Control Tower로 시작하는 안전한 클라우드 여정

AWS 멀티계정, 이제는 통제 가능한 방식으로 이용하자

양보승

양보승, Boseung Yang

- AWS Professional Services
- Cloud Architect
- Security.....?

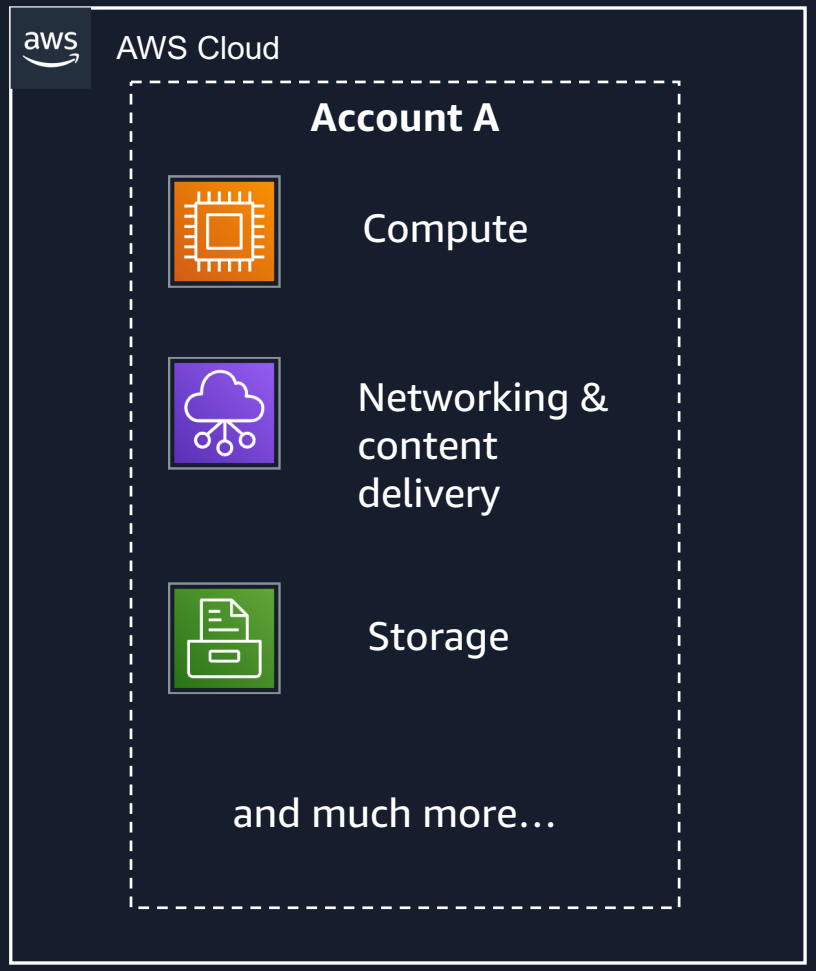
https://www.linkedin.com/in/bsyang/

AWS Account?

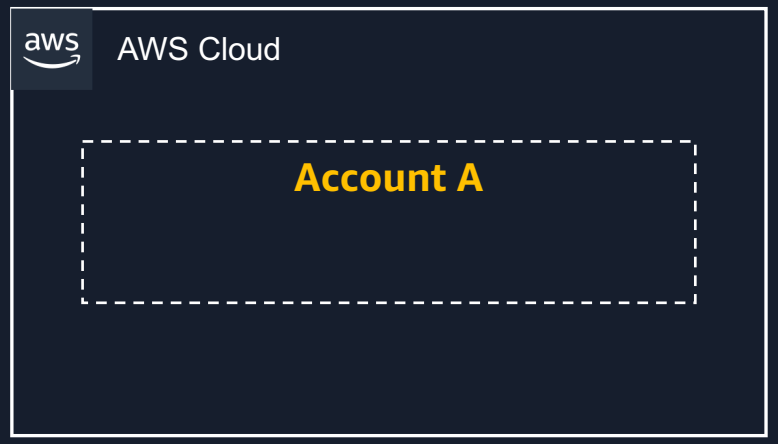
Each AWS Account:

- AWS 서비스의 **리소스 컨테이너**
- 명확한 **보안 경계**
- **비용 추적 및 청구**를 위한 컨테이너
- **서비스 한도 및 API 제한**과 같은 값을 관리하는 매커니즘

- 시간이 지남에 따라 더 많은 Application과 서비스를 위해 점점 더 많은 AWS 계정을 추가하게 됨



단일 Account

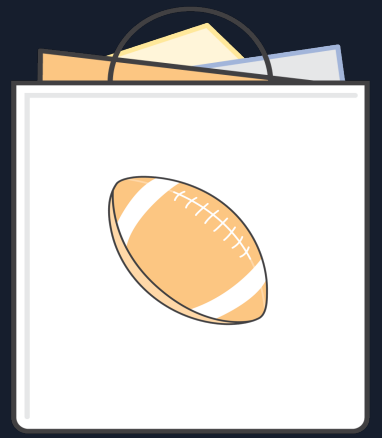


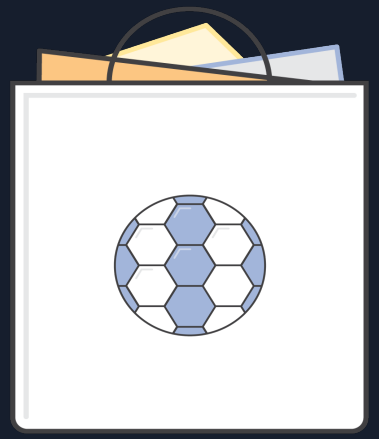
장점
- 단순한 운영 모델
- 소규모 운영팀에'만' 적합

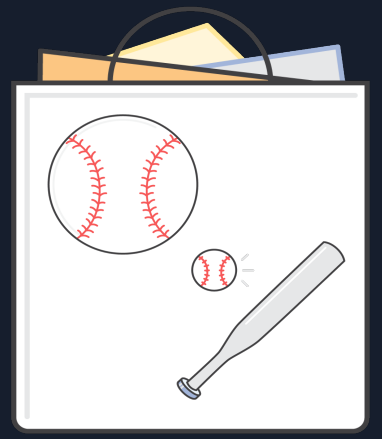
단점
- **확장성**이 제한됨
- 변경 및 위험 발생 시 **영향 범위가 매우 큼**
- 복잡한 보안 통제를 요구하게 됨
- 여러 비즈니스 조직에 **자율성 위임 어려움**
- **재무(청구) 가시성**을 명확하게 나누기 어려움
- 기타 여러 문제들...

몇 개의 AWS 계정을 사용하고 있나요?

Single Vs Multi







Multi Account는 어떨까?

관리 복잡성 증가

설정/정책 불일치

로그 분산

보안 취약점

Multi Account는 어떨까?

관리 복잡성 증가

AWS 계정 수가 증가할수록 관리의 복잡성 급증

로그 분산

분산 된 로그로 인한 보안 사고 대응 지연

설정/정책 불일치

계정마다 IAM 정책, VPC 구성이 제각각

보안 취약점

규모가 커질수록 보안 위험도 증가

"계정은 만들었는데 정책은 누가 넣죠?"

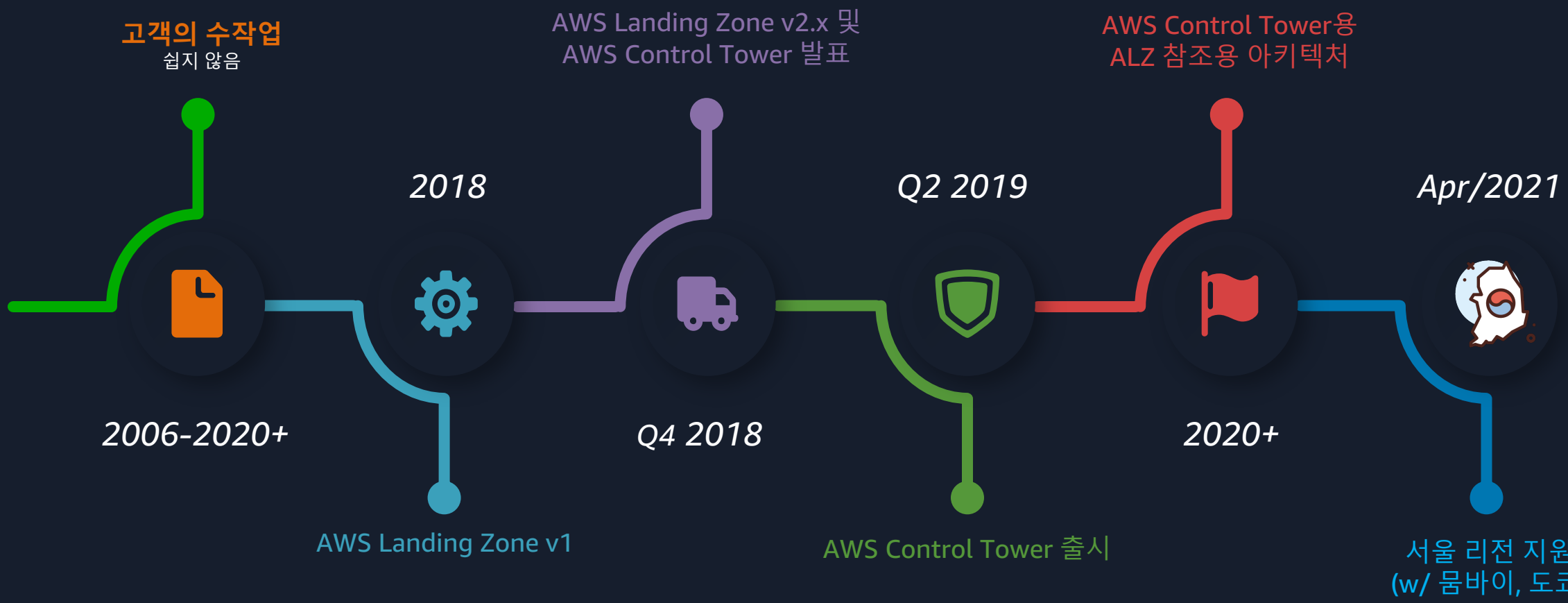
- 보안팀의 고민

"로그는 있는데 어디에 저장되어 있는지 모르겠어요..."

- 인프라팀의 고민

그래서 이 문제를 해결 할 방법은?

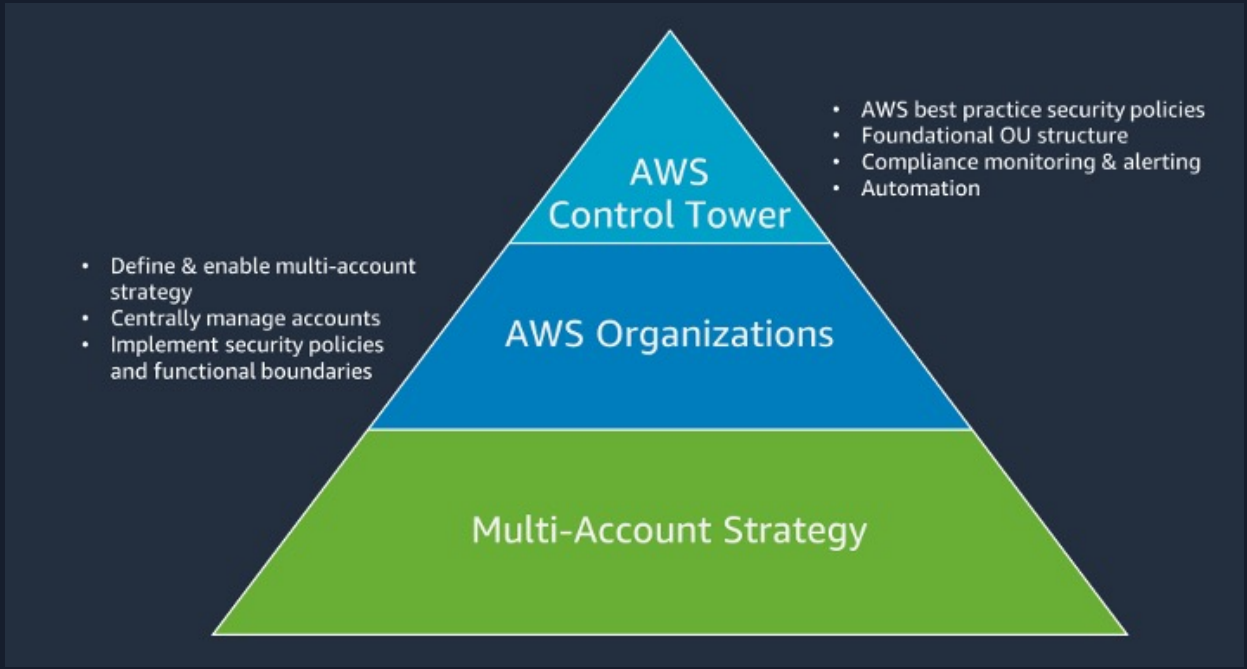
Landing Zone History



AWS Control Tower란?

멀티 계정 환경을 표준화&자동화하여 중앙에서 통제할 수 있도록 지원하는 **관리형 서비스**

복잡한 AWS 운영을 **'관제탑'**처럼, 가시성과 간소화 및 일관된 거버넌스 제공



Control Tower가 어떻게 해결을...?

계정 생성 자동화

- 수동 설정 번거로움 감소
- 미리 정의된 템플릿 기반 자동 프로비저닝
- 개발팀이 필요한 계정을 신속하게 확보 가능

보안 정책 사전 적용

- 계정 생성 단계부터 보안 정책 자동 적용
- 보안 설정 누락 방지
- AWS 모범 사례 기반 강력한 보안 태세 유지

로그 중앙화

- 모든 계정의 Cloudtrail 로그 자동 수집
- 포렌식 조사 및 감사 대응 능력 향상
- 로그 누락 및 조작 위험 감소

SSO 접근 제어 간소화

- AWS IAM Identity Center 통합
- 단일 ID로 여러 AWS 계정에 안전한 접근
- IAM 사용자 관리의 복잡성 감소

Organization

AWS Multi Account 환경에서 AWS Account 간 중앙 통제 및 관리하는 서비스

AWS Account 및 리소스를 중앙에서 프로비저닝

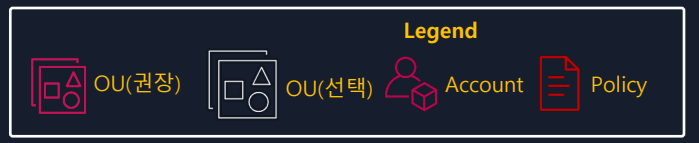
규정 준수를 위해 AWS 환경을 보호하고 감사

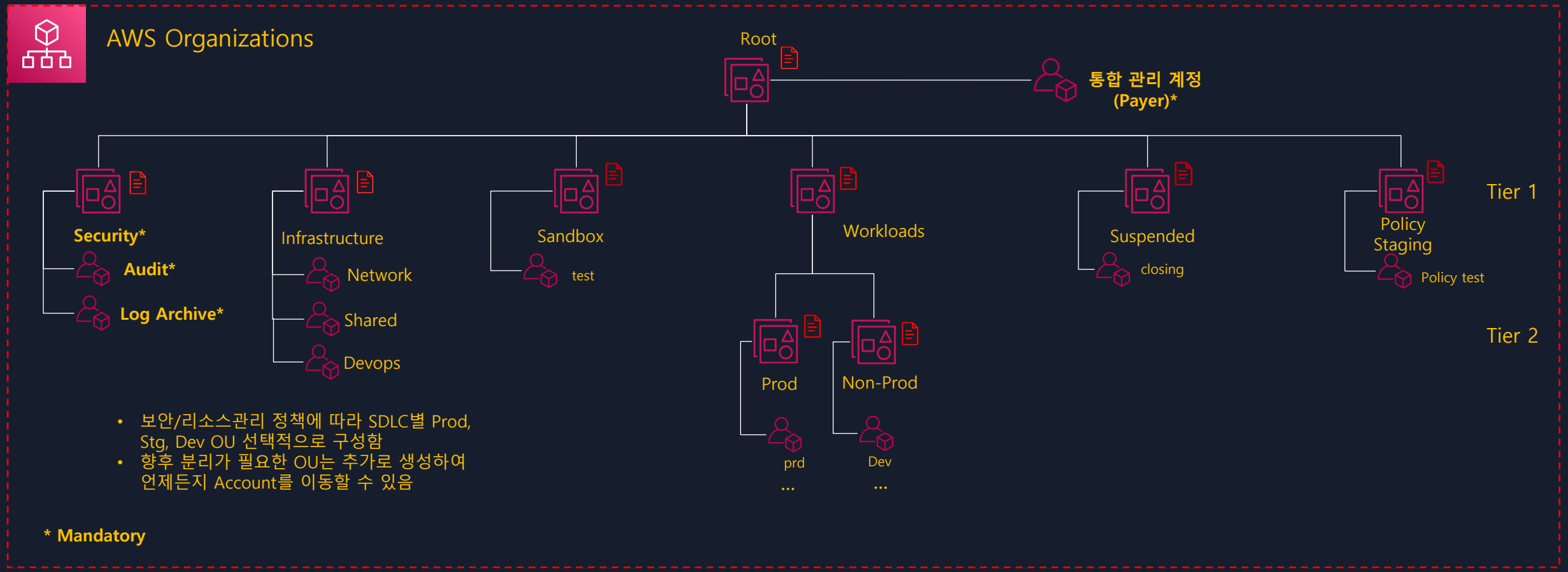
리소스를 공유하고 액세스 관리를 간소화

비용 관리 및 비용 절감 조치를 식별

OU(Organizational Units)

AWS Account을 그룹화하는 리소스





Guardrails

AWS Control Tower 환경에서 보안 및 컴플라이언스 요구사항에 따라 선택하고 적용할 수 있는 정책

예방형 Preventive Guardrails

서비스 제어 정책(SCP)을 통해 구현되며, 특정 규칙을 위반하는 리소스 생성을 사전에 방지

예시 -
퍼블릭 S3 버킷 생성 차단
Internet Gateway 생성 차단

탐지형 Detective Guardrails

AWS Config 규칙을 통해 구현되며, 이미 배포된 리소스가 설정된 규칙을 위반하는지 지속적으로 모니터링

예시 -
퍼블릭 쓰기/읽기 액세스가 허용된 S3 버킷 탐지
0.0.0.0/0 허용 정책 탐지

Account Factory

표준화된 템플릿을 기반으로 AWS 계정을 자동으로 프로비저닝하는 기능
템플릿에는 네트워크 구성, 보안 규칙 등이 포함되어 있어, 계정 생성 시 사전 정의한 표준 설정 자동 적용

자동화된 계정 생성

템플릿 기반으로 계정 생성 프로세스를
가속화하고 수동 작업을 최소화

보안 정책 사전 적용

새로운 계정에 Guardrails가 즉시 적용 되어
거버넌스 정책이 자동 준수

환경 일관성 확보

모든 계정의 일관성을 보장하여,
반복적 작업을 줄이고 운영 효율성 향상

Centralized Logging

여러 계정의 로그를 한 곳에 모아 보관·관리하는 기능으로, 보안 모니터링과 컴플라이언스 강화를 위해 로그를 중앙 로그 계정에 자동 집계함.

분석 향상

중앙 집중식으로 관리하여 보안 사고 발생 시 분석을 효율적으로 수행

위험 감소

로그 누락 및 조작 위험을 0%에 가깝게 줄이고, 모든 로그를 안전하게 보관

감사 대응 향상

규정 준수 감사를 위한 통합 로그 제공, 감사 속도 및 효율성 획기적 향상

IAM Identity Center (SSO)

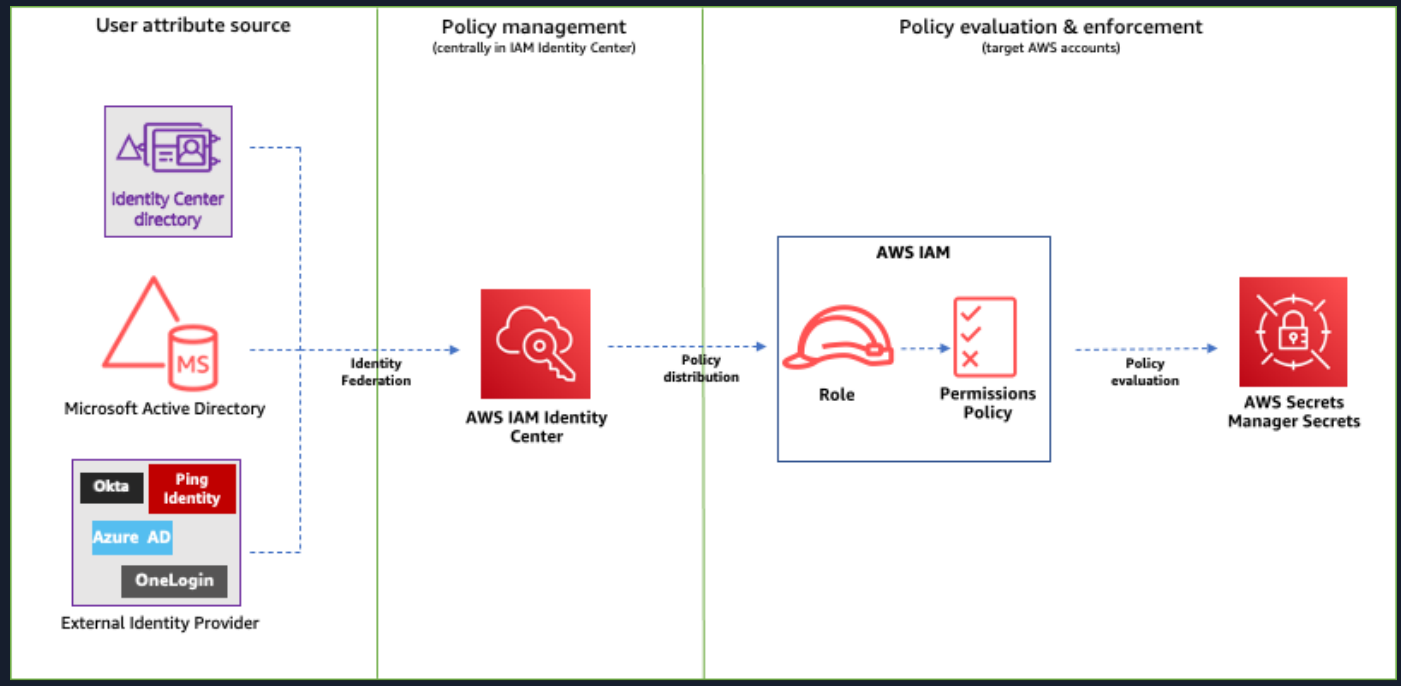
여러 AWS 계정과 애플리케이션의 접근 권한을 중앙에서 통합 관리하는 서비스로, 사용자 인증·권한 부여를 표준화해 보안성과 운영 효율성을 높임

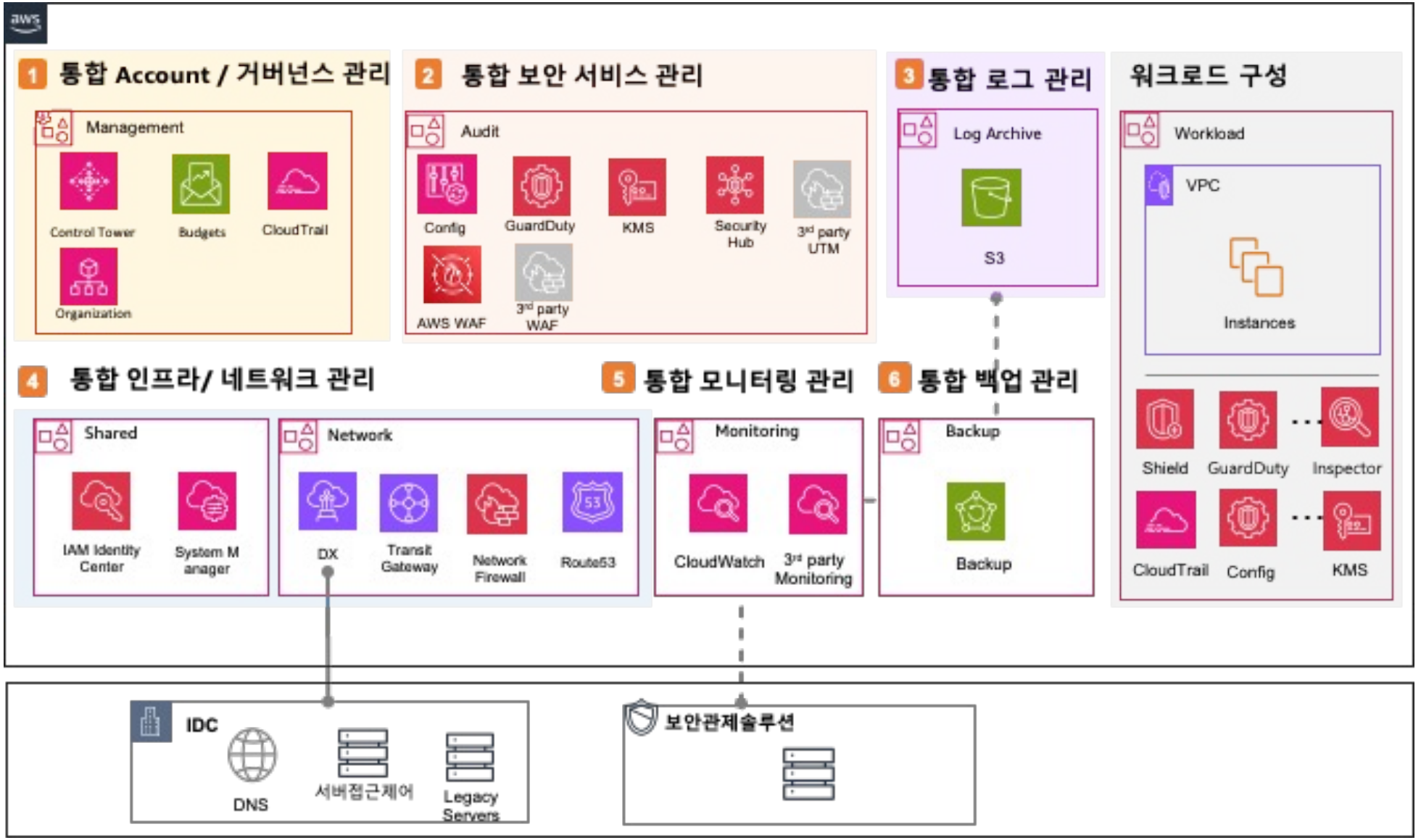
조직 단위 인증 통합

Control Tower와 IAM Identity Center를 통합하여 단일 로그인으로 여러 AWS 계정에 접근

계정 접근 제어 간소화

IAM 사용자 관리의 복잡성을 줄이고, 중앙에서 사용자 및 권한을 효율적으로 관리





Control Tower을 꼭 ..?

빠른 초기 설정

사전 통제 기반 보안 강화

커스터마이징 제약

작은 조직엔 과할 수도 있음

기존 계정 이관 난이도 높음

운영·보안팀 협업 효율 증가

요약하면,

계정 자동화

템플릿 기반 계정 생성으로
일관된 환경 제공

보안 정책 강제

Guardrails로 보안 및 규정 준수 유지

거버넌스 중앙 집중

중앙에서 정책 관리로 일관성 보장

FAQ

도입 시 가장 고민되는 점은 무엇인가요?

Control Tower 도입 시 OU 구조 설계, Guardrails 선택, 기존 계정 이관 등 여러 고려사항이 있습니다.

OU 구조를 어떻게 설계하나요?

비즈니스 요구사항과 보안 정책에 맞춰 OU 구조를 사전에 면밀히 설계해야 합니다.

기존 계정 이관 시 주의사항은 무엇인가요?

기존 계정의 설정과 Control Tower의 Guardrails 간에 정책 충돌이 발생할 수 있습니다.

Q&A

감사합니다.