RELATIONAL DATABASE SERVICE

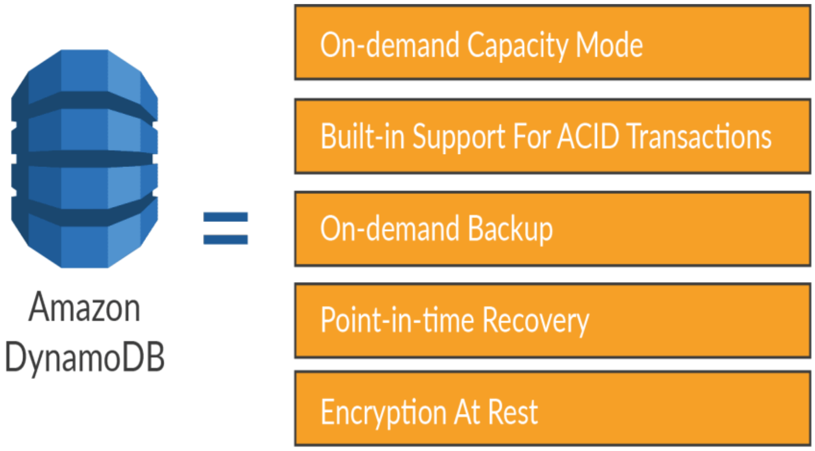
Introduction to Relational Database Service

Amazon Web service has a large spectrum of products and services to offer to businesses of all sizes and shapes. Among these, one of the most popular and remarkable services provided by Amazon is AWS RDS (Relational Database Service). With the availability of this relational database service, the tasks of setting up, operating, and scaling a relational database in the Amazon Cloud become easier for software programmers. The most common administrative tasks are automatically facilitated using hardware support, setup functioning, patching, and other database backup and functions. Businesses can scale storage and other computing resources using a single API Call using AWS RDS.

Different database services of AWS

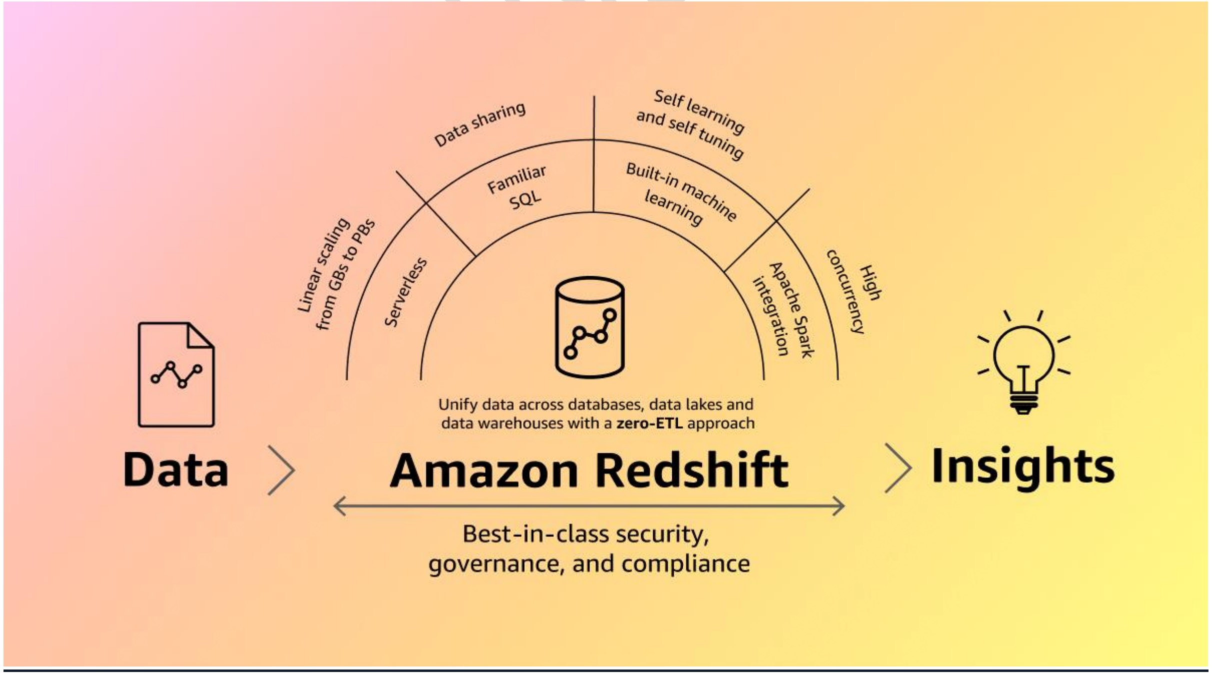
1.Amazon Relational Database Service (Amazon RDS) Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. Amazon RDS is a managed database service. It's responsible for most management tasks. By eliminating tedious manual tasks, Amazon RDS frees you to focus on your application and your users. We recommend Amazon RDS over Amazon EC2 as your default choice for most database deployments

2. **Amazon DynamoDB** Amazon Dynamo DB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling. DynamoDB also offers encryption at rest, which eliminates the operational burden and complexity involved in protecting sensitive data.



3.Amazon Redshift

Tens of thousands of customers use Amazon Redshift every day to modernize their data analytics workloads and deliver insights for their businesses. With a fully managed, AI powered, massively parallel processing (MPP) architecture, Amazon Redshift drives business decision making quickly and cost effectively. AWS's zero-ETL approach unifies all your data for powerful analytics, near real-time use cases and AI/ML applications. Share and collaborate on data easily and securely within and across organizations, AWS regions and even 3rd party data providers, supported with leading security capabilities and fine-grained governance.


Linear scaling from GBs to PBs
Data sharing
Self learning and self tuning
High concurrency
Serverless
Familiar SQL
Built-in machine learning
Apache Spark integration
Unify data across databases, data lakes and data warehouses with a zero-ETL approach
Data
Amazon Redshift
Insights
Best-in-class security, governance, and compliance

Configuring an Amazon RDS

You can create an Amazon RDS DB instance only in a virtual private cloud (VPC) based on the Amazon VPC service. Also, it must be in an AWS Region that has at least two Availability Zones. The DB subnet group that you choose for the DB instance must cover at least two Availability Zones. This configuration ensures that you can configure a Multi-AZ deployment when you create the DB instance or easily move to one in the future. To set up connectivity between your new DB instance and an Amazon EC2 instance in the same VPC, do so when you create the DB instance. To connect to your DB instance from resources other than EC2 instances in the same VPC, configure the network connections manually.

Configure automatic network connectivity with an EC2 instance

When you create an RDS DB instance, you can use the AWS Management Console to set up connectivity between an EC2 instance and the new DB instance. When you do so, RDS configures your VPC and network settings automatically. The DB instance is created in the same VPC as the EC2 instance so that the EC2 instance can access the DB instance.

The following are requirements for connecting an EC2 instance with the DB instance:

- The EC2 instance must exist in the AWS Region before you create the DB instance.

  If no EC2 instances exist in the AWS Region, the console provides a link to create one.

Amazon RDS Backup

AWS enables you to centralize and automate data protection across AWS services. AWS Backup is a fully managed, policy-based service that simplifies data protection at scale. The service is ideal for use cases such as regulatory compliance obligations, business policies for data protection, and business continuity goals.

In this how-to guide, we will use the AWS Management Console to set up automated backups of select AWS services using Amazon Relational Database Service (Amazon RDS), restore a backup, and clean up our resources to avoid unexpected costs. See this list for all the AWS and third-party services supported by AWS Backup. When going to production, remember to set up the correct schedules and retention management, and to monitor your costs.

To opt in to use AWS Backup to protect all supported resource types

1. Sign in to the AWS Management Console, and open the AWS Backup console at https://console.aws.amazon.com/backup.
2. In the left navigation pane, choose **Settings**.
3. Under **Service opt-in**, choose **Configure resources**.

4. Opt in to all AWS Backup-supported **Resources** by moving all the toggles to the right.
5. Choose **Confirm.**

Create a maintenance window (console)

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, choose **Maintenance Windows**.
3. Choose **Create maintenance window**.
4. For **Name**, enter a descriptive name to help you identify this maintenance window.
5. (Optional) For **Description**, enter a description to identify how this maintenance window will be used.
6. (Optional) If you want to allow a maintenance window task to run on managed nodes, even if you haven't registered those nodes as targets, choose **Allow unregistered targets**.

   If you choose this option, then you can choose the unregistered nodes (by node ID) when you register a task with the maintenance window.

   If you don't choose this option, then you must choose previously registered targets when you register a task with the maintenance window.
7. Specify a schedule for the maintenance window by using one of the three scheduling options.
8. For **Duration**, enter the number of hours the maintenance window will run. The value you specify determines the specific end time for the maintenance window based on the time it begins. No maintenance window tasks are permitted to start after the resulting end time minus the number of hours you specify for **Stop initiating tasks** in the next step.

   For example, if the maintenance window starts at 3 PM, the duration is three hours, and the **Stop initiating tasks** value is one hour, no maintenance window tasks can start after 5 PM.
9. Choose **Create maintenance window**. The system returns you to the maintenance window page. The state of the maintenance window you just created is **Enabled**.Connecting to an Amazon RDS