What is a Database?

A database is an organized collection of data that can be easily accessed, managed, and updated. It is used to store information like customer details, sales records, employee data, product catalogs, etc.

Databases are usually managed by Database Management Systems (DBMS), which provide tools to store, retrieve, and manipulate the data.

Types of Databases

Databases can be classified based on how they store, organize, and access data.

1. Relational Databases (RDBMS)

- Store data in tables (rows and columns)
- Use Structured Query Language (SQL) to manage data
- Examples:
  - Oracle Database
  - Microsoft SQL Server
  - MySQL
  - PostgreSQL

2. NoSQL Databases

- Used for unstructured or semi-structured data
- Don't use fixed table schemas
- Types of NoSQL databases:
  - Document-based (e.g., MongoDB)
  - Key-Value (e.g., Redis)
  - Column-family (e.g., Cassandra)
  - Graph (e.g., Neo4j)

3. In-Memory Databases

- Store data in RAM for ultra-fast access
- Suitable for real-time processing
- Examples: Redis, Memcached

4. Cloud Databases

- Hosted on cloud platforms like Azure, AWS, or GCP
- Highly scalable and managed (e.g., Azure SQL, Amazon RDS)

5. Object-Oriented Databases

- Store data in the form of objects (like in object-oriented programming)
- Support features like inheritance, polymorphism
- Example: db4o

6. Distributed Databases

- Data is stored across multiple locations or servers
- Can be replicated or partitioned
- Improves availability and scalability

7. Graph Databases

- Store data in nodes and edges to represent relationships
- Excellent for social networks, recommendation engines
- Example: Neo4j

Introduction to Neo4j

Neo4j is a **graph database** that is designed to store, query, and manage highly connected data efficiently. Unlike traditional relational databases, Neo4j uses **nodes** and **relationships** to represent and store data.

Key Features of Neo4j

- **Graph-based Storage**: Stores data as nodes (entities) and relationships (connections) rather than tables.
- **Cypher Query Language (CQL)**: A powerful and easy-to-read query language similar to SQL but optimized for graphs.
- **Schema-free Model**: No fixed schema required, allowing flexible data representation.
- **High Performance for Connected Data**: Optimized for traversing relationships efficiently.
- **ACID Compliance**: Ensures reliability, consistency, and integrity of transactions.
- **Built-in Indexing**: Uses native indexes for faster queries.
- **Scalability**: Supports horizontal and vertical scaling.

Components of Neo4j

1. **Nodes**: Represent entities (e.g., Person, Product, City).
2. **Relationships**: Define connections between nodes (e.g., FRIENDS_WITH, PURCHASED).
3. **Properties**: Key-value pairs stored in nodes and relationships.

4. **Labels**: Categorize nodes into types (e.g., User, Product).
5. **Indexes & Constraints**: Improve search performance and enforce data rules.

Basic Cypher Queries

Creating Nodes

```
CREATE (p:Person {name: 'Alice', age: 30})
```

Creating Relationships

```
MATCH (a:Person {name: 'Alice'}), (b:Person {name: 'Bob'})

CREATE (a)-[:FRIENDS_WITH]->(b)
```

Retrieving Data

```
MATCH (p:Person) RETURN p
```

Updating Data

```
MATCH (p:Person {name: 'Alice'})

SET p.age = 31

RETURN p
```

Deleting Nodes

```
MATCH (p:Person {name: 'Alice'})

DETACH DELETE p
```

When to Use Neo4j?

- **Social Networks**: Managing relationships between users.
- **Recommendation Systems**: Suggesting products, friends, or content.
- **Fraud Detection**: Identifying suspicious connections.
- **Network & IT Management**: Analyzing infrastructure dependencies.
- **Knowledge Graphs**: Storing and linking large datasets.

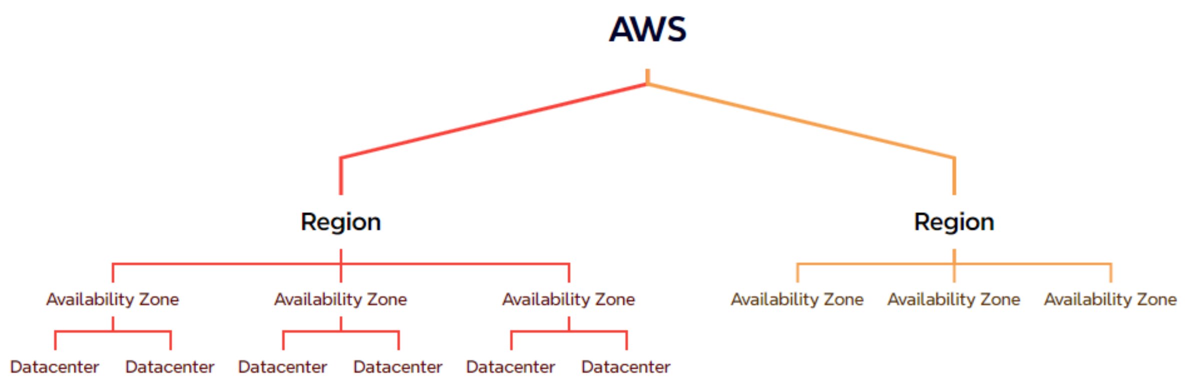
Getting Started with Neo4j

1. **Install Neo4j**: Download from neo4j.com
2. **Use Neo4j Browser**: Interactive web-based UI to execute Cypher queries.
3. **Try Neo4j Sandbox**: Cloud-hosted environment to experiment with.
4. **Explore Neo4j Aura**: Fully managed cloud database for production use.

Conclusion - Neo4j is a powerful tool for handling complex and connected data structures. With its graph-based approach, intuitive Cypher language, and scalability, it is widely used across various industries for analytical and operational applications.

AWS Cloud

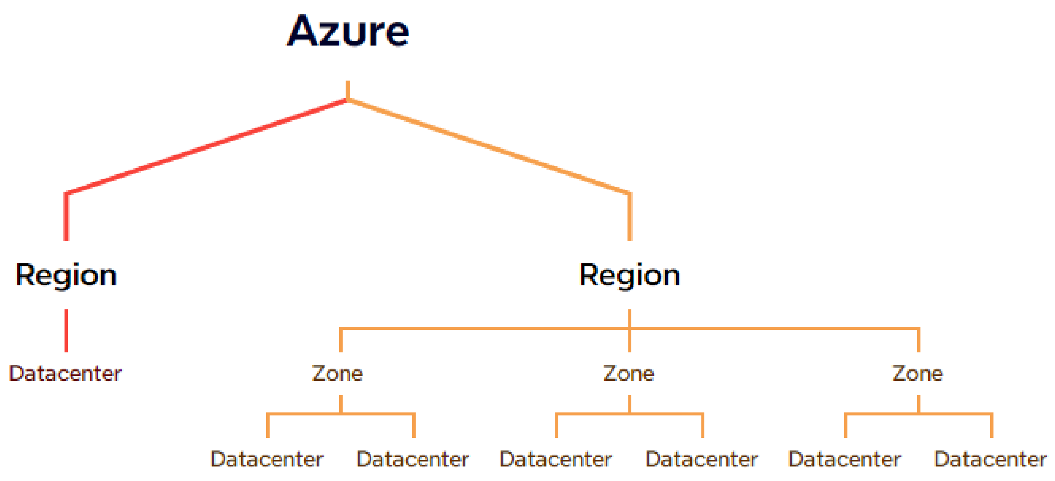
- Global Infra:
  - Regions
  - Availability Zones
  - Global Network



Azure Cloud

- Global Infra:

  - Regions
  - Zones
  - Backbone Network



- Refer Here for exploring datacenters of azure

Cloud Services

- Service: This is what is offered by Cloud Providers. Each service is metered and billing is generally pay as you go.
- Categories:
  - Compute
  - Storage
  - Network
  - Databases
  - AI/ML
  - DevOps
  - Big Data Services
- Each Service will have
  - billing
  - different options
  - Automation options
- On a Generic Note every cloud provides the following approaches to create infra
  - Website
    - AWS: Console
    - Azure: Portal
  - Command Line
    - AWS CLI
    - Azure CLI & Azure Powershell
  - Client Libraries/SDK
    - AWS SDK:
      - python: boto3
      - AWS SDK for Java ...
    - Azure SDK
  - Template
    - AWS Cloudformation
    - Azure ARM Template (Bicep)
    - Terraform (DevOps)
- Personal Accounts : We will be using free tier accounts for learning cloud.
  - By default you have admin access.
- Enterprise Accounts:
  - Landing Zones:
    - Governance rules are established
    - Roles established
    - Generally we will have multiple accounts
    - Templates
  - User Accounts
- Roles:
  - Solutions Architect
  - Developer
  - Cloud Engineer (Ops Engineer)

Activities around Databses

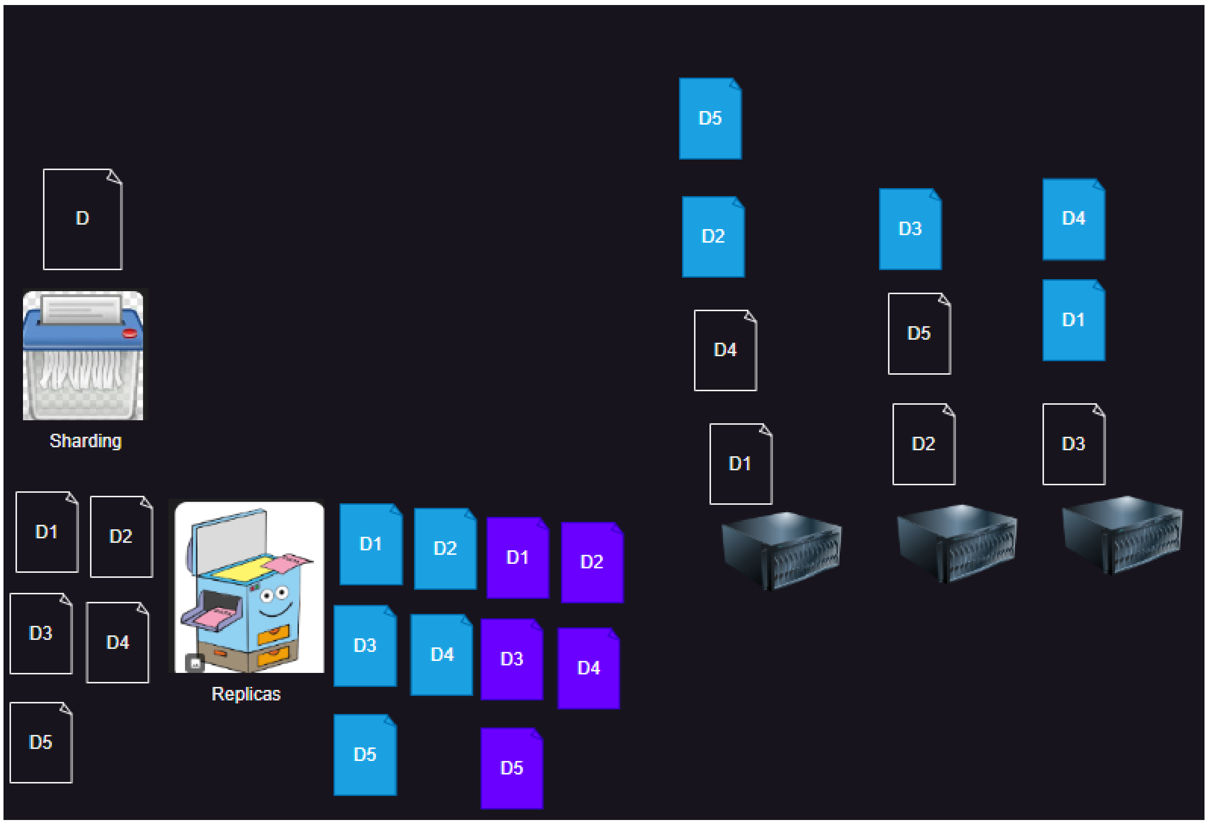
- Data Modelling

- Capacity Planning
- BCDR (Business Continuity and Disaster Recovery)
  - Backups
  - Recoveries
- Patching:
  - OS
  - DBMS Updates
- Storage needs
  - Expanding sizes
  - Archiving
- Migration

Databases offered by AWS

- RDS (Relational Database Services):
  - Engines:
    - mysql
    - PostgreSQL
    - Microsoft SQL Server
    - Oracle
    - IBM DB2
    - Aurora
      - mysql
      - postgresql
- DynamoDB (NOSQL)
- Document DB (Mongo DB)
- KeySpaces (Cassandra)
- QLDB (Quantum Ledger Database)
- Timestream:
  - IoT Devices
- Neptune (Graph Databases)
- Elastic Cache (Cache Databases)
- Memory DB (Cache Database)
- NOSQL Databases can scale horizontally using many techniques

- Shards and replicas



Databases on Azure

- Azure SQL (Microsoft SQL Server)
  - Ledger functionality
- Azure SQL for Postgres
- Azure SQL for mysql
- Azure Cosmos DB
  - Document
  - Key Value
  - Column Datbases
  - Graph
- Azure Cache

Database as a Service

- Database as a service is offering from cloud providers where we get
  - Database directly (rather than installing)
  - Options for
    - automated backups
    - replications
    - patching
    - failovers
- Drawbacks:
  - not all versions are supported

  - some features might not be supported.

- Advantages:

  - Minimum Adminstration
  - Easy configurations

- DBA Responsibilities (On-prem):

  - Backups and Recoveries
  - Replications
  - Patching
  - Performance Tuning

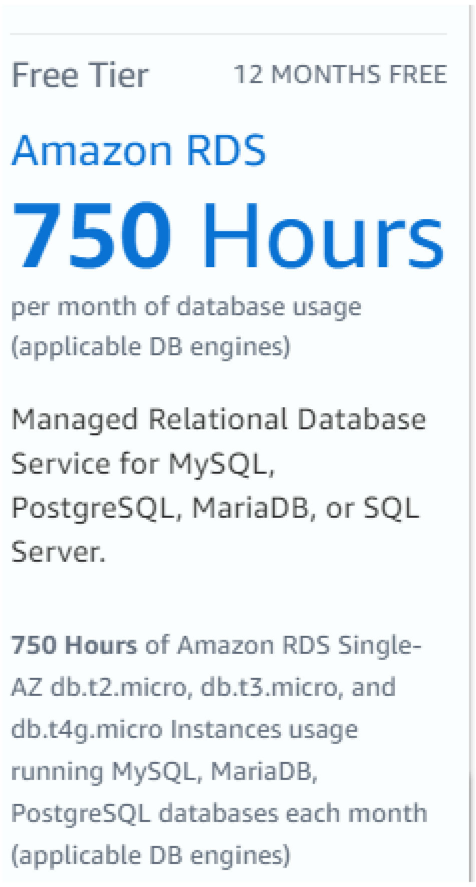
- With cloud databases or Database as a Service there is only one option partially left for DBA's i.e. Performance Tuning

- refer primary index, secondary index, full table scans.

AWS Relational Database Services (RDS)

- RDS is Database as a service offered by AWS supporting following database engines
  - Microsoft SQL Server
  - mySQL
  - PostgreSQL
  - Oracle
  - IBM DB2

- Free plan: (db.t2.micro/db.t3.micro)



Free Tier 12 MONTHS FREE

Amazon RDS

750 Hours

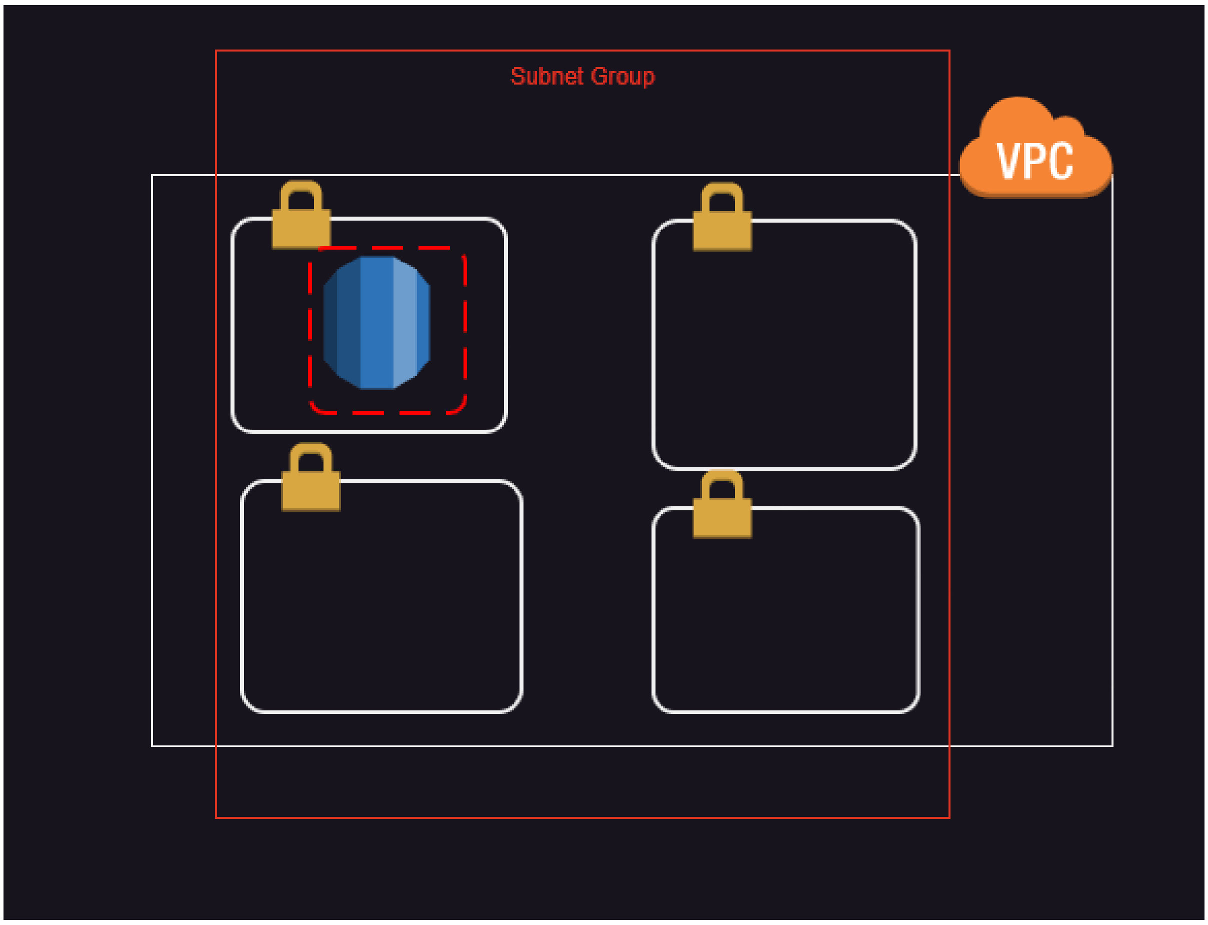
per month of database usage (applicable DB engines)

Managed Relational Database Service for MySQL, PostgreSQL, MariaDB, or SQL Server.

750 Hours of Amazon RDS Single-AZ db.t2.micro, db.t3.micro, and db.t4g.micro Instances usage running MySQL, MariaDB, PostgreSQL databases each month (applicable DB engines)

- RDS instances require
  - vpc
  - subnet group

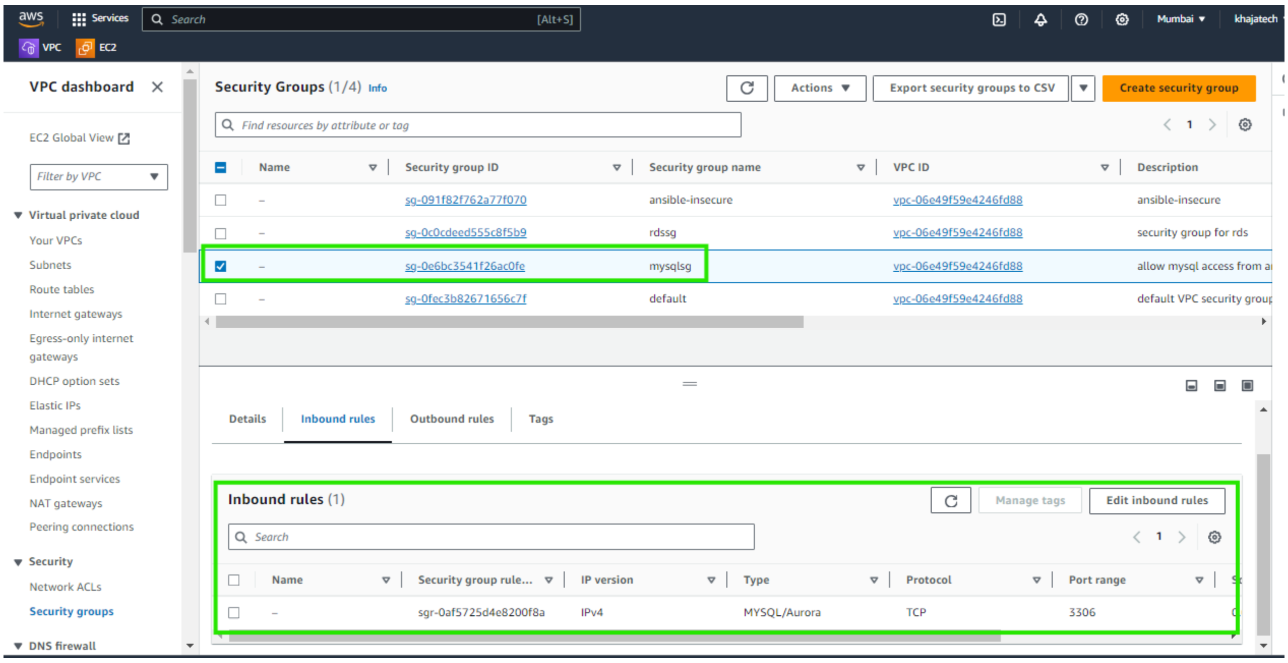
- security group



- RDS instances are sizes by db instance types

Create a free mysql database in mumbai region

- Create a security group



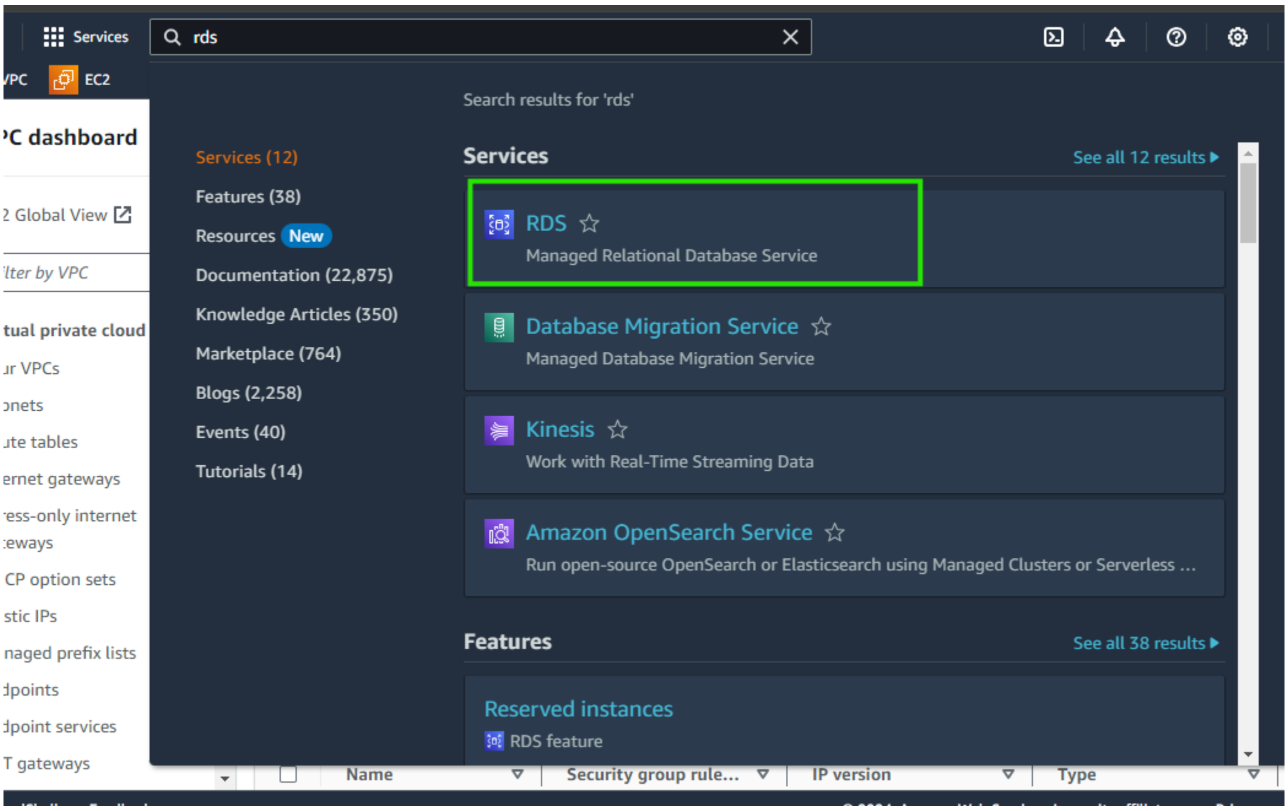
Security Groups (1/4) Info

Name	Security group ID	Security group name	VPC ID	Description
–	sg-091f82f762a77f070	ansible-insecure	vpc-06e49f59e4246fd88	ansible-insecure
–	sg-0c0cdeed555c8f5b9	rdssg	vpc-06e49f59e4246fd88	security group for rds
–	sg-0e6bc3541f26ac0fe	mysqlsg	vpc-06e49f59e4246fd88	allow mysql access from a
–	sg-0fec3b82671656c7f	default	vpc-06e49f59e4246fd88	default VPC security grou

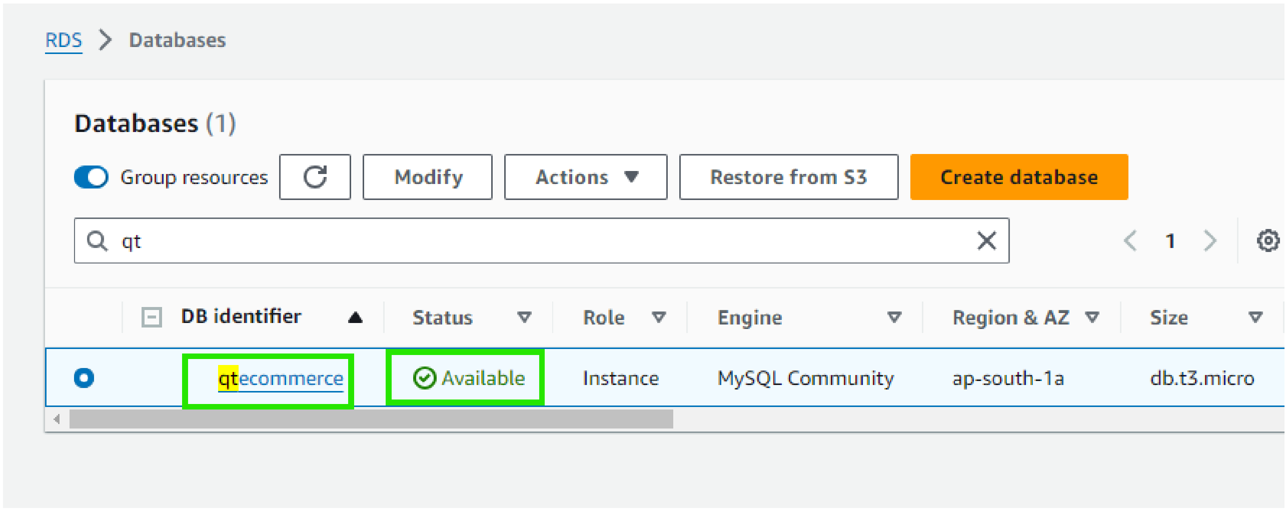
Inbound rules (1)

Name	Security group rule...	IP version	Type	Protocol	Port range
–	sgr-0af5725d4e8200f8a	IPv4	MYSQL/Aurora	TCP	3306

- Navigate to RDS



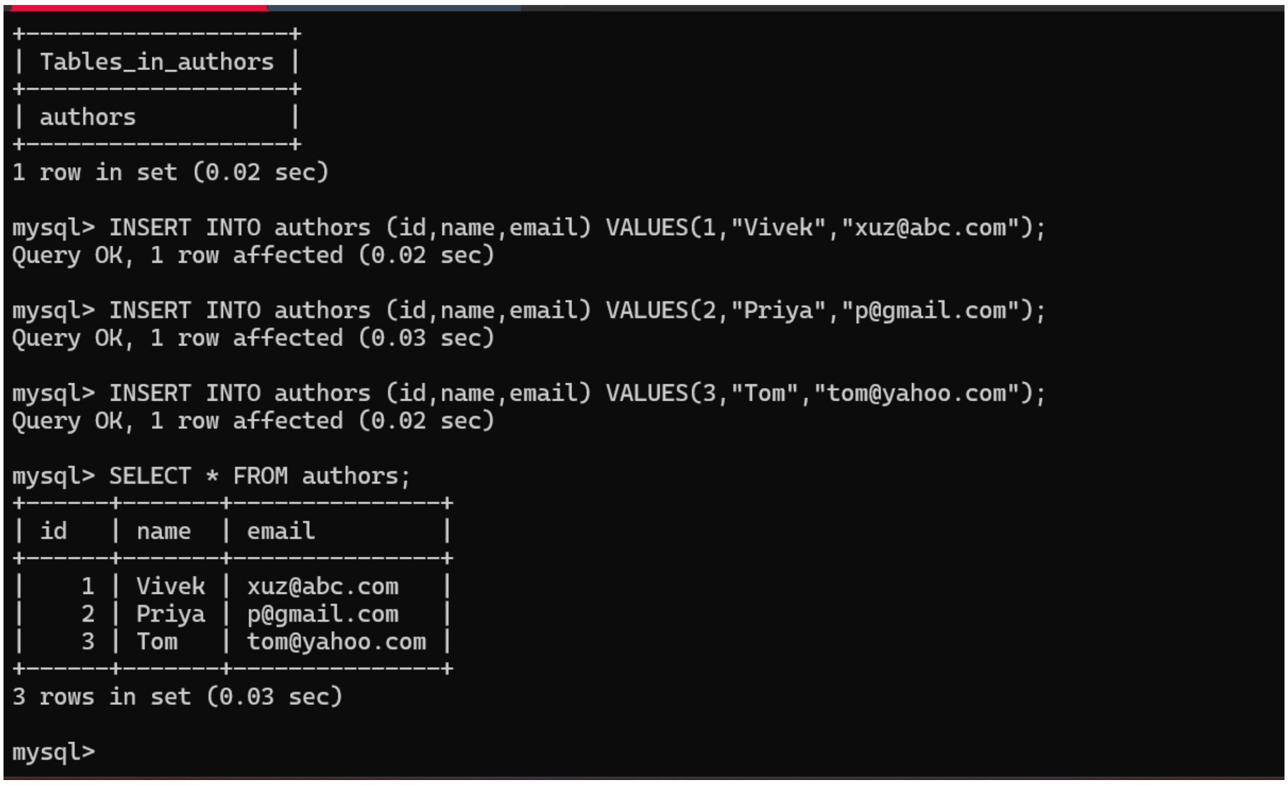
- For screenshots refer classroom video



DB identifier	Status	Role	Engine	Region & AZ	Size
qtecommerce	Available	Instance	MySQL Community	ap-south-1a	db.t3.micro

- As discussed we can use mysql client or azure data studio

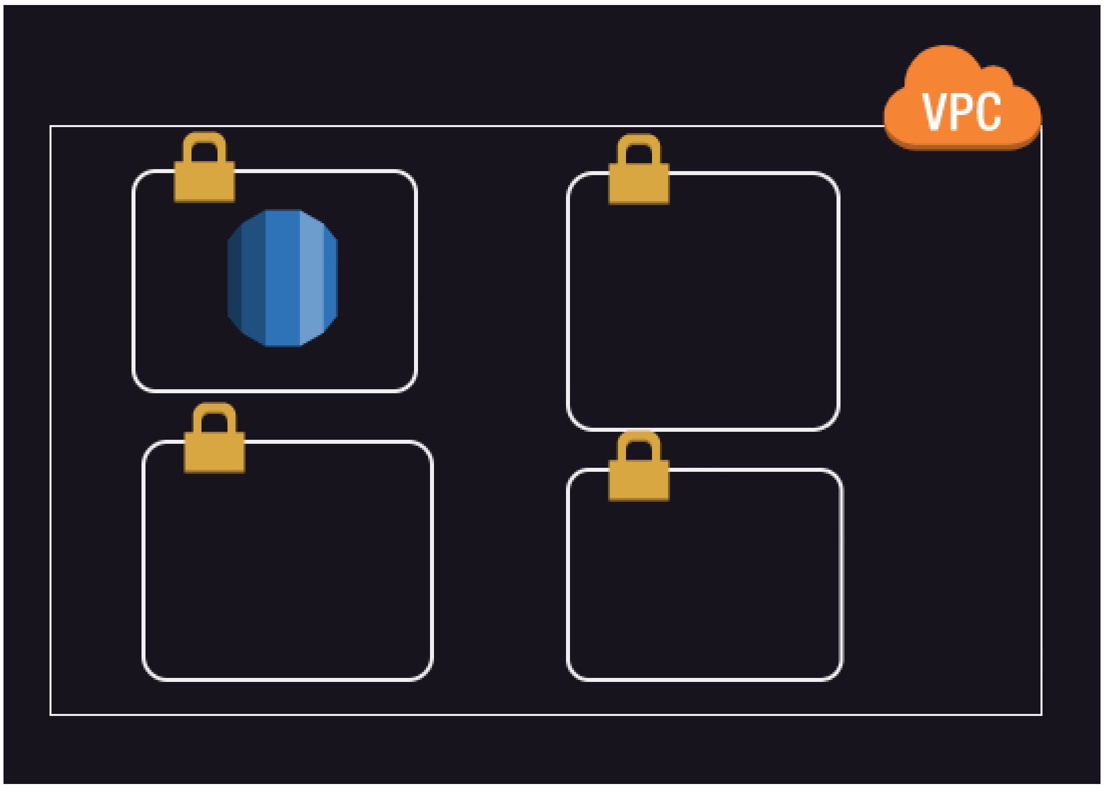
- Refer Here for creating a table and inserting data.



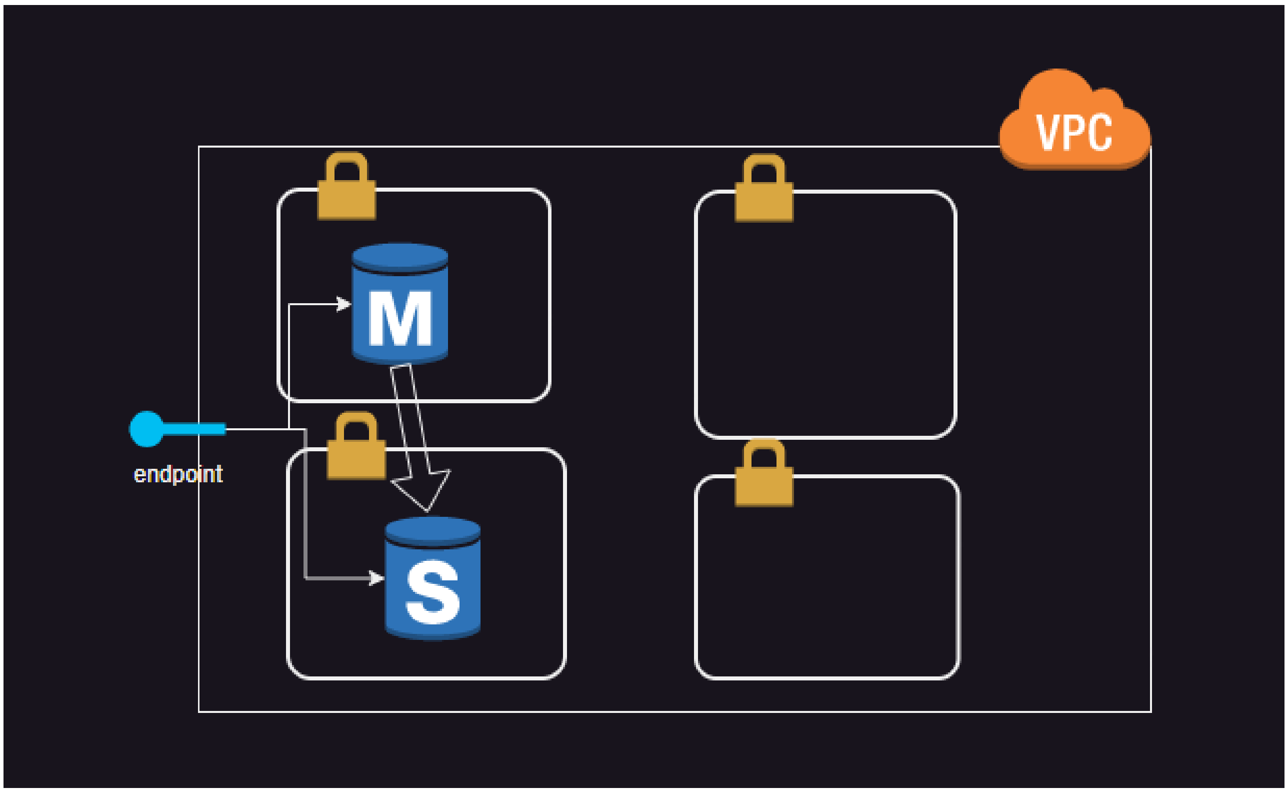
```
+-------------------+
| Tables_in_authors |
+-------------------+
| authors           |
+-------------------+
1 row in set (0.02 sec)

mysql> INSERT INTO authors (id,name,email) VALUES(1,"Vivek","xuz@abc.com");
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO authors (id,name,email) VALUES(2,"Priya","p@gmail.com");
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO authors (id,name,email) VALUES(3,"Tom","tom@yahoo.com");
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM authors;
+------+-------+---------------+
| id   | name  | email         |
+------+-------+---------------+
|    1 | Vivek | xuz@abc.com   |
|    2 | Priya | p@gmail.com   |
|    3 | Tom   | tom@yahoo.com |
+------+-------+---------------+
3 rows in set (0.03 sec)

mysql>
```

AWS Database Deployment Options

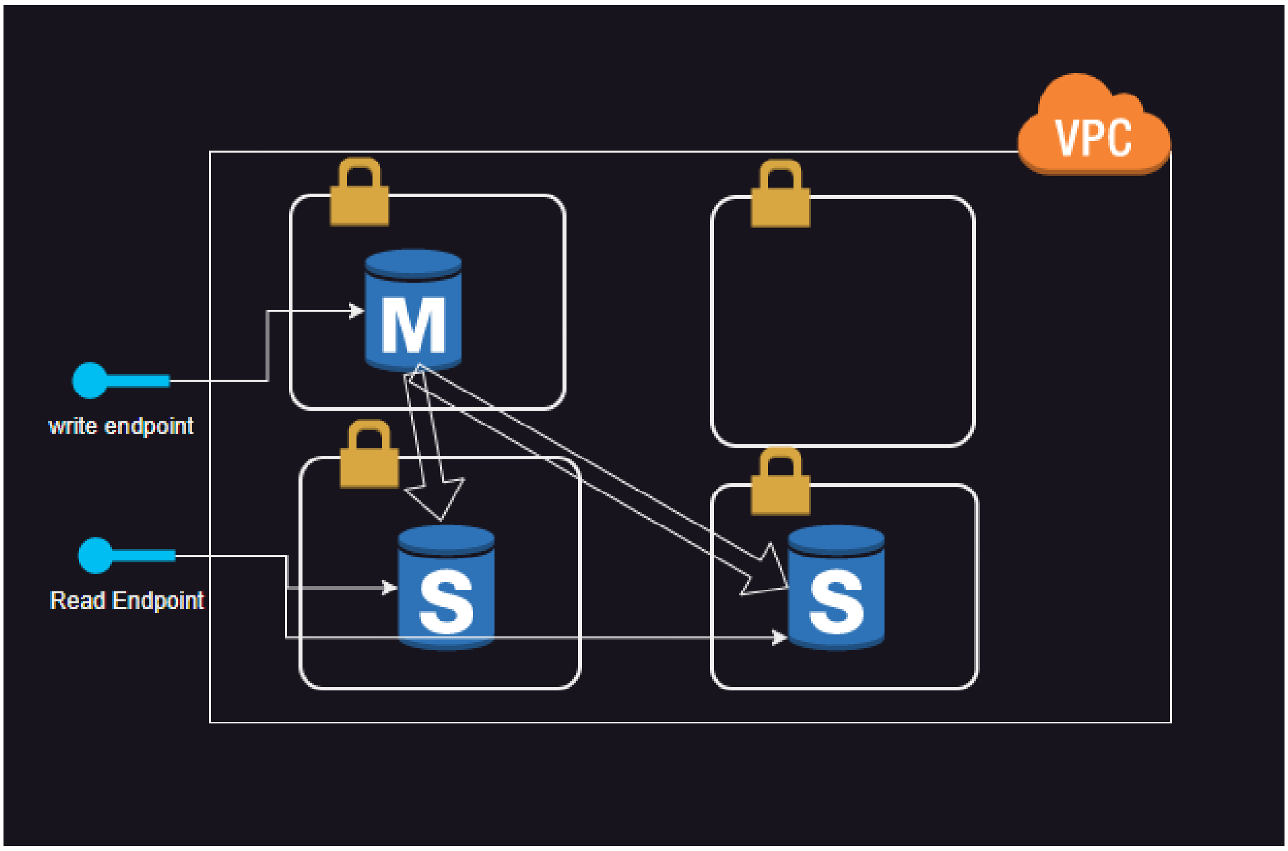
- Single Database



- Multi AZ Database



- Multi Cluster Database



AWS Database Instance Sizes

- By Purpose
  - Standard
  - Memory Optimized
  - Compute Optimized

    - Burstable
- Size Refer Here

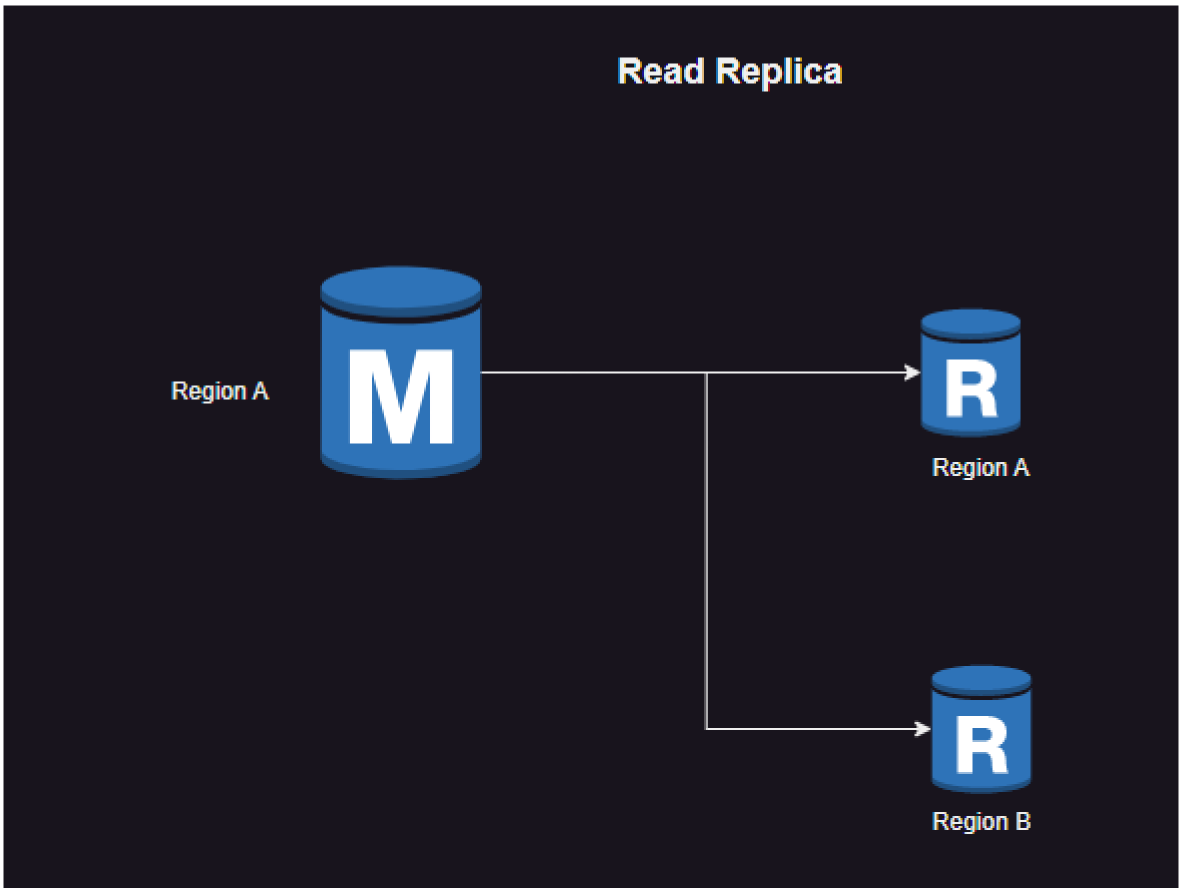
Backups

- Manual Backups can be taken to create snapshots
- Automated Backups can be enabled.
    - PITR (Point in time restore)

AWS RDS (Contd..)

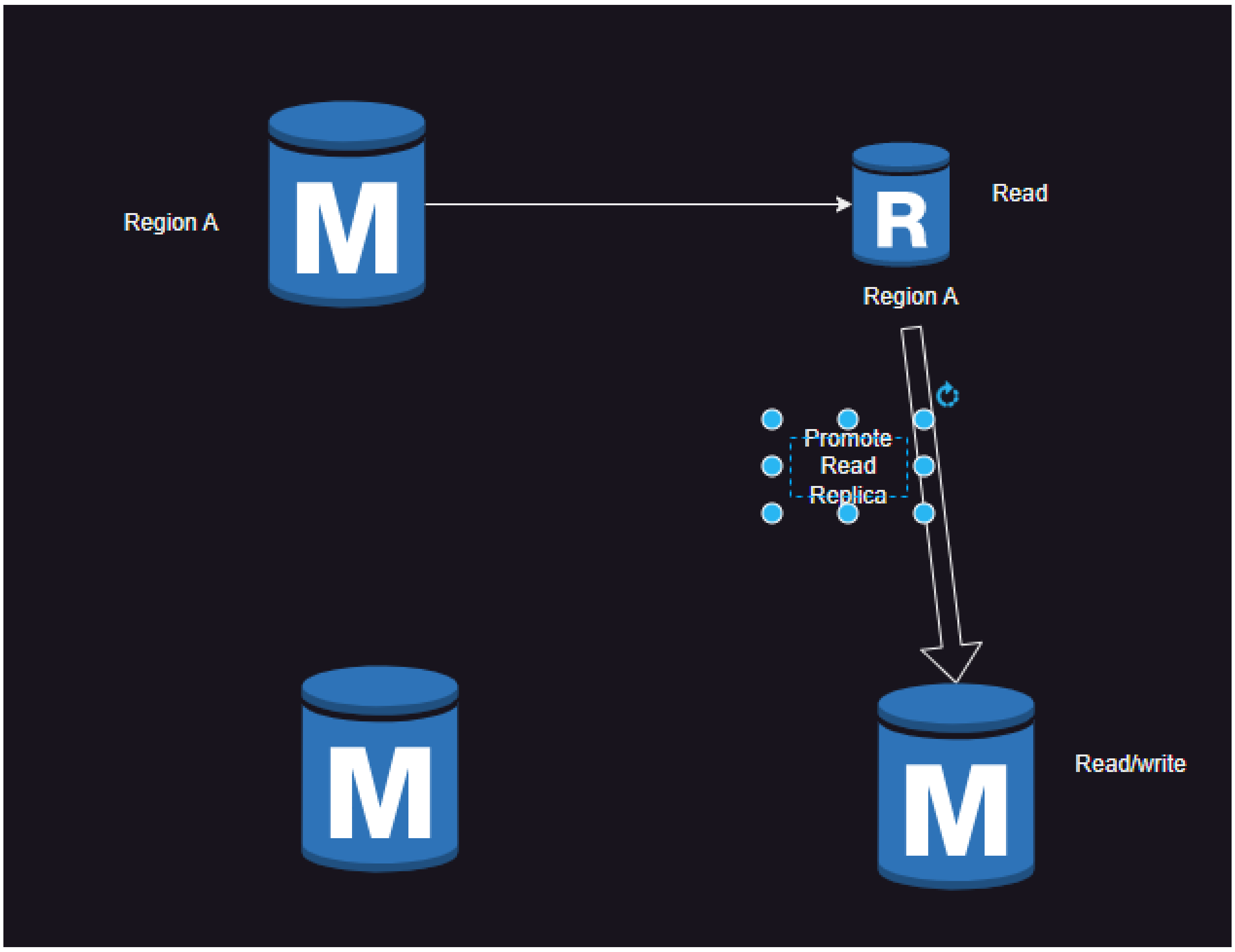
Read Replicas

- AWS RDS allows to create read replicas

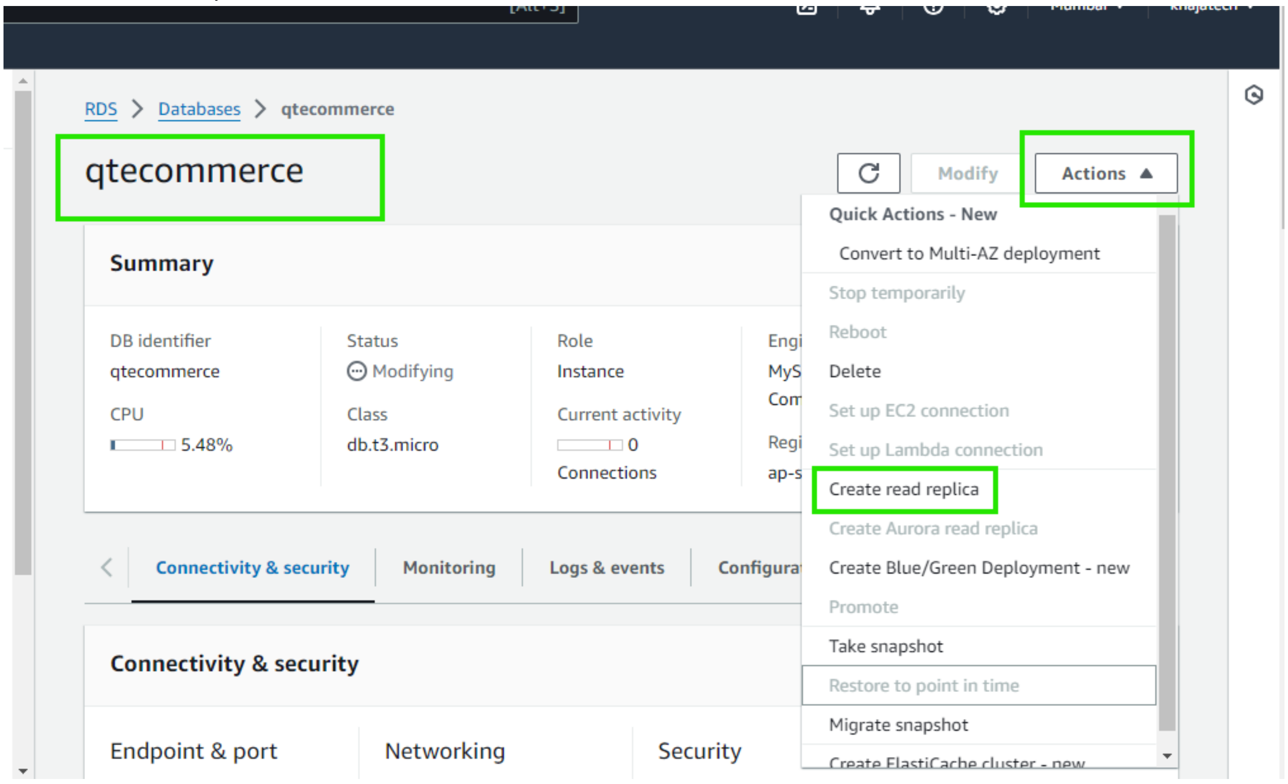


- Read Replicas can be in the same/different regions

- Read Replicas can be promoted to accept write requests but the replication will no longer continue



- Ensure the Backups are on (atleast for 1 day)
- Create a Read Replica


RDS > Databases > qtecommerce

qtecommerce

Modify
Actions ▲

Summary

DB identifier
qtecommerce

CPU
5.48%

Status
Modifying

Class
db.t3.micro

Role
Instance

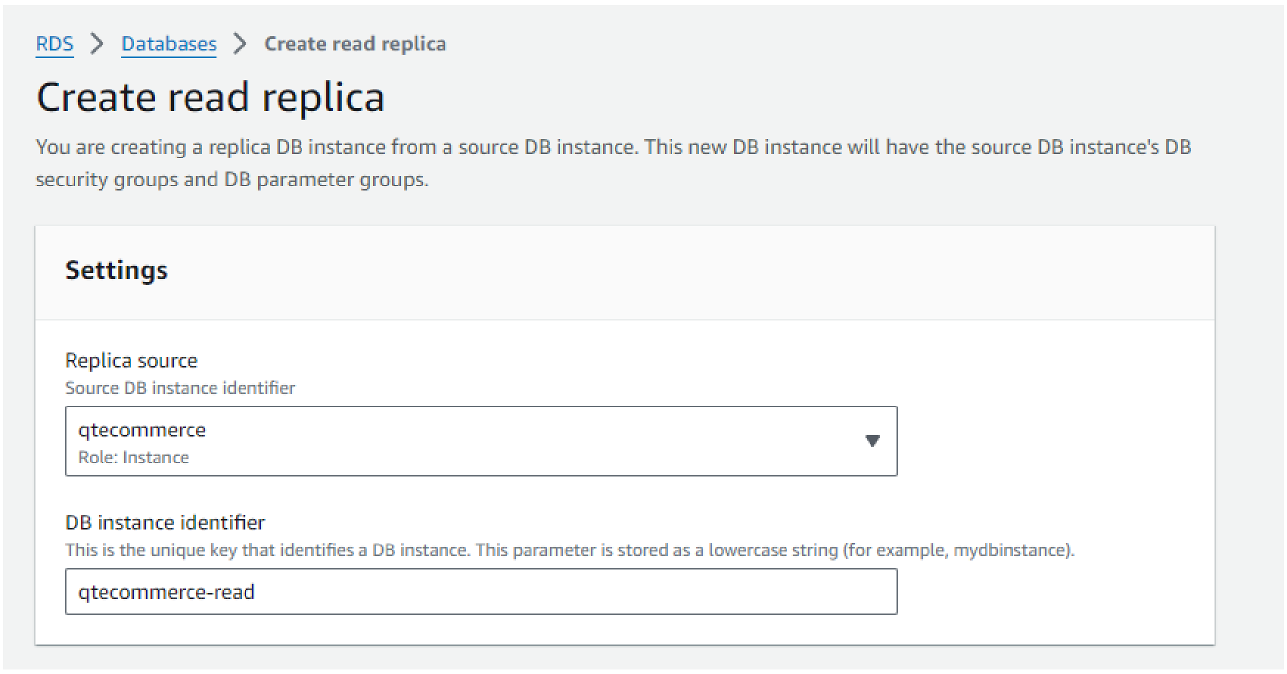
Current activity
0
Connections

Quick Actions - New
Convert to Multi-AZ deployment
Stop temporarily
Reboot
Delete
Set up EC2 connection
Set up Lambda connection
Create read replica
Create Aurora read replica
Create Blue/Green Deployment - new
Promote
Take snapshot
Restore to point in time
Migrate snapshot

Connectivity & security | Monitoring | Logs & events

Connectivity & security

Endpoint & port | Networking | Security



RDS > Databases > Create read replica

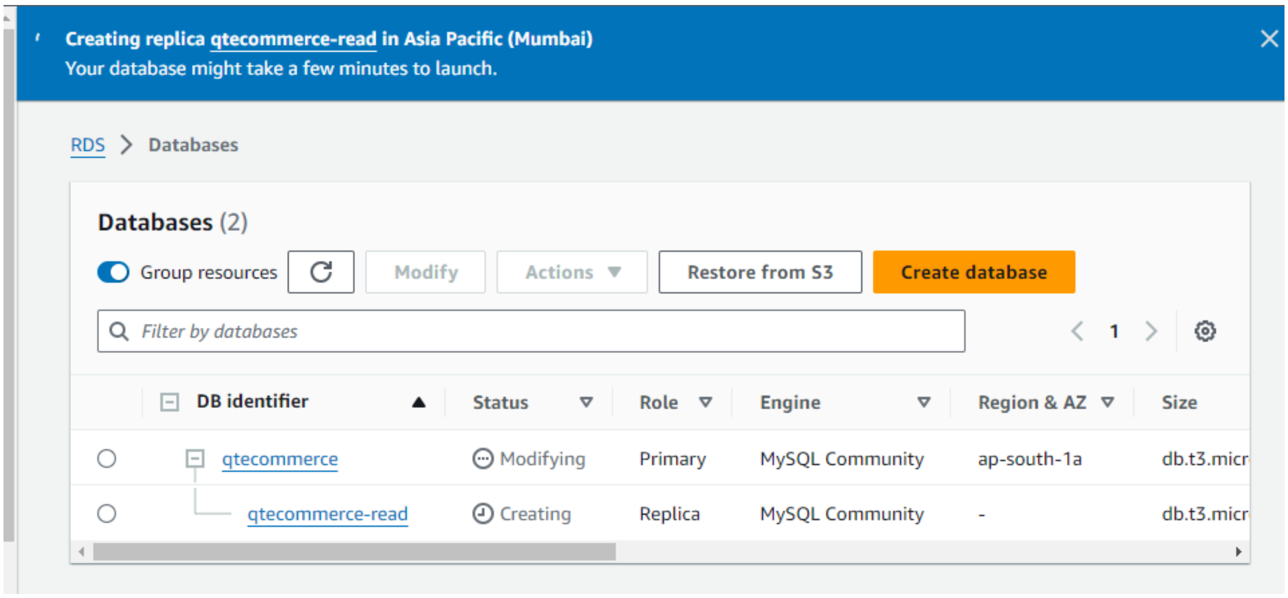
Create read replica

You are creating a replica DB instance from a source DB instance. This new DB instance will have the source DB instance's DB security groups and DB parameter groups.

Settings

Replica source
Source DB instance identifier
qtecommerce
Role: Instance

DB instance identifier
This is the unique key that identifies a DB instance. This parameter is stored as a lowercase string (for example, mydbinstance).
qtecommerce-read



Creating replica qtecommerce-read in Asia Pacific (Mumbai)
Your database might take a few minutes to launch.

RDS > Databases

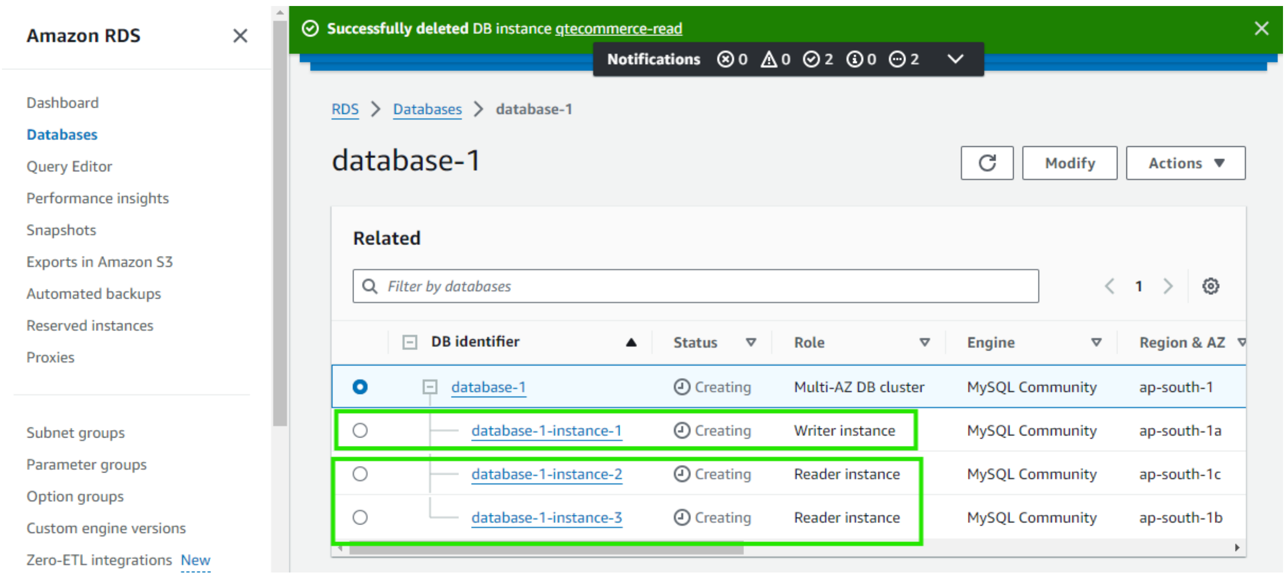
Databases (2)

DB identifier	Status	Role	Engine	Region & AZ	Size
qtecommerce	Modifying	Primary	MySQL Community	ap-south-1a	db.t3.micr
qtecommerce-read	Creating	Replica	MySQL Community	-	db.t3.micr

- We have experimented with read replicas in the class watch video

Multi Cluster DB Creation

- Overview



Successfully deleted DB instance qtecommerce-read

Notifications ⊗0 ⚠0 ✓2 ⓘ0 ⊖2

Amazon RDS

- Dashboard
- Databases
- Query Editor
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies
- Subnet groups
- Parameter groups
- Option groups
- Custom engine versions
- Zero-ETL integrations New

RDS > Databases > database-1

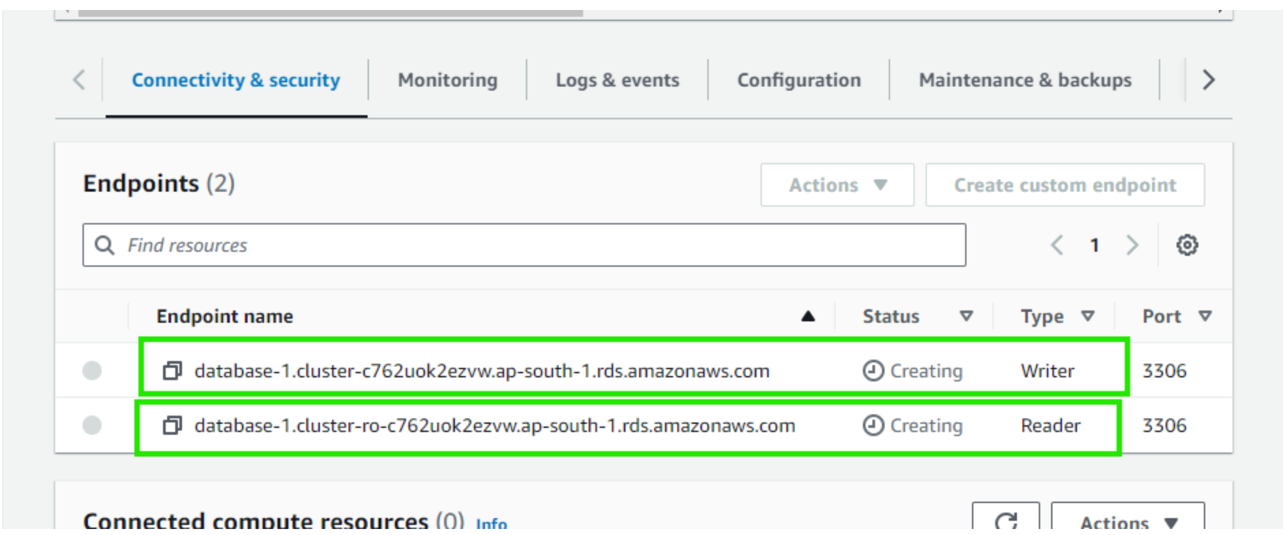
database-1

Modify | Actions ▼

Related

Filter by databases

	DB identifier	Status	Role	Engine	Region & AZ
●	database-1	Creating	Multi-AZ DB cluster	MySQL Community	ap-south-1
○	database-1-instance-1	Creating	Writer instance	MySQL Community	ap-south-1a
○	database-1-instance-2	Creating	Reader instance	MySQL Community	ap-south-1c
○	database-1-instance-3	Creating	Reader instance	MySQL Community	ap-south-1b



Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups

Endpoints (2)

Actions ▼ | Create custom endpoint

Find resources

	Endpoint name	Status	Type	Port
○	database-1.cluster-c762uok2ezvw.ap-south-1.rds.amazonaws.com	Creating	Writer	3306
○	database-1.cluster-ro-c762uok2ezvw.ap-south-1.rds.amazonaws.com	Creating	Reader	3306

Connected compute resources (0) Info

Actions ▼

Aurora

- Architecture Preview

Exercise

- Create a mysql rds single database (db.t2.micro) with 2 days of backup
- Create a read replica
- check the connectivity
- delete both instances

Amazon Dynamo DB

- To the Dynamo DB Table, we add indexes
- Indexes help in querying data
- There are two types of indexes
  - local secondary indexes (can be added during creation time only)
  - Global Secondary indexes (can be added post creation)

Amazon Document DB

- Mongo DB as a service offered by AWS
- Document DB gives us options to run mongo much like RDS
- We have to deal with
  - security groups
  - subnet groups
  - vpcs
- Once created we can use mongoshell or application libraries to connect.

Other Databases

- Cassandra ==> KeySpaces
- Graph => Neptune

Cache

- Redis
  - OSS => Elastic Cache
  - Enterprise => Memory DB
- Memcached => Elastic Cache

Database Backups in AWS

- Snapshot refers to a backup which is generally executed manually.
- AWS has a service to centralize all backups which is AWS Backup Service
- To Use AWS Backup Service, We need to
  - Create a vault
  - Backup plan
  - Protected Resources: Resources for which backup is enabled.

Backups in Azure

-

Azure has Backup Center which indirectly creates Backup Vaults

Create a Backup for any Service

DataWarehouses

- AWS Redshift: This is Datawarehouse as a service offered by AWS
- Azure Synapse: This is Datawarehouse as a service offered by Azure

Big Data

- As we have lots of data in the non-queryable format, to use this big data technology has started
- Hadoop
- Spark
- Hive
- Cloud has started giving a storage called as blob storages (s3 and azure storage account).
- Organizations have started adopting blob storages as data lakes
- Organization have started building pipelines for data ingestion, storage and analytics
- Treating data as code and building data pipelines.