

# 쿠버네티스를 위한 볼륨 프로비저닝 기법에 관한 연구

조혜영°, 정기문, 박준영, 손아영

한국과학기술정보연구원 슈퍼컴퓨팅본부

{chohy°, kmjeong, jypark, ayson}@kisti.re.kr

## A Study of Kubernetes Volume Provisioning Methods

Hyeyoung Cho°, Gi-Mun Jeong, Junyoung Park, A-young Son

Division of National Supercomputing

Korea Institute of Science and Technology Information

### 요약

최근 클라우드 네이티브 기술과 쿠버네티스의 활용이 애플리케이션 개발을 현대화하고 있다. 최근 마이크로 서비스와 컨테이너화 기술의 중요성이 강조 되고 있다. 특히, 쿠버네티스 환경에서의 Volume Provisioning은 데이터 관리와 시스템 안정성 강화를 위한 핵심적인 요소로 인식되고 있다. 본 논문에서는 쿠버네티스에 스토지지를 관리하는 볼륨 프로비저닝 기법에 대해서 분석하였다. 이를 통해 데이터 관리와 시스템의 안전성을 향상시키는 방안에 대한 분석을 제시한다.

### 1. 서론

최근 클라우드 네이티브 기술은 변화하는 고객의 요구에 부응하기 위해 애플리케이션 개발을 현대화에 하는데 쿠버네티스가 적극 활용되고 있다. 마이크로 서비스와 컨테이너를 통해 애플리케이션은 지곳적인 시장 변화에 유연하고 탄력적으로 대응할 수 있게 발전하고 있다.

이러한 쿠버네티스 기술에서 Volume Provisioning은 분산 시스템에서 데이터 지속성과 확장성을 강화하는 핵심적인 요소로 인식되고 있다 [1,2]. Volume Provisioning은 쿠버네티스 환경에서 데이터 관리와 시스템 안정성을 향상시키는 핵심적인 역할을 수행한다[3] 본 논문에서는 쿠버네티스 볼륨 프로비저닝 기법에 대한 기술 동향을 분석하고, 이를 통해 데이터 관리와 시스템의 안전성을 증진시키는 요인을 분석하였다.

본 논문의 구성은 다음과 같다. 2장에서는 쿠버네티스 볼륨 프로비저닝에 대해 설명하고, 3장에서는 쿠버네티스 볼륨 프로비저닝에 대해 소개하고, 4장에서는 Dynamic Provisioning에 대해 설명한다. 마지막으로 5장에서는 결론을 기술한다.

### 2. 쿠버네티스 볼륨 프로비저닝

쿠버네티스에서 볼륨 프로비저닝(Volume Provisioning)은 분산 시스템에서 데이터 지속성을 보장하고 확장성을 강화하는데 중요한 역할을 한다 [4]. 이는 다음과 같은 이점을 제공한다. 첫째, 동적 Volume Provisioning은 필요에 따라 스토리지를 동적으로 할당함으로써 자원 사용 효율성을 극대화한다. 둘째, 스토리지 리소스의 효율적인 관리를 통해 비용을 절감하고 클러스터의 성능을 향상시킨다. 셋째, 프로비저닝 정책을 통해 다양한 애플리케이션의 요구 사항에 맞게 스토리지를 최적화할 수 있어, 시스템의 유연성과 확장성을 높인다. 이러한 이유로, Volume Provisioning은 K8s 환경에서의 데이터 관리와 시스템 안정성을 증진시키는 핵심적인 요소로 인식되고 있다.

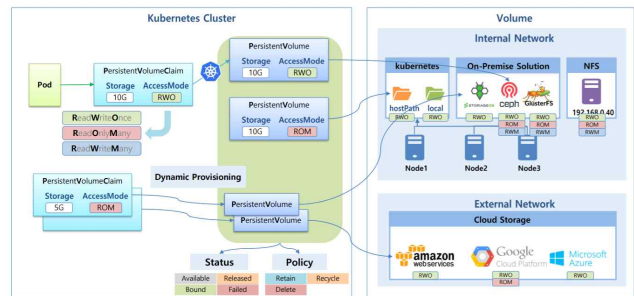


그림 1 Kubernetes Volume Provisioning

쿠버네티스에서 Volume은 데이터를 안정적으로 유지하기 위해서 사용한다. 이를 실제 데이터는 쿠버네티스 Cluster와 분리되어 관리된다. 그리고 이런 방식으로 관리될 수 있는 Volume의 종류들이 많은데, 크게 내부망에서 관리하는 경우와 외부망에서 관리하는 경우 2가지로 분류할 수 있다. 내부망에는 쿠버네티스를 구성하는 Node들이 있는데 기본적으로 쿠버네티스에서 이 Node들의 실제 물리적인 공간에 데이터를 만들 수 있는 hostPath나 localVolume이 있고 별도의 On-Premise Storage Solution들을 Node들에 설치할 수 있다. 또한 NFS를 사용해서 다른 서버를 Volume 자원으로 사용할 수 있다. 외부망에는 Google Colud, AWS, Azure 같은 클라우드 Storage를 두고 여기에 쿠버네티스 Cluster와 연결하여 사용할 수 있다. 쿠버네티스 Cluster 밖에 실제 Volume들이 마련되어 있다면 관리자는 PV를 만드는데 저장 용량과 AccessMode를 정하고 Volume을 선택해서 연결한다. 사용자는 원하는 용량과 AccessMode로 PVC를 만들면 쿠버네티스가 알아서 적절한 PV와 연결을 해준다. 그리고 이 PVC를 Pod가 사용하게 된다.

## AccessMode

1. ReadWriteOnce : 한 Node에서 읽기 쓰기
2. ReadOnlyMany : 여러 Node에서 읽기
3. ReadWriteMany : 여러 Node에서 읽기 쓰기

하지만 쿠버네티스에서 이렇게 AccessMode가 있다고 해서 실제 Volume들도 모두 이와 같이 지원되는 것이 아니고 각각의 Volume들마다 지원되는 AccessMode가 다르다. 하지만 이런식으로 사용한다면 Volume을 사용할 때마다 PV를 만들어줘야되고 또 원하는 PV와 연결을 할려고 Storage와 AccessMode를 확인해서 맞춰야하고 너무 복잡해진다. 그래서 쿠버네티스는 Dynamic Provisioning이란 것을 지원해준다.

## 3. Dynamic Provisioning

Dynamic Provisioning은 사용자가 PVC를 만들면 알아서 PV를 만들어주고 실제로 Volume과 연결해준다. 모든 PV에는 각각의 상태들이 존재한다. (Available, Released, Bound, Failed). Storage Solution을 선택해서 설치해 해야 한다.

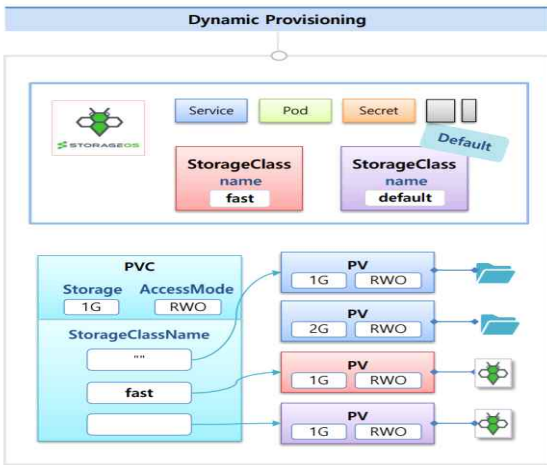


그림 2. Dynamic Provisioning

StorageClass 사용해서 동적으로 PV를 만들 수 있는데 PVC를 만들 때 StorageClassName 이라는 부분이 있다.

- StorageClassName: ""이면 적절한 PVC를 찾아준다.
- StorageClassName: 만들어져 있는 StorageClassName 이름으로 자동으로 StorageOs Volume을 가진 PV가 만들어진다.

StorageClass는 추가로 만들 수 있고 default라는 걸 설정할 수 있다. 이렇게 만들어 놓으면 PVC에 StorageClass 이름을 생략을 했을 때 default StorageClass가 적용이 되서 PV가 만들어진다. Status는 최초 PV가 만들어졌을 때 Available 상태다. PVC와 만나게 되면 Bound 상태다. PV를 직접 만드는 경우에는 아직 Volume에 실제 데이터가 만들어져 있는 상태가 아니고 Pod가 PVC를 사용해서 구동이 될 때 실제 Volume이 만들어진다. Pod에 서비스가 유지가 되다가 Pod가 삭제될 경우에는 PVC와 PV에는 아무런 변화가 없기 때문에 데이터에는 문제가 없다. PVC를 삭제해야지만 PV와 연결이 끊어지면서 PV의 상태는 Released가 된다. PV와 실제 데이터간에 문제가 생기면 PV의 상태는 Failed 상태가 된다.

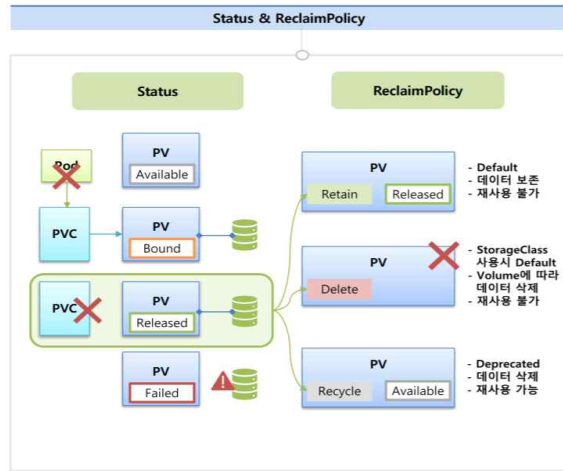


그림 3. Status & Reclaim Policy

PVC가 삭제가 되었을 때 상황에 대해서 PV의 ReclaimPolicy에 따라서 PV에 대한 상태가 달라진다. ReclaimPolicy에는 Retain, Delete, Recycle이 있다.

1. Retain: PVC가 삭제가 되면 PV의 상태가 Released가 됨. Default 실제 Volume의 데이터 보존됨. 재사용 불가 (PV를 다른 PVC에 연결할 수 없음)
2. Delete: PVC가 삭제가 되면 PV도 같이 지워짐. StorageClass 사용시 Default Volume에 따라 데이터 삭제되기도 하고 안되기도 함. 재사용 불가
3. Recycle: PVC가 삭제가 되면 PV의 상태가 Available이 Deprecated 데이터 삭제, 재사용 가능

## 5. 결론

본 연구는 클라우드 네이티브 기술의 발전과 쿠버네티스의 활용이 애플리케이션 개발을 현대화하고 있음을 확인하였다. 특히, 마이크로 서비스와 컨테이너화 기술을 통해 애플리케이션은 시장 변화에 더욱 유연하고 탄력적으로 대응할 수 있게 되었다. 본 연구에서는 이러한 기술적 발전과 함께 쿠버네티스 환경에서의 Volume Provisioning의 중요성을 강조하였다. 더불어, 쿠버네티스 볼륨 프로비저닝 기술의 동향과 데이터 관리, 시스템 안전성 강화에 대한 요인을 분석하였다. 이를 통해 쿠버네티스 기반 시스템에서의 데이터 관리 및 시스템 안정성을 높일 수 있는 방안에 대해 논의하였다. 특히, 쿠버네티스 환경에서 Volume Provisioning의 중요성을 강조하며, 이를 통해 데이터 관리와 시스템의 안전성을 향상시키는 방안에 대한 분석을 제시한다.

## ACKNOWLEDGMENTS

본 논문은 한국과학기술정보연구원(KISTI)에서 2024년도 정부(과학기술성정통부)의 지원(K-24-L02-M01-C07, 초고성능컴퓨팅 공동 활용 기술개발)으로 수행된 연구임.

## 참고문헌

- [1] Arpaci-Dusseau, Remzi H., and Andrea C. Arpaci-Dusseau. "Operating Systems: Three Easy Pieces." Arpaci-Dusseau Books, 2019.
- [2] Burns, Brendan, et al. "Design patterns for container-based distributed systems." O'Reilly Media, Inc., 2017.
- [3] Hajare, Vaibhav, and Harmeet Singh. "Kubernetes Handbook: Non-Programmer's Guide to Deploy Applications." Independently Published, 2021.
- [4] Kubernetes homepages, <https://kubernetes.io/docs>