

DevOps

CI / CD

Jenkins

Jamel ESSOUSSI: Architecte logiciel

jamel.essoussi@gmail.com

<https://github.com/jessoussi>

2025

Plan du cours

I. Introduction

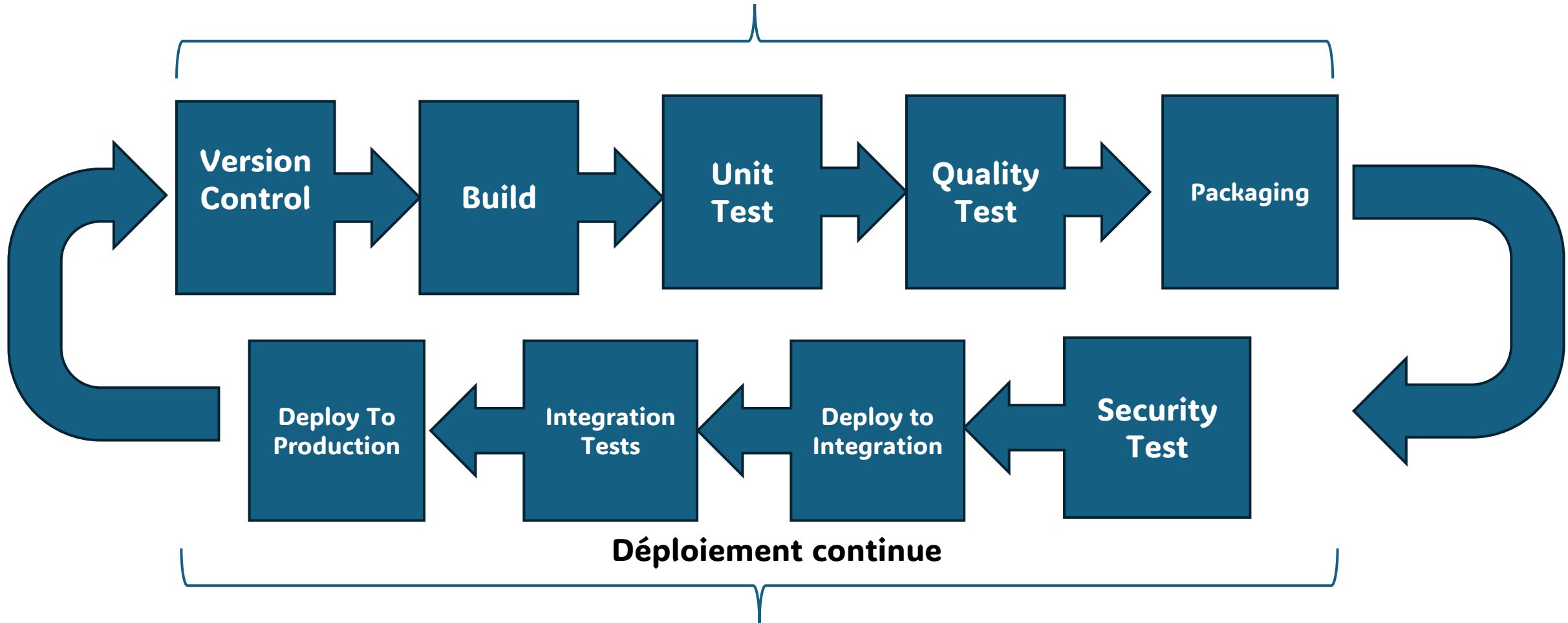
II. Job & Pipeline

III. Intégration avec les autres outils

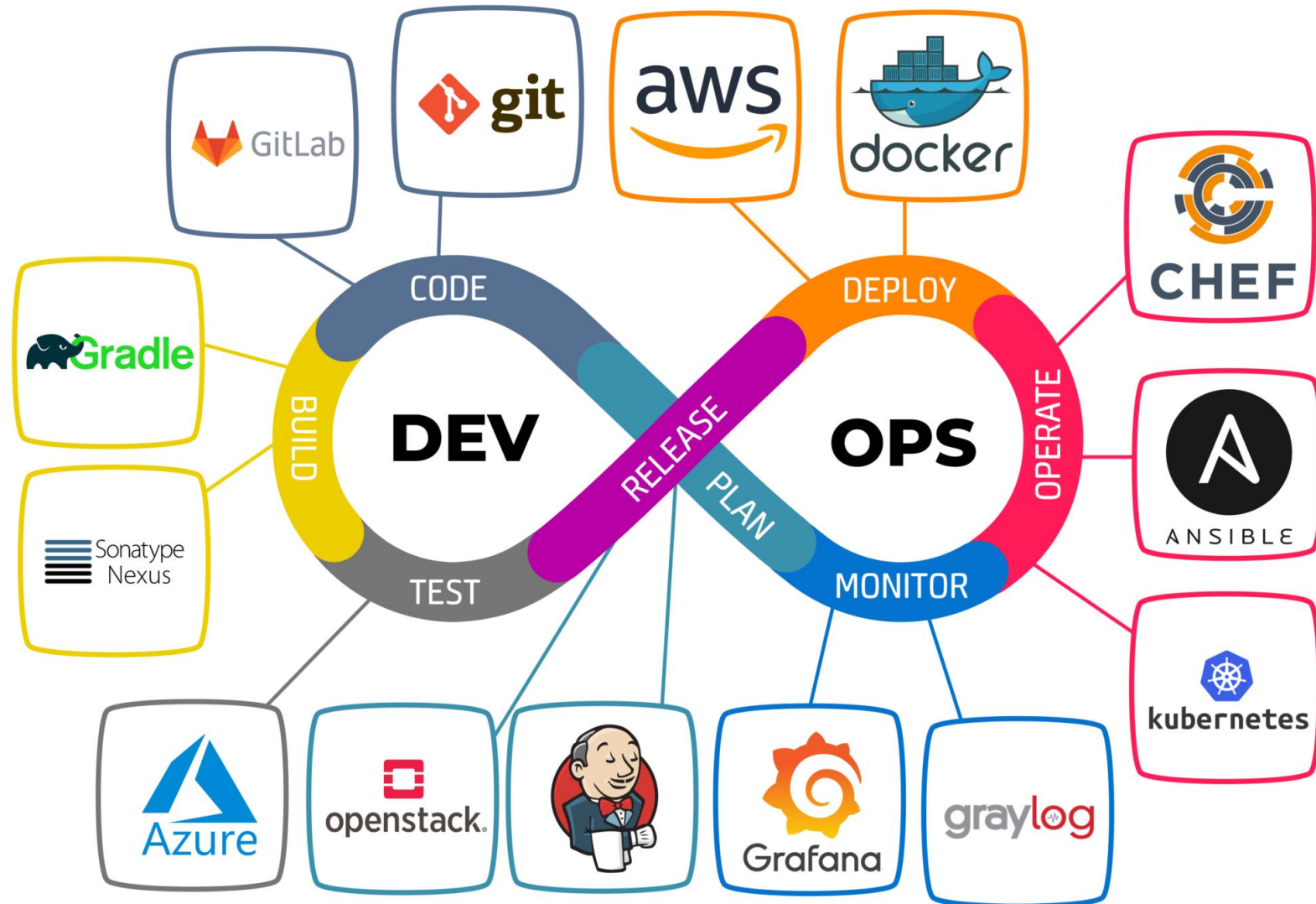
IV. Les bonnes pratiques

Introduction

Intégration continue



Outils du DevOps



Introduction

- Jenkins est un serveur d'automatisation open source qui aide à automatiser la construction, le test et le déploiement d'applications logicielles.
- Jenkins est un outil central pour la chaine DevOps

Introduction

- Il est écrit en Java et a été initialement publié en 2011 en tant que clone du projet Hudson.
- Jenkins est largement utilisé pour les pratiques d'intégration continue (CI) et de livraison continue (CD) dans le développement de logiciels.

Introduction

- En s'intégrant à divers outils de construction, systèmes de contrôle de version et cadres de test, Jenkins permet aux développeurs d'automatiser des tâches répétitives, telles que la construction de l'application, l'exécution de tests et le déploiement en production.

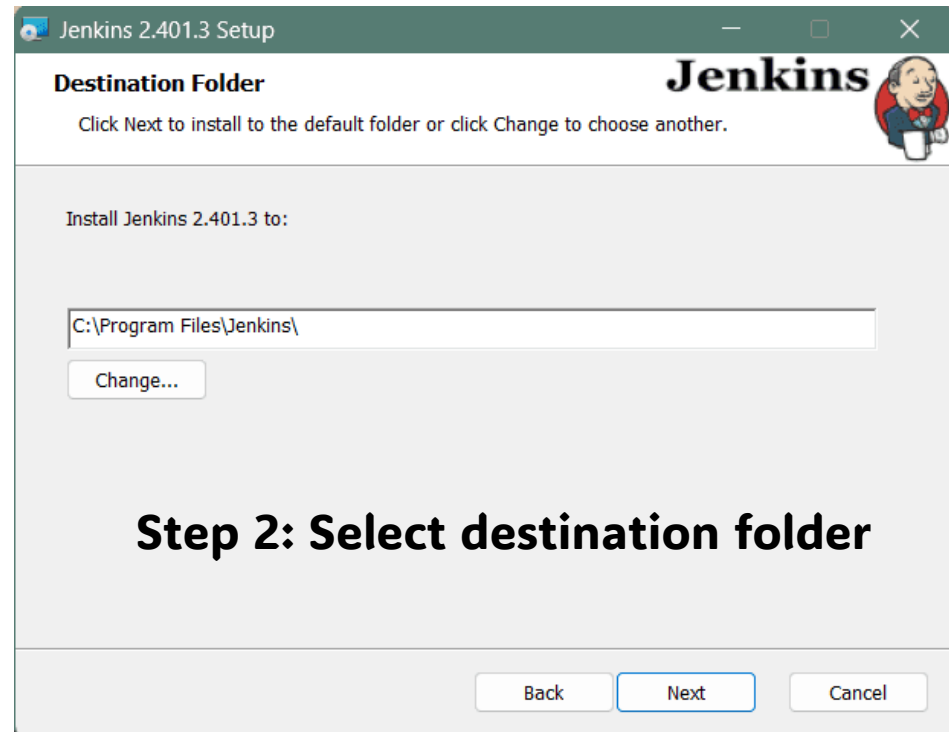
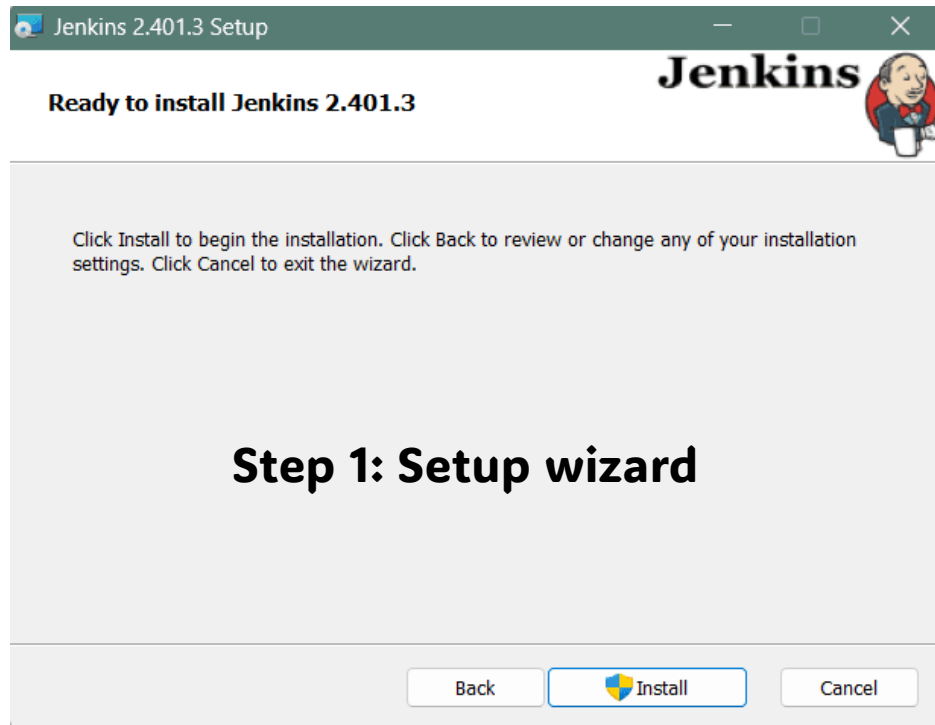
Introduction

- Cette automatisation permet d'augmenter la productivité, d'assurer la qualité du code et de permettre une livraison plus rapide des mises à jour logicielles.

Installation Jenkins

○ Téléchargement

- <https://www.jenkins.io/download/#downloading-jenkins>



Installation Jenkins

Jenkins 2.401.3 Setup

Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.401.3 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.401.3 to run successfully.


Logon Type:

☐ Run service as LocalSystem (not recommended)

☒ Run service as local or domain user:

Account:

Password:

 Credentials must be tested to continue

Step 3: Service logon credentials


Jenkins 2.401.3 Setup

Port Selection

Choose a port for the service.

Please choose a port.

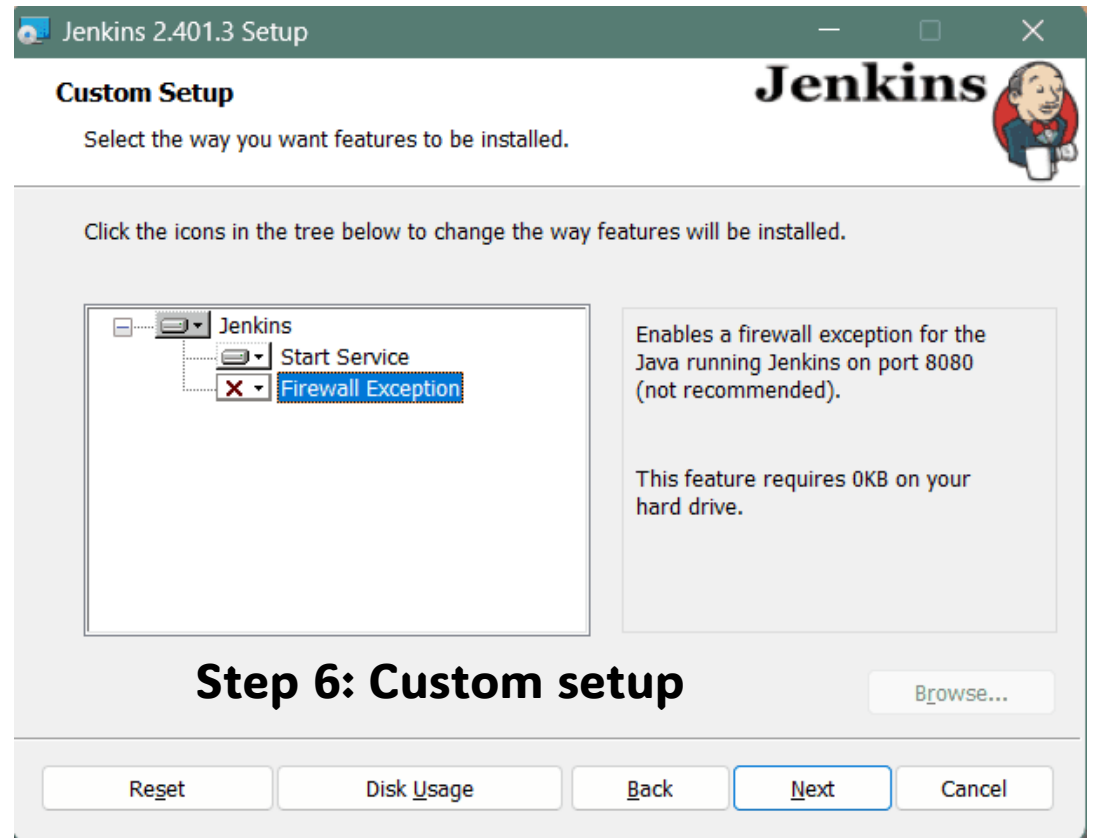
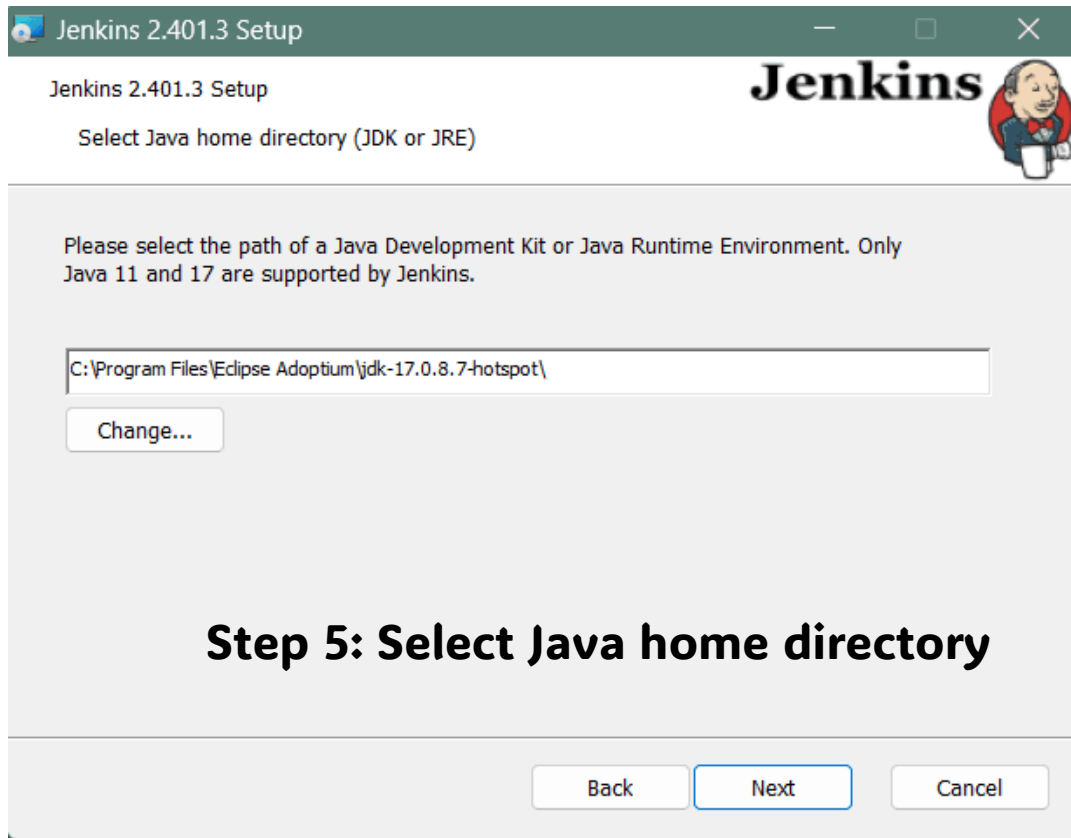
Port Number (1-65535):

 Click 'Test Port' button to proceed

It is recommended that you accept the selected default port.

Step 4: Port selection

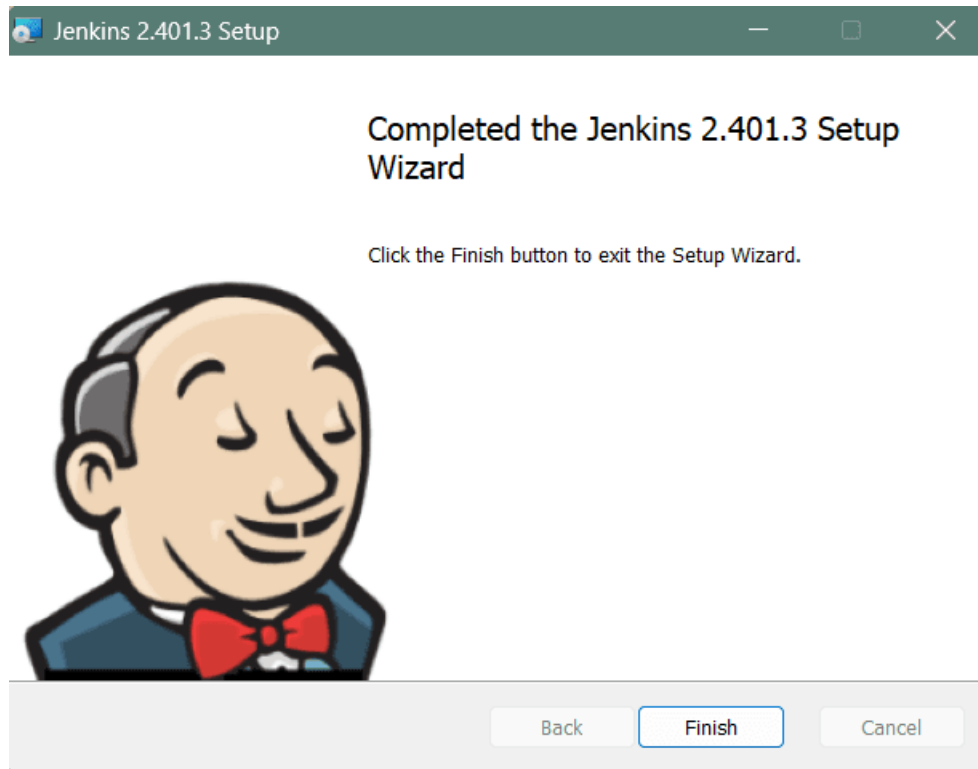
Installation Jenkins



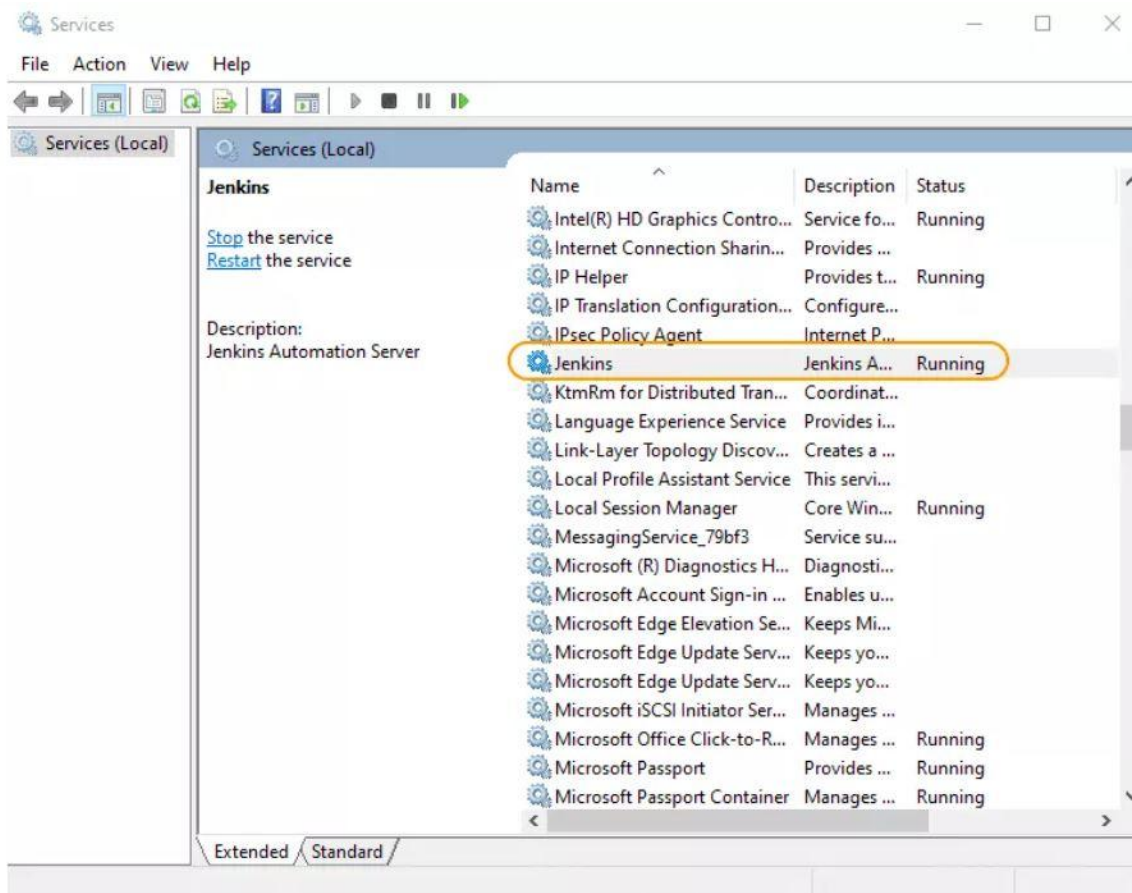
Installation Jenkins



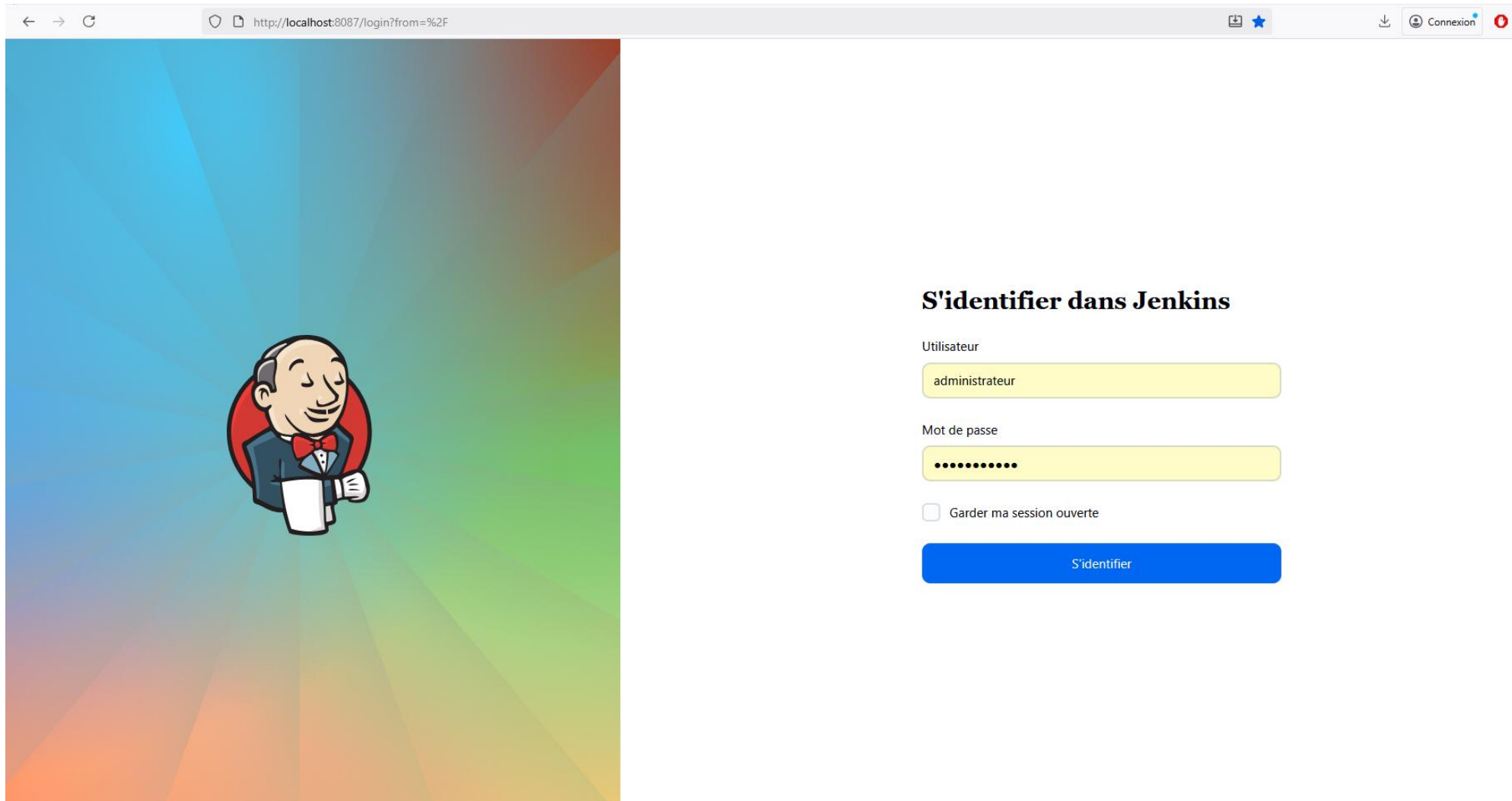
Installation Jenkins



Step 8: Finish Jenkins installation



Installation Jenkins



The screenshot shows a web browser window with the address bar displaying `http://localhost:8087/login?from=%2F`. The page features a large, colorful background image of the Jenkins mascot, a cartoon man in a tuxedo, holding a document. To the right of the mascot, the title "S'identifier dans Jenkins" is displayed. Below the title, there are two input fields: "Utilisateur" (User) with the text "administrateur" and "Mot de passe" (Password) with a masked password ".....". Below these fields is a checkbox labeled "Garder ma session ouverte" (Keep my session open). At the bottom of the login section is a blue button labeled "S'identifier" (Log in).

← → ↻ `http://localhost:8087/login?from=%2F` Connexion

S'identifier dans Jenkins

Utilisateur

administrateur

Mot de passe

.....

☐ Garder ma session ouverte

S'identifier

Plan du cours

I. Introduction

II. Job & Pipeline

III. Intégration avec les autres outils

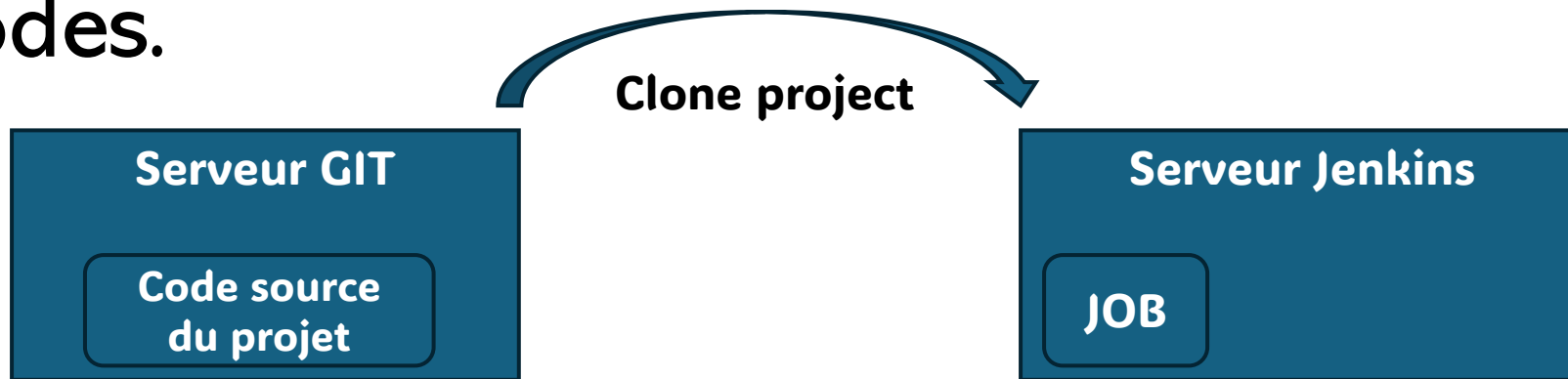
IV. Les bonnes pratiques

Job Jenkins

- Un Job dans Jenkins est une tâche qui peut être configurée et exécutée sur un serveur Jenkins, en fonction des besoins.

Job

- Avant de créer un Job sur Jenkins, il est essentiel d'avoir un serveur Jenkins installé et configuré, un projet prêt à être compilé, testé ou déployé, et un dépôt Git pour stocker les codes.



Job

Tableau de bord >

+ Nouveau Item

📋 Historique des constructions

😊 Relations entre les builds

🖐️ Vérifier les empreintes numériques

⚙️ Administrer Jenkins

📌 Mes vues

File d'attente des constructions ▾
File d'attente des constructions vide

État du lanceur de compilations 0/2 ▾

Cloud Statistics ▾

Job Jenkins

Ajouter une description

Tous +

S	M	Nom du projet ↓	Dernier succès	Dernier échec	Dernière durée	
✖	☁	devops-sample	6 mo. 1 j #53	27 j #57	3 mn 10 s	▶
✔	☀	kubectl	5 mo. 9 j #11	6 mo. 7 j #9	13 s	▶

Icône: S M L

+ Nouveau Item

Ajout d'un nouveau Job

✎ Ajouter une description

📁 Historique des constructions

🕒 Relations entre les builds

🔍 Vérifier les empreintes numériques

⚙️ Administrer Jenkins

📌 Mes vues

File d'attente des constructions

File d'attente des constructions vide

État du lanceur de compilations

0/2

Cloud Statistics

Tous

+

S	M	Nom du projet ↓	Dernier succès	Dernier échec	Dernière durée	
⊗	☁️	devops-sample	6 mo. 1 j #53	27 j #57	3 mn 10 s	▶
✅	☀️	kubectl	5 mo. 9 j #11	6 mo. 7 j #9	13 s	▶

Icône:

S

M

L

...

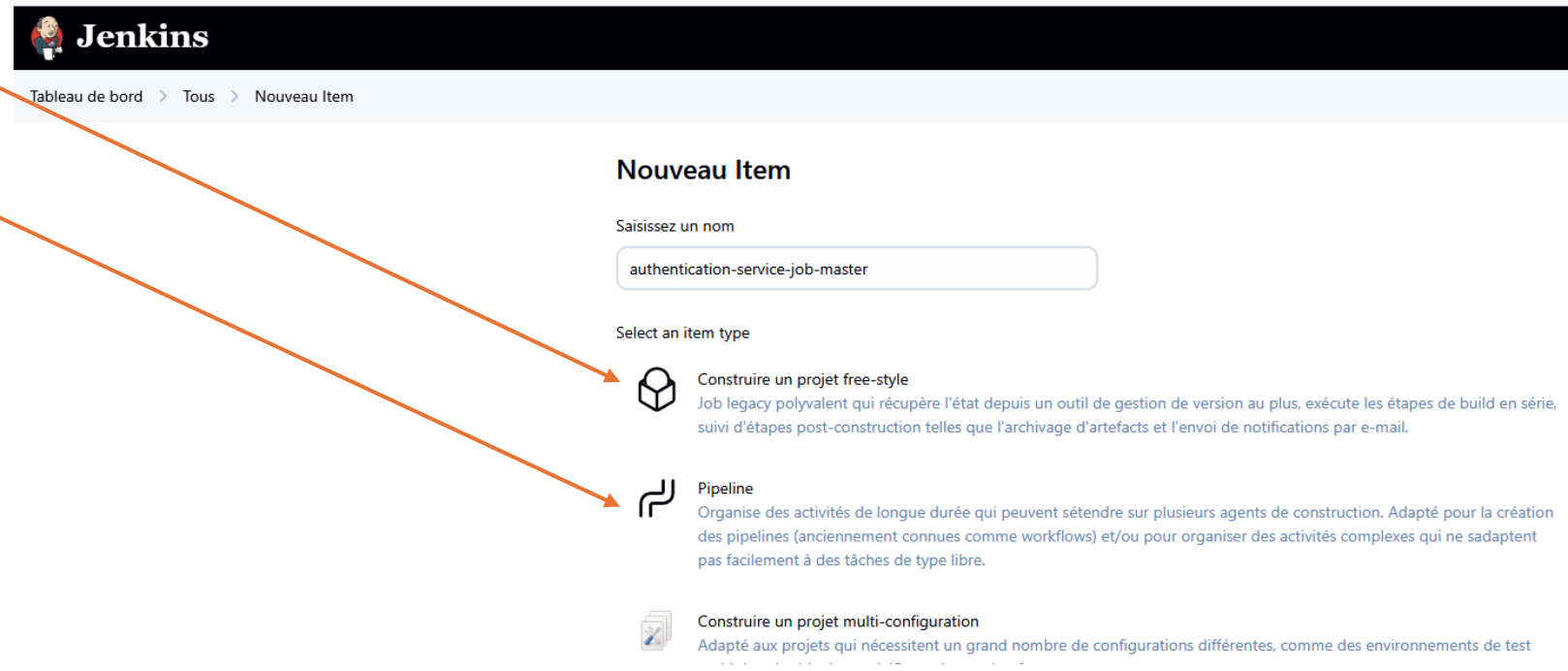
Job

- Plusieurs types de Job peuvent être créés:

- Freestyle

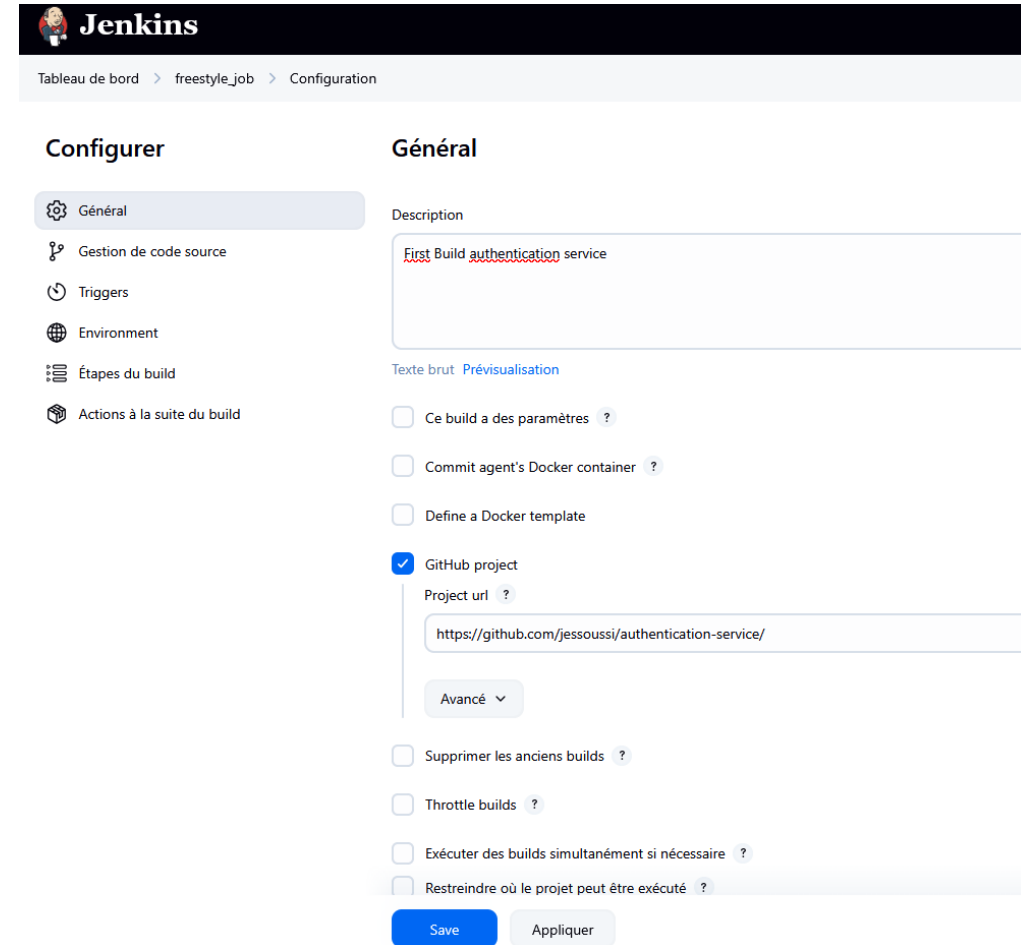
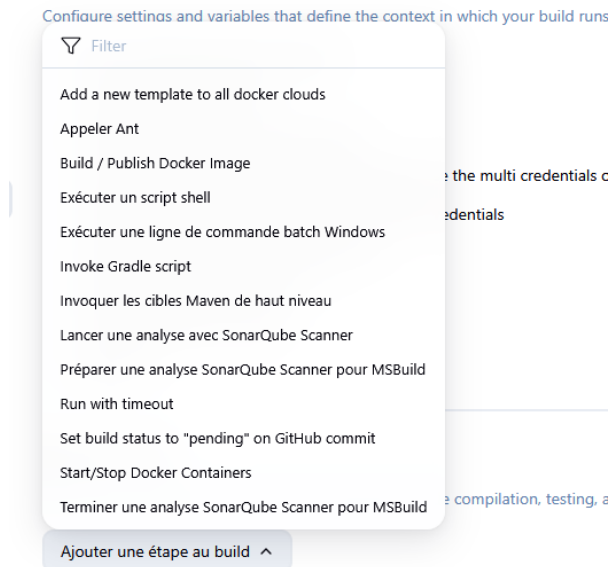
- Pipeline

- ...



Job

- Freestyle
 - Configuration manuelle du job
- GIT, étapes, Script, ...



Job

- Pipeline
 - Déclarative à travers des pipelines
 - Jenkinsfile

Tableau de bord > authentication-service-job-pipeline-master > Configuration

Configurer

- Général
- Triggers
- Pipeline**
- Advanced

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent any
3   tools {
4     maven 'M3'
5     jdk 'jdk17'
6   }
7   options {
8     skipStagesAfterUnstable()
9   }
10  stages {
11    stage('Git Checkout') {
12      steps {
13        script {
14          git branch: 'main', url: 'https://github.com/jessoussi/authentication-service/'
15        }
16      }
17    }
18  }
19 }
```

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

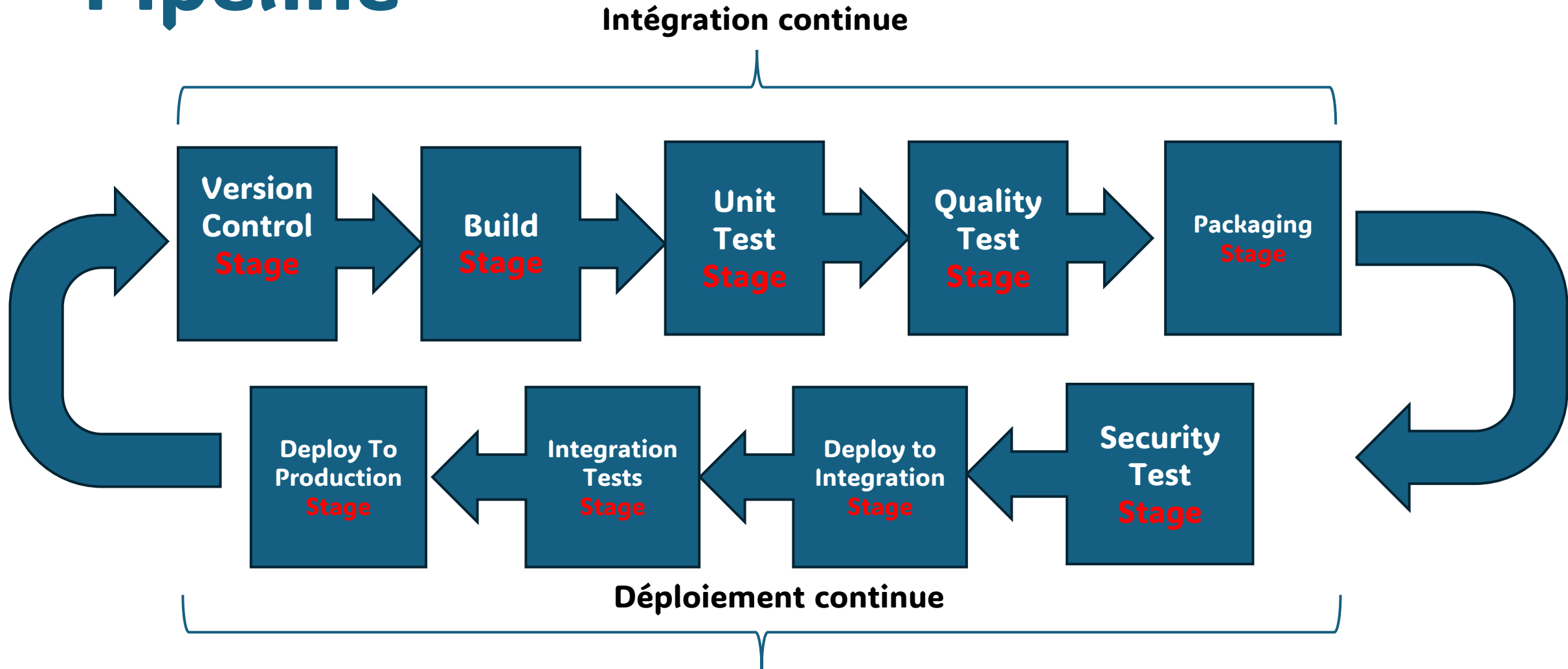
Advanced

Pipeline

- Dans Jenkins, on peut définir des pipelines pour décrire les étapes d'intégration et de déploiement continue.
- Les pipelines reposent sur le langage de script Groovy. La documentation dédiée à Jenkins est disponible sur cette page <https://www.jenkins.io/doc/book/pipeline/> .

Pipeline


Pipeline Jenkins



Pipeline Jenkinsfile

- Syntaxe (<https://www.jenkins.io/doc/book/pipeline/syntax/>)

```
pipeline {  
    /* insert Declarative Pipeline here */  
}
```

GROOVY | 

```
pipeline {  
    agent any  
    options {  
        // Timeout counter starts AFTER agent is allocated  
        timeout(time: 1, unit: 'SECONDS')  
    }  
    stages {  
        stage('Example') {  
            steps {  
                echo 'Hello World'  
            }  
        }  
    }  
}
```

Pipeline Jenkinsfile

- Syntaxe (<https://www.jenkins.io/doc/book/pipeline/syntax/>)

```
Jenkinsfile (Declarative Pipeline)
pipeline {
  agent any

  stages {
    stage('Build') {
      steps {
        echo 'Building..'
      }
    }
    stage('Test') {
      steps {
        echo 'Testing..'
      }
    }
    stage('Deploy') {
      steps {
        echo 'Deploying....'
      }
    }
  }
}
```

Pipeline

- Les scripts Pipeline apportent les notions suivantes :
 - **Pipeline**: est l'ensemble du processus à exécuter ;
 - **Node** : représente un environnement pouvant exécuter un pipeline (une machine esclave) ;
 - **Stage**: représente un ensemble d'étapes du processus (par exemple, la récupération des sources, la compilation...) ;
 - **Step**: représente ce qu'il y a à faire à un moment donné (l'action à proprement parler, comme maven).

Les concepts principaux

- Un pipeline (workflow) est une procédure automatisée associée à votre dépôt.
- Un pipeline est composé de un ou plusieurs jobs
 - Les jobs peuvent s'exécuter en parallèle par défaut
 - Possibilité de définir des dépendances entre les jobs (Concept de stage)
- Un job est composé de steps (étapes)
 - Une étape est une action/commande

Les concepts principaux

- Un job est exécuté par un Agent
 - Un agent en charge d'exécuter un job sur un serveur
 - Jenkins fournit des agents (et les ressources pour les exécuter)
 - Les jobs d'un pipeline sont exécutés dans des contextes d'exécutions différents
 - Les jobs ne peuvent pas partager de données par défaut
- Des événements déclenchent le lancement d'un pipeline

Les concepts principaux

- Les jobs peuvent s'exécuter en parallèle.

```
groovy
pipeline {
  agent any
  stages {
    stage('Parallèle') {
      parallel {
        stage('Python') {
          agent { docker 'python:latest' }
          steps {
            sh "python --version"
          }
        }
        stage('Java') {
          agent { docker 'openjdk:latest' }
          when {
            branch 'staging'
          }
          steps {
            sh "java -version"
          }
        }
      }
    }
  }
}
```

Les concepts principaux

- Utilisation de variables d'environnement

```
Jenkinsfile (Declarative Pipeline)
pipeline {
  agent {
    label '!windows'
  }

  environment {
    DISABLE_AUTH = 'true'
    DB_ENGINE    = 'sqlite'
  }

  stages {
    stage('Build') {
      steps {
        echo "Database engine is ${DB_ENGINE}"
        echo "DISABLE_AUTH is ${DISABLE_AUTH}"
        sh 'printenv'
      }
    }
  }
}
```

Les concepts principaux

- Utilisation des credentials
- Les credentials sont définis dans l'interface d'administration de Jenkins

Configuration du système



System

Configurer les paramètres généraux et les chemins de fichiers.



Tools

Configurer les outils, leur localisation et les installeurs automatiques.



Plugins

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



Nodes

Ajouter, supprimer, contrôler et monitorer les divers nœuds que Jenkins utilise pour exécuter les jobs.



Docker

Plugin for launching build Agents as Docker containers



Clouds

Ajouter, supprimer et configurer les instances de cloud afin de provisionner les agents à la demande.



Apparence générale

Configurer l'apparence générale de Jenkins

Sécurité



Security

Sécuriser Jenkins; définir qui est autorisé à accéder au système.



Credentials

Configure credentials



Credential Providers

Configure the credential providers and types



Users







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

Les concepts principaux


- Utilisation des credentials
- Liste des credentials

Tableau de bord > Administrer Jenkins > Identifiants

Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	kubectldcred	kubectldcred
		System	(global)	sonarqubecred	sonarqubecred
		System	(global)	github_credentials	jamel.essoussi@gmail.com/*****

Stores scoped to Jenkins

P	Store ↓	Domains
	System	(global)

Icône: S M L

Les concepts principaux

○ Utilisation des credentials

Jenkinsfile (Declarative Pipeline)

```
pipeline {
  agent {
    // Define agent details here
  }
  environment {
    AWS_ACCESS_KEY_ID = credentials('jenkins-aws-secret-key-id')
    AWS_SECRET_ACCESS_KEY = credentials('jenkins-aws-secret-access-key')
  }
  stages {
    stage('Example stage 1') {
      steps {
        // ❶
      }
    }
    stage('Example stage 2') {
      steps {
        // ❷
      }
    }
  }
}
```

Pipeline

- Pour définir un pipeline, il faut avoir le plugin Pipeline installé sur le serveur Jenkins.
- Ce plugin est pré-installé avec le serveur Jenkins.
- Deux Types de pipeline sont possibles:
 - Pipeline Déclaratives,
 - Pipeline Scripté.

Pipeline

- Un pipeline peut être créé de l'une des manières suivantes:
 - Dans Jenkins: Grâce à l'interface utilisateur de Jenkins.
 - Dans SCM: écrire un Jenkins file manuellement qui sera versionner avec le code source du projet dans GIT.
- La syntaxe pour définir un pipeline avec l'une ou l'autre approche est la même, utilisation de la syntaxe Groovy

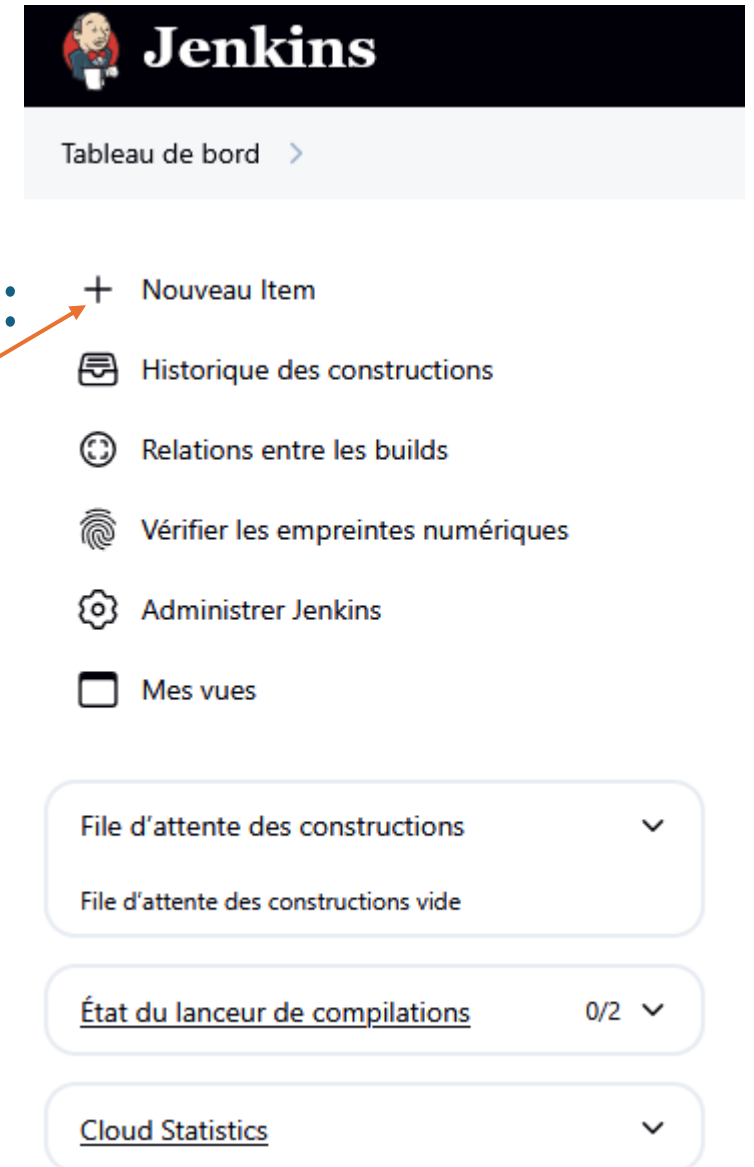
Pipeline

- Création à travers l'interface classique:
 - Un Jenkinsfile est créé à l'aide de l'interface utilisateur classique est stocké par Jenkins lui-même (à l'intérieur le répertoire d' accueil Jenkins).
 - Pour créer un pipeline de base à travers l'interface utilisateur classique de Jenkins:

Pipeline

○ Création à travers l'interface classique:

1 - Dans le tableau de bord Jenkins,
Selectionner nouvel élément:



Pipeline

○ Création à travers l'interface classique:

2 – Renseigner le nom de votre nouveau pipeline.

3 – Sélectionner Pipeline

4 - Valider



- Jenkins utilise ce nom d'élément pour créer des répertoires sur disque.
- C'est recommandé d'éviter d'utiliser des espaces dans les noms d'articles

Nouveau Item

Saisissez un nom

authentication-service-pipeline

Select an item type



Construire un projet free-style

Job legacy polyvalent qui récupère l'état depuis un outil de gestion de version au plus, exécute les étapes de build en série, suivi d'étapes post-construction telles que l'archivage d'artefacts et l'envoi de notifications par e-mail.



Pipeline

Organise des activités de longue durée qui peuvent s'étendre sur plusieurs agents de construction. Adapté pour la création des pipelines (anciennement connues comme workflows) et/ou pour organiser des activités complexes qui ne s'adaptent pas facilement à des tâches de type libre.



Construire un projet multi-configuration

Adapté aux projets qui nécessitent un grand nombre de configurations différentes, comme des environnements de test multiples, des binaires spécifiques à une plateforme, etc.

Pipeline

- Création à travers l'interface classique:

5- Cliquez sur Job créé.

6- Cliquez sur l'onglet Pipeline.

7- Sélectionnez le type du pipeline « Pipeline Script»

8- Entrez votre code Pipeline dans la zone de texte de script.

Pipeline

- Création à travers l'interface classique:

Configure

- General
- Advanced Project Options
- Pipeline**

Pipeline

Definition

Pipeline script

Script ?

```
1 pipeline {  
2   agent any  
3   stages {  
4     stage('Stage 1') {  
5       steps {  
6         echo 'Hello World'  
7       }  
8     }  
9   }  
10 }  
11
```

try sample Pipeline...

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

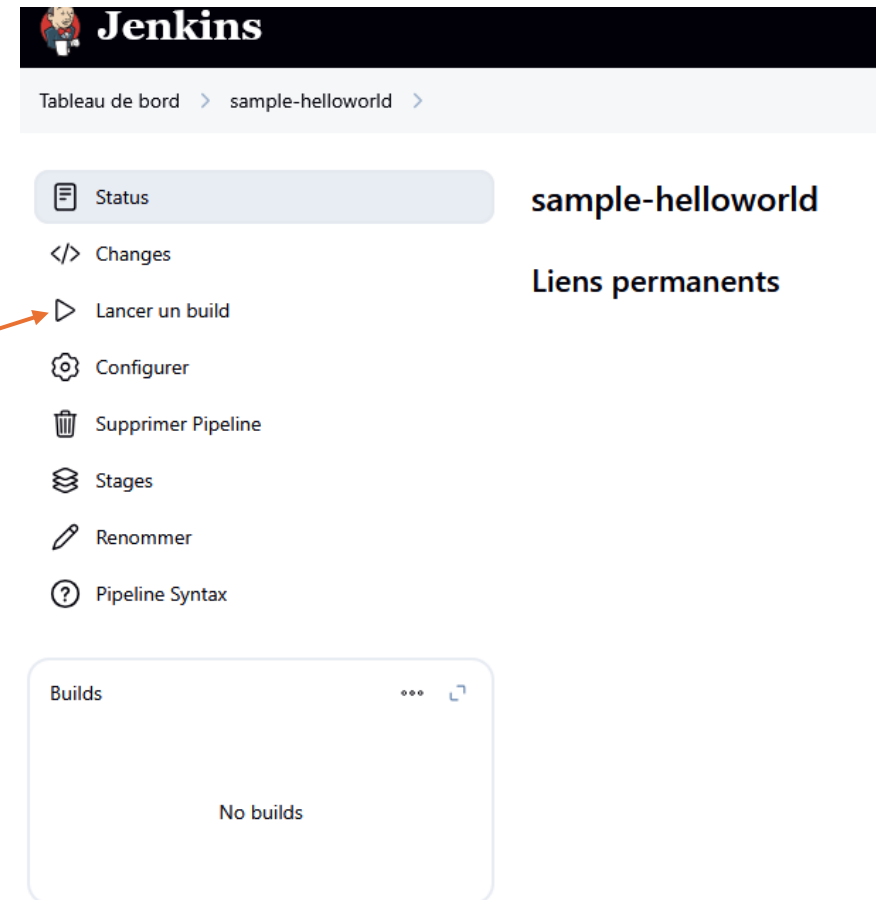
SaveApply

Pipeline

- Création à travers l'interface classique:

9 - Cliquez sur Enregistrer pour ouvrir la page de vue Projet/objet Pipeline.

10 - Sur cette page, cliquez sur Lancer un Build à gauche pour exécuter le pipeline.



Pipeline

- Création à travers l'interface classique:

10 – Une fois le build lancer, cliquer sur le dernier numéro de build

The screenshot shows the Jenkins web interface for a pipeline named 'sample-helloworld'. The breadcrumb navigation at the top reads 'Tableau de bord > sample-helloworld >'. On the left, a sidebar contains several actions: 'Status' (selected), 'Changes', 'Lancer un build', 'Configurer', 'Supprimer Pipeline', 'Stages', 'Renommer', and 'Pipeline Syntax'. The main content area is divided into two sections. The top section, titled 'sample-helloworld' with a green checkmark icon, lists 'Liens permanents' (permanent links) for various build states: 'Dernier build (#1), il y a 1 mn 23 s', 'Dernier build stable (#1), il y a 1 mn 23 s', 'Dernier build avec succès (#1), il y a 1 mn 23 s', and 'Dernier build complété (#1), il y a 1 mn 23 s'. The bottom section, titled 'Builds', features a search filter and a table of build records. Under the 'Today' header, there is one record: a green checkmark icon, the build number '#1', and the time '11:30'. A blue arrow originates from the text 'le dernier numéro de build' and points directly to the '#1' build entry in the 'Builds' table.

Tableau de bord > sample-helloworld >

Status

</> Changes

▶ Lancer un build

⚙ Configurer

🗑 Supprimer Pipeline

📁 Stages

✎ Renommer

❓ Pipeline Syntax

sample-helloworld

Liens permanents

- Dernier build (#1), il y a 1 mn 23 s
- Dernier build stable (#1), il y a 1 mn 23 s
- Dernier build avec succès (#1), il y a 1 mn 23 s
- Dernier build complété (#1), il y a 1 mn 23 s

Builds

🔍 Filter

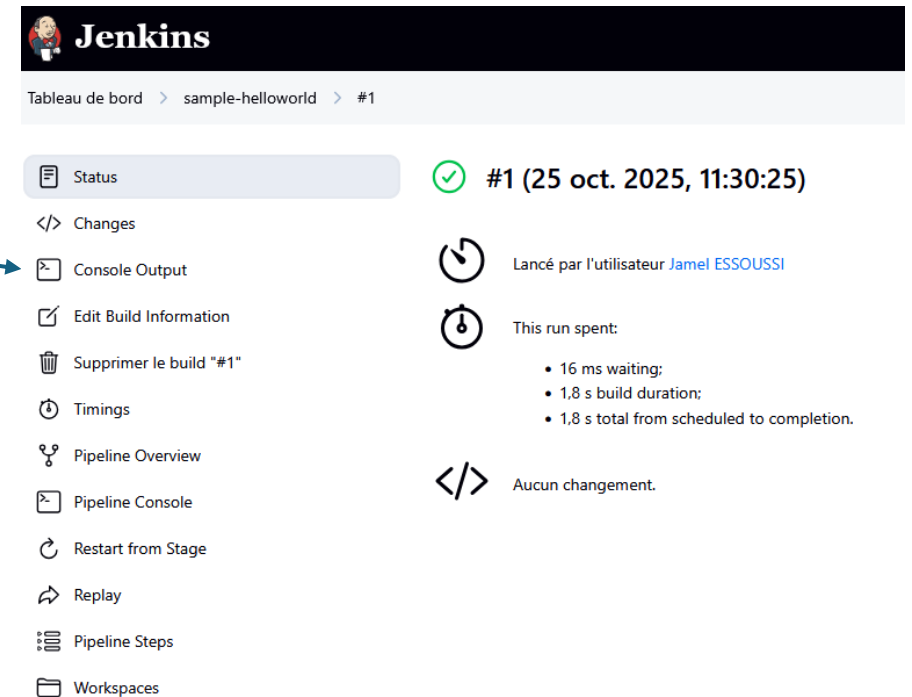
Today

- ✅ #1 11:30

Pipeline

- Création à travers l'interface classique:

11 – Ensuite sur Console output pour voir l'historique



Pipeline

- Création à travers l'interface classique:

11 – Consulter l'historique



The screenshot shows the Jenkins web interface. At the top, the Jenkins logo and name are visible. Below the header, the breadcrumb navigation shows 'Tableau de bord > sample-helloworld > #1'. On the left sidebar, the 'Console Output' option is selected. The main area displays the console output for the build. The output text is as follows:

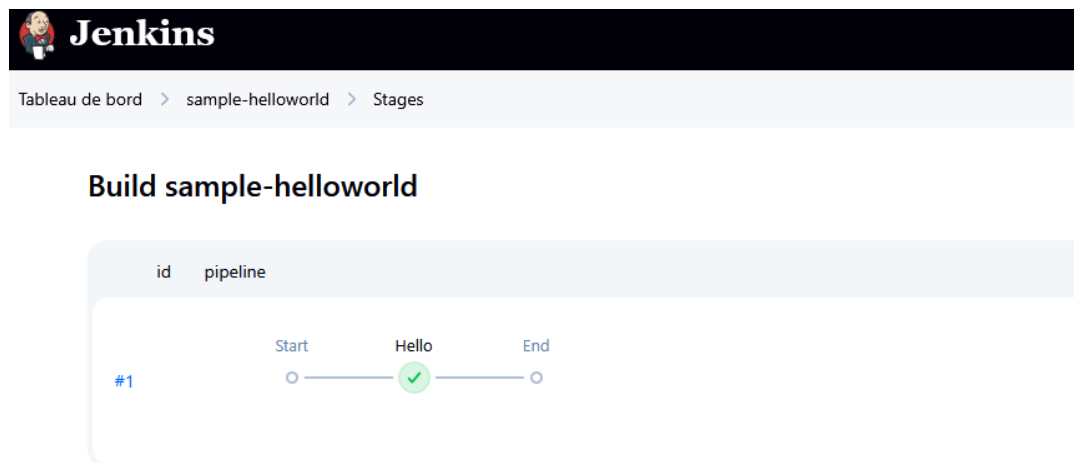
```
Démarré par l'utilisateur Jamel ESSOUSSI
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\sample-helloworld
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

The 'Hello World' line in the console output is highlighted with a red rectangular box.

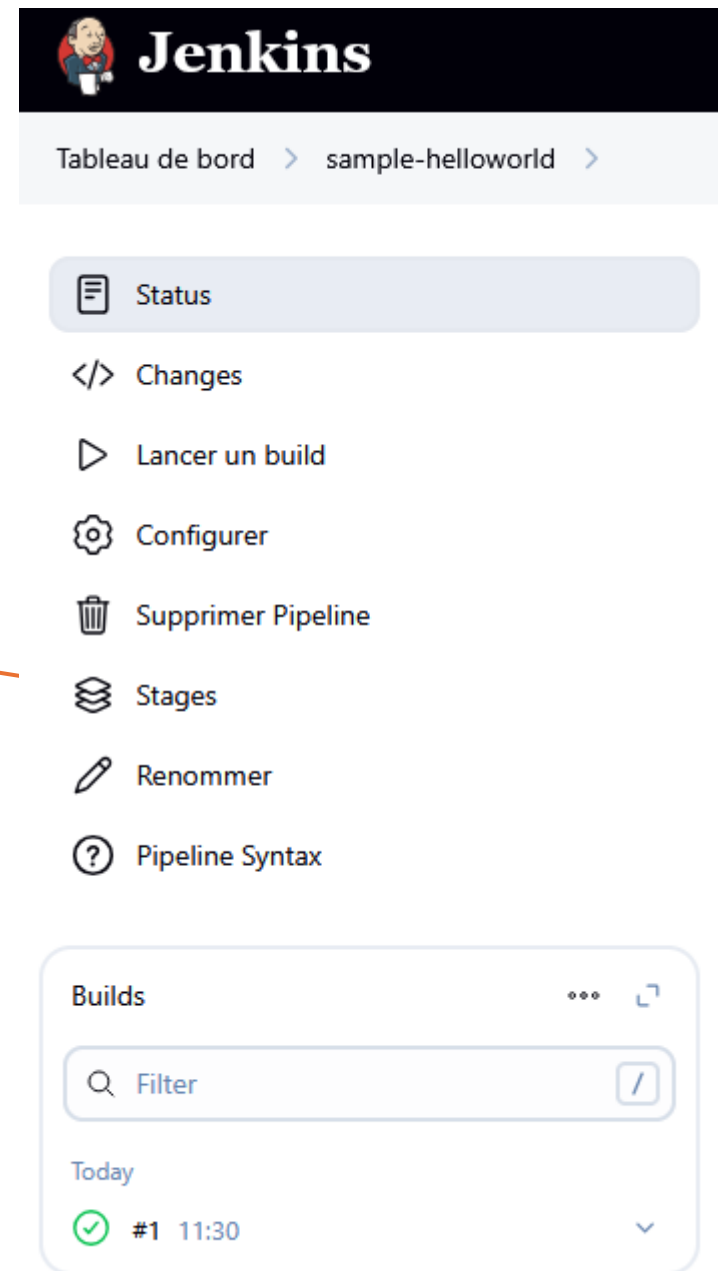
Pipeline

- Création à travers l'interface classique:

12 – Visualiser les stages



The screenshot shows the Jenkins interface for the 'sample-helloworld' pipeline. The breadcrumb navigation is 'Tableau de bord > sample-helloworld > Stages'. The main heading is 'Build sample-helloworld'. Below it, a table with columns 'id' and 'pipeline' shows a single entry with id '#1'. A visual representation of the pipeline stages is shown below the table, with nodes 'Start', 'Hello', and 'End'. The 'Hello' node is highlighted with a green checkmark, indicating a successful build.



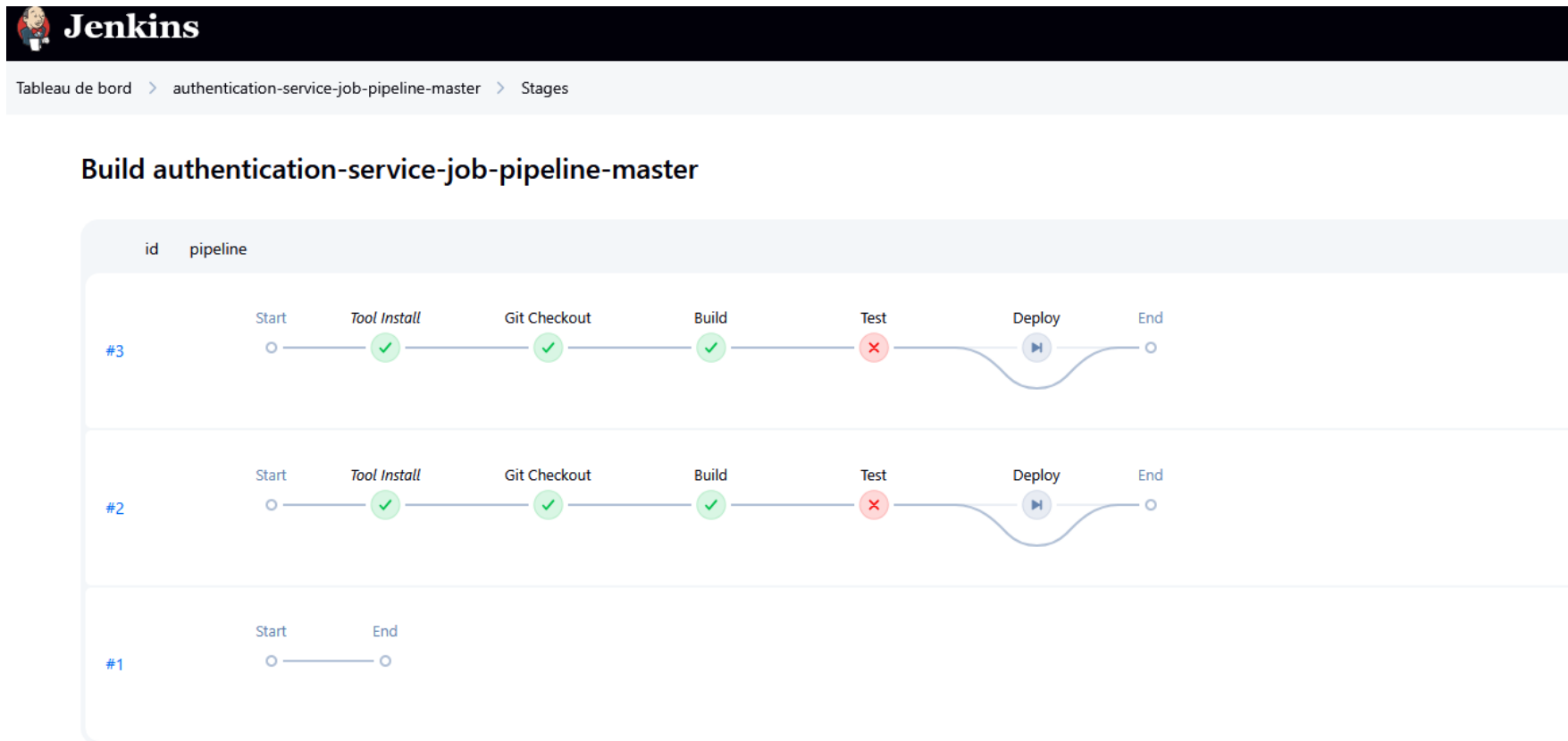
The screenshot shows the Jenkins interface for the 'sample-helloworld' pipeline. The breadcrumb navigation is 'Tableau de bord > sample-helloworld >'. The main heading is 'Build sample-helloworld'. Below it, a table with columns 'id' and 'pipeline' shows a single entry with id '#1'. A visual representation of the pipeline stages is shown below the table, with nodes 'Start', 'Hello', and 'End'. The 'Hello' node is highlighted with a green checkmark, indicating a successful build.

Builds

Filter	
Today	
✓ #1	11:30

Pipeline

- Notre exemple de TD:



Pipeline

- Notre exemple de TD:

```
1 pipeline {
2   agent any
3   tools {
4     maven 'M3'
5     jdk 'jdk17'
6   }
7   options {
8     skipStagesAfterUnstable()
9   }
10  stages {
11    stage('Git Checkout') {
12      steps {
13        script {
14          git branch: 'main', url: 'https://github.com/jessoussi/authentication-service/'
15        }
16      }
17    }
18    stage('Build') {
19      steps {
20        bat 'mvn clean compile'
21      }
22    }
23    stage('Scan SonarQube') {
24      steps {
25        withSonarQubeEnv(installationName: 'SonarQube'){
26          bat 'mvn clean install sonar:sonar'
27        }
28      }
29    }
30    stage('Test') {
31      steps {
32        bat 'mvn -Dmaven.test.failure.ignore=true clean install'
33      }
34    }
35    stage('Deploy') {
36      steps {
37        echo "Deployment application"
38      }
39    }
40  }
41 }
42 }
```


Plan du cours

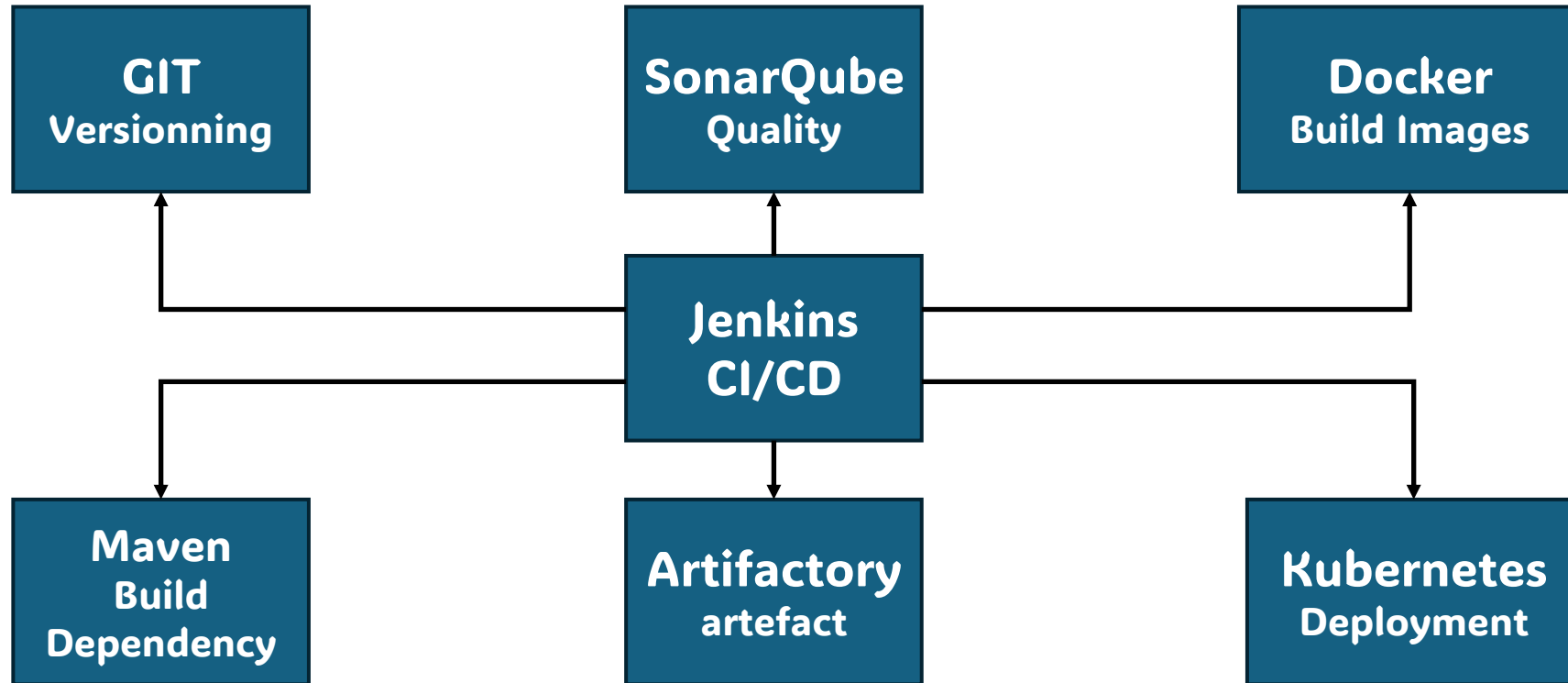
I. Introduction

II. Job & Pipeline

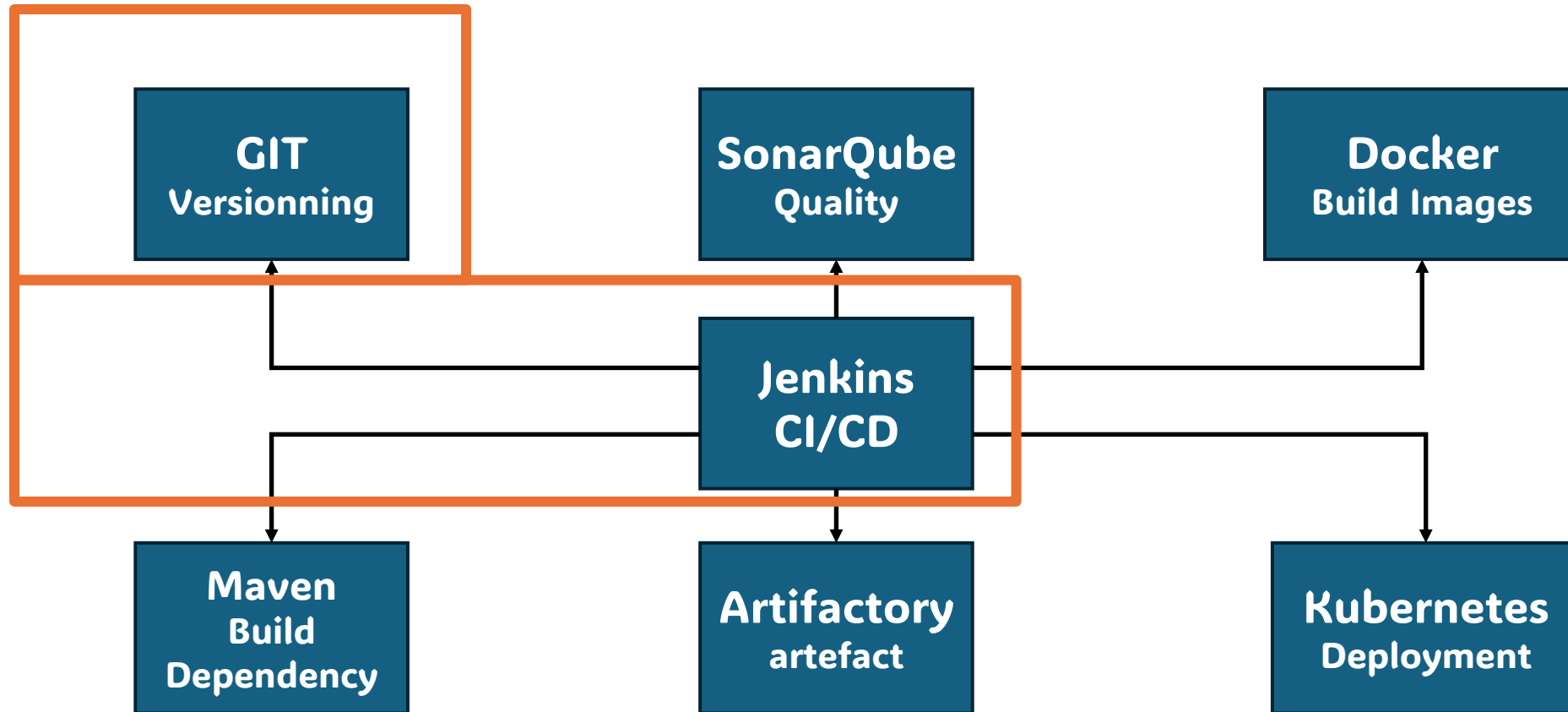
III. Intégration avec les autres outils

IV. Les bonnes pratiques

Intégration avec les autres outils



GIT



GIT

○ A travers le plugin GLT de Jenkins

The screenshot shows the Jenkins web interface. At the top, the Jenkins logo and user information (Jamel ESSOUSSI) are visible. The breadcrumb trail indicates the current location: 'Tableau de bord > Administrer Jenkins > Plugins'. The left sidebar contains navigation links: 'Mises à jour' (with a red badge showing 80 updates), 'Plugins disponibles', 'Plugins installés' (highlighted), 'Paramètres avancés', and 'Progression des téléchargements'. The main content area has a search bar with 'git' entered. Below the search bar, a table lists installed plugins. The first plugin is 'Git 5.7.0', which is active and has a description: 'This plugin integrates Git with Jenkins.' The second plugin is 'Git client 6.1.3', also active, with a description: 'Utility plugin for Git support in Jenkins'. A red warning box is displayed below the 'Git client' plugin, stating: 'Warning: The currently installed plugin version may not be safe to use. Please review the following security notices: • File system information disclosure vulnerability'. The third plugin is 'GitHub 1.43.0', which is active and has a description: 'This plugin integrates GitHub to Jenkins.'

Nom ↓	Activé
Git 5.7.0 This plugin integrates Git with Jenkins. Report an issue with this plugin	<input checked="" type="checkbox"/>
Git client 6.1.3 Utility plugin for Git support in Jenkins Report an issue with this plugin <div>Warning: The currently installed plugin version may not be safe to use. Please review the following security notices:<ul style="list-style-type: none">• File system information disclosure vulnerability</div>	<input checked="" type="checkbox"/>
GitHub 1.43.0 This plugin integrates GitHub to Jenkins. Report an issue with this plugin	<input checked="" type="checkbox"/>

GIT

- A travers le plugin GIT de Jenkins

```
stages {  
    stage('Git Checkout') {  
        steps {  
            script {  
                git branch: 'main', url: 'https://github.com/jessoussi/authentication-service'   
            }  
        }  
    }  
    stage('Build') {  
        steps {  
            script {  
                sh 'mvn clean install'   
            }  
        }  
    }  
}
```

GIT

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/jessoussi/authentication-service/

Credentials ?

jamel.essoussi@gmail.com/*****

+ Ajouter

Avancé ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/add-ci-cd

Add Branch

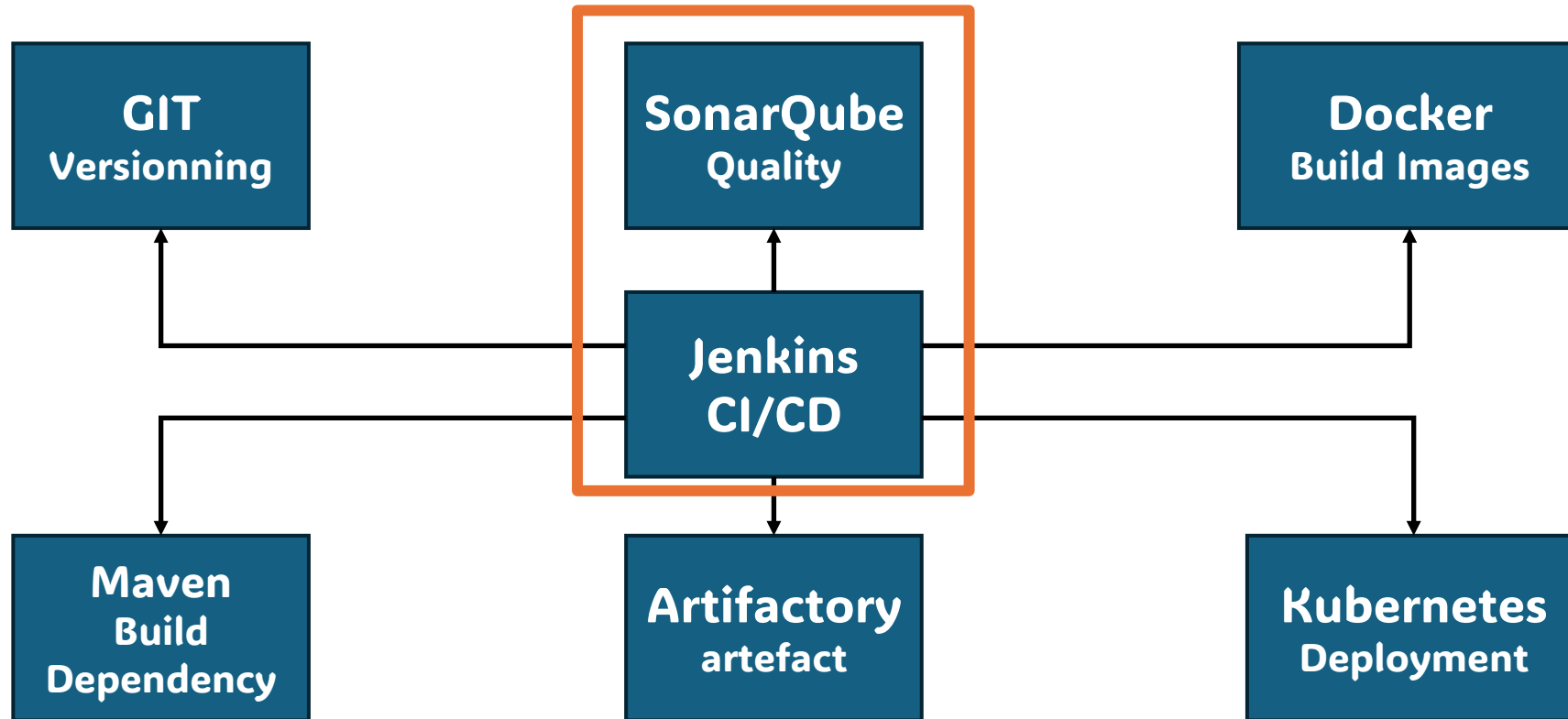
SonarQube

- SonarQube (précédemment Sonar) est un logiciel libre de qualimétrie en continu de code.
- Il aide à la détection, la classification et la résolution de défaut dans le code source, permet d'identifier les duplications de code, de mesurer le niveau de documentation et connaître la couverture de test déployée.

SonarQube

- SonarQube permet une surveillance continue de la qualité du code grâce à son interface web permettant de voir les défauts de l'ensemble du code et ceux ajoutés par la nouvelle version.
- Le logiciel peut être interfacé avec un système d'automatisation comme Jenkins pour inclure l'analyse comme une extension du développement.

SonarQube

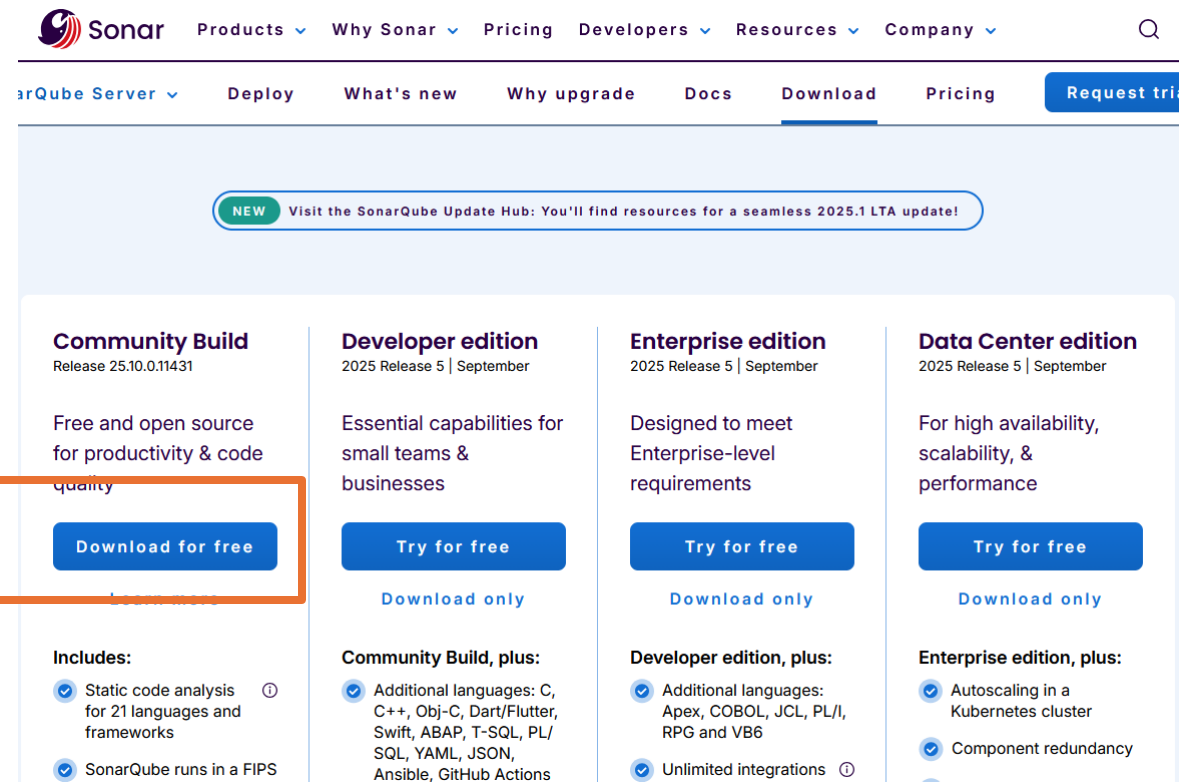


SonarQube

○ Installation de SonarQube

<https://www.sonarsource.com/products/sonarqube/downloads/>

Téléchargement



The screenshot shows the SonarQube download page. The navigation bar includes links for Products, Why Sonar, Pricing, Developers, Resources, and Company. Below this, a secondary navigation bar highlights the 'Download' section. A 'NEW' banner promotes the SonarQube Update Hub. The main content area features four columns for different editions: Community Build, Developer edition, Enterprise edition, and Data Center edition. Each column includes a 'Download for free' or 'Try for free' button. An orange box highlights the 'Download for free' button for the Community Build edition, with an arrow pointing from the 'Téléchargement' text on the left. Below the buttons, the 'Includes' section for each edition is listed.

Community Build	Developer edition	Enterprise edition	Data Center edition
Release 25.10.0.11431	2025 Release 5 September	2025 Release 5 September	2025 Release 5 September
Free and open source for productivity & code quality	Essential capabilities for small teams & businesses	Designed to meet Enterprise-level requirements	For high availability, scalability, & performance
Download for free	Try for free	Try for free	Try for free
Download only	Download only	Download only	Download only
Includes: <ul style="list-style-type: none">Static code analysis for 21 languages and frameworksSonarQube runs in a FIPS	Community Build, plus: <ul style="list-style-type: none">Additional languages: C, C++, Obj-C, Dart/Flutter, Swift, ABAP, T-SQL, PL/SQL, YAML, JSON, Ansible, GitHub Actions	Developer edition, plus: <ul style="list-style-type: none">Additional languages: Apex, COBOL, JCL, PL/I, RPG and VB6Unlimited integrations	Enterprise edition, plus: <ul style="list-style-type: none">Autoscaling in a Kubernetes clusterComponent redundancy

SonarQube

- Dézipper le package téléchargé:

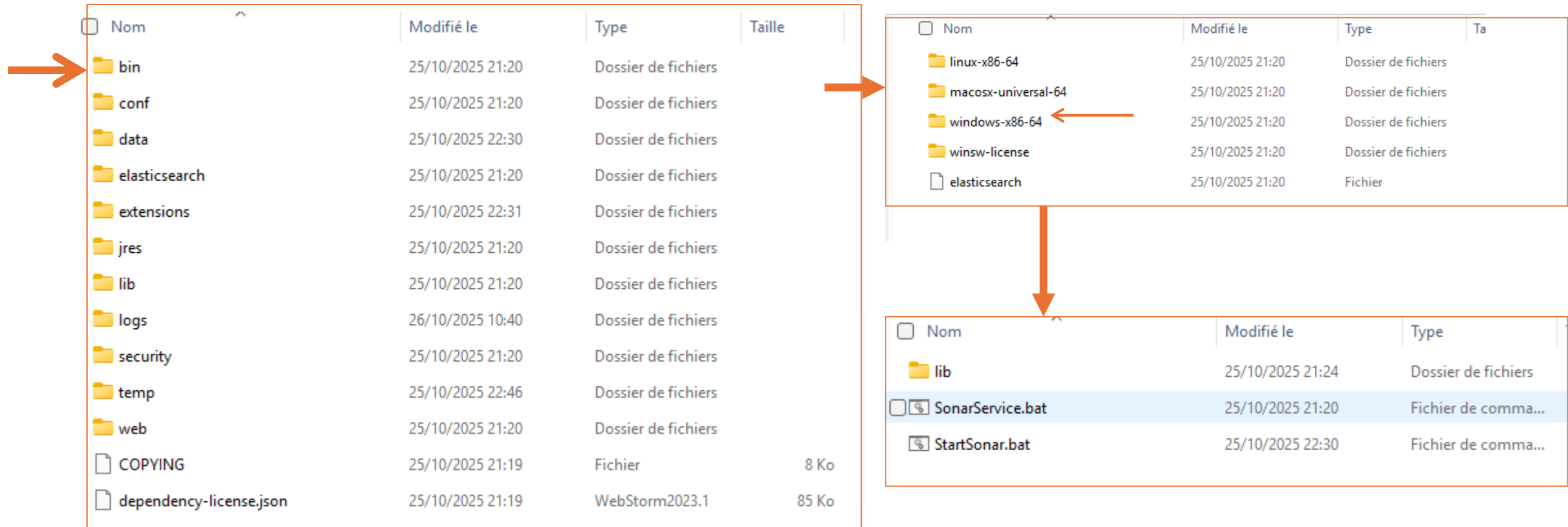
sonarqube-25.10.0.114319.zip



<input type="checkbox"/> Nom	Modifié le	Type	Taille
bin	25/10/2025 21:20	Dossier de fichiers	
conf	25/10/2025 21:20	Dossier de fichiers	
data	25/10/2025 22:30	Dossier de fichiers	
elasticsearch	25/10/2025 21:20	Dossier de fichiers	
extensions	25/10/2025 22:31	Dossier de fichiers	
jres	25/10/2025 21:20	Dossier de fichiers	
lib	25/10/2025 21:20	Dossier de fichiers	
logs	26/10/2025 10:40	Dossier de fichiers	
security	25/10/2025 21:20	Dossier de fichiers	
temp	25/10/2025 22:46	Dossier de fichiers	
web	25/10/2025 21:20	Dossier de fichiers	
COPYING	25/10/2025 21:19	Fichier	8 Ko
dependency-license.json	25/10/2025 21:19	WebStorm2023.1	85 Ko

SonarQube

○ Lancer le script de démarrage:



SonarQube

○ Lancer le script de démarrage:

```
$ ./StartSonar.bat
'jps' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.
Starting SonarQube...
2025.10.25 22:44:12 INFO app[][o.s.a.AppFileSystem] Cleaning or creating temp directory C:\dev_tools\sonarqube-25.10.0.114319\sonarqube-25.10.0.114319\temp
2025.10.25 22:44:12 INFO app[][o.s.a.es.EsSettings] Elasticsearch listening on [HTTP: 127.0.0.1:9011, TCP: 127.0.0.1:{}]
2025.10.25 22:44:12 INFO app[][o.s.a.ProcessLauncherImpl] Launch process[ELASTICSEARCH] from [C:\dev_tools\sonarqube-25.10.0.114319\sonarqube-25.10.0.114319\elasticsearch]: C:\Program Files\Java\jdk-17.0.2\bin\java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -Dcli.script=./bin/elasticsearch -Dcli.libs=lib/tools/server-cli -Des.path.home=C:\dev_tools\sonarqube-25.10.0.114319\sonarqube-25.10.0.114319\elasticsearch -Des.path.conf=C:\dev_tools\sonarqube-25.10.0.114319\sonarqube-25.10.0.114319\temp\conf\es -Des.distribution.type=tar -cp C:\dev_tools\sonarqube-25.10.0.114319\sonarqube-25.10.0.114319\elasticsearch\lib\*;C:\dev_tools\sonarqube-25.10.0.114319\sonarqube-25.10.0.114319\elasticsearch\lib\cli-launcher\* org.elasticsearch.launcher.CliToolLauncher
2025.10.25 22:44:12 INFO app[][o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
Standard Commons Logging discovery in action with spring-jcl: please remove commons-logging.jar from classpath in order to avoid potential conflicts
2025.10.25 22:44:34 INFO app[][o.s.a.SchedulerImpl] Process[es] is up
2025.10.25 22:44:34 INFO app[][o.s.a.ProcessLauncherImpl] Launch process[WEB_SERVER] from [C:\dev_tools\sonarqube-25.10.0.114319\sonarqube-25.10.0.114319]: C:\Program Files\Java\jdk-17.0.2\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\dev_tools\sonarqube-25.10.0.114319\sonarqube-25.10.0.114319\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/sun.management=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[::1] -cp ./lib/sonar-application-25.10.0.114319.jar;C:\dev_tools\sonarqube-25.10.0.114319\sonarqube-25.10.0.114319\lib\jdbc\h2\h2-2.3.232.jar org.sonar.server.app.WebServer C:\dev_tools\sonarqube-25.10.0.114319\sonarqube-25.10.0.114319\temp\sq-process12833236382398619013properties
Standard Commons Logging discovery in action with spring-jcl: please remove commons-logging.jar from classpath in order to avoid potential conflicts
2025.10.25 22:44:56 INFO app[][o.s.a.SchedulerImpl] Process[web] is up
```

SonarQube

- Interface SonarQube : <http://localhost:9000/>

The screenshot displays the SonarQube web interface in a browser window. The address bar shows `http://localhost:9000/projects`. A yellow banner at the top states: "Embedded database should be used for evaluation purposes only. It doesn't support scaling, upgrading to a new SonarQube Server version, or migration to another database engine. [Learn more](#)". Below this, a blue banner mentions a change in security, reliability, and maintainability counts and ratings, with a link to [Learn more in SonarQube documentation](#).

The main navigation bar includes the SonarQube logo and the following tabs: Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. The 'Projects' tab is active.

On the left sidebar, there are sections for 'My Favorites' (with 'All' selected), 'Filters', 'Quality Gate' (showing 1 Passed and 0 Failed), 'Security' (with a table of issue counts), and 'Reliability' (also with a table of issue counts).

The main content area shows a search bar for projects (minimum 2 characters), a 'Perspective' dropdown set to 'Overall Status', and a 'Sort by' dropdown set to 'Name'. It indicates '1 project(s)' are found.

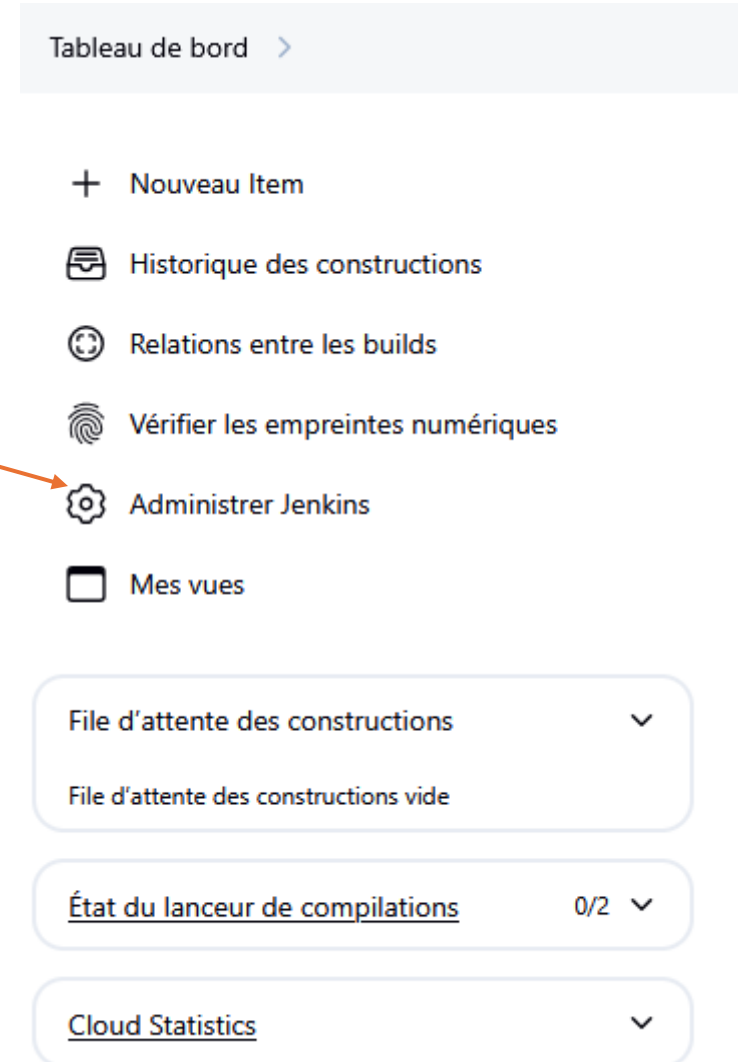
The project 'authentication-service' is displayed with a 'Public' label and a green checkmark indicating it 'Passed'. Below this, it states 'Last analysis: 13 hours ago • 260 Lines of Code • XML, Java'. A row of metrics is shown: Security (A 0), Reliability (A 0), Maintainability (A 7), Hotspots Reviewed (A —), Coverage (0.0%), and Duplications (0.0%).

At the bottom of the metrics row, it says '1 of 1 shown'.

SonarQube

○ Branchement avec Jenkins

Aller sur la page d'administration de Jenkins



SonarQube

○ Branchement avec Jenkins

Aller sur la page système



Configuration du système



System

Configurer les paramètres généraux et les chemins de fichiers.



Tools

Configurer les outils, leur localisation et les installeurs automatiques.



Docker

Plugin for launching build Agents as Docker containers



Clouds

Ajouter, supprimer et configurer les instances de cloud afin de provisionner les agents à la demande.



SonarQube

○ Branchement avec Jenkins

Credential: Token Sonar

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

☐ Environment variables

Installations de SonarQube

Liste des installations de SonarQube

Nom

SonarQube

URL du serveur

Par défaut à <http://localhost:9000>

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

sonarqubecred

+ Ajouter

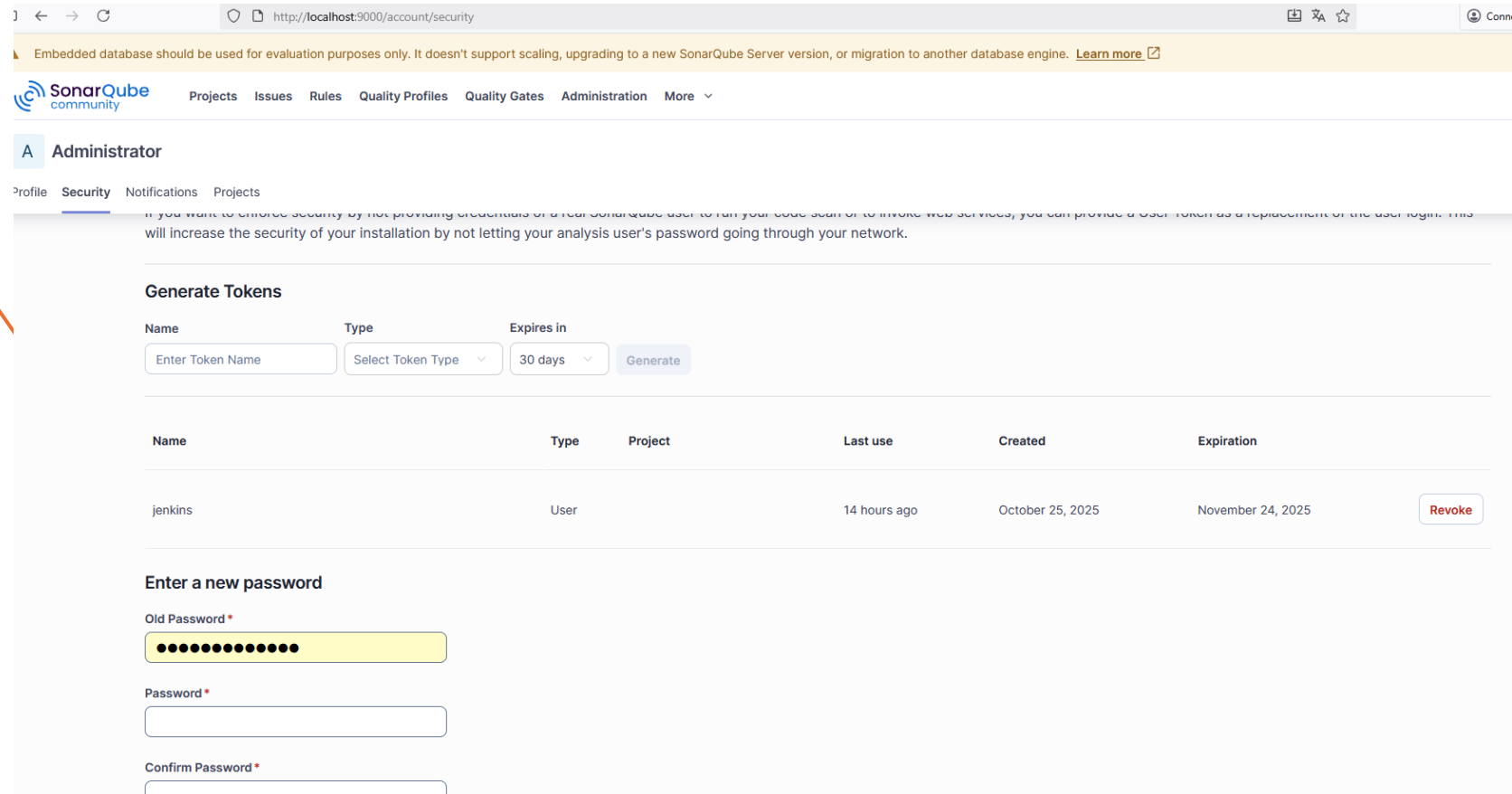
Avancé ▾

Ajouter une installation SonarQube

SonarQube

○ Générer un token sur SonrQube

Credential: Token Sonar



http://localhost:9000/account/security

Embedded database should be used for evaluation purposes only. It doesn't support scaling, upgrading to a new SonarQube Server version, or migration to another database engine. [Learn more](#)

SonarQube community

Projects Issues Rules Quality Profiles Quality Gates Administration More

A Administrator

Profile Security Notifications Projects

If you want to enhance security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a user token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Name	Type	Expires in	
<input type="text" value="Enter Token Name"/>	<input type="text" value="Select Token Type"/>	<input type="text" value="30 days"/>	<input type="button" value="Generate"/>

Name	Type	Project	Last use	Created	Expiration	
jenkins	User		14 hours ago	October 25, 2025	November 24, 2025	<input type="button" value="Revoke"/>

Enter a new password

Old Password *

••••••••••

Password *

Confirm Password *

SonarQube

○ Renseigner le token dans Jenkins

Si vous ne pouvez pas accéder à Jenkins, vous pouvez utiliser le lien ci-dessous pour accéder à votre compte.

☐ Environnement variables

Installations de SonarQube

Liste des installations de SonarQube

Nom
SonarQube

URL du serveur

Par défaut à <http://localhost:9000>


Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

sonarqubecred
+ Ajouter

Avancé ▾

Ajouter une installation SonarQube



SonarQube

Jenkins Credentials Provider: Jenkins

Type

Nom d'utilisateur et mot de passe ▾

Portée ?

Global (Jenkins, agents, items, etc...) ▾

Nom d'utilisateur ?

☐ Treat username as secret ?

Mot de passe ?

ID ?

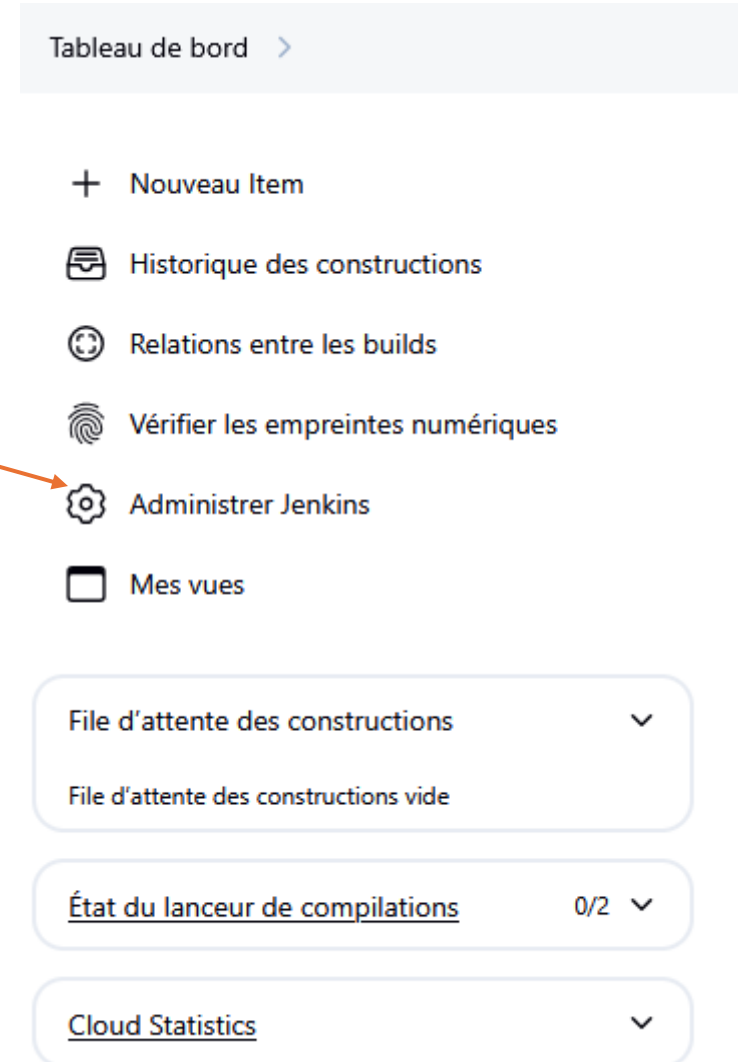
Description ?

Annuler Ajouter

SonarQube

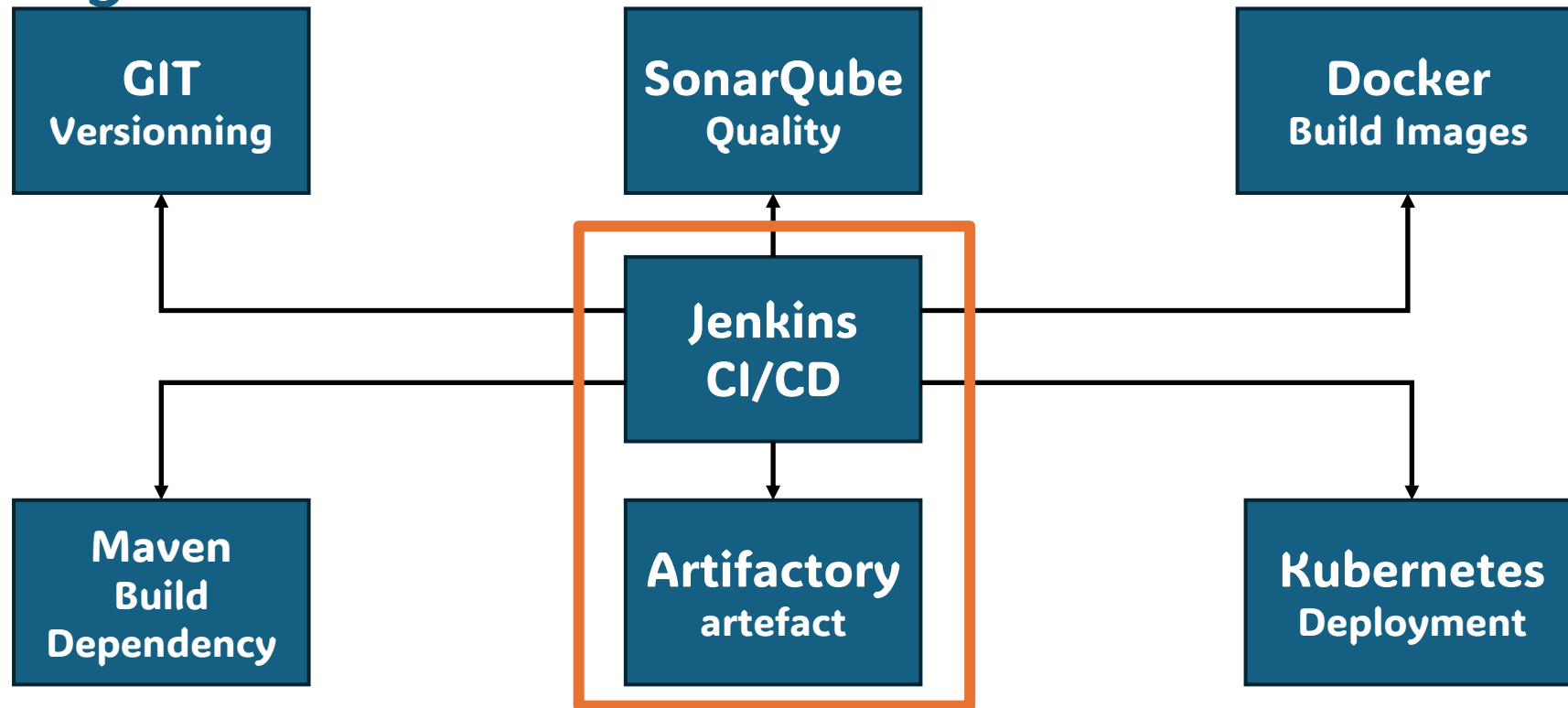
○ Branchement avec Jenkins

Aller sur la page d'administration de Jenkins



Artifactory

- Dépôt Centrale dans l'organisation
- Partage des livrables



Artifactory

- JFrog Artifactory est une solution pour héberger et gérer tous les artefacts logiciels, les modèles d'IA/ML, les binaires, les packages, les fichiers, les conteneurs, les composants et les versions utilisés et générés dans la chaîne d'approvisionnement logicielle de l'organisation.





Artifactory

- Artifactory sert de plaque tournante centrale pour le DevOps et les développeurs, en s'intégrant aux outils et processus pour améliorer l'automatisation, capturer les attestations, assurer l'intégrité des livraisons et fournir une visibilité inégalée sur vos processus de développement.

Artifactory Jenkins Plugin

Tableau de bord > Administrer Jenkins > Plugins

Plugins

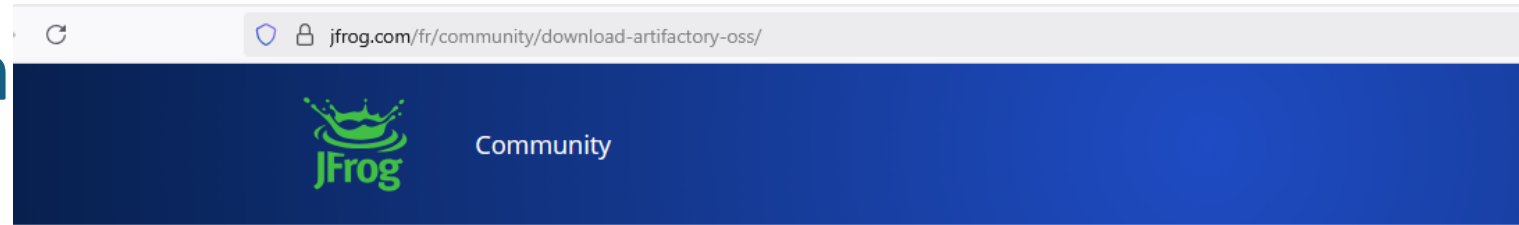
-  Mises à jour 80
-  Plugins disponibles
-  Plugins installés
-  Paramètres avancés

🔍 jfrog

Installer	Nom ↓	Publié
<input checked="" type="checkbox"/>	<div>JFrog 1.5.10</div> <div>Développement .NET Maven npm Déploiement docker</div> <p>The Jenkins JFrog Plugin allows for easy integration between Jenkins and the JFrog Platform. This integration allows your build jobs to deploy artifacts and resolve dependencies to and from Artifactory, and then have them linked to the build job that created them. It also allows you to scan your artifacts and builds with JFrog Xray and distribute your software package to remote locations using JFrog Distribution. This is all achieved by the plugin by wrapping JFrog CLI. Any JFrog CLI command can be executed from within your Jenkins Pipeline job using the JFrog Plugin.</p>	Il y a 1 mo. 26 j
<input type="checkbox"/>	<div>Artifactory Client API 2.19.0-116.vc6e7efff0e6b_</div> <p>This plugin provides the Artifactory Client (v2.19.0) for other plugins.</p>	Il y a 3 mo. 2 j
<input type="checkbox"/>	<div>Artifact Manager Artifactory 252.v944a_69ea_4873</div> <p>A Jenkins plugin to keep artifacts and Pipeline stashes in JFrog Artifactory</p>	Il y a 3 mo. 0 j
<input type="checkbox"/>	<div>Jobcacher Artifactory Storage Extension 198.vd6fa_ec5e1028</div> <p>Extension of jobcacher plugin that add JFrog artifactory for item storage.</p>	Il y a 3 mo. 2 j

Artifactory

○ Installation

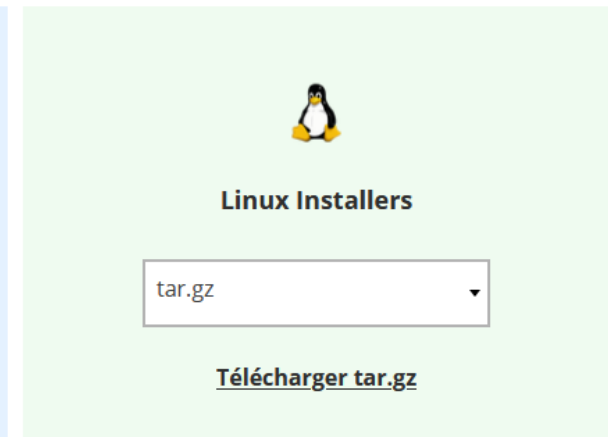
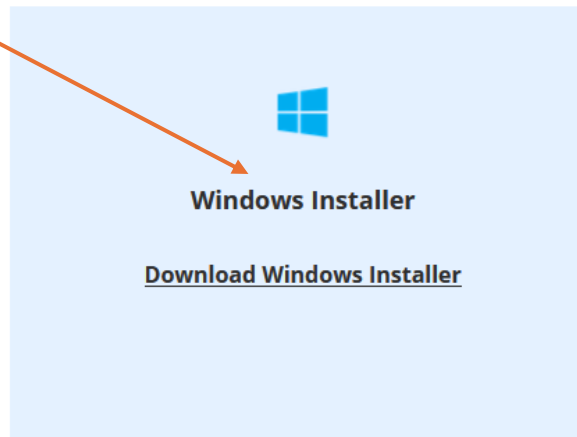


JFROG ARTIFACTORY OPEN SOURCE FOR ARTIFACT LIFE-CYCLE MANAGEMENT

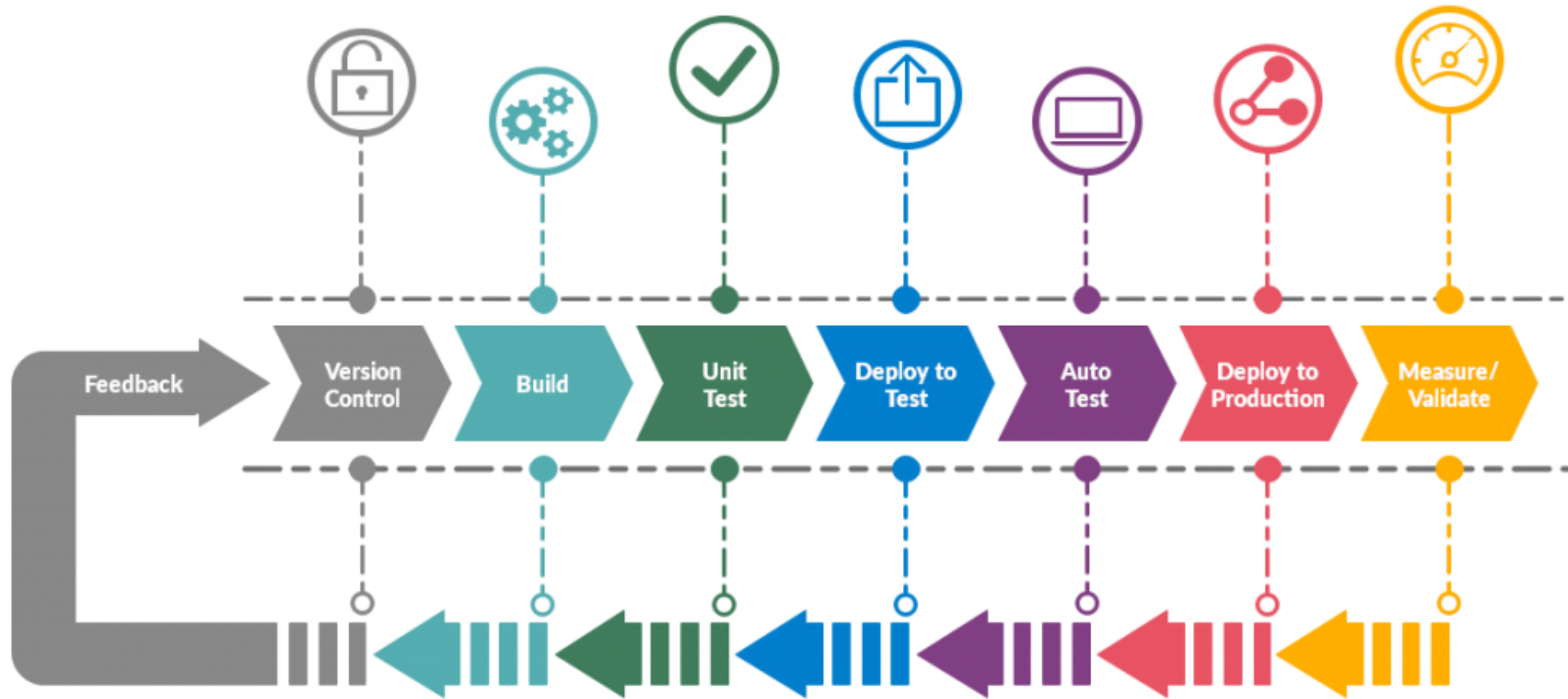
JFrog's Artifactory open source project was created to speed up development cycles using binary repositories. It's the world's most advanced repository manager, creating a single place for teams to manage all their binary artifacts efficiently.

[Get code source >](#)

Téléchargement



Introduction



Plan du cours

I. Introduction

II. Job & Pipeline

III. Intégration avec les autres outils

IV. Les bonnes pratiques

Les bonnes pratiques

- **Créer un Job Jenkins par Branche**

Merci