Service Name

AWS Identity and Access Management (IAM)

Service Overview

AWS Identity and Access Management (IAM) is a web service that helps securely control access to AWS resources by managing permissions for users and entities. IAM controls who is authenticated (signed in) and authorized (has permissions) to use AWS resources. When an AWS account is created, it includes a root user with complete access to all services, but IAM enables creation of additional identities with specific permissions. IAM provides the infrastructure necessary to control authentication and authorization across AWS accounts, offering centralized identity management, fine-grained access control, and security features like multi-factor authentication and identity federation for secure resource access.

Key Use Cases

- Enterprise user management: Connect corporate Active Directory to IAM Identity Center for centralized access control across multiple AWS accounts and applications
- Application security: Grant applications running on EC2 instances secure access to AWS services using IAM roles with temporary credentials
- Cross-account access: Enable secure access between different AWS accounts for partners, vendors, or organizational divisions using cross-account roles
- Compliance and governance: Implement least-privilege access controls, audit trails, and permissions boundaries to meet regulatory requirements
- Multi-factor authentication: Add extra security layer requiring users to provide codes from MFA devices in addition to passwords

Features

- **IAM Users and Groups**: Create individual users and organize them into groups with specific permissions for single persons or applications
- **IAM Roles**: Provide temporary credentials for federated access, cross-account access, and applications running on EC2
- **Policies and Permissions**: Define granular permissions using JSON policy language with identity-based and resource-based policies
- **Multi-Factor Authentication (MFA)**: Support for passkeys, security keys, virtual authenticator apps, and hardware TOTP tokens for enhanced security
- **Identity Federation**: Enable users with existing passwords from corporate networks or identity providers to access AWS using temporary credentials

- **IAM Access Analyzer**: Generate least-privilege policies, verify public access, and validate IAM policies for secure permissions
- **Permissions Boundaries**: Set maximum permissions for identity-based policies to establish guardrails for delegated administration
- **Service-Linked Roles**: Predefined IAM roles created automatically when AWS services need permissions to access resources on your behalf

Value Proposition

IAM provides centralized identity and access management with no additional charges, integrating seamlessly with over 250 AWS services. Unlike third-party solutions, IAM offers native integration with AWS services, automatic scaling, and built-in compliance features including PCI DSS validation:

- IAM enables fine-grained access control through JSON policies, supports multiple authentication methods including MFA and federation, and provides eventual consistency with high availability through multi-server replication
- Organizations benefit from reduced complexity in managing access across multiple AWS accounts, enhanced security through least-privilege principles, and comprehensive audit capabilities through CloudTrail integration

Service Integration

IAM integrates with virtually all AWS services as the foundational security service for access management. Core integrations include AWS CloudTrail for detailed logging and auditing, AWS Organizations for multi-account management, AWS STS for temporary credential generation, and AWS Identity Center for centralized user access:

- IAM works with compute services like EC2 for instance roles, storage services like S3 for bucket policies, and management services like CloudFormation for infrastructure-as-code deployments
- The service also integrates with AWS SDKs and CLI tools for programmatic access management, ensuring consistent security controls across all AWS service interactions

Onboarding and Offboarding

Getting started with IAM requires an existing AWS account and begins with securing the root user by enabling multi-factor authentication. Create an administrative user for everyday tasks rather than using root credentials:

- Set up IAM users for individuals or applications, organize them into groups for easier permission management, and attach appropriate AWS managed policies
- Configure IAM roles for applications running on EC2 instances or for cross-account access scenarios

- Implement MFA for enhanced security and use IAM Access Analyzer to validate policies
- Follow the principle of least privilege by starting with minimal permissions and adding access as needed

Getting Started Guide: https://docs.aws.amazon.com/serverless/latest/devguide/starter-iam.html

Data Removal

IAM data removal involves systematically deleting IAM identities and associated resources. Remove IAM users by first deleting their access keys, detaching policies, removing from groups, and finally deleting the user account. Delete IAM roles by first detaching policies and removing trust relationships. Clean up unused policies, groups, and service-linked roles. Use IAM Access Analyzer to identify unused access before removal. IAM changes are eventually consistent, so allow time for propagation across AWS regions.

Backup/Restore and Disaster Recovery

IAM configuration itself does not require traditional backup as it is a managed service with built-in high availability and eventual consistency through multi-server replication across regions. However, organizations should document IAM policies, roles, and user configurations as part of infrastructure-as-code practices using CloudFormation or Terraform. IAM changes are automatically replicated across AWS regions, providing inherent disaster recovery capabilities for identity and access management configurations.

Backup Capabilities

IAM does not provide traditional backup capabilities as it is a managed service with built-in redundancy and replication. IAM configurations are automatically maintained across multiple servers for high availability:

- Organizations should implement infrastructure-as-code practices to version control and recreate IAM configurations
- IAM does not integrate with AWS Backup service as the identity management data is inherently protected through AWS's managed service architecture

Disaster Recovery

IAM inherently supports disaster recovery through its eventually consistent, globally replicated architecture across multiple servers and regions. IAM does not integrate with AWS Elastic Disaster Recovery as it is a foundational service with built-in resilience:

- During regional failures, IAM continues operating from other regions, though some changes may experience temporary delays during propagation
- Cross-region IAM role assumptions and federated access continue functioning during regional outages

Recovery Objectives

IAM supports recovery objectives through its highly available, eventually consistent architecture that automatically replicates data across multiple servers. While IAM changes may take time to propagate (eventual consistency), the service maintains continuous availability during regional failures. Organizations achieve low RTO by implementing cross-region IAM roles and federation strategies. RPO is effectively zero for IAM configurations due to automatic replication, though policy changes may experience brief propagation delays.

Service Constraints

Service Quotas: https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_iam-quotas.html

FAQ: https://aws.amazon.com/iam/faqs/

Technical Requirements

Service Documentation: https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html

User Guide: https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html

API Reference: https://docs.aws.amazon.com/IAM/latest/APIReference/welcome.html

Pricing Overview

Please see the AWS UK G Cloud 15 Pricing Document accompanying this service in the Digital Marketplace.

Public Pricing: https://aws.amazon.com/iam/pricing/

Cost Optimization

IAM itself incurs no additional charges as it is provided free with all AWS accounts. Cost optimization focuses on managing the resources that IAM-authenticated users and roles access rather than IAM service costs:

- Use IAM policies to implement resource-based restrictions that prevent unnecessary resource provisioning or access to expensive services
- Implement permissions boundaries to limit maximum spending capabilities of delegated administrators
- Use IAM roles instead of long-term access keys to reduce security overhead and potential unauthorized usage costs
- Regular access reviews through IAM Access Analyzer help identify and remove unused permissions that could lead to unintended resource usage

Savings Considerations

Primary cost savings come from preventing unauthorized or accidental resource usage through proper IAM controls rather than direct IAM service costs. Implement least-privilege policies to prevent users from provisioning expensive resources unnecessarily:

- Use IAM conditions in policies to restrict actions based on time, IP address, or other factors to limit when expensive operations can occur
- Regular access reviews and cleanup of unused IAM users, roles, and permissions reduce potential for unauthorized spending
- Cross-account roles eliminate need for duplicate resources across accounts
- Proper IAM governance prevents shadow IT spending and ensures resources are only accessible to authorized personnel for legitimate business purposes