PART 2

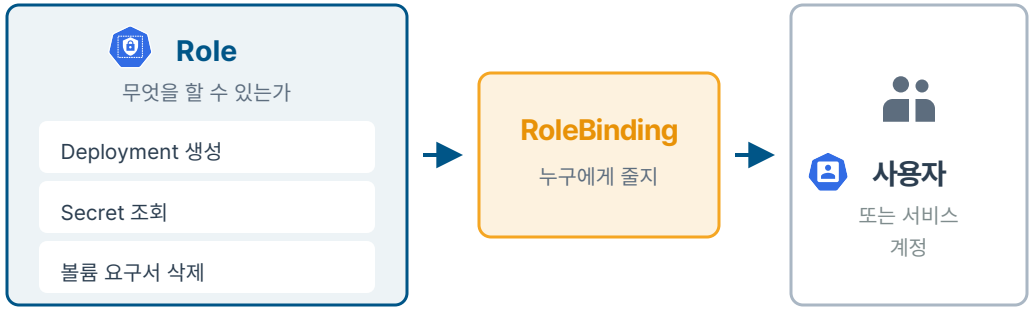
누가 무엇을 할 수 있는지 정하는 접근 제어

권한을 넘기고 그 사용을 기록으로 남기는 방법을 봄

클러스터 관리자가 손을 떼기 위해 넘기는 RBAC 권한

KEY INSIGHT 기본 관리자 역할은 무엇이든 가능 — 권한을 깎은 역할을 만들고 Namespace 관리 권한을 프로덕트팀에 넘겨야 책임이 실제로 이동

권한이 붙는 방식



역할의 유효 범위는 Namespace 하나 — 클러스터 전체에 걸려면 별도의 클러스터 범위 역할이 필요

권한을 넘기는 방향

클러스터 어드민팀은 클러스터 관리에만 머무르고 제품에 관한 판단은 넘김

권한 설계의 원칙

권한을 가진다는 것은 동시에 책임을 진다는 뜻

조직 차원의 책임 정의와 시스템 차원의 권한 설정은 함께 가야 함

한쪽만 하면 문서상의 책임과 실제 권한이 어긋남

전권 역할을 그대로 나눠 주면 통제 포기가 됨

함께 봐야 하는 다른 통제 · 현행화

RBAC 은 누가 무엇을 조작할 수 있는지를 통제

Pod 가 어떤 권한으로 실행되는지는 별도 통제의 몫

PodSecurityPolicy 폐기 후 Pod Security Admission 이 그 자리를 담당

두 통제를 하나로 착각하면 조작 권한만 막고 실행 권한이 열린 채 남음

Source: OMDM Blog · 쿠버네티스 구축부터 운영 완전 정복 — 10.4 RBAC · 10.3 서비스 어카운트

워크로드에 자동으로 붙는 서비스 계정 토큰

KEY INSIGHT 아무것도 지정하지 않아도 토큰이 주입됨 — 필요 없는 파드에서 끄지 않으면 침해 시 API 접근 경로가 됨

계정이 자동으로 생김

네임스페이스를 만들면 기본 서비스 계정이
함께 만들어짐

우리가 만들지 않아도 존재함

토큰이 파드에 주입됨

계정을 지정하지 않아도 기본 계정의 토큰이
파드 안 파일로 들어감

그 파드에 들어간 쪽은 API 를 호출할 수 있음

안 쓰면 끔

API 를 호출하지 않는 워크로드에서는 자동
주입을 꺼 둠

대부분의 애플리케이션이 여기에 해당함

쓰면 전용 계정을 만듦

API 가 필요한 워크로드에는 전용 계정을
만들고 최소 권한만 붙임

기본 계정에 권한을 붙이면 모든 파드가 갖게 됨

확인하는 방법

파드 안에서 토큰 파일이 있는지, 그 토큰으로 무엇을 조회할 수 있는지 실제로 시험해 봄

기본 계정이 아무 권한도 갖지 않는지 함께 확인함 — 편의를 위해 권한을 붙여 둔 클러스터가 종종 있음

Source: Sustainable Kubernetes, builderscon (2018)

만료되면 클러스터가 멈추는 인증서와 인증 방식

KEY INSIGHT kubeadm 클러스터의 인증서는 기본 1년 — 업그레이드를 하지 않으면 갱신도 되지 않으므로 만료일을 알림 대상에 넣지 않으면 어느 날 API 서버가 통째로 막힘

클러스터가 쓰는 인증서

기본 유효기간 1년

- API 서버·kubelet·etcd 가 서로를 확인하는 데 사용
- kubeadm 기준으로 대부분 1년짜리로 발급
- 업그레이드를 하면 그 과정에서 함께 갱신됨

만료되면 클러스터가 멈춤

1년 넘게 그대로 둔 클러스터가 위험함

사람이 붙는 방법

인증서 · OIDC · 토큰

- 기본 kubeconfig 는 클라이언트 인증서 방식
- 인증서는 폐기 목록이 없어 회수가 어려움
- 사람 계정은 OIDC 로 사내 계정과 묶는 편이 안전

떠난 사람의 접근을 끊을 수 있는가

인증서만 나눠 줬다면 회수가 사실상 안 됨

워크로드가 붙는 방법

서비스 계정 토큰

- 파드에 자동으로 주입되는 짧은 수명 토큰
- 대상과 만료가 지정되어 다른 곳에 못 씀
- 예전의 만료 없는 시크릿 토큰과는 다름

사람 자격 증명을 파드에 넣지 않기

워크로드는 서비스 계정으로만 붙게 함

운영 항목으로 올려 두어야 하는 것

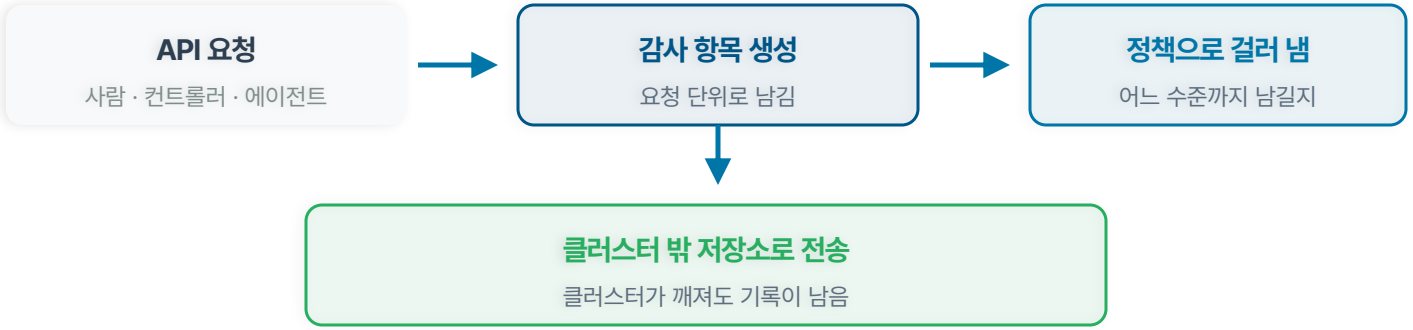
- 인증서 만료일을 알림 대상에 등록
- 사람 계정은 OIDC 로 사내 계정과 연동
- kubeconfig 를 공유하지 않기

Source: kubernetes.io — Certificate Management, Authenticating

누가 무엇을 했는지 남기는 감사 기록

KEY INSIGHT 모든 변경이 API 를 지나므로 여기서 전부 남음 — 기록을 켜지 않으면 사후 조사에서 쓸 근거 자체가 없음

기록이 만들어져 나가는 경로



⚠ 클러스터 안에만 두면 침해 시 기록도 함께 지워질 수 있음

기록 수준을 고름

전부 남기면 양이 감당되지 않고,
너무 줄이면 조사에 쓸 수 없음
자원 종류마다 다르게 정함

비밀값 접근은 상세히 남김

보관 기간을 정함

규정이 요구하는 기간과 실제
조사에 필요한 기간 중 긴 쪽을
기준으로 잡음

접근 권한도 함께 통제함

기록으로 답할 수 있어야 하는 질문

설정을 바꾼 주체와 시각 · 비밀값을 읽은 계정 · 실패한 권한 요청의 출처

이 셋에 답할 수 없음면 기록 수준이나 보관 기간을 다시 봄

사고가 난 뒤에는 설정을 바꿔도 과거 기록이 생기지 않음

Source: GKE security session, Google Cloud Kubernetes Day (2019)

무엇을 어디까지 남길지 리소스마다 다르게 정하는 감사 정책

KEY INSIGHT 전부 남기면 디스크가 차고 아무것도 안 남기면 사고 뒤에 볼 것이 없음 — 리소스별로 기록 수준을 달리 거는 것이 실무 기준

요청 하나에 매길 수 있는 네 가지 기록 수준

None	→	Metadata	→	Request	→	**RequestResponse**
남기지 않음		누가 무엇에 무엇을		보낸 내용까지		응답까지 전부

리소스마다 다르게 검

Secret 은 값이 남지 않게 Metadata 로,
RBAC 변경은 RequestResponse 로 남김
한 수준으로 통일하면 둘 중 하나가 틀림

볼륨을 먼저 재 봄

읽기 요청까지 남기면 양이 폭증함
조회성 동작은 대체로 제외하고
변경 동작 위주로 남김

클러스터 밖으로 뺌

노드 디스크에만 두면 그 노드가 죽을 때
함께 사라짐. 사고 조사에 쓰려면
밖으로 보내고 보관 기간을 정해 둠

Source: kubernetes.io — Auditing