



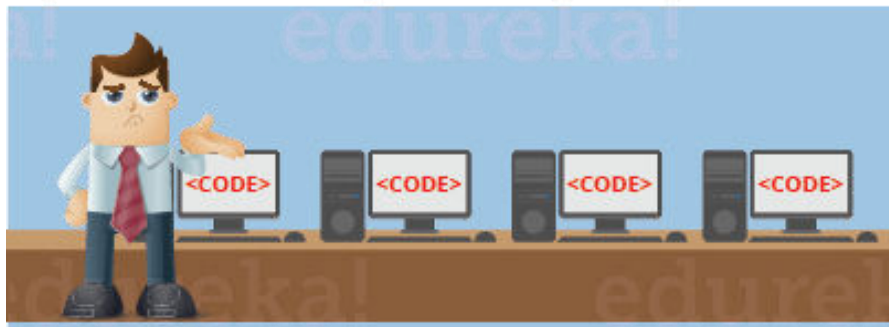
Amazon RDS

Relation Database Service (RDS)

On Prem Database Servers issues



- Customer has to managed on premises database servers.
- Updates, patches on OS by customer.
- Responsible for backups and restore
- No Monitoring dashboard.
- Performance issues, adding new servers for HA.
- Very difficult for DR
- Maintenance windows for upgrades
- Scaling capability Issues.



Amazon RDS



- Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud.
- So people often develop a misconception, when they confuse RDS with a database.
- RDS is not a database, it's a service that manages databases.

RDS Database Engines



Engine options

☒ Amazon Aurora

Amazon
Aurora

☐ MySQL



☐ MariaDB



☐ PostgreSQL



☐ Oracle

ORACLE®

☐ Microsoft SQL Server



Advantage over using RDS vs DB on EC2

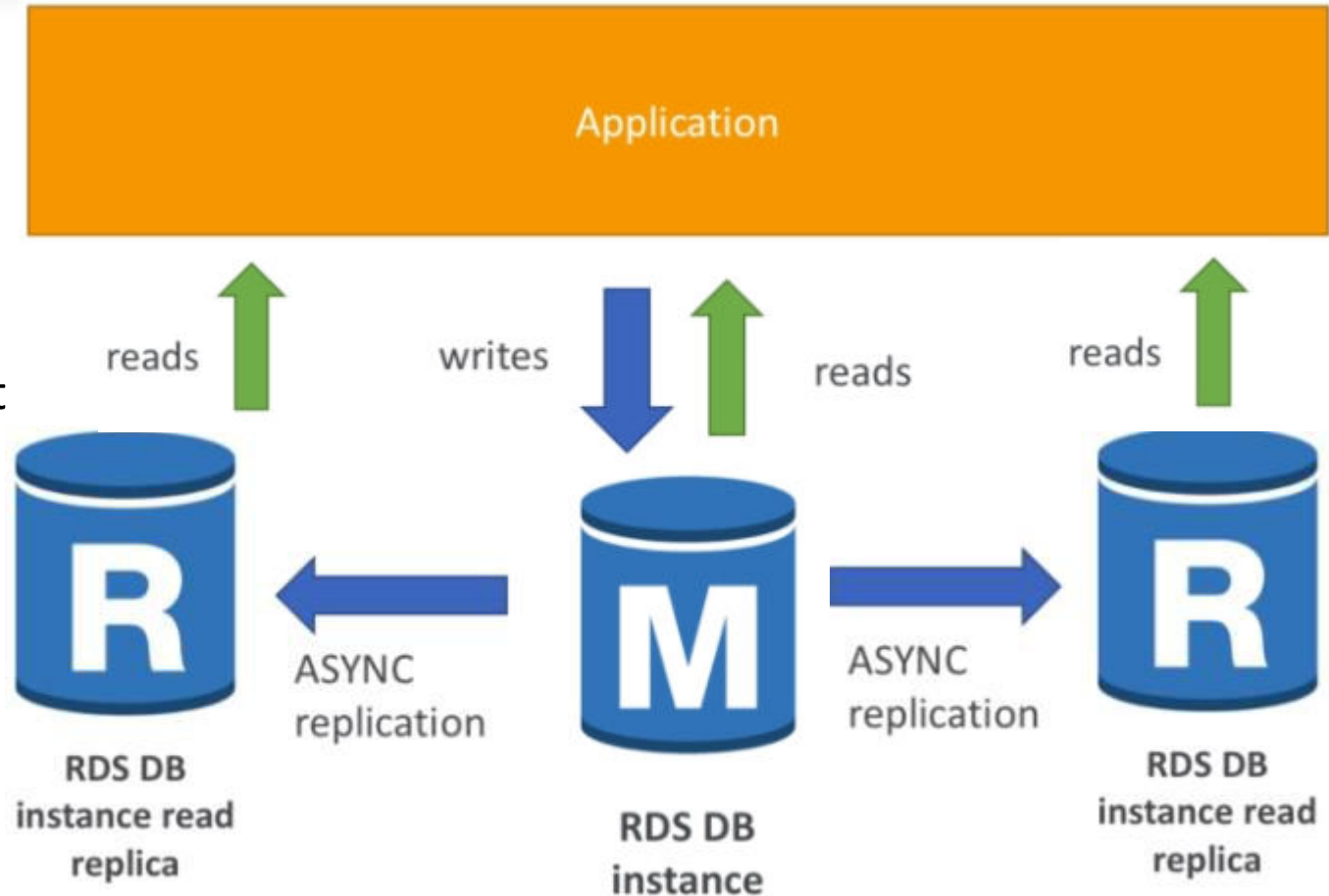


- RDS is a AWS managed Service.
- OS patching level
- Continuous backups and restore to specific timestamp (point in time restore)
- Monitoring dashboard.
- **Read replicas** for improved read performance
- **Multi AZ** setup for DR
- Maintenance windows for upgrades
- Scaling capability (vertical and horizontal)
- You cant SSH or Remote to RDS instance.

RDS Read Replicas for read scalability



- Up to 5 Read replicas
- With in 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.



Read Replica in Depth

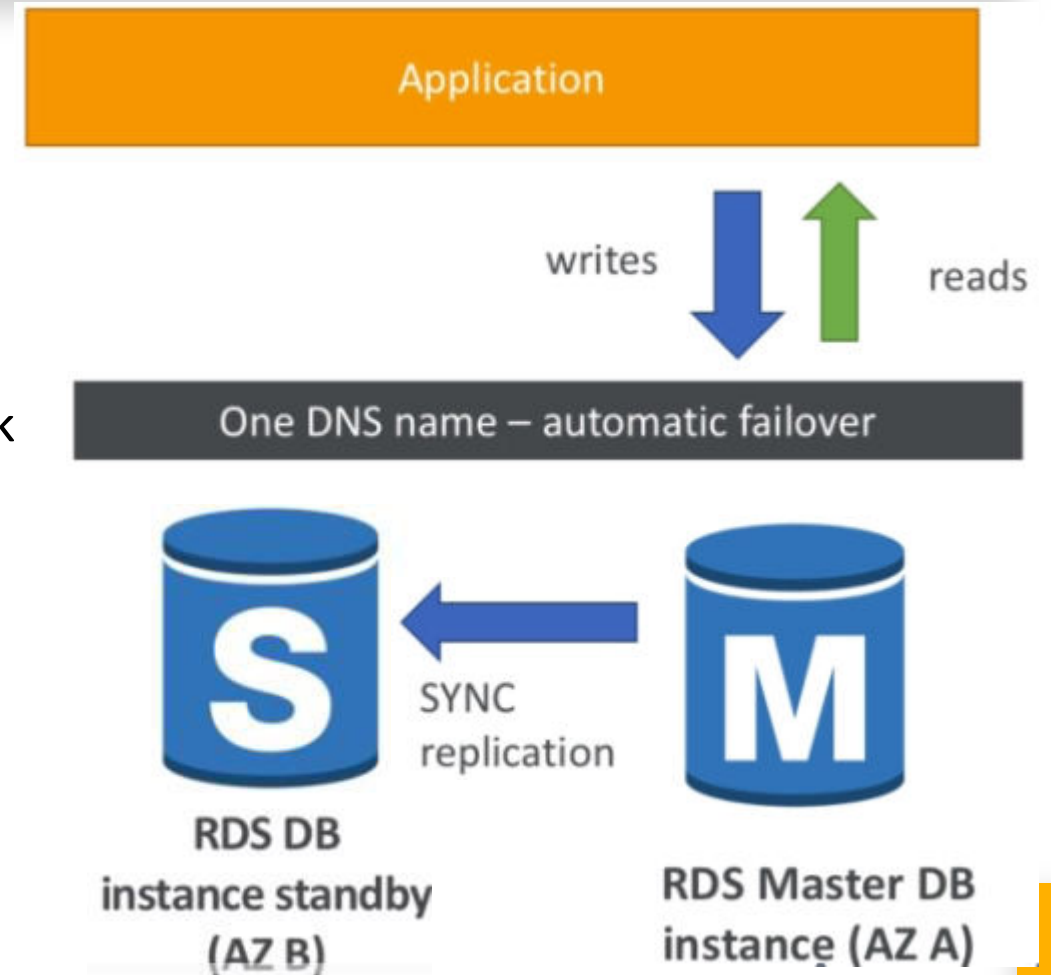


- Read Replicas help scaling read traffic
- A Read Replica can be promoted as a standalone database (manually).
- Read Replicas can be within AZ, Cross AZ or Cross Region.
- Each Read Replica has its own DNS endpoint.
- You can have Read replicas of Read Replicas.
- Read Replicas can be Multi-AZ
- Read Replicas help with DR by using cross region RR
- RDS Read Replicas are not supported for Oracle.
- Read Replicas can be used to run BI / Analytics Reports for example.

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
- Not used for scaling.



Multi AZ in Depth



- Multi AZ is not used to support the reads.
- The failover happens only in the following conditions:
 - The primary DB instance fails
 - An Availability Zone outage
 - The DB instance server type is changed.
 - The operating system of the DB instance is undergoing software patching.
 - A manual failover of the DB instance was initiated using Reboot with failover
- No failover for DB operations: long-running queries, deadlocks or database corruption errors.
- Endpoint is the same after failover (no URL change in application needed).
- Lower maintenance impact it happens on the standby, which is then promoted to master.
- Backups are created from the standby
- Multi AZ is only within a single region, not cross region. Region outages impact availability

RDS Backup



- Backups are automatically enabled in RDS
- **Automated backups:**
 - Daily full snapshot of the database
 - Capture transaction logs in real time
 - Ability to restore to any point in time
 - 7 days retention (can be increased to 35 days)
- **DB Snapshots:**
 - Manual triggered by the user
 - Retention of backup for as long as you want

RDS Encryption



- Encryption at rest capability with AWS KMS – AES-256 encryption .
- SSL certificates to encrypt data to RDS in flight
- 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;`
- To connect using SSL:
 - Provide the SSL Trust certificate (can be download from AWS).
 - Provide SSL options when connecting to database.

RDS 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 who can communicate with RDS.
- IAM policies help control who can **manage** AWS RDS.
- Traditional Username and Password can be used to login to the database.
- IAM users can now be used too (for MYSQL/Aurora - **NEW**)

RDS Security



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 SG
- In-database user creation and permissions
- 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

RDS DB Parameters Group



- You can configure the DB engine using Parameter Groups
- Dynamic parameters are applied immediately.
- Static parameters are applied after instance reboot.
- You can modify parameter group associated with a DB(must reboot).
 - PostgreSQL : rds.force_ssl = 1 in the AWS RDS Console (Parameter Groups)
 - MYSQL: Within the DB:
GRANT USAGE ON *.* TO 'mysqluser'@'%' REQUIRE SSL;

RDS with Cloudwatch



- CloudWatch metrics associated with RDS(gathered from the hypervisor).
 - DatabaseConnections
 - SwapUsage
 - ReadIOPS / WriteIOPS
 - ReadLatency / WriteLatency
 - ReadThroughPut / WriteThroughPut
 - DiskQueueDepth
 - FreeStorageSpace

- Enhanced Monitoring (gathered from an agent on the DB instance)
 - Useful when you need to see how different processes or threads use the CPU
 - Access to over 50 new CPU, memory, file system and disk I/O metrics

RDS Performance Insights



- Visualize your database performance and analyze any issues that affect it
- With the Performance Insight dashboard, you can visualize the database load and filter the load:
 - By waits => find the resource that is bottleneck(CPU, lock, IO, etc)
 - By SQL statements => find the SQL statement that is the problem
 - By Hosts => find the server that is using the most our DB
 - By Users => find the user that is using the most our DB
- DBLoad = the number of active sessions for the DB engine
- You can view the SQL queries that are putting load on your database.

RDS vs Aurora

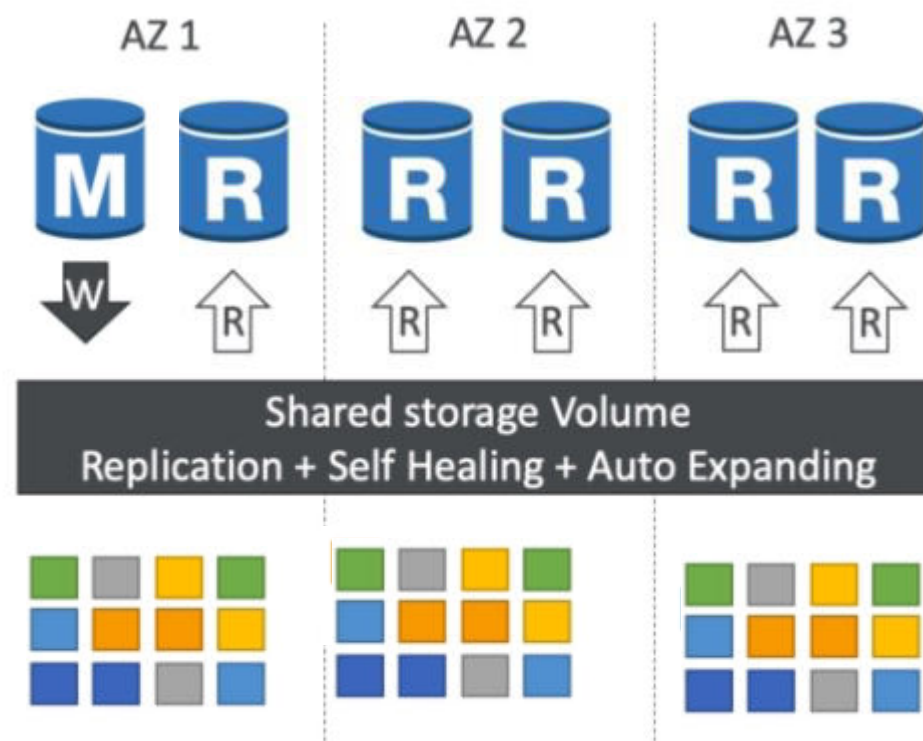


- Aurora is a proprietary technology from AWS(not open sourced).
- Postgres and MYSQL are both supported as Aurora DB (that means your drivers will work as if Aurora was a Postgres or MYSQL database).
- Aurora is “AWS Cloud Optimized” and claims 5x performance improvement over MYSQL on RDS, over 3x the performance of Postgres on RDS.
- Aurora storage automatically grows in increments of 10GB, up to 64TB.
- 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 . Its HA native
- Aurora costs more than RDS (20% more) – but is more efficient

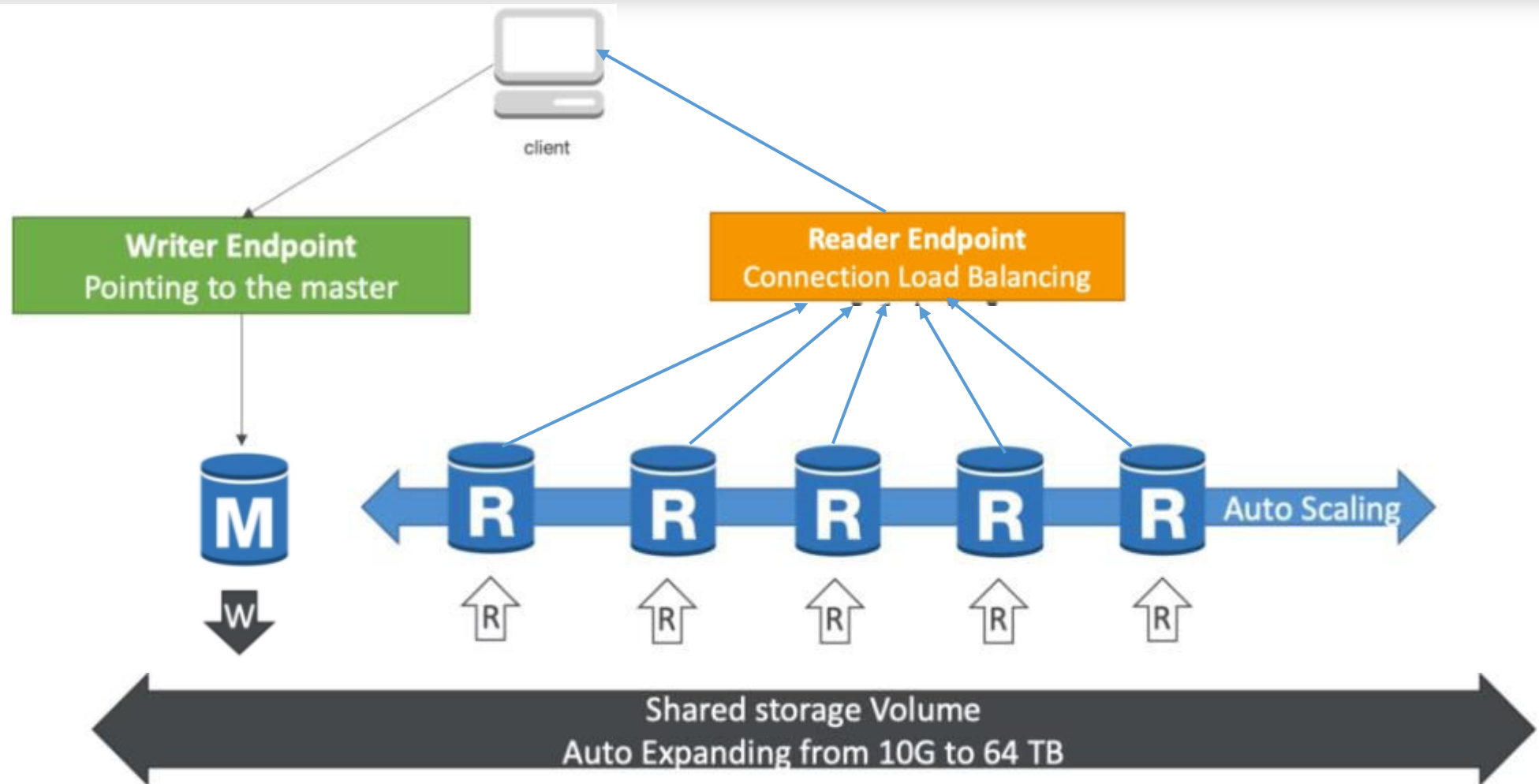
Aurora HighAvailability and Read Scaling



- 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
- ➡ One Aurora Instance takes writes(master)
- ➡ Automated failover for master in less than 30 seconds
- ➡ Master + up to 15 Aurora Read Replicas
- Supports for Cross Region Replication



Aurora DB Cluster



Features of Aurora



- Automatic Failover
- Backup and Recovery
- Isolation and Security
- Industry Compliance
- Push-button Scaling
- Automated Patching with Zero Downtime
- Advanced Monitoring
- Routine Maintenance
- Backtrack: restore data at any point in time without using backups

Aurora Security



- Encryption at rest using KMS
- Automated backups, snapshots and replicas are also encrypted
- Encryption in flight using SSL
- Authentication using IAM
- You are responsible for protecting the instance with security group.
- You cant SSH.

Aurora Serverless



- No need to choose an instance size
- Only supports MySQL 5.6 (as of Jan 2019) & PostgreSQL (beta)
- Helpful when you cant predict the workload
- DB cluster starts, shutdown and scales automatically based on CPU/connections
- Can migrate from Aurora Cluster to Aurora Serverless and vice versa
- Aurora Serverless usage is measured in ACU (Aurora Capacity Units)
- Billed in 5 minutes increment of ACU

Aurora Overview



- Compatible API for PostgreSQL / MySQL
- Data is held in 6 replicas, across 3 AZ
- Auto healing capability (failovers)
- Multi AZ, Auto Scaling read Replicas
- Read replicas can be Global
- Aurora database can be Global for DR or latency purpose
- Auto scaling of storage from 10 GB to 64 TB
- Define EC2 instance type for aurora instances
- Same security / monitoring / maintenance features as RDS
- Aurora Serverless
- Use Case: same as RDS, but less maintenance / more flexibility / more performance

Aurora for Solutions Architect



- **Operations:** less operations, auto scaling storage
- **Security :** AWS responsible for OS security, we are responsible for setting up KMS, security groups, IAM policies, authorizing users in DB, using SSL
- **Reliability:** Multi AZ, highly available, possibly more than RDS, Aurora Serverless option.
- **Performance:** 5x performance (according to AWS) due to architectural optimization. Up to 15 Read replicas (only 5 for RDS).
- **Cost:** Pay per hour based on EC2 and Storage usage. Possibly lower cost compared to Enterprise grade databases such as Oracle.



Neptune

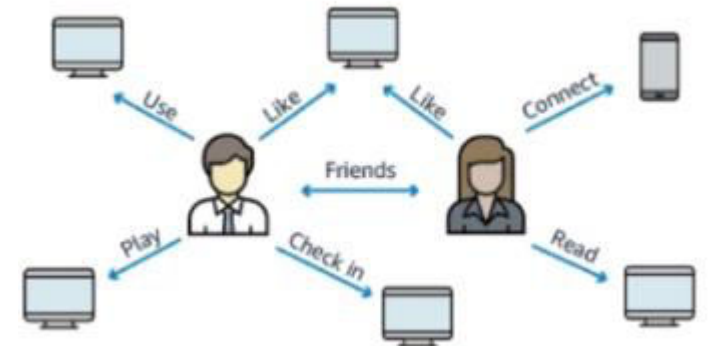
Neptune



- Fully managed **Graph** database

When do we use Graphs?

- High relationship data
 - Social Networking: Users friends with Users, replied to comment on post of user and like other comments
 - Knowledge graphs (Wikipedia)
- Highly available across 3 AZ, with up to 15 read replicas
 - Point in time recovery, continuous backup to Amazon S3
 - Support KMS encryption at rest + HTTPS



Neptune for Solutions Architect



- **Operations:** similar to RDS
- **Security:** IAM, VPC, KMS, SSL + IAM Authentication
- **Reliability:** Multi-AZ, Clustering
- **Performance:** best Suited for graphs, cluster to improve performance
- **Cost:** Pay per node provisioned (similar to RDS)

- **Remember:** Neptune = Graphs



ElasticSearch

ElasticSearch



- Example: In DynamoDB, you can only find by primary key or indexes
- With ElasticSearch, you can search any filed, even partially matches
- Its common to use ElasticSearch as a complement to another database
- ElasticSearch also has some usage for Big Data applications
- You can provision a cluster of instances
- Security though IAM, KMS, SSL and VPC

- Remember : ElasticSearch = Search / Indexing