🛡️ AWS SAA-C03 통합 가이드 - Part 1: 보안 아키텍처 (30%)

> **가장 높은 배점 영역 - 최우선 학습 필수!**

📋 보안 도메인 학습 전략

핵심 원칙

1. **최소 권한 원칙 (Least Privilege)** - 모든 보안의 기본
2. **심층 방어 (Defense in Depth)** - 여러 보안 계층
3. **암호화** - 전송 중(TLS) + 저장 시(KMS)
4. **추적 및 감사** - CloudTrail + Config

🔐 IAM (Identity & Access Management)

🧠 WHY (왜 중요한가?)

- AWS 보안의 핵심은 **최소 권한 원칙**
- Access Key 하드코딩은 보안사고의 주범
- 정책 평가 순서: **명시적 Deny > 명시적 Allow > 기본 Deny**

⚙️ HOW (어떻게 작동하는가?)

핵심 구성요소

```
User (사람) → Group (권한 묶음) → Policy (JSON 권한)
EC2/Lambda → Role (임시 권한) → Policy
```

User: 실제 사람 계정

- 루트 사용자는 **절대 사용 금지**
- MFA 필수 활성화
- Access Key는 최소화

Group: 공통 권한 묶음

- 정책을 User가 아닌 Group에 부여
- 예: Developers, Admins, ReadOnly

Role: AWS 리소스가 사용할 임시 권한

- EC2, Lambda 등에 부여
- Access Key 대신 사용

- STS로 임시 자격 증명 발급

Policy: JSON 기반 권한 정의

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::my-bucket/*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "203.0.113.0/24"
        }
      }
    }
  ]
}
```

정책 유형

유형	설명	사용 사례
Identity-based	User/Group/Role에 부여	일반적인 권한 관리
Resource-based	S3 버킷, SQS 등에 직접 부여	교차 계정 접근
Permissions Boundary	최대 권한 제한	권한 상한선 설정
SCP	Organizations 계정 제한	조직 전체 정책

고급 개념

STS (Security Token Service)

- 임시 자격 증명 발급 (15분~12시간)
- AssumeRole: 다른 역할 가정
- GetSessionToken: MFA 토큰
- GetFederationToken: 페더레이션 사용자

IAM Identity Center (AWS SSO)

- 여러 계정/서비스 통합 로그인
- SAML 2.0 기반
- Active Directory 통합

- 권장: 조직 규모 확장 시 필수

Permissions Boundary

- Role/User가 가질 수 있는 **최대 권한 상한선**
- 위임된 권한이라도 Boundary 초과 불가
- 예: 개발자는 S3만 접근 가능하도록 제한

Service Control Policy (SCP)

- Organizations에서 계정 전체 권한 제한
- **루트 사용자에도 적용** (매우 강력)
- 명시적 Allow + Deny 조합
- 예: 특정 리전만 사용 허용

조건 키 (Condition Keys)

- IP 주소: `aws:SourceIp`
- 시간: `aws:CurrentTime`
- MFA: `aws:MultiFactorAuthPresent`
- 태그: `aws:PrincipalTag`
- VPC: `aws:SourceVpc`

🧩 SCENARIO (실전 문제 패턴)

상황	정답	오답 함정
EC2가 S3 접근 필요	**IAM Role 연결**	Access Key를 EC2에 저장 ❌
여러 계정에서 리소스 공유	**Cross-Account Role + Trust Policy**	IAM User 생성 ❌
조직 전체 Single Sign-On	**IAM Identity Center**	각 계정마다 User 생성 ❌
최소 권한 보장 필요	**Permissions Boundary 적용**	관리자 권한 부여 ❌
외부 사용자 임시 접근	**STS AssumeRole**	영구 자격 증명 생성 ❌
루트 계정 보호	**MFA 활성화 + 사용 금지**	루트로 일상 작업 ❌
특정 리전만 사용 허용	**SCP로 다른 리전 Deny**	IAM Policy로 제한 ❌
개발자 권한 위임 (제한적)	**Permissions Boundary**	전체 관리자 권한 ❌

🎯 시험 키워드 매칭

- **"최소 권한"** → IAM Role + Permissions Boundary
- **"임시 자격 증명"** → STS
- **"교차 계정 접근"** → Cross-Account Role

- **"중앙 인증"** → IAM Identity Center
- **"조직 전체 제한"** → SCP
- **"MFA 필수"** → IAM Policy Condition
- **"감사 추적"** → CloudTrail

🚫 절대 하지 말아야 할 것 (TRAP)

- ❌ Root Account로 API 호출
- ❌ Access Key를 코드에 하드코딩
- ❌ 정책을 User에 직접 부착 (Role 중심 설계 권장)
- ❌ 와일드카드(*) 권한 남용
- ❌ IAM User 공유 (개인별 계정 생성)
- ❌ 오래된 Access Key 방치 (정기 교체)

📊 IAM Best Practices 체크리스트

- ✅ 루트 계정에 MFA 활성화
- ✅ 루트 계정 Access Key 삭제
- ✅ 개별 IAM User 생성 (공유 금지)
- ✅ Group으로 권한 관리
- ✅ 최소 권한 원칙 적용
- ✅ 강력한 암호 정책 설정
- ✅ Access Key 정기 교체
- ✅ CloudTrail로 API 호출 로깅
- ✅ IAM Access Analyzer로 외부 접근 검토

🌐 VPC (Virtual Private Cloud)

🧠 WHY

- 리소스를 논리적으로 분리하고 네트워크 트래픽을 완전 제어
- 보안, 통신 경로, 인터넷 접근을 직접 설계 가능
- **기본 개념**: CIDR 블록 = IP 주소 범위 (예: 10.0.0.0/16 = 65,536개 IP)

⚙️ HOW

VPC 기본 구성

CIDR 블록

- /16: 65,536개 IP (권장, 큰 VPC)
- /20: 4,096개 IP (중간 규모)
- /24: 256개 IP (작은 서브넷)
- /28: 16개 IP (최소 단위)

AWS 예약 IP (각 서브넷마다)

- .0: 네트워크 주소
- .1: VPC 라우터
- .2: DNS 서버
- .3: 미래 사용 예약
- .255: 브로드캐스트 (VPC는 지원 안 함)

서브넷 유형

```
Public Subnet
  ↓ (Route to 0.0.0.0/0)
Internet Gateway (IGW)
  ↓
Internet

Private Subnet
  ↓ (Route to 0.0.0.0/0)
NAT Gateway (in Public Subnet)
  ↓
Internet Gateway
  ↓
Internet
```

NAT 솔루션 비교

항목	NAT Gateway	NAT Instance
관리	AWS 완전 관리형	직접 관리 필요
가용성	**고가용성, AZ당 배포**	단일 장애점 가능
대역폭	**5-45 Gbps**	인스턴스 크기에 의존
비용	더 비쌈 ($0.045/시간 + 데이터)	저렴 (EC2 비용만)
보안 그룹	❌ (지원 안 함)	✅ (지원)
포트 포워딩	❌	✅
Bastion	❌	✅
사용 사례	**프로덕션 (권장)**	개발/테스트, 비용 절감

NAT Gateway 고가용성 구성

- **각 AZ마다 별도의 NAT Gateway 배포**
- 각 Private 서브넷은 같은 AZ의 NAT Gateway 사용
- 하나의 AZ 장애 시 다른 AZ는 정상 작동

Security Group vs NACL

항목	Security Group	NACL
적용 레벨	**Instance (ENI)**	**Subnet**
상태	**Stateful** (응답 자동 허용)	**Stateless** (명시적 규칙 필요)
규칙	**Allow만** 가능	**Allow + Deny** 가능
평가 순서	모든 규칙 평가	**규칙 번호 순서**대로
기본 정책	모든 아웃바운드 허용	모든 트래픽 허용
변경 적용	즉시	즉시

Security Group 예시

```
Inbound:
- Type: HTTP, Port: 80, Source: 0.0.0.0/0
- Type: SSH, Port: 22, Source: 203.0.113.0/24

Outbound:
- All traffic (기본)
```

NACL 예시

Inbound:
- Rule #100: Allow HTTP (80) from 0.0.0.0/0
- Rule #200: Allow HTTPS (443) from 0.0.0.0/0
- Rule #300: Allow SSH (22) from 203.0.113.0/24
- Rule #: Deny All (기본)

Outbound:
- Rule #100: Allow HTTP (80) to 0.0.0.0/0
- Rule #110: Allow Ephemeral (1024-65535) to 0.0.0.0/0
- Rule #*: Deny All (기본)

NACL 주의사항

- Stateless이므로 **Ephemeral Port (1024-65535) 허용 필수**
- 규칙 번호는 100 단위 권장 (중간 삽입 가능)
- 낮은 번호가 우선 평가

VPC Endpoint (매우 중요!)

Gateway Endpoint

- **S3, DynamoDB 전용**
- **무료** (데이터 전송 비용만)
- 라우팅 테이블에 추가
- NAT Gateway/IGW 불필요
- Private 서브넷에서 S3 접근 시 **필수**

Interface Endpoint (PrivateLink)

- 대부분의 AWS 서비스 지원
- ENI 생성 (서브넷에 IP 할당)
- **유료** ($0.01/시간 + 데이터)
- Security Group 적용 가능
- DNS 이름으로 접근

VPC Endpoint 장점

1. NAT Gateway 비용 절감
2. 인터넷 경유 불필요 (보안 향상)
3. 낮은 지연시간
4. 대역폭 제한 없음

VPC 연결 옵션

방법	특징	사용 사례	비용
VPC Peering	1:1 연결, 비전이적, CIDR 겹침 불가	소수 VPC 연결	무료 (데이터만)
Transit Gateway	허브 앤 스포크, 수천 개 VPC/온프레미스	대규모 네트워크	$0.05/시간/연결
PrivateLink	서비스를 다른 VPC에 비공개 노출	SaaS 제공	$0.01/시간
VPN	암호화 터널, 인터넷 경유	온프레미스 연결 (저비용)	$0.05/시간
Direct Connect	전용 회선, 고속/안정적	온프레미스 연결 (고성능)	포트 비용

VPC Peering 제약

- 전이적 Peering 불가 (A-B-C 연결 시 A와 C 직접 통신 불가)
- CIDR 블록 겹침 불가
- 리전 간 Peering 가능 (추가 비용)

Transit Gateway 장점

- 중앙 허브로 모든 VPC/온프레미스 연결
- 전이적 라우팅 지원
- VPN/Direct Connect 통합
- 라우팅 테이블로 세밀한 제어

VPC Flow Logs

- **용도**: 네트워크 트래픽 로깅 및 분석
- **수준**: VPC, 서브넷, ENI
- **저장소**: CloudWatch Logs 또는 S3
- **필터**: Accept, Reject, All
- **주의**: **로그만 수집, 트래픽 차단 불가** ❌

Flow Logs 형식

```
srcaddr dstaddr srcport dstport protocol packets bytes action
203.0.113.12 10.0.0.4 43418 443 6 10 5000 ACCEPT
198.51.100.5 10.0.0.4 80 50080 6 5 2500 REJECT
```

🧩 SCENARIO

상황	정답	이유
Private Subnet EC2가 S3 접근	**Gateway Endpoint**	무료, 인터넷 경유 불필요
Private Subnet EC2가 SNS 접근	**Interface Endpoint**	Gateway는 S3/DynamoDB만

상황	정답	이유
특정 IP 주소 차단	**NACL Deny 규칙**	Security Group은 Deny 불가
여러 AZ 고가용성	**각 AZ마다 서브넷**	AZ 장애 대응
온프레미스 ↔ AWS 대용량 전송	**Direct Connect**	VPN은 대역폭 제한
수백 개 VPC 연결	**Transit Gateway**	Peering은 1:1만
NAT Gateway 비용 절감	**Gateway Endpoint (S3/DynamoDB)**	데이터 전송 비용만
인스턴스 접근 차단	**Security Group 규칙 제거**	Deny 불가, Allow 제거
서브넷 전체 차단	**NACL Deny**	서브넷 레벨 제어

🎯 시험 키워드 매칭

- **"프라이빗 연결"** → VPC Endpoint
- **"특정 IP 차단"** → NACL
- **"고가용성"** → Multi-AZ 서브넷 + NAT Gateway (AZ별)
- **"대규모 네트워크"** → Transit Gateway
- **"온프레미스 전용 회선"** → Direct Connect
- **"비용 절감"** → Gateway Endpoint

🚫 TRAP (함정 주의!)

- ❌ Security Group에서 Deny 가능 (Allow만 가능)
- ❌ NAT Gateway는 Inbound 허용 (아웃바운드 전용)
- ❌ VPC Endpoint Gateway는 모든 서비스 지원 (S3/DynamoDB만)
- ❌ NACL은 Stateful (Stateless임, 양방향 규칙 필요)
- ❌ VPC Peering은 전이적 (비전이적)
- ❌ Flow Logs로 트래픽 차단 가능 (로그만 수집)

🔑 KMS & Secrets Manager

🧠 WHY

- 데이터 암호화는 필수 보안 요구사항 (규제 준수: GDPR, HIPAA 등)
- Key 관리 자동화로 운영 리스크 감소
- **Envelope Encryption**: 대용량 데이터 암호화의 핵심 패턴

⚙️ HOW

KMS (Key Management Service)

CMK (Customer Master Key) 유형

유형	관리 주체	교체	공유 가능	비용	사용 사례
AWS Managed	AWS	자동 (3년)	❌	무료	S3 기본 암호화
Customer Managed	고객	선택적 (1년)	✅	$1/월	**권장**, 교차 계정
AWS Owned	AWS	N/A	❌	무료	DynamoDB 내부
CloudHSM	고객	수동	✅	높음	규제 준수 (FIPS 140-2 Level 3)

Envelope Encryption 과정

1. 데이터 키(Data Key) 생성
   ↓
2. 데이터 키로 실제 데이터 암호화
   ↓
3. CMK(마스터 키)로 데이터 키 암호화
   ↓
4. 암호화된 데이터 + 암호화된 데이터 키 저장

복호화:
1. CMK로 데이터 키 복호화
   ↓
2. 데이터 키로 데이터 복호화

Envelope Encryption 장점

- 대용량 데이터 효율적 암호화 (네트워크 전송 최소화)
- CMK는 AWS KMS 내부에만 보관 (보안 강화)
- 데이터 키는 로컬에서 사용 (성능 향상)

KMS Key Policy

- **필수**: 모든 CMK는 Key Policy 필요
- IAM Policy만으로는 부족
- Root 계정 기본 허용

```
json
```

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::123456789012:role/MyRole"
  },
  "Action": [
    "kms:Decrypt",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

Grant (임시 권한)

- 프로그래밍 방식 권한 위임
- 자동 만료 가능
- 복잡한 워크플로에 유용

Secrets Manager vs Parameter Store

항목	Secrets Manager	Parameter Store
자동 교체	✅ (Lambda 연동)	❌
RDS 통합	✅	❌
버전 관리	✅	✅
암호화	KMS 자동	선택적 (KMS)
비용	**$0.40/secret/월**	**무료** (Standard), $0.05 (Advanced)
크기 제한	64KB	4KB (Standard), 8KB (Advanced)
API 호출 비용	$0.05/10,000	무료 (Standard), $0.05/10,000 (Advanced)
사용 사례	**DB 자격증명, API Key**	구성 데이터, 환경 변수

Secrets Manager 자동 교체

```
1. Lambda 함수 트리거 (일정 주기)
   ↓
2. 새 암호 생성
   ↓
3. DB/서비스에 새 암호 설정
   ↓
4. Secrets Manager 업데이트
   ↓
5. 애플리케이션은 자동으로 새 암호 사용
```

ACM (AWS Certificate Manager)

- **무료** HTTPS/TLS 인증서
- 자동 갱신 (만료 전)
- ALB, CloudFront, API Gateway 직접 연결
- 퍼블릭 인증서 + 프라이빗 CA

🧩 SCENARIO

상황	정답	이유
암호화된 EBS를 다른 계정 공유	**Customer Managed Key + 키 권한 공유**	AWS Managed Key는 공유 불가
Lambda가 RDS 암호 읽기	**Secrets Manager + 자동 교체**	보안 Best Practice
애플리케이션 구성 값 저장	**Parameter Store (Standard)**	무료 + 충분한 기능
HTTPS 인증서 관리	**ACM**	무료 + 자동 갱신
대용량 파일 암호화	**Envelope Encryption**	성능 + 보안
규제 준수 (FIPS 140-2 Level 3)	**CloudHSM**	하드웨어 보안 모듈
교차 계정 KMS 키 공유	**Customer Managed Key + Key Policy**	Principal 지정
매일 자동으로 DB 암호 변경	**Secrets Manager 자동 교체**	Lambda 연동

🎯 시험 키워드 매칭

- **"자동 교체"** → Secrets Manager
- **"교차 계정 공유"** → Customer Managed Key
- **"감사 추적"** → KMS (CloudTrail 통합)
- **"규정 준수"** → KMS + 암호화 필수, CloudHSM
- **"무료 인증서"** → ACM
- **"구성 데이터"** → Parameter Store

🚫 TRAP

- ❌ AWS Managed Key로 암호화된 리소스는 다른 계정 공유 불가
- ❌ Secrets Manager를 환경변수처럼 노출 (API 호출로 가져와야 함)
- ❌ Parameter Store Standard는 8KB 제한 (Advanced는 제한 없음)
- ❌ KMS는 4KB 이상 데이터 직접 암호화 불가 (Envelope Encryption 필요)
- ❌ ACM 인증서는 EC2에 직접 설치 불가 (ALB/CloudFront만)

🔒 CloudTrail / Config / GuardDuty

🧠 WHY

- **추적, 감사, 이상 탐지**는 보안의 3대 요소
- 사후 분석 + 실시간 탐지 조합으로 완벽한 보안 구현
- 규정 준수 필수 요소

⚙️ HOW

CloudTrail (API 호출 로깅)

기능

- 누가(Principal) 무엇을(Action) 언제(Timestamp) 어디서(SourceIP) 실행했는지 기록
- **90일 무료** 보관 (Event History)
- S3 저장 시 무제한 보관

이벤트 유형

유형	설명	기본 활성화	비용
관리 이벤트	리소스 생성/수정/삭제	✅	무료 (첫 Trail)
데이터 이벤트	S3 객체 접근, Lambda 호출	❌	유료
인사이트 이벤트	비정상 API 활동 감지	❌	유료

저장 및 무결성

- S3 버킷에 로그 저장
- CloudWatch Logs 통합 (실시간 모니터링)
- **Log File Integrity Validation** (로그 변조 탐지)
- SSE-S3 또는 SSE-KMS 암호화

Organization Trail

- 조직 내 **모든 계정** 로깅
- 중앙집중 관리
- 마스터 계정에서 생성

AWS Config (리소스 구성 감사)

기능

- 리소스 구성 상태 추적 및 변경 이력
- 컴플라이언스 검증

- **리소스 관계 매핑** (예: EC2 → Security Group → VPC)

Config Rules

유형	설명	예시
관리형 규칙	AWS 제공, 즉시 사용	s3-bucket-public-read-prohibited
사용자 지정 규칙	Lambda 기반	사용자 정의 컴플라이언스

Remediation (자동 수정)

- SSM Automation으로 자동 수정 가능
- 예: 퍼블릭 S3 버킷 자동 비공개 전환
- 수동 승인 또는 자동 실행 선택

주의사항

- **실시간 차단 불가**, 탐지 후 알림만
- 지속적 모니터링 (일정 주기)
- SNS 알림 연동 가능

GuardDuty (위협 탐지)

기능

- **AI/ML 기반** 이상 행위 탐지
- 데이터 소스: CloudTrail, VPC Flow Logs, DNS Logs
- 30일 무료 체험

탐지 예시

1. 비정상 API 호출
   - 루트 계정으로 대량 리소스 삭제
   - 이상한 시간대 API 호출
2. 암호화폐 채굴 활동
   - EC2에서 채굴 소프트웨어 실행
3. 알려진 악성 IP 통신
   - 명령제어(C&C) 서버 통신
4. 비정상 데이터 전송
   - 대량 데이터 유출 시도

자동 대응

GuardDuty Findings
↓
EventBridge Rule
↓
Lambda Function
↓
자동 조치 (예: Security Group 차단)

🧩 SCENARIO

상황	정답	이유
루트 계정으로 대량 API 호출 감지	**GuardDuty**	AI 기반 이상 행위 탐지
리소스 설정이 보안정책 위반	**AWS Config Rule + SNS**	컴플라이언스 감사
누가 S3 버킷 삭제했는지 추적	**CloudTrail Log**	API 호출 기록
퍼블릭 S3 버킷 자동 수정	**Config + SSM Automation**	자동 Remediation
조직 전체 감사	**Organization Trail**	중앙집중 로깅
로그 변조 탐지	**CloudTrail Log File Integrity**	무결성 검증
암호화폐 채굴 탐지	**GuardDuty**	이상 행위 탐지
규정 준수 리포트	**AWS Config**	컴플라이언스 대시보드

🎯 시험 키워드 매칭

- **"누가 언제 삭제했는지"** → CloudTrail
- **"규정 준수 위반 탐지"** → AWS Config
- **"이상 행위 탐지"** → GuardDuty
- **"자동 수정"** → Config Remediation
- **"로그 무결성"** → CloudTrail Validation
- **"중앙 감사"** → Organization Trail

🚫 TRAP

- ❌ CloudWatch Logs와 CloudTrail 혼동 (CloudWatch는 애플리케이션 로그)
- ❌ Config는 실시간 차단 불가, "탐지" 후 알림만
- ❌ GuardDuty는 기본으로 모든 계정에 활성화 X (수동 활성화 필요)
- ❌ CloudTrail은 데이터 이벤트 기본 로깅 X (추가 비용)

💼 S3 보안

🧠 WHY

- S3는 **기본적으로 프라이빗**이지만 잘못된 설정으로 유출 사고 빈번
- 버킷 정책(리소스 기반) vs IAM 정책(ID 기반) 혼동 주의
- Capital One 사고 (2019): 잘못된 S3 설정으로 1억 명 개인정보 유출

⚙️ HOW

보안 계층 (평가 순서)

1. Block Public Access (계정/버킷 레벨) ← 최우선
   ↓
2. Bucket Policy (리소스 기반)
   ↓
3. IAM Policy (ID 기반)
   ↓
4. ACL (레거시, 비추천)

Block Public Access 설정

- **권장**: 전체 계정 레벨에서 활성화
- 4가지 설정 (모두 체크 권장)
  1. BlockPublicAcls
  2. IgnorePublicAcls
  3. BlockPublicPolicy
  4. RestrictPublicBuckets
- **예외**: CloudFront OAC/OAI로만 접근 허용

암호화 옵션

유형	키 관리	사용 사례	감사 가능	성능
SSE-S3	AWS 관리	기본 암호화	❌	빠름
SSE-KMS	AWS KMS	감사 필요 시	✅ (CloudTrail)	중간
SSE-C	고객 제공	고객이 키 보유	❌	중간
Client-Side	클라이언트	클라이언트 암호화	❌	빠름

SSE-S3 (기본 권장)

- AES-256 암호화
- AWS 관리 키

- 추가 비용 없음
- 헤더: `x-amz-server-side-encryption: AES256`

SSE-KMS

- KMS CMK 사용
- CloudTrail 감사 가능
- 키 교체 자동
- 비용: KMS API 호출당
- 헤더: `x-amz-server-side-encryption: aws:kms`

버킷 정책 예시

HTTPS 강제

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::my-bucket/*",
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "false"
        }
      }
    }
  ]
}
```

특정 IP만 허용

```json
```

```
{
  "Effect": "Allow",
  "Principal": "*",
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "IpAddress": {
      "aws:SourceIp": "203.0.113.0/24"
    }
  }
}
```

S3 Versioning + MFA Delete

- **Versioning**: 객체 변경 이력 보존
  - 삭제 시 Delete Marker 추가 (실제 삭제 X)
  - 복구 가능
  - 버전마다 비용 발생
- **MFA Delete**: 버전 영구 삭제 시 MFA 인증 필수
  - 루트 계정만 활성화 가능
  - 최고 보안 수준

S3 Object Lock

- **Write-Once-Read-Many (WORM)**
- 버전 관리 필수

모드

모드	설명	삭제 가능
Compliance	루트도 삭제 불가	❌ (기간 만료 후만)
Governance	특정 권한으로 삭제 가능	✅ (s3:BypassGovernanceRetention)

Legal Hold

- 기간 제한 없는 잠금
- 무기한 보존
- 수동으로만 제거

S3 Access Analyzer

- IAM Access Analyzer 기반

- 외부 접근 가능한 버킷 탐지
- 자동 스캔 및 알림

🧩 SCENARIO

상황	정답	이유
모든 버킷 Public 접근 방지	**Block Public Access (계정 레벨)**	가장 강력한 보호
특정 계정만 접근 허용	**Bucket Policy + Principal**	교차 계정 접근
암호화 강제	**Bucket Policy + `aws:SecureTransport`**	HTTPS만 허용
삭제 방지 (규제)	**S3 Object Lock (Compliance)**	루트도 삭제 불가
CloudFront로만 접근	**OAC + Bucket Policy**	직접 접근 차단
암호화 감사 필요	**SSE-KMS**	CloudTrail 연동
실수 삭제 복구	**Versioning 활성화**	Delete Marker 제거
외부 접근 탐지	**S3 Access Analyzer**	자동 스캔

🎯 시험 키워드 매칭

- **"퍼블릭 접근 차단"** → Block Public Access
- **"암호화 감사"** → SSE-KMS
- **"교차 계정"** → Bucket Policy
- **"규제 준수"** → Object Lock Compliance
- **"CloudFront 전용"** → OAC
- **"삭제 방지"** → Versioning + MFA Delete

🚫 TRAP

- ❌ ACL로 퍼블릭 허용 설정 (Block Public Access로 차단됨)
- ❌ SSE-KMS 사용 시 KMS 권한 부여 누락
- ❌ Bucket Policy와 IAM Policy 중복 설정 (복잡도 증가)
- ❌ Object Lock은 버전 관리 필수 (사전 활성화)

📋 보안 도메인 최종 체크리스트

✅ IAM

- [ ] 루트 계정 MFA 활성화
- [ ] IAM Role 중심 설계 (Access Key 최소화)
- [ ] 최소 권한 원칙 적용
- [ ] SCP로 조직 전체 제한
- [ ] CloudTrail로 API 호출 로깅

✅ VPC

- [ ] Multi-AZ 서브넷 구성
- [ ] NAT Gateway (AZ별)
- [ ] Security Group vs NACL 차이 이해
- [ ] VPC Endpoint (Gateway/Interface)
- [ ] Flow Logs 활성화

✅ 암호화

- [ ] KMS Customer Managed Key
- [ ] Secrets Manager 자동 교체
- [ ] S3 SSE-KMS
- [ ] EBS 암호화
- [ ] RDS 암호화

✅ 모니터링

- [ ] CloudTrail Organization Trail
- [ ] AWS Config Rules
- [ ] GuardDuty 활성화
- [ ] S3 Access Analyzer

🎯 **이 파트가 시험의 30%를 차지합니다! 보안은 모든 아키텍처의 기본입니다.**

다음: Part 2 - 복원력 아키텍처 (26%)