Cloud Computing Lab

Experiment No.: 5

Create and migrate relational database on cloud (Amazon RDS)

Experiment No. 5

Name : Sarvagya Singh

SAPID : 60009200030 **BATCH : K1**

1. **Aim:** Create and migrate relational database on cloud
2. **Objectives:**
   - Launch a database using Amazon RDS
   - Configure a web application to connect to the database instance

3. **Outcomes:**

The learner will be able to launch, relational database

4. **Hardware / Software Required:** Internet, AWS console

5. **Theory:**

Amazon Relational Database Service (RDS) is a managed SQL database service provided by Amazon Web Services (AWS). Amazon RDS supports an array of database engines to store and organize data. It also helps with relational database management tasks, such as data migration, backup, recovery and patching.

Amazon RDS facilitates the deployment and maintenance of relational databases in the cloud. A cloud administrator uses Amazon RDS to set up, operate, manage and scale a relational instance of a cloud database. Amazon RDS is not itself a database; it is a service used to manage relational databases.

Task 1: Creating an Amazon RDS database

In this task, you create a MySQL database in your virtual private cloud (VPC). MySQL is a popular open-source relational database management system (RDBMS), so there are no software licensing fees.

1. On the **Services** menu, choose **RDS**.
2. Choose **Create database**
3. Under **Engine options**, select **MySQL**.
4. In the **Templates** section, select **Dev/Test**
5. In the **Settings** section, configure the following options:

DB instance identifier: inventory-db

Master username: admin

lab-password

lab-password

Master password:

Confirm password:

6. In the **DB instance class** section, configure the following options:
   Select **Burstable classes (includes t classes)**.
   Select **db.t3.micro**.

7. In the **Storage** section, for **Storage type**, select **General Purpose SSD (gp2)**.
8. In the **Connectivity** section, configure the following option:
   Virtual private cloud (VPC): Lab VPC

9. In the **Connectivity** section, for **Existing VPC security groups**, choose the **X** on default

*to remove this security group. Then choose the dropdown list, and select **DB-SG** to add*
it.

10. Scroll to the **Additional configuration** section, and choose to expand it. Configure the following settings:

    For **Initial database name**, enter inventor

    Clear (turn off) the **Enable Enhanced monitoring** option.

    *This is the logical name of the database that the application will use At the bottom of the page, choose **Create database***

You should receive this message: **Creating database inventory-db**.

- *Before you continue to the next task, the database instance status must be*

Available. This process could take several minutes.

Task 2: Configuring web application communication with a database instance

This lab automatically deployed an Amazon Elastic Compute Cloud (Amazon EC2) instance with a running web application. You must use the IP address of the instance to connect to the application.

11. On the **Services** menu, choose **EC2**.
12. *In the left navigation pane, choose **Instances**.*
13. In the center pane, there should be a running instance that is named **App Server**.
14. Select the check box for the **App Server** instance.
15. In the **Details** tab, copy the **Public IPv4 address** to your clipboard.

***Tip:** If you hover over the IP address, a copy icon appears. To copy the displayed value, choose the icon.*

16. Open a new web browser tab, paste the IP address into the address bar, and then press Enter.

The web application should appear. It does not display much information because the application is not yet connected to the database.

17. Choose **Settings**.

You can now configure the application to use the Amazon RDS database instance that you created earlier. You first retrieve the database endpoint so that the application knows how to connect to a database.

18. Return to the AWS Management Console, but do not close the application tab. (You will return to it soon.)
19. On the **Services** menu, choose **RDS**.
20. *In the left navigation pane, choose **Databases**.*
21. Choose inventory-db.
22. Scroll to the Connectivity & security section, and copy the Endpoint to your clipboard.

It should look similar to this example: inventory-db.crwxbgqad61a.rds.amazonaws.com

23. Return to the browser tab with the inventory application, and enter the following values:

For **Endpoint**, paste the endpoint you copied
earlier. For **Database** `nventory`
For **Username**, enter `admin`
For **Password**, enter `lab-password`
Choose **Save**.

The application will now connect to the database, load some initial data, and display information.

24. You can use the web application to Add inventory, edit, and delete inventory information.
25. Insert new records into the table. Ensure that the table has 5 or more inventory records before submitting your work.

You have now successfully launched the application and connected it to the database.

6. Result

Paste your screen shots for every task.

7. Conclusion:

Amazon Relational Database Service (Amazon RDS) is a collection of managed services that makes it simple to set up, operate, and scale databases in the cloud. Choose from seven popular engines — Amazon Aurora with MySQL compatibility, Amazon Aurora with PostgreSQL compatibility, MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server — and deploy on-premises with Amazon RDS on AWS Outposts.

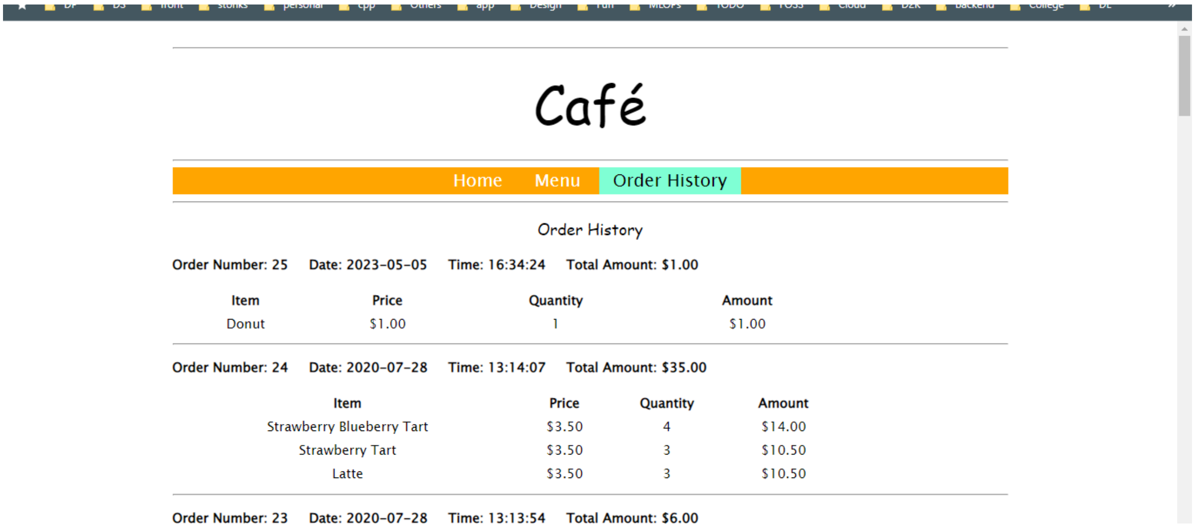
8. Viva Questions:

- What Are the Features of the Amazon Database Service?
- What is an RDS instance?
- List DB Storages supported by Amazon RDS?

9. References:

- https://aws.amazon.com/rds/?p=ft&c=db&z=3
- https://aws.amazon.com/rds/

SS :



Café

Order History

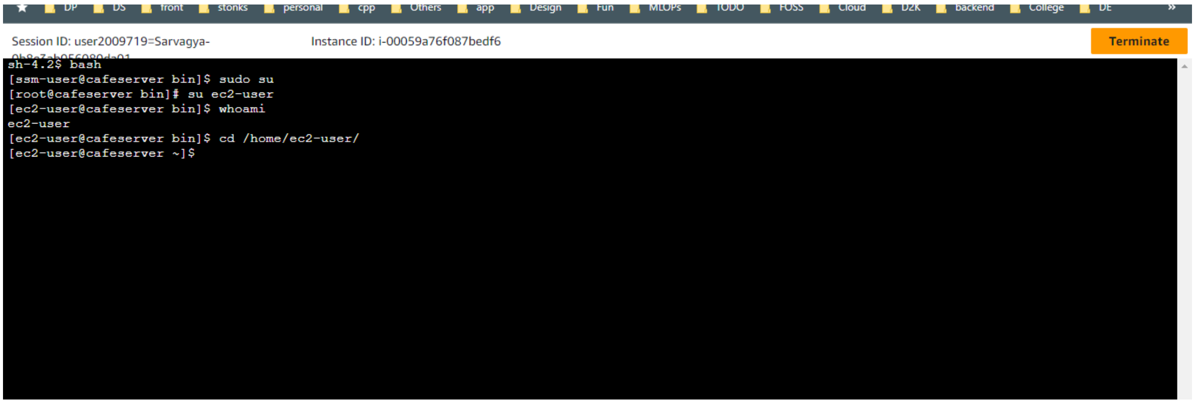
Order Number: 25 **Date: 2023-05-05** **Time: 16:34:24** **Total Amount: $1.00**

Item	Price	Quantity	Amount
Donut	$1.00	1	$1.00

Order Number: 24 **Date: 2020-07-28** **Time: 13:14:07** **Total Amount: $35.00**

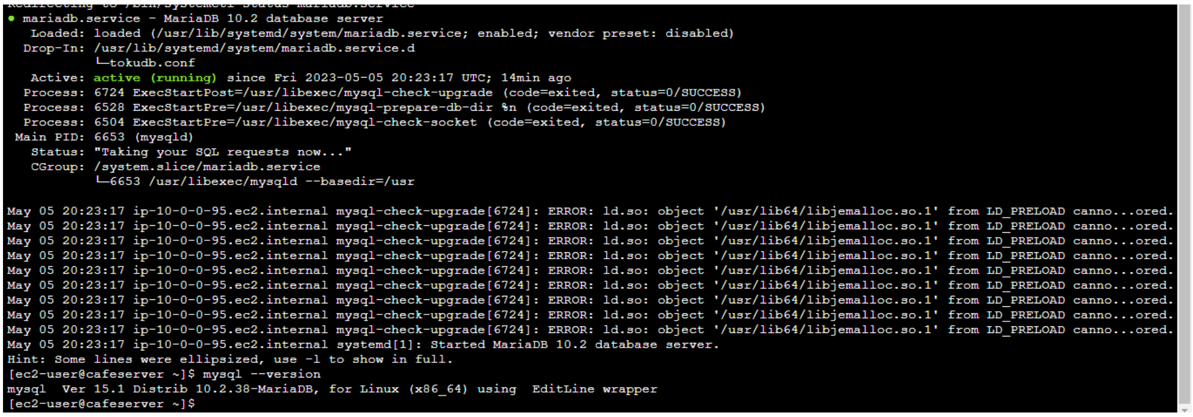
Item	Price	Quantity	Amount
Strawberry Blueberry Tart	$3.50	4	$14.00
Strawberry Tart	$3.50	3	$10.50
Latte	$3.50	3	$10.50

Order Number: 23 **Date: 2020-07-28** **Time: 13:13:54** **Total Amount: $6.00**



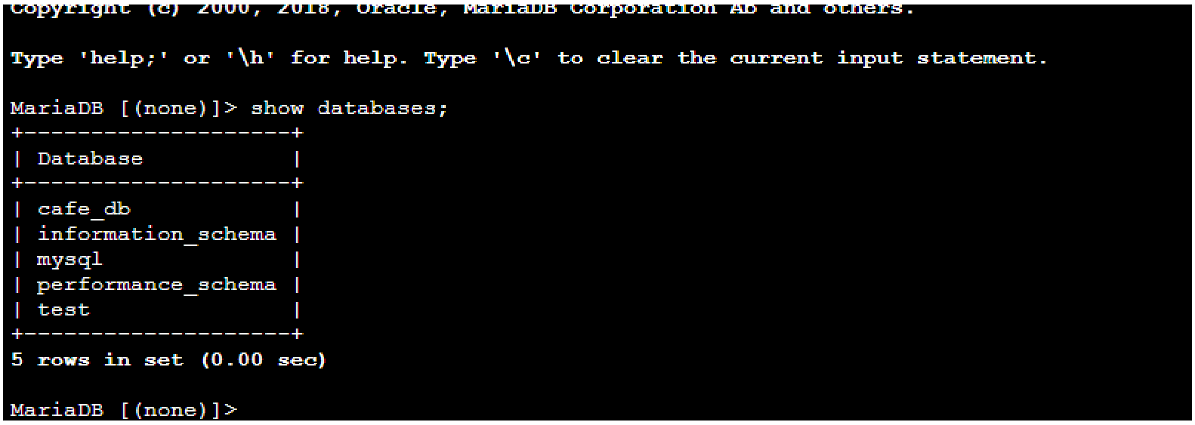
Session ID: user2009719=Sarvagya-0b8c7ab056080da01 Instance ID: i-00059a76f087bedf6

```
sh-4.2$ bash
[ssm-user@cafeserver bin]$ sudo su
[root@cafeserver bin]# su ec2-user
[ec2-user@cafeserver bin]$ whoami
ec2-user
[ec2-user@cafeserver bin]$ cd /home/ec2-user/
[ec2-user@cafeserver ~]$
```



```
Redirecting to /bin/systemctl status mariadb.service
● mariadb.service - MariaDB 10.2 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
  Drop-In: /usr/lib/systemd/system/mariadb.service.d
           └─tokudb.conf
   Active: active (running) since Fri 2023-05-05 20:23:17 UTC; 14min ago
  Process: 6724 ExecStartPost=/usr/libexec/mysql-check-upgrade (code=exited, status=0/SUCCESS)
  Process: 6528 ExecStartPre=/usr/libexec/mysql-prepare-db-dir %n (code=exited, status=0/SUCCESS)
  Process: 6504 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, status=0/SUCCESS)
 Main PID: 6653 (mysqld)
   Status: "Taking your SQL requests now..."
   CGroup: /system.slice/mariadb.service
           └─6653 /usr/libexec/mysqld --basedir=/usr

May 05 20:23:17 ip-10-0-0-95.ec2.internal mysql-check-upgrade[6724]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD canno...ored.
May 05 20:23:17 ip-10-0-0-95.ec2.internal mysql-check-upgrade[6724]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD canno...ored.
May 05 20:23:17 ip-10-0-0-95.ec2.internal mysql-check-upgrade[6724]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD canno...ored.
May 05 20:23:17 ip-10-0-0-95.ec2.internal mysql-check-upgrade[6724]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD canno...ored.
May 05 20:23:17 ip-10-0-0-95.ec2.internal mysql-check-upgrade[6724]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD canno...ored.
May 05 20:23:17 ip-10-0-0-95.ec2.internal mysql-check-upgrade[6724]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD canno...ored.
May 05 20:23:17 ip-10-0-0-95.ec2.internal mysql-check-upgrade[6724]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD canno...ored.
May 05 20:23:17 ip-10-0-0-95.ec2.internal mysql-check-upgrade[6724]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD canno...ored.
May 05 20:23:17 ip-10-0-0-95.ec2.internal mysql-check-upgrade[6724]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD canno...ored.
May 05 20:23:17 ip-10-0-0-95.ec2.internal systemd[1]: Started MariaDB 10.2 database server.
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@cafeserver ~]$ mysql --version
mysql  Ver 15.1 Distrib 10.2.38-MariaDB, for Linux (x86_64) using  EditLine wrapper
[ec2-user@cafeserver ~]$
```

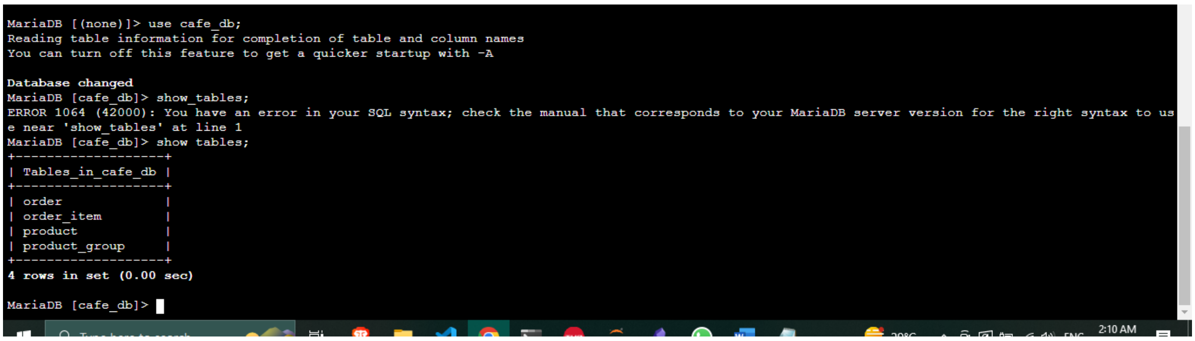


```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| cafe_db            |
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
5 rows in set (0.00 sec)

MariaDB [(none)]>
```



```
MariaDB [(none)]> use cafe_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [cafe_db]> show_tables;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to us
e near 'show_tables' at line 1
MariaDB [cafe_db]> show tables;
+-------------------+
| Tables_in_cafe_db |
+-------------------+
| order             |
| order_item        |
| product           |
| product_group     |
+-------------------+
4 rows in set (0.00 sec)

MariaDB [cafe_db]>
```