

Jenkins

PARTIE I Mise en place de Jenkins en local

Sur le site Jenkins : <https://jenkins.io/> récupérer le war jenkins.war (version : LTS Release 2.7.4)

Créer un répertoire nommé **SERVEUR_JENKINS**

Copier le .war dans le répertoire ../TravauxGSE-3\ **SERVEUR_JENKINS** \jenkins

Définir la **variable d'environnement** : JENKINS_HOME avec pour valeur le chemin du répertoire jenkins

Dans le répertoire **SERVEUR_JENKINS** au même niveau que « jenkins », créer un répertoire « gse3Repository ». Dans ce répertoire, créer le fichier « gradle.properties » contenant :

```
systemProp.http.proxyHost=proxy
systemProp.http.proxyPort=3128
systemProp.https.proxyHost=proxy
systemProp.https.proxyPort=3128
systemProp.http.proxyUser=deploy
systemProp.http.proxyPassword=Deploy
systemProp.http.nonProxyHosts=127.0.0.1;192.168.*.*;*.intranet.artal.fr
```

Pour démarrer Jenkins, exécuter "java -jar jenkins.war".

Dans un navigateur internet, connectez-vous à l'adresse <http://localhost:8080> et suivez les instructions :

- **Saisir le mot de passe d'initialisation** (se trouvant le répertoire jenkins\secrets)
- Configurer le proxy
- Créer le 1^{er} utilisateur administrateur
 - o **Nom d'utilisateur** : admin
 - o **Mot de passe** : admin

Gestion des plugins

Aller dans Administrer Jenkins>Gestion des plugins

Dans l'onglet « Disponibles » sélectionner :

- Git plugin
- Gradle Plugin

- EnvInject Plugin
- Checkstyle Plug-in
- FindBugs Plug-in
- PMD Plug-in
- Static Analysis Collector Plug-in
- Static Analysis Utilities
- jaCoCo plugin
- Post build task

Remarque : Si le proxy n'est pas bien configuré, il y aura aucun résultat.

Cliquer sur « Installer sans redémarrer »

Dans le répertoire **SERVEUR_JENKINS** créer un répertoire « gse3-EnvDev »

Copier les répertoires « gradle-2.13 », « innosetup », « jdk-8u77-64 » contenu dans C:\....\GSE3\gse3EnvDev (répertoire du projet GSE3) dans le répertoire C:\...\SERVEUR_JENKINS\gse3-EnvDev que l'on vient de créer.

Télécharger GIT

Sur le site <https://git-scm.com/download/win> Windows Portable **2.9.2**

Extraire le GIT dans le répertoire SERVEUR_JENKINS\gse3EnvDev\git

Configuration des outils

Aller dans Administrer Jenkins> Configuration globale des outils.

Dans la section **JDK**, cliquer sur « Installations JDK... », puis « Installer JDK »

Décocher « Install automatically », définir le nom et le chemin vers le répertoire de la JDK « jdk-8u77-64 »

JDK

Installations JDK

JDK
<div> <div>Nom</div> <div>jdk-8u77-64</div> </div> <div> <div>JAVA_HOME</div> <div>C:\TravauxGSE3\SERVEUR_JENKINS\gse3EnvDev\jdk-8u77-64</div> </div> <div> <input type="checkbox"/> Install automatically </div>

Ajouter JDK

Supprimer JDK

Nombre d'installations JDK sur ce système

Dans la section **Git**, décocher « Install automatically », définir le nom et le chemin vers git.exe

Git

Git installations

Git

Name: Default

Path to Git executable: C:\TravauxGSE3\SERVEUR_JENKINS\gse3EnvDev\git\bin\git.exe

☐ Install automatically

Delete Git

Add Git

description

Dans la section **Gradle**, cliquer sur « Installations Gradle... », décocher « Install automatically », définir le nom et le chemin vers le répertoire de gradle-2.13

Gradle

Installations Gradle

Gradle

name: 2.13

GRADLE_HOME: C:\TravauxGSE3\SERVEUR_JENKINS\gse3EnvDev\gradle-2.13

☐ Install automatically

Supprimer Gradle

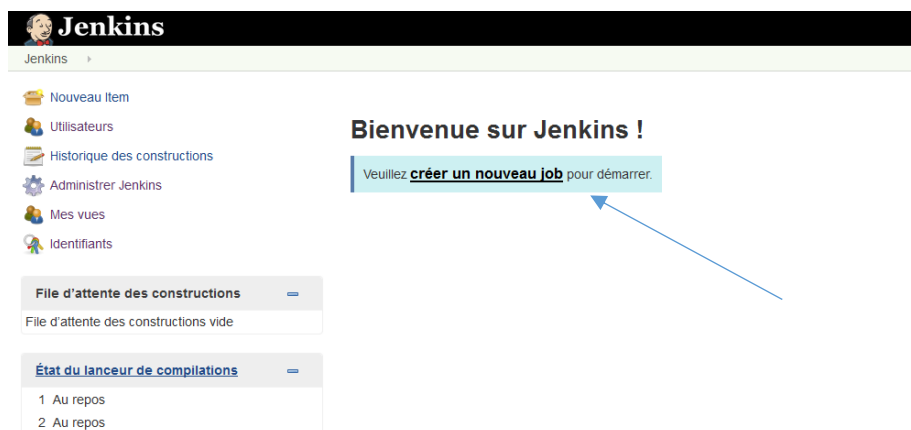
Ajouter Gradle

Nombre d'installations Gradle sur ce système

Enregistrer

Ajouter un JOB

Sur l'écran d'accueil de Jenkins, cliquer sur « créer un nouveau job ».



Créer un nouveau job nommé « GSE3 develop » et de type « projet free-style »

Dans la section « Gestion de code source », sélectionner Git.

- Définir le « repository URL » ssh://generic_user_name@extranet.artal.fr:29418/GSE3
- Cliquer sur Ajouter (pour définir le credential)
- Définir le credential associé
- Se positionner sur la branche : */develop

Jenkins Credentials Provider: Jenkins

Ajouter des identifiants

Domain: Identifiants globaux (illimité)

Type: SSH Username with private key

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

Username: ba

Private Key: ☒ Enter directly

Key:

```

PuTTY-User-Key-File-2: ssh-rsa
Encryption: none
Comment: rsa-key-20161003
Public-Lines: 6
AAAAB3NzaC1vc2EAAAABJQAAAQEA4eLQW01HKj3fCSXj6A3MzQXr3gJIZMhzzJXc

```

☐ From a file on Jenkins master

☐ From the Jenkins master ~/.ssh

Passphrase:

ID:

Gestion de code source

☐ Aucune

☒ Git

Repositories

Repository URL: ssh://ba@extranet.artal.fr:29418/GSE3

Credentials: ba [Ajouter](#)

[Avancé...](#)

[Add Repository](#)

Branches to build

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

[Add Branch](#)

Navigateur de la base de code: (Auto)

Additional Behaviours: [Ajouter](#)

Dans « Environnements de Build », cocher « Inject environment variables to the build process » et définir les propriétés :

JAVA_HOME= C:/TravauxGSE3/GSE3/gse3EnvDev/jdk-8u77-64

GRADLE_USER_HOME= C:/TravauxGSE3/SERVEUR_JENKINS/gse3Repository

GRADLE_HOME= C:/TravauxGSE3/SERVEUR_JENKINS/gse3EnvDev/gradle-2.13

INNO_HOME= C:/TravauxGSE3/SERVEUR_JENKINS/gse3EnvDev/innosetup

☒ Inject environment variables to the build process

Properties File Path

Properties Content

```
JAVA_HOME= C:/TravauxGSE3/GSE3/gse3EnvDev/jdk-8u77-64
GRADLE_USER_HOME= C:/TravauxGSE3/SERVEUR_JENKINS/gse3Repository
GRADLE_HOME= C:/TravauxGSE3/SERVEUR_JENKINS/gse3EnvDev/gradle-2.13
INNO_HOME= C:/TravauxGSE3/SERVEUR_JENKINS/gse3EnvDev/innosetup
```

Script File Path

Script Content

Evaluated Groovy script

☐ Inject passwords to the build as environment variables

Remarque : Dans Properties content (de Environnements Build), mettre des slashes « / » et non des backslashes « \ » pour le chemin vers les répertoires JAVA_HOME et GRADLE_USER_HOME.

Dans Section « Build » :

- Sélectionner la version de Gradle « 2.13 »
- Tasks: clean build
- Root Build script: \${workspace}\gse3-parent
- Cocher « Pass job parameters as Gradle properties » **(Utile ?)**

Build

Invoke Gradle script

☒ Invoke Gradle

Gradle Version

2.13

Use Gradle Wrapper

☐ Use Gradle Wrapper

Build step description

Switches

Tasks

clean
build

Root Build script

\${workspace}/gse3-parent

Build File

Force GRADLE_USER_HOME to use workspace

☐

Pass job parameters as Gradle properties

☒

Specify Gradle build file to run. Also, [some environment variables are available to the build script](#)

Dans « Actions à la suite du build », ajouter les 7 actions suivantes :

- [Publier les résultats d'analyse FindBugs](#)

Publier les résultats d'analyse FindBugs

FindBugs results

**/build/reports/findbugs/main.xml

[Fileset includes](#) setting that specifies the generated raw FindBugs XML report files, such as `**/findbugs.xml` or `**/findbugsXml.xml`. Basedir of the fileset is [the workspace root](#). If no value is set, then the default `**/findbugsXml.xml` or `**/findbugs.xml` are used for maven or ant builds, respectively. Be sure not to include any non-report files into this pattern.

Use rank as priority

☐

Uses the bug rank when evaluating the priority of the warnings (otherwise the FindBugs priority is used).

Avancé...

- [Publier les résultats d'analyse Checkstyle](#)

Publier les résultats de l'analyse Checkstyle

Checkstyle results

**/reports/checkstyle/test.xml

[Fileset includes](#) setting that specifies the generated raw CheckStyle XML report files, such as `**/checkstyle-result.xml`. Basedir of the fileset is [the workspace root](#). If no value is set, then the default `**/checkstyle-result.xml` is used. Be sure not to include any non-report files into this pattern.

Avancé...

- [Publier les résultats d'analyse PMD](#)

Publier les résultats de l'analyse PMD

PMD results

Fileset includes setting that specifies the generated raw PMD XML report files, such as **/pmd.xml. Basedir of the fileset is [the workspace root](#). If no value is set, then the default **/pmd.xml is used. Be sure not to include any non-report files into this pattern.

Avancé...

- [Publish combined analysis results](#)

Publish combined analysis results

Checkstyle warnings ☒

FindBugs warnings ☒

PMD warnings ☒

Avancé...

- [Pour voir les rapports de Test JUnit :](#)

Toujours dans « **Ajouter une action après le build** », cocher publier le rapport des résultats des tests JUnit puis définir le chemin du fichier .xml du test.

Publier le rapport des résultats des tests JUnit

XML des rapports de test

Une configuration du type [Fileset 'includes'](#) qui indique où se trouvent les fichiers XML des rapports de test, par exemple 'myproject/target/test-reports/*.xml'. Le répertoire de base (basedir) du fileset est [la racine du workspace](#).

☐ Retain long standard output/error

Health report amplification factor

1% failing tests scores as 99% health. 5% failing tests scores as 95% health

Avancé...

- [Jacoco coverage dans Jenkins:](#)

- 1- Ajouter une action après le build -> Record JaCoCo coverage report
- 2- Afficher le coverage

Record JaCoCo coverage report

Path to exec files (e.g.: ****/target/*.exec, **/jacoco.exec**) Path to class directories (e.g.: ****/target/classDir, **/classes**) Path to source directories (e.g.: ****/mySourceFiles**)

Inclusions (e.g.: ****/*.class**) Exclusions (e.g.: ****/*Test*.class**)

	Instruction	% Branch	% Complexity	% Line	% Method	% Class
	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

☐ Change build status according the thresholds

- [Post Build Task: launching the packaging script :](#)

Ajouter une action après le build -> Post build task

Sélectionner «Run script only if all previous steps were successful» et «Escalate script execution status to job status»

Post build task

Tasks

Log text

Operation -- OR --

Ajouter

Script

gse3-parent\build.bat

Run script only if all previous steps were successful ☒

Escalate script execution status to job status ☒

Add another task

Remarque : Record jaCoCo coverage report, se positionne avant le Post build task.

A la fin de l'exécution du build.bat le programme fait un clean ce qui efface tous les tests générés, d'où l'intérêt de positionner le jaCoCo coverage avant le post build task.

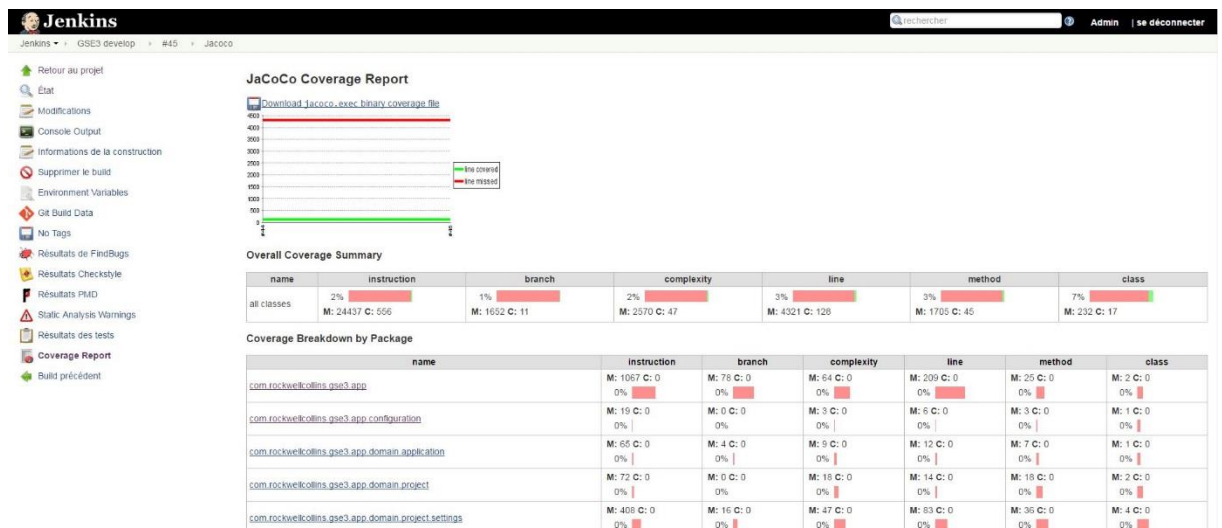
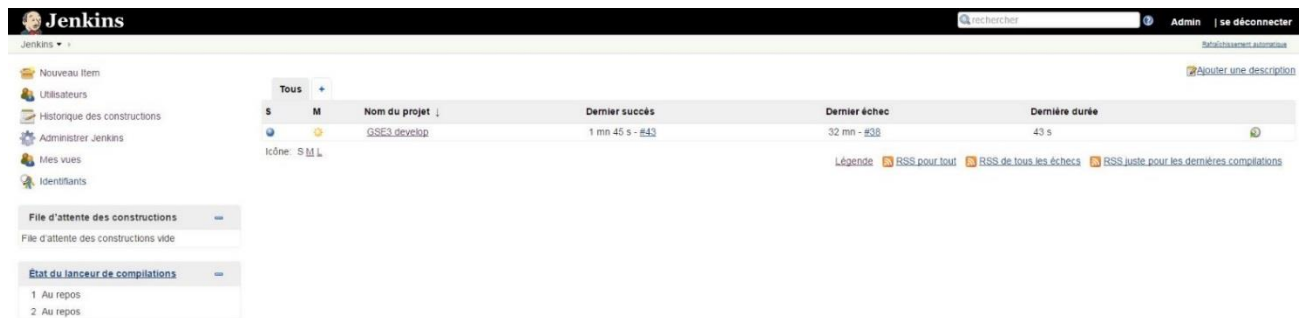
Apply puis Sauver (pour sauvegarder)

Lancement du build



Lancer un build

RESULTATS :



Sortie console

Jenkins

Jenkins

GSE3 develop

#77

Retour au projet

Etat

Modifications

Console Output

View as plain text

Informations de la construction

Supprimer le build

Environment Variables

Git Build Data

No Tags

Résultats de Findings

Résultats Checkstyle

Résultats PMD

Static Analysis Warnings

Résultats des tests

Coverage Report

Build précédent

Sortie de la console

Démarré par l'utilisateur admin

[EnvInject] - Loading node environment variables.

Building in workspace C:\Travaux\GSE3\SERVER_JENKINS\jenkins\workspace\GSE3 develop

> C:\Travaux\GSE3\SERVER_JENKINS\gse3\env\git\bin\git.exe rev-parse --is-inside-work-tree --timeout=10

Fetching changes from the remote git repository

> C:\Travaux\GSE3\SERVER_JENKINS\gse3\env\git\bin\git.exe config remote.origin.url ssh://ba@extranet.artal.fr:29418/GSE3 # timeout=10

Fetching upstream changes from ssh://ba@extranet.artal.fr:29418/GSE3

> C:\Travaux\GSE3\SERVER_JENKINS\gse3\env\git\bin\git.exe --version # timeout=10

using GIT_SSH to set credentials

> C:\Travaux\GSE3\SERVER_JENKINS\gse3\env\git\bin\git.exe fetch --progress ssh://ba@extranet.artal.fr:29418/GSE3 --refs/heads/*:refs/remotes/origin/*

> C:\Travaux\GSE3\SERVER_JENKINS\gse3\env\git\bin\git.exe rev-parse --refs/remotes/origin/develop [commit] # timeout=10

> C:\Travaux\GSE3\SERVER_JENKINS\gse3\env\git\bin\git.exe rev-parse --refs/remotes/origin/origin/develop [commit] # timeout=10

Checking out Revision 2512431a2081aaacbb4818dbaac5cf6dabaae (refs/remotes/origin/develop)

> C:\Travaux\GSE3\SERVER_JENKINS\gse3\env\git\bin\git.exe config core.sparsecheckout # timeout=10

> C:\Travaux\GSE3\SERVER_JENKINS\gse3\env\git\bin\git.exe checkout -f 2512431a2081aaacbb4818dbaac5cf6dabaae

> C:\Travaux\GSE3\SERVER_JENKINS\gse3\env\git\bin\git.exe rev-list --list 859fa497255a3a6e9f9f8e28a7855a701207a07 # timeout=10

[EnvInject] - Executing scripts and injecting environment variables after the SCM step.

[EnvInject] - Injecting as environment variables the properties content

GRADLE_USER_HOME=C:\Travaux\GSE3\SERVER_JENKINS\gse3\repository

JAVA_HOME=C:\Travaux\GSE3\GSE3\gse3\env\jdk-8u77-d8

GRADLE_HOME=C:\Travaux\GSE3\SERVER_JENKINS\gse3\env\gradle-2.13

IMMGR_HOME=C:\Travaux\GSE3\SERVER_JENKINS\gse3\env\immoSetup

[EnvInject] - Variables injected successfully.

[gradle] - Launching build.

[gse3-parent] \$ cmd.exe /C "C:\Travaux\GSE3\SERVER_JENKINS\gse3\env\gradle-2.13\bin\gradle.bat clean build && exit %ERRORLEVEL%"

Using online dependencies resolution...

Using cots repositories : [https://repo.maven.org/maven2/]

using gradle repository : https://plugins.gradle.org/m2/

Processing profile properties for gse3-app

No target profile defined, no target properties loaded.

No user profile defined, no user properties loaded.

Processing profile properties for gse3-common

Target properties file does not exist : C:\Travaux\GSE3\SERVER_JENKINS\jenkins\workspace\GSE3 develop\gse3-parent\gradle-target-HOME.properties

Target properties file does not exist : C:\Travaux\GSE3\SERVER_JENKINS\jenkins\workspace\GSE3 develop\gse3-common\gradle-target-HOME.properties

User properties file does not exist : C:\Travaux\GSE3\SERVER_JENKINS\jenkins\workspace\GSE3 develop\gse3-parent\gradle-user-HOME.properties

User properties file does not exist : C:\Travaux\GSE3\SERVER_JENKINS\jenkins\workspace\GSE3 develop\gse3-common\gradle-user-HOME.properties

clean

gse3-app:clean

gse3-common:clean

gse3-common-test:clean

gse3-coreGeneric:clean

gse3-cores350:clean

compileJava

processResources

classes

jar

assemble

compileTestJava

processTestResources

test:classes

gse3-app:assemble

gse3-app:check

gse3-app:build

gse3-common:sourceJar

gse3-common:assemble

gse3-common:check

gse3-common:build

gse3-common-test:compileJava

gse3-common-test:processResources UP-TO-DATE

gse3-common-test:classes

gse3-common-test:jar

gse3-common-test:sourceJar

gse3-common-test:assemble

gse3-common-test:check

gse3-common-test:build

gse3-coreGeneric:sourceJar

gse3-coreGeneric:assemble

gse3-coreGeneric:check

gse3-coreGeneric:build

gse3-cores350:sourceJar

gse3-cores350:assemble

gse3-cores350:check

gse3-cores350:build

gse3-app:fixJar

gse3-app:fixNative

Running [C:\Travaux\GSE3\GSE3\gse3\env\jdk-8u77-64\jre\bin\java, -version]

Creating app bundle: gse3 in C:\Travaux\GSE3\SERVER_JENKINS\jenkins\workspace\GSE3 develop\gse3-app\build\jfx\native

Using custom package resource [application icon] (loaded from package\windows\gse3.ico)

Running [C:\Users\ba\AppData\Local\Temp\Iconspad414409386\64x39580.exe, C:\Users\ba\AppData\Local\Temp\fixbundle17028964113581142422\windows\gse3.ico, C:\Travaux\GSE3\SERVER_JENKINS\jenkins\workspace\GSE3 develop\gse3-app\build\jfx\native\gse3\gse3.exe]

Icon File Name: C:\Users\ba\AppData\Local\Temp\fixbundle17028964113581142422\windows\gse3.ico

Executable File Name: C:\Travaux\GSE3\SERVER_JENKINS\jenkins\workspace\GSE3 develop\gse3-app\build\jfx\native\gse3\gse3.exe

Result application bundle: C:\Travaux\GSE3\SERVER_JENKINS\jenkins\workspace\GSE3 develop\gse3-app\build\jfx\native

Config files are saved to C:\Users\ba\AppData\Local\Temp\fixbundle17028964113581142422\windows. Use them to customize package.

Running [C:\Travaux\GSE3\GSE3\gse3\env\jdk-8u77-64\jre\bin\java, -version]

Detected [jssc.exe] version 0 but version 5 is required.

Running [C:\Travaux\GSE3\GSE3\gse3\env\jdk-8u77-64\jre\bin\java, -version]

Detected [candle.exe] version 0 but version 3 is required.

Detected [light.exe] version 0 but version 3 is required.

gse3-app:generate_exe

C:\Travaux\GSE3\SERVER_JENKINS\jenkins\workspace\GSE3 develop\gse3-app\build\...\gse3-app\build\installer,"

BUILD SUCCESSFUL

Total time: 56,949 secs

"Installer generated in "C:\Travaux\GSE3\SERVER_JENKINS\jenkins\workspace\GSE3 develop\gse3-parent\...\gse3-app\build\installer,"

POST BUILD TASK : SUCCESS

END OF POST BUILD TASK : 0

Finished: SUCCESS

- Importation et exportation configuration Jenkins:

Dans Administrer Jenkins → Gestion des plugins → Disponibles sélectionner : backup plugin → Installer sans redémarrer

Retour dans Administrer Jenkins → Il y a **Gestionnaire de backup** qui apparait

A partir de là, on peut importer ou exporter une configuration.

PARTIE II Création d'un nœud esclave sur un serveur Jenkins

Aller sur <http://kador.intranet.artal.fr:8080/>

S'identifier

Configuration des outils

Aller dans Administrer Jenkins> Configuration globale des outils.

Dans la section **JDK**, cliquer sur « Installations JDK... », « Ajouter JDK » puis « Installer JDK »

Décocher « Install automatically », définir le nom et le chemin vers le répertoire de la JDK « jdk-8u77-64 » de la machine esclave.

The screenshot shows the 'Add JDK' configuration form in Jenkins. The 'JDK' icon is visible. The 'Nom' field contains 'JDK 8 Islamabad'. The 'JAVA_HOME' field contains 'C:\TravauxGSE3\SERVEUR_JENKINS\gse3EnvDev\jdk-8u77-64'. A yellow warning message below the field states: 'C:\TravauxGSE3\SERVEUR_JENKINS\gse3EnvDev\jdk-8u77-64 is not a directory on the Jenkins master (but perhaps it exists on some slaves)'. The 'Install automatically' checkbox is unchecked. A red button labeled 'Supprimer JDK' is at the bottom right.

Dans la section **Gradle**, cliquer sur « Installations Gradle... », décocher « Install automatically », définir le nom et le chemin vers le répertoire de gradle-2.13 de la machine esclave

The screenshot shows the 'Add Gradle' configuration form in Jenkins. The 'Gradle' icon is visible. The 'name' field contains 'Gradle islamabad'. The 'GRADLE_HOME' field contains 'C:\TravauxGSE3\SERVEUR_JENKINS\gse3EnvDev\gradle-2.13'. A yellow warning message below the field states: 'C:\TravauxGSE3\SERVEUR_JENKINS\gse3EnvDev\gradle-2.13 is not a directory on the Jenkins master (but perhaps it exists on some slaves)'. The 'Install automatically' checkbox is unchecked. A red button labeled 'Supprimer Gradle' is at the bottom right.

Remarque : Si Gradle n'apparaît dans la liste des configurations, c'est que le plugin Gradle n'est pas installé dans Jenkins donc il faut l'ajouter (voir la section [Gestion des plugins](#) de la partie I).

Dans la section **Git**, décocher « Install automatically », définir le nom et le chemin vers git.exe de la machine esclave

The screenshot shows the 'Add Git' configuration form in Jenkins. The 'Git' icon is visible. The 'Name' field contains 'Git islamabad'. The 'Path to Git executable' field contains 'C:\TravauxGSE3\SERVEUR_JENKINS\gse3EnvDev\git\bin\git.exe'. The 'Install automatically' checkbox is unchecked.

Création du nœud

Pour créer un nœud esclave sur le master :

Dans Administrer Jenkins -> Gérer les nœuds → Créer un nœud → Nom de la machine esclave →

→ Sélectionner Esclave passif
compléter les champs

Nom	islamabad	?
Description	Windows 7	?
Nb d'exécuteurs	1	?
Répertoire de travail du système distant	C:\TravauxGSE3\SERVEUR_JENKINS\jenkins\	?
Étiquettes		?
Utilisation	Utiliser cet esclave autant que possible	?
Méthode de lancement	Lancer les agents esclaves par JNLP	?
	Avancé...	
Disponibilité	Garder cet esclave en ligne autant que possible	?

Ajout des propriétés des nœuds :

Sélectionner « Emplacement des outils » et « Variable d'environnement ».

Cliquer sur Ajouter afin d'ajouter des emplacements

☒ Emplacement des outils

Liste des emplacements où les outils peuvent être trouvés

Nom	(JDK) JDK 8 Islamabad	
Site	C:\TravauxGSE3\SERVEUR_JENKINS\gse3EnvDev\jdk-8u77-64	Supprimer
Nom	(Git) Git islamabad	
Site	C:\TravauxGSE3\SERVEUR_JENKINS\gse3EnvDev\git\bin\git.exe	Supprimer
Nom	(Gradle) Gradle islamabad	
Site	C:\TravauxGSE3\SERVEUR_JENKINS\gse3EnvDev\gradle-2.13	Supprimer
Ajouter		

Dans variables d'environnement

☒ Variables d'environnement
Liste des paires clé-valeur


nom	GRADLE_HOME	
valeur	C:/TravauxGSE3/SERVEUR_JENKINS/gse3EnvDev/gradle-2.13	Supprimer
nom	GRADLE_USER_HOME	
valeur	C:/TravauxGSE3/SERVEUR_JENKINS/gse3Repository	Supprimer
nom	INNO_HOME	
valeur	C:/TravauxGSE3/SERVEUR_JENKINS/gse3EnvDev/innosetup	Supprimer
nom	JAVA_HOME	
valeur	C:/TravauxGSE3/SERVEUR_JENKINS/gse3EnvDev/jdk-8u77-64	Supprimer
nom	JENKINS_MODE	
valeur	OK	Supprimer

Ajouter

Ces variables, sont des variables d'environnement qui seront injectées lors du processus de build.

Enregistrer.

Connect slave to Jenkins one of these ways:

-  **Launch** Lancer l'agent à partir du navigateur sur l'esclave
- Run from slave command line:

```
java -jar slave.jar -jnlpUrl http://kador:8080/computer/zozor/slave-agent.jnlp -secret 1f59ff4cbb94934381af07beb61513960a25dddbcd9364706299f43b9d100bf5
```

Created by [BA Alpha-Amadou](#)

- Cliquer sur [slave.jar](#) et copier le fichier quelque part sur la machine local (esclave)

- Lancer la ligne de commande générée sur le terminal

```
java -jar slave.jar -jnlpUrl http://kador:8080/computer/zozor/slave-agent.jnlp -secret 1f59ff4cbb94934381af07beb61513960a25dddbcd9364706299f43b9d100bf5
```

La machine esclave est connectée au master (serveur).

Remarque :

Afin de lancer cette commande à chaque démarrage de Windows, on peut créer un fichier .bat en y copiant la ligne de commande et le sauvegarder dans le dossier Démarrage de Windows.

[Ajouter un projet](#)

- Se positionner dans Jenkins

Cliquer ici pour ajouter un JOB

Jenkins

rechercher ? BA Alpha Amadou | se déconnecter

Rafraîchissement automatique

Nouveau Item

Utilisateurs

Historique des constructions

Relations entre les builds

Vérifier les empreintes numériques

Administrer Jenkins

Query and Trigger Gerrit Patches

Mes vues

Leader board

Status Monitor

Sounds

Job Import Plugin

Ajouter une description

S	M	Nom du projet	Dernier succès	Dernier échec	Dernière durée
		AirbusMinor_SynSync	14 h - #1344	s. o.	19 s
		ALTO	s. o.	18 h - #55	16 mn
		Anonymous comments	2 an. 8 mo. - #516	2 an. 11 mo. - #70	3.8 s
		ASIA Deploy	1 an. 9 mo. - #16	1 an. 9 mo. - #15	5 mn 49 s
		Attractive3 RELEASE	s. o.	s. o.	ND
		Attractive3 Tag 1.0.3 RELEASE	2 an. 7 mo. - #5	s. o.	23 s
		Attractive3 Trunk	5 mo. 19 j. - #842	1 mo. 22 j. - #941	42 s

Créer un nouveau job nommé « GSE3 » et de type « projet free-style »

- Sélectionner « Activer la sécurité basée projet » pour donner des droits aux membres du projet.
- Supprimer les anciens builds : mettre nombre max de builds à conserver 10
- Choisir le JDK à utiliser pour ce projet « JDK 8 Islamabad »
- Restreindre où le projet peut être exécuté : mettre le nom du nœud esclave « islamabad »

[Safe HTML] [Prévisualisation](#)

☒ Activer la sécurité basée projet

☐ Block inheritance of global authorization matrix

Utilisateur/groupe	Identifiants	Job	Historique des builds	Gestion de version	Env. Inject
muno	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Anonyme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Utilisateur/groupe à ajouter:

☐ Ce build a des paramètres

☐ GitHub project

☒ Supprimer les anciens builds

Strategy

Log Rotation

Nombre de jours de conservation des builds

Nombre maximum de builds à conserver

si non vide, les enregistrements de build seront conservés au maximum ce nombre de jours

si non vide, pas plus de ce nombre de builds ne sera conservé

☐ Throttle builds

☐ Prepare an environment for the run

☐ Désactiver le projet (Aucun nouveau build ne sera exécuté jusqu'à ce que le projet soit réactivé.)

☐ Exécuter des builds simultanément si nécessaire

JDK

JDK 8 Islamabad

Le JDK à utiliser pour ce projet

☒ Restreindre où le projet peut être exécuté


Expression


islamabad

Label is serviced by 1 node

Dans la section « Gestion de code source », sélectionner Git.

- Définir le « repository URL » `ssh://generic_user_name@extranet.artal.fr:29418/GSE3`
- Cliquer sur Ajouter (pour définir le credential)
- Définir le credential associé
- Se positionner sur la branche : `*/develop`

 **Jenkins Credentials Provider: Jenkins**

 **Ajouter des identifiants**

Domain: Identifiants globaux (illimité)

Type: SSH Username with private key

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

Username: ba

Private Key: ☒ Enter directly

Key: `PuTTY-User-Key-File-2: ssh-rsa
Encryption: none
Comment: rsa-key-20161003
Public-Lines: 6
AAAAB3NzaC1yc2EAAAABJQAAQEA4eLQW01HKJ3fCSXi6A3MzQXr3gJIZMhzzJXc`

☐ From a file on Jenkins master
☐ From the Jenkins master ~/.ssh

Passphrase:


ID:


Gestion de code source

☐ Aucune
☐ CVS
☐ CVS Projectset
☒ Git

Repositories

Repository URL: `ssh://ba@extranet.artal.fr:29418/GSE3`

Credentials: `ba` 



Add Repository

Branches to build

Branch Specifier (blank for 'any'): `*/develop`

Add Branch Delete Branch

Git executable: `Git islamabad`

Navigateur de la base de code: `(Auto)`

Pour les étapes suivantes, faire pareil que la [Partie I](#)

Build :

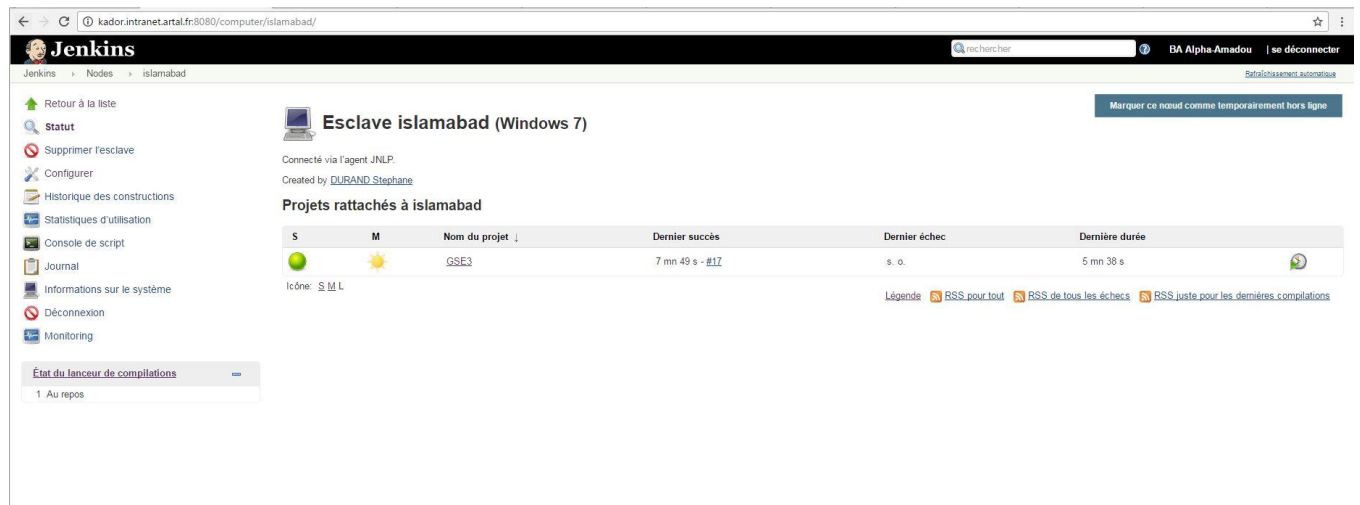
- Sélectionner la version de Gradle « 2.13 »
- Tasks: clean build
- Root Build script: `${workspace}\gse3-parent`

Dans « Actions à la suite du build », ajouter les 7 actions suivantes :

- Publier les résultats de l'analyse FindBus
- Publier les résultats de l'analyse Checkstyle
- Publier les résultats de l'analyse PMD
- Publish combined analysis results
- Publier le rapport des résultats des tests JUnit
- Record JaCoCo coverage report
- Post build task

Remarque : Comme dans la partie I, Record jaCoCo coverage report, se positionne avant le Post build task.

Appliquer puis Sauver (pour sauvegarder)



Dans nœud islamabad, il y a maintenant le projet GSE3 qui lui est rattaché.

Lancement du build sur le master



FIN.

Notes