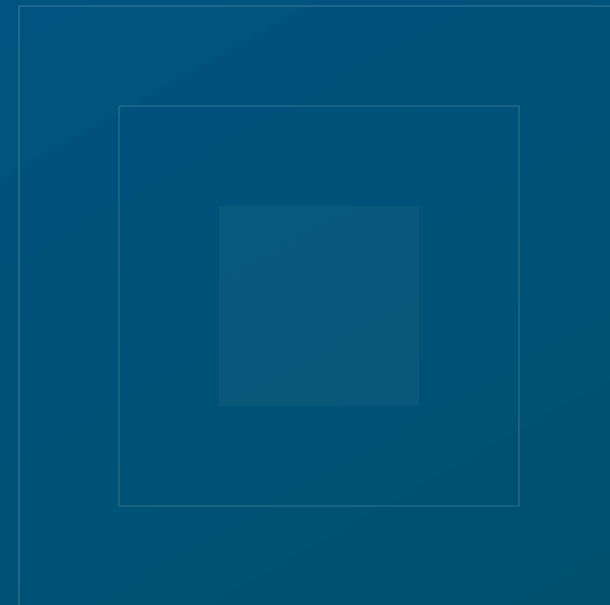


02

PART 2 · BENEFIT

클라우드 네이티브 그래서 무엇이 개선되는가

관리 · 속도 · 백업 · AI — 한 번의 구조 결정이 네 효과로



관리는 '수리'에서 '통째 교체'로 바뀐다

핵심 관리 대상 OS 100 → 노드 10, 금융기관 패치·보안 인력 약 1/10로 감소

VM — 고치기

도는 OS 안에 들어가 직접 손본다

- 커널·패키지 업데이트
- 재기동해도 되는지 사람이 판단
- 문제 생기면 원인 찾아 수리

서버 100대 = 이 일을 100번 반복

컨테이너 — 교체

안전한 새 이미지로 통째 바꾼다

- 새 이미지 빌드 → 순차 자동 교체
- 쿠버네티스가 상자를 내리고 올림
- 사람은 "이 이미지가 안전한가"만

판단이 자동화된 반복 작업으로

돌봐야 할 관리 대상

100→10

관리 대상 OS 100개 →
호스트 노드 10개로 (OS가 0이 되진 않음)

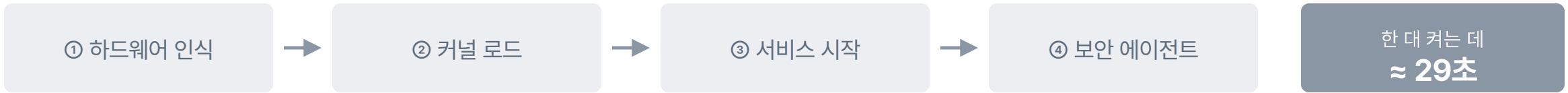
금융기관 사례

인력 약 1/10

초 단위 기동이 오토스케일을 '실제로' 작동시킨다

핵심 VM 29초 vs 컨테이너 1초 — '피크의 순간'에 오토스케일이 실제로 작동한다

가상화 (VM) — 계단을 처음부터 오른다



컨테이너 — 계단이 아예 없다



발전기 vs 전등

정전에 필요한 건
예열하는 발전기가 아니라,
누르면 바로 켜지는 전등

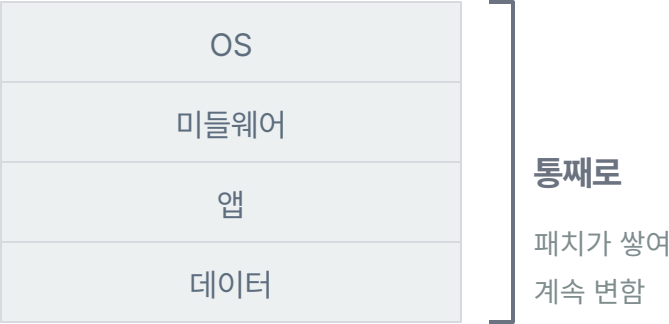
그래서 실전에서

트래픽이 몰릴 때 **'피크의 순간'에**
오토스케일이 실제로 작동
VM은 준비되기 전에 피크가 지나감

백업 대상이 '디스크 전체'에서 '데이터'로 줄어든다

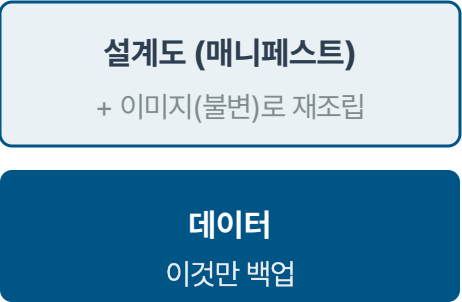
핵심 이미지·설정은 재현 가능 → 백업은 데이터만, DR 7단계→3단계 · RTO 시간→분

VM — 집 전체를 매번 사진



매번 디스크 전체를 스냅샷 — 무겁고 느림

컨테이너 — 설계도 + 데이터만



- 이미지는 변하지 않음
- 설정은 설계도로 문서화
- OS·앱은 백업 대상에서 빠짐
- **지킬 것은 데이터뿐**

집이 무너져도 설계도만 있으면 다시 짓는다

재해복구(DR) 절차

7단계 → 3단계

복구 목표 시간 (RTO)

시간 → 분

AI 워크로드의 공통 실행 기반도 쿠버네티스다

핵심 LLM·VectorDB·RAG·GPU·메시가 모두 컨테이너 위 → 기존 역량을 AI에 그대로 재사용



LLM 서버

추론 서버를
컨테이너로 배포

KServe · vLLM



Vector DB

검색 저장소를
컨테이너로 운영

Milvus · Weaviate



RAG · 에이전트

선언형으로
배포·확장

Deployment · CRD



GPU 공유

값비싼 GPU를
나눠 써 활용률 2배+

MIG · Time-slicing



서비스 메시

트래픽·보안을
앱 수정 없이

Istio · mTLS

쿠버네티스 — AI 워크로드의 공통 실행 기반

다섯 조각이 모두 하나의 표면 위에서 돈다

기존 클라우드 네이티브 역량(**배포·확장·복구·보안**)을 AI에 **그대로 재사용**

AI를 위해 인프라를 새로 만들 필요가 없다

만능은 아니다 — 한계는 보완하고, 적합한 것부터

핵심 이점은 도입 즉시 구조적, 한계는 조건부(보완 가능) — 무상태부터 단계적으로 전환

이럴 땐 신중해야 합니다

① 커널 공유의 보안

커널 하나를 함께 쓰니, 런타임 취약점이
여러 컨테이너로 번질 수 있음

보완 → gVisor·Kata로 민감 워크로드만 격리 강화

② 레거시 · 데이터베이스

강결합 레거시, 상태를 가진 DB는
전환이 까다롭거나 신중이 필요

보완 → 핵심 DB는 클러스터 밖 관리형으로 분리

시작은 이렇게 — 작게, 검증하며

1

무상태부터

웹·API·배치처럼 재시작해도 안전한 것부터

2

작게 검증

1~2개 먼저 옮겨 운영 방식·도구를 검증

3

확대

검증되면 점진 확대 — 노하우가 이어짐

무엇부터? 세 질문이면 갑니다

- ① 재시작해도 데이터가 안전한가?
- ② 최근 3년 내 코드가 수정됐는가?
- ③ 담당자가 지금도 사내에 있는가?

핵심은 다섯 문장으로 수렴한다

핵심 차이는 하나(칸마다 OS) → 가볍고·빠르고·자동 → 무상태부터 작게 시작

1

차이는 '칸마다 OS를 두느냐' 하나뿐입니다

VM은 칸마다 OS(단독주택), 컨테이너는 커널을 공유(아파트)

2

컨테이너는 커널을 공유해 가볍고 빠릅니다

무게 GB→MB · 집적률 3~10배 · 기동 분→초

3

클라우드 네이티브는 컨테이너를 자동으로 운영합니다

쿠버네티스(관리사무소)가 배포·확장·복구·감시를 대신

4

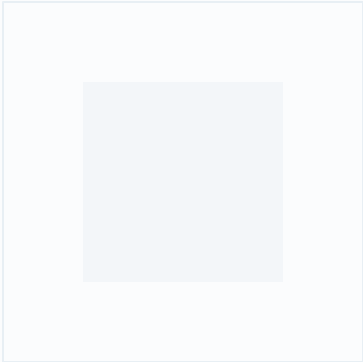
관리·속도·백업·비용이 동시에 가벼워집니다

관리 OS 100→10 · 기동 초 단위 · 백업은 설계도+데이터만

5

무상태부터 작게 시작해 검증하며 넓힙니다

만능은 아니다 — 한계는 보완하고, 적합한 것부터



THANK YOU

질문을 바꾸면 답이 달라집니다



어떤 OS로 바꿀까가 아니라, OS를 어떻게 운영할까

MSAP.ai

msap.ai · Cloud Native Practice

 <https://www.msap.ai/>

 02-6953-5427

 hello@msap.ai

