AWS 아키텍처 설계

Chapter 07. IAM

설계 시나리오

큰 규모의 조직에서 각 팀의 팀원들이 전문적인 역할을 맡고 있는 상황이다.

각 팀원들에게 업무에 필수적인 권한만 부여하여 AWS 리소스 보호 및 접근 제어를 하고자 한다.

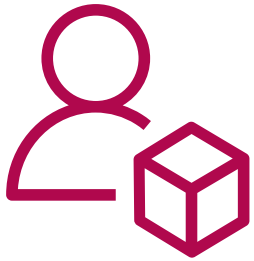
이러한 요구 사항을 충족할 수 있는 서비스는 무엇이 있을까?

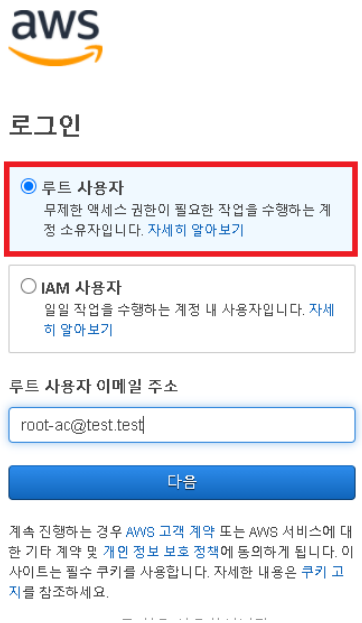
IAM (Identity and Access Management)

01 AWS Account / IAM

AWS Account (루트 사용자)

- **AWS 계정은 AWS 서비스 및 리소스에 대한 모든 권한을 갖는다.**
- **강력한 권한을 갖고, 어떠한 제한도 받지 않는다.**





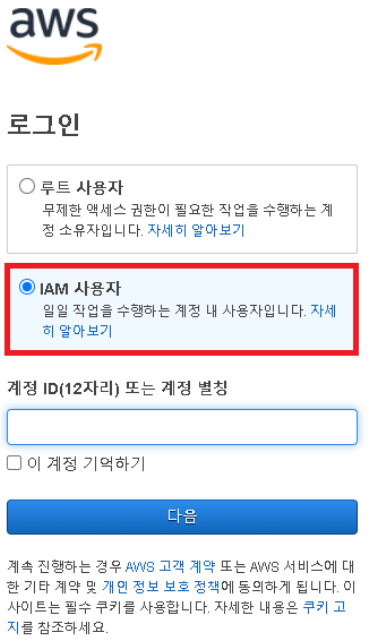
IAM User

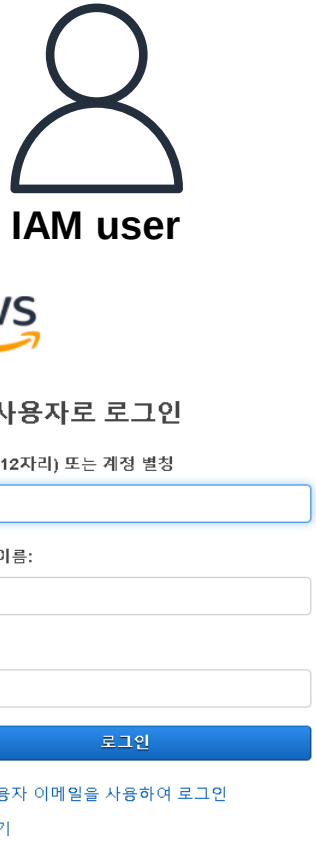
- **IAM 사용자**는 **AWS 계정(루트 사용자)** 내의 사용자를 의미한다.
- 각 사용자는 **자체 자격 증명**을 갖는다.
- IAM 사용자는 할당된 **권한**으로만 AWS 작업을 수행할 수 있다.

IAM user

IAM User

- **AWS Management Console**
  - 사용자 이름 / 암호





IAM User

- **프로그래밍 방식 액세스**
  - AWS CLI 및 SDK
  - 액세스 키 ID 및 비밀 액세스 키

IAM user

액세스 키 ID: AKIAT4ZXEDSN7EXAMPLE
보안 액세스 키: Gr8rXAAtnFEMI/T6MDENG/bfjRfiCYEXAMPLEKEY

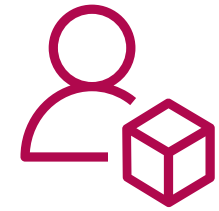
AWS CLI

```
[root@ip-172-31-35-241 ec2-user]# aws configure
AWS Access Key ID [None]: [redacted]
AWS Secret Access Key [None]: [redacted]
Default region name [None]: ap-northeast-2
Default output format [None]:
```

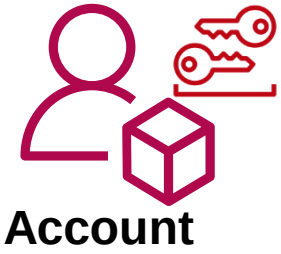
AWS SDK

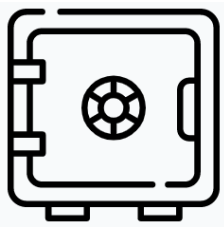
Python Java .NET

안전한 계정 관리



- IAM 관리 사용자 생성



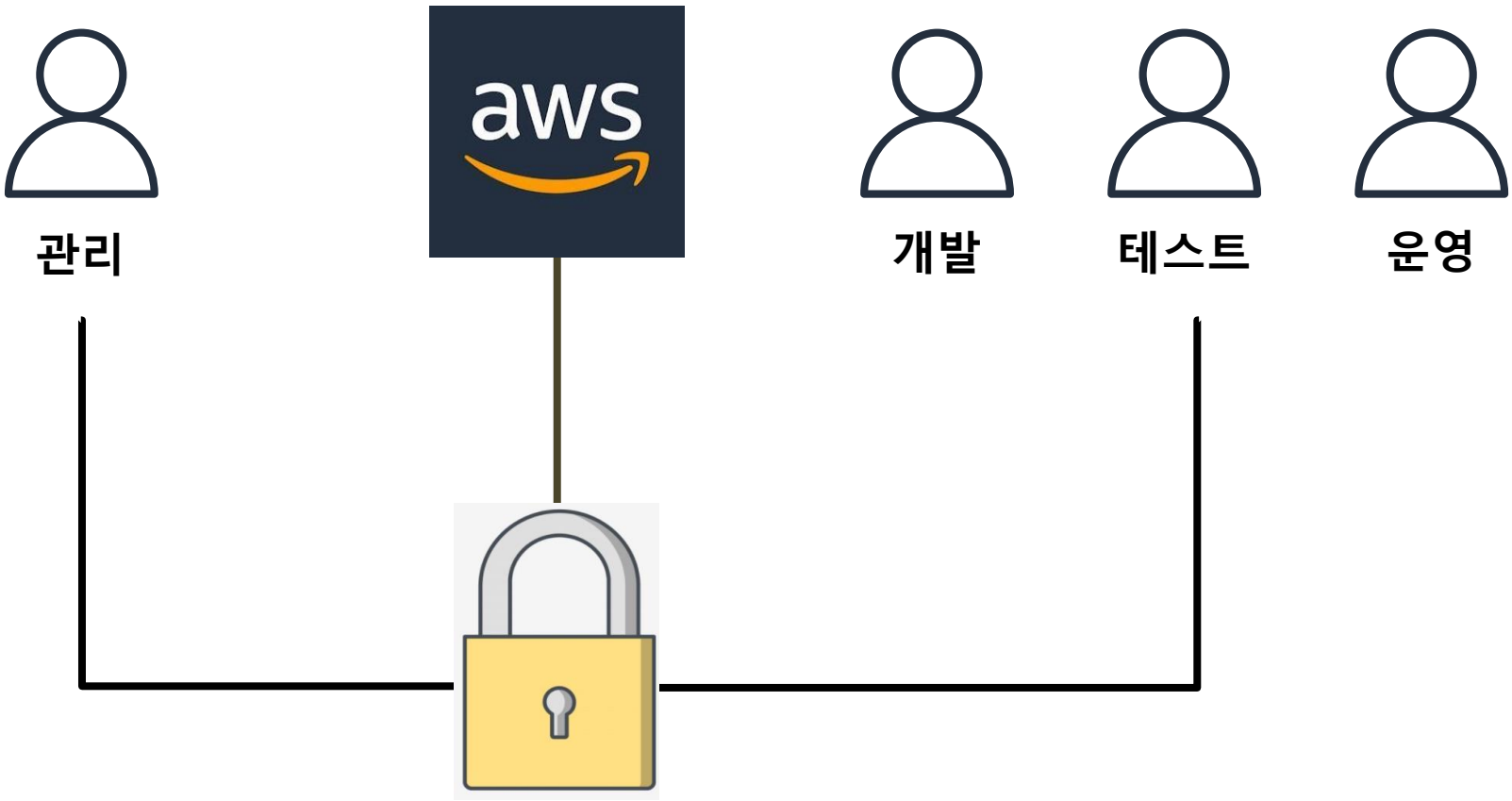


- AWS 계정 자격 증명 잠금

- IAM 관리 사용자로 작업

최소 권한의 원칙

- 업무에 따라 AWS 접근 권한을 세분화하여 제어해야 한다.

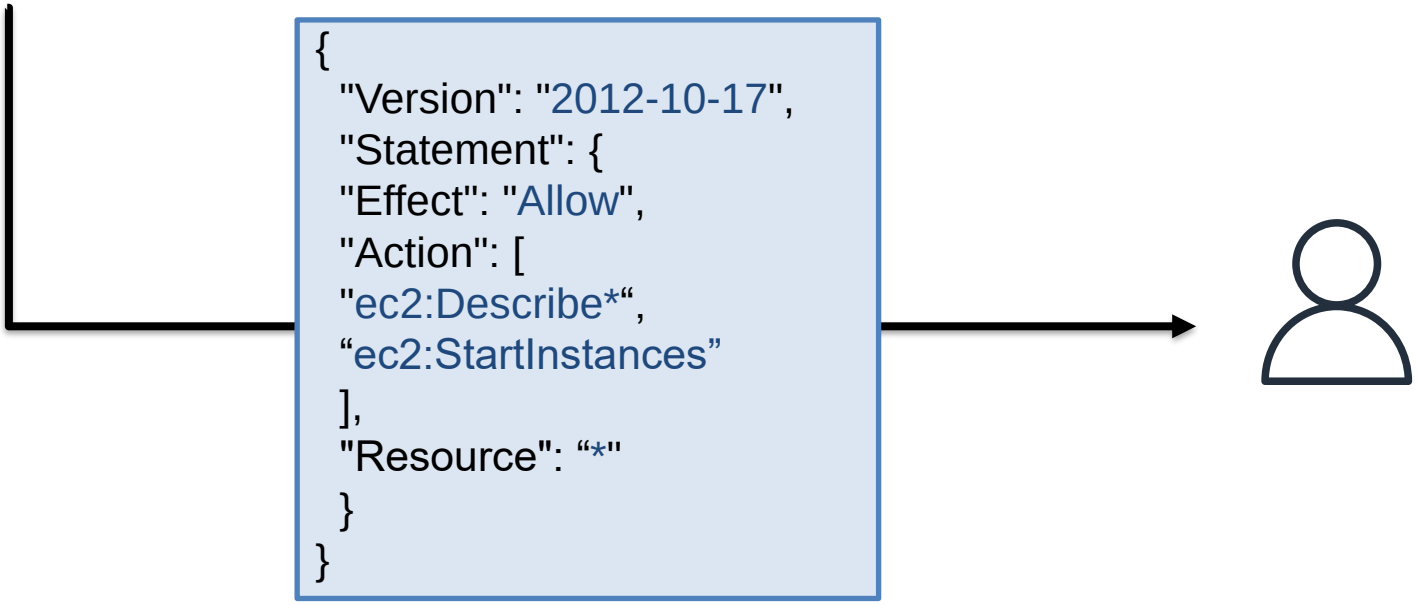


AWS IAM(Identity and Access Management)

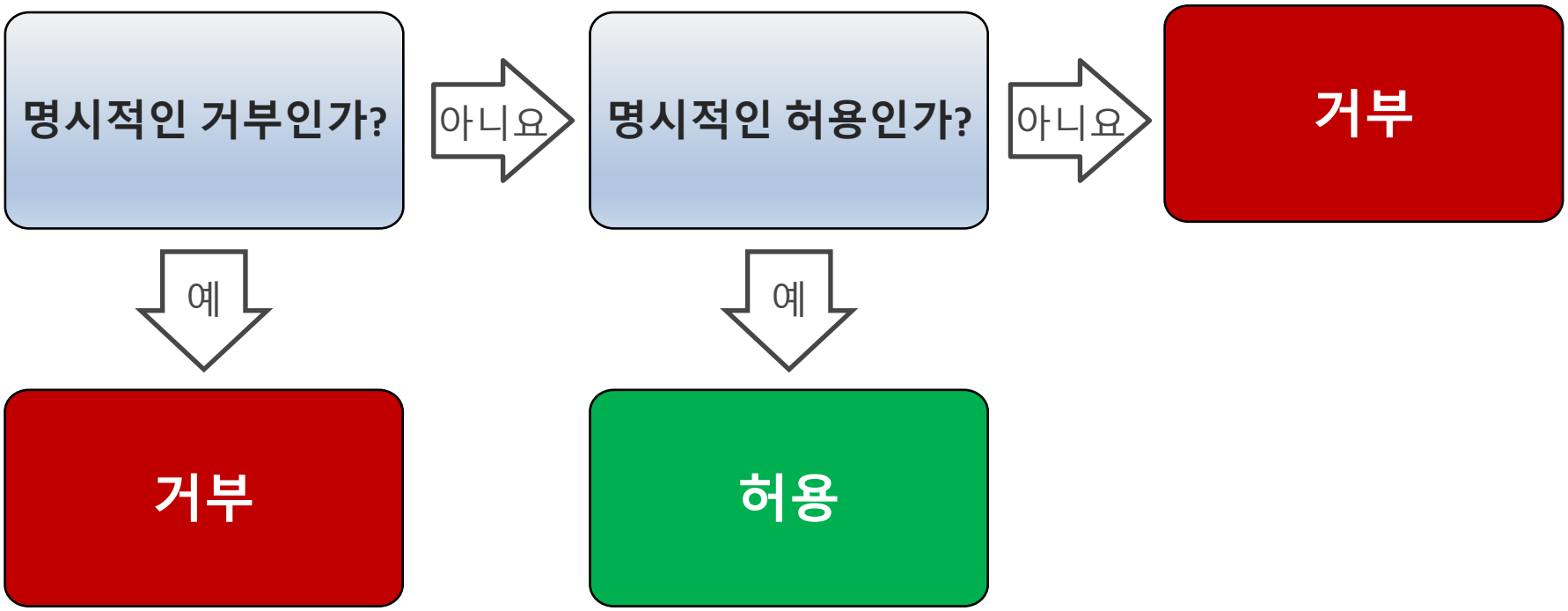
권한 부여

정책

- IAM 사용자에게 기본적으로 주어진 권한은 없다.
- 정책은 권한을 설명하는 JSON 문서
- 작업 요청 시 해당 작업에 대한 권한이 있는지 평가
- IAM 정책은 AWS 서비스에 대한 접근만 제어



권한 평가 방법



권한 부여 방법

정책

- 자격 증명 기반 – IAM 보안 주체와 정책 연결
- 리소스 기반 – AWS 리소스와 정책 연결

자격 증명 기반 정책

자격 증명 기반 정책

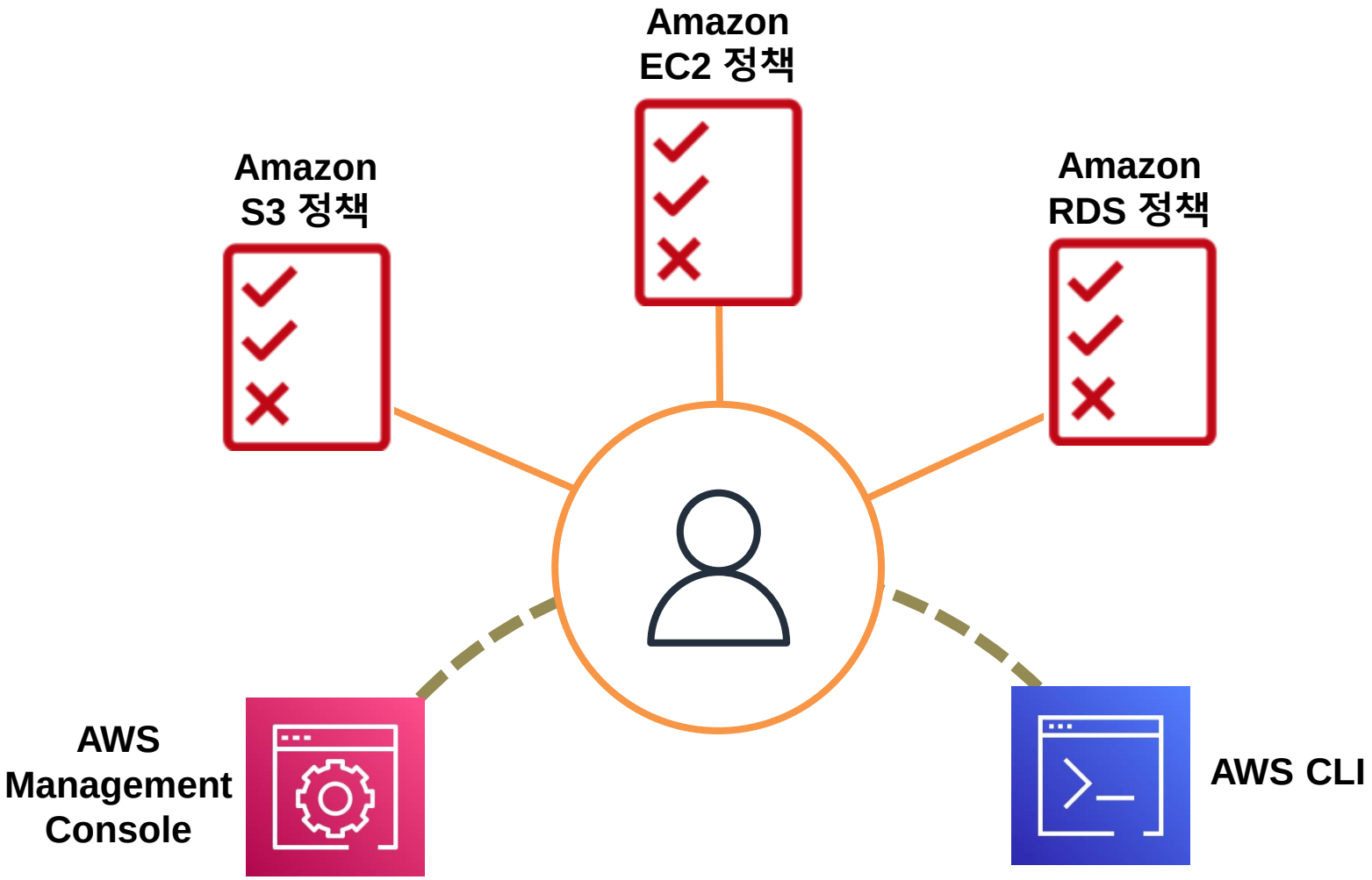
- **연결 대상** – IAM User / Group / Role
- **정책 유형** – AWS 관리형 / 고객 관리형 / 인라인
- **제어** – Action / Resource / Condition

리소스 기반 정책

리소스 기반 정책

- **연결 대상** – AWS 리소스 (S3 / KMS / SQS 등)
- **정책 유형** – 항상 인라인

다수의 권한이 부여된 자격증명



IAM 정책

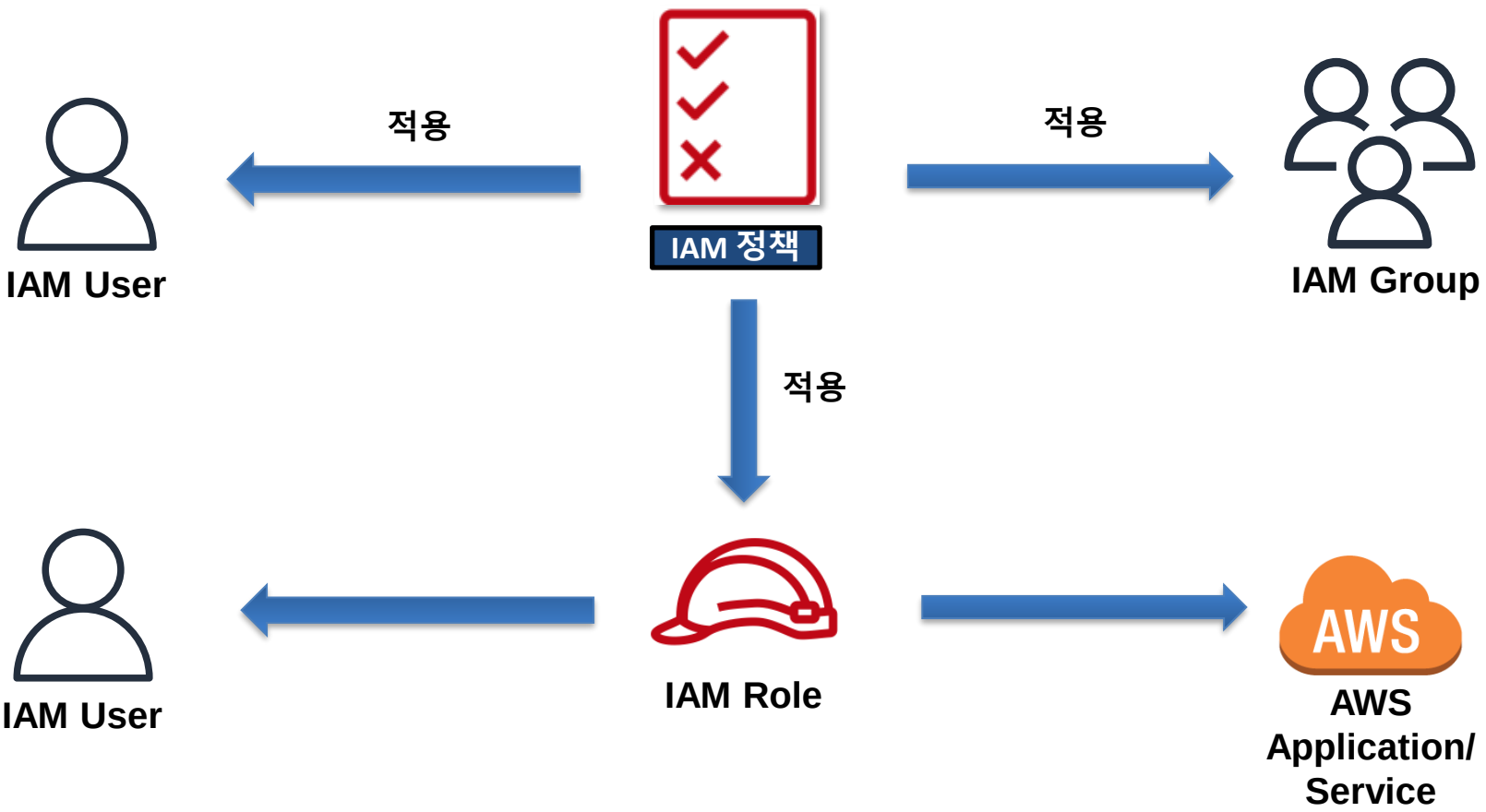
```
{
 "Version": "2012-10-17",
 "Statement":[{
 "Effect":"Allow",
 "Action":["s3:*", "dynamodb:*"],
 "Resource":["arn:aws:dynamodb:ap-northeast-2:123456789123:table/table-name",
 "arn:aws:s3:::bucket-name",
 "arn:aws:s3:::bucket-name/*"]
 },
 {
 "Effect":"Deny",
 "Action":["dynamodb:*","s3:*"],
 "NotResource":["arn:aws:dynamodb:ap-northeast-2:123456789123:table/table-name",
 "arn:aws:s3:::bucket-name",
 "arn:aws:s3:::bucket-name/*"]
 } ]
}
```

명시적인 거부는 허용보다 우선 적용

IAM Group

IAM Group

IAM Group


적용
적용
IAM 정책
IAM User
IAM Group
적용
IAM User
IAM Role
AWS
AWS
Application/
Service

02 사용자 연동

IAM 역할

- **AWS 리소스에 대한 액세스 권한이 없는 사용자나 서비스에 액세스 권한을 위임.**

IAM Role

- **임시 자격 증명**

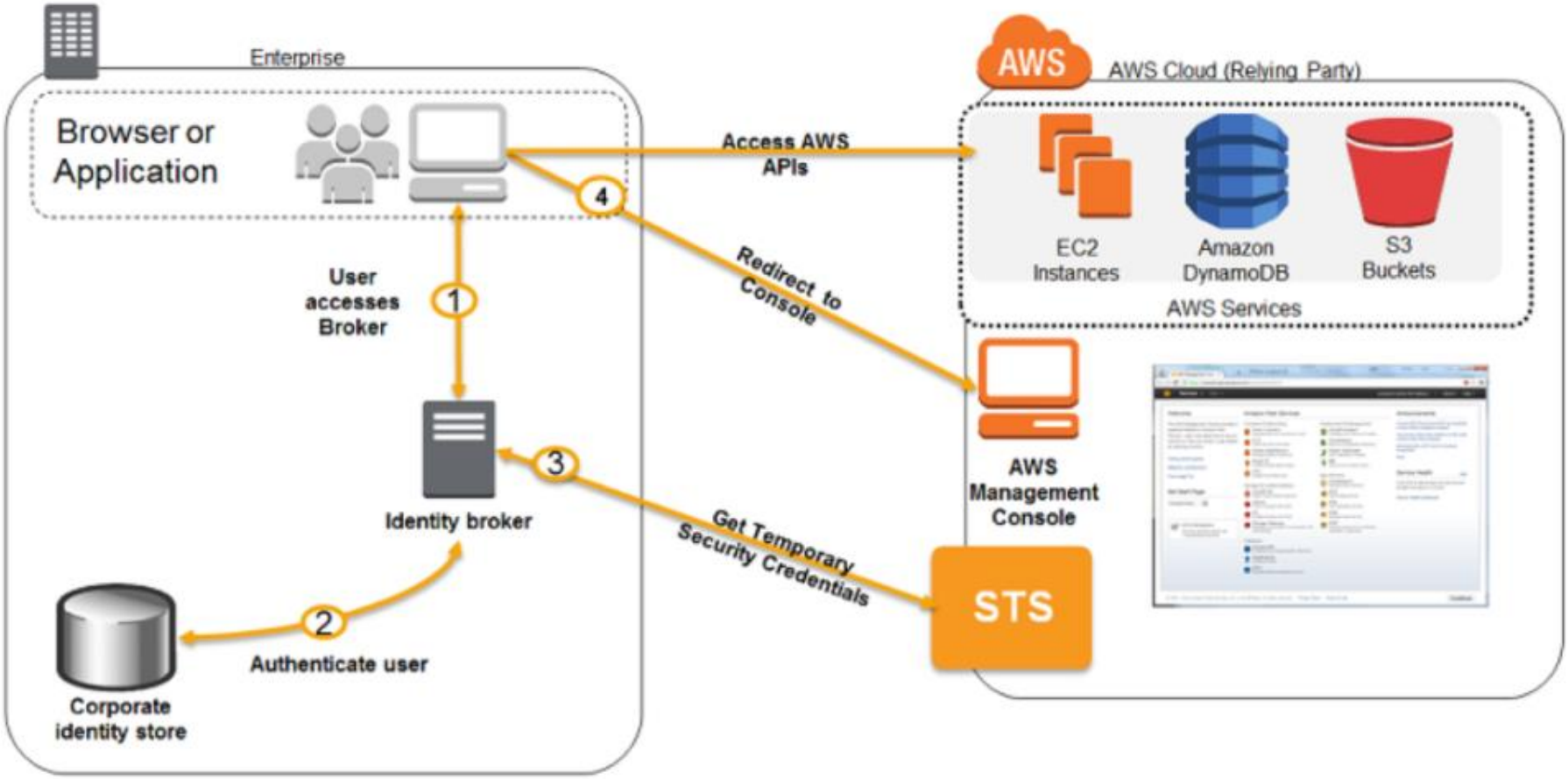
- **IAM 역할을 통해 액세스 권한을 부여할 수 있는 신뢰할 수 있는 엔티티는 다음과 같다.**
  - AWS 서비스 / 다른 AWS 계정 / 웹 ID / SAML 2.0 연동

- **신뢰할 수 있는 엔티티에 역할을 부여하면 AWS STS (Security Token Service)를 사용하여 기간 한정 보안 토큰 발급.**

IAM 역할 사용 시나리오

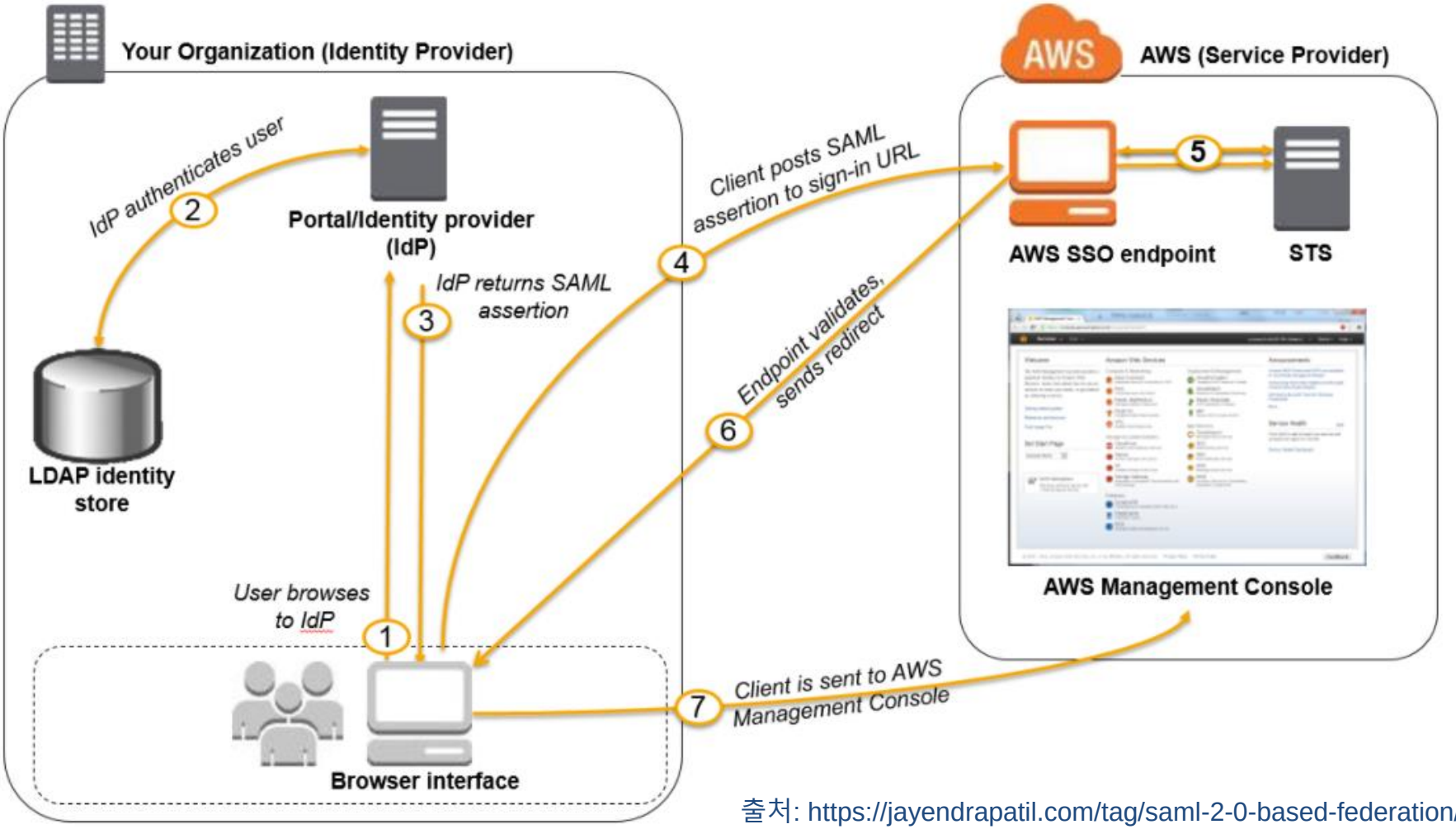
- Amazon EC2 인스턴스에서 실행되는 애플리케이션에 AWS 리소스에 대한 액세스 권한 부여
- 자신의 AWS 계정/다른 AWS 계정의 리소스 액세스 허용
- AWS 서비스에 권한 부여
- 외부 인증 사용자에게 액세스 허용

STS Identity Broker



출처: https://jayendrapatil.com/tag/saml-2-0-based-federation/

SAML



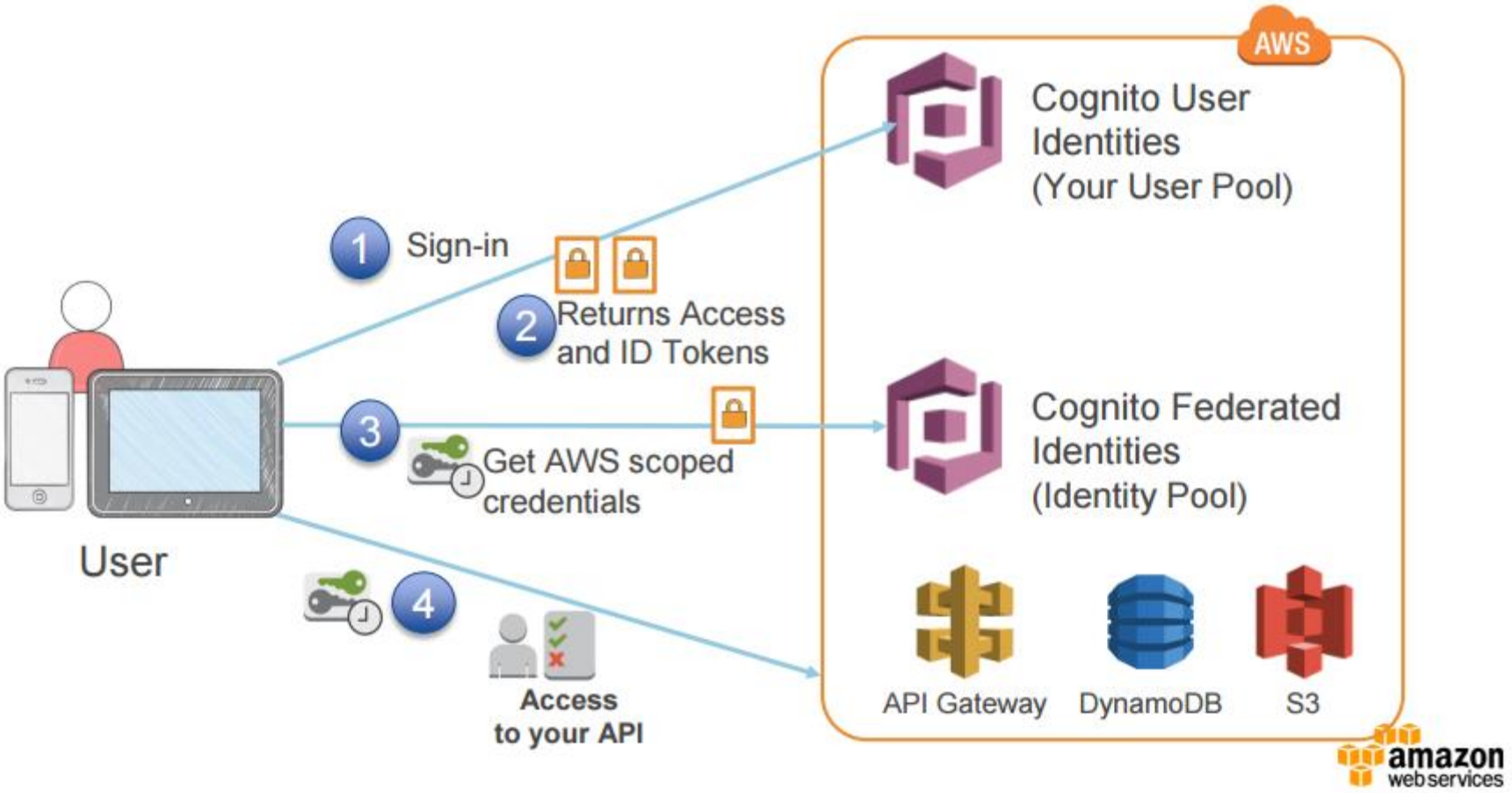
출처: https://jayendrapatil.com/tag/saml-2-0-based-federation/

Amazon Cognito

- **Amazon Cognito**는 웹 및 모바일 앱에 대한 인증, 권한 부여 및 사용자 관리를 제공. (완전 관리형 서비스)
- 사용자 이름/암호로 로그인하거나 타사 계정 (Facebook, Amazon, Google, Apple)을 통해 로그인.
- 두 가지 주요 구성 요소
  - User Pool
  - Identity Pool

Amazon Cognito

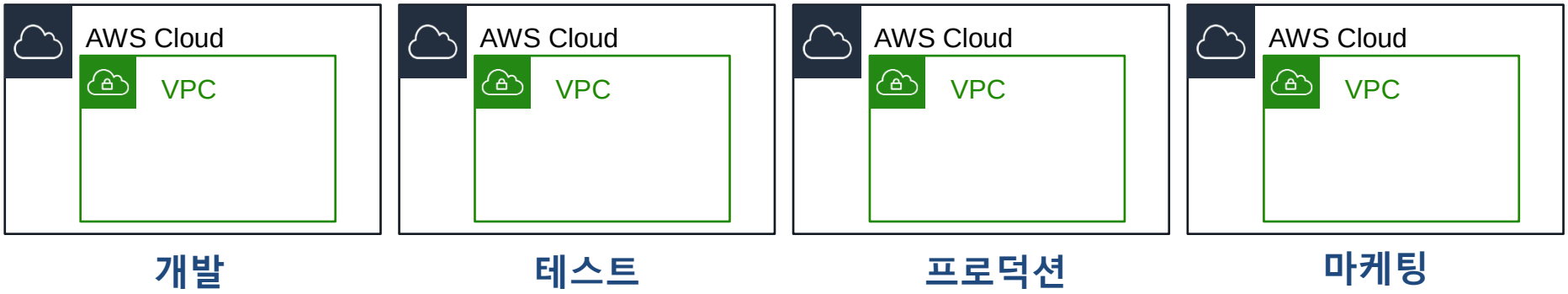
Amazon Cognito



출처: http://blog.jacobmarks.com/2016/12/amazon-cognito-user-pool-admin.html

03 다중 계정 관리

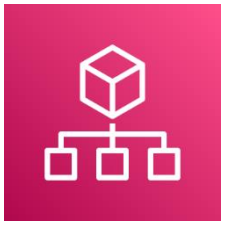
다중 계정 환경



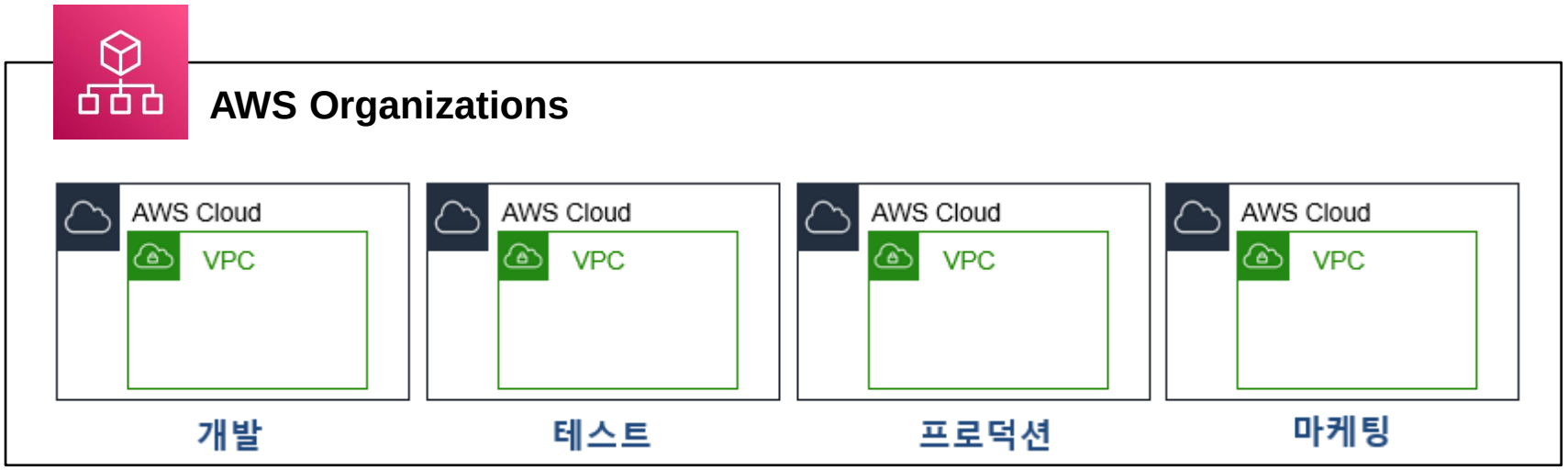
- **AWS 계정**은 권한, 보안, 비용 및 워크로드에 대한 경계 역할을 수행
- 각 부서/팀/환경을 격리할 수 있다.
- 필요한 경우 **교차 계정 액세스**를 통해 다른 계정의 서비스 접근이 가능
- 계정을 어떻게 관리할 것인가를 사전에 검토

AWS Organizations

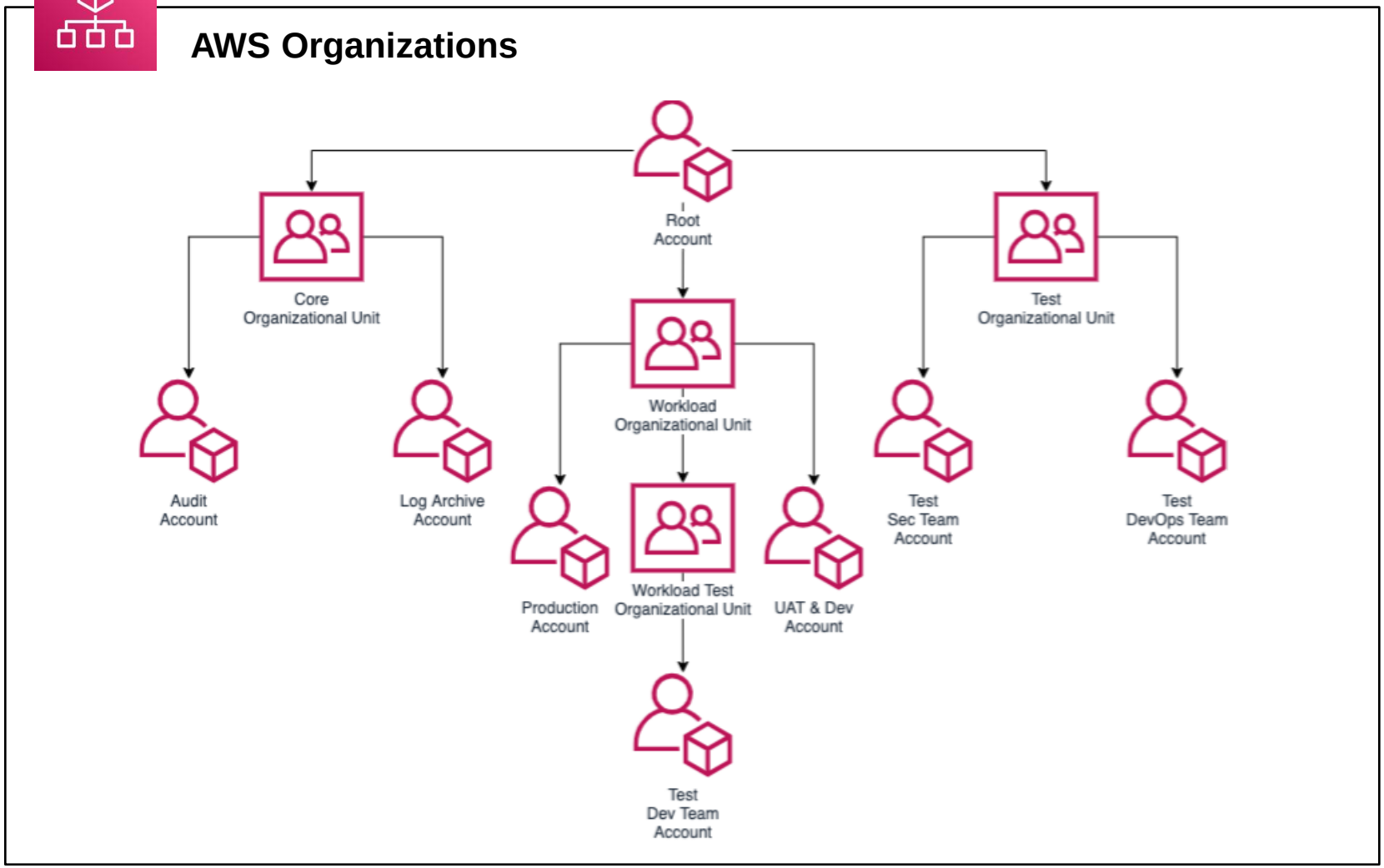
- 여러 계정을 그룹화하여 계정 그룹을 일원화해서 관리할 수 있는 서비스
- 결제 일원화
- 계정에서 사용 가능한 AWS 서비스에 제한 설정



AWS Organizations

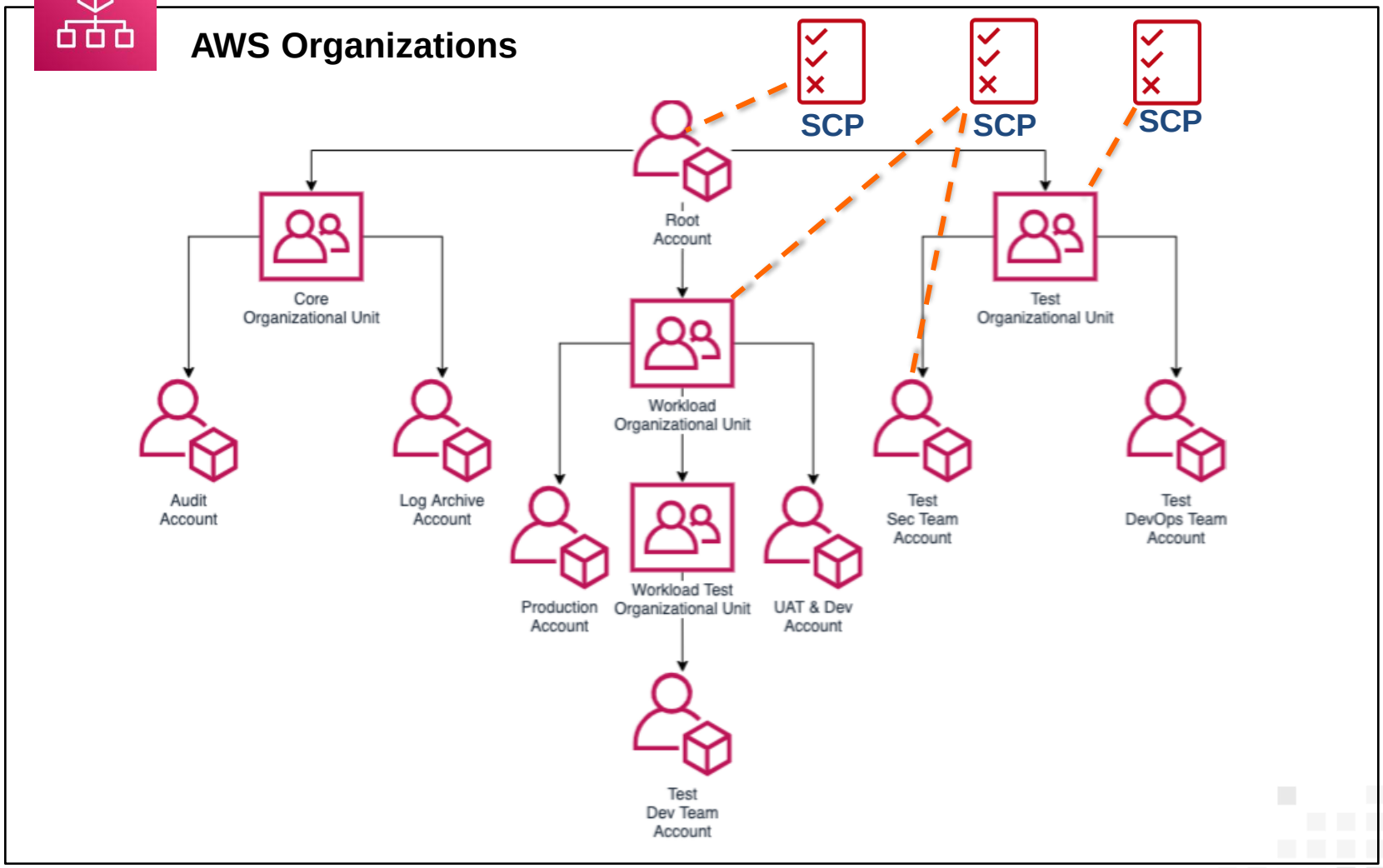


AWS Organizations



출처: https://www.lozanomatheus.com/post/exploring-the-multi-account-aws-environment-and-reducing-costs-with-it

AWS Organizations



출처: https://www.lozanomatheus.com/post/exploring-the-multi-account-aws-environment-and-reducing-costs-with-it