AWS Certified Solutions Architect - Associate

Week 2 – Content Review

15 JULY 2021

DHAVAL SONI

infostretch

Agenda

- Week 2 content review
  - AWS RDS
  - RDS Read Replicas & Multi-AZ
  - RDS Encryption & Security
  - AWS Aurora & Advance Concepts
  - Amazon ElastiCache, Amazon Route53
  - Amazon Elastic Beanstalk
  - Amazon S3 & Advance Concepts
  - Amazon Athena, Amazon CloudFront
  - AWS Global Accelerator, AWS Snow Family
  - AWS Storage Gateway, Amazon FSx
- Topics for Week 3
- Q & A


infostretch
CONTENT REVIEW

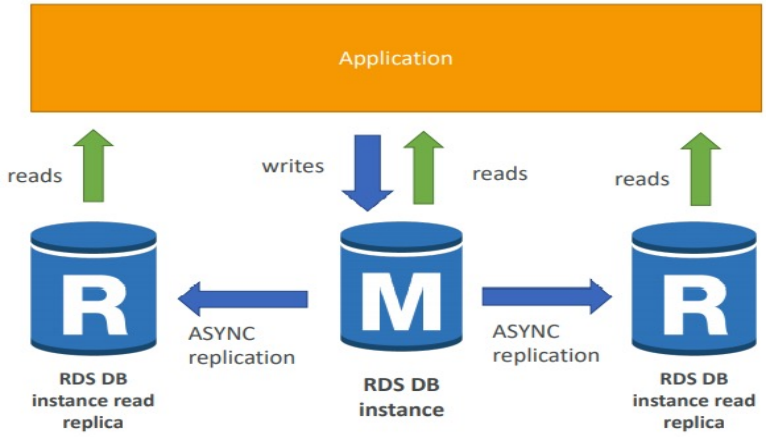
AWS RDS

- RDS stands for Relational Database Service
- It’s a managed DB service for DB use SQL as a query language.
- It allows you to create databases in the cloud that are managed by AWS
  - Postgres
  - MySQL
  - MariaDB
  - Oracle
  - Microsoft SQL Server
  - Aurora (AWS Proprietary database)
- RDS is a managed service:
  - Automated provisioning, OS patching
  - Continuous backups and restore to specific timestamp (Point in Time Restore)! Monitoring dashboards
  - Read replicas for improved read performance
  - Multi AZ setup for DR (Disaster Recovery)
  - Maintenance windows for upgrades
  - Scaling capability (vertical and horizontal)
  - Storage backed by EBS (gp2 or io1)
- Backups are automatically enabled in RDS

RDS Read Replicas & Multi AZ

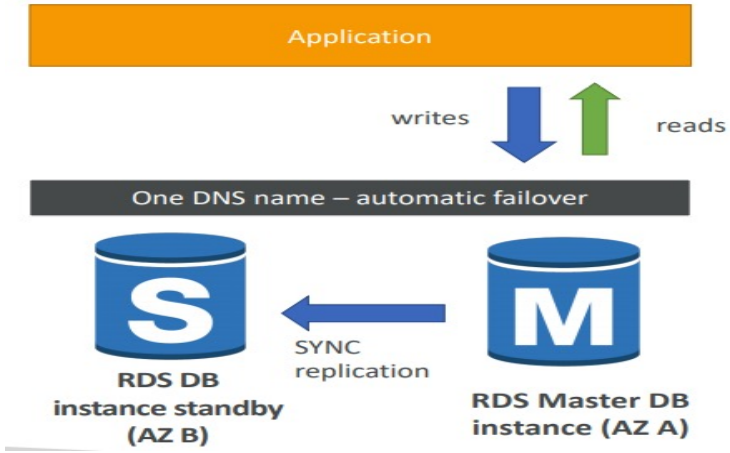
RDS Read Replicas

- RDS Read Replicas used for Read Scalability
- Up to 5 Read Replicas within AZ, without AZ or Cross Region
- Replication is ASYNC, so reads are eventually consistent
- Read Replicas used when you have a production DB that is taking on normal Load or you want to run a reporting application to run some analytics
- In AWS there's a network cost when data goes from one AZ to another and to reduce the cost, you can have your Read Replicas in the same AZ
- The Read Replicas be setup as Multi AZ for Disaster Recovery (DR)



RDS Multi AZ

- RDS Multi AZ is used for Disaster Recovery
- SYNC replication
- One DNS name – automatic app failover to standby
- Increase availability
- Failover in case of loss of AZ, loss of network, instance or storage failure
- No manual intervention in apps
- Not used for scaling



RDS Encryption & Security

infostretch

RDS Encryption

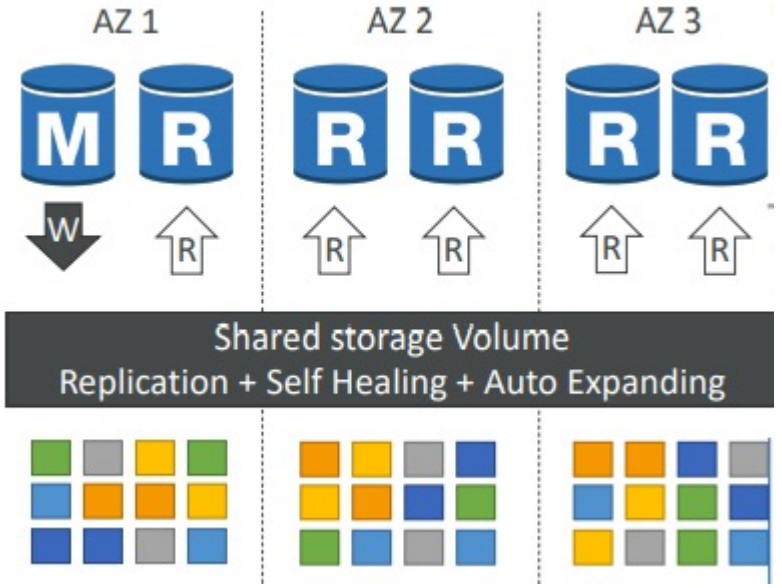
- **At rest encryption**
  - Possibility to encrypt the master & read replicas with AWS KMS - AES-256 encryption
  - Encryption has to be defined at launch time
  - If the master is not encrypted, the read replicas cannot be encrypted
  - Transparent Data Encryption (TDE) available for Oracle and SQL Server
- **In-flight encryption**
  - SSL certificates to encrypt data to RDS in flight
  - Provide SSL options with trust certificate when connecting to database
  - To enforce SSL:
    - PostgreSQL: rds.force_ssl=1 in the AWS RDS Console (Parameter Groups)
    - MySQL: Within the DB: GRANT USAGE ON *.* TO 'mysqluser'@'%' REQUIRE SSL

RDS Security

- **Network Security**
  - RDS databases are usually deployed within a private subnet, not in a public one
  - RDS security works by leveraging security groups (the same concept as for EC2 instances) – it controls which IP / security group can communicate with RDS
- **Access Management**
  - IAM policies help control who can manage AWS RDS (through the RDS API)
  - Traditional Username and Password can be used to login into the database
  - IAM-based authentication can be used to login into RDS MySQL & PostgreSQL

AWS Aurora

- Aurora is a proprietary technology from AWS (not open sourced)
- Postgres and MySQL are both supported as Aurora DB (that means your drivers will work as if Aurora was a Postgres or MySQL database)
- Aurora is “AWS cloud optimized” and claims 5x performance improvement over MySQL on RDS, over 3x the performance of Postgres on RDS
- Aurora storage automatically grows in increments of 10GB, up to 64 TB.
- Aurora can have 15 replicas while MySQL has 5, and the replication process is faster (sub 10 ms replica lag)
- Failover in Aurora is instantaneous. It’s HA (High Availability) native.
- Aurora costs more than RDS (20% more) – but is more efficient
- For Aurora High Availability you need 6 copies of your data across 3 AZ:
  - 4 copies out of 6 needed for writes
  - 3 copies out of 6 need for reads
  - Self healing with peer-to-peer replication
  - Storage is striped across 100s of volumes
- Automated failover for master in less than 30 seconds
- Master + up to 15 Aurora Read Replicas serve reads
- Support for Cross Region Replication

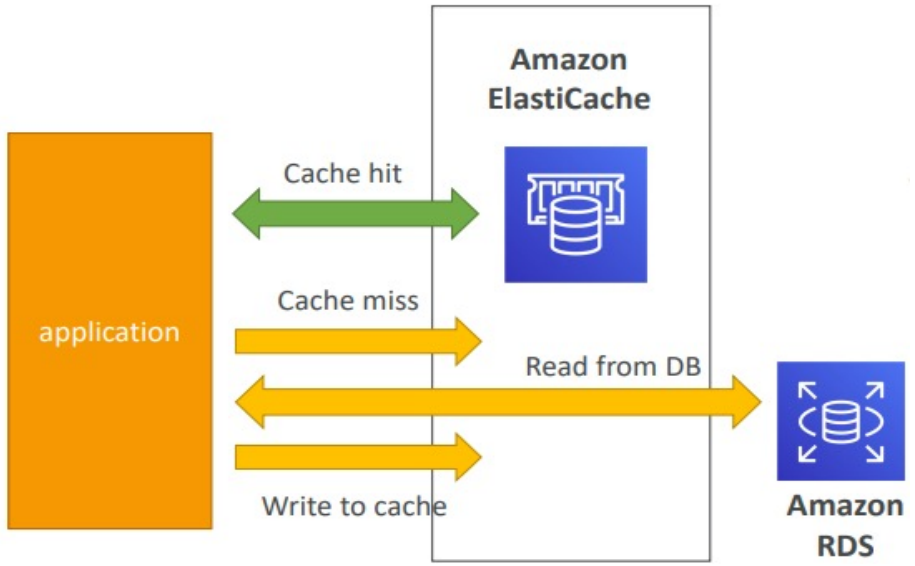


Aurora - Advanced Concepts

- **Aurora Security**
  - Similar to RDS because uses the same engines
  - Encryption at rest using KMS
  - Automated backups, snapshots and replicas are also encrypted
  - Encryption in flight using SSL (same process as MySQL or Postgres)
  - Possibility to authenticate using IAM token (same method as RDS)
- **Aurora Serverless**
  - Automated database instantiation and auto - scaling based on actual usage
  - Good for infrequent, intermittent or unpredictable workloads
  - No capacity planning needed
  - Pay per second, can be more cost -effective
- **Aurora Global Database**
  - 1 Primary Region (read / write)
  - Up to 5 secondary (read-only) regions, replication lag is less than 1 second
  - Up to 16 Read Replicas per secondary region
  - Helps for decreasing latency
  - Promoting another region (for disaster recovery) has an RTO of < 1 minute

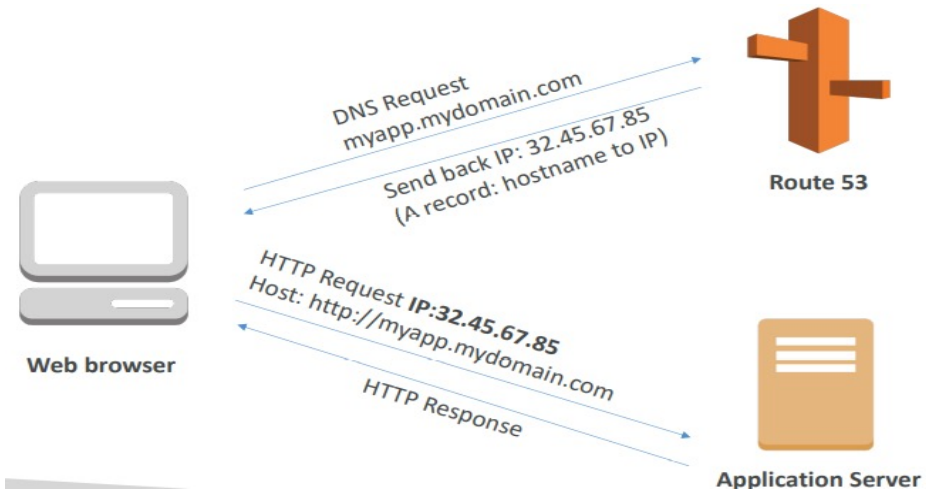
Amazon ElastiCache

- Caches are in-memory databases with really high performance, low latency
- Helps reduce load off of databases for read intensive workloads
- Helps make your application stateless
- Using ElastiCache involves heavy application code changes
- Applications queries ElastiCache, if not available, get from RDS and store in ElastiCache.
- Helps relieve load in RDS
- Cache must have an invalidation strategy to make sure only the most current data is used in there.
- All caches in ElastiCache support SSL in flight encryption but Do not support IAM authentication
- IAM policies on ElastiCache are only used for AWS API-level security
- Redis AUTH is used to set a “password/token” when you create a Redis cluster. This an extra level of security for your cache (on top of security groups)
- Memcached supports SASL-based authentication (advanced)
- Patterns for ElastiCache
  - Lazy Loading: all the read data is cached, data can become stale in cache
  - Write Through: Adds or update data in the cache when written to a DB (no stale data)
  - Session Store: store temporary session data in a cache (using TTL features)



Amazon Route53

- Route53 is a Managed DNS (Domain Name System)
- DNS is a collection of rules and records which helps clients understand how to reach a server through URLs.
- In AWS, the most common records are:
  - A: hostname to IPv4
  - AAAA: hostname to IPv6
  - CNAME: hostname to hostname
  - Alias: hostname to AWS resource
- Route53 has below Routing Policies:
  - **Simple routing policy** – Use for a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.
  - **Failover routing policy** – Use when you want to configure active-passive failover.
  - **Geolocation routing policy** – Use when you want to route traffic based on the location of your users.
  - **Geoproximity routing policy** – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.
  - **Latency routing policy** – Use when you have resources in multiple AWS Regions and you want to route traffic to the Region that provides the best latency with less round-trip time.
  - **Multivalue answer routing policy** – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.
  - **Weighted routing policy** – Use to route traffic to multiple resources in proportions that you specify.



Amazon Elastic Beanstalk

- Elastic Beanstalk is a developer centric view of deploying an application on AWS
- It uses all the component's like: EC2, ASG, ELB, RDS, etc But it's all in one view that's easy to make sense of!
- We still have full control over the configuration
- Beanstalk is free but you pay for the underlying instances
- Just the application code is the responsibility of the developer
- Three architecture models:
  - Single Instance deployment: good for dev
  - LB + ASG: great for production or pre-production web applications
  - ASG only: great for non-web apps in production (workers, etc..)
- Elastic Beanstalk has three components
  - Application
  - Application version: each deployment gets assigned a version
  - Environment name (dev, test, prod…): free naming
- You deploy application versions to environments and can promote application versions to the next environment
- Rollback feature to previous application version
- Full control over lifecycle of environments

Amazon Simple Storage Service (S3)

infostretch

- Amazon S3 allows people to store objects (files) in “buckets” (directories)
- Buckets must have a globally unique name
- Buckets are defined at the region level
- Naming convention
    - No uppercase
    - No underscore
    - 3-63 characters long
    - Not an IP
    - Must start with lowercase letter or number
- Objects (files) have a Key
- The key is the FULL path:
    - s3://my-bucket/my_file.txt
    - s3://my-bucket/my_folder1/another_folder/my_file.txt
- The key is composed of prefix + object name
    - Object values are the content of the body:
    - Max Object Size is 5TB (5000GB)
- If uploading more than 5GB, must use “multi-part upload”
- Metadata (list of text key / value pairs – system or user metadata)
- Tags (Unicode key / value pair – up to 10) – useful for security / lifecycle
- Version ID (if versioning is enabled)

Amazon S3 (Cont.)

- You can version your files in Amazon S3
- It is enabled at the bucket level
- Same key overwrite will increment the “version”: 1, 2, 3….
- It is best practice to version your buckets:
  - Protect against unintended deletes (ability to restore a version)
  - Easy roll back to previous version
- Any file that is not versioned prior to enabling versioning will have version “null”
- Suspending versioning does not delete the previous versions
- There are 4 methods of encrypting objects in S3
  - SSE-S3: encrypts S3 objects using keys handled & managed by AWS
  - SSE-KMS: leverage AWS Key Management Service to manage encryption keys
  - SSE-C: when you want to manage your own encryption keys
  - Client Side Encryption
- IAM policies are used as User based security in S3 - which API calls should be allowed for a specific user from IAM console
- Resource Based Security can be done by:
  - Bucket Policies - bucket wide rules from the S3 console - allows cross account
  - Object Access Control List (ACL) – finer grain
  - Bucket Access Control List (ACL) – less common

Amazon S3 Advanced concepts

infostretch

- **Consistency Models:**
  - Read after write consistency for PUTS of new objects
    - As soon as a new object is written, we can retrieve it ex: (PUT 200 => GET 200)
    - This is true, except if we did a GET before to see if the object existed ex: (GET 404 => PUT 200 => GET 404) – eventually consistent
  - Eventual Consistency for DELETES and PUTS of existing objects
    - If we read an object after updating, we might get the older version ex: (PUT 200 => PUT 200 => GET 200 (might be older version))
    - If we delete an object, we might still be able to retrieve it for a short time ex: (DELETE 200 => GET 200)
- **S3 MFA Delete**
  - MFA (multi factor authentication) forces user to generate a code on a device (usually a mobile phone or hardware) before doing important operations on S3
  - To use MFA-Delete, enable Versioning on the S3 bucket
  - Only the bucket owner (root account) can enable/disable MFA-Delete
  - MFA-Delete currently can only be enabled using the CLI
- **S3 Access Log**
  - For audit purpose, you may want to log all access to S3 buckets
  - Any request made to S3, from any account, authorized or denied, will be logged into another S3 bucket
  - That data can be analyzed using data analysis tools…

Amazon S3 Advanced concepts

infostretch

- **S3 Object Lock**
  - Adopt a WORM (Write Once Read Many) model
  - Block an object version deletion for a specified amount of time
- **Glacier Vault Lock**
  - Adopt a **WORM** (Write Once Read Many) model
  - Lock the policy for future edits (can no longer be changed) • Helpful for compliance and data retention
- **S3 Lifecycles Rules:**
  - **Transition actions:** It defines when objects are transitioned to another storage class.
    - Move objects to Standard IA class 60 days after creation
    - Move to Glacier for archiving after 6 months
  - **Expiration actions:** configure objects to expire (delete) after some time
    - Access log files can be set to delete after a 365 days • Can be used to delete old versions of files (if versioning is enabled)
    - Can be used to delete incomplete multi-part uploadsAs soon as a new object is written, we can retrieve it ex: (PUT 200 => GET 200)
- **S3 Select & Glacier Select**
  - Retrieve less data using SQL by performing server-side filtering
  - Can filter by rows & columns (simple SQL statements)
  - Less network transfer, less CPU cost client-side

Amazon S3 Storage Classes

infostretch

- **S3 Standard** — The default storage class. If you don't specify the storage class when you upload an object, Amazon S3 assigns the S3 Standard storage class.

- **Reduced Redundancy** — The Reduced Redundancy Storage (RRS) storage class is designed for noncritical, reproducible data that can be stored with less redundancy than the S3 Standard storage class. (omitted)

- **S3 Standard-IA** — Amazon S3 stores the object data redundantly across multiple geographically separated Availability Zones (similar to the S3 Standard storage class). S3 Standard-IA objects are resilient to the loss of an Availability Zone. This storage class offers greater availability and resiliency than the S3 One Zone-IA class.

- **S3 One Zone-IA** — Amazon S3 stores the object data in only one Availability Zone, which makes it less expensive than S3 Standard-IA. However, the data is not resilient to the physical loss of the Availability Zone resulting from disasters, such as earthquakes and floods. The S3 One Zone-IA storage class is as durable as Standard-IA, but it is less available and less resilient.
-
- **S3 Glacier** — Use for archives where portions of the data might need to be retrieved in minutes. Data stored in the S3 Glacier storage class has a minimum storage duration period of 90 days and can be accessed in as little as 1-5 minutes using expedited retrieval. If you have deleted, overwritten, or transitioned to a different storage class an object before the 90-day minimum, you are charged for 90 days.

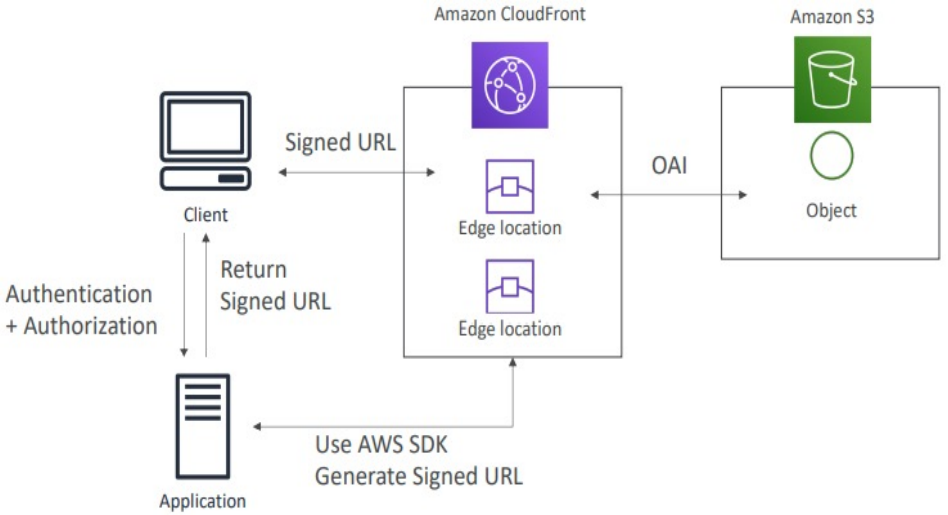
- **S3 Glacier Deep Archive** — Use for archiving data that rarely needs to be accessed. Data stored in the S3 Glacier Deep Archive storage class has a minimum storage duration period of 180 days and a default retrieval time of 12 hours. If you have deleted, overwritten, or transitioned to a different storage class an object before the 180-day minimum, you are charged for 180 days.

Amazon Athena

- Serverless service to perform analytics directly against S3 files
- Uses SQL language to query the files
- Has a JDBC / ODBC driver
- Charged per query and amount of data scanned
- Supports CSV, JSON, ORC, Avro, and Parquet (built on Presto)
- Use cases: Business intelligence / analytics / reporting, analyze & query VPC Flow Logs, ELB Logs, CloudTrail trails, etc.

Amazon CloudFront

- Content Delivery Network (CDN)
- Improves read performance, content is cached at the edge
- DDoS protection, integration with Shield, AWS Web Application Firewall
- CloudFront origins an be S3, EC2, ALB etc
- You can restrict who can access your distribution
  - Whitelist: Allow your users to access your content only if they're in one of the countries on a list of approved countries.
  - Blacklist: Prevent your users from accessing your content if they're in one of the countries on a blacklist of banned countries.
- The “country” is determined using a 3rd party Geo-IP database.
- Great for static content that must be available everywhere
- To distribute paid shared content to premium users over the world, We can use CloudFront Signed URL / Cookie.
- Signed URL = access to individual files (one signed URL per file)
- Signed Cookies = access to multiple files (one signed cookie for many files)



AWS Snow Family

Snowball
- Physical data transport solution that helps moving TBs or PBs of data in or out of AWS
- Alternative to moving data over the network (and paying network fees)
- Secure, tamper resistant, uses KMS 256 bit encryption
- Tracking using SNS and text messages. E -ink shipping label
- Pay per data transfer job

Use cases: large data cloud migrations, DC decommission, disaster recovery
TiP: If it takes more than a week to transfer over the network, use Snowball devices

Snowball Edge
- Snowball Edges add computational capability to the device
- 100 TB capacity with either:
- Storage optimized – 24 vCPU
- Compute optimized – 52 vCPU & optional GPU
- Supports a custom EC2 AMI so you can perform processing on the go
- Supports custom Lambda functions
- Very useful to pre-process the data while moving

Use case: data migration, image collation, IoT capture, machine learning

Snowmobile
- Transfer exabytes of data (1 EB = 1,000 PB = 1,000,000 TBs)
- Each Snowmobile has 100 PB of capacity (use multiple in parallel)
- Better than Snowball if you transfer more than 10 PB

AWS Storage Gateway

Storage Gateway is a Bridge between on-premise data and cloud data in S3

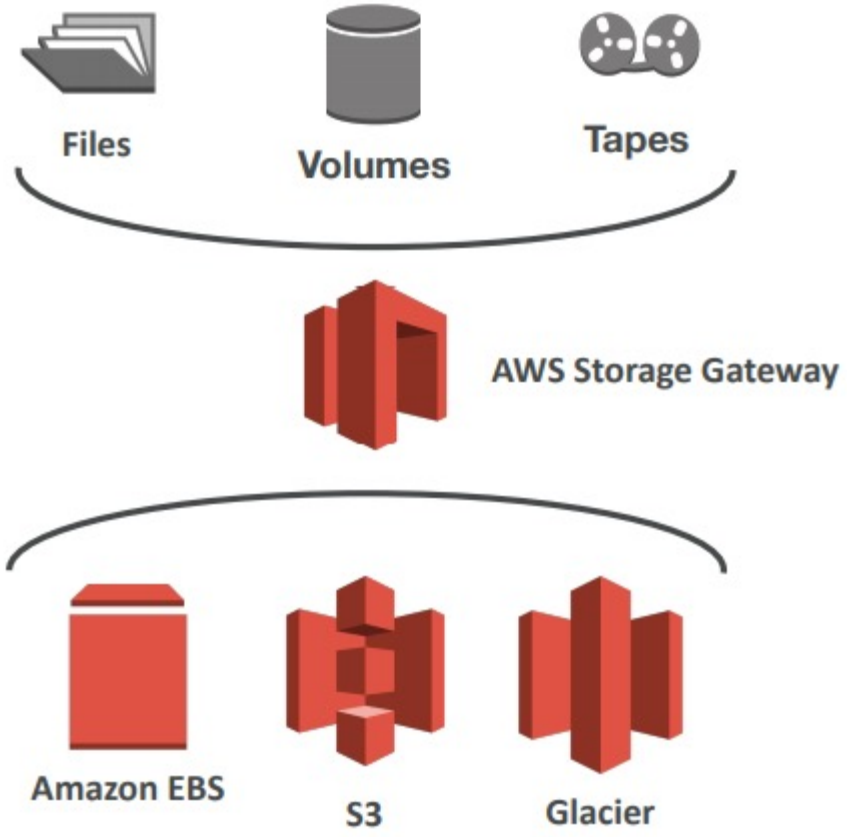
Use cases: disaster recovery, backup & restore, tiered storage

3 types of Storage Gateway:

File Gateway: Amazon S3 File Gateway supports a file interface into Amazon Simple Storage Service (Amazon S3) and combines a service and a virtual software appliance. By using this combination, you can store and retrieve objects in Amazon S3 using industry-standard file protocols such as Network File System (NFS) and Server Message Block (SMB).

Tape Gateway: A tape gateway provides cloud-backed virtual tape storage. The tape gateway is deployed into your on-premises environment as a VM running on VMware ESXi, KVM, or Microsoft Hyper-V hypervisor.

Volume Gateway: A volume gateway provides cloud-backed storage volumes that you can mount as Internet Small Computer System Interface (iSCSI) devices from your on-premises application servers.



Amazon FSx

infostretch

For Windows:

- FSx for Windows is a fully managed Windows file system share drive
- Supports SMB protocol & Windows NTFS
- Microsoft Active Directory integration, ACLs, user quotas
- Built on SSD, scale up to 10s of GB/s, millions of IOPS, 100s PB of data
- Can be accessed from your on-premise infrastructure
- Can be configured to be Multi-AZ (high availability)
- Data is backed-up daily to S3.

For Lustre:

- Lustre is a type of parallel distributed file system, for large-scale computing
- The name Lustre is derived from “Linux” and “cluster”
- Machine Learning, High Performance Computing (HPC)
- Video Processing, Financial Modeling, Electronic Design Automation
- Scales up to 100s GB/s, millions of IOPS, sub-ms latencies
- Seamless integration with S3
    - Can “read S3” as a file system (through FSx)
    - Can write the output of the computations back to S3 (through FSx)
    - Can be used from on-premise servers

Topics we'll cover in week - 3

- Decoupling Applications: SNS, SQS, Active MQ, Kinesis
- Containers on AWS: ECS, Fargate, EKS
- IAM Advanced
- AWS Monitoring & Audit: CloudWatch, CloudTrail
- Databases in AWS
- Lambda, DynamoDB, API Gateway

infostretch

Q & A

Thank you

CONNECT WITH US

EMAIL INFO@INFOSTRETCH.COM

CALL +1-408-727-1100