Test Axioms for Module 4

* Expect "Single AZ" will never be a right Answer.
* Using AWS managed services should always be preferred
* Fault tolerant & high availability are not the same thing
* Expect that everything will fail at some point & design accordingly

[Fully Managed]

* Amazon S3: (Pay Per Use Policy) (Object Storage) 31/03/2020

We can use it to store & retrieve the data at any time, from anywhere on the web.

What: It is an object storage service that offers industry leading scalability, data availability, security & performance. It is a service offered by AWS that provides object storage through a web service interface. It is storage for the internet.

Storage Classes:

1. S3 standard: General-purpose storage of Frequently accessed data.
2. S3 Intelligent-Tiering: for data with unknown (or) changing access patterns.
3. S3 Standard-Infrequent Access: for less frequently accessed data
4. S3 One Zone-IA: for less frequently accessed data
5. S3 Glacier: &
6. S3 Glacier Deep Archive: for long-term archive & digital preservation.

Description:

1. S3 standard: It offers high availability, durability & performance for frequently accessed data.

Features: Supports SSL for data in transit & encryption of data at Rest.

Bucket can contain upto 5TB of data

2. S3 Intelligent-Tiering Storage class: Designed to optimize costs by automatically moving data to most cost-effective access tier without performance impact (or) operational overhead.
   - Works by storing objects in 2 access tiers.
     1st is optimized for FA.
     2nd is for IA.
3. S3 Standard-Infrequent Access: It is for data that is accessed less frequently, but requires rapid action when need
4. S3 one-zone-IA: It is for data that is accessed less frequ but requires rapid access when needed.
   Unlike other storage classes which store data in a min. amount of 3 AZs, S3 one-zone-IA stores data in single and costs 20% less than S3 standard-IA.
5. S3 Glacier: It is a secure, durable & low-cost storage class for data "archiving".
   - Supports SSL for data in transit & encryption of data at rest

* Because S3 one-zone-IA stores data in a single AWS AZ, data stored in this storage class will be lost in the event of AZ destruction.

* Glacier is imp. for longterm storage & archiving files cheap.

Features:

* Data is stored as objects within resources called "buckets"
& single object can be upto 5 TB in size.
* Objects can be accessed through S3 Access Points / directly through the bucket hostname.
→ Amazon S3 stores data as objects within buckets.
→ An object consists of a file & optionally any metadata that describes the file.

* We can store virtually any kind of data, in any format.
→ Capacity, the volume & no. of objects we can store in S3 is unlimited.
* Bucket is an objec. It consists of data, key & metadata
* Bucket is a logical unit of storage in S3.
4 parameters to chose the optimal region: pricing; user/client locat^n; Latency; Service Availability.
S3 has/supports "Cross Region Replicat^n":

Cloud Front: Caching Service; Data from client side is transferred to nearest edge locat^n & from there it is routed to AWS S3 bucket over an optimized n/w path.
Snowball: Transfer data physically; Avg. TurnAround Time is 5-7 days
Can transfer upto 75 TB
Cost of Snowball: 50TB: 200$ } Fixed.
80TB: 250$ }

* Block Storage can be accessed by only one machine at a time where as Object Storage can be accessed directly by multiple machines.
* S3 is available on Web APIs. S3 is web based storage.

Bucket:
- Used to store objects; contains data & metadata
- It can be configured & created in any specific region.
- When object is added to a bucket, S3 generates unique ID and assigns it to the object.
- Only 100 buckets can be created in each AWS account.

eg: johnsmith -> Bucket
obj: photos/puppy.jpg
URL: http://johnsmith.s3.amazonaws.com/photos/puppy.jpg

- Bucket can be at any region but recommended to have in same region.

Bucket Policy:
- It allows user to authorize policies which either grant/deny access to any no. of accounts & across a range (or) set of keys.
- With this, you can also define security rules that apply to more than one file, including all files (or) a subset of files within a bucket.
- If we have to restrict a user 'Alice' to access studentdata bucket then we can apply security with the help of JSON script & after that user will not be able to access the bucket.

* 3 & more characters should be there
  - number ✓
  - caps (or) - " ✗

max - 63 characters.

Reduced Redundancy Storage (RRS): Cheaper than standard S3
* Stores non-critical, reproducible data at lower levels of redundancy than S3's standard storage.
* High durability & availability 99.99% of objects/yr
  standard - 99.999999999 (11 9's)
* Cost-effective.
* Stores objects in multiple devices across multiple facilities but does not replicate objects.

Cross Region Replication:
* It is a bucket level feature that enables automatic, asynchronous copying of objects across buckets in diff. AWS regions.
* replicas in destination bucket = replicas in source bucket.
AWS never copies data across regions but we can do it

Versioning: Keeping multiple variants of obj in same bucket
It can restore objects from deletion.

1) How can I access files in my S3 bucket through my existing EC2 instance.
A: S3 objects can be accessed using http from anywhere, as long you have permissions.
Access: a) browser: goto url of object
b) AWS CLI: commands on your EC2 to PUT (or) get
c) AWS SDK: use classes & networks from your app to put or get.

* Do you need to write JSON for enforcing policies(or) is there a GUI for managing security within AWS?
A. There is a policy generator tool to generate the JSON
* S3 is cheaper than EC2.
More than 100

Common use Scenarios

1. Backup & Storage
2. Applicat^{n} Hosting
3. Media Hosting
4. Slw Delivery

Cloud Formation:

Why: manage, create & provision all resources at single place.

A complex applicat° on AWS can have many resources and managing all these resources can be a mundane ordinary task.

What: model & setup aws resources

Cloud Formation is a service that helps you model & setup your AWS resources so that you can spend less time managing those resources & more time focusing on your applications that run in AWS.

We can create templates. which is used to create as many copies as we want (any region).

For creating templates we use JSON script.

JSON - JavaScript Object Notat°.

It is an open-standard Format that uses human readable text to transmit data objects consisting of attribute - value pairs.

Structure: Fields

AWS Template Format Version
Descript°
Metadata
Parameters
Mappings (dependencies)
Conditions
Outputs
** Resources: {Name of bucket: Type: } }

AWS cloudFormat° sample templates demonstrate how you can create templates for various cases.

IAM: Identity - Access Management (Asymmetric Encrypt^{n})

Why?

Giving specific roles that they deserve

What?

It is a service from AWS using which you can give permissions to different users who using the same AWS account that we have created.

It is a web service that helps you securely control access to ~~AWS~~ resources for your users. You use IAM to control who can use your AWS resources (authentiat^{n}) and what resources they can use & in what ways (authorizat^{n}).

Components of IAM (4): 1. Users 2. Groups 3. Roles 4. Policies

Using IAM, we can create & manage AWS users, and use permissions to allow & deny their access to AWS resources.

1. Users: ~~we need~~ Users ~~bcz we need~~ are used to give permissions to someone

Any user who wants to access your AWS account has to be added in IAM ~~users~~ & then you can attach policies.

Assigned to people.

2 access: a) programatic access: It enables an Access key ID and secret access key for AWS API, CLI, SDK & other development tools

b) AWS management console Access: Enables a password that allows users to sign-in to AWS management console.

2. Groups: The users created can also be devided among groups, & then the rules & policies that apply on the group, apply on the user level as well.

3. Roles: Are assigned to applicat?

An IAM role is an IAM entity that defines a set of permissions for making AWS service requests.

These are not associated with a specific user/group.

4. Policies: nothing but permisions.

To assign permisions to a user, group, role or resource you can create a policy, which is a document that explicitly lists permissions.

Multi-Factor Authenticat^{n} (MFA): Same as OTP procedure

2 layer of security: password
OTP.