**Domain: Databases**

**Topic: RDS**

**Service overview**

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 AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

Amazon RDS takes over many of the difficult and tedious management tasks of a relational database:

* When you buy a server, you get CPU, memory, storage, and IOPS, all bundled together. With Amazon RDS, these are split apart so that you can scale them independently. If you need more CPU, less IOPS, or more storage, you can easily allocate them.
* Amazon RDS manages backups, software patching, automatic failure detection, and recovery.
* To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances. It also restricts access to certain system procedures and tables that require advanced privileges.
* You can have automated backups performed when you need them, or manually create your own backup snapshot. You can use these backups to restore a database. The Amazon RDS restore process works reliably and efficiently.
* You can use the database products you are already familiar with: MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server.
* You can get high availability with a primary instance and a synchronous secondary instance that you can fail over to when problems occur. You can also use MariaDB, Microsoft SQL Server, MySQL, Oracle, and PostgreSQL read replicas to increase read scaling.
* In addition to the security in your database package, you can help control who can access your RDS databases by using AWS Identity and Access Management (IAM) to define users and permissions. You can also help protect your databases by putting them in a virtual private cloud.

**Usecases**

*Basically, RDS is a SQL database managed by Amazon. So usecases are the same as SQL db’s usecases (e.g. situations where data integrity is absolutely paramount, like financial applications, defense and security, private health information, or highly structured data).*

**Constrains**

* By default, you can have up to a total of 40 DB instances. RDS DB instances, Aurora DB instances, Amazon Neptune instances, and Amazon DocumentDB instances apply to this quota.
* The maximum number of simultaneous database connections varies by the DB engine type and the memory allocation for the DB instance class. Refer to the [documentation](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Limits.html).
* There are some naming constraints in Amazon RDS. Refer to the [documentation](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Limits.html).

**Task: Setup highly available RDS cluster**

**Problem to Be Solved**

*You’re a system engineer. Company you work for badly needs a relational database in AWS cloud that can run their critical production workloads without worrying about a database becoming unavailable due to the failure of a database component. For critical application workloads, a couple of minutes of application downtime can result in significant revenue losses. To reduce the load on the source instance and improve performance for read-heavy workloads, Cloud Databases users can redirect writes and reads to source and replica instances within the HA setup, respectively.*

## Explanation of the Solution

[*Storage autoscaling*](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_PIOPS.StorageTypes.html#USER_PIOPS.Autoscaling)

[*Read replicas*](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html)

[*Multi-AZ deployment*](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html)

[*Difference between read replicas and multi-AZ deployment*](https://medium.com/awesome-cloud/aws-difference-between-multi-az-and-read-replicas-in-amazon-rds-60fe848ef53a)

## Implementation Details

* *Open RDS dashboard*
* *Click “create database”*
* *Choose mysql as an engine*
* *Choose “dev/test” template*
* *Input a strong admin password*
* *Choose “burstable class” as your db instance class and choose “db.t3.small” as instance type*
* *Choose “Create a standby instance” at “Availability & durability”*
* *Create new security group at “Connectivity” and type some valid name for it*
* *Click “create database”*
* *Wait until database status be “Ready”*
* *Run virtual machine in chosen VPC*
* *Install mysql client on it*
* *Connect to the RDS instance*
* *Reboot RDS with failover enabled*

## Benefits / Outcomes / Pros and Cons / Summary

*Highly available solutions are mostly used at production environments in order to make them more resilient and stable in a case of emergency. But every single 9 in your SLA makes your infrastructure cost more. In case of RDS to provide high availability to your instance your bill grows significantly with each extra instance.*

## Pricing

[*Link to cost estimation of HA RDS deployment with 3 instances with 100 GB (about ~370$ per month)*](https://calculator.aws/#/estimate?id=0452daa80baacaac5a6365e8ab437f69bd16cc92)

## Tearing down

* *Select instance you created*
* *Click “Actions” -> “Delete”*
* *Do not create final snapshot and retain automated backups*