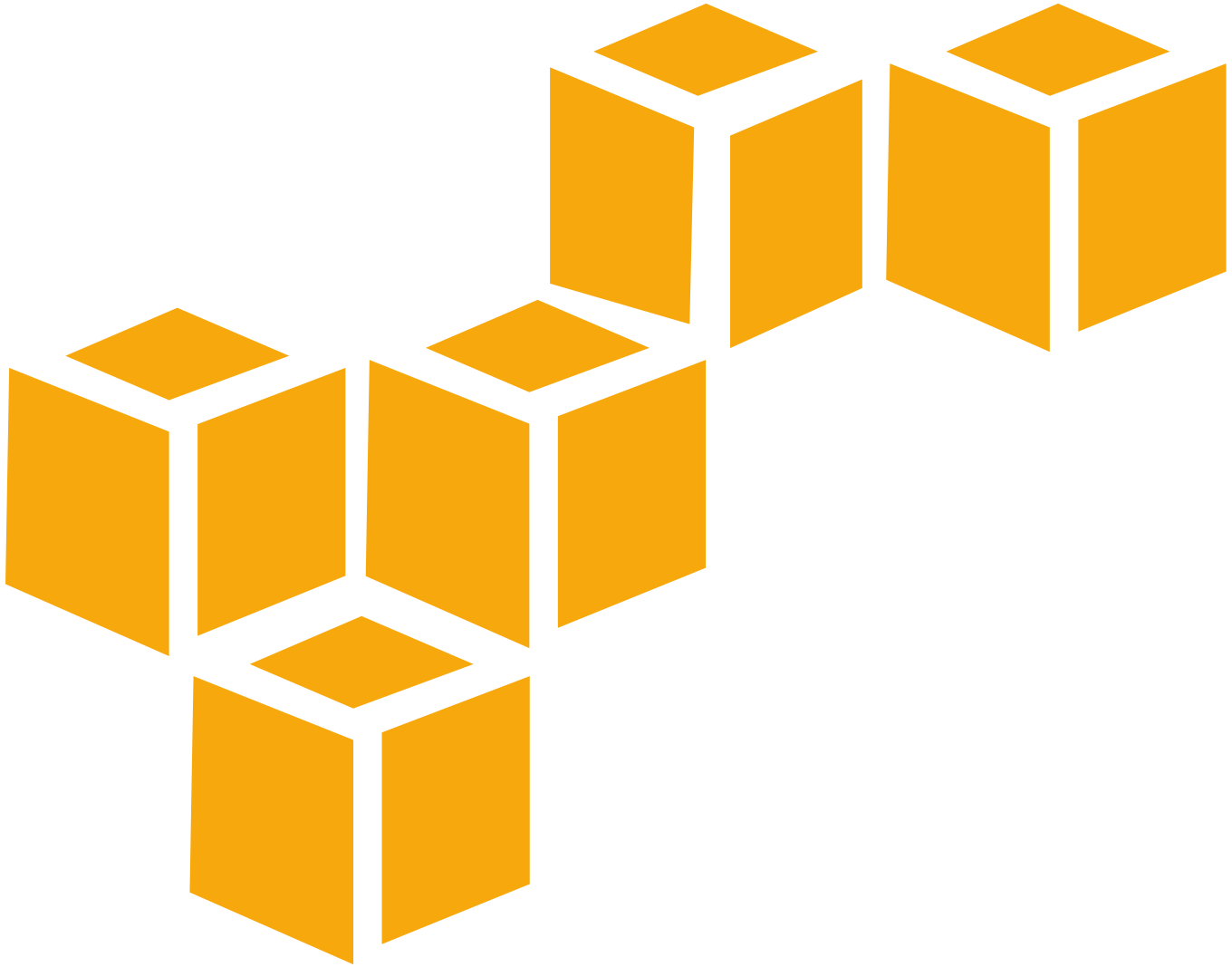


AWS Database Services

Content :

- Database services by AWS
- Amazon RDS
- Amazon DynamoDB
- Amazon Aurora
- Amazon ElastiCache
- Amazon Redshift

AWS Database Services

Database type	Use cases	AWS service
Relational	Traditional applications, ERP, CRM, e-commerce	Amazon Aurora, Amazon RDS, Amazon Redshift
Key-value	High-traffic web apps, e-commerce systems, gaming applications	Amazon DynamoDB
In-memory	Caching, session management, gaming leaderboards, geospatial applications	Amazon ElastiCache for Memcached Amazon ElastiCache for Redis
Document	Content management, catalogs, user profiles	Amazon DocumentDB (with MongoDB compatibility)
Wide column	High scale industrial apps for equipment maintenance, fleet management, and route optimization	Amazon Keyspaces (for Apache Cassandra)
Graph	Fraud detection, social networking, recommendation engines	Amazon Neptune
Time series	IoT applications, DevOps, industrial telemetry	Amazon Timestream
Ledger	Systems of record, supply chain, registrations, banking transactions	Amazon QLDB

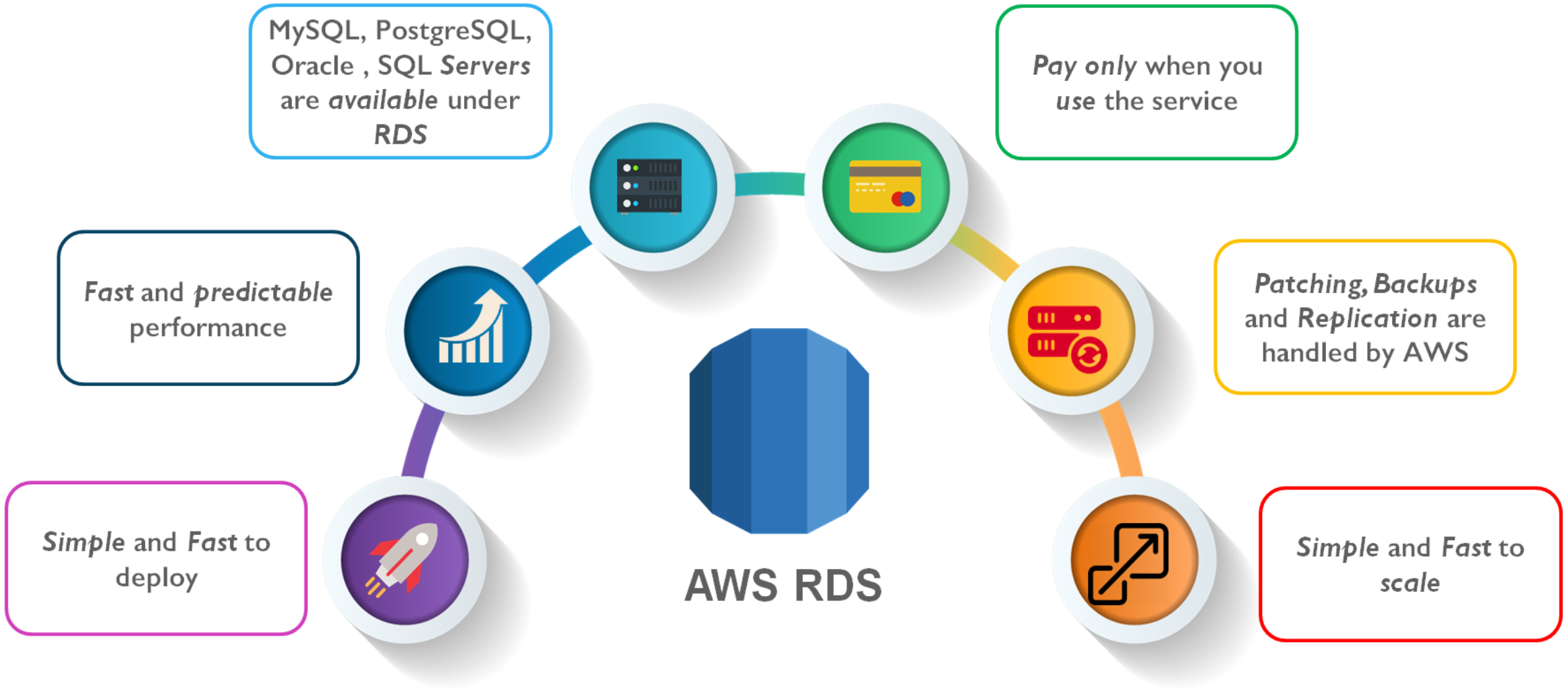
Database Services by AWS

RDS	• Manages relational database • Handles structured and tabular data
AURORA	• MySQL and PostgreSQL– compatible relational database with 3x/5x performance
DYNAMODB	• No SQL database • Document and key-value store • Handles unstructured data
ELASTICACHE	• In memory cache • Increase the performance of data-intensive applications
REDSHIFT	• Data warehouse • Used for data analysis and reporting

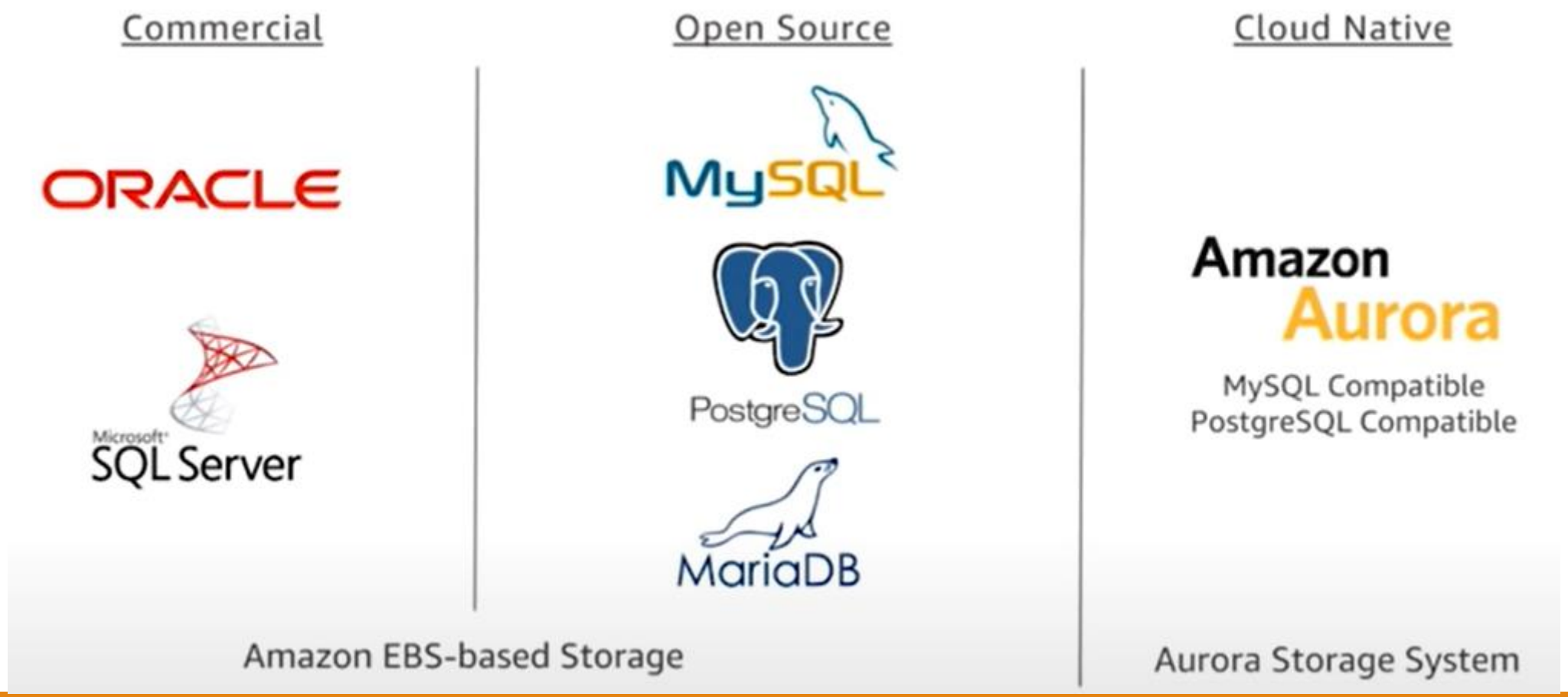
Amazon RDS

RDS is a relational database management service which manages relational databases for users.

- Managed service that handles set up, software patching, update, backups, recovery and automatic failure detection.
- Either manual backup can be created via snapshot or can have an automated backup.
- Can use replication to enhance availability and durability and can also scale up for read-heavy database workloads.
- Mainly used to manage data of e-commerce, gaming, apps, websites and many more.


MySQL, PostgreSQL, Oracle , SQL Servers are available under RDS
Pay only when you use the service
Fast and predictable performance
Patching, Backups and Replication are handled by AWS
Simple and Fast to deploy
Simple and Fast to scale
AWS RDS

RDS Database engines



RDS – DB Instance

Database Instance is a set of memory structures that manages the database.

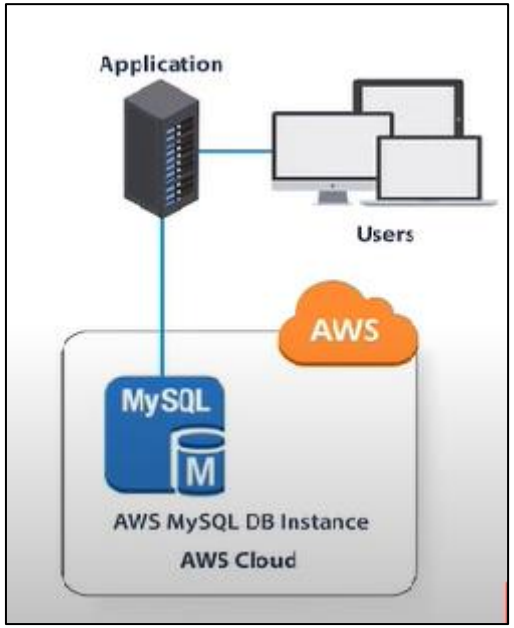
01 • It runs on DB engine

02 • The computation and memory capacity of a DB instance is determined by its DB instance class

03 • Each DB instance can host multiple user created databases or a single Oracle database with multiple schemas

04 • Each DB Instance runs on a DB engine

05 • By default one can have 40 RDS instances



Database schemas within DB engines

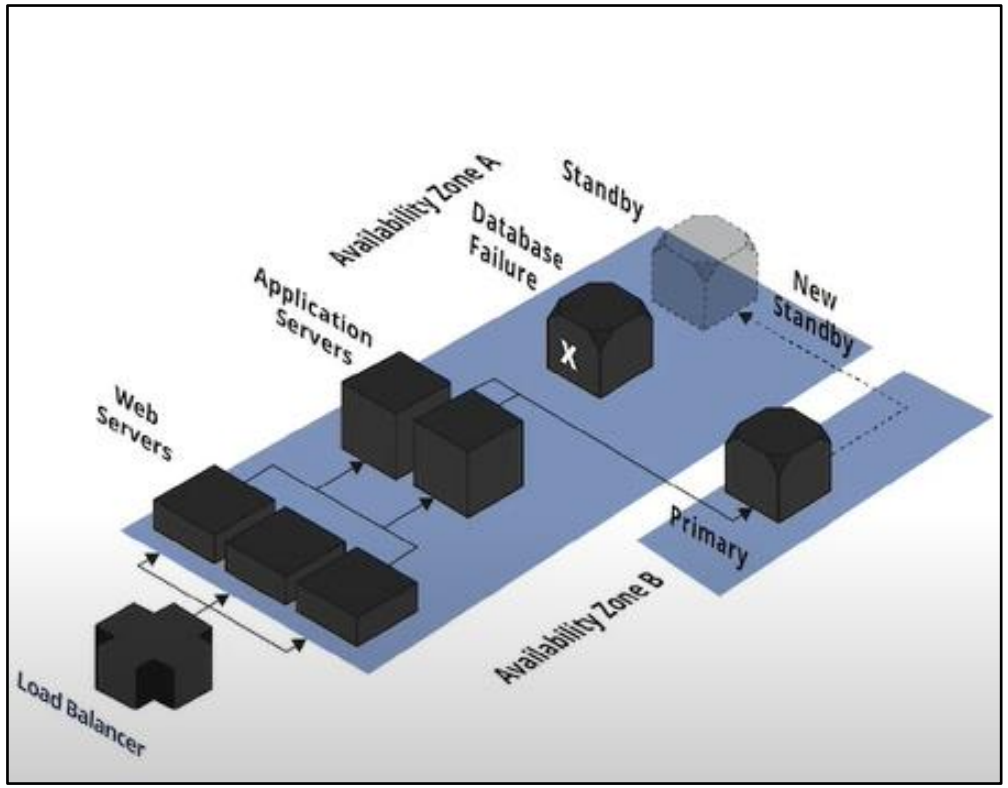
Database Engine	Number of Databases within an instance
Amazon Aurora	No limit imposed
MySQL	No limit imposed
MariaDB	No limit imposed
Oracle	1 database per instance; No limit of number of schemas per database
SQL Server	100 databases per instance
PostgreSQL	No limit imposed

Hands On 1: RDS using EC2

- Launch a RDS MySQL instance in default VPC
- Launch an EC2 instance in default VPC with public IP
- Connect to EC2
- Install MySQL client on EC2 instance
- Connect to MySQL RDS instance
- Create a database
- Add a table
- Insert few records
- Select the inserted records

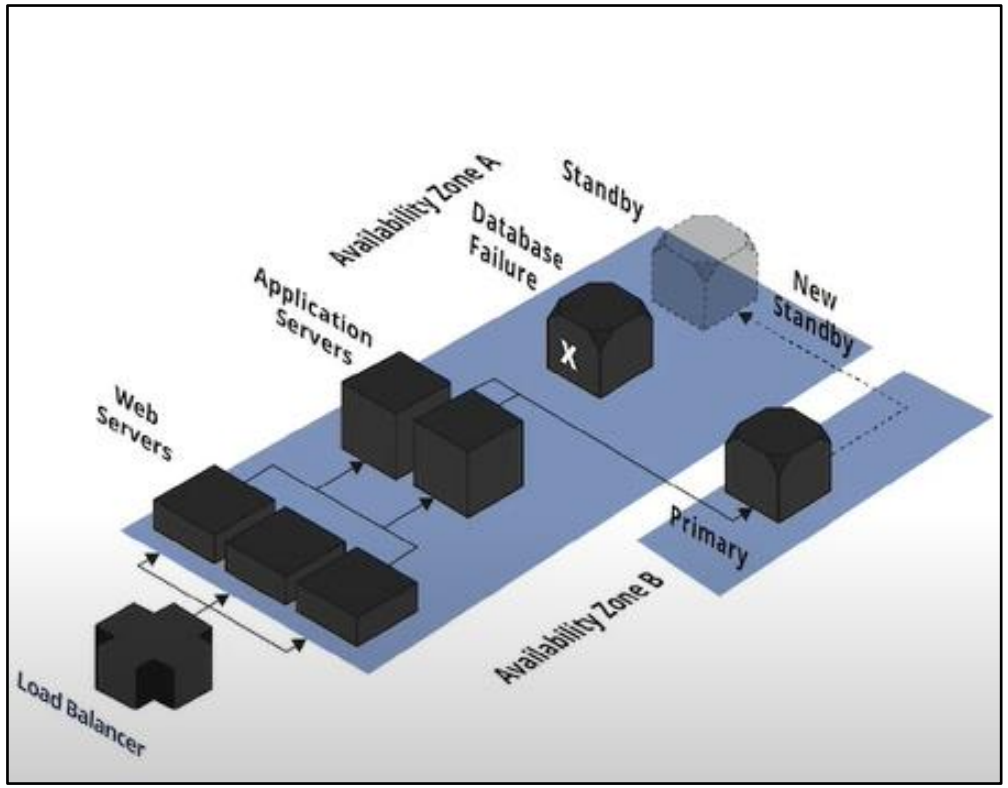
yum install httpd php php-mysql mysql –y
mysql -h <DNS port> -P 3306 -u <username> -p

RDS – Multi A-Z



- Allows to have an exact copy of production database in another AZ. AWS handles the replication, so when production database is written to, will automatically be synchronized to the standby database.
- In the event of planned database maintenance, DB Instance failure, or an AZ failure, Amazon RDS will automatically failover to the standby so that database operations can resume quickly without interventions.
- Both DB servers should have same DNS endpoints.

RDS – Multi A-Z



- Enhances the durability
- Increases the availability
- Protection of database performance
- Automatic failover.

- Failover Conditions :
  - Loss of availability in primary Availability Zone.
  - Loss of network connectivity to primary
  - Compute unit failure on primary
  - Storage failure on primary

RDS Backups

RDS Backups are of 2 types : **Automated Backup**
DB Snapshots

Automated Backup

- Allows to recover database to any **point in time** within a “retention period”. The retention period can be between 7 and 35 days.
- It will take a full day snapshot and will store transaction logs . AWS will choose the most recent daily back up.
- Automated backups are enabled by default.
- Only works when engine in InnoDB enabled.

DB Snapshots

- They are done manually and cost effective.

RDS Back and Restore

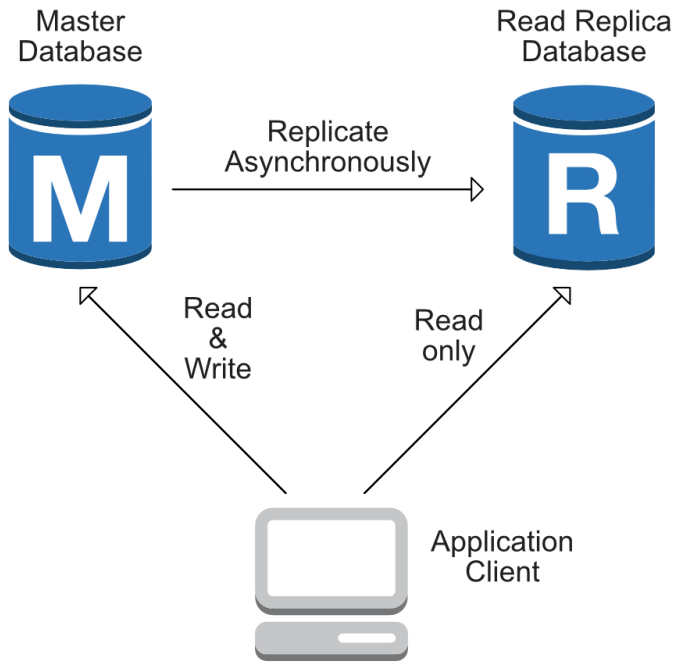
AWS RDS carries the automated backups of DB instances as per the specified backup retention period

The backup retention period can be set between 7-35 days

Even manually backups can be created via Snapshots

When a DB instance is deleted the automated backups too get deleted but manual Snapshots remain the same.

RDS – Read Replica



- Allows to have a read-only copy of production database. This is achieved using Asynchronous replication from primary RDS instance to the read replica.
- Primarily used for heavy database workload.
- Use for scaling.
- Can have up to 5 RR of any database.
- Can have read replicas of read replicas but with latency.
- Each RR will have its own DNS end point.

Hands On 7: Read Replica

Region : Ohio:

- Create a private VPC of CIDR (10.2.0.0/16)
- Create 3 subnets :
  - public (10.2.0.0/28)
  - private 1 (10.2.0.32/28)
  - private 2 (10.2.0.16/28)
- Create an IG , attach with VPC
- Create 2 Route tables :
  - public – with IG and public subnet
  - private – with 2 private subnets
- Create 1 subnetGroup under RDS with 2 private subnets

Region : Singapore

- Create one RDS instance with default configurations.
- Create a table using EC2.
- Once the instance is up, go to Options and select “Create Read Replica”
- Select the Destination as “Ohio”
- Select the subnet group as the group created in Ohio.
- Create read replica.

- Check the databases of Ohio, read replica will be added.
- Create an EC2 instance in Ohio with public subnet and SG open as SSH to all and 3306 to VPC CIDR
- Login to EC2 and connect to readreplica in Ohio.
- Insert a new value in Singapore RDS and the value will be visible in Ohio db.

Billing of RDS

Parameters	Billing Procedure
DB instance hours	Pricing is per DB instance-hour consumed, from the time a DB instance is launched until it is stopped or deleted. Can be On-Demand or Reserved instance.
Storage (per GB per month)	With General Purpose (SSD), charges will be for storage provisioned and not based on the I/Os.
I/O requests per month	With Provisioned IOPS, charges will be for the IOPS and storage provisioned.
Data transfer	Data transfer out of your DB instance on internet. Transfer between RDS and EC2 Instances in the same AZ is free.