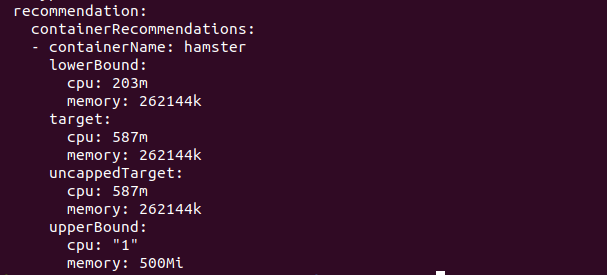
1. 저번 미팅 내용

* 우리가 진행하던 HPA는 여유 리소스가 있을때만 동작하기 때문에 주제의 실현가능성이 매우 낮아졌다.
* VPA를 조사해서 다른 돌파구가 있는지 찾아보자
* VPA 로드를 더 주었을 때 가상 환경에서 리소스를 더 할당 받을 수 있는지 확인할 수 있을지 여부 확인

1. VPA 조사

* VPA는 Auto 모드와 Manual모드가 있고, 이는 VPA의 yaml 파일에서 updatePolicy의 updateMode에서 지정할 수 있다.
* Pod resource의 request값을 변경해주는 방식이기 때문에 원치않는 pod 재실행이 될 수 있다.
* OOM 이벤트를 감시할 수 있음
* VPA와 HPA는 양립할 수 없다.
* manual모드로 운영하면 적절한 request양을 추천해준다.



* target 이 추천 resource 양이다.
* 기본 예제로 테스트한 결과 cpu request양이 200 -> 587로 변했다.

1. 개선점 및 의문점

* 하나의 서비스에 HPA와 VPA를 타겟 리소스를 다르게 해서 둘 다 적용시키면 어떻게 될까? (에러 발생 확률 농후)
* VPA 실험해봤는데 v1beta2버전부터 매뉴얼 모드가 지원됨, 하지만 내가 그걸 하려고 하면 validation 오류가 뜸.
* VPA가 널리 사용되는 기능인 느낌보다는 개발중인 것 같은 미완성 기능 같았다.
* 로드를 없애고 5분이지나도 원래 할당량으로 돌아오지 않음
* 이유를 알아보려고 자잘한 수정을 반복하며 테스트 했지만 모르겠음
* beta버전에 쓰이는 예시말고 hpa 테스트 때 사용한 php-apache로 VPA를 사용하려 하면 update가 자동으로 진행되지 않음 (1)
* beta버전도 초기 replica를 1로 바꾸면 update가 진행되지 않음 내가 이상한건지 아직 미완성이라 호환성문제가 있는건지 구분이 안감 (2)
* (1) (2) 원인 해결 : replica를 2개 이상으로 설정해줘야함 VPA는 파드의 리소스를 수정할 때 아예 재시작하므로 서비스가 유지될 수 있게 최소한 2개의 pod가 필요함

1. 생각

* Long-lived-service에 대부분의 리소스를 할당하고 short-lived-service에 갑작스런 대규모 로드가 들어왔다는 가정하에 VPA를 돌려서 추천 request 값으로 바꿔서 가지고 있던 리소스의 일부분을 돌려받는다.
* 돌려받아 생긴 여유 리소스는 short-lived-service의 hpa로 리소스할당을 진행한다. 이게 첫번째 실험 계획.
* 잘된다면 마지막에 보고서에 제안으로 “현재는 VPA와 HPA가 호환이 안되지만 만약 가능하게 되면 ㅁㅁ 방식으로 개선할 수 있을 것이다.”로 마무리할 수 있을 것 같다.
* 이 실험이 진행이 안되거나 교수님 판단하에 의미 없는 실험 같으면 다른 방법을 생각해봐야 할 듯