Course and Labs

AWS Identity and Access Management (IAM)

Mohanraj Shanmugam

Overview

- AWS Identity and Access Management (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.

Functionality

- Manage IAM users and their access – You can create users in IAM, assign them individual security credentials ,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.
- Manage IAM roles and their 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.
- Manage federated users and their permissions – You can enable identity federation to allow existing identities (users, groups, and roles) in your enterprise to access the AWS Management Console, call AWS APIs, and access resources, without the need to create an IAM user for each identity.

Use Cases

- Fine-grained access control to AWS resources
  - 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.
- Integrate with your corporate directory
  - IAM can be used to grant your employees and applications federated access to the AWS Management Console and AWS service APIs, using your existing identity systems such as Microsoft Active Directory.
  - You can use any identity management solution that supports SAML 2.0, or feel free to use one of our federation samples

Manage access control for mobile applications with Web Identity Providers

- You can enable your mobile and browser-based applications to securely access AWS resources by requesting temporary security credentials that grant access only to specific AWS resources for a configurable period of time.

Multi-factor authentication for highly privileged users

- 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.

IAM Best Practices

- Manage Users
  - Manage IAM users and their access—You can create users in IAM, assign them individual security credentials (such as access keys, passwords, and multi-factor 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. IAM users can be:
  - Privileged administrators who need console access to manage your AWS resources.
  - End users who need access to content in AWS.
  - Systems that need privileges to programmatically access your data in AWS.

Create Groups

- A group is a collection of IAM users.
- Groups let you assign permissions to a collection of users, which can make it easier to manage the permissions for those users.
- Create Multiple groups as per the role

Grant Least privilege

- Permissions
  - Permissions let you specify who has access to AWS resources and which actions they can perform on those resources
  - Every AWS Identity and Access Management (IAM) user starts with no permissions.
  - In other words, by default, users can do nothing, not even view their own access keys.
  - To give a user permission to do something, you can add the permission to the user (that is, attach a policy to the user), or add the user to a group that has the desired permission.

Grant Least privilege

- Policies
- To assign permissions to a user, group, role, or resource, you create a policy that lets you specify:
  - Actions – Which AWS actions you allow. For example, you might allow a user to call the Amazon S3 ListBucket action. Any actions that you don't explicitly allow are denied.
  - Resources – Which AWS resources you allow the action on. For example, what Amazon S3 buckets will you allow the user to perform the ListBucket action on? Users cannot access any resources that you do not explicitly grant permissions to.
  - Effect – Whether to allow or deny access. Because access is denied by default, you typically write policies where the effect is to allow.
  - Conditions – Which conditions must be present for the policy to take effect. For example, you might allow access only to the specific S3 buckets if the user is connecting from a specific IP range or has used multi-factor authentication at login.

Auditing

- AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you.
- The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service.
- With CloudTrail, you can get a history of AWS API calls for your account, including API calls made via the AWS Management Console, AWS SDKs, command line tools, and higher-level AWS services
- The AWS API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing.

Configure Strong Password

- AWS Identity and Access Management (IAM) lets you manage several types of long-term security credentials for IAM users:
  - Passwords – Used to sign in to secure AWS pages, such as the AWS Management Console, the AWS Discussion Forums, and the AWS Premium Support site.
  - Access keys – Used to make secure REST or Query protocol requests to any AWS service API.
  - Key pairs – Used for Amazon CloudFront.

MFA – Enable MFA for privileged users.

- AWS Multi-Factor Authentication (MFA) is a simple best practice that adds an extra layer of protection on top of your user name and password.
- With MFA enabled, when a user signs in to an AWS website, they will be prompted for their user name and password (the first factor—what they know), as well as for an authentication code from their AWS MFA device (the second factor—what they have).
- Taken together, these multiple factors provide increased security for your AWS account settings and resources.

Roles – Use IAM roles for Amazon EC2 instances.

- IAM roles allow you to delegate access to users or services that normally don't have access to your organization's AWS resources.
- IAM users or AWS services can assume a role to obtain temporary security credentials that can be used to make AWS API calls. Consequently, you don't have to share long-term credentials or define permissions for each entity that requires access to a resource.

Best Practices

- 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.