In AWS Identity and Access Management (IAM), authentication and authorization are key concepts that control who can access AWS resources and what actions they can perform.

**1. Authentication vs. Authorization:**

* **Authentication:** This is the process of verifying the identity of a user or system before granting access to AWS. It answers the question, ***"Who are you****?"*  
  In AWS, authentication happens when you sign in using your credentials (password or access key). IAM provides different methods to authenticate users, such as:
  + **IAM Users** with usernames and passwords for the AWS Management Console.
  + **Access Keys** for programmatic access (e.g., using the AWS CLI or SDKs).
  + **Multi-Factor Authentication (MFA)** for added security.
* **Authorization:** This determines what actions an authenticated user or system can perform. It answers the question, *"****What are you allowed to do****?"*
  + After authentication, IAM policies are checked to determine which resources you can access and what actions you are allowed to perform.

**2. IAM Users:**

* **IAM Users** are entities created for individual people or applications to interact with AWS.
* Each IAM user has a unique name within the AWS account and can be assigned:
  + **Programmatic access** through access keys (used with AWS CLI or SDK).
  + **Console access** through a password.
* Users are granted permissions by attaching policies that define what actions they are allowed to perform.

**3. IAM Groups:**

* **IAM Groups** are collections of IAM users that share similar permissions.
  + Instead of attaching policies to individual users, you can attach policies to a group.
  + For example, you might create a group for administrators, developers, or auditors and assign policies to these groups to streamline permissions management.

**4. IAM Policies:**

* **IAM Policies** are JSON documents that define permissions and specify who can do what in AWS.
  + A policy consists of **statements** which include elements like:
    - **Effect:** Allow or Deny.
    - **Action:** Specifies the action(s) (e.g., s3:ListBucket).
    - **Resource:** Defines the specific resources the action applies to (e.g., a specific S3 bucket). Specifies **which AWS resources** the action applies to.

Uses **ARN (Amazon Resource Name)** to define specific resources

* + - **Condition (optional):** Adds restrictions or conditions for when the policy applies (e.g., allowing access only from a certain IP range).

There are two types of policies:

* **Managed Policies:**
  + **AWS Managed Policies:** Predefined by AWS for common use cases (e.g., AdministratorAccess, AmazonS3FullAccess).
  + **Customer Managed Policies:** Created by users for custom permissions.
* **Inline Policies:** Policies directly embedded within a user, group, or role, offering more specific control.

**5. IAM Roles:**

* **IAM Roles** are similar to users, but they don’t have long-term credentials like passwords or access keys.
  + Instead, roles are assumed by users, applications, or services temporarily, providing access with temporary security credentials.
  + Common use cases include:
    - **Cross-account access:** Granting permissions to users from another AWS account.
    - **EC2 instance roles:** Allowing EC2 instances to access AWS services without embedding credentials.