1

# Section 9 : Databases & Analytics

## # Databases Intro

- Sorting data on disk (EFS, EBS, EC2 Instan store, S3) can have its limits.

- Sometimes, you want to store data in db.

- You can structure the data.

- You build indexes to efficiently query/ search through the data

- You define relationships between your ~~databases~~. datasets.

- Databases are optimized for a purpose and come with different features, shapes and constraints.

# Databases & Shared Responsibility on AWS.

- AWS offers use to manage different databases

- Benefits include:

1] Quick Provisioning, High availability, Vertical & Horizontal Scaling

2] Automatic Backup & Restore, Operations, upgrades

3] Operating System Patching is handled by AWS.

4] Monitoring, alerting

- Note :- many databases technologies could be run on EC2, but you must handle yourself the resiliency, backup, patching, high availability, fault tolerance, scaling.-

# Amazon RDS Overview

- RDS stands for Retional Database Service
- It's 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

1] Postgres
2] MySQL
3] MariaDB
4] Oracle
5] Microsoft SQL Server
6] IBM DB2
7] Aurora (AWS Proprietary database)

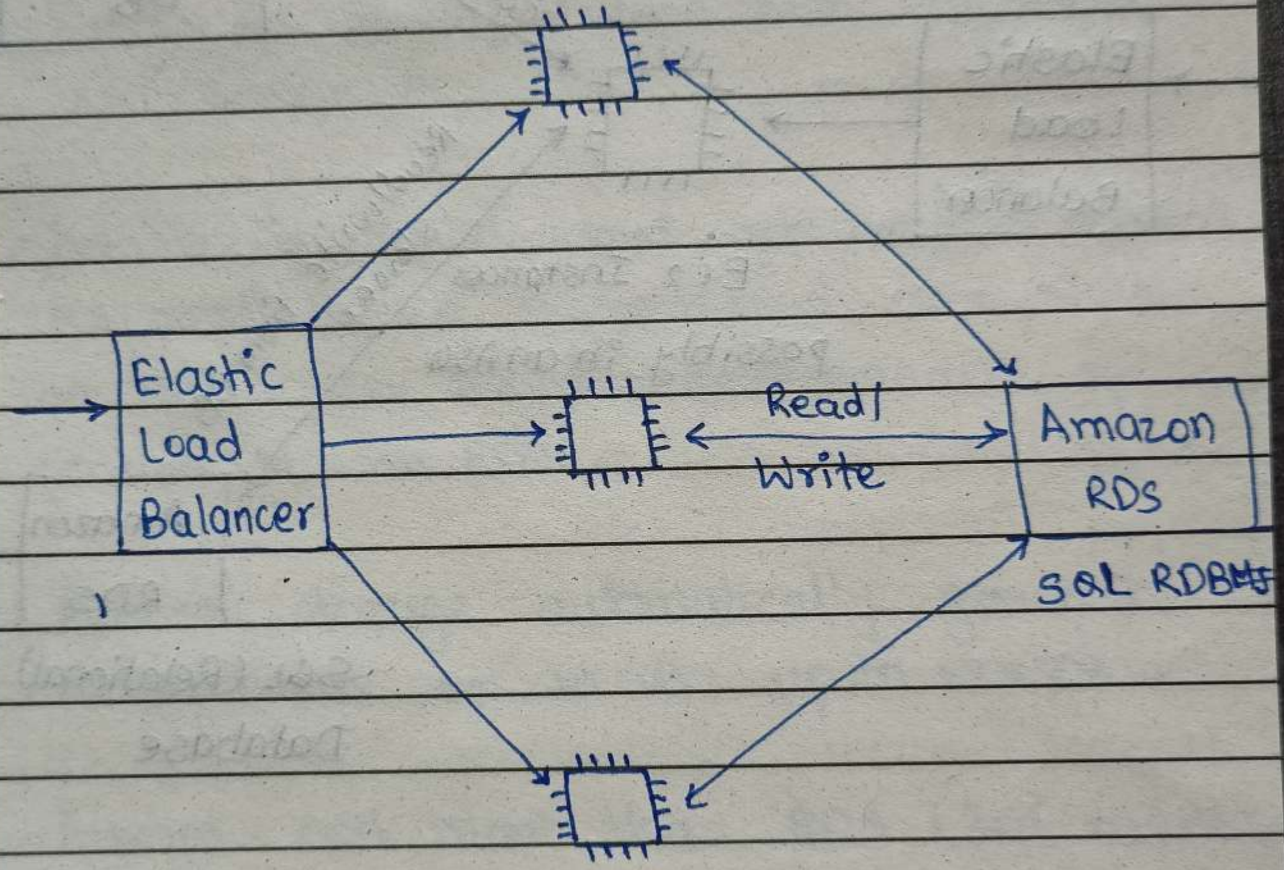
# Advantages over using RDS versus Deploying DB on EC2.

* RDS is managed service:

1] Automated provisioning, OS patching
2] Continuous backups & restore to specific timestamp (Point in Time Restore)!
3] Monitoring dashboards.
4] Read replicas for improved read performance.
5] Multi AZ setup for DR (Disaster Recove
6] Maintenance windows for upgrades
7] Scaling capabilities (vertical & Horizontal)
8] Storage backed by EBS.

- BUT you can't SSH into your instances

# * RDS Solution Architecture



EC2 Instances
Possibly in an ASG

# Amazon Aurora (Relational DB)

- Aurora is a proprietary technology from AWS (not open sourced)

- PostgreSQL & MySQL are both supported as Aurora DB.

- Aurora is "AWS cloud optimized" & claims 5x performance improvement over MySQL on RDS, over 3x the performance op postgresQL on RDS.

- Aurora storage automatically grows in increaments op 10 GB, up to 128 TB.

- Aurora cost more than RDS (20% more) -but its more eppicient

- Not in the free tier

# Amazon Aurora serverless

- Automated database instantiation & auto-scaling based on actual usage.

- PostgreSQL & MySQL are both supported as Aurora Serverless DB.

- No capacity planning needed.

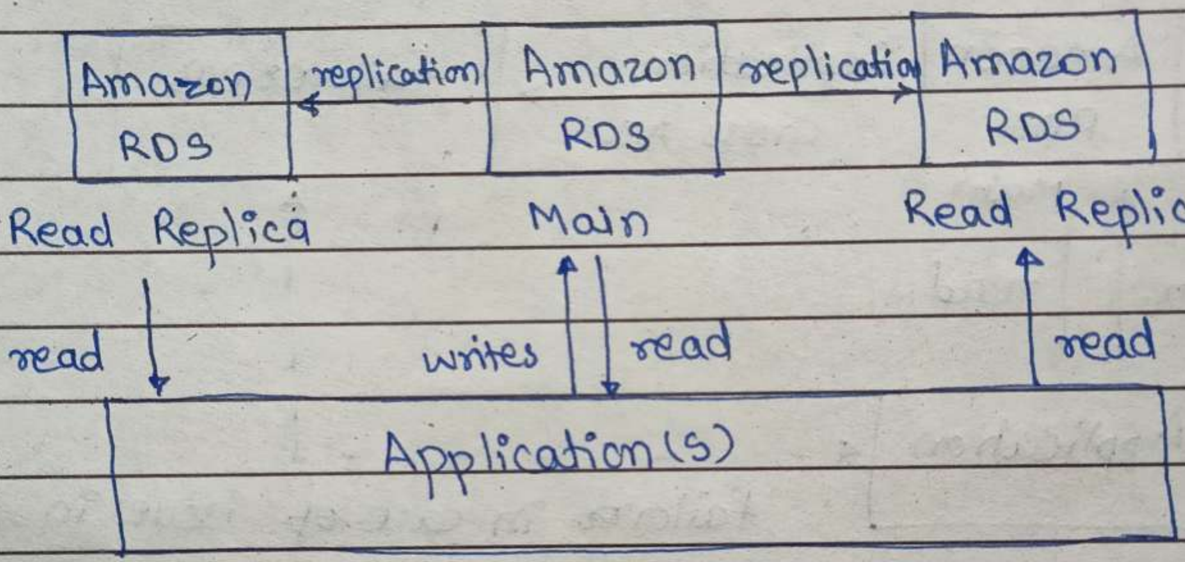
- Least management overhead.

- Pay per second, can be more cost-effective

- Use cases: good for infrequent, intermittent or unpredictable workloads.....

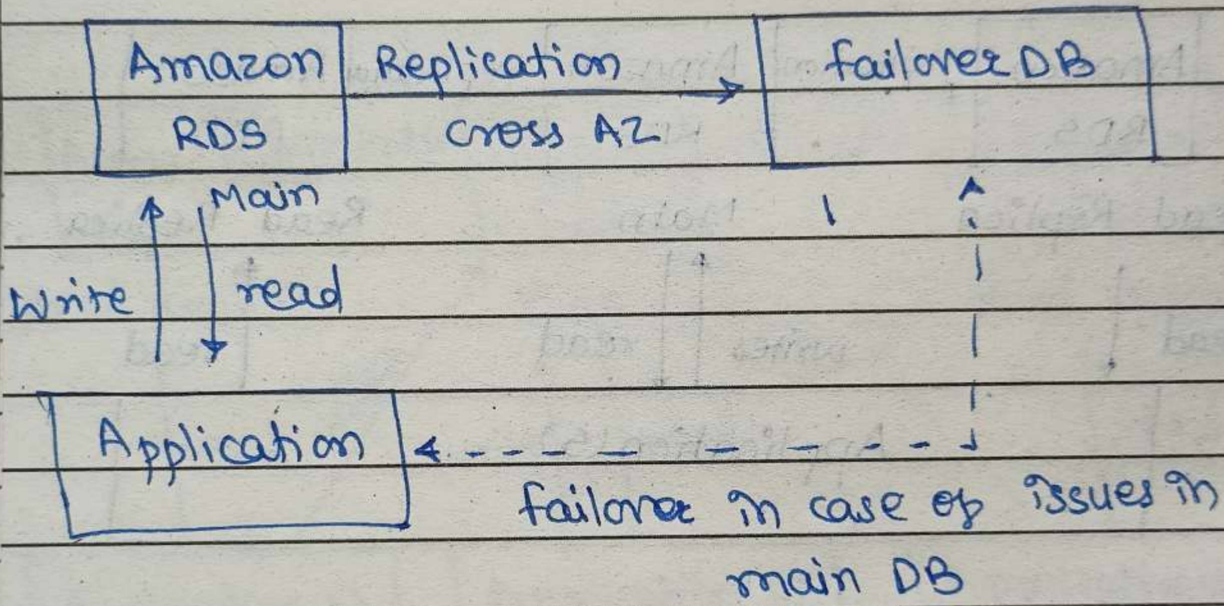
# RDS Deployments : Read Replicas, Multi-Az

- Read Replicas:
  - Scale the read workload of your DB.
  - Can create upto 15 Read Replicas.
  - Data is only written to the main DB.



- Multi-AZ :

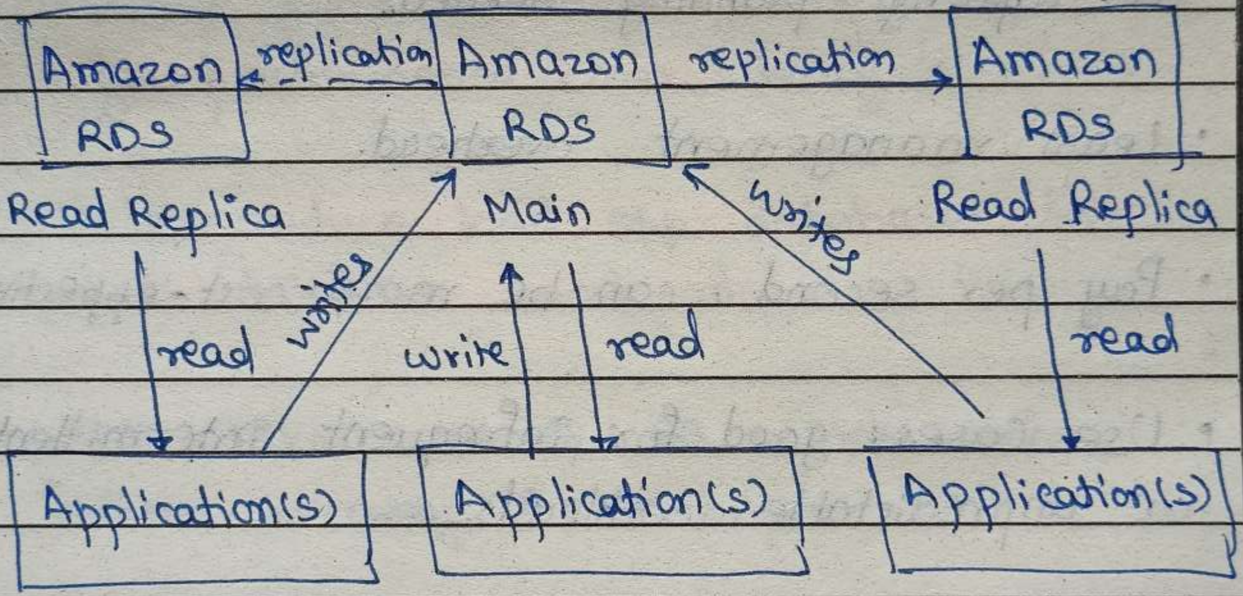
- failover in case of AZ outage (high availability)



- Data is only read / written to the main database.
- Can only have 1 other AZ as failover.

# # RDS Deployments : Multi- Region

+ Multi - Region (Read Replicas)
  - Disaster recovery in case op region issue
  - local performance for global reads.
  - Replication cost



It's used for OLTP.

# Amazon Elasticache Overview

- The same way RDS is ~~go~~ to get managed Relational Databases....

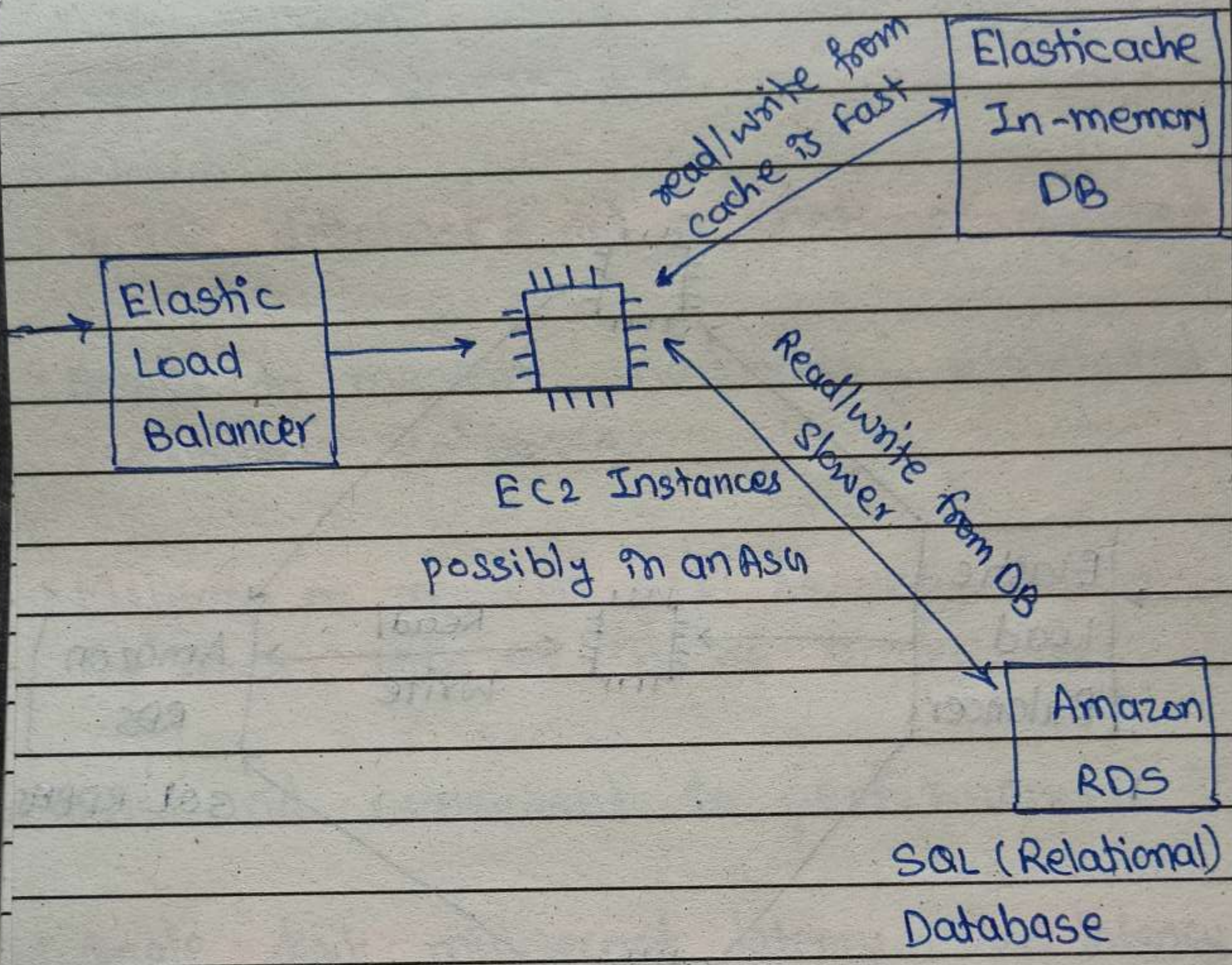
- ElastiCache is to get managed Redis or Memcached.

- Caches ~~g~~ are in-memory databases with high-performance, low latency.

- Helps reduce load off databases for read intensive workloads.

- AWS takes care of OS maintenance, failure recovery & backups

* Elasticache Solution Architecture - Cache



# DynamoDB

- Fully managed Highly available with replication across 3 AZ.

- NoSQL database - not a relational database

- Scales to massive workloads, distributed "serverless" database.

- Millions of requests per seconds, trillions of rows, 100s of TB of storage

- Fast and consistent in performance.

- Single digit millisecond latency - low latency retrieval.

- Integrated with IAM for security, authorization & administration.

- Low-cost & auto scaling capabilities.

- Standard & Infrequent Access (IA) Table Class.

DynamoDB - type op data

- DynamoDB is a key/value database

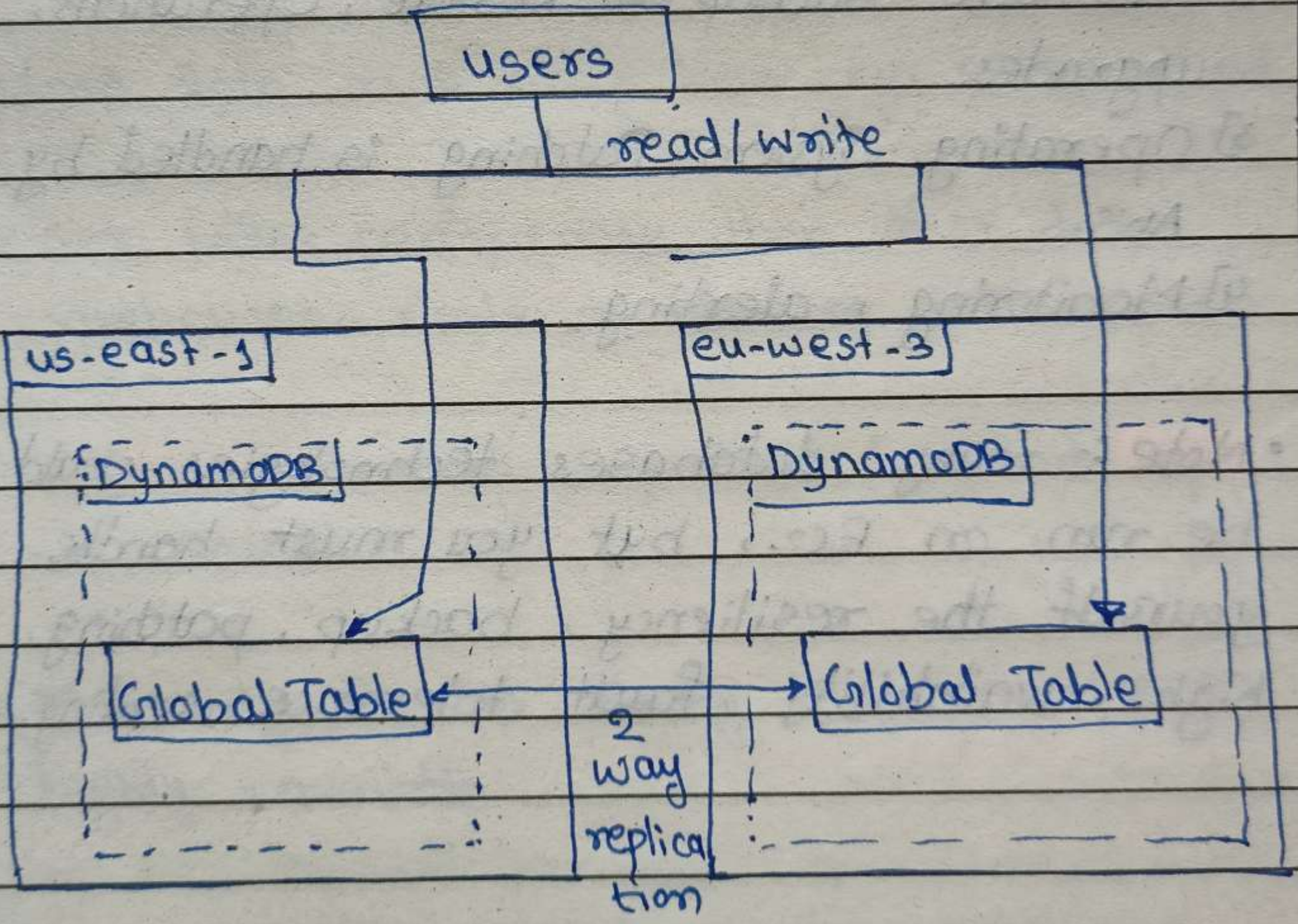
# DynamoDB Accelerator - DAX

- fully Managed in-memory cache for DynamoDB
- 10x Performance improvement - single-digit millisecond latency to microseconds latency-when accessing your DynamoDB tables.
- Secure, highly scalable & highly available
- Difference with ElastiCache at the CCP level: DAX is only used for and is integrated with DynamoDB, while Elasticache can be used for other databases.

# DynamoDB - Global Tables

- Make a DynamoDB table accessible with low latency in multiple-regions.

- Active-Active replication (read/write to any AWS region)


users
read/write
us-east-1
DynamoDB
Global Table
eu-west-3
DynamoDB
Global Table
2 way replica tion