

Grup 4 – Lab 274

Yonetim Academy re-Start II

Date: August 2023

Team Members and Roles:

Koordinator: Emine Akoz

Team: Tugba, Busra, Meltem, Ibrahim, Mehmet Baran, Onur,

Research: Huseyin



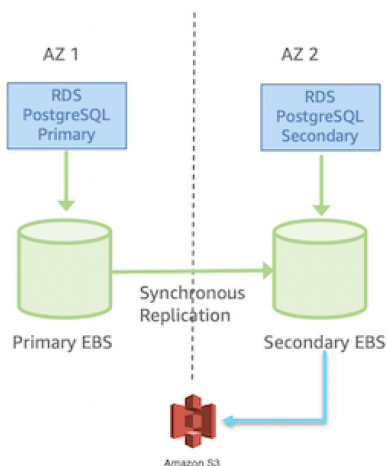
Differences Between AWS RDS – AWS Aurora from Open Source:

- High Availability and Durability:

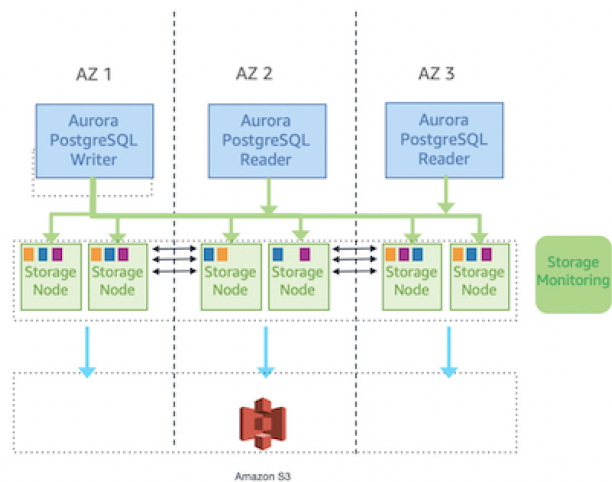
- RDS: To achieve reliability, you need to **enable the Multi-AZ** feature on your RDS instance and replicate it synchronously to a standby replica in another Availability Zone.

- Aurora: Aurora's database storage is separate from the instances. In Aurora, data has 6 copies (as 10GB chunks distributed) to three Availability Zones. Hence, even if you have **only one Aurora instance, your data will still have 6 copies**.

RDS PostgreSQL



Amazon Aurora PostgreSQL



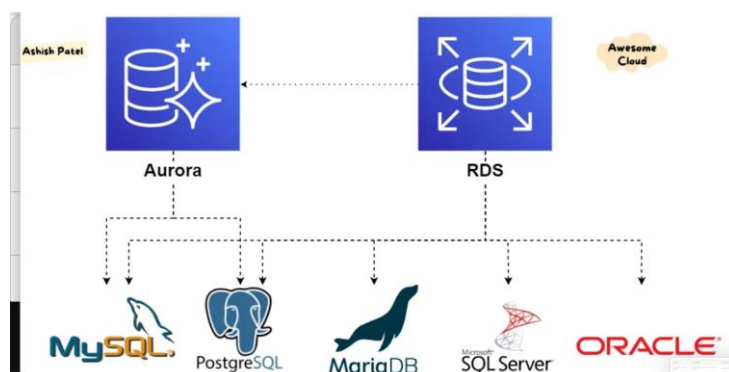
- With Aurora, you can provision up to **fifteen** replicas compared to just **five** in RDS MySQL.

- Performance and Caching:

- Aurora Serverless: This feature offers the user to set up automatic starts up, shuts down, and scales capacity up and down based on the workload of the application

➤ RDS does not have this feature.

- Available DB:



- Cost:

- Aurora provides higher performance and scalability but usually comes with a **higher cost**.

Feature	Amazon RDS	Amazon Aurora
Database engines	MySQL, PostgreSQL, Oracle, SQL Server, MariaDB	MySQL, PostgreSQL
Performance	Similar to traditional database servers	Faster than traditional database servers
Scalability	Horizontally scalable	Horizontally and vertically scalable
Availability	99.95% guaranteed	99.99% guaranteed
Security	Amazon Web Services' security features	Amazon Web Services' security features
Price	Based on the database engine, instance type, and storage	Based on the database engine, instance type, and storage
Features	Support for a wide range of features, including backups, replication, and monitoring	Support for a wider range of features, including backups, replication, monitoring, and more

High Availability And Disaster Recovery:

Automated backups are taken from the standby instance. Incidents such as primary instance failure, storage failure, instance scale-up, and network failure trigger failover to make the secondary instance a new primary instance. Amazon RDS for PostgreSQL also supports replicas in the same Region as well as cross-Region. This replication is based on database transaction logs, and replication lag can increase depending on the workload at the source instance. You can copy and share Amazon RDS snapshots across AWS accounts and Regions for DR purposes.

If a failure occurs and no Aurora replica has been provisioned, it attempts to create a new database instance automatically. Amazon Aurora Global Database offers cross-Region replication. The typical cross-Region replication latency is below 1 second. You can also copy and share Aurora cluster snapshots across AWS accounts and Regions for DR purposes.

If you fulfil the high availability requirement of your data by replicating in two Availability Zones, and cross-Region replication lag is acceptable, Amazon RDS for PostgreSQL should meet your requirements. For higher data availability and to design a DR solution with minimal Recovery Point Objective (RPO) and Recovery Time Objective (RTO), Aurora PostgreSQL is the preferred choice.

Backup:

Amazon RDS automatically takes daily backups of PostgreSQL instances during a backup window. There is a slight performance impact when the backup initiates for single Availability Zone deployments. In addition, it also backs up transaction logs (WALs). For PITR, the full backup is restored first, followed up by replaying WALs until the desired time. In write-intensive RDS for PostgreSQL instances, replaying transaction logs may take a long time. Frequently taking manual backups can reduce PITR duration.

Aurora PostgreSQL backs up cluster volume automatically and retains backups for the length of the defined retention period. Aurora backups are incremental, so it can be quickly restored to a point within the backup retention period. There is no performance impact or interruption of database service during backups. For PITR, a new copy of the database cluster is created from the backup of the database at any point in time within the backup retention period. The continuous and incremental nature of backup improves the PITR restore time.

If you're looking for a database solution with lower RPO and RTO, and if your workloads are sensitive to performance degradation during backups, Aurora PostgreSQL is a better solution. If momentary performance degradation in a single Availability Zone configuration during backup and higher RPO and RTO aren't business critical, Amazon RDS for PostgreSQL can meet your requirements. In Amazon RDS for PostgreSQL, you can modify the automated backup window to avoid performance degradation during peak hours. In Multi-AZ configuration, the backups are taken from the standby instance.

Database Instance Classes:

Amazon RDS for PostgreSQL supports several instance classes, including general purpose T2, T3, M3, M4, M5 instances, and memory optimized R3, R4, R5 instances.

Aurora PostgreSQL supports limited instance classes, including T3, R4, and R5 instances. In many Regions, Amazon RDS for PostgreSQL and Aurora PostgreSQL support AWS Graviton2 based instances. Graviton2 processors are custom built by AWS using 64-bit Arm Neoverse cores to deliver the best price performance for your workloads.

Amazon RDS for PostgreSQL offers more instance types so you can choose the appropriate size for your workload. If your database workload requires a smaller instance class that isn't available with Aurora PostgreSQL, Amazon RDS for PostgreSQL is a better choice. (ref 6)

RESULTS:

Databases (2) Group resources Modify Actions Restore from S3 Create database

Filter by databases

DB identifier	Status	Role	Engine	Region & AZ	Size	Actions
aurora	Available	Regional cluster	Aurora MySQL	us-west-2	1 instance	3 Actions
aurora-instance-1	Available	Writer instance	Aurora MySQL	us-west-2a	db.t3.small	-

Endpoints (2) Actions Create custom endpoint

Find resources

name	Status	Type	Port
aurora.cluster-cmuqulgykpv.us-west-2.rds.amazonaws.com	Available	Writer instance	3306
aurora.cluster-ro-cmuqulgykpv.us-west-2.rds.amazonaws.com	Available	Reader instance	3306

➤ AURORA CREATES TWO DIFFERENT CLUSTERS

Manage IAM roles

☒ Select IAM roles to add to this cluster
Add an existing IAM role to this cluster.

☐ Select a service to connect to this cluster

END OF THE LAB:

```
Query OK, 1 row affected (0.01 sec)

MySQL [world]>
MySQL [world]> INSERT INTO 'country' VALUES ('THA','Thailand','Asia','Southeast Asia',513115.00,1350,61399000,68.6,116416.00,153907.00,'Prathet Thai',
,'Constitutional Monarchy',3320,'TH');
Query OK, 1 row affected (0.00 sec)

MySQL [world]>
MySQL [world]> INSERT INTO 'country' VALUES ('CRI','Costa Rica','North America','Central America',51100.00,1821,4023000,75.8,10226.00,9757.00,'Costa
Rica','Republic',584,'CR');
Query OK, 1 row affected (0.00 sec)

MySQL [world]>
MySQL [world]> INSERT INTO 'country' VALUES ('AUS','Australia','Oceania','Australia and New Zealand',7741220.00,1901,18886000,79.8,351182.00,392911.0
0,'Australia','Constitutional Monarchy, Federation',135,'AU');
Query OK, 1 row affected (0.02 sec)

MySQL [world]> SELECT * FROM country WHERE GNP > 35000 and Population > 10000000;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Code | Name | Continent | Region | SurfaceArea | IndepYear | Population | LifeExpectancy | GNP | GNPold | LocalName |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| AUS | Australia | Oceania | Australia and New Zealand | 7741220.00 | 1901 | 18886000 | 79.8 | 351182.00 | 392911.00 | Australia |
| THA | Thailand | Asia | Southeast Asia | 513115.00 | 1350 | 61399000 | 68.6 | 116416.00 | 153907.00 | Prathet Thai |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MySQL [world]>
```

Process Time:



```
Query OK, 0 rows affected (0.05 sec)

MySQL [world]> INSERT INTO `country` VALUES ('GAB','Gabon','Africa','Central Africa',267668.00,1960,1226000,50.1,5493.00,5279.00,'Le Gabon','Republic',902,'GA');
Query OK, 1 row affected (0.05 sec)
```

REFERENCES

1. <https://aws.amazon.com/tr/training/restart/>
2. <https://aws.amazon.com/blogs/database/is-amazon-rds-for-postgresql-or-amazon-aurora-postgresql-a-better-choice-for-me/>
3. [Amazon](#) RDS
4. https://aws.amazon.com/tr/rds/aurora/features/#High_Performance_and_Scalability
<https://aws.amazon.com/tr/rds/aurora/>
5. <https://www.percona.com/blog/when-should-i-use-amazon-aurora-and-when-should-i-use-rds-mysql/>
6. <https://tutorialsdodojo.com/amazon-aurora-vs-amazon-rds/>