AWS IAM: Working, Components, and Features Explained

Amazon Web Services (AWS) cloud provides a secure virtual platform where users can deploy their applications. Compared to an on-premises environment, AWS security provides a high level of data protection at a lower cost to its users. There are many types of security services, but Identity and Access Management (IAM) is one the most widely used. AWS IAM enables you to securely control access to AWS services and resources for your users. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

Let us begin this AWS IAM tutorial by understanding AWS security.

What is AWS Security?

Cloud security is the highest priority in AWS. When you host your environment in the cloud, you can be assured that it's hosted in a data center or in a network architecture that's built to meet the requirements of the most security-sensitive organization. Additionally, this high level of security is available on a pay-as-you-go basis, meaning there is really no upfront cost, and the cost for using the service is a lot cheaper compared to an on-premises environment.

There are many types of security services available but some of them are widely used by AWS, such as:

- IAM
- Key Management System (KMS)
- Cognito
- Web Access Firewall (WAF)

We shall deal with IAM in this tutorial.

IAM enables you to manage access to AWS services and resources in a very secure manner. With IAM you can create groups and allow those users or groups to access some servers, or you can deny them access to the service.

Why IAM?

Before AWS or IAM, passwords were often shared in corporate environments in a very insecure manner: over the phone or through email. Often only one admin password existed, which was commonly stored in a set location, or there was only one person who could reset it, and you needed to call the person to ask for the admin password over the phone. That was not secure at all, because anybody could walk by and eavesdrop and then walk away with the password and access to your system and information.

Today we have a more secure communication tool: a third-party application called Slack, which is hosted on AWS. It helps people to share a document through the application so that eavesdropping is eliminated.

In the next section of the AWS IAM tutorial, let us understand what IAM is.

What is IAM?

AWS Identity and Access Management (IAM) is a web service for securely controlling access to AWS resources. It enables you to create and control services for user authentication or limit access to a certain set of people who use your AWS resources.

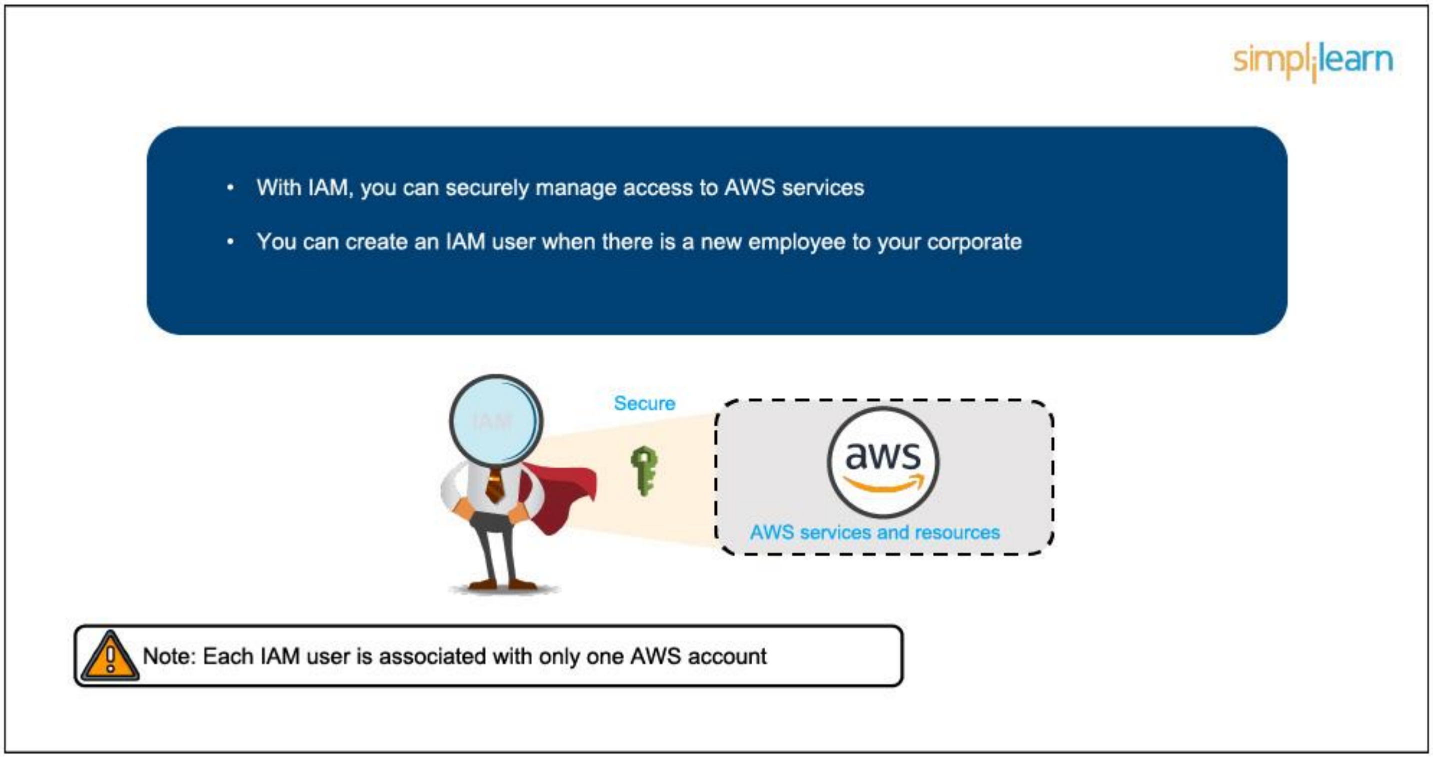
How Does IAM Work?

The IAM workflow includes the following six elements:

1. A principal is an entity that can perform actions on an AWS resource. A user, a role or an application can be a principal.
2. Authentication is the process of confirming the identity of the principal trying to access an AWS product. The principal must provide its credentials or required keys for authentication.
3. Request: A principal sends a request to AWS specifying the action and which resource should perform it.
4. Authorization: By default, all resources are denied. IAM authorizes a request only if all parts of the request are allowed by a matching policy. After authenticating and authorizing the request, AWS approves the action.
5. Actions are used to view, create, edit or delete a resource.
6. Resources: A set of actions can be performed on a resource related to your AWS account.

Let us explore the components of IAM in the next section of the AWS IAM tutorial.

Components of IAM



- With IAM, you can securely manage access to AWS services
- You can create an IAM user when there is a new employee to your corporate

Note: Each IAM user is associated with only one AWS account

There are other basic components of IAM. First, we have the **user**; many users together form a **group**. **Policies** are the engines that allow or deny a connection based on policy. **Roles** are temporary credentials that can be assumed to an instance as needed.

- **Users**

An IAM user is an identity with an associated credential and permissions attached to it. This could be an actual person who is a user, or it could be an application that is a user. With IAM, you can securely manage access to

AWS services by creating an IAM user name for each employee in your organization. Each IAM user is associated with only one AWS account. By default, a newly created user is not authorized to perform any action in AWS. The advantage of having one-to-one user specification is that you can individually assign permissions to each user.

- **Groups**

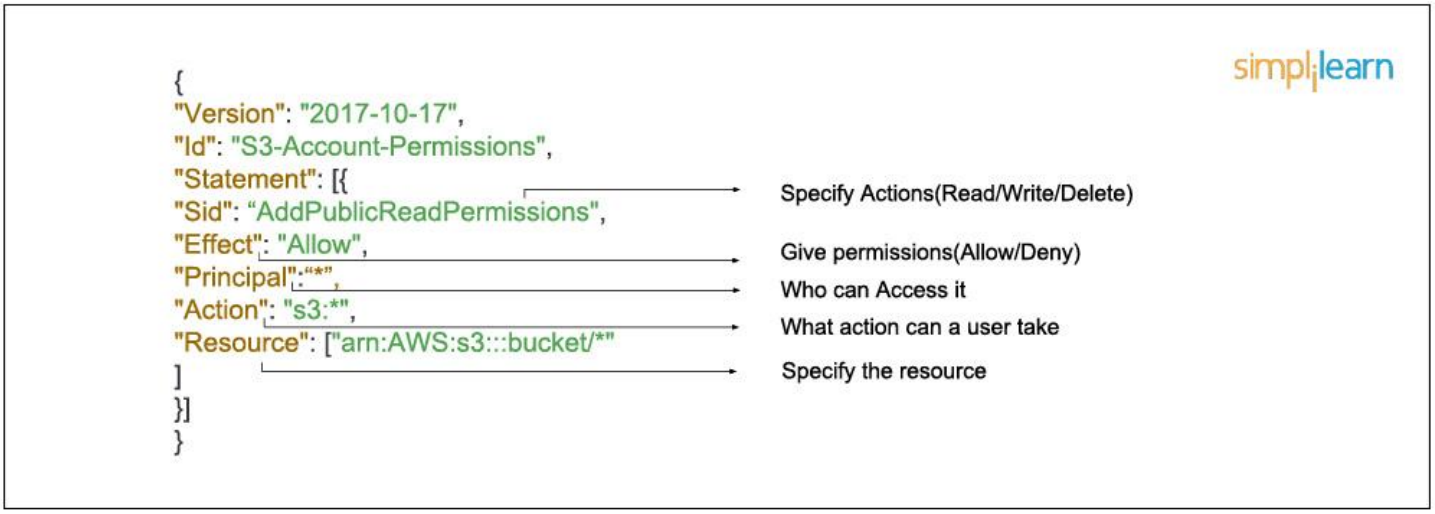
A collection of IAM users is an IAM group. You can use IAM groups to specify permissions for multiple users so that any permissions applied to the group are applied to the individual users in that group as well. Managing groups is quite easy. You set permissions for the group, and those permissions are automatically applied to all the users in the group. If you add another user to the group, the new user will automatically inherit all the policies and the permissions already assigned to that group. This lessens the administrative burden.

- **Policies**

An IAM policy sets permission and controls access to AWS resources. Policies are stored in AWS as JSON documents. Permissions specify who has access to the resources and what actions they can perform. For example, a policy could allow an IAM user to access one of the buckets in Amazon S3. The policy would contain the following information:

1. Who can access it
2. What actions that user can take
3. Which AWS resources that user can access
4. When they can be accessed

In JSON format that would look like this:



```
{
"Version": "2017-10-17",
"Id": "S3-Account-Permissions",
"Statement": [{
"Sid": “AddPublicReadPermissions",
"Effect": "Allow",
"Principal":“*”,
"Action": "s3:*",
"Resource": ["arn:AWS:s3:::bucket/*"
]
}]
}
```

simplilearn
Specify Actions(Read/Write/Delete)
Give permissions(Allow/Deny)
Who can Access it
What action can a user take
Specify the resource

There are two types of policies: managed policies and inline policies.

1. A **managed policy** is a default policy that you attach to multiple entities (users, groups, and roles) in your AWS account. Managed policies, whether they are AWS-managed or customer-managed, are stand-alone identity-based policies attached to multiple users and/or groups.
2. **Inline policies** are policies that you create that are embedded directly into a single entity (user, group or role).

- **Roles**

An IAM role is a set of permissions that define what actions are allowed and denied by an entity in the AWS console. It is similar to a user in that it can be accessed by any type of entity (an individual or AWS service). Role permissions are temporary credentials.

For example, you might want to allow a mobile app to use AWS resources, but you do not want it to save the key, credential or password. Or you might want to give access to resources to a user who already has an identity defined outside of AWS, such as a user who already has Google or Facebook authentication. If you want to provide someone with a service or let someone access resources in your account, you can use roles for that purpose too. You also might want to grant temporary access to your account to a third party, such as a consultant or an auditor. They're not permanent users, just users with temporary access to your environment.

Let us explore the features of IAM in the following section of the AWS IAM tutorial.

Gain proficiency in AWS IAM and the security attributes with the AWS Certification Training. Check out the course preview now.

Features of IAM

To review, here are some of the main features of IAM:

- **Shared access to the AWS account.** The main feature of IAM is that it allows you to create separate usernames and passwords for individual users or resources and delegate access.
- **Granular permissions.** Restrictions can be applied to requests. For example, you can allow the user to download information, but deny the user the ability to update information through the policies.
- **Multifactor authentication (MFA).** IAM supports MFA, in which users provide their username and password plus a one-time password from their phone—a randomly generated number used as an additional authentication factor.
- **Identity Federation.** If the user is already authenticated, such as through a Facebook or Google account, IAM can be made to trust that authentication method and then allow access based on it. This can also be used to allow users to maintain just one password for both on-premises and cloud environment work.
- **Free to use.** There is no additional charge for IAM security. There is no additional charge for creating additional users, groups or policies.
- **PCI DSS compliance.** The Payment Card Industry Data Security Standard is an information security standard for organizations that handle branded credit cards from the major card schemes. IAM complies with this standard.
- **Password policy.** The IAM password policy allows you to reset a password or rotate passwords remotely. You can also set rules, such as how a user should pick a password or how many attempts a user may make to provide a password before being denied access.

In the last section of the AWS IAM tutorial, let us go through a demo on how to create an S3 bucket using the multifactor authentication (MFA) feature.

Demo: Create an S3 Bucket Using the MFA Feature

The final segment of this article puts together all of the information presented and uses it to solve a basic problem.

Problem statement: To create an S3 bucket for a company in which each user can read and write data with multifactor authentication.

Task: To create policies and assign permissions for a user and a group.

- Provide access (read and write) to the developer group.
- Provide a policy in which a user is allowed to read or denied permission to write an object in an S3 bucket.

This is a very good use case if you have sensitive data in an S3 bucket and you want only privileged or MFA-authenticated users to make changes to those buckets. For those privileged users, you would enable multifactor authentication.

Conclusion

The information provided in this AWS IAM tutorial gave you a clear idea of AWS security and IAM. Amazon Web Services offers many remote computing services apart from security services. As companies across the world are adopting AWS Cloud, there will be a huge demand for professionals who have in-depth knowledge of AWS principles and services.

Source: Simplilearn.com