Identity Access management - IAM

Ponnam Phani Krishna
PONNAM.PHANI@GMAIL.COM

Identity Access Management (IAM)

IAM Allows you to manage users and their level of access to the AWS Console. It is important to understand IAM and how it works, both for the exam and for administrating a company's AWS Account.

IAM Allows you to setup user account, policies, roles, groups etc.

Features of IAM:

- Centralised control of your AWS account
- Shared access to your AWS account
- Granular Permissions
- Identity Federation (Including Active Directory, Facebook, etc.)
- Multifactor Authentication
- Provide temporary access for users/devices and services
- Allows you to setup your own password rotation policies
- Integrated with many different AWS services
- Supports PCI DSS Compliance.

Shared access to your account:

You can grant other people permission to administer and use resources in your AWS account without having to share your password or access key

Granular Permissions:

You can grant different permissions to different people for different resources. For example, you might allow some users complete access to Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Redshift, and other AWS services. For other users, you can allow read-only access to just some S3 buckets, or permission to administer just some EC2 instances, or to access your billing information but nothing else.

Multi-factor authentication (MFA)

You can add two-factor authentication to your account and to individual users for extra security. With MFA you or your users must provide not only a password or access key to work with your account, but also a code from a specially configured device.

Identity federation

You can allow users who already have passwords elsewhere—for example, in your corporate network or with an internet identity provider—to get temporary access to your AWS account.

PCI DSS Compliance

IAM supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS).

Key Terminology:

1. Users
2. Groups
3. Policies
4. Roles

Users: End Users such as people, employees of an organization etc. two different types of access available for IAM users, Web console access & Programmatic access.

- **Web console access** : This is the access type which allows the user to access all amazon services through a web browser.
- **Programmatic access:** This is the access type which allows the user to access using cli, API or SDK.

Web Console Access type users will get ***username*** and ***password*** to login to amazon account

Programmatic Access type users will get ***access key ID*** & ***Secret access key*** to login to amazon account using CLI

Groups: A collection of users. Each user in the group will inherit the permissions of the group.

Policies: Policies are made up of documents, called policy documents. These documents are in a format called JSON and they give permissions as to what a user/group/role is able to do.

Roles: You can create roles and then assign them to AWS Resources. Roles can be assigned to services, so that the service can manage other service.

IAM is **Universal**, It Does not apply to regions at this time.

The 'root account' is simply the account created when first setup your AWS account. It has complete admin access.

New Users have no permissions when first created.

New Users are assigned Access Key ID & Secret Access Keys when first created.

These are not the same as a password. You cannot use the access key ID & Secret Access Key to login to the console. You can use this to access AWS via the APIs and Command Line.

Pre-requisites: Access to your AWS Account

Identity Access Management – LAB

1. Secure AWS Account
    a. Activate MFA on your root account
    b. Create individual IAM Users
    c. Use Groups to assign Permissions
    d. Apply an IAM Password Policy
2. IAM Users
    a. User with Programmatic Access
    b. User with console access
3. Enable/ Disable Console access for a User
4. Regenerate secret access key if we lost

ToDo List 1:

1. Activate MFA on your root account
2. Create a password policy with the below options
    a. Min password length 10 char
    b. Uppercase letters
    c. Lower case letters
    d. Numbers
    e. Special characters
    f. Password history 3
    g. Password age 45 days

ToDo List 2:

(Web console access type users)

1. Create User1 with no permissions (default user)
2. Create User2 with Admin permissions (attach appropriate policy to the user)
3. Create user3 with Read only permission across your AWS Account
4. Create user4 with Full permissions on IAM Service
5. Create User5 with full permissions on EC2 service
6. Create a group called Admin Group and attach appropriate policy to the group so that users in that group will get Admin privileges across your AWS Account
7. Add user1 to the group Admin Group to make him as an administrator
8. Create user6 and add him to the group Admin Group
9. Create user7 and copy the permissions from user3
10. Reset the password of user4

Note: Above mentioned users should get Web console access type only

ToDo List 3:

(programmatic access type users)

1. Create user11 with full admin permissions
2. Install AWS CLI on your computer by downloading from https://aws.amazon.com/cli
3. Configure user11 on your computer by issuing the below command
    a. Aws configure
4. Check the connectivity by accessing aws s3 ls
5. Regenerate new accesskeyid and secretaccess key for user11
6. Re configure the cli with new accesskeyid and secret access key

ToDo List 4:

1. Change the access type of user1 from Web console access to programmatic access
2. Assign web console access to user11 by keeping the programmatic access.

ToDo List 5:

1. Force IAM user to enable MFA, so that he will not be allowed to access any of your amazon services without MFA Enabled.