Uniwersytet Gdański
Wydział Matematyki, Fizyki i Informatyki
Instytut Informatyki

Wordle

Karol Wiśniewski

Projekt z przedmiotu technologie chmurowe na kierunku informatyka profil praktyczny na Uniwersytecie Gdańskim.

Gdańsk

11 czerwca 2023

Spis treści

1 Opis projektu

Pewna firma o nazwie "Language Learners Inc."zajmujaca sie edukacja jezykowa postanowiła wdrożyć innowacyjne narzedzie, które pomogłoby osobom uczacym sie jezyków obcych rozwijać swoje umiejetności słownictwa w sposób interaktywny i angażujacy. Przez wiele lat nauki słownictwa w tradycyjny sposób było monotonne i nieprzyjemne, co czesto powodowało brak motywacji u uczniów.

Aby temu zaradzić, firma Language Learners Inc. zdecydowała sie zamówić aplikacje o nazwie Wordle, która pozwoliłaby użytkownikom na nauke słownictwa w ciekawy i zabawny sposób. Celem aplikacji Wordle jest dostarczenie interaktywnego doświadczenia, w którym użytkownicy mogliby zgadywać słowa na podstawie podpowiedzi i zyskiwać punkty za poprawne odpowiedzi.

1.1 Opis architektury - 8 pkt

Aplikacja Wordle jest oparta na Kubernetes, który jest otwartoźródłowym systemem zarzadzania kontenerami. Kubernetes umożliwia skalowalność, zarzadzanie, automatyzacje i odizolowanie aplikacji w kontenerach. Dodatkowo, aby uprościć poruszanie sie po aplikacji, adres IP klastra jest utożsamiany z adresami project.baw (main), api.project.baw (backend), keycloak.project.baw (keycloak). Każdy z tych adresów jest odpowiednio interpretowany dzieki dodatkowej warstwie Ingress, która działa bardzo podobnie do default.conf serwera nginx. Jest to "interpreter"poruszania sie po aplikacji. Sztuczne domeny stworzone zostały za pomoca konfiguracji pliku "hosts"w katalogach źródłowych systemu Windows. Działaja one na takiej samej zasadzie jak 127.0.0.1 i "localhost".

1.2 Opis infrastruktury - 6 pkt

Cały klaster Kubernetes jest hostowany przez maszyne wirtualna, stworzona za pomoca narzedzia minikube. Jej specyfikacja to: 10GB pamieci masowej, 6GB pamieci operacyjnej (RAM) oraz 2 wirtualne CPU.

1.3 Opis komponentów aplikacji - 8 pkt

Frontend (React): Moduł odpowiedzialny za interfejs użytkownika aplikacji. Zbudowany w oparciu o framework React. Został wdrożony poprzez customowy Dockerfile, zbudowany na Dockerze samej maszyny wirtualnej, na której uruchomiony jest klaster. Jego ścieżki obsługuje dodatkowy serwer nginx z dostosowanym plikiem konfiguracyjnym default.conf.

Backend (Express): Moduł odpowiedzialny za logike biznesowa aplikacji. Wykorzystuje framework Express do obsługi żadań HTTP. Tak samo jak frontend, został on wdrożony poprzez customowy Dockerfile na maszynie wirtualnej. Jego endpointy sa chronione przez biblioteke keycloak-connect i wymagaja zalogowania sie w serwisie. Dodatkowo, jeden z nich wymaga od użytkownika statusu admina.

Keycloak: Moduł odpowiedzialny za autentykacje i zarzadzanie tożsamościami użytkowników. Keycloak jest otwartoźródłowym narzedziem do zarzadzania tożsamościami opartym na protokole OpenID Connect.

PostgreSQL: Moduł bazy danych, który przechowuje dane zwiazane z tożsamościami użytkowników Keycloak. Dzieli swoje dane z PVC (Persistent Volume Claim) na klastrze, aby konfiguracja serwisu keycloak pozostawała taka sama, nawet po całkowitym restarcie klastra Kubernetes.

1.4 Konfiguracja i zarzadzanie - 4 pkt

Frontend: korzysta wyłacznie z jednego serwisu, który definiuje na jakim porcie ma stać ta cześć klastra.

Backend: wykorzystuje jedna ConfigMape oraz jeden serwis. ConfigMapa ma za zadanie dostarczyć backendowi niezbedne zmienne, takie jak FRONT-ORIGIN (skad beda przychodzić zapytania do API) oraz kilka zmiennych konfiguracyjnych dla biblioteki keycloak-connect. Serwis zaś, ma za zadanie zdefiniować port, na którym bedzie nasłuchiwał backend.

Postgres: w przypadku tej warstwy potrzebna była porzadna konfiguracja. Wykorzystuje ona jeden Secret, jedna ConfigMape oraz jeden PVC (Persistent Volume Claim). Secret to bezpieczne miejsce dla zmiennych środowiskowych wykorzystywanych przez dany serwis. Przechowuje takie dane jak nazwa użytkownika bazy danych, hasło oraz nazwe samej bazy. ConfigMapa w tym przypadku jest inicjalizowana przez customowy skrypt bashowy, który bazuje na query SQL'owym, tworzacym baze Keycloak, potrzebna dla samego Keycloaka do persystencji danych. Tak jak w poprzednich przypadkach, Serwis ma za zadanie zdefiniować port nasłuchiwania dla danej warstwy.

Keycloak: ta warstwa korzysta z jednego Serwisu i jedego Secreta. Serwis dzieli takie samo zadanie, jak w przypadku reszty warstw. Secret przechowuje delikatne dane do konfiguracji samego serwisu Keycloak oraz jego połaczenia z wcześniej opisana baza PostgreSQL.

1.5 Zarzadzanie błedami - 2 pkt

1.6 Skalowalność - 4 pkt

1.7 Wymagania dotyczace zasobów - 2 pkt

Frontend: wykorzystuje 200MB pamieci podrecznej i 0.2 CPU.

Backend: podobnie jak frontend, potrzebuje 0.2CPU, ale za to już 600MB pamieci podrecznej.

Keycloak: wykorzystuje najwiecej zasobów, bo aż 0.4 CPU i prawie 2GB pamieci podrecznej. Sa to minimalne wymagania dla tego serwisu.

Postgres: ma niewielkie wymagania, wykorzystuje zaledwie 0.1 CPU i 128MB pamieci podrecznej.

1.8 Architektura sieciowa - 4 pkt