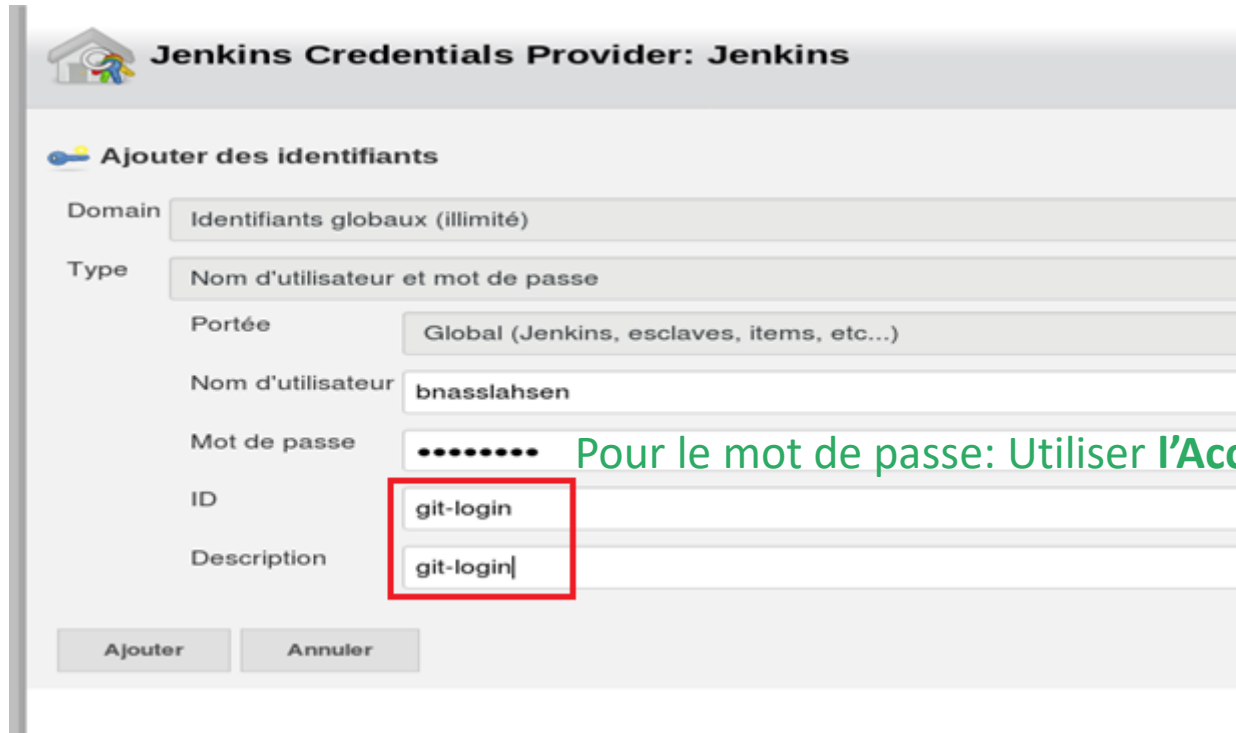
☒ Git

Repositories

Repository URL

Credentials

CRÉATION DE JOB FREE STYLE SOUS JENKINS



The screenshot shows the 'Jenkins Credentials Provider: Jenkins' interface. The main heading is 'Ajouter des identifiants'. The form fields are as follows:

- Domain: Identifiants globaux (illimité)
- Type: Nom d'utilisateur et mot de passe
- Portée: Global (Jenkins, esclaves, items, etc...)
- Nom d'utilisateur: bnasslahsen
- Mot de passe: (masked)
- ID: git-login
- Description: git-login

At the bottom, there are two buttons: 'Ajouter' and 'Annuler'.

Pour le mot de passe: Utiliser l'Access Token GITHUB

CRÉATION DE JOB FREE STYLE SOUS JENKINS

- Choisir votre login/mdp git

Gestion de code source

☐ Aucune

☒ Git

Repositories

Repository URL

Credentials [Ajouter](#)

[Avancé...](#)

[Add Repository](#)

Branches to build

Branch Specifier (blank for 'any')

[Add Branch](#)

Navigateur de la base de code

Additional Behaviours [Ajouter](#)

CRÉATION DE JOB FREE STYLE SOUS JENKINS

- Dans la section build,
 - Choisir invoquer les cibles maven de haut niveau



The screenshot shows the 'Build' section of the Jenkins job configuration. It features a header 'Build' and a sub-header 'Invoquer les cibles Maven de haut niveau'. Below this, there are two input fields: 'Version de Maven' with the value 'MAVEN-3' and 'Cibles Maven' with the value 'clean package'.

- A la fin cliquer sur Sauver
- Puis Lancer un build



The screenshot shows the 'Build' section of the Jenkins job configuration. It features a header 'Build' and a dropdown menu 'Ajouter une étape au build'. The dropdown menu is open, showing a list of build steps: 'Appeler Ant', 'Exécuter un script shell', 'Exécuter une ligne de commande batch Windows', 'Invoke Gradle script', 'Invoquer les cibles Maven de haut niveau' (which is highlighted with a dotted border), 'Run with timeout', and 'Set build status to "pending" on GitHub commit'. At the bottom of the dropdown, there are buttons for 'Sauver' and 'Annuler'.

CONFIGURATION: SERVEUR DE MAIL

1. Administrer Jenkins
2. Configurer le système
 - Vous pouvez utiliser votre mail pour tester la configuration
- Login: jenkins.orsys
- MDP: jenkins.orsys99
- Si vous avez une erreur de connexion, valider la connexion sur un autre onglet au compte:

Notification par email

Serveur SMTP	smtp.gmail.com
Suffixe par défaut des emails des utilisateurs	
<input checked="" type="checkbox"/> Use SMTP Authentication	
Nom d'utilisateur	jenkins.orsys
Mot de passe	*****
Utiliser SSL	<input checked="" type="checkbox"/>
Use TLS	<input type="checkbox"/>
Port SMTP	
Adresse de réponse	jenkins.orsys@gmail.Com
Jeu de caractères	UTF-8
<input checked="" type="checkbox"/> Tester la configuration en envoyant un e-mail de test	
Destinataire du courriel de test	jenkins.orsys@gmail.Com

Email was successfully sent

Tester la configuration

• <https://gmail.com/>

CONFIGURATION DES NOTIFICATIONS PAR MAIL

- Aller dans le job

Actions à la suite du build

Notifier par email

Destinataires

Liste des destinataires, séparés par un espace. Un email sera envoyé lors d'un échec d'un build.

- ☒ Envoyer un email pour chaque build instable
- ☐ Envoyer des emails séparés aux personnes qui ont cassé le build

Ajouter une action après le build ▼

CRÉATION DE JOB FREE STYLE SOUS JENKINS

- Afin de programmer un build toutes les 45 min,
- Aller dans la section build trigger
- Renseigner la conf dans planning: H/45 * * * *

Ce qui déclenche le build

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

☐ Construire après le build sur d'autres projets

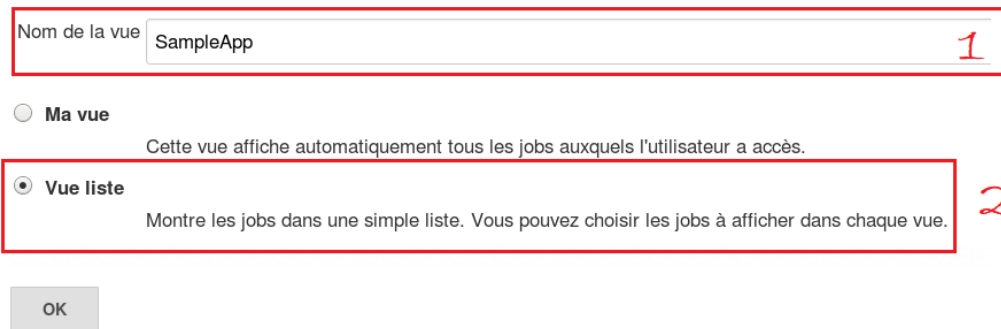
☒ Construire périodiquement

Planning

H/45 * * * *

UTILISATION DES VUES

- Créer une vue SampleApp

A screenshot of the 'Create New View' dialog box in Jenkins. The dialog has a title bar 'Jenkins' and a main area with two sections. The first section, labeled 'Nom de la vue', contains a text input field with the value 'SampleApp'. The second section has two radio button options: 'Ma vue' and 'Vue liste'. The 'Vue liste' option is selected. Below the options, there is an 'OK' button. The 'SampleApp' input field and the 'Vue liste' option are both highlighted with red rectangular boxes. A red handwritten '1' is next to the input field, and a red handwritten '2' is next to the 'Vue liste' option.

UTILISATION DES VUES

- Ajouter à cette vue le job: App-IC

Nom: SampleApp

Description:

[Plain text] [Prévisualisation](#)

Filtrer la file d'attente de constructions: ☐

Filtrer les lanceurs de constructions: ☐

Filtres pour les tâches

État du filtre: Toutes les tâches sélectionnées

Rechercher récursivement dans les sous répertoires: ☐

Tâches: ☒ APP-IC

☐ Utilisez une expression régulière pour inclure les jobs dans la vue

Ok Apply



**LEARNING FROM
EXPERIENCE**

TP UTILISATION DES PIPELINES

CONVERTIR LE JOB FREE-STYLE EN PIPELINE

- Aller sur le menu jenkins de gauche,
 1. Choisir Administrer jenkins
 2. Gestion des plugins

Jenkins

- Nouveau Item
- Utilisateurs
- Historique des constructions
- Relations entre les builds
- Vérifier les empreintes numériques
- Administrer Jenkins** 1
- Mes vues
- Identifiants
- New View

File d'attente des constructions -

File d'attente des constructions vide

État du lanceur de compilations -

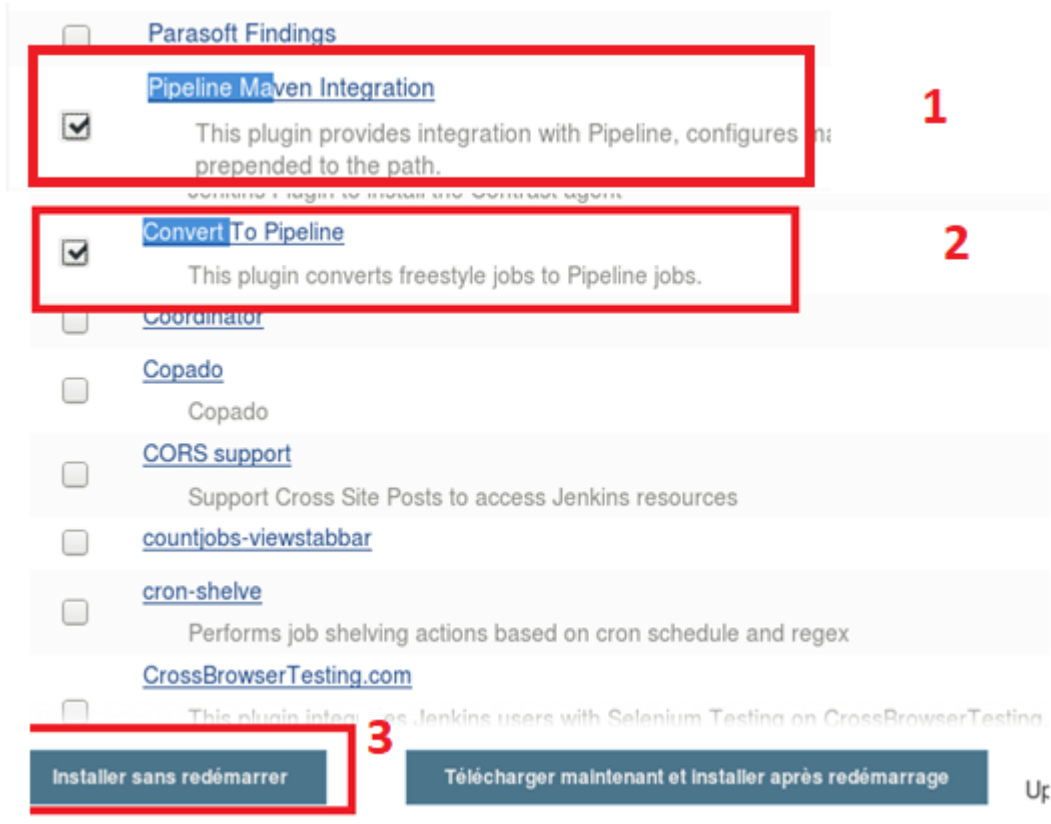
- 1 Au repos
- 2 Au repos

Administrer Jenkins

- Configurer le système**
Configurer les paramètres généraux et les chemins de fichiers.
- Configurer la sécurité globale**
Sécuriser Jenkins; définir qui est autorisé à accéder au système.
- Configure Credentials**
Configure the credential providers and types
- Configuration globale des outils**
Configurer les outils, leur localisation et les installateurs automatiques.
- Recharger la configuration à partir du disque**
Supprimer toutes les données en mémoire et recharger tout à partir du système de fichiers. Utile quand v
- Gestion des plugins** 2
Ajouter, supprimer, activer ou désactiver des plugins qui peuvent étendre les fonctionnalités de Jenkins.

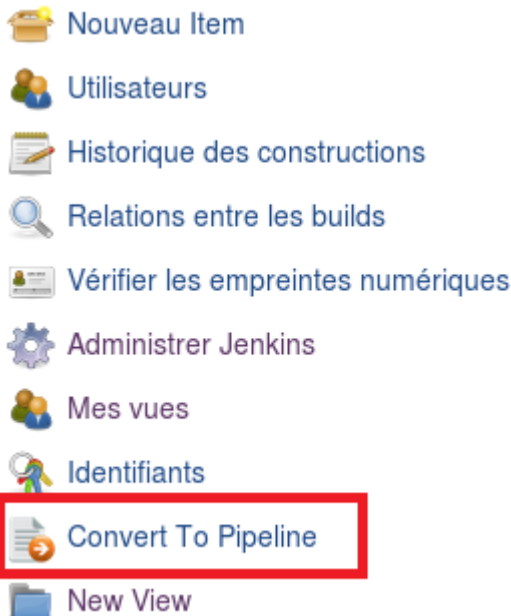
CONVERTIR LE JOB FREE-STYLE EN PIPELINE

- Aller dans l'onglet disponibles
- Choisir
 - Pipeline maven integration
 - convert to Pipeline
- Installer sans redémarrer



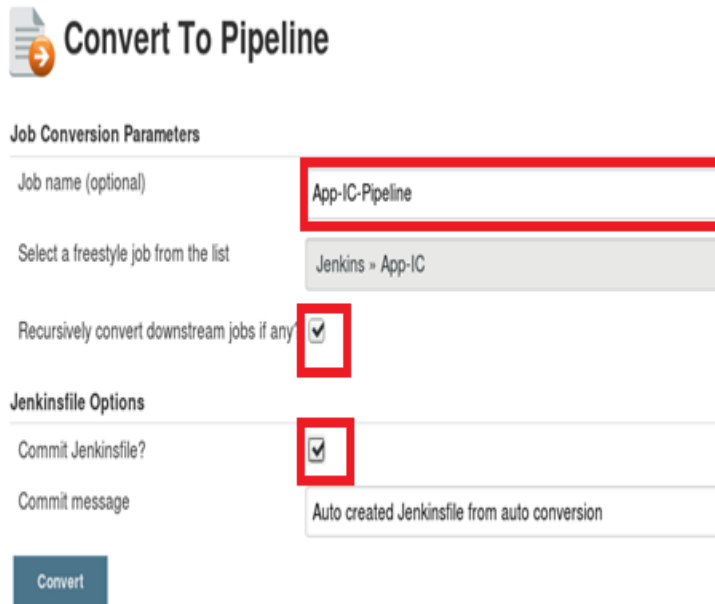
CONVERTIR LE JOB FREE-STYLE EN PIPELINE


- Aller sur le job App-IC
- Choisir Convert To Pipeline



CONVERTIR LE JOB FREE-STYLE EN PIPELINE

- Renseigner les paramètres de conversion
- Cliquer sur convert



 **Convert To Pipeline**

Job Conversion Parameters

Job name (optional)

Select a freestyle job from the list Jenkins » App-IC

Recursively convert downstream jobs if any ☒

Jenkinsfile Options

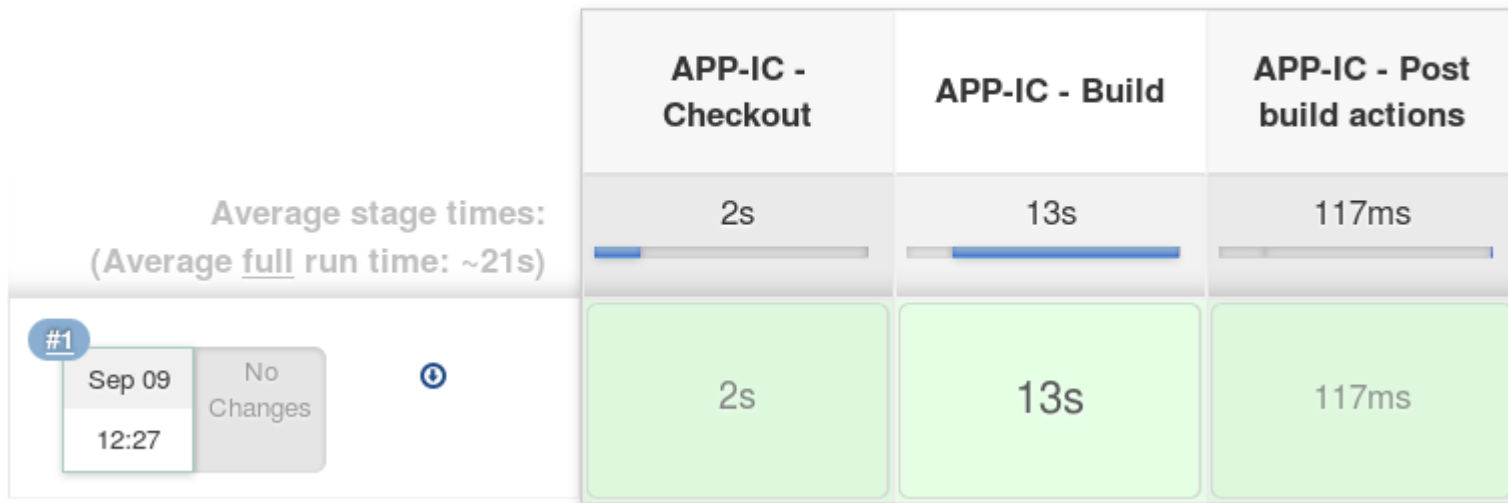
Commit Jenkinsfile? ☒

Commit message

CONVERTIR LE JOB FREE-STYLE EN PIPELINE

- Lancer le job: App-Ic-Pipeline

Stage View



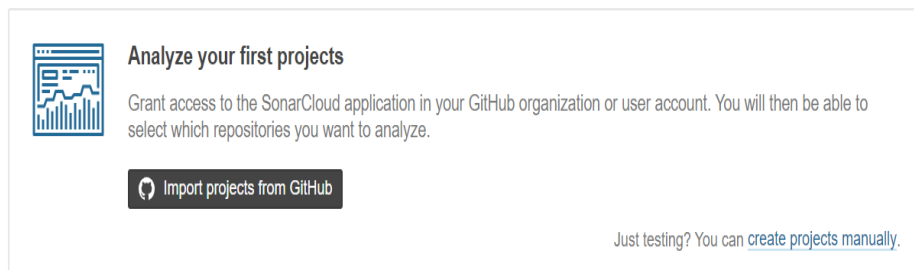


**LEARNING FROM
EXPERIENCE**

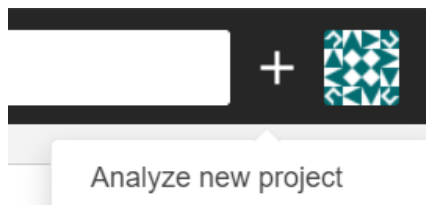
TP SONAR ET JENKINS

CONFIGURATION DE SONAR

- Se connecter avec votre compte GITHUB dans sonarcloud:
 - <https://sonarcloud.io/>
- Choisir: Import projects from GITHUB




- Ou bien, pour les gens qui ont déjà des projets sur sonarcloud:



CONFIGURATION DE SONAR

- Choisir le repository à analyser

☐ Select all available repositories

☒  jenkins-sample

1

☐  spring-boot

Don't see your repo? Check your [GitHub app configuration](#).

1 repository selected

1 repository will be created as a public project on SonarCloud

Set Up


2

CONFIGURATION DE SONAR


- Choisir le mode manuel


Configure Issues Security Hotspots Measures Code Activity Administration ▼

Choose your Analysis Method




With GitHub Actions >

 Recommended



With Travis CI >



With CircleCI >

With other CI tools >

SonarCloud integrates with your workflow no matter which CI tool you're using.

Manually >

Use this for testing. Other modes are recommended to help you set up your CI environment.

CONFIGURATION DE SONAR

- Choisir Maven, puis garder ces informations pour paramétrer Jenkins

Configure Issues Security Hotspots Measures Code Activity Administration ▼

Analyze from your local sources

Run analysis on your project

What option best describes your build?

1 **Maven** Gradle .NET C, C++ or ObjC Other (for JS, TS, Go, Python, PHP, ...)

Configure the SONAR_TOKEN environment variable

1 Name of the environment variable: SONAR_TOKEN ⓘ

2 Value of the environment variable 6b4704b3757bd7c381490dfb6acf5db03a093c20 ⓘ ✎

2 - le token qui sera utilisé dans jenkins

Execute the SonarScanner for Maven from your computer

Update your pom.xml file with the following properties:

CONFIGURATION DE SONAR

- Récupérer la valeur de **sonar.projectKey** pour la suite:

Run the following command in the project folder:

```
mvn verify org.sonarsource.scanner.maven:sonar-maven-plugin:sonar -Dsonar.projectKey=bnasslahsen_jenkins-sample
```

- Conserver la configuration de votre projet:

Update your `pom.xml` file with the following properties:

```
<properties>
  <sonar.organization>bnasslahsen</sonar.organization>
  <sonar.host.url>https://sonarcloud.io</sonar.host.url>
</properties>
```

UTILISATION D'UN CONTEXTE SONAR

- Aller sur le menu jenkins de gauche,
 - Choisir Administrer jenkins
 - Gestion des plugins

Jenkins

- Nouveau Item
- Utilisateurs
- Historique des constructions
- Relations entre les builds
- Vérifier les empreintes numériques
- Administrer Jenkins** 1
- Mes vues
- Identifiants
- New View

File d'attente des constructions -

File d'attente des constructions vide

État du lanceur de compilations -

1 Au repos

2 Au repos

Administrer Jenkins

- Configurer le système**
Configurer les paramètres généraux et les chemins de fichiers.
- Configurer la sécurité globale**
Sécuriser Jenkins; définir qui est autorisé à accéder au système.
- Configure Credentials**
Configure the credential providers and types
- Configuration globale des outils**
Configurer les outils, leur localisation et les installateurs automatiques.
- Recharger la configuration à partir du disque**
Supprimer toutes les données en mémoire et recharger tout à partir du système de fichiers. Utile quand v
- Gestion des plugins** 2
Ajouter, supprimer, activer ou désactiver des plugins qui peuvent étendre les fonctionnalités de Jenkins.

INSTALLATION DU PLUGIN SONAR SOUS JENKINS

- Aller dans l'onglet disponibles
- Choisir SonarQube scanner
- Installer sans redémarrer

Fortran, SQL, Ruby, Python, etc...

<input type="checkbox"/>	SonarQube Scanner	2.7.1
<input type="checkbox"/>	Sonargraph Integration	2.1.2
<input type="checkbox"/>	Sonargraph	1.6.4
<input type="checkbox"/>	Splunk	
<input type="checkbox"/>	Splunk plugin for Jenkins provides deep insights into your Jenkins master and slave infrastructure, job and build details such as console logs, status, artifacts, and an incredibly efficient way to analyze test results.	1.6.4
<input type="checkbox"/>	Statistics Gatherer	2.0.2
<input type="checkbox"/>	Captures Statistics related to Jenkins Builds, Build Step, SCM checkouts, Jobs and Queue and sends them where you want.	
<input type="checkbox"/>	StepCounter	2.0.0
<input type="checkbox"/>	Summary Display	1.15
<input type="checkbox"/>	Extend build and project page with extended results in XML format	

TAB

Installer sans redémarrer

Télécharger maintenant et installer après redémarrage

Update information obtained: 2 h 41 mn ago

Vérifier maintenant

CONFIGURATION DU PLUGIN SONAR SOUS JENKINS

- Aller sur le menu Jenkins ► Administrer Jenkins ► Configurer le Système
- Aller à la section "SonarQube servers"
- Cliquer sur "Add SonarQube"
- Renseigner les informations, comme indiqué sur l'image:
 - Nom: Sonar
 - URL: https://sonarcloud.io

SonarQube servers

Environment variables

☒ Enable injection of SonarQube server configuration as build environment variables

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Installations de SonarQube

Nom

Sonar

URL du serveur

https://sonarcloud.io

Par défaut à http://localhost:9000

Server authentication token

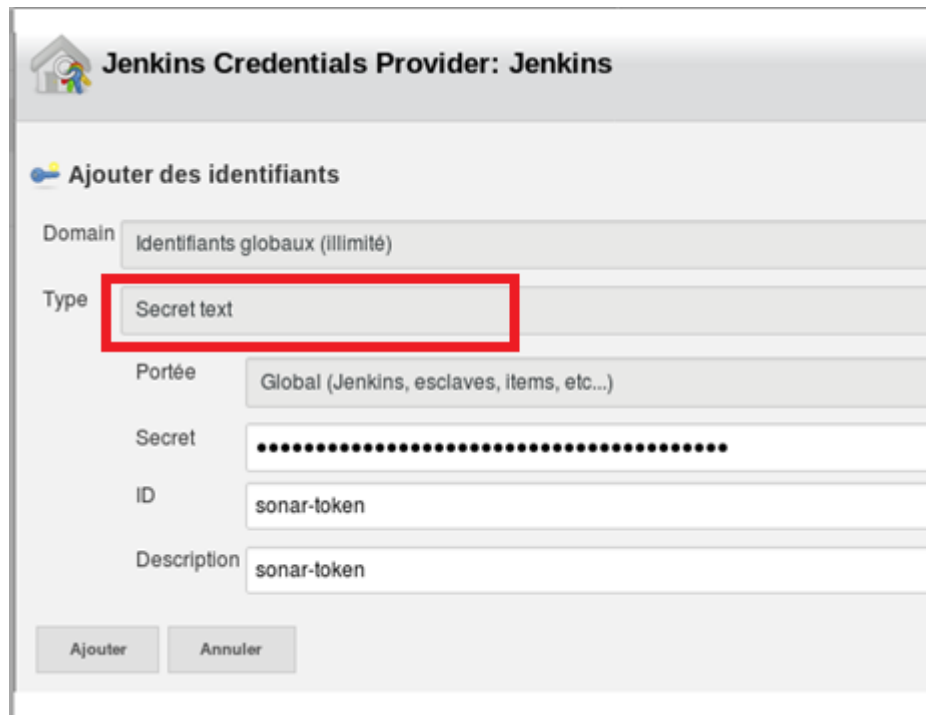
sonar-token ▼

Ajouter ▼

SonarQube authentication token. Mandatory when anonymous access is disabled.

CONFIGURATION DU PLUGIN SONAR SOUS JENKINS

- Configurer le token d'authentification

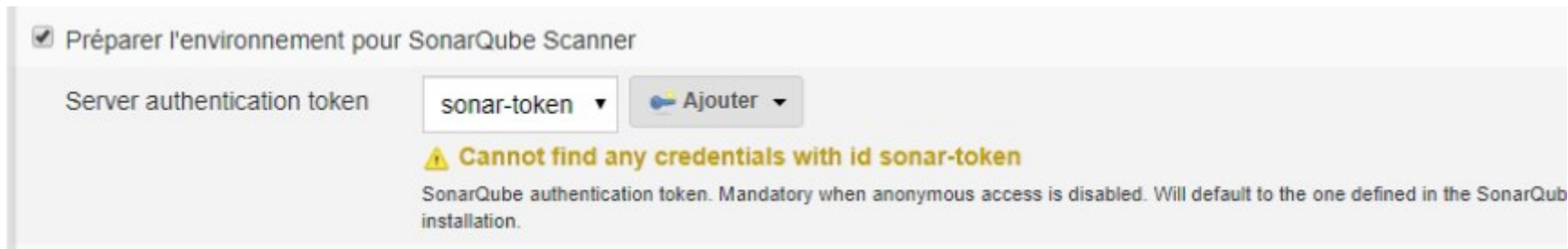


The screenshot shows the 'Jenkins Credentials Provider: Jenkins' configuration page. The 'Ajouter des identifiants' section is active. The 'Domain' field is set to 'Identifiants globaux (illimité)'. The 'Type' field is highlighted with a red box and contains the text 'Secret text'. The 'Portée' field is set to 'Global (Jenkins, esclaves, Items, etc...)'. The 'Secret' field is filled with dots. The 'ID' field contains the text 'sonar-token'. The 'Description' field contains the text 'sonar-token'. At the bottom, there are two buttons: 'Ajouter' and 'Annuler'.


Field	Value
Domain	Identifiants globaux (illimité)
Type	Secret text
Portée	Global (Jenkins, esclaves, Items, etc...)
Secret
ID	sonar-token
Description	sonar-token

UTILISATION DE SONAR DANS UN PROJET FREE STYLE

- Aller dans la section : Environnements de Build
- Cocher la case préparer l'environnement pour SonarQube Scanner
- Choisir le token sonar, paramétré dans les étapes précédentes



☒ Préparer l'environnement pour SonarQube Scanner

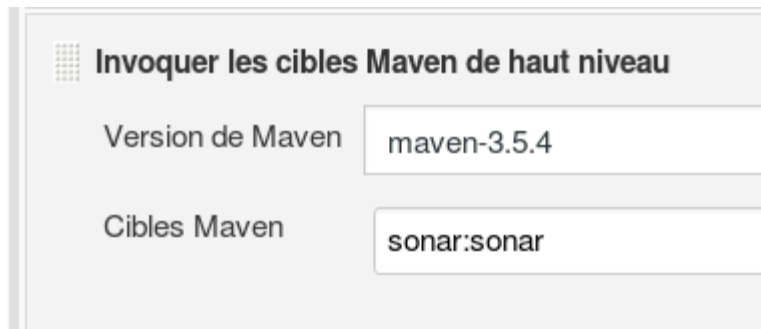
Server authentication token: sonar-token 

⚠ Cannot find any credentials with id sonar-token

SonarQube authentication token. Mandatory when anonymous access is disabled. Will default to the one defined in the SonarQube installation.

UTILISATION DE SONAR DANS UN PROJET FREE STYLE

- Ajouter une étape de build Maven au projet APP-IC.
 - Elle invoque les cibles maven suivante : **sonar:sonar**



The screenshot shows a configuration window titled "Invoquer les cibles Maven de haut niveau". It contains two input fields: "Version de Maven" with the value "maven-3.5.4" and "Cibles Maven" with the value "sonar:sonar".

Invoquer les cibles Maven de haut niveau	
Version de Maven	maven-3.5.4
Cibles Maven	sonar:sonar

UTILISATION DE SONAR DANS UN PROJET FREE STYLE

- Editer le fichier pom.xml pour ajouter la configuration fournie par sonarcloud:
 - la balise parente est <project>, dans le fichier pom.xml
 - N'oublier pas de rajouter la propriété: **sonar.projectKey**

```
<properties>
<sonar.projectKey>          .jenkins-sample</sonar.projectKey>
<sonar.organization>        </sonar.organization>
<sonar.host.url>https://sonarcloud.io</sonar.host.url>
</properties>
```

- Lancer le build du projet

ANALYSE SONAR POUR LES PROJETS AVEC PIPELINE

- Ajouter le bloc suivante au niveau du pipeline du projet

```
stage('Quality check') {  
    withSonarQubeEnv('Sonar') {  
        bat "mvn sonar:sonar"  
    }  
}
```

- Identifier les violations les plus critiques
- La configuration du profil d'analyses SonarQube peut être évoquée mais ce n'est pas l'objet du TP
- Définir un plan d'action



**LEARNING FROM
EXPERIENCE**

TP JUNIT ET JENKINS

TESTS: APPLICATION AU PROJET SAMPLE-APP

- Créer un test junit
- Consulter l' exécution du test avec Maven sur la console jenkins
- Relancer un Build sous Jenkins et consulter les rapports de tests
- Pour aller plus loin, vous pouvez consulter la configuration du plugin jacoco au niveau configuration du pom.xml du projet ddd-sample:
 - <https://github.com/bnasslahsen/ddd-sample.git>



**LEARNING FROM
EXPERIENCE**

TP AUTOMATISATION DU DÉPLOIEMENT

UTILISATION DU PLUGIN NEXUS



- Aller dans l'onglet disponibles
- Choisir
 - **Nexus platform**
 - **Maven integration**
- Installer sans redémarrer

Maven Integration		
<input checked="" type="checkbox"/>	This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTS, automated configuration of various Jenkins publishers (JUnit, ...).	3.15.1
Désinstaller		
Nexus Platform		
<input checked="" type="checkbox"/>	This plugin integrates Sonatype Nexus to Jenkins.	3.13.20211118-115856.c81882e
Désinstaller		

UTILISATION DU PLUGIN NEXUS

- Aller sur le menu Jenkins ► Administrer Jenkins ► Configurer le Système
- Aller à la section "Sonatype Nexus"
- Cliquer sur « Nexus Repository Manager Servers »
- Renseigner les informations, comme indiqué sur l'image:
 - Login/mdp nexus (voir page suivante)
 - url: <http://138.3.244.133:8081/>

Nexus Repository Manager 3.x Server

Display Name	<input type="text" value="nexus"/>	
Server ID	<input type="text" value="nexus"/>	
Server URL	<input type="text" value="http://130.61.57.189:8081"/>	
Credentials	<input type="text" value="admin/***** (nexus-login)"/>	 Ajouter ▼

Nexus Repository Manager 3.x connection succeeded

Test connection



NXRM OSS 3.30.0-01 found. Some operations require Nexus Repository Manager Professional server version 3.13.0 or newer; use of an incompatible server could result in failed builds.

UTILISATION DU PLUGIN NEXUS

- Le mot de passe est le suivant:
 - 1c5d93e9-b36f-4746-81dc-a7027eb6ec69
 - Ne pas changer le mot de passe Nexus!



The screenshot shows the 'Jenkins Credentials Provider: Jenkins' configuration page. It features a header with a house icon and the title. Below the header is a section titled 'Ajouter des identifiants' with a key icon. The form contains several fields: 'Domain' set to 'Identifiants globaux (illimité)', 'Type' set to 'Nom d'utilisateur et mot de passe', 'Portée' set to 'Global (Jenkins, esclaves, items, etc...)', 'Nom d'utilisateur' set to 'admin', 'Mot de passe' represented by dots, 'ID' set to 'nexus-login', and 'Description' set to 'nexus-login'. At the bottom are 'Ajouter' and 'Annuler' buttons.

Jenkins Credentials Provider: Jenkins

Ajouter des identifiants

Domain: Identifiants globaux (illimité)

Type: Nom d'utilisateur et mot de passe

Portée: Global (Jenkins, esclaves, items, etc...)

Nom d'utilisateur: admin

Mot de passe:

ID: nexus-login

Description: nexus-login

Ajouter Annuler

CONFIGURATION DU GIT PLUGIN

Git plugin

Global Config user.name Value

username GITHUB

Global Config user.email Value

mail GITHUB

☐ Create new accounts based on author/committer's email

UTILISATION DU PLUGIN NEXUS

- Enregistrer les modifications

Sonatype Nexus

Nexus Repository Manager Servers

Nexus Repository Manager 3.x Server

Display Name:

Server ID:

Server URL:

Credentials: Ajouter

Nexus Repository Manager 3.x connection succeeded

⚠ NXRM OSS 3.30.0-01 found. Some operations require Nexus Repository Manager Professional server version 3.13.0 or newer; use of an incompatible server could result in failed builds.

Test connection Supprimer

Add Nexus Repository Manager Server

Nexus IQ Server

Add Nexus IQ Server

Build-timeout Plugin > BuildStep Action

☐ Enable BuildStep Action

Git plugin

Global Config user.name Value:

Global Config user.email Value:

☐ Create new accounts based on author/committer's email

☐ Show the entire commit summary in changes

☐ Hide credential usage in job output

☐ Disable performance enhancements

☐ Preserve second fetch during checkout

☐ Add git tag action to jobs

Chall

Enregistrer Appliquer

CREATION D'UN JOB DE RELEASE

- Configurer le fichier settings.xml dans le serveur jenkins en ajoutant:
 - C:\opt\apache-maven-3.5.3\conf\settings.xml

```
<servers>
  <server>
    <id>nexus-snapshots</id>
    <username>admin</username>
    <password>1c5d93e9-b36f-4746-81dc-a7027eb6ec69</password>
  </server>
  <server>
    <id>nexus-releases</id>
    <username>admin</username>
    <password>1c5d93e9-b36f-4746-81dc-a7027eb6ec69</password>
  </server>
</servers>
<mirrors>
  <mirror>
    <id>central</id>
    <name>central</name>/
    <url>http://138.3.244.133:8081/repository/maven-public/</url>
    <mirrorOf>*</mirrorOf>
  </mirror>
</mirrors>
```


CREATION D'UN JOB DE RELEASE

- Créer un fork du repository suivant: [jenkins-sample-1](https://github.com/bnasslahsen/jenkins-sample-1)
 - <https://github.com/bnasslahsen/jenkins-sample-1.git>

CREATION D'UN JOB DE RELEASE

- Créer UN JOB MAVEN App-Deploy, en choisissant:
 - Utiliser l'url GIT de votre projet:
 - jenkins-sample-1
 - Construire un projet maven
 - Dans les Goal du job mettre
 - **clean deploy**
- Lancer un build pour valider



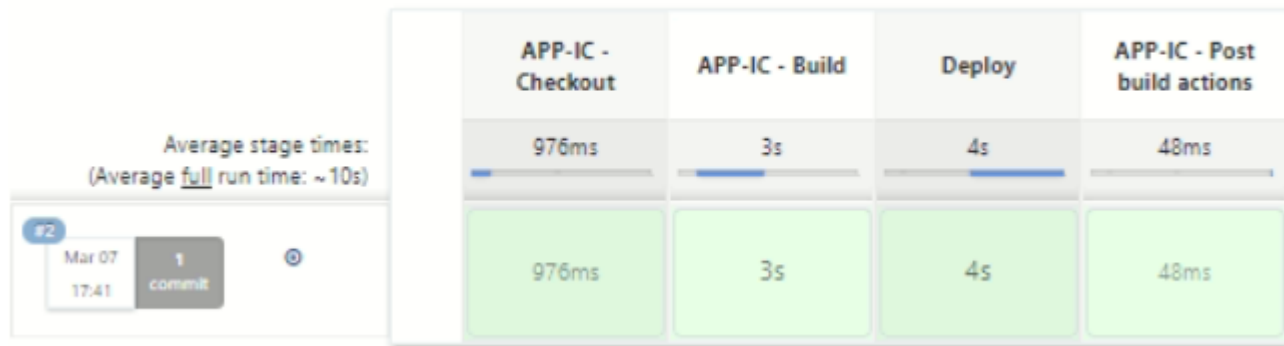
Construire un projet maven

Construit un projet avec maven. Jenkins utilise directement les retours.

UTILISATION DU PIPELINE AVEC UN STAGE DEPLOY

- Créer un job de type pipeline: App-Deploy-Pipeline
- Le fichier Jenkinsfile du projet jenkins-sample-1 va vous permettre le déploiement d'un artefact à partir du pipeline

Stage View



UTILISATION DU PLUGIN RELEASE

- Aller sur le menu jenkins de gauche,
 - Choisir Administrer jenkins
 - Gestion des plugins

Jenkins

- Nouveau Item
- Utilisateurs
- Historique des constructions
- Relations entre les builds
- Vérifier les empreintes numériques
- Administrer Jenkins 1**
- Mes vues
- Identifiants
- New View

File d'attente des constructions —
File d'attente des constructions vide

État du lanceur de compilations —
1 Au repos
2 Au repos

Administrer Jenkins

- Configurer le système**
Configurer les paramètres généraux et les chemins de fichiers.
- Configurer la sécurité globale**
Sécuriser Jenkins; définir qui est autorisé à accéder au système.
- Configure Credentials**
Configure the credential providers and types
- Configuration globale des outils**
Configurer les outils, leur localisation et les installeurs automatiques.
- Recharger la configuration à partir du disque**
Supprimer toutes les données en mémoire et recharger tout à partir du système de fichiers. Utile quand v
- Gestion des plugins 2**
Ajouter, supprimer, activer ou désactiver des plugins qui peuvent étendre les fonctionnalités de Jenkins.

UTILISATION DU PLUGIN RELEASE

- Aller dans l'onglet disponibles
- Choisir
 - Maven release Plug-in
- Installer sans redémarrer

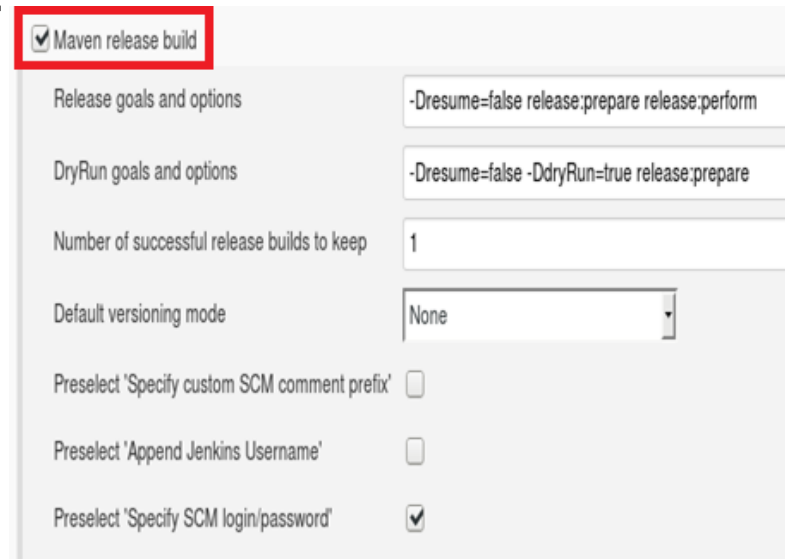
The plug-in generates the release

<input checked="" type="checkbox"/>	Maven Release Plug-in A plug-in that enables you to perform Maven releases.
<input type="checkbox"/>	Maven Artifact ChoiceListProvider (New) A ChoiceListProvider to read Maven artifacts.
<input type="checkbox"/>	Maven Deployment Linker A linker to deploy Maven artifacts.
<input type="checkbox"/>	Maven Info A plugin that provides information about the Maven project.
<input type="checkbox"/>	Maven Invoker Reports on Maven Invoker it tests.
<input type="checkbox"/>	Maven Metadata Plugin for Jenkins CI Adds a new build parameter type.
<input type="checkbox"/>	Maven Cascade Release Configure and perform maven release.

[Installer sans redémarrer](#)[Télécharger](#)

UTILISATION DU PLUGIN RELEASE

- Cocher dans le job, App-Déploy la case
 - Maven release build
- Mettre dans les paramètres du build:
 - clean package



<input checked="" type="checkbox"/> Maven release build	
Release goals and options	-Dresume=false release:prepare release:perform
DryRun goals and options	-Dresume=false -DdryRun=true release:prepare
Number of successful release builds to keep	1
Default versioning mode	None
Preselect 'Specify custom SCM comment prefix'	<input type="checkbox"/>
Preselect 'Append Jenkins Username'	<input type="checkbox"/>
Preselect 'Specify SCM login/password'	<input checked="" type="checkbox"/>

Build

POM Racine

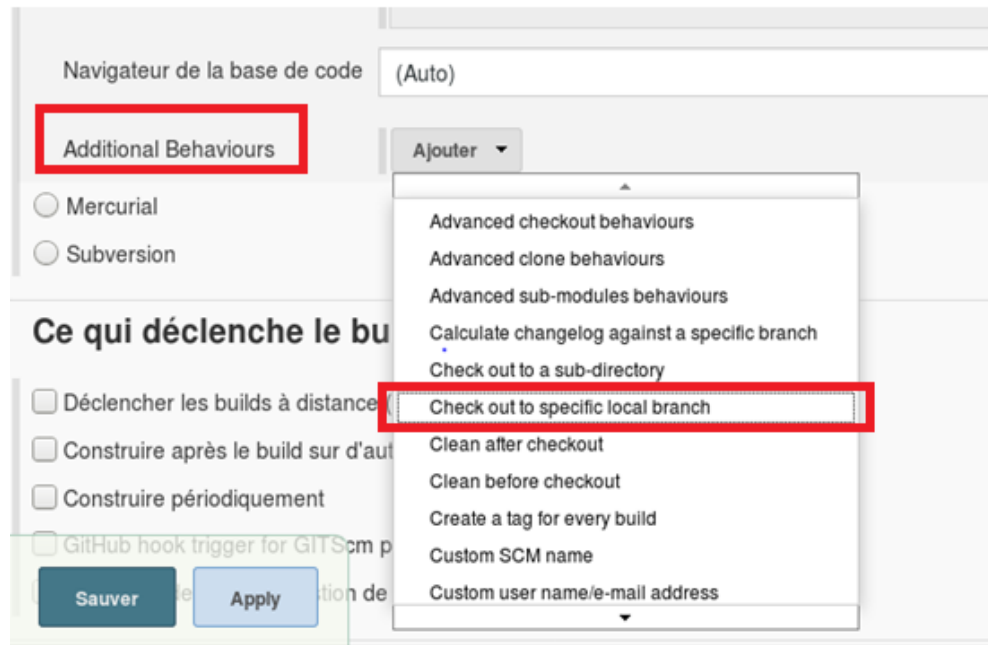
pom.xml

Goals et options

clean package

CREATION D'UN JOB DE RELEASE

- Aller dans additional behaviours
 - Choisir:
 - Check out to specific local branch



CREATION D'UN JOB DE RELEASE

- Adapter la section suivante du fichier pom.xml, du projet [jenkins-sample-1](#) afin d'utiliser votre repository GITHUB (en adaptant votre **username**)

```
<scm>
    <connection>scm:git:https://github.com/username/jenkins-sample-1.git</connection>
    <url>https://github.com/bnasslahsen/jenkins-sample-1</url>
    <developerConnection>scm:git:https://github.com/bnasslahsen/jenkins-sample-1.git</developerConnection>
</scm>
</project>
```

- Lors de la release, cette configuration vous permettra de créer des tags dans votre repository GITHUB
- Choisir un numéro de version SNAPSHOT différent des autres participants
- Sauvegarder les modifications du pom.xml
- Vous êtes prêt pour la release !

CREATION D'UNE RELEASE

- Il est possible de cocher dry run pour tester avant le release effective
- Tester le job release

 Retour au tableau de bord

 État

 Modifications

 Répertoire de travail

 Lancer un build

 Supprimer Maven project

 Configurer

 Modules

 Favoris

 Perform Maven Release

 Open Blue Ocean

 Rename

Perform Maven Release

Release Version

1.6

Development version

1.7-SNAPSHOT

Dry run only?

☐

☒ Specify SCM login/password

Username

1

Password

2

☐ Specify custom SCM comment prefix



☐ Specify custom SCM tag



3

Schedule Maven Release Build



Historique des builds

tendance



QUESTIONS / RÉPONSES