

Session 1 : What is Data,Databases and DBMS

Data : In simple words,data can be facts related to any object.Ex : Age,Job,House Num,Contact Num Name,etc related to us

Database : Database is a Systematic collection of Data.Databases support storage and manipulation of data.Ex : Facebook,Telecom Companies,Amazon etc

Database Management System(DBMS) : It is a collection of programmes which enable its users to access database,manipulate data,reporting/representation of data

- *Types of DBMS :* Hierarchial,Network,Relational,Object Oriented

Session 2 : Relational Database (Attribute,Tuple,Horizontal and Vertical Scaling in RDBMS)

Relational Database :

- A relational database is a data structure that allows us to link information from different 'tables' or different types of data bucket
- Tables are related to each other
- All fields must be filled
- Best suites for OLTP(Online Transaction Processing)
- Relational DB : MySQL,Oracle DBMS,IBM DB2 etc..

Attribute	Attribute	Attribute	Attribute
Name	Age	Location	Phone
X	21	LA	+8100
Y	33	CA	+9300
z	18	IN	+9191

- A single column is called attribute or fields and a whole row is called Tuple
- A row of a table is also called record and it contains the specific information of each individual entry in the table
- Each table has its own primary key
- A schema(Design of Database) is used to strictly defines table,colum,indexes and relation between tables
- Relational DB are usually used in Enterprise applications/scenario but exception is MySQL which is used for web application
- Common application for MySQL include PHP and Java based web applications that require a database storage backend...EX : Joomla
- Cannot scale-out horizontally
- Virtually all Relational DB uses SQL

Session 3 : NoSQL Database(Types-Documents,Key-value,Graph based and Columnar)

Non-Relational DB/No-SQL DB :

- Non-Relational databases store data without a structured mechanism to link data from different

tables

- Require low cost hardware
- Much faster performance(read/write) compared to relational DB
- Horizontal scaling is possible
- Never provide table with flat fixed column records. It means schema-free
- Best suited for online analytical processing
- Ex : No-SQL Databases- MongoDB, Cassandra, DynamoDB, Postgresql, Raven, Redis

Types of No-SQL Databases :

1. Columnar DB : Cassandra, HBase
2. Document DB : MongoDB, CouchDB, RavenDB
3. Keyvalue DB : Redis, Riak, DynamoDB, Tokyo Cabinet
4. Graph DB : Neo4J, Flock DB

1. Columnar DB :

- It is a DBMS that stores data in columns instead of rows

ID	Name	Age	Bonus
1	Bob	30	3000
2	Alex	26	4000
3	Vijay	22	2000

- In a columnar DB, all the column1 values are physically together, followed by all the column2 values
- In a row oriented DBMS, the data would be stored like (1.Bob,30,3000.....2.Alex,26,4000.....ect)
- In a column based DBMS, the database would be stored like (1,2,3 ; Bob,Alex ; 30,26 ; 3000,4000....etc)
- Benefit is that because a column-based DBMS is self indexing, it uses less disk space than a RDBMS containing the same data
- It easily performs operations like Min, Max & Avg

2. Document DB :

- Document DB makes it easier for developers to store and query data in a DB by using the same document model format they use in their application
- Document DBs are efficient for storing catalogues
- Store semi-structured data as documents typically in JSON or XML format
- In the following example, a JSON-like document describes a book

```
[
  {
    "year" : 2010,
    "title" : "AWS Fundamentals",
    "info" : {
      "writer" : ['Rajput', 'Bhupinder'],
      "release date" : "2010-01-07",
      "rating" : 62,
      "genre" : ["it", "science"]
    }
  }
]
```

- A document database is a great choice for content management applications such as blogs and video platforms

3. Key-Value-DB :

- A key-value-db is a simple DB that uses an associative array (think of a dictionary) as a Fundamental model where each key associated with one and only one value in a collection
- It allows horizontal scaling
- Use cases : shopping cart, and session store in app like FB & Twitter
- They improve application performance by storing critical pieces of data in memory for low latency access
- Amazon elastic cache as an in-memory key-value stores

4.Graph Based DB :

- A graph DB is basically a collection of nodes and edges.Each node represent an entity(like person) and each edge represent a connection or relationship between two nodes

Session 4 : AWS Relational Database Services - Template Types,Storage Types and DB Engine Available

- In an AWS fully managed relational DB engines service where AWS is responsible for :
 - ☐ Security and patching
 - ☐ Automated Backup
 - ☐ Software updates for the DB engine
 - ☐ If selected, multi-AZ with synchronous replication between the active and standby DB instance
 - ☐ Automatic failover if multi-AZ option was selected
 - ☐ By default,every DB has weekly maintenance window(max 35 days)
- Settings managed by the users :
 - ☐ Managing DB settings
 - ☐ Creating relational database schema
 - ☐ Database performance tuning

Relational Database Engine Options :

- 1.MS SQL Server
- 2.My SQL → Support 64TB of DB
- 3.Oracle
- 4.AWS aurora → High Throughput
- 5.Postgre SQL → Highly Reliable & Stable
- 6.MariaDB → MySQL Compatible,64TB DB

There are two Licensing Options :

- 1.BYOL---Bring uour own license
- 2.License from AWS on hourly Basis

RDS Limits :

- Upto 40DB instances per account
- 10 of this 40 can be oracle or MS-SQL servers under License include model (or) under BYOL model,all 40 can be any DB engine you need

RDS Instance Storage :

- Amazon RDS use EBS volumes(not instance store) for DB and logs storage

1.General Purpose : Use for DB workloads with moderate I/O requirementLimits : Min 20GB....Max 16384GB

2.Provisional IOPS RDS Storage : Use for high performance OLTP workloads.....Limits : MIn

100GB....Max 16384GB.....This is recommend to use when we want to create multi-az environment for standby db

Templates Available in RDS :

- 1.Production* : Use defaults for high availability and fast,consistent performance
- 2.Dev/Test* : This instance is intended for development use outside of a production environment
- 3.Free-Tier* : Use free-tier to develop new app,test existing app,or gain hands on experience with Amazon RDS

DB Instance Size :

- 1.Standard Class* : Include M class
 - Max 96 VCPU....384GB Ram.....EBS 14000 MBPS
- 2.Memory Optimized Class* : Include r & x class
 - Max 96VCPU.....768GB ram....EBS 14000 MBPS
- 3.Burstable Class* : Include t class
 - Max 8VCPU....32GB Ram.....EBS 1500MBPS

Session 5 : AWS Relational Database Services - Concept of Multi-AZ in RDS

Multi-AZ in RDS :

- We can select Multi-AZ option during RDS DB instance launch
- RDS service creates a standby instances in a diff AZ in the same region,and configure "Synchronous Replication" between the primary and standby
- We can not read/write to the standby RDS DB instances directly
- We cannot select,which AZ in the region will be choosen to create the standby DB instance
- We can however view which AZ is selected after the standby is created
- Depending on the insatnce class,it may take 1 to few minutes to failover to the standby instance
- AWS recommends the use of provisioned IOPS instances for Muti-AZ RDS Instances

When Multi-AZ RDS Failover Triggers :

- 1.In case of failure of primary DB instance failure
- 2.In case of AZ failure
- 3.Loss of Network connectivity to primary DB
- 4.Loss of primary EC2 instance failure
- 5.EBS failure of primary DB instance
- 6.The primary DB instance or settings are changed
- 7.Patching the O.S of the DB instance
- 8.Manual failover(incase of rebooting)

Multi-AZ RDS Failover Consequences :

- 1.During failover,the CNAME of the RDS DB instance is updated to Map to the standby IP address
 - 2.It is recommended to use the endpoints(URL) to reference our DB instances and not its IP address
 - 3.The CNAME does not change,because RDS endpoint does not change region
- RDS endpoint does not change by selecting Multi-AZ option.However the primary and standby instances will have different Ip addresses,as they are in diff AZ
 - It is always recommended that we do not use the IP address to point RDS instances,always use endpoint.By using endpoint,there will be no change whenever a failover happens

When we do Manual Failover :

- In case of rebooting
- This is by selecting the “reboot with failover” reboot options on the primary RDS DB instances
- A DB instance reboot is required for changes to take effect when we change the DB parameter group or when we change a static DB parameter

Session 6 : AWS Relational Database Services - AWS RDS Backup and Retention Period

- Whenever failover occurs, AWS RDS sends SNS notification
- We can use API calls to find out the RDS events occurred in the last 14 days
- Even, we can use CLI to view last 14 days events
- Using AWS console, we can view only last one day events
- In case of OS-patching, system upgrades, DB scaling, these things happen on standby first, then on primary to avoid outage
- In multi-AZ, snapshots and automated backups are done on standby instance to avoid I/O suspension on primary

RDS Multi-AZ Deployment-Maintenance

- Firstly, perform maintenance on standby
- Now, convert standby into primary so that maintenance can be done on primary (currently)
- We can manually upgrade a DB instance to a supported DB engine version from AWS console as follows : RDS---->DB instance---->Modify DB----->Set DB Engine Version
- By default, change will take effect during the next maintenance window
- Or we can force an immediate upgrade if we want
- In multi-az, version upgrade will be conducted on both primary and standby at the same time, which will cause an outage
- So do it during maintenance window

Methods to Backup :

- There are 2 methods to backup and restore our RDS DB instances
 1. AWS RDS automated backup
 2. User Initiated manual backup
- Either we can take backup of entire DB instance or just the DB
- We can create a storage volume snapshots of our entire DB instances

Automated Backup

- Automated backups by AWS backup our DB data to multiple-AZ to provide for data durability
- Select -Automated Backup in AWS console
- Stored in Amazon S3
- Multi-AZ automated backups will be taken from the standby instance
- The DB instance must be in “Active” state for automated backup
- RDS automatically backs up the DB instance daily, by creating a storage volume snapshot of our DB instance (fully daily snapshot) including the DB transaction logs
- We can decide, when we would like to take backup (window)
- No additional charge for RDS backing up our RDS instances
- For multi-AZ deployment, backups are taken from the standby DB instance (True for MariaDB, MySQL, Oracle, PostgreSQL)
- Automated backups are deleted when we delete our RDS DB instance
- An outage occurs if we change the backup retention period from zero to non-zero value or the other way around
- Retention period of automated backup is 7 days (By default) via AWS console but AWS Aurora is

an exception its default is one day

- Via CLI or API, 1 day by default
- We can increase it upto 35 days
- If we do not want backup, put zero in retention period

Session 7 : AWS Relational Database Services - RDS Encryption, Manual Backup and Billing

- In case of manual snapshot, point-in-time recovery is not possible
- Manual snapshot is also stored in S3
- They are not deleted automatically, if we delete RDS instance
- Take a final snapshot before deleting our RDS DB instance
- We can store manual snapshot directly with other AWS account
- When we restore a DB instance, only the default DB parameters and security groups are associated with the Restored instance
- We cannot restore a DB snapshot into an existing DB instance rather it has to create a new DB instance. It has new endpoint
- Restoring from a backup as a DB snapshot changes the RDS instance endpoint
- At the time of restoring, we can change the storage type (general purpose or provisioned)

RDS Encryption :

- We cannot encrypt an existing unencrypted DB instance
- To do that, we need to :
 - create a new, encrypted instance and migrate our data to it (from unencrypted to encrypted) (OR)
 - We can restore from a backup/snapshot into a new encrypted RDS instance
- RDS supports encryption at rest for all DB engines using KMS

What actually encrypted when data at rest :

1. All its snapshots
2. Backups of DB (S3 Storage)
3. Data on EBS volume
4. Read replica created from the Snapshot

Points related to RDS Billing :

- No upfront cost
- We have to pay only for :
 - DB instances hours (partial hours chargeable)
 - Storage GB/Month
 - Internet data transfer
 - Backup storage (i.e S3)... This increases by increasing DB backups retention period

Aws also Charge for :

- Multi-AZ DB hours
- Provisioned storage (Multi-AZ)
- Double write I/O
- We are not charged for DB data transfer doing replication from primary to standby

LABS :

Session 8 : Accessing MySQL DB instance from Linux

