

6.6 퍼시스턴트 볼륨의 동적 프로비저닝

🕒 생성일	@2021년 4월 18일 오전 12:46
🏷️ 태그	

여전히 클러스터 관리자는 실제 스토리지를 미리 프로비저닝 해줘야함..

But .. k8s는 pv의 **동적 프로비저닝** 을 통해 자동으로 이 작업을 수행할 수 있음

클러스터 관리자가 퍼시스턴트볼륨을 생성하는 대신 **퍼시스턴트볼륨 프로비저너** 를 배포하고 사용자가 선택 가능한 퍼시스턴트볼륨의 타입을 하나 이상의 **스토리지클래스 오브젝트**로 정의할 수 있음

[장점] : 누군가 pvc를 통해 요청시 새로 pv를 생성 → pv 볼륨이 부족할 일이 없다...

1. 스토리지클래스 리소스를 통해 사용가능한 스토리지 유형 정의하기

사용자가 퍼시스턴트 볼륨 클레임을 생성 → pv이 프로비저닝됨 → 하나 이상의 스토리지클래스 리소스를 생성해야함

storageclass-fast-gcepd.yaml 참조

`provisioner: kubernetes.io/gce-pd` pv 프로비저닝을 위해 사용되는 볼륨 플러그인을 추가

2. 퍼시스턴트 볼륨클레임에서 스토리지 클래스 요청하기

특정 스토리지클래스를 요청하는 pvc wjddml todtjdgkrl

Mongodb-pvc를 동적 프로비저닝을 사용하도록 수정

mongodb-pvc-dp.yaml 참조

`spec: storageClassName: fast` 수정

클레임 생성시 fast 스토리지클래스 리소스에 참조된 프로비저너가 퍼시스턴트 볼륨을 생성함

동적 프로비저닝된 PV와 생성된 PV 검사하기

`kubectl get pvc mongodb-pvc`

하면,, volume 열에 클레임에 바인딩된 퍼시스턴트 볼륨을 표시함..

`kubectl get pv` 하면 동적으로 프로비저닝된 퍼시스턴트볼륨을 볼 수 있다

→ pv 외에 프로비저너는 실제 스토리지도 프로비저닝했다

`gcloud compute disk list` 를 하면 첫번째 퍼시스턴트 디스크의 이름은 동적으로 프로비저닝됐음을 나타내고.. 유형은 이전에 스토리지 클래스에서 지정한 대로 SSD임

3. 스토리지 클래스를 지정하지 않은 동적 프로비저닝

pv를 파드에 연결하는 가장 간단한 방법을 살펴보자

스토리지 클래스 조회하기

fast라는 사용자 정의 스토리지 클래스를 생성할 때 클러스터에 스토리지 클래스가 이미 정의돼 있는지 확인하지 않았지만.. 지금 확인 → `kubectl get sc`

- fast 스토리지 클래스 외에 standard 스토리지 클래스가 default로 표시된것을 알 수 있음

`kubectl get sc` 를 해보면 여기서도 default가 조회된다..

기본 스토리지 클래스 확인하기

`kubectl get` 을 사용해 GKE 클러스터의 standard 스토리지 클래스를 자세히 살펴보자

`kubectl get sc standard -o yaml`

- 기본 스토리지 클래스는 pvc에서 명시적으로 어떤 스토리지 클래스를 사용할지 지정하지 않은 경우 pv를 동적 프로비저닝하는 데 사용됨

스토리지 클래스를 지정하지 않고 퍼시스턴트볼륨클레임 생성하기

storageClassName 속성을 지정하지 않고 pvc생성시 GKE에서 pd-standard 유형의 GCE 퍼시스턴트 디스크가 프로비저닝된다.

- mongodb-pvc-dp-nostorageclass.yaml을 확인해보면
- storageClassName 속성을 지정하지 않음

퍼시스턴트볼륨클레임을 미리 프로비저닝된 퍼시스턴트볼륨으로 바인딩 강제화하기

storageClassName을 빈 문자열로 지정하지 않으면 미리 프로비저닝된 pv가 있어도 동적 볼륨 프로비저너는 새 pv를 프로비저닝 함