

AWS IAM과 친해지기

김태수

솔루션즈 아키텍트, AWS



AWS 사용 시작



AWS 서비스 사용 – 계정 생성

Account Owner ID (Root Account)

- 구독한 모든 서비스에 대한 접근
- 과금 정보에 대한 접근
- 콘솔 및 API 사용
- 기술 지원 계약 변경



IAM 사용자, 역할, Federated 사용자

- 지정된 일부 서비스에 대한 접근
- 콘솔 및 API 사용
- 기술지원 요청



어플리케이션을 위한 임시 보안 자격 증명

- 지정된 일부 서비스에 대한 접근
- 콘솔 및 API 사용



AWS 서비스 사용 – Console 기반 작업

Run a command Actions

Filter by attributes

	Command ID	Instance ID	Document name	Status	Requested date	Comment
<input checked="" type="checkbox"/>	65555b90-ee60-45...	i-8fd6aa30	AWS-RunPowerSh...	Success	October 21, 2015 at...	Listing services
<input type="checkbox"/>	65555b90-ee60-45...	i-d583f76a	AWS-RunPowerSh...	Success	October 21, 2015 at...	Listing services
<input type="checkbox"/>	65555b90-ee60-45...	i-8ed6aa31	AWS-RunPowerSh...	Success	October 21, 2015 at...	Listing services
<input type="checkbox"/>	ca4b10c6-cee1-437...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	getting list of pro
<input type="checkbox"/>	561e5f4a-27d2-419...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	ipconfig on the b

Command ID: 65555b90-ee60-4520-9dc3-e42e94445469 Instance ID: i-8fd6aa30

Description Output

Command ID	65555b90-ee60-4520-9dc3-e42e94445469	Instance ID	i-8fd6aa30
Document name	AWS-RunPowerShellScript	Status	Success
Date requested	October 21, 2015 at 3:56:59 PM UTC-7	Comment	Listing services
Output S3 bucket	run-command-test	Document parameters	



👍 쉽게 시작할 수 있다.

😞 반복작업에 적합하지 않다.

😞 시간이 오래 걸린다.

수작업

High level

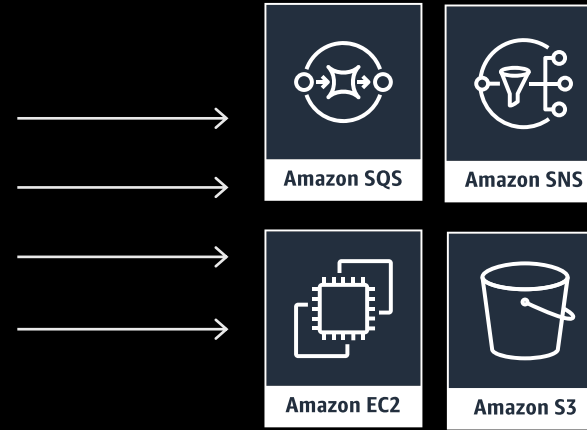
Low level

AWS 서비스 사용 – Script 기반 작업

```
require 'aws-sdk-ec2'

ec2 = Aws::EC2::Resource.new(region: 'us-west-2')

instance = ec2.create_instances({
  image_id: 'IMAGE_ID',
  min_count: 1,
  max_count: 1,
  key_name: 'MyGroovyKeyPair',
  security_group_ids: ['SECURITY_GROUP_ID'],
  instance_type: 't2.micro',
  placement: {
    availability_zone: 'us-west-2a'
  },
  subnet_id: 'SUBNET_ID',
  iam_instance_profile: {
    arn: 'arn:aws:iam::' + 'ACCOUNT_ID' + ':instance-profile/aws-opsworks-ec2-role'
  }
})
```



👍 반복 작업에 적합하다.

👍 원하는 항목에 대한 수정이 용이하다.

😞 리소스의 준비 상태 확인이 어렵다.

😞 문제 발생 시 복원이 어렵다.

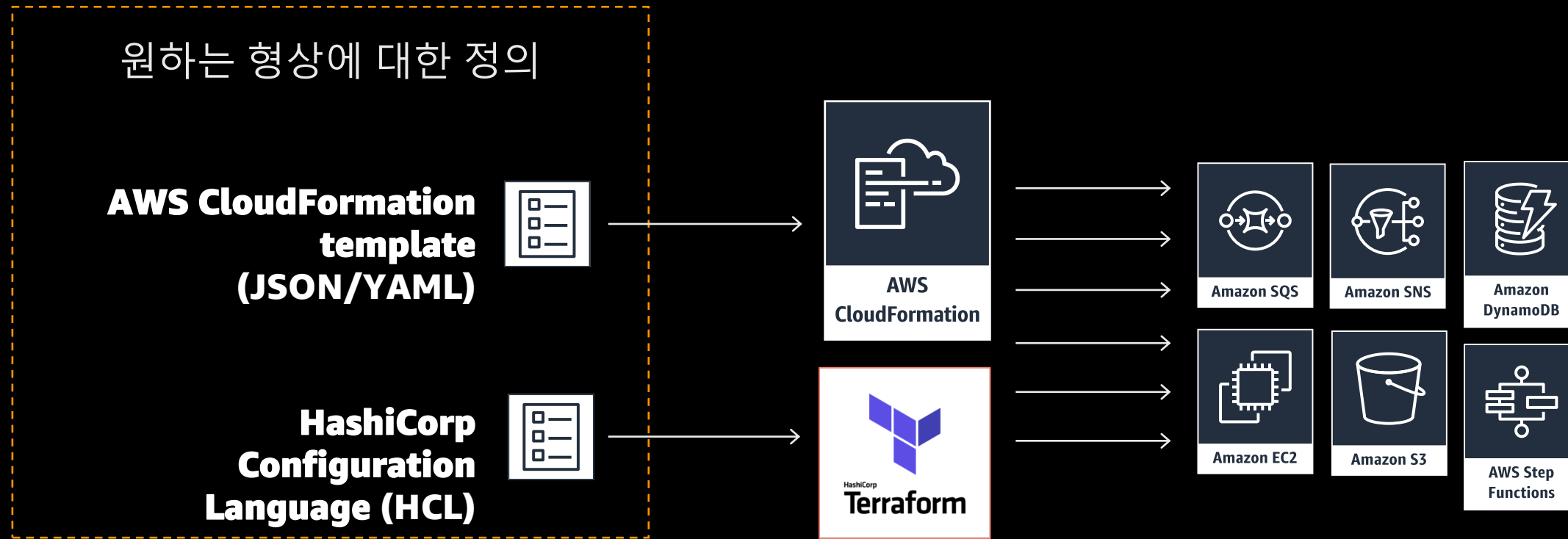
Script 기반

수작업

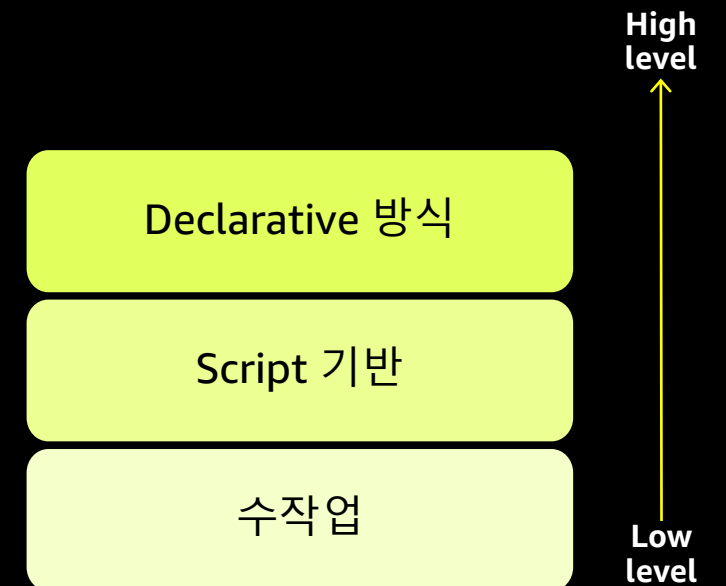
High level

Low level

AWS 서비스 사용 - 프로비저닝 엔진 사용

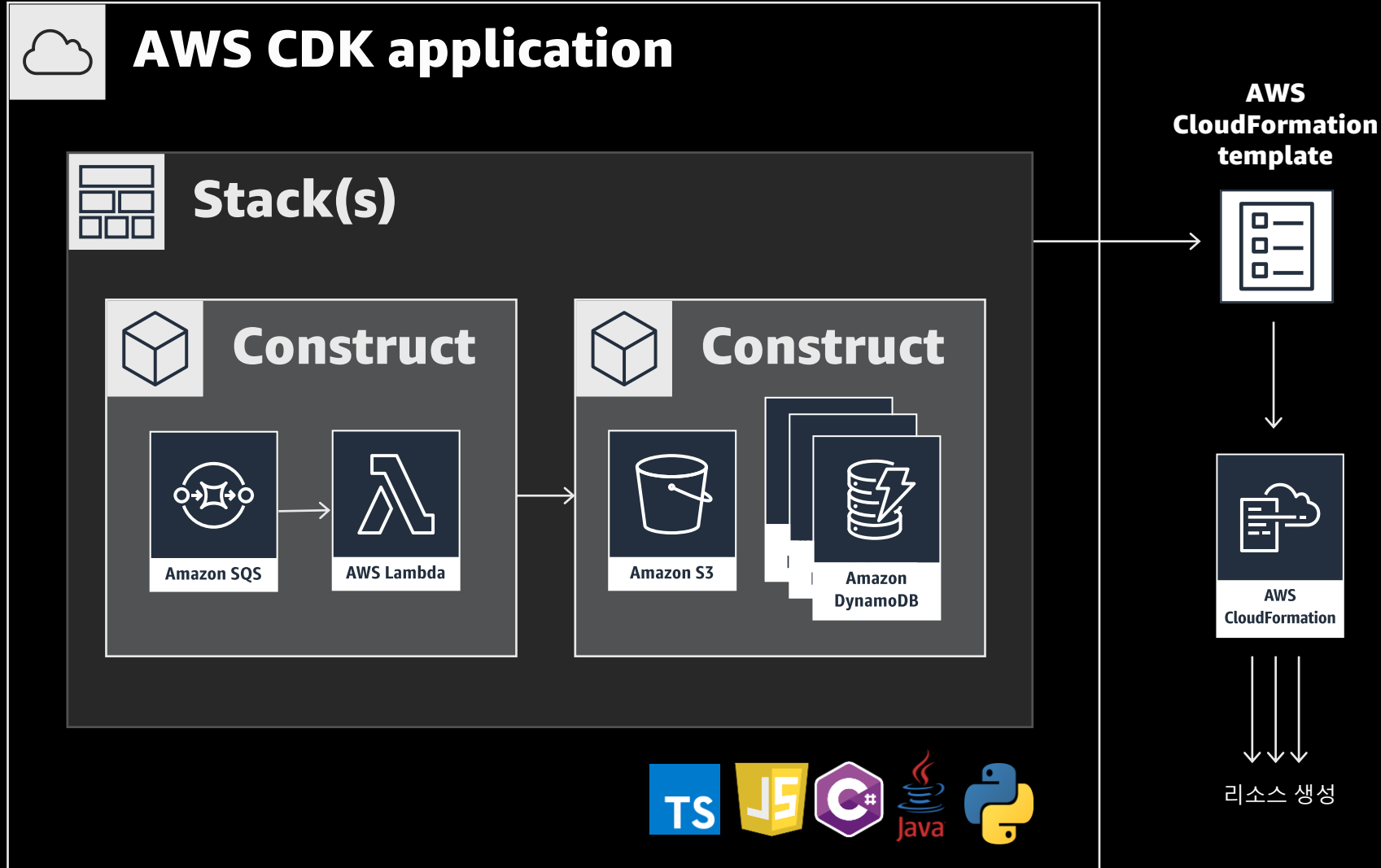


- 👍 자동화 구현에 용이하다.
- 👍 반복작업에 적합하다.
- 👍 에러 발생 시 원복이 쉽다.
- 😓 최초 구현이 복잡하다.



AWS 서비스 사용 – CDK 사용

Cloud Development Kit



👍 코드 기반

👍 원하는 형상에 대한 정의

😞 초기 코딩의 복잡성

Component 화

Declarative 방식

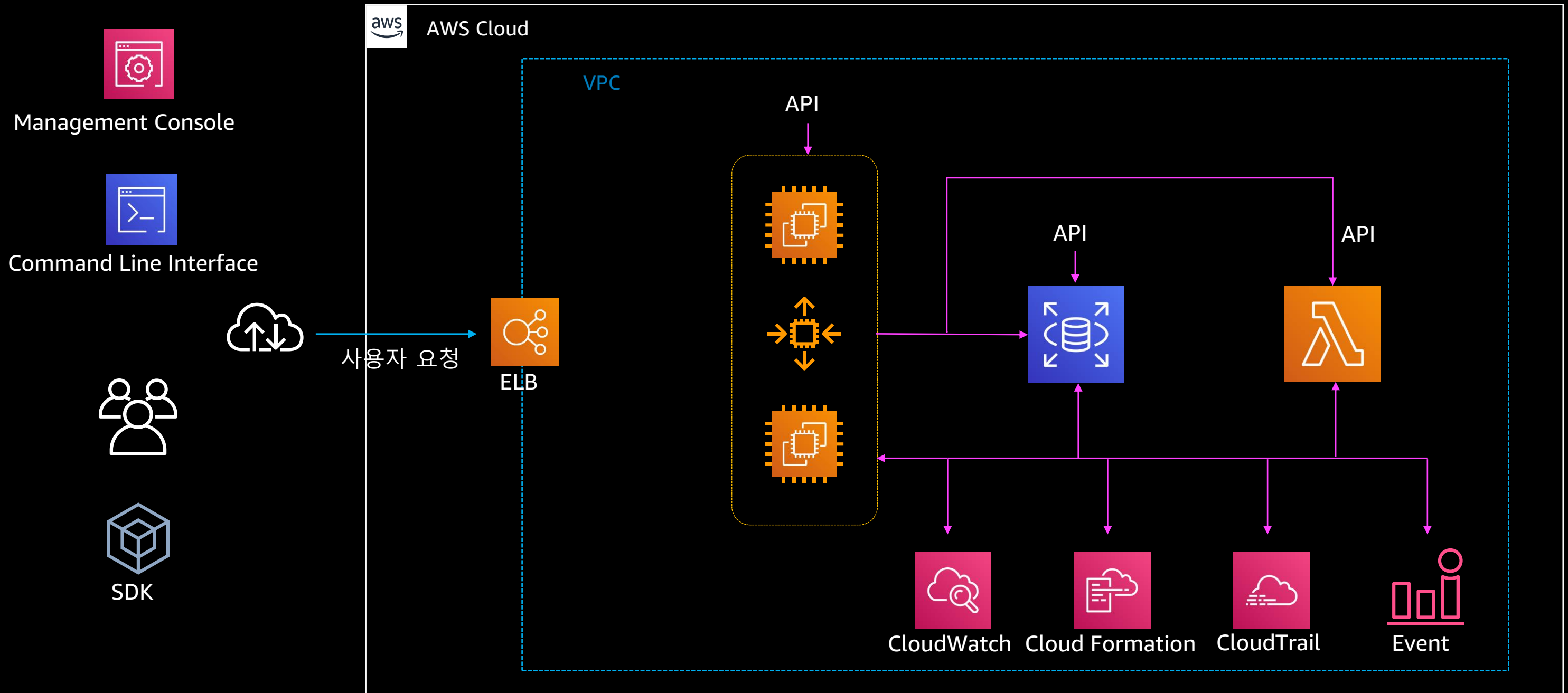
Script 기반

수작업

High level

Low level

결국 중요한 건 - API



AWS에서의 API 인증



```
POST /API-Request HTTP/1.1
Host: any-aws-services.us-east-1.amazonaws.com
X-Amz-Date: 20190418T123600Z
Content-Type: application/x-www-form-urlencoded
Authorization: AWS4-HMAC-SHA256
Credential=ASIAXXXXXXXX/20181126/us-east-1/sqs/aws4_request,SignedHeaders=content-type;host;x-amz-date,Signature=b97dfa904a5beff61c982a1b6f458b799221646efd99d3219ec94cdf2500
```

Access Key ID로 요청주체 (IAM Principal)를 인증

비밀키로 생성한 HMAC 서명값(Sig v4) 검증

Signature 생성 코드 예제
(Python)



```
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()

def getSignatureKey(key, dateStamp, regionName, serviceName):
    kDate = sign(("AWS4" + key).encode("utf-8"), dateStamp)
    kRegion = sign(kDate, regionName)
    kService = sign(kRegion, serviceName)
    kSigning = sign(kService, "aws4_request")
    return kSigning
```

AWS Identity and Access Management

AWS Identity and Access Management (IAM)

Identity

Access Management



IAM 은 AWS 전체의 권한 통제 시스템.

I is for Identity : 실 사용자 → IAM 사용자(User)

AWS 액세스 유형 선택

해당 사용자가 AWS에 액세스하는 방법을 선택합니다. 마지막 단계에서는 액세스 키와 자동 생성된 비밀번호가 제공됩니다. [자세히 알아보기](#)

IAM 사용자의 AWS 액세스 유형

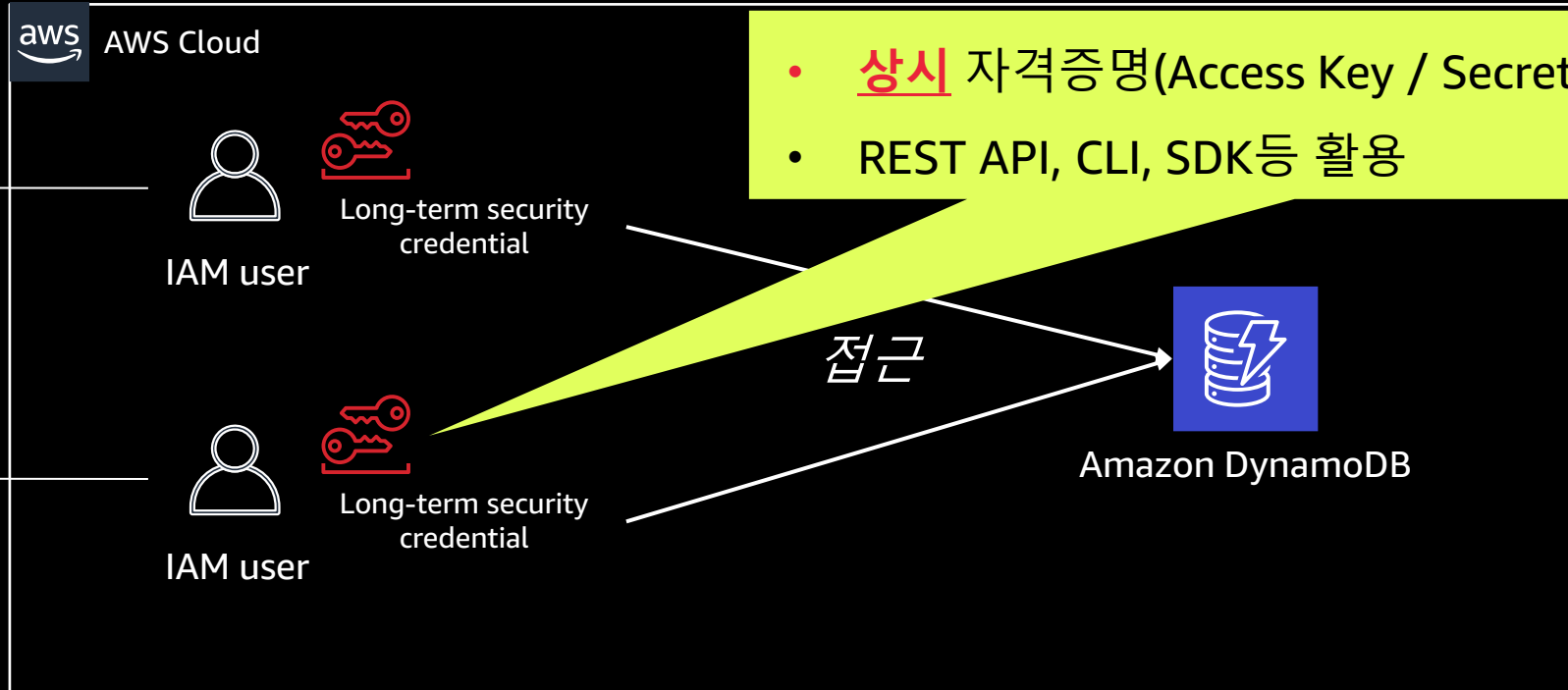
- 액세스 유형*
- ☐ **프로그래밍 방식 액세스**
AWS API, CLI, SDK 및 기타 개발 도구에 대해 액세스 키 ID 및 비밀 액세스 키 을(를) 활성화합니다.
 - ☐ **AWS Management Console 액세스**
사용자가 AWS Management Console에 로그인할 수 있도록 허용하는 비밀번호 을(를) 활성화합니다.



Human user



Human user



IAM User

IAM Role

Policy

I is for Identity : AWS IAM Role

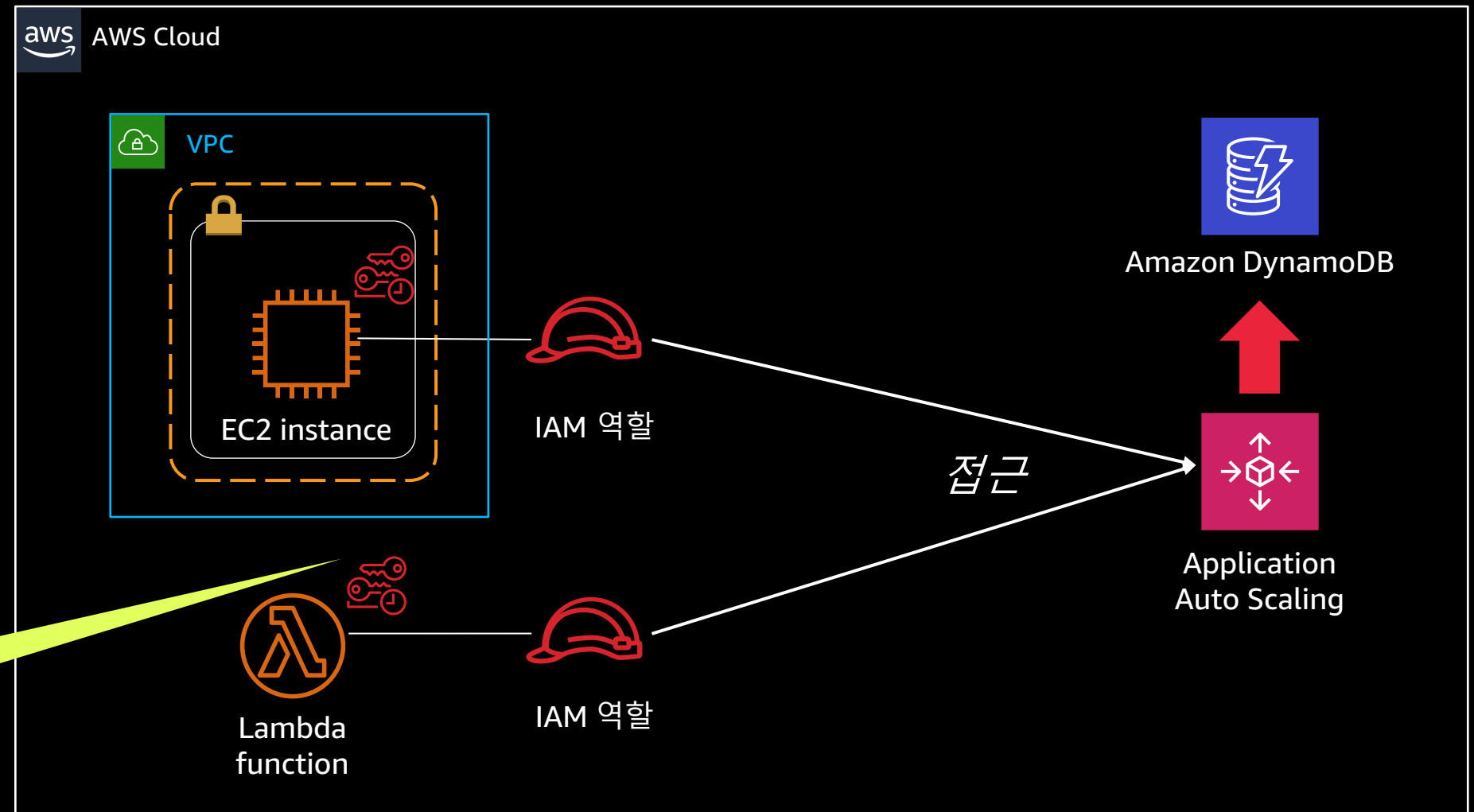
- 정의된 권한 범위 내 AWS API 를 사용할 수 있는 임시 자격 증명
- **IAM Role**를 사용하면 사용자 권한을 공유하거나 매번 필요한 권한 부여 불 필요
- **IAM Role** 활용 예
 - EC2 Role
 - Federations : Cross-Account, SAML2.0, Web Identity Provider

IAM User

IAM Role

Policy

I is for Identity : 인스턴스/타서비스 → IAM role



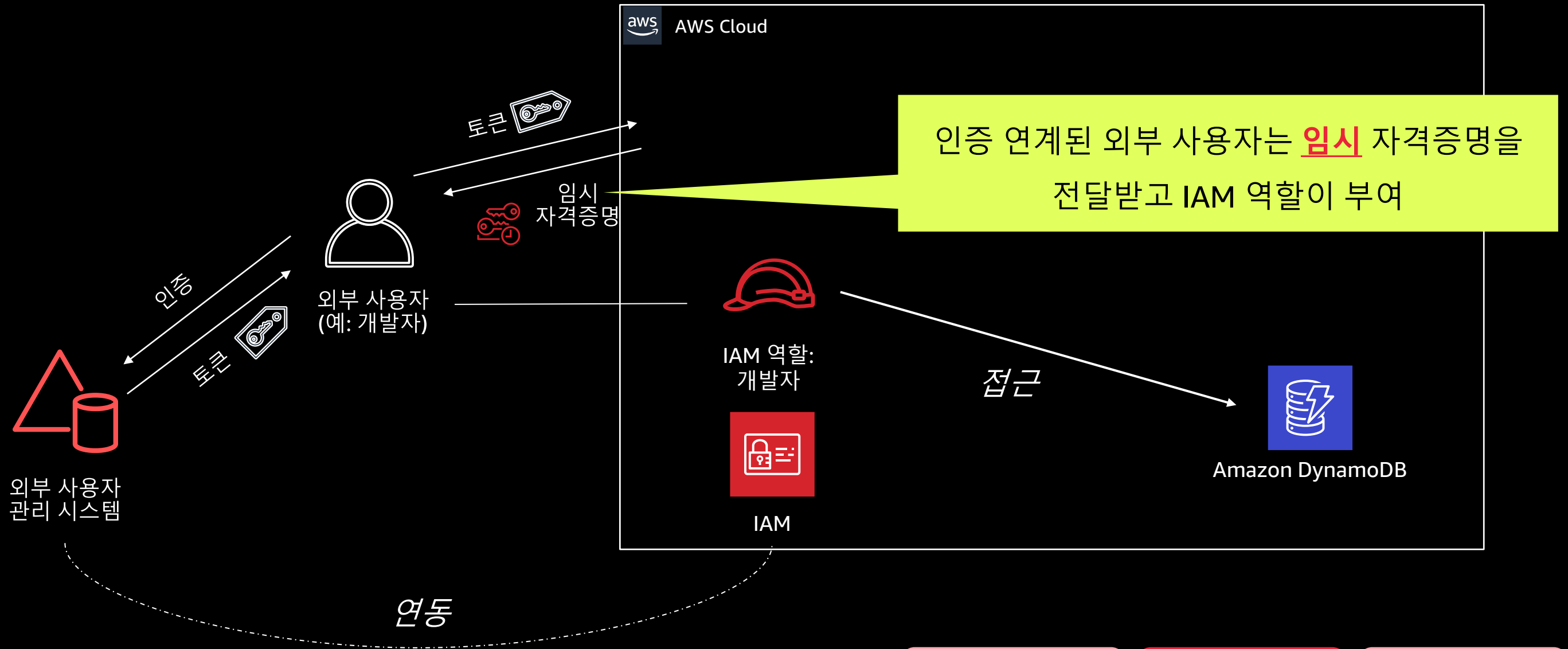
IAM 역할에는 **임시** 자격증명 부여

IAM User

IAM Role

Policy

I is for Identity: 외부 사용자 → IAM role

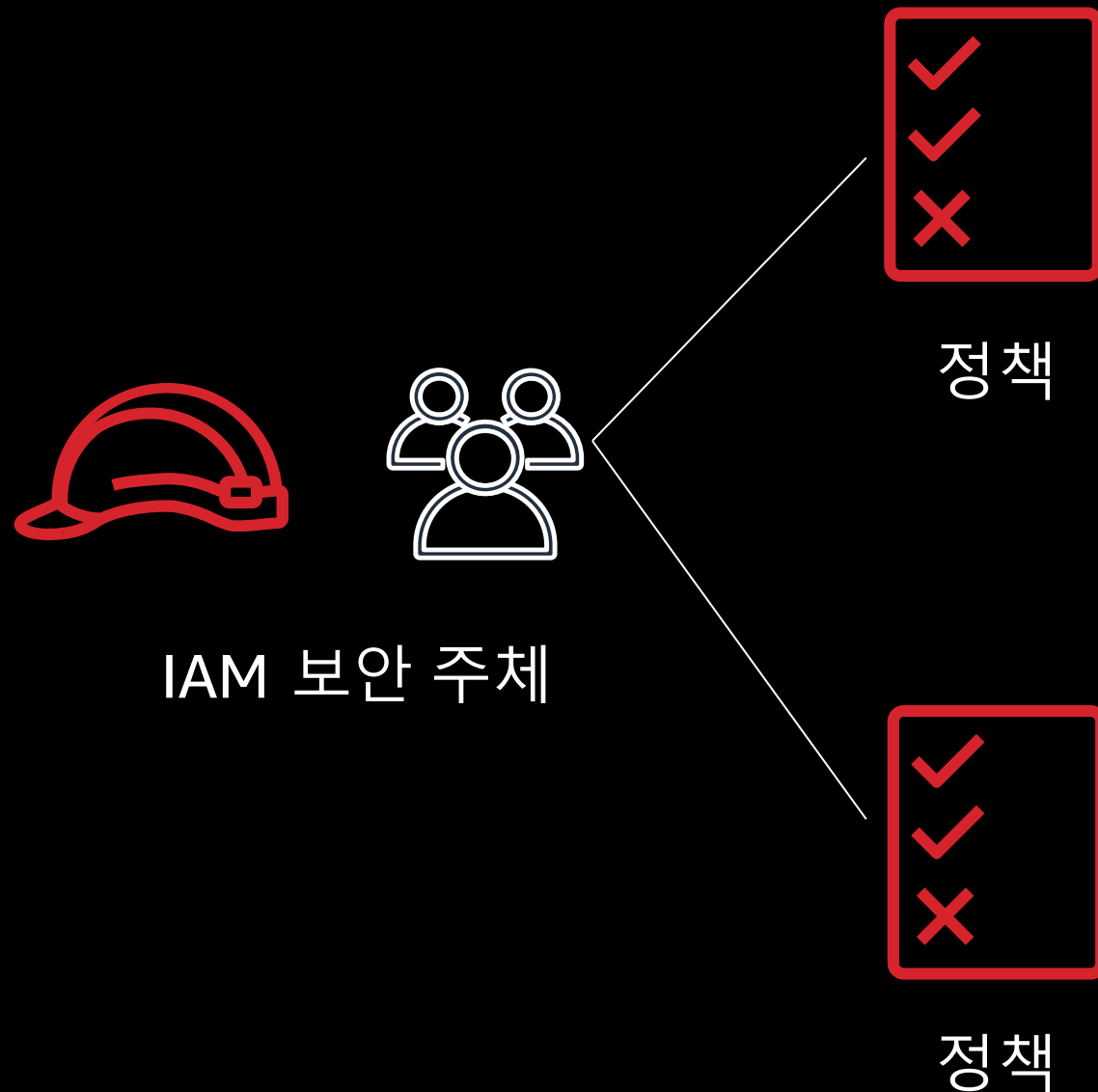


IAM User

IAM Role

Policy

AM is Access Management: AWS에서의 인가



- 모든 AWS 서비스는 **접근제어 정책**을 기반으로 인가됨
- 매 API호출 시, 적용된 정책을 통해 인가 수행
- 정책은 IAM 역할/사용자/그룹, AWS 리소스, 임시 자격증명 세션, OU 등에 적용할 수 있음
- AWS Root 어카운트는 기본적으로 AWS 리소스에 대한 모든 권한을 가짐.
- AWS 정책은 기본 디폴트가 **Deny**이고, **명시적 Allow < 명시적 Deny**의 우선순위.

IAM User

IAM Role

Policy

IAM Policy 의 구조

```
{  
  "Statement": [{  
    "Effect": "Allow or Deny",  
    "Principal": "principal",  
    "Action": "action",  
    "Resource": "arn",  
    "Condition": {  
      "condition": {  
        "key": "value" }  
      }  
    }  
  ]  
}
```

Effect – 명시된 정책에 대한 허용 혹은 차단

Principal – 접근을 허용 혹은 차단하고자 하는 대상
"Principal": "AWS": "arn:aws:iam::123456789012:user/username"

Action – 허용 혹은 차단하고자하는 접근 타입
"Action": "s3:GetObject"

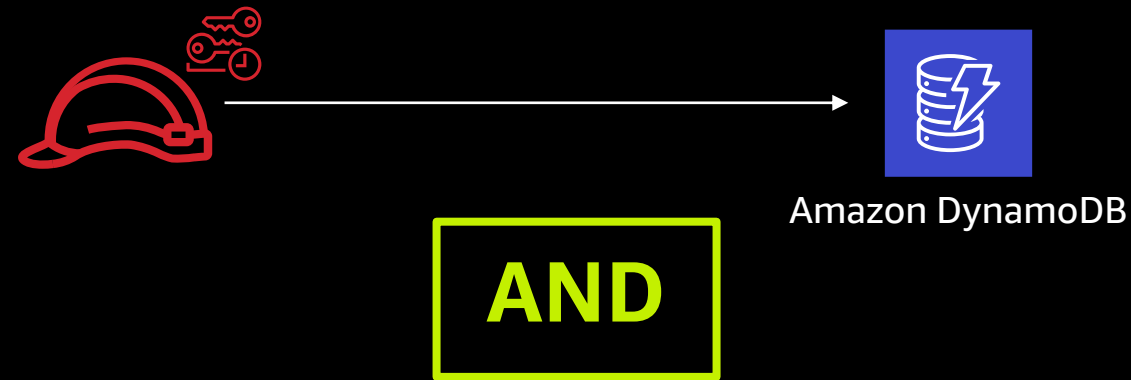
Resource – 요청의 목적지가 되는 서비스
"Resource": "arn:aws:sqs:us-west-2:123456789012:queue1"

Condition – 명시된 조건 유효하다고 판단될 수 있는 조건
"StringEqualsIfExists": {"aws:RequestTag/project": ["Pickles"]}

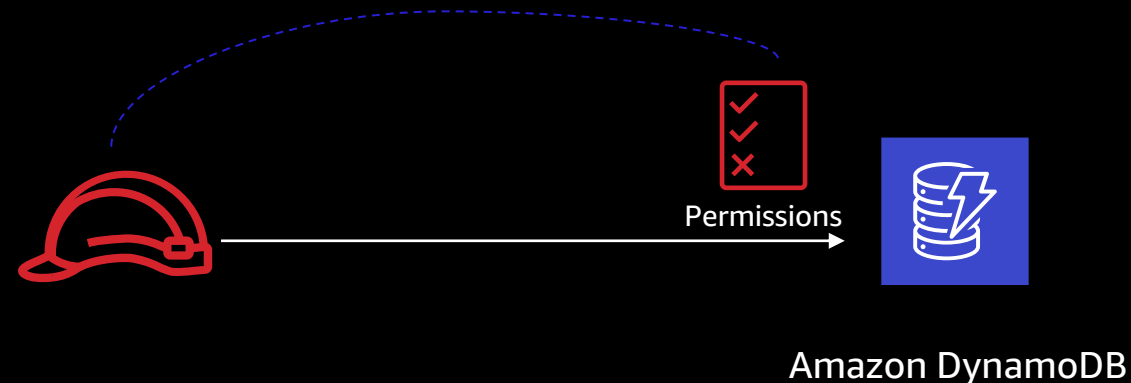
AM is Access Management: 요청의 성공 조건

제출된 요청이 성공하기 위해선,

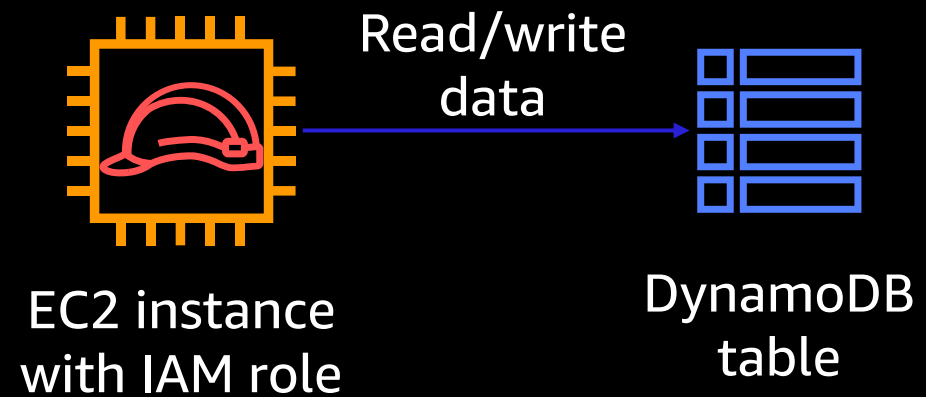
- IAM 보안 주체의 적법한 서명값이 포함되어 있고(인증),



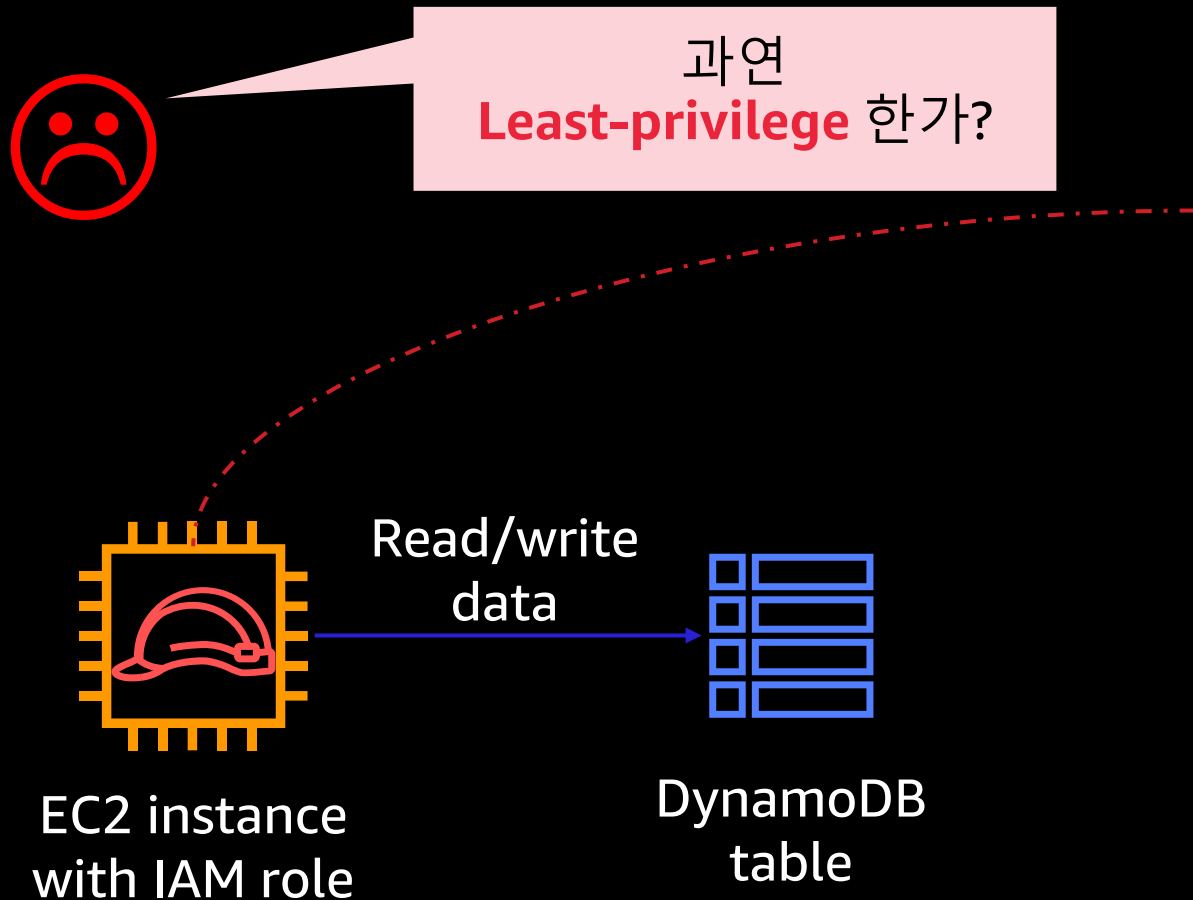
- 정책(Policy)에 의해 해당 요청이 정확하게 인가되어야 함.



예제 1: DynamoDB로부터 데이터 읽기



예제 1: DynamoDB로부터 데이터 읽기

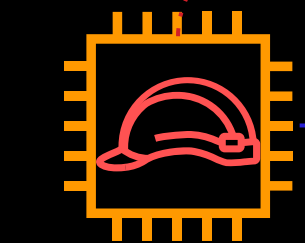


```
{  
  "Effect": "Allow",  
  "Action": "*",  
  "Resource": "*"  
}
```

모든 액션

모든 리소스

예제 1: DynamoDB로부터 데이터 읽기



EC2 instance
with IAM role

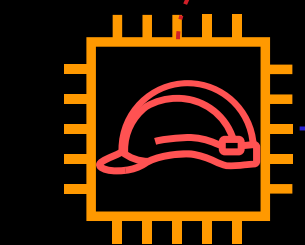
Read/write
data



DynamoDB
table

```
{  
  "Effect": "Allow",  
  "Action": "dynamodb:*",  
  "Resource": "*"  
}
```

예제 1: DynamoDB로부터 데이터 읽기



EC2 instance
with IAM role

Read/write
data



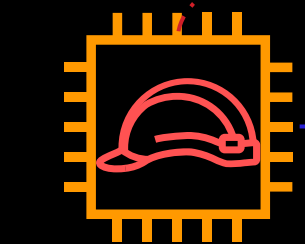
DynamoDB
table

```
{  
  "Effect": "Allow",  
  "Action": [  
    "dynamodb:GetItem",  
    "dynamodb:PutItem"  
  ],  
  "Resource": "*" }  
}
```

구체적인
DynamoDB 액션

모든 리소스

예제 1: DynamoDB로부터 데이터 읽기



EC2 instance
with IAM role

Read/write
data

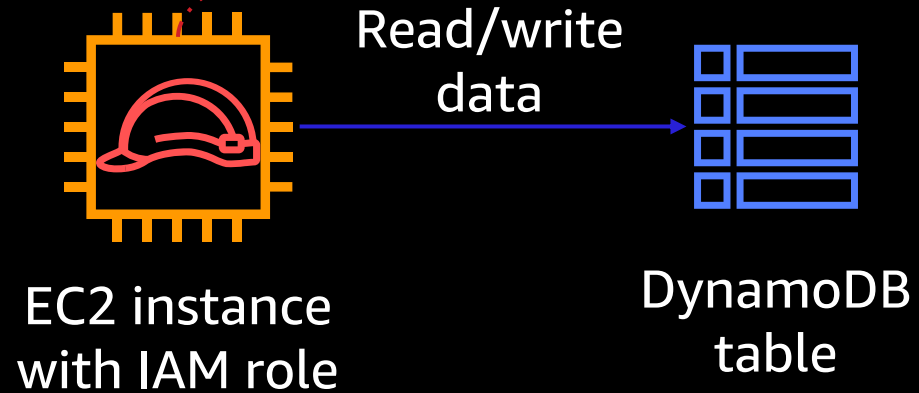


DynamoDB
table

```
{  
  "Effect": "Allow",  
  "Action": [  
    "dynamodb:GetItem",  
    "dynamodb:PutItem"  
  ],  
  "Resource": [  
    "arn:aws:dynamodb:us-east-2:111122223333:table/MyTable"  
  ]  
}
```

특정 Dynamodb 리소스

예제 1: DynamoDB로부터 데이터 읽기



```
{  
  "Effect": "Allow",  
  "Action": [  
    "dynamodb:GetItem",  
    "dynamodb:PutItem"  
  ],  
  "Resource": [  
    "arn:aws:dynamodb:us-east-2:111122223333:table/MyTable"  
  ],  
  "Condition": {  
    "IpAddress": {  
      "aws:SourceIp": "1.1.1.1"  
    },  
  },  
}
```

특정 조건

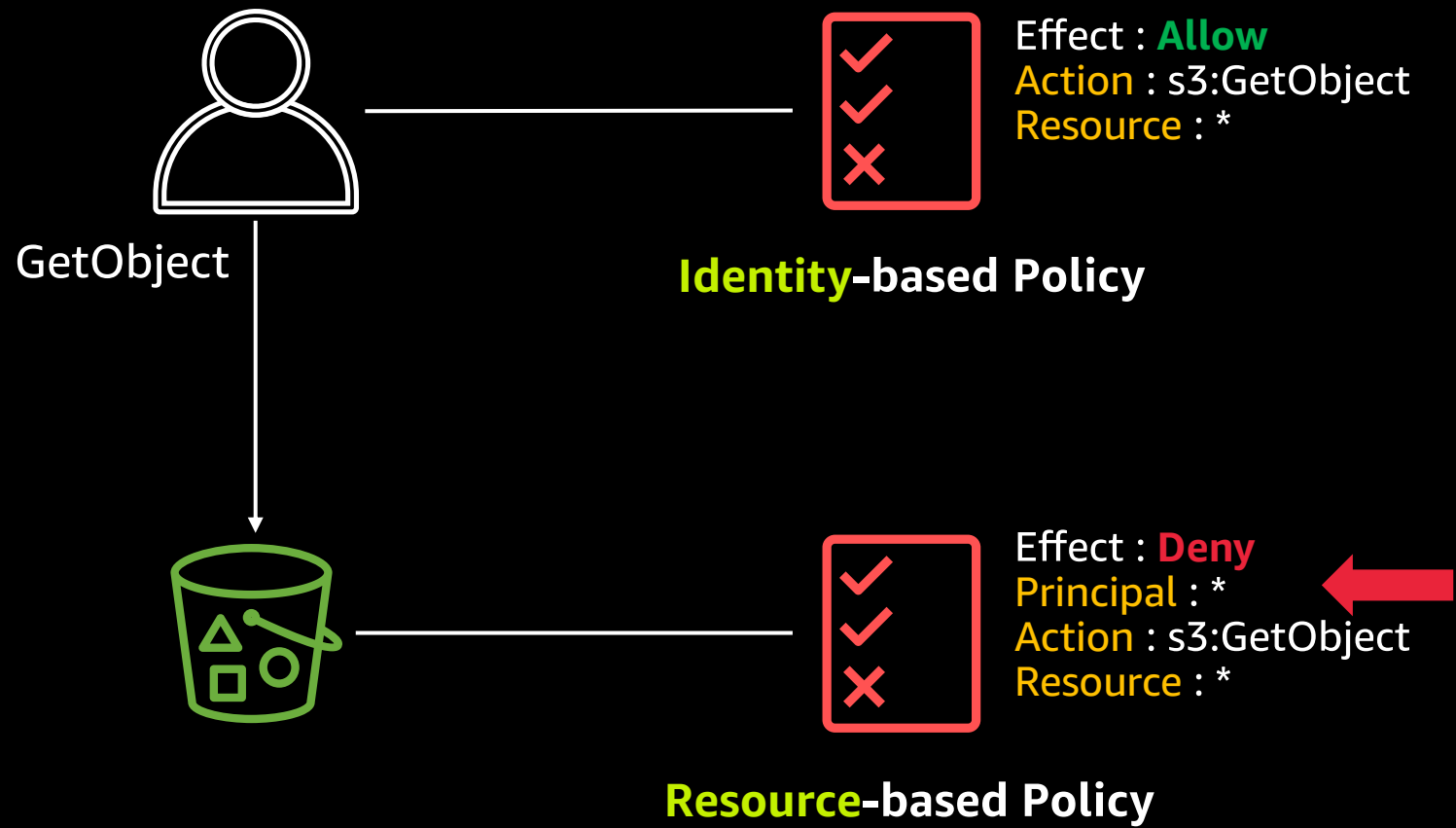
IAM 정책의 종류

정책	설명	포맷	정의 및 관리
Identity-based 정책	IAM 보안 주체(IAM 사용자, IAM 그룹의 사용자 집합, IAM 역할)에 할당되어 해당 주체의 권한을 규정	JSON	IAM
Resource-based 정책	정책이 할당될 리소스를 기준으로 어떤 보안 주체가 할 수 있(없)는 작업을 규정.	JSON	개별 서비스들
IAM Permission Boundary 정책	IAM 보안 주체 별로 획득할 수 있는 권한의 최대치를 규정	JSON	IAM
Organization SCP	Organization의 OU 또는 개별 어카운트 별로 권한의 최대치를 규정 주로 Root 어카운트의 권한을 제한 시킬 때 사용.	JSON	Organization
Session 정책	임시 자격증명의 기존 퍼미션을 해당 세션에 대해서만 제한할 때 사용 AssumeRole*, GetFederationToken API의 파라미터 로 전달됨.	JSON	STS
ACL 정책	리소스 기준으로 정의하며, 주로 Cross-Account 간의 리소스 공유시, 보안 주체에 대한 접근을 규정	XML	개별 서비스들
Endpoint 정책	VPC G/W Endpoint에 적용되는 접근제어 정책. 일종의 Resource-based 정책 임.	JSON	VPC

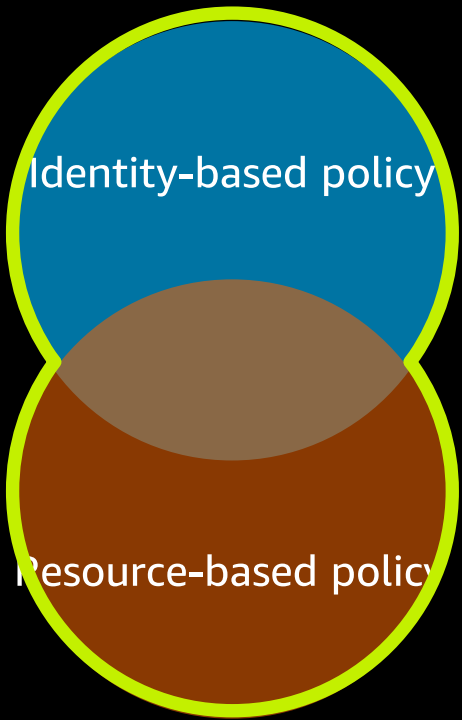
상세 정보 : https://docs.aws.amazon.com/ko_kr/IAM/latest/UserGuide/access_policies.html



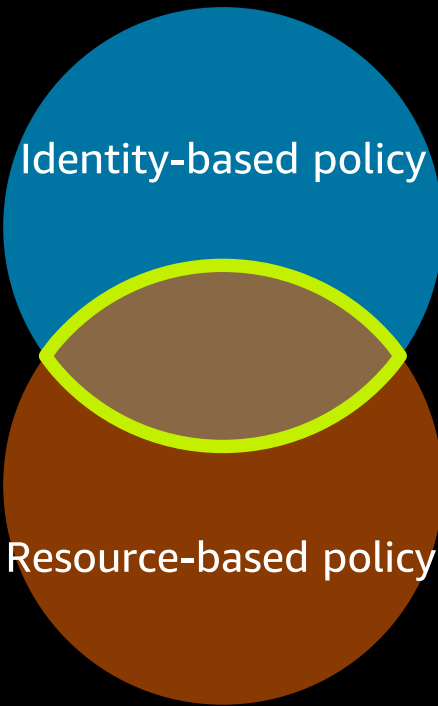
IAM 정책의 종류



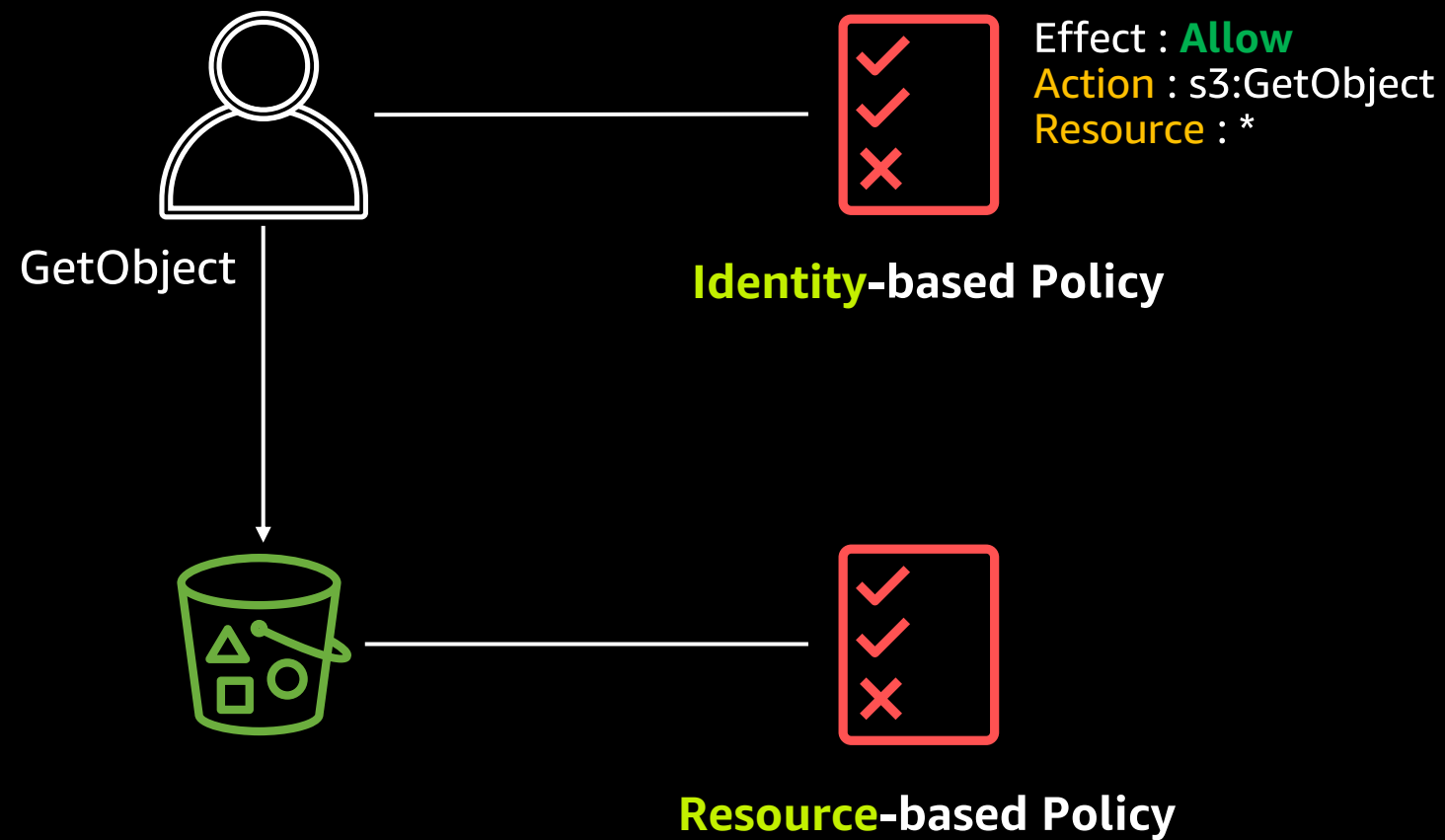
In-Account



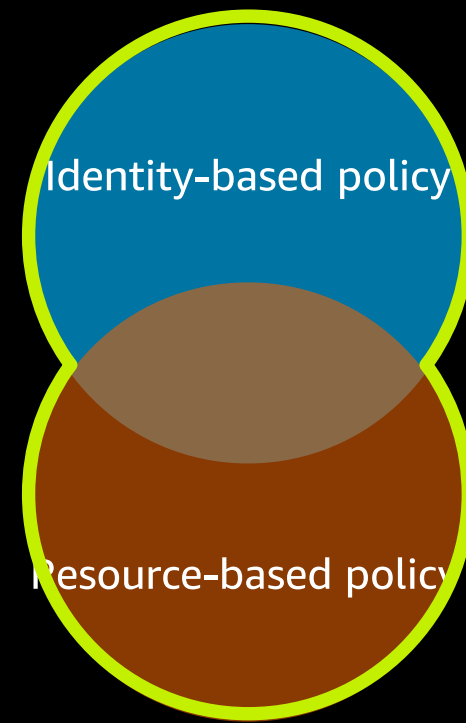
Cross-Account



IAM 정책의 종류

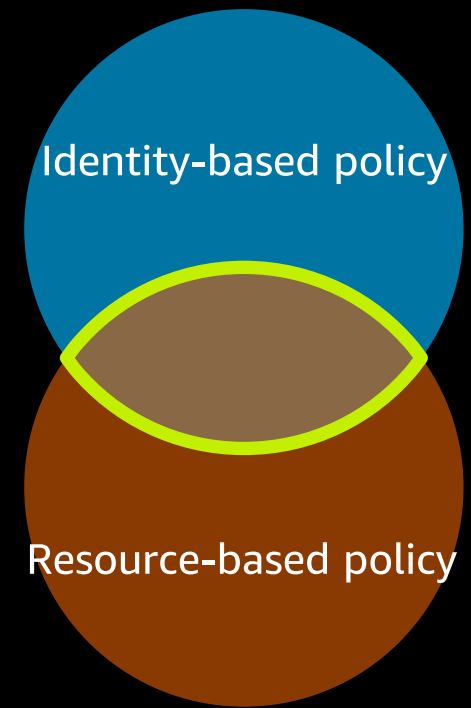


In-Account



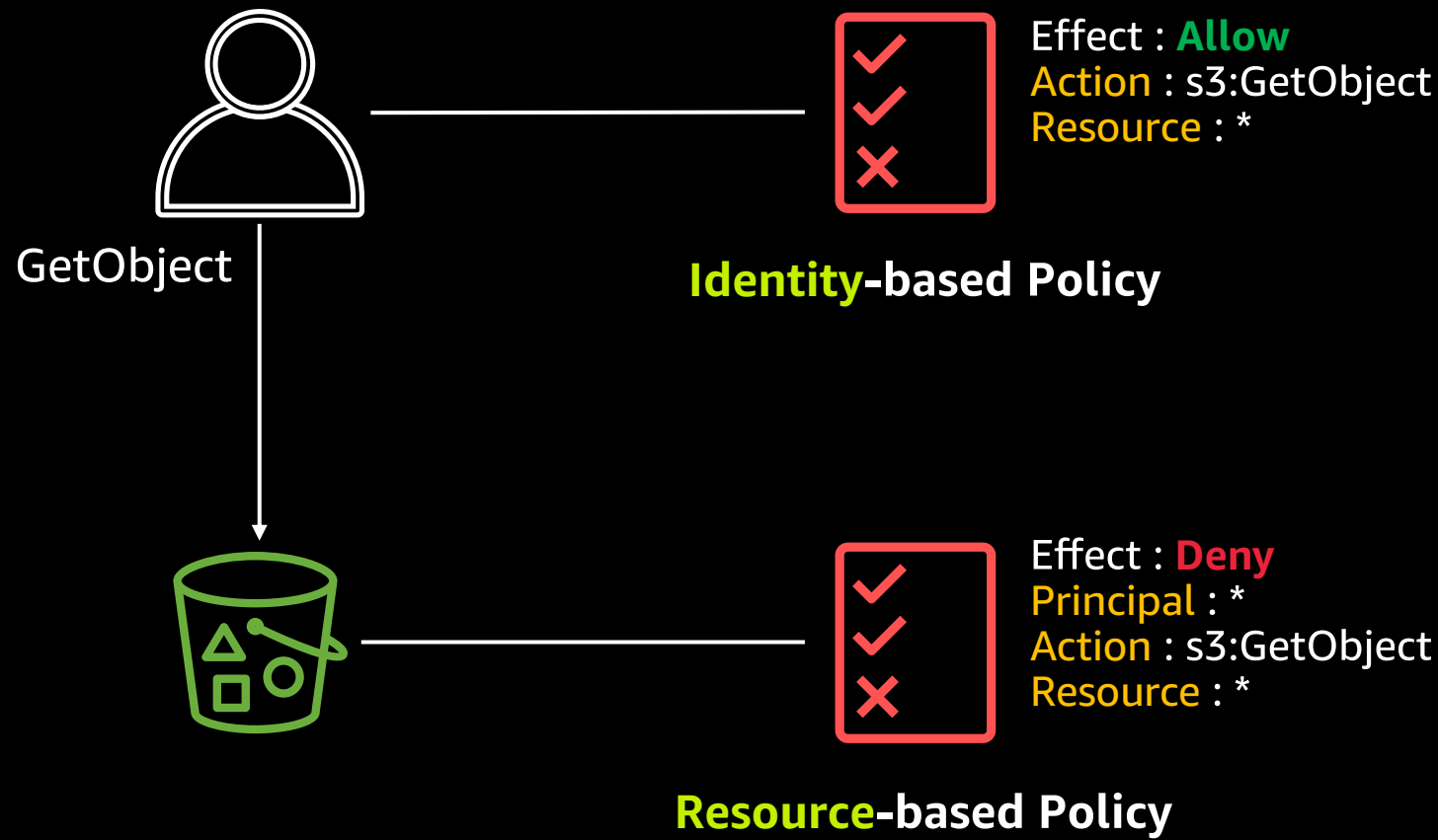
GetObject **O**

Cross-Account

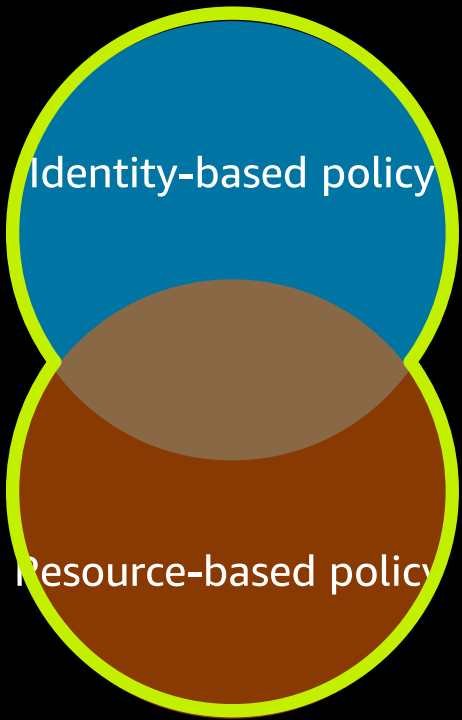


GetObject **X**

IAM 정책의 종류

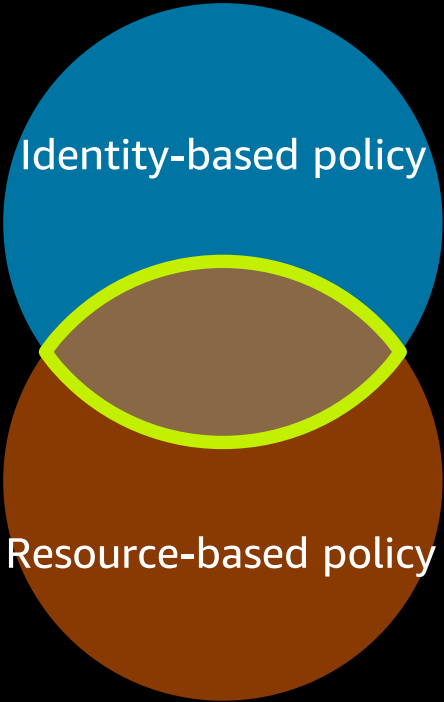


In-Account



GetObject **X**

Cross-Account



GetObject **X**

AWS IAM Best practice

IAM 모범사례

1. AWS 계정 root 사용자 액세스 키 잠금
2. 권한 있는 사용자에게 MFA 활성화
3. 개별 IAM 사용자 만들기
4. 그룹을 사용하여 IAM 사용자에게 권한을 할당
5. 최소 권한 부여
6. 서비스 권한 제어에 역할 사용
7. 역할을 사용하여 권한 위임
8. 자격 증명을 정기적으로 교체
9. 보안 강화를 위해 정책 조건 사용
10. AWS 계정의 활동 모니터링 및 감사

더 자세한 모범사례: https://docs.aws.amazon.com/ko_kr/IAM/latest/UserGuide/best-practices.html

루트 잠그기

- AWS 계정 root 사용자 액세스 키 잠금

Created	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Dec 11th 2020		N/A	N/A	N/A	Active	Make Inactive Delete

Create New Access Key

- 권한 있는 사용자에게 대해 MFA 활성화

Manage MFA device

Choose the type of MFA device to assign:

☒ Virtual MFA device
Authenticator app installed on your mobile device or computer

☐ U2F security key
YubiKey or any other compliant U2F device

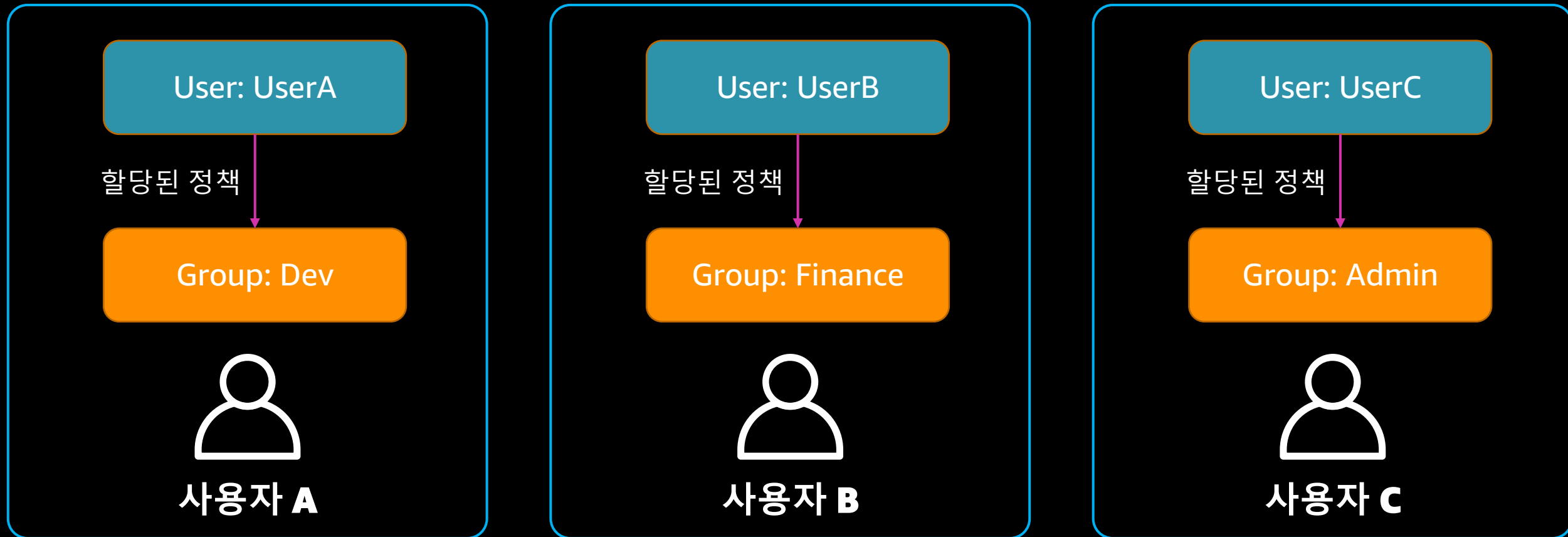
☐ Other hardware MFA device
Gemalto token

For more information about supported MFA devices, see [AWS Multi-Factor Authentication](#)

Cancel

Continue

IAM 사용자 관리



- 개별 IAM 사용자 만들기
- 공통된 권한 그룹을 사용하여 IAM 사용자에게 권한을 할당

최소 권한 부여

❖ Access Advisor

- 최대 1년간의 기간 동안 저장된 데이터를 기준으로 마지막 접속 서비스에 대한 정보를 제공

- Access Advisor 정보를 API 를 통하여 조회 가능

```
"ServicesLastAccessed": [  
  {  
    "LastAuthenticated": "2018-11-21T17:41:15Z",  
    "LastAuthenticatedEntity": "arn:aws:iam::123456789012:role",  
    "ServiceName": "Amazon EC2",  
    "ServiceNamespace": "ec2", "TotalAuthenticatedEntities": 1  
  },  
  ]
```

Permissions	Groups (1)	Tags (2)	Security credentials	Access Advisor
Access advisor shows the service permissions granted to this user and when those services were last accessed. You can use this information to revise your policies. Learn more				
Note: Recent activity usually appears within 4 hours. Data is stored for a maximum of 365 days, depending when your region began supporting this feature. Learn more				
Filter: No filter <input type="text" value="Search"/> Showing 171 results				
Service Name	Policies Granting Permissions	Last Accessed		
Amazon Macie	AdministratorAccess	71 days ago		
AWS IoT	AdministratorAccess	71 days ago		
Amazon API Gateway	AdministratorAccess	71 days ago		
Amazon Elastic File System	AdministratorAccess	80 days ago		
DataSync	AdministratorAccess	80 days ago		
AWS Backup	AdministratorAccess	80 days ago		
AWS Performance Insights	AdministratorAccess	84 days ago		
AWS Transfer for SFTP	AdministratorAccess	85 days ago		
AWS Batch	AdministratorAccess	85 days ago		
AWS Firewall Manager	AdministratorAccess	99 days ago		
Alexa for Business	AdministratorAccess	Not accessed in the tracking period		
AWS Accounts	AdministratorAccess	Not accessed in the tracking period		
AWS Amplify	AdministratorAccess	Not accessed in the tracking period		

서비스 간 권한 제어에 **Role** 사용

EC2에 IAM역할을 부여할 수 있는 정책

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "iam:PassRole",
      "iam:ListInstanceProfiles",
      "ec2:*"
    ],
    "Resource": "*"
  }]
}
```

부여자 보다 많은 권한을 갖는
IAM역할을 EC2에 줄 수 있는 가능성

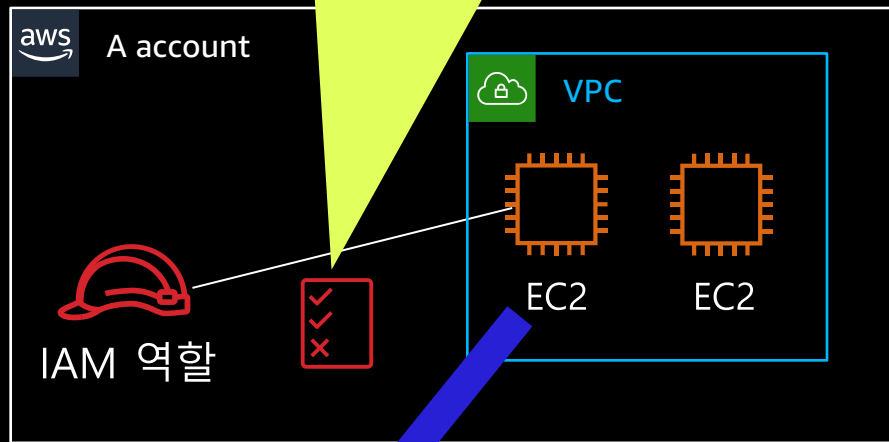
EC2에 특정 IAM역할만 부여할 수 있도록 제한

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::<account-id>:role/Get-pics"
    }
  ]
}
```

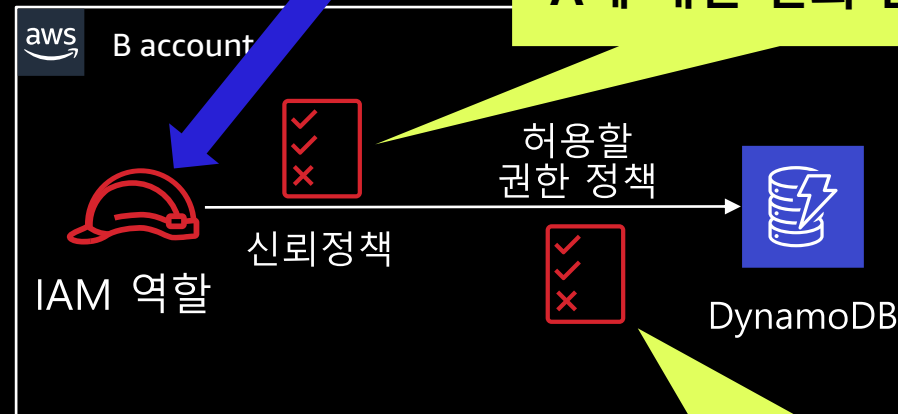
부여자에게 EC2에 줄 수 있는 역할을 한정시켜줌.

Role을 사용하여 권한 위임

다른 어카운트의 IAM 역할을 Assume할 수 있는 in-line 정책을 IAM 역할에 설정



A에 대한 신뢰 관계 설정



위임할 권한 정책 정의

A account의 IAM role 권한

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource":
    "arn:aws:iam::Account_B_ID:role/Account_B_IAM_Role_Name"
  }
}
```

B account의 IAM role 신뢰정책(Trust Policy)

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::123456789012:root"
    },
    "Action": "sts:AssumeRole",
    "Condition": {}
  }
]
```

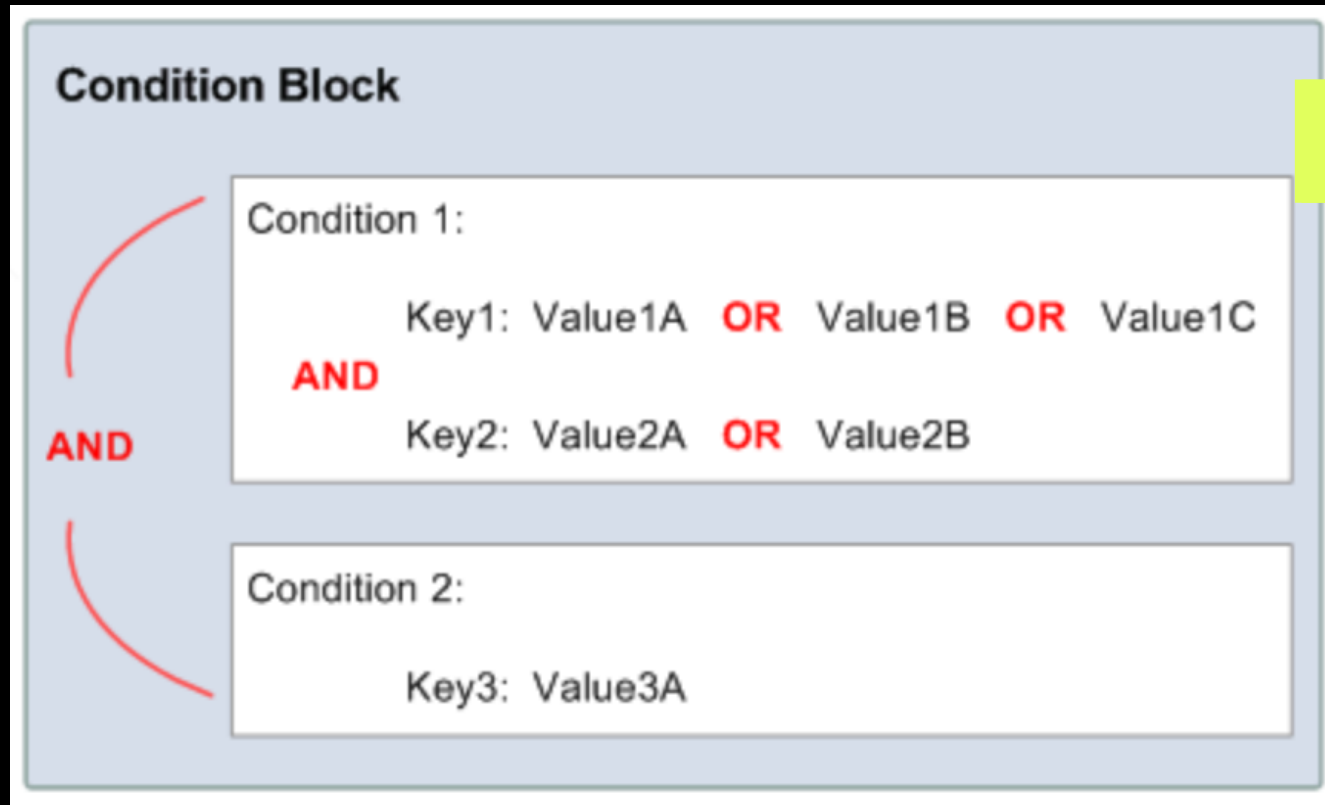
자격 증명을 정기적으로 교체

<input type="checkbox"/>	User name ▾	Groups	Access key age	Password age	Last activity	MFA
<input checked="" type="checkbox"/>	[REDACTED] VPCE_User	Administrators	! 416 days	735 days	410 days	Not enabled
<input type="checkbox"/>	[REDACTED] User1	Administrators	! 416 days	973 days	735 days	Not enabled
<input type="checkbox"/>	[REDACTED] g	Administrators	! 416 days	416 days	315 days	Not enabled
<input type="checkbox"/>	[REDACTED]	Administrators	! 416 days	559 days	Today	Not enabled

콘솔이나 Credential Report를 통해 교체 주기를 지난 IAM사용자의 자격증명들을 파악해서 조치

user	arn	user_creation	password_enabled	password_last	password_last_changed	password_next_rotation	mfa_active	access_key	access_key_1_last_rotated	access_key_1_
<root_account>	arn:aws:iam::8	2015-05-08T	not_supported	2017-04-12T	not_supported	not_supported	FALSE	TRUE	2015-06-29T09:31:32+00:00	N/A
DDB_VPCE_U	arn:aws:iam::8	2016-12-26T	TRUE	2017-11-16T	2016-12-26T07:57:59+00:00	N/A	FALSE	TRUE	2017-11-10T01:36:58+00:00	N/A
DemoUser1	arn:aws:iam::8	2016-05-02T	TRUE	2016-12-26T	2016-05-02T04:43:46+00:00	N/A	FALSE	TRUE	2017-11-10T01:32:47+00:00	N/A

보안 강화를 위해 정책 조건 사용



AND

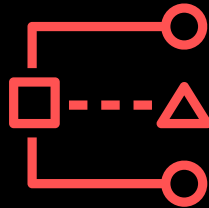
```
"Condition" : {  
  "DateGreaterThan" : {  
    "aws:CurrentTime" : "2013-08-16T12:00:00Z"  
  },  
  "DateLessThan": {  
    "aws:CurrentTime" : "2013-08-16T15:00:00Z"  
  },  
  "IpAddress" : {  
    "aws:SourceIp" : ["192.0.2.0/24", "203.0.113.0/24"]  
  }  
}
```

OR

AWS 계정의 활동 모니터링 및 감사



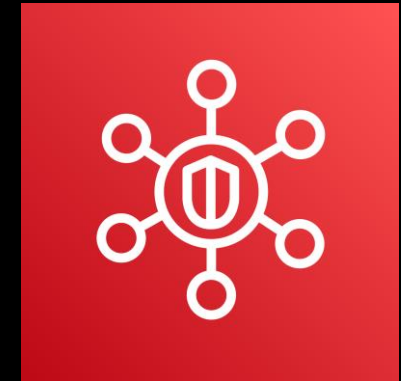
AWS CloudTrail



AWS IAM
Access analyzer



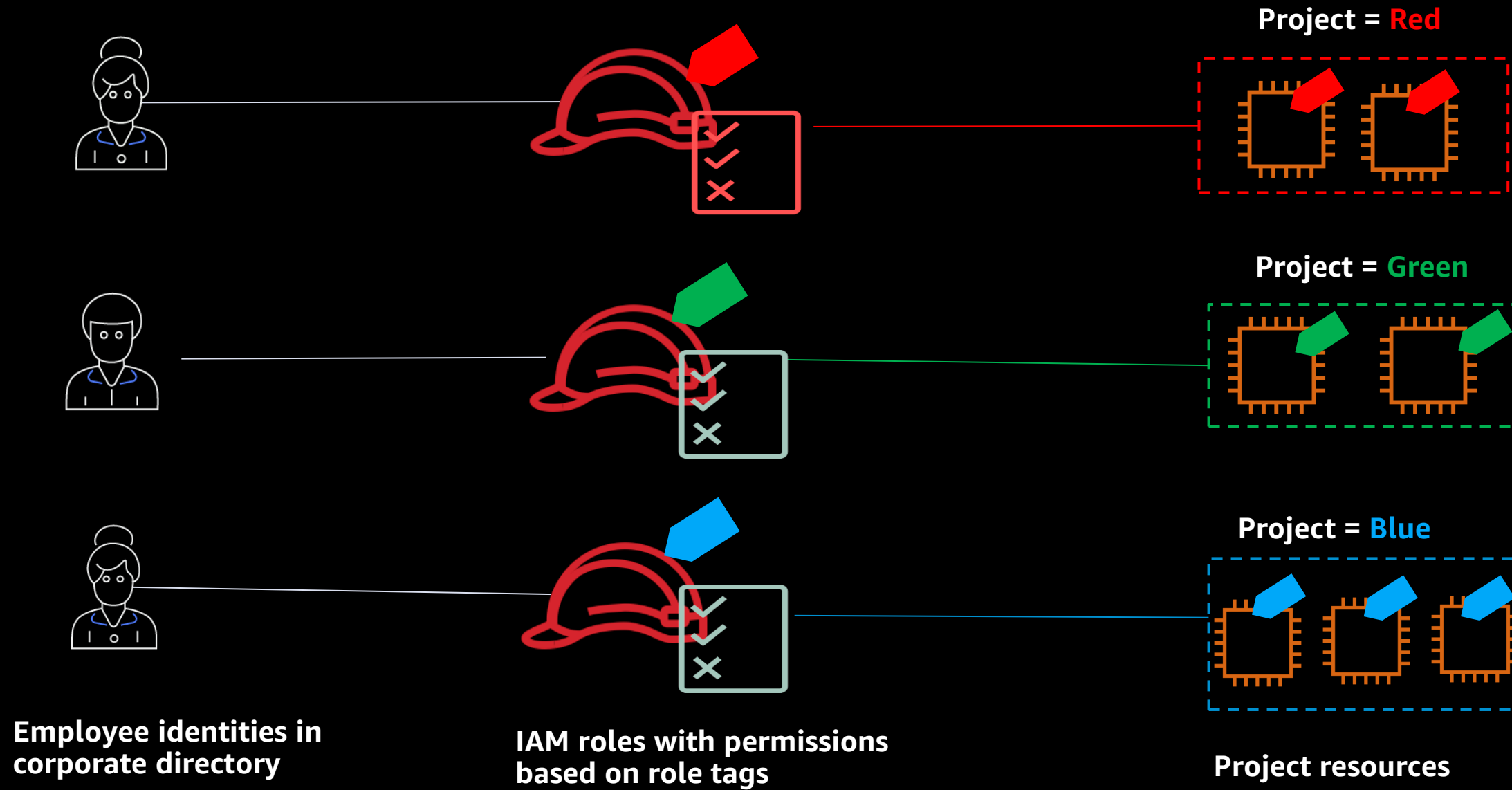
Amazon GuardDuty



AWS Security Hub

AWS IAM Advanced

속성 기반 접근 제어 : Before



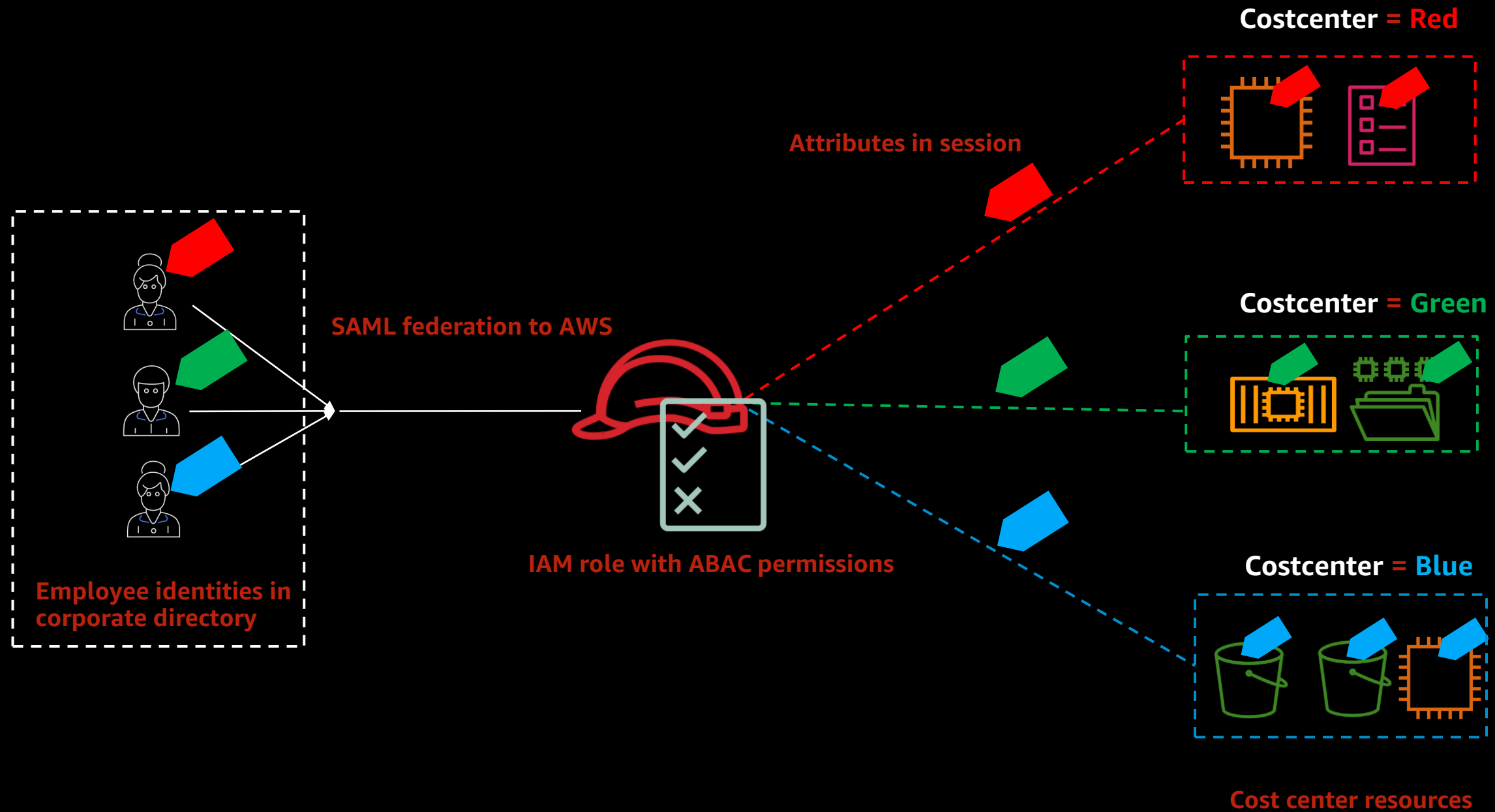
속성 기반 접근 제어

- **ABAC : Attribute Based Access Control**

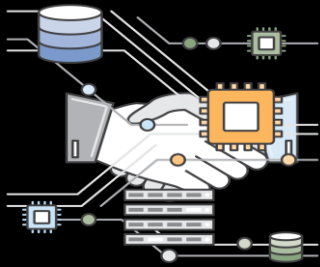
IAM 보안 주체의 요소(attribute, 태그)를 이용해서 단일 policy로 각기 다른 리소스에 재사용가능한 접근 제어

Condition key	Description	Actions that support the condition key
<code>aws:RequestTag</code>	Tags that you request to be added or removed.	<code>iam:CreateUser</code> , <code>iam:Create Role</code> , <code>iam:TagRole</code> , <code>iam:UntagRole</code> , <code>iam:TagUser</code> , <code>iam:UntagUser</code>
<code>aws:TagKeys</code>	Tag keys that are checked before the actions are executed.	<code>iam:CreateUser</code> , <code>iam:Create Role</code> , <code>iam:TagRole</code> , <code>iam:UntagRole</code> , <code>iam:TagUser</code> , <code>iam:UntagUser</code>
<code>aws:PrincipalTag</code>	Tags that exist on the user or role making the call.	A global condition (all actions across all services support this condition key)
<code>iam:ResourceTag</code>	Tags that exist on an IAM resource.	All IAM APIs that supports an IAM user or role and <code>sts:AssumeRole</code>

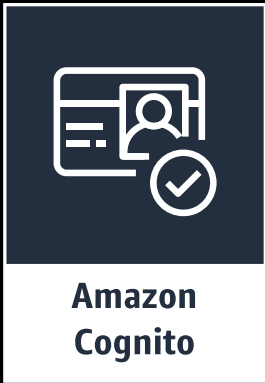
속성 기반 접근 제어 : **After**



클라우드 확장

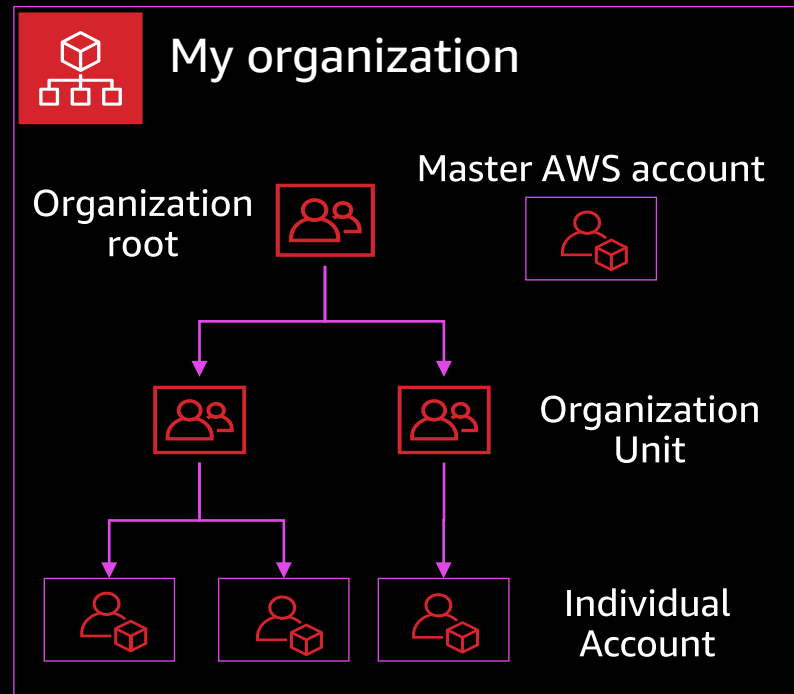


Custom Identity Broker

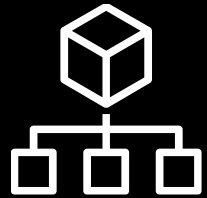


	IAM Federation	AWS SSO	IAM Custom Identity Broker	Cognito User Pool	Cognito Identity Pool
저장소(Directory)	고객 / 소셜	고객 / AWS SSO	고객	Cognito User Pool	고객 / 소셜 / CUP
자격증명 공급자(IdP)	고객 / 소셜	AWS SSO	고객	Cognito User Pool	고객 / 소셜 / CUP
AWS 리소스 접근시, 자격증명 교환	IAM	AWS SSO	고객	Cognito Identity Pool	Cognito Identity Pool
주요 용도	AWS 관리콘솔 연계	AWS 관리콘솔 연계	AWS 관리콘솔 연계	비즈니스 User 인증 / 인가	비즈니스 User 인가
지원 방식	SAML / OIDC	SAML	Custom	SAML / OIDC	SAML / OIDC / Custom

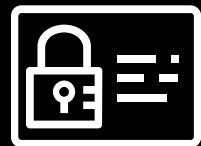
AWS Organizations



멀티 AWS 계정 환경을 위한
AWS 계정의 중앙 거버넌스 및 관리
서비스



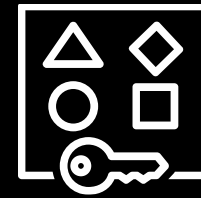
조직 및 계정 관리
정의



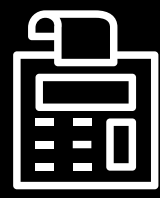
액세스 및 권한
제어



규정 준수를 위한
환경 감사,
모니터링 및 보안

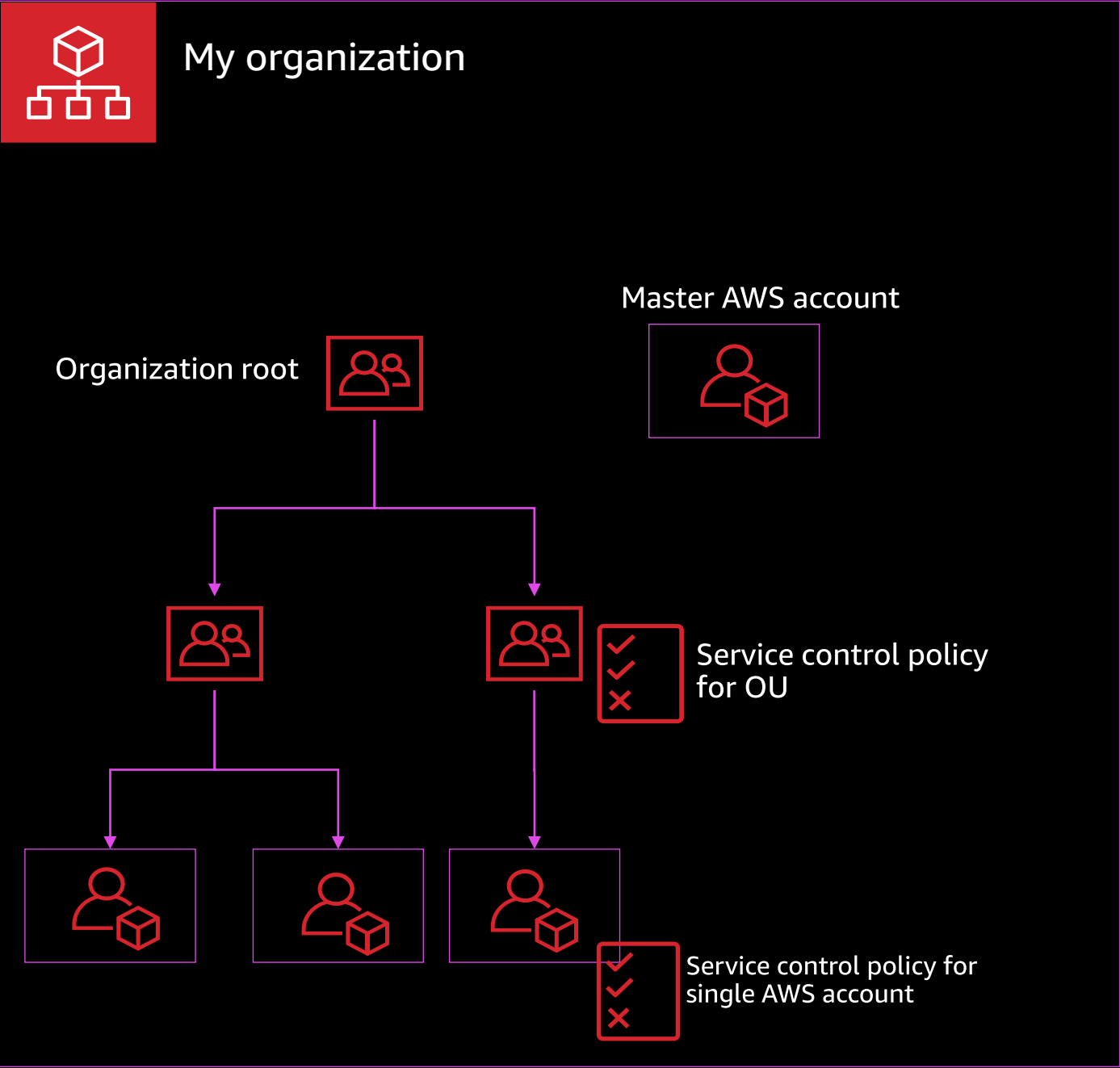



여러 계정에서
리소스 공유

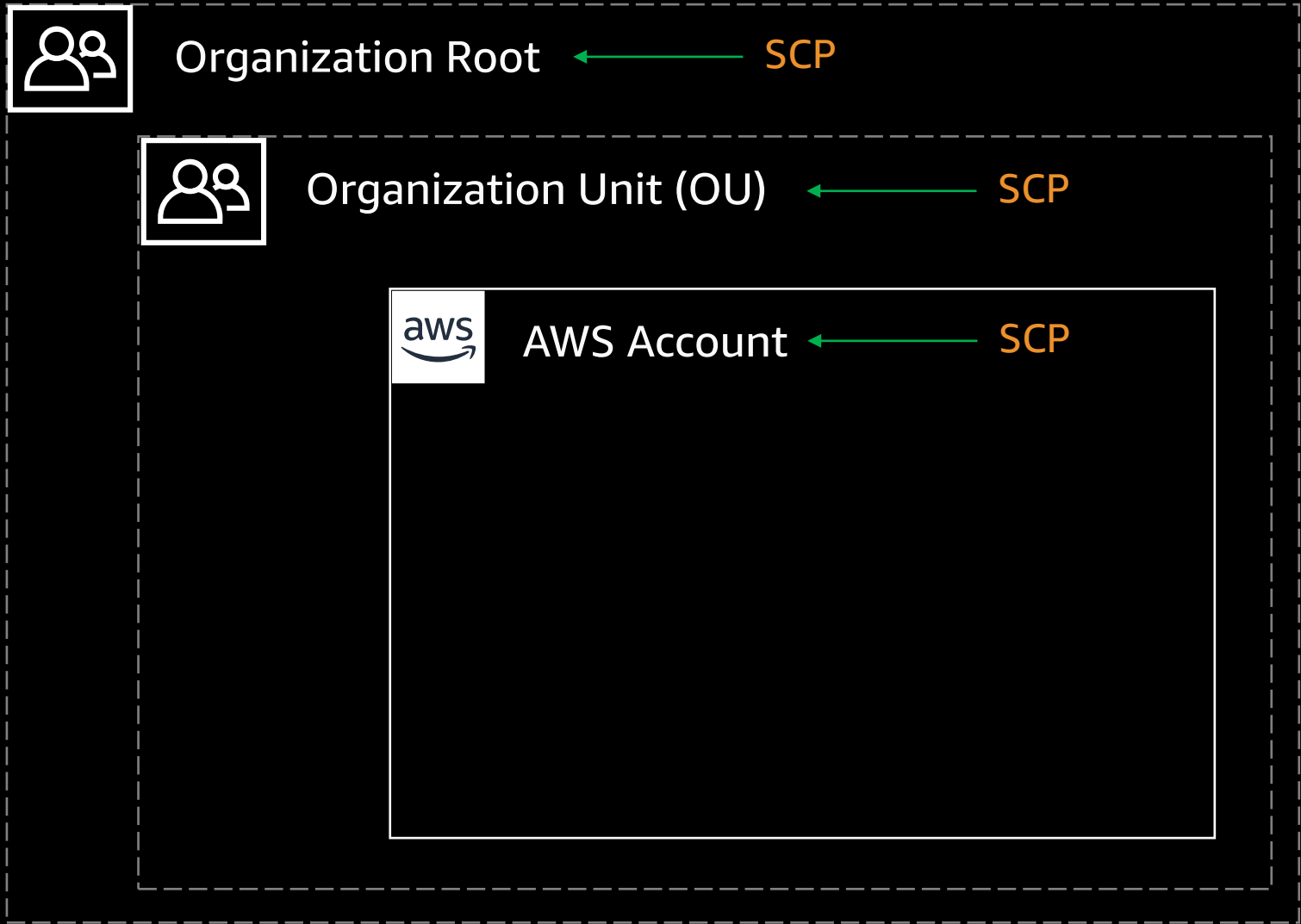


비용 및 청구를
중앙에서 관리

Service Control Policies Overview



 Service control policy as an organization level guardrail



Service Control Policies Overview

Whitelist - only allow specific actions¹

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:*", "ds:*", "s3:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Blacklist – block specific actions²

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "organizations:LeaveOrganization",
        "ec2:TerminateInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS Organizations – Organizational unit & Service Control Policies



AWS Cloud



AWS Organizations



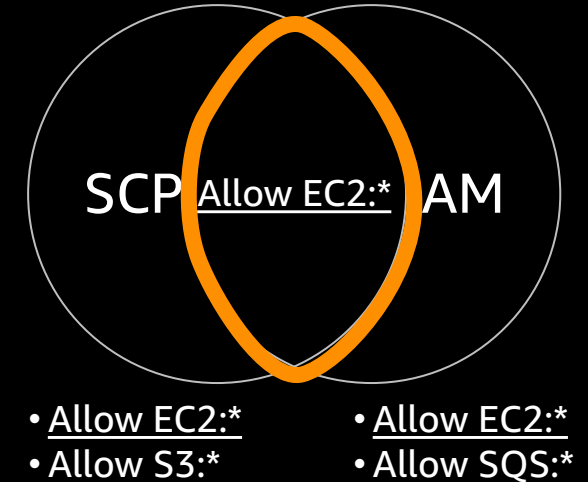
Organizational unit (OU)

- AWS 계정 그룹
- 그룹에 대한 서비스 제어 정책 (SCP)
- 권한 그룹핑 사용



Service Control Policies (SCP)

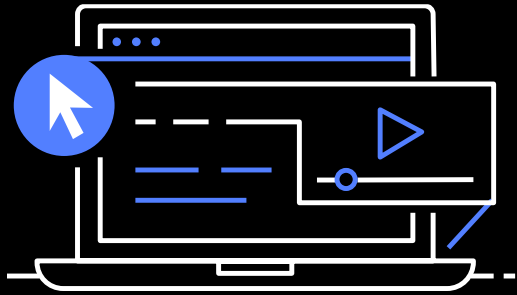
- AWS 서비스 API를 제어
 - API 허용 목록 정의 – allowlist
 - API 차단 목록 정의 – denylist
- SCP 기본 특징
 - 루트를 포함하여 하위 모든 계정에 보이지 않음
 - 루트를 포함하여 하위 모든 계정에 적용



Key take away

- IAM의 인증과 인가
- 정책의 구조와 종류
- IAM 운영은 모범 사례에 맞춰서
- Beyond IAM, 워크로드 보안, Organizations

AWS 디지털 교육



Flexibility to learn
your way

550개 이상의
무료 디지털 교육 및
심층적 강의실 교육을 통해
클라우드 기술 역량을
업그레이드 하세요!

추천 과정 (한글 자막)

- [AWS Cloud Practitioner Essentials](#)
AWS 클라우드 기초에 대해 학습하고, 기초 자격증인 AWS Certified Cloud Practitioner 시험을 준비할 수 있습니다.
- [Amazon DynamoDB for Serverless Architectures](#)
Amazon DynamoDB의 전반적인 소개와 Amazon DynamoDB가 서버리스 아키텍처 구축에 어떻게 활용되는지 알아봅니다.
- [AWS Security Fundamentals](#)
AWS 액세스 제어 및 관리, 거버넌스, 로깅, 그리고 암호화 방법을 포함한 기본적인 클라우드 컴퓨팅 및 AWS 보안 개념에 대해 알아봅니다.
- [AWS Database Offerings](#)
다양한 데이터베이스 기술 및 아키텍처에 대한 기본 개요와 AWS 데이터베이스 서비스를 소개합니다.

AWS Builders Online Series에 참석해주셔서 대단히 감사합니다.

저희가 준비한 내용, 어떻게 보셨나요?
더 나은 세미나를 위하여 **설문을 꼭 작성해 주시기 바랍니다.**



aws-korea-marketing@amazon.com



twitter.com/AWSKorea



facebook.com/amazonwebservices.ko



youtube.com/user/AWSKorea



linkedin.com/company/amazon-web-services



twitch.tv/aws

Thank you!

김태수

솔루션즈 아키텍트

AWS

