

쿠버네티스 가비지 수집 (Garbage Collection)

개념 요약

쿠버네티스의 가비지 수집(Garbage Collection)은

“더 이상 필요 없는 리소스를 자동으로 정리하는 시스템”

즉, 클러스터 안의 **찌꺼기 리소스**(죽은 파드, 사용 안 하는 이미지 등)를

주기적으로 찾아서 지워주는 청소기 역할

왜 필요한가?

- 쿠버네티스는 수많은 리소스(파드, 잡, 이미지, 볼륨)를 계속 생성/삭제해.
- 이 중 일부가 남아 있으면 **디스크 낭비, 리소스 누수, 성능 저하** 발생.
- 그래서 kubelet과 컨트롤러는 주기적으로 “정리 대상”을 찾아 없앤다.

| 🖱️ “자동 정리 시스템”이 없다면, 쿠버네티스는 점점 느려지고 불안정해진다.

어떤 걸 지우나?

가비지 수집은 다음 리소스들을 정리

- **종료된 Job**
 - 소유자 참조(Owner Reference)가 없는 오브젝트
 - **사용되지 않는 컨테이너/이미지**
 - 삭제 정책이 Delete인 PV
 - 만료된 CSR (CertificateSigningRequests)
 - 오래된 노드 정보
-

핵심 구조

Owner & Dependent

- 대부분의 쿠버네티스 오브젝트는 **소유자(owner)** 와 **종속(dependent)** 관계로 연결
- 예:
 - ReplicaSet → 여러 Pod
 - Service → EndpointSlice

즉, “누가 만들었는가” 관계가 있어야 함.

만약 **owner** 없는 오브젝트는 가비지로 간주되어 삭제

삭제 방식 (Cascading Deletion)

쿠버네티스가 오브젝트를 삭제할 때 **종속 리소스를 함께 삭제할지 여부**를 설정할 수 있다.

이걸 **캐스케이딩 삭제**라고 한다.

◆ 포그라운드(Foreround)

- 먼저 소유자 오브젝트를 “삭제 중 상태”로 표시
- 종속 오브젝트들을 **모두 삭제한 뒤**, 마지막에 소유자 삭제
- 즉, **순차적·안전한 삭제**
- `blockOwnerDeletion=true` 인 자식이 있으면 삭제 대기함

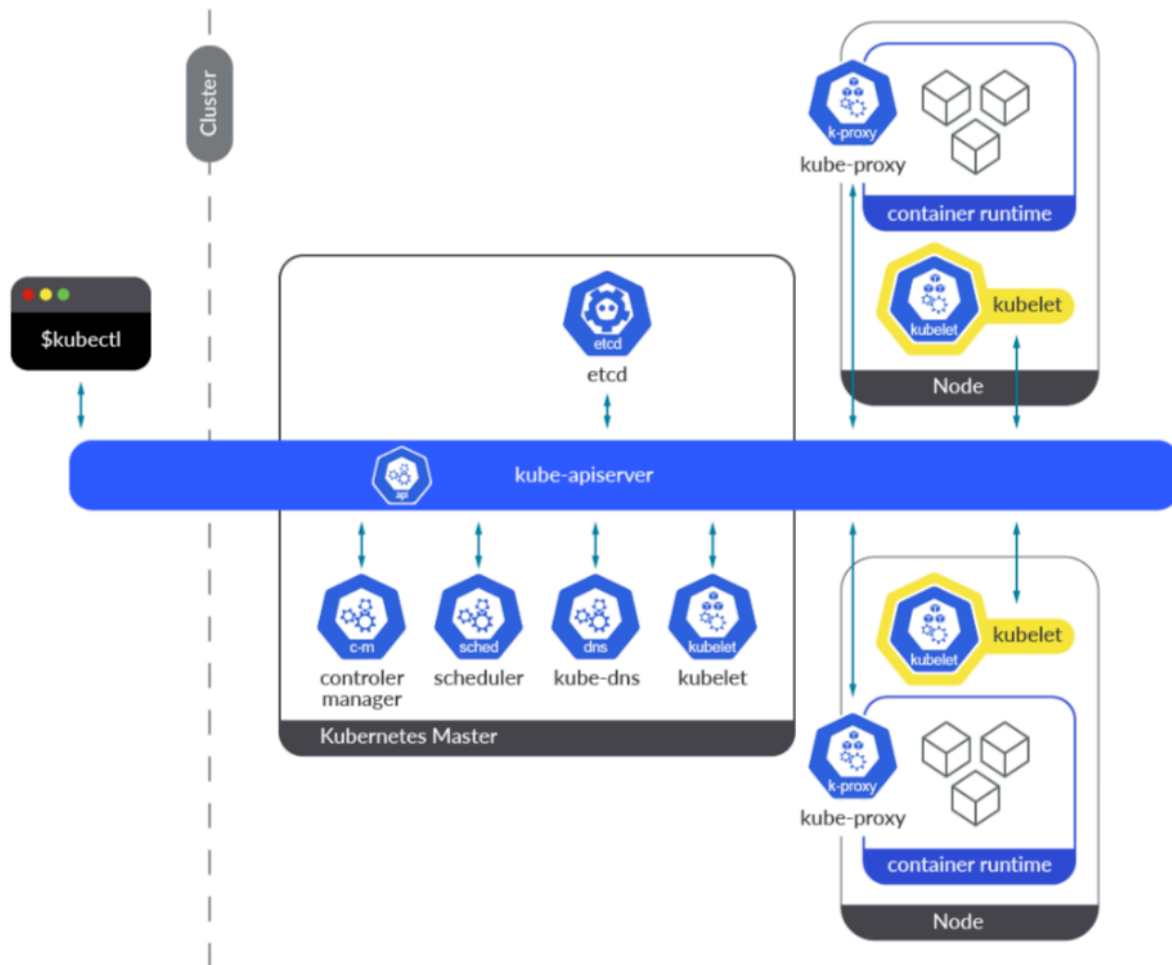
◆ 백그라운드(Background)

- 소유자 오브젝트를 **즉시 삭제**
- 이후 백그라운드에서 종속 오브젝트들 자동 정리
- 기본 모드

◆ Orphan (분리된 종속)

- 부모를 지워도 자식을 남기고 싶을 때
- `-cascade=orphan` 설정 시 사용

이미지·컨테이너 가비지 수집 (Kubelet 역할)



kubelet은 **노드 레벨**에서 불필요한 컨테이너와 이미지를 주기적으로 삭제

| 대상 | 주기 | 설명 |
|------|----|-----------------|
| 컨테이너 | 1분 | 사용되지 않는 컨테이너 정리 |
| 이미지 | 5분 | 안 쓰는 이미지 삭제 |

⚠ 외부에서 임의로 삭제 도구를 쓰면, kubelet의 관리 흐름을 깨트릴 수 있음.

디스크 사용량 기준

이미지 삭제는 디스크 용량을 기준으로 트리거

- **HighThresholdPercent** 초과 → 오래된 이미지부터 삭제 시작
 - **LowThresholdPercent** 도달할 때까지 삭제 진행
-

8 주요 파라미터 (kubelet 설정)

- **MinAge** : 삭제 대상이 되기 위한 최소 나이
- **MaxPerPodContainer** : 각 파드가 가질 수 있는 죽은 컨테이너 최대 개수
- **MaxContainers** : 전체 노드가 가질 수 있는 죽은 컨테이너 총합