

FORMATION JENKINS - TP

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

Badr NASS LAHSEN
Le 16/09/2019

Plan des TP

- TP - Découverte de Jenkins
- TP MAVEN et GIT
- TP Intégration du projet dans JENKINS
- TP Sonar et JENKINS
- TP Junit et JENKINS
- TP automatisation du déploiement
- TP Docker et JENKINS - Bonus



TP - DÉCOUVERTE DE JENKINS

I

3

INSTALLATION CENTOS

- Lancer VirtualBox-*.exe
- Lancer Oracle_VM_VirtualBox_Extension_Pack-*.vbox-extpack
- Lancer VirtualBox
- Importer CentOS-*.ova
- Cette VM est à la base avec les propriétés suivantes:
 - RAM: 8 Go
 - Stockage: 25 Go, fixe , vdi
 - Mémoire Vidéo 128 Mo
 - CPU:4 vCPU
- login/password
 - root/rootroot

I

4

INSTALLATION DE JENKINS SOUS CENTOS COMME SERVICE

- Les commandes suivantes vont permettre d'installer jenkins avec les paramètres suivants:
 - Jenkins sera lancé comme daemon au démarrage. Pour plus de détails se reporter au fichier `/etc/init.d/jenkins`.
 - L'utilisateur 'jenkins' sera créé pour lancer le service. En cas de changement à un autre utilisateur, il faudra également lui donner les droits sur `/var/log/jenkins`, `/var/lib/jenkins`, and `/var/cache/jenkins`.
 - Les logs seront disponible sous `/var/log/jenkins/jenkins.log`.
 - Le fichier `/etc/sysconfig/jenkins`, contient le paramétrage de configuration de lancement
 - Par défaut, Jenkins sera lancé sur le port 8080.

5

INSTALLATION DE JENKINS SOUS CENTOS COMME SERVICE

- Installation
 - Se connecter en **root** sur la VM CenOS

```
cd /opt
wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo
rpm --import https://jenkins-ci.org/redhat/jenkins-ci.org.key
yum install jenkins
```
- Arrêt / Relance

```
service jenkins start
service jenkins stop
service jenkins restart
```
- Jenkins est accessible depuis l'url:
 - <http://localhost:8080/>

6

TP9: JENKINS - INSTALLATION

- Tester l'accès à l'url
 - <http://127.0.0.1:8080>
 - Se connecter avec le mot de passe

```
vim /var/lib/jenkins/secrets/initialAdminPassword
```

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 :

```
/root/.jenkins/secrets/initialAdminPassword
```

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

Mot de passe administrateur

Continuer

TP9: 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.

TP9: 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"/>

9



TP MAVEN ET GIT

10

INSTALLATION D'UN ENVIRONNEMENT DE DÉVELOPPEMENT SOUS WINDOWS

- Installation de JDK 8:
 - Lancer jdk-8u181-windows-i586.exe
 - Ajouter la variable d'environnement JAVA_HOME qui doit pointer sur le répertoire d'installation de JDK-8
 - Ajouter dans le path: %JAVA_HOME%\bin
 - echo %JAVA_HOME%
 - Vérifier l'installation: lancer la commande java -version
- Installation de 7zip
 - Lancer 7z1805-x64
- Installation de Notepad++
 - Lancer npp.7.5.8.Installer.exe

11

INSTALLATION D'UN ENVIRONNEMENT DE DÉVELOPPEMENT

- Installation Maven
 - Dans MesDocuments
 - Créer le répertoire m2repo
 - Dézipper apache-maven-3.5.0-bin.zip dans MesDocuments(avec 7zip)
 - Ajouter la variable d'environnement M2_HOME qui doit pointer sur le répertoire d'installation de Maven
 - Aller dans le fichier %M2_HOME%\conf\settings.xml et ajouter:
<localRepository>D:/DEV/m2repo</localRepository>
 - Ajouter dans le path: %M2_HOME%\bin
 - Vérifier l'installation: mvn -version
- Installation Eclipse
 - Créer le répertoire workspace dans MesDocuments
 - Dézipper eclipse-jee-oxygen-3a-win32-x86_64.zip au sein du répertoire tools
 - Lancer eclipse en utilisant le répertoire workspace créé précédemment
 - Paramétrer eclipse pour pointer sur settings.xml de l'installation maven locale

12

CRÉATION D'UN PREMIER PROJET AVEC MAVEN

- Créer un répertoire de travail TP1 dans workspace
- Aller dans le répertoire workspace/TP1 et lancer la commande:
- Ajouter le fichier .gitignore fourni à la racine du projet SampleProject
- `mvn archetype:generate -DgroupId=fr.orsys.samples -DartifactId=SampleProject -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false`
- Importer le projet créé, comme un projet maven sous eclipse
- Lancer la classe de tester AppTest, afin de valider le projet

I

13

GÉNÉRATION DE LA DOCUMENTATION PROJET AVEC MAVEN

- Pour cette question, il faudra se référer à la documentation officielle de Maven, disponible sous <https://maven.apache.org/>.
- En ligne de commande:
 - Générer le package sous format jar
 - Générer la javadoc du projet créé
 - Publier le projet ainsi que les résultats de tests sur un site web.

I

14

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)

I

15

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>>>`

I

16

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>`

17

PUSH DU PROJET SOUS UN SCM

- Installation de GIT
 - Lancer Git-2.14.1-64-bit.exe
- Créer un compte sous github:
 - <https://github.com/>
- Créer un repository public: jenkins-sample-1
- Puis lancer les commandes suivantes en adaptant votre username et le chemin de votre workspace:

```
cd $workspace/TP1/SampleProject
git init
git add .
git commit -m "first commit"
git remote add origin https://github.com/$username/jenkins-sample-1.git
git push -u origin master
```

18



TP INTÉGRATION DU PROJET DANS JENKINS

I

19

INSTALLATION DE JDK-8 SOUS CENTOS7

- Lancer la commande

```
yum install java-1.8.0-openjdk-devel
```

- Puis pour la vérification:
 - java -version

I

20

INSATALLATION DE MAVEN SOUS CENTOS7

```
cd /opt
wget http://mirrors.ircam.fr/pub/apache/maven/maven-3/3.5.4/binaries/apache-maven-3.5.4-bin.tar.gz
tar -xf apache-maven-3.5.4-bin.tar.gz
```

- Vérifier l'installation

```
cd /opt/apache-maven-3.5.4/bin
./mvn -version
```

I

21

INSTALLATION DE GIT SOUS CENTOS

- Lancer la commande:

```
yum install git
git --version
```


- Relancer Jenkins:

```
service jenkins restart
```

I

22

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 installeurs automatiques.

23

PARAMÉTRAGE DE JDK

JDK

Installations JDK

Ajouter JDK

JDK

Nom

java-8

JAVA_HOME

/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.222.b10-0.el7_6.x86_64

☐ Install automatically

Supprimer JDK

Ajouter JDK

Liste des installations JDK sur ce système

24

PARAMÉTRAGE DE MAVEN

Maven

Installations Maven

Ajouter Maven

Maven

Nom

maven

MAVEN_HOME

/opt/apache-maven-3.5.4/

☐ Install automatically

25

CRÉATION DE JOB FREE STYLE SOUS JENKINS

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

Jenkins

Nouveau Item

Utilisateurs

Historique des constructions

Administrer Jenkins

Mes vues

Identifiants

New View

26

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 Jenkins pour tout autre chose qu'un build logiciel.

27

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

- aucun -

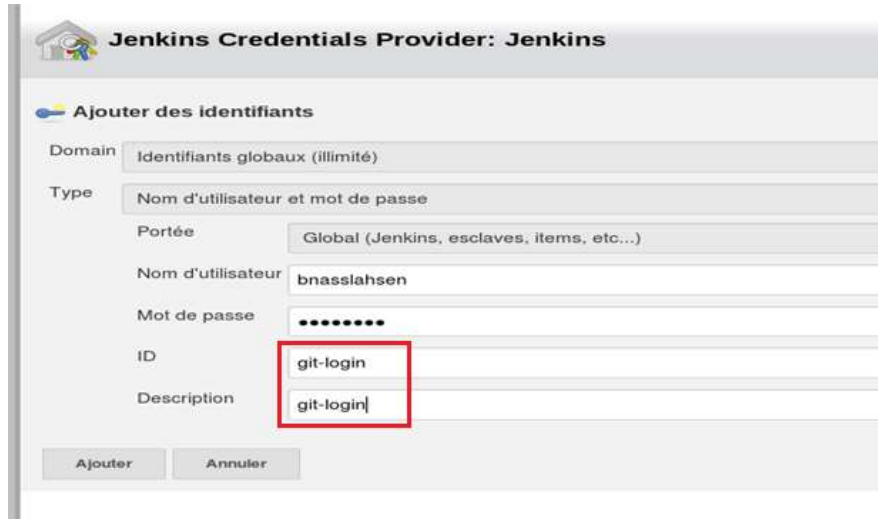
Ajouter

Avancé...

Add Repository

28

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:

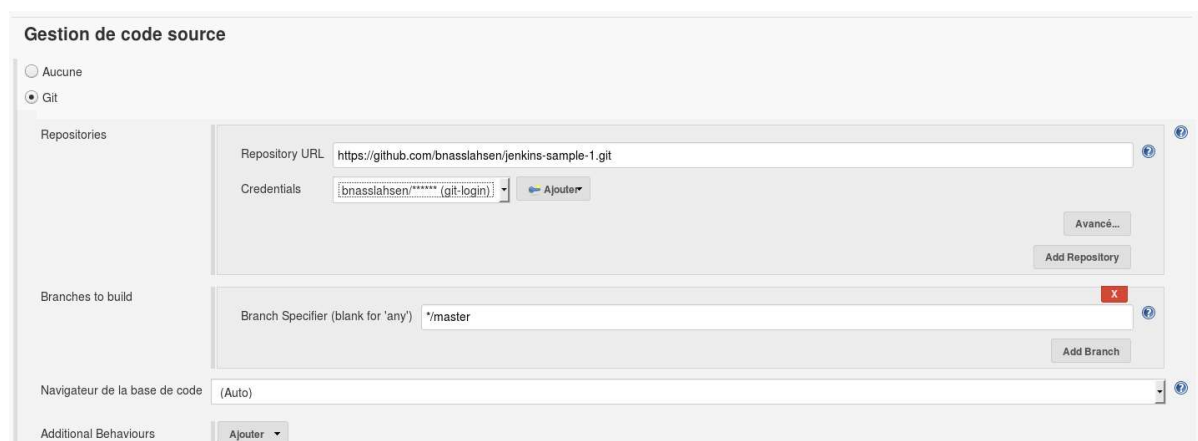
- Domain: Identifiants globaux (illimité)
- Type: Nom d'utilisateur et mot de passe
- Portée: Global (Jenkins, esclaves, items, etc...)
- Nom d'utilisateur: bnasslahsen
- Mot de passe: [masked with dots]
- ID: git-login (highlighted with a red box)
- Description: git-login| (highlighted with a red box)

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

29

CRÉATION DE JOB FREE STYLE SOUS JENKINS

- Choisir votre login/mdp git



The screenshot shows the 'Gestion de code source' configuration page in Jenkins. The 'Git' radio button is selected. The form fields are as follows:

- Repositories:
 - Repository URL: https://github.com/bnasslahsen/jenkins-sample-1.git
 - Credentials: bnasslahsen/***** (git-login) (with an 'Ajouter' button next to it)
- Branches to build:
 - Branch Specifier (blank for 'any'): */master
- Navigateur de la base de code: (Auto)
- Additional Behaviours: Ajouter

Buttons 'Avancé...' and 'Add Repository' are visible on the right side of the 'Repositories' section.

30

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 configuration interface. Under the heading 'Invoquer les cibles Maven de haut niveau', there are two input fields: 'Version de Maven' with the value 'MAVEN-3' and 'Cibles Maven' with the value 'clean package'.

The screenshot shows the 'Build' section of the Jenkins configuration interface. A dropdown menu is open, showing various build steps. The option 'Invoquer les cibles Maven de haut niveau' is highlighted.

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

31

CONFIGURATION: SERVEUR DE MAIL

1. Administrer Jenkins
2. Configurer le système

The screenshot shows the 'Notification par email' configuration page in Jenkins. The 'Serveur SMTP' field is set to 'smtp.gmail.com'. The 'Suffixe par défaut des emails des utilisateurs' field is empty. The 'Utiliser l'authentification par SMTP' checkbox is checked. The 'Nom d'utilisateur' field is set to '@gmail.com'. The 'Mot de passe' field is masked with '*****'. The 'Utiliser SSL' checkbox is checked. The 'Port SMTP' field is empty. The 'Reply-To Address' field is set to '@gmail.com'. The 'Jeu de caractères' field is set to 'UTF-8'. The 'Tester la configuration en envoyant un e-mail de test' checkbox is checked. The 'Destinataire du courriel de test' field is set to '@gmail.com'. The 'Email was successfully sent' message is displayed. The 'Tester la configuration' button is highlighted. At the bottom, there are 'Enregistrer' and 'Appliquer' buttons.

32

CONFIGURATION: SERVEUR DE MAIL

- Par défaut, le compte Gmail est hautement sécurisé.
- Lorsque nous utilisons gmail smtp à partir d'un outil non gmail, les courriers électroniques sont bloqués
- Si vous obtenez l'erreur
 - Connectez-vous à Gmail.
 - Accédez à l'URL à l'adresse <https://www.google.com/settings/security/lesssecureapps>.
 - Sélectionnez "Activer"

Failed to send e-mail

javax.mail.AuthenticationFailedException: 535-5.7.8 Username and Password not accepted. Learn more at 535 5.7.8 <https://support.google.com/mail/?p=BadCredentials> y3sm12758922wrl.78 - gsmt

```
at com.sun.mail.smtp.SMTPTransport$Authenticator.authenticate(SMTPTransport.java:809)
at com.sun.mail.smtp.SMTPTransport.authenticate(SMTPTransport.java:752)
at com.sun.mail.smtp.SMTPTransport.protocolConnect(SMTPTransport.java:669)
at javax.mail.Service.connect(Service.java:317)
at javax.mail.Service.connect(Service.java:176)
at javax.mail.Service.connect(Service.java:125)
at javax.mail.Transport.send0(Transport.java:194)
at javax.mail.Transport.send(Transport.java:124)
at hudson.tasks.MailDescriptorImpl.doSendTestMail(Mailer.java:671)
at java.lang.invoke.MethodHandle.invokeWithArguments(MethodHandle.java:627)
at org.kohsuke.stapler.Function$MethodFunction.invoke(Function.java:396)
at org.kohsuke.stapler.Function$InstanceFunction.invoke(Function.java:408)
at org.kohsuke.stapler.interceptor.RequirePOST$Processor.invoke(RequirePOST.java:77)
at org.kohsuke.stapler.PreInvokeInterceptedFunction.invoke(PreInvokeInterceptedFunction.java:26)
at org.kohsuke.stapler.Function.bindAndInvoke(Function.java:212)
at org.kohsuke.stapler.Function.bindAndInvokeAndServeResponse(Function.java:145)
at org.kohsuke.stapler.MetaClass$11.doDispatch(MetaClass.java:535)
at org.kohsuke.stapler.NameBasedDispatcher.dispatch(NameBasedDispatcher.java:58)
at org.kohsuke.stapler.Stapler.tryInvoke(Stapler.java:747)
at org.kohsuke.stapler.Stapler.invoke(Stapler.java:878)
at org.kohsuke.stapler.MetaClass$4.doDispatch(MetaClass.java:288)
at org.kohsuke.stapler.NameBasedDispatcher.dispatch(NameBasedDispatcher.java:58)
at org.kohsuke.stapler.Stapler.tryInvoke(Stapler.java:747)
at org.kohsuke.stapler.Stapler.invoke(Stapler.java:878)
at org.kohsuke.stapler.Stapler.invoke(Stapler.java:676)
at org.kohsuke.stapler.Stapler.service(Stapler.java:238)
```

33

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 ▼

34

CRÉATION DE JOB FREE STYLE SOUS JENKINS

- Afin de programmer un build toutes les 5 min,
- Aller dans la section build trigger
- Renseigner la conf dans planning: H/5 * * * *

Build Triggers

☐ Construire après le build sur d'autres projets

☒ Construire périodiquement

Planning: H/5 * * * *

Aurait été lancé à mardi 8 mai 2018 14 h 37 CEST; prochaine exécution à mardi 8 mai 2018 14 h 42 CEST.

☐ 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)

35

UTILISATION DES VUES

- Créer une vue SampleApp

Jenkins

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

Nom de la vue: SampleApp 1

☐ Ma vue
Cette vue affiche automatiquement tous les jobs auxquels l'utilisateur a accès.

☒ **Vue liste** 2
Montre les jobs dans une simple liste. Vous pouvez choisir les jobs à afficher dans chaque vue.

OK

36

UTILISATION DES VUES

- Ajouter à cette vue le job: App-IC

Nom: SampleApp

Description:

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 1

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

Ok 2 Apply

37

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 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

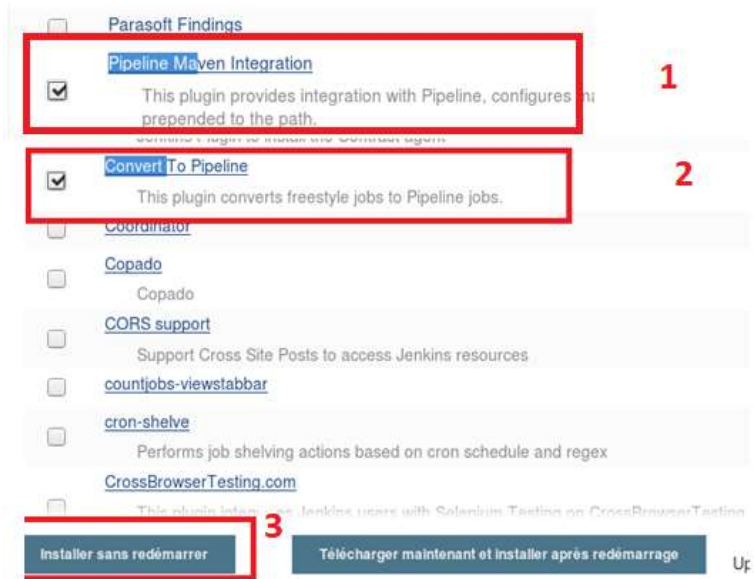
Gestion des plugins 2

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

38

CONVERTIR LE JOB FREE-STYLE EN PIPELINE

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



39

CONVERTIR LE JOB FREE-STYLE EN PIPELINE

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



40

CONVERTIR LE JOB FREE-STYLE EN PIPELINE

- Renseigner les paramètres de conversion
- Cliquer sur convert

Convert To Pipeline

Job Conversion Parameters

Job name (optional) **App-IC-Pipeline**

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

Recursively convert downstream jobs if any ☒

Jenkinsfile Options

Commit Jenkinsfile? ☒

Commit message Auto created Jenkinsfile from auto conversion

Convert

41

CONVERTIR LE JOB FREE-STYLE EN PIPELINE

- Lancer le job: App-Ic-Pipeline

Stage View



42



TP SONAR ET JENKINS

43

INSTALLATION DE SONAR SOUS CENTOS

- Lancer les commandes suivantes

```
wget -O /etc/yum.repos.d/sonar.repo  
http://downloads.sourceforge.net/project/sonar-pkg/rpm/sonar.repo  
yum -y install sonar
```

- La configuration de Sonar est disponible dans le fichier sonar.properties (à titre informatif):
 - /opt/sonar/conf/sonar.properties
- Démarrer Sonar à l'aide de la commande suivante:

```
/etc/init.d/sonar start
```

44

INSTALLATION DE SONAR

- Accès à sonar depuis l'url:
 - <http://localhost:9000/>
 - Connecter vous à sonar avec le login/mdp:
 - admin/admin
- Générer votre token et mémoriser le dans éditeur de texte par exemple

Welcome to SonarQube!

[Skip this tutorial](#)

Want to quickly analyze a first project? Follow these 2 easy steps.

[Find it back anytime in the Help section](#)

1 Provide a token

admin-token: `4c0c045a18e30a3365c70173794aa4851319f205` ✖

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point of time in your user account.

[Continue](#)

45

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

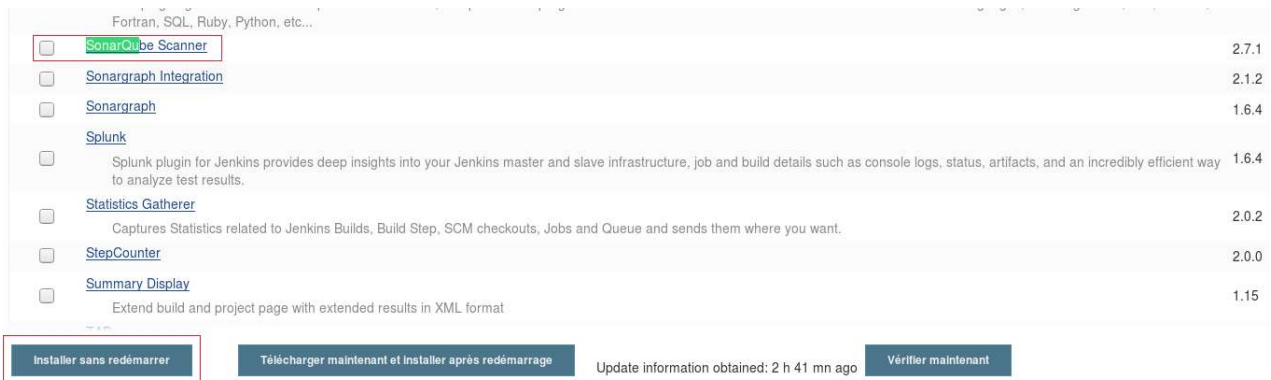
Administrer Jenkins

- 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.

46

INSTALLATION DU PLUGIN SONAR SOUS JENKINS

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



47

CONFIGURATION DU PLUGIN SONAR SOUS JENKINS

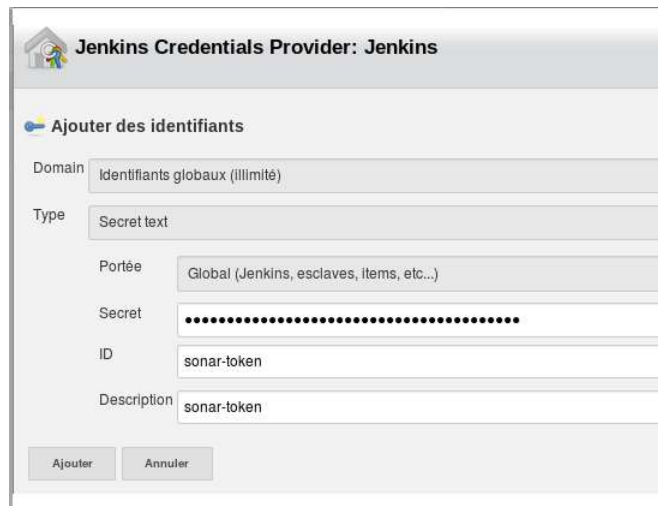
- Aller sur le menu Jenkins ► Administrer Jenkins ► Configurer le Système
- Aller à la section "SonarQube servers"
- Cliquer sur "Add SonarQube"
- Renseigner les informations, comme indiqué sur l'image:
 - Nom: Sonar
 - URL: <http://localhost:9000/>



48

CONFIGURATION DU PLUGIN SONAR SOUS JENKINS

- Configurer le token d'authentification

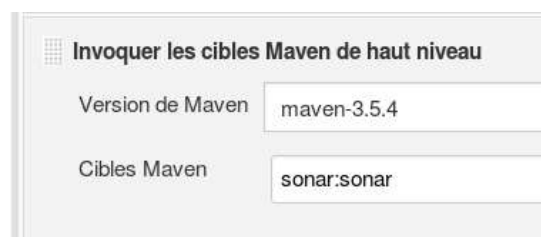


The screenshot shows the 'Jenkins Credentials Provider: Jenkins' configuration page. It has a title bar with a Jenkins logo and the text 'Jenkins Credentials Provider: Jenkins'. Below the title bar is a section titled 'Ajouter des identifiants' with a key icon. The form contains the following fields: 'Domain' with the value 'Identifiants globaux (illimité)', 'Type' with the value 'Secret text', 'Portée' with the value 'Global (Jenkins, esclaves, items, etc...)', 'Secret' with a masked value represented by dots, 'ID' with the value 'sonar-token', and 'Description' with the value 'sonar-token'. At the bottom of the form are two buttons: 'Ajouter' and 'Annuler'.

49

UTILISATION CONTEXTE SONAR SUR 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 box titled 'Invoquer les cibles Maven de haut niveau'. It contains two fields: 'Version de Maven' with the value 'maven-3.5.4' and 'Cibles Maven' with the value 'sonar:sonar'.

50

ANALYSE SONAR SUR LES PROJETS AVEC PIPELINE

- Ajouter le bloc suivante au niveau du pipeline du projet: jenkins-sample-1

```
stage ('APP-IC - Quality Analysis') {  
  withMaven(maven: 'maven-3.5.4') {  
    if(isUnix()) {  
      sh "mvn sonar:sonar"  
    } else {  
      bat "mvn sonar:sonar"  
    }  
  }  
}
```

- Identifier les violations qui vous paraissent 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

51



TP BLUE OCEAN PIPELINE

52

UTILISATION DU PLUGIN BLUE OCEAN PIPELINE

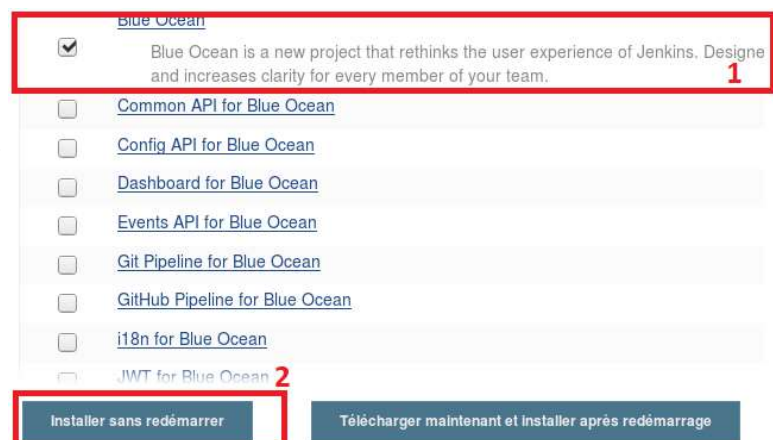
- Aller sur le menu jenkins de gauche,
 - Choisir Administrer jenkins
 - Gestion des plugins



53

UTILISATION DU PLUGIN BLUE OCEAN PIPELINE

- Aller dans l'onglet disponibles
 - Choisir
 - Blue Ocean
 - Installer sans redémarrer



54

UTILISATION DU PLUGIN BLUE OCEAN PIPELINE

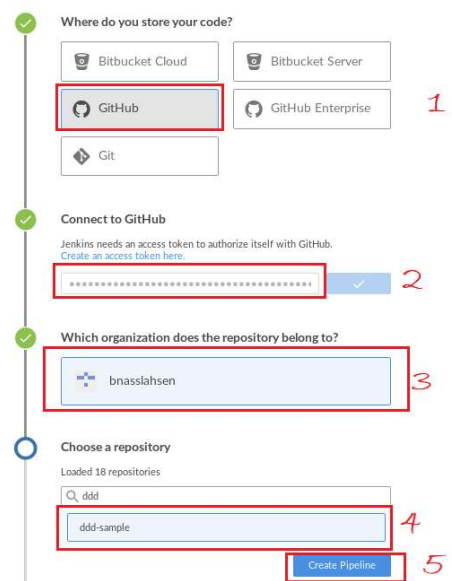
- Ouvrir la console Blue Ocean



55

UTILISATION DU PLUGIN BLUE OCEAN PIPELINE

- Créer un nouveau pipeline en vous branchant sur le projet
 - <https://github.com/bnasslahsen/ddd-sample.git>
 - En utilisant l'access token:
2e28ebf275efcaf007fa09c18dfd474ae690da5a



56

UTILISATION DU PLUGIN BLUE OCEAN PIPELINE

- Lancer un build:



57



TP JUNIT ET JENKINS

58

TESTS SAMPLE-APP

- Créer un test junit
- Valider l' exécution du test avec Maven
- Relancer un Build sous Jenkins
- Pour aller plus loin, vous pouvez configurer le plugin jacoco au niveau configuration du pom.xml du projet.

I

59



TP AUTOMATISATION DU DÉPLOIEMENT

I

60

INSTALLATION DE NEXUS

- Téléchargement et installation de nexus

```
cd /opt
wget http://download.sonatype.com/nexus/3/latest-unix.tar.gz
tar xvf latest-unix.tar.gz -C /opt
ln -s /opt/nexus-3.18.1-01/ /opt/nexus
```

- Ajout de l'utilisateur nexus

```
adduser nexus
chown -R nexus:nexus /opt/nexus-3.18.1-01
chown -R nexus:nexus /opt/sonatype-work
```

I

61

INSTALLATION DE NEXUS

- Lancement de nexus comme service

```
## Modifier run_as_user="nexus"
vim /opt/nexus-3.18.1-01/bin/nexus.rc
ln -s /opt/nexus-3.18.1-01/bin/nexus /etc/init.d/nexus
systemctl enable nexus
## Démarrage de nexus
systemctl start nexus
```

I

62

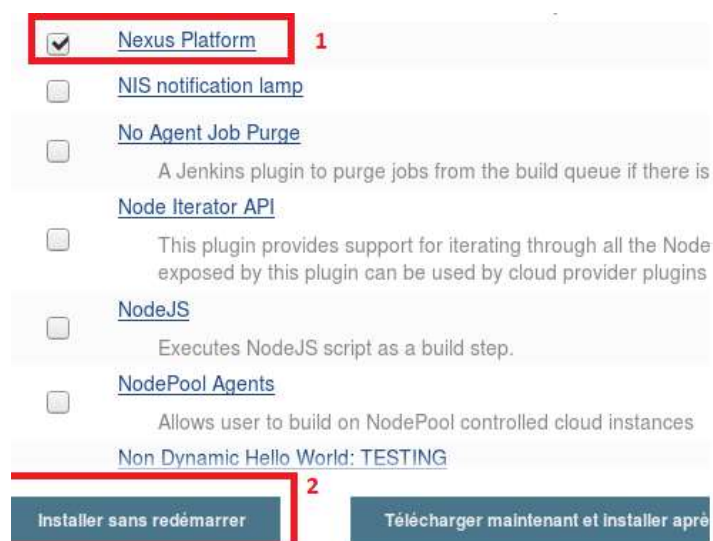
INSTALLATION DE NEXUS

- Les logs sont disponible sous:
 - vi sonatype-work/nexus3/log/nexus.log
- Url d'accès: <http://localhost:8081/>
- Login/mdp: admin/admin123
- Pour l'arrêt
 - systemctl stop nexus

63

UTILISATION DU PLUGIN NEXUS

- Aller dans l'onglet disponibles
- Choisir
 - Nexus platform plugin
- Installer sans redémarrer



64

UTILISATION DU PLUGIN NEXUS

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



Sonatype Nexus

Nexus Repository Manager Servers

Nexus Repository Manager 3.x Server

Display Name: nexus

Server ID: nexus

Server URL: http://localhost:8081/

Credentials: admin***** (nexus-login) [Ajouter]

Nexus Repository Manager 3.x connection succeeded

⚠ NXRM OSS 3.18.1-01 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

65

UTILISATION DU PLUGIN NEXUS

- Le mot de passe se trouve dans le fichier suivant:

`/opt/sonatype-work/nexus3/admin.password`



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

66

CREATION D'UN JOB DE RELEASE

- Configurer le fichier settings.xml de maven en ajoutant:
 - vim apache-maven-3.5.4/conf/settings.xml

```
<servers>
  <server>
    <id>nexus-snapshots</id>
    <username>admin</username>
    <password>admin123</password>
  </server>
  <server>
    <id>nexus-releases</id>
    <username>admin</username>
    <password>admin123</password>
  </server>
</servers>
<mirrors>
  <mirror>
    <id>central</id>
    <name>central</name>
    <url>http://localhost:8081/repository/maven-public/</url>
    <mirrorOf>*</mirrorOf>
  </mirror>
</mirrors>
```

67

CREATION D'UN JOB DE RELEASE

- Modifier le pom.xml du projet [jenkins-sample-1](#) en ajoutant

```
<repositories>
  <repository>
    <id>maven-group</id>
    <url>http://localhost:8081/repository/maven-public/</url>
  </repository>
</repositories>
<distributionManagement>
  <snapshotRepository>
    <id>nexus-snapshots</id>
    <url>http://localhost:8081/repository/maven-snapshots/</url>
  </snapshotRepository>
  <repository>
    <id>nexus-releases</id>
    <url>http://localhost:8081/repository/maven-releases/</url>
  </repository>
</distributionManagement>
</project>
```

68

CREATION D'UN JOB DE RELEASE

- Créer UN JOB MAVEN App-BUILD, en choisissant:

- Utiliser l'url GIT du projet:
 - jenkins-sample-1
- Construire un projet maven
- Dans les Goal du job mettre
 - clean deploy



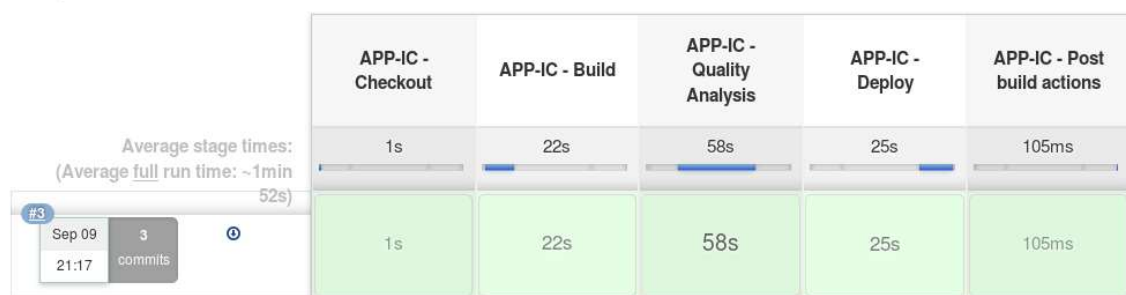
- Lancer un build pour valider

69

AJOUT AU NIVEAU DU PIPELINE DU DEPLOY

- Adapter le Jenkinsfile du projet jenkins-sample-1 pour permettre le déploiement d'un artefact à partir du pipeline

Stage View



70

UTILISATION DU PLUGIN RELEASE

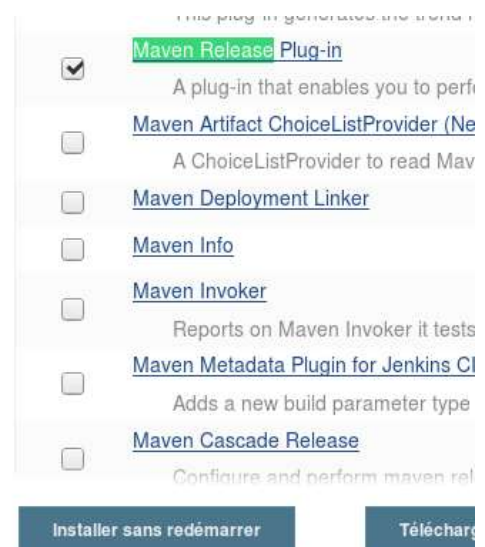
- Aller sur le menu jenkins de gauche,
 - Choisir Administrer jenkins
 - Gestion des plugins



71

UTILISATION DU PLUGIN RELEASE

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



72

UTILISATION DU PLUGIN RELEASE

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

☒ 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': ☐

Preselect 'Append Jenkins Username': ☐

Preselect 'Specify SCM login/password': ☒

Build

POM Racine: pom.xml

Goals et options: clean package

73

CREATION D'UN JOB DE RELEASE

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

Navigateur de la base de code: (Auto)

Additional Behaviours

☐ Mercurial

☐ Subversion

Ce qui déclenche le build

☐ Déclencher les builds à distance

☐ Construire après le build sur d'aut

☐ Construire périodiquement

☒ Check out to specific local branch

☐ Clean after checkout

☐ Clean before checkout

☐ Create a tag for every build

☐ Custom SCM name

☐ Custom user name/e-mail address

Sauver Apply

74

CREATION D'UN JOB DE RELEASE

- Modifier le pom.xml du projet [jenkins-sample-1](#) en ajoutant et 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>
```

75

CREATION D'UNE RELEASE

- 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

Historique des builds [tendance](#)

Perform Maven Release

Release Version

Development version

Dry run only? ☐

☒ Specify SCM login/password

Username

Password

☐ Specify custom SCM comment prefix

☐ Specify custom SCM tag

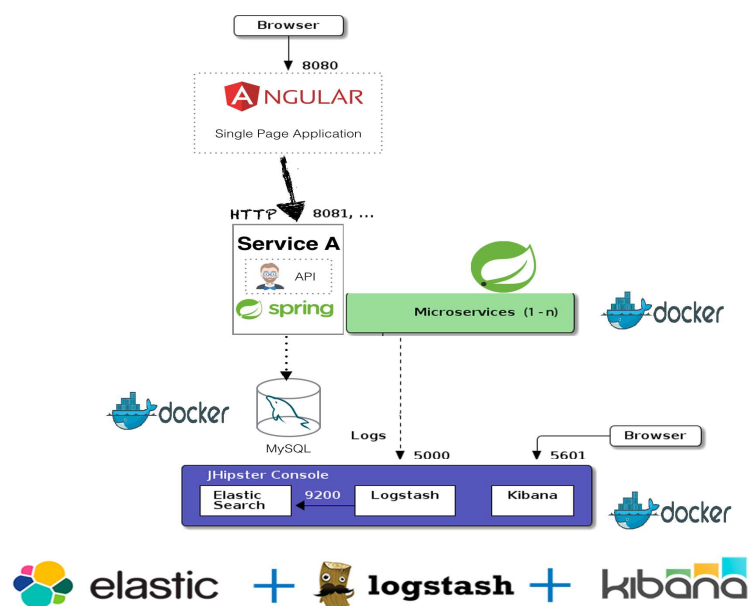
76

LEARNING FROM EXPERIENCE

TP JENKINS ET DOCKER - BONUS

77

ARCHITECTURE DU PROJET EXEMPLE



78

INSTALLATION DE DOCKER SUR LE SERVEUR JENKINS

- Lancer le terminal (en root), puis lancer les commandes:

```
yum check-update  
curl -fsSL https://get.docker.com/ | sh
```

- Démarrer le service:

```
systemctl start docker
```

- Vérifier l'installation:

```
systemctl status docker  
docker run --rm hello-world
```

- Démarrer docker comme service

```
systemctl enable docker
```

- Ajout de l'utilisateur au groupe docker

```
usermod -aG docker $(whoami)  
usermod -a -G docker jenkins
```

79

INSTALLATION DE DOCKER SUR LE SERVEUR JENKINS

- Ajouter Jenkins à la liste des sudoers

```
echo "jenkins ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
```

- Relancer docker

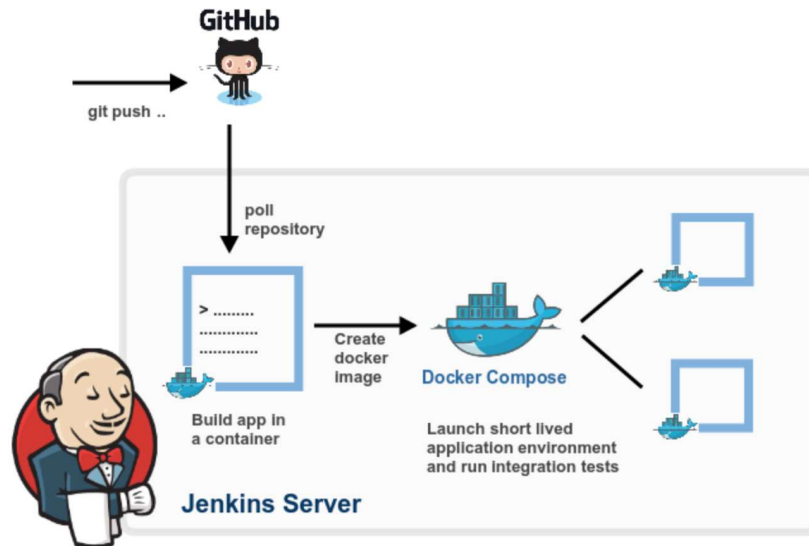
```
service docker restart
```

- Relancer jenkins

```
service jenkins restart
```

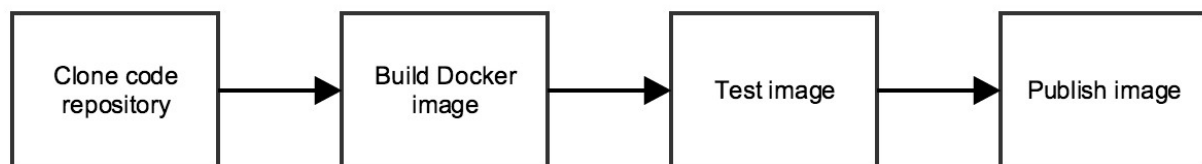
80

ARCHITECTURE DE L'INTEGRATION AVEC JENKINS



81

ILLUSTRATION DES ÉTAPES A IMPLÉMENTER SUR JENKINS



82

CRÉATION DE REPOSITORY DANS DE DOCKERHUB

- Connectez-vous sur le site web:
 - <https://hub.docker.com/>
- Créer un compte (login / mdp), qui sera réutilisé dans la conf jenkins dans étapes suivantes
- Créer un repository sous le hub docker, avec le nom:

jenkins-repo

83

MODIFICATION DU POM.XML

- Créer un nouveau repository dans votre GITHUB, avec le nom: sample-docker-repo
- Cloner ce repository dans votre répertoire de travail
- Télécharger le zip du projet: sample-docker-repo
 - <https://github.com/bnasslahsen/sample-docker-repo/archive/master.zip>
- Dézipper le contenu dans votre nouveau repository local créé: sample-docker-repo

84

MODIFICATION DU POM.XML

- Editer le fichier pom.xml
- Aller dans la configuration de jib-maven-plugin
- Valoriser la ligne suivante en utilisant votre login dockerhub.
 - Voici un exemple:

```
<to>  
  <image>registry.hub.docker.com/bnasslahsen/jenkins-repo</image>  
</to>
```

I

85

CREATION D' UN PREMIER JOB AVEC DOCKER

- Push des modifications sur GITHUB

```
git add .  
git commit -m "ajout jenkins pipeline"  
git push origin master
```

I

86

AJOUT DE L'UTILISATEUR DOCKER REGISTRY SUR JENKINS 1/3

- Aller dans le menu jenkins:
 - **Identifiants -**
 > System
 -> identifiants
 globaux
- Choisir:
 - **Add Credentials**



87

AJOUT DE L'UTILISATEUR DOCKER REGISTRY SUR JENKINS 2/3

Kind:

Scope:

Username:

Password:

ID:

Description:

OK

88

CONFIRMATION DE L'AJOUT DE L'UTILISATEUR DOCKER REGISTRY SUR JENKINS 3/3

Identifiants globaux (illimité)

Credentials that should be available irrespective of domain specification to requirements matching.

	Nom	Type	Description
	bnasslahsen*****	Nom d'utilisateur et mot de passe	

Icône: [S](#) [M](#) [L](#)

89

CRÉATION DE JOB SOUS JENKINS 1/8

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



90

CRÉATION DE JOB SOUS JENKINS 2/8

- Choisir le projet de type Pipeline
- Renseigner un nom de projet:
 - Docker-App-IC

Enter an item name

→ Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
A container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate space, so you can have multiple things of the same name as long as they are in different folders.

91

CRÉATION DE JOB SOUS JENKINS 3/8

- Pour la définition de, choisir Pipeline script from SCM
- Renseigner le l'url git de votre repository : sample-docker-repo
- Dans Credentials, cliquer sur ajouter

Pipeline

Definition: Pipeline script from SCM

SCM: Git

Repositories

Repository URL: https://github.com/bnasslahsen/devops.git

Credentials: - aucun - [Ajouter]

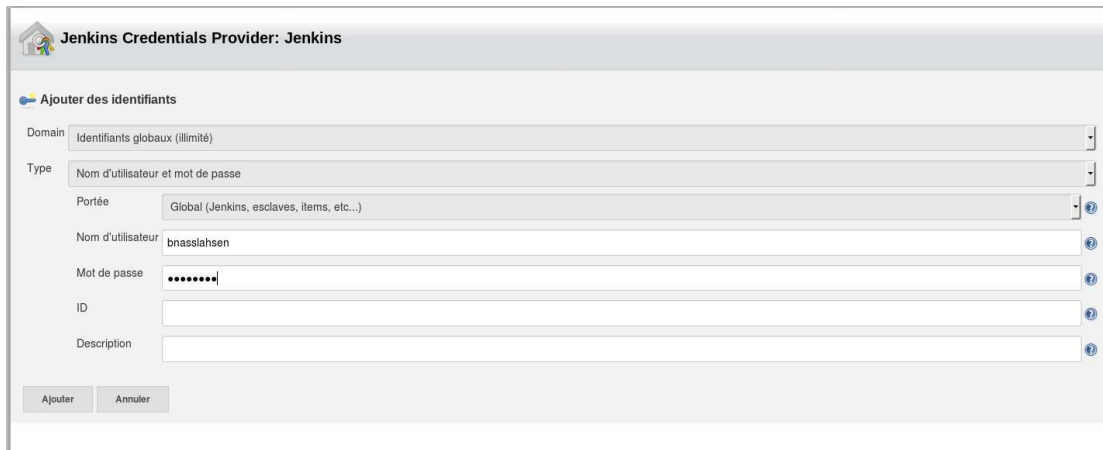
Branches to build

Branch Specifier (blank for 'any'): */master

92

CRÉATION DE JOB SOUS JENKINS 4/8

- Ajouter votre login et mdp de github



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: bnasslahsen

Mot de passe:

ID:

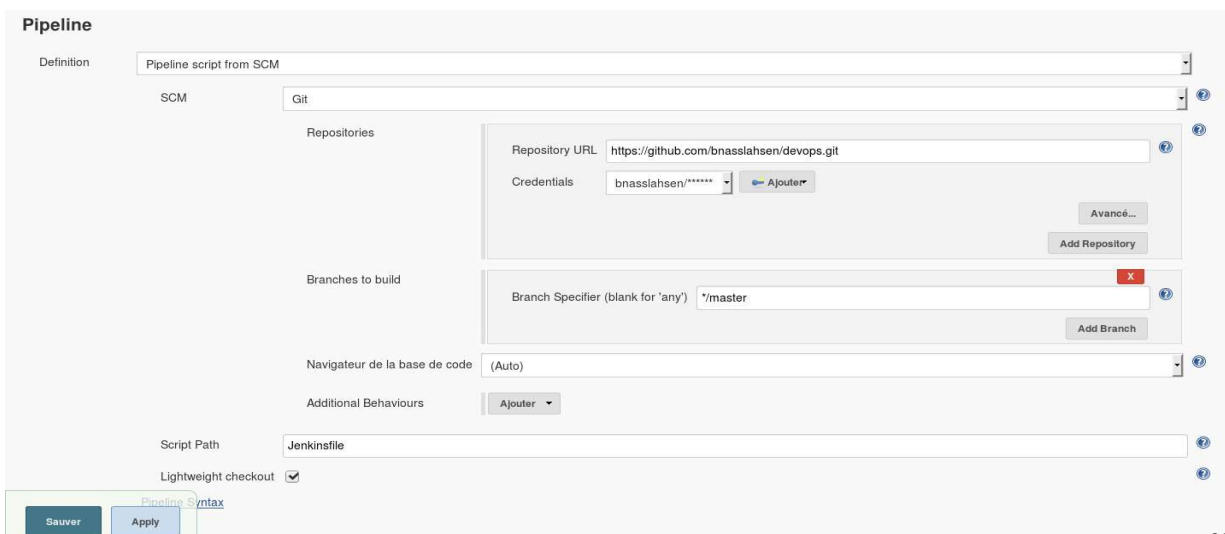
Description:

Ajouter Annuler

93

CRÉATION DE JOB SOUS JENKINS 6/8

- Cliquer sur Save



Pipeline

Definition: Pipeline script from SCM

SCM: Git

Repositories:

Repository URL: https://github.com/bnasslahsen/devops.git

Credentials: bnasslahsen/*****

Branches to build:

Branch Specifier (blank for 'any'): */master

Navigateur de la base de code: (Auto)

Additional Behaviours: Ajouter

Script Path: Jenkinsfile

Lightweight checkout: ☒

Sauver Apply

94


CRÉATION DE JOB SOUS JENKINS 7/8

- Choisir: (Menu de gauche)
 - Lancer un build

Jenkins > App-IC >

[Back to Dashboard](#)
[Status](#)
[Changes](#)
[Lancer un build](#)
[Supprimer Pipeline](#)
[Configurer](#)
[Full Stage View](#)
[Pipeline Syntax](#)

Pipeline App-IC

 [Recent Changes](#)

Stage View

No data available. This Pipeline has not yet run.

Historique des builds [tendance](#)

95

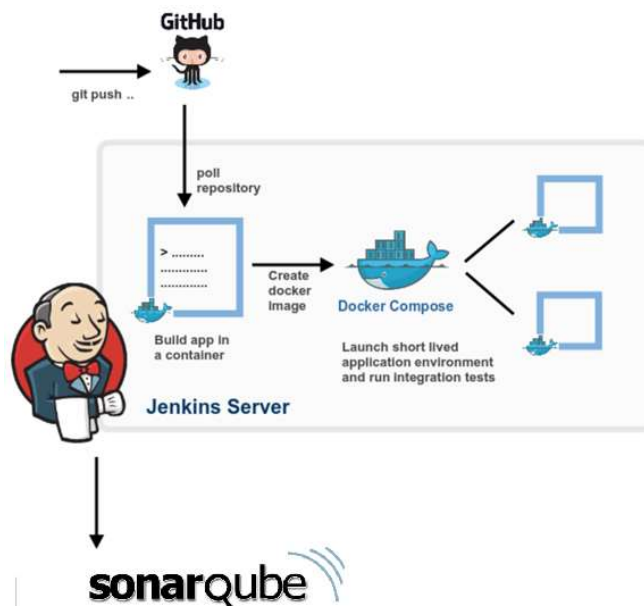
EXEMPLE DE RÉSULTAT D'EXÉCUTION 8/8

Stage View

	checkout	check java	clean	install tools	yarn install	backend tests	frontend tests	packaging	build docker	publish docker
Average stage times: (Average full run time: ~30min)	3s	1s	21s	29s	3min 45s	5min 48s	1min 39s	6min 3s	47s	4min 50s
#3 May 08 14:52 No Changes	2s	1s	7s	14s	6min 53s	5min 48s	1min 39s	6min 3s	47s	4min 50s

96

ARCHITECTURE DE L'INTEGRATION ENTRE JENKINS ET SONAR



97

SONAR - CONFIGURATION DU FIREWALL

- Nous allons autoriser l'accès à n'importe quel port du conteneur vers le host.
- Pour cela, lancer les commandes suivantes en root:

```
nmcli connection modify docker0 connection.zone trusted
systemctl stop NetworkManager.service
firewall-cmd --permanent --zone=trusted --change-interface=docker0
systemctl start NetworkManager.service
nmcli connection modify docker0 connection.zone trusted
systemctl restart docker.service
```

98

SONAR - CONFIGURATION DU FIREWALL

- Ajout des règles sur le firewall:

```
sudo firewall-cmd --zone=public --add-port=5000/udp --permanent
sudo firewall-cmd --zone=public --add-port=5000/tcp --permanent
sudo firewall-cmd --zone=public --add-port=9000/tcp --permanent
firewall-cmd --reload
```

- Pour valider l'ouverture des ports:
 - Se connecter au conteneur cible: **docker exec -it <mycontainer> bash**
 - Lancer la commande: **wget http://172.17.0.1:9000**

I

99

SONAR - ANALYSE DE QUALITE DE CODE

- Ajouter une étape dans le fichier Jenkinsfile, qui déclenche l'analyse de qualité de code avec sonar
- Push des modifications sur GITHUB

```
git add .
git commit -m "ajout conf sonar"
git push origin master
```

- Aller sur jenkins puis re-lancer un build pour tester

I

100



QUESTIONS / RÉPONSES

I

101