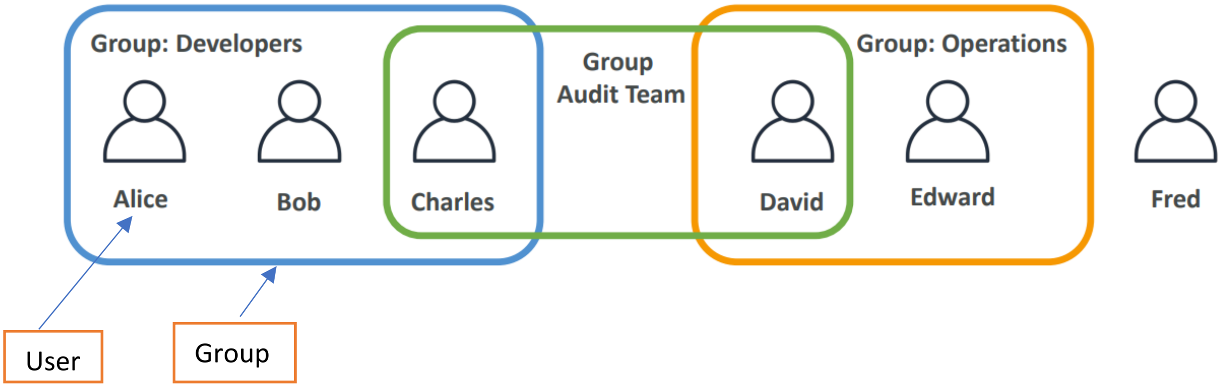
CNAD MSE

IAM: Identity Access Management

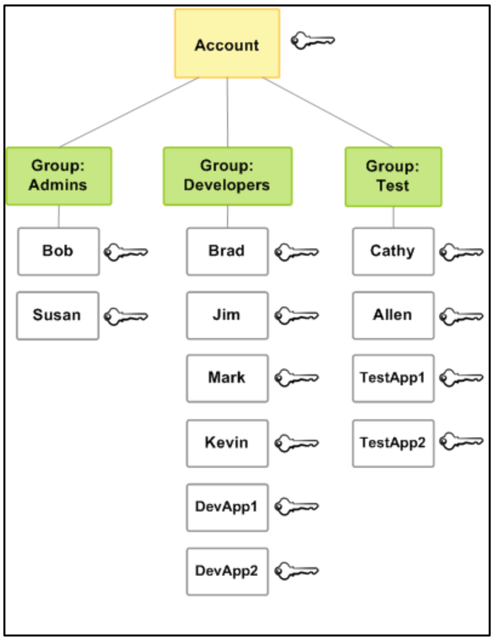
1. AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources.
2. It is a global service
3. With IAM, you can centrally manage permissions that control which AWS resources users can access.
4. It contains Users, Group

Users :

1. An AWS Identity and Access Management (IAM) user is an entity that you create in AWS.
2. A user in AWS consists of a name and credentials.
3. A user is associated with a set of security credentials that enable them to access and perform actions on AWS resources.
4. IAM allows you to create and manage users with different levels of access to AWS resources, control their permissions, and monitor their activities.
5. Users can be created and managed using the AWS Management Console, AWS Command Line Interface (CLI), or AWS SDKs.
6. Users can be granted permissions to access AWS services and resources through policies attached to their IAM user accounts.



Groups :



1. An IAM user group is a collection of IAM users.
2. Groups can be created and managed using the AWS Identity and Access Management (IAM) service.
3. IAM allows you to create groups and attach policies that define the permissions the group members should have.
4. Groups only contain users, not other groups.
5. You can use groups to organize users based on their roles, departments, or projects.

Roles :

1. An IAM role is an IAM identity that you can create in your account that has specific permissions.
2. Roles allow you to define a set of permissions for making AWS service requests.
3. Roles can be used to grant access to AWS resources across different AWS accounts.
4. IAM Roles can be used with AWS services: IAM Roles can be used with a variety of AWS services, such as EC2, Lambda, and Code Pipeline.
5. By using roles, you can improve the security of your AWS environment and simplify access management.

6. Overall, roles are a powerful tool for managing access to AWS resources and controlling permissions for service requests.

Policies and permissions in IAM :

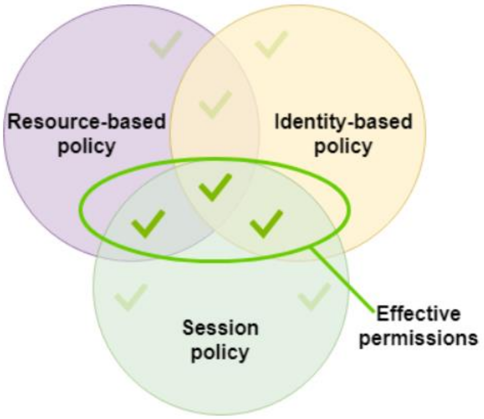
Policy types

A policy is an object in AWS that, when associated with an identity or resource, defines their permissions.

The following policy types, listed in order from most frequently used to less frequently used, are available for use in AWS.

- **Identity-based policies** – Attach managed and inline policies to IAM identities (users, groups). Identity-based policies grant permissions to an identity.
- **Resource-based policies** – Attach inline policies to resources. The most common examples of resource-based policies are Amazon S3 bucket policies and IAM role trust policies. Resource-based policies grant permissions to the principal that is specified in the policy.
- **Permissions boundaries** – Use a managed policy as the permissions boundary for an IAM entity (user or role). That policy defines the maximum permissions that the identity-based policies can grant to an entity, but does not grant permissions. Permissions boundaries do not define the maximum permissions that a resource-based policy can grant to an entity.
- **Organizations SCPs** – Use an AWS Organizations **service control policy (SCP)** to define the maximum permissions for account members of an organization or organizational unit (OU). SCPs limit permissions that identity-based policies or resource-based policies grant to entities (users or roles) within the account, but do not grant permissions.
- **Access control lists (ACLs**) – Use ACLs to control which principals in other accounts can access the resource to which the ACL is attached. ACLs are similar to resource-based policies, although they are the only policy type that does not use the JSON policy document structure.
  ACLs cannot grant permissions to entities within the same account.

- **Session policies** – Session policies limit permissions for a created session, but do not grant permissions.



Types of access to user :

1. Console (userid and password)
2. Programming (accesskey id and secret access key)

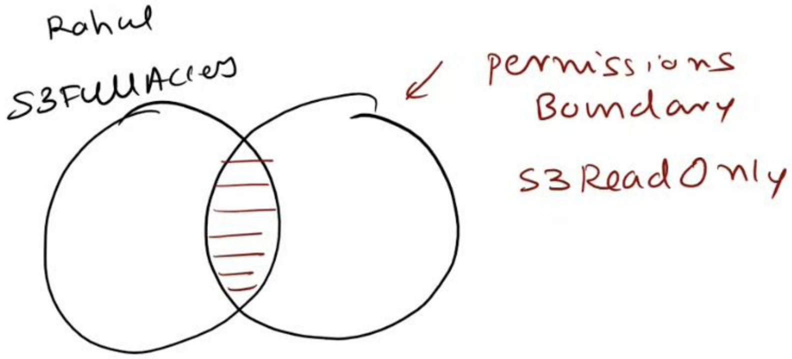
Permission boundary :

It will not assign the permission, it only set maximum permission for user (limit). When extra permissions are assigned by admin by mistake , permission boundary will help us.

Permission boundary is assigned individually not in group of user.

eg. Sam security specialist, S3 full access by mistake, handover to another, by mistake, purposefully

Maximum permission use is allowed to get.



S3 ReadOnly is a part of S3 FullAccess

Readonly
Full
Access

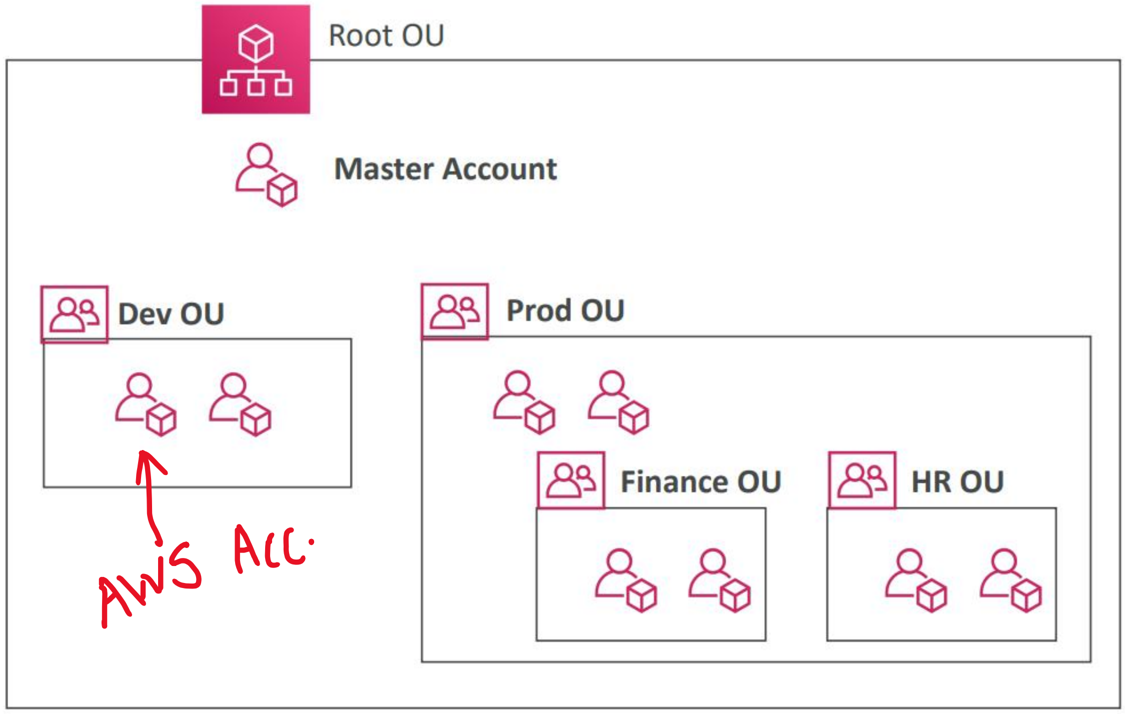
AWS Organizations :

It is build for organization which has many aws accounts.

Allows to manage multiple AWS accounts **centrally**

Eg. Infosys

Organizational Units (OU)



AWS Organization supports :

- Security policy
- Backup policy

- Consolidated Billing (client pay the bill to company and company will pay the bill to aws, to reduce multiple transactions)
- Cost saving (eg. For 500 instances aws give discount 5%(150 accounts combined 1 organization))

Service Control Policies (SCP) :

Applied at the OU or Account level.

SCP is applied to all the Users and Roles of the Account

S3 (Simple Storage Service)

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data.

Features of Amazon S3

- **Storage classes**

Each object in Amazon S3 has a storage class associated with it. if you list the objects in an S3 bucket, the console shows the storage class for all the objects in the list. All of these storage classes offer high durability.

1.