	DevSecOps : Tests Unitaires et chaîne d'intégration, cas d'une application web MVC

I. Introduction

I.1 Description

L'utilisation d'un Framework moderne, la vérification du code via des tests unitaires et la mise en place d'une chaîne d'intégration sont des éléments indispensables lors du développement d'une application moderne et sécurisée.
Dans cet atelier, à partir d'une application créée avec le Framework MVC Spring Boot, nous vous proposons de mettre en place les tests unitaires Junit et de vérifier le taux de couverture du code.
Vous mettrez également en place la chaîne d'intégration du projet sur la plateforme de DevSecOps Gitlab pour automatiser le cycle de vie de l'application.

I.2 Présentation

Ayant la volonté d'intégrer dans ses cours la méthodologie DevSecOps, il est parfois difficile de savoir comment commencer, quels outils choisir et comment les utiliser.
Je vous propose d'utiliser un Framework MVC Java open source, Spring boot.
Spring boot a plusieurs qualités que nous allons utiliser dans cet atelier :

- C'est un Framework MVC offrant un cadre clair de développement
- Il s'appuie sur le gestionnaire d'automatisation Maven
- Il contient un serveur web Tomcat intégré.

Concernant la mise en place de la chaîne d'intégration et les jobs associés, nous nous appuierons sur la forge logicielle Gitlab. Elle contient nativement tous les outils nécessaires à la réalisation de l'objectif.
Les concepts présentés ci-après sont adaptables à d'autres plateformes et langages comme Laravel en fonction des besoins mais présente un panorama des étapes clés à répliquer.
L'exemple utilisé dans cet atelier est basé sur une séquence pédagogique réalisée avec des étudiants de CIEL-IR1 réplicable ou adaptable dans vos sections de BTS.

II. Prérequis

II.1 Prérequis de connaissances

- Utilisation de git pour versionner les projets logiciels
- Connaissance d'un langage de programmation objet

II.2 Prérequis logiciels

- Un JDK (openjdk-17)
- Maven
- VsCode et les extensions nécessaires.
- Un compte Gitlab. Nous vous proposons d'utiliser la Forge des communs numériques éducatifs dans cette étude.

II.3 Procédure d'installation des prérequis logiciel et vérification

*L'installation est proposée sur **Linux - Debian 12** mais est adaptable sur les autres OS (Windows et MacOs)*

- Installer un jdk (Outil Java Development Kit - Je conseille openjdk 17 pour la suite de l'étude)

```
sudo apt install openjdk-17-jdk
```

- Installer Maven (outil de gestion et d'automatisation de production des projets logiciels Java)

```
sudo apt install maven
```

- Installer VsCode (éditeur de codes)

- Installer les extensions VsCode suivantes
  - Extension Pack for Java
    - Langage support for java (by Redhat)
    - Degugger for Java
    - Test runner for Java
    - Maven for Java

- Project Manager for Java
- Django

- Django template support
- Live preview
- Live server
- Red Hat Dependency Analytics

II.4 Valider l'installation

L'installation sera validée au début de la première étude.

III. DevSecOps – 2 applications

III.1 Objectif de l'étude :

Comme indiqué dans la description, l'objectif de l'étude est de créer les Tests Unitaires et d'Intégration d'une application MVC dans une logique DevSecOps.
Dans cette logique, un maximum d'étapes du développement doivent être automatisées dans la chaîne d'intégration de la forge logicielle pour que l'équipe de développement soit immédiatement avertie en cas de problème, comme un bug de régression ou une faille de sécurité dans une dépendance ou le code.
Dans la mise en place du projet, idéalement, l'environnement de développement est le même que tous les environnements du projet, comme le staging ou la production.
C'est pour cela qu'ici, je conseille notamment l'utilisation de open-jdk 17 car c'est la version du JDK qui sera utilisé dans le pipeline d'intégration et en production.

III.2 Déroulé :

Dans cette étude, je vais vous guider pour progressivement créer

- Une application MVC (Spring boot)
- Le pipeline d'intégration sur Gitlab
- Les Tests Unitaires de l'application
- Les Tests d'intégration
- Le pipeline de déploiement

Pour cela, vous allez créer deux applications

- Application CyberDemo.
  - Une application simple pour étudier la chaîne d'intégration
- Application GMP (Gestionnaire de Mots de Passe).
  - Une application MVC réutilisant la chaîne d'intégration et incluant les Tests Unitaires et d'Intégration.
  - L'objectif final est de déployer automatiquement sur un serveur cloud (support AWS) l'application.

III.3 Application 1 - CyberDemo

Comme vous allez utiliser Gitlab, je vous propose de retrouver les explications pour réaliser cette application sur la Forge des communs numériques éducatifs.

Adresse de la forge : https://forge.apps.education.fr
Adresse de l'étude : https://forge.apps.education.fr/pnf-ciel/application1-cyberdemo

III.3.1 Réalisation de l'application 1 – CyberDemo

Vous retrouverez le pas à pas pour créer la première application sur le wiki du projet hébergé sur la forge des communs.

Wiki du projet : https://forge.apps.education.fr/pnf-ciel/application1-cyberdemo/-/wikis/home

III.3.2 Conclusion de cette étude

Dans cette première étude, vous avez pu découvrir Spring boot MVC et son serveur Tomcat intégré qui facilite le développement d'application web.
Vous avez pu réaliser un pipeline d'intégration (CI ou Continuous Integration) avec la forge logicielle Gitlab et notamment les étapes (jobs) du pipeline :

- Build pour la compilation du projet
- Tests Unitaire pour automatiser l'exécution des Tests Unitaires sur Gitlab
- Code coverage pour vérifier le pourcentage du code couvert lors de l'exécution des Tests Unitaires.
- Templates DevSecOps Gitlab. Ce sont des modèles proposées par Gitlab, disponibles pour différents langages de programmation pour éviter des failles de sécurité.

Quel que soit le projet logiciel, il est indispensable de mettre en place la chaîne d'intégration dès le démarrage de celui-ci.
Selon votre version de Gitlab (free ou ultimate), vous n'aurez pas à votre disposition les mêmes templates ni résultats visuels. Cependant, même la version basique offre des capacités intéressantes à exploiter.

En tant qu'établissement scolaire, nous pouvons bénéficier gratuitement de la version **Ultimate** en rejoignant le programme Gitlab for education.

Adresse du corrigé de l'application 1 : https://forge.apps.education.fr/pnf-ciel/corrig-s/cyberdemo-corrige

III.4 Application 2 – Gestionnaire de Mots de Passe

Cet exemple est tiré d'une séquence proposée en BTS CIEL-IR1
L'objectif de la séquence étudiant était de :

- Créer un gestionnaire de mots de passe (keypass) sécurisé
- Application console puis Web (optionnel)

En ce qui concerne cette étude, je vous propose l'application GMP fonctionnelle, l'objectif de l'étude est :

- D'intégrer les Tests Unitaires et d'Intégration (TU et TI)
- De réaliser la partie déploiement continue (CD ou Continuous Delivery)

Adresse de l'étude : https://forge.apps.education.fr/pnf-ciel/application2-gmp

III.4.1 Réalisation de l'application 2 – GMP.

Vous retrouverez le pas à pas pour créer la première application sur le wiki du projet hébergé sur la forge des communs.

Wiki du projet : https://forge.apps.education.fr/pnf-ciel/application2-gmp/-/wikis/home

III.4.2 Conclusion de cette étude.

Cette étude propose un cas d'utilisation réaliste sur une application qui pourrait être réalisé par vos étudiants.
Ici, je vous ai proposé le code des TU et TI pour pouvoir s'attarder sur le process de mise en place de ceux-ci dans la logique DevSecOps du développement moderne d'applications.
Dans le développement logiciel, la sécurité d'une application demande de la rigueur et la logique DevSecOps nous offre un cadre de travail automatisant les contrôles et donc augmentant la productivité du développeur pour qu'il puisse se concentrer sur les tâches métiers.

Adresse du corrigé de l'application 2 : https://forge.apps.education.fr/pnf-ciel/corrigs/application2-gmp-corrige