



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
Wydział Fizyki i Informatyki Stosowanej

KATEDRA INFORMATYKI STOSOWANEJ I FIZYKI KOMPUTEROWEJ

Praca dyplomowa

*System do akwizycji danych z rozproszonych systemów
pomiarowych*
*System for data acquisition from distributed measurement
systems*

Autor:
Kierunek studiów:
Opiekun pracy:

Mateusz Barnacki
Informatyka Stosowana
dr inż. Antoni Dydejczyk

Kraków, 2023r.

Spis treści

1. Wstęp
2. Przegląd wzorców architektury aplikacji
3. Elementy charakterystyczne dla architektury mikrousługowej
4. Zapewnienie jakości w projekcie IT
5. Architektura aplikacji
6. Technologie wykorzystane w projekcie
7. Dokumentacja komponentów
8. Podsumowanie

1. Wstęp

2. Przegląd wzorców architektury aplikacji

3. Elementy charakterystyczne dla architektury mikrousługowej

4. Zapewnienie jakości w projekcie IT

5. Architektura aplikacji

6. Technologie wykorzystane w projekcie

7. Dokumentacja komponentów

7.1. Wdrożenie aplikacji na serwer wydziału

Jednym z problemów jakie należało rozwiązać w trakcie projektowania aplikacji było znalezienie sposobu na względnie szybkie zbudowanie i uruchomienie systemu na dowolnym środowisku. Istotnymi elementami przy wyborze odpowiedniego narzędzia było uwzględnienie popularności wykorzystania danej technologii przez środowisko programistów oraz przejrzyste napisana dokumentacja. Ponadto ze względów formalnych narzędzie musi być darmowe.

Uwzględniając wyżej wymienione kryteria autor pracy podjął decyzję o wykorzystaniu technologii konteneryzacji przy użyciu narzędzia Docker. Każdy kontener zawiera w sobie wszystkie potrzebne zależności do zbudowania oraz uruchomienia danego komponentu aplikacji. Kontenery są tworzone na podstawie obrazów, które w terminologii Dockera oznaczają niezmiennie szablony definiujące reguły budowania kontenerów. Obrazy są definiowane za pomocą instrukcji zawartych w specjalnych plikach o nazwie Dockerfile.

W przypadku zastosowania architektury mikrousługowej uruchomienie programu wymaga stworzenia większej ilości kontenerów. Narzędziem dedykowanym do zarządzania rozbudowaną infrastrukturą aplikacji jest Docker Compose, który umożliwia zbudowanie i uruchomienie programu przy użyciu jednej komendy. Reguły budowania projektu są umieszczone w pliku o nazwie docker-compose.yml. W ramach budowania wielokontenerowej aplikacji można ustalić m.in. kolejność budowania oraz uruchamiania kontenerów, lokalizację plików Dockerfile na podstawie których budowane są kontenery, lokalizację pliku zawierającego kopię zapasową dla baz danych.

Ważnym aspektem związanym z wykorzystaniem technologii konteneryzacji jest oddzielenie kontenera od środowiska zewnętrznego. Aplikacja działa w taki sam sposób na dowolnym systemie operacyjnym lub maszynie wirtualnej. Wobec tego można przetestować program na lokalnej maszynie nie wpływając na działanie systemu na środowisku produkcyjnym. W przypadku naprawy błędu wystarczy zreprodukować dane wejściowe, które spowodowały błąd i wprowadzić konieczne poprawki.

W celu uruchomienia aplikacji wymagana jest instalacja na docelowym urządzeniu systemu kontroli wersji Git oraz wyżej opisanego narzędzia Docker. System kontroli wersji Git służy do stworzenia lokalnej kopii zdalnego repozytorium. Jeżeli aplikacja jest uruchamiana na systemie operacyjnym Windows należy zainstalować dodatek o nazwie Git Bash. Uzasadnienie wykorzystania konsoli Git Bash nastąpi w kolejnej sekcji. Narzędzie Docker umożliwia automatyczne zbudowania oraz uruchomienia aplikacji.

W momencie pisania niniejszej pracy kod źródłowy programu jest umieszczony na platformie GitHub. W celu zabezpieczenia wrażliwych informacji dotyczących sposobu budowania API-Key oraz JWT repozytorium zostało oznaczone jako prywatne. Wydział udostępnił maszynę wirtualną wykorzystującą system operacyjny Rocky Linux. Maszyna znajduje się pod adresem 172.20.40.211. W celu uruchomienia systemu zainstalowałem wszystkie potrzebne aplikacje wymienione powyżej. Działanie aplikacji można sprawdzić tylko i wyłącznie będąc zalogowanym do sieci wydziału. Strona logowania do systemu znajduje się pod adresem <http://172.20.40.211:3000>.

7.2 Pierwsze uruchomienie systemu

Poniższa instrukcja zawiera opis wykorzystania komend w dowolnym wierszu poleceń. Część niżej wymienionych operacji może zostać zastąpiona poprzez wykorzystanie narzędzi posiadających interfejs użytkownika np. IntelliJ IDEA lub Docker Desktop.

Pierwszym krokiem wymaganym do uruchomienia systemu na dowolnej maszynie jest utworzenie lokalnej kopii repozytorium za pomocą komendy `git clone`. Aplikacja przy generowaniu JWT wykorzystuje parę asymetrycznych kluczy służących do kodowania oraz dekodowania żetonów. Wobec tego należy utworzyć klucz publiczny oraz klucz prywatny. Należy przenieść się do lokalizacji `master-thesis/auth-service/src/main/resources`. Następnie użytkownik musi utworzyć nowy katalog o nazwie `certs`. W katalogu `certs` należy wykonać poniższą sekwencję komend.

```
1 openssl genrsa -out keypair.pem 2048
2 openssl rsa -in keypair.pem -pubout -out public.pem
3 openssl pkcs8 -topk8 -inform PEM -outform PEM -nocrypt -in keypair.pem -out private.pem
```

Rysunek 1 Utworzenie klucza publicznego oraz klucza prywatnego

Biblioteka OpenSSL jest dostępna na systemach operacyjnych MacOS oraz Linux. W przypadku systemu operacyjnego Windows rekomendowane jest zainstalowanie konsoli Git Bash, która jest dodatkiem przy instalacji systemu kontroli wersji Git i zawiera wbudowaną wersję biblioteki OpenSSL. Pierwsza linia generuje klucz prywatny za pomocą algorytmu RSA o rozmiarze 2048 bitów i zapisuje go do pliku `keypair.pem`. Druga instrukcja procesuje wygenerowany klucz prywatny i generuje na jego podstawie klucz publiczny zapisując go do pliku `public.pem`. Ostatnia komenda procesuje klucz prywatny z pliku `keypair.pem` w formacie PKCS8 i zapisuje go do pliku `private.pem`. Opcja `-nocrypt` została użyta ze względu na wykorzystywanie niezakodowanych kluczy asynchronicznych we frameworku Spring Boot Security. Po utworzeniu plików `public.pem` oraz `private.pem` można usunąć plik `keypair.pem` oraz skopiować cały katalog do folderu `master-thesis/api-gateway/src/main/resources`.

Kolejnym etapem jest wykorzystanie narzędzia Docker Compose. W tym celu należy przejść do folderu `master-thesis/` i wykonać poniższą komendę.

```
1 docker compose up -d
```

Rysunek 2 Zbudowanie oraz uruchomienie aplikacji

Powyższa komenda umożliwia wygenerowanie obrazów Dockera na podstawie instrukcji zawartych w plikach Dockerfile oraz stworzenie kontenerów na podstawie powstałych obrazów. Opcja `-d` umożliwia wyłączenie śledzenia logów aplikacji po zakończeniu procesu budowania kontenerów.

Aplikacja po pierwszym uruchomieniu zawiera puste bazy danych. Utworzenie schematu bazy danych wymaga wykorzystania narzędzia konsolowego `psql`. Bazy danych znajdują się w kontenerach, wobec tego komunikacja musi odbyć się za pośrednictwem Dockera. Poniższa komenda umożliwi uruchomienie konsoli użytkownika PostgreSQL.

8. Podsumowanie

Bibliografia