DevOps 엔지니어가 ISMS를 핑계로 접근통제를 제대로 다시 설계한 이야기

— 핑계로 시작했지만, 기왕 하는 거 제대로. 더 편하고 지속 가능한 게 더 좋은 보안이다.

구경열 ● 케이타운포유 ● AWSKRUG 보안소모임 · 2026.06.25

01 —— 출발점 – 공격자의 눈으로 본 우리

01

접근통제가 사실상 없었다

ISMS 준비를 시작하며 우리 자신을 공격자의 시점으로 다시 봤다. 통제의 부재는 곧 침해 경로였다.

- × 백오피스가 전부 public — 도메인만 알면 누구나 닿는다 → 무인증 노출 표면
- × MFA 미지원 — 비밀번호 하나가 뚫리면 끝 → 계정 탈취
- × 퇴사자 · 현직 구분 없이 접근 → 권한 회수 부재 · 내부 위협
- × 개발본부 Client VPN — 한번 붙으면 내부망 신뢰 → 측면 이동(lateral movement)

02 —— 복선 – 그때 스쳐 지나간 것

VPN 대안을 찾다,
물리장비 없는 ZTNA 하나가 눈에 들어왔다

별도 HW 없이 동작하고, 네트워크가 아니라 신원 단위로 접근을 통제한다고 했다. 스펙도 제대로 모른 채 "한번 써보고 싶다" 정도의 호기심이었다. — 정답은 이미 여기 있었지만, 우리는 먼 길을 돌아간다.

ZTNA SaaS · 호기심 ••• 하지만 일단 보류

※ 미리 밝혀둡니다 — 오늘은 제품 자랑이 아니라 설계 이야기입니다. 도구는 갈아끼울 수 있고, 원칙이 주인공입니다.

02

"ISMS면 폐쇄망 + 물리 VPN"이라는 통념을 따라가다

ISMS는 "이 장비를 쓰라"고 말한 적이 없다. 그건 통제 목표가 아니라 업계의 관성이었다. 우리는 그 관성을 따라 IP 경계 모델로 들어갔다.

선택 **국내 UTM 장비 도입**

- SSL / IPsec VPN이 아주 제한된 방식만 지원
- device · OS 의존성이 컸다
- "터널 안에 들어오면 신뢰" — IP 경계 모델

요구사항 **우리가 실제로 필요했던 것**

- iOS · macOS · Windows · Android 전부 지원
- MFA 필수 · 사용자/장치 단위 통제
- → UTM으로는 처음부터 불가능했다

04 —— 우회 ② – 더 비싼 길

글로벌 방화벽 벤더로 전환

— 주변 팀을 억지로 설득해서, 적잖은 비용을 들여

결과 **그래도 안 됐다**

- SSL VPN 구성 → **모바일 디바이스 의존성 여전**
- 일부 영업장은 비용 때문에 구형 장비 잔존
- 여전히 "한번 인증하면 내부망 신뢰" — 측면 이동 가능

그리고 **보안 마감은 다가오고**

- 완전치 않은 채로 전사 적용을 밀어붙여야 했다
- 더 비싼 장비를 샀지만 **통제 모델은 그대로**였다

05 —— 바닥 – 추석 즈음

전사 VPN을 강행했고,

즉시 장애 → 전면 롤백

적용 → 현장 곳곳에서 접속 실패 → **당일 롤백**

현장이 멈추면 통제는 우회된다 — 가용성도 보안이다.

고치려 할수록 벽만 늘어났다

버전 올렸더니

› 벤더가 상위 버전 위험성을 들어 지원 거부

SSL VPN 사양길

› 벤더 향후 미지원 방향 — 미래가 없는 선택

IPsec 대안?

› 설정 지옥, 제대로 구성 안 됨, Android 의존성 여전

AWS Verified Access

› 정책·인가 거점이 AWS 내부에 묶인다 — 신원 일원화가 깨짐

› 구성 난이도 · 비용 압박

지속할 수 없는 통제는 통제가 아니다 — 우리에게 필요한 건 장비가 아니라 모델이었다.

07 —— 회귀 – 도구가 아니라, 무엇을 신뢰하느냐

장비를 다섯 번 바꿔도 안 됐다. 바꿀 건 신뢰의 단위였다

전 **네트워크를 신뢰**

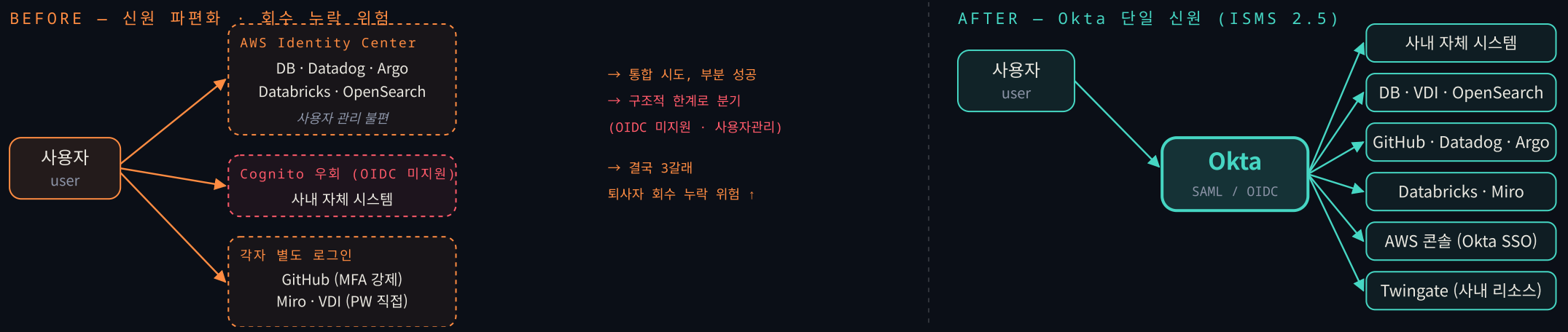
- "터널 안 = 신뢰" — 한번 붙으면 내부망 전체
- 인증은 **접속 시 1회**, 이후엔 통과
- 측면 이동(lateral movement)에 취약

후 **신원을 신뢰 — 지속검증**

- 매 접근마다 **사용자 · 인증수단 · 장치 · 리소스** 재검증
- 신뢰의 단위가 네트워크가 아니라 **신원**
- 리소스 단위 인가 — 닿을 수 있는 것만 닿는다

처음 스쳤던 ZTNA를 이번엔 **우리 위협모델에 대고** 다시 검증했다(트라이얼 연장까지 받아가며). 확신은 특정 제품이 아니라 **모델**에서 왔다 — 신뢰를 네트워크가 아니라 '신원'에 두는 설계. 거기서부터 접근통제를 다시 짰다.

무엇을 얹기 전에, 신원이 먼저 하나여야 했다



Identity Center 통합은 **OIDC 미지원·사용자 관리**의 구조적 한계로 3갈래로 갈라졌다 → **Okta 한 곳**으로 모았다. 사람을 도구마다 만들지 않는다 — **그룹 = 조직도, 어트리뷰트 = 상태, Workflows = 자동화**. 이 신원 하나가 다음 장의 모든 접근을 결정한다.

신원 하나에 모든 접근 경로를 매달았다

신원이 별이고 접근 경로가 그 둘레다 — 입·퇴사 한 번이 **모든 궤도에 동시에** 반영된다. 그런데 별이 하나면, **그 하나가 뚫리면 전부** 아닌가 — 다음 장에서 정면으로 답한다.

하나에 다 걸었다 — 그래서 그 하나를 가장 단단히

피싱저항 MFA

비밀번호로는 못 들어온다

- Okta **FastPass 강제** — OTP·Push가 아니라 피싱에 견디는 factor
- 탈취한 비밀번호 하나로는 신원이 서지 않는다

슈퍼어드민 격리

최고권한은 망 안에서만

- Okta 관리 콘솔(슈퍼어드민)은 **폐쇄망 안에서만**
- 가장 위험한 표면엔 망 격리를 **한 겹 더** (망분리는 만능 통념일 땐 문제, 최고권한 평면 심층방어로는 정답)

break-glass

비상계정은 물리적으로

- 비상 우회계정 MFA 디바이스는 **별도 시건** 보관
- 평시 노출 0 · 쓰는 순간이 곧 기록

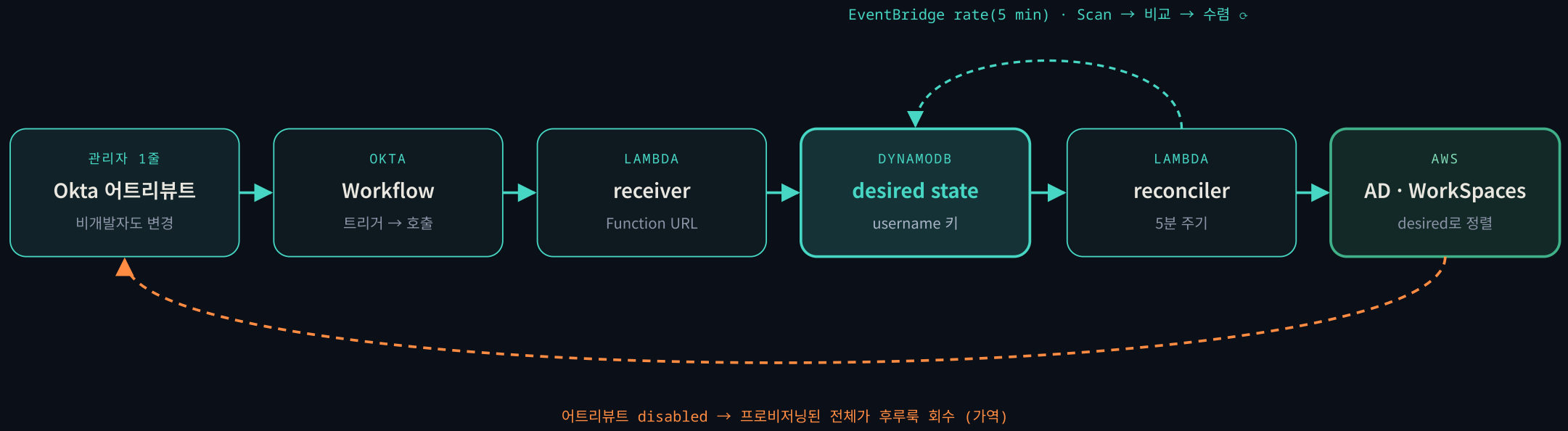
상시 모니터링

누가·언제·어디서

- Okta 시스템 로그를 **별도 대시보드**로 상시 감시
- 이상 접근을 사후가 아니라 **지금** 본다

"단일 신원"이 "단일 약점"이 되지 않게 — 허브에 **피싱저항·망 격리·물리 시건·상시 감시**를 겹쳤다. 다음 장: 그렇게 지킨 신원 위에서, 접근은 선언적 루프로.

관리자가 어트리뷰트 한 줄 — 나머지는 스스로 맞춰진다



WorkSpaces(VDI)는 Okta가 **직접 프로비저닝하지 못한다** — 그 빈자리를 선언적 루프로 메웠다. k8s 컨트롤러처럼, 원하는 상태를 선언하면 루프가 현실을 맞춘다. 입사도 퇴사도 **어트리뷰트 한 줄**. 구현 상세는 부록 A5-2.

편해진 게 아니라, 더 강해졌다

회수 보장 **고아 계정 0**

- 퇴사 = 어트리뷰트 한 줄 → 모든 궤도에서 동시 회수
- 사람 손이 아니라 **구조로** 보장 · 가역적 (ISMS 2.5)

drift 자가치유 **워크스페이스 표류 차단**

- 이 루프가 관리하는 범위에선, 콘솔에서 손댄 변경도 **5분 내 desired로 교정**
- 관리 대상 안에서 권한 표류가 쌓이지 않는다

최소권한 **선언적**

- 조직도 = 그룹·어트리뷰트 = desired access
- 권한이 사람 판단이 아니라 **모델**에서 나온다

3중 통제 **계층 방어**

- **신원**(Okta) + **네트워크**(Twingate inbound 0) + **장치**
- WorkSpaces device_type DENY · 도달 IP는 Twingate 뒤로만

그리고 이 전부가 **증적**이다 — DynamoDB desired state · Okta 변경 이력 · 접근 로그. 운영하면 저절로 쌓인다.

심사 기준에 이 설계를 그대로 얹다

"핑계로" 시작했지만 통제는 진지하게. ISMS-P 접근통제 항목을 ZTNA 구조로 충족하고, 그 충족을 **증적**으로 남겼다.

통제항목	요지	구현 (ZTNA)	증적
2.5 인증·권한관리	사용자 등록·해지, 권한 부여·변경·말소 / 퇴직 시 지체 없이 회수	**Okta(SAML/OIDC)** + 그룹 정책, 입·퇴사 자동 회수	IdP 그룹 변경 이력 · 접근 로그
2.6.1 네트워크 접근	비인가 접근 통제, 네트워크 분리	**inbound port 0**, per-resource authorization	리소스·정책 정의(코드) · 접근 감사
2.6.2 정보시스템 접근	접속 위치 제한, 특수권한 강화 인증, 미사용 계정 검토	**device posture** + 정책을 코드로 관리	정책 PR 이력 · 장치 검증 로그
2.6.6 원격접근	원격·재택 접근 통제	ZTNA **단일 경로로 일원화**	단일 접근 경로 감사 로그

출처: KISA **ISMS-P 인증기준 안내서 (2022.4.22 정본)**. 통제별 매핑·증적 생성 방식 상세는 **부록 A6**.

접근통제는 산출물일 뿐 — 핵심은 반복되는 ISMS 파이프라인

가장 의미 있었던 건 도구 도입이 아니라 — KISA 인증기준 가이드를 우리 회사 지침으로 번역하고, 그 충족을 증적으로, 그것도 반복·자동으로 만들어낸 것이다.

가이드
KISA ISMS-P 인증기준
원문 — 추상적 통제 목표

→

번역
우리 회사 지침으로
우리 맥락·시스템에 맞게 구체화

→

구현
통제 = 코드
정책=PR · 콘솔 클릭 아님

→

증적
운영의 부산물
반복·자동으로 저절로 쌓임

⟳ **한 번 만들고 끝이 아니라 계속 도는 파이프라인.** 그 단순함을 유지하려고 Okta·Twingate를 골랐다 — 도구가 주인공이 아니라, 파이프라인을 단순히 두기 위한 선택. 매핑·증적 부록 A6 · Terraform 부록 A7.

통제가 현장에 정착했다.
우회되지 않았다.

보안팀

✕ 통제 기준 · 심사 증적

운영 · DevOps

› 코드화된 정책 · 자동화

사용자 · 현장

› 매뉴얼 없이 쓰는 경험

셋이 **같은 정책**을 공유한다 — 보안과 운영의 사일로 없이. 통제가 불편하면 우회되고, 우회된 통제는 통제가 아니다.

완벽한 선택은 없다 — 감추지 않고 정면에서

Q SaaS인데 끊기면?

- 데이터는 **P2P 직접** — 맺은 세션은 영향 작다
- Connector **2 AZ 다중화** · 신규 세션만 영향(SLA)

Q 트래픽을 SaaS가 보나?

- P2P 직접(미경유) · Relay도 **복호화 불가**(zero-knowledge)
- 비밀번호 **미보관** — 인증은 Okta에 위임

Q M365 AD는 왜 안 붙였나?

- Okta↔Entra **이중 디렉터리 동기화**의 복잡도·실패지점 회피
- 신원 단일 소스는 **Okta 고정**, AD는 도메인 조인 백엔드로만

데이터플레인 도식 부록 A4 · 트레이드오프 전체(국외이전·망분리·M365) 부록 A9.

핑계로 시작했지만,
기왕 하는 거 제대로 했다.

우회할 이유가 없는 통제가,
끝까지 살아남는다.

자격증명 **Okta** + 네트워크 **Twingate**

진짜 '제대로'는 도구가 아니라 **ISMS를 반복 파이프라인으로** 만든 것 — 도구는 그 단순함을 위한 선택이었다. 더 편하고 지속 가능한 통제가, 결국 **더 좋은 보안**이었다.

감사합니다 —— Q & A

구경열 Kyungyeol Gu

케이타운포유 · DevOps Engineer

클라우드 인프라와 접근통제를, 보안과 운영이 갈라지지 않게 — 코드로 다룹니다.

in linkedin.com/in/kyungyeol-gu-402391170

더 깊은 근거 — 아키텍처 · ISMS 증적 · 트레이드오프 · 비용 — 은 **부록**(A1–A11)에 있습니다. 필요하시면 펼치겠습니다.

— APPENDIX · 부록 —

근거는 전부 여기에

본 발표는 서사 중심으로 가볍게 갔습니다. 아키텍처 원리·통제 증적·트레이드오프·출처는 이 부록에서 다룹니다. Q&A는 "부록 N번"으로 답하겠습니다.

A4 Twingate 아키텍처
A4-2 데이터플레인 정통 도식
A6 ISMS 매핑·증적
A3 감사 로그·탐지
A5 Okta SSO
A5-2 Reconcile 구현
A9 트레이드오프
A1 벤더 비교
A2 SSL VPN 사양길
A11 AWS 권한 일원화
A7 Terraform 운영
A8 도입 후 지표
A10 비용 TCO

A1 —— 벤더 비교 매트릭스

같은 기준으로 나란히 놓고 봤다

기준	국내 UTM 장비	글로벌 방화벽 벤더	AWS Verified Access	Cloudflare	Twingate
물리장비 불필요	✕	✕	○	○	○
전 OS 지원 iOS·Mac·Win·Android	△	△	○	○	○
MFA / SSO	제한적	부가구성	○	○	○
접근 모델	IP/터널	IP/터널	ZTNA	ZTNA	ZTNA
무매뉴얼 UX	✕	✕	△	△	○
운영 부담	높음	높음	중간	중간	낮음
비용 성격	장비+라이선스	장비+라이선스↑	사용량 과금	구독	구독
우리 조직 적합도	✕	✕	△	△	우리 컨텍스트 최적

○ 충족 · △ 부분 · ✕ 미충족. **UTM·방화벽은 IP 경계 모델**이라 멀티 디바이스+MFA를 처음부터 못 맞췄다(public IP 노출·blast radius 큼). 정답은 ZTNA라는 **모델**이고 제품은 조직마다 다르다 — 실패 사례는 본문 익명, 채택 스택만 실명. 출처: Twingate — Architecting Network Connectivity for a Zero Trust Future (2022)

"미래가 없는 선택"의 근거

FortiOS 7.6.3부터 SSL VPN tunnel mode가 IPsec VPN으로 대체되어 GUI·CLI에서 제거. 전 모델 적용, 이전 설정은 업그레이드 시 이관되지 않음. 2GB RAM 이하 모델은 web·tunnel mode 자체 제거.

FortiOS 7.6.3 Release Notes — SSL VPN tunnel mode no longer supported

docs.fortinet.com/document/fortigate/7.6.3/fortios-release-notes/173430/ssl-vpn-tunnel-mode-no-longer-supported

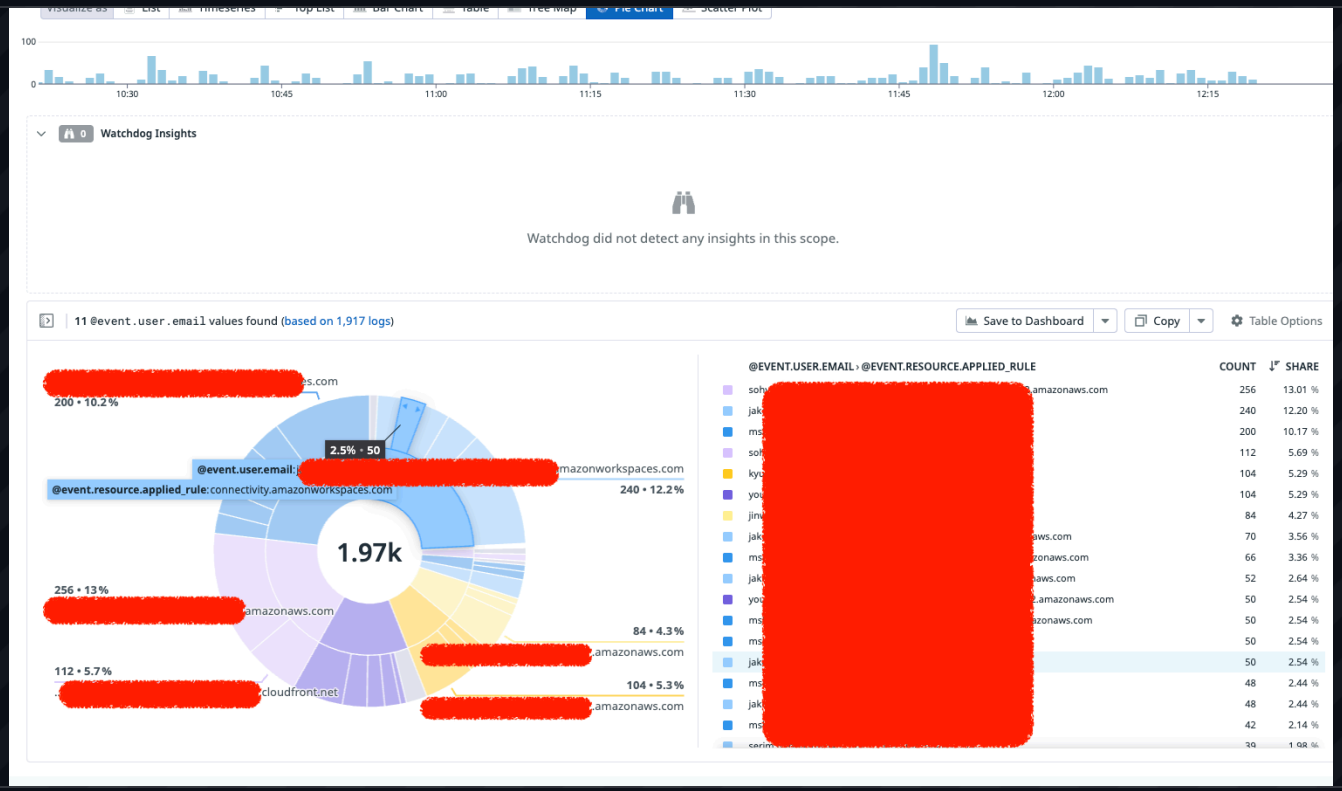
2GB RAM 모델 — SSL VPN (tunnel·web mode) 제거

docs.fortinet.com/document/fortigate/7.6.0/fortios-release-notes/877104/ssl-vpn-removed-from-2gb-ram-models-for-tunnel-and-web-mode

SSL VPN → IPsec VPN 마이그레이션 가이드 (7.6.3)

docs.fortinet.com/document/fortigate/7.6.0/new-features/155142/migration-from-ssl-vpn-tunnel-mode-to-ipsec-vpn-7-6-3

통제가 동작했다는 증거 — 1,917건의 접근



Twingate 접근 로그를 **Datadog**로 분석 · 사용자 이메일 × 적용 규칙(applied_rule)별 분포 · **이상징후 0**(Watchdog) · 민감 필드 블러 처리

접근 단위 감사 — 사용자 · 리소스 · 적용 규칙

Watchdog 이상탐지 상시 가동

접근 패턴 시계열 추적

inbound port 0 — 어떻게 가능한가

CLIENT
사용자 디바이스
인증된 사용자·장치. 리소스로 직접 향하지 않음

→

CONTROLLER
제어 평면
인증·권한·정책 결정 (data plane 불통과)

→

RELAY
중계
양측 아웃바운드 연결을 이어줌

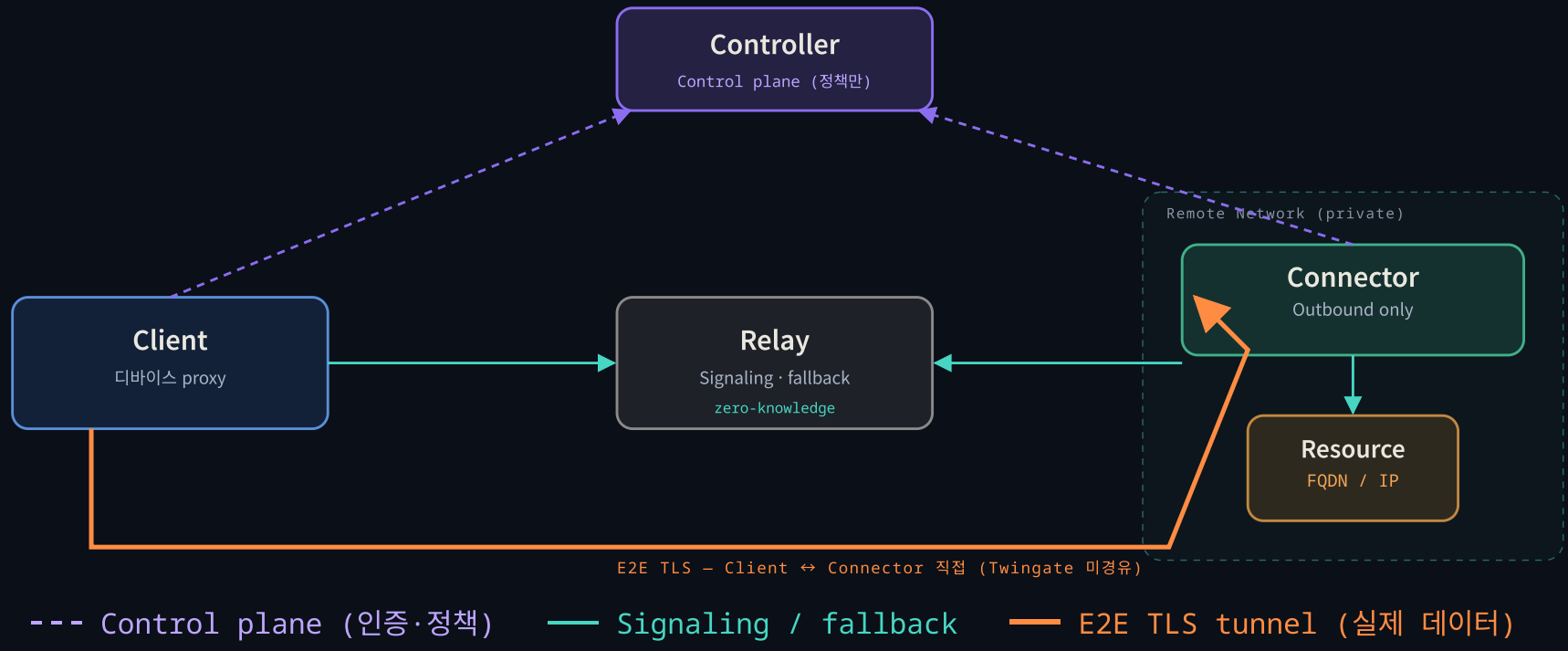
→

CONNECTOR
리소스 측
사내망에서 **아웃바운드**로만 연결 — 인바운드 포트 개방 0

`인바운드 오픈 포트 0개` `Client·Connector 모두 outbound-only` `단일 컴포넌트는 단독으로 트래픽 허용 결정 불가`

리소스(Connector)는 외부에서 향하는 포트를 열지 않는다. 양쪽이 각자 아웃바운드로 붙고, 트래픽은 기본적으로 **Client↔Connector P2P로 직접** 흐른다(Twingate 인프라 미경유). 인증 흐름과 데이터 흐름은 **별도 컴포넌트가 분리 검증**한다.
우리 배포 — Connector를 **ECS Fargate 2개 독립 서비스(2 AZ)로 HA**, private 서브넷에서 **egress-only**. 출처: Twingate Security Whitepaper (2022.8)

데이터는 Twingate 서버를 안 지난다



역할 분리가 핵심. Controller는 정책만, Relay는 신호·fallback만, Connector는 사내망에서 밖으로만. **실제 트래픽(주황)은 Twingate 서버를 안 지난다 — Client↔Connector 직접**. P2P가 막히는 경우(symmetric NAT 등)에만 Relay 경유 fallback이 일어나며, 이때도 Relay는 양 끝 E2E 암호화의 **봉인을 풀 수 없다**. 출처: Twingate Security Whitepaper (2022.8) · §단일 컴포넌트 단독 허용 결정 불가 / Relay zero-knowledge

A5 — OKTA SSO 연동 흐름

신원은 IdP가, 접근은 정책이

01
접근 시도
사용자가 보호 리소스에 접근

→

02 · OKTA
SAML/OIDC 인증
IdP에서 신원 + MFA 검증

→

03
그룹 → 정책 매핑
Okta 그룹이 접근 권한으로 연결

→

04
장치 검증 후 허용
device posture 확인 → 리소스별 인가

입·퇴사 = IdP 그룹 변경으로 자동 회수

단일 신원 소스

ISMS 2.5 인증·권한관리 충족

k8s 컨트롤러처럼, 5분마다 현실을 desired로

OKTA WF

어트리뷰트 변경 수신

프로파일 변경 → Lambda receiver(Function URL) 호출

→

DYNAMODB

desired state 저장

username 키 · TTL. 사용자별 원하는 상태

→

EVENTBRIDGE

rate(5 min)

주기적으로 reconciler 기동 — 선언적 루프

→

RECONCILER

AD · WorkSpaces 정렬

Scan → 비교 → Create/Terminate/Modify · 최대 10회 재시도 · SES 알림

```
# 원하는 상태를 선언 — 실제 구성은 루프가 맞춘다
{ "username": "hong.gildong", "state": "enabled", "tier": "standard" }
# → reconciler: WorkSpace 없으면 생성 · tier→bundle 매핑 · 디렉터리 등록
{ "username": "hong.gildong", "state": "disabled" }
# → reconciler: 프로비저닝됐던 리소스 회수(Terminate) — 가역
```

실제 구성 (`assistance/workspaces`): receiver·reconciler 모두 python3.12. 백엔드 디렉터리는 **AWS Managed Microsoft AD 단독**(M365/Entra 미연동), WorkSpaces는 **SAML·CBA** 연동. SSM Hybrid Activation 자동 갱신 rate(25 days), 미사용 WorkSpaces auto-stop(KST 20:00)으로 비용까지 코드로. `desired state =`

통제 충족을 증적 산출물로

통제항목	구현 (ZTNA)	제출한 증적 산출물
2.5 인증·권한관리	**Okta(SAML/OIDC)** + 그룹 정책. 입·퇴사 자동 회수	IdP 사용자·그룹 목록, 그룹 변경 이력, 퇴직자 회수 로그
2.6.1 네트워크 접근	**inbound port 0**, per-resource authorization	리소스·접근정책 정의(Terraform 코드), 보안그룹·노출포트 현황
2.6.2 정보시스템 접근	**device posture** + 정책을 코드로 관리	정책 변경 PR 이력, 장치 검증 로그, 미사용 계정 검토 결과
2.6.6 원격접근	ZTNA **단일 경로로 일원화**	원격 접근 단일 경로 구성도, 접근 감사 로그

핵심 — 증적이 운영의 부산물이다. 정책을 코드로 운영하니 심사 때 따로 모으지 않아도 PR 이력·접근 로그가 그대로 증적이 된다. 출처: KISA **ISMS-P 인증기준 안내서 (2022.4.22 정본)**, IT위키 2.5 / 2.6 접근통제.

접근 정책 = PR 리뷰 = 증적

```
# 리소스 단위 접근을 코드로 — 포트까지 최소권한
resource "twingate_resource" "rds" {
  name       = "*.rds.amazonaws.com"
  remote_network_id = local.remote_network_id
  protocols = { tcp = { policy = "RESTRICTED", ports = ["3306"] } } # MySQL만
  access_group { group_id = local.group_everyone_id }
}
```

실제 운영 구성. 리소스 39개 · 그룹 2개(Everyone / WorkSpaces)를 코드로 관리. Connector 토큰은 ECS에 **ARN만** 넘겨 state 미저장 · provider api_token은 secret_string 참조라 **state에 평문으로 남음** → **S3 backend 암호화·접근통제가 방어선**(개선: env 분리). 정책 변경 = PR = 증적.

`provider Twingate/twingate v4.2 · backend s3 (use_lockfile)`

A8 —— 도입 후 지표

무엇이 바뀌었나

TODO · 실수치 입력 필요

리허설 전 채울 항목

- ☐ 접근 관련 사고 / 비인가 접근 시도 — 도입 전 vs 후
- ☐ 신규 입사자 온보딩 시간 (VPN 설정 → ZTNA 클릭)
- ☐ ISMS 심사 대응 공수 (증적 수집 시간)
- ☐ 퇴사자 접근 회수 소요 시간 — 수동 → 자동
- ☐ 지원 티켓 / 문의 건수 (무매뉴얼 UX 효과)

완벽한 선택은 없다

의존성 **SaaS control plane**

- 제어 평면 장애 시 신규 세션 수립 영향 범위
- 가용성 SLA · 장애 시 운영 연속성 계획 필요

규제 **개인정보 · 국외이전**

- 위탁·국외이전 고지는 필요 — 다만 방어 근거가 분명
- 트래픽 **P2P 직접**(SaaS 미경유) · Relay 경유 시 **복호화 불가·미저장** · 비밀번호 **미보관**(IdP 위임)
- 계약상 **DPA·표준계약조항(SCC)** 적용 (약관 §6.1·6.2)

조화 **망분리와 SaaS**

- 망분리 정책과 SaaS 기반 접근통제의 양립 설명 부담
- 심사원에게 ZTNA 개념을 이해시키는 과정 필요

경계 **M365 AD 미연동**

- Okta↔Entra **이중 디렉터리 동기화**의 관리 복잡도·실패지점 회피
- 신원 단일 소스는 Okta 고정, AD는 WorkSpaces 도메인 조인 백엔드로만

상쇄 **그럼에도 선택한 이유**

- 물리장비 운영부담·매몰비용 제거

A10 —— 비용 — TCO 비교

우회로의 매몰비용 vs 구독

TODO · 실수치 입력 필요

채울 비용 항목

- ☐ 우회로 매몰비용 — UTM 장비 + 글로벌 방화벽 라이선스 (구매·연환)
- ☐ 구형 장비 잔존·교체 비용 (영업장 단위)
- ☐ Verified Access 검토 시 예상 사용량 과금
- ☐ Twingate 구독 TCO — 좌석/리소스 기준 연 비용
- ☐ 운영 인건비 — 물리장비 유지보수 시간 → ZTNA 코드 운영

A11 —— 클라우드 권한도 같은 신원으로

접근통제는 사내 리소스에서 멈추지 않았다

오늘 본편은 "사용자 → 사내 리소스" 접근에 집중했지만, 같은 원칙을 클라우드 권한에도 적용했다. AWS 콘솔·리소스 접근을 IAM Identity Center + Okta로 연동해 신원을 일원화했다 — 한 사람의 입·퇴사가 사내 리소스와 클라우드 권한 양쪽에서 한 번에 반영된다.

Okta · 단일 신원 → Twingate · 사내 리소스 → IAM Identity Center · AWS 권한

신원 소스 하나 — 회수도 한 번에

AWS 콘솔 · WorkSpaces 접근도 ZTNA 경로로

선택과 집중 — 오늘은 사내 리소스 중심