Практична робота №5

З дисципліни «Прикладна загальна теорія систем безпеки»

Студенту I курсу групи РЗ-201

Погорєльцева Павла Михайловича

Завдання:

1. Встановіть Docker;
2. Створіть Dockerfile;
3. Зберіть image на основі Dockerfile. Перевірте його наявність в списку образів. Запустіть контейнер. Перевірте працездатність вашого додатку. Зупиніть контейнер.
4. В кореневому каталозі проєкту створіть docker-compose.yml. Внесіть в нього конфігурацію для запуску вашого web-додатку.
5. Запустіть стек за допомогою docker compose up і впевніться що ваш додаток доступний ззовні.

Вказати:

1. Посилання на GitHub репозиторій проєкту;
2. Кроки виконання роботи;
3. Вміст файлів Dockerfile, docker-compose.yml;
4. Скріншоти images;
5. Підтвердження роботи додатку.

Хід роботи

На мій комп’ютер (з операційною системою Windows) було встановлено Docker Desktop.

Було створено наступний Dockerfile:

|  |
| --- |
| ARG BUILD\_HOME=/build  FROM gradle:8-jdk-21-and-23 AS build-image  ARG BUILD\_HOME  WORKDIR $BUILD\_HOME  COPY --chown=gradle:gradle build.gradle settings.gradle ./  COPY --chown=gradle:gradle src ./src  RUN gradle --no-daemon build  FROM eclipse-temurin:21  ARG BUILD\_HOME  ENV BUILD\_HOME=$BUILD\_HOME  WORKDIR /app  # App name is dependant on the version  COPY --from=build-image $BUILD\_HOME/build/libs/pr4-direct-1.0.0.jar app.jar  ENTRYPOINT ["java", "-jar", "app.jar"] |

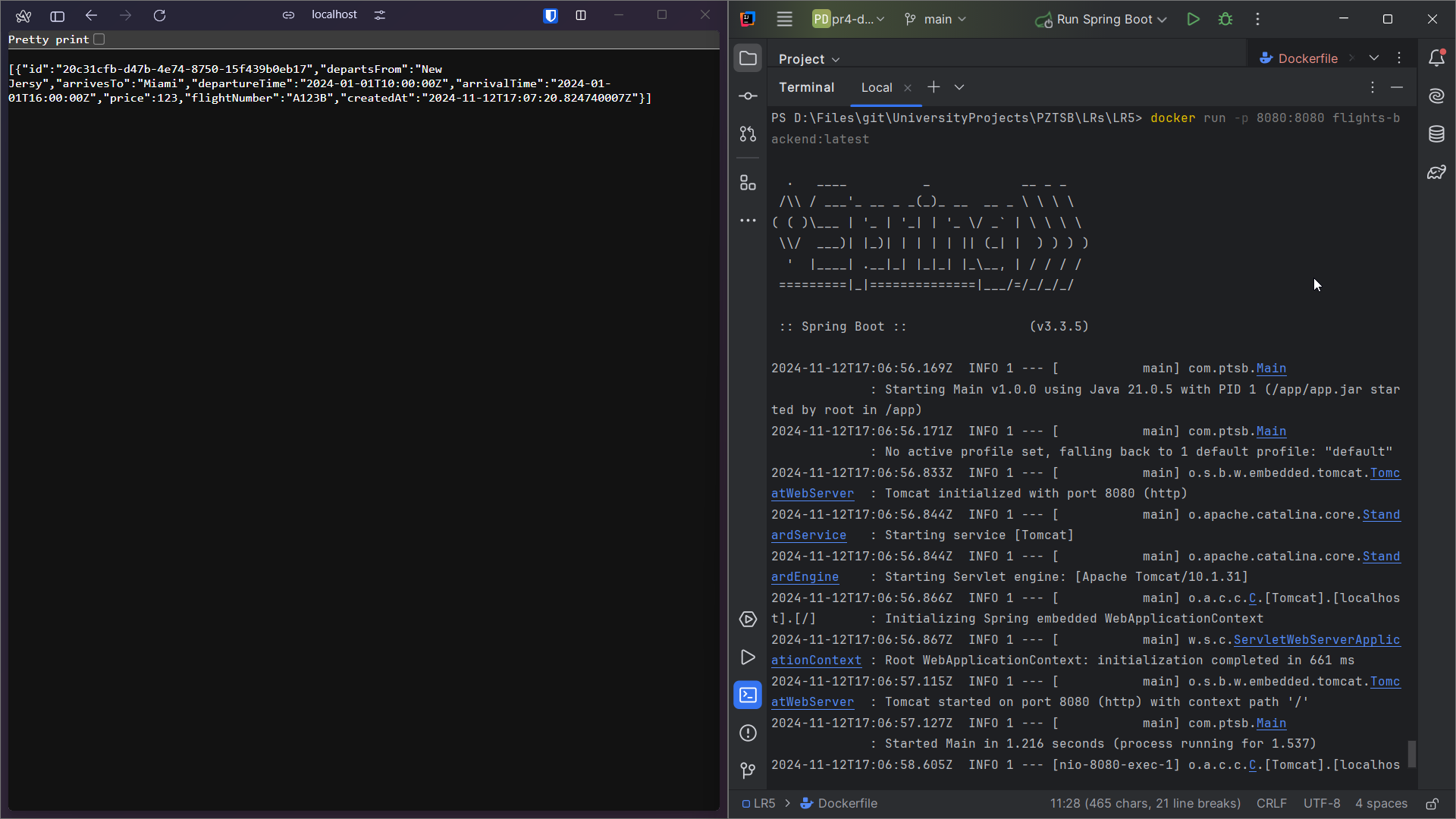
Створимо Docker Image:

|  |
| --- |
| PS D:\Files\git\UniversityProjects\PZTSB\LRs\LR5> docker build . -t flights-backend:latest  [internal] load build definition from Dockerfile  transferring 525/0 0.006  [internal] load metadata for docker.io/library/eclipse-temurin:21  [internal] load metadata for docker.io/library/gradle:8-jdk-21-and-23  [internal] load .dockerignore  [stage-1 2/3] WORKDIR /app  [build-image 2/5] WORKDIR /build  [internal] load build context  transferring 673/0 0.004  [build-image 1/5] FROM docker.io/library/gradle:8-jdk-21-and-23@sha256:5d98730af4ca79c37bf75a0def8598e93058980e7269253beb7093b6e4c468ea  [stage-1 1/3] FROM docker.io/library/eclipse-temurin:21@sha256:5ad4efff3364b06c61578b267138359bcba92acc20dfd533f35b75c709a6f10b  [build-image 3/5] COPY --chown=gradle:gradle build.gradle settings.gradle ./  [build-image 4/5] COPY --chown=gradle:gradle src ./src  [build-image 5/5] RUN gradle --no-daemon build  To honour the JVM settings for this build a single-use Daemon process will be forked. For more on this, please refer to https://docs.gradle.org/8.10.2/userguide/gradle\_daemon.html#sec:disabling\_the\_daemon in the Gradle documentation.  Daemon will be stopped at the end of the build  > Task :compileJava  > Task :processResourcesNO-SOURCE  > Task :classes  > Task :resolveMainClassName  > Task :bootJar  > Task :jar  > Task :assemble  > Task :compileTestJavaNO-SOURCE  > Task :processTestResourcesNO-SOURCE  > Task :testClassesUP-TO-DATE  > Task :test NO-SOURCE  > Task :check UP-TO-DATE  > Task :build    BUILD SUCCESSFULin 27s  4 actionable tasks: 4 executed  [stage-1 3/3] COPY --from=build-image /build/build/libs/pr4-direct-1.0.0.jar app.jar  exporting to image  exporting layers 0/0 0.127  writing image sha256:8c56d6a7395701f04ebfcf0b29816341c8a8bd595c2f8e282151a0e779f4497a 0/0 0.006  naming to docker.io/library/flights-backend:latest 0/0 0.012 |

Запустимо додаток:

|  |
| --- |
| docker run -p 8080:8080 flights-backend:latest |

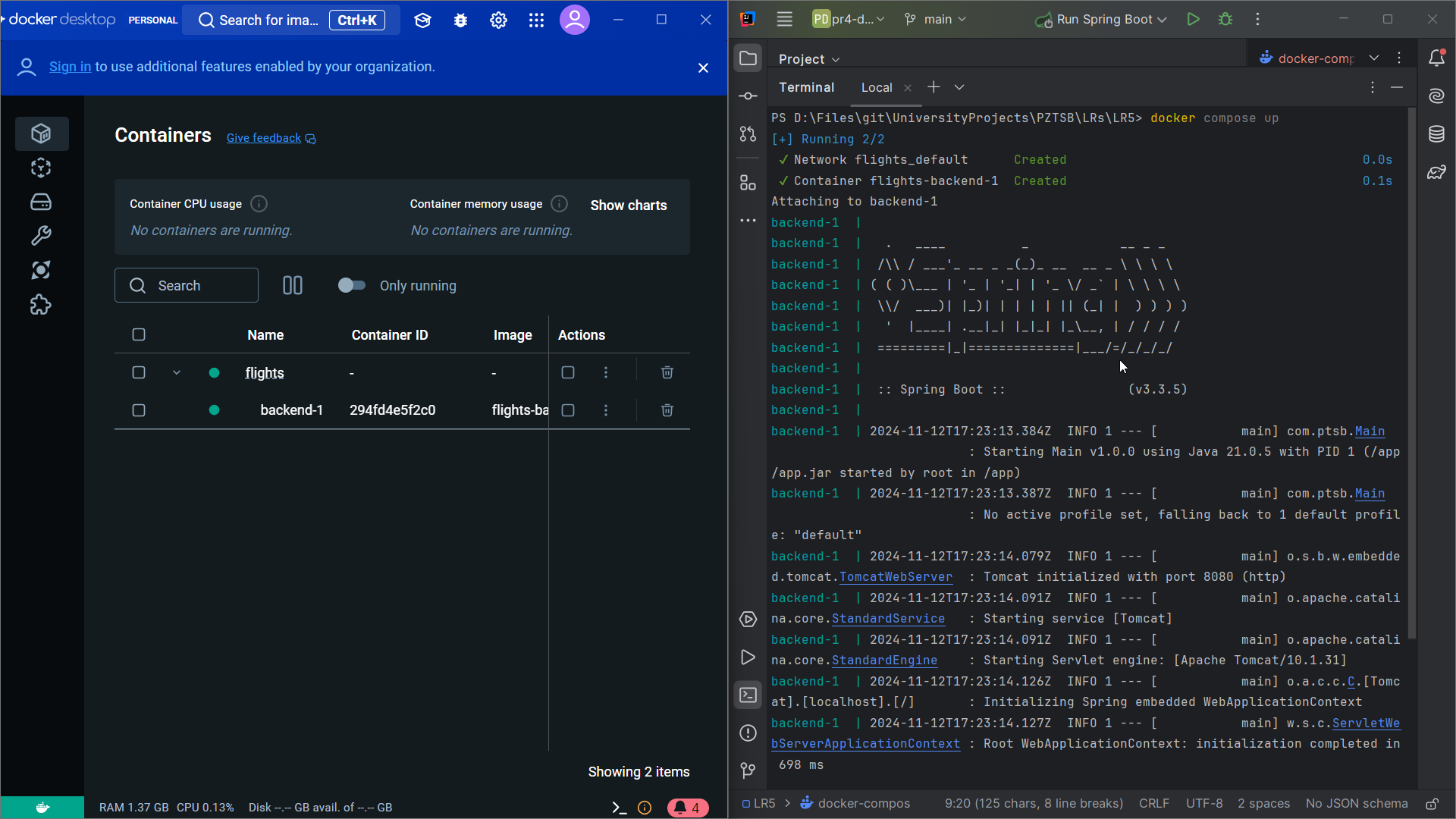
Результат (та перевірка роботи сервісу) на зображенні нижче:



Створимо compose файл для цього сервісу з наступним змістом:

|  |
| --- |
| name: "Flights"  services:  backend:  build:  context: .  dockerfile: Dockerfile  ports:  - "8080:8080" |

Запустимо сервіс з використанням docker compose up.



Перевіримо роботу додатку у браузері:

