AWS Identity and Access Management (IAM)

- Manages access to AWS services
- IAM permissions are very granular and they control access to both the data plane and control plane.
- You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

Benefits:

- Set permission guardrails and fine-grained access
- Manage workloads and workforce identities across Aws accounts
- Use temporary credentials and permissions sets to access AWS resources
- Analyze access and validate IAM policies keeping least privilege

Accessing IAM

- AWS management console
- AWS command line tools – AWS CLI , AWS tools for Windows PowerShell.
- AWS SDKs
- IAM Query API

Authentication

- A principal must be authenticated (signed in) to use AWS services.
- Root user – sign in with email and password
- A federated user – authenticated by identity provider and granted access to AWS by assuming role
- IAM user – sign with account ID or alias, then username and password
- Workloads from API – temporary credentials or long-term credentials using access key and secret key
- For additional security use MFA.

When you are authorized to access AWS resources

- You must be authenticated(signed in) as root user, IAM user or by assuming an IAM role.
- Can sign in to AWS as a federated identity (e.g. company's single sign-on, google, Facebook credentials)
- To access AWS programmatically, use SDK and CLI to cryptographically sign requests using your credentials.
- If not using AWS tools (SDK / CLI) you must sign request yourself.
- Regardless of authentication method, for addition security use MFA (multifactor authentication)

Authorization

- Once authenticated, during authorization AWS uses values from request to check for policies.
- AWS uses policies to determine whether to accept or deny request.
- The evaluation logic is applied in a account :
  - By default, all requests are denied
  - Explicit allow in identity-based or resource-based policy overrides this default.
  - Organizations SCP, IAM permissions boundary, or a session policy overrides the allow
  - An explicit deny in any policy overrides any allows.
  -

Users

- IAM and AWS IAM Identity Center both can be used to create new users or federate existing users into AWS.
- IAM users are granted long-term credentials
- IAM Identity Center have temporary credentials that are established each time the user signs-in to AWS.

root user :

- has complete access to all AWS services and resources in the account.

- accessed by signing in with the email address and password
- never use the root user for your everyday tasks
- Tasks that require root user:
  - Change account settings : name, email id, root user password, root user access keys
  - Restore IAM user permission: If the only IAM administrator accidentally revokes their own permissions, you can sign in as the root user to edit policies and restore those permissions.
  - Close AWS account
  - View certain tax invoices
  - Register as seller in Reserved instance marketplace.
  - Configure S3 bucket to enable MFA
  - Edit or delete S3 bucket policy that denies all permissions.
  - Edit or delete SQS queue that denies all principals.

IAM-user
- Single person or application with specific permissions.
- IAM user has long-term credentials (password & access keys).
- AWS recommends to rotate access keys
- IAM group = collection of IAM users.
- Cannot sign as a group. Use groups to specify permissions for multiple users at a time.

Federating existing users
- Users already exist in a corporate directory.(corporate directory compatible with SAML 2.0, or Microsoft AD, if not create identity broker app)
- Users already have Internet identities.(Amazon, Facebook, Google, OpenID Connect (OIDC) compatible identity provider)

Assume an IAM role
IAM role has specific permissions. Similar to IAM user but not associated with a specific person.
You can temporarily assume an IAM role

- All IAM actions, resources, and condition keys available for AWS services are listed in this documentation page.
- Build policies with the AWS Policy Generator to help formulate the syntax.
- Test your policies with the IAM Policy Simulator, and see the documentation for this tool.
- Using the AWS IAM role-comparison tool to extract and compare IAM roles from different AWS accounts.

Permissions and policies

Principal - a person or application that is authenticated using an IAM entity (user or role).
Identities - users, groups of users, or roles
Resource – AWS resources (S3 bucket, lambda function)

Two types of policies : Identity-based policy and
Identity based policy
- Permissions policy attached to identities (users, groups of users, or roles)
- *Identity-based policies* control what actions the identity can perform, on which resources, and under what conditions.
- Further categorized in : Managed policies (AWS and Customer) and inline policies.
-

Resource-based policy
- Permissions policies that you attach to a resource such as an Amazon S3 bucket or an IAM role trust policy.

- *Resource-based policies control what actions a specified principal can perform on that resource and under what conditions.*
- Resource-based policies are inline policies, and there are no managed resource-based policies.
- Used to enable cross-account access
-

Permissions boundaries

- advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity.
- When this is set, entity can perform only the actions that are allowed by both its identity-based policies and its permissions boundaries.
- Resource-based policies are not limited by the permissions boundary.

Service control policies (SCPs)

- SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU).
- The SCP limits permissions for entities in member accounts, including each AWS account root user.

Access control lists (ACLs)

- ACLs are similar to resource-based policies, although they are the only policy type that does not use the JSON policy document format.
- These are service policies that allow you to control which principals in another account can access a resource.
- Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs.

Session Policies

- Advanced policies passed as a parameter when programmatically creating a temporary session for a role or federated user.
-

ABAC (Attribute-based access control)

- Defines permissions based on attributes called tags
- Can attach tags to IAM entities or AWS resources

IAM Access Analyzer

- Helps **:**
  - Identifying resources shared with an external entity.
  - Identifying unused access granted to IAM users and roles
  - Validating policies against policy grammar and AWS best practices
  - Validating policies against your specified security standards
  - Generating policies based on access activity in your AWS CloudTrail logs.
  - You can also use IAM Access Analyzer APIs to preview public and cross-account access for your Amazon S3 buckets, AWS KMS keys, IAM roles, Amazon SQS queues and Secrets Manager secrets by providing proposed permissions for your resource.

Managing Access permissions

- Granting access to billing information and tools :
  - By default, IAM users do not have access to AWS billing and cost management tools. To grant access
    - Activate IAM access
    - Attach IAM policy that grants permissions to access billing and cost console.