

Module 8 - Databases

[Slides](#)

Objectives / Topics

- Explain Amazon Relational Database Service (Amazon RDS)
- Identify the functionality in Amazon RDS
- Explain Amazon DynamoDB
- Identify the functionality in Amazon DynamoDB
- Explain Amazon Redshift
- Explain Amazon Aurora
- Perform tasks in an RDS database, such as launching, configuring, and interacting

Labs / Activities

- [Knowledge Check](#)
- [Lab: Build a Database Server](#) --- [Lab Instructions](#)

Section 1: Amazon Relational Database Service (RDS)

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. RDS provides you with six familiar database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB.

Managed vs Unmanaged Service

Unmanaged Challenges

- Server maintenance and energy footprint
- Software installation and patches
- Database backups and high availability
- Limits on scalability
- Data security
- Operating system (OS) installation and patches

RDS is a managed service that sets up and operates a relational database in the cloud. AWS Manages:

- OS installation and patches
- Database software installation and patches
- Database backups
- High availability
- Scaling
- Power and racking and stacking servers
- Server maintenance

RDS Features

The database instance is the basic building block of Amazon RDS. The DB instance is an isolated database environment that can contain multiple user created databases. When setting up the database, you pick an instance class and the type of storage you need for your database. You also need to specify which database engine to run: MySQL, Amazon Aurora, Microsoft SQL Server, PostgreSQL, MariaDB, or Oracle

Amazon RDS allows you to configure your DB instance for high availability with a Multi-AZ (availability zone) deployment. When you configure a Multi-AZ deployment, RDS automatically generates a standby copy of the database instance in another availability zone within the same VPC. After seeding the database copy, transactions are synchronously replicated to the standby copy. If the main database instance fails in a Multi-AZ deployment, RDS automatically brings the standby database instance

online as the new main instance.

RDS also supports the creation of read replicas. Updates that are made to the source database instance are asynchronously copied to the read replica instance. You can reduce the load on your source DB instance by routing read queries from your applications to the read replica.

Cost and Usage of RDS

When to use RDS

Use when you require:

- Complex transactions or complex queries
- A medium to high query or write rate – Up to 30,000 IOPS (15,000 reads + 15,000 writes)
- No more than a single worker node or shard
- High durability

Do not use when:

- Massive read/write rates (for example, 150,000 write/second)
- Sharding due to high data size or throughput demands
- Simple GET or PUT requests and queries that a NoSQL database can handle
- Relational database management system (RDBMS) customization

Billing

Multiple factors influence the cost of RDS

- Clock hour billing - resources incur charges when running
- Database characteristics effect cost
- DB Purchase Type
- On-Demand Instances - Compute capacity by the hour
- Reserved Instances - Low, one-time, upfront payment for database instances that are reserved with a 1-year or 3-year term
- Number of DB instances
- Provisioned storage
- No charge - Backup storage of up to 100 percent of database storage for an active database
- Charge (GB/month) - Backup storage for terminated DB instances
- Additional Storage - Charge (GB/Month) for backup storage in addition to the provisioned storage
- Number of Requests
- Deployment type – Storage and I/O charges vary, depending on whether you deploy to a single availability zone or multiple
- Data transfer – No charge for inbound, tiered charges for outbound

[Lab: Build a Database Server](#) --- [Lab Instructions](#)

Section 2: Amazon DynamoDB

- Fast and flexible NoSQL database service for any scale.
- NoSQL database tables with no limits
- Virtually unlimited storage
- Items can have differing attributes
- Low-latency queries
- Scalable read/write throughput with no limits
- Supports document and key-value store models.
- Replicates your tables automatically across your choice of AWS Regions
- Works well for mobile, web, gaming, adtech, and Internet of Things (IoT) applications
- Provides consistent, single-digit millisecond latency at any scale

Section 3: Amazon Redshift

[Amazon Redshift](#) is a fully managed, petabyte-scale data warehouse service in the cloud. The Redshift service manages all of the work of setting up, operating, and scaling a data warehouse. These tasks include provisioning capacity, monitoring and backing up the cluster, and applying patches and upgrades to the Amazon Redshift engine.

- Columnar storage and parallel processing architectures
- Automatically and continuously monitors cluster
- Encryption is built in

Section 4: Amazon Aurora

[Amazon Aurora](#) is a MySQL and PostgreSQL-compatible relational database built for the cloud, that combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases. Aurora features a distributed, fault-tolerant, self-healing storage system that auto-scales up to 64TB per database instance. It delivers high performance and availability with up to 15 low-latency read replicas, point-in-time recovery, continuous backup to Amazon S3, and replication across three Availability Zones (AZs). It also automates time-consuming tasks such as provisioning, patching, backup, recovery, failure detection, and repair.

[Knowledge Check](#)