IAM (Identity & Access Management)

IAM is a web service that helps you securely control access to AWS resources. You can use IAM to control who is Authenticated (signed-in) & Authorized (has permission) to use resources.

1. When you first create an AWS account you begin with a single sign-in identity that has completely access to all AWS service and resources in the account. This identity is called the AWS account “ROOT USER”
2. AWS strongly recommended that you do not use the Root User for your everyday task, even the administrative one.
3. Use other IAM user accounts to manage the Administrative task of your account and securely lock away the Root User Credentials & use them to perform only a few tasks.

Some important points for interview

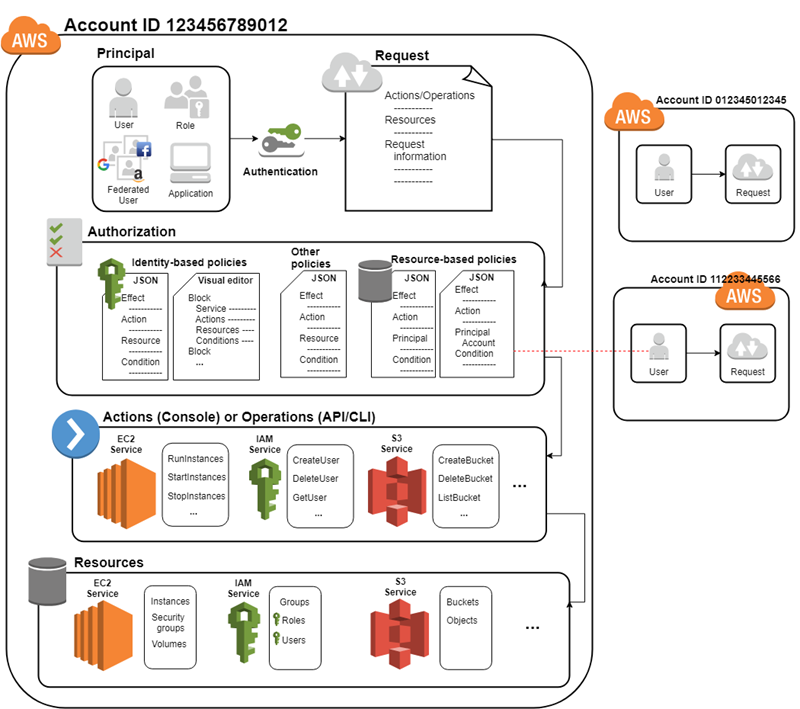
1. IAM user limit is 5000 per AWS account by default. You can exceed this limit using support center.
2. IAM user groups are 300 per AWS account.
3. You can add up to 10 users at one time.
4. IAM user can be a member of 10 groups.(max)

IAM Feature

1. **Shared access to your AWS account-** You can grant other people permission to administer and use resources in your AWS account without having to share your credential (password or access key)
2. **Granular permission –** (I) you can grant different permission to different people for different resources.

(II) For example you can allow some users complete access to EC2, S3, Dynamo DB etc. while others users you can allow read only access.

1. **Multifactor Authentication (MFA) -** You can add two factor authentications to your account and to individual user for extra security. You can use physical hardware or virtual MFA (ex.- Google authenticator)
2. **Identity Federation-** You can allow users who already have password elsewhere for ex- in your corporate network or with an internet identity provider to get temporary access to your aws account.
3. **PCI- DSS** **Compliance –** IAM support the processing storage and transmission of credit card by a merchant or service and has been validated as being complaint with payment card industries data security standard.
4. **Eventually consistent-** if a request to change some data is successfully the change is committed and safely stored however the change must be replicate across IAM which can take some time.
5. **Free to use-** AWS IAM is a feature of your AWS account offered at no additional charges. You will be charged only for use of other AWS services by your IAM users.



How IAM Work?

1. Principal – A principal is a person or application that can make a request for an action or operation on AWS resource.

Your administrative IAM user is your first principal.

IAM users, Roles, Federated users and applications are all AWS principal.

1. Request –
2. When a principal tries to use the AWS management console, AWS API, AWS CLI that principal sends a request to AWS. The request includes the following information- Action, Resources, Principal, Environment data, Resource data
3. Actions- That the principal want to perform.
4. Resources- Resources upon which the action are performed.
5. Principal- Principal Information including the environment from which the request was made.
6. Environment data- such as IP Address, user agent, SSL enabled status or the time of day.
7. Resources data – data related to the resource that is requested.
8. Authentication- A principal sending a request must be authenticated (signed-into AWS) to send request to AWS.
9. Authorization- To authorize request, IAM user value from the request context to check for matching policies and determine whether to allow or deny the request.

Two types policies here

1. User (identity) based policy- specify permission allowed/denied for principal.
2. Note- by default the AWS root user has access to all the resources in that account.

**Resources based policy**- specifies permission allowed/denied for resources popular for granting cross account permission.

1. IAM checks each policy that matches the context of your request.
2. If a single policy includes a denied action, IAM denies the entire request and step evaluating. This is called Explicit Deny.
3. Action**- A**ction is defined by a service & is the things that you can do to a resources such as viewing, creating, editing & deleting that resources.
4. Resources- Example is EC2 instances, S3 bucket, Dynamo DB tables.

IAM 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. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session.

1. AWS Service Role- Many AWS services require that you use roles to allow the service to access resources in other services on your behalf. A role that a service assumes to perform actions on your behalf is called a service role. When a role serves a specialized purpose for a service, it is categorized as a service role for EC2 instances.
2. Another AWS Account- You can use IAM roles to delegate access to your AWS resources. With IAM roles, you can establish trust relationships between your trusting account and other AWS trusted accounts.

The trusting account owns the resource to be accessed and the trusted account contains the users who need access to the resource. However, it is possible for another account to own a resource in your account.

For example, the trusting account might allow the trusted account to create new resources, such as creating new objects in an Amazon S3 bucket. In that case, the account that creates the resource owns the resource and controls who can access that resource.

1. Web Identity- You can use Web Identity or OpenID Connect Federation (OIDC) identity providers instead of creating IAM users in your AWS account. With an identity provider (IdP), you can manage your user identities outside of AWS and give these external user identities permissions to access AWS resources in your account.
2. SAML 2.0 federation- You can use SAML 2.0 federation instead of creating IAM users in your AWS account. With an identity provider (IdP), you can manage your user identities outside of AWS and give these external user identities permissions to access AWS resources in your account.

IAM permissions boundaries- A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity. When you set a permissions boundary for an entity, the entity can perform only the actions that are allowed by both its identity-based policies and its permissions boundaries.

LABS

1. How to create users & groups
2. Policy two types we discuss (A) policy created by AWS (B) Inline policy we create
3. Roles with AWS services & Roles with another AWS account.

Steps for roles with another AWS account

1. For this lab we need two account
2. Take one account and create one group & add two users & give permission at group EC2 read only.
3. Create inline policy here.
4. Now go to another account & create one bucket.
5. Same account create role & select type of roles (another aws account) & give permission S3 read only permission.
6. Now sign-in one user & check he can access only EC2 not any other services.
7. Now go to user name & drop down you get switch roles click here then open new tab.
8. Provide necessary field here & you can see this user can access another account s3 bucket but we didn’t given permission this user for s3.