**AWS Solution Architect Associates**

Exam Tips: -

**Understand the difference between Region, an Availability Zones (AZ) and Edge Locations.**

* A region is physical location in the world which consist of two or more Availability Zones (AZ’s)
* An AZ is one or more discrete Data Centers, each with redundant power, Networking and connectivity, housed in separate facilities.
* Edge locations are Endpoints for AWS which are used for caching content. Typically, this consist of CloudFront, Amazon Content Delivery Network (CDN).

**AWS Support Plans?**

* Basic
* Developer
* Business
* Enterprise

**Identity Access Management: -**

IAM allows you to manage users and their level of access to the AWS console.

**Key Features of IAM: -**

**Identity Access Management offers following feature.**

* Centralized control of your AWS account
* Shared access to your AWS accounts
* Granular Permissions
* Identity Federation (including Active Director, Facebook, LinkedIn etc.)
* Multifactor Authentication
* Provide temporary access for users/device and services where necessary
* Allows you to setup own password rotation policy
* Integrates with many different AWS Services
* Support PCI DSS Compliance

**Key Terminology For IAM: -**

**1. Users: -** End Users such as people, employees of an organization etc.

**2. Groups: -** A collection of users. Each user in the group will inherit the permission of the group.

**3. Policies: -** Policies are made up of documents called Policy documents. These documents are in a format called JSON and they give permission as to what a User/Group/Role can do.

**4. Roles: -** You can create roles and then assign them to AWS Resources.

**Exam Tips: -**

* **IAM is Universal.** It does not apply to region currently.
* The **“Root Account”** is simply the account created when we setup your AWS Account. It has complete Admin Access.
* New User have No Permissions
* New users are assigned Access Key ID & Secret Access Keys when it’s created.
* Access Key ID & Secret Access Keys are not the same as a password. You cannot use the Access Key ID & Secret Access Key to login in to the AWS console. You can use this access via the AWS APIs and Command Line, however.
* Access Key ID & Secret Access Keys you only get to view once. If you lose them, you have regenerate. So, save them in secure location.
* Always Setup a Multifactor Authentication on your root account
* You can create and customize your own password rotation policies.

**Simple Storage Service (S3): -**

**What is S3?**

S3 provides developers and IT teams with secure, durable, Highly scalable object storage. Amazon S3 is easy to use, with a simple web services interface to store and retrieve any amount of data from anywhere on the web.

* S3 is a safe place to store your files
* It is Object-based storage
* The data is spread across multiple devices and facilities.

**The basics of S3 are as follows;**

* S3 is object based – i.e. allows you to upload files.
* Files can be from **0 Bytes to 5 TB.**
* There is unlimited storage.
* Files are stored in Buckets.
* S3 is a universal namespace. That is, name must be unique globally.
* When you upload a file to S3, you will receive a **HTTP 200 code** if the upload was successful.

**How does data consistency work for S3?**

* Read after Write consistency for PUTS of new Objects
* Eventual Consistency for overwrite **PUTS** and **DELETES** (can take some time to propagate)

**In other Words;**

* If you write a new file and read it immediately after words, you will be able to view that data
* If you update AN EXSITING files or delete a file and read it immediately, you may get the older version, or you may not. Basically, changes to objects can take a little bit of time to propagate.

**S3 – Guarantees**

**S3 has the following guarantees from Amazon;**

* Built for **99.99% availability** for the S3 platform.
* Amazon Guarantee **99.9% availability**
* Amazon guarantees **99.99999999999% durability** for S3 information. (Remember 11 x 9s).

**S3 – Features: -**

**S3 has the following features;**

* Tired storage available
* Lifecycle Management
* Versioning
* Encryption
* MFA Delete
* Secure your data using Access Control List and Bucket Policies

**S3 – Storage Classes: -**

1. **S3 Standard**

* **99.99% Availability**
* **99.9999999999% Durability**
* Stored redundantly across multiple devices in multiple facilities
* Designed to sustain loss of 2 facilities concurrently

1. **S3 IA (Infrequently accessed)**

* For data that is accessed less frequently, but requires rapid access when needed
* Lower fee than S3, but you are charged a retrieval fee

1. **S3 One Zone – IA**

* For where you want a lower cost option for infrequent data
* But do not required multiple Availability Zone data resilience

1. **S3 Intelligent Tiering**

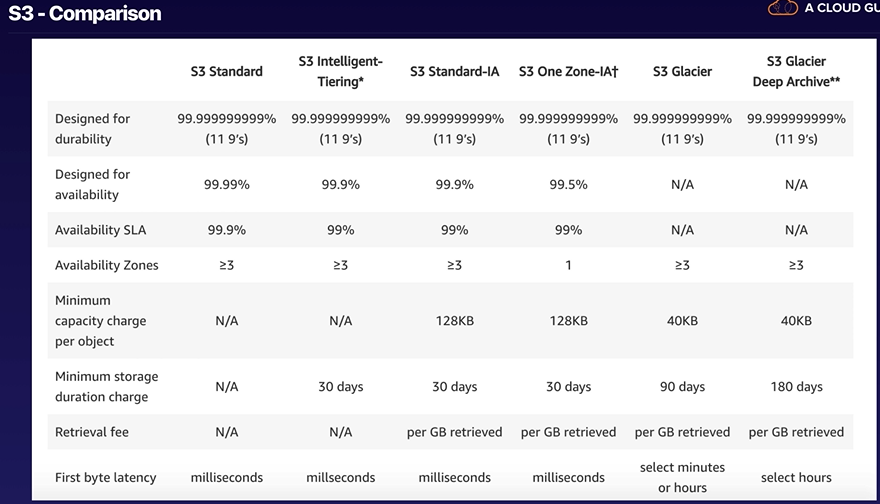
* Designed to optimize cost by automatically moving data to the most cost-effective access tier
* Without performance impact or operational overhead

1. **S3 Glacier**

* S3 Glacier is a Secure, durable and low-cost storage class for data archiving
* You can reliably store any amount of data at costs that are competitive with or cheaper than on-premises solutions.
* **Retrieval Times configurable from Minute to Hours**

1. **S3 Glacier Deep Archive**

* S3 Glacier Deep Archive is Amazon’s S3’s lowest-cost storage class where **a retrieval time of 12 Hours is acceptable**



**S3 Charges**

* Storage
* Request
* Storage Management Pricing
* Data Transfer Pricing
* Transfer Acceleration
* Cross Region Replication Pricing

**S3 Transfer Acceleration**

Amazon S3 Transfer Acceleration enables fast, easy and secure transfer of files over long distance your end user and an S3 bucket.

Transfer Acceleration takes advantage of Amazon CloudFront’s globally distributed edge locations. As the data arrives an edge location, data is routed to Amazon S3 over an optimized network path.

**Exam Tips: -**

* You can control access to buckets using either a Bucket ACL or using the Bucket Policies.

**S3 Security and Encryption: -**

**Encryption in Transit is achieved by**

* SSL/TLS

**Encryption at Rest (Server Side) is achieved by**

* S3 Managed Keys -SSE-S3
* AWS Key Management Service, Managed Keys – SSE-KMS
* Server-Side Encryption with customer provided keys -SSE-C

**Client-Side Encryption**

**S3 Cross Region Replication: -**

**Exam Tips: -**

* Versioning must be enabled on source and destination buckets
* Regions must be unique
* Files in an existing bucket are not replicated automatically.
* All subsequent updated files will be replicated automatically.
* Delete markers are not replicated.
* Deleting individual versions or delete markers will not be replicated.

**AWS CloudFront**

**What is CloudFront?**

A Content Delivery Network (CDN) is a system distributed servers (Network) that deliver webpages and other web content to users based on the geographic locations of the users, the origin of the webpage and a content delivery server.

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services. CloudFront works seamlessly with services including AWS Shield for DDoS mitigation, Amazon S3, Elastic Load Balancing or Amazon EC2 as origins for your applications, and Lambda @Edge to run custom code closer to customers’ users and to customize the user experience. Lastly, if you use AWS origins such as Amazon S3, Amazon EC2 or Elastic Load Balancing, you don’t pay for any data transferred between these services and CloudFront.

**CloudFront Key Terminology: -**

* Edge Location – This is location where content will be cached. This is to separate to an AWS Region/AZ
* Origin – This is the origin of all the files that CDN will distribute. This can be S3 Bucket, an EC2 Instances, an Elastic Load Balancer, or Route 53.
* Distribution – This is the name given the CDN which consists of a collection of Edge locations.

**Types of Distributions: -**

1. Web Distribution – Typically used for Websites
2. RTMP – Used for Media Streaming

**Exam Tips: -**

* Edge Location are not just READ Only – You can write to them too (i.e. Put an object on to them).
* Objects are cashed for the life of the TTL (Time to Live).
* You can clear cashed objects, but you will be charged.

**AWS EC2 (Elastic Compute Cloud): -**

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the Cloud. Amazon EC2 reduce the required time to obtain and boot a new server instances within minutes, allowing you to quickly scale the capacity, both up and down, as your compute requirements change.

**EC2 Pricing Model: -**

* **On Demand: -** Allows you to pay fixed rate by the hour (or by the second) with no commitment.
* **Reserved: -** Provides you with a capacity reservation and offer a significant discount on the hourly charge for an instance. Contract Terms are 1 Year to 3 Year Terms.
* **Spot Instances: -** Enables you to bid whatever price you want for instance capacity, providing for even grater savings if your applications have flexible start and end times.
* **Dedicated Host: -** Physical EC2 server dedicated for your use. Dedicated hosts can help you to reduce cost by allowing you to use your existing server-bound software licenses.

**Reserved Instance Pricing Model: -**

* **Standard Reserved Instances: -** These offers up to 75% off on Demand Instances. The more you pay up front and the longer the contract, the grater the discount.
* **Convertible Reserved Instances:** - These offers up to 54% off on demand capacity to change the attributes of the RI as long as the exchange results in the creation of Reserved Instances of equal or grater value.
* **Scheduled Reserved Instances: -** These are available to launch within the time window you reserve. This option allows you to match your capacity reservation to predictable recurring schedule that only requires a fraction of day, a week or a month.

**Spot Instances Pricing Model: -**

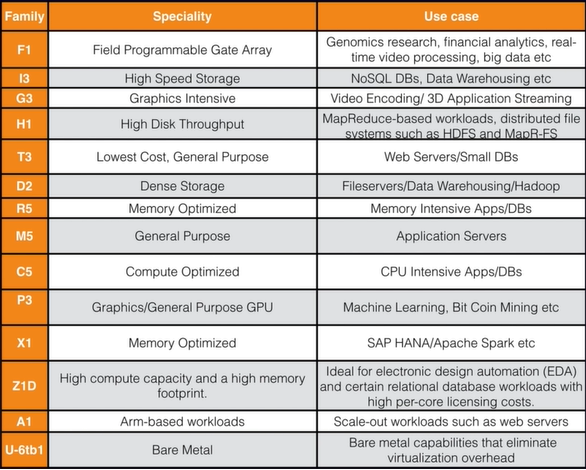
* Application that have flexible start and end times.
* Applications that are only feasible at very low compute price.
* Users with urgent computing needs for large amount of additional capacity.

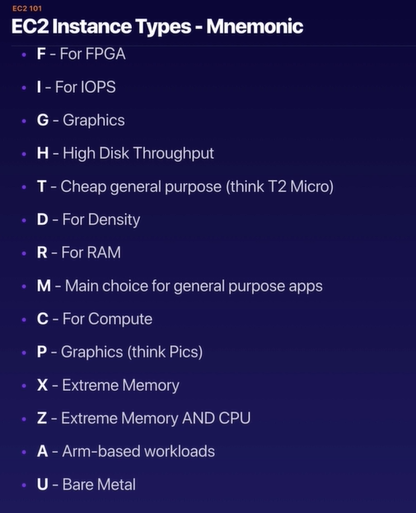
**Dedicated Host Pricing: -**

**Useful for;**

* Useful for regularity requirements that may not support multi-tenant virtualization.
* Great for licensing which does not support multi-tenancy or cloud deployments.
* Can be purchased on Demand (hourly).
* Can be purchased as reservation for up to 70% off the demand price.

**EC2 Instance Types: -**

****



**Exam Tips: -**

* If the spot instance is terminated by Amazon EC2, you will be not charged for partial hour of usage. However, If you terminate the instance yourself, you will be charged for any hour in which the instance ran.
* Termination protection is **turned off by default**, you must turn it on.
* On EBS-backed Instances **the default action is for the root EBS volume to be delated** when the instance is terminated.
* EBS root volume of your default AMI’s **can** be encrypted. You can also use third party tolls (such as Bit Locker etc.) to encrypt the root volume, or this can be done when creating AMI’s in AWS console or using the API’s.
* Additional volumes can be encrypted.

**AWS Security Groups: -**

**Exam Tips: -**

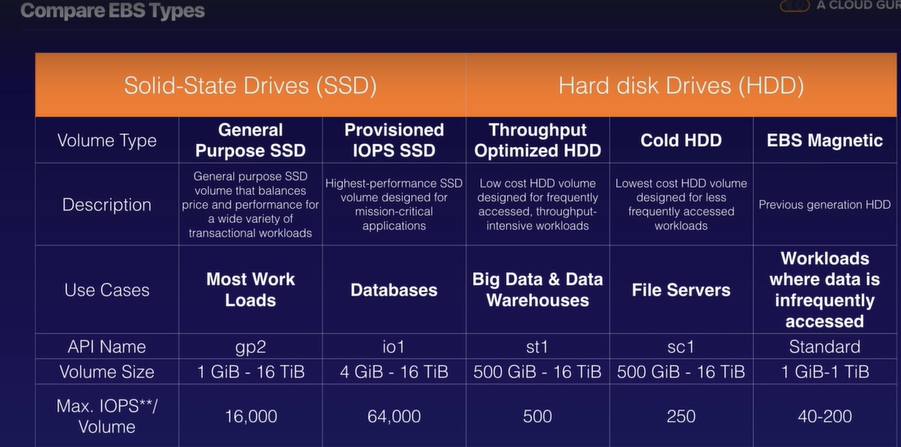
* All Inbound traffic is blocked by default
* All Outbound traffic is allowed.
* Changes to Security Groups take effect immediately
* You can have any number of EC2 instances within a Security Group.
* You can have multiple Security Groups attached to EC2 Instances.
* Security Groups are STATEFUL
* If you create an Inbound rule allowing traffic in, that traffic is automatically allowed back out again.
* You cannot block specific IP addresses using the Security Group, instead use Network Access Control Lists.
* You can Specify allow rules, but not deny rules.

**Amazon Elastic Block Store (EBS): -**

Amazon Elastic Block Store (EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within it’s Availability Zone to protect you from component failure, offering high availability and durability.

**Types of Amazon Elastic Block Store (EBS): -**

* General Purpose SSD
* Provisioned IOPS SSD
* Throughput Optimized HDD
* Cold HDD
* EBS Magnetic



**Exam Tips: -**

* Volumes Exits on EBS. Think of EBS as Virtual Hard Disk.
* Snapshots exits on S3. Think of Snapshots as a photograph of the disk.
* Snapshots are point in time copies of Volumes.
* **Snapshots are incremental –** This means that only the block that has changed since your snapshot moved to S3 Bucket.
* If this is your Frist Snapshot, it may take some time to create.
* To create snapshot for Amazon EBS Volume that serve as a root device, you should stop the EC2 instance before taking the snapshot.
* However, you can take snapshot while the instance is running.
* You can create AMI’s from both Volumes and Snapshots.
* You can change the EBS Volume size on the fly, including changing the Size and Storage type.
* Volume will be always in the same Availability Zone as the EC2 Instance.
* To move an EC2 Volumes from one AZ to another, take a snapshot of it, create an AMI from the Snapshot and then use the AMI to launch the EC2 instance in a New AZ
* To move an EC2 Volume form one region to another region, take a snapshot of it, Create AMI from the Snapshot and then copy AMI from one region to another region. Then use copied AMI to launch the New EC2 Instance in New region.

**AWS AMI’s Types: -**

* **EBS Volumes: -** The root Device from an Instance launched from the AMI is an Amazon EBS Volume created from an EBS snapshot

* **Instance Store Volumes: -** The root device from an instance launched from the AMI is an instance store volume crested from a template stored in S3.

**Exam Tips: -**

* Instance store Volumes are sometimes called as Ephemeral Storage
* Instance store Volume can not be stopped. If the underlaying host fails, you will lose your data.
* EBS backed instance can be stopped. You will not lose the data on this instance if it is stopped.
* You can reboot both, you will not lose your data.
* By default, both ROOT volumes will be deleted on termination. However, with EBS volumes, you can tell AWS to keep the root device volumes.

**Encrypted and Unencrypted Volumes: -**

**Exam Tips: -**

* Snapshots of encrypted volumes are encrypted automatically
* Volumes restored from encrypted snapshots are encrypted automatically
* You can share snapshots, but if only they are unencrypted
* These snapshots can be shared with other AWS Accounts or made Public.
* You can now encrypt root device volume upon creation of the EC2 instance.

**How to encrypt the Unencrypted root device Volumes?**

**Steps: -**

1. Create the snapshot of the unencrypted root device volume
2. Create the copy of the Snapshot and select the encrypt option
3. Create an AMI from the encrypted snapshot
4. Use the AMI to launch new encrypted instance.

**What is AWS CloudWatch?**

**Amazon CloudWatch is monitoring service to monitor your AWS resources, as well as the applications that you run on AWS.**

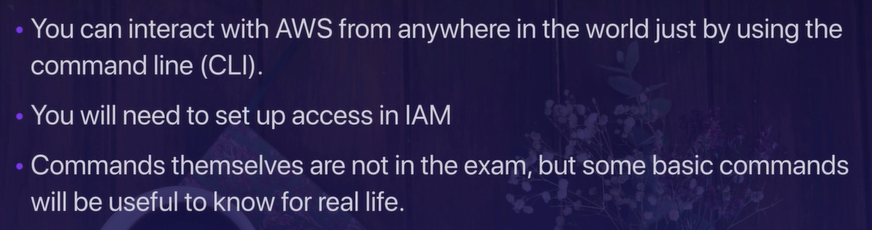
**Exam Tips: -**

* CloudWatch is used for monitoring performance
* CloudWatch can monitor most of AWS as well as your applications that rum on AWS
* CloudWatch with EC2 will monitor the events every 5 minutes by default
* You can have 1-minute intervals by turning on detailed monitoring.
* You can create CloudWatch alarms which trigger notifications.
* CloudWatch is all about performance. CloudTrail is all about auditing.
* **Standard Monitoring = 5 minutes**
* **Detailed Monitoring = 1 minutes**

**What can do with CloudWatch?**

* **Dashboards –** Creates awesome dashboards to see what is happening with your AWS environment.
* **Alarms** – Allows you to set Alarms that notify you when particular thresholds are hit
* **Events –** CloudWatch Events helps you to respond to state changes in your AWS resources
* **Logs** – CloudWatch Logs help you to aggregate, Monitor and store logs
* **CloudWatch Monitor performance**
* **CloudTrail monitors API calls in the AWS platform.**

**AWS CLI: -**



**AWS IAM Roles: -**

**Exam Tips: -**

* Roles are more secure than storing your access kay and secrete access key on individual EC2 instances.
* Roles are easier to Manage
* Roles can be assigned to EC2 instances after it is created using both the console and command line.
* Roles are universal – You can use them in any region.

**AWS Meta-Data**

* **Used to get the information about an instance (such as public IP)**
* **Curl** [**http://169.254.169.154/latest/meta-data**](http://169.254.169.154/latest/meta-data)
* **Curl** [**http://169.254.169.154/latest/user-data**](http://169.254.169.154/latest/user-data)

**AWS Elastic File System (EFS): -**

Amazon Elastic File System (Amazon EFS) is a file storage service for Amazon Elastic Compute Cloud (Amazon EC2) instances. Amazon EFS is easy to use and provides a simple interface that allows you to create and configure file systems quickly and easily. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.

**Exam Tips: -**

* Supports the Network File System version 4 (NFSv4) protocol
* You only pay for the storage you use (no pre-provisioning required).
* Can scale up to the petabytes
* Can support thousands of concurrent NFS connections
* Data is stored across multiple AZ’s within a region.
* Read after Write consistency.

**EC2 Placement Groups: -**

**Types of Placement Groups**

1. **Clustered Placement Group: -** A cluster placement group is a grouping of instances within a single Availability Zone. Placement groups are recommended for applications that need low network latency, high network throughput, or both.

Only certain instances can be launched in to a clustered placement group.

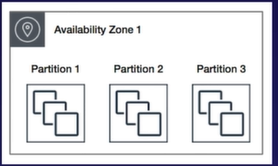
1. **Spread Placement Groups: -** Spread Placement Group is a group of instances that are each placed on distinct underlying hardware.

Spread placement groups are recommended for applications that have small number of critical instances that should be kept separate from each other.

**THINK INDIVIDUAL INSTANCES**

1. **Partitioned Placement Groups: -** When using the partitioned placement groups, Amazon EC2 divides each group in to logical segment called partition. Amazon EC2 ensures that each partition within a placement group has its own set of rack. Each rack has its own Network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure with your application.

**THINK MULTIPLE INSANCES**

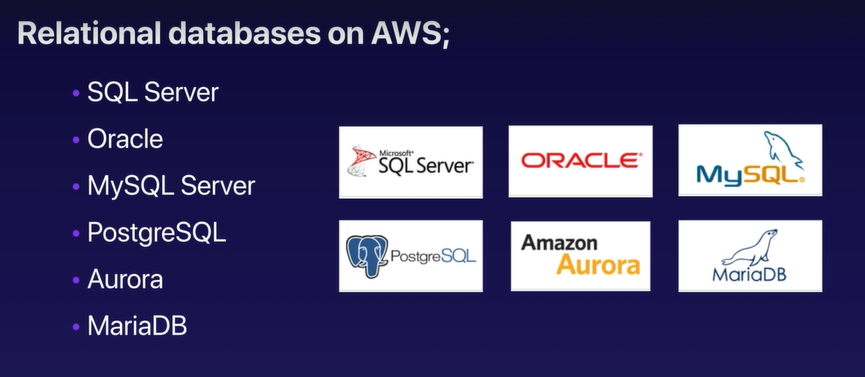


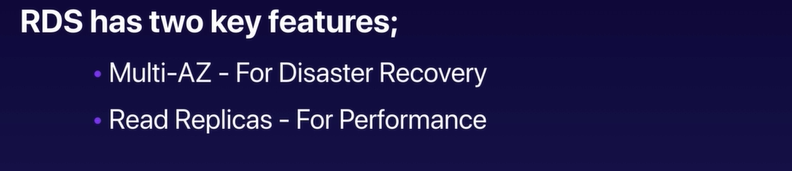
**Exam Tips: -**

**Types of Placement Groups: -**

* **Clustered Placement Group: -**
  + **Low Network Latency / High Network Throughput**
* **Spread Placement Group: -** 
  + **Individual critical EC2 Instances**
* **Partitioned: -**
  + **Multiple EC2 instances HDFS, HBase and Cassandra**
* A clustered placement group cannot span multiple Availability Zones.
* A spread placement and partitioned group can span in to multiple AZ’s.
* The name you specify for a placement group must be unique within your AWS account.
* Only certain types of instances can be launched in a placement group (Compute Optimized, GPU, Memory Optimized, Storage Optimized)
* AWS recommend homogeneous instances within clustered placement groups.
* You can’t merge placement groups.
* You can’t move an existing instance in to a placement group. You can create an AMI form your exiting instance, and then launch the new instance from the AMI into a placement group.

**AWS Database Services: -**





**AWS RDS Backups: -**

**Automated Backups: -**

* Automated Backups allow you to recover your database to any point in time within a “retention period”.
* The retention period can be between 1 to 35 days.
* Automated backup will take a full daily snapshot and will also store transaction logs throughout the day.
* When you do recovery, AWS will first choose the most recent daily backup and then apply transaction logs relevant to that day.
* This allows you to do a point in time recovery down to a second, within the retention period.

**Database Snapshot: -**

* DB snapshot are done manually (they are user initiated)
* They are stored even after you deleted the original RDS Instance, unlike automated backups.

**Restoring Backups: -**

Whenever you restore either an Automatic or Manual Snapshot, the restored version of the database will be a new RDS with a new DNS endpoint.

**Encryption at REST: -**

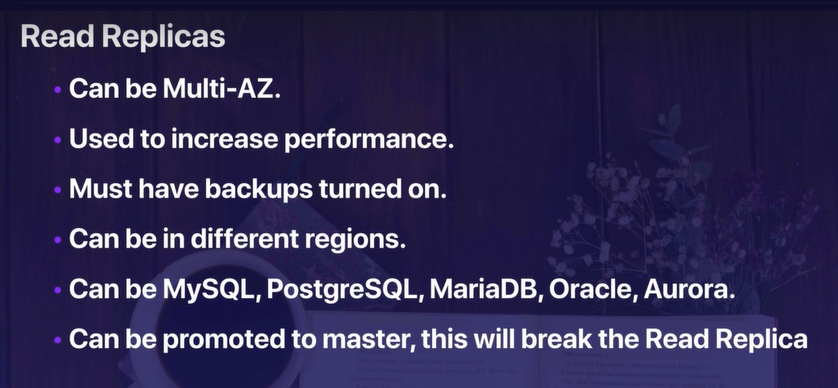
* Encryption at REST is supported for MySQL, Oracle, SQL Server, PostgreSQL, MariaDB and Aurora.
* Encryption is done using the AWS Key Management Service (KMS)
* Once your RDS instances is encrypted, the data stored at rest in the underlaying storage is encrypted, as are its automated backups, read replicas and snapshots.

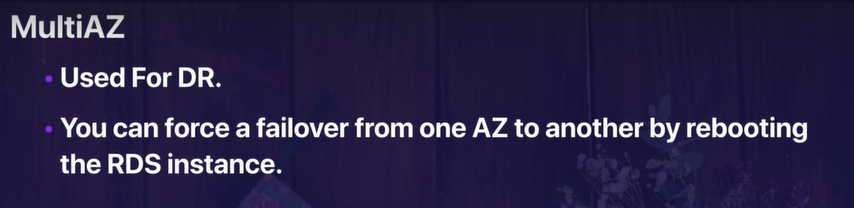
**What is Multi AZ: -**

Multi AZ allows you to have an exact copy of your production database in another Availability Zone. AWS handles the replication for you, so when your production database is written to, this write will automatically be synchronized to the stand by database.

**In the event of planned database maintenance, DB instance failure, or an Availability Zone failure, Amazon RDS will automatically failover to the standby so that operations can resume quickly without administrative intervention.**

**Exam Tips: -**





**What is AWS DynamoDB?**

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale.

It is fully managed database and supports both document and key-value data models.

It is flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad-Tech, IoT, and many other applications.

**The Basic of DynamoDB is: -**

* Stored on SSD storage
* Spread across 3 geographically distinct data centers
* Eventual Consistent Reads (Default)
* Strongly Consistent Reads

**Eventual Consistent Read: -**

Consistency across all copies of data is usually reached within a second. Repeating a read after a short time should return updated data. (Best Read Performance)

**Strongly Consistent Reads: -**

A Strongly Consistent Reads returns a result that reflects all writes that received a successful response prior to the read.

**AWS Redshift: -**

Amazon Redshift is a fast and powerful, fully managed, petabyte-scale data warehouse service in the cloud.

Customers can start small for just $0.25 per hour with no commitments or upfront costs and scale to petabyte or more for $1,000 per terabyte per year, less than a tenth of most other data warehousing solutions.

**Redshift can be configured as follows: -**

* **Single Node (160 GB)**
* **Multi Node** 
  + **Leader Node (manages client connections and receives queries)**
  + **Compute Node (store data and perform queries and computations). Up to 128 Compute Nodes.**

**Advanced Compression: -**

Columnar data stores can be compressed much more than row-based data stores because similar data is stored sequentially on disk.

Amazon Redshift employs multiple compression techniques and can often achieve significant compression relative to traditional relational data stores.

In addition, Amazon Redshift doesn’t require indexes or materialized views, and so uses less space than traditional relational database systems.

When loading data into an empty table, Amazon Redshift automatically samples your data and select the most appropriate compression scheme.

**Massively Parallel Processing (MPP): -**

Amazon Redshift automatically distribute data and query load across all nodes. Amazon redshift makes it easy to add nodes to your data warehouse and enables you to maintain fast query performance as your data warehouse grows.

**Redshift Backups: -**

* Enabled by default with a 1-day retention period.
* Maximum retention period is 35 days
* Redshift always attempts to maintain at least three copies of your data (the original and replica on the compute nodes and a backup in amazon S3).
* Redshift can also asynchronously replicate your snapshots to S3 in another region for disaster recovery.

**Redshift is priced as follows:**

* Compute node Hours (total number of hours you run across all your compute nodes for the billing period. You are billed for 1 unit per node per hour, so a 3-node data-warehouse cluster running persistently for an entire month would incur 2,160 instance hours. You will not be charged for leader node hours; only compute nodes will incur charges.)
* Backup
* Data Transfer (Only within a VPC, not outside it)

**Security Considerations:**

* Encrypted in transit using SSL
* Encrypted at REST using AES-256 encryption
* By default, RedShift takes care of key management.
  + Manage your own keys through HSM
  + AWS Key Management Service

**Redshift Availability: -**

* Currently only available in 1 AZ
* Can restore snapshots to new AZs in the event of an outage

**Amazon Aurora: -**

Amazon Aurora is a MySQL-compatible, relational database engine that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open source databases.

Amazon Aurora provides up to five times better performance than MySQL at a price point one tenth that of a commercial database while delivering similar performance and availability.

**Things Know About Aurora: -**

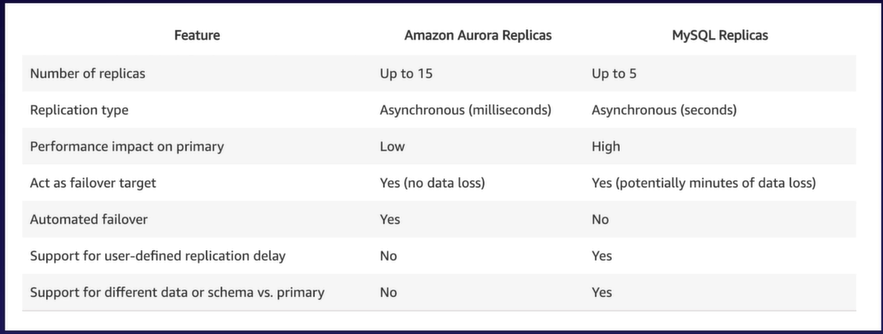
* Start with 10GB, Scale in 10GB increments to 64TB (Storage Autoscaling)
* Compute resources can scale up to 32vCPUs and 244GB of Memory.
* 2 copies of your data are contained in each availability zone, with minimum of 3 availability zones. 6 copies of your data.

**Scaling Aurora: -**

* Aurora is designed to transparently handle the loss of up to two copies of data without affecting database write availability and up to three copies without affecting read availability.
* Aurora storage is also self-healing. Data blocks and disk are continuously scanned for errors and repaired automatically.

**Two Types of Aurora Replicas are available: -**

* **Aurora Replicas (currently 15)**
* **MySQL Read Replicas (currently 5)**



**Backups wit Aurora: -**

* Automated backups are always enabled on Amazon Aurora DB Instances. Backups do not impact database performance.
* You can also take snapshots with Aurora. This also does not impact om performance.
* You can share Aurora Snapshots with other AWS accounts.

**Exam Tips: -**

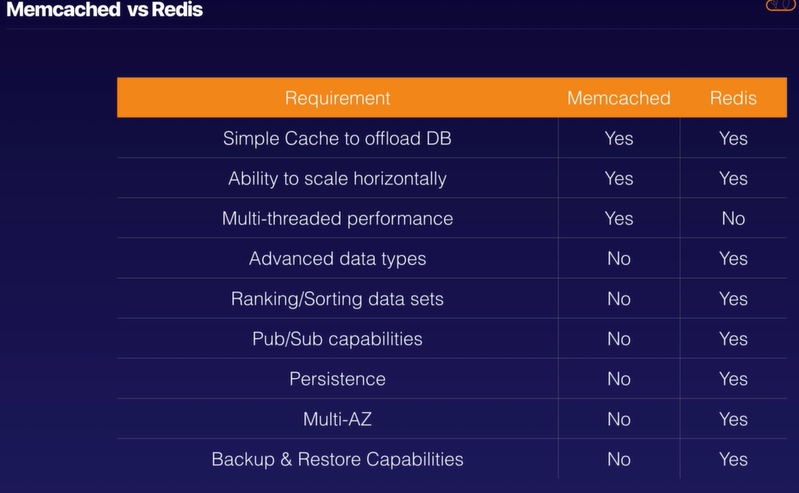
* 2 copies of your data is contained in each availability zone, with minimum of 3 availability zones. 6 copies of your data.
* You can share Aurora Snapshots with other AWS accounts.
* 2 types of replicas available. Aurora Replicas and MySQL Replicas. Automated failover is only available with Aurora Replicas.
* Aurora has automated backups turned on by default. You can also take Snapshots with Aurora. You can share these snapshots with other AWS accounts.

**AWS ElastiCache: -**

ElastiCache is a web service that makes it easy to deploy, operate and scale an in-memory cache in the cloud. The service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.

**ElastiCache support two open-source in-memory caching engines:**

* **Memcached**
* **Redis**



**Exam Tips: -**

* Use ElastiCache to increase database and web application performance.
* Redis is Multi-AZ
* You can do backups and restores of Redis.

**AWS Load Balancers: -**

* Application Load Balancer
* Network Load Balancer
* Classic Load Balancer

**Application Load Balancer: -**

Application Load Balancers are best suited for load balancing of HTTP and HTTPS traffic. They operate at Layer 7 and are application-aware. They are intelligent, and you can create advanced request routing, sending specified requests to specific web servers.

**Network Load Balancers: -**

N/W Load Balancers are best suited for load balancing of TCP traffic where extreme performance is required. Operating at the connection level (Layer 4), Network Load Balancer are capable of handling millions of requests per second, while maintaining ultra-low latencies.

**Use for Extreme Performance!!**

**Classic Load Balancers: -**

Classic Load Balancers are the legacy Elastic Load Balancers. You can load balance HTTP/HTTPS applications and use Layer 7 -specific features, such as X-Forwarded and sticky sessions. You can also use strict Layer 4 load balancing for applications that rely purely on the TCP protocol.

If your applications stop responding, the ELB (Classic Load Balancer) responds with a 504 error.

This means that the application is having issues. This could be either at the Web Server layer or at the Database Layer.

Identify where the application is failing, and scale it up or out where possible.

**Exam Tips: -**

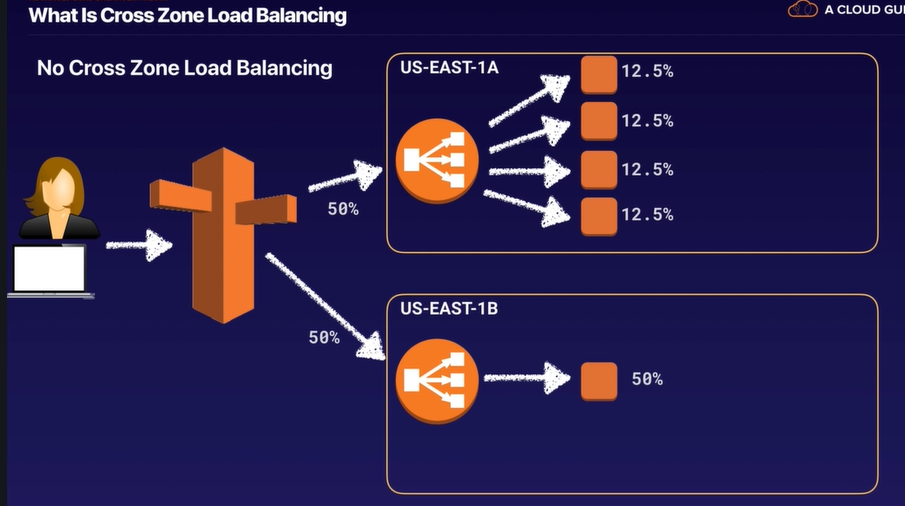
* 504Error means the gateway has timed out. This means that the application not responding within the idle timeout period.
* Trouble shoot the application. Is it the Web Server or Database Server?
* If you need the IPv4 address of your end user, look for the X-Forwarded-For Header.
* Instances monitored by ELB are reported as; InService, or OutofService
* Health Checks check the instance healthy by talking to it.
* Load Balances have their own DNS name. you are never given and IP address.

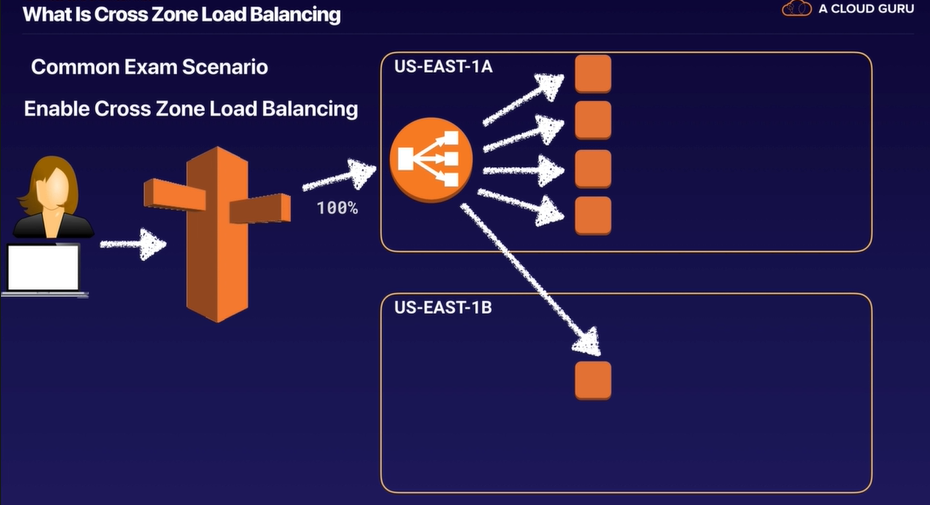
**What are Sticky Sessions?**

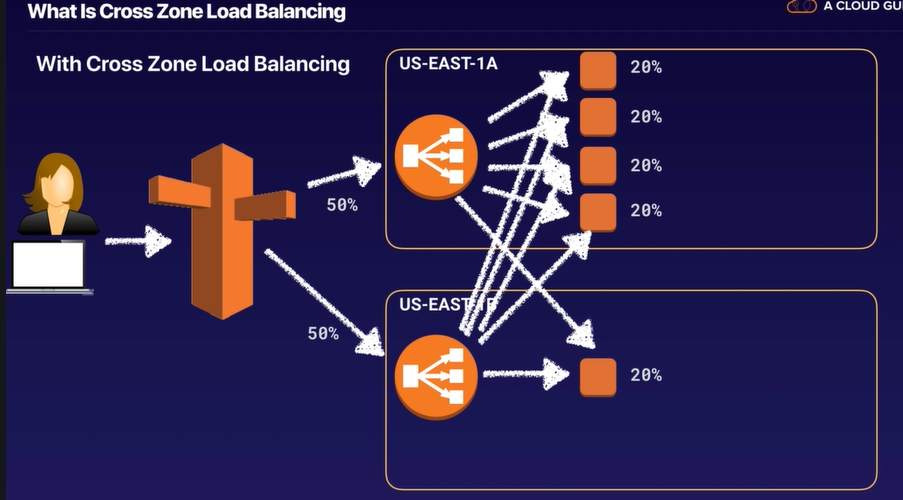
Classic Load Balancer routes each request independently to the registered EC2 instance with the smallest load.

Sticky sessions allow you to bind a user’s session to a specific EC2 instance. This ensures that all requests from the user during the session are sent to the same instance.

You can enable Sticky Sessions for Application Load Balancers as well, but the traffic will be sent at the Target Group Level.







**What are Path Patterns?**

You can create a listener with rules to forward requests based on the URL path. This is known as path-based routing. If you are running microservices, you can route traffic to multiple back-end services using the path-based routing. For Example, you can route general requests to one target group and requests to render images to another target group.

**Exam Tips: -**

* Sticky Sessions enable your users to stick to the same EC2 instance. Can be useful if you are storing information locally to that instance.
* Cross Zone Load Balancing enables you to load balance across multiple availability zones.
* Path patterns allow you to direct traffic to different EC2 instances based on the URL contained in the request.

**AWS Route 53 (AWS DNS Service): -**

If you’ve used the internet, you’ve used DNS. DNS is used to convert human friendly domain names (such as <http://devops.cloud>) into an Internet Protocol (IP) address (such as <http://82.34.124.2>).

IP addresses are used by computer to identify each other on the network. IP addresses commonly come in 2 different forms, IPv4 and IPv6.

**DNS Records: -**

1. **Start of Authority (SOA): - The SOA record stores information about:**
   * **The name of the server that supplied the data for the Zone**
   * **The administrator of the Zone**
   * **The current version of Data File**
   * **The default number of seconds for the time-to-live file on resource records.**
2. **Name Server Records (NS Records): -**

They are used by Top Level Domain servers to direct traffic to the Content DNS Server which contains the authoritative DNS records.

What’s an A Record: - An “A” record is the fundamental type of DNS record. The “A” in A record stands for “Address”. The A record is used by a computer to translate the name of the domain to an IP address. **For example,** [**http://www.devops.cloud**](http://www.devops.cloud) **might point to http://123.10.10.80**

**What’s an TTL?**

The length that a DNS record is cached on either the Resolving Server or the users own local PC is equal to the value of the “Time To Live” (TTL) in seconds. The lower the time to live, the faster changes to DNS records take to propagate throughout the internet.

**What is CNAME?**

A Canonical Name (CNAME) can be used to resolve one domain name to another.

**Alias Records: -**

Alias records are used to map resource record sets in your hosted zone to Elastic Load Balancers, CloudFront distributions, or S3 buckets that are configured as websites.

Alias records work like a CNAME record in that you can map one DNS name to another ‘target’ DNS name.

**Routing Policies: -**

**The following Routing Policies are Available with Route53:**

* **Simple Routing**
* **Weighted Routing**
* **Latency-based Routing**
* **Failover Routing**
* **Geo-proximity Routing (Traffic Flow Only)**
* **Multi-value Answer Routing**

**Simple Routing Policy: -**

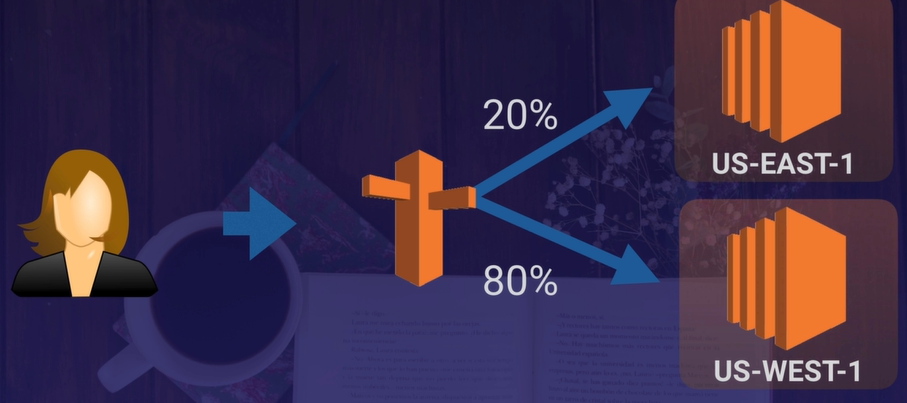
If you choose the simple routing policy, you can only have one record with multiple IP addresses. If you specify multiple values in a record, Route 53 returns all values to the user in a random order.



**Weighted Routing Policy: -**

Allows you to split your traffic based on different weights assigned.

For example, you can set 10% of your traffic to go to US-EAST-1 and 90% to go to EU-WEST-1.



**DNS Health Checks: -**

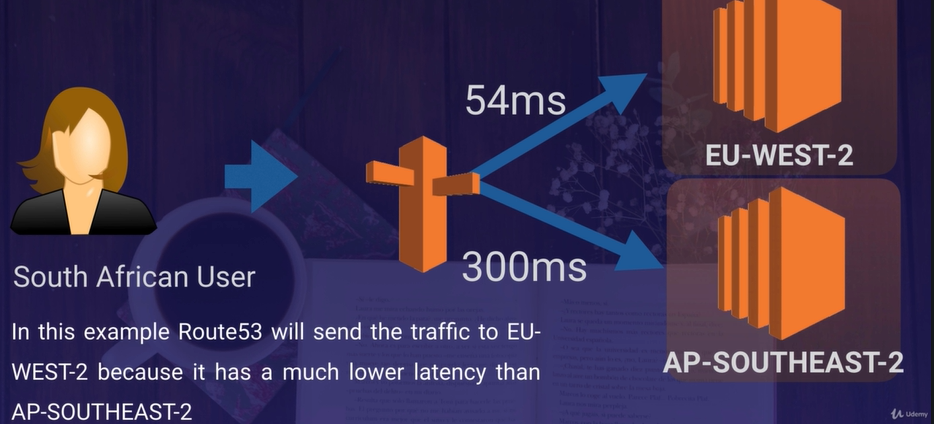
* You can set health checks on individual record sets.
* If a record set fails a health check it will be removed from Route53 until it passes the health check
* You can set SNS notifications to alert you if a health check is failed

**Latency-Based Routing: -**

Allows you to route your traffic based on the lowest network latency for you end user (i.e. which region will give them the fastest response time).

To use latency-based routing, you can create a latency resource record set for the Amazon EC2 (or ELB) resource in each region that hosts your website.

When Amazon Route 53 receives a query for your site, it selects the latency resource records set for the region that gives the user the lowest latency. Route 53 then responds with the value associated with that resources record set.

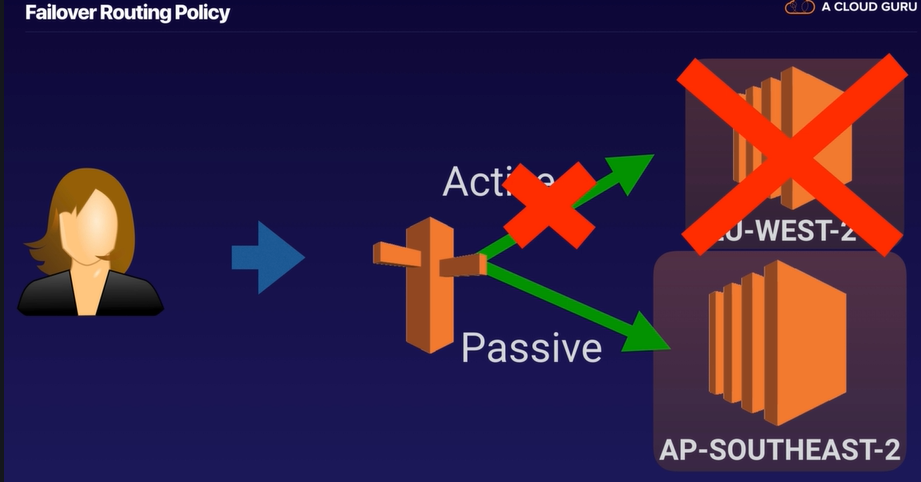


**Failover Routing Policy: -**

Failover routing policies are used when you want to create an active/passive set up. For example, you may want your primary site to be in EU-WEST-2 and your secondary DR site in AP-SOUTHEAST-2

Route53 will monitor the health of your primary site using a health check.

A health check monitors the health of your end points.



**Geolocation Routing Policy: -**

Geolocation routing lets you choose where your traffic will be sent based on the geographic location of your users (i.e. the location from which DNS queries originate).

For example, you might want all queries from Europe to be routed to a fleet of EC2 instances that are specifically configured for your European customers. These servers may have the local language of your European customers and all prices are displayed in Euros.



**Geo-proximity Routing (Traffic Flow Only): -**

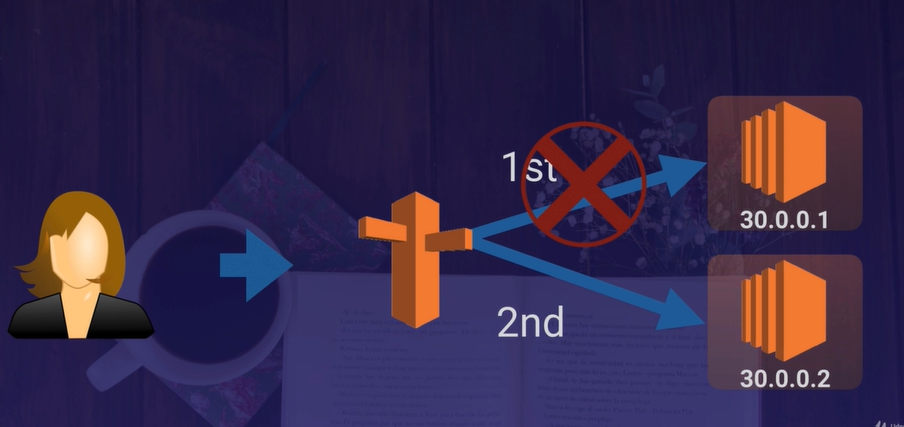
Geo-proximity routing lets Amazon Route 53 route traffic to your resources based on the geographic location of your users and your resources. You can also optionally choose to route more traffic or less to a given resource by specifying a value, known as a bias. A bias expands or shrinks the size of the geographic region from which traffic is routed to a resource.

**To use geo-proximity routing, you must use Route 53 traffic flow.**

**Multi-value Answer Routing Policy: -**

Multi-value answer routing lets you configure Amazon Route 53 to return multiple values, such as IP addresses for your web servers, in response to DNS queries. You can specify multiple values for almost any record, but multi-value answer routing also lets you check the health of each resource, so Route 53 returns only values for healthy resources.

**This is similar to simple routing however it allows you to put health checks on each record set.**

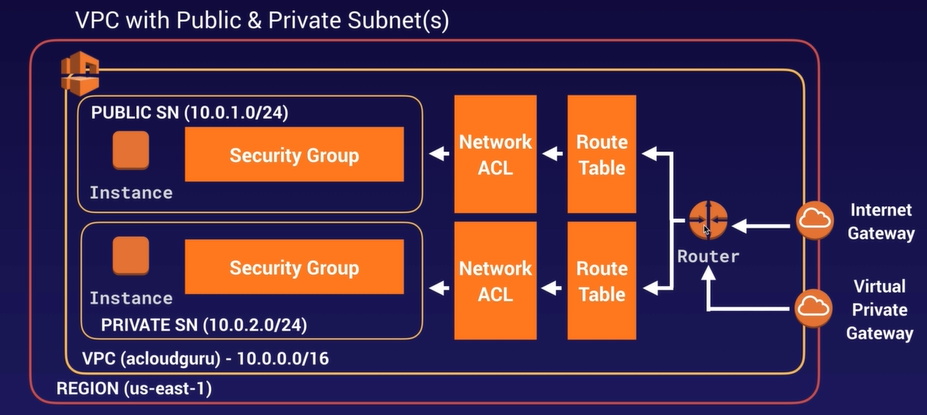


**AWS Virtual Public Cloud (VPC): -**

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the Amazon Web Services (AWS) Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

You can easily customize the network configuration for your Amazon Virtual Private Cloud. For example, you can create a public-facing subnet for your webservers that has access to the Internet and place your backend systems such as databases or application server in private-facing subnet with no internet access. You can leverage multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet.

Additionally, you can create a Hardware Virtual Private Network (VPN) connection between your corporate datacenter and your VPC and leverage the AWS cloud as an extension of your corporate datacenter.



**What can we do with a VPC?**

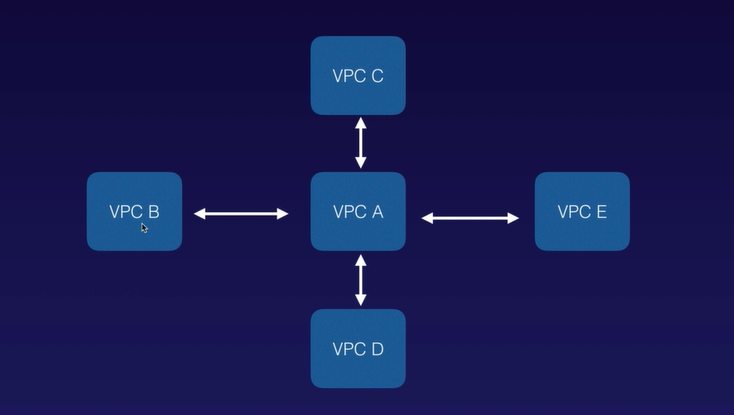
* Launch instances into a subnet of your choosing
* Assign custom IP address ranges in each subnet
* Configure route tables between subnets
* Create internet gateway and attach it to our VPC
* Much better security control over your AWS resources
* Instance security groups
* Subnet network access control list (ACLS)

**Default VPC vs Custom VPC: -**

* Default VPC is user friendly, allowing you to immediately deploy instances.
* All subnets in default VPC have a route out to the internet
* Each EC2 instance has both a public and private IP address.

**VPC Peering :-**

* Allows you to connect one VPC with another via a direct network route using private IP addresses.
* Instances behave as if they were on the same private
* You can peer VPC’s with other AWS accounts as well as with other VPC’s in the same account.
* Perring is in a star configuration: i.e. 1 central VPC peers with 4 others. **NO TRANSITIVE PEERING!!!!**
* You can peer between regions.



**Remember the following;**

* Think of a VPC as a logical datacenter in AWS
* Consists of IGWs (Or Virtual Private Gateways), Route Tables, Network Access Control Lists, Subnets and Security Groups.
* 1 Subnet = 1 Availability Zone
* Security Groups are stateful; Network Access Control Lists are Stateless
* **NO TRANSITIVE PEERING**
* When you create a VPC a default Route Table, Network Access Control List (NACL) and a default Security Group.
* It won’t create any subnets, nor will it create a default internet gateway.
* US-East-1A in your AWS accounts can be a completely different availability zone to US-East-1A in another AWS accounts. The AZ’s are randomized.
* Amazon always reserve 5 IP addresses within your subnets.
* You can only have 1 Internet Gateway per VPC.
* Security Groups can’t span VPCs.

**NAT Instances and NAT Gateways: -**

**Exam Tips: -**

**NAT Instances: -**

* When creating a NAT instance, Disable Source/Destination Check on the Instance.
* NAT instances must be in a public subnet.
* There must be a route out of the private subnet to the NAT instance, in order for this to work.
* The amount of traffic that NAT instances can support depends on the instance size. If you are bottlenecking, increase the instance size.
* You can create High Availability using Autoscaling Groups, multiple subnets in different AZs, and a script to automate failover.
* Behind the Security Group.

**NAT Gateways: -**

* Redundant inside the Availability Zone
* Preferred by the enterprise
* Start at 5Gbps and scale currently to 45Gbps
* No need to patch
* Not associated with security groups
* Automatically assigned a public IP address
* Remember to update your route tables.
* No need to disable Source/Destination Checks

If you have resources in multiple Availability Zones and they share one NAT gateway, in the event of the NAT gateway’s Availability Zone is down, resources in the other Availability Zones lose internet access. To create an Availability Zone-independent architecture, create a NAT gateway in each Availability Zone and configure your routing to ensure that resources use the NAT gateway in the same Availability Zone.

**Network Access Control List: -**

**Exam Tips: -**

* Your VPC automatically comes with a default Network ACL, and by default it allows all outbound and inbound traffic.
* You can create custom network ACLs. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.
* Each subnet in your VPC must be associated with a network ACL. If you don’t explicitly associated a subnet with network ACL, the subnet is automatically associated with the default network ACL.
* Block IP Addresses using network ACLs not using the Security Groups.
* You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time. When you associate a network ACL with a subnet, the previous association is removed.
* Network ACLs contain a numbered list of rules that is evaluated in order, starting with the lowest numbered rule.
* Network ACLs have a separate inbound and outbound rule, and each rule can either allow or deny traffic.
* Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).



**VPC Flow Logs: -**

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data is stored using Amazon CloudWatch Logs. After you’ve created a flow logs, you can view and retrieve its data in Amazon CloudWatch Logs.

**Flow Logs can be created at 3 Levels: -**

* VPC
* Subnet
* Network Interface Level

**Exam Tips: -**

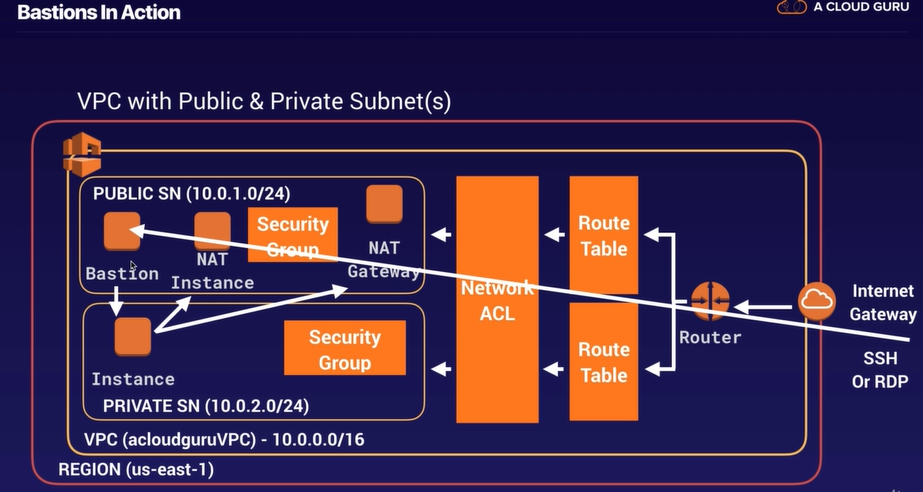
* You cannot enable flow logs for VPCs that are peered with your VPC unless the peer VPC is in your account.
* You cannot tag a flow log.
* After you’ve created a flow log, you cannot change its configuration; for example, you can’t associate a different IAM role with the flow log.

**Not all IP Traffic in monitored;**

* Traffic generated by instances when they contact the Amazon DNS server. If you use your own DNS server, then all traffic to that DNS server is logged.
* Traffic generated by a Windows instance for Amazon Windows license activation
* Traffic to and from 169.254.169.254 for instance metadata.
* DHCP traffic
* Traffic to the reserved IP address for the default VPC router.

**Bastions Host: -**

A bastion host is a special purpose computer on a network specifically designed and configured to withstand attacks. The computer generally hosts a single application, for example a proxy server, and all other services are removed or limited to reduce the threat to the computer. It is hardened in this manner primarily due to its location and purpose, which is either on the outside of a firewall or in a demilitarized zone (DMZ) and usually involves access from untrusted networks or computers.



**Exam Tips: -**

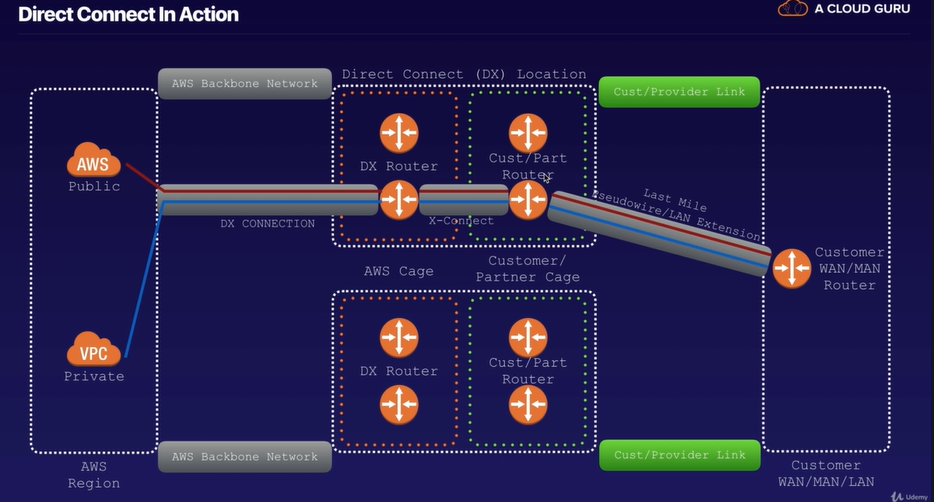
* A NAT Gateway or NAT Instance is used to provide internet traffic to EC2 instance in a private subnet.
* A Bastion is used to securely administer EC2 instances (using SSH or RDP). Bastions are called Jump Boxes in Australia.
* You cannot use a NAT Gateway as Bastion host.

**AWS Direct Connect: -**

AWS Direct Connect is a cloud service solution that makes it easy to establish a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your datacenter, office or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

**Exam Tips: -**

* Direct connect directly connects your data center to AWS
* Useful for high throughput workloads (i.e. lots of network traffic)
* Or if you need a stable and reliable secure connection.



**AWS VPC Endpoints: -**

A VPC endpoints enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by Private Link without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.

Endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components that allows communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.

**Exam Tips: -**

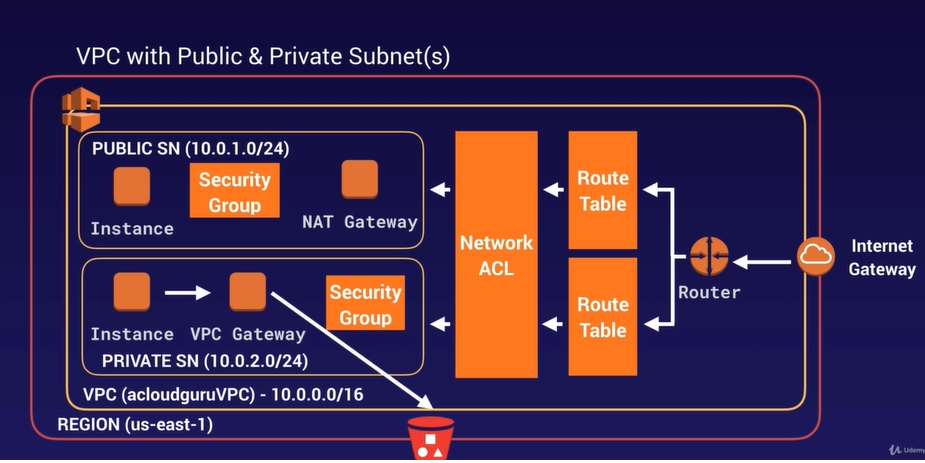
**There are two types of VPC endpoints:**

* Interface Endpoints
* Gateway Endpoints

**Currently Gateway Endpoints Support:**

* Amazon S3
* DynamoDB





**AWS SQS: -**

Amazon SQS is a web service that gives you access to a message queue that can be used to store message while waiting for a computer to process them.

Amazon SQS is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component. A queue is temporary repository for messages that are awaiting processing.

Using Amazon SQS, you can decouple the components of an application, so they run independently, easing message management between components. Any components of a distributed application can store messages in a fail-safe queue. Messages can contain up to 256 KB of text in any format. Any component can later retrieve the messages programmatically using the Amazon SQS API.

The queue acts as a buffer between the component producing and saving data, and the component receiving the data for processing. This means the queue resolves issues that arise if the producer is producing work faster than the consumer can process it, or if the producer or consumer are only intermittently connected to the network.

**There are two types of Queue:**

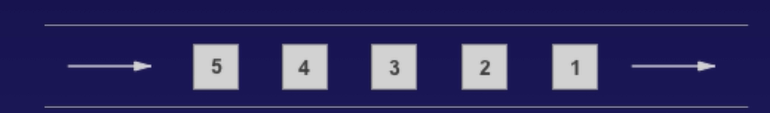
* Standard Queues (default)
* FIFO Queues

**Standard Queues (default): -**

Amazon SQS offers standard as the default queue type. A standard queue lets you have a nearly-unlimited number of transactions per second. Standard queues guarantee that a message is delivered at least once. However, occasionally (because of the highly-distributed architecture that allows high throughput), more than one copy of a message might be delivered out of order. Standard queues provide best-effort ordering which ensures that messages are generally delivered in the same order as they are sent.

**FIFO Queues: -**

The FIFO queue complements the standard queue. The most important features of this queue type are FIFO (first-in-first-out) delivery and exactly-once processing: The order in which messages are sent and received is strictly preserved and a message is delivered once and remains available until a consumer processes and deletes it; duplicates are not introduced in the queue.



FIFO queues also support message group that allow multiple ordered message groups within a single queue. FIFO queue are limited to 300 transactions per second (TPS) but have all the capabilities of standard queues.

**Exam Tips: -**

* SQS is pull based, not push based
* Messages are 256 KB in size
* Messages can be kept in the queue from 1 minute to 14 days; the default retention period is 4 days.
* Visibility Time Out is the amount of time that the message is invisible in the SQS queue after a reader picks up that message. Provided the job is processed before the visibility time out expires, the message will then be deleted from the queue. If the job is not processed within that time, the message will become visible again and another reader will process it. This could result in the same message being delivered twice.
* Visibility timeout maximum is 12 hours.
* SQS guarantees that you message will be processed at least once.
* Amazon SQS long polling is a way to retrieve messages from your Amazon SQS queues. While the regular short polling returns immediately (even if the message queue being polled is empty), long polling doesn’t return a response until a message arrives in the message queue, or the long poll times out.

**Amazon Simple Workflow Service (Amazon SWF): -**

Amazon Simple Workflow Service (Amazon SWF) is a web service that makes it easy to coordinate work across distributed application components. SWF enables applications for a range of use cases, including media processing, web application backends, business process workflows, and analytics pipelines, to be designed as a coordination of tasks.

Task represents invocations of various processing steps in an application which can be performed by executable code, web service calls, human actions, and scripts.

**SWF vs SQS: -**

* SQS has a retention period of up to 14 days; with SWF, workflow executions can last up to 1 year.
* Amazon SWF presents a task-oriented API, whereas Amazon SQS offers a message-oriented API.
* Amazon SWF ensures that a task is assigned only once and is never duplicated. With Amazon SQS, you need to handle duplicated messages and may also need to ensure that a message is processed only once.
* Amazon SWF keeps track of all the tasks and events in an application. With Amazon SQS, you need to implement your own application-level tracking, especially if your application uses multiple queues.

**SWF Actors: -**

* Workflow Starters – An application that can initiate (start) a workflow. Cloud be your e-commerce website following the placement of an order, or a mobile app searching for bus times.
* Deciders – Control the flow of activity tasks in a workflow execution. If something has finished (or failed) in a workflow, a Decider decides what to do next.
* Activity Workers – Carry out the activity tasks.

**Amazon Simple Notification Service (Amazon SNS): -**

Amazon Simple Notification Service (SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud.

It provides developers with a highly scalable, flexible, and cost-effective capability to publish messages from an application and immediately deliver them to subscribers or other applications.

Push notifications to Apple, Google, Fire OS, and Windows devices, as well as Android devices in China with Baidu Cloud Push.

Besides pushing cloud notifications directly to mobile devices, Amazon SNS can also deliver notifications by SMS test message or email to Amazon Simple Queue Service (SQS) queues, or to any HTTP endpoint.

**What is SNS Topic?**

SNS allows you to group multiple recipients using topics. A topic is an “access point” for allowing recipients to dynamically subscribe for identical copies of the same notification.

One topic can support deliveries to multiple endpoint types-for example, you can group together iOS, Android and SMS recipients. When you publish once to a topic, SNS delivers appropriately formatted copies of your message to each subscriber.

**SNS Availability: -**

To prevent messages from being lost, all messages published to Amazon SNS are stored redundantly across multiple availability zones.

**Exam Tips: -**

* Instantaneous, push-based delivery (no polling)
* Simple APIs and easy integration with applications
* Flexible message delivery over multiple transport protocols
* Inexpensive, pay-as-you-go model with no up-front costs
* Web-based AWS Management Console offers the simplicity of a point-and click-interface

**AWS API Gateway: -**

Amazon API Gateway is a fully managed service that makes it easy for developers to publish, maintain, monitor, and secure APIs at any scale.

With a few clicks in the AWS Management Console, you can create an API that acts as a “front door: for applications to access data, business logic, or functionality from your back-end services, such as application running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, or any web application.

**What Can API Gateway Do?**

* Expose HTTPS endpoints to define a RESTful API
* Serverless-ly connect to services like Lambda and DynamoDB
* Send each API endpoint to a different target
* Run efficiently with low cost
* Scale effortlessly
* Track request to prevent attacks
* Connect to CloudWatch to log all requests for monitoring
* Maintain multiple versions of your API

**How Do I configure API Gateway?**

* Define an API (container)
* Define Resources and nested Resources (URL paths)
* For each Resource:
  + Select supported HTTP methods (verbs)
  + Set security
  + Choose target (such as EC2, Lambda, DynamoDB, etc.)
  + Set request and response transformations

**How Do I Deploy API Gateway?**

* **Deploy API to a stage**
  + Uses API Gateway domain, by default
  + Can use custom domain
  + Now supports AWS Certificate Manager: free SSL/TLS certs

**API Gateway Caching: -**

You can enable API caching in Amazon API Gateway to cache your endpoint’s response. With caching, you can reduce the number of calls made to your endpoint and also improve the latency of the requests to your API. When you enable caching for a stage, API Gateway caches responses from your endpoint for a specified time-to-live (TTL) period, in seconds. API Gateway then responds to the request by looking up the endpoint response from the cache instead of making a request to your endpoint.

**Same Origin Policy: -**

In Computing, the same-origin policy is an important concept in the web application security model. Under the policy, a web browser permits scripts contained in a first web page to access data in a second web page, but only if both web pages have the same origin.

**This is done to prevent Cross-Site Scripting (XSS) attacks.**

* Enforced by web browser
* Ignored by tools like PostMan and curl

**Exam Tips: -**

* Remember what API Gateway is at a high level
* API Gateway has caching capabilities to increase performance
* API Gateway is low cost and scales automatically
* You can throttle API Gateway to prevent attacks
* You can log results to CloudWatch
* If you are using JavaScript/AJAX that uses multiple domains with API Gateway, ensure that you have enabled CORS on API Gateway
* CORS is enforced by the client

**CORS Explained: -**

CORS is one way the server at the other end (not the client code in the browser) can relax the same-origin policy.

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts) on a web page to be requested from another domain outside the domain from the first resource was served.

**AWS Kinesis: -**

**What is Streaming Data?**

Streaming Data is data that is generated continuously by thousands of data sources, which typically send in the data records simultaneously, and in small sizes (order of kilobytes.)

* Purchase from online stores (think amazon.com)
* Stock Price
* Game data (as gamer plays)
* Social network data
* Geospatial data (think uber.com)
* iOT sensor data

**What is Kinesis?**

Amazon Kinesis is a platform on AWS to send your streaming data to. Kinesis makes it easy to load and analyze streaming data, and also providing the ability for you to build your own custom applications for your business needs**.**

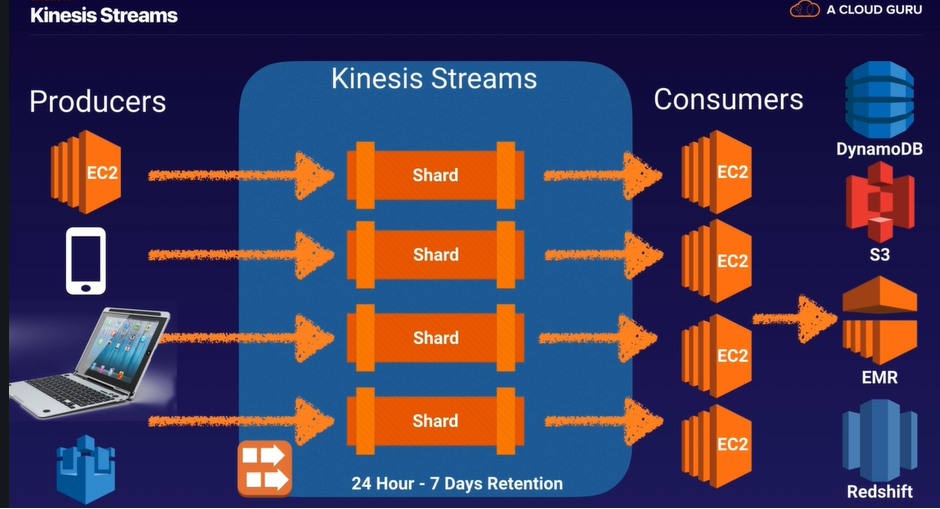
**3 Different Types of Kinesis: -**

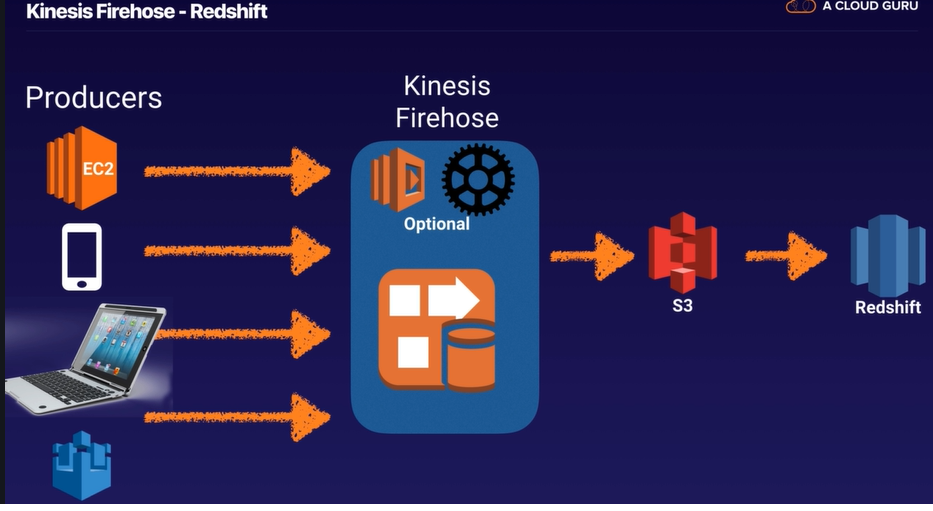
* **Kinesis Streams**
* **Kinesis Firehose**
* **Kinesis Analytics**

**Kinesis Streams: -**

**Kinesis Streams Consist of Shards;**

* 5 transactions per second for reads, up to a maximum total data rate of 2 MB per second and up to 1,000 records per second for writes, up to a maximum total data write rate of 1 MB per second (including partition keys)
* The data capacity of your stream is a function of the number of shards that you specify for the stream. The total capacity of the stream is the sum of the capacities of its shards.







**AWS Web Identity Federation – Cognito: -**

**What is Web Identity Federation?**

Web Identity Federation lets you give your users access to AWS resources after they have successfully authenticated with a web-based identity provider like Amazon, Facebook, or Google.

Following successful authentication, the user receives an authentication code from the Web ID provider, which they can trade for temporary AWS security credentials.

**Amazon Cognito provides Web Identity Federation with the following features:**

* Sign-up and sign-in to your apps
* Access for guest users
* Acts as an Identity Broker between your application and Web ID providers, so you don’t need to write any additional code.
* Synchronizes user data for multiple devices
* Recommended for all mobile applications AWS services.

**Cognito User Pools: -**

User Pools are user directories used to manage sign-up and sign-in functionality for mobile and web applications.

Users can sign-in directly to the User Pool, or using Facebook, Amazon, or Google. Cognito acts as an Identity Broker between the identity provider and AWS. Successful authentication generates a JSON Web Token (JWTs)

**Cognito Identity Pools: -**

Identity Pools enable provide temporary AWS credentials to access AWS services like S3 or DynamoDB.

**Cognito Synchronization: -**

Cognito tracks the association between user identity and the various devices they sign-in from. In order to provide a seamless user experience for your application, Cognito uses Push Synchronization to push updates and synchronize user data across multiple devices. Cognito uses SNS to send a notification to all the devices associated with a given user identity whenever data stored in the cloud changes.

**Exam Tips: -**

* Federation allows users to authenticate with a Web Identity Provider (Google, Facebook, Amazon)
* The user authenticates first with the Web ID Provider and receives an authentication token, which is exchanged for temporary AWS credentials allowing them to assume an IAM Role.
* Cognito is an Identity Broker which handles interaction between your applications and the Web ID provider (you don’t need to write your own code to do this)
* User Pool is user based. It handles things like user registration, authentication, and account recovery.
* Identity pools authorize access to your AWS resources.