



## AWS Solutions Architect Training

### Module-7: Database Services

## Module 7: MariaDB Assignment

#### **Problem Statement:**

You work for XYZ Corporation. Their application requires a SQL service that can store data which can be retrieved if required. Implement a suitable RDS engine for the same.

#### **While migrating, you are asked to perform the following tasks:**

1. Create a MariaDB Engine based RDS Database.
2. Connect to the DB using the following ways:
  - a. SQL Client for Windows
  - b. Linux based EC2 Instance

## Module 7: Aurora Assignment

#### **Problem Statement:**

You work for XYZ Corporation. Their application requires a SQL service that can store data which can be retrieved if required. Implement a suitable RDS engine for the same.

#### **While migrating, you are asked to perform the following tasks:**

1. Create an AuroraDB Engine based RDS Database.
2. Create 2 Read Replicas in different availability zones for better infrastructure availability.

## Module 7: DynamoDB Assignment

### **Problem Statement:**

You work for XYZ Corporation. Their application requires a database service that can store data which can be retrieved if required. Implement a suitable service for the same.

### **While migrating, you are asked to perform the following tasks:**

1. Create a DynamoDB table with partition key as ID.
2. Add 5 items to the DynamoDB table.
3. Take backup and delete the table.

## Module 7: Redshift Assignment

### **Problem Statement:**

You work for XYZ Corporation. Their application requires a database service that can store data which can be retrieved if required. Implement suitable service for the same.

### **While migrating, you are asked to perform the following tasks:**

1. Create a Redshift data warehouse.
2. Using the query editor:
  - a. Load some data
  - b. Query the data

## MariaDB Assignment

Go to AWS Management Console → RDS → Databases → Create Database

The screenshot shows the AWS Management Console with the RDS service selected. The left sidebar includes links for Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, Recommendations, and Certificate update. The main content area features a banner for 'Introducing Aurora I/O-Optimized'. Below it, the 'Resources' section lists DB Instances (0/40), DB Clusters (0/40), and DB Snapshots (0). The 'Create database' section contains a 'Create database' button. To the right, there's a 'Recommended services' section with links to IAM, AWS Organizations, Elastic Container Service, Amazon OpenSearch Service, and Elastic Container Registry. Another section titled 'Recommended for you' includes links for Amazon RDS Backup and Restore using AWS Backup and Implementing Cross-Region DR.

This screenshot shows the 'Create database' wizard. The first step, 'Choose a database creation method', offers 'Standard create' (selected) and 'Easy create'. The second step, 'Engine options', lists several database engines: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL, MariaDB (selected), PostgreSQL, Oracle, Microsoft SQL Server, and IBM Db2. On the right side, a detailed description of MariaDB is provided, highlighting its MySQL compatibility, support for open source commands, and various instance classes.

Select MariaDB

Templates → Free Tier

The screenshot shows the AWS RDS console for creating a new database instance. In the top left, there's a search bar and a dropdown menu for 'Services'. On the right, there's a user profile for 'mariadbadmin' and a region selector for 'N. Virginia'. The main area has a title 'MariaDB' with a close button. Below it, a detailed description of the MariaDB engine is provided, mentioning MySQL compatibility and various features like automated backups and cross-region replication. On the left, there are filters for 'Engine version' (set to 'MariaDB 10.11.6') and 'Show versions that support the Amazon RDS Optimized Writes' (which is checked). Under 'Templates', three options are listed: 'Production' (radio button), 'Dev/Test' (radio button), and 'Free tier' (radio button, which is selected). A note says 'Choose a sample template to meet your use case.'

The screenshot shows the 'Settings' page for creating a new DB instance. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, and a search bar. To the right of the search bar is a button '[Alt+S]'. Below the search bar, there's a callout 'experience with Amazon RDS.' with a 'Info' link. The main content area has a title '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.  
 Info

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.  
 Info

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](#)  
 Info

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

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

Master password is 12345678 and confirm master password 12345678

The screenshot shows the AWS RDS Instance Configuration page. At the top, there is a navigation bar with the AWS logo, 'Services' (selected), a search bar, and a '[Alt+S]' keyboard shortcut. Below the navigation bar, a message states: 'Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / " @'. A 'Confirm master password' field contains '.....'. In the main content area, under 'Instance configuration', it says 'The DB instance configuration options below are limited to those supported by the engine that you selected above.' Under 'DB instance class', there is a 'Hide filters' button and two filter options: 'Show instance classes that support Amazon RDS Optimized Writes' (selected) and 'Include previous generation classes'. Below these filters, three class types are listed: 'Standard classes (includes m classes)', 'Memory optimized classes (includes r and x classes)', and 'Burstable classes (includes t classes)'. The 'Burstable classes' option is selected. A dropdown menu shows 'db.t3.micro' with details: '2 vCPUs', '1 GiB RAM', and 'Network: 2,085 Mbps'. The entire configuration section is enclosed in a light gray border.

Keep default settings for Connectivity, Tags and Database Authentication

[Go to Additional Configuration](#)

AWS Services Search [Alt+S]

## Connectivity Info

**Compute resource**  
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

**Don't connect to an EC2 compute resource**  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

**Connect to an EC2 compute resource**  
Set up a connection to an EC2 compute resource for this database.

**Network type Info**  
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

**IPv4**  
Your resources can communicate only over the IPv4 addressing protocol.

**Dual-stack mode**  
Your resources can communicate over IPv4, IPv6, or both.

**Virtual private cloud (VPC) Info**  
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-0f0b59488a783673d)  
6 Subnets, 6 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

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

default

**Public access Info**  
 **Yes**  
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

**No**  
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

**VPC security group (firewall) Info**  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

**Choose existing**  
Choose existing VPC security groups

**Create new**  
Create new VPC security group

**Existing VPC security groups**

Choose one or more options

default X

**DB subnet group** [Info](#)

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

default-vpc-0f0b59488a783673d

6 Subnets, 6 Availability Zones

**Public access** [Info](#) Yes

RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

 No

RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

**VPC security group (firewall)** [Info](#)

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

 Choose existing

Choose existing VPC security groups

 Create new

Create new VPC security group

## Existing VPC security groups

*Choose one or more options*



The screenshot shows the AWS RDS (Relational Database Service) configuration page for a new database instance. At the top, the AWS logo and 'Services' menu are visible, along with a search bar and keyboard shortcut [Alt+S].

**Certificate Authority:** A dropdown menu shows 'rds-ca-rsa2048-g1 (default)' with an expiry date of 'May 26, 2061'. A note states: 'If you don't select a certificate authority, RDS chooses one for you.'

**Additional configuration:** A section with a '▶ Additional configuration' button.

**Tags:** A section for adding key-value pairs. It shows 'No tags associated with the resource.' and a button to 'Add new tag'. A note says: 'You can add up to 50 more tags.'

**Database authentication:** A section titled 'Database authentication options' with an 'Info' link. It shows two options: 'Password authentication' (selected, with a note 'Authenticates using database passwords.') and 'Password and IAM database authentication' (with a note 'Authenticates using the database password and user credentials through AWS IAM users and roles.').

**Monitoring:** A section with a checkbox for 'Enable Enhanced Monitoring' (unchecked), which is described as useful for seeing CPU usage by processes or threads.

Go to Additional Configuration

Give a name to the initial database

Firtstdb

AWS Services Search [Alt+S]

Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Additional configuration

Database options

Initial database name [Info](#) firstdata

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#) default.mariadb10.11

Option group [Info](#) default:mariadb-10-11

Backup

Enable automated backups  
Creates a point-in-time snapshot of your database

Backup retention period [Info](#) 1 day

The number of days (1-35) for which automatic backups are kept.

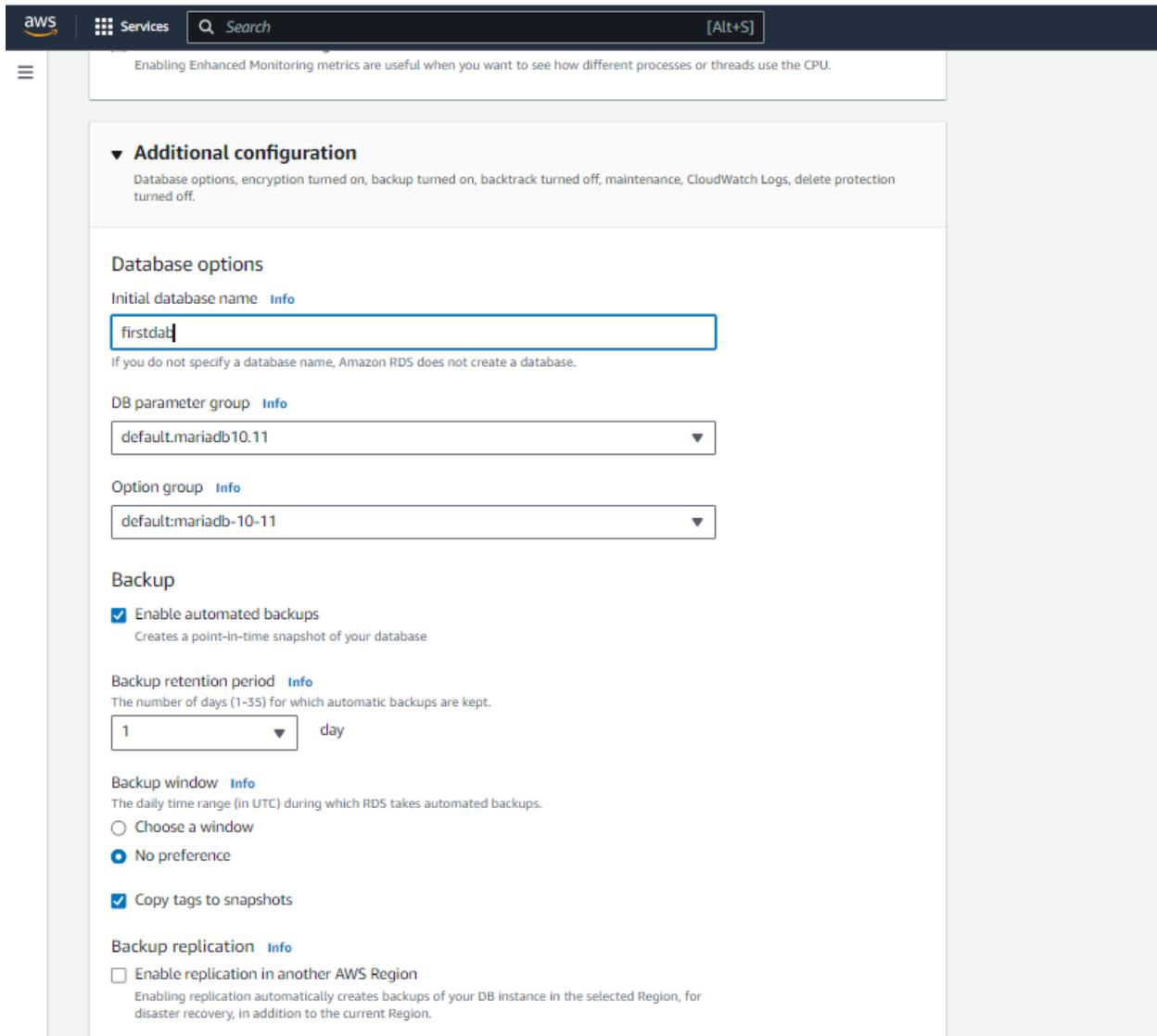
Backup window [Info](#) The daily time range (in UTC) during which RDS takes automated backups.

Choose a window  
 No preference

Copy tags to snapshots

Backup replication [Info](#)

Enable replication in another AWS Region  
Enabling replication automatically creates backups of your DB instance in the selected Region, for disaster recovery, in addition to the current Region.



Keep the remaining settings to defaults and click on Create database

AWS Services Search [Alt+S]

Auto minor version upgrade [Info](#)  
 **Enable auto minor version upgrade**  
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window [Info](#)  
Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.  
 Choose a window  
 No preference

Deletion protection

**Enable deletion protection**  
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

### Estimated Monthly costs

|              |                  |
|--------------|------------------|
| DB instance  | 12.41 USD        |
| Storage      | 2.30 USD         |
| <b>Total</b> | <b>14.71 USD</b> |

This billing estimate is based on on-demand usage as described in [Amazon RDS Pricing](#). Estimate does not include costs for backup storage, I/Os (if applicable), or data transfer.

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#).

### 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, db.t3.micro or db.t4g.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 your 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](#).

**ⓘ You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.**

Cancel **Create database**

Screenshot of the Amazon RDS console showing the creation of a database named "mariadb-assignment". The database is currently "Creating". A tooltip suggests creating a Blue/Green Deployment. The table lists the database details.

| DB Identifier      | Status   | Role     | Engine  | Region & AZ | Size        | Recommendations | CPU | Current activity | Maintenance | VPC                  | Multi-AZ |
|--------------------|----------|----------|---------|-------------|-------------|-----------------|-----|------------------|-------------|----------------------|----------|
| mariadb-assignment | Creating | Instance | Mariadb | us-east-1c  | db.t3.micro | -               | -   | -                | none        | vpc-0f0b0948a783672d | No       |

## Suggested add-ons for mariadb-assignment

Simplify the configuration of the following suggested add-ons by using settings from your new database.



**Create an ElastiCache cluster from RDS using your DB settings - *new***

You can save costs and improve read performance by using ElastiCache with RDS versus running on RDS alone.

*\*For example: you can save up to 55% in cost and gain up to 80x faster read performance using ElastiCache with RDS for MySQL (vs. RDS for MySQL alone).*

[Learn more](#)

[Create ElastiCache cluster](#)



**Use RDS Proxy**

Using a proxy allows your applications to pool and share database connections to help them scale. A proxy simplifies connection management and makes applications more resilient to database failures.

[Learn more](#)

[Create proxy](#)

**i** You can hide these suggestions so they don't appear after database creation. All these actions can be taken from the database list page or database details page.

Hide add-ons for 30 days [Close](#)

Click on close

The screenshot shows the AWS RDS console. On the left, there's a sidebar with various options like Dedicated Databases, Query Editor, Performance insights, etc. The main area shows a message about creating the database and then displays the 'Connection details to your database mariadb-assignment'. It shows the master username as 'admin' and the master password as '12345678'. There are buttons for 'Actions', 'Restore From S3', and 'Create database'. Below this, there's a table with columns for DB identifier, Status, Role, Engine, Region & AZ, Size, Recommendations, CPU, and Current activity. One row is visible for 'mariadb-assignment'.

## View credentials details

This screenshot shows the AWS RDS console after the database has been successfully created. A green banner at the top says 'Successfully created database mariadb-assignment'. Below it, there's a message about Aurora I/O-Optimized. The main area shows the 'Databases' section with a table. The table has columns for DB identifier, Status, Role, Engine, Region & AZ, Size, Recommendations, CPU, and Current activity. One row is visible for 'mariadb-assignment'.

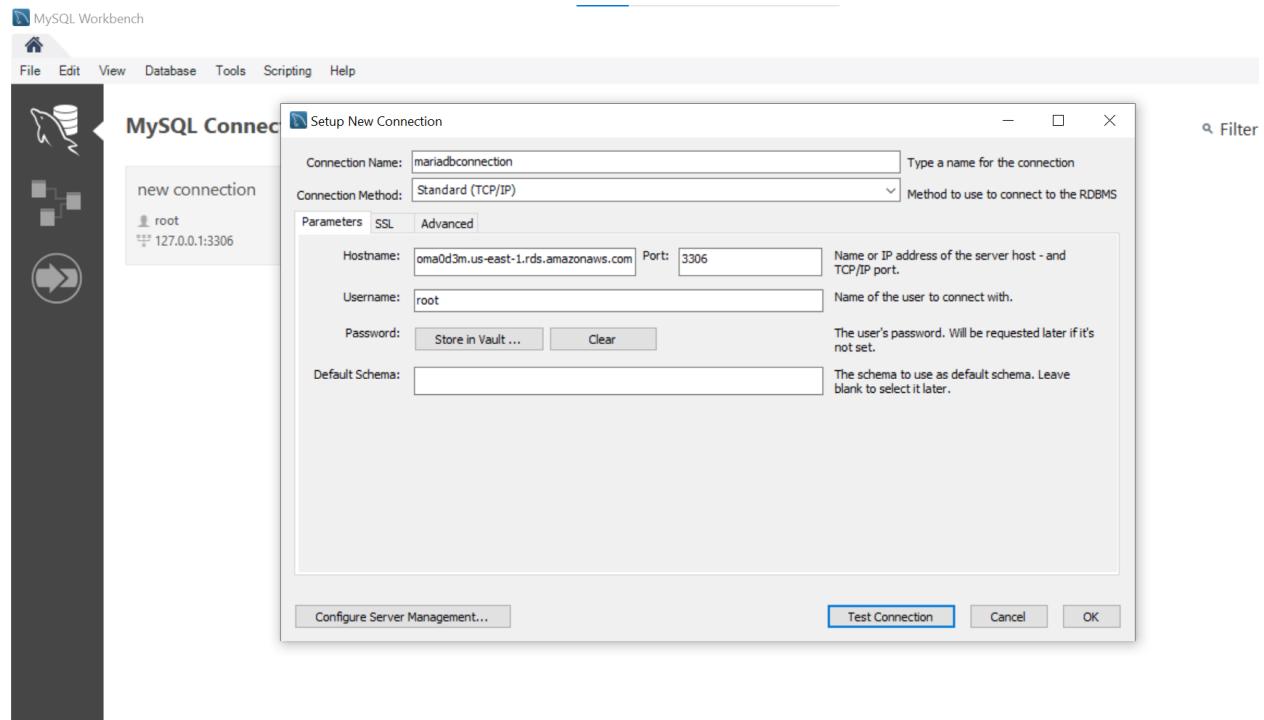
Database successfully created

It takes some time to create this database. Meanwhile download and install MySQL Workbench

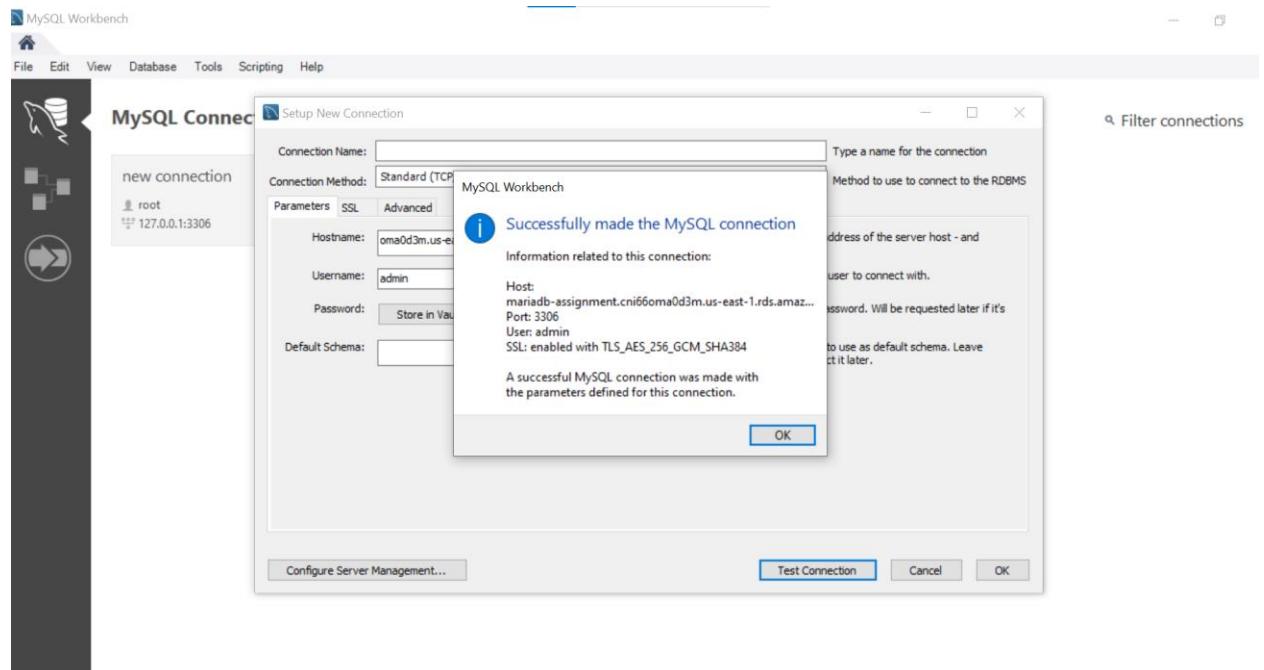
MySQL Workbench installed

The screenshot shows the MySQL Workbench interface. At the top, there's a menu bar with File, Edit, View, Database, Tools, Scripting, and Help. The main area is titled 'MySQL Connections' and shows a list with one item: 'new connection' under 'root' with the host '127.0.0.1:3306'. On the left, there's a sidebar with icons for new connection, schema browser, and query editor.

Now open MySQL workbench and enter the required details. Click on Store in Vault to enter the password



Click on test connection



Then click on OK

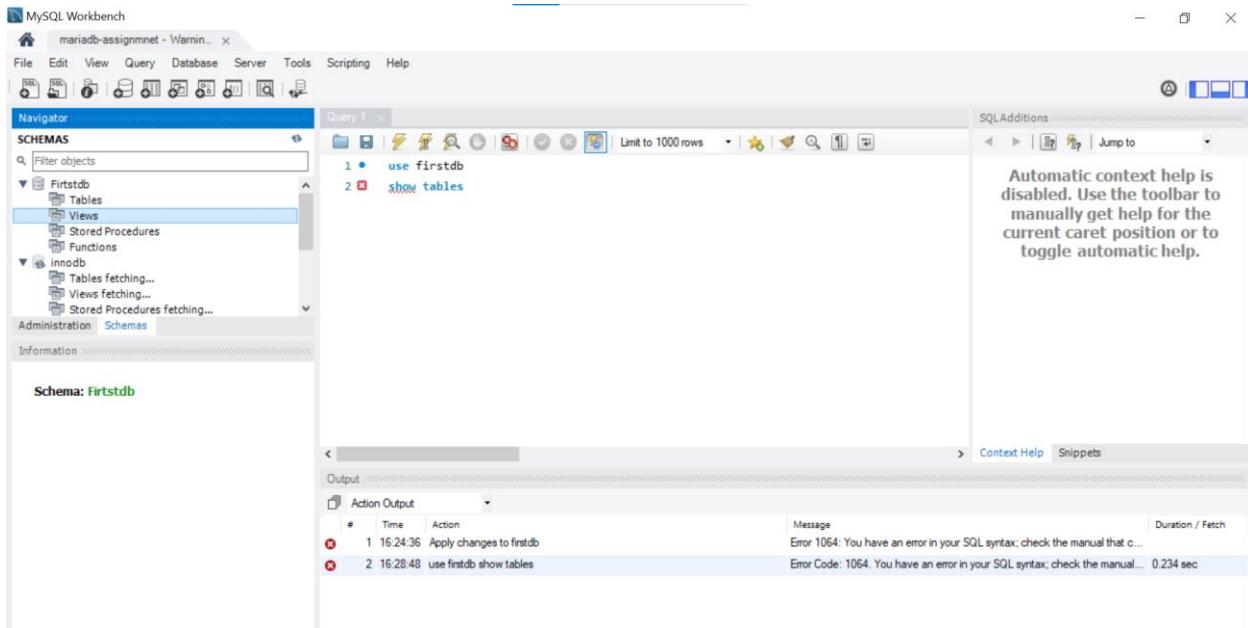
The screenshot shows two windows of MySQL Workbench.

**MySQL Connections Window:**

- File Edit View Database Tools Scripting Help
- MySQL Connections
- new connection
- mariadb-assignmnet
- root 127.0.0.1:3306
- admin mariadb-assignment.cn166oma0d3...

**Schema Details Window:**

- File Edit View Query Database Server Tools Scripting Help
- mariadb-assignmnet - Warning
- Query 1 Firtstdb Firtstdb - Schema
- Info Tables Columns Indexes Triggers Views Stored Procedures Functions Grants Events
- Navigator
- SCHEMAS Filter objects
- Firtstdb Tables Views Stored Procedures Functions
- innodb sys
- Administration Schemas
- Information
- Schema: Firtstdb
- Schema Details
- Default collation: latin1\_swedish\_ci
- Default characterset: latin1
- Table count: 0
- Database size (rough estimate): 0.0 bytes
- SQLAdditions
- Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
- Output Action Output
- Action Output # Time Action
- 1 16:24:36 Apply changes to firstdb
- Message Error 1064: You have an error in your SQL syntax; check the manual that c...
- Duration / Fetch
- Context Help Snippets



You can now run SQL queries and manage your database from the SQL client

## Linux based EC2 Instance

Let's Connect to the DB using a Linux-Based EC2 Instance. Create an Amazon Linux instance in the same VPC as the database.

Then install MariaDB client.

```
sudo yum update -y      >>>>  sudo yum install mariadb105 -y
```

Note the **Endpoint** and **Port** from the RDS console as mentioned before.

Connect to MariaDB using the below command

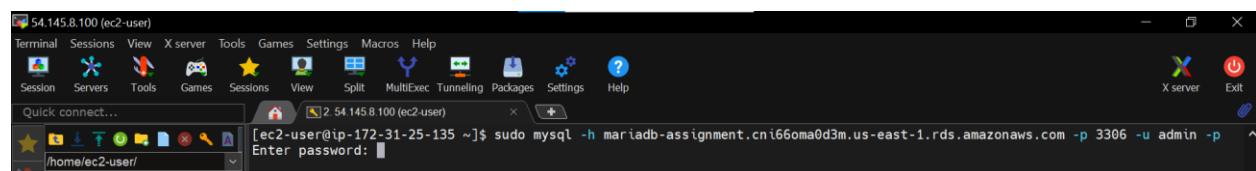
```
mysql -h <endpoint> -P <port> -u <username> -p
```

```
mysql -h mariadb-assignment.cn66oma0d3m.us-east-1.rds.amazonaws.com -p 3306 -u
```

```
admin -p
```

**Enter password when prompted**

**12345678**



```
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 82
Server version: 10.11.6-MariaDB-log managed by https://aws.amazon.com/rds/

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)]> 
```

It is successfully connected. you can run SQL commands directly from the terminal

We've successfully set up a MariaDB RDS instance and connect to it from both a Windows SQL client and a Linux-based EC2 instance

1. Next, we'll do AuroraDB Assignment

## **2. Problem Statement:**

3. You work for XYZ Corporation. Their application requires a SQL service that can
4. store data which can be retrieved if required. Implement a suitable RDS engine
5. for the same.

## **6. While migrating, you are asked to perform the following**

### **7. tasks:**

8. 1. Create an AuroraDB Engine based RDS Database.
9. 2. Create 2 Read Replicas in different availability zones for better
10. infrastructure availability.

Go to AWS Management Console → RDS → Databases → 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](#)

**Aurora (MySQL Compatible)**

**Aurora (PostgreSQL Compatible)**

**MySQL**

**MariaDB**

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

## Master user admin

Password is admin#12345678

**Credentials Settings**

Master username: [Info](#)  
Type a login ID for the master user of your DB instance.

1 to 32 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](#)

Password strength: **Strong**

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

Confirm master password: [Info](#)

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

## Keep default settings for Connectivity, Tags and Database Authentication

**Cluster storage configuration - new** [Info](#)

Choose the storage configuration for the Aurora DB cluster that best fits your application's price predictability and price performance needs.

Configuration options  
Database instance, storage, and I/O charges vary depending on the configuration. [Learn more](#)

**Aurora Standard**

- Cost-effective pricing for many applications with moderate I/O usage (I/O costs <25% of total database costs).
- Pay-per-request I/O charges apply. DB instance and storage prices don't include I/O usage.

**Aurora I/O-Optimized**

- Predictable pricing for all applications. Improved price performance for I/O-intensive applications (I/O costs >25% of total database costs).
- No additional charges for read/write I/O operations. DB instance and storage prices include I/O usage.

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms

## Select db.t3.medium

**Instance configuration**

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class: [Info](#)  
 **db.t3.medium**  
2 vCPUs 4 GiB RAM Network: 2,085 Mbps

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones

## Go to Additional Configuration

**Availability & durability**

Multi-AZ deployment: [Info](#)  
 **Don't create an Aurora Replica**  
Creates an Aurora Replica for fast failover and high availability.

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

**Compute resource**  
Creates an Aurora Replica for fast failover and high availability.

Don't create an Aurora Replica

**Connectivity Info**

**Compute resource**  
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource  
Set up a connection to an EC2 compute resource for this database.

**Network type Info**  
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4  
Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode  
Your resources can communicate over IPv4, IPv6, or both.

**Virtual private cloud (VPC) Info**  
Choose the VPC. The VPC defines the virtual networking environment for this DB cluster.

Default VPC (vpc-0fb59488a783673d)  
6 Subnets, 6 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

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

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

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

**DB subnet group Info**  
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB cluster can use in the VPC that you selected.

default-vpc-0fb59488a783673d  
6 Subnets, 6 Availability Zones

**Public access Info**  
 Yes  
RDS assigns a public IP address to the cluster. Amazon EC2 instances and other resources outside of the VPC can connect to your cluster. Resources inside the VPC can also connect to the cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

No  
RDS doesn't assign a public IP address to the cluster. Only Amazon EC2 instances and other resources inside the VPC can connect to your cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

**VPC security group (firewall) Info**  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing  
Choose existing VPC security groups

Create new  
Create new VPC security group

**Existing VPC security groups**  
Choose one or more options

default

**Availability Zone Info**  
No preference

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

## Choose default VPC

Choose one or more options

default

**Availability Zone Info**  
No preference

**RDS Proxy**  
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

Create an RDS Proxy  
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

**Certificate authority - optional Info**  
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)  
Expiry: May 26, 2061

If you don't select a certificate authority, RDS chooses one for you.

**Additional configuration**

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

Go to Additional Configuration

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

Go to additional configuration

Give a name to the initial database.

Keep the remaining settings to defaults and click on Create

**AWS Services** Search [Alt+S] N. Virginia maheshzadeitm

**Additional configuration**

Database options  
Initial database name: [Info](#) myfirstAuroraDB

If you do not specify a database name, Amazon RDS does not create a database.

DB cluster parameter group: [Info](#) default.aurora-mysql8.0

DB parameter group: [Info](#) default.aurora-mysql8.0

Option group: [Info](#) default.aurora-mysql-8-0

Failover priority: No preference

**Backup**

Backup retention period: [Info](#) 1 day

Copy tags to snapshots

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

**AWS Services** Search [Alt+S] N. Virginia maheshzadeitm

Copy tags to snapshots

**Encryption**

Enable encryption Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. [Info](#)

AWS KMS key: [Info](#) (default) aws/rds

Account: 058264440832

KMS key ID: ca2478b0-5065-4c28-ba6a-4f54735c4da6

**Backtrack**

Backtrack lets you quickly rewind the DB cluster to a specific point in time, without having to create another DB cluster. [Info](#)

Enable Backtrack Enabling Backtrack will charge you for storing the changes you make for backtracking.

**Log exports**

Select the log types to publish to Amazon CloudWatch Logs

Audit log  
 Error log  
 General log  
 Slow query log

**IAM role**

The following service-linked role is used for publishing logs to CloudWatch Logs.

[cloudShell](#) [Feedback](#) © 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

The screenshot shows the AWS RDS configuration interface for creating a new database. On the left, there are sections for 'Maintenance' (with options for auto minor version upgrade and maintenance window), 'Deletion protection' (disabled), and 'Estimated Monthly costs' (showing a total of \$59.86 USD). A modal window titled 'Aurora MySQL-Compatible Edition' provides detailed information about the service, including its benefits like up to 128 TB of storage and six-way replication. At the bottom, a note states responsibility for third-party services and includes links for pricing and a simple monthly calculator.

**Maintenance**

Auto minor version upgrade [Info](#)  
 **Enable auto minor version upgrade**  
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window [Info](#)  
Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.  
 Choose a window  
 No preference

**Deletion protection**

**Enable deletion protection**  
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

**Estimated Monthly costs**

|              |                  |
|--------------|------------------|
| DB instance  | 59.86 USD        |
| <b>Total</b> | <b>59.86 USD</b> |

This billing estimate is based on on-demand usage as described in [Amazon Aurora Pricing](#). Estimate does not consider reserved instance benefits and costs for instance storage, IOs, or data transfer.

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#).

ⓘ You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Aurora MySQL-Compatible Edition

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

The screenshot shows a confirmation step for the estimated monthly costs. It displays the same cost information as the previous screen (\$59.86 USD) and includes a note about the billing estimate being based on on-demand usage. Below this, there is a link to the AWS Simple Monthly Calculator. At the bottom, a note reiterates responsibility for third-party services, and there are 'Cancel' and 'Create database' buttons.

**Estimated Monthly costs**

|              |                  |
|--------------|------------------|
| DB instance  | 59.86 USD        |
| <b>Total</b> | <b>59.86 USD</b> |

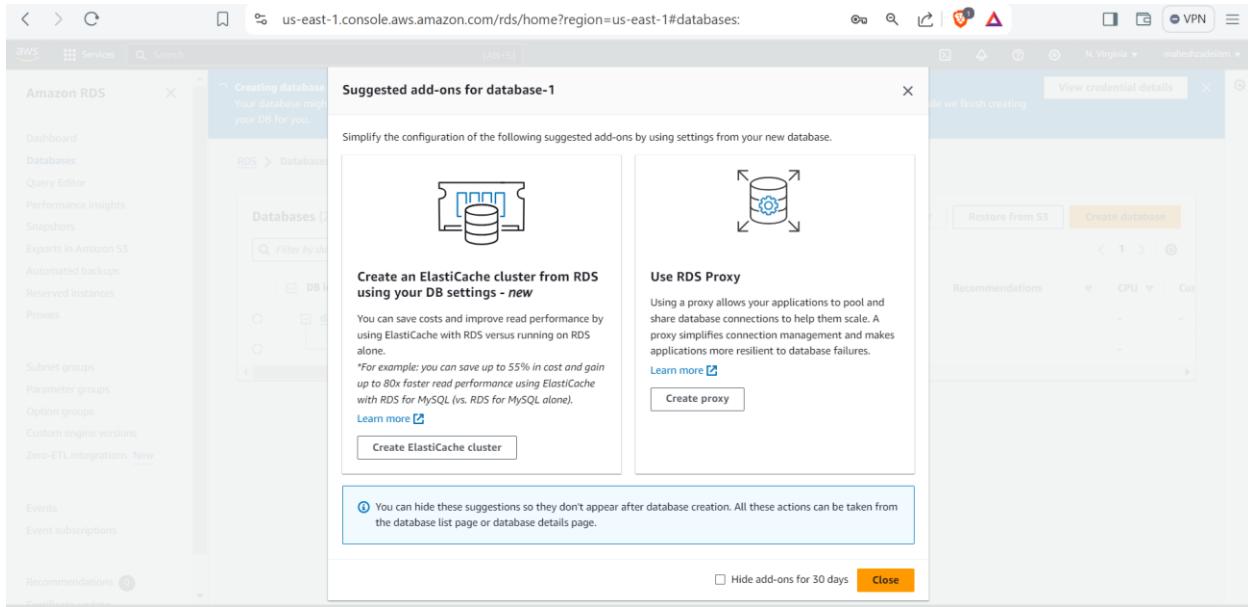
This billing estimate is based on on-demand usage as described in [Amazon Aurora Pricing](#). Estimate does not consider reserved instance benefits and costs for instance storage, IOs, or data transfer.

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#).

ⓘ You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Cancel **Create database**

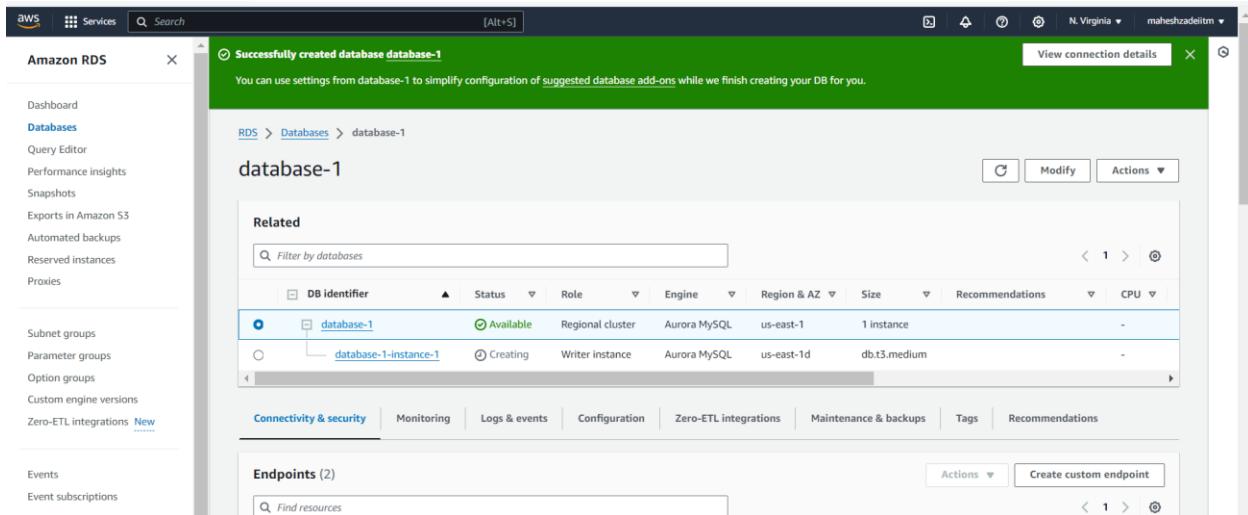
Click **Create database**



Close

database is created and is available now wait till database instance created .

note endpoint name : database-1.cluster-cni66oma0d3m.us-east-1.rds.amazonaws.com



Databases (2)

| DB identifier         | Status    | Role             | Engine       | Region & AZ | Size         | Recommendations | CPU    |
|-----------------------|-----------|------------------|--------------|-------------|--------------|-----------------|--------|
| database-1            | Available | Regional cluster | Aurora MySQL | us-east-1   | 1 instance   | -               | -      |
| database-1-instance-1 | Available | Writer instance  | Aurora MySQL | us-east-1d  | db.t3.medium | 43.45%          | 43.45% |

click on view connection details

Successfully created database database-1

You can use settings from database-1 to simplify configuration of suggested database add-ons while we finish creating your DB for you.

View connection details

Connection details to your database database-1

This is the only time you can view this password. Copy and save the password for your reference. If you lose the password, you must modify your database to change it. You can use a SQL client application or utility to connect to your database.

Learn about connecting to your database

Master username  
admin

Master password  
admin#12345678

Endpoint  
database-1.cluster-cni66oma0d3m.us-east-1.rds.amazonaws.com

Master username : admin

Master password : admin#12345678

Endpoint : database-1.cluster-cni66oma0d3m.us-east-1.rds.amazonaws.com

Now Let's create read replicas in 2 different Azs. Go to Actions -> Add reader

AWS Services Search [Alt+S] N. Virginia maheshzadeitm

Amazon RDS Databases

Databases (2)

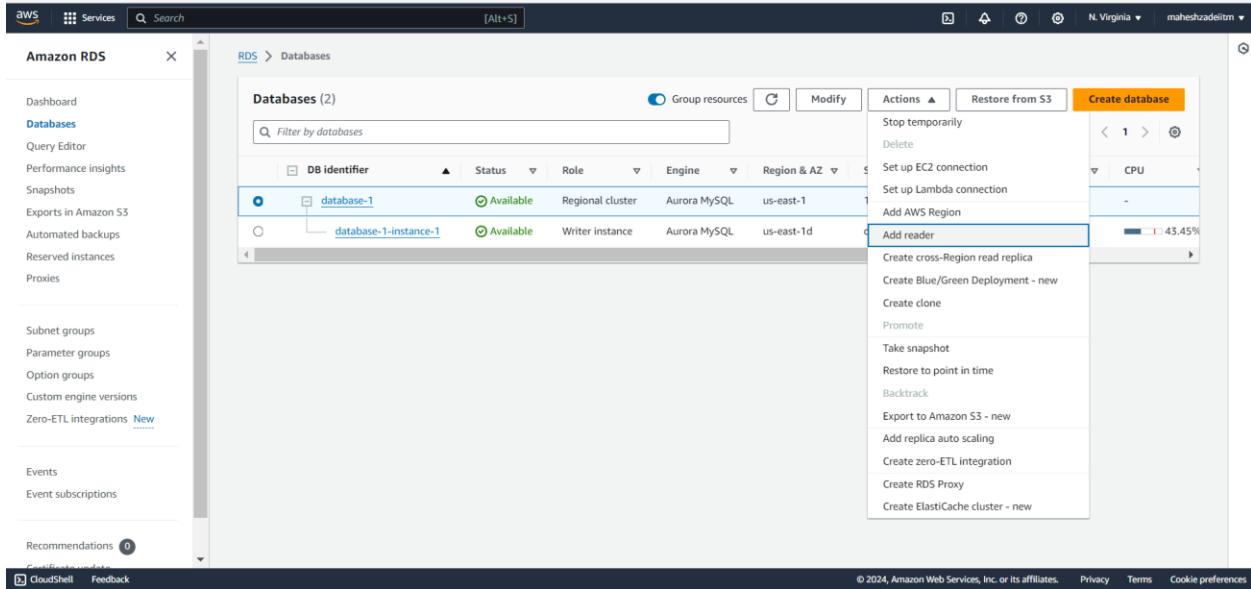
| DB identifier         | Status    | Role             | Engine       | Region & AZ |
|-----------------------|-----------|------------------|--------------|-------------|
| database-1            | Available | Regional cluster | Aurora MySQL | us-east-1   |
| database-1-instance-1 | Available | Writer instance  | Aurora MySQL | us-east-1d  |

Actions ▾ Restore from S3 Create database

Stop temporarily Delete Set up EC2 connection Set up Lambda connection Add AWS Region Add reader 43.45% Create cross-Region read replica Create Blue/Green Deployment - new Create clone Promote Take snapshot Restore to point in time Backtrack Export to Amazon S3 - new Add replica auto scaling Create zero-ETL integration Create RDS Proxy Create ElastiCache cluster - new

Subnet groups Parameter groups Option groups Custom engine versions Zero-ETL integrations New Events Event subscriptions Recommendations 0 CloudShell Feedback

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



AWS Services Search [Alt+S] N. Virginia maheshzadeitm

RDS > Databases > Add reader

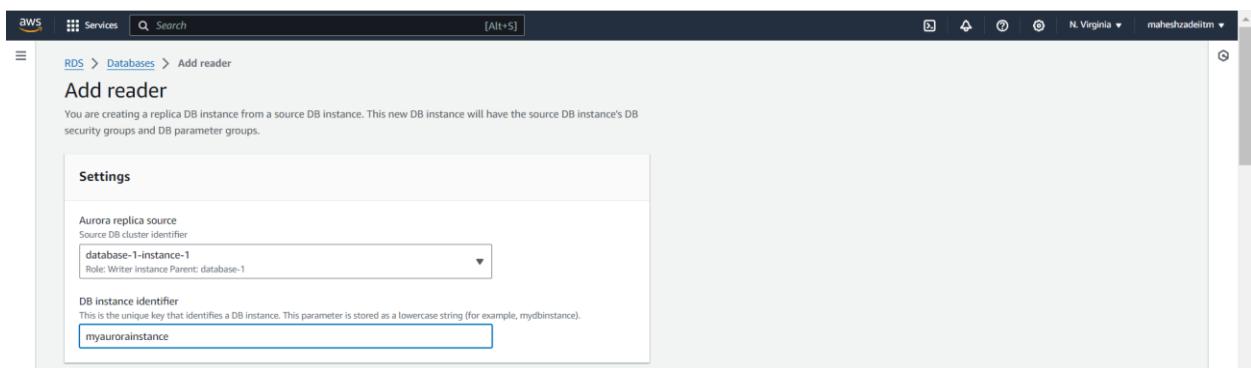
### Add reader

You are creating a replica DB instance from a source DB instance. This new DB instance will have the source DB instance's DB security groups and DB parameter groups.

**Settings**

Aurora replica source  
Source DB cluster identifier: database-1-instance-1  
Role: Writer Instance Parent: database-1

DB instance identifier  
This is the unique key that identifies a DB instance. This parameter is stored as a lowercase string (for example, mydbinstance).  
myaurorainstance

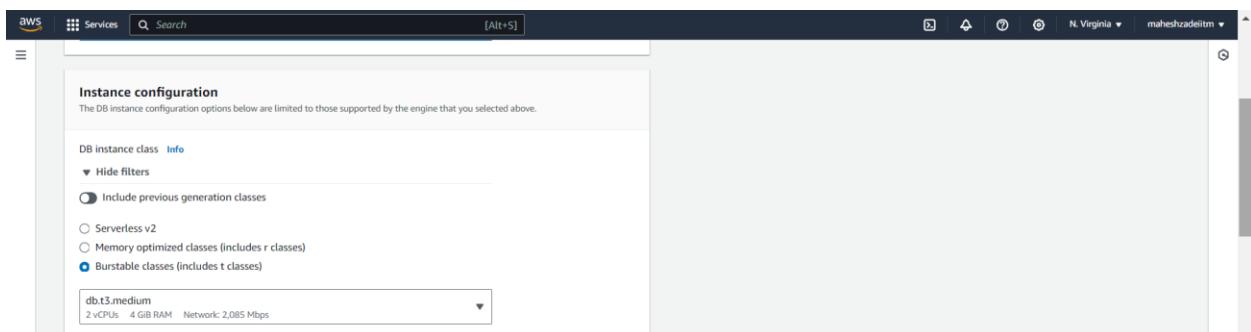


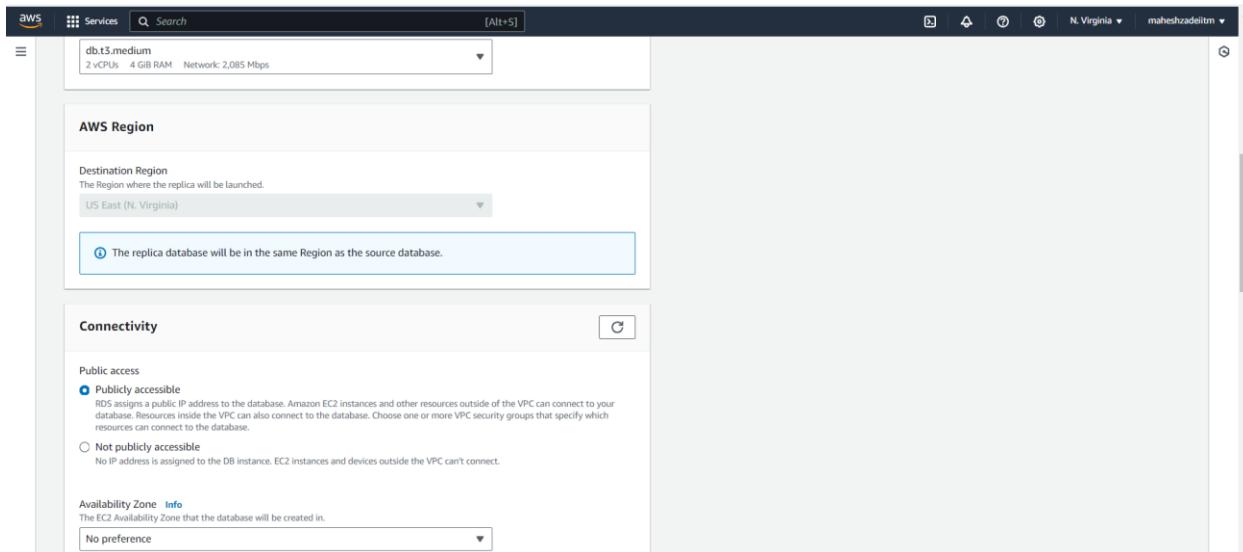
AWS Services Search [Alt+S] N. Virginia maheshzadeitm

Instance configuration

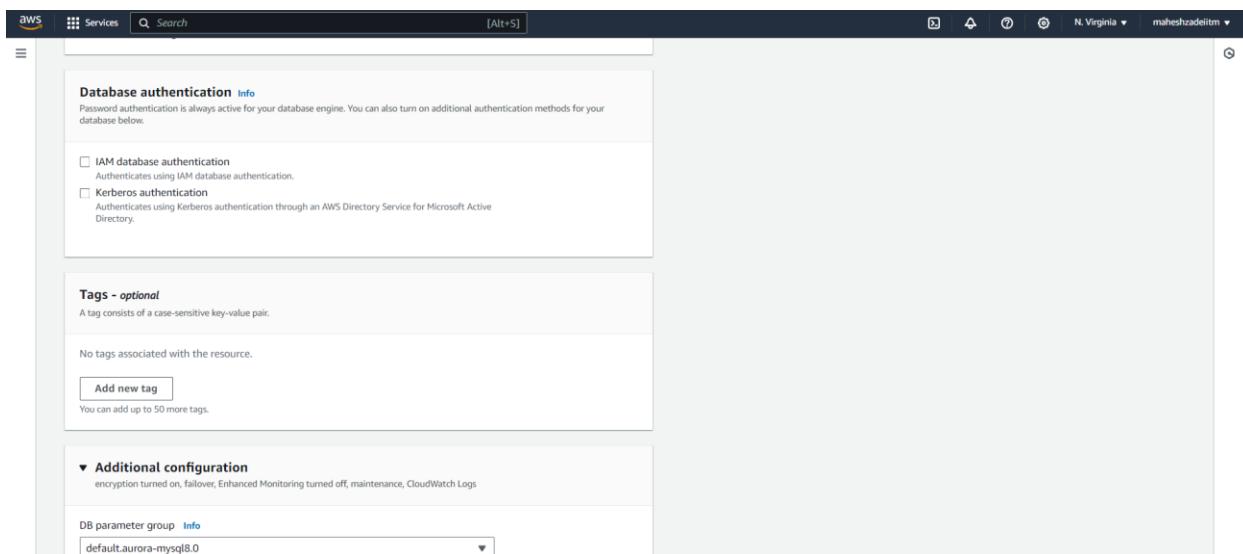
The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class: db.t3.medium  
Info Hide filters Include previous generation classes  
Serverless v2 Memory optimized classes (includes r classes)  
Burstable classes (includes t classes)





**Keep the remaining settings to defaults and click on Add Reader**



Screenshot of the AWS RDS console showing the configuration for creating a new database instance. The 'No preference' dropdown under 'Tunner priority' is selected.

**Encryption**

- Enable encryption
 

Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. [Info](#)
- AWS KMS key [Info](#)
- Amazon Resource Name (ARN)
 

Example: arn:aws:kms:<region>:<accountID>/key/<key-id>
- Account: 05826440832
- KMS key ID: ca2478b0-5065-4c28-ba6a-4f54735c4da6

**Maintenance**

- Auto minor version upgrade [Info](#)
- Enable auto minor version upgrade
 

Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Cancel Add reader

Read replica is getting created. Meanwhile, we go and create another read replica in a different AZ.

Choose cluster → Go to Actions -> Add reader

Screenshot of the AWS RDS console showing the 'Creating replica myaurorainstance in US East (N. Virginia)' status bar. The database might take a few minutes to launch.

**Databases**

RDS > Databases

**Databases (3)**

| DB identifier         | Status    | Role             | Engine       | Region & AZ | Size         | Recommendations | CPU |
|-----------------------|-----------|------------------|--------------|-------------|--------------|-----------------|-----|
| database-1            | Available | Regional cluster | Aurora MySQL | us-east-1   | 2 instances  | -               | -   |
| database-1-instance-1 | Available | Writer instance  | Aurora MySQL | us-east-1d  | db.t3.medium | 11.21%          | -   |
| myaurorainstance      | Creating  | Reader instance  | Aurora MySQL | us-east-1c  | db.t3.medium | -               | -   |

Creating replica myaurainstance in US East (N. Virginia)  
Your database might take a few minutes to launch.

RDS > Databases

Databases (3)

| DB identifier         | Status    | Role             | Engine       | Region & AZ |
|-----------------------|-----------|------------------|--------------|-------------|
| database-1            | Available | Regional cluster | Aurora MySQL | us-east-1   |
| database-1-instance-1 | Available | Writer instance  | Aurora MySQL | us-east-1d  |
| myaurainstance        | Creating  | Reader instance  | Aurora MySQL | us-east-1c  |

Actions ▾

- Group resources
- Modify
- Actions ▾
- Restore from S3
- Create database
- Stop temporarily
- Delete
- Set up EC2 connection
- Set up Lambda connection
- Add AWS Region
- Add reader
- Create cross-Region read replica
- Create Blue/Green Deployment - new
- Create clone
- Promote
- Take snapshot
- Restore to point in time
- Backtrack
- Export to Amazon S3 - new
- Add replica auto scaling
- Create zero-ETL integration
- Create RDS Proxy
- Create ElastiCache cluster - new

CPU 11.21%

Creating replica myaurainstance in US East (N. Virginia)  
Your database might take a few minutes to launch.

RDS > Databases > Add reader

### Add reader

You are creating a replica DB instance from a source DB instance. This new DB instance will have the source DB instance's DB security groups and DB parameter groups.

Settings

Aurora replica source  
Source DB cluster identifier  
database-1-instance-1  
Role: Writer instance Parent: database-1

DB instance identifier  
This is the unique key that identifies a DB instance. This parameter is stored as a lowercase string (for example, mydbinstance).  
myauradbinsatnce2

Creating replica myaurainstance in US East (N. Virginia)  
Your database might take a few minutes to launch.

RDS > Databases > Create

### Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

▼ Hide filters

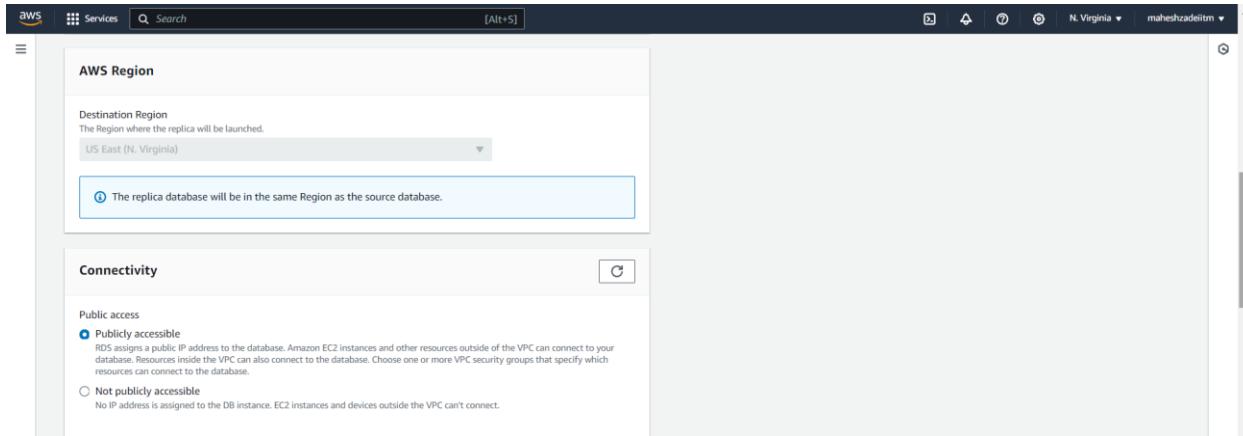
Include previous generation classes

Serverless v2

Memory optimized classes (includes r classes)

Burstable classes (includes t classes)

db.t3.medium  
2 vCPUs 4 GiB RAM Network: 2,085 Mbps



## AZ change to us-east-1a

