

Exercices:

Pour réaliser correctement les exercices. Il faudrait avoir un compte github, créer un projet maven ou nodeJS.

L'objectif des exercices est de construire son pipeline au fur et à mesure en partant du build du test, de packaging et du déploiement de l'artefact et de l'application en intégrant les tests de performance.

Pour la création de projet maven on pourra générer un projet spring boot:

1. Spring Initializer : <https://start.spring.io/> génération d'un projet spring boot admin vide
2. Jhipster online : <https://start.jhipster.tech/> génération d'une application complète voir en Annexe 3
3. Pour un projet pur NodeJS. On pourra utiliser le projet Ghost détaillé dans l'annexe du cours.
4. fork du cargo-tms, simulation d'un transport management system : git clone [git@github.com:bilonjea/cargo-tms.git](https://github.com/bilonjea/cargo-tms.git) ; projet Java **21**, Maven **3.9+** Node **20+** (Angular 20) pnpm (gestionnaire de dépendances recommandé), Docker & Docker Compose (via WSL Ubuntu) :

Pour le jenkins on a le choix entre:

- Une VM disponible
- Un docker local

L'application pourra être déployée:

- sur une VM Mise disposition:

Pour l'analyse de code Sonar: on aura le choix

- Sonar Cloud: <https://sonarcloud.io/sessions/new>
- Une VM Sonar sera disponible
- Un docker sonar pourra tourner en local

Pour l'archivage des artefacts :

- **JFrog Artifactory** - auto-hébergé ou Cloud (SaaS) : <https://jfrog.com/>
- **Nexus Repository Community/OSS** - auto-hébergé
- **GitHub Packages (GHCR)** - gratuit pour les projets publics (facturation possible pour le privé).

Exercice 1

Création et configuration d'un Projet Maven Jenkins basé sur un repo Git.

Exercice 2

Adaptation du projet Jenkins pour inclure les outils d'analyse de qualité de code et suivi de la qualité dans le temps: **SpotBugs + Warnings NG**.

Exercice 3

Adaptation du projet ajouter **JaCoCo XML** et brancher SonarQube ou SonarCloud.

Exercice 4

Publier un artefact Maven/npm sur un depot de package de votre choix

Exercice 5

déployer sur une VM (scp + systemd + Nginx).

Exercice 6:

pipeline complet avec **Jenkinsfile déclaratif**, étapes *build* → *test* → *analyse* → *artefacts* → *déploiement*.

Integration de notification jira

test de perf jmeter