Academic Year: 2024-25 **Semester: VI**
Class/Branch: T.E.D.S. **Subject: Cloud Computing Lab**

EXPERIMENT NO. 7

Aim: To implement Database as a Service on SQL/NOSQL database like AWS RDS

1. **Objectives:** From this experiment, the student will be able to,
   - To implement a Database as a Service (DBaaS) solution using SQL or NoSQL databases such as AWS RDS (Relational Database Service) on the Amazon Web Services (AWS) platform.
   - To Launch an Amazon RDS DB instance with high availability.
   - To Configure the DB instance to permit connections from your web server.
   - Open a web application and interact with your database
   - To access and utilize databases without the need for managing the underlying hardware, software, or infrastructure
2. **Outcomes:** The learner will be able,
   - Implement a scalable infrastructure that can automatically adjust to changing workloads and resource demands.
   - Able to Launch and configure DB instance
   - Able to open, access and utilize databases
   - Ensure the database service is highly available and reliable by utilizing features such as automatic backups, multi-AZ deployment, and failover mechanisms.
3. **Hardware / Software Required:** Ubuntu operating system, An active AWS account. AWS Management Console access for configuration.Utilize AWS RDS for managing relational databases such as MySQL, PostgreSQL, SQL Server, etc., or NoSQL databases like Amazon DynamoDB.
4. **Theory:**

Database-as-a-service (DBaaS) is a cloud computing service. As a hosted/managed service, users don't have to worry about setting up hardware or installing software. Everything related to managing the database is handled by the service provider.

Cloud database management is often much simpler than traditional on-premises equivalents. The database administration tools themselves are almost identical, allowing you to provision databases quickly and easily on the hosted infrastructure.

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 managing time-consuming database administration tasks, which allows you to focus on your applications and business. Amazon RDS provides you with six familiar database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB

5. **Procedure:**

Task 1: Create a Security Group for the RDS DB Instance

- Create Security Groups and Add rules to it
- Configures the Database security group to permit inbound traffic on port 3306 from any EC2 instance that is associated with the *Web Security Group*.

Task 2: Create DB subnet Group to tell RDS which subnets can be used for the database. Each DB subnet group requires subnets in at least two Availability Zones.

- From Services select RDS, choose subnets
- Create Subnets and configure

Task 3: Create an amazon RDS DB Instance

- choose Create Database, choose MySql
- Choose template as Dev/Test
- Choose Multi-AZ DB instance under Availability
- Configure the BD instance
- Under Storage configure SSD with allocation as 20
- Choose and configure the security groups
- Configure database name and create the database
- Wait until Info changes to Modifying or Available.
- Scroll down to the Connectivity & security section and copy the Endpoint field.It will look similar to: *lab-db.xxxx.us-east-1.rds.amazonaws.com*.

Task 4: Interact with Your Database

- Choose on the AWS Details drop down menu above these instructions. Copy the IP address value.
- Open a new web browser tab, paste the *WebServer* IP address and press Enter. The web application will be displayed, showing information about the EC2 instance.
- Choose the RDS link at the top of the page. and configure the application to connect to your database.
- Configure the database for endpoints and database name
- After some time the address book will be displayed

Conclusion: