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 en pourra générer un projet spring boot ont peut utiliser les générateur de projet en ligne :

1. Spring Initializer : https://start.spring.io/ génération d'un projet spring boot admin vide
2. Jhispter online : https://start.jhipster.tech/ génération d'un application complète

Pour le projet NodeJS. On pourra utiliser le projet Ghost détaillé dans l'annexe du cours.

Pour le jenkins on a le choix entre:

- Une VM disponible
- Un docker local

L'application pourra être déployée:

- sur les serveur heroku : https://www.heroku.com/
- Ou 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 artéfact

- Le gestionnaire de dépôts d'objets binaires Jfrog https://jfrog.com/ de Artifactory (https://www.jfrog.com/confluence/display/RTF/Welcome%20to%20Artifactory)
- Une VM Jfrog
- Un docker local
- Sur Jenkins

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.

Exercice 3

Adaptation du projet Jenkins pour inclure JUnit et suivi des tests de performance avec JMeter.

Exercice 4

Adaptation du projet Jenkins pour automatiser le déploiement de l'artefact construit.

Exercice 5

Adaptation du projet Jenkins pour automatiser le déploiement de l’application sur serveur distant.