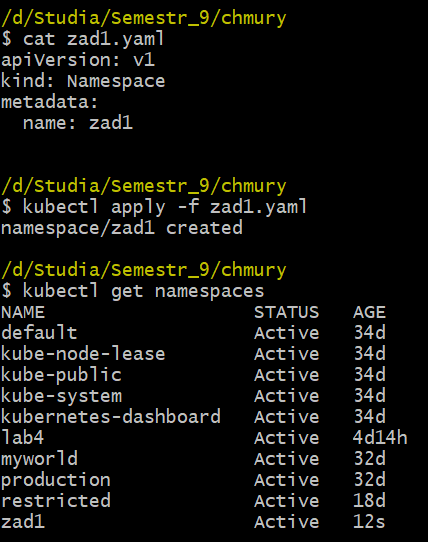
Sprawozdanie 1

Dawid Grabek

# Wstęp

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:



Rysunek 1. Tworzenie przestrzeni nazw **zad1** do sprawozdania

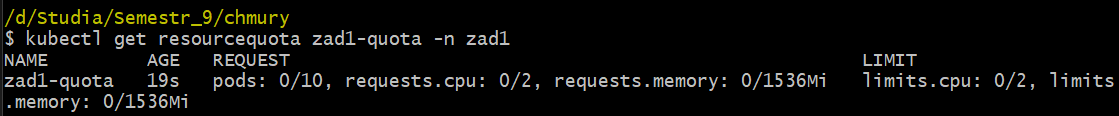
# Zadanie 1

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

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 2 Stworzenie ograniczeń zasobów



Rysunek 3 Weryfikacja zadania

# Zadanie 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

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie**

Rysunek 4 Stworzenie pod'a worker

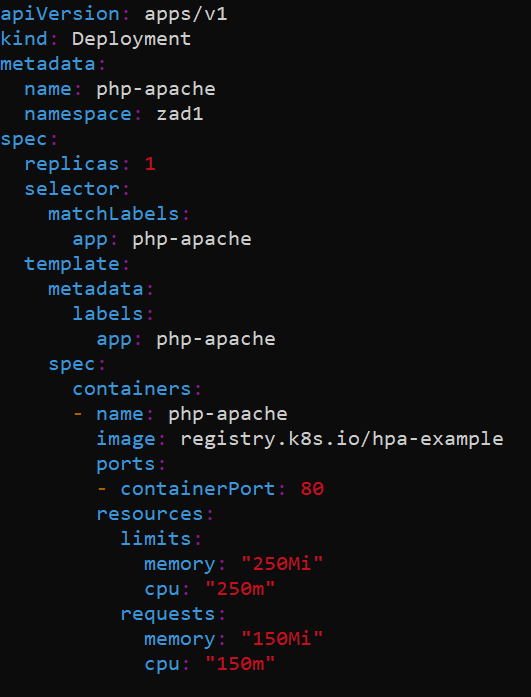
***Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie***

Rysunek 5 Weryfikacja zadania

# Zadanie 3

*Bazując na przykładzie application/php-apache.yaml i/lub z dokumentacji Kubernetes: https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscalewalkthrough/ 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*



Rysunek 6 deployment.yaml

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 7 service.yaml

# Zadanie 4

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

Obraz zawierający tekst, zrzut ekranu, Czcionka

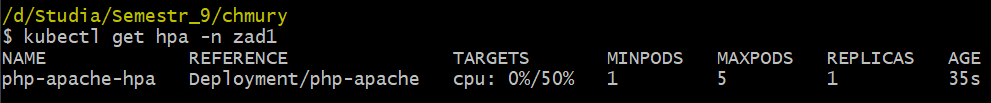
Opis wygenerowany automatycznie

Rysunek 8 hpa.yaml

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Rysunek 9 Uruchomienie hpa.yaml



Rysunek 10 Weryfikacja zadania

## Uzasadnienie:

W zadaniu 4 ustaliliłem wartość maxReplicas na 5. Uzasadnienie wynika z ograniczeń nałożonych przez Resource Quota w przestrzeni nazw zad1, które definiują

* Maksymalna liczba podów: **10**
* Dostępne zasoby CPU: **2000m (2 CPU)**
* Dostępna ilość pamięci RAM: **1.5Gi**

**Obliczenia:**

1. Każdy pod w Deployment ma przypisane:
   1. CPU
      1. limits.cpu = 250m
      2. requests.cpu = 150m
   2. RAM:
      1. limits.memory = 250Mi
      2. requests.memory = 150Mi.
2. Zasoby wymagane dla 1 poda:
   1. CPU = 250m
   2. RAM = 250Mi
3. Maksymalna liczba podów ograniczona zasobami:
   1. CPU: *2000m / 250m=8*
   2. RAM: *1536Mi / 250Mi=6.144*
4. Najmniejszy wynik (CPU vs RAM) decyduje: maksymalnie **6 podów**.
5. Biorąc pod uwagę, że w namespace może znajdować się również pod worker, który zajmuje:
   1. CPU: 200m
   2. RAM: 200Mi
6. Ograniczenia zasobów dla pozostałych podów:
   1. CPU: *2000m−200m=1800m*
   2. RAM: *1536Mi−200Mi=1336Mi*
7. Po podzieleniu zasobów na pod:
   1. CPU: *1800m / 250m=7.2*
   2. RAM: *1336Mi / 250Mi=5.344*

Finalna liczba replik: **5** (dla zachowania limitów RAM).

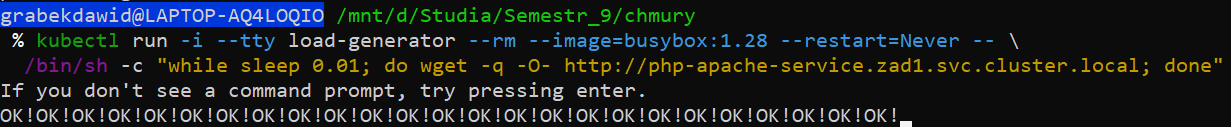
# Zadanie 5

*Należy utworzyć obiekty zdeklarowane w opracowanych plikach yaml. Następnie potwierdzić ich poprawne uruchomienie za pomocą samodzielnie dobranego polecenia (poleceń).*

**W poprzednich zadaniach utworzyłem już obiekty oraz wtedy sprawdziłem poprawność.**

# Zadanie 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 inicjalizujące proces autoskalowania wdrożenia tej aplikacji. Za pomocą samodzielnie dobranych poleceń i wyniku ich działania proszę potwierdzić dobór parametrów z punktu 4.*



Rysunek 11 Uruchomienie przeciążenia

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Rysunek 12 Weryfikacja autoskalowania

Zgodnie z wynikami, HorizontalPodAutoscaler (HPA) działa poprawnie. Oto, co się dzieje:

**Obciążenie CPU**: Widać, że CPU w podach przekracza próg 50%, co skutkuje skalowaniem aplikacji.

Początkowo liczba replik to 1.

Gdy obciążenie CPU wzrastało do 75%, HPA zwiększyło liczbę replik do 2, a później do 4.

**Skalowanie**: HPA poprawnie reaguje na wzrost obciążenia CPU i skaluje liczbę podów, nie przekraczając ustawionego limitu maxReplicas (5).

**Podsumowanie**

* HPA działa poprawnie, a liczba replik dynamicznie rośnie w odpowiedzi na obciążenie CPU.
* Wszystko jest skonfigurowane zgodnie z oczekiwaniami, a mechanizm autoskalowania działa efektywnie.