Identity and Access Management in AWS, Azure, GCP

클라우드 환경에서 보안 문제는 중요한 이슈입니다. 자원 접근에 대한 권한 관리와 자격 증명 관리 실패로 인한 보안 문제가 많이 발생하고 있습니다. 특히 멀티 클라우드 환경에서는 복잡성 때문에 위험이 더 커집니다. 클라우드 환경에서 자원에 대한 접근 제어와 권한 관리가 어떻게 이루어져 있는지 알아보고, AWS, AZURE, GCP의 IAM 정책을 비교해 보았습니다.

1. RBAC과 ABAC

RBAC(Role-Based Access Control)	ABAC(Attribute-Based Access Control)
RBAC 은 현재 대부분의 퍼블릭 클라우드에서 사용되고 있는 권한 부여 방식입니다. Who(주체), what(권한), where(자원)의 세 가지 요소에 대하여, 주체에 자원에 접근할 수 있는 권한을 포함하는 역할을 부여함으로써 접근 제어를 합니다.	ABAC 은 RBAC 이 진화한 개념으로써, 추가적인 속성 조건을 기반으로 역할을 부여하는 방식입니다. 더 세밀하고 간단한 권한 관리가 가능합니다. 사용될 수 있는 대표적인 속성으로는 시간이나 장소같은 컨텍스트 속성이 있고, 그 외 주체, 액션, 자원의 속성이 있습니다.

2. AWS, Azure, GCP의 IAM 시스템 비교

	AWS	Azure	GCP
IAM service	AWS IAM	Azure RBAC, Azure Active Directory	GCP IAM
account management hierarchy	-root account -organizations -accounts	-management group -subscriptions -resource groups -resources	-organization -folders -projects -resources
elements	-principal -role -resource -policy	-security principal -role -scope	-identity/principal -role -resource
Security principals	-root user -IAM user -user group -federated user	-user -group -service principal -managed identity	-google account -service account -google group -G suite domain -cloud identity domain

policy	An object that defines permissions	-	A collection of role bindings
types of roles	-role -AWS service role -AWS service-linked role	-classic subscription administrator roles, Azure roles, Azure AD roles -built-in roles, custom roles	-primitive roles -predefined roles -custom roles
service to resource access	service role	service principal	service account
fundamental built-in roles	-	-owner, contributor, reader (Azure roles)	-owner -editor -viewer
security credentials	-id, password -access key, secret key -key pair (EC2)	-id, password -certificates	-id, password -service account keys -API key
Authentication methods	-CLI: aws configure -application: Amazon Cognito	-CLI: az login --service-principal -u app-id -p password-or-cert --tenant tenant	-CLI: gcloud auth application-default login -application: Application Default Credentials(ADC), OAuth 2.0, API key

access tools	-AWS Management Console -AWS CLI -AWS SDK -IAM HTTPS API	-Azure portal -Azure CLI -Azure PowerShell -Azure SDKs -REST APIs	-Google Cloud Console -gcloud CLI tools -REST API -Resource Manager client library
IAM security best practices	• AWS 계정 루트 사용자 액세스 키 잠금 • 역할을 사용하여 권한 위임 • 최소한의 권한 부여 • 사용자에 대한 강력한 암호 정책 구성 • MFA 활성화 • Amazon EC2 인스턴스에서 실행되는 애플리케이션에 역할 사용 • 자격 증명을 정기적으로 교체 • 불필요한 자격 증명 삭제 • 보안 강화를 위해 정책 조건 사용 • AWS 계정의 활동 모니터링	• 사용자가 필요한 권한만을 부여 • 역할을 사용자가 아닌 그룹에 부여 • 기본 보안 경계로 ID 처리 • ID 관리 중앙 집중화 • 연결된 테넌트 관리 • Single Sign-On 사용 • 조건부 액세스 설정 • 정기 보안 강화 계획 • 암호 관리 사용 • 사용자에 대한 MFA 적용 • 역할 기반 액세스 제어 사용 • 권한 있는 계정의 낮은 노출 • 리소스를 만든 위치 제어 • 의심스러운 활동을 적극적으로 모니터링 • 스토리지 인증에 Azure AD 사용	• 액세스 제어에 리소스 계층 구조 사용 • 개별 사용자가 아닌 Google 그룹에 역할을 부여 • 서비스 계정에 최소한의 권한 부여 • 서비스 계정 키를 공유하면 안 됨 • 임시 위치에 서비스 계정 키를 남겨두면 안 됨 • 사용자 계정에 MFA 적용 • Cloud 모니터링 로그의 로그를 사용하여 IAM 정책의 변경사항을 정기적으로 감사

• Security principals

AWS	**-root user:** 처음 계정을 생성할 때만 사용 권장 **-IAM user:** 조직 내부의 사용자에 대응하는 계정 **-user group:** 사용자의 집합 **-federated user:** 조직이 이미 사용자 인증 시스템을 사용하고 있는 경우 따로 사용자를 생성하지 않고 IdP 로 자격 증명을 연동하는 사용자
Azure	**-user:** 사용자 **-group:** 사용자들의 집합 **-service principal:** 자원에 접근이 필요한 애플리케이션이나 서비스 **-managed identity:** Azure AD 인증을 지원하는 service principals 에 부여하여 인증이 가능하게 하는 특수한 ID
GCP	**-google account:** google email 주소를 사용하는 사용자 **-service account:** 애플리케이션이나 컴퓨트 자원 **-google group:** google account 와 service account 의 집합 **-G suite domain:** G Suite 하에 있는 google account **-cloud identity domain:** G suite 하에 있지 않은 모든 google account

• Types of roles

AWS	**-role:** 권한의 집합으로 해당 역할이 필요한 사용자 누구에게나 부여 가능 **-AWS service role:** 서비스가 리소스에 액세스하는 데 필요한 권한의 집합 **-AWS service-linked role:** AWS 서비스에 사전 정의되어 직접 연결된 service role
Azure	**-classic subscription administrator roles:** Subscriptions 에 대한 접근에 관한 역할. Account Administrator, Service Administrator, and Co-Administrator 를 포함.

	-**Azure roles:** Azure 자원에 대한 접근을 관리하는 역할 -**Azure AD roles:** Azure Active Directory 자원에 대한 접근을 관리하는 역할
GCP	-**primitive roles:** Owner, editor, viewer -**predefined roles:** 더 세밀하고 세분화 된 역할 -**custom roles:** 조직이 필요한 특정 권한들로만 생성 가능한 역할

• Security credentials

AWS	access key, secret key 생성	aws iam create-access-key --user-name
Azure	certificates 생성	az ad sp create-for-rbac --create-cert
GCP	service account keys 생성	gcloud iam service-accounts keys create key-file \ --iam-account=sa-name@project-id.iam.gservice account.com