(Access Rights for Identity on AWS - graph visualization)

ARIA-gv 워크플로 총정리

실행 목적, 주요 단계 및 도출 결과 분석

https://github.com/aws-samples/sample-visualizing-access-rights-for-identity-on-aws

ARIA-gv 솔루션 A to Z
기반 아키텍처 상세 구조화

도입 배경

이것이 왜 필요한가

What kind of questions are we getting?

Internal Audit & Compliance

"Who in our company can access our cloud resources and what can they do to them?"

External Auditors & Regulators

"Can you show me how Bob was able to update the customer data in our production account?"

Senior Stakeholders

"Do users with access to our cloud resources have access rights that follow least privilege?"

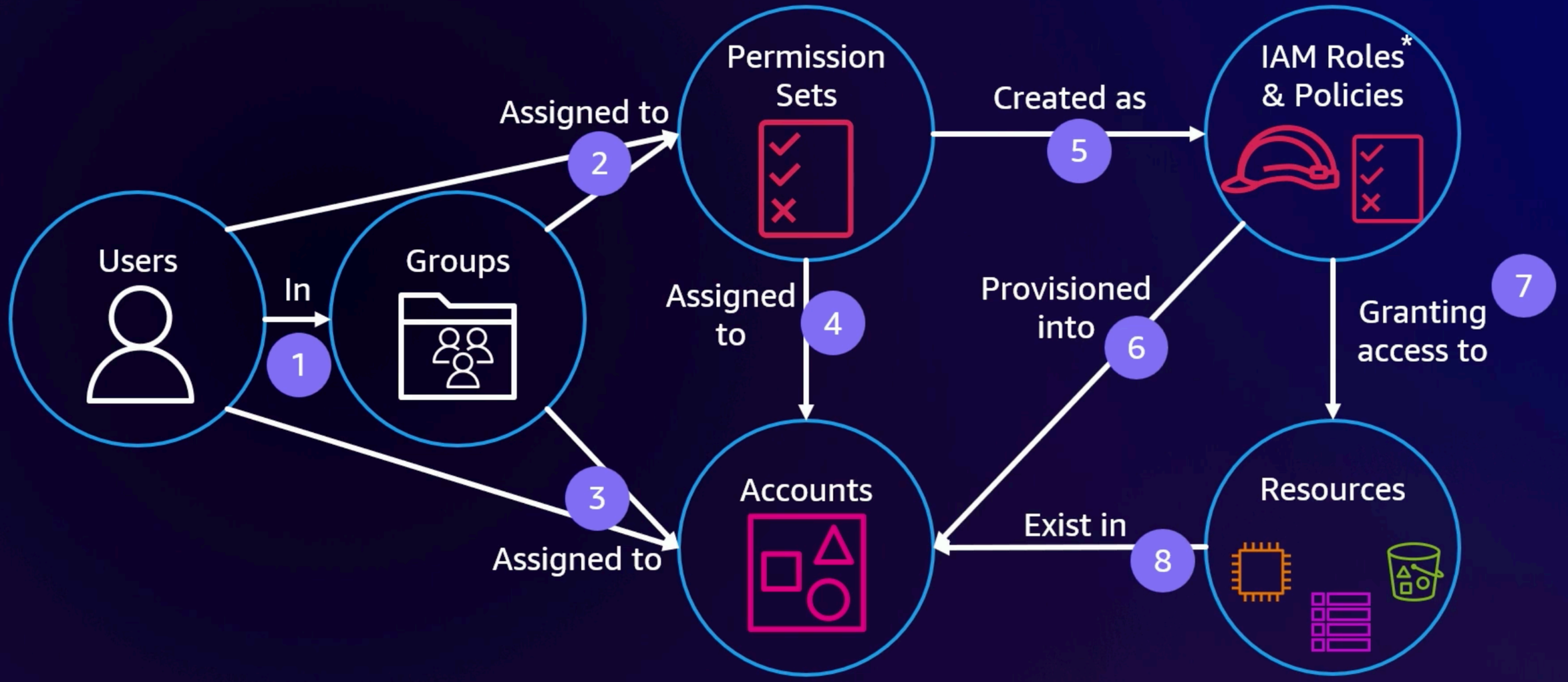
"Can you give me a report of everything that Alice has access to in our production account?"

Identity Administrators

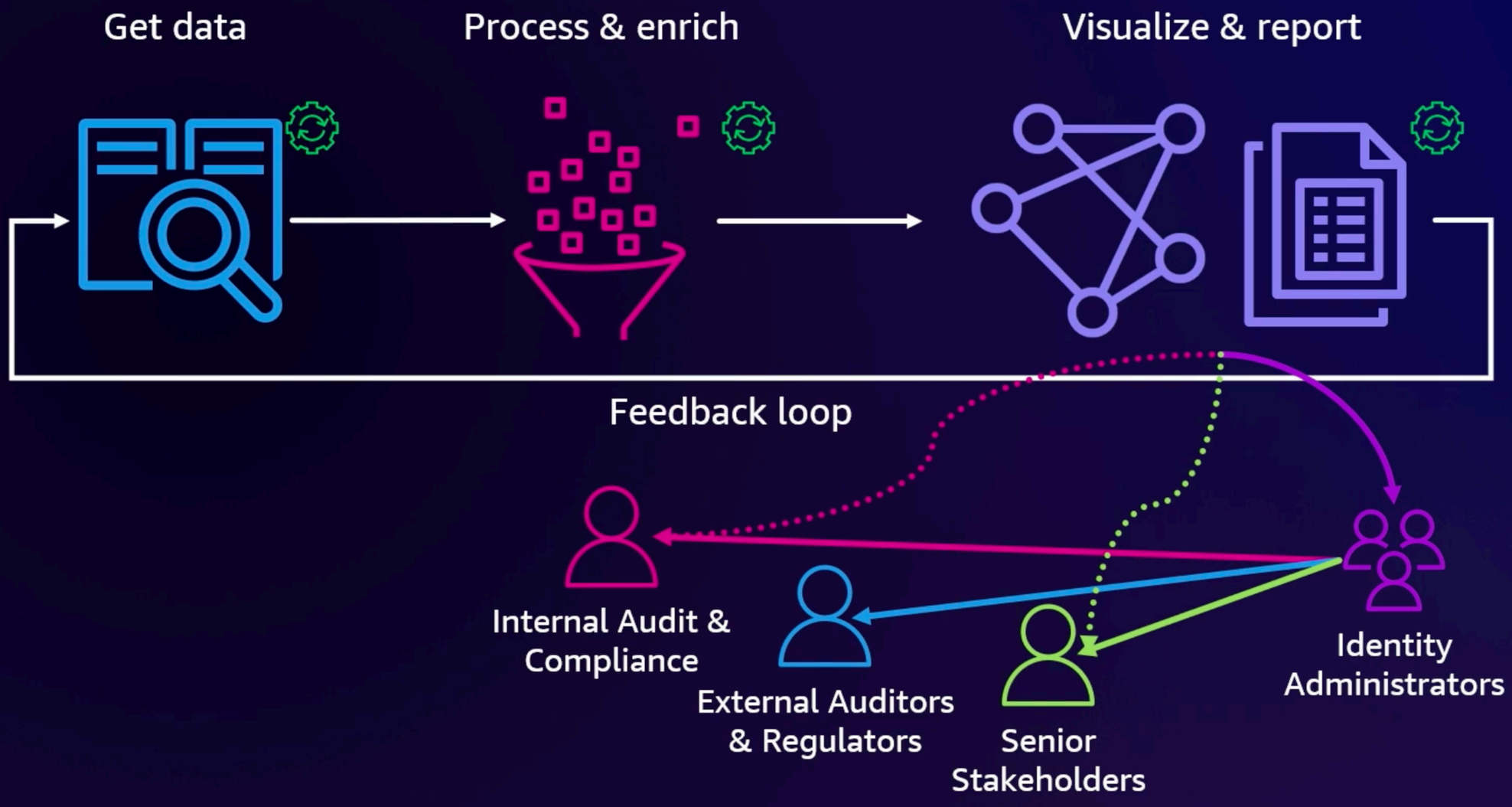
What are the challenges?

- Basing resource access assumptions on IdP group membership doesn’t tell the whole story
- Resource access may be granted using a combination of
    - Identity-based policies
    - Resource-based policies
    - Service Control Policies (SCPs)
    - Resource Control Policies (RCPs)
    - Permissions Boundaries
    - Session Policies
- Teams might deploy custom IAM roles & policies into accounts
- Providing AWS account & resource access visibility to teams beyond just CloudOps

Resource access through IAM Identity Center

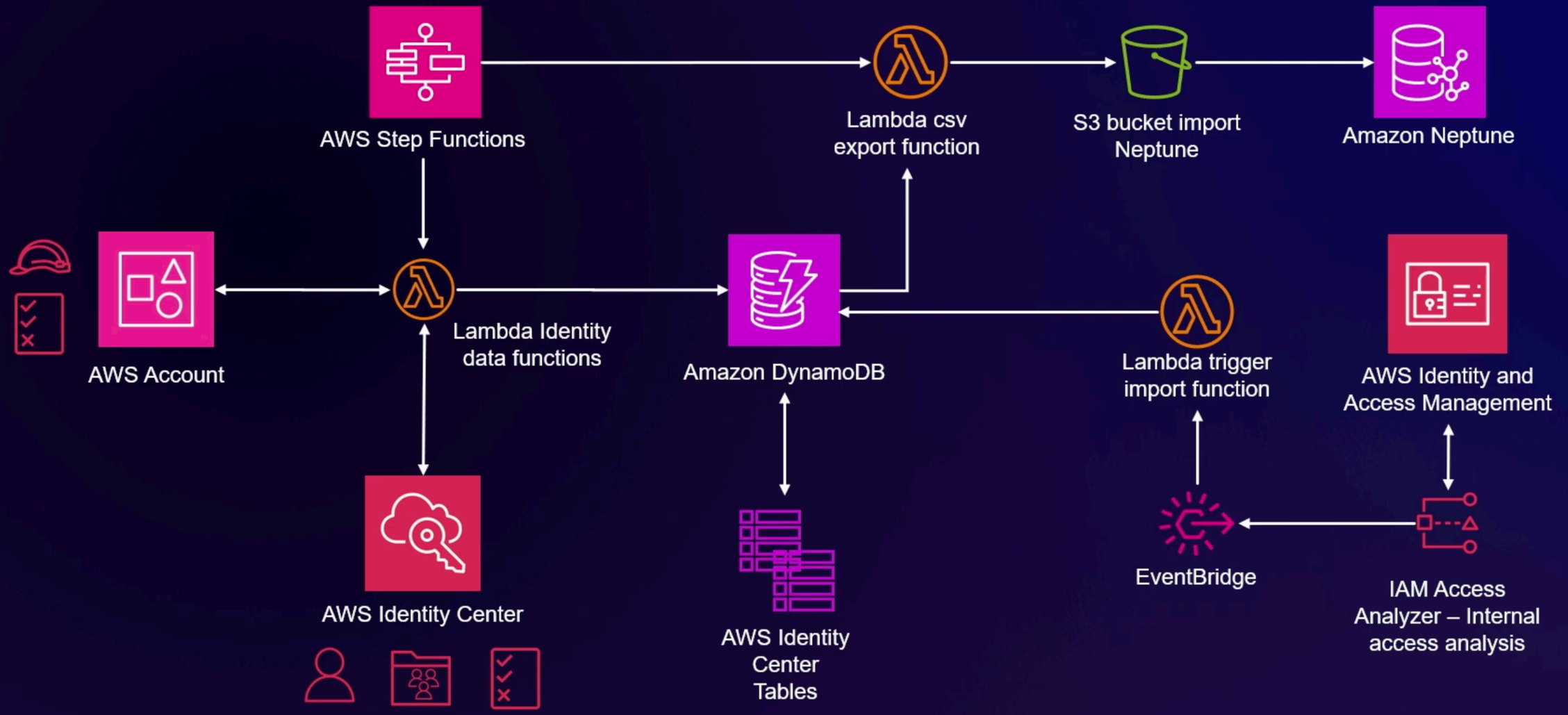


Our Goal



Our Updated Architecture

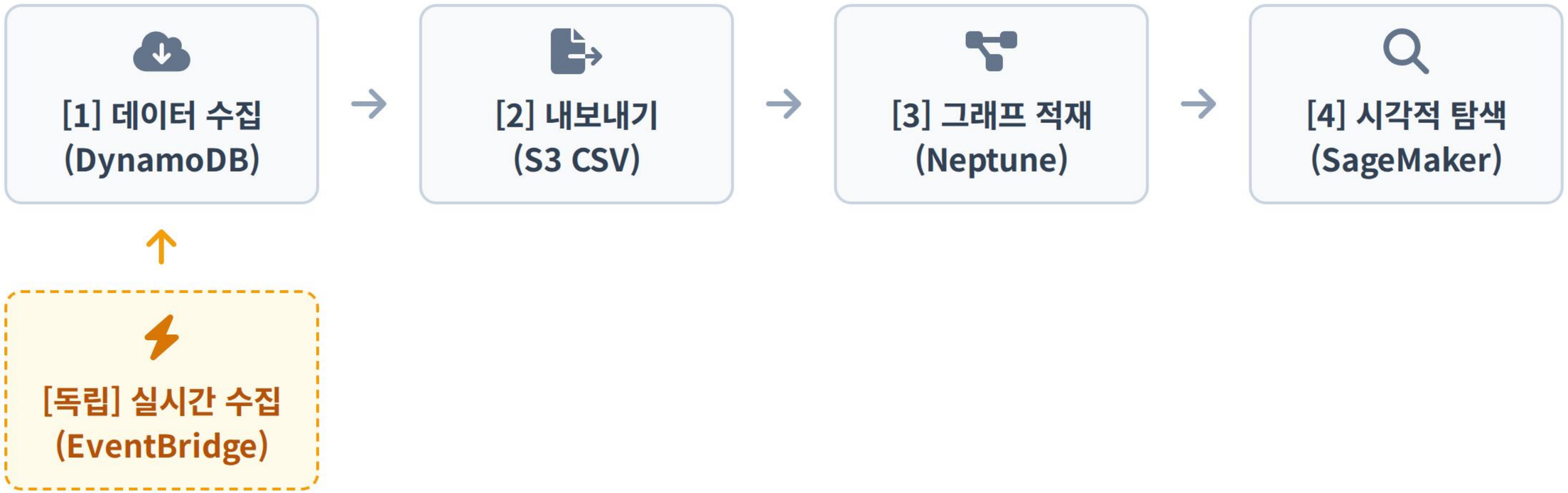
New



제 1 장

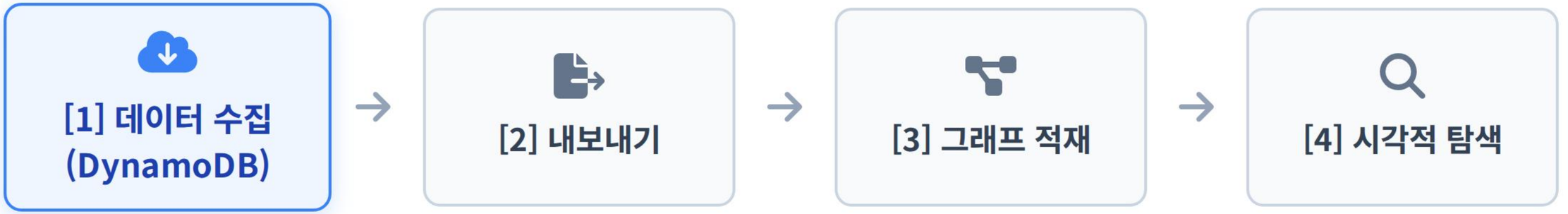
전체 파이프라인 개요

전체 파이프라인 흐름도



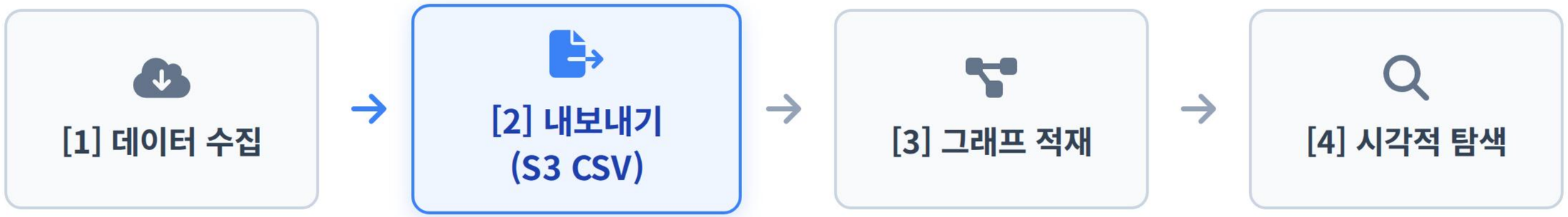
- 분산된 AWS 권한 데이터의 **중앙 집중식 수집 체계** 구축
- 수집 데이터를 **단일 연결망(그래프)으로 시각화**하는 자동화 파이프라인

파이프라인 1단계: 데이터 수집



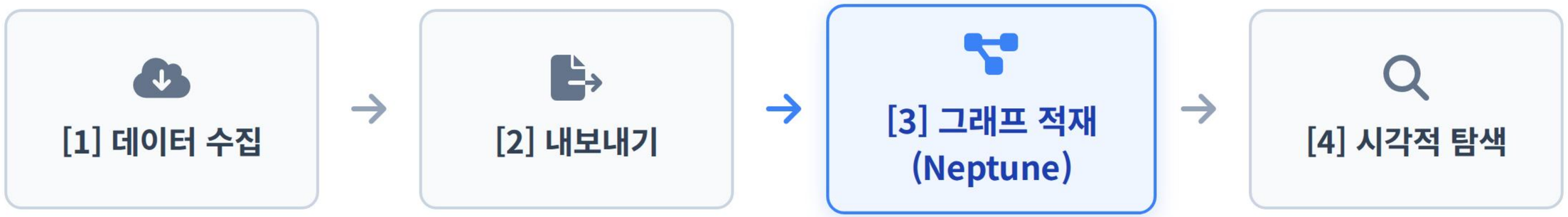
- **AriaStateMachine 구동**: Identity Center, Organizations, IAM API 일괄 호출
- **임시 캐싱 저장소 활용**: 후속 처리를 위해 수집 데이터를 DynamoDB 12개 테이블에 안전하게 저장

파이프라인 2단계: 내보내기 (Export)



- **AriaExportGraphStateMachine 구동**: DynamoDB 적재 원시 데이터 일괄 변환
- **그래프 DB 전용 포맷 변환**: Neptune 규격에 맞춘 S3 CSV 파일(노드/엣지) 형태로 내보내기 수행

파이프라인 3단계: 그래프 적재 (Import)



- **기존 그래프 DB 초기화 (Reset)**: 데이터 무결성 및 최신 상태 반영을 위한 전체 캔버스 삭제
- **Neptune 데이터 임포트**: S3 내보내기 CSV 파일을 읽어 그래프 노드 및 엣지 형태로 물리적 적재

파이프라인 4단계: 시각적 탐색

- **전용 뷰어 환경 구축**: SageMaker Notebook 기반의 대화형 쿼리 및 분석 인터페이스 제공
- **권한 토폴로지 시각화**: 노드와 엣지로 렌더링된 AWS 인프라 전체 권한망의 도식적 확인 및 추적

독립 모듈: 실시간 이벤트 수집

- **비동기 파이프라인 분리**: 메인 Step Functions 워크플로 주기와 독립적으로 동작
- **보안 경고 즉각 반영**: Access Analyzer 발생 Finding을 EventBridge로 감지하여 DynamoDB 실시간 기록

제 2 장

기반 인프라 핵심 용어 정리

솔루션 구성 3대 핵심 인프라

DynamoDB

- 키-값 기반 NoSQL 데이터베이스
- API 수집 데이터의 임시 저장 캐시 역할 수행

Step Functions

- 다수 Lambda 함수의 워크플로 오케스트레이션 엔진
- 복잡한 순차 및 병렬 실행 흐름 제어

EventBridge

- 서버리스 이벤트 버스 및 라우팅 서비스
- 특정 이벤트 기반의 즉각적 규칙 실행

AWS Organizations의 활용 목적

- 다수 AWS 계정의 통합 및 중앙 집중식 관리를 지원하는 근간 서비스
- `organizations:ListAccounts` API를 통한 조직 내 전체 멤버 계정 목록 일괄 확보
- **사각지대 없는 권한 프로비저닝 및 IAM 역할 전수 조사를 위한 필수 기반 데이터 제공**

제 3 장

AWS IAM Identity Center 구조 분석

IAM Identity Center (구 AWS SSO)

- 회사의 인증 시스템(Okta, Azure AD 등)과 연동하여 조직 구성원 식별
- 다수 계정에 대한 **AWS 접근 권한을 중앙 집중식으로 통합 관리**하는 핵심 서비스
- **본 솔루션이 수집하는 핵심 데이터 구조의 근본적 출처 (Source of Truth) 제공**
  - 접근 주체 (누가)
  - 대상 환경 (어디에)
  - 접근 방식 (어떻게 접근하는가)

권한 통제 기본 객체: 사용자 및 그룹

사용자 (User)

- Identity Center에 등록된 실제 개별 직원 (예: Alice, Bob)
- 외부 IdP 환경 동기화 또는 AWS 내부 수동 생성 지원

그룹 (Group)

- 권한 할당의 효율성을 위한 사용자 논리적 집합 체계
- 단일 사용자의 다수 그룹 중복 소속 허용 (예: Dev-Team 겸 Data-Team)

권한 통제 단위: 권한 세트 (Permission Set)

- "특정 사용자나 그룹에게 **어떤 AWS 권한**을 부여할 것인가"를 상세히 정의한 정책 모음
- AWS 관리형 정책(AmazonS3ReadOnlyAccess 등) 및 커스텀 정책 연결 지원
- 명명 규칙 사례: S3-ReadOnly, DynamoDB-Full 등

권한의 실체화: 프로비저닝과 계정 할당

프로비저닝

- 중앙에서 정의한 '권한 세트'를 특정 대상 계정에 **물리적으로 배포**하는 행위
- 배포 완료 시 AWSReservedSSO_{권한명} 형태의 **IAM 역할** 자동 생성 완료

계정 할당

- 특정 주체에게 특정 권한으로 특정 계정에 접근토록 **3자 연결 작업** 수행
- (예: Dev-Team ➔ S3-ReadOnly ➔ Target Account 허용)

접근 권한 5단계 관계 흐름 (Topology)

사용자 → 그룹 → 권한세트 → 계정 → IAM역할

- 다단계에 걸쳐 엮인 복잡한 연결 고리의 끊김 없는 추적 확보
- **파편화된 권한 구조를 하나의 가시성 높은 그래프망으로 완성하는 것이 최종 목적**

제 4 장

IAM Access Analyzer 분석

Access Analyzer 및 Finding 정의

- **IAM Access Analyzer:** AWS 리소스 접근 권한을 수학적으로 검증하여 의도치 않은 권한 및 보안 위험을 자동 식별하는 서비스
- **Finding (탐지 결과):** 분석기가 권한 검증을 통해 생성한 **보안 위험 티켓**
- 발생 원인 및 탐지 영역에 따라 크게 3가지 핵심 유형으로 구분 및 관리 진행

탐지 유형 1: External Access

- 외부망(타 계정, 퍼블릭 인터넷 등)에서 조건 없이 무단 접근 가능한 취약 리소스 식별
- 사례: "특정 S3 버킷의 퍼블릭 읽기 허용", "외부 계정에서의 IAM 역할 Assume 허용"
- **솔루션 기술 제약:** EventBridge 수집은 동작하나, 원본 코드 결함(KeyError)으로 인해 현재 DB 적재 실패 발생 (보완 요구)

탐지 유형 2: Internal Access

- 계정 내부 망에서 특정 주체(역할/사용자)가 내부 민감 리소스에 접근 가능한 상세 내역 분석
- 사례: "S3-ReadOnly 역할이 특정 버킷 내 데이터 GetObject 직접 접근 가능"
- **그래프 구조 연동:** 분석 완료 시 그래프 상에 `GRANTS_ACCESS_TO` 관계 엣지 생성의 핵심 소스로 동작

탐지 유형 3: Unused Access (미사용 권한)

- 할당 후 장기간 미사용 상태로 방치된 유휴 권한 및 역할 탐지 (AWS 유료 서비스 적용 구간)
- 사례: "권한 정책은 부여되었으나, 최근 90일간 API 호출 이력 전무"
- **그래프 구조 연동:** 그래프 상에 HAS_UNUSED_ACCESS 경고 엣지로 표기되어 관리자의 보안 점검 포인트로 활용

Finding 생명주기 (Lifecycle) 상태 관리

ACTIVE 상태

- 보안 취약점 및 과잉 권한 할당이 현재 **미해결 상태로 존재**함
- 해당 상태의 티켓 데이터만 그래프 시각화 및 분석의 주 대상으로 활성화됨

RESOLVED 상태

- 관리자의 권한 정책 수정 조치를 통해 위험이 **완전히 해소된 상태**
- 해결 감지 시 솔루션 워크플로가 DB 내 해당 데이터를 영구적으로 안전 삭제 처리

제 5 장

시각화 도구의 명확한 역할 정의

아키텍처 도구 활용에 대한 명확한 분리

- 다수의 AWS 서비스가 혼합된 형태의 아키텍처 환경 이해 필수
- 데이터의 **직접 가공/적재 시스템**과 단순 **시각적 뷰어(Viewer) 시스템**의 명확한 역할 경계선 구분 요구

Amazon SageMaker Notebook의 제한적 역할

- 기계 학습 모델링 플랫폼 기능 대신 **'웹 기반 컴퓨팅 환경'** 요소만 선별 차용
- 인스턴스 내부에서는 데이터 수집 및 변환 로직이 일절 동작하지 않음
- 데이터베이스에 질의 후 결과를 화면에 렌더링하는 **단순 시각화 인터페이스 (UI)** 역할에 국한됨

Jupyter 및 시각화 확장 패키지 구성

- **Jupyter Notebook 환경:** DB 대상 텍스트 기반 쿼리 송수신 인터페이스 제공
- 텍스트 응답만으로는 복잡한 그래프 토폴로지 분석에 한계 존재
- **graph_notebook 패키지 도입:** 텍스트 응답을 노드/엣지 다이어그램으로 렌더링하는 **AWS 공식 확장 프로그램** 필수 설치
- 인터랙티브한 팝업창 (Graph Explorer) 동작 지원 체계 마련

라이프사이클(Lifecycle) 스크립트 기반 자동화

- 사용자의 번거로운 수동 확장 패키지 설치 과정을 전면 배제
- SageMaker 인스턴스 최초 구동 시 **백그라운드 셸 스크립트 자동 실행** 체계 구현
- DB 연결 엔드포인트 세팅 및 필수 시각화 패키지 `graph_notebook`의 무인 설치 지원 프로세스 완성

제 6 장

Amazon Neptune Analytics 및 그래프 모델링

Neptune Analytics와 그래프 데이터베이스

- 수백만 건의 데이터 간 복잡한 **'관계'** 저장 및 초고속 탐색에 특화된 고성능 그래프 DB 엔진
- 전통적 관계형 DB(RDB)의 테이블(행/열) 격리 관리 방식 탈피
- 데이터 자체를 원천적으로 **노드(점)와 엣지(선)**가 연결된 거미줄 망 형태로 구성 및 영구 저장

그래프 모델링 핵심 객체: 노드와 엣지

● 노드 (Node)

- 핵심 엔티티(주체 및 객체)를 물리적 점으로 표현
- 사용자, 그룹, 권한 세트, 계정, IAM 역할, 보안 Finding 등이 각각 독립적 객체로 적재

→ 엣지 (Edge)

- 분리된 노드 사이의 상태 및 인과 관계를 물리적 선으로 표현
- 소속, 할당, 배포, 리소스 접근 등의 모든 동작이 방향성을 지닌 선으로 강 결합됨

데이터 대량 적재용 CSV 포맷 규격화

- 초고속 벌크 렌더링을 위한 **특수 규격 S3 CSV 파일** 대량 임포트(Import) 방식 채택
- **노드 CSV 필수 포맷:** `~id` (고유 식별자), 속성 데이터, `~label` (엔티티 타입) 명시
- **엣지 CSV 필수 포맷:** `~from` (출발 주체), `~to` (도착 객체), `~label` (관계 명칭) 명시

강력한 그래프 탐색 언어: openCypher

- 기존 SQL 문법 구조와 유사성을 가지나, **노드와 선의 화살표 방향을 텍스트로 그려 질의**하는 특화 탐색 언어

```
MATCH path = (u:UserName)-[*]-(a:AccountName) RETURN path
```

- **의미:** 시작점(임의의 사용자)부터 종착점(특정 대상 계정)까지 연결된 모든 다단계 통로망을 전면 추적 후 반환

그래프 데이터베이스 도입의 당위성

- 'Alice → 그룹 → 권한세트 → 계정 → 최종 역할'의 다단계 탐색 시, RDB 환경에서는 **끝없는 JOIN 연산**에 따른 시스템 부하 유발
- 그래프 DB는 이러한 구조적 한계를 우회하여, **단 한 줄의 경로 탐색 쿼리**만으로 즉각적이고 신속한 결과 도출 지원
- AWS Neptune 내장 쿼리 에디터 활용으로 부가적인 서버 세팅 없는 즉각적 시각화 확보

제 7 장

솔루션 활용 및 비즈니스 가치 분석

솔루션의 최종 핵심 목표 (Core Purpose)

> '우리 회사에서 **누가** 구체적으로 어떤 **계정**에
> 어떤 수준의 **권한**으로 접근할 수 있는가'

- 위와 같은 보안의 핵심 질문에 대하여, 추상적 문서가 아닌 **명확한 시각적 그래픽**으로 즉시 검증 및 해답 제공

시각화 활용: 경로 추적 및 논리-물리 매핑

- **사용자 권한의 흐름 추적 (End-to-End):**
  - 직원 Alice의 현재 소속 그룹 검증 완료
  - 해당 그룹이 확보한 정책을 통한 최종 타겟 계정 도달 루트 입체적 증명
- **추상적 명칭의 물리적 실체화 파악:**
  - 관리자가 임의 설정한 '권한 세트' 이름의 실제 프로비저닝 상황 추적
  - 대상 계정 내 렌더링된 IAM 역할의 난해한 객체명을 투명하게 식별 및 매핑 완료

시각화 활용: 과잉 권한 다이어트 및 내부 통제

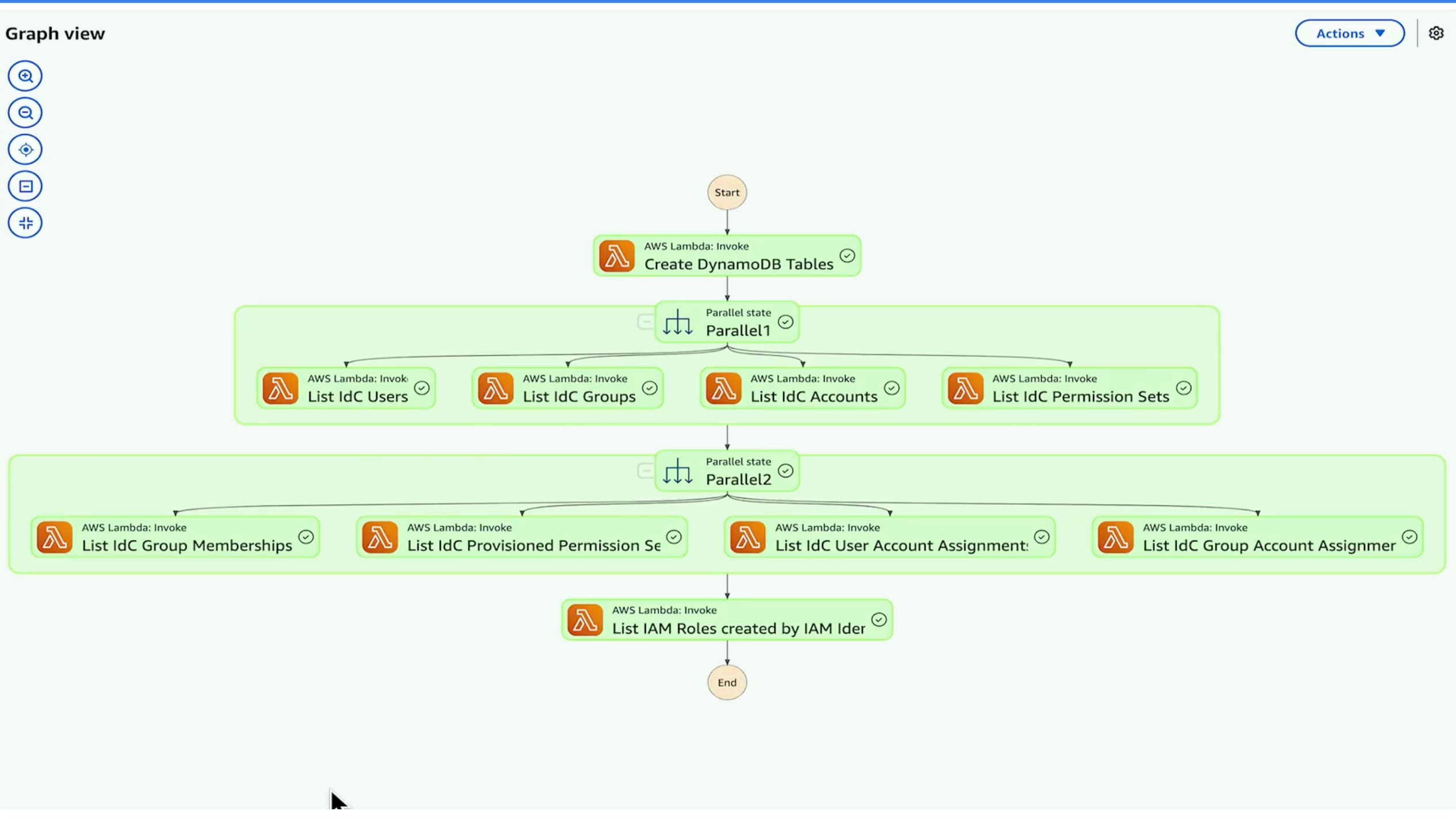
- **유휴 권한 적발 및 보안 경고 시각화:**
  - 권한 공식 할당 이후 장기간 사용 이력이 없는 잉여 권한 붉은색 노드 표출
  - 보안 점검 담당자의 즉각적 회수 및 PoLP(최소 권한 원칙) 이행 유도
- **내부 민감 리소스 접근망 탐지:**
  - 외부 방어망 통과 후, 내부망에 위치한 S3, DB 등에 대한 은밀한 접근 통로 즉시 파악 및 차단 고려

종합 결과 도출 및 통찰 획득

- 수천 장의 JSON 문서 속에 복잡하게 얽힌 클라우드 정책 데이터의 완벽한 재구성 완료
- Neptune 그래프 캔버스 단 한 곳에 **통합 보안 가시성 대시보드** 집중 구현
- 인터랙티브 마우스 조작을 통한 입체적 권한망 탐색 체계 확보

제 8 장

1단계 워크플로: 데이터 수집 상세 (AriaStateMachine)


Graph view
Actions
Start
AWS Lambda: Invoke
Create DynamoDB Tables
Parallel state
Parallel1
AWS Lambda: Invok
List IdC Users
AWS Lambda: Invoke
List IdC Groups
AWS Lambda: Invoke
List IdC Accounts
AWS Lambda: Invoke
List IdC Permission Sets
Parallel state
Parallel2
AWS Lambda: Invoke
List IdC Group Memberships
AWS Lambda: Invoke
List IdC Provisioned Permission Se
AWS Lambda: Invoke
List IdC User Account Assignment
AWS Lambda: Invoke
List IdC Group Account Assignmer
AWS Lambda: Invoke
List IAM Roles created by IAM Ider
End

AriaStateMachine의 도입 목적

- Identity Center, Organizations, IAM 등 다수 서비스 환경에 고립 분산된 데이터 타겟팅
- 통합 API 일괄 호출을 통한 **엔터프라이즈 보안 기초 데이터 1차 긁어모으기** 수행
- 빠른 가공을 위한 DynamoDB 임시 중앙 저장소 일괄 캐싱(Caching) 역할 수행

수집 1단계: 테이블 컨테이너 사전 구축

- 파이프라인 최우선 순위로 단독 가동되는 `createtables` Lambda 엔진 구동
- 방대한 수집 데이터를 받아낼 **12개의 거대한 빈 DynamoDB 테이블 사전 생성** 조치
- 이후 후발 병렬 수집기들의 무차별적인 데이터 삽입 시 발생할 수 있는 DB 락(Lock) 및 에러 원천 차단

수집 2단계: 기본 엔티티 1차 병렬 스캔

- 서로 간 데이터 종속성이 없는 **4개의 기본 객체 수집 Lambda 프로세스 동시 점화**
- 단일 실행 대비 병목 현상 제거 및 파이프라인 처리 속도 극대화 도모

2단계 병렬 스캔: 사용자 및 그룹 확보

listusers 함수

- 조직 내 등록된 전체 구성원 목록 API 전수 호출
- AriaIdCUsers 테이블에 영구 식별 저장 완료

listgroups 함수

- 인증 시스템에 편성된 모든 논리적 그룹의 물리적 뼈대 스캔
- AriaIdCGroups 테이블에 기초 구조 적재 완료

2단계 병렬 스캔: 계정 및 권한 정책 확보

listaccounts 함수

- AWS Org 울타리 내 구동 중인 전체 멤버 계정 활동 상태 검사
- AriaIdCAccounts 테이블 내 계정 풀(Pool) 등록 완료

listpermissionsets 함수

- 관리자가 작성해둔 중앙 집중식 통제 정책 세트 전체 열람 수행
- AriaIdCPermissionSets 테이블 내 정책 원본 집중 보관

수집 3단계: 복합 관계망 2차 병렬 매핑

- 이전 단계에서 수집 완료된 기초 엔티티 뼈대(사용자, 그룹, 권한 등) 데이터 스캔 수행
- 엔티티와 엔티티 간 연결선에 해당하는 **'상호 교차 관계(Mapping)'** 정보 정밀 수집 목적
- 4개의 고도화된 매핑 전용 Lambda 동시 병렬 가동 시작

3단계 병렬 매핑: 소속 및 배포 현황 확인

listgroupmembership

- 단일 그룹 내 소속된 전체 사용자 리스트 역추적 조회
- 그룹↔사용자 간 양방향 결속 데이터 테이블 생성

listprovisioned...

- 개별 계정 단위로 실제 밀어넣어진 권한 세트 현황 스캔
- 계정↔권한세트 간 종속 및 배포 이력 테이블 매핑

3단계 병렬 매핑: 난해한 3방향 할당 구조 해석

listuseraccount...

- 개인 사용자의 '계정' 및 '할당 권한 세트' 동시 보유 관계망 파악
- 개인 단위의 난해한 3방향 락(Lock) 구조 해제 후 테이블 적재

listgroupaccount...

- 거대 조직 그룹의 다수 '계정' 분산 배치 및 '정책 룰' 스캔
- 조직 규모의 거대한 3방향 연결 고리를 테이블상에 완전 기록

수집 4단계: 타겟 계정 물리 역할 심층 추적

- 선행 스캔이 완벽히 끝난 후, **단독 순차적으로 구동되는 최후의** `getiamroles` **단계**
- 대상 타겟 계정 내부에 `sts:AssumeRole` 합법적 침투 후 전체 IAM 역할 목록 강제 열람 수행
- 조회된 문자열 중 `AWSReservedSSO_*` 패턴의 프로비저닝 산출물만 추출 필터링 완료
- 긴 역할명 안에 은닉된 기존 '권한 세트 명칭'을 파싱하여 이전 단계 데이터와 최종 연결 고리 완성

최종 순차 배치의 당위성 및 수집의 완료

- **후방 고정 배치 원인:** 3단계에서 사전 구성된 '프로비저닝 데이터베이스'가 온전히 적재되어 있어야만 역할 텍스트 매칭 검증 연산이 가능하기 때문
- **성공적 1막 종료의 의미:**
  - DynamoDB 내 12개 공간에 조직의 전체 권한, 엔티티, 매핑 테이블 구축 완료
  - 시각화를 위한 무결한 '원시 그래프 식재료' 조달 프로세스 최종 마감

제 9 장

2단계 워크플로: 데이터 포맷 변환 및 적재

Graph view

Actions

AriaExportGraph 워크플로 가동 목적

- DynamoDB에 캐싱 보관된 투박한 원시 테이블 데이터의 대량 추출 진행
- Neptune 그래프 엔진 아키텍처에 맞춘 **최적화된 포맷(CSV)으로의 데이터 구조 가공 및 개편**
- 낡은 캔버스를 비우고 새로운 시각화용 최신 권한망을 물리적 갱신(Update) 하는 중추 과정

적재 1단계: S3 대량 포맷 변환 추출

- 대용량 메모리 처리 전용 `s3export` Lambda 엔진 풀가동 스캔 진입
- DynamoDB 12개 영역 데이터를 목적에 맞게 두 부류로 찢어서 재가공 실시
  - **노드 구성 CSV 풀:** `AriaIdCUsers.csv` 등 단일 점 데이터 분리
  - **엣지 구성 CSV 풀:** `GroupMembership_Edge.csv` 등 관계 연결선 데이터 분리
- 가공 완료된 수만 건의 파일을 S3 내보내기 전용 안전 버킷에 대량 업로드 커밋 완료

적재 2/3단계: 기존 캔버스 완전 백지화 제어

Reset Neptune Graph

- ResetGraph 강력 API 파괴 명령 하달
- 기존 캔버스 데이터를 흔적 없이 제거
  (전면 엎어쓰는 **Full-Import** 전략 구사)

Wait 120초 (초기화 보장)

- 대규모 삭제 I/O 스레드 백그라운드 구동 대비
- DB 클러스터 안정화 확보를 위한 워크플로 진행 임시 차단 조치 수행

적재 4/5단계: 신규 그래프망 재건축 및 완료

Start Import Task

- S3에 대기 중인 렌더링용 CSV 파이프라인 개방
- 엔진 자율 파싱을 통한 거미줄 같은 신규 노드 및 엣지 물리적 재구성 점화

Wait 180초 (안정화 보장)

- 대량 데이터 적재 및 DB 내부 검색 인덱싱 빌드 백그라운드 작업 시간 보장
- 완벽한 콘솔 쿼리 실행을 위한 최후 대기 후 대단원 마감 처리

제 10 장

비동기 실시간 보안 이벤트 감시망 (EventBridge)

이벤트 버스 기반 실시간 탐지 라우팅

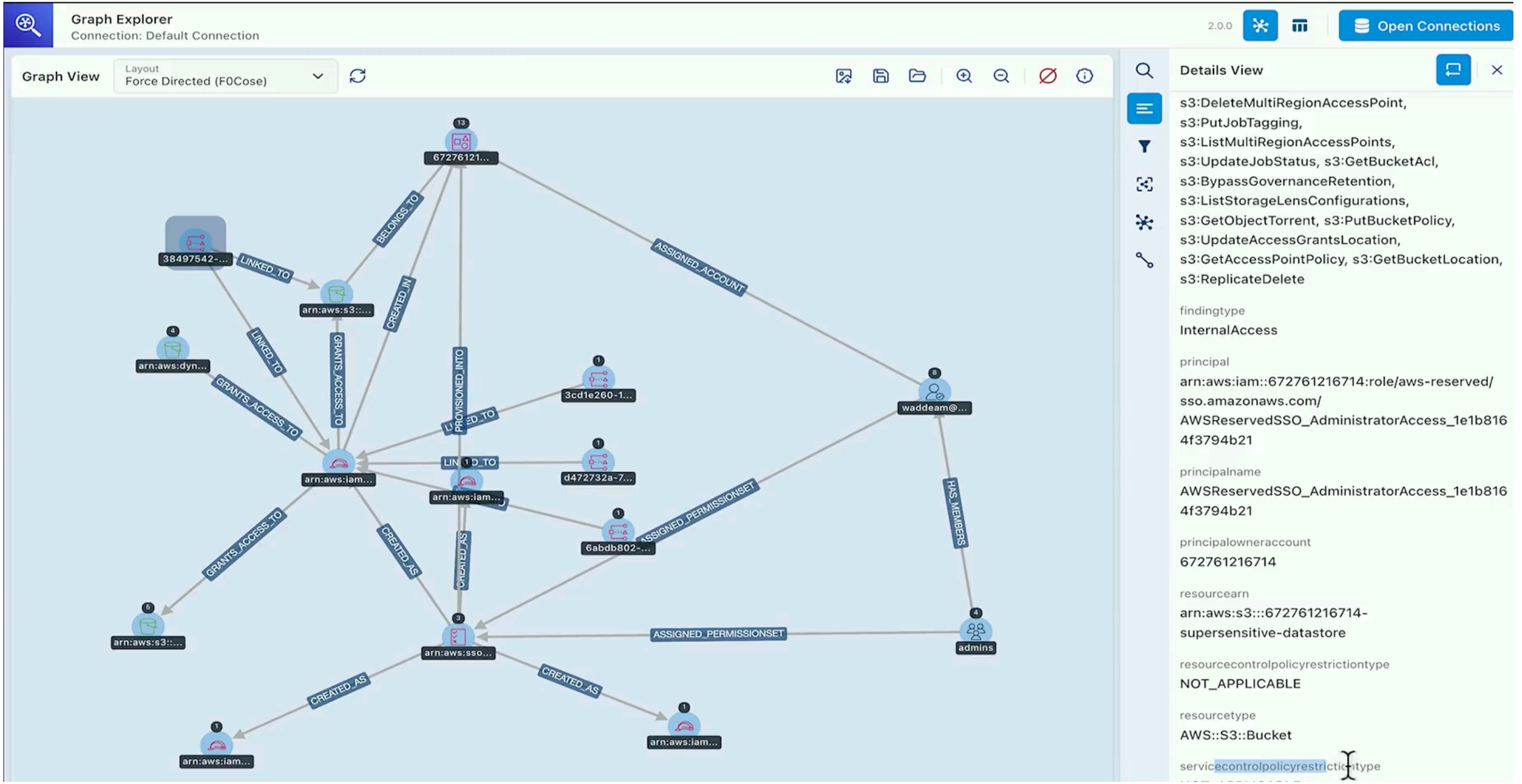
- Access Analyzer가 포착한 심각한 보안 침해 징후를 정기 워크플로 배치 스케줄과 완전히 분리
- **EventBridge 이벤트 버스망**에 위험 신호가 떨어지는 즉시 1초 내 비동기 포착 수행
- 대기 중이던 `accessanalyzerfindingingestion` 함수가 깨어나 DynamoDB 경고망 테이블에 긴급 적재 기록

이벤트 기반 코드 무중단 자동 갱신

- 관리자가 솔루션 성능 개선 목적으로 S3 소스 버킷에 신규 압축(ZIP) 코드를 덮어쓰기 하는 즉시 트리거 발동
- 변경 신호를 낚아챈 `updatefunctioncode` 함수가 백그라운드에서 구동 진입
- 솔루션 생태계 전체 구동 환경의 서비스 다운타임 없는 **무중단 핫-업데이트(Hot-Update) 자동화** 완수

제 11 장

Neptune 시각화 데이터 완전 해부도


Graph Explorer
Connection: Default Connection
2.0.0
Open Connections
Graph View
Layout
Force Directed (F0Cose)
Details View
s3:DeleteMultiRegionAccessPoint,
s3:PutJobTagging,
s3:ListMultiRegionAccessPoints,
s3:UpdateJobStatus, s3:GetBucketAcl,
s3:BypassGovernanceRetention,
s3:ListStorageLensConfigurations,
s3:GetObjectTorrent, s3:PutBucketPolicy,
s3:UpdateAccessGrantsLocation,
s3:GetAccessPointPolicy, s3:GetBucketLocation,
s3:ReplicateDelete
findingtype
InternalAccess
principal
arn:aws:iam::672761216714:role/aws-reserved/
sso.amazonaws.com/
AWSReservedSSO_AdministratorAccess_1e1b816
4f3794b21
principalname
AWSReservedSSO_AdministratorAccess_1e1b816
4f3794b21
principalowneraccount
672761216714
resourcearn
arn:aws:s3:::672761216714-
supersensitive-datastore
resourcecontrolpolicyrestrictiontype
NOT_APPLICABLE
resourcetype
AWS::S3::Bucket
servicecontrolpolicyrestrictiontype
67276121...
38497542-...
arn:aws:s3:...
arn:aws:dyn...
arn:aws:iam...
arn:aws:iam...
3cd1e260-1...
d472732a-7...
6abdb802-...
waddeam@...
admins
arn:aws:s3:...
arn:aws:sso...
arn:aws:iam...
arn:aws:iam...
BELONGS_TO
LINKED_TO
CREATED_IN
ASSIGNED_ACCOUNT
GRANTS_ACCESS_TO
PROVISIONED_INTO
ASSIGNED_PERMISSIONSET
HAS_MEMBERS
CREATED_AS
ASSIGNED_PERMISSIONSET
CREATED_AS
CREATED_AS

최종 도출된 핵심 점(Node) 객체 목록

- **UserName 노드:** Identity Center 인증을 통과한 인프라 진입 주체 (직원)
- **GroupName 노드:** 다수 사용자를 권한 제어 효율성을 위해 결합한 묶음망체
- **PermissionSet 노드:** 부여할 권한 정책 문서를 캡슐화한 권한 보따리 본체
- **AccountName 노드:** 최종 접근 대상이 되는 격리된 단일 AWS 클라우드 환경 풀
- **RoleName 노드:** 대상 계정 내부에 실질적으로 작동 가능하도록 빚어진 권한의 최종 형태

최종 도출된 보안 점(Security Node) 객체

- **Finding 노드 (Internal / Unused):**
  - AI 분석기가 캔버스에 경고 목적으로 띄워둔 독립적 보안 위반 티켓
- **CriticalResources 노드:**
  - 뚫린 권한을 타고 최종 도달하게 되는 민감 데이터 저수지(DB, S3) 적색 알람 객체

엔티티 간 기본 연결선(Edge)의 의미

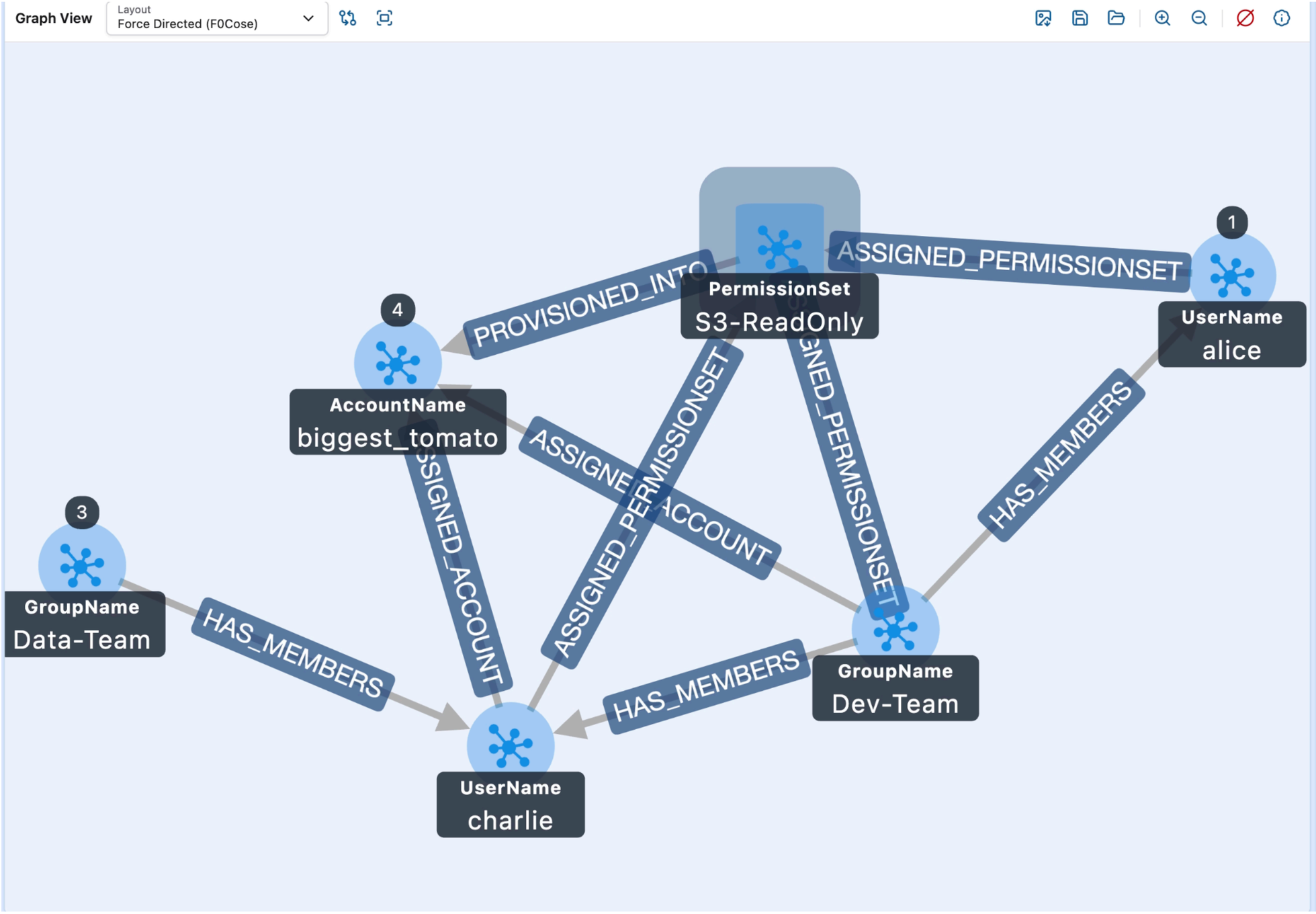
- **HAS_MEMBERS 엣지:** 껍데기 그룹이 알맹이 사용자를 멤버로 포섭하고 있음을 증명하는 선
- **ASSIGNED_PERMISSIONSET 엣지:** 특정 사람이나 집단에 강력한 정책의 룰이 배급 완료되었음을 잇는 선
- **ASSIGNED_ACCOUNT 엣지:** 권한 주체가 목적지 특정 타겟 계정의 철조망을 넘을 수 있도록 결재된 증빙선

물리적 렌더링 연결선(Edge)의 의미

- **PROVISIONED_INTO 엣지:** 관념적 상태의 정책안이 목적지 계정에 강제로 배포 및 이식 완료됨을 확정하는 지표
- **CREATED_AS / CREATED_IN 엣지:** 배포된 아이디어가 실제 작동 가능한 배역(역할)으로 몸을 입고(CREATED_AS), 해당 계정 무대 안에 완전히 착지하여 안착했음(CREATED_IN)을 증명

심층 보안 취약점 연결선(Security Edge)

- ⚠️ **GRANTS_ACCESS_TO 엣지:**
  - 접근 주체가 시스템 뱃속 깊은 곳 민감 자산까지 침투 가능한 치명적 권한이 살아있음을 가리키는 적색 선
- 🧹 **HAS_UNUSED_ACCESS 엣지:**
  - 발급된 지 오래되어 주인이 까먹었으나, 언제든 탈취 악용될 수 있는 위험한 유휴 권한이 널부러져 있음을 고발하는 경고선



Details View

PermissionSet › S3-ReadOnly
S3-ReadOnly

Neighbors (4)

AccountName	1	1
GroupName	1	1
UserName	2	2

Properties

Node Id
arn:aws:sso:::permissionSet/ssoins-7230dbc3bcdc1b90/ps-723076bca36bfd67

Node Label
PermissionSet

description
S3 Read Only Access

name
S3-ReadOnly

결론 및 발전 방향 (Future Scope)

현 아키텍처의 비즈니스적 통찰과 당면한 한계점 분석

솔루션 가치: 직관적 가시성 혁신 도달

- 글자 중심의 파편화된 복잡한 IAM 정책 문서들의 기계적 해독 과정 생략 구조 확보
- 관리자의 뇌 구조가 이해하기 가장 편한 방식인 **'그림과 선(토폴로지 그래프)'으로 통합 권한 지형도 완성**
- 조직의 보안 블라인드 스팟(사각지대) 제거 및 신속한 리스크 대응 거버넌스 확립 완료

한계점 1: 정규식 파싱 의존에 따른 불완전성

- 그래프의 핵심인 엣지(관계선)를 시스템이 100% 자율적으로 규명하는 아키텍처 아님
- **Lambda 함수 내부에서 개발자가 하드코딩한 문자열 정규식 파싱 로직**에 전적으로 의존하는 태생적 한계 지님
- AWS 내부에 존재하는 변칙적이거나 숨겨진 모든 권한 교차로를 완벽한 엣지로 자동 구축해 내는 데는 기술적 불완전성 잔존 우려

한계점 2: 프리미엄 인프라 조합에 따른 TCO 증가

- Amazon Neptune Analytics, SageMaker Notebook, 유료 Access Analyzer 등 과금 체계가 무거운 프리미엄 서비스 동시 조합 설계 채택
- 대형 엔터프라이즈 환경에서 지속적인 데이터 갱신(Full-Import) 수행 시, **막대한 클라우드 총소유비용(TCO) 폭증**이라는 현실적 족쇄 존재

AI 시대 도래에 따른 차세대 보안 엔진으로의 도약

- 구조적 결함과 유지보수 비용 한계에도 불구하고, 그래프 DB가 던지는 보안 인사이트 자체의 본연적 가치 인정
- **생성형 AI(Generative AI) 및 LLM 모델과의 전격 결합 고도화 제안**
  - 수동 파싱 로직을 탈피한 AI 기반 변칙 권한 자동 추론 및 연결망 자율 생성 아키텍처 구상
  - 자연어 프롬프트("최근 90일간 안 쓴 위험한 관리자 권한 가진 사람 찾아줘") 기반의 사이버 보안 감사 에이전트로 지속 디벨롭(Develop) 수행 가능성 확보

프레젠테이션 종료

지금까지 ARIA-gv 솔루션 상세 워크플로의 모든 과정을
경청해 주셔서 대단히 감사합니다.