**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