Training Course

Amazon Web Service

Module 8:

AWS Fundamentals
RDS, Aurora, ElastiCache

Module 8: RDS, Aurora, ElastiCache

- **Goal**: Understanding Database in AWS
  - ✓ RDS
  - ✓ Aurora

  Lab: Create and configuring database

AWS RDS Overview

- 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
  - Postgre
  - MySQL
  - MariaDB
  - Oracle
  - Microsoft SQL Server
  - Aurora (AWS Proprietary database)

Advantage over using RDS versus deploying DB on EC2

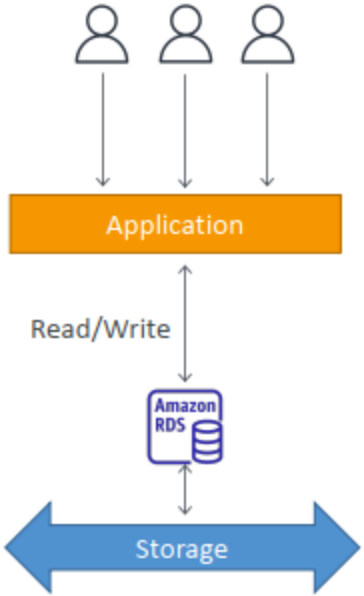
- RDS is a managed service:
  - Automated provisioning, OS patching
  - Coninuous backups and restore to specific timestamp (Point in Time Restore)!
  - Monitoring dashboards
  - Read replicas for improved read performance
  - Multi AZ setup for DR (Disaster Recovery)
  - Maintenance windows for upgrades
  - Scaling capability (Vertical and horizontal)
  - Storage backed by EBS (gp2 or io1)
- BUT you can't SSH into your instances

RDS Backups

- Backups are automatically enabled in RDS
- Automated backups:
  - Daily full backup of the database (during the maintenance window)
  - Transaction logs are backed-up by RDS every 5 minutes -> ability to restore to any point in time (from oldest backup to 5 minutes ago)
  - 7 days retention (can be increased to 35 days)
- DB Snapshots:
  - Manually triggerd by the user
  - Retention of backup for as long as you want

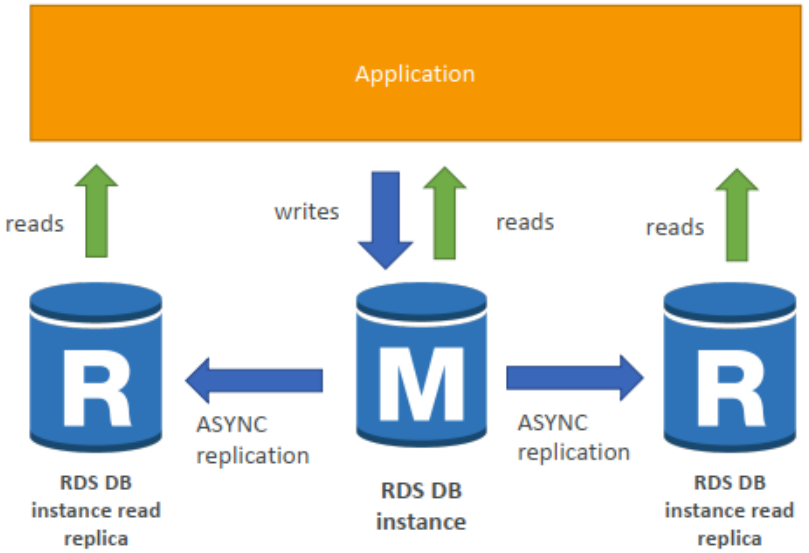
RDS – Storage Auto Scaling

- Helps you increase storage on your RDS DB instance dynamically
- When RDS detects you are running out of free database storage, it scales automatically
- Avoid manual scaling your database storage
- You have to set Maximum Storage Threhold (maximum limit for DB storage)
- Useful for applications with unpredictable workloads
- Supports all RDS database engines (MariaDB, MySQL, PostgreSQL, SQL Server, Oracle)



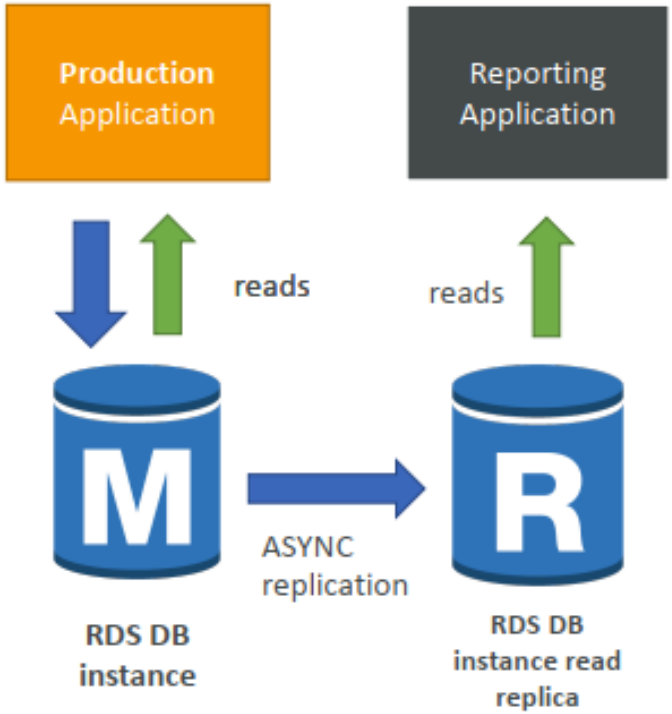
RDS Read Replicas for read scalability

- Up to 5 Read Replicas
- Within AZ, Cross AZ or Cross Region
- Replication is ASYNC, so reads are eventually consistent
- Replicas can be promoted to their own DB
- Applications must update the connection string to leverage read replicas



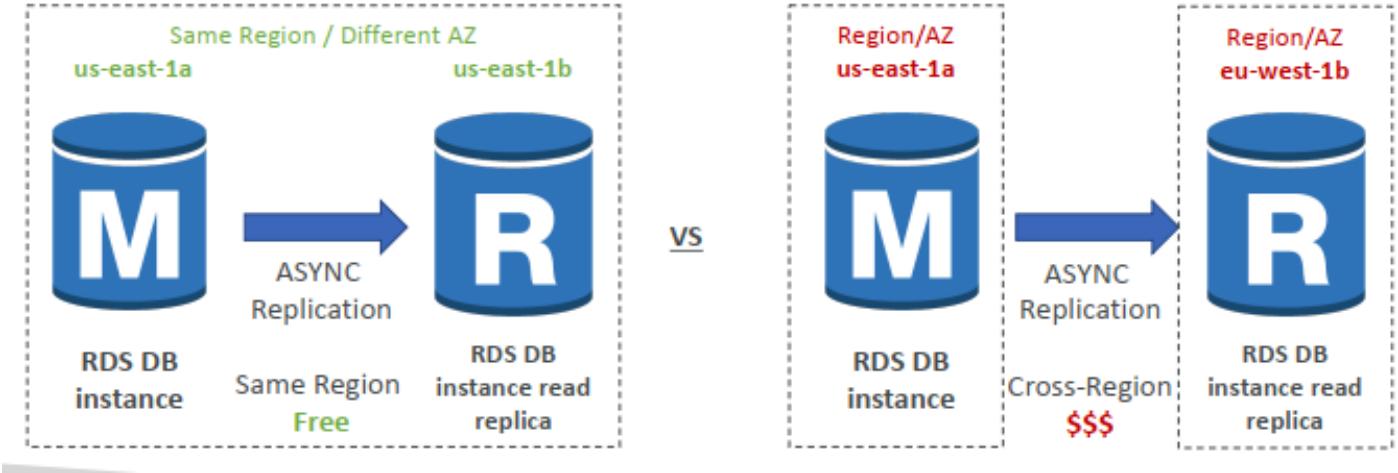
RDS Read Replicas – Use Cases

- You have a production database that is taking on normal load
- You want to run a reporting application to run some analytics
- You create a Read Replica to run the new workload there
- The production application is unaffected
- Read replicas are used for SELECT (=read) only kind of statements (not INSERT, UPDATE, DELETE)



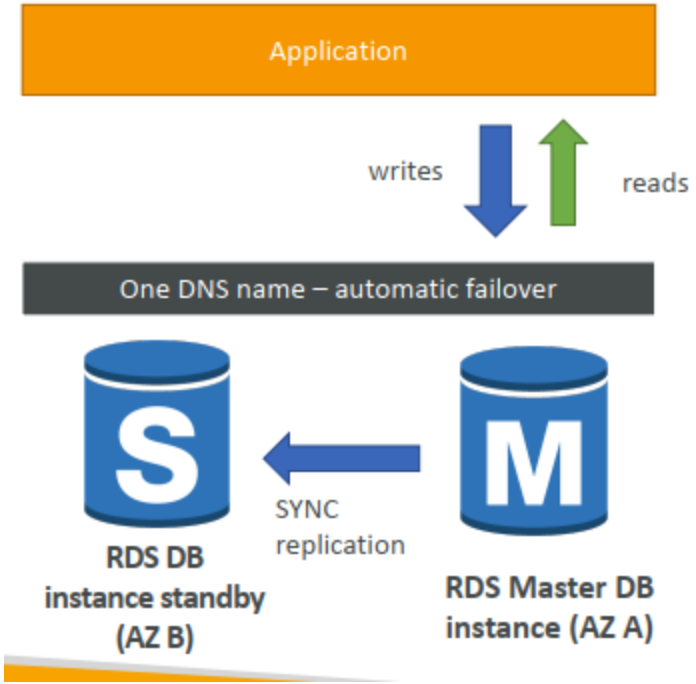
RDS Read Replicas – Network Cost

- In AWS there's a network cost when data goes from one AZ to another
- For RDS Read Replicas within the same region, you don't pay that fee



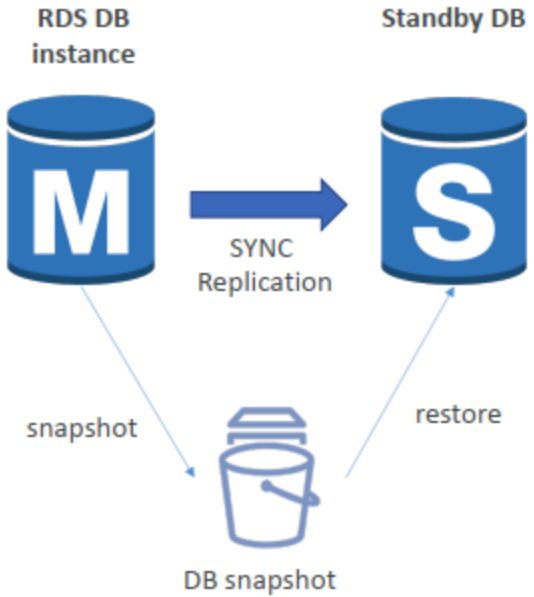
RDS Multi AZ (Disaster Recovery)

- 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
- Multi-AZ replication is free
- Note: The Read Replicas be setup as Multi AZ for Disaster Recovery (DR)



RDS – From Single AZ to Multi AZ

- Zero downtime operation (no need to stop the DB)
- Just click on “modify” for the database
- The following happens internally
  - A snapshot is taken
  - A new DB is restored from the snapshot in a new AZ
  - Synchronization is established between the two databases



RDS Security - 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 Encryption Operation

- Encrypting RDS backups
  - Snapshots of un-encrypted RDS databases are un-encrypted
  - Snapshot of encrypted RDS databases are encypted
  - Can copy a snapshot into an encrypted one
- To encrypt an un-encrypted RDS database
  - Create a snapshot of the un-encrypted database
  - Copy the snapshot and enable encryption for the snapshot
  - Restore the database from the encrypted snapshot
  - Migrate applications to the new database, and delete the old database

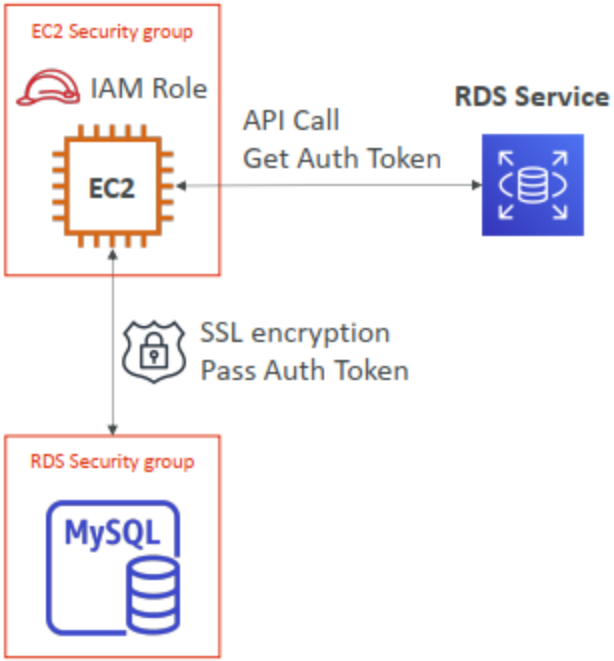
RDS Security – Network & IAM

- 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

RDS – IAM Authentication

- IAM database authentication works with MySQL and PostgreSQL
- You don't need a password, just an authentication token obtained through IAM & RDS API calls
- Authe token has a lifetime of 15 minutes

- Benefits:
  - Network in/out must be encrypted using SSL
  - IAM to centrally manage users instead of DB
  - Can leverage IAM Roles and EC2 Instance profiles for easy integration



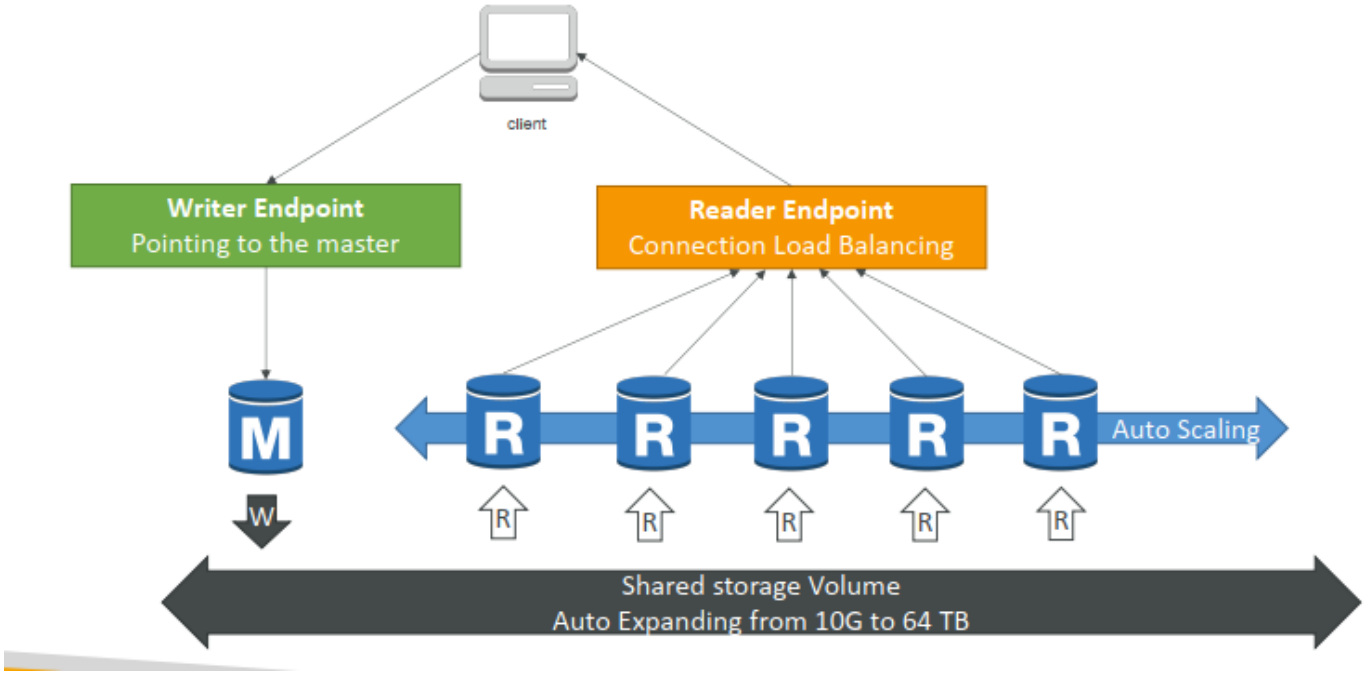
RDS Security - Summary

- Encryption at rest
  - Is done only when you first create the DB instance
  - Or: unencrypted DB -> snapshot -> copy snapshot as encrypted -> create DB from snapshot
- Your responsibility
  - Check the ports/ IP/ security group inbound rules in DB's SG
  - In-database user creation and permissions or manage through IAM
  - Creating a database with or without public access
  - Ensure parameter groups or DB is configured to only allow SSL connections
- AWS responsibility
  - No SSH access
  - No manual DB patching
  - No manual OS patching
  - No way to audit the underlying instance

Amazon Aurora

- Aurora is a proprietary technology from AWS (not open source)
- 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
- Failover in Aurora is instantaneous. It’s HA native
- Aurora costs more than RDS (20% more) – but it more efficient

Amazon Aurora Cluster



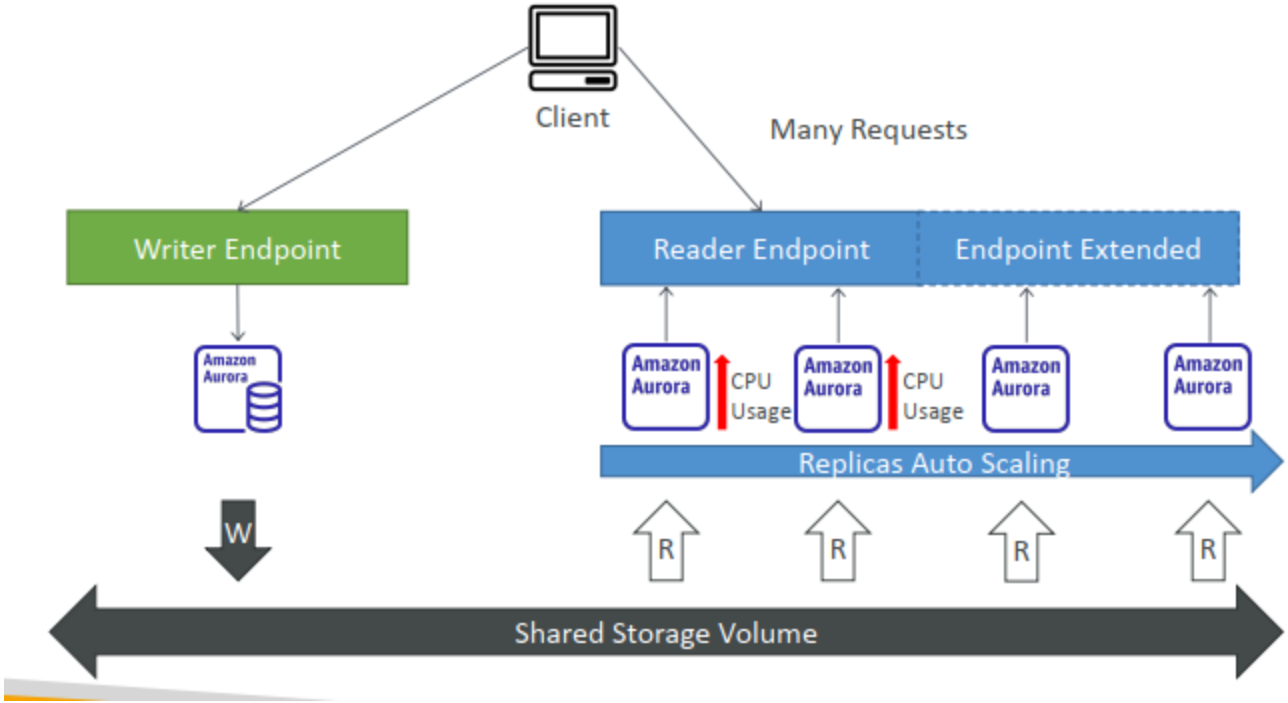
Features of Aurora

- Automatic fail-over
- Backup and Recovery
- Isolation and security
- Industry compliance
- Push-button scaling
- Automated Patching with Zero Downtime
- Advaced Monitoring
- Routine Maintenance
- Backtrack: restore data at any point of time without using backups

Aurora Security

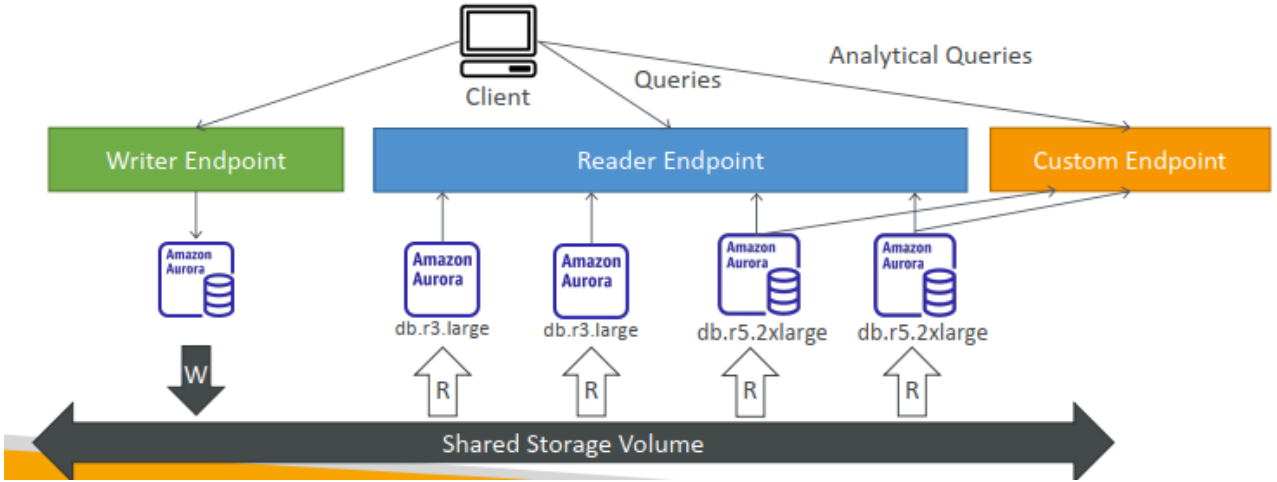
- Similar to RDS because uses the same engines
- Encryption at rest using KMS
- Automated backups, snapshots and replicas are also encrypted
- Encryption in flight using SSL (same process as MySQL or Postgres)
- Possibility to authenticate using IAM token (same method as RDS)
- You are responsible for protecting the instance with security groups
- You can’t SSH

Aurora Replicas – Auto Scaling



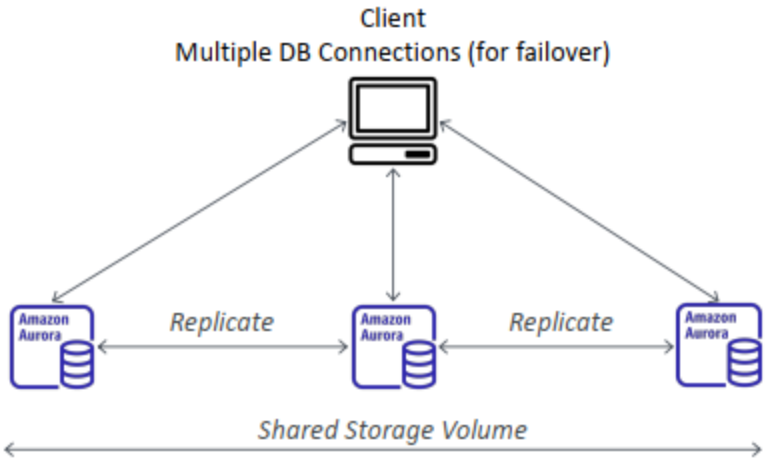
Aurora – Custom Endpoints

- Define a subset of Aurora Instances as a Custom Endpoint
- Example: Run analytical queries on specific replicas
- The reader: Endpoint is generally not used defining Custom Endpoint



Aurora Multi-Master

- In case you want immediate failover for write node (HA)
- Every node does R/W – vs promoting a RR as the new master



Global Aurora

- Aurora Cross Region Read Replicas
  - Useful for disaster recovery
  - Simple to put in place
- Aurora Global Database (recommend)
  - I 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

Thank you!!!