UNIT –V

STORAGE, INDEXING, QUERY PROCESSING, AND EMERGING TRENDS

Amazon RDS - Introduction to Amazon Relational Database System

Amazon RDS Is a relational database management system along with the facilities of the AWS cloud platform. It facilitates us in creating database instances as per our requirements, i.e. resizable, variety of database types, etc.

What is Amazon Relational Database Service (Amazon RDS)?

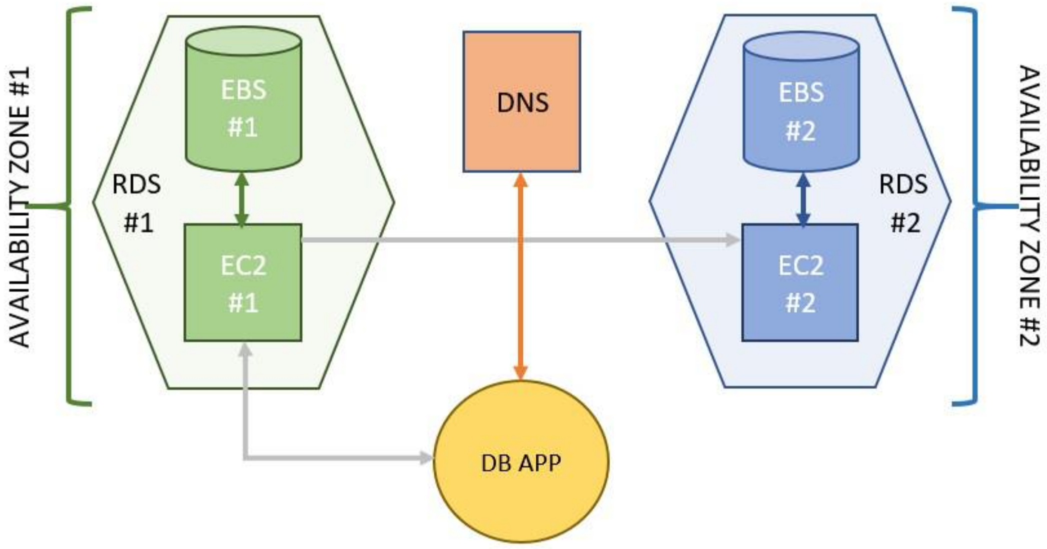
Amazon Web Services offers Amazon RDS a service where it is managed completely by AWS and also it offers wide range data base engines like the following:

1. MySQL.
2. PostgreSQL.
3. Oracle.
4. SQL Server.

The backup of the data and the infrastructure will be taken care of by the AWS scaling and balancing the load the security is very high the data will be encrypted at rest can control the accesses to the data with the help of IAM(Identity Access Management). The DR (Disaster Recovery) will automatically take by the AWS automatically by the AWS

How Amazon RDS Works?

Traditionally, database management used to be a very scattered service, from the webserver to the application server and then finally to the database. For the maintenance of such a vast system a team was required, to shrink this workforce, AWS came across an amazing all-in-one service, RDS. The whole architecture of RDS includes every aspect of the traditional management system, all in place. Thus, it includes everything from **EC2 (Elastic Compute Cloud)** to **DNS (Domain Name System).** Every part of the RDS architecture has its own separate set of features completely different from each other. A diagrammatical representation of RDS has been attached ahead.



Comparison of responsibilities with Amazon EC2 and on-premises deployments

Responsibility	Amazon EC2	On-Premises Deployments
Infrastructure Management	AWS handles the physical hardware also networking and infrastructure management	Organizations manage all aspects of hardware also networking and infrastructure
Scalability	EC2 offers easy scaling with just a few clicks or automatically based on demand	Scaling requires purchasing, installing and configuring additional hardware
Security	AWS provides security for the cloud infrastructure including physical security and offers built-in encryption	Organizations are responsible for physical security also network security and software security
Patching and Maintenance	AWS manages the underlying hardware and applies updates to the infrastructure	IT teams handle patching, updates and maintenance for both hardware and software
Cost Structure	Pay-as-you-go with no upfront hardware costs	Significant upfront investments for hardware, with ongoing maintenance costs

Amazon RDS shared responsibility model

The Amazon RDS shared responsibility model divides responsibilities between AWS and the customer ensuring the security and availability of database services.

- **AWS Responsibilities**: AWS is responsible for managing the infrastructure which includes the underlying hardware data centers and network security. AWS also handles database software installation, patching and automated backups. This includes physical security failover management and ensuring high availability.
- **Customer Responsibilities**: The customer is responsible for managing the data stored within the database including encryption access controls and compliance with data governance policies. Customers also need to configure security settings fine-tune database performance and optimize queries to ensure efficient usage.

This model ensures that AWS provides a secure and reliable platform while giving customers control over their data application-specific configurations and access management.

Exploring the DB Engines , DB Instance Classes and DB instance storage

DB engines

Amazon RDS offers a range of DB engines that runs on your DB instance. Some of them are Listed Below

- MySQL
- PostgreSQL
- Oracle
- MariaDB
- PostgreSQL
- IBM Db2

Each of these engines allows developers to leverage familiar technology while enjoying the benefits of cloud management through Amazon RDS.

DB instance classes

A DB instance class contain DB instance class type and the DB instance class size. we explained in below table also the asterisk (*) represents the generation, optional attribute, and size

Instance Class Type	**Description**	**Use Cases**
*General Purpose (db.m)**	Balanced compute and memory resources, suitable for a variety of applications	Ideal for most workloads requiring moderate performance
Memory Optimized (db.r, db.x)**	High memory-to-compute ratios designed for in-memory databases and large data sets.	Suitable for memory-intensive applications needing fast access to data
*Compute Optimized (db.c)**	Prioritizes compute power over memory ideal for high CPU performance tasks.	Best for compute-heavy workloads like complex calculations.
*Burstable Performance (db.t)**	Variable performance operating at baseline levels with the ability to burst for higher performance.	Cost-effective for workloads with occasional performance spikes.

DB instance storage

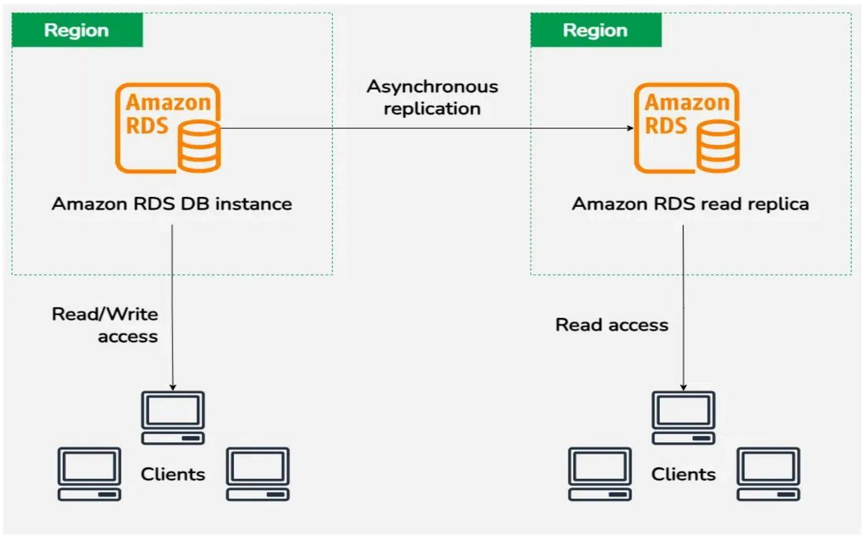
DB instance storage comes in the following types

Storage Type	**Description**	**Use Cases**
General Purpose (SSD)	Cost-effective, SSD-based storage that offers a good balance of performance for a wide range of workloads	Ideal for development, testing, and medium-sized database instances
Provisioned IOPS (PIOPS)	High-performance SSD storage designed for workloads that require low latency and consistent throughput.	Best for I/O-intensive and critical production environments.
Magnetic (Legacy)	Older storage option that is now primarily used for backward compatibility with existing systems	Recommended to transition to SSD storage for modern applications.

Understanding AWS Regions, Availability Zones and Multi-AZ Deployments in Amazon RDS.

AWS Regions and Availability Zones

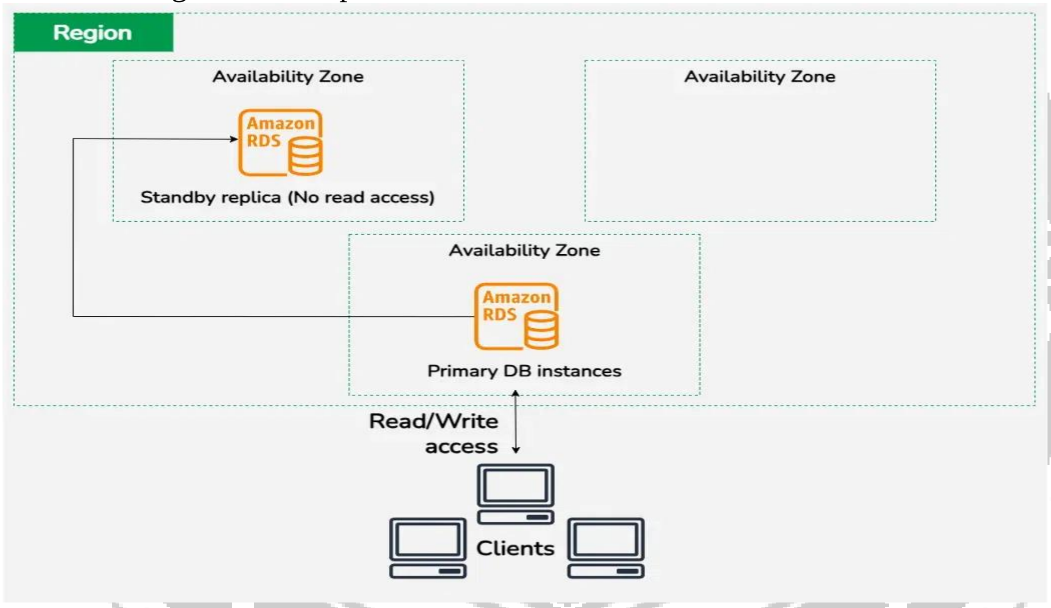
AWS Regions are geographically separated locations that contain multiple Availability Zones. Each region is isolated from the others to ensure geographical redundancy. When you create an RDS instance you select a region which defines where your database physically resides. Availability Zones are distinct data centers within a region each with independent power, cooling and networking. RDS can deploy your databases across multiple AZs to ensure high availability. The image above illustrates Amazon RDS's cross-region replication which helps enhance availability and data durability across regions.



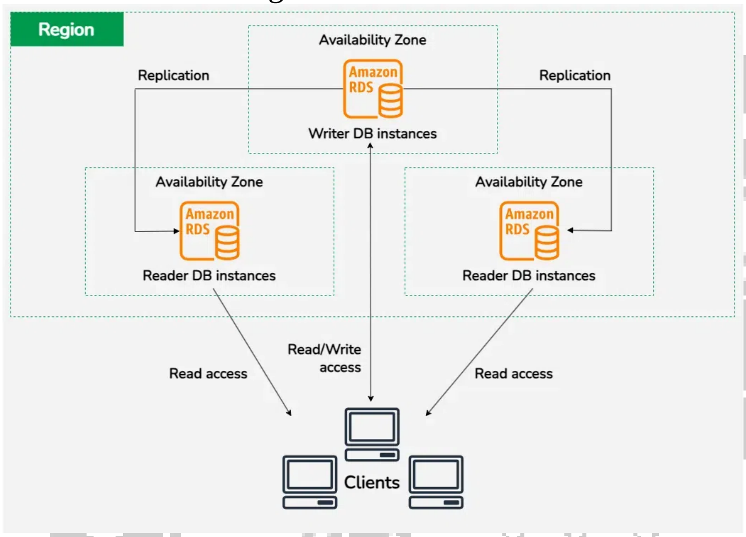
This configuration provides a resilient solution to scale applications globally while ensuring data redundancy and improved fault tolerance

Multi-AZ deployments

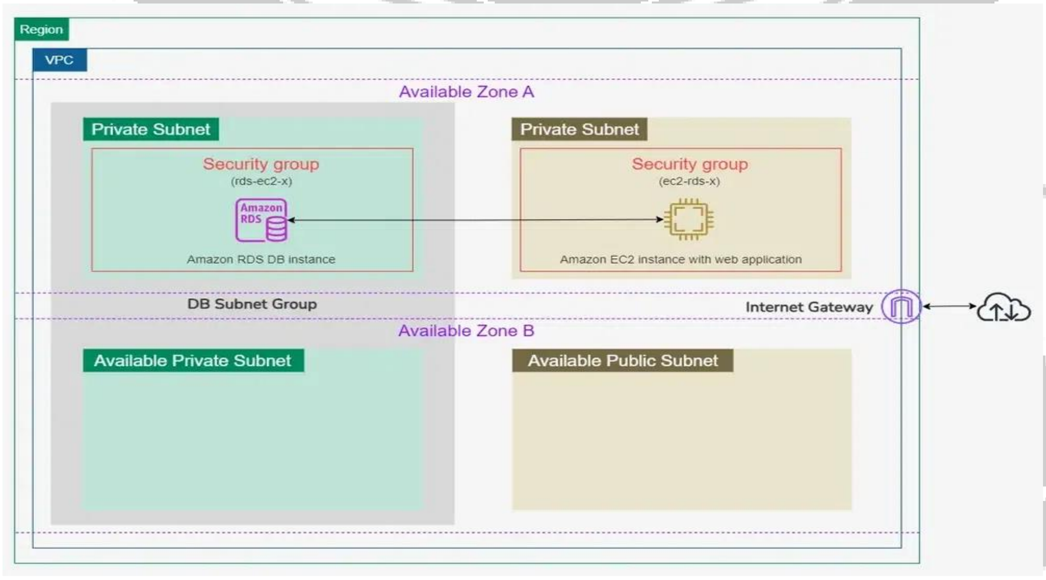
In Multi-AZ deployments with Amazon RDS your primary database is automatically replicated to a standby instance in another Availability Zone within the same AWS Region. The primary instance handles read/write access while the standby instance stays in sync for disaster recovery. If the primary instance or its AZ fails RDS automatically switches to the standby instance ensuring minimal downtime. Check the diagram below for a clearer understanding of this setup



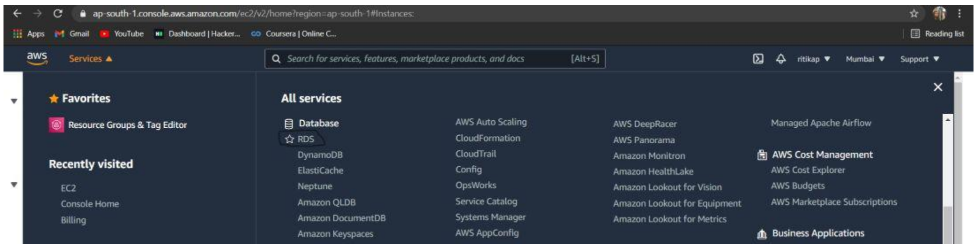
In a **Multi-AZ DB cluster** deployment, there is one writer DB instance and two reader DB instances spread across three separate Availability Zones within the same AWS Region. The writer instance handles both read and write requests, while the reader instances are dedicated to handling read traffic.

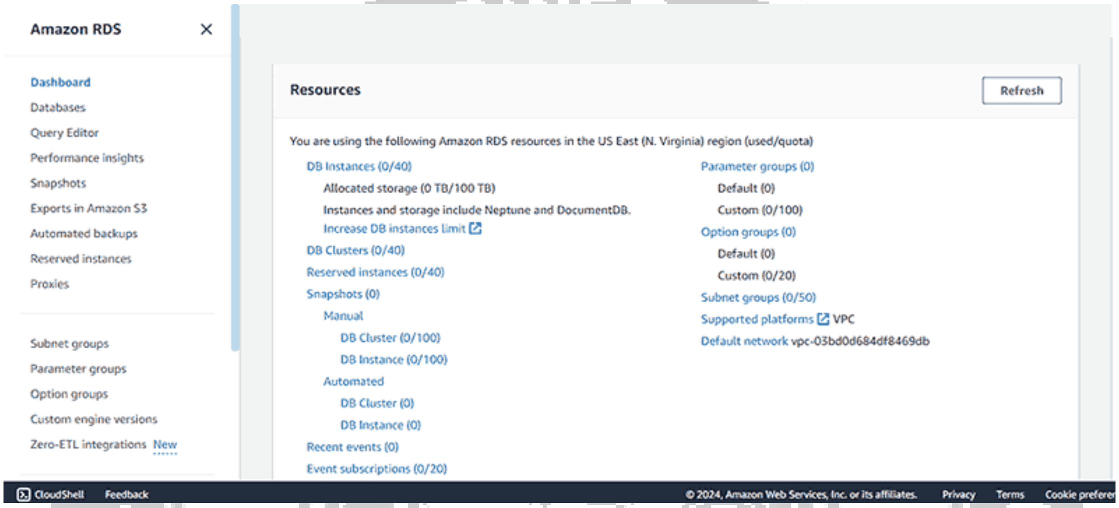


Access control with security groups For RDS DB Instances
Security groups play a crucial role in controlling network access between EC2 instances and RDS DB instances within a VPC. Below Is the Architecture to where its Clearly show how Security Groups configure for EC2 and RDS In VPC



For more information about security groups, Please Refer security groups Creation
Ways to Interact with Amazon RDS
You can interact with Amazon RDS in multiple ways
AWS Management Console
The AWS Management Console provides an easy-to-use web-based interface to manage and monitor your RDS instances including creating databases, managing backups and scaling instances





Resources

You are using the following Amazon RDS resources in the US East (N. Virginia) region (used/quota)

- DB Instances (0/40)
  - Allocated storage (0 TB/100 TB)
  - Instances and storage include Neptune and DocumentDB.
  - Increase DB instances limit
- DB Clusters (0/40)
- Reserved instances (0/40)
- Snapshots (0)
  - Manual
    - DB Cluster (0/100)
    - DB Instance (0/100)
  - Automated
    - DB Cluster (0)
    - DB Instance (0)
- Recent events (0)
- Event subscriptions (0/20)
- Parameter groups (0)
  - Default (0)
  - Custom (0/100)
- Option groups (0)
  - Default (0)
  - Custom (0/20)
- Subnet groups (0/50)
- Supported platforms VPC
- Default network vpc-03bd0d684df8469db

Command line interface

The AWS CLI enables users to automate tasks by executing commands in scripts. For more information Please Refer to this Configure CLI for RDS

Amazon RDS APIs

Amazon RDS API allows developers to programmatically manage and interact with their RDS instances offering seamless integration into applications and services.

Use Cases Of Amazon RDS (AWS)

Below are some use cases of Amazon RDS mostly used for secured and highly configured applications like gaming servers and health and financial applications.

1. **WebApplication:** The Amazon RDS is mainly used for the backend for web applications where it can support maximum no.of in and output operation. And also is easy to scale up and down.
2. **Managed Database:** Instead of you managing the database AWS will provide Amazon RDS as a service by just doing some configuration your database will be available to perform the operations.
3. **Isolation:** You can integrate and configure multiple applications with secure isolation by protecting the data of each application's customers while managing the underlying infrastructure.
4. **Highly Secured:** You can use Amazon RDS for domains like health care and banking because the data used in this type of application is highly secure which can be achieved with the help of AWS RDS.

Features Of Amazon RDS

The following are some key features of Amazon RDS:

- **Availability:** The **"Automated Backup"** feature of RDS makes the recovery of the database instance much easier and makes it available for access quickly. Other than that, **"Database Snapshots"** are user-driven backup features initiated by Amazon RDS, which makes it easier for the user to monitor all the alterations made on the Database Instance. These snapshots can be shared among multiple AWS accounts in order to expand the availability of the DB instance, along with maintaining the security of the confidential data.

- **Security:** While creating a new database, you have to create a password that is totally restricted and known to you only. And by default, you are given the **"Admin role"** which has the maximum authority on that particular database. Amazon RDS also allows its users to encrypt the databases using **"keys"** which is managed by **KMS** (Key Management Service) under Amazon RDS.
- **Backups:** RDS provides us the facility to have backups. We can have backups in multiple forms. **Snapshots** are basically non-editable backups used for maintaining records. We also can create **Automated Backups** simply by altering the configurations during creating the database. **Reserved instances** are also another type of backup facility available here.
- **Scalability:** RDS enables us to automatically scale up or scale down depending upon the number of transactions happening on your database per minute. We can do both **"Horizontal Scaling"** and **"Vertical Scaling".** Let us go through the difference between both of them.
    - **Horizontal Scaling** deals with scenarios where the amount of traffic is increased on your database exponentially, in such cases, this scaling comes into the picture. This simply creates multiple hardware & software which are look-alike of the previously existing ones on the cloud in order to tackle the traffic.
    - **Vertical Scaling** deals with situations, where the traffic is not very much increased but the current configurations of the hardware & software are not able to handle the demands of the client anymore. Using this scaling method, we are capable of adding additional storage and processors to our pre-existing resources.
- **Performance:** RDS gives two SSD-backed storage options for its users, i.e. **General Purpose & Provisioned.** All these variants directly impact the level of performance of the resource and its attached services. The general SSD is very cost-effective and is used at places where a broad workforce is required. Provisioned, as the name suggests are designed for temporary or lower workloads purposes.
- **Pricing:** RDS only asks you to pay for what you use, once you are done with a certain resource delete it and don't pay for it anymore. There is no compulsory minimal charge decided for using RDS. Depending upon the **Database Engines** and the type of database, a bill is calculated and sent to you at the end of the month. For free tier accounts, special configurations are bound to choose and you won't get any bills if you delete all the resources you used before logging out.

Amazon RDS Alternatives

1. **MySQL -** It is the 2nd most preferred open-source RDBMS in the world. It is developed by Oracle. It is not typically cloud-based in nature like Amazon RDS, i.e. it can be used on PC as well. It is also offered as one of the options on RDS to choose as Database Engine. It supports five server operating systems. The main application of MySQL is in the e-commerce domain, data warehouse, and logging application.

2. **PostgreSQL -** It is one of the oldest RDBMS. It is also one of the popularly used open-source RDBMS. It was developed by PostgreSQL Global Development Group in 1989. It is a cross-platform software, and it supports more operating systems as compared to others. Its primary focus is maintaining the security of the data and it is a vast kingdom of user-defined functions.
3. **MariaDB -** It is the most compatible RDBMS, and it supports both secondary database models, i.e. **Spatial & Graph.** It was released in 2009, by MariaDB Corporation Ab (MariaDB Enterprise). It supports a wide range of programming languages and also allows users to introduce server-side scripts. One of the best features of MariaDB is that it focuses on high-level security in the community of MariaDB continuously finding and fixing the issues for MariaDB.

All these alternatives are found useful for users to meet their requirements at a certain level. AWS introduced, RDS to ensure that the ultimate control resides in the hands of the users. RDS is not of query-driven structure rather it is more like a console in its structure.

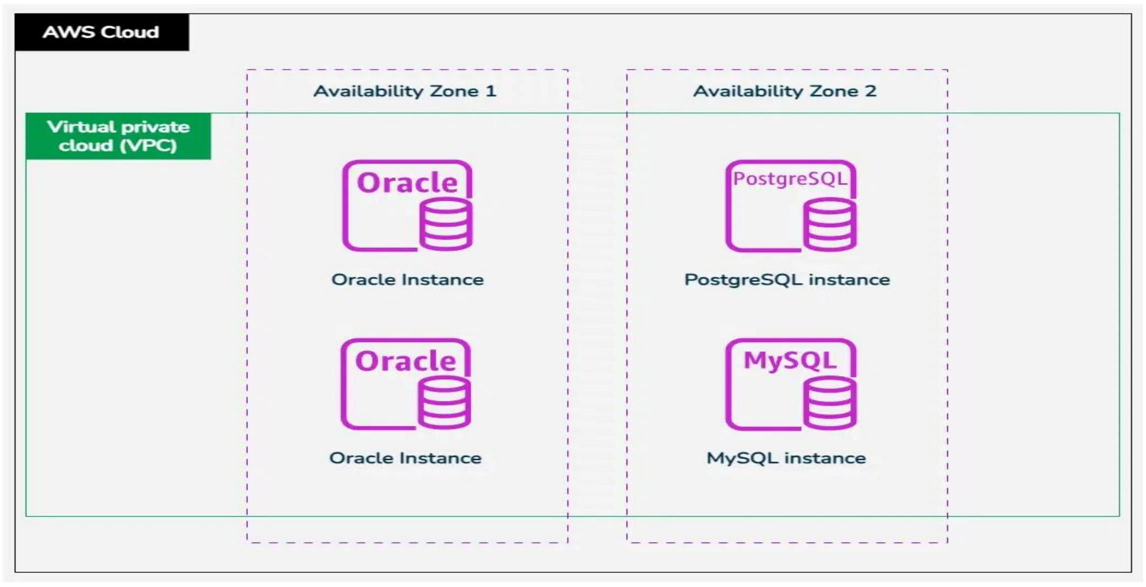
What are the drawbacks of Amazon RDS?

Here are some potential drawbacks of Amazon RDS:

1. **Limited Customization**: Since it's a managed service, customization options for server configuration and software updates are limited compared to self-hosted databases.
2. **Cost**: For large-scale deployments, the cost can increase significantly, especially when using Multi-AZ deployments or higher instance classes.
3. **Manual Scaling**: While scaling is possible, it is not fully automatic like in Amazon Aurora, and it requires manual intervention to adjust based on workload changes.
4. **Backup and Restore Time**: During heavy usage periods, backups and restores can take longer, which may affect database performance.
5. **Vendor Lock-In**: Once integrated deeply with RDS, migrating to other platforms can be complex and time-consuming

Amazon RDS database instances

Amazon RDS database instances are managed virtual servers used to run relational databases in the AWS cloud. Each instance comes pre-configured with the necessary hardware and software resources to support your chosen database engine whether it's MySQL also PostgreSQL, Oracle, SQL Server or MariaDB. RDS handles common tasks like backups also patching and database management allowing you to focus on your applications. You can choose instance sizes based on your performance needs with the option to scale up or down as required. Additionally Multi-AZ deployment options are available to ensure high availability and fault tolerance



Amazon Aurora and Amazon Aurora Server less vs Amazon RDS

Feature	Amazon Aurora	Amazon Aurora Serverless	Amazon RDS
Performance	AWS Aurora is optimized for high performance and scalability	Automatically scales to meet workload demands, performance varies based on usage	AWS RDS offers good performance but may have limitations for extremely high transaction volumes
Scalability	Automatic storage scaling up to 128 TB	Automatically scales resources up and down based on demand	Manual scaling required for instance and storage resources
High Availability	Multi-AZ replication for fault tolerance	Multi-AZ replication available, scales with demand	Multi-AZ option available but requires manual setup
Cost	Pay for provisioned instances and storage	Pay only for the capacity you use making it cost-effective for variable workloads	Cost-effective for smaller workloads but may require manual adjustments for scaling

Amazon RDS Pricing

AWS RDS pricing is designed to be flexible and cost-effective allowing users to pay only for the resources they use. Below Is The Table To Explain It in Much Better Way

Pricing Category	**Details**
Amazon RDS Free Tier	db.t2.micro, db.t3.micro, db.t4g.micro instances are included in the Free Tier, providing 750 hours per month for free usage with 20 GB of storage
Amazon RDS Pricing By Database Engine	Costs vary based on the engine used: MySQL, PostgreSQL, Oracle, SQL Server, MariaDB or Amazon Aurora.
Amazon RDS Pricing By Database Instance	Pricing depends on instance type (e.g., General Purpose, Memory Optimized), and size (micro, small, large, etc.).
Amazon RDS Pricing By Database Region	Prices differ by region. For example, RDS instances in the US East region may be cheaper than in regions like Asia Pacific (Mumbai).
Amazon RDS On-Demand Instance Pricing	Pay-as-you-go model where users are billed hourly based on the instance class and storage. Ideal for short-term or unpredictable workloads.
Amazon RDS Reserved Instance Pricing	Users commit to a specific instance class for a 1 or 3-year term in exchange for discounted rates compared to on-demand pricing.
Amazon RDS Pricing By DB Storage	
General Purpose (SSD) storage	Charged per GB provisioned per month, typically used for a wide range of database workloads
Provisioned IOPS (SSD) storage	High-performance storage optimized for I/O-intensive workloads, priced based on the IOPS provisioned and the GB stored
Magnetic storage	Legacy storage type, priced lower than SSD, but offers less performance. Charged per GB per month, often phased out for new deployments

Amazon DynamoDB

DynamoDB allows users to create databases capable of storing and retrieving any amount of data and comes in handy while serving any amount of traffic. It dynamically manages each customer's request and provides high performance by automatically distributing data and traffic over servers. It is a fully managed NoSQL database service that is fast, predictable in terms of performance, and seamlessly scalable. It relieves the user from the administrative burdens of operating and scaling a distributed database as the user doesn't have to worry about hardware provisioning, patching Software, or cluster scaling. AWS DynamoDB – Creating a Table

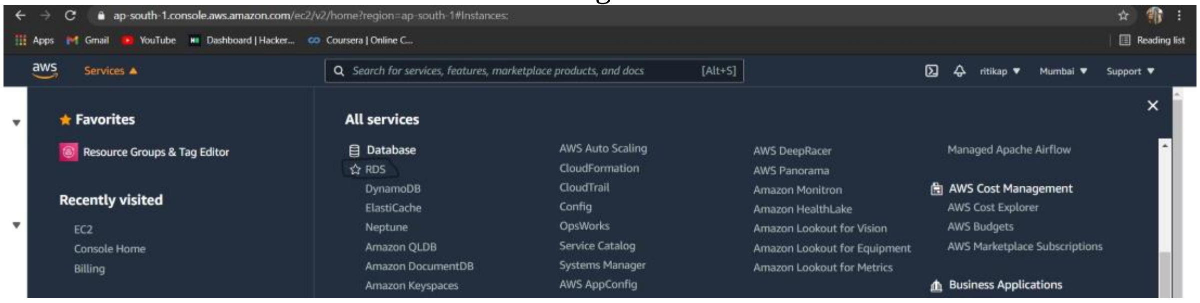
Amazon RedShift

It is a data warehouse that is based on the cloud. Amazon Redshift has a commercial license and is a part of Amazon's web services. It handles large-scale of data and is known for its scalability. It does parallel processing of multiple data. It uses the ACID properties as its working principle and is very popular. It is implemented in C language and has high availability. To Know more about Amazon RedShift refer to Amazon Redshift

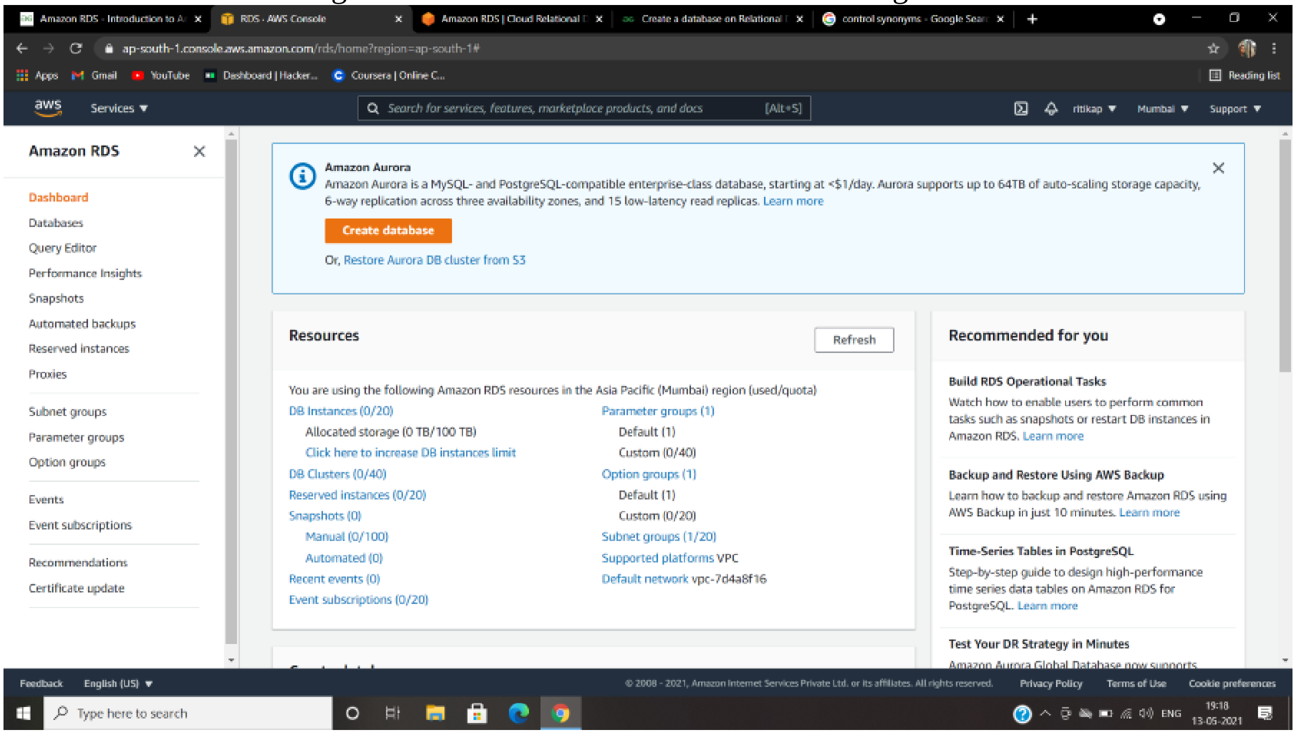
Steps To Configure Amazon RDS

Now, let us look at the AWS Relational Database Service management console.

Step 1: To reach, the RDS management console. First login into your AWS account to create AWS free tier account refer to Amazon Web Services (AWS) – Free Tier Account Set up. Once you are directed to the primary screen, at the leftmost part of it, click on **"Services".** From the long list, look for the sub-heading **"Databases"** and under it, you will find **"RDS"**. Click on it. Here is the image to refer to.



Step 2: Once you tap on RDS, in a while, you will be able to see the RDS management console. Refer to the image attached ahead for a better understanding.



Amazon Aurora

Amazon Aurora is a MySQL- and PostgreSQL-compatible enterprise-class database, starting at <$1/day. Aurora supports up to 64TB of auto-scaling storage capacity, 6-way replication across three availability zones, and 15 low-latency read replicas. Learn more

Or, Restore Aurora DB cluster from S3

Resources

You are using the following Amazon RDS resources in the Asia Pacific (Mumbai) region (used/quota)

DB Instances (0/20)	Parameter groups (1)
Allocated storage (0 TB/100 TB)	Default (1)
Click here to increase DB instances limit	Custom (0/40)
DB Clusters (0/40)	Option groups (1)
Reserved instances (0/20)	Default (1)
Snapshots (0)	Custom (0/20)
Manual (0/100)	Subnet groups (1/20)
Automated (0)	Supported platforms VPC
Recent events (0)	Default network vpc-7d4a8f16
Event subscriptions (0/20)	

Recommended for you

Build RDS Operational Tasks

Watch how to enable users to perform common tasks such as snapshots or restart DB instances in Amazon RDS. Learn more

Backup and Restore Using AWS Backup

Learn how to backup and restore Amazon RDS using AWS Backup in just 10 minutes. Learn more

Time-Series Tables in PostgreSQL

Step-by-step guide to design high-performance time series data tables on Amazon RDS for PostgreSQL. Learn more

Test Your DR Strategy in Minutes

Amazon Aurora Global Database now supports

© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

This is what the RDS dashboard looks like. On the left, there is the navigation pane to direct you to all the services under RDS. You can create your database from here, by tapping on the orange box saying, **"Create database".** For creating a database in RDS follow the linked article.