26/02/2025

IAM : Identity and Access Management

→ We will be using the Root user most of the time.

↳ Root user has the complete access (to create, delete, modify...etc)

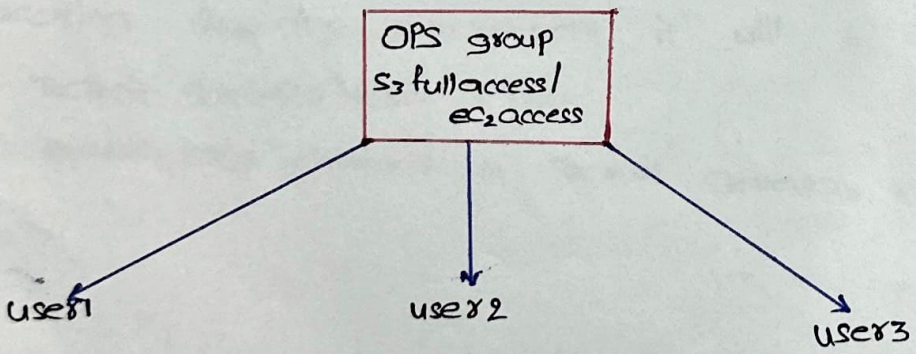
↳ IAM is used to create individual user access to each and every resource.

↳ IAM user is an individual identity with specific permissions to access resources with in a system.

IAM identity: Identity can be user, group, roles.

1) GUI access
2) CLI access / programmetic access.

→ Instead of giving permission/specific access to each and every of the user. I can add the users to a group, and what all the permissions were required, i can attach to the group.



To create a group:

User groups
↓
create group
↓
Name the group
↓
Add users to the group
↓
Attach permissions policies
↓
create group

Policy:

Policy defines the permissions of the IAM identity.

* Types of policies:

1) Identity Based Policies:

a) In-line policy – An inline policy is a policy created for a single IAM identity (a user, group, or role).

Inline policies maintain a strict one-to-one relationship between a policy and an identity.

They are deleted when you delete the identity.

IAM → users → Add permissions → create inline policy → → service (choose a service) → Actions (manual actions) → Resources (all resources) → review policy → Name → create policy

b) Managed policies:

(i) AWS managed policies: that are created and managed by AWS.

(ii) customer managed policies: Managed policies that you create and manage in your AWS account. customer managed policies provide more precise control over your policies than AWS managed policies.

2) Resource based policies:

are attached directly to resources and specify permissions for specific actions on the resource by some principals.

3) IAM permissions boundaries:

Define the maximum permissions for an IAM entity and are used as safe guards.

4) Access control lists (ACLs)

are attached to resources and control cross-account permissions for principals from other accounts.

5) Organizations service control policies (SCPs):

Specify the maximum level of permissions for an organization's accounts. These policies are used to limit the permissions that can be assigned within member accounts.

6) session policies:

are advanced policies used during temporary sessions for roles or federated users.

Role:

IAM role is an identity within your AWS account that has specific permissions.

It is similar to an IAM user, but is not associated with a specific person.

Types of Roles:

1) AWS service Role:

A service Role is an IAM role that a service assumes to perform actions on your behalf.

2) AWS account:

Allows entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

3) Web identity:

Allows users federated by the specific external web identity provider to assume this role to perform actions in this account.

4) SAML 2.0 federation (security Assertion markup language):

Allows users federated with SAML 2.0 from a corporate directory to perform actions in this account.

5) custom trust policy:

create a custom trust policy to enable others to perform actions in this account.

AWS - S3 BUCKET:

28/02/2025

* Simple storage service

↳ S3 is a global service.

↳ S3 service is not restricted to any region.

↳ S3 Bucket is a region specific.

↳ S3 is used for fixed objects.(files, images, video, documents).
Fixed means, the moment we upload it to the S3 we cant edit them. If we want to make any changes, we should download it and make changes.

→ S3 bucket has a storage of upto 5TB.
The largest object that can be uploaded in a single PUT is 5GB.

→ S3 Bucket name should be unique globally.

In AWS we have 3 different storages:

1) EBS (Elastic Bock storage).
2) S3 (Simple storage service)
3) EFS (Elastic File Systems).

* In EBS, we can connect multiple EBS to single EC_2.
we cant connect multiple EC_2 to single EBS.

* In S3, we can transfer data from multiple EC_2 to S3. from different region.

* EFS can also be used across different EC_2. in same region.

create Bucket → Name → AWS region → object ownership(ACLs ~~disabled~~ enable) →
→ Block all public access (disable) → Bucket versioning → Default encryption → create Bucket

→ Access control list (ACL's) grants permissions at object ~~or~~ level.
~~bucket level~~

→ Bucket policy applies only to buckets.

→ By default S3 storage class is standard.

↳ when we upload any object, that will be moved to the standard storage class.

→ we have different classes in S3.

1) standard IA
2) S3 glacer
3) Deep archieve Glacier
4) Standard

→ These are called as lifecycle rules for S3 objects.

→ S_3 standard - High durability, low-latency, and used for frequently accessed data.

→ S_3 standard-IA - lower cost than standard but with retrieval fees; best for less frequently accessed data.

→ S_3 -one zone -IA - cheaper than standard-IA but stored in a single AZ, making it less resilient.

→ S_3- Glacier Deep archive- cheapest storage for long-term retention, but retreival takes 12-48 hours.

→ S_3- Glacier Instant Retrieval - low cost, instant access for archive data that is rarely accessed.

→ S_3- Glacier Flexible Retrieval - Lower cost than Instant Retrieval but retrieval times range from minutes to hours.

* we can also use our S3 bucket to host a website.

Static websites can be hosted using S3.

↓

Amazon S3 → Buckets → Select the bucket → permissions → Static website hosting → edit → Enable → ~~save changes~~ → Hosting type (Host a static website) → index document (file name which was created) → Error document (file which was created) → save changes.

↓

Amazon S3 → Buckets → Select the Bucket → ~~upload~~ objects → upload → Add files → upload.

↳ Till now we are using the object URL to open the file.

↳ We need bucket URL to access my S3 as a static website.

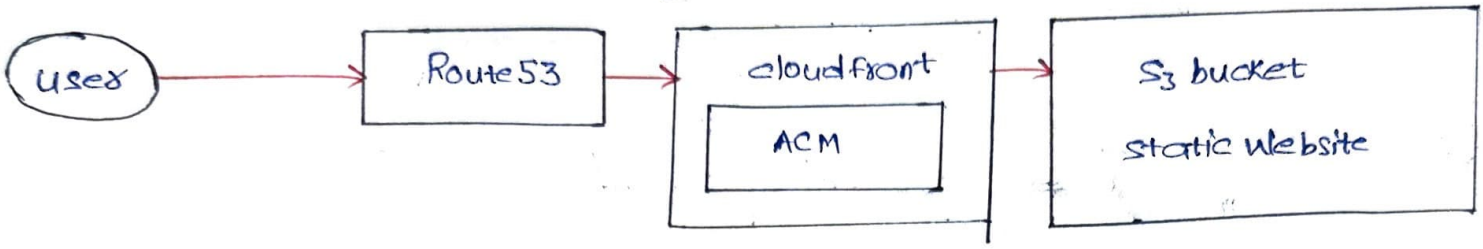
↓

Buckets → select the bucket → Properties → Static website hosting → URL

03/03/2025

What have you stored in S3 during your project?

↳ We have used S3 to store VPC flow logs, images related to our application and to deploy one static website.



Previously,

→ Whenever a user wants to access the static website, we were just copying the bucket URL and browsing it.

→ CloudFront will help us to improve the performance and also reduce the latency.

→ Route53 is used to configure the Domain and servers.

→ AWS ACM (Amazon certificate Manager) is a service that helps you manage SSL/TSL certificates for securing network communications and websites.

SSL - secure socket layer

TSL - Transport Layer security.

↳ with the help of ACM, i can purchase SSL certificate to my domain.

↳ ACM will add security to my URL with the help of SSL.

↳ By adding SSL certificate to my URL, we can ensure that the data is encrypted.

↳ SSL is used to establish an encrypted connection between a web server and a client. SSL ensures a secure data transmission.

→ No one can steal the data if SSL is configured.

→ If SSL is not configured users will not trust our website.

↳ we always need to make sure that our application is SSL configured.

http	https
↓	↓
not secure	secured

→ As it is hard to remember each and every URL.

↓

we will be buying a domain. So, whenever we try to access that domain, that will redirect the access to the URL attached to that domain automatically.

R53 → 53 is the port number for the DNS service.

DNS → Domain name system.

Route 53 → ~~Register domain~~ → Hosted zones → create Hosted zone →

→ Domain name → public Hosted zone → Create Hosted zone

↓

↳ It will create four different records by NS (Name server).

↳ A Name server (NS) record is a type of DNS record that specifies which authoritative name servers are responsible for a domain. These name servers handle DNS queries and direct traffic to the correct locations.

→ we have our domain in Godaddy & we are trying to host in the Route53. So, we need to map them.

↓

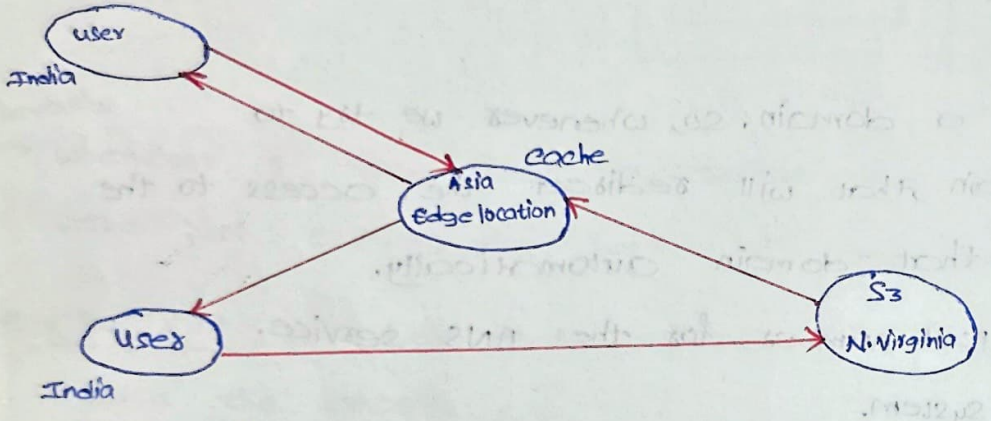
with the help of Name servers.

we need to configure the Name servers in Godaddy domain name.

→ create S3 Bucket with same Domain name.

→ Cloud Front: cloud Front will help us to improve the performance of websites and also it will secure our application.

↳ Edge locations are used in cloud Front.



↳ for the first time, when the user tries to connect to S3 in N. virginia, while responding back, Edge location will store some cache and respond back to the user.

↳ If any other user from India is trying for the same request, instead of connecting to the S3 in N.virginia, it will connect to the cache location (Edge location). Then the Edge location will respond back to the user.

↳ so, the time taken to travel the packets has been reduced.

↳ so, the performance and speed of the website will be improved.

cloudFront → Distributions → create distribution → origin domain → HTTP only → → viewer protocol policy (HTTP and HTTPS) → price class (use N.America, Asia, Middle-East and Africa) → create distribution.

↓

copy the Distribution domain name (run in browser).