**Контеjнеризациjа и инсталациjа во облак**

* **Dockerfile (За Java Spring Boot апликацијата)**

Ова е главниот **Dockerfile**, кој се користи за да се спакува и стартува **StockMap** апликација што е напишана во Java со Spring Boot.

**Што прави?**

1. **Користи официјална Java 17 слика** за извршување на Java апликации.
2. **Поставува работен директориум /app** во кој ќе се сместат сите фајлови.
3. **Ги копира сите проектни фајлови** од компјутерот во контејнерот.
4. **Го гради проектот со Maven** и го создава **JAR фајлот** што ќе се стартува.
5. **Отвора порта 8080** за да може апликацијата да биде достапна.
6. **Стартува Java апликација со командата java -jar target/StockMap-0.0.1-SNAPSHOT.jar**.

* **Dockerfile.python (За Python анализите)**

Бидејќи имаме **Python скрипти** (LSTM.py, stockMap.py, technical\_analysis.py), потребен е посебен **Dockerfile** за да се извршуваат Python скриптите.

**Што прави?**

1. **Користи Python 3.9 слика** за извршување на Python код.
2. **Го поставува директориумот /app** за да ги сместиме сите Python фајлови.
3. **Го копира requirements.txt** и ги инсталира сите Python библиотеки.
4. **Ги копира останатите Python фајлови во контејнерот**.
5. **Стартува Python скрипта (stockMap.py) која анализира податоци.**

* **requirements.txt (Python зависности)**

**requirements.txt** се користи за да ги наведеме сите Python библиотеки што се потребни за анализите.

**Што содржи?**

Сите библиотеки што ги користат Python скриптите:

1. pandas
2. numpy
3. matplotlib
4. tensorflow
5. flask
6. requests

* **docker-compose.yml (За стартување на Java и Python заедно)**

Овој фајл овозможува **да ги стартуваме Java и Python контејнерите заедно** со една команда.

**Што прави?**

1. **Креира два сервиси (Java и Python)** што работат заедно.
2. **Java сервис (stockmap-app)**:
   * Се гради од главниот **Dockerfile**.
   * Ја стартува StockMap апликацијата.
   * Го отвора **портот 8080**.
3. **Python сервис (python-service)**:
   * Се гради од **Dockerfile.python**.
   * Ги користи Python анализите (stockMap.py).
   * Ја користи папката static/ за податоци и модели.
4. **Java апликацијата ќе го чека Python сервисот** пред да стартува.

* **Како работи сè заедно?**

**1. Се гради и стартува проектот со една команда**

docker-compose up --build

**2. Што се случува кога ќе се стартува командата?**

1. Docker **ја гради Java апликацијата (stockmap-app)** и ја стартува на порта 8080.
2. Docker **ја гради Python околината (python-service)**, ги инсталира зависностите и стартува stockMap.py.
3. Java апликацијата комуницира со Python сервисот за да земе податоци и анализи.
4. **Сите сервиси работат во Docker контејнери и се независни од локалниот систем.**

**Како се стопираат контејнерите?**

Кога сакаме да ги стопираме сервисите, притискаме **CTRL + C** или ја користиме командата:

docker-compose down