IAM security tools

Credentials report (Account-level)

Access Advisor(user-level)

Assign users to groups and assign permissions to groups

Create a strong password policy

Crate and use roles for giving programmatic access

Role: permissions for making AWS service requests

Policies are for Users, groups, and roles’ permissions.

Budget can be created for account

2 thresholds can be setup for it.

EC2

Renting virtual machines ec2

EBS

ELB

Auto scaling groups

OS, CPU, RAM, storage space and network and security group

Bootstrapping means launching commands when a machine starts

T2.micro gives 750 hours per month

Stop and terminate instance

Ipv4 is going to change every time we start an instance

Compute optimized

Memory optimized (high performance relational/non-relational database

Security groups:

Can reference by IP or by security group

Will control access to ports, IP ranges and controls of inbound and outbound rules

**ELB(EC2 load balancer) and ASG**

*High scalability*

Vertical scalability -> increase the load to be handled, t2.micro to t2.large (increasing the size of your instance) t2.nano 0.5g RAM and 1 vCPU MAX: 12.3TB of RAM and 448 vCPUs

Horizontal scalability -> increase the no of instances (distributed systems) auto scaling group and load balancer

*High availability*

Running the application in 2 AZs, auto scaling group and load balancer supporting multi AZ

*Load balancing*

Load balancers are servers that forward internet to multiple servers (EC2 instances)

Hostname of LB is all that is needed and does regular health checks

*3 kinds of LB*

Classic LB – http, https and tcp (v1 generation)

Application LB – http, https and websocket

Network LB – tcp, tls(secure tcp) and udp

Overall, it is recommended to use the v2 generation LBs

Users can contact LB using http and https

But LB can contact EC2 only using http (different SGs assigned to each of them)

Classic LB

Supports TCP, HTTP and HTTPS

Health checks are TCP or HTTP based

Application LB

LB to multiple HTTP applications across machines (target groups)

LB to multiple applications on the same machine (containers)

Routing based on path in URL, hotname in URL and query string or headers.

ALB is used in docker and amazon ECS

Has a port mapping feature

*TGs*

EC2 instances, ECS tasks, Lambda functions and IP addresses

ALB can route to multiple target groups.

Health checks are at the TG level

The true IP of the client is inserted in the header

*Network LB*

Forward TCP and UDP traffic to your instances

Handle millions of request per second

Less latency

NLB has one statis IP per AZ and supports assigning elastic IP

It doesn’t support HTTP based

LB Stickiness

Same client to forwarded to the same instance behind a LB

This works for ALB and CLB

The cookie used for stickiness has an expiration date you control

Might bring imbalance

Stickiness is at the TG level and not an LB level (a time can be specified)

*Cross zone LB*

Splitting the EC2 instances across present In all AZs

SSL/TLS certificate (traffic bw client and LB)

SSL in flight encryption

TLS is the newer version.

Certificates can be managed using ACM (AWS certificate manager)

HTTPS listener must specify a default certificate.

Clients can use SNI (server name indication) to specify the hostname they reach.

SNI – solves the problem of loading multiple SSL certificates.

Only works for ALB and NLB

Does not work for CLB

*Connection draining*

CLB - connection draining

ALB and NLB – Deregistration delay

Time to complete ‘in flight requests” while the instance is draining (1 - 3600 seconds, avg is 300 secs)

*Auto Scaling groups*

Create and get rid of servers very quickly.

Automatically register instances into LB

Min, actual size and max

Scaling policies

Auto scaling alarms, scale based on CloudWatch alarms.

Done based on 1. Target average CPU usage, 2.Customer metrics etc

To update an ASG, you must provide a new launch configuration.

Scaling policies

1.Target Tracking scaling

Ex: I want the average ASG CPU to stay at around 40%

2.Simple/Step Scaling

Using CloudAlarm

The cooldown perios helps to ensure that your auto scaling group doesn’t launch or terminate additional instances before the previous scaling activity takes effect.

Scaling specific cooldown

**RDS + Aurora + Elasticache – 22/02/2021**

Relational DB service

Postgres

MySQL

MariaDB

Oracle

Microsoft SQL Server

RDS is a managed service – automated provisioning, OS patching, continuous backup and restore to specific timestamp, monitoring dashboards, multi-AZ for disaster recovery, vertical and horizontal scaling.

Backups are automatically enabled in RDS, daily full backup, transaction logs are backed up by RDS every 5 mins, 7 days retention (Can be increased to 35 days)

DB Snapshots – manually triggered, retention if you want.

RDS- storage auto scaling

Maximum storage threshold

Applicable for unknown capacities

Read replicas – scales up reading upto 5 replicas within AZ, cross AZ, cross region – ASYNC

Applications must update the connection string to leverage read replicas.?

Only select keyword can be used.

Cost will be charged for cross-region replicas

RDS Multi AZ(disaster recovery) sync replication

From single AZ to multi AZ, you use sync replication – a snapshot is taken and then sync is established

*RDS Security*

At rest encryption – encrypting master and read replicas with AWS KMS, encryption to be defined at launch time, if the master is not encrypted then the read replicas can’t be encrypted.

In-flight encryption – SSL certificates, for postgres, rds.force\_ssl=1(to make it mandatory)

Can copy a snapshot into an encrypted one.

How to encrypt an un-encrypted RDS db :

Create a snapshot, make the snapshot encrypted, restore the db from from the encrypted snapshot, migrate application to the new db and delete the old db

Network security

RDS should be deployed in a private subnet

RDS security group works by leveraging security groups

Can use IAM

For MySQL and PostgreSQL alone, IAM based authentication can be used – just need a authentication token with a lifetime of 15mins

API call to get Auth token from RDS Service

*Amazon Aurora:*

AWS cloud optimized

Postgres and MySQL are both supported as Aurora DB, Aurora can have 15 RR

Aurora high availability and read scaling

6 copies of your data across 3 AZs, self healing, storage is striped across – one aurora takes writes

Reader endpoints(writer endpoint is of the master) – connection load balancing (connects to all RRs so that endpoint will always be mapped incase of failure/auto scaling

Hands on

Db features –> Serverless – unpredictable workloads

KMS – key management service

Amazon Elasticache – Redis or Memcached

Cache – are in memory dbs with really high performance and low latency, helps reduce off od dbs for read intensive work loads

Cache hit and cache miss

User session store – the application writes the session data into elasticache

If the user hits another instance of our application, the instance retrieves data and the user is already logged in

*Redis* – multi AZ with auto failover and RR to scale reads and have high availability

Backup and restore features

*Memcached* –multi node and multi architecture with no high availability

Caching implementation considerations

Cache- should be consistent, data should be structured

Which cache is best?

1.Lazy loading

|  |  |
| --- | --- |
| Pros | Only req dats is cached, node failures are not fatal |
| cons | Cache miss penalty that results in 3 round trips, stale cache |

2.Write through

|  |  |
| --- | --- |
| Pros | Data is never stale, reads are quick |
| cons | Missing data (could be combined with lazy), Cache chum |

Cache evictions and TTL

Delete explicitly, LRU, TTL

Elasti Cache replication’

Redis (Cluster mode disabled) – upto 5, all nodes present under one shard, multi AZ enabled by default for failover

Redis (cluster mode enabled) – multiple shards , upto 500 nodes per cluster

Due to the failover process, the primary and the standby instances use the same endpoint, RR will have their own DNS names

PostgreSQL does not support TDE(transparent data encryption) on top of KMS.

VPC – virtual private cloud

VPC is AZ specific

Subnets can be there – public and private subnet.

To define access to the internet and between subnets, we use routing tables.

Internet gateway is used for public subnets to talk to www.

NAT gateways are used in public subnets which will be contacted by the private subnet to connect to IGW which in turn will talk to www.

Network ACL

A fire wall which controls traffic from and to subnet, can have allow and deny rules and are attached at the subnet level, the rules only include IP addresses.

security groups (can only deny rules)

a firewall that controls traffic to and from an EC2 instance

2 firewall, first NACL and then security group

VPC flow logs – capture information about IP traffic going into your interfaces.

Would help in monitoring and troubleshooting connectivity issues. Like

Subnets to internet, subnets to subnets and internet to subnets

VPC peering connection.

Must not have overlapping CIDR (IP address range)

VPM peering is not transitive.

VPC endpoints(private)

Endpoints allow you to connect to AWS services using a private network instead of a public www network, which gives enhanced security and lower latency to access AWS services.

VPC endpoint gateway: S3 & Dynamo DB

VPC endpoint interface: the rest

Site to Site VPN & Direct connect:

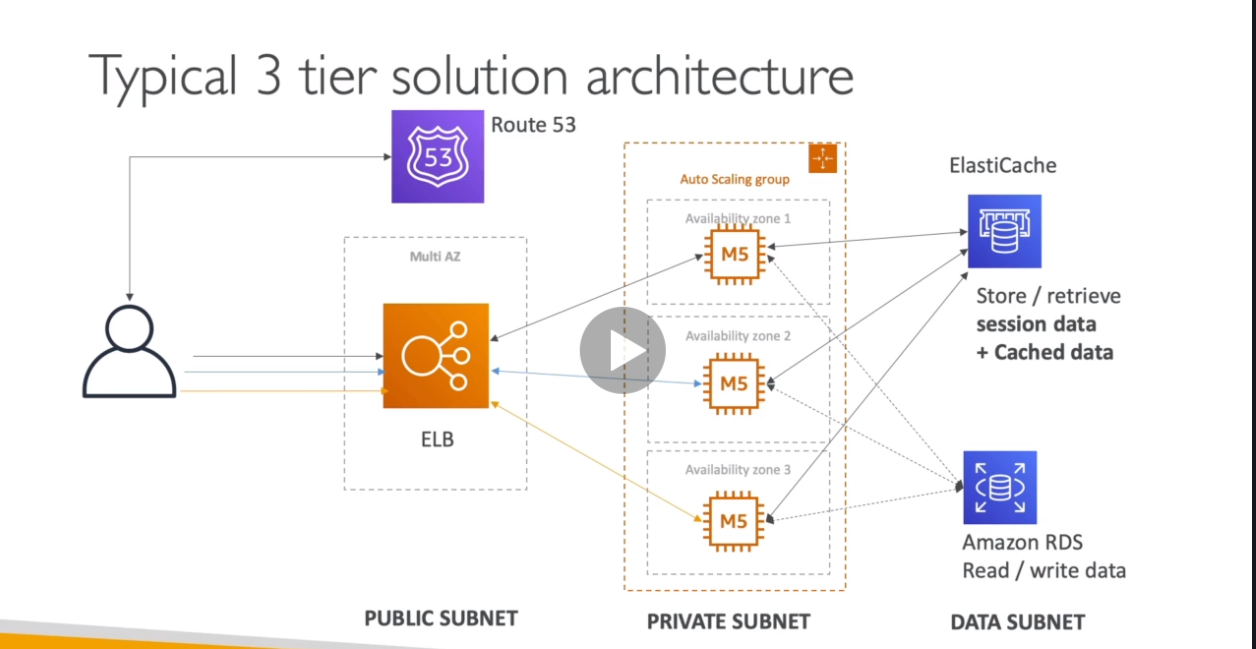
Site to Site VPN

1. The connection is automatically encrypted.
2. Goes over the public internet.

Direct connect.

1. A physical connection
2. Private secure and fast
3. Takes time for setup.

3 tier architectures



LAMP stack on EC2

Linux: OS for EC2 instances

Apache: Web Server that run on Linux.

MySQL: database on RDS

PHP: application logic

Redis/Memcached

EBS drive: to store local application data & software.

WordPress: blogging tool, stored on EFS (elastic file system).

**Advanced S3 & Athena**

S3 MFA Delete:

Forces users to generate a code on a device before doing important operations on S3.

To use MFA, delete, enable versioning on the S3 bucket.

You will need MFA to

1. Permanently delete on object version
2. Suspend versioning on the bucket.

You will not need MFA to

1. Enable versioning.
2. Listing deleted versions.

Only the bucket owner can enable/disable MFA-Delete.

It can be done only through CLI.

Use commands to enable/disable MFA in a particular bucket.

S3 Access Logs:

Any request made into S3 from any account, authorized or denied will be logged into another S3 bucket. Data analysis tools such as Athena could be used to do so.

Warning:

Do not set your logging bucket to be the monitored bucket, else it will create a logging loop and the bucket will grow exponentially.

S3 Replication (CRR & SRR)

Cross region and same region replication

Must enable versioning in source and destination, buckets can be in different accounts, copying in asynchronous and must give proper IAM permissions to S3.

CRR – compliance, lower latency access and replication across accounts.

SRR – Log aggregation, live replication between production and test accounts.

S3 replication:

After activating, only new objects are replicated. (Bucket versioning must able to enabled)

For DELETE operations:

Can replicate delete markers from source to target, deletions with a version ID are not replicated.

There is no chaining of replication, meaning if bucket 1 has replication into bucket 2, which has replication into bucket 3. Then objects created in bucket 1 are not replicated to bucket 3.

Deleted objecs (specific version IDs) cannot be replicated.

When a file is deleted, the deleted file gets replicated but not the delete marker. (Can be selected to behave otherwise).

S3 pre-signed URLs:

Can generate pre-signed URLs using SDK (uploads generally) and CLI (downloads generally).

To generate pre signed URLs, we need to use a CLI command. As the link provided in the bucket page will not work.

S3 Storage classes (6)

1. S3 standard
2. S3 standard – infrequent use
3. S3 one zone – infrequent use (recreate data)
4. S3 intelligent tiering
5. S3 Glacier (Archiving)
6. S3 Glacier deep archive (less needed)

S3 reduced redundancy storage.

*S3 Standard – General purpose (99.99% available)*

High durability of objects across multiple AZ.

Can sustain 2 concurrent facility failure.

Use cases: Big-data analytics, mobile & gaming applications, content distribution.

*S3 Standard – infrequent access (IA)*

Suitable for data that is less frequently accessed but requires rapid access when needed.

Availability is 99.9% across multiple AZ.

Low cost compared to amazon s3 standard. And can sustain 2 concurrent facility failures.

Use cases: as a data store for disaster recovery, backups etc.

*S3 One Zone – Infrequent Access (IA)*

Same as IA but stored in a single AZ, hence data will be lost when AZ is destroyed.

99.5% availability.

Low latency and high throughput performance.

Supports SSL for data at transit and encryption at rest.

Low cost compared to IA (by 20%)

Use Cases: storing secondary backup copies of on-premises data or storing data you can recreate.

*S3 intelligent tiering*

Same low latency and high throughput performance of S3 standard.

Small monthly monitoring and auto-tiering fee.

Automatically moves objects between 2 access tiers based on changing access patterns.

Designed for durability of 99.9…… of objects across multiple AZs.

Resilient against events that impact an entire AZ.

*Amazon Glacier*

Low-cost object storage meant for archiving/backup.

Data is retained for the longer term (10s of years)

Alternative to on-premises magnetic tape storage.

Cost per storage per month is very less.

Each item in glacier is called “Archive” (up to 40TB) and archives are stored in “Vaults”.

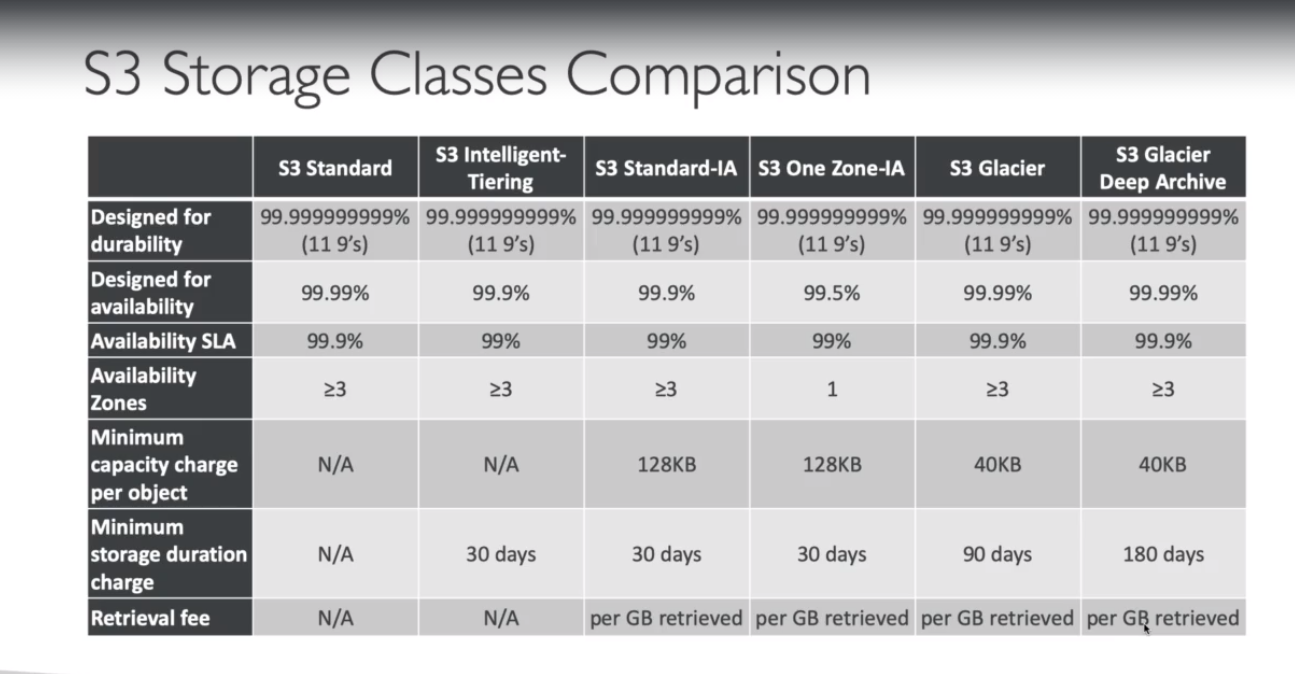
Amazon glacier & glacier deep archive.

Amazon glacier – 3 retrieval options:

1. Expedited (1 to 5 minutes)
2. Standard (3 to 5 hours)
3. Bulk (5 to 12 hours)
4. Minimum storage duration of 90 days

Amazon glacier deep archive – for long term storage – cheaper:

1. Standard (12 hours)
2. Bulk (48 hours)
3. Minimum storage duration of 180 days



*S3 Lifecycle Rules:*

Transition actions: it defines when objects are transitioned to another storage class.

1. Move objects to standard IA class 60 days after creation.
2. Move to glacier for archiving after 6 months.

Expiration actions: configure objects to expire (delete) after some time.

1. Access log files can be set to delete after 365 days.
2. Can be used to delete old versions of files.
3. Can be used to delete incomplete multi-part uploads.

Rules can be created for a certain prefix or certain object tags.

*S3 Lifecycle rules hands on:*

Multiple lifecycle rules can be configured.

*S3 baseline performance:*

Amazon S3 automatically scales to high request rates, latency 100-200 ms.

3500 PUT/COPY/POST/DELETE requests per prefix in a bucket.

5500 GET/HEAD requests per prefix in a bucket.

Example (object path => prefix)

Bucket/folder1/sub1/file => /folder1/sub1 is the prefix.

*S3 KMS Limitation:*

If you use SSE-KMS, you may be impacted by the KMS limits.

When you upload and download, an API is called.

A service quota increase using the service quota console can be done.

*S3 performance:*

Multi part upload:

Recommended for > 100mb.

Must use for > 5 gb.

Can help parallelize uploads into S3 buckets.

S3 transfer acceleration:

Increase transfer speed by transferring file to an AWS edge location using public www and from there use fast private AWS to transfer to destination.

File in USA ->Edge location (USA) ->S3 Bucket Australia.

S3 Performance – S3 byte-range fetches.

To parallelize GETs by requesting specific byte ranges.

Better resilience in case of failures, can be used to speed up downloads. Can be used to retrieve only partial data (for example the head of a file).

*S3 Select & glacier select.*

S3 will do the filtering before sending the information using server-side filtering, which is 400% faster and up to 80% cheaper.

S3 Event notifications:

S3: object created, S3: object removed.

Use case: generate thumbnails of images uploaded to S3.

Can create as many ‘S3 events’ as desired.

SNS, SQS and Lambda function.

If you want to ensure that an event notification is sent for every successful write, you can enable versioning on your bucket.

*AWS Athena*

Serverless service to perform analytics directly against S3 files.

User SQL language to query the files.

Has a JDBC / ODBC language driver, supports CSV< JSON, ORC, Avro, and Parquet.

Use cases: Business intelligence / analytics/ reporting, analyze.

Exam tip: Analyze data directly on S3 => use Athena.

Athena Hands-On: