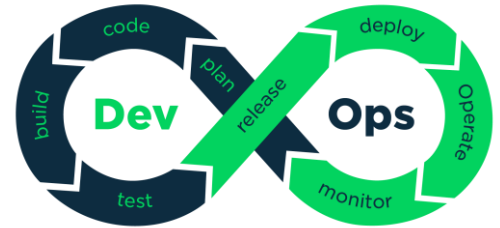
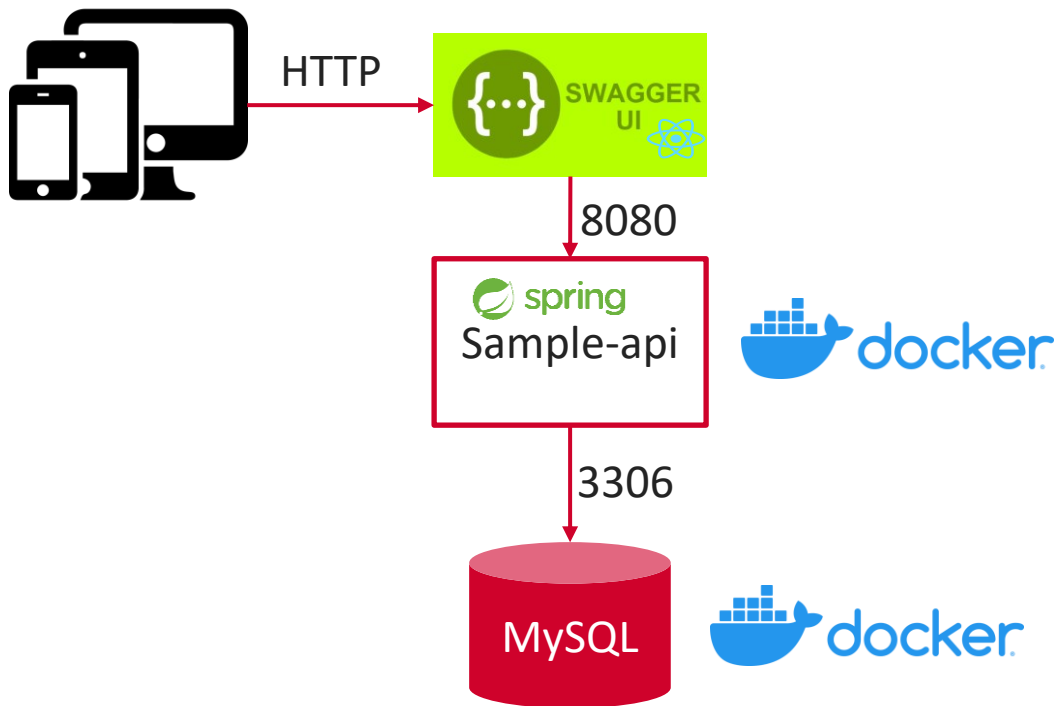


# FORMATION JENKINS – TP DOCKER

La démarche pour délivrer en continu

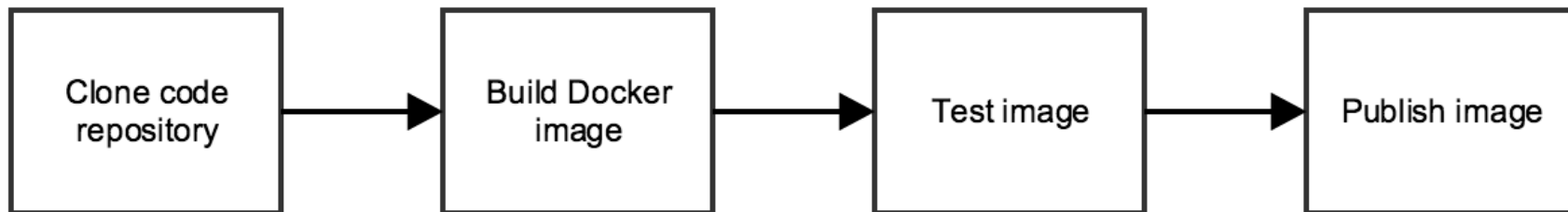


# PRESENTATION DE L'ARCHITECTURE DU PROJET



## TP9: JENKINS - ILLUSTRATION DES ÉTAPES A IMPLÉMENTER SUR JENKINS

---



## TP9: JENKINS - 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**

## TP9: JENKINS – FORK DU PROJET

---

- Se connecter à GITHUB
- Se rendre dans le repository GIT suivant:
  - <https://github.com/bnasslahsen/twitter-api.git>
- Créer un fork du repository suivant:

 bnasslahsen / twitter-api

 Unwatch ▾

1

 Star

0

 Fork

0

## TP9: JENKINS – ADAPTATION DU PROJET

---

- Editer le fichier pom.xml dans GITHUB
- Remplacer dans la ligne qui valorise votre login et nom de repository récemment créés sous dockerhub.
- Voici un exemple:

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

## TP9: JENKINS - MODIFICATION DE LA CONF SETTINGS.XML

---

- Configurer le fichier settings.xml de maven en ajoutant votre login/mdp au serveur DOCKERHUB:
  - vim /opt/apache-maven-3.5.4/conf/settings.xml

```
<servers>
```

```
  <server>
```

```
    <id>registry.hub.docker.com</id>
```

```
    <username>your-dockerhub-login</username>
```

```
    <password>your-dockerhub-pass</password>
```

```
  </server>
```

```
  <server>
```

```
    <id>registry-1.docker.io</id>
```

```
    <username>your-dockerhub-login</username>
```

```
    <password>your-dockerhub-pass</password>
```

```
  </server>
```

# TP9: INSTALLATION DU PIPELINE MAVEN PLUGIN

- 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

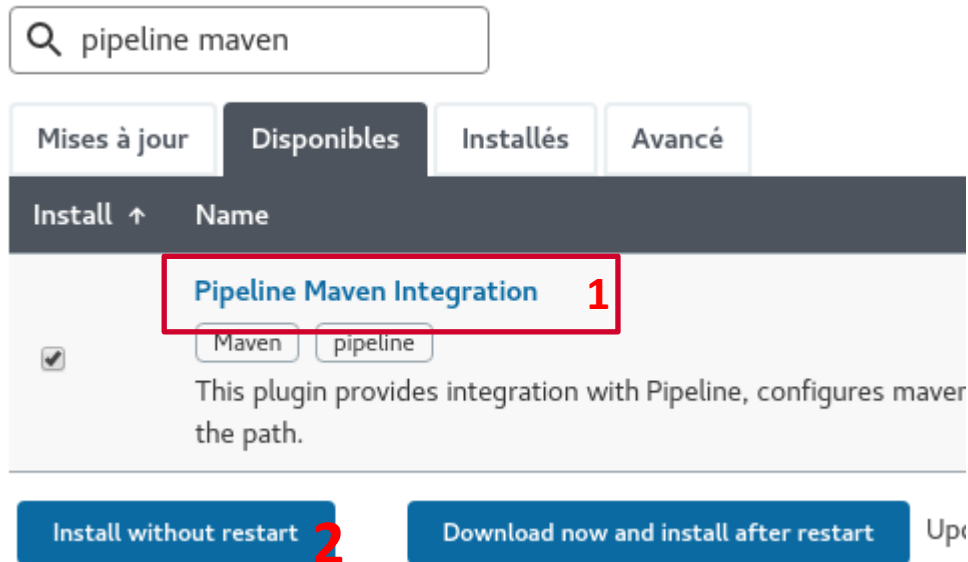
## Administrer Jenkins

- 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.



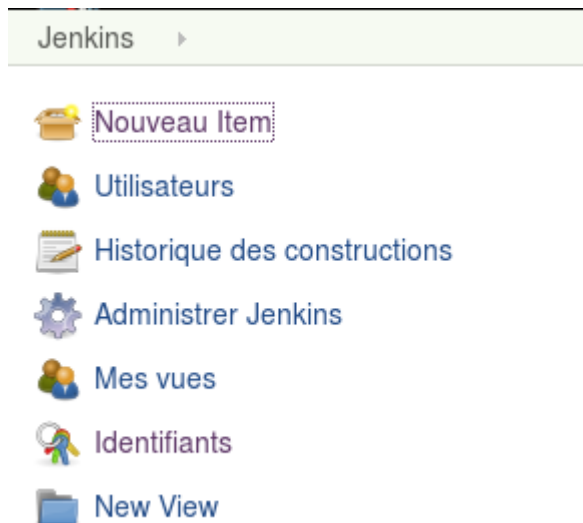
## TP9: CONVERTIR LE JOB FREE-STYLE EN PIPELINE

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



## TP9: JENKINS - CRÉATION DE JOB SOUS JENKINS


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





## TP9: JENKINS - CRÉATION DE JOB SOUS JENKINS


- Choisir le projet de type Pipeline
- Renseigner un nom de projet:
  - Twitter-api-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**  
Creates 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 namespace, so you can have multiple things of the same name as long as they are in different folders.

## TP9: JENKINS - CRÉATION DE JOB SOUS JENKINS

- Pour la définition de, choisir Pipeline script from SCM
- Renseigner le l'url git de votre repository
- Dans Credentials, choisir votre login/mdp
- Cliquer sur Save

Repositories

Repository URL

https://github.com/bnasslahsen/twitter-api.git

Credentials

m.ouledsaber@gmail.Com/\*\*\*\*\* (git-login) Ajouter

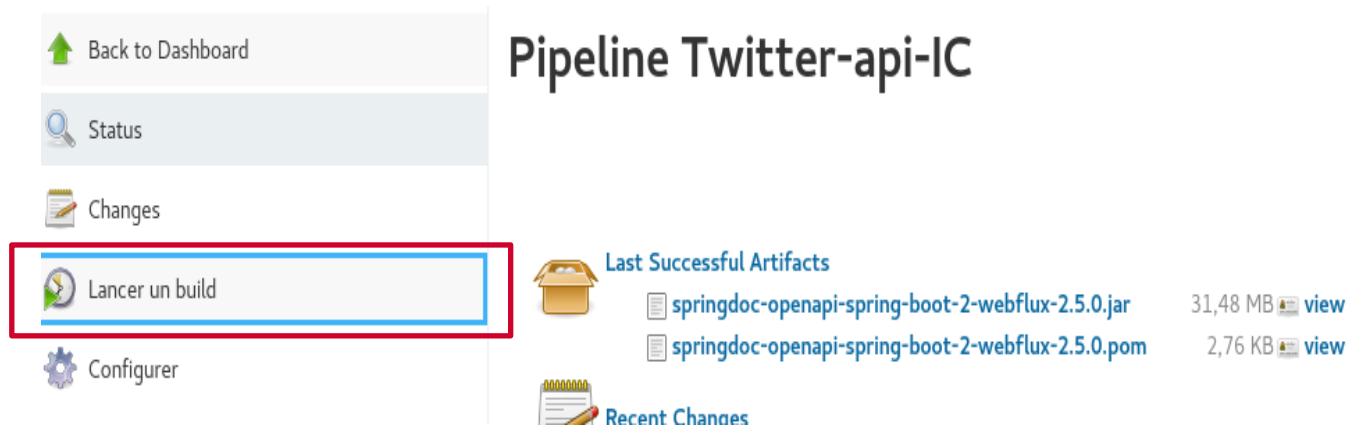
Branches to build

Branch Specifier (blank for 'any')

\*/main

## TP9: JENKINS - CRÉATION DE JOB SOUS JENKINS

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



Back to Dashboard

Status

Changes

**Lancer un build**

Configurer

### Pipeline Twitter-api-IC

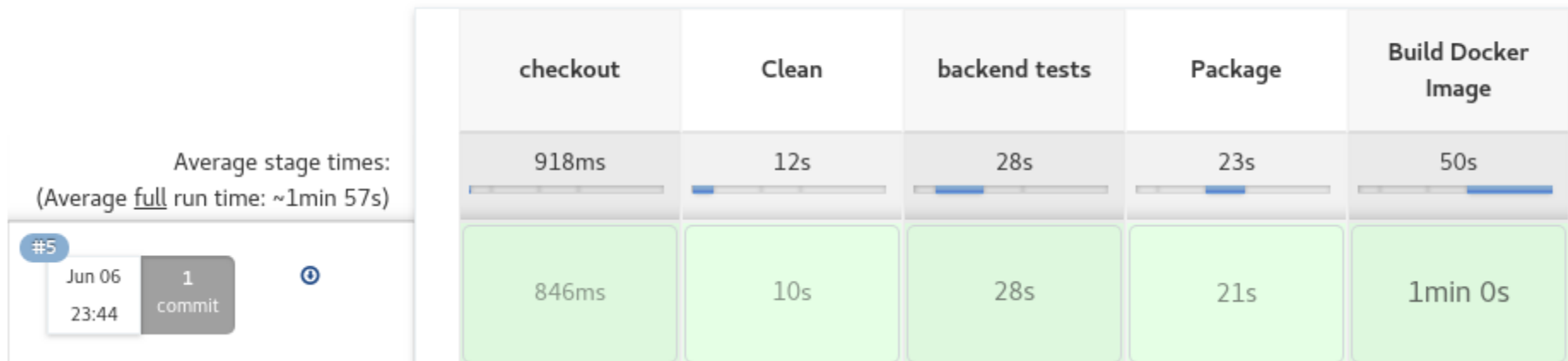
Last Successful Artifacts

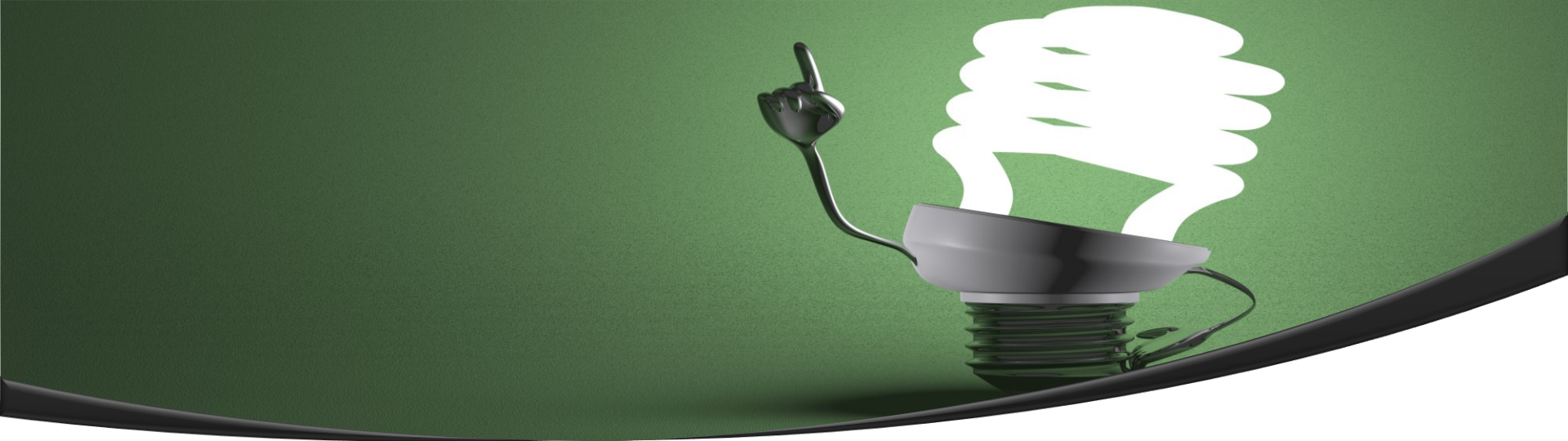
springdoc-openapi-spring-boot-2-webflux-2.5.0.jar	31,48 MB	<a href="#">view</a>
springdoc-openapi-spring-boot-2-webflux-2.5.0.pom	2,76 KB	<a href="#">view</a>

Recent Changes

# TP9: JENKINS - EXEMPLE DE RÉSULTAT D' EXÉCUTION

## Stage View





# DOCKER COMPOSE

## TP10: DOCKER COMPOSE - INSTALLATION DOCKER COMPOSE

- La dernière version de docker-compose est disponible sous github:
  - <https://github.com/docker/compose/releases>
  - Cela permet d'identifier la dernière version qui sera à installer
- Lancer les commandes suivantes pour installer docker-compose:

```
curl -L https://github.com/docker/compose/releases/download/1.29.2/docker-  
compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose
```

```
chmod +x /usr/local/bin/docker-compose
```

- Tester l'installation:

```
docker-compose --version
```



## TP10: DOCKER COMPOSE - TESTER L'IMAGE

- Modifier le fichier app.yml sous github: répertoire src/main/docker
- Changer dans image: **login-docker-hub/nom-repo:latest**

```
version: '2'
services:
  devops1-app:
    image: bnasslahsen/devops-repo:latest
    environment:
      - SPRING_PROFILES_ACTIVE=prod
      - SPRING_DATASOURCE_URL=jdbc:mysql://devops1-mysql:3306/devops1?useUnicode=true&characterEncoding=utf8&useSSL=false
      - SPRING_JPA_HIBERNATE_DDL-AUTO=update
    ports:
      - 8081:8080
  devops1-mysql:
    extends:
      file: mysql.yml
      service: devops1-mysql
```

## TP10: DOCKER COMPOSE - TESTER L'IMAGE

- Sous Centos, aller dans le répertoire /opt:

- Créer et aller dans le sous-répertoire: app

```
cd /opt; mkdir app; cd app
```

- Télécharger les conf docker-compose (fichiers dans src/main/docker), en adaptant votre username:

```
curl -O https://raw.githubusercontent.com/username/twitter-api/main/src/main/docker/app.yml  
curl -O https://raw.githubusercontent.com/username/twitter-api/main/src/main/docker/mysql.yml
```

- Lancer l'application, avec la commande:

```
docker-compose -f app.yml up
```

- Tester l'accès depuis l'url: <http://localhost:8081>

- Arrêter l'application:

```
docker-compose -f app.yml stop
```

- Pour supprimer le conteneur:

```
docker-compose -f app.yml rm
```



## QUESTIONS / RÉPONSES