AWS Identity and Access Management (IAM)

AWS Identity and Access Management (IAM) is a security service that helps you control who can access AWS resources and what actions they can perform. It ensures secure access to AWS services through **authentication** (who can access) and **authorization** (what actions they can perform).

---

1. Key Features of IAM

1. **Granular Access Control** – Define permissions at a fine-grained level.
2. **Secure Authentication** – Use passwords, access keys, and multi-factor authentication (MFA).
3. **Policy-Based Authorization** – Control user actions using JSON-based policies.
4. **Temporary Credentials** – Issue short-lived credentials using IAM roles.
5. **Integration with AWS Services** – IAM is integrated with almost all AWS services.
6. **No Additional Cost** – IAM is free to use (you only pay for AWS resources used).

---

2. IAM Components

A. Users

- An **IAM User** represents a person or application that needs access to AWS resources.
- Users have **credentials**:
    - **Console Access** → Username & Password
    - **Programmatic Access** → Access Key ID & Secret Access Key
- **Best Practice**: Never use the root user for daily operations.

Example: Creating a User (AWS CLI)

```
aws iam create-user --user-name DeveloperUser
```

B. Groups

- **IAM Groups** allow you to assign permissions to multiple users at once.
- Users inherit permissions from the group they belong to.
- Example groups: `Admins`, `Developers`, `Billing`, `Support`.

Example: Creating a Group (AWS CLI)

```
aws iam create-group --group-name Developers
```

Example: Adding a User to a Group

```
aws iam add-user-to-group --user-name DeveloperUser --group-name Developers
```

C. Roles

- **IAM Roles** allow AWS services, applications, or external users to assume a specific set of permissions.
- IAM Roles are identities with attached policies
- No permanent credentials; uses temporary security tokens.
- **Use Cases:**
  - Allow EC2 to access S3 without storing credentials.
  - Grant third-party applications limited AWS access.
  - Enable federated access via SSO.

Example: Creating a Role (AWS CLI)

```
aws iam create-role --role-name S3AccessRole --assume-role-policy-document
file://trust-policy.json
```

Example: Trust Policy (`trust-policy.json`)

D. Policies

- **Policies** define permissions using JSON documents.
- They specify **who** can perform **what actions** on **which resources**.
- AWS provides **Managed Policies**, but you can also create **Custom Policies**.

Example: S3 Read-Only Access Policy

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::my-bucket/*"
  }
}
```

Types of Policies:

1. **AWS Managed Policies** – Predefined by AWS (e.g., `AdministratorAccess`).
2. **Customer Managed Policies** – Created by users.
3. **Inline Policies** – Embedded directly in users, groups, or roles.
4. **Service Control Policies (SCPs)** – Applied at the AWS Organization level.

---

3. IAM Authentication Methods

A. Access Keys

- Used for programmatic access (CLI, SDKs, APIs).
- **Best Practice:** Rotate keys regularly, and never hard-code them.

Example: Creating Access Keys

```
aws iam create-access-key --user-name DeveloperUser
```

B. Multi-Factor Authentication (MFA)

- Adds an extra layer of security (password + OTP).
- Required for the root user and recommended for IAM users.

Enable MFA (CLI Example)

```
aws iam enable-mfa-device --user-name DeveloperUser --serial-number
arn:aws:iam::123456789012:mfa/DeveloperUser --authentication-code-1 123456 --
authentication-code-2 654321
```

---

4. IAM Security Best Practices

1. **Enable MFA for all users, especially the root account.**
2. **Follow the principle of least privilege (grant only necessary permissions).**
3. **Use IAM roles instead of hard-coded access keys in applications.**
4. **Rotate IAM access keys regularly.**
5. **Monitor IAM activity using AWS CloudTrail.**
6. **Use AWS Organizations & SCPs for centralized policy enforcement.**
7. **Use temporary credentials instead of long-lived access keys.**

---

5. IAM Policy Evaluation Logic

AWS IAM evaluates policies using the following order of precedence:

1. **Explicit Deny –** If a policy denies access, it is final.
2. **Explicit Allow –** If there's an allow policy and no deny, access is granted.
3. **Implicit Deny –** By default, AWS denies all actions unless explicitly allowed.

Example: Denying S3 Deletion Globally

```
{
  "Effect": "Deny",
  "Action": "s3:DeleteObject",
  "Resource": "*"
}
```

---

6. Advanced IAM Features

A. IAM Identity Center (AWS SSO)

- Centralized management for user authentication across AWS accounts.
- Allows integration with Microsoft AD, Okta, Google Workspace, etc.

B. IAM Access Analyzer

- Helps identify overly permissive policies.
- Detects unintended external access to resources.

Enable IAM Access Analyzer via CLI

```
aws accessanalyzer create-analyzer --analyzer-name MyAnalyzer --type ACCOUNT
```

C. AWS Organizations & SCPs

- **Organizations**: Manage multiple AWS accounts centrally.
- **Service Control Policies (SCPs)**: Restrict permissions at the account level.

Example: SCP Blocking S3 Public Access

```
{
  "Effect": "Deny",
  "Action": "s3:PutBucketPolicy",
  "Resource": "*",
  "Condition": { "Bool": { "aws:SecureTransport": "false" } }
}
```

---

7. IAM Monitoring & Logging

- **AWS CloudTrail**: Logs all IAM-related activities.
- **AWS Config**: Tracks IAM policy changes.
- **AWS GuardDuty**: Detects suspicious IAM activity.

---

8. IAM Pricing

- **IAM is Free –** You only pay for the AWS resources you use.

---

9. Common IAM Scenarios

Use Case	Solution
Allow EC2 to access S3	Attach an IAM role to EC2
Grant temporary access	Use AWS STS AssumeRole
Secure root account	Enable MFA and create separate admin users
Restrict a user to only one service	Attach a service-specific IAM policy

---

10. Summary

- IAM controls access to AWS resources using **users, groups, roles, and policies**.
- **Best practice**: Use IAM roles instead of access keys.
- **Always enable MFA and follow the least privilege principle**.
- **Monitor IAM activity using CloudTrail and Access Analyzer**.