

Experiment No : 8

Aim: To study and implement Database as a service on SQL database using AWS RDS

Theory:

Explain the concept of Database as a Service.
List some Database as a service provider along with name of service.

- ⇒ Database as a service (DBaaS) is a cloud computing service model that provides users with access to managed database system.
- Instead of setting up and maintaining their own database infrastructure, users can leverage a cloud provider's resources to store, manage and access their data.
- Some well-known DBaaS providers include:
 - 1) Amazon Web Services (AWS)
 - ⇒ Amazon RDS, Amazon Aurora
 - 2) Microsoft Azure
 - ⇒ Azure SQL database, Cosmos DB

3) Google Cloud Platform

⇒ cloud SQL, firestore

Cloud storage, cloud database, Bigtable

Firebase, real-time database, Cloud Functions

Cloud storage, Cloud functions

8.1 AWS Relational Database

- 1) List the most popular database supported by AWS and its architecture.
- AWS supports various SQL and NoSQL databases, including:
 - 1) SQL database: MySQL, PostgreSQL, SQL Server, MariaDB, Oracle, SQLite, Aurora MySQL.
 - 2) NoSQL database: Amazon DynamoDB, Amazon Kinesis, Amazon ElastiCache, Amazon RDS, Amazon DocumentDB.
 - 3) Difference between RDS and Aurora:

Amazon RDS	Aurora
1) Amazon RDS is one of the managed relational database services that supports multiple database engines such as MySQL, PostgreSQL, MariaDB, Oracle and SQL Server.	1) Aurora is a MySQL and PostgreSQL compatible relational database engine built for the cloud.
2) Aurora replicates data six ways across three availability zones by default, providing greater durability and redundancy.	2) Aurora's storage system is highly available with multiple instances; allowing for faster replication and failover compared to RDS where each instance has its own storage.

* Explain the following terms with AWS DBaaS.

DBaaS is a fully managed service that provides a secure and reliable database solution for your applications.

⇒ Database solutions are provided by Amazon RDS.

1) Storage types:

⇒ AWS offers various storage types for databases, including General Purpose SSD (GP2), Provisioned IOPS SSD (T1) (Magnetic standard), and Amazon Aurora storage.

These options differ in terms of performance,

durability, and cost.

2) Endpoint

⇒ In AWS DBaaS, an endpoint refers to the network address used to access a database instance.

Endpoints are used by applications to connect to the database stored either over the internet or within the AWS network.

3) Snapshot

⇒ A snapshot is a point-in-time copy of a database instance. It captures the entire state of a database at the moment the snapshot is taken.

Snapshots are used for backup, recovery, and replication purpose.

- 4) Read Replicas → primary with replica
- A read replica is a copy of a source database instance that allows read-only access to the data.
- Read replicas can be used to offload read operations from the primary database, improving performance and scalability.
- 5) Single AZ and Multi AZ instances
- In AWS, a single Availability Zone instance runs in a single data center within a specific AWS region.
- In contrast, a multi-AZ Availability Zone instance replicates data synchronously across multiple availability zones within the same region, providing high availability and fault tolerance in case of a failure in one AZ instance.

A*

Sep 2024

18/3/2024

In your writing, if you write with a red pen, it is considered as independent work, quoted in your own handwriting → original authorship by you

Output:

Amazon RDS Dashboard

The screenshot shows the Amazon RDS dashboard in the EU North (Stockholm) region. It displays various resources and metrics. A prominent blue banner at the top introduces the Aurora I/O-Optimized storage configuration.

Create Database

This section allows users to create a new database. It includes a "Create database" button and a note that DB instances will launch in the EU North (Stockholm) region.

Service health

The service is currently operating normally.

Additional Information

Links to Getting started with RDS, Overview and features, Documentation, Articles and tutorials, Data import guide for MySQL, Data import guide for Oracle, Data import guide for SQL Server, New RDS feature announcements, Pricing, and Forums.

Database Preview Environment

Get early access to new DB engine versions.

MySQL Configuration

The MySQL configuration page provides details about MySQL support and features, including support for up to 64 TiB, General Purpose, Memory Optimized, and Burstable Performance instance classes, automated backup, point-in-time recovery, and up to 15 Read Replicas per instance.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL Community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Templates

Choose a sample template to meet your use case.

Production Use defaults for high availability

Dev/Test This instance is intended for

Free tier Use RDS Free Tier to develop new

Availability and durability

Deployment options [Info](#)

The deployment options below are limited to those supported by the engine you selected above.

Multi-AZ DB Cluster
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.

Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.

Single DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a single DB instance with no standby DB instances.

Settings

DB instance identifier [Info](#)

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

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. 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.

admin

Master password

Specify a string that defines the password for the master user. Master Password must be at least eight characters long, as in "mypassword".

DB instance identifier [Info](#)

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

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. 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.

admin

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management

You can use AWS Secrets Manager or manage your master user credentials.

Managed in AWS Secrets Manager - most secure
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

Self managed
Create your own password or have RDS create a password that you manage.

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

Master password [Info](#)

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / * @

Confirm master password [Info](#)

Storage

Allocated storage GiB

Master password

Specify a string that defines the password for the master user. Master Password must be at least eight characters long, as in "mypassword".

Virtual private cloud (VPC)

DB subnet group

Public access

VPC security group (firewall)

RDS Proxy

Certificate authority - optional

Screenshot 1: AWS RDS Free Tier Configuration

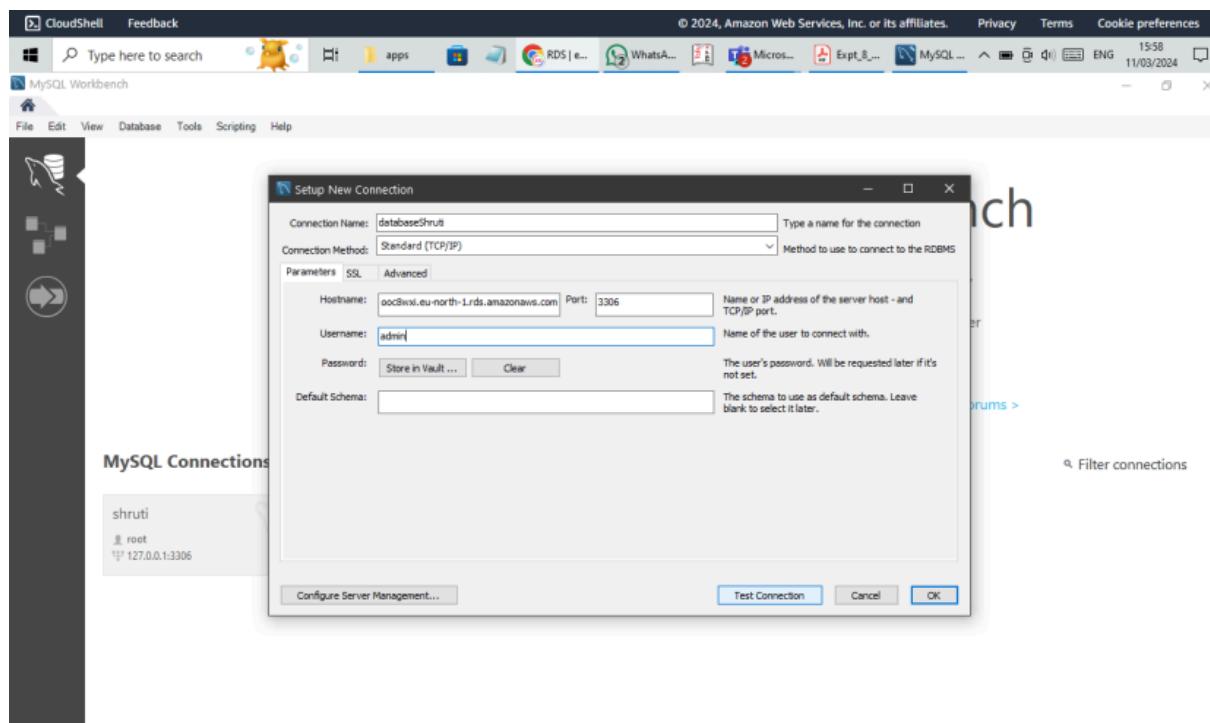
The screenshot shows the AWS RDS Free Tier configuration page. It includes sections for 'Enable Enhanced Monitoring', 'Additional configuration' (with notes on encryption, backup, and maintenance), 'Estimated monthly costs' (noting the free tier for 12 months), and a note about responsibilities. At the bottom is a 'Create database' button.

Screenshot 2: Creating a Database

This screenshot shows the 'Creating database database-1' step. It displays a progress bar and a note about Aurora I/O-Optimized and Blue/Green deployments. The 'Databases' table shows 'database-1' in the 'Creating' state.

Screenshot 3: Database Overview

This screenshot shows the 'database-1' overview page. It displays basic information such as DB identifier, status, role, engine, and connectivity details like endpoint and port. Tabs for connectivity, monitoring, logs, configuration, maintenance, tags, and recommendations are present.



VPC dashboard EC2 Global View

Filter by VPC: Select a VPC

Your VPCs (1) Info

VPC ID : **vpc-001671399ec9d2700** X Clear filters

Name	VPC ID	State	IPv4 CIDR
-	vpc-001671399ec9d2700	Available	172.31.0.0/16

Select a VPC above

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

EC2 > Security Groups > sg-01031f3c06b9fd512 - default > Edit inbound rules

Edit inbound rules Info

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

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0eb19093ec456da26	All traffic	All	All	C... X	sg-01031f3c06b9fd512

Add rule

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

EC2 Dashboard X Inbound security group rules successfully modified on security group (sg-01031f3c06b9fd512 | default) Details

EC2 Global View

Events

Instances

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations
- New

Images

AMIs

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

MySQL Workbench

File Edit View Database Tools Scripting Help

Manage Server Connections

MySQL Connections

- shruti
- databaseShruti

Connection Name: databaseShruti

Connection Method: Standard (TCP/IP)

Parameters SSL Advanced

Hostname: databaseshruti.cn04aoc8exi.eu-north-1.rds.amazonaws.com Port: 3306

Username: admin

Password: Store in Vault ... Clear

Default Schema:

New Delete Duplicate Move Up Move Down Test Connection Close

MySQL Connect

shruti

root

127.0.0.1:3306

MySQL Workbench

MySQL Connections

- shruti
- databaseShruti

Connection Name: databaseShruti

Connection Method: Standard (TCP/IP)

MySQL Workbench

MySQL Workbench

Successfully made the MySQL connection

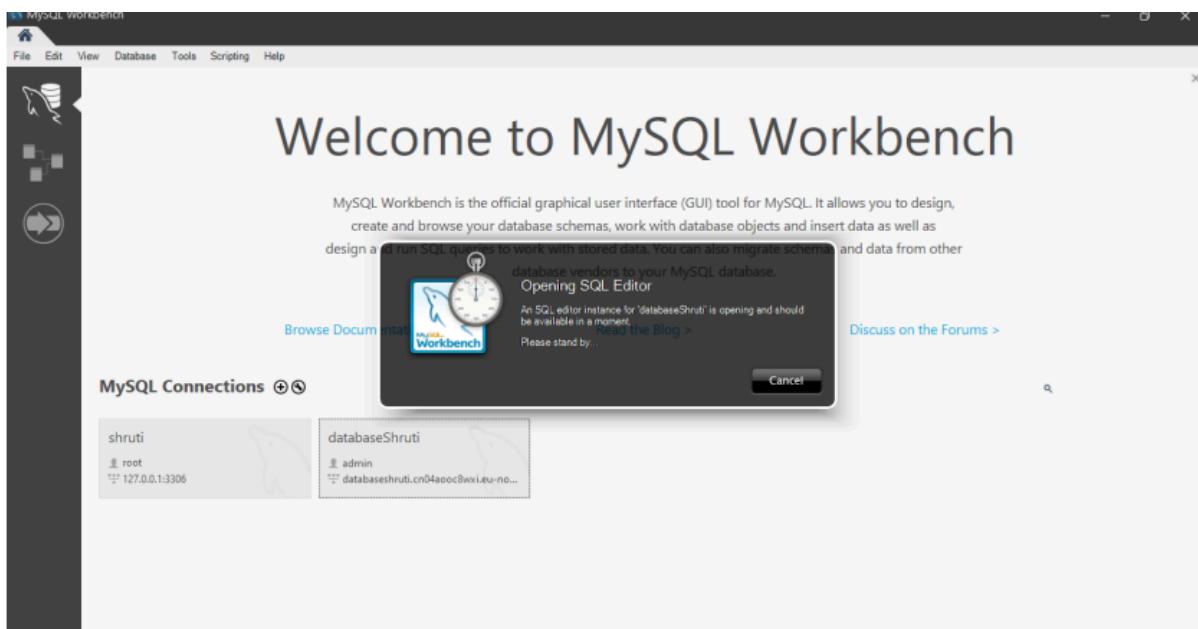
Information related to this connection:

Host: databaseshruti.cn04aoc8exi.eu-north-1.rds.amazonaws.com
Port: 3306
User: admin
SSL enabled with TLS_AES_128_GCM_SHA256

A successful MySQL connection was made with the parameters defined for this connection.

OK

New Delete Duplicate Move Up Move Down Test Connection Close



This screenshot shows the MySQL Workbench interface after connecting to the "databaseShruti" database. The title bar now says "databaseShruti". The left sidebar includes sections for MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), and PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup). The main area has tabs for Administration and Schemas, with "Administration" selected. In the center, the "Query 1" tab shows two SQL statements: "create database shruti;" and "use shruti;". The "Output" pane at the bottom displays the results of these queries, showing "Action Output" with two entries: "1 16:18:29 create database shruti" and "2 16:18:31 use path". The "Message" column indicates "1 row(s) affected" for the first query and "Error Code: 1049: Unknown database 'path'" for the second. The "Duration / Fetch" column shows "0.210 sec" and "0.219 sec" respectively. A note in the top right of the main area says: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or toggle automatic help."

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:


```

1 •  create database shruti;
2 •  use shruti;
3
4 •  CREATE TABLE template_table (
5      id INT AUTO_INCREMENT PRIMARY KEY,
6      name VARCHAR(255),
7      age INT,
8      email VARCHAR(255)
9  );
10
11 •  INSERT INTO template_table (name, age, email) VALUES ('John Doe', 30, 'john@example.com');
12 •  INSERT INTO template_table (name, age, email) VALUES ('Jane Smith', 25, 'jane@example.com');
13 •  INSERT INTO template_table (name, age, email) VALUES ('Alice Johnson', 35, 'alice@example.com');
14
            
```
- Output Window:** Shows the execution log with the following entries:

Action	Time	Message	Duration / Fetch
use shruti	3 16:19:44	0 row(s) affected	0.234 sec
CREATE TABLE template_table (id INT AUTO_INCREMENT PRIMARY KEY, name VA...	4 16:19:05	0 row(s) affected	0.234 sec
INSERT INTO template_table (name, age, email) VALUES ('John Doe', 30, 'john@example.co...	5 16:19:09	1 row(s) affected	0.234 sec
INSERT INTO template_table (name, age, email) VALUES ('Jane Smith', 25, 'jane@example.co...	6 16:19:11	1 row(s) affected	0.235 sec
INSERT INTO template_table (name, age, email) VALUES ('Alice Johnson', 35, 'alice@example.com')	7 16:19:15	1 row(s) affected	0.218 sec

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:


```

9 );
10
11 •  INSERT INTO template_table (name, age, email) VALUES ('John Doe', 30, 'john@example.com');
12 •  INSERT INTO template_table (name, age, email) VALUES ('Jane Smith', 25, 'jane@example.com');
13 •  INSERT INTO template_table (name, age, email) VALUES ('Alice Johnson', 35, 'alice@example.com');
14
15 •  select * from template_table;
            
```
- Result Grid:** Displays the data from the template_table:

	id	name	age	email
1	1	John Doe	30	john@example.com
2	2	Jane Smith	25	jane@example.com
3	3	Alice Johnson	35	alice@example.com
- Output Window:** Shows the execution log with the following entries:

Action	Time	Message	Duration / Fetch
INSERT INTO template_table (name, age, email) VALUES ('Alice Johnson', 35, 'alice@example.com')	7 16:19:15	1 row(s) affected	0.218 sec
SELECT * FROM template_table	8 16:19:18	Error Code: 1054. You have an error in your SQL syntax; check the manual that corresponds ...	0.235 sec
SELECT * FROM template_table	9 16:19:31	Error Code: 1054. You have an error in your SQL syntax; check the manual that corresponds ...	0.219 sec
select * from template_table LIMIT 0, 400	10 16:20:24	3 row(s) returned	0.219 sec / 0.000 sec

eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#databases:

You are stopping this DB instance for up to 7 days. You can restart the DB instance manually at any time. To stop the DB instance permanently, save it in a snapshot and delete it. [Learn more](#)

Acknowledgement
 I acknowledge that the DB instance will restart automatically after 7 days, on March 18, 2024, 16:21 (UTC+05:30).

Snapshot - optional
 Save the DB instance in a snapshot
The snapshot enables you to restore the DB instance to its last state before it was stopped.

Cancel **Stop temporarily**

Permanently delete database-1 DB instance. You can't undo this action.

⚠ Proceeding with this action will delete the instance with all its content and can affect related resources. [Learn more](#)

Create final snapshot
Determines whether a final DB Snapshot is created before the DB instance is deleted.

Final snapshot name
The identifier of the new DB snapshot that is created.

Retain automated backups
Determines whether retaining automated backups for 1 day after deletion

ⓘ You will be billed for retained backup storage at the rate described as 'Additional backup storage' found in Backup Storage. [Learn more](#)

To avoid accidental deletion provide additional written consent.
To confirm deletion, type *delete me* into the field.

Cancel **Delete**

Deleting DB instance database-1

Databases (1)

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations	CPU
database-1	Deleting	Instance	MySQL Community	eu-north-1b	db.t3.micro		

Group resources Modify Actions Restore from S3