Experiment No.
9

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.

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 east 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?

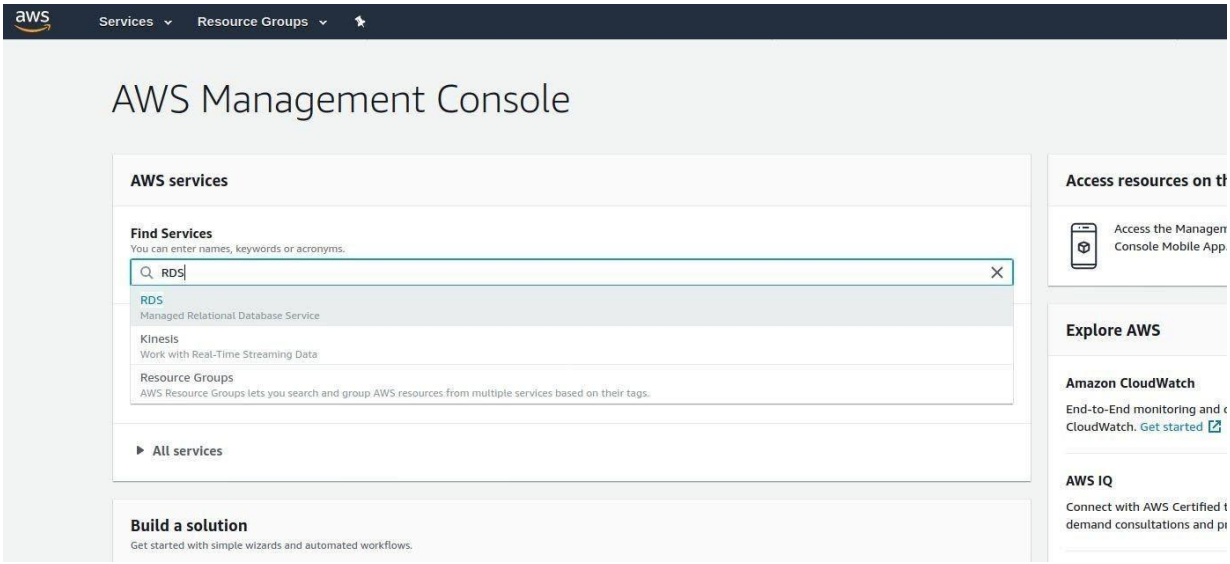
2. Which of the AWS DB services is a NoSQL database and server-less, and delivers consistent single-digit millisecond latency at any scale?

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

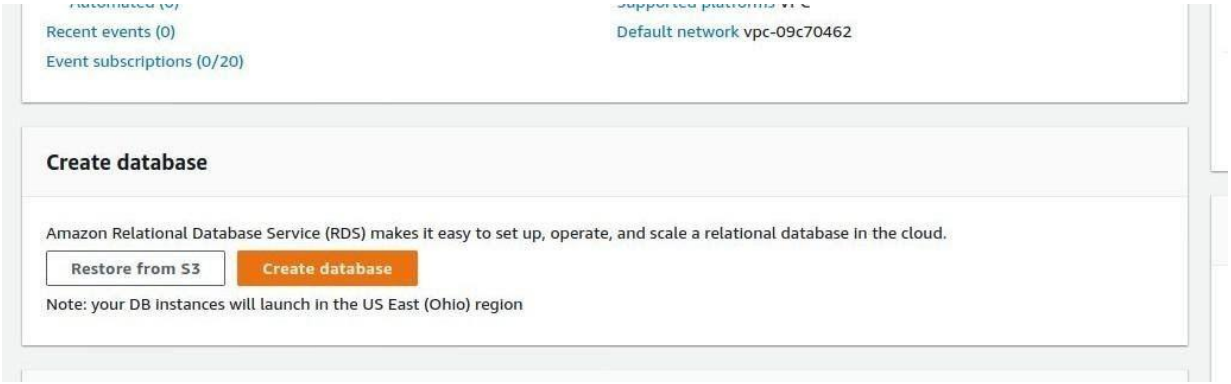
3. What is DynamoDB?

4. How is Security implemented in Amazon RDS?

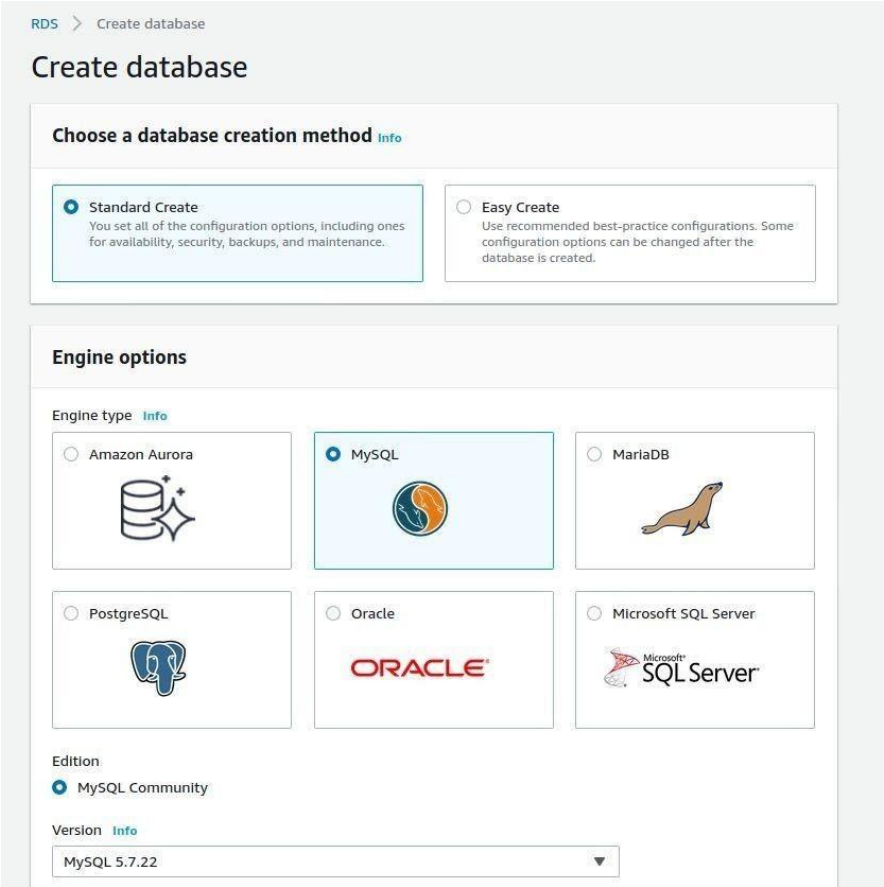
Step 1: Open RDS in your AWS Console



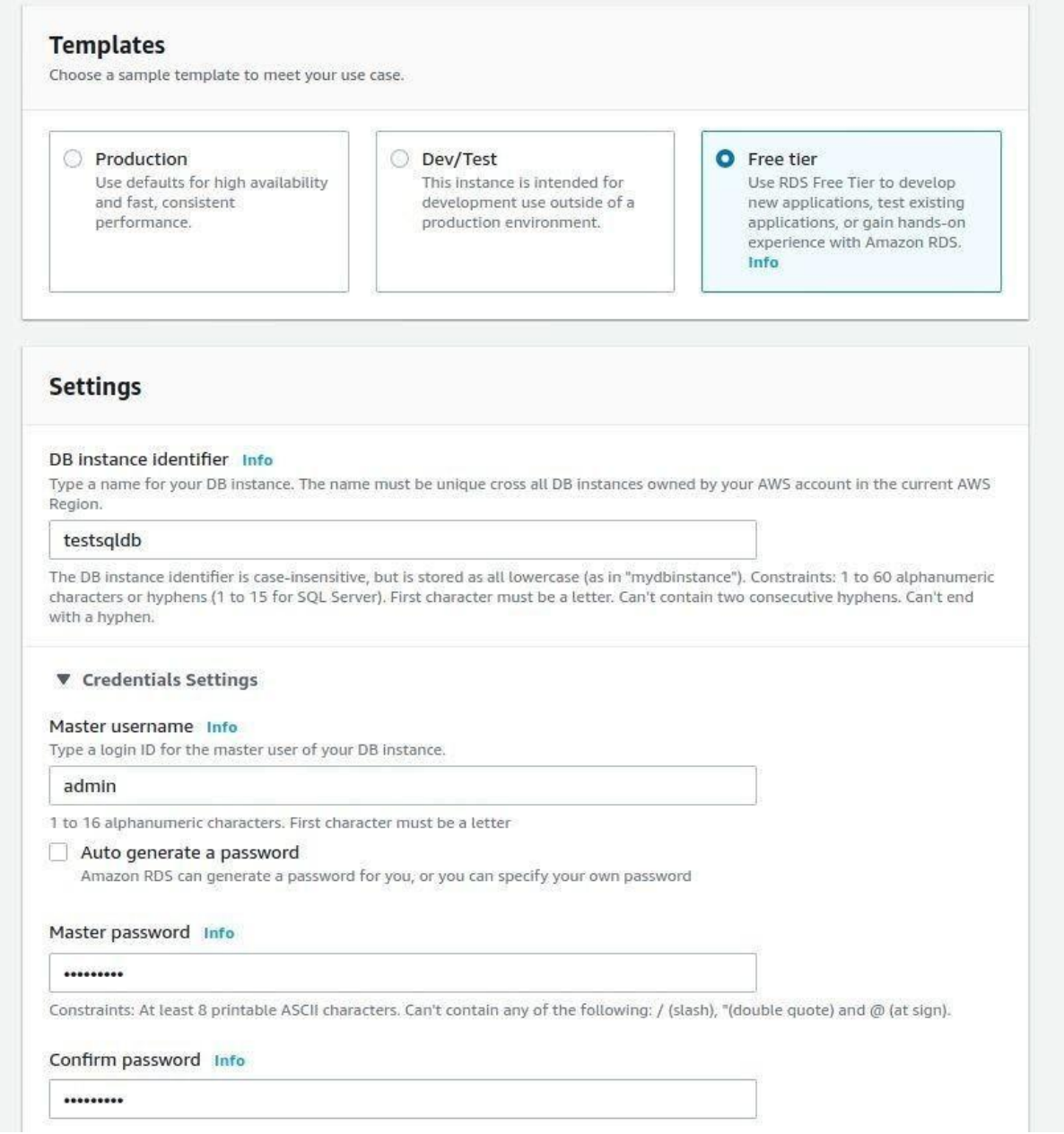
Step 2: Select Create Database



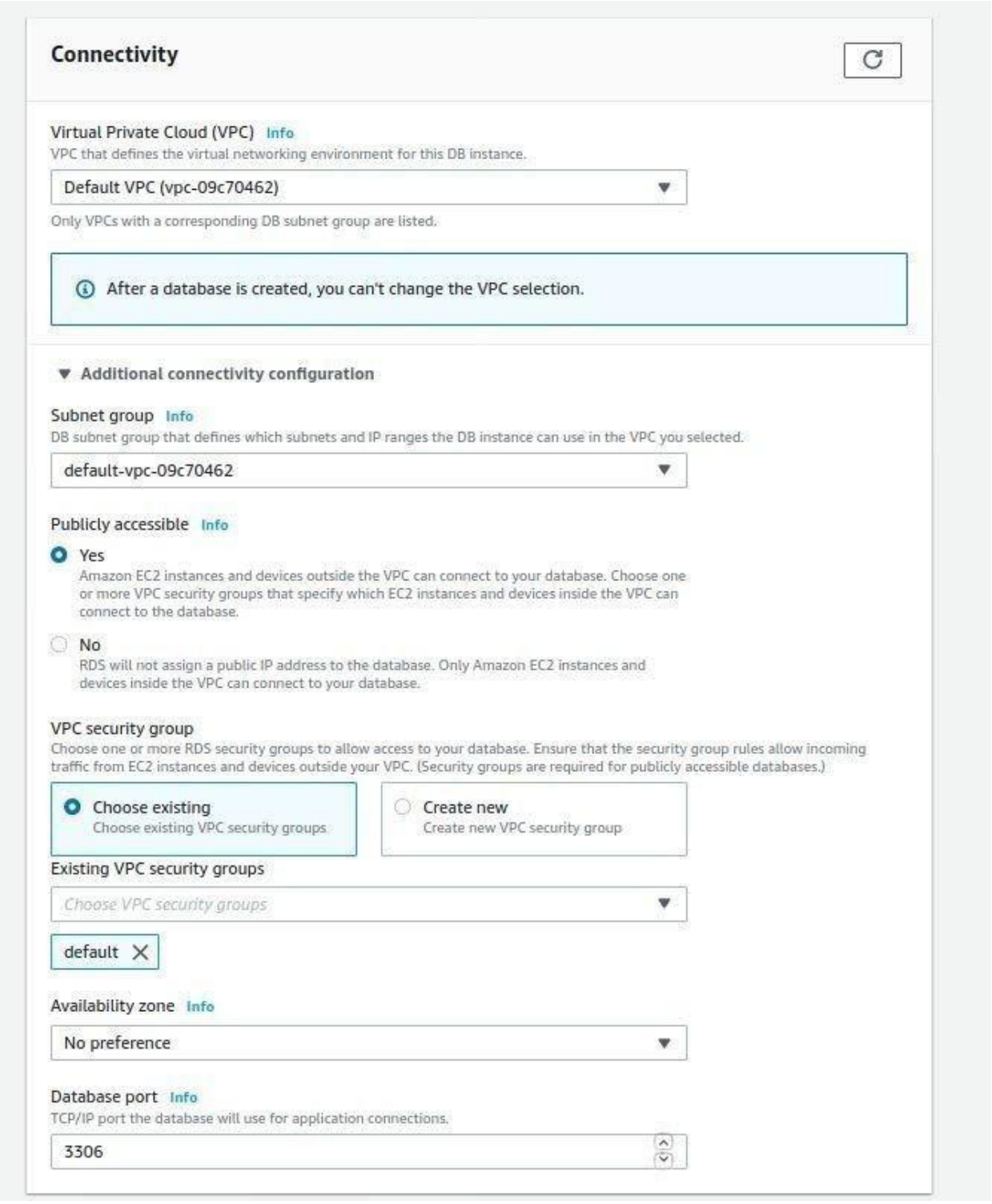
Step 3: Select Standard Create -> Engine Type MySQL (if you SQL Workbench installed)
or
PostgreSQL (if you pgadmin installed)



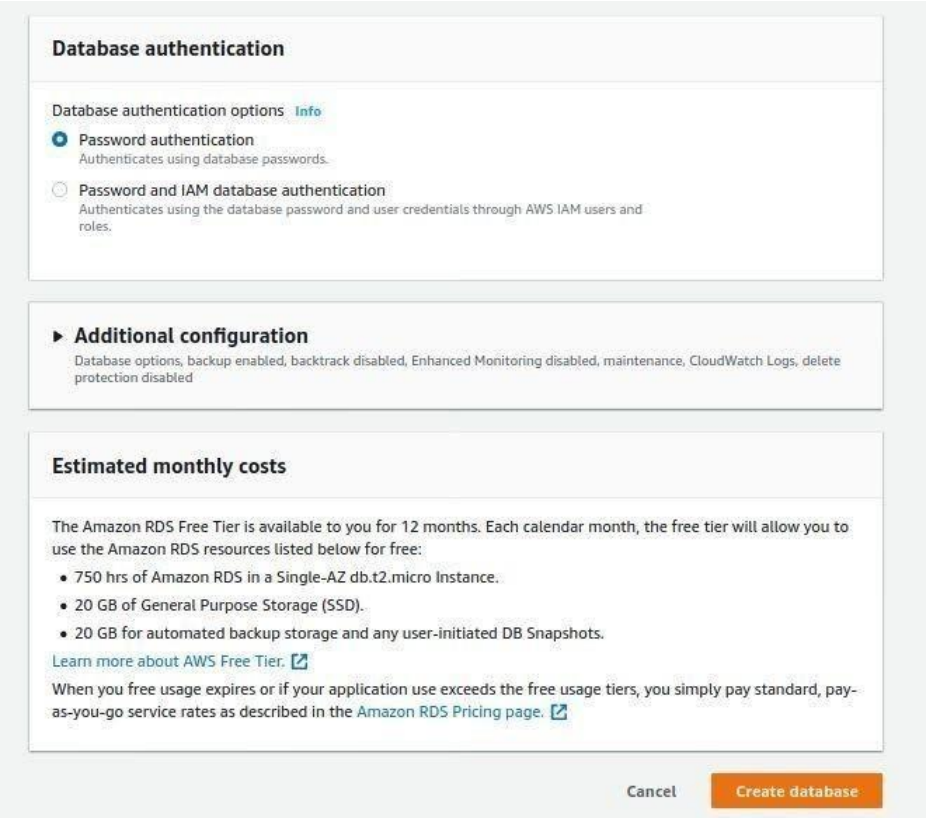
Step 4: Select Free Tier (important) , name you db and add username , password.



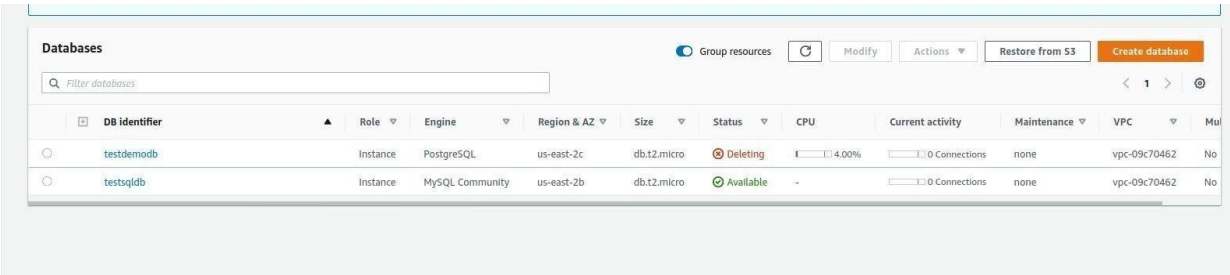
Step 5: Select Default VPC Group & Publicly accessible as YES



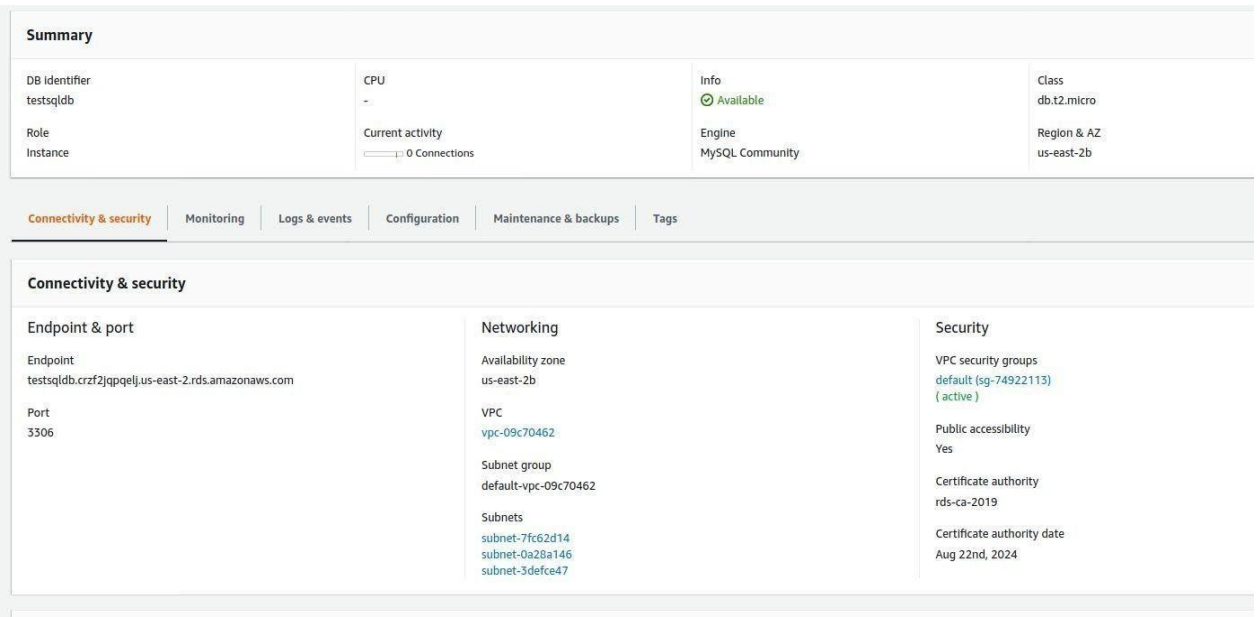
Step 6: Create Database



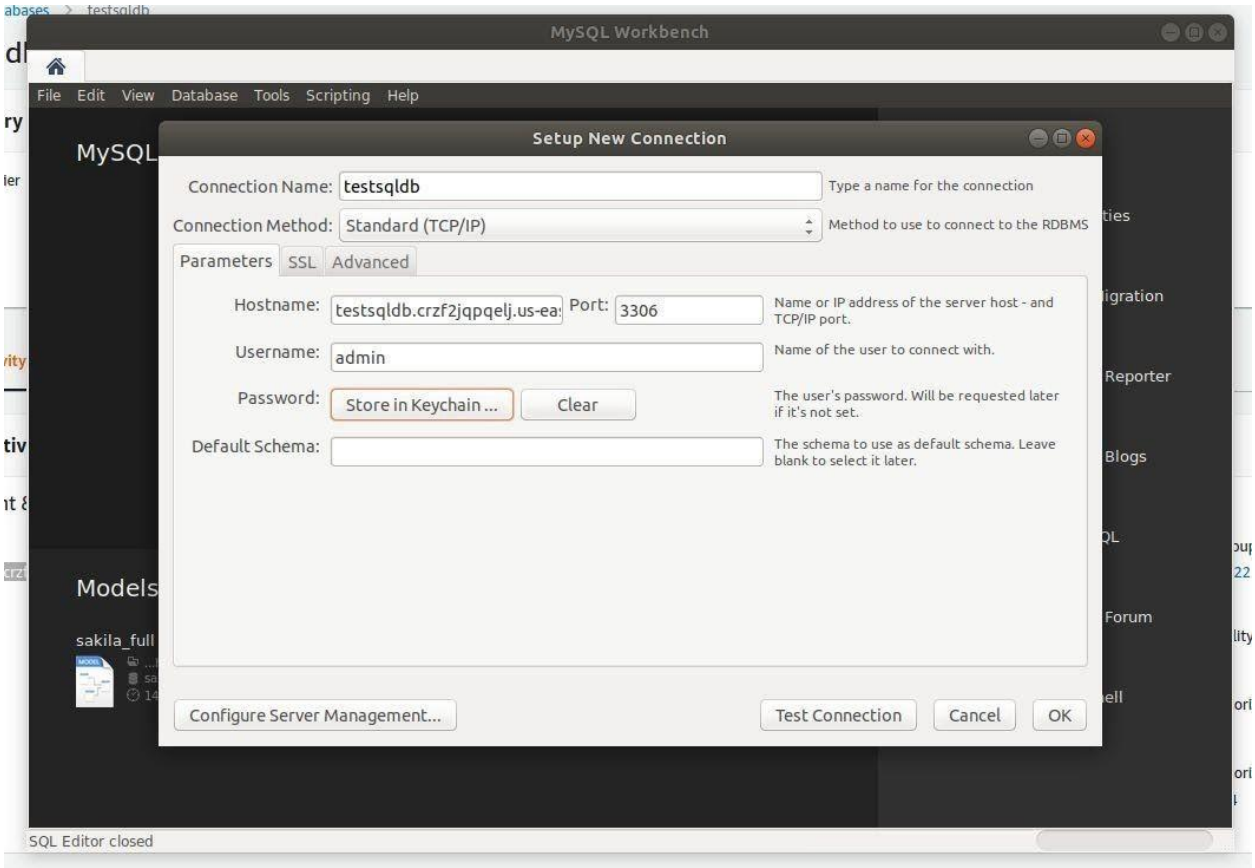
Step 7: Wait until your database is available (this will take sometime) , then click on your database



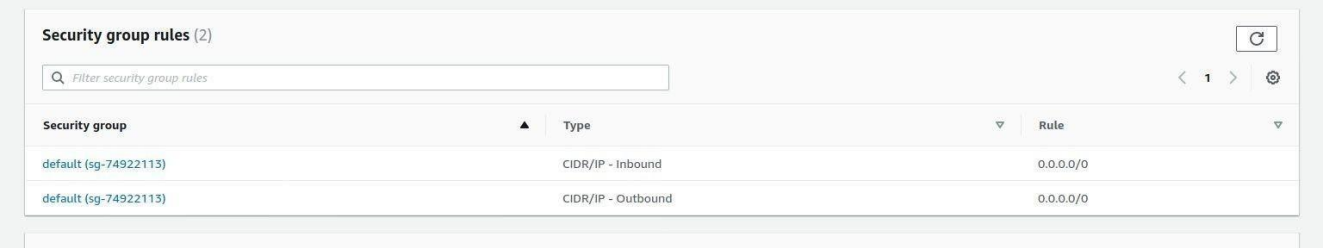
Step 8: Copy the Endpoint given in Connectivity & Security



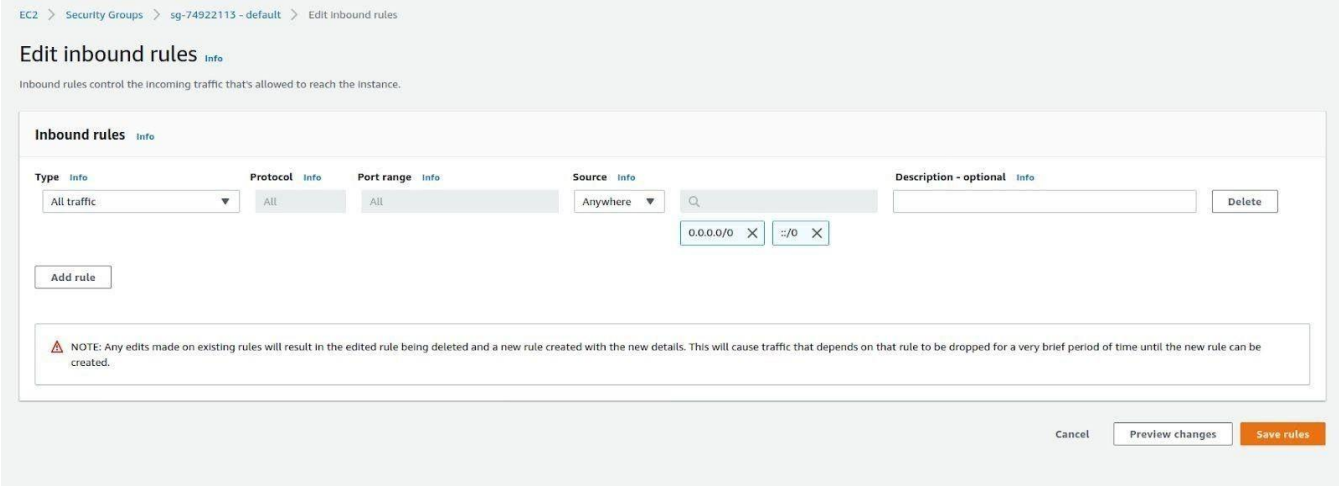
Step 9: Open MySQL workbench & test your connection after entering credentials



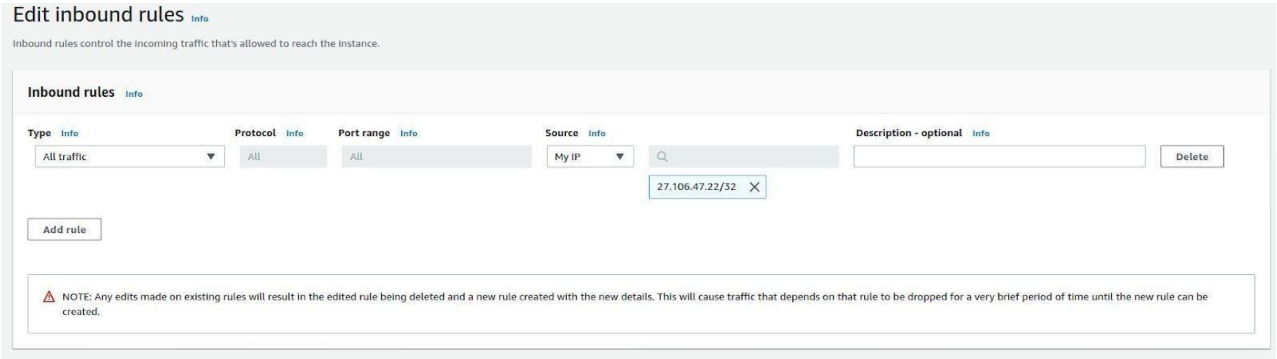
If you fail to establish connection, check if security group inbound rules have 0.0.0.0/0 as RULE



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



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

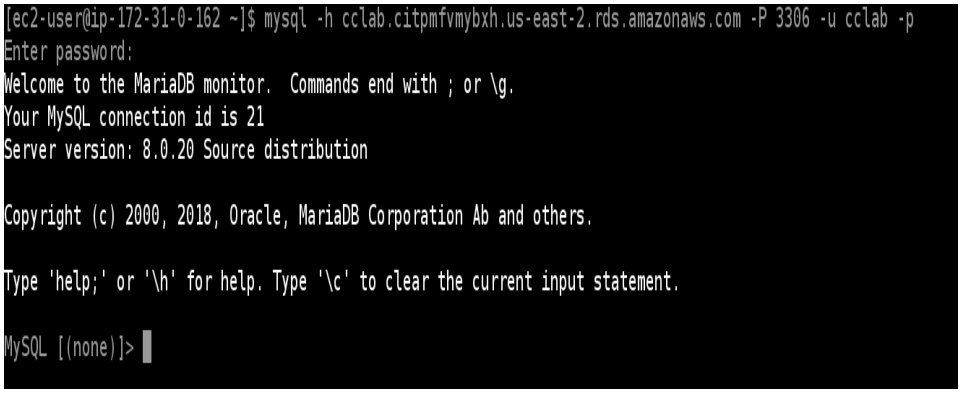


Step 10: Open the EC2 terminal

Enter following Commands to setup MySQL connection

a) sudo yum install mysql

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



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','Advaith 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.