aws certified Solutions Architect ASSOCIATE

이준형

보안 · 모니터링 · 관측

세부 구현 내용

1 보안 & 감사

CloudTrail 감사 로깅 · 실시간 Teams 알림 3종 · WAF 탐지 · 비용 경보 · 멀티클라우드 DR/SSO

2 인프라 모니터링

Prometheus·Grafana 수집 · Alertmanager 알림 4종 · 리소스 대시보드 · HTTPS 접속

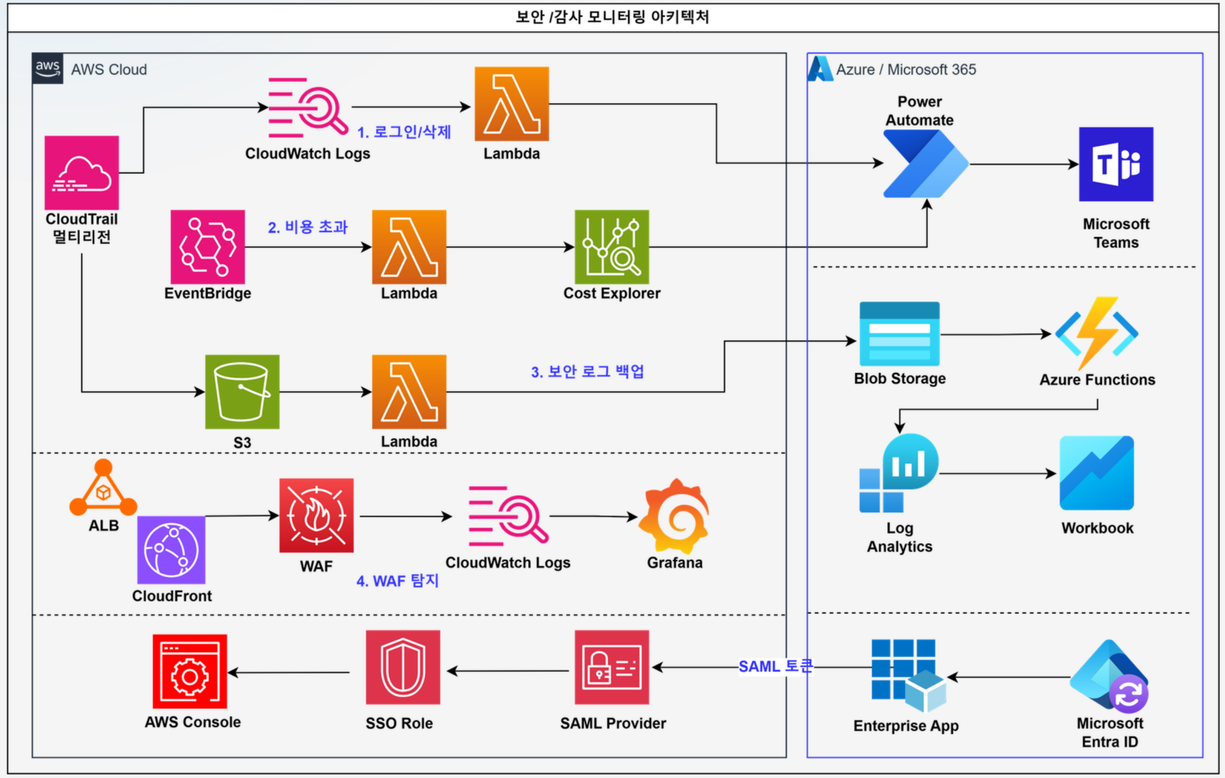
3 애플리케이션 관측

로그·메트릭·추적(X-Ray) 3축 · 멀티리전 · IoT 파이프라인

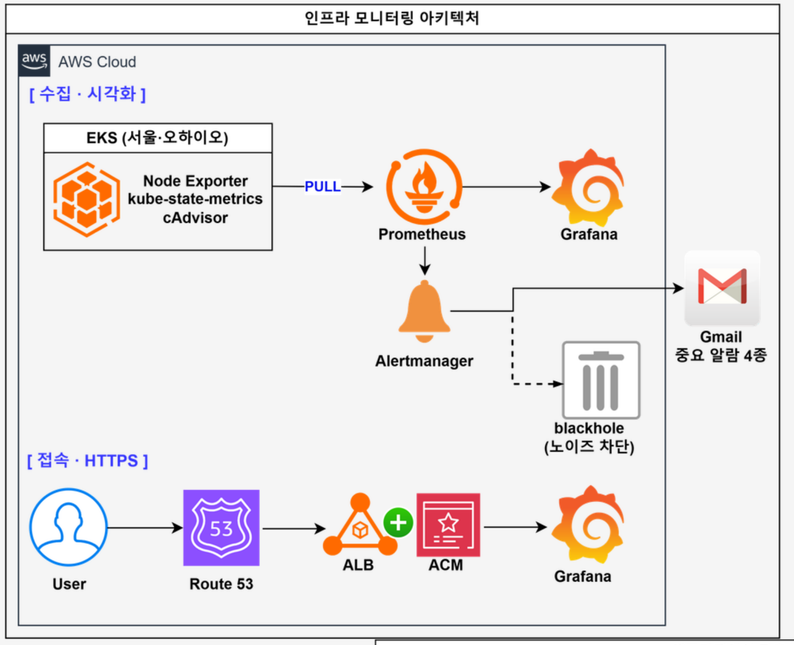
사용 서비스 및 기술

🔐 보안	CloudTrail · CloudWatch · S3 · WAF · Lambda · EventBridge · Cost Explorer · Power Automate · Teams · IAM SAML
📊 모니터링	Prometheus · Grafana · Alertmanager · Helm · IRSA
🔍 관측	Fluent Bit · OpenTelemetry(ADOT) · X-Ray · IoT Core · Firehose · Glue · Athena
☁️ 멀티클라우드	Azure (Blob · Functions · Log Analytics · Entra ID SSO)
⚙️ 공통	EKS · Terraform · Route 53 · ACM · S3 Backend · 멀티리전

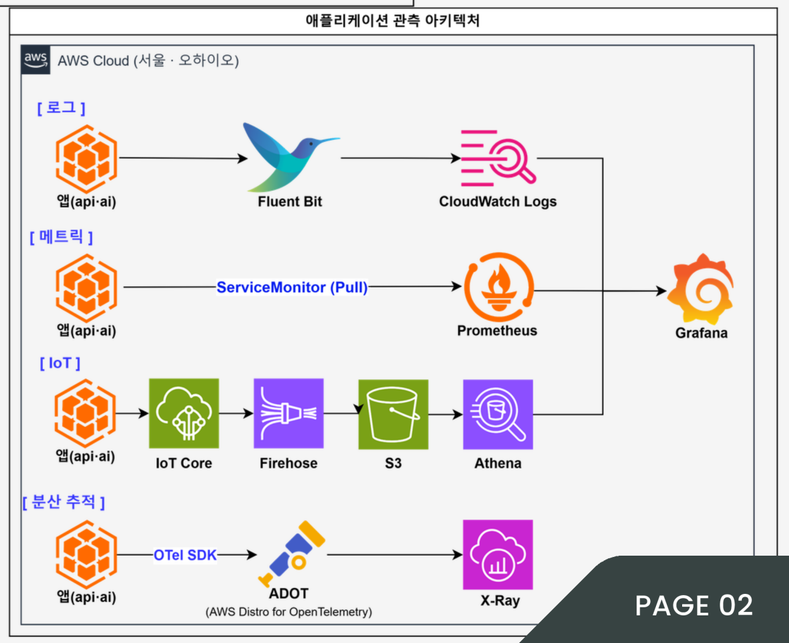
1 보안 & 감사



2 인프라 모니터링



3 애플리케이션 관측



멀티클라우드 계정 체계

AWS 계정: IAM Identity Center

- 장기 액세스 키 → 임시 자격증명 (AWS 권장사항)
- root·IAM User 키 발급 없이 SSO 단일 진입
- 팀원 4명 권한을 코드(IaC)로 중앙 관리

Azure 계정: Entra ID 연동

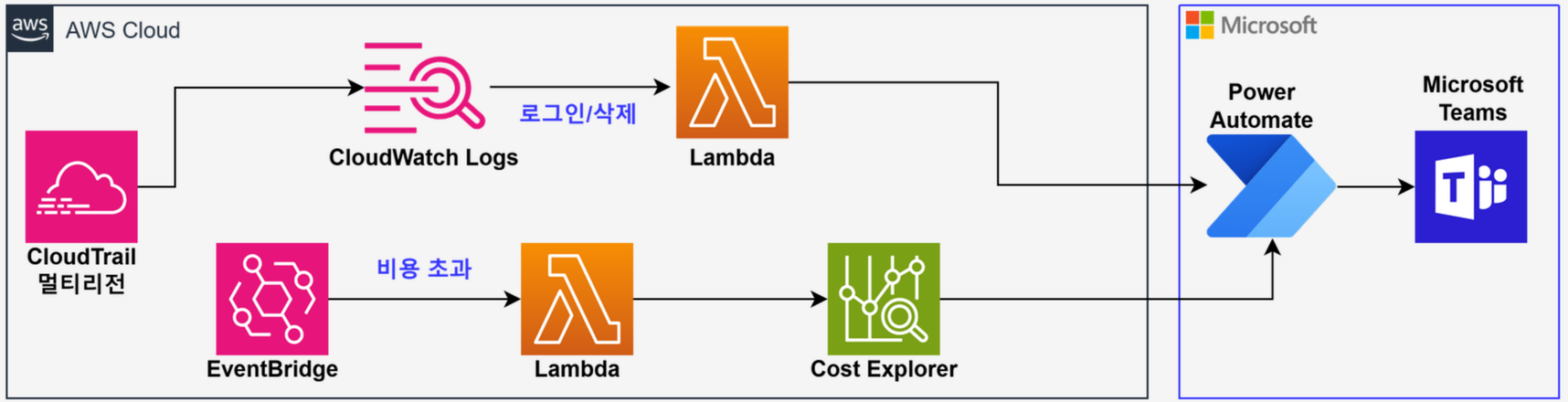
- 알림(Teams·Power Automate)이 이미 M365 기반 → Entra ID로 자연스럽게 통합
- 별도 인증 체계 추가 없이 기존 Azure 자산 활용
- 사용자·디렉터리를 Entra ID에서 일원 관리

멀티클라우드 SSO 통합

- Azure 계정 하나로 AWS 콘솔 단일 로그인
- 퇴사자·권한 변경을 중앙에서 즉시 통제 (운영 확장성 고려)
- 자격증명 분산 없는 중앙 관리 (연동 상세 → PAGE 06)

실시간 감사 알림

로그인·삭제·비용 알림 흐름



01

Lambda & Power Automate

⚡ Lambda

→ CloudTrail 로그 파싱 (사용자·IP·시각 추출)

→ 이벤트 감지 3종 (로그인/삭제/비용)

→ AWS 서비스 노이즈 필터링 (destroy 등 자동작업 제외)

함수 이름
siseon-lambda-billing-alert
siseon-lambda-delete-alert
siseon-l### Power Automate

이름
aws-billings
aws-alerts
aws-logins

Power Automate

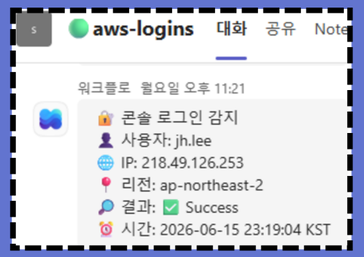
→ Webhook 중계 (Lambda → Teams)

→ 채널별 분배 (aws-logins/alerts/billing)

→ 메시지 동적 렌더링

02

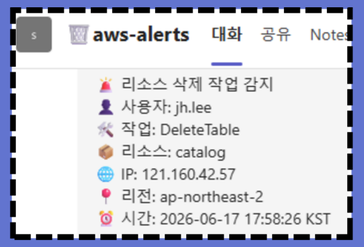
Teams 채팅



1

채널별 분리 발송

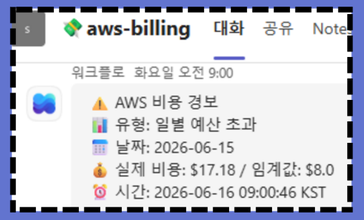
→ 로그인/삭제/비용



2

실시간

→ 이벤트 발생 즉시



3

상세 정보 파싱

→ 사용자 · IP · 시각

03

billing 구성

```python
import json
import urllib.request
import urllib.error
import os
import boto3
from datetime import datetime, timezone, timedelta

WEBHOOK_URL = os.environ["TEAMS_WEBHOOK_URL"]
DAILY_THRESHOLD = 8.0
MONTHLY_THRESHOLD = 135.0
KST = timezone(timedelta(hours=9))

ce = boto3.client("ce", region_name="us-east-1")

def get_cost(granularity, start, end):
    response = ce.get_cost_and_usage(
        TimePeriod={"Start": start, "End": end},
        Granularity=granularity,
        Metrics=["UnblendedCost"]
    )
    return float(response["ResultsByTime"][0]["Total"]["UnblendedCost"]["Amount"])
```

📅 EventBridge — 매일 KST 09:00 스케줄

🔍 Cost Explorer — 전일 비용 조회

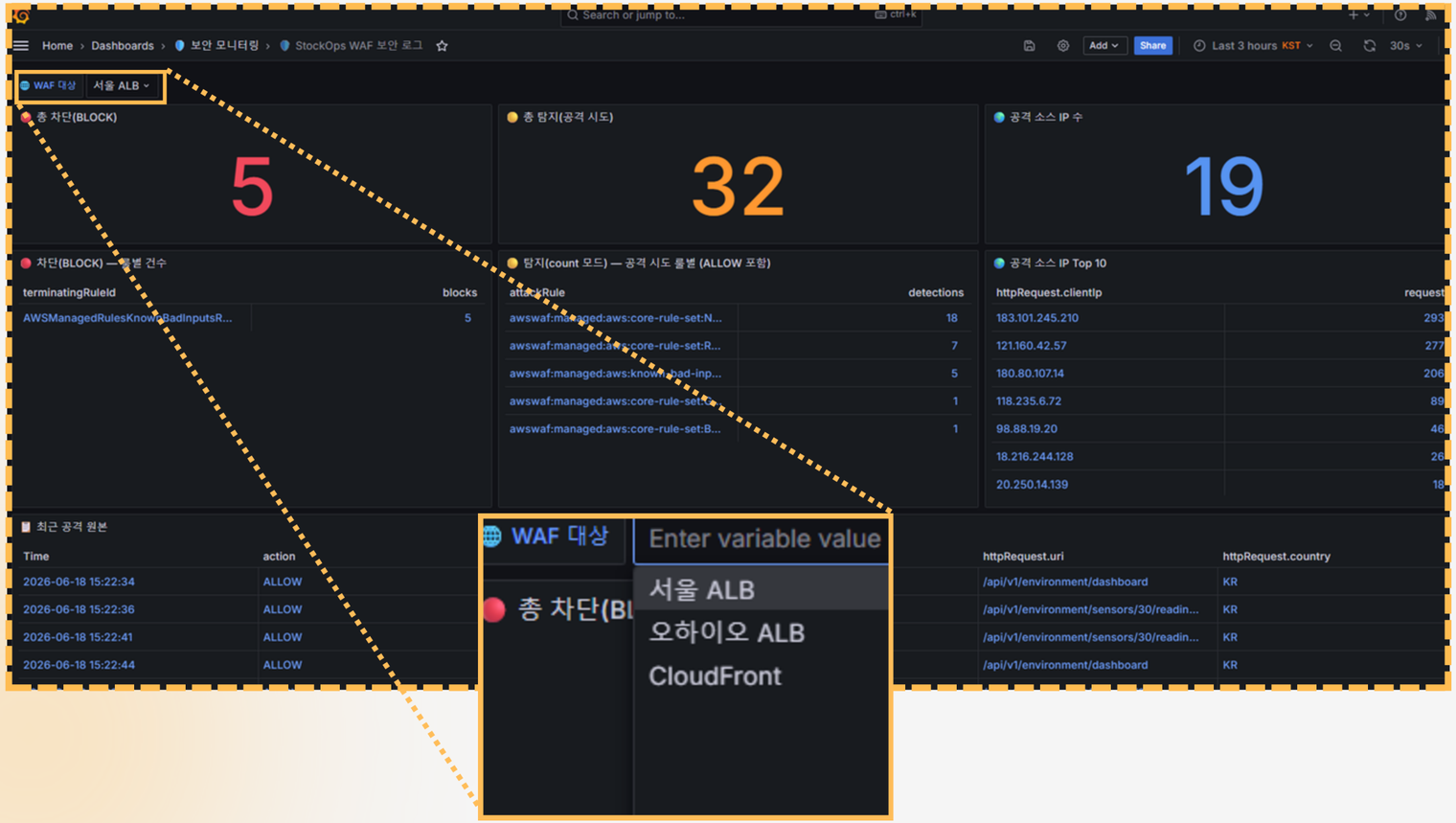
⚠️ 임계값 — 일 $8 / 월 $135 초과 시 알림

🔄 Alarm/Budgets 한계

"상태 변경 시에만" 발송 → 능동 폴링으로 해결

멀티리전 WAF 통합 관제

WAF 로그 수집 흐름



Point 01

🛡️ 핵심 설계: 차단(BLOCK)과 탐지(count) 분리

운영 안전을 위해 대부분 룰을 count 모드로 운영. 단순 BLOCK 집계 시 SQLi·XSS 공격이 화면에서 누락되는 함정 → awswaf:managed 라벨을 파싱해 ALLOW로 통과된 공격 시도까지 포착

Point 02

🌐 멀티리전: 드롭다운 하나로 3개 WAF

서울 ALB / 오하이오 ALB / CloudFront — 리전·로그그룹 다른 3개를 단일 대시보드에서 전환

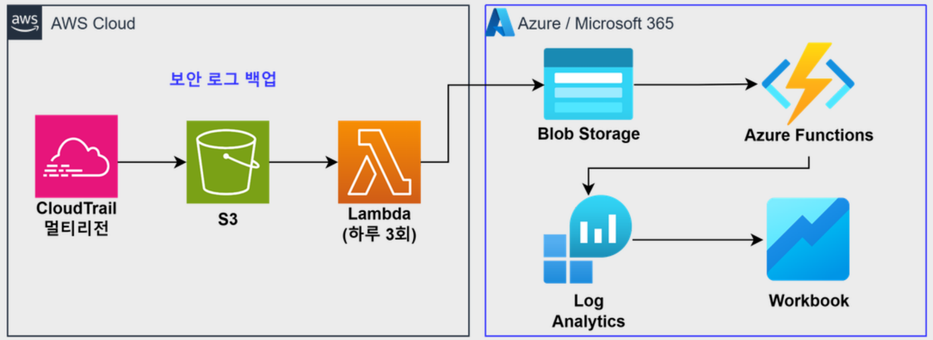
Point 03

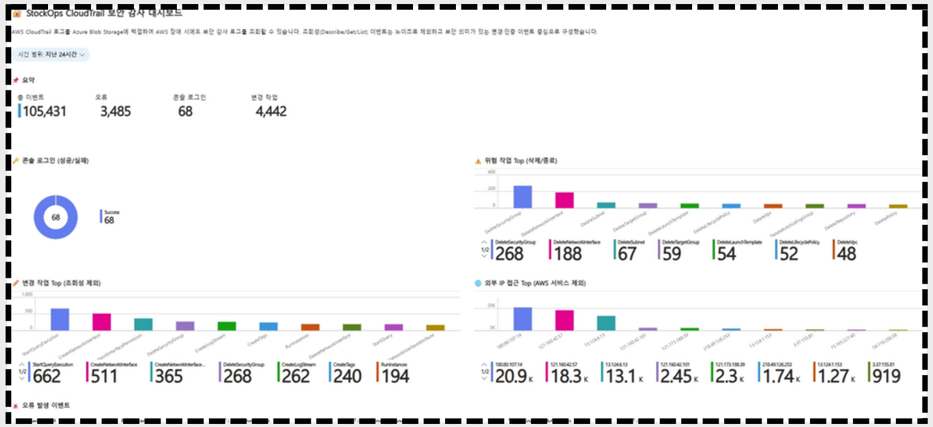
🎯 실제 공격으로 검증

외부에서 유입된 phpunit 스캐너 봇이 KnownBadInputs 룰에 BLOCK으로 차단되는 실제 공격을 포착. 직접 날린 SQLi·XSS 패턴으로 탐지 로직까지 검증

AWS↔Azure 백업 · SSO 연동

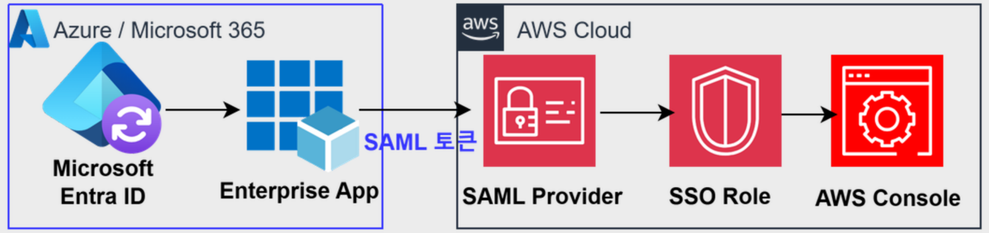
☁️ 재해복구 — CloudTrail 로그 Azure 백업





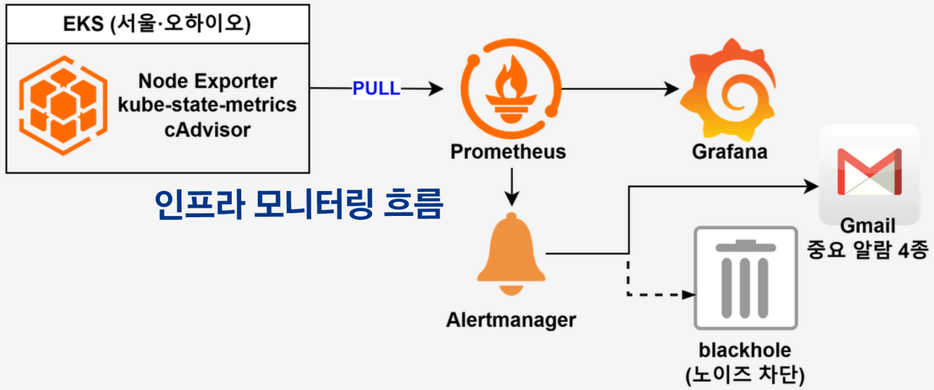
- CloudTrail 감사 로그를 Azure Blob에 백업 → AWS 장애 시에도 감사 지속
- 하루 3회 동기화 — 백업 공백 최대 8시간
- Azure Monitor Workbook으로 페일오버 시각화

🔐 통합 인증 — Entra ID SAML Federation



- AWS 콘솔 로그인을 Azure Entra ID로 페더레이션
- 별도 IAM 사용자 없이 Azure 계정으로 AWS 접근
- 자격증명 일원화 -> 퇴사자 차단·권한 회수를 Entra ID 한 곳에서

Prometheus 기반 인프라 모니터링



인프라 모니터링 흐름

템플릿 없이 직접 설계

- Grafana 공식 템플릿 그대로 X
- EKS 환경에 맞춰 패널 직접 구성
- 대시보드 정의를 Terraform 코드로 관리 → GitOps

IRSA 기반 권한

- Grafana → AWS 리소스(Athena 등) 접근
- 액세스 키 없이 IRSA로
- "키 없이 역할 기반" 보안 원칙 적용

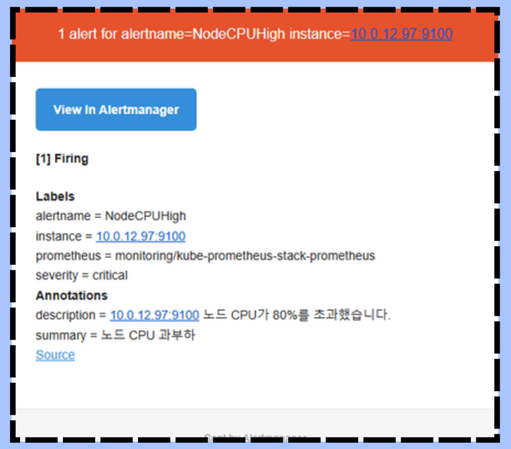
직접 구성한 핵심 패널

- 노드별 CPU 사용률 (Karpenter 오토스케일링 검증)
- 노드별 메모리 사용률 (로그 스케일 — 파드·노드 편차 흡수)
- 노드별 네트워크 I/O (트래픽 분산 확인)
- 파드 상태 / 재시작 현황 (OOM·크래시 조기 감지)
- 노드 범례 테이블 (다수 노드 한눈에)

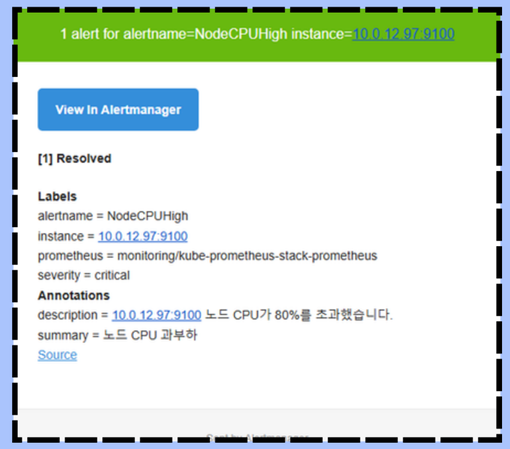
알림 노이즈 제어 · HTTPS 보안

🔔 Alertmanager — 노이즈 억제 알림

Firing



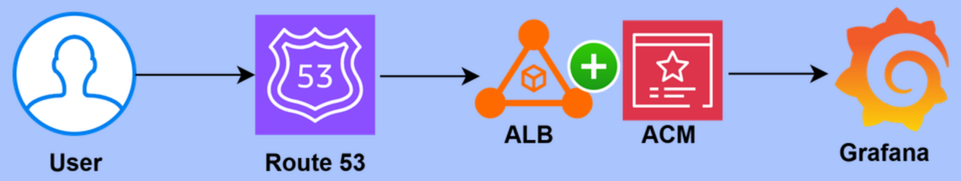
Resolved



- 중요 4종만 Gmail 발송 (Pod장애/재시작/노드CPU/노드메모리)
- blackhole 라우팅으로 나머지 노이즈 차단 → 화이트리스트 방식
- Firing → Resolved 상태까지 추적 (발생·해소 둘 다 알림)

🔒 Grafana HTTPS — 도메인 + 인증서

[접속 · HTTPS]



https://grafana.siseon.live

- grafana.siseon.live 도메인 (Route 53)
- ACM 와일드카드 인증서 (*.siseon.live) → HTTPS 암호화
- NLB → ALB Ingress 전환 + Route 53 alias 자동 연결

센서 데이터 분석 · 시각화

📡 MQTT 센서 수집

- 창고 센서 7종 → MQTT → IoT Core
- IoT Rule이 토픽 와일드카드 구독·라우팅 (+/sensors/+)

🗄️ 서버리스 분석 파이프라인

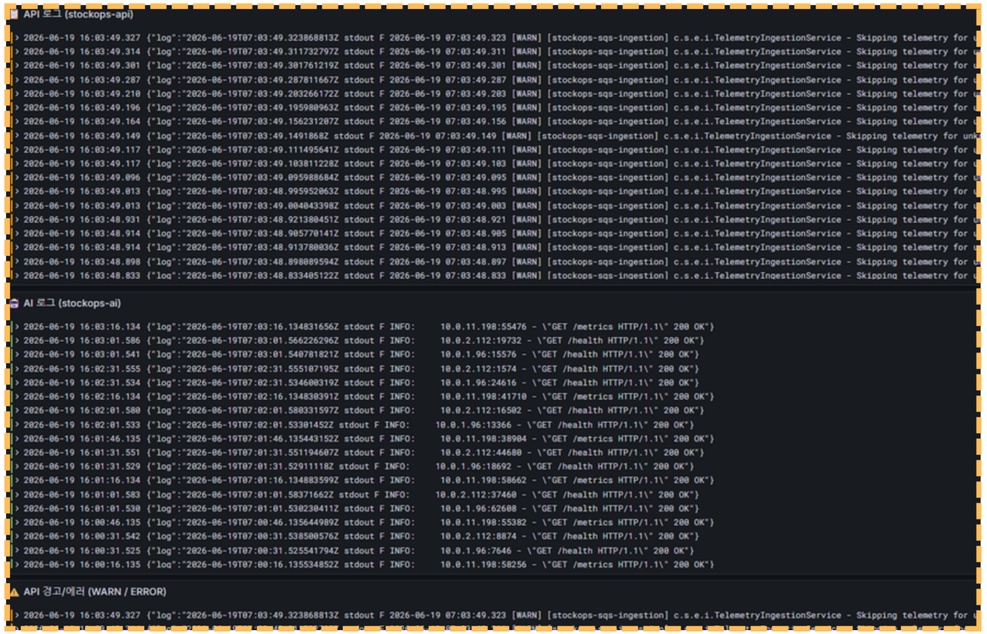
- Firehose → S3(Hive 파티션) → Glue 카탈로그 → Athena
- 별도 DB·서버 없이 S3 직접 쿼리

⚡ 파티션 프로젝션 최적화

- year/month/day 파티션 자동 인식
- 쿼리 시 해당 날짜만 스캔
  - 스캔량·비용 최소화

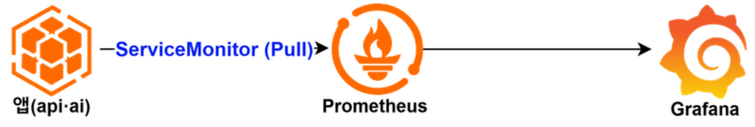
애플리케이션 로그 · 메트릭 관측

로그 : Fluent Bit → CloudWatch



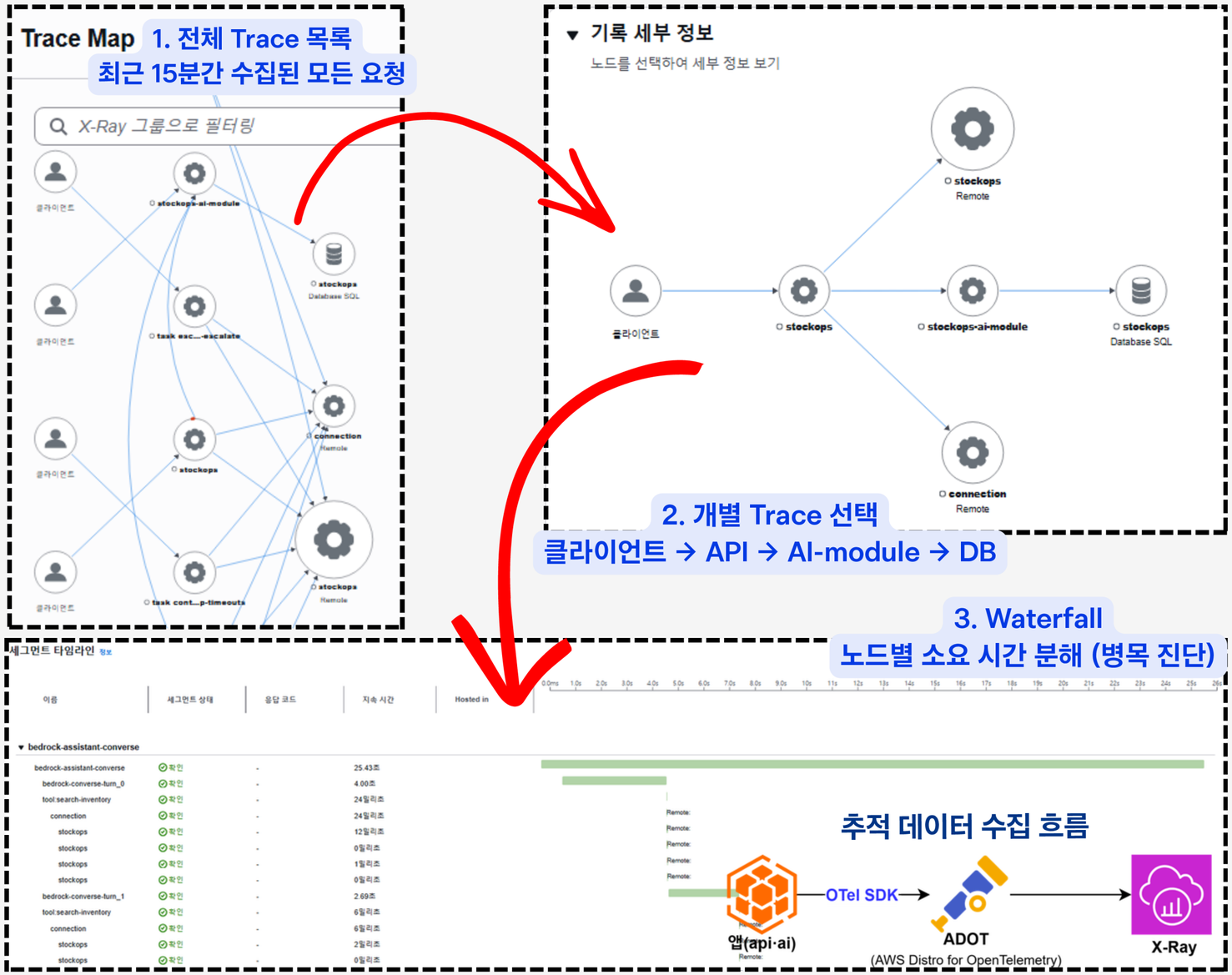
- 로그그룹 선택·쿼리 없이 검색어·레벨 필터로 즉시 조회
- 서울·오하이오 멀티리전 통합 (드롭다운)
- Fluent Bit DaemonSet + IRSA (노드별 수집, K8s 메타데이터 부착)

메트릭 : ServiceMonitor→Prometheus



- api 6 + ai 4 패널 (ServiceMonitor 기반 Pull 수집)
- api=summary(평균) / ai=histogram(p95 분위수)
- ai 예측 캐시 적중률 ~97%
- 로그=Push, 메트릭=Pull (수집 방향 분리)
- Bedrock 회로차단기·DB 커넥션풀 등 앱 내부 지표까지

LLM 호출 분산 추적 (X-Ray)



🔭 앱 코드 레벨 OTEL 계측

- api (Spring): micrometer-tracing-bridge-otel로 OTel 브리지 + @Observed 커스텀 span
- ai (FastAPI): OpenTelemetry SDK + Instrumentor로 HTTP·DB 자동 계측
- sampling 1.0 전량 수집 → ADOT Collector → X-Ray

🤖 LLM 에이전트 호출 체인 추적

- 질문 1건 → Bedrock 멀티턴 → forecast 툴 → prophet 예측 → RDS
- 클라이언트 → api → ai-module → DB 단일 trace 4노드
- 핵심: 계측된 HTTP 클라이언트가 W3C traceparent를 전파 → 코드로 안 이어도 서비스 경계 자동 연결

⏱️ 구간별 지연 분해

- Waterfall로 각 구간(보안필터·LLM 추론·prophet·DB) 소요시간 분리
- 어느 구간이 병목인지 정량 확인 — 실측 병목은 Bedrock LLM 추론
- 동기 호출 체인이라 구간별 시간이 누적되는 구조까지 가시화

트러블슈팅

영역	문제	원인 진단	해결
보안	일부 로그인이 알림 없이 누락됨	CloudWatch Alarm은 상태 변경 시에만 발송 → (이미 ALARM이면 무시)	Subscription Filter → Lambda 이벤트 방식 전환
인프라	백엔드(API) 메트릭만 No data (AI는 정상)	대시보드 아닌 Prometheus 수집 차단 → 인증에 막혀 스크랩 거부 (재배포 시 설정 누락)	메트릭 공개 설정 주입 → Prometheus 타겟 정상화, 패널 복원
관측	API 호출은 200 성공인데 X-Ray 추적 선이 안 생김	성공 응답 ≠ 정상 동작 → 연동·DB·데이터 3단 막혀 AI 호출 폴백	연동 env + 시드 데이터 → 실호출 복원, 추적 선 생성

프로젝트 핵심 성과

1 📦 전 구간 IaC

EKS·대시보드·알림·추적까지 Terraform으로 코드화하고 GitOps로 관리. 수작업 없이 재현 가능한 인프라 구성

2 🔭 Observability 3축 직접 구축

로그·메트릭·추적을 공식 템플릿 없이 SDK 계측부터 대시보드 시각화까지 직접 구성. 모놀리식 앱의 AI 호출까지 분산 추적

3 🌐 멀티리전·멀티클라우드 운영

서울·오하이오 크로스리전 피어링으로 로그·WAF 통합 + AWS 감사로그를 Azure로 백업하는 멀티클라우드 DR 구성