AWS Identity and Access Management (IAM)

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. It enables you to:

- **Manage identities** (users, groups, and roles)
- **Control permissions** through policies
- **Provide temporary access** for federated users
- **Enforce least privilege** access

IAM is a global service that's not region-bound, though some IAM resources are region-specific.

Why IAM ?

Security Foundation

- **Prevent unauthorized access**: IAM ensures only authorized entities can access your AWS resources
- **Implement least privilege**: Grant only the permissions required to perform tasks
- **Audit access**: Track who did what and when with CloudTrail integration

Compliance Requirements

- **Regulatory compliance**: Meet requirements like HIPAA, PCI DSS, GDPR
- **Governance**: Implement centralized access control across your organization
- **Accountability**: Establish clear ownership and responsibility

Operational Excellence

- **Centralized management**: Control access across all AWS services from one place
- **Granular permissions**: Define precise access controls at resource level
- **Automation**: Integrate with DevOps workflows and infrastructure as code

Core IAM Concepts

Identities

Identity Type	Description	Use Case
AWS Account Root User	Complete access to all AWS resources	Initial account setup, emergency access
IAM Users	Entities created in AWS representing people or applications	Individual access for team members
IAM Groups	Collections of IAM users	Apply permissions to multiple users
IAM Roles	Entities with permissions that can be assumed	Temporary access for services, federated users
Temporary Credentials	Short-lived credentials	Enhanced security for applications

Policies

Policy Type	Description	Example
Managed Policies	AWS-managed or customer-managed reusable policies	AmazonS3ReadOnlyAccess
Inline Policies	Embedded directly into an identity	Custom policy for a specific user
Permissions Boundaries	Maximum permissions an identity can have	Restrict even administrators
Service Control Policies (SCPs)	Applied at organization level	Restrict services across accounts

Credentials

Credential Type	Description	Security Key
Password	Used for console access	Medium (with MFA: High)
Access Keys	Long-term credentials for API/CLI	Medium
Multi-Factor Authentication (MFA)	Additional security layer	High
Temporary Security Credentials	Short-lived credentials	Very High

How AWS Implements IAM ?

Key Technologies Behind IAM

1. Distributed Policy Engine

- **Component**: IAM Policy Evaluation Service
- **Technology**: Distributed system with low-latency evaluation
- **How it works**:
- Policies are parsed into abstract syntax trees (ASTs)
- During request processing, the engine evaluates all applicable policies
- Uses efficient algorithms to determine allow/deny decisions
- Caches frequently used policy evaluation

2. Global Replicated Database

- **Component**: IAM Database
- **Technology**: Multi-region replicated database
- **How it works**:
- IAM data is replicated across multiple AWS regions
- Uses eventual consistency model for most operations
- Provides high availability and durability
- Implements conflict resolution for concurrent updates

3. Credential Management System

- **Component**: AWS STS (Security Token Service)
- **Technology**: Cryptographic token generation
- **How it works**:
- Generates temporary credentials with limited lifetime
- Uses cryptographic signing to prevent tampering
- Integrates with identity providers for federation
- Supports role assumption and delegation

4. Access Logging and Auditing

- **Component**: CloudTrail Integration
- **Technology**: Distributed logging system
- **How it works**:
- Captures all IAM-related API calls
- Encrypts logs and stores in S3
- Provides searchable event history
- Enables security analysis and compliance reporting

Security Model

1. Policy Evaluation Logic

IAM uses a specific algorithm to evaluate policies:

1. **Default deny**: By default, all requests are denied
2. **Explicit allow**: An explicit allow overrides the default
3. **Explicit deny**: An explicit deny always overrides an allow
4. **Permissions boundaries**: Cap the maximum permission

2. Policy Types Evaluation Order

1. **Organizations SCPs** (if applicable)
2. **Resource-based policies**
3. **Identity-based policies**
4. **Session policies** (for temporary credentials)

3. Context-Based Evaluation

IAM evaluates policies in context of:

- **Principal**: Who is making the request
- **Action**: What operation they're trying to perform
- **Resource**: What resource they're trying to access
- **Conditions**: Additional constraints (time, IP, MFA status)

References

- AWS IAM Documentation
- IAM Policy Reference
- IAM Best Practices
- AWS Security Blog