AWS Security – IAM

IAM provides the infrastructure necessary to control authentication and authorization for your account. It includes following elements.

IAM Resources

The user, group, role, policy, and identity provider objects that are stored in IAM. As with other AWS services, you can add, edit, and remove resources from IAM.

IAM Identities

The IAM resource objects that are used to identify and group. You can attach a policy to an IAM identity. These include users, groups, and roles.

IAM Entities

The IAM resource objects that AWS uses for authentication. These include IAM users and roles.

Principals

A person or application that uses the AWS account root user, an IAM user, or an IAM role to sign in and make requests to AWS. Principals include federated users and assumed roles.

Actions or operations

IAM supports approximately 40 actions for a user resource, including the following actions:

* CreateUser
* DeleteUser
* GetUser
* UpdateUser

To allow a principal to perform an operation, you must include the necessary actions in a policy that applies to the principal or the affected resource.

## **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.

AWS IAM Resources

1. User
   1. Can be user/application.
2. Group
   1. Collection of users.
   2. All users in the group will inherit permission of the group
3. Role
   1. Roles created are assigned to AWS resources.
4. Policy
   1. Defines Permissions
   2. JSON formatted document
   3. Can be assigned to User/Group/Role

Role Vs Policy

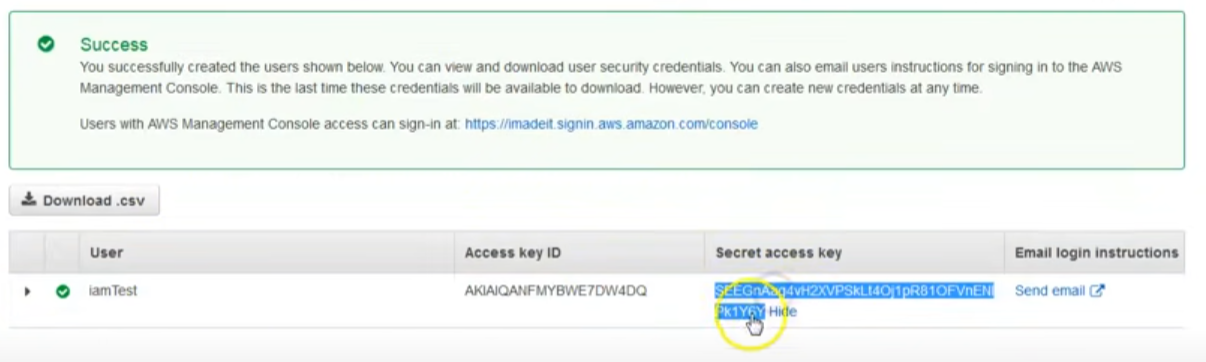
Role: An IAM role is an IAM identity that you can create in your account that has specific permissions.

Policy:  IAM policies define permissions for an action regardless of the method that you use to perform the operation

Typically, you have a role and you assign polices to your role.

User

1. Create AWS user
2. For user, you have defined Access type.
3. Can assign permission using Policy Engine
4. Access ID/Keys.
   1. After creating user and permissions. You can Have
      1. Access Key ID
      2. Secrete Key Access



AWS Access type

1. Programmatic Access
2. AWS Management Console Access

Policy

1. Collections of permission
2. AWS managed (created by AWS)
3. Customer managed (created/mainte by AWS users based on need and custom requirements)

Studey material

<https://www.youtube.com/watch?v=DXNS-EP9sXM&ab_channel=KnowledgeIndiaAWSAzureTutorials>

AWS IAM Example: Policies and Permissions in Amazon S3

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/access-policy-language-overview.html>

Policy

* This diagram is an example of AWS IAM Policy
* This IAM Policy defines permission for AWS S3 bucket

Resources:

Buckets, objects, access points, and jobs are the Amazon S3 resources for which you can allow or deny permissions. Policy

Actions –

For each resource, Amazon S3 supports a set of operations. You identify resource operations that you will allow (or deny) by using action keywords.

For example, the s3:ListBucket permission allows the user to use the Amazon S3 GET Bucket (List Objects) operation.

Action: 'lambda:InvokeFunction', // The privilege you are giving to API gateway api's

Effect – What the effect will be when the user requests the specific action—this can be either allow or deny.

If you do not explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource. You might do this to make sure that a user can't access the resource

Principal – The account or user who is allowed access to the actions and resources in the statement. In a bucket policy, the principal is the user, account, service, or other entity that is the recipient of this permission.