AWS-IAM

Introduction:

AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

IAM is a feature of your AWS account offered at no additional charge. You will be charged only for use of other AWS services by your users.

Features:

1. [Manage IAM users](https://aws.amazon.com/iam/details/manage-users/)and [their access](https://aws.amazon.com/iam/details/managing-user-credentials/)

You can create users in IAM, assign them individual security credentials (in other words, access keys, passwords, and authentication devices), or request temporary security credentials to provide users access to AWS services and resources. You can manage permissions in order to control which operations a user can perform.

1. [Manage IAM roles](https://aws.amazon.com/iam/details/manage-roles/) and [their permissions](https://aws.amazon.com/iam/details/manage-permissions/)

You can create roles in IAM and manage permissions to control which operations can be performed by the entity, or AWS service, that assumes the role. You can also define which entity is allowed to assume the role. In addition, you can use [service-linked roles](https://docs.aws.amazon.com/console/iam/service-linked-role) to delegate permissions to AWS services that create and manage AWS resources on your behalf.

1. Shared access to your AWS 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. Users can share the resources for the collaborative projects.

1. Granular permissions

 It is used to set a permission that user can use a particular service but not other services. 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.

1. Secure access to AWS resources for applications that run on Amazon EC2

You can use IAM features to securely provide credentials for applications that run on EC2 instances. These credentials provide permissions for your application to access other AWS resources. Examples include S3 buckets and DynamoDB tables.

1. 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 special configured device.

1. Identity federation

An Identity Federation means that we can use Facebook, Active Directory, LinkedIn, etc. with IAM. Users can log in to the AWS Console with same username and password as we log in with the Active Directory, Facebook, etc.

1. Identity information for assurance

If you use [AWS Cloud Trail](https://aws.amazon.com/cloudtrail/), you receive log records that include information about those who made requests for resources in your account. That information is based on IAM identities.

1. PCI DSS Compliance

PCI DSS (Payment Card Industry Data Security Standard) is a compliance framework. If you are taking credit card information, then you need to pay for compliance with the framework.

10. Integrated with many AWS services

There are a lot of or a list of AWS services that work with IAM,

* Service
* Actions
* Authorization based on tags
* Resource-level permissions
* Resource-based policies
* Service-linked roles

11. Eventually Consistent

IAM, like many other AWS services, is [eventually consistent](https://wikipedia.org/wiki/Eventual_consistency). IAM achieves high availability by replicating data across multiple servers within Amazon's data centers around the world. If a request to change some data is successful, the change is committed and safely stored. However, the change must be replicated across IAM, which can take some time. Such changes include creating or updating users, groups, roles, or policies.

1. Free to use

 AWS IAM is a feature of AWS account which is offered at no additional charge. You will be charged only when you access other AWS services by using IAM user.

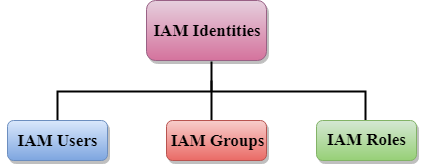
1. Provide temporary access for users/devices and services where necessary**:**

If you are using a mobile app and storing the data in AWS account, you can do this only when you are using temporary access.

IAM Identities:

IAM identities are created to provide authentication for people and processes in your aws account.

**IAM identities are categorized as given below:**



IAM Roles:

A role is a set of permissions that grant access to actions and resources in AWS. An IAM User can use a role in the same AWS account or a different account. Role is also an AWS identity with permission policies that determine what the identity can and cannot do in AWS.

Important terms associated with the "IAM Roles":

* **Delegation:**

Delegation is a process of granting the permissions to the user to allow the access to the AWS resources that you control. Delegation sets up the trust between a trusted account (an account that owns the resource) and a trusting account (an account that contains the users that need to access the resources).

To delegate permission to access the resources, an IAM role is to be created in the trusting account that has the two policies attached.

**a. Permission Policy:** It grants the user with a role the needed permissions to carry out the intended tasks.

b**. Trust Policy:** It specifies which trusted account members can use the role.

* **Federation:**

Federation is a process of creating the trust relationship between the external service provider and AWS. For example, Facebook allows the user to login to different websites by using their facebook accounts.

* **Trust policy:**

  A document was written in JSON format to define who is allowed to use the role. This document is written based on the rules of the IAM Policy Language.

* **Permissions policy:**

 A document written in JSON format to define the actions and resources that the role can use. This document is based on the rules of the IAM Policy Language.

* **Permissions boundary:**

It is an advanced feature of AWS in which you can limit the maximum permissions that the role can have. The permission boundaries can be applied to IAM User or IAM role but cannot be applied to the service-linked role.

* **Principal:**

A principal can be AWS root account user, an IAM User, or a role. The permissions that can be granted in one of the two ways:

* + Attach a permission policy to a role.
  + The services that support resource-based policies, you can identify the principal in the principal element of policy attached to the resource.

An IAM Role can be used in the following ways**:**

* **IAM User:**

IAM Roles are used to grant the permissions to your IAM Users to access AWS resources within your own or different account. An IAM User can use the permissions attached to the role using the IAM Console. A Role also prevents the accidental access to the sensitive AWS resources.

* **Applications and Services:**

 You can grant the access of permissions attached with a role to applications and services by calling the AssumeRole API function. The AssumeRole function returns a temporary security credentials associated with a role. An application and services can only take those actions which are permitted by the role. An application cannot exit the role in the way the IAM User in Console does, rather it stops using with the temporary credentials and resumes its original credentials.

* **Federated Users:**

Federated Users can sign in using the temporary credentials provided by an identity provider. AWS provides an IDP (identity provider) and temporary credentials associated with the role to the user. The credentials grant the access of permissions to the user.

Accessing IAM

You can work with AWS Identity and Access Management in any of the following ways.

AWS Management Console

The console is a browser-based interface to manage IAM and AWS resources.  When IAM Users working in the IAM Console and want to use the role, then they access the permissions of the role temporarily. An IAM Users give up their original permissions for this.

AWS Command Line Tools

You can use the AWS command line tools to issue commands at your system's command line to perform IAM and AWS tasks. Using the command line can be faster and more convenient than the console. The command line tools are also useful if you want to build scripts that perform AWS tasks.

AWS provides two sets of command line tools:  [AWS Command Line Interface](https://aws.amazon.com/cli/) (AWS CLI) and the [AWS Tools for Windows PowerShell](https://aws.amazon.com/powershell/).

AWS SDKs:

AWS provides SDKs (software development kits) that consist of libraries and sample code for various programming languages and platforms (Java, Python, Ruby, .NET, iOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to IAM and AWS. For example, the SDKs take care of tasks such as cryptographically signing requests, managing errors, and retrying requests automatically. For information about the AWS SDKs, include Tools for Amazon Web Services.

IAM HTTPS API:

You can access IAM and AWS programmatically by using the IAM HTTPS API, which lets you issue HTTPS requests directly to the service. When you use the HTTPS API, you must include code to digitally sign requests using your credentials.