

**Topic : Relational Database System**

**Aim :** To understand and connect to Relational Database System on AWS / GCP Cloud Platform

**Theory:**

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. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

**Amazon RDS database engines**



**Main benefits of Amazon RDS:**

Availability : It also runs on the same infrastructure as the Amazon Web Services, which tend to be very reliable. The Amazon Relational Database Service is able to duplicate data from a Multi-AZ database instance to a totally different availability zone, which is a standby instance.

Scalability : With just a few mouse clicks, you can easily scale the compute and storage resources of your database. This can also be achieved through an API call that doesn't come with any downtime. Most Amazon Relational Database Service engines will permit you to start one or multiple read replicas to unburden your original DB instance off its read traffic.

Administering : Amazon Relational Database Service is easy to administer thus, enabling you to move from the conception stage of your project to execution. To do this, you simply need to make use of the AWS Management Console, API calls or just the AWS RDS interface that requires you to enter commands.

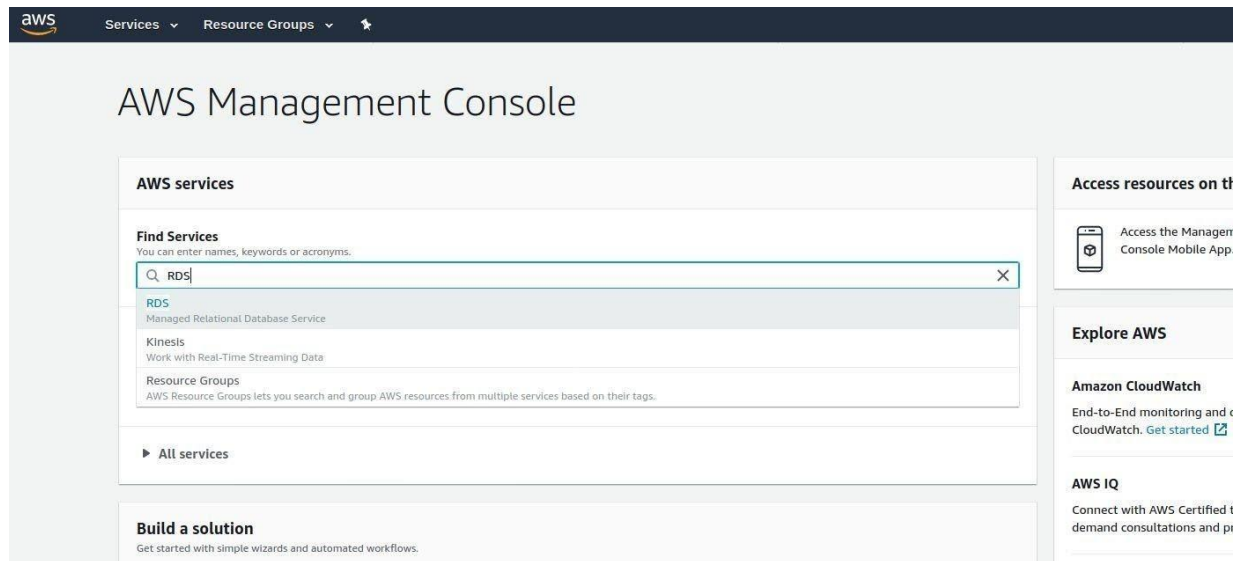
Fast : It is responsible for supporting very demanding DB applications. You have the freedom to choose either of the SSD-backed storage alternatives. They include one that is meant for high-acting OLTP apps while the other is for cheaper general use purposes

## **Conclusion :**

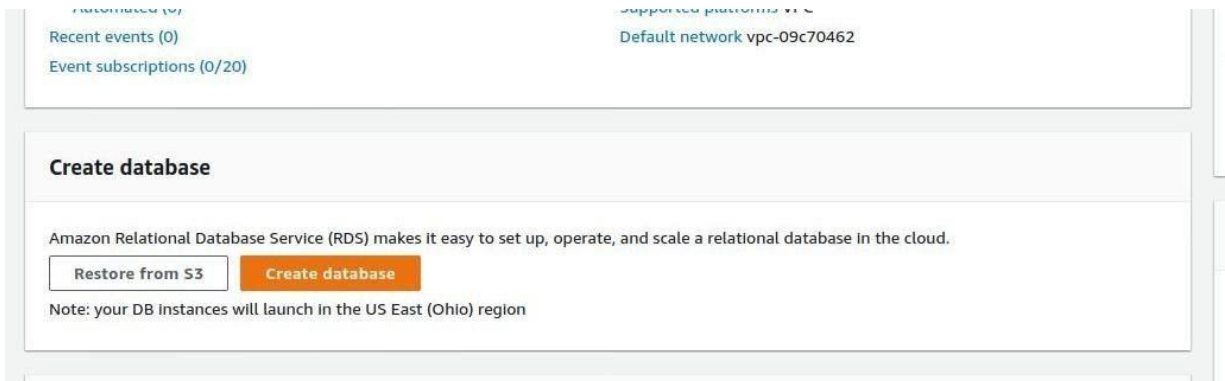
### **Postlabs :**

1. What are the features of Amazon Database?
2. Which of the AWS DB services is a NoSQL database and server-less, and delivers consistent single-digit millisecond latency at any scale?
  - a. Amazon Aurora
  - b. MariaDB
  - c. DynamoDB
  - d. Amazon Redshift
3. What is DynamoDB?
4. How is Security implemented in Amazon RDS?

## Step 1: Open RDS in your AWS Console



## Step 2: Select Create Database



Step 3: Select Standard Create -> Engine Type MySQL ( if you SQL Workbench installed )  
or  
PostgreSQL ( if you pgadmin installed )

RDS > Create database

## Create database


**Choose a database creation method** [Info](#)


☒ **Standard Create**  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.


☐ **Easy Create**  
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.


**Engine options**


Engine type [Info](#)


☐ Amazon Aurora  


☒ MySQL  


☐ MariaDB  


☐ PostgreSQL  


☐ Oracle  


☐ Microsoft SQL Server  


Edition

☒ MySQL Community

Version [Info](#)

MySQL 5.7.22 ▼

Step 4: Select Free Tier ( important ) , name you db and add username , password.

## Templates

Choose a sample template to meet your use case.



### Production

Use defaults for high availability and fast, consistent performance.



### Dev/Test

This instance is intended for development use outside of a production environment.



### Free tier

Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. [Info](#)

## Settings

### DB instance identifier [Info](#)

Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

### ▼ Credentials Settings

#### Master username [Info](#)

Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter



#### Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password

#### Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), "(double quote) and @ (at sign).

#### Confirm password [Info](#)

**Step 5: Select Default VPC Group & Publicly accessible as YES**

## Connectivity

### Virtual Private Cloud (VPC) [Info](#)

VPC that defines the virtual networking environment for this DB instance.

Default VPC (vpc-09c70462) ▼

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change the VPC selection.

### ▼ Additional connectivity configuration

#### Subnet group [Info](#)

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

default-vpc-09c70462 ▼

#### Publicly accessible [Info](#)

☒ Yes

Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

☐ No

RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

#### VPC security group

Choose one or more RDS security groups to allow access to your database. Ensure that the security group rules allow incoming traffic from EC2 instances and devices outside your VPC. (Security groups are required for publicly accessible databases.)

☒ Choose existing  
Choose existing VPC security groups

☐ Create new  
Create new VPC security group

##### Existing VPC security groups

Choose VPC security groups ▼

default ✕

#### Availability zone [Info](#)

No preference ▼

#### Database port [Info](#)

TCP/IP port the database will use for application connections.

3306 ▲ ▼

Cloud Computing Lab Manual By Prof. Sunil Chaudhari

## Step 6: Create Database

### Database authentication

Database authentication options [Info](#)

☒ Password authentication  
Authenticates using database passwords.

☐ Password and IAM database authentication  
Authenticates using the database password and user credentials through AWS IAM users and roles.

### ► Additional configuration

Database options, backup enabled, backtrack disabled, Enhanced Monitoring disabled, maintenance, CloudWatch Logs, delete protection disabled

### Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier.](#) [↗](#)

When you free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page.](#) [↗](#)

Cancel Create database

Step 7: Wait until your database is available (this will take sometime) , then click on your database

Databases										
<div><div>Group resources</div><div>Modify</div><div>Actions</div><div>Restore from S3</div><div>Create database</div></div>										
<div>Filter databases</div>										
DB Identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maintenance	VPC	Mu
testdemodb	Instance	PostgreSQL	us-east-2c	db.t2.micro	Deleting	4.00%	0 Connections	none	vpc-09c70462	No
testsqldb	Instance	MySQL Community	us-east-2b	db.t2.micro	Available	-	0 Connections	none	vpc-09c70462	No

**Step 8: Copy the Endpoint given in Connectivity & Security**

The screenshot displays the AWS RDS console for an instance named 'testsqladb'. The 'Connectivity & security' tab is selected, showing details for the endpoint, networking, and security.

Summary			
DB Identifier testsqladb	CPU -	Info Available	Class db.t2.micro
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ us-east-2b

Navigation tabs: Connectivity & security (selected), Monitoring, Logs & events, Configuration, Maintenance & backups, Tags.

**Connectivity & security**

Endpoint & port	Networking	Security
Endpoint testsqladb.crzf2jqpqelj.us-east-2.rds.amazonaws.com	Availability zone us-east-2b	VPC security groups default (sg-74922113) (active)
Port 3306	VPC vpc-09c70462	Public accessibility Yes
	Subnet group default-vpc-09c70462	Certificate authority rds-ca-2019
	Subnets subnet-7fc62d14 subnet-0a28a146 subnet-3defce47	Certificate authority date Aug 22nd, 2024

**Step 9: Open MySQL workbench & test your connection after entering credentials**

The screenshot shows the MySQL Workbench application with the 'Setup New Connection' dialog box open. The dialog is configured with the following details:

- Connection Name: testsqladb
- Connection Method: Standard (TCP/IP)
- Parameters tab selected
- Hostname: testsqladb.crzf2jqpqelj.us-east-2.rds.amazonaws.com
- Port: 3306
- Username: admin
- Password: (masked with 'Store in Keychain ...' button)
- Default Schema: (empty)

Buttons at the bottom: Configure Server Management..., Test Connection, Cancel, OK.



If you fail to establish connection, check if security group inbound rules have 0.0.0.0/0 as RULE

Security group rules (2)			
<input type="text" value="Filter security group rules"/>			< 1 > ⚙
Security group	Type	Rule	
default (sg-74922113)	CIDR/IP - Inbound	0.0.0.0/0	
default (sg-74922113)	CIDR/IP - Outbound	0.0.0.0/0	

If you don't have 0.0.0.0/0 here, you add it by doing:

EC2 > Security Groups > sg-74922113 - default > Edit inbound rules

### Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

**Inbound rules** [Info](#)

**Type** [Info](#)

**Protocol** [Info](#)

**Port range** [Info](#)

**Source** [Info](#)

**Description - optional** [Info](#)

All traffic

All

All

Anywhere

Delete

Add rule

**NOTE:** Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel

Preview changes

Save rules

NOTE: 0.0.0.0 means it's accessible to everyone with hostname , username and password. This is not a good practice for real world application. It should only be accessible from your IP. Hence in the source, select "My IP" option, which will automatically detect and add you IP

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

**Inbound rules** [Info](#)

**Type** [Info](#)

**Protocol** [Info](#)

**Port range** [Info](#)

**Source** [Info](#)

**Description - optional** [Info](#)

All traffic

All

All

My IP

Delete

Add rule

**NOTE:** Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Step 10: Open the EC2 terminal

Enter following Commands to setup MySQL connection

a) `sudo yum install mysql`

b) `mysql -h <<mysql instance dns>> -P 3306 -u cclab -p`

```
[ec2-user@ip-172-31-0-162 ~]$ mysql -h cclab.citpmfvmybxh.us-east-2.rds.amazonaws.com -P 3306 -u cclab -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.20 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> █
```

c) Commands to create a Database

1) create database BookDB

```
MySQL [(none)]> create database bookDB;  
Query OK, 1 row affected (0.00 sec)
```

```
MySQL [(none)]> show databases;
```

```
+-----+  
| Database |  
+-----+  
| BookDB   |  
| BooksDB  |  
| book     |  
| bookDB   |  
| information_schema |  
| mysql    |  
| performance_schema |  
+-----+  
7 rows in set (0.00 sec)
```

- 2) create table if not exists books(  
book\_id INT AUTO\_INCREMENT,  
title VARCHAR(255) NOT NULL,  
author VARCHAR(255),  
publish\_date DATE,  
description TEXT,  
PRIMARY KEY (book\_id)  
) ENGINE=INNODB;

```
MySQL [bookDB]> create table if not exists books(  
-> book_id INT AUTO_INCREMENT,  
-> title VARCHAR(255) NOT NULL,  
-> author VARCHAR(255),  
-> publish_date DATE,  
-> description TEXT,  
-> PRIMARY KEY (book_id)  
-> ) ENGINE=INNODB;  
Query OK, 0 rows affected (0.03 sec)
```

- 4) INSERT INTO books(title,author) VALUES ('Learn AWS RDS','Shubham Mishra');  
INSERT INTO books(title,author) VALUES ('MySQL Tips and Tricks','Advait Poojary');

S

```
MySQL [bookDB]> INSERT INTO books(title,author) VALUES
-> ('Learn AWS RDS','Shubham Mishra');
Query OK, 1 row affected (0.01 sec)

MySQL [bookDB]>
MySQL [bookDB]>
MySQL [bookDB]>
MySQL [bookDB]> INSERT INTO books(title,author) VALUES ('Learn AWS RDS','Advaith Poojary');
Query OK, 1 row affected (0.00 sec)
```

## Q1) What are the features of Amazon Database?

### Lower administrative burden

- Easy to use
- You can use the [AWS Management Console](#), the [Amazon RDS Command Line Interface](#), or simple [API calls](#) to access the capabilities of a production-ready relational database in minutes.
- Amazon RDS database instances are pre-configured with parameters and settings appropriate
- Automatic software patching
- Amazon RDS will make sure that the relational database software powering your deployment stays up-to-date with the latest patches. You can exert optional control over when and if your database instance is patched.

### Performance

- General Purpose (SSD) Storage
- Amazon RDS General Purpose Storage is an SSD-backed storage option delivers a consistent baseline of 3 IOPS per provisioned GB and provides the ability to burst up to 3,000 IOPS above the baseline. This storage type is suitable for a broad range of database workloads.

### Scalability

- Push-button compute scaling
- You can scale the compute and memory resources powering your deployment up or down, up to a maximum of 32 vCPUs and 244 GiB of RAM. Compute scaling operations typically complete in a few minutes.
- Easy storage scaling
- As your storage requirements grow, you can also provision additional storage. The Amazon Aurora engine will automatically grow the size of your database volume as your database storage needs grow, up to a maximum of 64 TB or a maximum you define. The MySQL, MariaDB, Oracle, and PostgreSQL engines allow you to scale up to 64 TB of storage and SQL Server supports up to 16 TB. Storage scaling is on-the-fly with zero downtime.

### Availability and durability

- Automated backups
- Database snapshots
- Multi-AZ deployments

- Automatic host replacement
- Amazon RDS will automatically replace the compute instance powering your deployment in the event of a hardware failure.

### Security

- Encryption at rest and in transit
- Amazon RDS supports Transparent Data Encryption in SQL Server and Oracle. Transparent Data Encryption in Oracle is integrated with [AWS CloudHSM](#), which allows you to securely generate, store, and manage your cryptographic keys in single-tenant Hardware Security Module (HSM) appliances within the AWS cloud.
- Amazon RDS supports the use of [SSL to secure data in transit](#).
- Network isolation
- Resource-level permissions

### Manageability

- Monitoring and metrics
- Event notifications
- Configuration governance

.

### Cost-effectiveness

- Pay only for what you use
- Reserved instances
- Stop and start
- Amazon RDS allows you to easily stop and start your database instances for up to 7 days at a time. This makes it easy and affordable to use databases for development and test purposes, where the database is not required to be running all of the time.

Q2) . Which of the AWS DB services is a NoSQL database and server-less, and delivers consistent single-digit millisecond latency at any scale?

- a. Amazon Aurora b. MariaDB c. DynamoDB d. Amazon Redshift

Ans. C) Dynamo DB

Q3) What is DynamoDB?

Ans. Amazon DynamoDB 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. For more information, see [DynamoDB Encryption at Rest](#).

With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. You can use the AWS Management Console to monitor resource utilization and performance metrics.

DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs. For more information, see [On-Demand Backup and Restore for DynamoDB](#).

Q4) How is Security implemented in Amazon RDS?

Ans . Amazon RDS is a managed relational database service that provides you six familiar database engines to choose from, including [Amazon Aurora](#), [MySQL](#), [MariaDB](#), [Oracle](#), [Microsoft SQL Server](#), and [PostgreSQL](#).

Amazon RDS and Amazon Aurora provide a set of features to ensure that your data is securely stored and accessed. Run your database in Amazon Virtual Private Cloud (VPC) for network-level isolation. Use security groups to control what IP addresses or Amazon EC2 instances can connect to your databases. This built-in firewall prevents any database access except through rules you specify.

Use AWS Identity and Access Management (IAM) policies to assign permissions that determine who is allowed to manage RDS resources. Use the security features of your database engine to control who can log in to the databases, just as you do if the database was on your local network. You can also map database users to IAM roles for federated access.

Use Secure Socket Layer / Transport Layer Security (SSL/TLS) connections to encrypt data in transit. Encrypt your database storage and backups at rest using

Amazon Key Management Service (KMS). Monitor database activity and integrate with partner database security applications with Database Activity Streams.