Identity & Access Management (IAM)

DOCUMENTATION DISTILLED

What is IAM?

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources for your users. You use IAM to control who can use your AWS resources (authentication) and what resources they can use and in what ways (authorization).

IAM Features:

- Shared access to your AWS account
- Granular permissions
- Secure access to AWS resources for applications that run on Amazon EC2
- Multi-factor authentication (MFA)
- Identity federation
- Identity information for assurance (e.g. CloudTrail)
- PCI DSS Compliance
- Integrated with many AWS services
- Eventually Consistent (Global information that is replicated across regions)
- Free to use

Accessing IAM:

- AWS Management Console
- AWS Command Line Tools
- AWS SDKs
- IAM HTTPS API

Federating Existing Users:

Federation is particularly useful in these cases:

- Your users already have identities in a corporate directory (SAML 2.0 or identity broker app).
- Your users already have Internet identities (Facebook, Google etc).

IAM Entity Name Limits

- Names of users, groups, roles, policies, instance profiles, and server certificates must be alphanumeric, including the following common characters: plus (+), equal (=), comma (,), period (.), at (@), underscore (_), and hyphen (-).
- Names of users, groups, and roles must be unique within the account. They are not distinguished by case, for example, you cannot create groups named both "ADMINS" and "admins".
- AWS account ID aliases must be unique across AWS products, and must be alphanumeric following DNS naming conventions. An alias must be lowercase, it must not start or end with a hyphen, it cannot contain two consecutive hyphens, and it cannot be a 12 digit number.

IAM Entity and Object Limits

- Customer managed policies in an AWS account: 1500
- Groups in an AWS account: 100
- Roles in an AWS account: 500
- Users in an AWS account: 5000 (not Federated users)
- Access keys assigned to an IAM user: 2
- Groups an IAM user can be a member of: 10
- Aliases for an AWS account: 1

Identities (Users, Groups, and Roles)

Users

By default, a brand new IAM user has no permissions to do anything.

When you create a user, IAM creates these ways to identify that user:

- A "friendly name" for the user, which is the name that you specified when you created the user, such as Bob or Alice. These are the names you see in the AWS Management Console.
- An Amazon Resource Name (ARN) for the user.
- A unique identifier for the user. This ID is returned only when you use the API, Tools for Windows PowerShell, or AWS CLI to create the user; you do not see this ID in the console.

You can access AWS using:

- Console password
- Access keys: A combination of an access key ID and a secret access key.
- SSH keys for use with AWS CodeCommit
- Server certificates: SSL/TLS certificates that you can use to authenticate with some AWS services.

Password Policies

You can use a password policy to do these things:

- Set a minimum password length.
- Require specific character types, including uppercase letters, lowercase letters, numbers, and non-alphanumeric characters. Be sure to remind your users that passwords are case sensitive.

- Allow all IAM users to change their own passwords.
    - Note: When you allow your IAM users to change their own passwords, IAM automatically allows them to view the password policy. IAM users need permission to view the account's password policy in order to create a password that complies with the policy.
- Require IAM users to change their password after a specified period of time (enable password expiration).
- Prevent IAM users from reusing previous passwords.
- Force IAM users to contact an account administrator when the user has allowed his or her password to expire.

Access Keys

Users need their own access keys to make programmatic calls to AWS from the AWS Command Line Interface (AWS CLI), Tools for Windows PowerShell, the AWS SDKs, or direct HTTP calls using the APIs for individual AWS services.

You can give your users permission to list, rotate, and manage their own keys.

You can generate and download a credential report that lists all users in your account and the status of their various credentials, including passwords, access keys, and MFA devices. You can get a credential report from the AWS Management Console, the AWS SDKs and Command Line Tools, or the IAM API.

Unique IDs

When IAM creates a user, group, role, policy, instance profile, or server certificate, it assigns to each entity a unique ID. For the most part, you use friendly names and ARNs when you work with IAM entities, so you don't need to know the unique ID for a specific entity.

However, the unique ID can sometimes be useful when it isn't practical to use friendly names. For example:

> Within your account, a friendly name for a user, group, or policy must be unique. For example, you might create an IAM user named David. Your company uses Amazon S3 and has a bucket with folders for each employee; the bucket has a resource-based policy (a bucket policy) that lets users access only their own folders in the bucket. Suppose that the employee named David leaves your company and you delete the corresponding IAM user. But later another employee named David starts and you create a new IAM user named David. If the bucket policy specifies the IAM user named David, the policy could end up granting the new David access to information in the Amazon S3 bucket that was left by the former David.

Changing a user's friendly name does not change their unique id and permissions associated to it.

IAM Policies

A policy lets you specify the following:

- Actions: what actions you will allow.

- Resources: which resources you allow the action on.
- Effect: what the effect will be when the user requests access—either allow or deny.

Policies are documents that are created using JSON. A policy consists of one or more statements, each of which describes one set of permissions.

You cannot save any policy that does not comply with the established policy syntax. You can use Policy Validator to detect and correct invalid policies.

If your policy includes multiple statements, a logical OR is applied across the statements at evaluation time. Similarly, if multiple policies are applicable to a request, a logical OR is applied across the policies at evaluation time.

When a policy statement contains a Condition element, the statement is only in effect when the Condition element evaluates to true. If your policy includes multiple conditions, a logical AND is applied across the conditions at evaluation time.

[IAM Request Evaluation Logic]