

LABORATORIUM

Programowanie Full-Stack w Chmurze Obliczeniowej

ZADANIE nr 1

CZĘŚĆ OBOWIĄZKOWA

Całość zadania ma być zrealizowana w przestrzeni nazw (namespace) o nazwie *zad1*. Na wstępie należy utworzyć manifest (plik yaml) deklarujący przestrzeń nazw *zad1*. Następnie uruchomić ten obiekt (tą przestrzeń nazw).

Następnie należy utworzyć zestaw plików manifestów (plików yaml) opisujących obiekty środowiska Kubernetes zgodnie z poniższymi założeniami:

1. Utworzyć plik yaml tworzący dla przestrzeni nazw *zad1* zestaw ograniczeń na zasoby (quota) o następujących parametrach:
 - maksymalna liczba Pod-ów: 10
 - dostępne zasoby CPU: 2 CPU (2000m)
 - dostępna ilość pamięci RAM: 1,5Gi
2. Utworzyć plik yaml tworzący Pod-a w przestrzeni nazw *zad1* o nazwie *worker*. Pod ma bazować na obrazie *nginx* i mieć następujące ograniczenia na wykorzystywane zasoby:

```
limits:
  memory: 200Mi
  cpu: 200m
requests:
  memory: 100Mi
  cpu: 100m
```

3. Bazując na przykładzie [application/php-apache.yaml](https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale-walkthrough/) i/lub z dokumentacji Kubernetes: <https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale-walkthrough/> należy zmodyfikować wskazany wyżej plik yaml tak by obiekty Deployment i Service utworzone zostały w przestrzeni nazw *zad1*. Jednocześnie obiekt Deployment ma mieć następujące ograniczenia na wykorzystywane zasoby:

```
limits:
  memory: 250Mi
  cpu: 250m
requests:
  memory: 150Mi
  cpu: 150m
```

4. Należy utworzyć plik yaml definiujący obiekt HorizontalPodAutoscaler, który pozwoli na autoskalowanie wdrożenia (Deployment) *php-apache* z zastosowaniem następujących parametrów:

```
minReplicas: 1
maxReplicas: ???????
```

targetCPUUtilizationPercentage: 50

Wartość `maxReplicas` należy określić samodzielnie, tak by **nie przekroczyć parametrów `quoty`** dla przestrzeni nazw `zad5`.

W sprawozdaniu należy **UZASADNIĆ PRZYJĘTY DOBÓR WARTOŚCI**.

5. Należy utworzyć obiekty zadeklarowane w opracowanych plikach `yaml`. Następnie potwierdzić ich poprawne uruchomienie za pomocą samodzielnie dobranego polecenia (poleceń).
6. Ponownie, bazując na przykładach z instrukcji do lab5 i/lub linku podanego w punkcie 3, należy uruchomić aplikację generującą obciążenie dla aplikacji `php-apache` i tym samym inicjalizując proces autoskalowania wdrożenia tej aplikacji. Za pomocą samodzielnie dobranych poleceń i wyniku ich działania proszę **potwierdzić dobór parametrów** z punktu 4.

ZADANIE nr 1

CZĘŚĆ NIEOBOWIĄZKOWA

(Ta część jest punktowana oddzielnie – warunkiem otrzymania dodatkowych punktów jest oczywiście wykonanie części obowiązkowej)

Należy odpowiedzieć na następujące pytania:

1. Czy możliwe jest dokonanie aktualizacji aplikacji (np. wersji obrazu kontenera) gdy aplikacja jest pod kontrolą autoskalera HPA ? Proszę do odpowiedzi (TAK lub NIE) dodać link do fragmentu dokumentacji, w którym jest rozstrzygnięta ta kwestia.
2. Jeśli odpowiedź na poprzednie pytanie jest pozytywna to proszę podać przykładowe parametry strategii `rollingUpdate`, które zagwarantują, że:
 - a) Podczas aktualizacji zawsze będą aktywne 2 pod-y realizujące działanie przykładowej aplikacji oraz
 - b) Nie zostaną przekroczone parametry wcześniej zdefiniowanej `quoty` dla przestrzeni `zad1`.
 - c) Jeśli należy skorelować (zmieć) ustawienia autoskalera HPA z części obowiązkowej w związku z zaplanowaną strategią aktualizacji to należy również przedstawić te zmiany.

Odpowiedź na pytanie 2 oraz dobór parametrów należy **krótko uzasadnić**.

Przydatny link podczas rozwiązywania zadania:

Poza przytoczonym wyżej linkiem warto również w dokumentacji referencyjnej (<https://kubernetes.io/docs>) zapoznać się z treścią sekcji: **Configure Memory and CPU Quotas for a Namespace**

Sprawozdanie (z części obowiązkowej oraz ewentualnie z części nieobowiązkowej) należy przesłać na moodle w postaci pliku `pdf` lub (preferowane) linku do repozytorium `git` z kodem pliku (-ów) `.yaml` i plikiem `Readme.md` opisującym rozwiązanie.