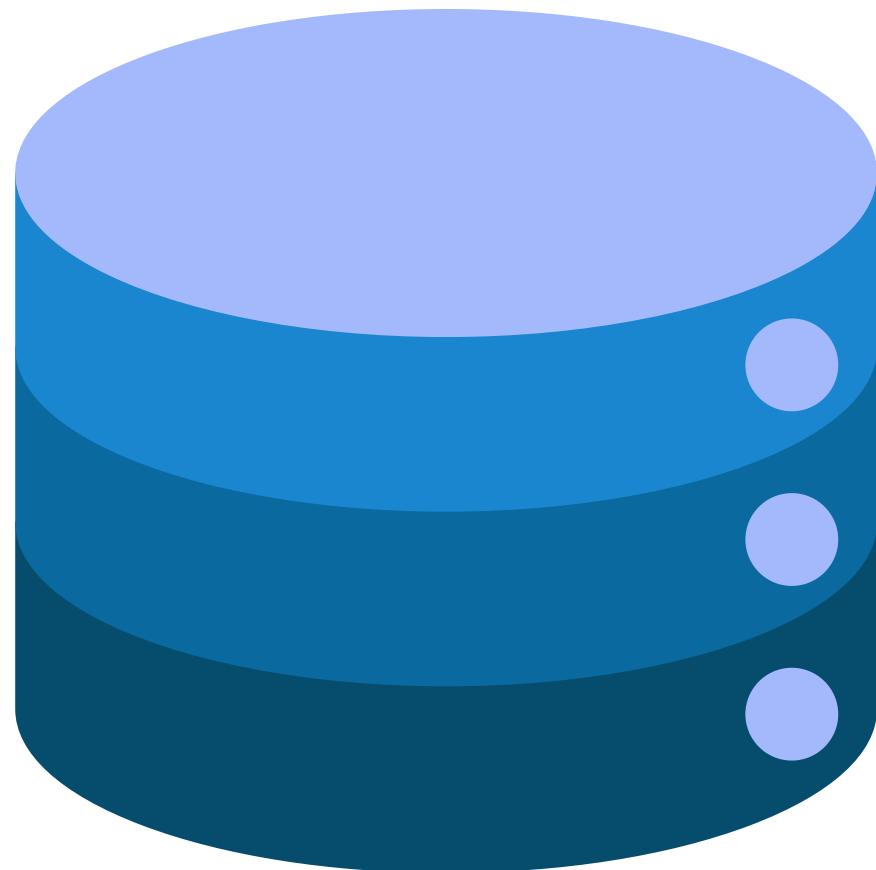


# RDS



# INTRO:

- Amazon Relational Database Service (Amazon RDS).
- it easier to set up, operate, and scale a relational database in AWS.
- It provides cost-efficient, resizable capacity.
- It's responsible for most management tasks.
- It eliminating tedious manual processes.
- Amazon RDS frees you to focus on your application and your users.

# **ADVANTAGES:**

- You can use database engines that you are already familiar with.
- Amazon RDS manages backups, software patching, automatic failure detection, and recovery.
- You can turn on automated backups & restore a database.
- You can get high availability with a primary DB instance secondary DB
- You can also use read replicas to increase read scaling.
- you can control access by using IAM to define users and permissions.

# ENGINES:

Engine type [Info](#)

Aurora (MySQL Compatible)



Aurora (PostgreSQL Compatible)



MySQL



PostgreSQL



MariaDB



Oracle

**ORACLE®**

Microsoft SQL Server

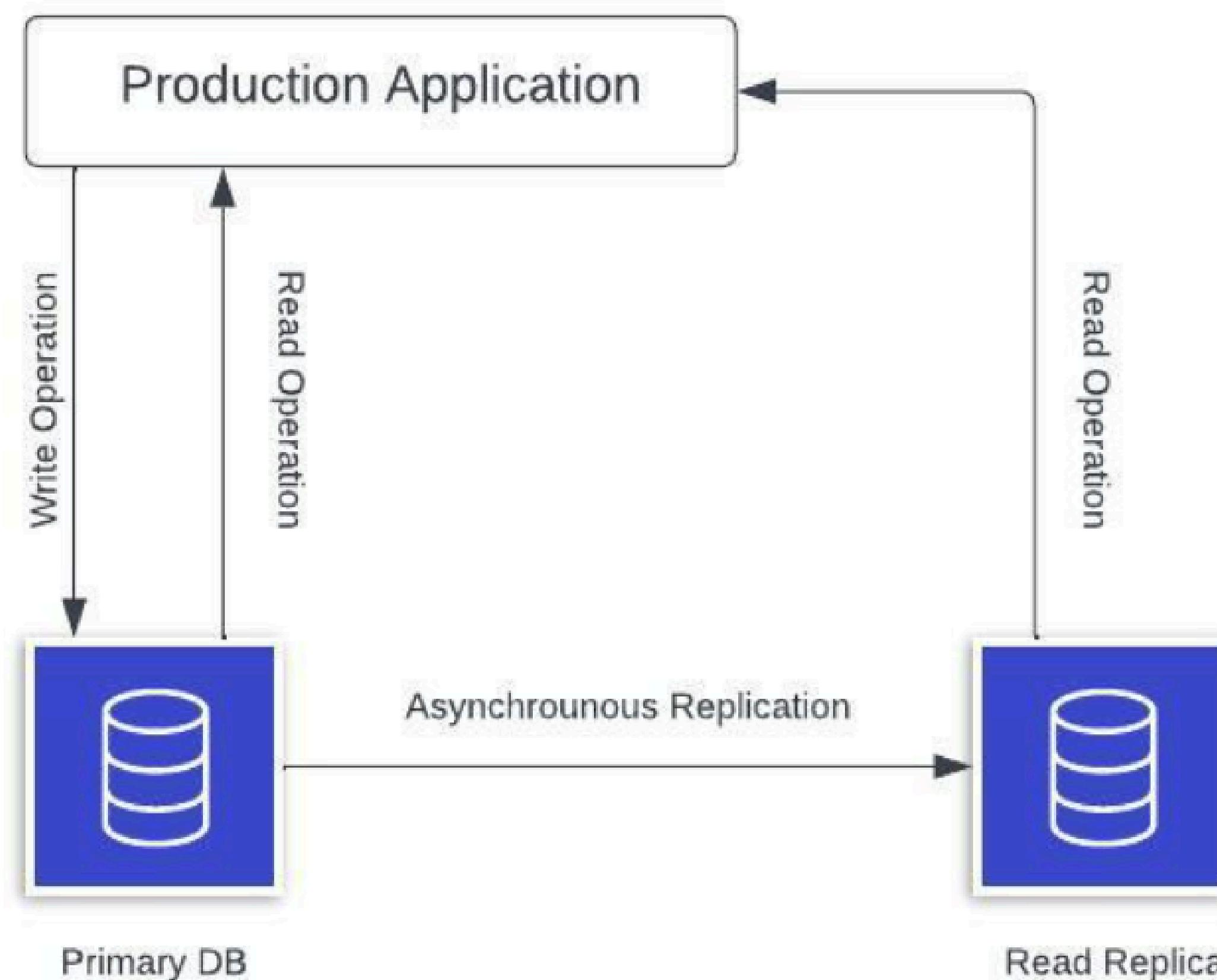


IBM Db2

**IBM Db2**

# **READ REPLICA:**

- Read Replicas are copies of your primary database instance.
- used to offload read queries from the primary database.
- improving performance, scalability, and availability.
- Our Prod App will not be slow down,
- Data will be asynchronously replicate from primary DB to Read replicas.
- Read Replicas will be helpful in Disaster Recovery.
- used to test application changes without affecting the primary database.
- reducing latency and improving performance for geographically distributed applications.
- We can Create upto 15 Read Replicas.
- Replica can become its own DB.



## CORNS:

- light delay between data changes and replication.
- All write operations must be done on the primary database.
- Read replicas cannot accept write operations.
- Cost is A bit High.

## MULTI-AZ:

- Used for Synchronous Replication.
- In case of one AZ fails we have standby DB to act as Main DB.
- We use one DNS name for all of them.
- Not used for scaling.
- Note: RR can be setup in Multi-AZ for Disaster Recovery.

# RR VS MULTI AZ:

- Multi AZ deployments are mainly used for backups and disaster recovery strategies. Multi AZ deployments contain a primary database and a standby database, when the primary database goes down (or) becomes unavailable the standby database becomes active.
- While read replicas are used to scale reads in a database, multi-AZ deployments are used to increase availability of data. Multi-AZ deployments uses synchronous mode of replication.

# **SYNC VS ASYNC:**

- In **synchronous replication**, changes made to the primary database are written to the replica at the same time, ensuring that both the primary and replica databases are always in sync.
- In **asynchronous replication**, changes made to the primary database are not immediately written to the replica. The changes are propagated to the replica with some delay.

## **Synchronous Replication:**

- Use it for high-availability scenarios, where data loss and downtime must be minimized.
- Ideal for disaster recovery, failover, and critical applications requiring strong data consistency (e.g., Multi-AZ for RDS).

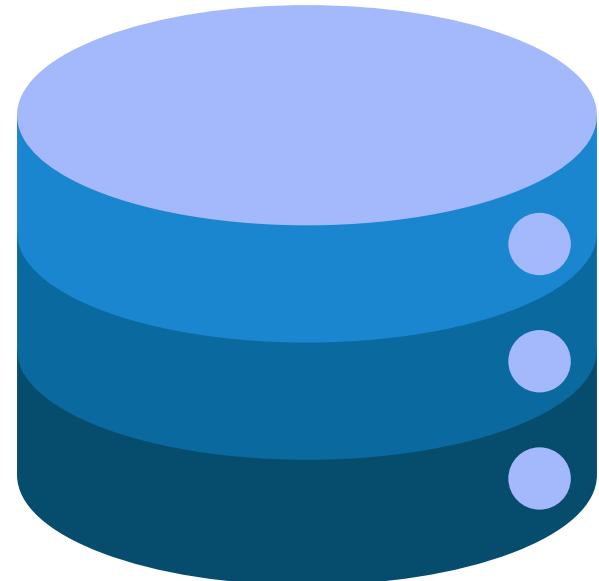
## **Asynchronous Replication:**

- Use it when you need to scale read-heavy workloads.
- Suitable for applications where write performance is critical, and some level of replication lag is acceptable.

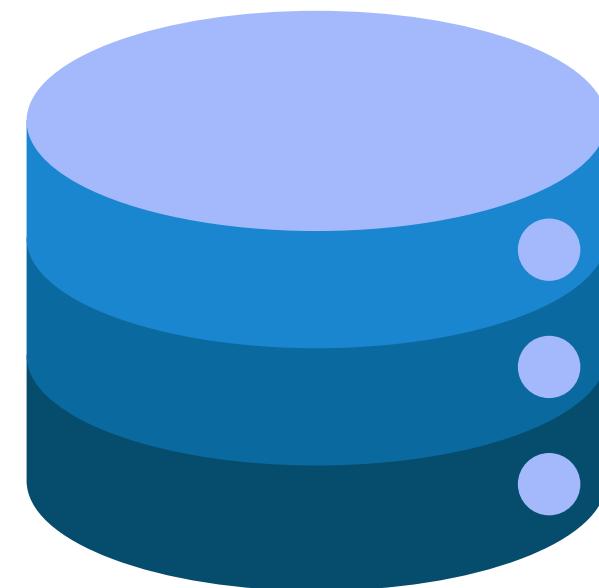
ATTRIBUTES	RDS READ REPLICAS	RDS MULTI-AZ DEPLOYMENT
PURPOSE	Used to scale reads in a database by creating multiple read only databases.	Used to increase availability of a database by creating an exact replica of the entire database and deploying it in another AZ.
MODE OF REPLICATION	Used asynchronous mode of replication.	Used synchronous mode of replication.
SUPPORTED OPERATIONS	Other than source databases, all the read replicas only support select clause queries.	Both primary and standby databases support all types of queries.
NO. OF DUPLICATE DB INSTANCES	Supports up to 15 read replicas for a single DB instance.	Supports only 2 databases: a primary database and a secondary database.
OTHERS	Read replicas can be promoted to its own databases.	The standby database becomes active only when the primary database is not available.

# COST:

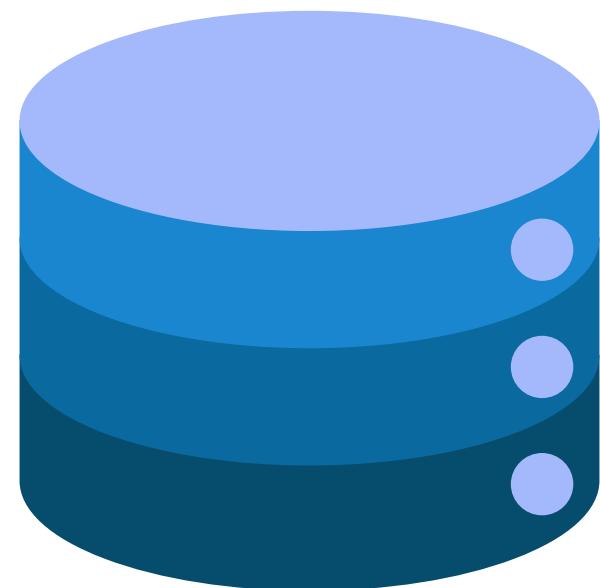
- For same Region its Free.
- For Different Region its paid.



us-east-1a



us-east-1b



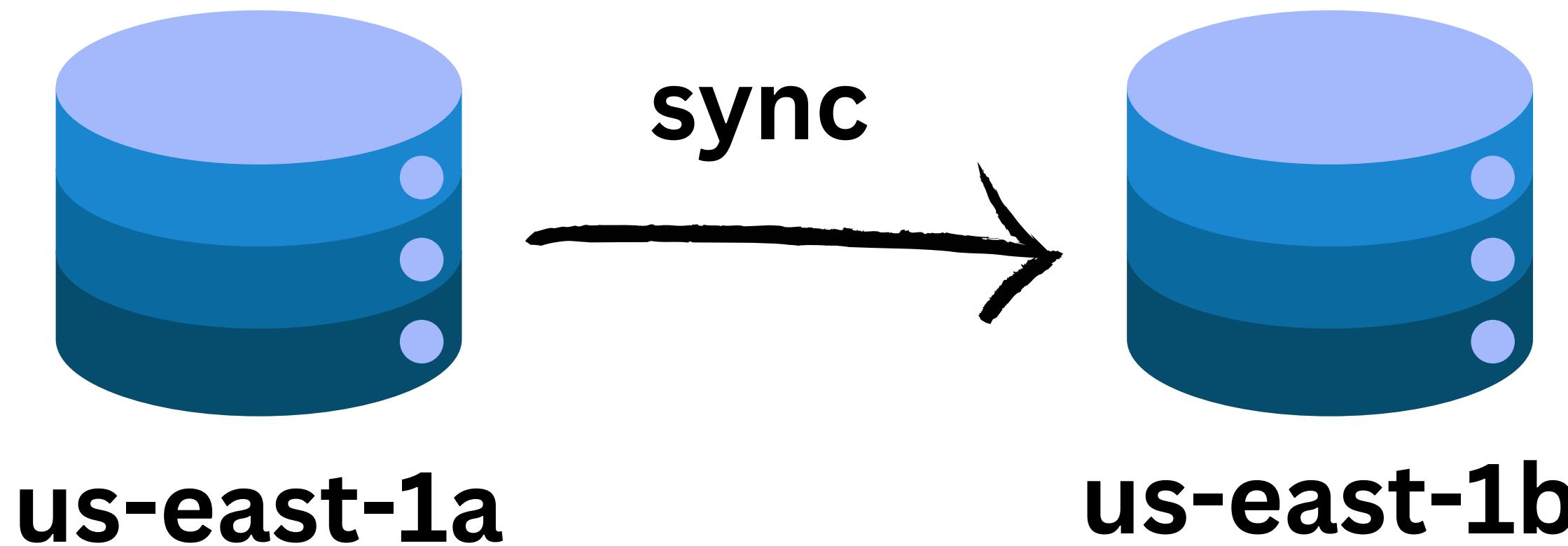
us-east-1a



us-west-1a

# SINGLE TO MULTI-AZ

- It Required Zero Downtime operation.
- we should not stop it.
- just click on modify option.
- Internally db-1 will create snapshot and restore to db-2 automatically.



- **DATABASES**
- **CREATE**
- **Standard create**
- **ENGINE: MYSQL**
- **VERSION: 8.0.39**
- **TEMPLATE: PRODUCTION**
- **AVALIABILITY: Multi-AZ DB Cluster**
- **DB cluster identifier: database-1**
- **Master username: admin**
- **Password: systemmanager**
- **standard: db.m5d.large**
- **storage: 400**
- **CONNECTIVITY: Connect to ec2**
- **NOTE: All the config of ec2 will be applicable for RDS**
- **NOTE: SECRET MANAGER IS OUT OF SCOPE BUT WE CAN USE IT FOR OUR CREDs ROTATION.**

# **SETUP:**

- **sudo wget https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm**
- **sudo dnf install mysql80-community-release-el9-1.noarch.rpm -y**
- **sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2023**
- **sudo dnf install mysql-community-client -y**
- **sudo dnf install mysql-community-server -y**
- **CONNECTION:**
- **mysql -u admin -h endpoint -p**

- create database raham;
- show databases;
- use raham;

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,  
name VARCHAR(100),  
email VARCHAR(100),  
address VARCHAR(100),  
created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

INSERT INTO users (name, email) VALUES ('ramu', 'ramu@gmail.com');  
INSERT INTO users (name, email) VALUES ('remo', 'remo@gmail.com');  
INSERT INTO users (name, email) VALUES ('aparichith', 'aparichith@gmail.com')  
SELECT \* FROM users;  
exit

## RDS BACKUPS:

- It will take backups automatically for every 5 minutes.
- Retention period is only 1 to 35 days.
- we can take manual backup as snapshots and gets long retention period.
- in real time when we use db we run it and if we dont want we take backup and restore for next time when we want to use it.

## RESTORE:

- we can restore the data from snapshots immediate.