Homework_Lesson_30

Задание:
1. Изучите основы pipeline CI/CD и настройте интеграцию своего проекта с платформой управления исходным кодом (например, GitHub).
2. Определите этапы pipeline для своего проекта (например, сборка, тестирование, развертывание), и автоматизируйте каждый этап с помощью инструментов pipeline CI/CD, таких как Jenkins, Gitlab или Github Actions.
3. Создайте репозиторий на Github и добавьте исходный код вашего приложения.
4. Создайте Dockerfile для сборки образа приложения.
5. Установите VSC на свой компьютер.
6. Откройте ваш проект в VSC.
7.Настройте автоматическое форматирование кода и проверку ошибок в реальном времени в VSC, используя соответствующие расширения (например, Pylance для Python, ESLint для JavaScript и т.д.).
8.Разработайте решение для сборки вашего проект с использованием механизмов параллельной сборки.
9.Проверьте свое решение, запустив pipeline CI/CD для вашего проекта, отслеживая метрики качества и оптимизируя процесс сборки.

Jenkins + SonarCloud
mvn clean verify package sonar:sonar -X \
-Dsonar.host.url="https://sonarcloud.io/" \
-Dsonar.organization="YOUR_PROJECT_ORGANIZATION" \
-Dsonar.projectKey="YOUR_PROJECT_KEY" \
-Dsonar.projectName="YOUR_PROJECT_NAME" \
-Dsonar.login=$(sonar_token)

Jenkins+Snyk
export SNYK_TOKEN=$(snyk_Token)
mvn snyk:test -fn
displayName: "Integrate SCA scan using Snyk in ADO DevSecOps Pipeline"

*****Jenkins+OWASP ZAP для проверки вашего кода (Static Application Security Testing (SAST))**
export JAVA_OPTS="-Xmx512m"
wget https://github.com/zaproxy/zaproxy/releases/download/v2.16.0/ZAP_2.16.0_Linux.tar.gz
tar -xvf ZAP_2.16.0_Linux.tar.gz
cd ZAP_2.16.0
./zap.sh -cmd -quickurl https://www.example.com -quickprogress -quickout /home/vsts/work/1/a/zap_report.html

Параллелизм уже добавлен в 28 домашку, продолжать не буду, стало скучно