LABORATORIUM: Programowanie Aplikacji w Chmurze Obliczeniowej

Zadanie 1
Zadanie podzielone jest na dwie części : obowiązkową i dodatkową (nieobowiązkową)

CZĘŚĆ OBOWIĄZKOWA

1. (max. 20%)
Proszę napisać program serwera (dowolny język programowania), który realizować będzie następującą funkcjonalność:

a. po uruchomieniu kontenera, serwer pozostawia w logach informację o dacie uruchomienia, imieniu i nazwisku autora serwera (imię i nazwisko studenta) oraz porcie TCP, na którym serwer nasłuchuje na zgłoszenia klienta.
b. na podstawie adresu IP klienta łączącego się z serwerem, w przeglądarce powinna zostać wyświetlona strona informująca o adresie IP klienta i na podstawie tego adresu IP, o dacie i godzinie w jego strefie czasowej.

W sprawozdaniu proszę umieścić kod oprogramowania wraz z niezbędnymi komentarzami.

2. (max. 50%)
Opracować plik Dockerfile, który pozwoli na zbudowanie obrazu kontenera realizującego funkcjonalność opisaną w punkcie 1. Przy ocenie brane będzie sposób opracowania tego pliku (wieloetapowe budowanie obrazu, ewentualne wykorzystanie warstwy scratch, optymalizacja pod kątem funkcjonowania cache-a w procesie budowania, optymalizacja pod kątem zawartości i ilości warstw, healthcheck itd). Dockerfile powinien również zawierać informację o autorze tego pliku (ponownie imię oraz nazwisko studenta).

W sprawozdaniu proszę umieścić plik Dockerfile wraz z niezbędnymi komentarzami.

3. (max. 30%)
Należy podać polecenia niezbędne do:

a. zbudowania opracowanego obrazu kontenera,
b. uruchomienia kontenera na podstawie zbudowanego obrazu,
c. sposobu uzyskania informacji, które wygenerował serwer w trakcie uruchamiana kontenera (patrz: punkt 1a),
d. sprawdzenia, ile warstw posiada zbudowany obraz.

W sprawozdaniu należy podać treść poleceń (punkty a – d) w raz w ewentualnymi komentarzami oraz zrzut ekrany okna przeglądarki, potwierdzający poprawne działanie systemu.

UWAGA: Wszystkie informacje, które trzeba dostarczyć w sprawozdaniu (punkty 1-4) należy opracować w postaci pliku *zadanie1.md* a ten plik umieścić na koncie GitHub jako zwyczajowy opis zawartości repozytorium git. Na tym repozytorium proszę umieścić też opracowane źródła dla serwera oraz przygotowany plik Dockerfile (oraz wszystkie inne, niezbędne Państwa zdaniem pliki). ***Jako sprawozdanie należy przekazać wyłącznie plik tekstowy zawierający linki do użytego repozytorium na GitHub oraz DockerHub .***

Punkty dodatkowe (max. 100%)
Wśród wszystkich opracowań Zadania 1, część obowiązkowa wybrane zostanie rozwiązanie (ewentualnie rozwiązania) wykorzystujące obraz o najmniejszym rozmiarze i "nagrodzone" dodatkowymi punktami – 80%. **Warunkiem uwzględnienia obrazu w „konkursie" jest wykazanie, że żadne jego składowe nie uzyskują oceny CVSS w zakresie High lub Critical (powyżej 7.0) za pomocą narzędzia Docker Scount.** Spełnienie tego warunku należy udokumentować w sprawozdaniu za co zostanie przyznane 20% punktów.

CZĘŚĆ DODATKOWA ----------

UWAGA:
- Aby otrzymać punkty za części dodatkowe, należy wykonać **WSZYSTKIE** punkty z części obowiązkowej zadania 1.
- Należy przedstawić rozwiązanie TYLKO JEDNEGO z punktów poniżej (każdy punkt jest rozszerzenie kolejnego czyli zawiera zakres poprzedniego). Punktacja za rozwiązanie jest sumowana zgodnie z danymi poniżej.

1. (max. 10%)
Zbudować obrazy kontenera z aplikacją opracowaną w punkcie nr 1, które będą pracował na architekturach: **linux/arm64** oraz **linux/amd64** wykorzystując domyślnie skonfigurowany QEMU w Docker Desktop.

2. (max. +20%)
Zbudować obrazy kontenera z aplikacją opracowaną w punkcie nr 1, które będą pracował na architekturach: **linux/arm64** oraz **linux/amd64** wykorzystując sterownik docker-container.

3. (max. +30%)
Zbudować obrazy kontenera z aplikacją opracowaną w punkcie nr 1, które będą pracował na architekturach: **linux/arm64** oraz **linux/amd64** wykorzystując sterownik docker-container. Dockerfile powinien wykorzystywać rozszerzony frontend i umożliwiać wykorzystanie danych cache w procesie budowania obrazu (deklaracje wewnątrz Dockerfile).

4. (max. +40%)
Zbudować obrazy kontenera z aplikacją opracowaną w punkcie nr 1, które będą pracował na architekturach: **linux/arm64** oraz **linux/amd64** wykorzystując sterownik docker-container. Dockerfile powinien wykorzystywać rozszerzony frontend, zawierać deklaracje wykorzystania cache (jak w p.3) i umożliwiać bezpośrednie wykorzystanie kodów aplikacji umieszczonych w swoim repozytorium publicznym na GitHub.

Opracowane obrazy należy przesłać do swojego repozytorium na DockerHub. W sprawozdaniu należy podać wykorzystane instrukcje wraz z wynikiem ich działania oraz ewentualnymi komentarzami. W przypadku podpunktu 4 proszę nie przekazywać danych wrażliwych (jeśli takie będą wykorzystywane) a jedynie przedstawić wynik wykonywanych działań (poleceń).

UWAGA: Ponownie wszystkie informacje, które trzeba dostarczyć w sprawozdaniu z części nieobowiązkowej (punkty 1-4) należy opracować w postaci pliku *zadanie1_dod.md* a ten plik

umieścić na koncie GitHub jako zwyczajowy opis zawartości repozytorium git. Na tym repozytorium proszę umieścić też opracowane źródła dla serwera oraz przygotowany plik Dockerfile (oraz wszystkie inne, niezbędne Państwa zdaniem pliki). ***Jako sprawozdanie należy przekazać wyłącznie plik tekstowy zawierający linki do użytego repozytorium na GitHub oraz DockerHub .***

UWAGI KOŃCOWE:

1. Zadanie należy wykonać SAMODZIELNIE. W przypadku kopii (tak typu Ctr C – Ctr V) lub modyfikacji nieistotnych merytorycznie) ocena z zadania zostanie PODZIELONA przez liczbę „współtwórców".
2. Oceniający zastrzega sobie prawo do podwyższenia oceny jeśli dane rozwiązanie chwyci go za serce albo umysł.