

ZADANIE NR 2

Podstawą do wykonania zadań w zadaniu nr 2 jest aplikacja **Multifib**, której kody oraz omówienie zostały przedstawione w instrukcji do laboratorium 11 a przekładowe wdrożenie w oparciu o Github Actions (ponownie kod i omówienie) były przedmiotem instrukcji do laboratorium nr 12.

CZĘŚĆ OBOWIĄZKOWA

Część obowiązkowa obejmuje wdrożenie wielokontenerowej aplikacji typu full-stack w usłudze ElasticBeanstalk AWS. Przy tym wdrożeniu należy wykorzystać udostępniony kod lub przerobić/opracować aplikację o identycznej funkcjonalności (w tym drugim przypadku, PROSZĘ zapoznać się z warunkami otrzymania dodatkowych punktów (w „CZĘŚĆ NIEOBOWIĄZKOWA”))

1. Wymagane zmiany w dostarczonym kodzie aplikacji:

W skład interfejsu aplikacji wchodzi dwie strony webowe: pierwsza przeznaczona do wprowadzania współczynników ciągu i prezentacji wyników oraz druga: obecnie pusta. Do tej pustej strony należy dodać kod pozwalający na wyświetlenie następujących danych:

- 1) nazwę programu,
- 2) imię i nazwisko autora,
- 3) numer grupy dziekańskiej.

2. Sposób przygotowania aplikacji:

Docelowe wdrożenie oparte ma być o „produkcyjną” wersję pliku *docker-compose.yml* definiującego strukturę aplikacji. Wewnątrz tego pliku należy wykorzystać gotowe obrazy poszczególnych mikrouslug, przygotowanych wcześniej w fazie rozwojowej i przechowywanych w wybranym przez studenta repozytorium. Ponownie, można wykorzystać dostarczone pliki lub zmodyfikować/przygotować nowe, zgodne z przyjętą koncepcją aplikacji.

3. Sposób wdrożenia aplikacji:

Do realizacji wdrożenia należy użyć Github Actions. Analogicznie jak w poprzednich punktach, można wykorzystać dostarczony plik definiujący workflow GA lub opracować własne rozwiązanie. **UWAGA:** W tej części, w pliku workflow proszę nie wykorzystywać automatycznych metod tag-owania obrazów (np. poznanego semver).

4. Przygotowanie środowiska w chmurze AWS:

W chmurze AWS muszą być zdefiniowane wszystkie niezbędne usługi – analogicznie jak w przypadku instrukcji do laboratorium 12. Zgodnie z założeniem na wstępie, bazą wdrożenia ma być usługa ElasticBeanstalk. W ramach tej usługi należy wybrać dowolną strefę (region) – w przypadku braku pomysłu” us-east1 oraz zdefiniować nazwę aplikacji w ElasticBeanstalk, która ma mieć format `<inicjały+multifib>`, przykładowo, w moim przypadku byłaby to nazwa `spmultifib`. Reszta parametrów zależy od studenta **UWAGA:** Proszę pamiętać o ograniczaniu się w ramach „free tier”.

Kod aplikacji wraz ze wszystkimi plikami, niezbędnymi do realizacji opisanego wdrożenia, mają znajdować się na GitHub, w dowolnym repo publicznym studenta.

Sprawozdanie z części podstawowej może mieć dwie formy (do wyboru):

1. Plik tekstowy z linkiem do utworzonego i wykorzystywanego przy wykonywaniu zadania repozytorium na GitHub. Na tym repozytorium powinien być utworzony pliki *README.md* z sprawozdanie z wykonania zadania.
2. Klasyczny plik dot/doc/pdf z opisem przygotowanego rozwiązania oraz linkiem do utworzonego i wykorzystywanego przy wykonywaniu zadania repozytorium na GitHub.

W każdym wypadku sprawozdanie należy wgrać do przygotowanego katalogu na moodle.

Zawartość sprawozdania:

Należy umieścić dowód poprawnego wdrożenia aplikacji w postaci:

1. Zrzut ekranu z oknem aplikacji zawierającym interfejs kalkulatora. Na zrzucie powinien być widoczny adres, pod którym wyświetlono tą stronę.
2. Analogicznie jak w punkcie 1, zrzut ekranu z oknem aplikacji zawierającym dane studenta i adresem, pod którym wyświetlono tą stronę.
3. Wynik działania polecenia:

```
aws elasticbeanstalk describe-environments --environment-names \
multifib-env --output json
```

UWAGA1: Proszę zmienić zaznaczoną na czerwono nazwę na nazwę własnego środowiska, utworzonego w usłudze Elastic Beanstalk. Przykład działania polecenia jest zaprezentowany w instrukcji do laboratorium 12, strona 30.

UWAGA2: Powyższe polecenie można uruchomić w uprzednio zainstalowanym i skonfigurowanym środowisku AWS CLI lub (dla studentów ceniących czas i prostotę) w usłudze Cloud Shell – usługa wspomniana w instrukcji do laboratorium 12, strona 27.

Maksymalna ilość punktów z części obowiązkowej – 100%

Oceny częściowe:

p.1 – 30%

p.2 – 40%

p.3 – 30%

Dodatkowo, osoba, która w ramach punktu p.3 użyje AWS CLI otrzyma **ekstra 20% punktów**.

CZĘŚĆ NIEOBOWIĄZKOWA

Część ta zawiera dwa tematy. Można zrealizować jeden (wybrany) temat lub oba. W tym drugim przypadku punkty sumują się.

Zadanie nieobowiązkowe 1

To zadanie obejmuje modyfikację kodu. Jeżeli student zdecyduje się na zmiany (albo całkiem nowe rozwiązanie) to zostanie to ocenione (niestety, subiektywnie i w odniesieniu do

wszystkich zaproponowanych rozwiązań) w skali od 0% do 100%. Warunki, jakie należy spełnić by otrzymać punkty „ekstra” to:

- należy przygotować oddzielny opis (w postaci pliku: [README1.md](#) lub [README1.pdf](#)), w którym krótko zostaną opisane zmiany lub przedstawiona koncepcja nowej aplikacji,
- należy zapewnić dostępność wszystkich obrazów na dowolnym repozytorium publicznym, tak bym mógł uruchomić przygotowany docker-compose i zapoznać się z funkcjonalnością zaproponowanej aplikacji. UWAGA: Obrazy mogą być dla architektury arm64 (M1) lub amd64 (Intel),
- funkcjonalność aplikacji musi być zbieżna z funkcjonalnością aplikacji wymaganej w części podstawowej i MUSI zawierać przynajmniej jedną (dowolną) bazę danych.

UWAGA: W przypadku realizacji zmodyfikowanej nowej aplikacji **PROSZĘ KONIECZNIE** wdrożyć ją zgodnie z wymogami części podstawowej (inaczej: wykorzystać do realizacji części obowiązkowej). W ten sposób ta aplikacja będzie podstawą do przyznania punktów tak z części obowiązkowej jak i z tej części czyli zadania dodatkowego 1.

Maksymalna ilość punktów z nieobowiązkowego zadania nr 1 – 100%

Zadanie nieobowiązkowe 2

Zadanie składa się z dwóch części.

- **Pierwsza część** polega na wykorzystaniu poleceń AWS CLI do przedstawienia konfiguracji poszczególnych usług w chmurze AWS, z których korzysta wykonane wdrożenie. Przykładowo mogą to być EC2, S3, VPC itd. Podobne polecenia odnoszące się do Elastic Beanstalk jest wymagane w części obowiązkowej. W ramach sprawozdania z tej części należy przygotować oddzielny opis (w postaci pliku: [README2A.md](#) lub [README2A.pdf](#)) w którym umieszczone zostaną zrzuty ekranowe kolejnych poleceń wraz z efektem ich działania.

UWAGA: Polecenia muszą odnosić się maksymalnie po jednym dla danej usługi.

Maksymalna ilość punktów z tej części nieobowiązkowego zadania nr 2 – ilość poprawnie użytych poleceń x 10%

- **Druga część** dotyczy zagadnienia tagowania obrazów i wersji aplikacji w przygotowanym wdrożeniu. W części obowiązkowej nie wymagane jest definiowanie żadnego mechanizmu, który mógłby całościowo (tzn. w odniesieniu do kolejnych commit-ów, powiązanych tag-ów obrazów jak i wynikających z tego kolejnej wersji wdrażanej aplikacji) zorganizować proces kontroli nad tagowaniem. Pytanie: czy taki mechanizm istnieje a jeśli tak, to jak on funkcjonuje? W sprawozdaniu z tej części należy przygotować oddzielny opis (w postaci pliku: [README2B.md](#) lub [README2B.pdf](#)) w którym krótko należy przedstawić ideę/schemat możliwego rozwiązania.

UWAGA: Dla uzyskania maksymalnej ilości punktów należy podać odwołania (linki) do wykorzystanych źródeł informacji ale nie jest konieczna implementacja danego mechanizmu w opracowanym wdrożeniu.

Maksymalna ilość punktów z tej części nieobowiązkowego zadania nr 2 – 60%

UWAGI KOŃCOWE:

1. Przed rozpoczęciem realizacji zadania 2, **PROSZĘ KONIECZNIE** zdefiniować budżet typu „zero wydatków” zgodnie z informacjami przedstawionymi na laboratorium oraz w pliku **aws_monitoring_kosztów.pdf**, dostępnym w katalogu do laboratorium 12.
2. Po zakończeniu prac nad zadaniem 2 **PROSZĘ KONIECZNIE** usunąć (wyłączyć) wszystkie wykorzystywane usługi. Ocena wszystkich zadań (tak części obowiązkowej jak i części nieobowiązkowych) **NIE WYMAGA** działania wdrożonej aplikacji w chmurze AWS.
3. Oceniający zastrzega sobie prawo do podwyższenia oceny jeśli dane rozwiązanie chwyci go za serce albo umysł.