AWS Database

A system in which we can access the structured data is called a database. In other words, a database is an organized collection of data in which we can fetch the information based on the desired format and queries.

It is used to store data in a specific format to create conclusions from this stored data with queries made in its own language.(e.g. SQL)

### **Type of Database**

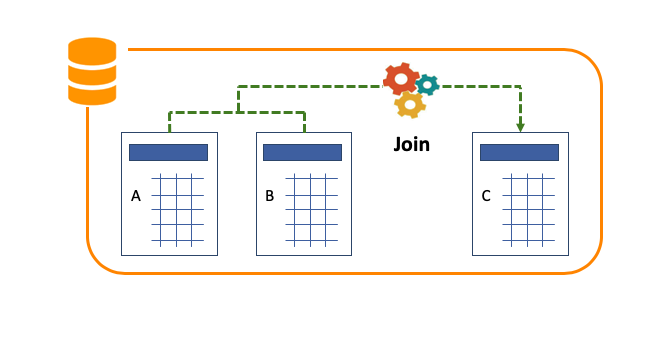
In the database environment, there are two leading types of Database storage:

* **Relational Database -SQL**
* **Non-Relational Database-NoSQL**

We call them SQL and NoSQL, referring to whether or not they’re written solely in structured query language SQL.

SQL stands for **Structured Query Language** and it is a standard language for dealing with Relational Databases. So we usually prefer to call Relational Database as SQL and Non-Relational Database as NoSQL.

### **What is SQL?**



**Relational Databases:**

* Relational databases are the oldest and currently the most widely used database type. Relational databases store data as rows and columns, just like in a Microsoft Excel Sheet.
* Tables in SQL are pre-determined and created as a schema. The user can store the data in this database by adhering to these schemes. Thanks to SQL (Structured Query Language), which is the basic language of relational databases, data can be imported into these tables, read or updated, etc.
* The main reason why this type of database is called relational is that it concludes by using separate tables as you see in the picture above.
* These processes are called **Join** in the database environment and the main power of relational databases is due to this **Join** process.
* Almost 90% of the database environment consists of these relational databases.
* But they have some disadvantages:
  + First of all, we have to update/add data into the database according to a determined scheme. Therefore, the requests must be determined correctly in advance. This causes low flexibility.
  + Besides, it requires strict coordination with database developers.
* Therefore, SQL presents a huge challenge in dynamic and constantly renewed workspaces.

### **What is NoSQL?**

As the name suggests, Non-Relational(NoSQL) database is a type of database that is not based on SQL. The Non-Relational database is a new database model that is getting popular in recent years. It is designed to solve the problems caused by the relational databases depend on predetermined schemes and it has been improved and is becoming increasingly common.

The Non-Relational database keeps data as key data mappings in documents rather than tables. So, instead of tables, there are documents called **Collections** that store data in **JSON** format.

Assume that, if we say Microsoft Excel Sheet for SQL, we can say Microsoft Word for NoSQL.

As for the advantage of NoSQL; Unlike SQL, NoSQL eliminates the need to coordinate with the Database developer when writing database programs.

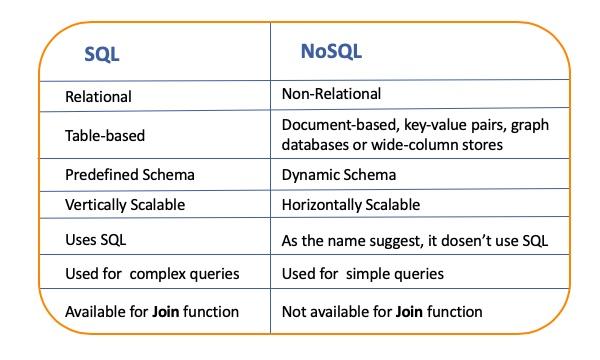
Also, we don't need to know the SQL language.

But, in NoSQL, we can not use the **Join** process and lose the opportunity to make relational data analysis over multiple tables.

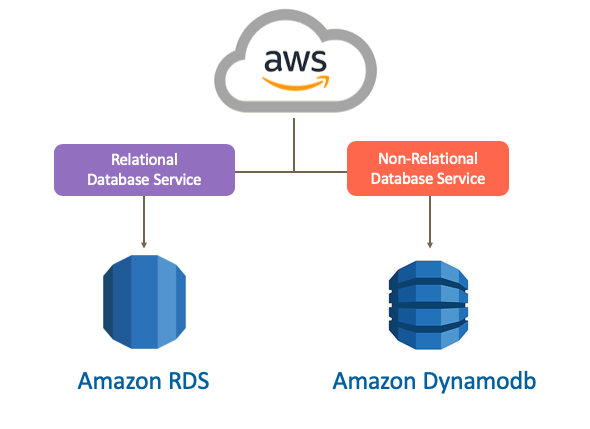
The main difference between SQL and No-SQL is the **Join** process.

While SQL is used to query complex data, NoSQL is designed for simpler databases.

### **SQL vs. NoSQL**



### **Database Options in AWS**



AWS supports both SQL and NoSQL database types.

AWS provides a relational database solution with **Amazon RDS** service. Amazon RDS offers different types of relational database engines (Oracle, Microsoft SQL Server, MySQL, e.g. ) which are widely used in the world.

Besides, Amazon Aurora, which is built on the open-source MySQL and PostgreSQL infrastructure and AWS's database engine, serves under the Amazon RDS service.

Normally, you can install these databases inside EC2 machines. However, the Amazon RDS service has taken the management burden of these databases from customers and made this service a SaaS (Software as a Service) service.

As for NoSQL, the service offered by AWS is **Amazon DynamoDB**. Unlike the SQL database service, AWS offers only its DynamoDB option in the NoSQL field.

Amazon RDS

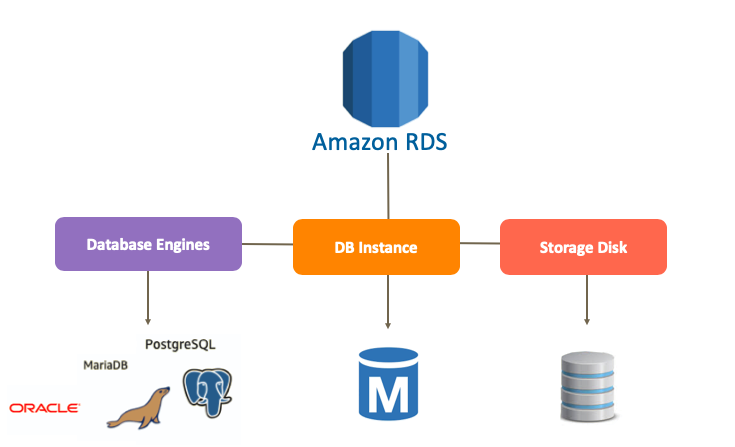
Amazon RDS is the SQL database service managed by AWS.

Since Amazon RDS base is a managed service as we said above, the setup, resource management and maintenance operations of the computer environment where the database runs are performed by AWS.

Naturally, the design and data modification processes of the database belong to the user.

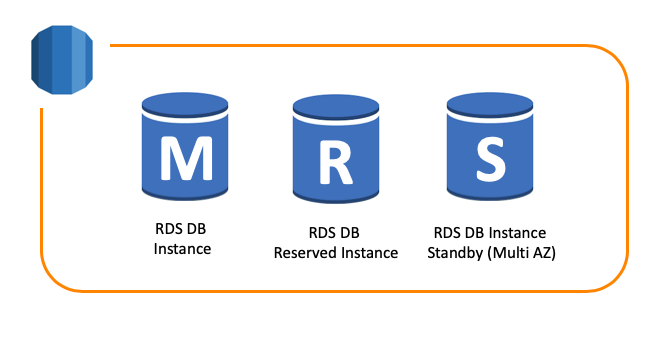
Amazon RDS enables to use of popular database engines in the database environment like Oracle, Microsoft SQL Server, MySQL, PostgreSQL, and MariaDB. Amazon RDS also offers its own relational database product, Amazon Aurora.

### **Components of Amazon RDS**



* Amazon RDS basically consists of 3 main components:
  + The first and most basic component is a suitable **Database Engines** for our needs such as MySQL, PostgreSQL, MariaDB, etc. It is a kind of database software.
  + The second component is the **DB Instance** on which the database software we choose will run.
  + The third component is the virtual **Storage Disk** which will be connected to the DB instance.
* In fact, viewed from the perspective of the components, we can say the Amazon RDS is similar to EC2

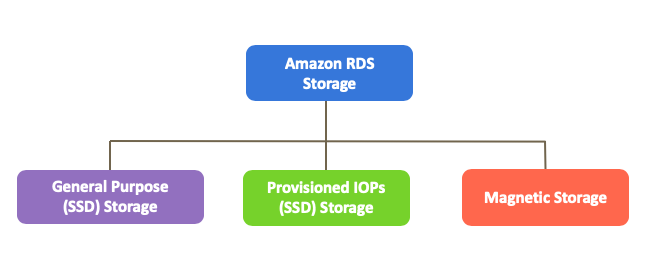
Relational Database Instance



You can think of a DB instance as a database environment in the cloud with the compute and storage resources you specify.

* According to the size of our database, we select a DB Instance considering the CPU and RAM power in the RDS environment.
* RDS offers **On-Demand** and **Reserved Instance** options.
* **Start** and **Stop** status are available just like EC2 virtual machines, So we can stop for a while and run it again. However, in the Amazon RDS service, a DB instance can only remain in "Stop" status for 7 days. If the machine is not put into operation after the 7th day, the machine is automatically started.
* RDS also allows the DB instance to be transferred to a more advanced and powerful machine without any interruption. This feature is valid for increasing the storage disk size.
* You can create and delete DB instances, define/refine infrastructure attributes of your DB instance(s), and control access and security via the AWS Management Console, Amazon RDS APIs, and AWS Command Line Interface.
* You can run one or more DB instances, and each DB instance can support one or more databases or database schemas, depending on the engine type.

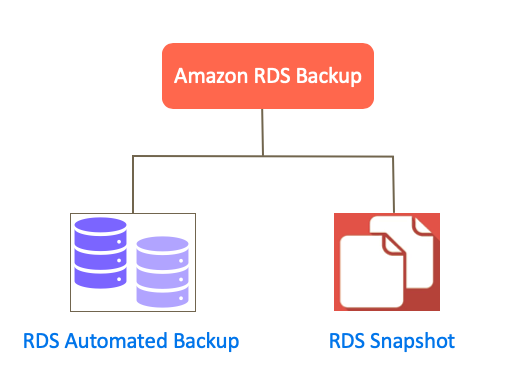
### **Amazon RDS DB Instance Storage**



There is an important factor in the databases as much as CPU and RAM power, which is the value of IOPs. We determine the IOPs value while choosing a storage disk. There are 3 types of Storage disks in Amazon RDS:

* **General Purpose (SSD) Storage:**
  + General Purpose SSD storage offers **cost-effective storage** that is acceptable for most database workloads.
  + RDS offers the **General Purpose (SSD) Storage** option, which has a storage capacity of 20 GB to 64 TB as part of the disk infrastructure and gives a value of **3 IOPS per GB**, provided that it has an upper limit of up to 3000 IOPS in total.
* **Provisioned IOPs (SSD) Storage:**
  + For production application that requires **fast and consistent I/O performance**, AWS recommends Provisioned IOPS storage. Provisioned IOPS storage is a storage type that delivers predictable performance and consistently low latency.
  + Amazon RDS offers the **Provisioned IOPs (SSD) Storage** option, with 1000 to 80000 IOPs and a storage capacity of 20GB to 64 TB.
* **Magnetic Storage:**
  + Amazon RDS also supports magnetic storage for backward compatibility. AWS recommends that you use General Purpose SSD or Provisioned IOPS SSD for any new storage needs.
  + Magnetic storage doesn't allow you to scale storage when using the SQL Server database engine. It is limited to a maximum size of 3 TB and 1,000 IOPS. And it also doesn't support elastic volumes.

### **Automated Backups**



Amazon RDS provides two different methods for backing up your DB instance:

**Automated Backups:**

Thanks to Automated Backups, RDS performs the necessary updates and maintenance of the database at the time intervals we will choose. AWS call this time interval as a **Backup Window**.

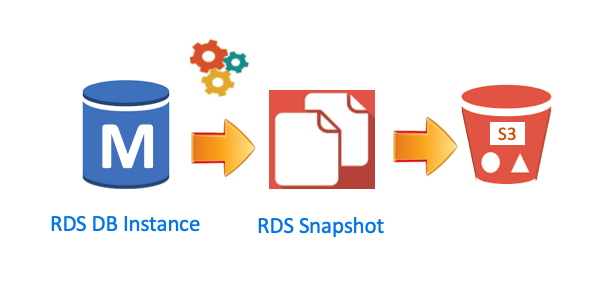
We can choose a backup window while creating a DB instance. In this backup window, RDS provides an automated backup opportunity so that we can return up to 35 days.

In addition to creating a complete copy of the database in automated backups, RDS backs up the **Transaction Log** records on S3 every 5 minutes, where all the transactions made in the database are kept. In this way, we have an opportunity to return to a desired moment of the day in seconds.

When you initiate a point-in-time recovery, transaction logs are applied to the most appropriate daily backup in order to restore your DB instance to the specific time you requested.

\*\*\* In the RDS environment, system restores are never made on the existing RDS DB instance. Instead, a new DB instance is created and registered for each occasion.

**Database Snapshots**

Amazon RDS supports manual backups as well as automated backups. Amazon RDS allows taking Snapshot of RDS DB instance at any time.

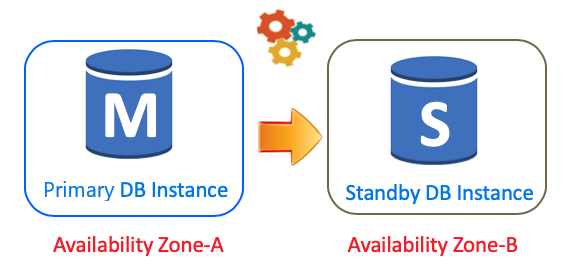
DB Snapshots are user-initiated and enable you to back up your DB instance in a known state as frequently as you wish, and then restore to that specific state at any time.

You can also identify DB Snapshots as an Automated Snapshot Type. It take Snapshot in a specific time you determined.

* When the RDS DB instance is deleted, an Automated Backups are deleted along with it. However, DB Snapshots (taken manually) remain on AWS even if the RDS DB instance is deleted.

Amazon RDS DB snapshots and automated backups are stored in S3.

### **RDS Multi-AZ Deployment**



Thanks to the Multi-AZ Deployment, RDS creates a replica of the **primary database** for multiple Availability Zones within the same region. We call the second replica database as a **standby database**.

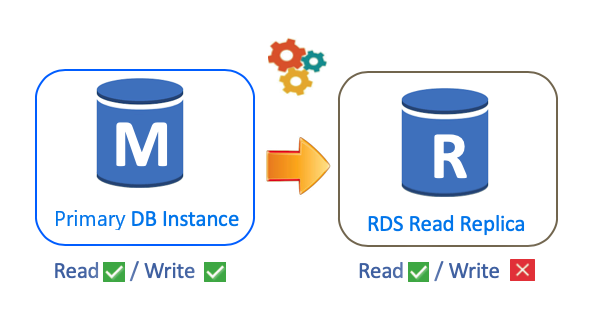
Amazon RDS automatically provisions and maintains a synchronous standby replica. Every information recorded in the primary database is synchronized instantly to the standby database located in the other AZ.

However, only the **primary database** responds to data queries. The standby database is promoted in failover scenarios. If a fault occurs in the primary database, the standby database is automatically activated.

Also, automated backups are taken from the standby database. In this way, possible interruptions and delays are also prevented.

* Multi-AZ Deployment certainly doesn't provide an increase of performance like a Load Balancer. It only ensures the continuity of the data flow.

### **Amazon Read Replicas**



Actually, the main workloads of the databases are caused by reads.

For example, while looking for something on the Amazon website you make dozens of read processes from the Amazon databases. But only data is being written when you purchase something.

So, the biggest load of a database is to fetch data from it, that is, to read.

That's why RDS offers Read Replica. Thanks to Read Replica, RDS allows creating asynchronous database copies that we can create in the same or different regions.

But, unlike Multi-AZ, Read Replicas are designed to serve read traffic. Data writing processes are made only through our **source database**, and then Read Replica is synchronized with our source DB instance.

* Unlike RDS Multi-AZ feature, Amazon Read Replica improves database performance.