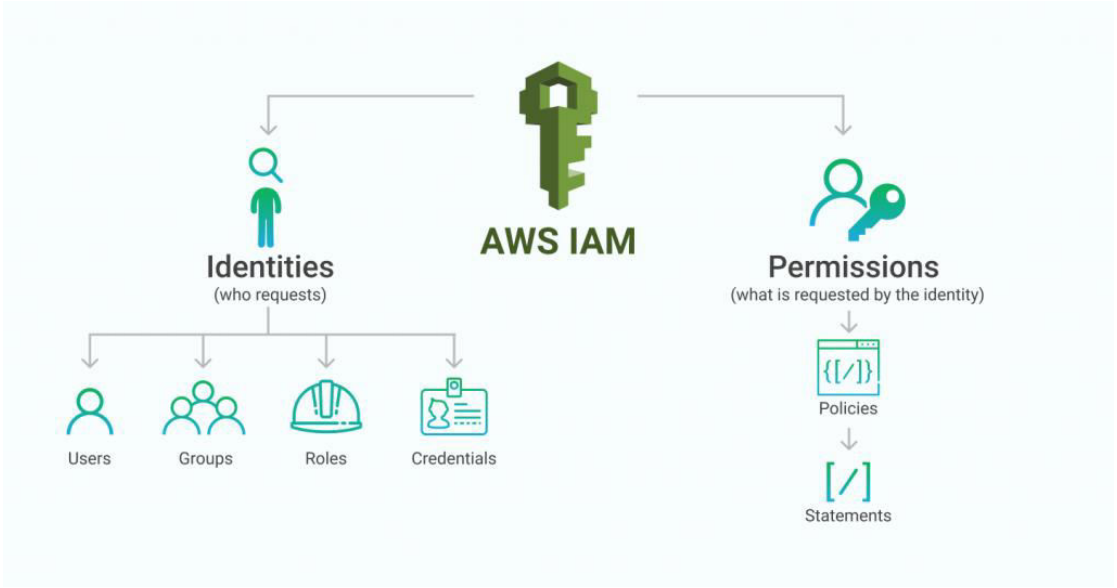
AWS IAM

What is AWS IAM?



- 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.

Users

- A user in IAM is a user of the AWS account and is a user of the organization that uses the AWS account.
- A user in IAM can be a developer, a HR or any member of the organization using the AWS account.
- A user can have AWS Management Console access or Programmatic access.
- A user can change his password after his first sign-in to the console based on the permissions given by the organization’s cloud administrator.
- A user uses username and password to login to the AWS Management Console.
- A user uses access key id and secret access key to access AWS resources APIS via programmatic access.
- A policy can be attached to the user to restrict the user actions in the AWS account.
- A user can be added to a group or can be an independent user of the AWS account.

Groups

- A group in an organization can be a collection of users such as HR Team or Development Team or DevOps Team or Data Analyst Team and so on.
- A group in the AWS account is the same as that of a group in an organization.
- A group in the AWS account is a collection of IAM users performing the same actions in the AWS account.
- For Eg: The HR team in the organization is using an S3 bucket to store resumes. Therefore IAM users are created in AWS IAM and added to a group called HR who have access to only S3 by the usage of policies.
- A policy can be attached to a group to restrict access to AWS services for all members of the group.
- For Eg : All members of the developer IAM group can launch virtual machines.

Roles

- An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS.
- However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it.
- In summary, a role can be used by any AWS resource or a person in order to perform actions with AWS resources. It provides the person or AWS resource temporary credentials which are refreshed every few minutes by AWS internally.
- A policy is attached to the role to define the actions that the role can do with AWS resources that assumes that role.
- For Eg : An EC2 instance can assume a role to access S3 to download web content to serve users.

Policies

- A policy is an entity that, when attached to an identity or resource, defines their permissions. Policies are stored in AWS as JSON documents and are attached to principals as identity-based policies in IAM.
- You can attach an identity-based policy to a principal (or identity), such as an IAM group, user, or role. Identity-based policies include AWS managed policies, customer managed policies, and inline policies.
- **Import** — You can import a managed policy within your account and then edit the policy to customize it to your specific requirements. A managed policy can be an AWS managed policy, or a customer managed policy that you created previously.
- **Visual editor** — You can construct a new policy from scratch in the visual editor. If you use the visual editor, you do not have to understand JSON syntax.
- **JSON** — In the JSON tab, you can create a policy using JSON syntax. You can type a new JSON policy document or paste an example policy.

Policies

- An IAM policy to provide administrator access to the AWS account.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "*",
         "Resource": "*"
      }
   ]
}
```

The IAM policy has 5 elements

- Version
- Statement
- Effect – Allow/Deny
- Action
- Resource

An IAM policy has versions where an IAM policy can have maximum up to 5 versions. The 6th version replaces the 1st version created.

Types of Policies

- An IAM policy can be of three types :
- **AWS Managed Policy** – There are many AWS managed policies in IAM that serve most of the purposes for restricting access to AWS resources. The only disadvantage is that there are not AWS managed policies for all customized actions on the AWS account. There comes in customer managed policies. These policies can be applied across users, groups and roles.
- **Customer Managed Policy** – An administrator can create his own customer managed policy with specific actions and conditions. It can be created using visual editor or JSON template. For Eg: Providing access to only writing content to a specific folder within a specific AWS S3 bucket. These policies can be applied across users, groups and roles.
- **Inline Policy** - An inline policy is a policy that's embedded in a principal entity (a user, group, or role). An inline policy is applicable only to that specific user, group or role.

Features of IAM

- Fine-grained access control to AWS resources
- Multi-factor authentication for highly privileged users
- Manage access control for mobile applications with Web Identity Providers
- Integrate with your corporate directory
- Manage and control password policy
- Create users, groups, roles
- Attach custom policies to users, groups and roles
- Enable and disable regions to use in the AWS account to launch resources

Fine Grained Access Control

- IAM enables your users to control access to AWS service APIs and to specific resources. IAM also enables you to add specific conditions such as time of day to control how a user can use AWS, their originating IP address, whether they are using SSL, or whether they have authenticated with a multi-factor authentication device.
- The administrator managing the AWS account will be able to create users and groups and attach custom policies or AWS managed policies to restrict the actions done by the user in the AWS management console or through AWS service APIs.
- A policy is a JSON document with specific elements.

Multi-factor Authentication

- Protect your AWS environment by using AWS MFA, a security feature available at no extra cost that augments user name and password credentials. MFA requires users to prove physical possession of a hardware MFA token or MFA-enabled mobile device by providing a valid MFA code.
- Use the following mobile application in the play store for android or app store for IOS devices to install the following application on your phone to authorize second level of security access to your AWS account.
- Google Authenticator for android/iOS devices.
- Authenticator for windows devices.

Best Practices

Users – Create individual users.

Groups – Manage permissions with groups.

Permissions – Grant least privilege.

Auditing – Turn on AWS CloudTrail.

Password – Configure a strong password policy.

MFA – Enable MFA for privileged users.

Roles – Use IAM roles for Amazon EC2 instances.

Sharing – Use IAM roles to share access.

Rotate – Rotate security credentials regularly.

Conditions – Restrict privileged access further with conditions.

Root – Reduce or remove use of root.

Demo - IAM

- Create two users in IAM with AWS Management Console and Programmatic Access.
- These two users should be able to create new passwords for themselves on signing into the AWS account.
- Add these two users to the HR group.
- Add a policy to the HR group to access AWS S3 service. Attach the AWS Managed policy AmazonS3FullAccess to the group.
- Login with any one of the user credentials to the AWS account and access S3 service and try accessing other services too and notice the difference.
- Remove or detach the managed policy from the group.
- Create a customer managed policy denoting the same access as that of the AWS managed policy for S3. Attach the customer managed policy to the IAM group called HR.
- Now try accessing the AWS S3 service as a user part of the HR group.
- Install the AWS CLI. Configure the AWS CLI with the one of the user's credentials and try accessing the AWS S3 API's such as listing all buckets.
- View password policy, Set up MFA and view the policy versions on the console.