Wykonanie zadania rozpocząłem od znalezienia w WSLu katalogu, w którym znajdowały się pliki z zajęć. Następnie z poziomu konsoli przeszedłem do VSCode. Kolejnym krokiem, jaki poczyniłem było zbudowanie obrazu poleceniem „sudo docker build –t moj_image:0.2 .” Następnie poleceniem „docker run moj_image:0.2” uruchomiłem stworzony wcześniej obraz. Kolejne etapy przeniesienia aplikacji do chmury odbywały się na platformie Azure. Początkowo stworzyłem grupę zasobów i rejestr kontenerów. Następnie wypchnąłem stworzony wcześniej obraz przy pomocy poleceń „docker login NAZWA_REJESTRU.azurecr.io”, „docker tag moj_image:0.2 rejestrpiotrek22.azurecr.io/moj_image:0.2” oraz „docker push rejestrpiotrek22.azurecr.io/moj_image:0.2”. Następnie uruchomiłem AppServices i wybrałem opcję „Aplikacja internetowa”, wybrałem stworzoną wcześniej grupę zasobów, nazwałem aplikację i wybrałem opcję Container. Kolejno przeszedłem do bazy danych. Jako źródło obrazu kliknąłem Azure Container Registry i wybrałem rejestr, który stworzyłem, zpushowany obraz, oraz tag, który nadałem temu obrazowi. Następnie przeszedłem do Serwerów elastycznych usługi Azure Database for PostgreSQL i kliknąłem utwórz, a następnie wybrałem stworzoną wcześniej grupę zasobów, nazwałem swój serwer jako „serwerpiotrek22” i ustawiłem dostępność jako 99,9%. Kolejno uzupełniłem pola dotyczące nazwy użytkownika administratora jako „piotrek22” oraz ustawiłem hasło do bazy danych. Następnie w zakładce „sieci” kliknąłem w plusa, obok którego znajdował się niebieski napis „dodaj 0.0.0.0 - 255.255.255.255”, a w zakładce zabezpieczenia pozostawiłem klucz przez usługę. Następnie przeszedłem do Azure Database for PostgreSQL i tam wybrałem stworzony wcześniej serwer. Wszedłem w mniejsze menu i wybrałem pozycję „Bazy danych” i kliknąłem dodaj, po czym nazwałem swoją bazę danych jako „fastapidb22”. Następnie przeszedłem do zakładki App Services i tam wybrałem swoją aplikację. Kliknąłem w zmienne środowiskowe, dodaj i uzupełniłem kolejno rubryki w taki sposób (Nazwa = Wartość):

- PORT = 8000

- WEBSITES_PORT = 8000

- CONNECTION_STRING = postgresql:// piotrek22:hasło_do_bazy_danych @serwerpiotrek22.postgres.database.azure.com:5432/fastapidb22. Przy connection_string zaznaczyłem kwadracik jako ustawienie miejsca wdrożenia. Finalnie przeszedłem do zakładki app services i kliknąłem w link obok napisu „Domena domyślna”. Po tej czynności otworzyło się okno z działającą aplikacją w chmurze.

(Link do aplikacji: https://aplikacjapiotrek22.azurewebsites.net/)

Instrukcja – jak uruchomić i zbudować obraz aplikacji lokalnie.

Do uruchomienia i zbudowania obrazu aplikacji potrzebny jest program Docker.

1. **Instalacja dockera w Ubuntu:**

   sudo apt update sudo apt install docker.io

2. **Pobranie kodu źródłowego aplikacji z repozytorium gita:**

   git clone

3. **Przejście do katalogu, do którego pobrany został kod źródłowy:**

   cd <lokalny_katalog>

4. **Budowanie obrazu Docker:**

   Należy przejść do głównego katalogu aplikacji, gdzie znajduje się plik Dockerfile. Następnie należy uruchomić poniższe polecenie, aby zbudować obraz Docker. sudo docker build -t nazwa_obrazu:tag .

5. **Uruchomienie kontenera Docker:**

   Po zbudowaniu obrazu można uruchomić kontener Docker korzystając z polecenia: sudo docker run -p port_hosta:port_kontenera nazwa_obrazu:tag

   port hosta – port, na którym chcemy uruchomić aplikację na swoim lokalnym systemie port kontenera – port, na którym aplikacja jest wystawiona wewnątrz kontenera Docker

Instrukcja – jakie kroki należy podjąć, aby uruchomić aplikację w chmurze Azure.

1. **Utworzenie konta w chmurze Azure:**

   Zalogowanie się na swoje konto Azure lub utworzenie nowego konta. Wybór odpowiedniej subskrypcji.

2. **Tworzenie zasobów:**

   Na portalu Azure znajdują się gotowe szablony zasobów, które ułatwiają konfigurację. Dla aplikacji webowych warto użyć Azure App Service, dla kontenerów Azure Container Instances lub Azure Kubernetes Service (AKS). W przypadku baz danych można użyć usługi Azure SQL Database lub Azure Cosmos DB w zależności od potrzeb.

3. **Konfiguracja zasobów:**

   Dostosowanie konfiguracji zasobów w zależności od potrzeb aplikacji, takich jak wielkość, lokalizacja, dostępność, zabezpieczenia. Upewnienie się, że wszystkie wymagane ustawienia, takie jak zmienne środowiskowe, klucze uwierzytelniania itp., są poprawnie skonfigurowane.

4. **Pobieranie obrazu aplikacji:**

Jeśli aplikacja jest w kontenerze Docker, można umieścić jej obraz w Azure Container Registry lub użyć istniejącego repozytorium (np. Docker Hub).

5. **Wdrażanie aplikacji:**

Wybór odpowiedniej usługi do wdrożenia aplikacji w chmurze Azure (np. Azure App Service, Azure Kubernetes Service, Azure Container Instances). Postępowanie zgodnie z instrukcjami, aby wdrożyć swoją aplikację, wskazując odpowiednie ustawienia i konfiguracje. Monitorowanie postępu wdrożenia i sprawdzenie, czy aplikacja działa poprawnie.

6. **Monitorowanie i zarządzanie:**

Konfiguracja odpowiednich narzędzi do monitorowania i logowania, takich jak Azure Monitor, aby śledzić wydajność i dostępność aplikacji. Zapewnienie skalowalności i elastyczności zasobów w zależności od obciążenia aplikacji. Regularne aktualizowanie aplikacji i zasobów, w celu utrzymania bezpieczeństwa i wydajności.

Instrukcja – jak postawić obraz bazy danych lokalnie.

1. **Instalacja Dockera:**

Początkowo należy zainstalować Docker na systemie Ubuntu, korzystając z poniższych poleceń:

sudo apt update

sudo apt install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update

sudo apt install docker-ce

Następnie poleceniem” sudo systemctl status docker” należy sprawdzić, czy Docker został poprawnie zainstalowany

Jeśli usługa nie działa należy ją uruchomić:

sudo systemctl start docker

sudo systemctl enable docker

2. **Pobranie obrazu PostgreSQL:**

sudo docker pull postgres

3. **Uruchomienie kontenera PostgreSQL:**

Należy uruchomić kontenet PostgreSQL, definiując przy tym nazwę użytkownika, hasło i nazwę bazy danych:

sudo docker run --name nazwa_kontenera_postgres -e POSTGRES_USER=nazwa użytkownika -e POSTGRES_PASSWORD=hasło -e POSTGRES_DB=nazwa_bazy_danych -p 5432:5432 -d postgres

4. **Sprawdzenie uruchomionego kontenera:**

Sudo docker ps

5. **Łączenie się z bazą danych w kontenerze:**

sudo docker exec -it nazwa_kontenera_postgres psql -U nazwa_użytkownika -d nazwa_bazy_danych