



Introduction à la CI/CD/CD et Jenkins



Dans ce module :

- Qu'est-ce que la CI/CD/CD ?
- Jenkins et son rôle dans la mise en place d'une pipeline de CI/CD
- Installation de Jenkins sous Docker
- Job freestyle VS pipeline
- Création d'un premier job

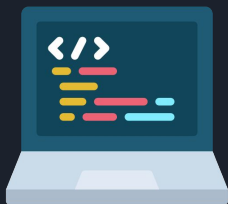
Qu'est-ce que la CI/CD/CD ?



CI (Continuous integration | Intégration continue)

“C'est le processus dans lequel les développeurs intègrent leur code dans un référentiel partagé plusieurs fois par jour.

À chaque intégration, des tests automatiques sont exécutés pour vérifier si le code nouvellement intégré ne provoque pas de régressions ou d'erreurs dans l'application.”





CD (Continuous delivery | Livraison continue)

“Il s'agit de la pratique qui consiste à automatiser le processus de mise à disposition du logiciel en production, à un rythme rapide et fiable.

Cela signifie que chaque modification de code qui passe les tests de CI est potentiellement déployable en production.”



CD (Continuous deployment | Déploiement continu)

“Cette pratique va un pas plus loin que la Continuous Delivery en automatisant complètement le déploiement du logiciel en production à chaque fois qu'un changement de code passe avec succès les tests de CI.

Ainsi, chaque fois qu'un développeur fusionne son code dans la branche principale (ou toute autre branche configurée), il est automatiquement déployé en production sans intervention humaine supplémentaire.”

TODO :
Retravailler le
texte ici



Jenkins et la CI/CD



Jenkins

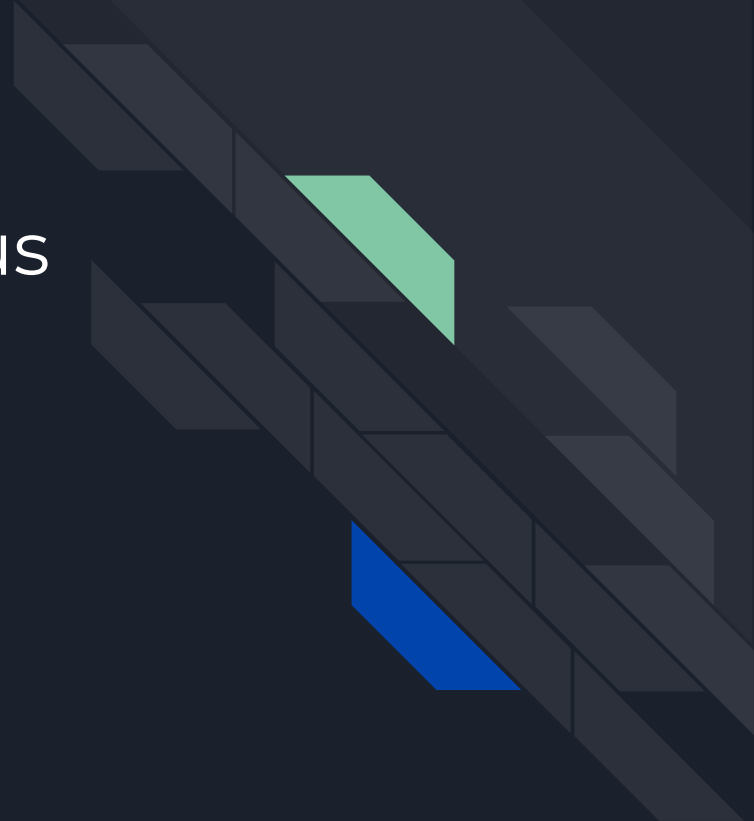
“Jenkins est un outil d'intégration continue (CI) open-source largement utilisé dans le domaine du développement logiciel. Son rôle principal est d'automatiser les processus de construction, de test et de déploiement des applications, ce qui permet aux équipes de développement de livrer des logiciels de manière plus rapide, fiable et efficace.”



Jenkins et la CI/CD



Installation de Jenkins sous Docker



Installation sous Docker



```
mkdir jenkins
```

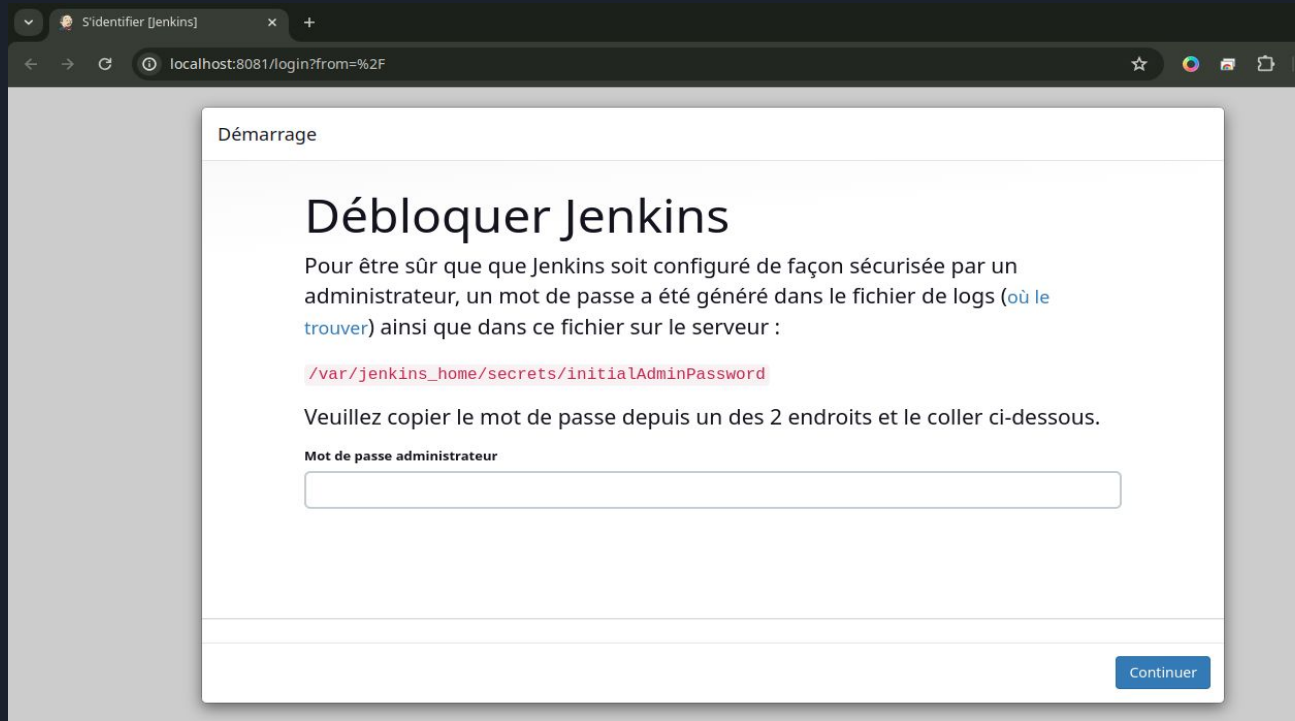
```
cd jenkins
```

```
docker network create jenkins
```

```
docker run -d --restart on-failure --network jenkins -p 8080:8080 -p 50000:50000 --name jenkins_container  
jenkins/jenkins:its
```

```
docker inspect jenkins_container
```

Configuration de Jenkins



S'identifier [Jenkins]

localhost:8081/login?from=%2F

Démarrage

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 :

`/var/jenkins_home/secrets/initialAdminPassword`

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

Mot de passe administrateur

Continuer

http://localhost:8080

docker logs jenkins_container

Job freestyle vs pipeline





Job freestyle

- Les jobs freestyle sont basés sur une interface utilisateur graphique (GUI) conviviale qui permet aux utilisateurs de configurer leurs tâches CI/CD en utilisant des options prédéfinies.
- Ils sont bien adaptés aux workflows simples et linéaires où les tâches sont exécutées dans un ordre spécifique sans besoin de logique de programmation complexe.
- Ils offrent une flexibilité limitée en termes de gestion du code source, de gestion des dépendances et de gestion des erreurs.



Job pipeline

- Les pipelines sont définis comme du code, généralement en utilisant le langage de script Jenkins appelé Groovy. Cette approche permet une plus grande flexibilité et un meilleur contrôle sur le processus CI/CD.
- Ils sont conçus pour modéliser des workflows complexes avec des dépendances entre les tâches, des boucles, des conditions et des logiques avancées.
- Les pipelines offrent une meilleure visibilité et traçabilité du processus d'intégration et de déploiement, car tout le processus est décrit en tant que code.
- Ils sont plus appropriés pour les projets complexes, les environnements distribués et les équipes de développement nécessitant un haut degré d'automatisation et de reproductibilité.

Création d'un premier job



Mon premier Job freestyle

localhost:8081/job/My-first-job/configure

Tableau de bord > My-first-job > Configuration

Configure

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

☐ Aucune

☒ Git ?

Repositories ?

Repository URL ?

Please enter Git repository.

Credentials ?

- aucun -

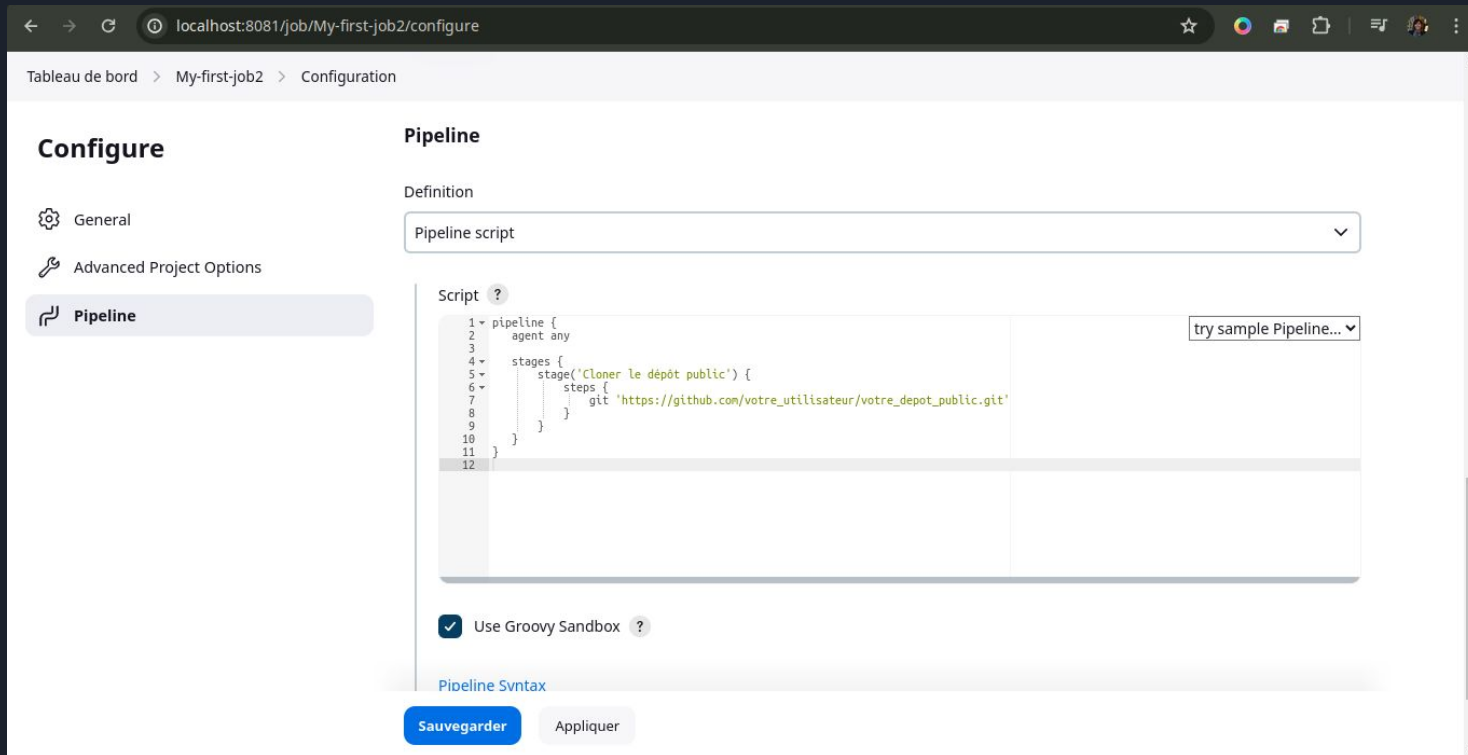
+ Ajouter

Avancé

Add Repository

Sauvegarder Appliquer

Mon premier Job pipeline



The screenshot shows the Jenkins web interface in a browser. The address bar displays `localhost:8081/job/My-first-job2/configure`. The breadcrumb navigation at the top reads "Tableau de bord > My-first-job2 > Configuration".

On the left, the "Configure" section has three tabs: "General" (with a gear icon), "Advanced Project Options" (with a wrench icon), and "Pipeline" (with a pipeline icon and highlighted in light blue).

The main area is titled "Pipeline" and contains a "Definition" dropdown menu set to "Pipeline script". Below this is a "Script" section with a line number editor (1-12) and a "try sample Pipeline..." button. The script content is as follows:

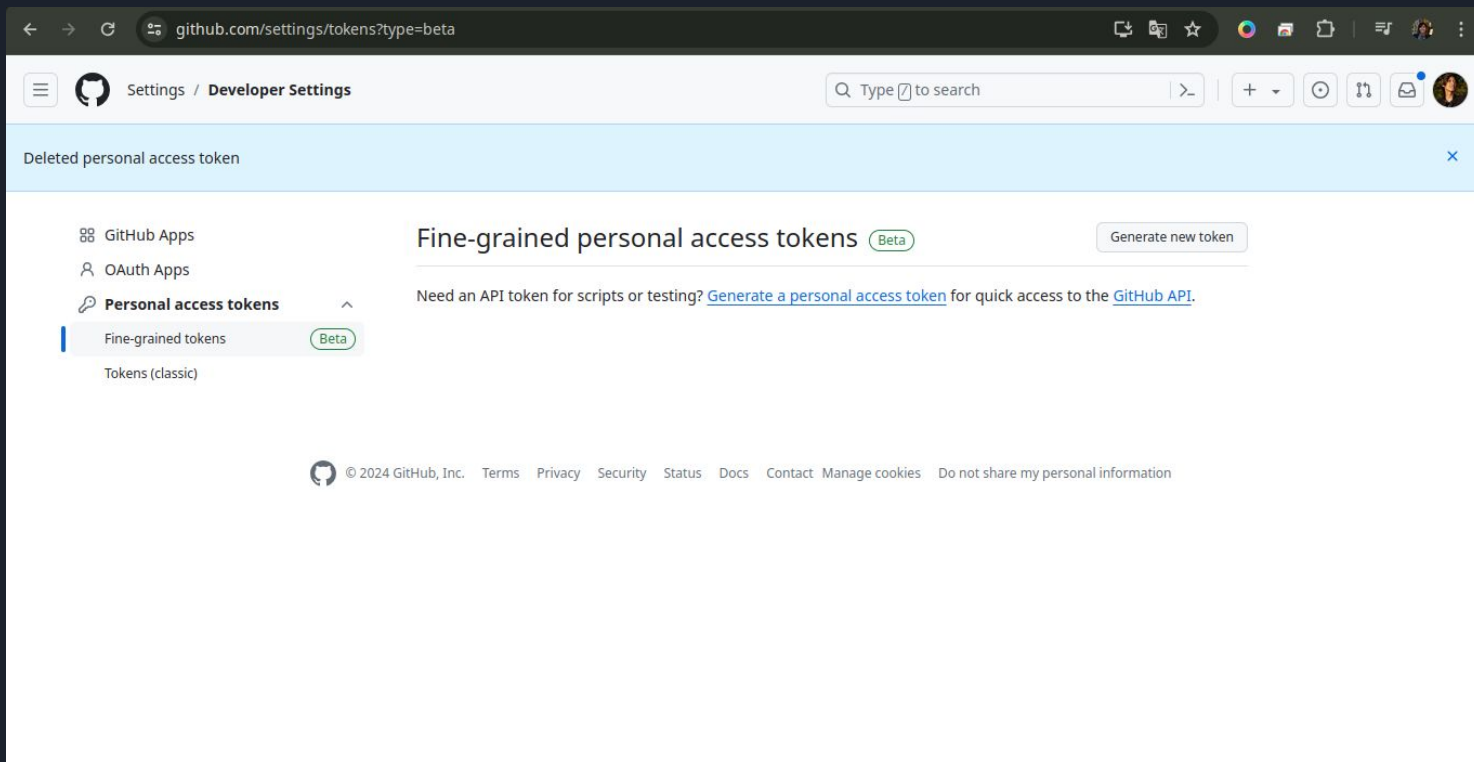
```
1 pipeline {  
2   agent any  
3  
4   stages {  
5     stage('Cloner le dépôt public') {  
6       steps {  
7         git 'https://github.com/votre_utilisateur/votre_depot_public.git'  
8       }  
9     }  
10  }  
11 }  
12
```

Below the script editor, there is a checkbox labeled "Use Groovy Sandbox" which is checked. At the bottom, there are two buttons: "Sauvegarder" (Save) and "Appliquer" (Apply).

Cloner un repo privé



PAT



The screenshot shows the GitHub Developer Settings page for Personal Access Tokens. The browser address bar displays `github.com/settings/tokens?type=beta`. The page header includes the GitHub logo, the text "Settings / Developer Settings", a search bar, and various utility icons. A light blue banner at the top states "Deleted personal access token" with a close button. The left sidebar contains a menu with "GitHub Apps", "OAuth Apps", "Personal access tokens" (selected), "Fine-grained tokens" (marked as Beta), and "Tokens (classic)". The main content area is titled "Fine-grained personal access tokens" (also marked as Beta) and includes a "Generate new token" button. Below the title, a message reads: "Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the [GitHub API](#)." The footer contains the GitHub logo, copyright notice "© 2024 GitHub, Inc.", and links for Terms, Privacy, Security, Status, Docs, Contact, Manage cookies, and Do not share my personal information.

github.com/settings/tokens?type=beta

Settings / Developer Settings

Deleted personal access token

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens Beta

Tokens (classic)

Fine-grained personal access tokens Beta Generate new token

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the [GitHub API](#).

© 2024 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

Mon premier Job pipeline

localhost:8081/job/My-first-job2/pipeline-syntax/

Tableau de bord > My-first-job2 > Pipeline Syntax

Online Documentation
Examples Reference
IntelliJ IDEA GD

Jenkins Credentials Provider: Jenkins

Ajouter des identifiants

Domain

Identifiants globaux (illimité)

Type

Nom d'utilisateur et mot de passe

Portée ?

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

Nom d'utilisateur ?

☐ Treat username as secret ?

Generate Pipeline Script

Mon premier Job pipeline

← → ↻ localhost:8081/job/My-first-job2/configure ☆ 🌈 📁 🗑️ 📄 👤 ⋮

Tableau de bord > My-first-job2 > Configuration

Configure

- ⚙️ General
- 🔧 Advanced Project Options
- 📄 Pipeline**

Pipeline

Definition

Pipeline script ▼

Script ?

```
1 ~ pipeline {
2   agent any
3
4   environment {
5     GITHUB_TOKEN = credentials('nom_de_votre_token')
6   }
7
8   stages {
9     stage('Cloner le dépôt privé') {
10      steps {
11        git credentialsId: 'GITHUB_TOKEN', url: 'https://github.com/votre_utilisateur/votre_depot_prive.git'
12      }
13    }
14  }
15 }
16
17
```

try sample Pipeline... ▼

☒ Use Groovy Sandbox ?

Pipeline Syntax

Sauvegarder Appliquer

Activité :
Première pipeline avec
repo public et privé





Pipeline vide

```
pipeline {  
  agent any  
  
  stages {  
    stage('My stage') {  
      steps {  
        //My step  
      }  
    }  
  }  
}
```




A vous !



Why ?

C'est le premier point de départ

Objectif ?

Générer un projet html avec bootstrap

Cloner un repo public et privé depuis une pipeline jenkins

Outils ?

- Ossature d'une pipeline
- Ligne de commande pour récupérer un repo public ou privé
- PAT Github
- Variable d'environnement jenkins