Guided Lab: Integrating Amazon RDS to Amazon EC2 Instance

Description

Welcome to the hands-on lab for integrating EC2 (Elastic Compute Cloud) and RDS (Relational Database Service) in AWS! This lab will guide you through the process of setting up an EC2 instance and an RDS database, and connecting the two. By the end of this lab, you'll have a basic understanding of how these core AWS services work together to create a scalable and secure environment for your applications.

What is EC2?

Amazon EC2 is a web service that provides resizable compute capacity in the cloud. It allows you to launch virtual servers (instances) in minutes, enabling you to scale up or down as your computing needs change. EC2 is commonly used for hosting websites, running applications, performing data processing, and more.

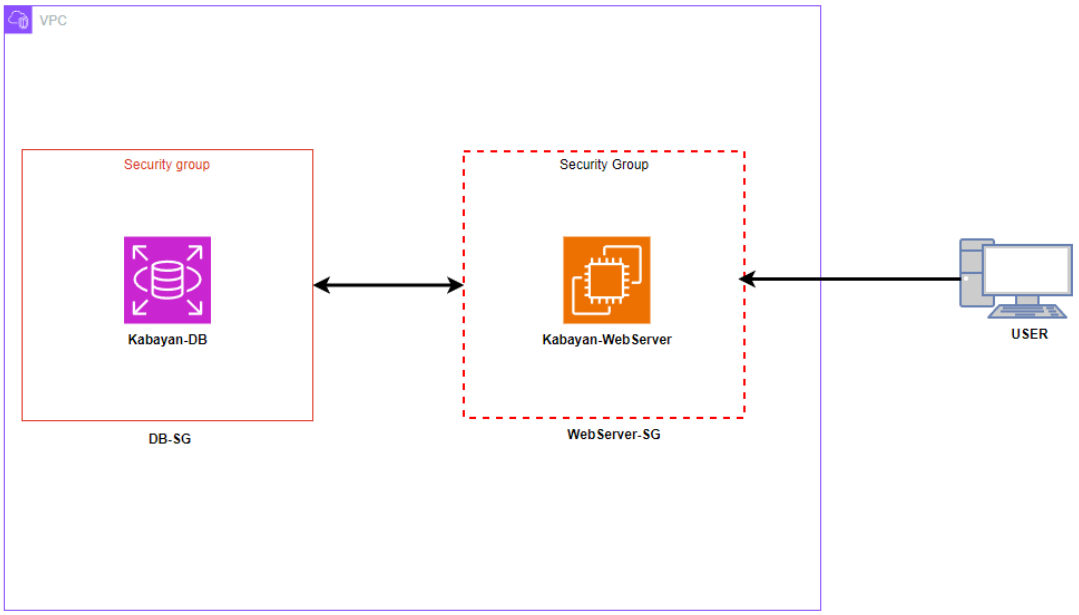
What is RDS?

Amazon RDS is a managed relational database service that makes it easy to set up, operate, and scale a relational database in the cloud. RDS supports several popular database engines, including MySQL, PostgreSQL, MariaDB, Oracle, and Microsoft SQL Server. It automates time-consuming tasks such as hardware provisioning, database setup, patching, and backups.

Why Integrate EC2 and RDS?

Integrating EC2 and RDS is a common architecture for many web applications. By separating the compute (EC2) and database (RDS) layers, you can achieve better performance, scalability, and security. Here are some benefits of this integration:

- **Performance:** Offload database management to RDS, allowing your EC2 instances to focus on application logic.
- **Scalability:** Scale your EC2 instances and RDS database independently to meet changing demands.
- **Security:** Isolate your database from the internet by placing it in a private subnet, accessible only from your EC2 instances.
- **Simplicity:** Simplify database management with RDS's automated backups, updates, and monitoring.



Prerequisites

This lab assumes you have basic knowledge of creating an Amazon RDS database, an Amazon EC2 instance (Linux), and basic AWS networks.
If you find any gaps in your knowledge, consider taking the following lab:

- Creating an Amazon RDS database
- Creating an Amazon EC2 instance (Linux)
- Security Group VS Network Access Control List

Objectives

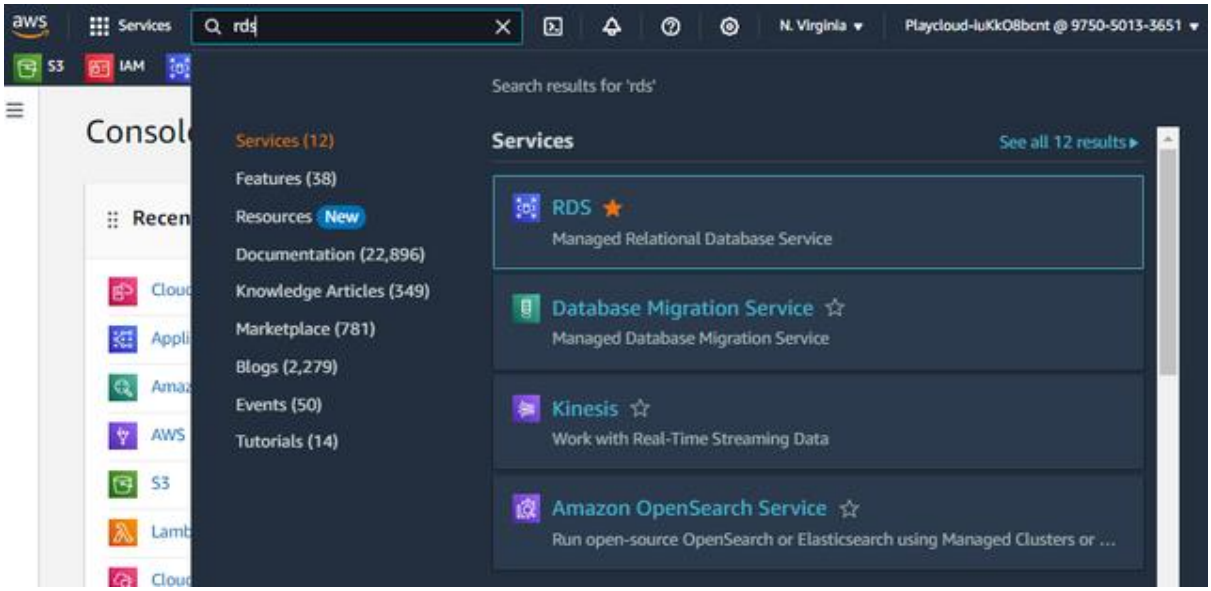
In this lab, you will:

- Learn how to set up an EC2 instance and RDS database and connect the EC2 instance to the RDS database.

Subscribe to access AWS PlayCloud Labs

Lab Steps

Set Up an Amazon RDS Instance


aws
Services
rds
N. Virginia
Playcloud-iuKkO8bcnt @ 9750-5013-3651
S3
IAM
Search results for 'rds'
Services (12)
Features (38)
Resources New
Documentation (22,896)
Knowledge Articles (349)
Marketplace (781)
Blogs (2,279)
Events (50)
Tutorials (14)
Services
See all 12 results
RDS
Managed Relational Database Service
Database Migration Service
Managed Database Migration Service
Kinesis
Work with Real-Time Streaming Data
Amazon OpenSearch Service
Run open-source OpenSearch or Elasticsearch using Managed Clusters or ...

-
-
  -
  -

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
- ( ) PostgreSQL
- ( ) Oracle
- ( ) Microsoft SQL Server
- ( ) IBM Db2

Edition

(•) MySQL Community

Engine version Info

View the engine versions that support the following database features.

▼ Hide filters

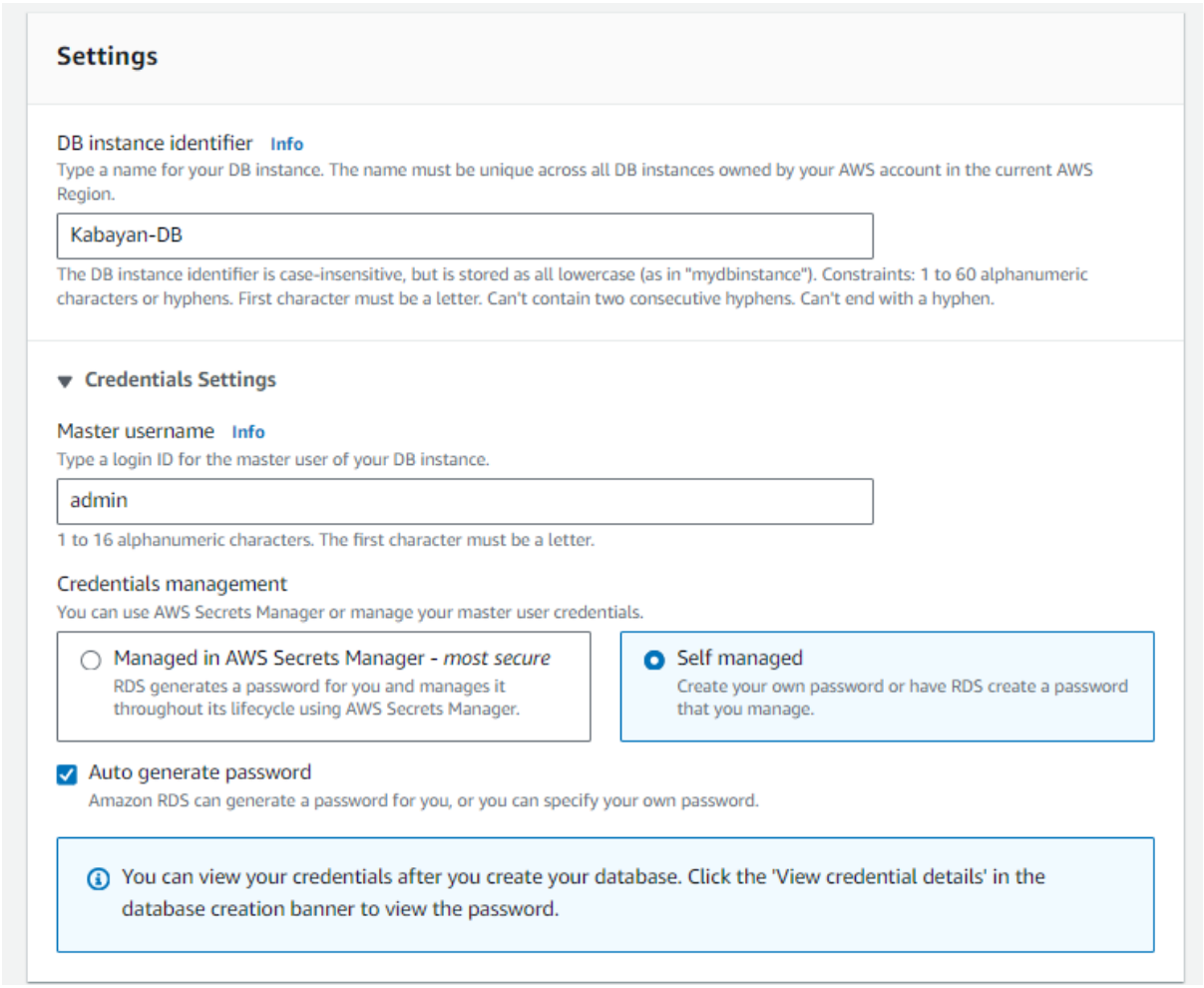
- [ ] Show versions that support the Multi-AZ DB cluster Info
  Create a A Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.
- [ ] Show versions that support the Amazon RDS Optimized Writes Info
  Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine Version

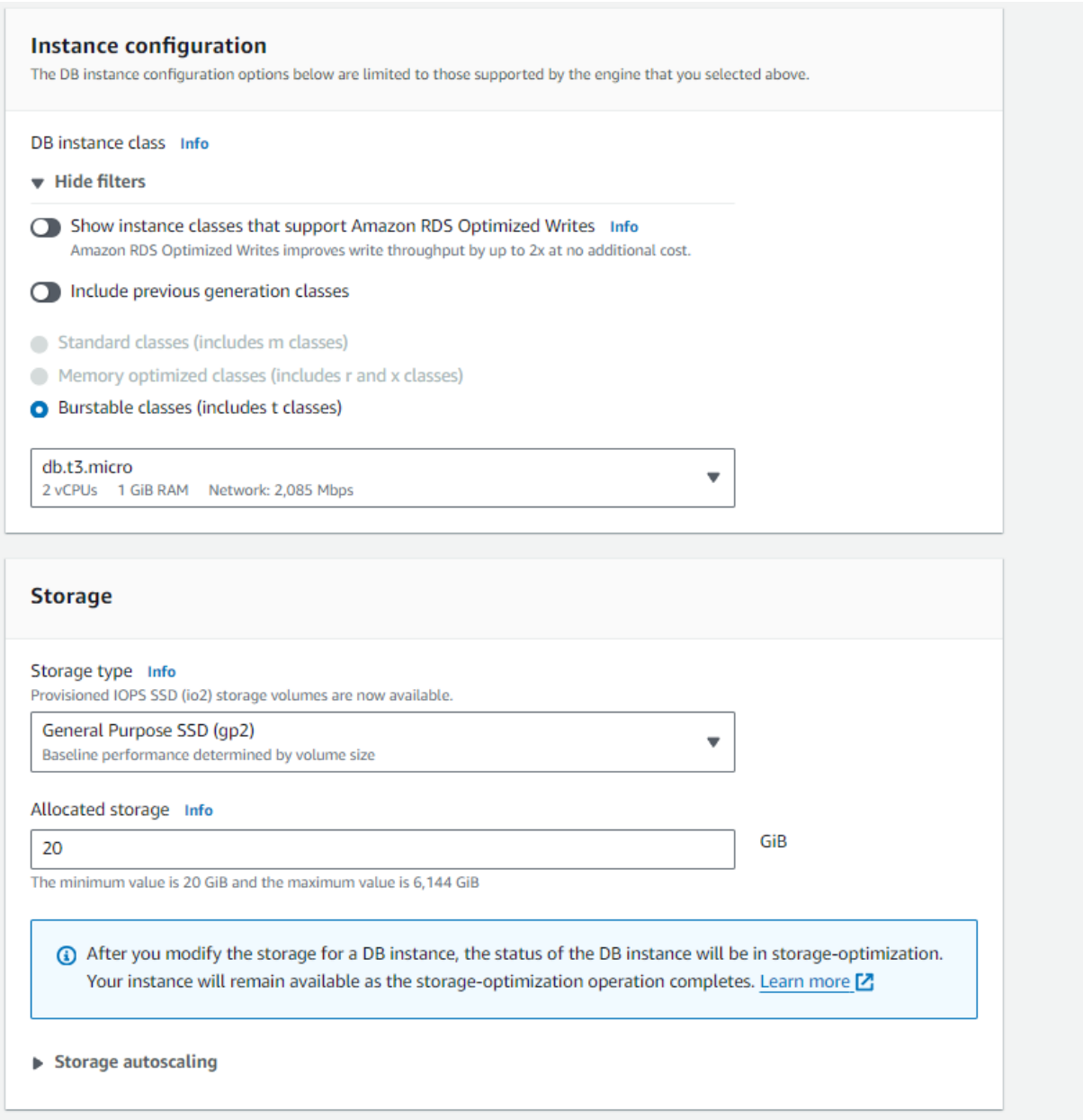
MySQL 8.0.35 ▼

- [ ] Enable RDS Extended Support Info
  Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the RDS for MySQL documentation.

-
-
    -
    -
    -
    -


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.
Kabayan-DB
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.
You can view your credentials after you create your database. Click the 'View credential details' in the database creation banner to view the password.

-
    -
-
    -
    -


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
Show instance classes that support Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.
Include previous generation classes
Standard classes (includes m classes)
Memory optimized classes (includes r and x classes)
Burstable classes (includes t classes)
db.t3.micro
2 vCPUs 1 GiB RAM Network: 2,085 Mbps
Storage
Storage type Info
Provisioned IOPS SSD (io2) storage volumes are now available.
General Purpose SSD (gp2)
Baseline performance determined by volume size
Allocated storage Info
20
GiB
The minimum value is 20 GiB and the maximum value is 6,144 GiB
After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization.
Your instance will remain available as the storage-optimization operation completes. Learn more
Storage autoscaling

-
    -

-
    -
    -
    -
    -
        -

-
  -

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.

Virtual private cloud (VPC) Info

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

vpc-092f3b7dd98fe3bee
3 Subnets, 3 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

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 instance can use in the VPC that you selected.

Create new DB Subnet Group

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

New VPC security group name

DB-SG

Availability Zone Info

us-east-1a

RDS Proxy

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

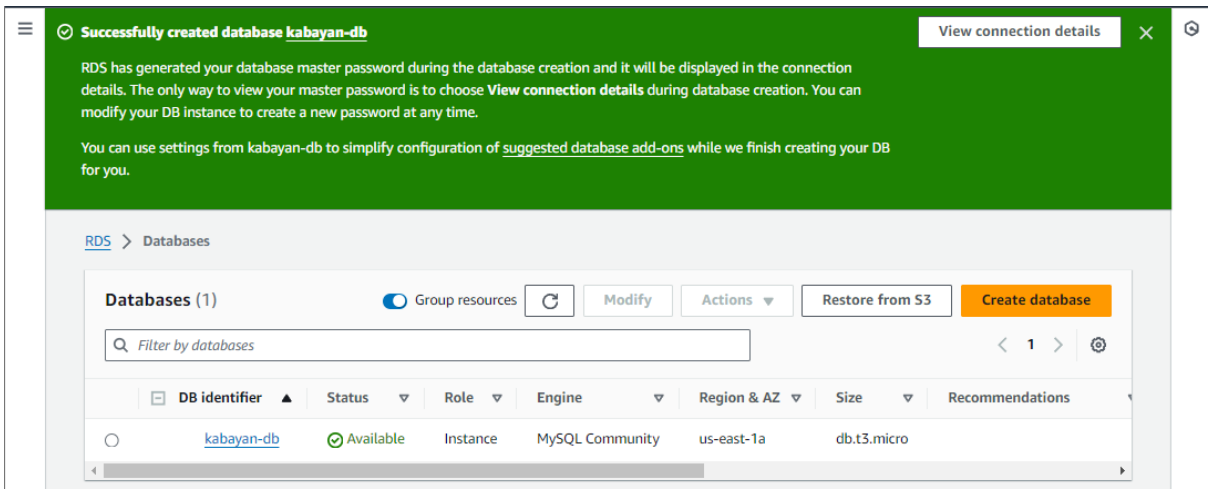
[ ] Create an RDS Proxy Info
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see Amazon RDS Proxy pricing.

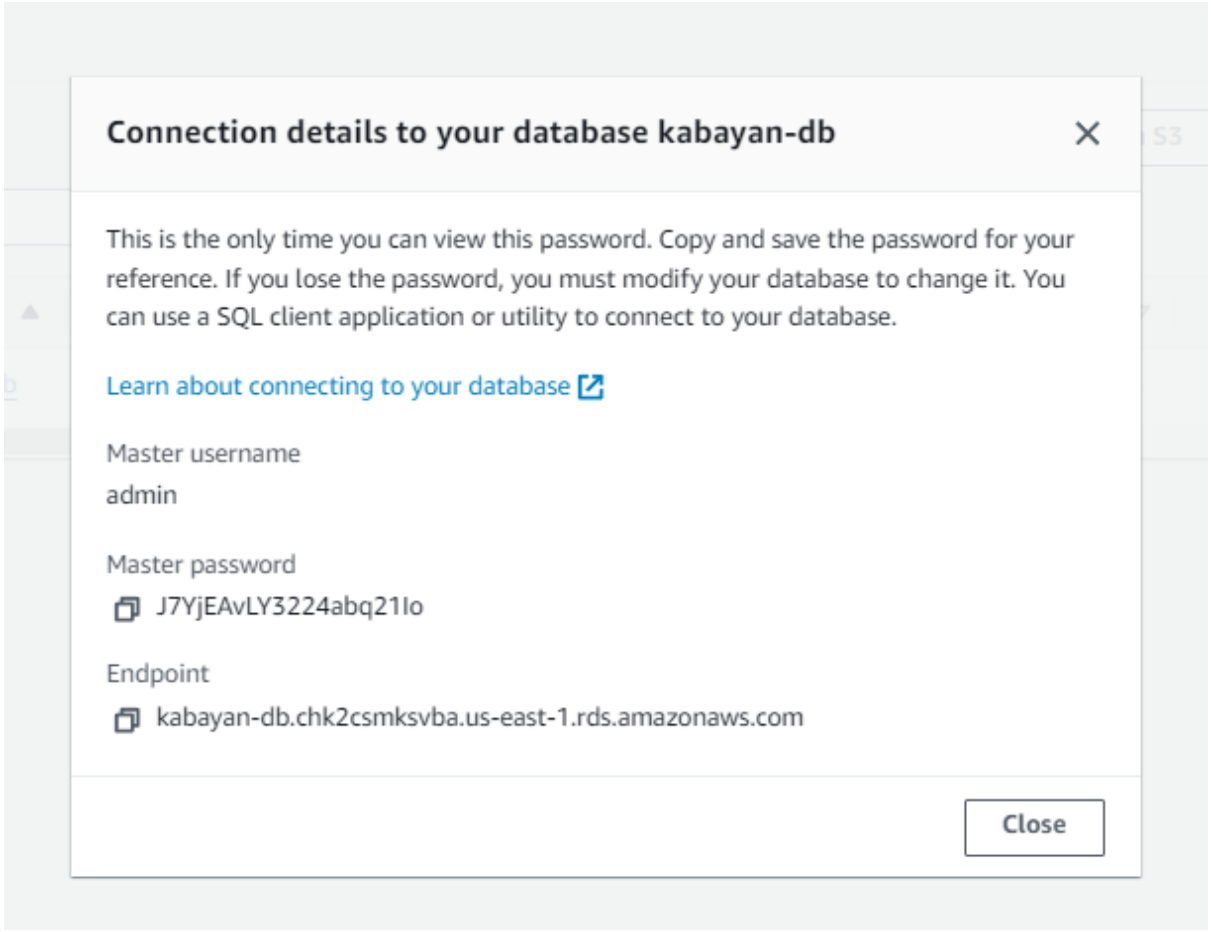
Certificate authority - *optional* Info

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

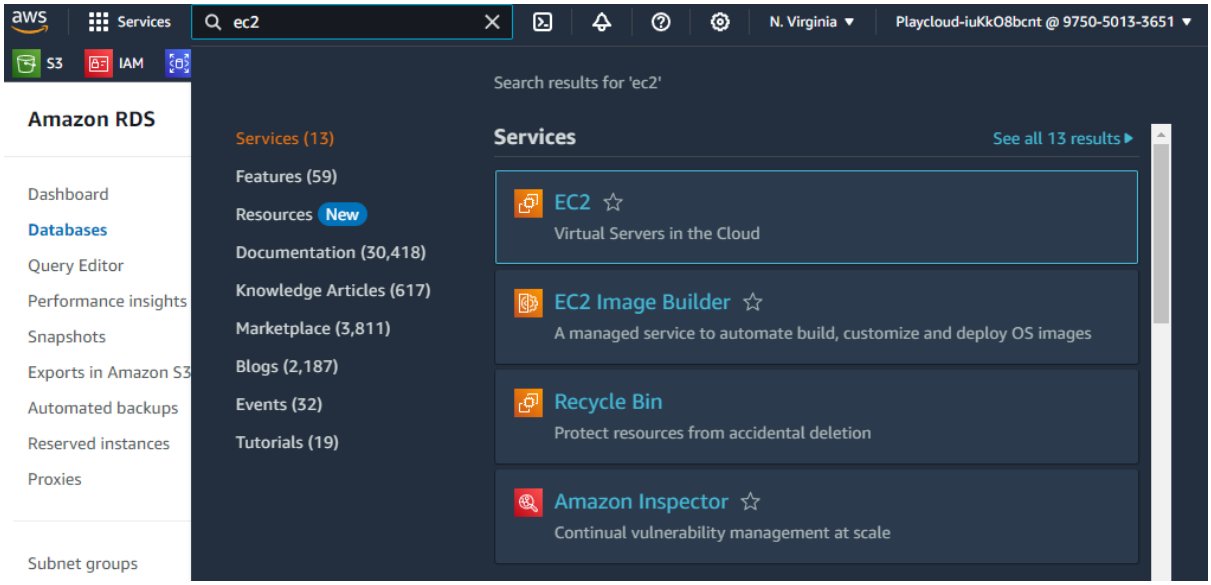
-





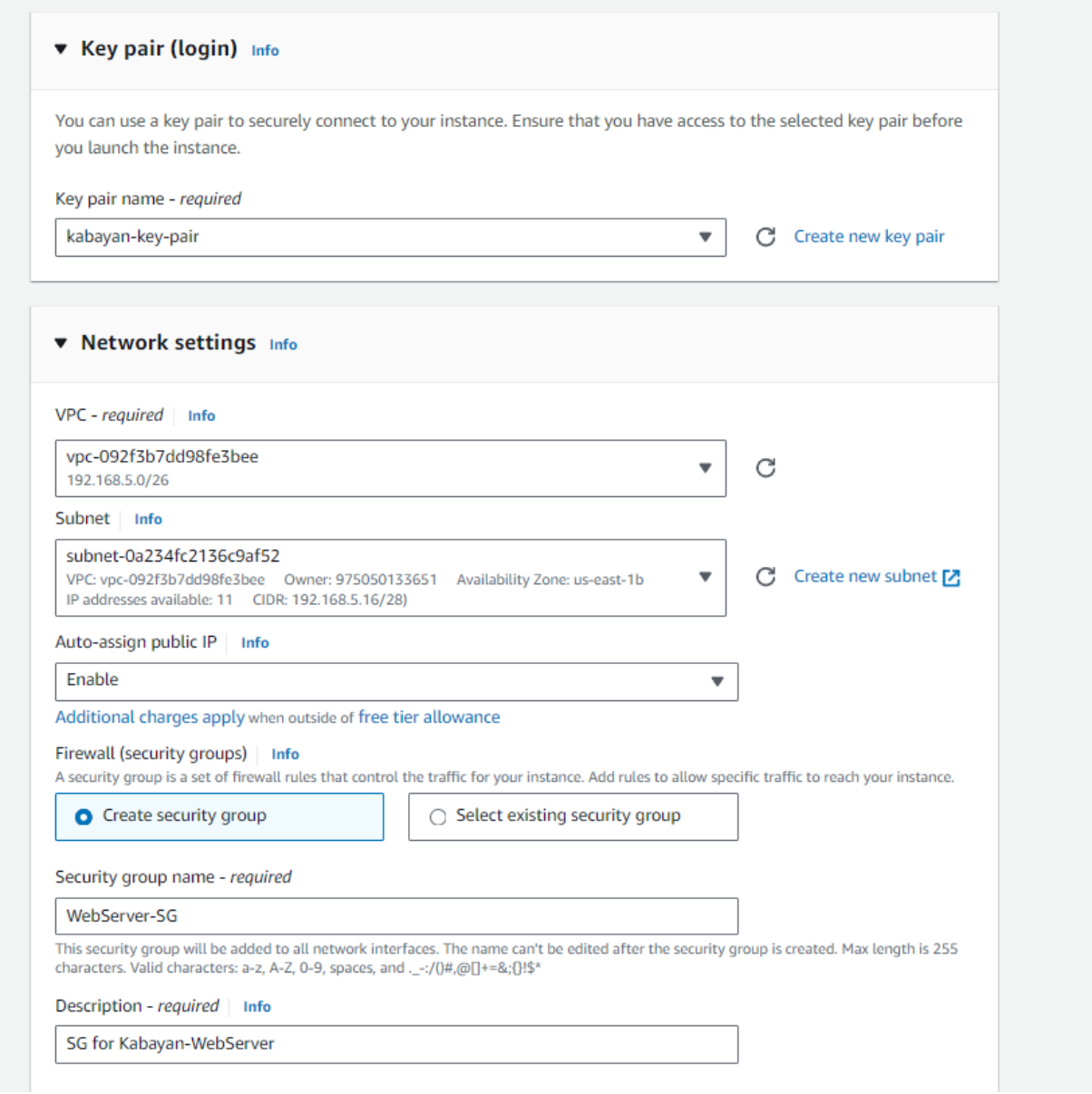
Set Up an Amazon EC2 Instance

1. **Navigate to EC2:**



2.**Launch an EC2 Instance with the following configuration:**

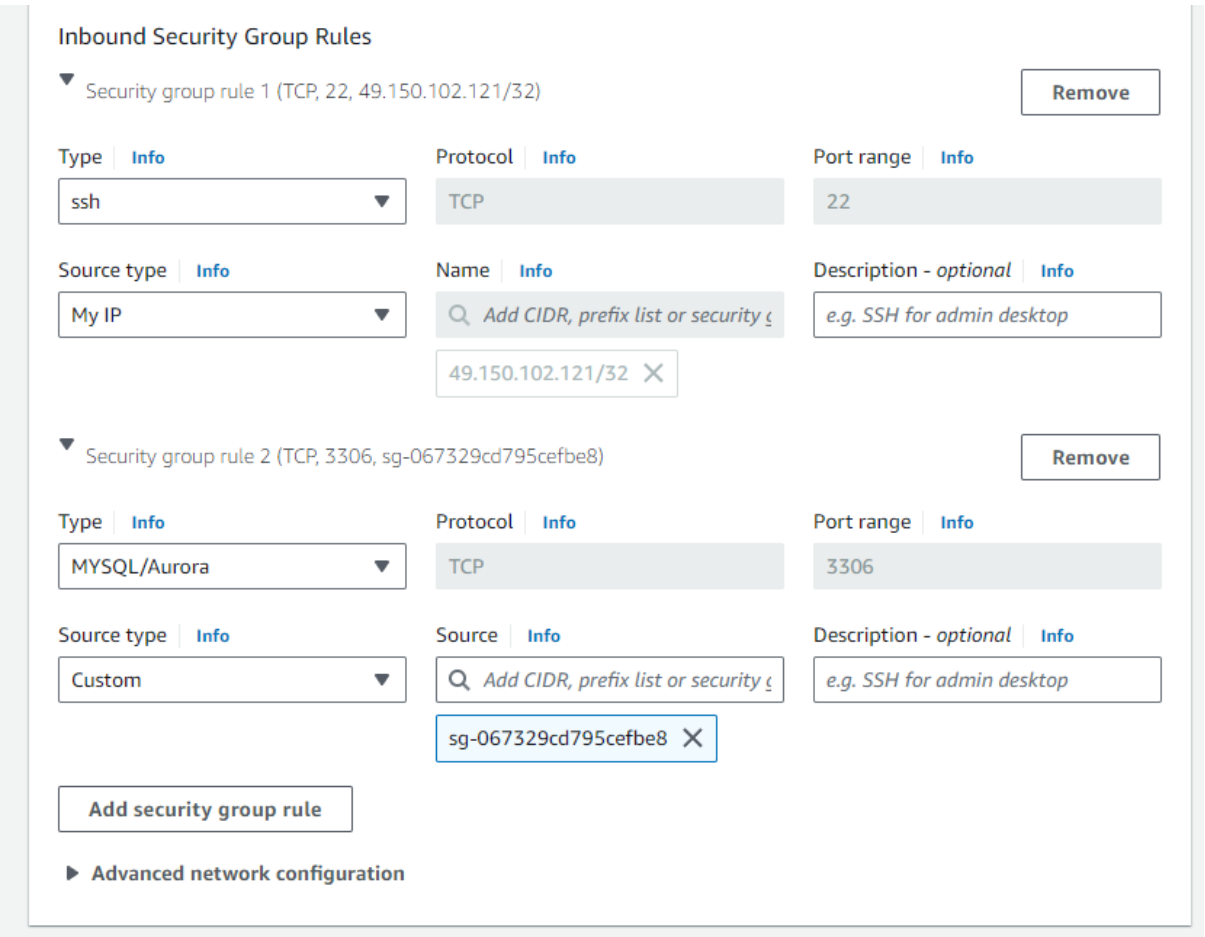
- **Name:** Kabayan-WebServer
- **AMI:** Amazon Linux
- **Instance type:** t2.micro
- **Key pair:** (Please **create a new one.**)
    - **Key pair name:** kabayan-key-pair
    - **Key pair type:** RSA
    - **Private key file format:** .pem
- **Network settings: (**Click **"Edit")**
    - Leave **VPC** and **Subnet** with the default
    - **Auto-assign public IP:** Select Enable
    - Firewall (security groups): tick on the **Create security group**
        - **Security group name – *required*:** WebServer-SG
        - **Description – *required:*** SG for Kabayan-WebServer



- **Inbound Security Group Rules: (**add the following security group rule)
    - **Type:** ssh
        - **Source type:** My IP
    - **Type:** MYSQL/Aurora
        - **Source:** Select DB-SG that you created in the creation of your RDS instance earlier

---

NOTE: This Inbound rules allows the Security Group for our Kabayan-WebServer(EC2 Instance) to connect with the Kabayan-DB's(RDS Instance) Security Group DB-SG. This is very important rule to let the communication between these Security Group secure and not let anyone connect to our RDS Except for our instance. We will also need to modify the Security Group Rule for the DB-SG later in the succeeding steps.

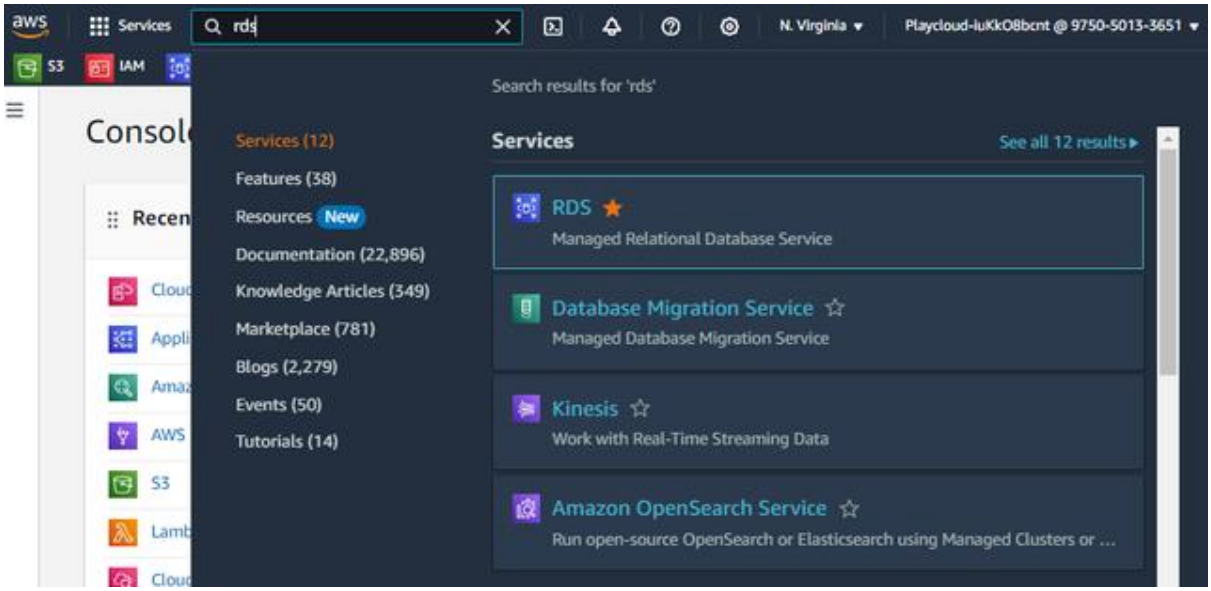


-
    - Click **Launch instance**

Lab Steps

Set Up an Amazon RDS Instance

1. Navigate to RDS:



2. Create a Database with the following configuration:

- **Choose a database creation method:** Standard create
- **Engine options:**
    - **Engine type:** MySQL
    - **Engine Version:** Leave it as default

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
- ( ) PostgreSQL
- ( ) Oracle
- ( ) Microsoft SQL Server
- ( ) IBM Db2

Edition

(•) MySQL Community

Engine version Info

View the engine versions that support the following database features.

▼ Hide filters

- [ ] Show versions that support the Multi-AZ DB cluster Info
  Create a A Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.
- [ ] Show versions that support the Amazon RDS Optimized Writes Info
  Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine Version

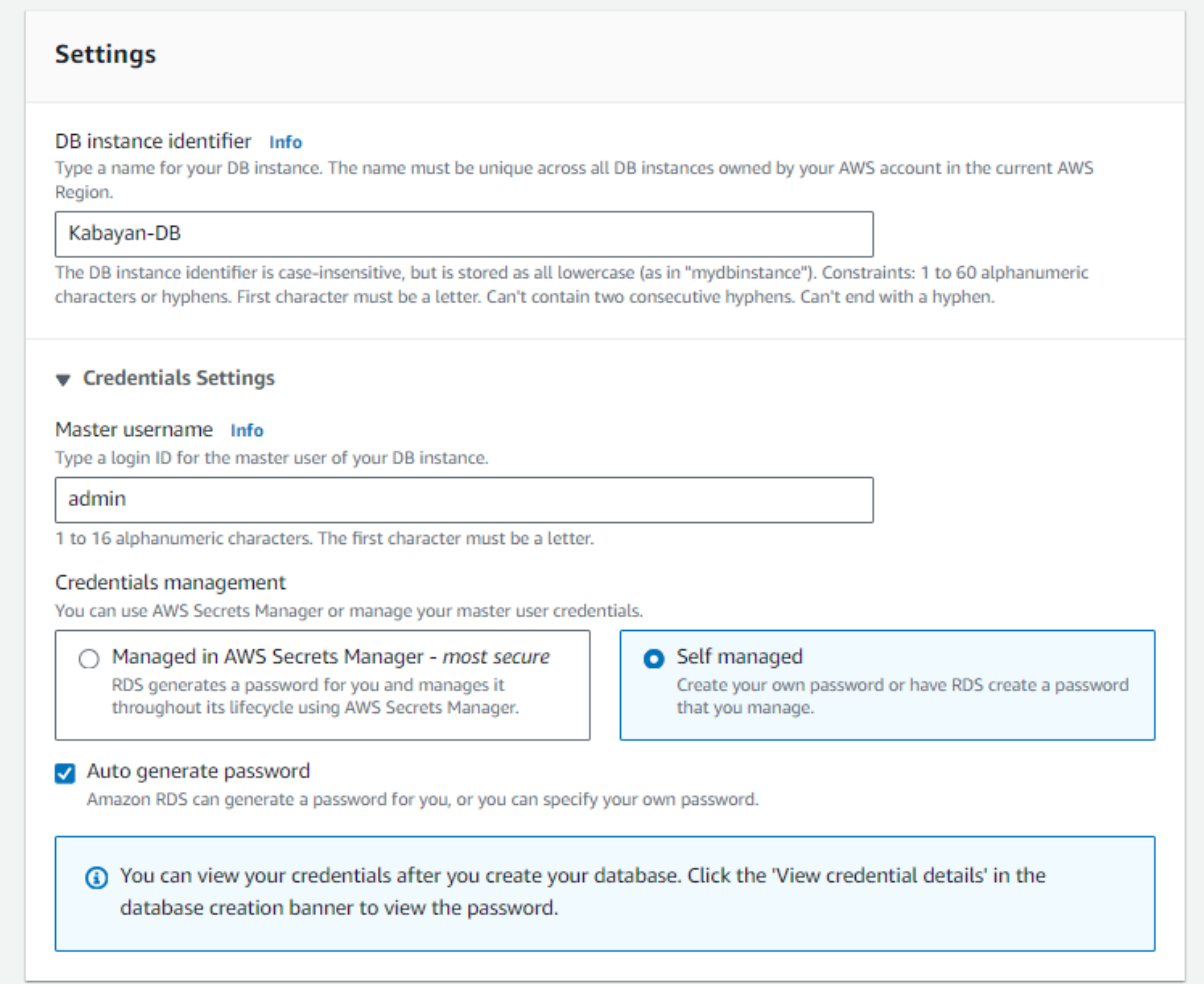
MySQL 8.0.35 ▼

- [ ] Enable RDS Extended Support Info
  Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the RDS for MySQL documentation.

- **Templates:** Free tier
- **Settings:**
    - **DB instance identifier:** Kabayan-DB
    - **Master username:** admin (you can change this as you desire)
    - **Credentials management:** Self managed
    - For the simplicity of this lab we will tick the checkbox of **Auto generate password**.

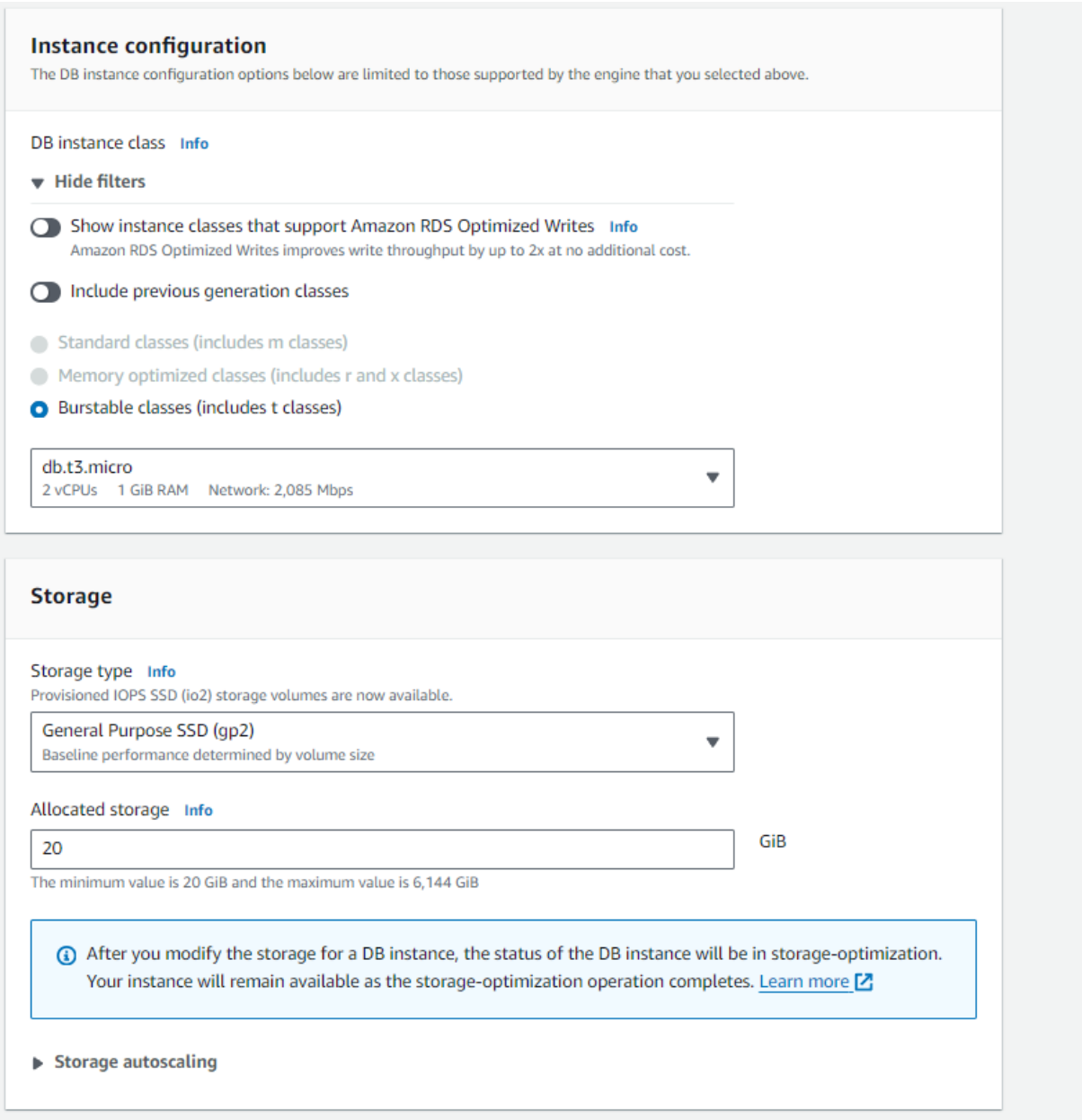
But take note that in PRODUCTION, you need to create a strong password for better security of your Database

---



- **Instance configuration**:
    - **DB instance size**:, db.t3.micro
- **Storage:**
    - **Storage type:** General Purpose SSD (gp2)

- **Allocated storage:** 20



- **Connectivity:**
  - **Compute resource:** Don't connect to an EC2 compute resource

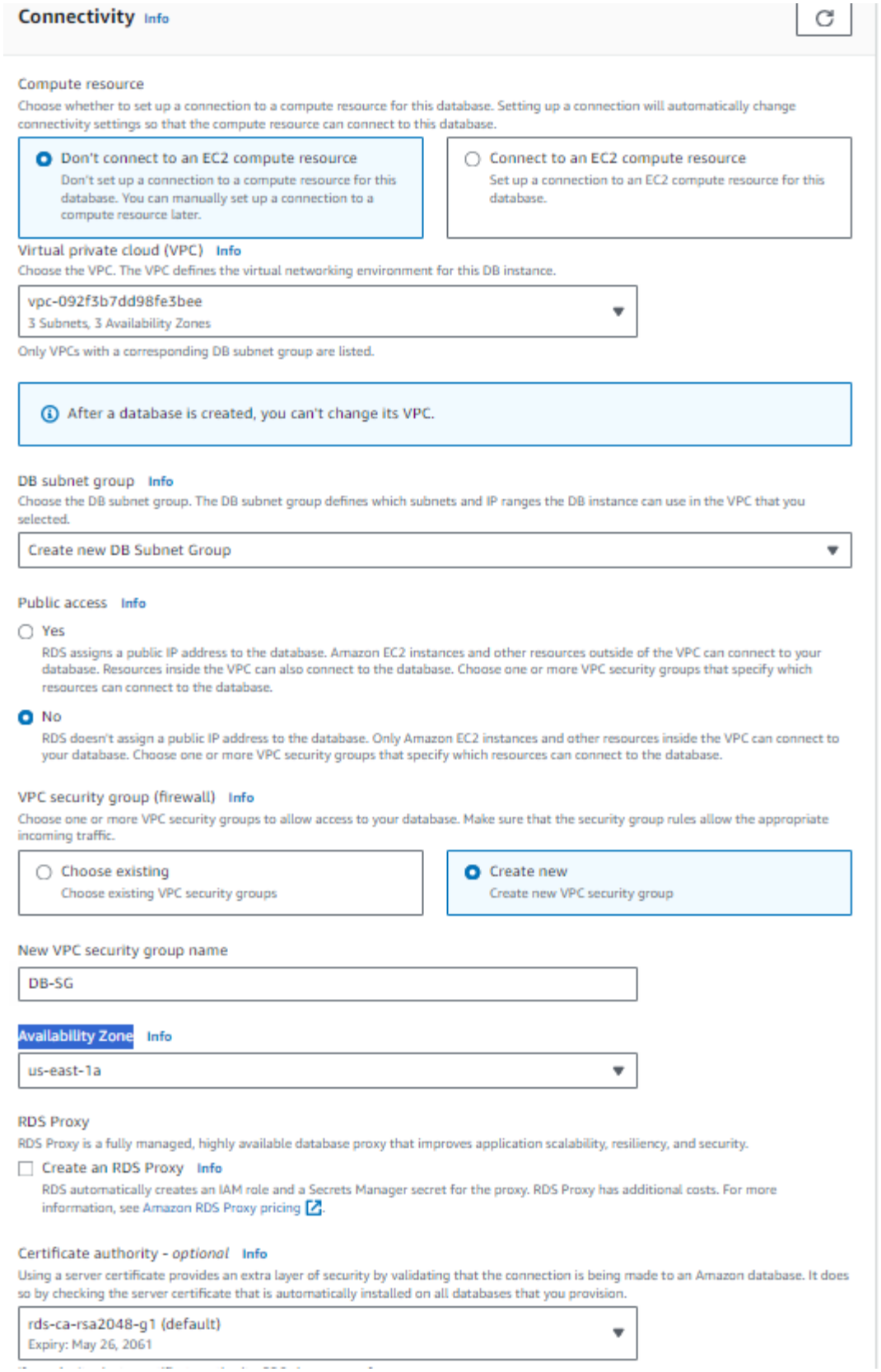
---

Note that in this lab we will do this manually, so lets select Don't connect to an EC2 compute resource

---

-
  - **Virtual private cloud (VPC):** leave it as the default
  - **DB subnet group:** Create new DB Subnet Group
  - **Public access:** No

- **VPC security group (firewall):** Select Create new
    - *New VPC security group name:* DB-SG
    - *Availability Zone:* select whichever you prefer *(e.g.* us-east-1a)
- Leave the rest as default

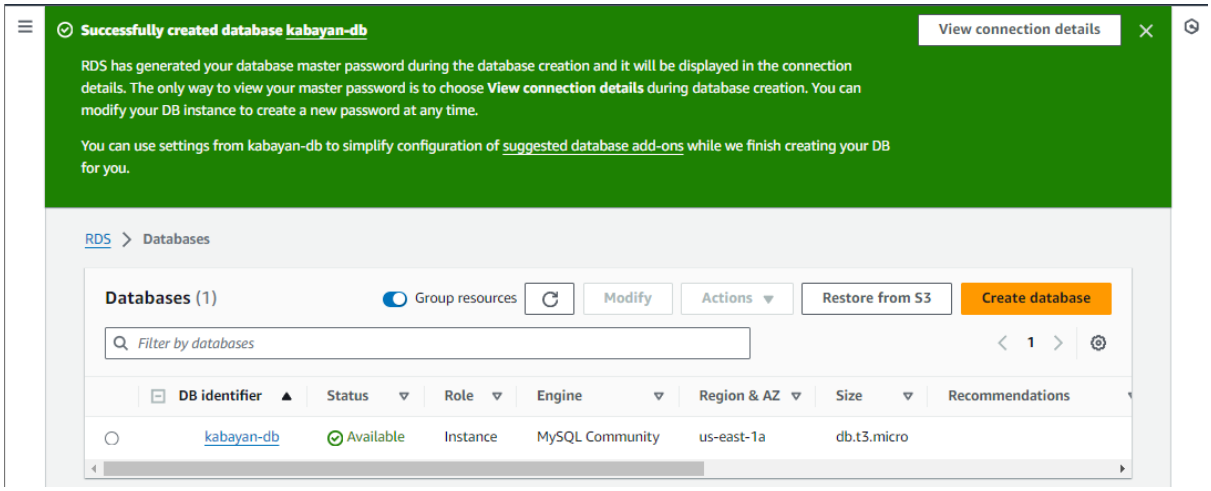


-
    - Click **Create database**

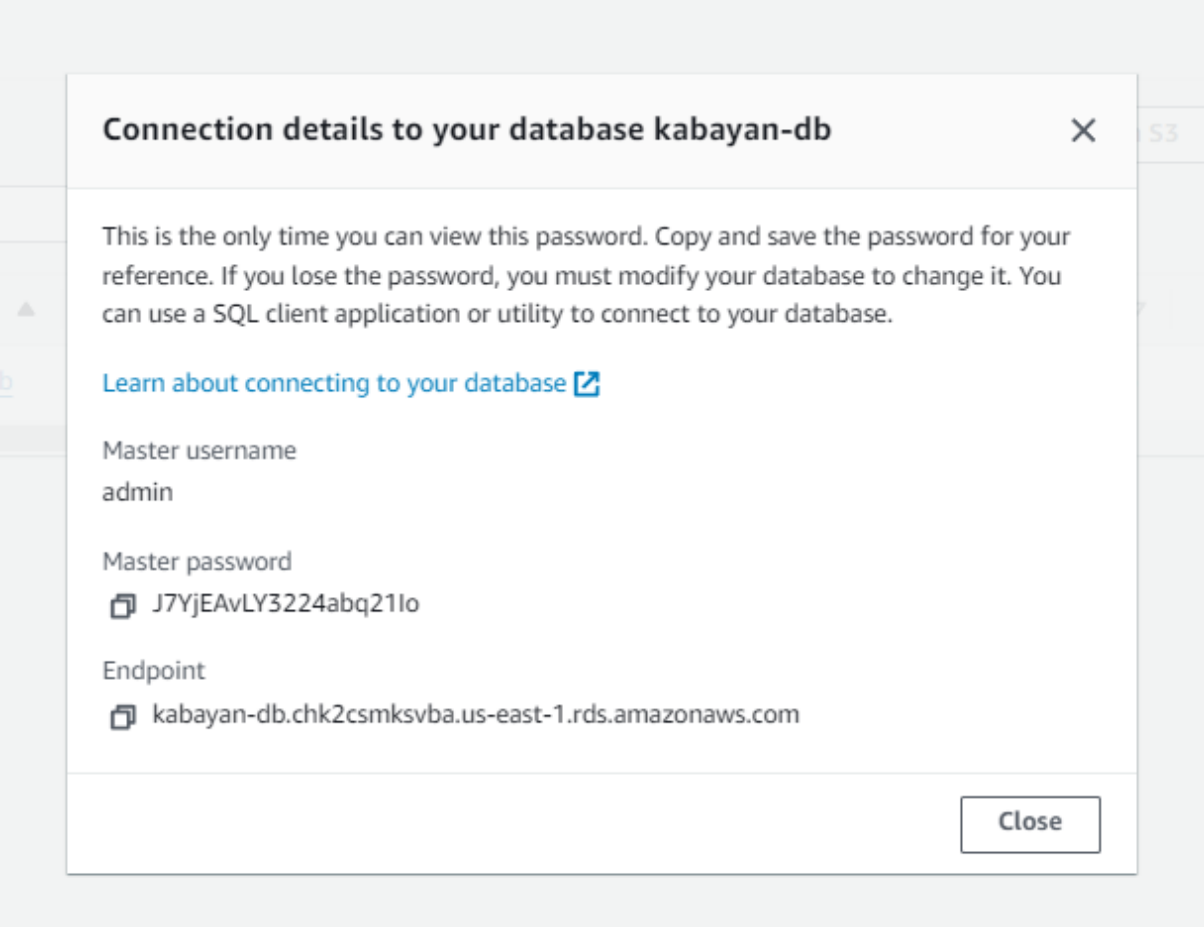
Take note that this will take time, around 10-15 minutes to finished.

---

3. **Wait for the RDS Instance to Be Available:**

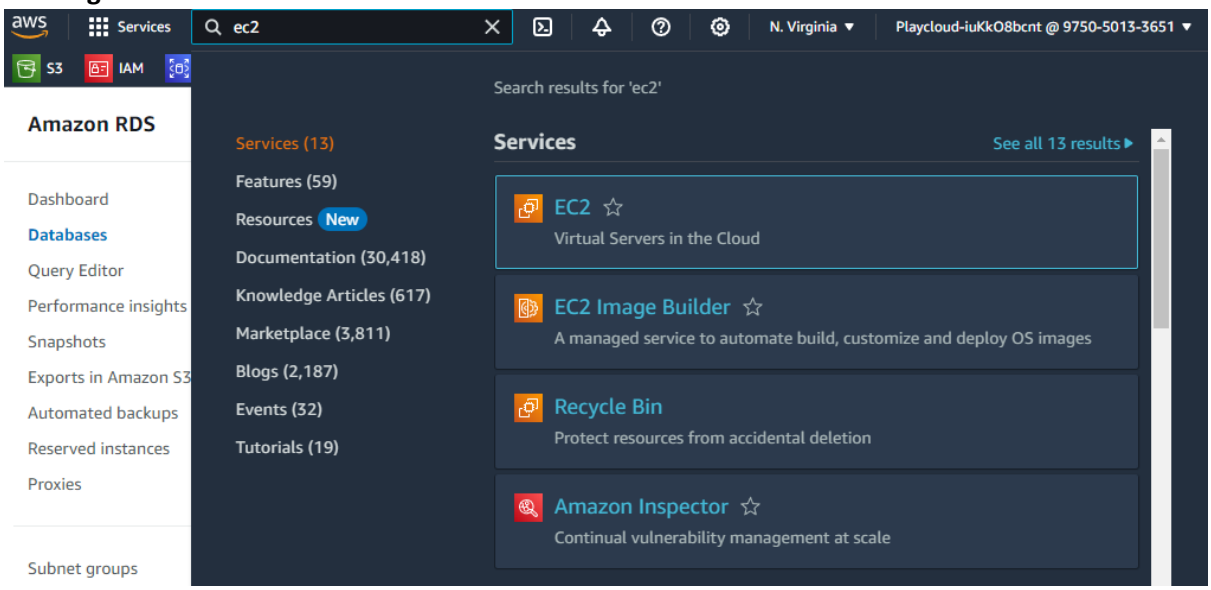


4. Click **View connection details** and take note of the details



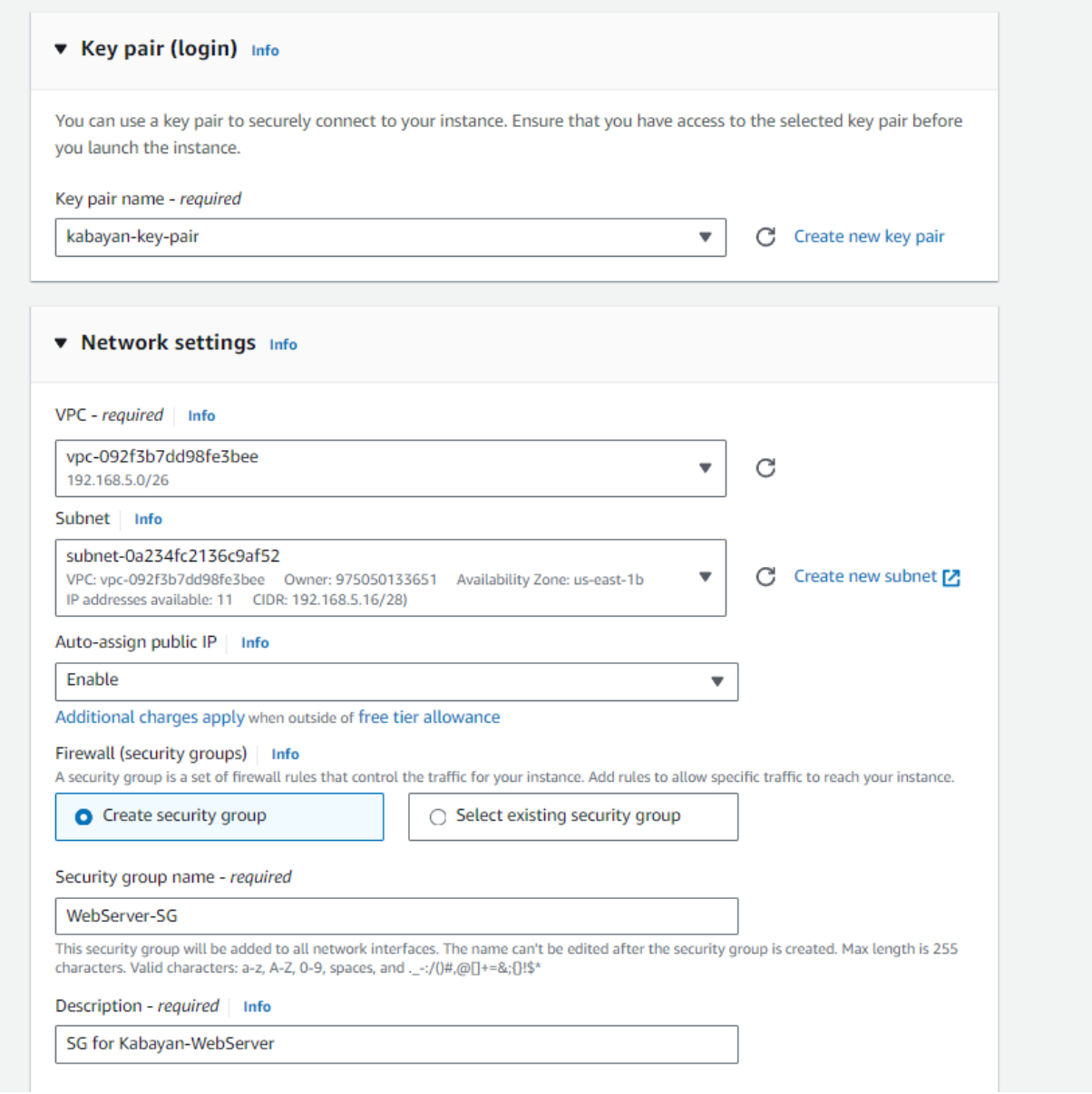
Set Up an Amazon EC2 Instance

1. **Navigate to EC2:**



2.**Launch an EC2 Instance with the following configuration:**

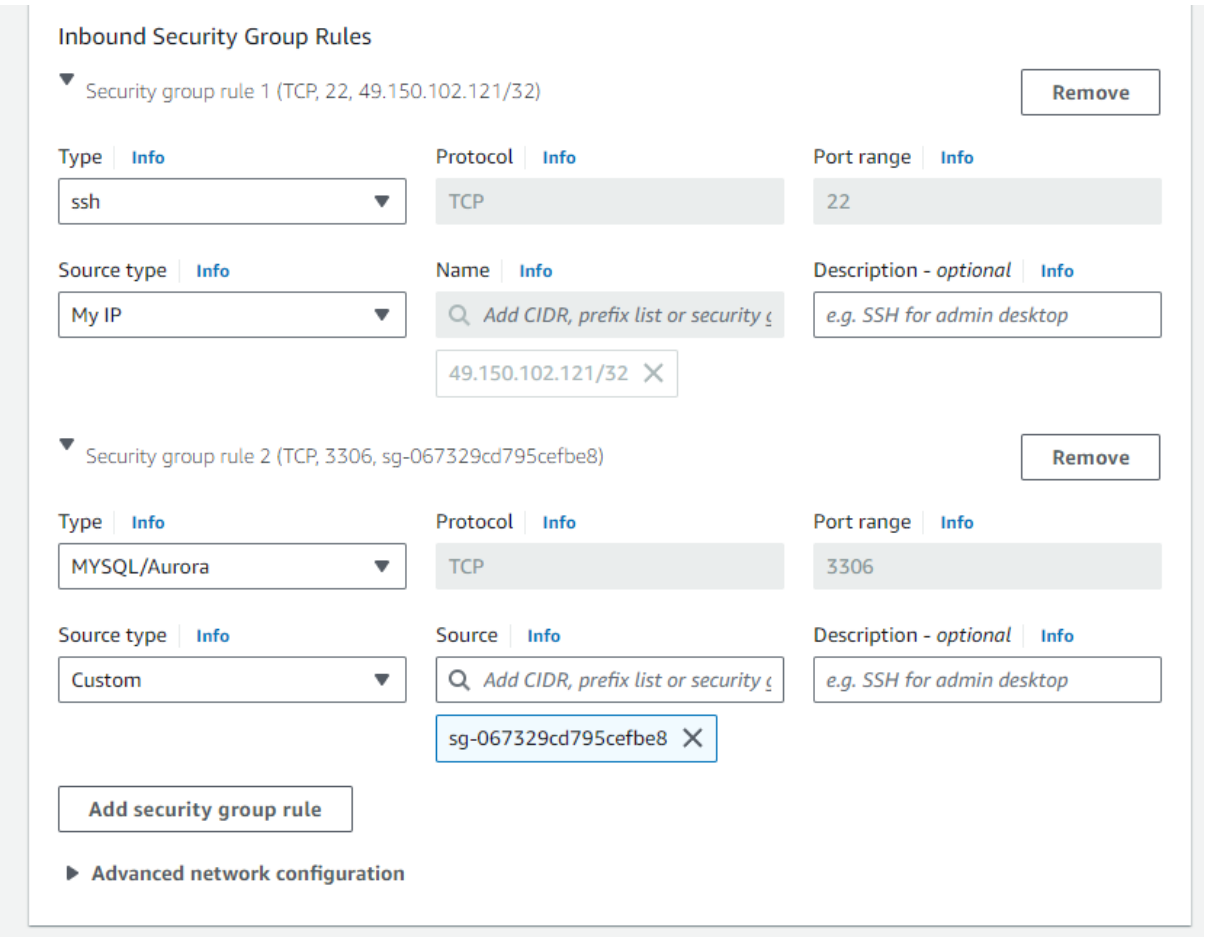
- **Name:** Kabayan-WebServer
- **AMI:** Amazon Linux
- **Instance type:** t2.micro
- **Key pair:** (Please **create a new one.**)
    - **Key pair name:** kabayan-key-pair
    - **Key pair type:** RSA
    - **Private key file format:** .pem
- **Network settings: (**Click **"Edit")**
    - Leave **VPC** and **Subnet** with the default
    - **Auto-assign public IP:** Select Enable
    - Firewall (security groups): tick on the **Create security group**
        - **Security group name – *required*:** WebServer-SG
        - **Description – *required:*** SG for Kabayan-WebServer



- **Inbound Security Group Rules: (**add the following security group rule)
    - **Type:** ssh
        - **Source type:** My IP
    - **Type:** MYSQL/Aurora
        - **Source:** Select DB-SG that you created in the creation of your RDS instance earlier

---

NOTE: This Inbound rules allows the Security Group for our Kabayan-WebServer(EC2 Instance) to connect with the Kabayan-DB's(RDS Instance) Security Group DB-SG. This is very important rule to let the communication between these Security Group secure and not let anyone connect to our RDS Except for our instance. We will also need to modify the Security Group Rule for the DB-SG later in the succeeding steps.



-
    - Click **Launch instance**

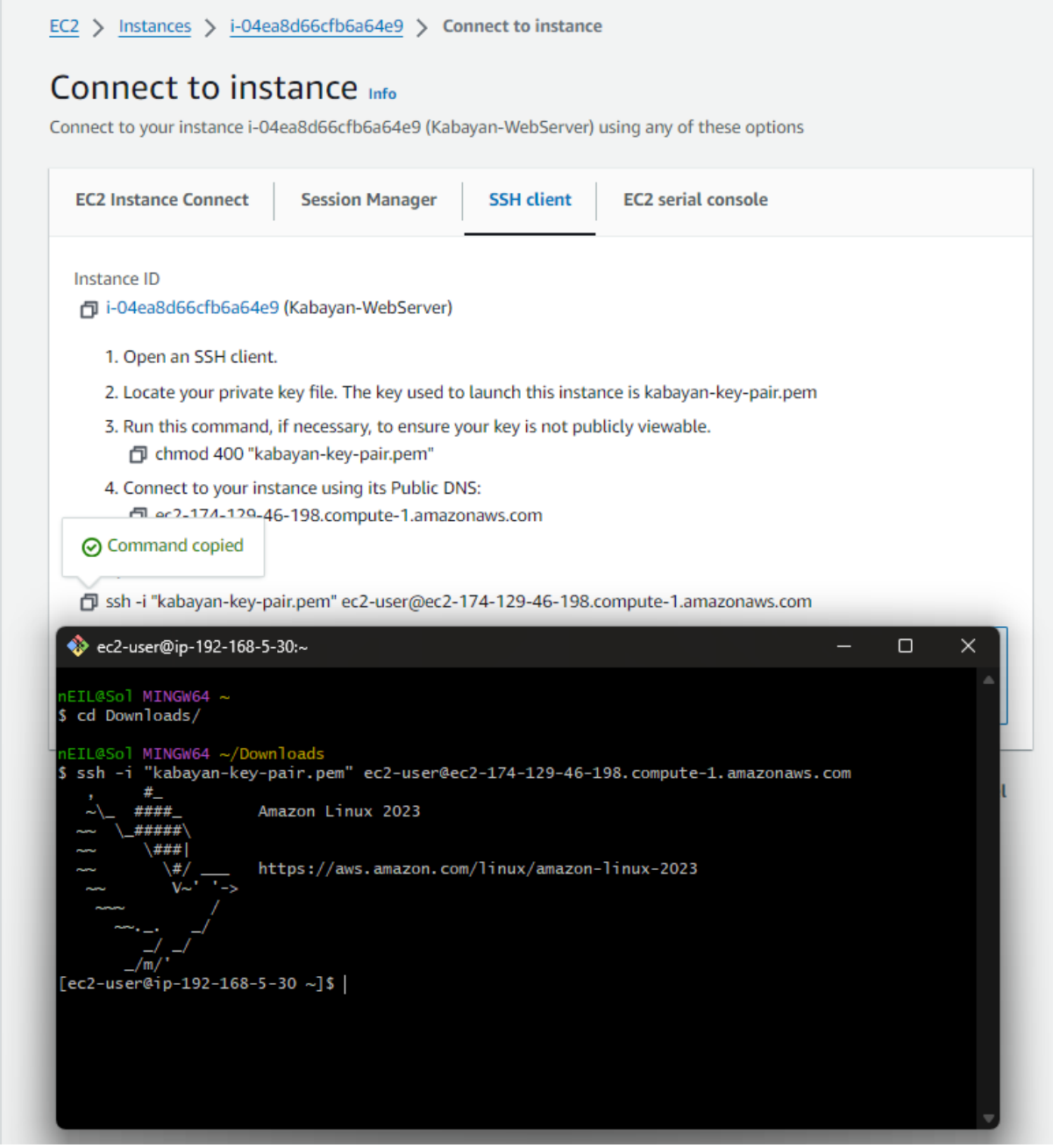
Integrating Amazon RDS to Amazon EC2 Instance

1.**Connect to EC2 Instance:**

- Open your SSH client. (e.g. GitBash)
- Connect to the EC2 instance using the public DNS name and key pair:

Do not forget to change the value of the placeholders in the following command

ssh -i </path/to/your-key-pair.pem> ec2-user@<your-ec2-public-dns>

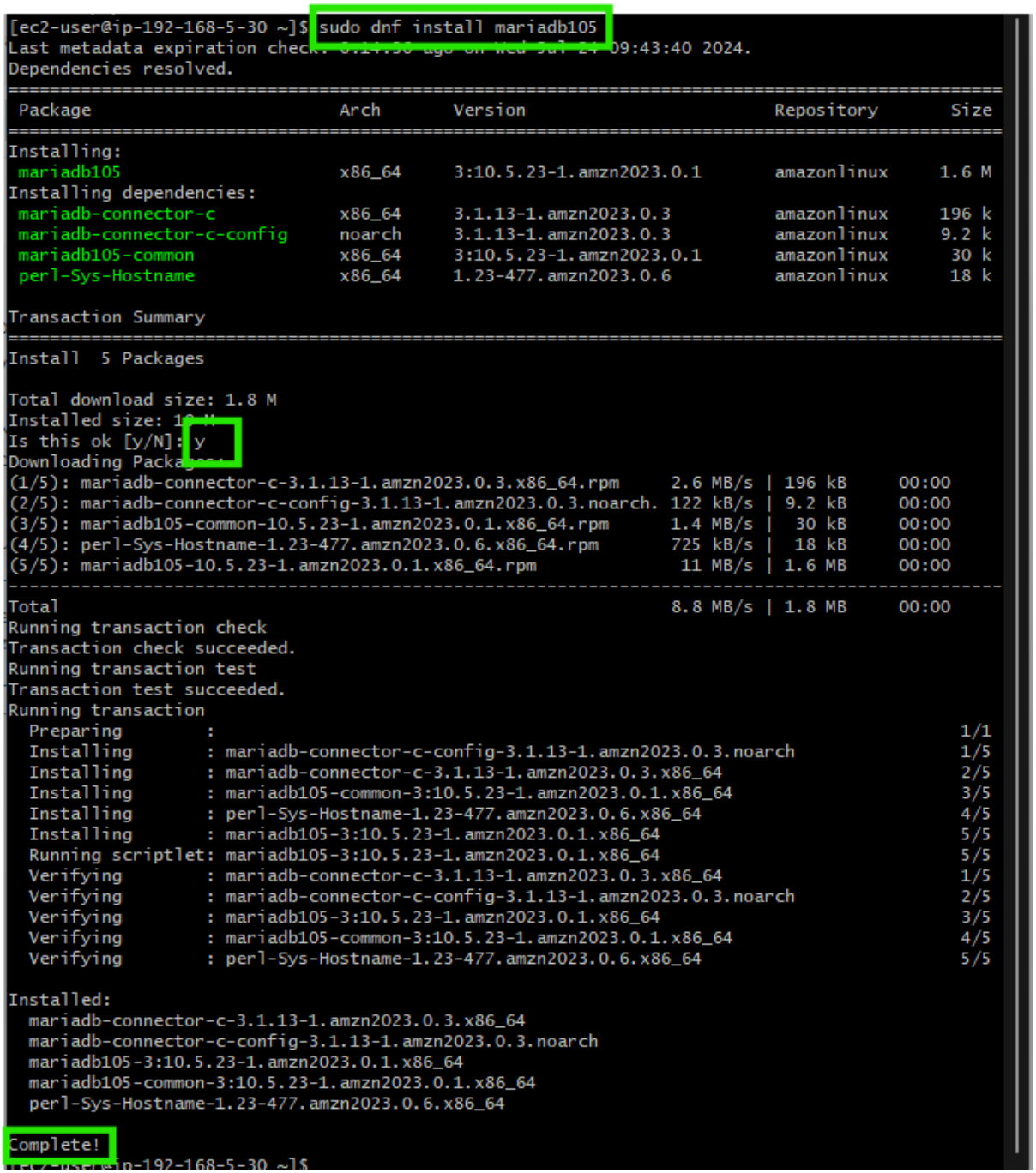


2. **Install a MySQL Client application (if using MySQL RDS):**

MySQL Client Applications: This is a command-line tool or application that allows users to interact with the MySQL server. The client sends SQL queries to the MySQL server and retrieves the results. It is essentially a way for users to communicate with the MySQL server to perform database operations like querying, updating, and managing data.

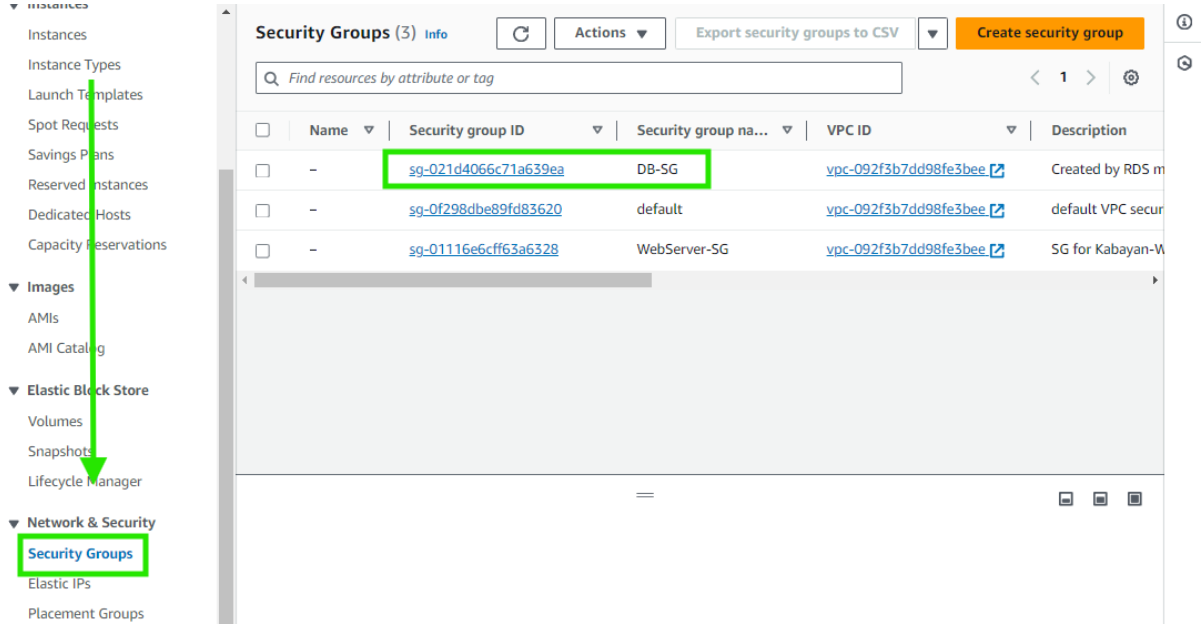
Type the following command and press **ENTER**

sudo dnf install mariadb105



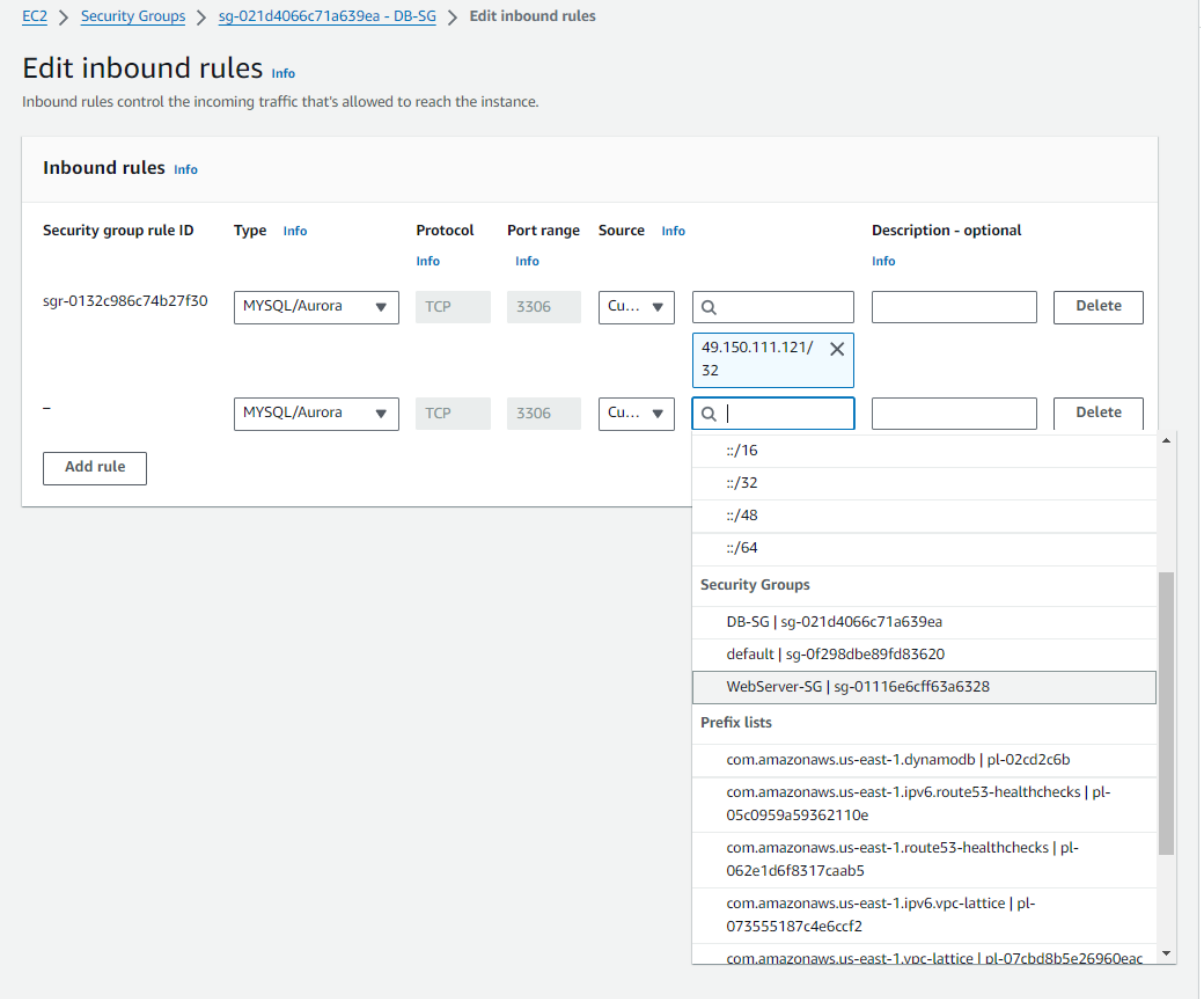
3. Before we connect to the RDS Instance, as mentioned earlier, we need to ensure that our Security Groups Inbound Rules are configured correctly, as well as for DB-SG:

a. Navigate to the **EC2 Console > Security Groups > DB-SG**



b. Edit the inbound rule, and add this rule: (This is similar inbound rule we did in the WebServer-SG earlier)

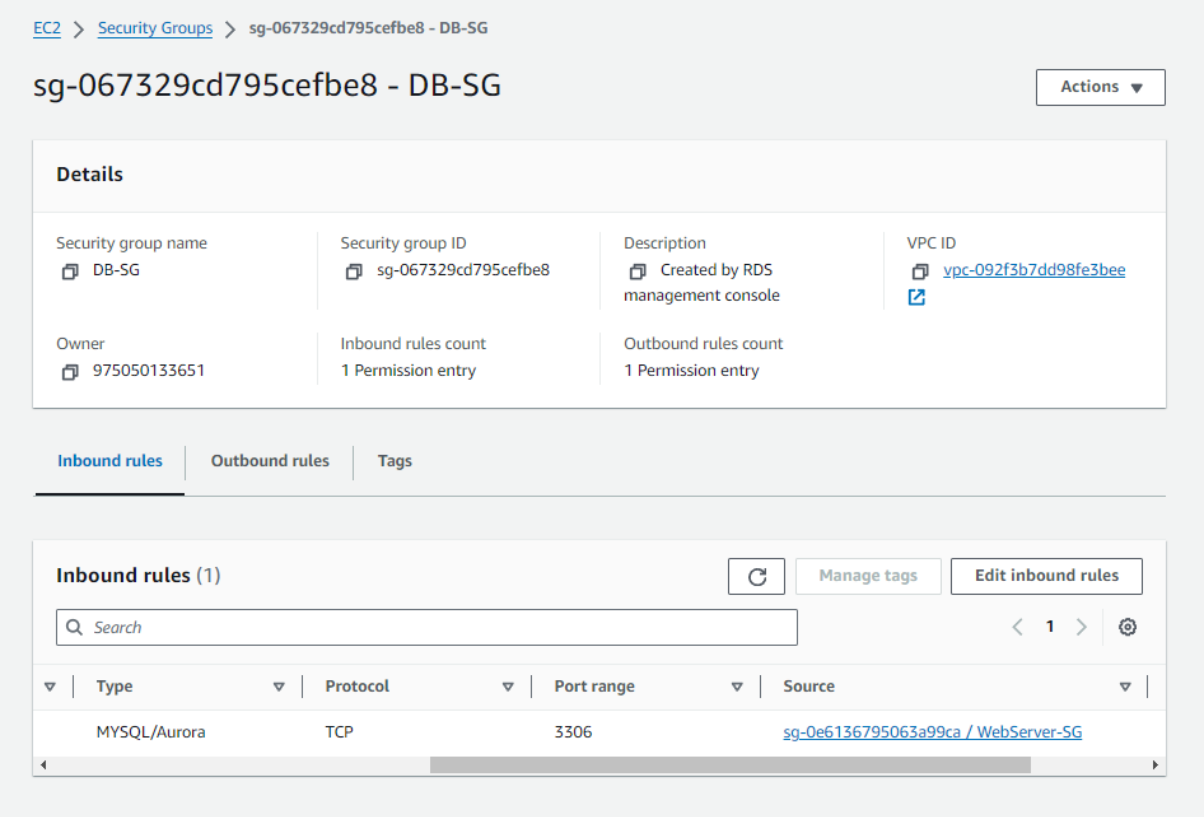
-
    - **Inbound rule 1**:
        - **Type:** MYSQL/Aurora
        - **Source:** Select theWebServer-SG



This will allow connection between the DB-SG and WebServer-SG.

---

c. Click **Save rules**

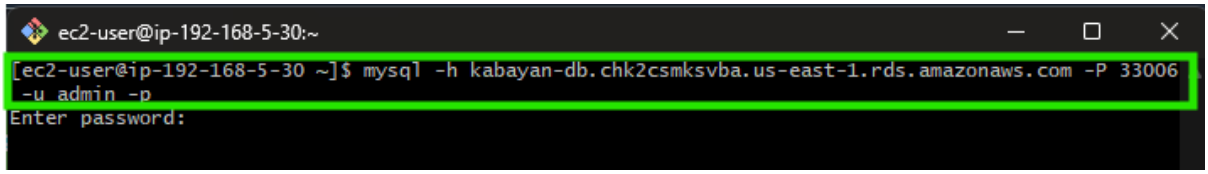


4. **Connect to the RDS Instance from EC2:**

- Use the MySQL client to connect to your RDS instance:

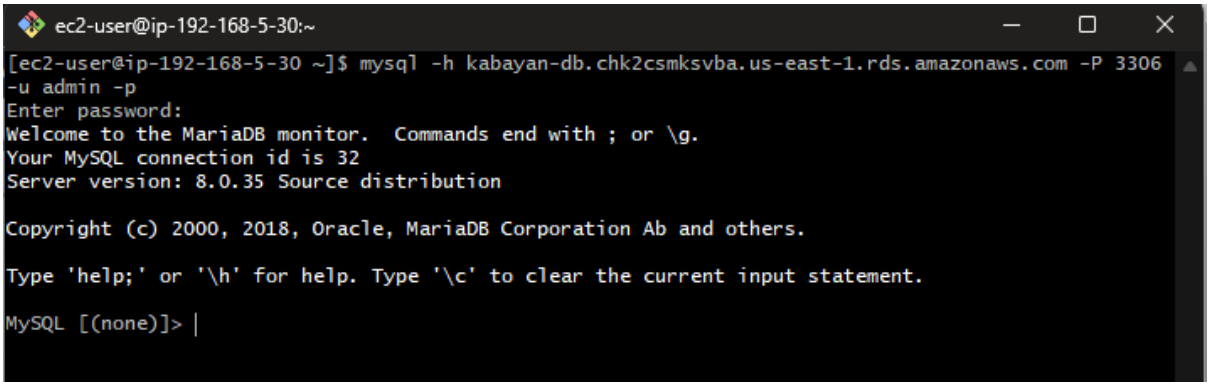
You gonna be needing the RDS endpoint you copied earlier, the username, the password and the port which is usually 3306.
Do not forget to change the value of the placeholders

mysql -h <your-rds-endpoint> -P 3306 -u <your-master-username> -p



- Enter your master password when prompted.

Note you wont be able to see the password you type here. Just Type or paste it correctly and hit Enter



5. **Verify Connection:**

- Once connected, run some SQL commands to verify the connection:

SHOW DATABASES;

```
MySQL [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.007 sec)

MySQL [(none)]>
```

6. **Create a Database and Table:**

a. You can now interact freely with your RDS Databsee like creating Database using the command:

Create DATABASE EmployeesDB;

```
MySQL [(none)]> Create DATABASE EmployeesDB;
Query OK, 1 row affected (0.007 sec)
```

You can check this by the same command

SHOW DATABASES;

```
MySQL [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| EmployeesDB        |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.005 sec)

MySQL [(none)]>
```

b. Choose the database that has been created.

USE EmployeesDB;

```
MySQL [(none)]> USE EmployeesDB;
Database changed
MySQL [EmployeesDB]>
```

c. Creating a table.

```
CREATE TABLE Employees (
    ID INT PRIMARY KEY,
    Name VARCHAR(50),
    Age INT,
```

Salary DECIMAL(10, 2)

);

```
MySQL [EmployeesDB]> CREATE TABLE Employees (
    ->     ID INT PRIMARY KEY,
    ->     Name VARCHAR(50),
    ->     Age INT,
    ->     Salary DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (0.059 sec)
```

d. Describe the table.

DESCRIBE Employees;

```
MySQL [EmployeesDB]> DESCRIBE Employees;
+--------+---------------+------+-----+---------+-------+
| Field  | Type          | Null | Key | Default | Extra |
+--------+---------------+------+-----+---------+-------+
| ID     | int           | NO   | PRI | NULL    |       |
| Name   | varchar(50)   | YES  |     | NULL    |       |
| Age    | int           | YES  |     | NULL    |       |
| Salary | decimal(10,2) | YES  |     | NULL    |       |
+--------+---------------+------+-----+---------+-------+
4 rows in set (0.004 sec)
```

Congratulations on completing the lab! You've successfully set up an EC2 instance and an RDS database, and connected the two, leveraging the power of AWS to create a scalable, secure, and efficient architecture for your applications. This is just one of the many ways to securely connect your RDS database to your instance.

You've gained practical experience with AWS services, including creating and configuring EC2 and RDS instances, setting up security groups, and connecting to a database from a virtual server. These skills are fundamental for deploying and managing applications in the cloud like:

- **Web Applications:** Use EC2 for your web servers and RDS for your databases to build robust, scalable web applications.
- **Data Processing:** Leverage EC2 for processing tasks and RDS for storing processed data

By mastering the integration of EC2 and RDS, you've taken a significant step towards building modern, cloud-native applications. This foundational knowledge empowers you to explore more advanced AWS services and architectures, further enhancing your ability to design and deploy scalable, reliable, and secure applications in the cloud. Happy learning!