

공공기관 클라우드 네이티브 기반 DR(재해복구) 구축 방안

무엇을, 왜 바뀌야 하는가 — 가상화 DR의 한계와 대안

정량(RTO·자원) · 정성(자동화·신뢰 부트) · 정책(전자정부법·ISMS-P) 3축 입증

PROJECT CODE
CN-DR-WP-2026

DATE
2026.06



PUBLIC

목차 — 6개 장 · 본문 22쪽

01

왜 전환인가

1장 · 외부 압력 · 선언적 재현 · 5차원 비교 · 복구 시간축

02

Active-Active 아키텍처

2장 · 4중 종속 해소 · 트래픽 분기 3계층 · Pod 부팅

03

이미지·레지스트리 동기화

3장 · SSoT · Cosign 서명 · 4 동기화 패턴 · 망분리 DMZ

04

데이터 동기화와 복구 절차

4장 · 4 카테고리 매트릭스 · 7단계 vs 3단계

05

도입 효과와 해외 사례

5장 · 자원·라이선스 30~60% 절감 · 글로벌 5개 사례

06

정책 적합성과 도입 권장

6장 · 5개 정책 문서 수렴 · 의사결정 매트릭스

KEY FOCUS

선언적 재현 모델 전환 → Active-Active 구조 → 데이터 동기화 신뢰성 → 비용·RTO 우위 → 정책 적합·도입 권장의 인과 사슬을 6개 장이 입증

KEY INSIGHT 독립 결론이 아닌 인과 사슬 — 모델 전환이 구조를, 구조가 데이터 선택을, 그 결합이 정량 우위를, 우위가 정책 적합성을 만든다

번호	핵심 메시지	본문 입증
M1	DR 모델이 시점 복원에서 선언된 상태의 재현으로 이동했다 백업 부팅이 아니라 매니페스트·이미지 다이제스트를 다른 클러스터에서 재현	1·6장
M2	쿠버네티스가 Active-Active 운영의 구조적 전제를 갖췄다 상태 비저장 워크로드 · 선언적 매니페스트 · Service Mesh 멀티클러스터 라우팅	2장
M3	데이터베이스 동기화는 워크로드별 4 카테고리 매트릭스로 선택한다 동기 복제 / CDC 비동기 / 분산 SQL / NoSQL multi-DC	4장
M4	자원·라이선스 약 30~60% 절감 + RTO 한 자릿수 분 단축 동일 워크로드 기준 — 정량 ROI 의 핵심 결론	5·6장
M5	국내 정책 적합성이 확보된다 전자정부법 제49조 · 행안부 지침 · KISA ISMS-P · DPG 로드맵 · NIA 가이드	9장

CHAPTER 01

가상화 DR 과 클라우드 네이티브 DR

모델 전환 — 시점 복원에서 선언된 상태의 재현으로

KEY INSIGHT

세 가지 압력이 함께 밀려온다 — 컨테이너 확산, 라이선스 비용 상승, 랜섬웨어 위협이 겹치면서 DR 전환은 선택이 아니라 필수가 된다



워크로드 컨테이너 이동

Container Adoption

- 새로 만드는 업무 시스템은 대부분 쿠버네티스로 간다
- DR만 가상화로 남겨두면 시스템을 두 벌 운영해야 한다



가상화 라이선스 상승

License Pressure

- 가상화 라이선스 비용이 해마다 오르고 있다
- 평소 꺼둔 대기 장비에도 똑같이 비용이 청구된다



사이버 재해 빈도 증가

Ransomware Risk

- 백업 데이터까지 공격 대상이 된다
- 과거 시점으로 복구해도 깨끗하다고 보장할 수 없다

가상화 DR의 구조적 한계 — ROI 시트에 누적되는 4항목

5~15%

평시 자원 활용률
보험성 대기 비용

×2

3단 라이선스 이중 지불
하이퍼바이저·OS·미들웨어

IP·스토리지

종속 → 광역 L2 전제
페일오버 자동화 한계

5/7

단계가 인적 개입
야간·주말 호출 비용

KEY INSIGHT

선언적 매니페스트·이미지 불변성·데이터 복제 — 세 메커니즘이 결합해 상태를 '복원'하는 대신 '재현'한다

1



선언적 매니페스트 동기

Argo CD ApplicationSet · Git = SSoT

변경·드리프트를 선언 상태로 자동 수렴

2



이미지 불변성·서명

Harbor · Sigstore Cosign · OCI 표준

같은 다이제스트 = 비트 단위 동일 보증

3



데이터 복제

Velero · DB 네이티브 복제 · CSI 스냅샷

애플리케이션 상태와 의도적으로 분리

선언적 재현이 만드는 차이

가상화 DR

클라우드 네이티브 DR



복구 시간

수 시간 ~ 수 일



수 분



평시 활용률

30 ~ 50%



70 ~ 90%



라이선스

3단(HV+OS+MM)



단일 · 30~60%↓



복구 절차

직렬 7단계



병렬 3단계

상세 비교표·복구 시간축은 이어지는 2개 장 참조

KEY INSIGHT 다섯 개 차원 모두에서 클라우드 네이티브 DR이 우위 — 복구 시간과 평시 자원 효율에서 격차가 가장 크다

평가 차원	가상화 DR	클라우드 네이티브 DR
복구 시간 (RTO)	수 시간 ~ 수 일	수 분
복구 시점 (RPO)	분 ~ 시간	초 ~ 분
자동화 수준	낮음 — 수동 절차 의존	높음 — GitOps 선언적 재현
평시 자원 활용률	30 ~ 50%	70 ~ 90%
라이선스 운영 비용	3단 (HV + OS + MM)	단일 (약 30~60% 절감)

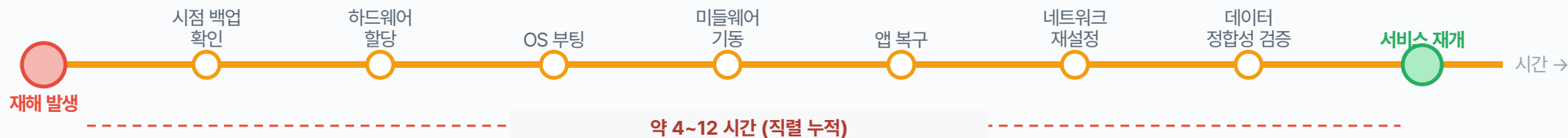
Source: 백서 1.3 — CNCF Annual Survey 2024 [S1] · Argo CD·GitOps DR [S4] · Velero 복제 패턴 [S5]

애플리케이션 페일오버는 데이터 복제 완료를 기다리지 않고 시작 가능

KEY INSIGHT

차이의 본질은 단계 수가 아니라 직렬→병렬 이동 — 가상화 DR은 단계가 차례로 누적돼 수 시간, 클라우드 네이티브는 병렬 처리로 수 분에 복구

재해 발생 후 두 모델의 복구 시간축 비교 — 직렬 처리 7단계 vs 병렬 처리 3단계

가상화 DR
시점 복원 모델클라우드 네이티브 DR
선언적 재현 모델

시간축 길이 자체가 두 모델의 RTO 격차를 드러낸다 — 직렬 7단계 vs 병렬 3단계

[그림] 시점 복원(직렬 7단계) vs 선언적 재현(병렬 3단) 시간축

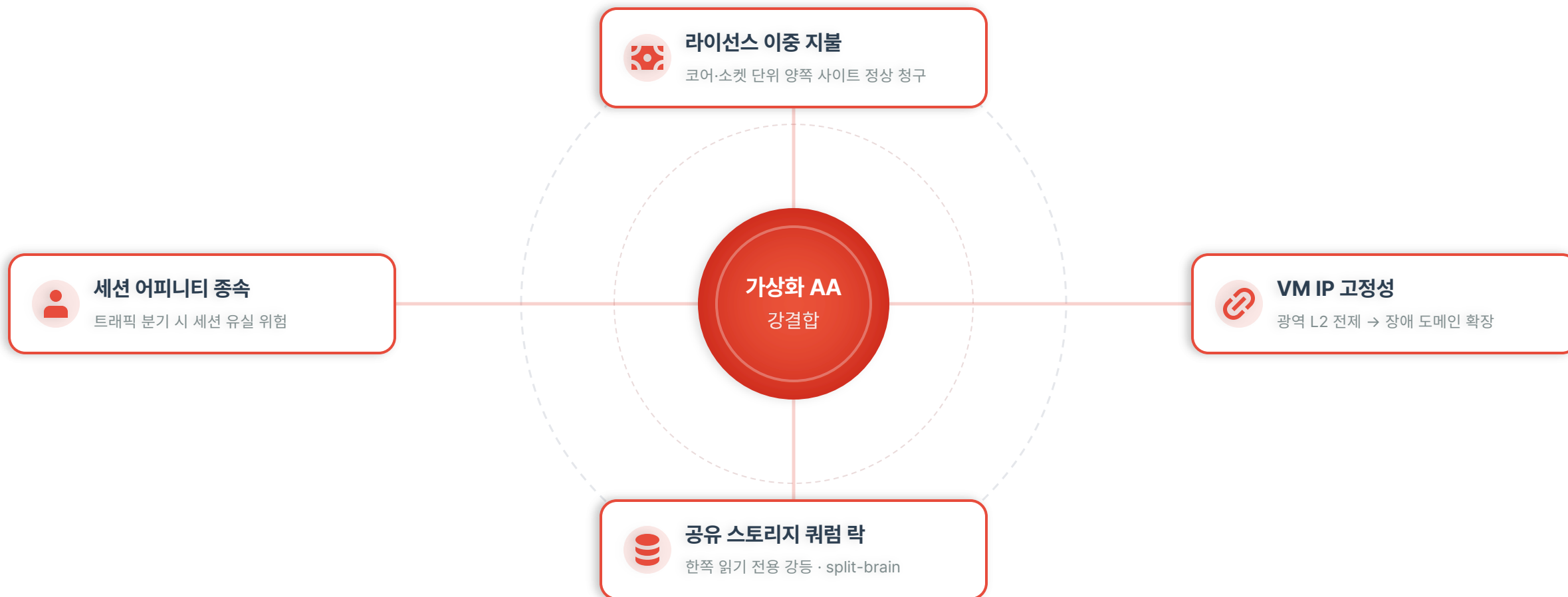
CHAPTER 02

쿠버네티스 **Active-Active** 의 구조적 이유

4중 종속의 해소 · 이미지 불변성과 Pod 부팅

KEY INSIGHT

4중 종속이 서로를 강화 — 어느 하나도 단독 해결 불가, 대부분 Active-Standby 로 회귀했다



KEY INSIGHT

4구조가 한 묶음일 때만 안정 — 하나라도 빠지면 4중 종속이 부활한다

선언적 배포

reconcile 루프가 두클러스터를 동일 desired state 로 수렴

상태 비저장 분리

Deployment vs StatefulSet 명시 분리로 DB 쿼리를 실행경로 밖에

GitOps 동기

ApplicationSet 가 drift 를 자동 복구 · 변경 = 커밋 이력으로 추적

Service Mesh 라우팅

Istio Multi-Primary · VirtualService 가중치로 IP 종속 흡수

트래픽 분기 3계층

계층 / 메커니즘	전환 지연	세션 정합	운영 부담
DNS TTL 기반 · 가중치 · GSLB	분 ~ 시간	약함	낮음
부하분산기 (LB) L4 분배 · Anycast · VIP	초 단위	중간	중간
Service Mesh L7 라우팅 · Istio	초 미만	강함	높음

DNS · LB · Service Mesh — 누적 다단 구조

Active-Active 패턴 대조 — 가상화 실패 vs 쿠버네티스 성공

라이선스·IP·스토리지·세션 4중 종속(좌) vs 선언적 매니페스트·Service Mesh·Stateless 분리·GitOps 4중 분리(우)

X 가상화 Active-Active — 강결합 실패 패턴

사이트 A (VM 클러스터)
고정 IP · 라이선스 코어 종속사이트 B (VM 클러스터)
동일 IP 대역 · 라이선스 이중공유 스토리지 + 쿼럼 락
(광역 L2 종속 · SAN 동기)

↓ 4중 종속이 서로를 강화 (강결합 cycle)

X 라이선스 코어 이중 지불
(양 사이트 모두 활성화)X IP 고정 종속
(광역 L2 · 사이트 이동 불가)X 스토리지 쿼럼 충돌
(SAN 동기 끊김 → split-brain)X 세션 종속
(sticky session · 외부 저장소 없음)결과 — 단일 요소 해결 불가, Active-Active 실패 회귀
(비용 폭증 · 운영 복잡도 누적 · 장애 시 양쪽 동시 마비)

✓ 쿠버네티스 Active-Active — 느슨한 결합 성공 패턴

클러스터 A (K8s)
Stateless Pod · 독립 컴퓨트클러스터 B (K8s)
Stateless Pod · 독립 컴퓨트GitOps SSoT (Argo CD · Flux)
선언적 매니페스트 양쪽 동기

↓ 4중 분리가 서로를 독립화 (loose coupling)

✓ 선언적 매니페스트
(YAML SSoT · 라이선스 코어 무관)✓ Service Mesh 라우팅
(Istio · IP 종속 흡수)✓ Stateless / Stateful 분리
(독립 스토리지 · 쿼럼 충돌 무관)✓ GitOps 자동 동기
(매니페스트 오류 자동 복구)결과 — 각 구조가 독립 작동, Active-Active 실패 안착
(점진 전환 · 카나리 배포 · 한쪽 장애 시 다른 쪽 정상 운영)

핵심 — 차이의 본질은 비용·라이선스가 아니라 워크로드 분해 단위. 가상화는 4중 종속이 서로를 강화하고, 쿠버네티스는 4중 분리가 서로를 독립한다.

[그림] 가상화 강결합 vs 쿠버네티스 느슨한 결합

KEY INSIGHT

Pod 는 VM 부팅 7단계 중 5단계 생략(커널 공유) — RTO 결정 변수가 인프라에서 운영 거버넌스로 이동



패치 의존성 제거

"패치 작업"이 아니라 "재배포"
새 이미지·다이제스트로 교체



신뢰 모델 강화

복구 단위 = 매니페스트 +
다이제스트 + 볼륨 스냅샷



감사 가능성 향상

GitOps 커밋 이력 = 감사 증적
별도 변경관리 시스템 불필요

동일 가로 폭에 그린 두 시간축의 길이 = 부팅 소요 시간 비율 (VM ~30배)

VM 부팅

BIOS 부팅

POST · UEFI

게스트 OS 부팅

커널 · 드라이버 · init

미들웨어 시작

WAS · DB · 구성 파일 적재

애플리케이션 ready

트래픽 수용 가능

분~10분 단위

RTO 하한 = 부팅 시간

Pod 부팅

image pull

캐시 시 0초

컨테이너 start

ns · cgroup · entrypoint

애플리케이션 ready

트래픽 수용 가능

초 단위

RTO 결정 변수 아님

← Pod 는 호스트 커널 공유로 BIOS · 커널 · init · 미들웨어 5단계 생략

시간축 가로 길이 비 (VM:Pod = 30:1) — 가상화 DR 의 RTO 하한이 부팅 시간으로 정해지던 시절을 컨테이너가 끝낸다 [S1] [S5] [S12]

CHAPTER 03

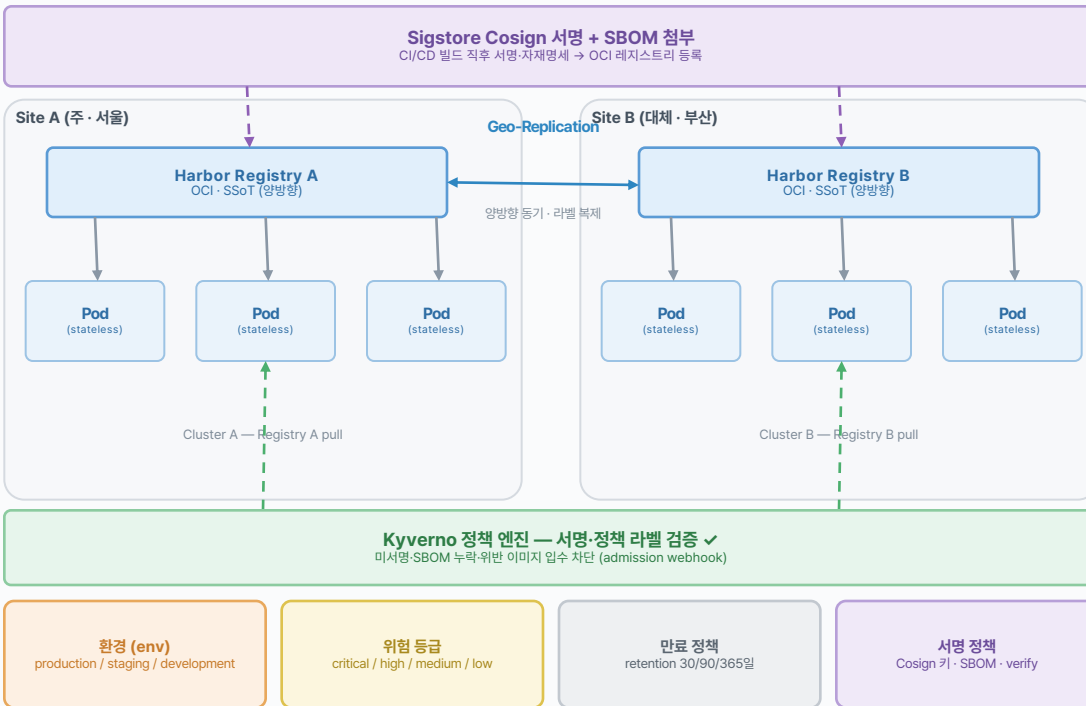
사이트 간 이미지·레지스트리 동기화

SSoT · Cosign 서명 · 4 동기화 패턴과 망분리 DMZ

KEY INSIGHT

SSoT 미정이 가장 위험 — 같은 태그가 두 사이트에서 다른 이미지를 가리키면 정산 처리에서 사고화된다

이미지 Geo-Replication 토폴로지



[그림] 주 사이트 SSoT → 부 사이트 복제 흐름

SSoT 정의 — 거버넌스 보드 의결 사항

단방향 — 주 사이트 SSoT, 부는 복제본

운영·권한 모델 단순 · 충돌 0

단점: SSoT 가 단일 장애점

양방향 — Active-Active 정합

이름공간 분리·태그정책·정책엔진 3장치 필요

CI 는 단일 빌드를 양쪽 동시 등록

Geo-Replication 정책 4변수

레지스트리	정책 단위	동기 주기	충돌 정책	무결성	운영 부담
Harbor	프로젝트 단위	즉시·cron	last-write-wins	Cosign 통합	중간
Quay (Red Hat)	저장소 단위	즉시·예약	last-write-wins	서명 통합	중간
JFrog Artifactory	저장소 단위	즉시·cron	push/pull 동시	SBOM 첨부	높음

[그림] 복제 주기 · 트리거 · 범위 필터 · 충돌·재시도 정책

양방향 Geo-Replication 토폴로지 — Cosign 서명 + Kyverno 검증

두 사이트 레지스트리가 서로 SSoT — 입수 시점에 정책 엔진이 서명·정책 라벨을 검증

Sigstore Cosign 서명 + SBOM (Software Bill of Materials) 첨부

CI/CD 파이프라인이 빌드 직후 이미지에 서명·SBOM 첨부 · keyless OIDC 서명 · Rekor 투명성 로그 기록 → OCI Distribution Spec으로 레지스트리 등록

Site A (주 사이트 · 서울)

Harbor Registry A

OCI Distribution Spec · SSoT (양방향)

Pod
(stateless)

Pod
(stateless)

Pod
(stateless)

Kubernetes Cluster A — Pod들이 Registry A에서 이미지 pull

Site B (대체 사이트 · 부산)

Harbor Registry B

OCI Distribution Spec · SSoT (양방향)

Pod
(stateless)

Pod
(stateless)

Pod
(stateless)

Kubernetes Cluster B — Pod들이 Registry B에서 이미지 pull

Geo-Replication
(양방향 동기)

이벤트 트리거 · 정책 라벨 단위 복제

Kyverno 정책 엔진 — 입수 시점 서명·정책 라벨 검증 ✓

Cosign 서명 미부착·SBOM 누락·정책 위반 이미지는 양 사이트 모두에서 입수 차단 (admission webhook · enforce 모드)

정책 라벨 (이미지별 메타데이터) — 라벨 단위로 복제 범위·검증 규칙 운영

환경 (env)

production / staging / development

위험 등급

critical / high / medium / low

만료 정책

retention 30/90/365일 · 태그 garbage collect

서명 정책

Cosign 키 ID · SBOM 필수 · Kyverno verify 규칙

KEY INSIGHT

위조 이미지를 부 사이트에 밀어 넣어도 정책 엔진이 서명 검증 실패로 Pod 기동 차단 — 백업·복원보다 강한 신뢰 부트



거버넌스 보드 안건 — 7개 의결 항목 (회의 1건으로 통과)

1 SSoT 사이트 지정
단방향 또는 양방향+이름공간 분리 명시

2 SSoT 변경 권한자
변경 절차·감사 로그 보존 기간 문서화

3 복제 4변수 채움
주기·트리거·범위필터·충돌정책

4 OCI v1.1+ RFP 항목
평가표 필수 항목으로 등록 → 교체 비용 상한

5 CI 서명+SBOM 표준
빌드 직후 파이프라인 표준 단계로 정의

6 입수 시점 검증 강제 — 최우선
운영·DR 두 클러스터 모두에 동시 적용

7 ISMS-P 매핑 문서화
변경통제·악성코드·재해복구 무결성 항목 매핑

⚠ ⑥ 없이 ①~⑤ 만 도입되면
거버넌스 형식만 갖추고 실제 차단 효과는 없다

KEY INSIGHT

만능 패턴은 없다 — 망분리는 DMZ 게이트웨이 단방향 결합으로 양방향 구현, Kyverno 가 운영 시점 서명 검증 강제

망분리 양방향 Geo-Replication + 서명 검증

외부망(인터넷) — 운영망 A · DMZ · 운영망 B 어느 곳도 직접 통신 금지



정책 라벨 + Kyverno 정책 엔진 — 운영 시점 강제 통제
환경 등급 · 위험 등급 · 만료 시점 · 서명 정책 · ClusterPolicy verifyImages → Admission 차단

단방향 Push

충돌 0 · SSoT 단순
SSoT 가용성 의존

양방향 동기

단일 장애점 제거
충돌 정책 설계 필수

단방향 Pull

대상 사이트 회선 자율
조회 주기 = RPO lag

Pull-through Cache

회선·스토리지 절감
캐시미스 시 SSoT 의존

레지스트리 동기화 4 가지 패턴 트레이드오프

패턴	SSoT 명확성	회선 사용	충돌 위험	적합 시나리오	운영 부담
단방향 Push	매우 명확 (원본 일방)	변경분 즉시 (빌드 시점)	0 (구조적 차단)	Active-Standby DR (DR read-only)	낮음
단방향 Pull	명확 (소비자 주도)	장기 폴링 (조회 주기)	0 (구조적 차단)	망분리-점진 도입 (대상 자율)	낮음
양방향 동기	분산 SSoT (충돌 정책 필수)	양쪽 변경분 (양방향)	있음 (정책 필요)	Active-Active (양쪽 빌드 권한)	높음
Pull-through Cache	명확 (lazy fetch)	최소 (요청 시만)	0 (digest 검증)	회선·스토리지 절감 (보조 패턴 권장)	중간

셀 배경 — 녹색: 운영 안전 · 적색: 정책 설계 필수 · 황색: 보조 패턴 결합 권장. 변경 빈도·회선 비용·신뢰 모델 3변수로 워크로드별 선택 [S6][S7]

CHAPTER 04

데이터베이스 동기화와 복구 절차

워크로드 4 카테고리 매트릭스 · 7단계 vs 3단계

KEY INSIGHT

출발점은 DB 제품이 아니라 워크로드 유형 — 각 카테고리는 회피 불가능한 비용(자연·충돌정책·운영전환·의미제약)을 진다

비교 차원	동기 복제 Patroni·MGR·Galera	CDC 비동기 Debezium+Kafka MM2	분산 SQL Cockroach·Yugabyte·TiDB	NoSQL multi-DC Mongo·Cassandra·Scylla
일관성 모델	강한 일관성 linearizable	결과적 일관성 eventual	글로벌 강한 일관성 직렬화 가능	쿼리 단위 선택 LOCAL~EACH_QUORUM
RPO (이상 시)	0	수 ms~수 초	0	쿼리별 선택
사이트 단절 시	쿼럼손실 시 쓰기거부	양쪽 단독 쓰기	쿼럼손실 영역 차단	양쪽 단독 쓰기
충돌 정책	불필요 (단일 순서)	LWW / CRDT / 규칙	내부 단일 순서	LWW / 응용 책임
권장 워크로드	금융거래·재고·결제 RPO 0 필수	세션·로그·이벤트 가용성 우선	신규 글로벌 OLTP 신규 우선 권장	시계열·카탈로그·로그 대용량 처리량

KEY INSIGHT

매트릭스에서 워크로드 행 하나만 짚으면 해당 워크로드의 DR 카테고리가 곧바로 결정된다 — 분류 회의 시간을 분 단위로 단축

DB 동기화 4 카테고리 x 4 워크로드 매트릭스

각 카테고리의 일관성·RPO·비용·운영 난이도·추천 워크로드를 한 표로 정리 — 카테고리 결정의 출발점

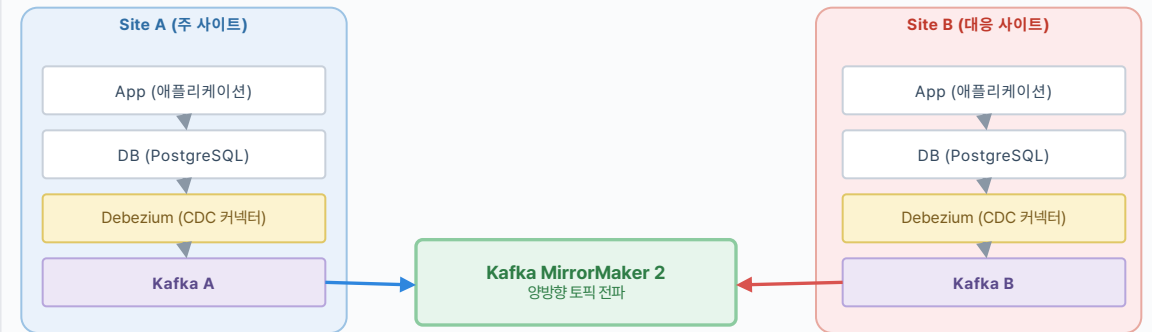
카테고리	일관성 모델	복구 시점 (RPO)	비용	운영 난이도	추천 워크로드
동기 복제 Patroni-MySQL GR-Galera	강한 일관성 (퀵 동기)	0 (동기)	높음 (round-trip)	중간 (모니터링)	트랜잭션 (결제·재고) RPO 0 필수
CDC 비동기 Debezium+Kafka MM2	결과적 일관성 (eventual)	초~분	낮음 (비동기)	중상 (정책 합의)	로그·이벤트 결과적 일관성 충분
분산 SQL CockroachDB·Yugabyte·TiDB	글로벌 일관성 (Raft 합의)	0 (합의)	중간 (RTT 의존)	높음 (모델 전환)	신규 OLTP 글로벌 일관성 필수
NoSQL multi-DC MongoDB·Cassandra·Scylla	가용성 우선 (CAP→AP)	초	낮음 (키 단위)	중간 (샤딩 설계)	세션·분석 시계열·대용량

셀 배경색 = 속성 유리도 유리 보통 불리 (비용·난이도 高)

[그림] 트랜잭션·세션·로그·분석 x 4 카테고리 권장값

CDC 비동기 충돌 해결 토폴로지

Debezium + Kafka MirrorMaker 2 양방향 전파 · 충돌 해결 3 방식



충돌 해결 3 방식

방식 1 — Last-Write-Wins

타임스탬프·시퀀스 기반 · 충돌 시 큰 값 자동 채택
적합: 캐시·카운터·세션값
부담: 시계 동기화 정확도

방식 2 — CRDT

충돌 없는 복제 자료형 · 자료형 자체가 병합 보장
적합: 좋아요·태그·세션
부담: 자료형 설계 난이도

방식 3 — 비즈니스 규칙 기반

자동 규칙 또는 수동 규칙 · 응용·운영자 결정
적합: 결제·재고·회원 정보
부담: 규칙 유지·운영자 개입

[그림] 두 사이트 동시 쓰기 충돌 검출·해결 경로

KEY INSIGHT

7단계 중 6단계가 인적 개입 — 행안부 핵심등급 RTO 4시간 목표를 구조적으로 초과한다

RTO 단계 분해 - 가상화·물리 DR 7단계 vs 클라우드 네이티브 3 단계

좌측은 인적 개입 분기점이 단계마다 누적 → 수 시간~수 일 · 우측은 트리거 1회 후 자동 동기화 → 분 단위

가상화·물리 DR · 7단계

단계	작업 · 담당	평균 소요
1	백업 매체 위치 확인 운영자 호출	30분~2시간
2	하드웨어 할당 인프라팀	1~4시간
3	OS 부팅 자동	5~15분
4	미들웨어 재구성 운영자 수동	30분~2시간
5	애플리케이션 복구 운영자 + DBA	1~3시간
6	데이터 적재 DBA	30분~수 시간
7	네트워크 재설정 네트워크팀	15분~1시간

합계 · 수 시간 ~ 수 일

클라우드 네이티브 DR · 3단계

단계	작업 · 담당	평균 소요
1	페일오버 트리거 운영자 1회 클릭	수 초
2	Argo CD ApplicationSet 동기화	1~3분
3	DNS · Service Mesh 트래픽 전환 자동	30초~2분

합계 · 분 단위



5.5~17시간

누적 RTO (훈련된 환경의 최선 시나리오)

실측은 훈련치의 2~3배에 이른다는 보고



6 / 7

단계가 인적 개입 분기점

단계 간 인수인계 누락 시 재시도



×3~5

야간·주말 호출 단가 (평일 대비)

6개 직군 동시 호출 시 단가 × 6배

KEY INSIGHT

6단계가 자동화되어 인적 개입은 트리거 1회 — RTO 목표를 가깝스로 달성이 아니라 재정의할 여유를 만든다

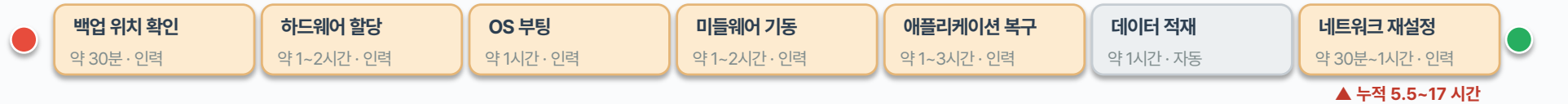
장애복구 절차 비교 - 가상화 7단계 vs 클라우드 네이티브 3단계 (같은 시간축)

가로 길이 = 누적 소요 시간 (스케일 일치) · 짙은 색 = 인적 개입 · 옅은 색 = 자동화

가상화·물리 DR

직렬 7단계 · 누적 5.5~17시간

재해 발생



클라우드 네이티브 DR

병렬 3단계 · 누적 2~9분

재해 발생



← 가로 길이 비율 = 약 10:1 (시간 스케일 그대로) — 시간 자체가 본질 차이

범례 인적 개입 자동화

가상화 6/7 인적 (자동 1) · CN 1/3 인적 (자동 2)

1 페일오버 트리거

거버넌스 의결

0.5~5분

인적 1회

2 동기 + Pod 기동

ApplicationSet

1~3분

자동

3 트래픽 전환

DNS-Mesh 가중치

0.5~1분

자동

🔒 사이버 재해 — ISMS-P 4통제 1:1 매핑 (별도 도구 없이)

- 무결성 검증 → 이미지 불변성 + Cosign 서명
- 신뢰 시점 → GitOps 커밋 이력
- 권한 분리 → Argo RBAC + Git 브랜치 보호
- 감사 추적 → GitOps 동기화 로그

CHAPTER 05

도입 효과와 해외 사례

자원·라이선스 30~60% 절감 · 글로벌 5개 사례 공통 패턴

KEY INSIGHT

DR 사이트가 "보험성 지출"에서 "평시 가용 자원"으로 재정의 — Bin Packing+HPA/VPA 와 라이선스 단순화의 결합 효과

DR 자원·라이선스 비용 비교

동일 워크로드(100 vCPU / 200GB) 기준 · 절대 금액 없이 정성 + 상대 비율로 표현

차원	가상화 (VM + 하이퍼바이저)	컨테이너 (Pod + K8s)
DR 사이트 자원	100 vCPU / 200GB (운영과 1:1)	40~70 vCPU / 80~140GB · 30~60% ↓
평시 자원 활용률	30~40% · 대기 위주(유향)	70~90% · Active-Active 활용
하이퍼바이저 라이선스	필요 (코어·소켓, 운영+DR)	불필요 (호스트 커널 공유)
게스트 OS 라이선스	필요 (VM·코어, 운영+DR)	불필요 (경량 이미지 내장)
미들웨어 라이선스	필요 (인스턴스·코어, 별도)	절감 (컨테이너 단위 산정)
총 환산 비용 (상대)	기준 (100%)	약 40~70% · 30~60% 절감

30~60%

동일 워크로드 DR 자원·라이선스 절감



자원 효율 — Bin Packing + HPA/VPA

스케줄러가 소수 노드에 집약 배치 · 트래픽에

따라 복제본·자원 동적 조정

DR 사이트가 평시 비핵심 워크로드 수용



3단 라이선스 단순화

하이퍼바이저 제거 · 게스트 OS 경량화 ·

미들웨어 컨테이너 단위로 이동

절감 폭은 자원요청 정확도·컨테이너 친화 비율에 비례

KEY INSIGHT

채택 스택은 달라도 3결정(GitOps·멀티클러스터 거버넌스·복제경로 분리)은 모두 동일 — 운영은 SRE·플랫폼 엔지니어링 재편

사례	전환 전 아키텍처	전환 동기	채택 스택	측정 복구 지표	운영 변화
Adidas (KubeCon EU)	모놀리식 e-Commerce + 자체 데이터센터 단일 사이트	Black Friday 단기 트래픽 피크 + 사이트 장애 시 DR	쿠버네티스 멀티 리전 + GitOps + 컨테이너 레지스트리 복제	RTO 분 단위 (시간 단위 → 분 단위)	SRE · 플랫폼팀 재편 + 평시 · DR 운영 통합
Lufthansa Systems (Red Hat Summit)	항공 운항 레거시 + 가상화 단일 사이트	미션 크리티컬 24/7 운항 + 시점 백업 회복 시간 한계	쿠버네티스 + 멀티 클러스터 + GitOps + 이미지 · 데이터 복제 분리	RTO 분 단위 (시간 단위 → 분 단위)	플랫폼 엔지니어링 도입 + 인력 재교육 약 12개월
ING Bank	분산 모놀리식 + 사이트 · 도구 분리	금융 감독 변경 통제 + 평시 · DR 운영 분리 부담	멀티 클러스터 거버넌스 + GitOps + 정책 엔진 (Kyverno 등)	MTTR 한 자릿수배 단축 (분 단위)	평시 · DR 플랫폼팀 통합 + 야간 · 주말 호출 ½ 이하 감소
JPMorgan Chase	분산 모놀리식 + 단일 사이트 시점 백업	글로벌 24시간 거래 미션 크리티컬 RTO · RPO	쿠버네티스 멀티 사이트 + GitOps + 분산 데이터 복제	RTO 분 단위 + RPO 분 단위 미만 목표	글로벌 분산 SRE 모델 + GitOps 매니페스트 SSoT
SAP S/4HANA on Kubernetes	ERP 미들웨어 통합 + 가상화 라이선스 종속	클라우드 전환 + 롤링 업데이트 무중단 패치	쿠버네티스 Operator + 멀티 클러스터 거버넌스	ERP RTO 분 단위 (시간 단위 → 분 단위)	ERP 인력 플랫폼 편입 + 재교육 약 6~12개월

① GitOps 통합

변경 이력 = 감사 증빙 · 페일오버 = GitOps 동기

② 멀티클러스터 거버넌스

정책 일원화 · 표류 차단 · 인증·접근통제 통합

③ 앱·데이터 복제 경로 분리

동기화 자유도 · 영향 범위 축소 · 신뢰 부트 유지

운영 모델 재편

DR 전담조 소멸 → 플랫폼 엔지니어링 통합

GitOps 통합 · 멀티 클러스터 거버넌스 · 데이터 복제 분리 — 애플리케이션 경로와 데이터 복제 경로를 의도적으로 분리해 독립 운영.

App 계층

매니페스트 · 이미지
GitOps SSoT

Git Repository
(매니페스트 SSoT)

Argo CD
ApplicationSet

단일 Git 매니페스트가
여러 사이트에 동일 적용

거버넌스 계층

정책 · RBAC · 인증
표류 차단

Cluster Hub · 정책 엔진
(RBAC · 인증)

Cluster A (사이트 1)

정책 동기 · 표류 차단
인증·접근 통제 일원화

Cluster B (사이트 2)

데이터 계층

별도 복제 경로
CDC · 분산 SQL · 동기

DB Cluster B (사이트 2)

DB 복제 (별도 경로)

DB Cluster A (사이트 1)

동기 복제 · CDC 비동기
분산 SQL · NoSQL multi-DC

핵심 메시지 — 애플리케이션 경로(파란색)와 데이터 복제 경로(주황색)는 의도적으로 분리. 한쪽 동기화 실패가 다른 쪽 운영을 막지 않음.

CHAPTER 06

정책 정합성과 도입 권장

5개 정책 문서의 수렴 · 의사결정 매트릭스

KEY INSIGHT

"사전 구축" 의무는 GitOps 동기 루프에서 평시 배포의 부산물로 충족 — 핵심등급 분 단위 RTO·RPO 0 직접 매핑

정책 요구사항 ↔ 클라우드 네이티브 구성 요소

법·정책	적용 대상	핵심 요구사항	측정 지표	클라우드 네이티브 DR 정합
「전자정부법」 제49조	공공기관 정보시스템 전반	재해복구체계 의무 구축 (일반 의무)	RTO·RPO 등급 분류	직접 충족 (클라우드 네이티브 DR 전 구성 요소가 의무 이행)
행정안전부 재해복구 체계 구축 지침	정보시스템 등급별 (핵심~보통 4등급)	등급별 RTO·RPO 목표 + 정기 훈련 의무	시간(RTO)·시점(RPO) 정량, 훈련 빈도	Active-Active 로 RTO 단축 + Argo CD-Chaos Mesh 로 훈련 일·주 단위 자동화
디지털플랫폼정부 전환 로드맵	공공부문 정보시스템 (신규·재구축 우선)	클라우드 네이티브 아키텍처 전환 (컨테이너·MSA 채택)	채택률·전환 일정, 사업 평가 점수	본 백서가 권고하는 참조 모델 그 자체 (미채택 시 감점 책임)
NIA 클라우드 네이티브 가이드	공공기관 신규·전환 사업	컨테이너·MSA 도입 절차, 레퍼런스 아키텍처, 검증 항목	단계별 체크리스트 충족률	직접 적용 (가이드의 실행 청사진을 그대로 사용)
KISA ISMS-P 재해복구 통제	ISMS-P 인증 대상 조직 (공공·민간)	DR 통제 항목 (복구 절차·매체 보관·무결성 검증)	통제 충족도 (인증 심사 기준)	이미지 불변성·서명·매니페스트 동기화 통제 자동 충족

☐ 녹색 열 = 클라우드 네이티브 DR 정합 (다섯 정책 문서의 의무·권고를 직접 또는 자동으로 충족)

[그림] 5개 문서 요구사항 행 × 구성 요소 열 매핑



「전자정부법」 제49조 + 시행령

- 재해복구체계 "사전 구축·운영" 일반 의무
- 구체 기술 요건은 미제시 (모호성)
- 사고 시 사전 구축 여부 = 책임 1차 기준



행안부 재해복구 체계 구축 지침

- 4등급 정량화 — 핵심등급 분 단위 RTO · RPO 0
- 정기 훈련 의무 → 카오스 자동화와 정합
- 등급은 기관 분류 — 충족 수단 선택의 문제



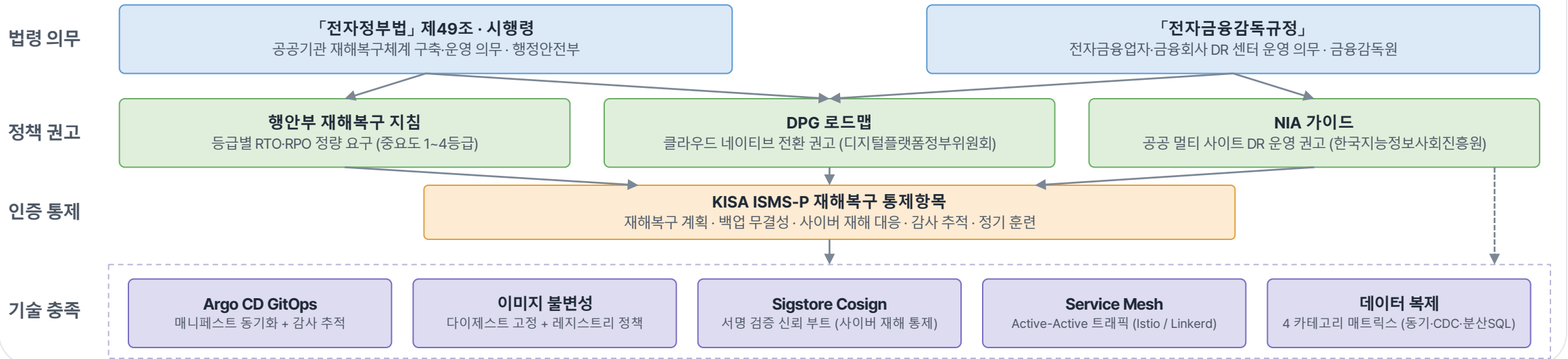
미채택 시 입증 책임 발생

감사원·국정감사·ISMS-P 갱신에서 직접 지적 대상 — 기관장 행정 책임 영역

KEY INSIGHT

발행처 다른 5개 문서가 3단 구조(GitOps+불변성+데이터복제)로 수렴 — 채택이 아니라 미채택의 입증 책임이 더 무겁다

클라우드 네이티브 DR ↔ 5개 정책 문서 기술 매핑 (의사결정 보고용)



DPG 로드맵

신규·재구축 우선 전환 · 평가 지표(자원·자동화·RTO·라이선스)
사업 기획·예산 심의 평가 항목 → 실질적 강제력
5·6장 정량 4지표와 직접 맞물림

NIA 가이드

6영역(런타임·오케스트레이션·CI/CD·관측·보안·DR) 요건
DR 영역 = GitOps + 이미지 불변성 + 데이터 복제 3단 반영
RFP 평가 골자·현대화 마일스톤으로 활용

✓ KISA ISMS-P — 통제 자동화 3메커니즘

- 불변성+Cosign → 사이버 재해 통제 1차 충족
- GitOps 매니페스트 → 감사 추적 자동 충족
- 카오스 엔지니어링 → 정기 훈련 운영 부산물

결론: "전환 여부"가 아니라 "우선순위와 일정"

KEY INSIGHT

정량 우위가 한 슬라이드 결론이라면, 정성 우위는 그 결론이 운영 모델 전체로 확산되는 메커니즘

정량 4지표 (가상화 → 클라우드 네이티브)

복구 시점 RPO

일 → 초·분

 동기 복제 시 사실상 0

복구 시간 RTO

시간 → 분

 한 자릿수 분

자원 절감

30~60%

 동일 워크로드 기준

라이선스

3단 → 1~2단

 계층 수 감소·예측성↑

정성 4측면 — 운영 모델로의 확산

 자동화 수준

선언된 상태의 지속 재현
→ 매일 배포가 곧 DR 훈련

평시 운영의 부산물화

 신뢰 부트

이미지 불변성 + 서명 검증
→ 미서명 배포 차단

통제 자동화

 정책 적합성

등급 + 로드맵·가이드 정합
→ 5개 문서 단일 모델로

정책 적합성 확정

 인력 모델

분리 운영팀 → 통합
플랫폼 엔지니어링 팀

인적 자원 재배치

KEY INSIGHT

5개 결정은 상호 종속 — 거버넌스 보드 한 번의 회의에서 한 매트릭스로 합의해야 순서 충돌이 사라진다

결정 영역	권장 / 근거
거버넌스 누가 결정하는가	DR 거버넌스 보드 공동 근거 1·9장
워크로드 분류 무엇을 먼저 옮길까	신규 시스템 우선 근거 4·5장
데이터 복제 4 카테고리 선택	워크로드별 분리 결정 근거 4장
다음 검증 후보 어떤 범위로	신규·분석 후보 근거 5·6장
인력 재편 조직 모델 구성	규모별 SRE/기존 전환 근거 1·8장

5개 영역 × 결정 옵션 × 권장

영역	결정 옵션 A	결정 옵션 B	결정 옵션 C	권장 (CNCF vendor-neutral)
거버넌스	CIO 단독 결정	DR 거버넌스 보드	정보보호 책임자 공동	DR 거버넌스 보드 + 정보보호 책임자 참여
워크로드 분류	트랜잭션 우선	세션 우선	신규 시스템 우선	신규 시스템 우선 (전환 비용 최소)
데이터 복제 모델	4 카테고리 단일 채택	워크로드별 매트릭스	분산 SQL 일괄	워크로드별 매트릭스
다음 검증 후보	전체 시스템	단일 비핵심 시스템	다중 신규 시스템	단일 비핵심 시스템 → 핵심 시스템 단계 확장
인력 재편	플랫폼 엔지니어링 전담 팀 신설	기존 인프라 운영팀 전환	SRE 통합	단계적 SRE / 플랫폼 엔지니어링 통합

KEY INSIGHT

3축을 한 장에 겹치면 실행 순서가 정해진다 — 신규 시스템부터, 복제 모델은 워크로드가 결정한다

워크로드 분류 × 데이터 복제 × 플랫폼 거버넌스 — 3축 통합 실행 지도



THANK YOU

전환 여부가 아니라 우선순위와 일정

다음 분기 거버넌스 보드에 5개 영역 매트릭스 한 페이지를 안건으로

1

모델 진단

시점복원 vs
선언적 재현

2

검증 후보 식별

신규·분석
워크로드

3

복제 워크북

4 카테고리
매트릭스 채움

4

정합성 갭 식별

등급별 요구사항
매핑

5

조직 모델 선정

SRE 통합 /
기존 팀 전환

Cloud Native Forum

Cloud Native Forum

<https://cncf.co.kr>

hello@cncf.co.kr