IAM - Identity & Access Management

By : LAKSHMIKANT DESHPANDE

Overview of IAM

IAM (Identity and Access Management) is a service that helps you securely control access to AWS services and resources. IAM allows you to create and manage users, groups, roles, and policies.

- **IAM Users**
- **IAM Groups**
- **IAM Roles**
- **IAM Policies**

IAM Users

An **IAM User** represents an individual person or application that interacts with AWS resources. Each user has its own set of credentials (username/password or access keys).

Type of access:

- **Programmatic access**: This will give the user access to the AWS CLI, SDK, or API.
- **AWS Management Console access**: This allows the user to sign in to the AWS Console.

IAM Groups

IAM Groups are collections of IAM users. By adding users to groups, you can grant permissions to multiple users at once. For example, you might have an **Admin group** or a **Developer group**.

Attach Policies to the Group:

- You can attach existing policies like **AdministratorAccess**, **PowerUserAccess**, etc.
- Select **AdministratorAccess** if you want this group to have full access to AWS resources.

IAM Roles

An **IAM Role** is a set of permissions that you can assign to AWS services or other AWS accounts. A role is meant to be assumed by a service, user, or entity.

IAM Policies

IAM Policies define the specific permissions associated with users, groups, or roles. Policies are written in JSON format and specify what actions are allowed or denied for specific AWS resources.

In the **Visual editor**, select a service (e.g., **S3**) and then define the specific permissions.

- For example, you can give the user permission to list S3 buckets (`s3:ListBucket`) and put objects in a specific bucket (`s3:PutObject`).

Summary of IAM Components

IAM Users: Represent individuals or applications that need to access AWS resources.

- Created users (e.g., JohnDoe) and assigned credentials for console or programmatic access.

IAM Groups: A way to assign permissions to multiple users at once.

- Created a group (Admins) and attached a policy (AdministratorAccess).

IAM Roles: Used to allow services or entities to assume a set of permissions.

- Created a role (e.g., EC2FullAccessRole) for EC2 or other services to use.

IAM Policies: Define permissions for users, groups, and roles.

- Created a custom policy (S3ReadWritePolicy) to grant access to specific resources.

Key IAM Concepts to Emphasize

Users, Groups, Roles, Policies:

- IAM **Users** represent people or applications that need AWS access.
- **Groups** are collections of users, and permissions can be assigned to groups.
- **Roles** are sets of permissions that can be assumed by AWS services or external AWS accounts.
- **Policies** define what actions are allowed or denied on which resources.

Access Analyzer

Access Analyzer helps you identify resources in your AWS environment that are shared with an external entity. It's useful for detecting unintended access to your resources.

- **Analyze Resource Policies**: Access Analyzer analyzes IAM, S3, and other resource policies to determine if the resource is publicly accessible or shared with external accounts.
- **Generate Findings**: It generates findings about who can access a resource (external accounts or public users).

IAM Best Practices

- **Use groups to assign permissions**: Assign IAM users to groups (e.g., Admins, Developers) and manage permissions at the group level instead of individual users.
- **Apply the principle of least privilege**: Grant only the minimum permissions necessary for a user, group, or role to perform their job.
- **Enable MFA**: Always use Multi-Factor Authentication (MFA) for critical accounts (like the root account and IAM users with broad access).
- **Use roles for EC2 instances**: Assign roles to EC2 instances instead of embedding AWS credentials in application code.
- **Rotate credentials regularly**: Regularly rotate IAM access keys, passwords, and secrets to reduce the risk of compromised credentials.
- **Avoid using the root account**: The root account should be used minimally. Instead, create IAM users and assign appropriate permissions.