we will cover everything you need to know about **Amazon RDS**, **Features and Benefits of RDS, Amazon RDS Pricing,** and a step-by-step tutorial to create a Relational Database on AWS.

Topics covered in this blog:

What is Amazon RDS?

Amazon RDS is a **Relational Database Service** by Amazon Web Services. A relational database is a type of database that stores data in tables with rows and columns. Amazon RDS provides an industry-standard relational database that is cost-efficient with resizable capacity. It gives users solutions for easier setup, operation, and scale of a relational database in the cloud.

Check Also: Free AWS Training and Certifications

Features of Amazon RDS

- It manages software patching, automatic failure detection, backups, and recovery.
- Offers flexibility to opt for automated backups or manual backup snapshots. In case of data loss, you can restore the backups at any time.
- Flexibility to pick your database engine among **MySQL**, **MariaDB**, **PostgreSQL**, **Oracle**, and **Microsoft SQL Server**.
- Enable more security with **AWS IAM** (Identity and Access Management) to ensure that access remains only for the selected users with limited permissions.

Benefits of Amazon RDS

- **Ease of use:** AWS RDS is a fully managed service, which means that Amazon takes care of all the details of database administration, such as provisioning hardware, installing software, and managing backups. This frees up your developers and IT staff to focus on other tasks.

- **Scalability**: AWS RDS is highly scalable, so you can easily add or remove database capacity as needed. This makes it ideal for applications that experience fluctuating traffic patterns.
- **Security:** AWS RDS offers a wide range of security features, such as encryption, access control, and auditing. This helps you to protect your data from unauthorized access.
- **Cost-effectiveness:** AWS RDS is a cost-effective way to host your databases. You only pay for the resources that you use, so you can save money by only provisioning the capacity that you need.

Database Engine Selection in RDS

When using AWS RDS (Relational Database Service), choosing the right database engine is a crucial decision that impacts the performance, compatibility, and functionality of your applications. AWS RDS supports several popular database engines, including MySQL, PostgreSQL, Oracle, SQL Server, and Amazon Aurora. Each engine has its strengths and use cases, so it's essential to consider the following factors when making your selection:

1. **Performance and Scalability:** Evaluate the performance requirements of your application. Consider factors such as transaction throughput, read and write operations, and response time. MySQL and PostgreSQL are known for their performance and scalability capabilities, making them suitable for high-traffic applications. Amazon Aurora, a MySQL and PostgreSQL-compatible database, offers enhanced performance and scalability with its innovative storage and replication architecture.
2. **Features and Functionality:** Examine the specific features and functionality provided by each database engine. MySQL is popular for its ease of use, wide community support, and compatibility with various applications. PostgreSQL offers advanced features such as JSON support, full-text search capabilities, and robust transaction management. Oracle and SQL Server are suitable for enterprises requiring comprehensive enterprise-grade features, such as advanced analytics, security, and high availability options.
3. **Application Compatibility:** Consider the compatibility of the database engine with your existing applications and frameworks. If you have applications developed with MySQL or PostgreSQL, it may be beneficial to stick with the same engine for seamless migration and reduced development effort. AWS RDS provides tools and resources to facilitate database migration between engines, allowing you to switch engines if needed.
4. **Vendor Lock-In and Portability:** Evaluate the potential vendor lock-in and portability concerns associated with the database engine. AWS RDS provides managed services for popular open-source engines like MySQL and PostgreSQL, allowing you to migrate your databases to self-managed environments or other cloud providers if necessary. If you anticipate the need for portability and flexibility in the future, choosing an open-source engine might be a suitable option.

5. **Licensing and Cost:** Consider the licensing requirements and associated costs of the database engine. Oracle and SQL Server are commercial database engines with additional licensing costs, while MySQL and PostgreSQL are open-source and typically have lower licensing costs. AWS RDS offers options for both licensed and open-source engines, allowing you to select the engine that aligns with your budget and licensing requirements.

Amazon RDS Pricing

Amazon RDS is free to try, and you can test your environment with basic settings. If you are setting up Amazon RDS for your organization with more frequent usage, then the charges depend on the usage of the resources. There are various database engines on which the Amazon RDS pricing depends like **Amazon Aurora**, **My SQL**, **PostgreSQL**, **Maria DB**, **Oracle**, and **Microsoft SQL Server**. On running Amazon RDS, the charges are made on the following parameters.

- On-Demand DB Instances
- Reserved Instances
- Database Storage
- Backup Storage
- Snapshot Export
- Data Transfer

If you want to learn the exact Amazon RDS Pricing, you can check here.

Also Check: Our blog post on **Cloud Service Models**.

Amazon RDS Tutorial

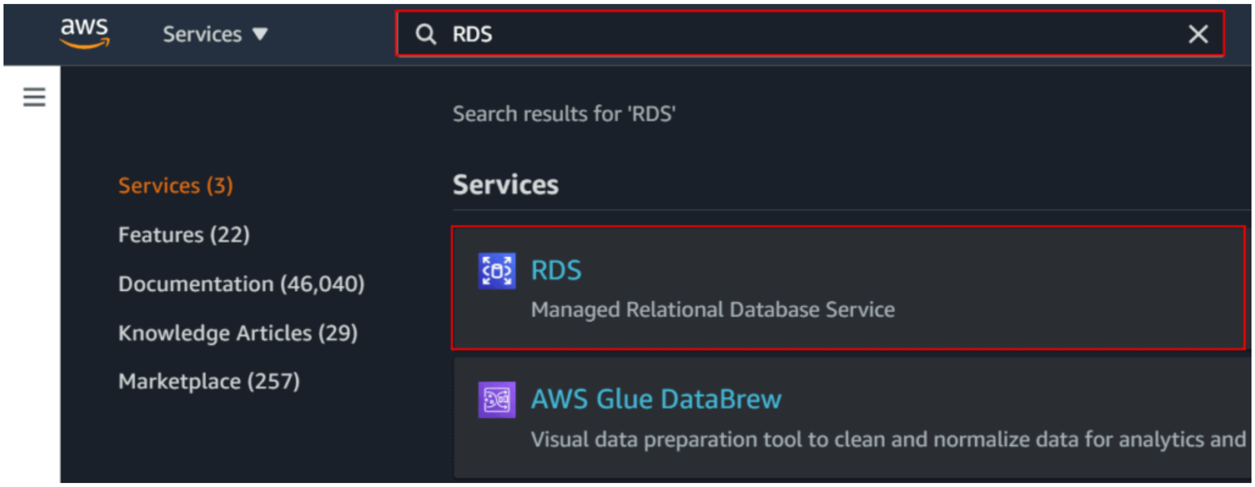
If you are a beginner and don't have an AWS account, see our blog on creating an **AWS Free Tier Account.**

Now, I assume that you already have an AWS account ready with you. To start using **Relational Database Service**, visit **console.aws.amazon.com** and log in with your credentials.

Step 1) Create Amazon RDS

After log in to the AWS console, follow the below steps.

1) After visiting your AWS console, search for RDS in the top search bar. Click on the RDS Service as highlighted in the image below.



aws Services ▼

RDS ✕

Search results for 'RDS'

Services (3)

Features (22)

Documentation (46,040)

Knowledge Articles (29)

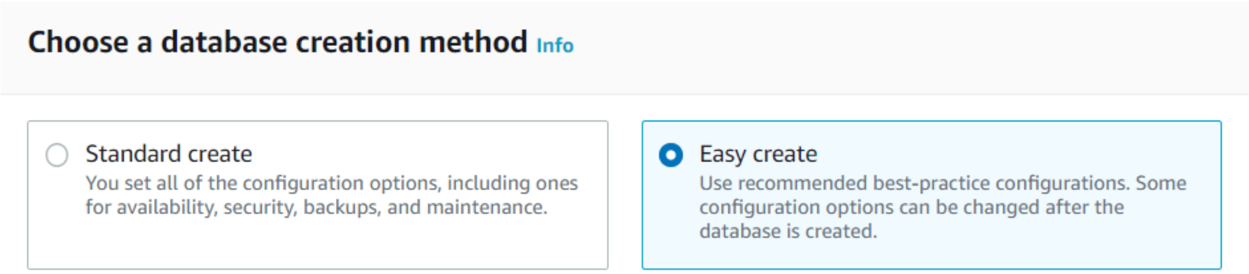
Marketplace (257)

Services

RDS
Managed Relational Database Service

AWS Glue DataBrew
Visual data preparation tool to clean and normalize data for analytics and

2) After clicking on the RDS Service, a new window will appear on your screen. You will be provided with two options to create. One is **Standard Create** which will offer you to choose all the additional settings and configurations. The other is **Easy Create,** which will automatically pick the default configurations for your RDS. Depending on your need, you can pick any option, but I will pick the Easy Create option for a simple creation process for beginners.

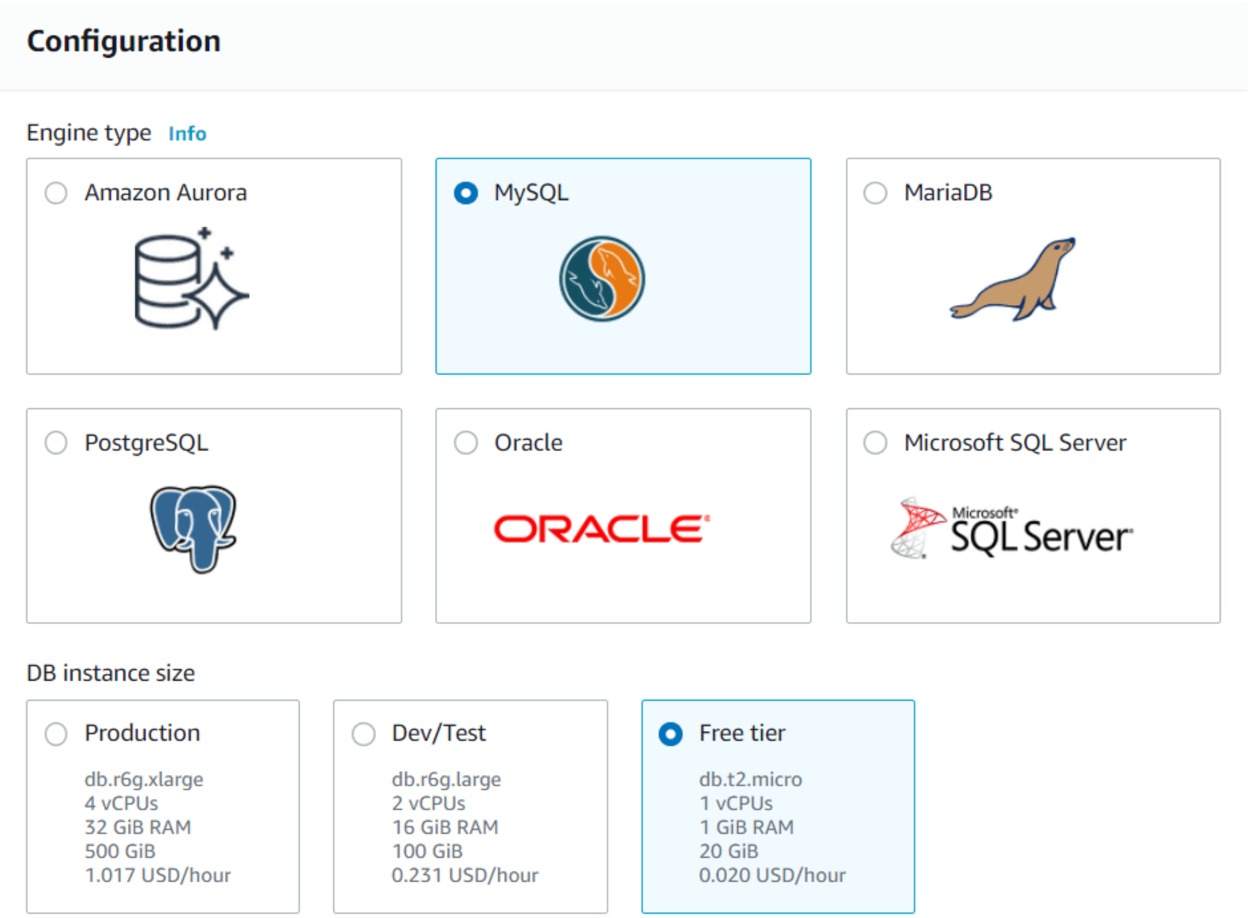


Choose a database creation method Info

○ 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.

3) Now, you need to pick the RDS Engine. The types of RDS Engines available are Amazon Aurora, My SQL, Maria DB, PostgreSQL, Oracle, and Microsoft SQL Server. Based on your need, you can pick any of the RDS engines. I have picked MySQL in my case.



Configuration

Engine type Info

○ Amazon Aurora

● MySQL

○ MariaDB

○ PostgreSQL

○ Oracle

○ Microsoft SQL Server

DB instance size

○ Production
db.r6g.xlarge
4 vCPUs
32 GiB RAM
500 GiB
1.017 USD/hour

○ Dev/Test
db.r6g.large
2 vCPUs
16 GiB RAM
100 GiB
0.231 USD/hour

● Free tier
db.t2.micro
1 vCPUs
1 GiB RAM
20 GiB
0.020 USD/hour

After selecting RDS Engine, there will be some options to choose the DB instance size, and these options change with the selection you made above. Also, based on your RDS Engine and DB instance size, the payment is made accordingly.

4) Now, in the settings of the Amazon RDS, fill in all the details. First is the DB cluster identifier field. Fill it with the name you want to assign to your database.

Then, assign the username and password for your database. You can also pick the auto-generated password option that will generate a strong password for you.

5) For the Easy Create option, only this much is needed to be configured, and for the rest, all the default settings are used. In the end, all the default settings used for your AWS RDS are shown, and you can match them with your needs.

6) Now, the final step is to click on the Create Database button at the end, as highlighted below.

Also Check: Our bog post on **AWS SNS**.

Step 2) Manage Amazon RDS

After the database is created, you are redirected to the screen where all your databases are present. You can search your database name in the search field if you have multiple databases created. Soon after creating your database, the status is shown. It will take a few minutes to be completely available. From this window, you can also modify and delete your created databases.

Also Read: Our blog post on **AWS Secrets Manager**.

Step 3) Connect to Amazon RDS

To connect your database, you need to get the endpoint and port for your database. And to access it, you will require an SQL client.

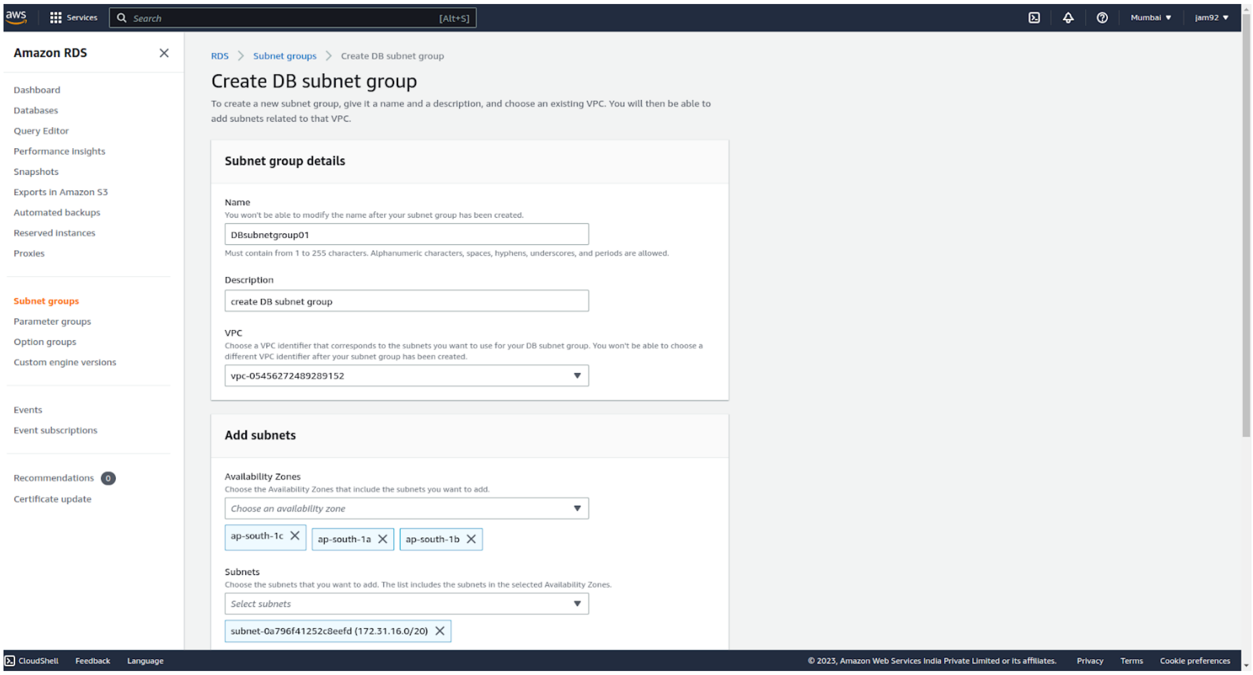
1) Visit your database by clicking on your created RDS name as highlighted in the above image. Now you will see all the details of your database. Click on connectivity & security and copy your endpoint URL and port number. Both these values are used to connect a database.

2) Download your SQL client and configure it. For the MySQL engine, the **MySQL Workbench** tool is needed. You can download it from here.

```
PROMPT> mysql -h <endpoint> -P 3306 -u <mymasteruser> -p
```

Now, to create a connection to your database, run the above command. After running it, the database is connected to your SQL client.

We need to create the subnet group first



Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

Name
You won't be able to modify the name after your subnet group has been created.
DBsubnetgroup01
Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

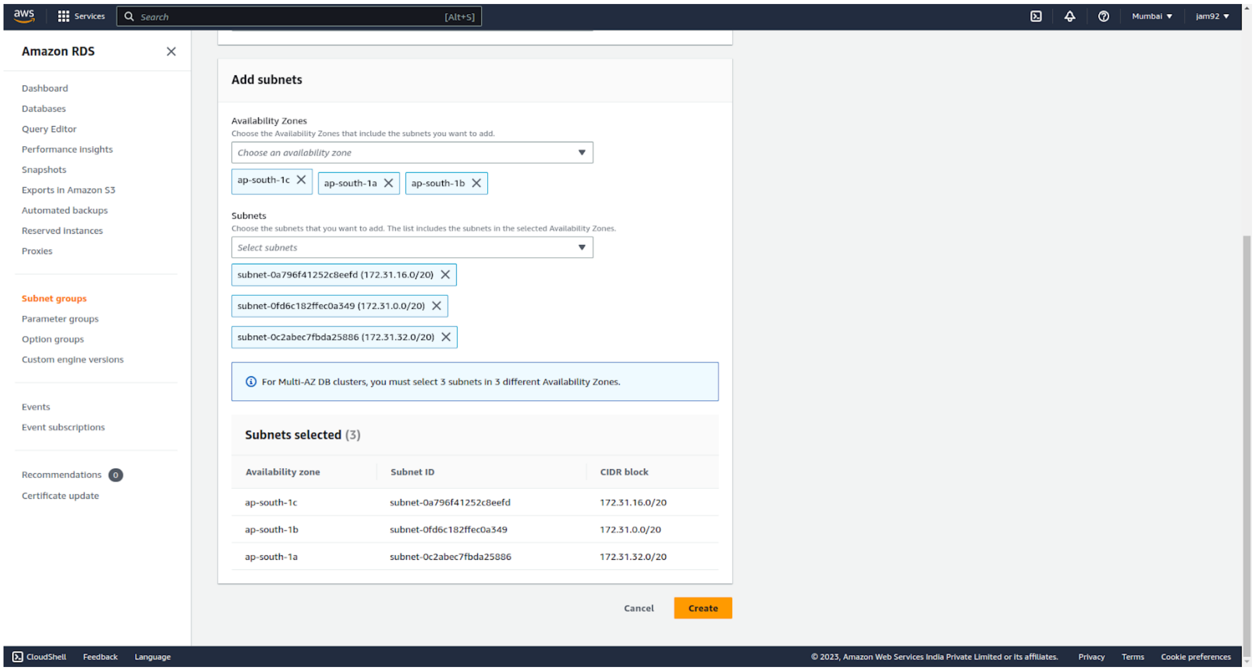
Description
create DB subnet group

VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.
vpc-05456272489289152 ▾

Add subnets

Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone ▾
ap-south-1c ✕ ap-south-1a ✕ ap-south-1b ✕

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets ▾
subnet-0a796f41252c8eefd (172.31.16.0/20) ✕



Add subnets

Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone ▾
ap-south-1c ✕ ap-south-1a ✕ ap-south-1b ✕

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets ▾
subnet-0a796f41252c8eefd (172.31.16.0/20) ✕
subnet-0fd6c182ffec0a349 (172.31.0.0/20) ✕
subnet-0c2abec7fbda25886 (172.31.32.0/20) ✕

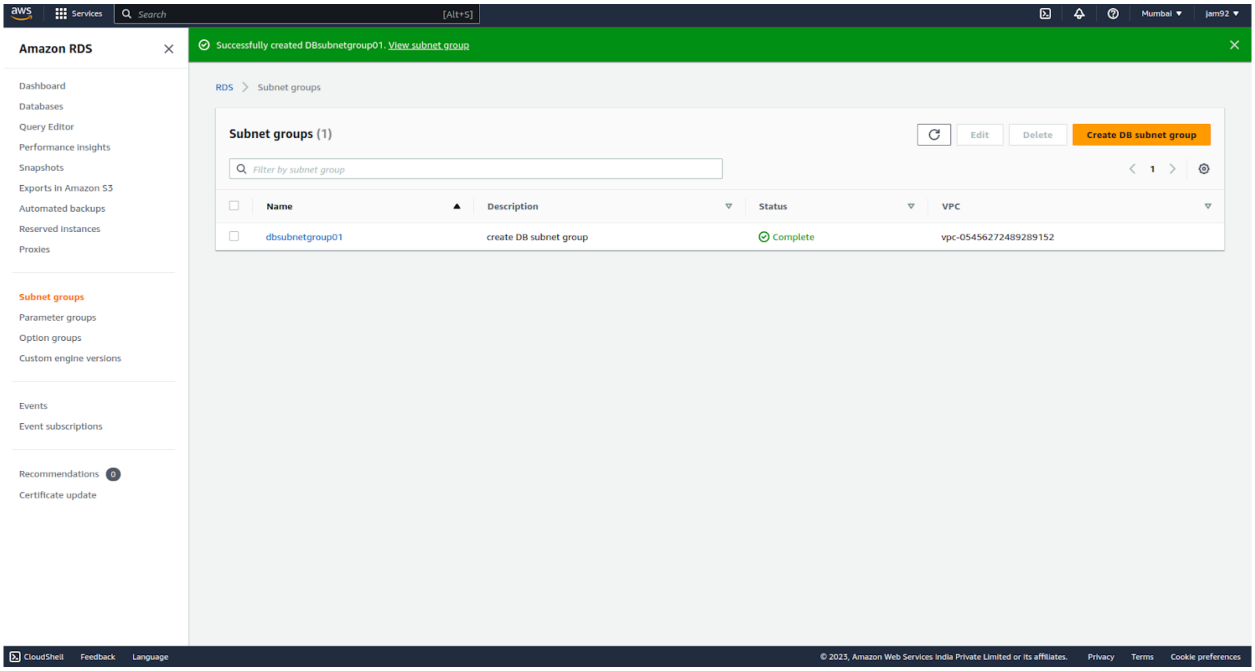
ⓘ For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Subnets selected (3)

Availability zone	Subnet ID	CIDR block
ap-south-1c	subnet-0a796f41252c8eefd	172.31.16.0/20
ap-south-1b	subnet-0fd6c182ffec0a349	172.31.0.0/20
ap-south-1a	subnet-0c2abec7fbda25886	172.31.32.0/20

Cancel Create

You can see we had created the DB subnet group in the below image

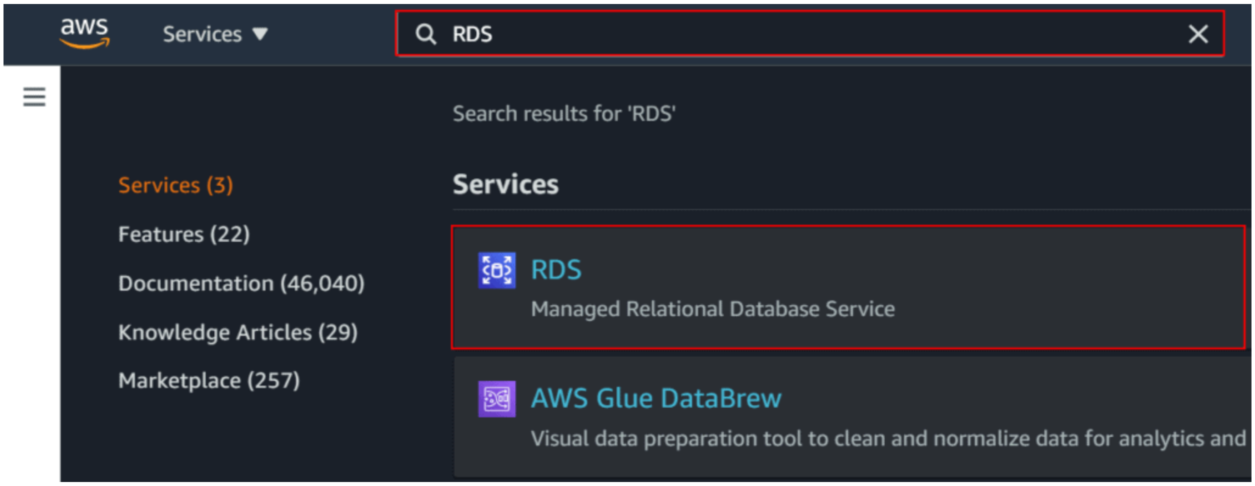


Successfully created DBsubnetgroup01. View subnet group

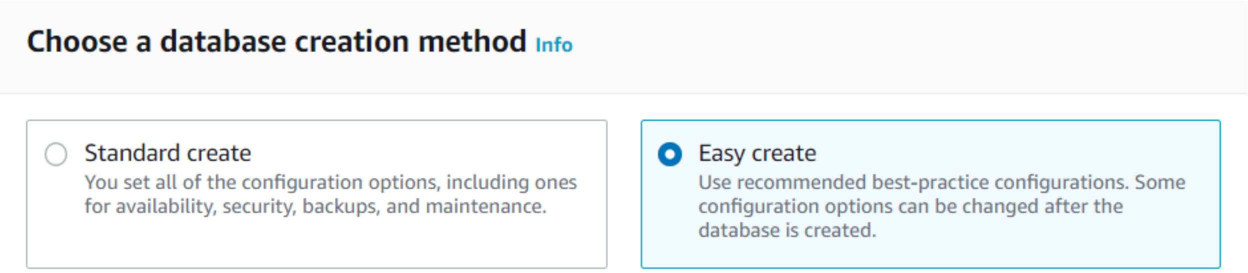
Subnet groups (1)

□	Name	Description	Status	VPC
□	dbsubnetgroup01	create DB subnet group	✓ Complete	vpc-05456272489289152

1) After visiting your AWS console, search for RDS in the top search bar. Click on the RDS Service as highlighted in the image below.



2) After clicking on the RDS Service, a new window will appear on your screen. You will be provided with two options to create. One is **Standard Create** which will offer you to choose all the additional settings and configurations. The other is **Easy Create,** which will automatically pick the default configurations for your RDS. Depending on your need, you can pick any option, but I will pick the Easy Create option for a simple creation process for beginners.

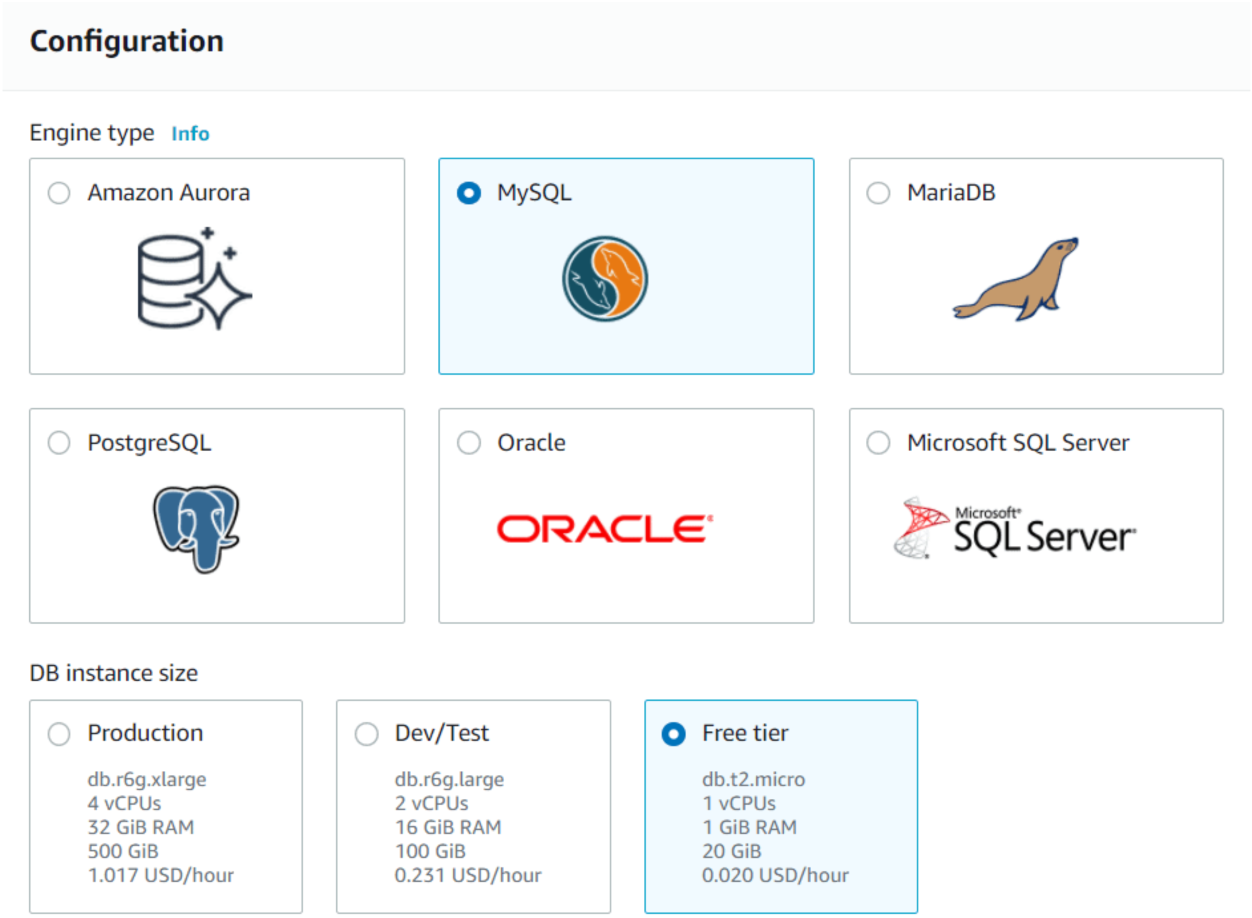


Choose a database creation method Info

○ 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.

3) Now, you need to pick the RDS Engine. The types of RDS Engines available are Amazon Aurora, My SQL, Maria DB, PostgreSQL, Oracle, and Microsoft SQL Server. Based on your need, you can pick any of the RDS engines. I have picked MySQL in my case.



Configuration

Engine type Info

○ Amazon Aurora

● MySQL

○ MariaDB

○ PostgreSQL

○ Oracle

○ Microsoft SQL Server

DB instance size

○ Production
db.r6g.xlarge
4 vCPUs
32 GiB RAM
500 GiB
1.017 USD/hour

○ Dev/Test
db.r6g.large
2 vCPUs
16 GiB RAM
100 GiB
0.231 USD/hour

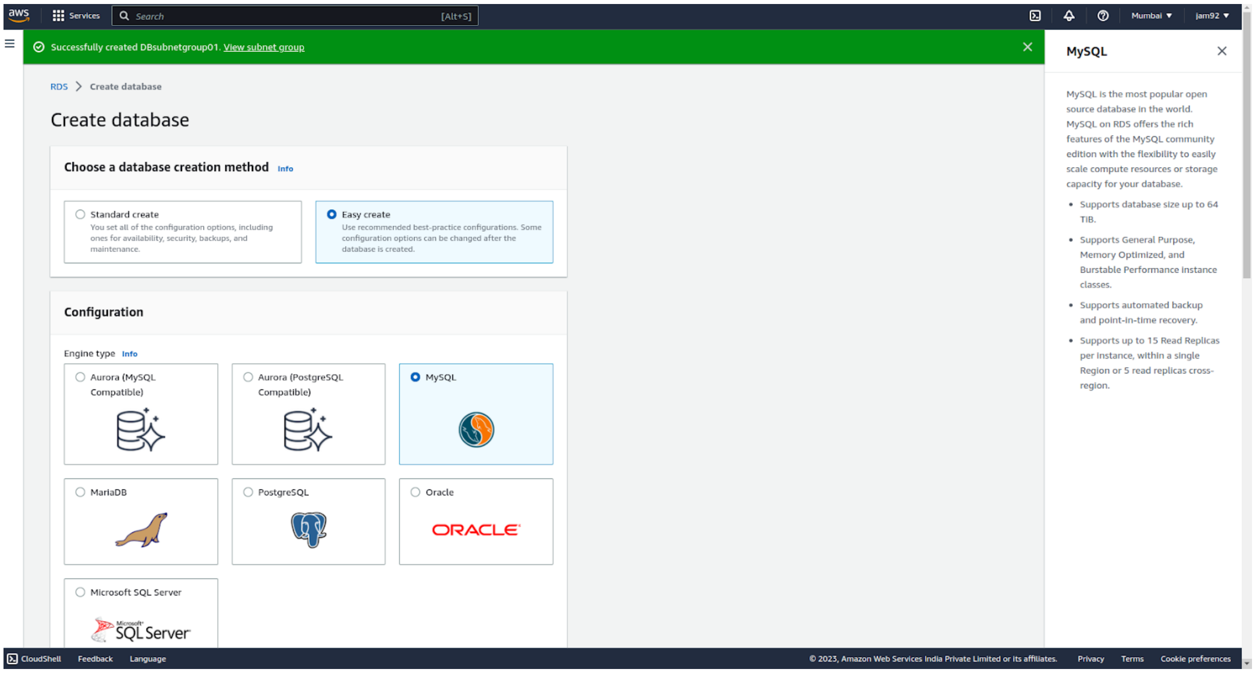
● Free tier
db.t2.micro
1 vCPUs
1 GiB RAM
20 GiB
0.020 USD/hour

After selecting RDS Engine, there will be some options to choose the DB instance size, and these options change with the selection you made above. Also, based on your RDS Engine and DB instance size, the payment is made accordingly.

4) Now, in the settings of the Amazon RDS, fill in all the details. First is the DB cluster identifier field. Fill it with the name you want to assign to your database.

Then, assign the username and password for your database. You can also pick the auto-generated password option that will generate a strong password for you.

5) For the Easy Create option, only this much is needed to be configured, and for the rest, all the default settings are used. In the end, all the default settings used for your AWS RDS are shown, and you can match them with your needs.



Successfully created DBsubnetgroup01. View subnet group

RDS > 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.

Configuration

Engine type Info

○ Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

● MySQL

○ MariaDB

○ PostgreSQL

○ Oracle

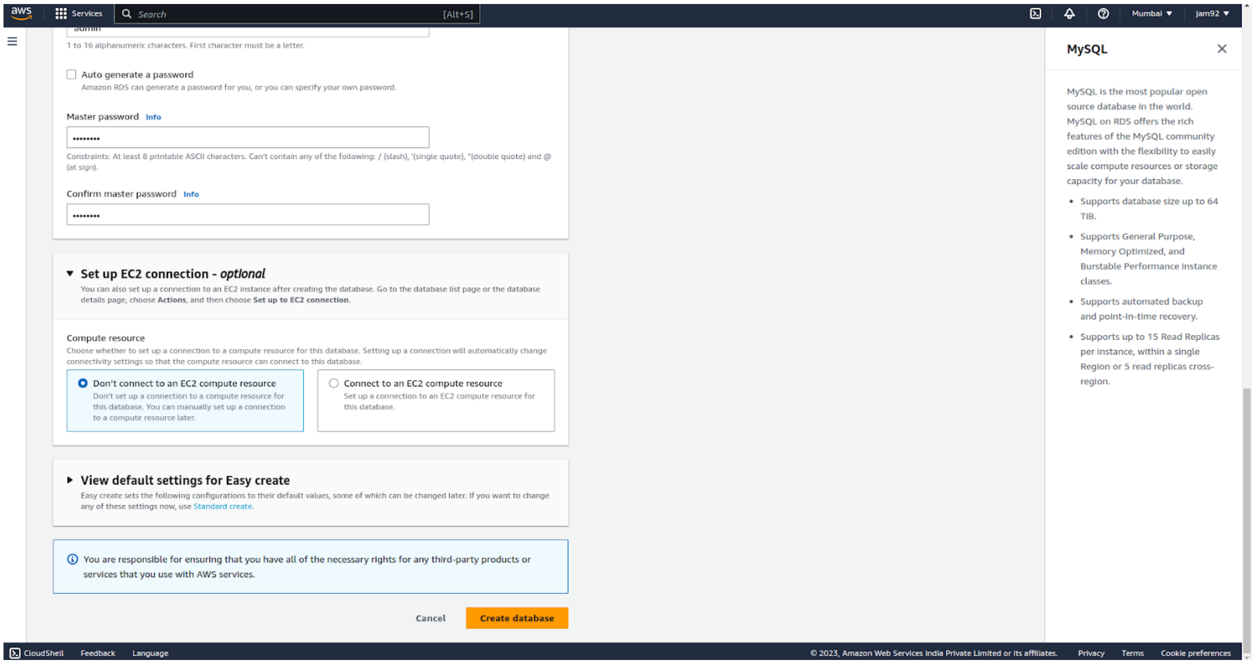
○ Microsoft SQL Server

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences



admin

1 to 16 alphanumeric characters. First character must be a letter.

□ Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

•••••••••

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password Info

•••••••••

▼ Set up EC2 connection - *optional*

You can also set up a connection to an EC2 instance after creating the database. Go to the database list page or the database details page, choose **Actions**, and then choose **Set up to EC2 connection**.

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.

▶ View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use Standard create.

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.

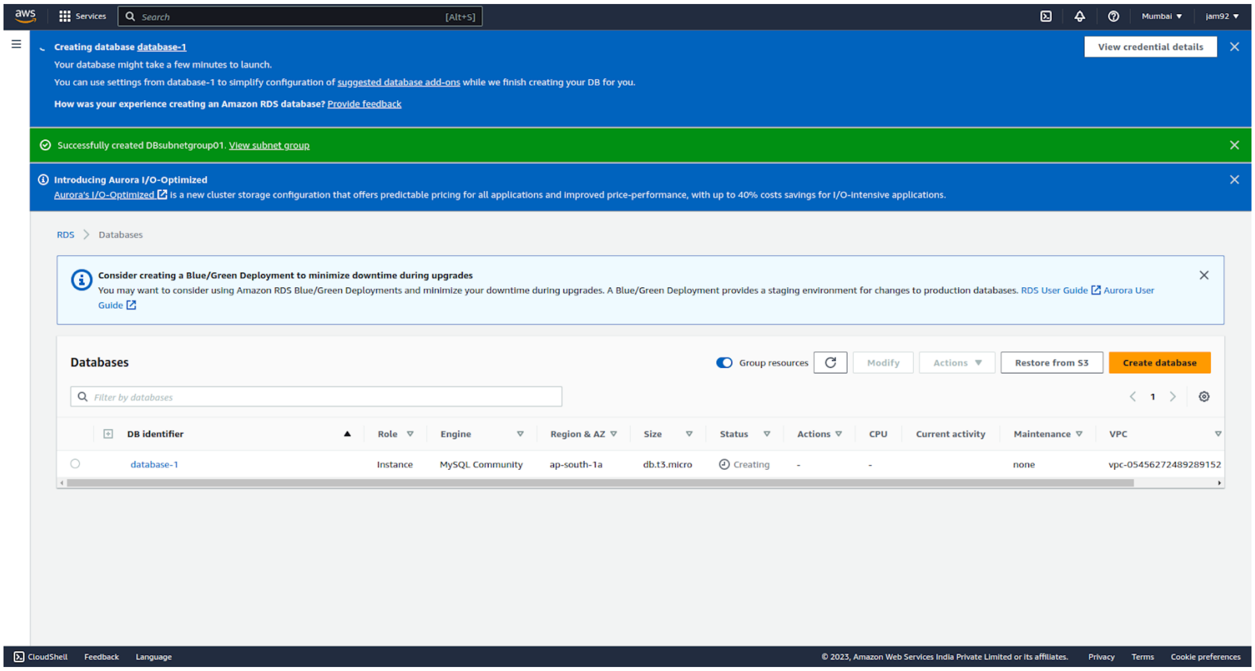
MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

Now, I had created our db,



Creating database database-1
Your database might take a few minutes to launch.
You can use settings from database-1 to simplify configuration of suggested database add-ons while we finish creating your DB for you.
How was your experience creating an Amazon RDS database? Provide feedback

Successfully created DBsubnetgroup01. View subnet group

Introducing Aurora I/O-Optimized
Aurora's I/O-Optimized is a new cluster storage configuration that offers predictable pricing for all applications and improved price-performance, with up to 40% costs savings for I/O-intensive applications.

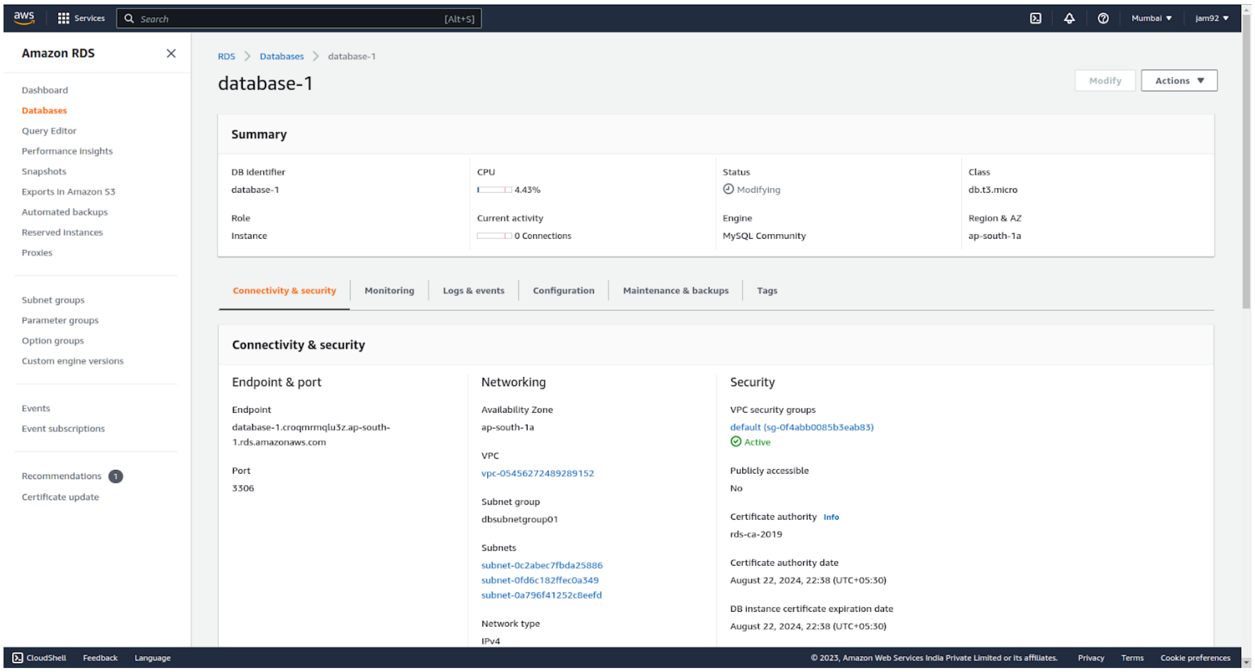
RDS > Databases

Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide

Databases

	DB identifier	Role	Engine	Region & AZ	Size	Status	Actions	CPU	Current activity	Maintenance	VPC
○	database-1	Instance	MySQL Community	ap-south-1a	db.t3.micro	Creating	-	-		none	vpc-05456272489289152

CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences



database-1

Summary

DB identifier
database-1

Role
Instance

CPU
4.43%

Current activity
0 Connections

Status
Modifying

Engine
MySQL Community

Class
db.t3.micro

Region & AZ
ap-south-1a

Connectivity & security

Endpoint & port

Endpoint
database-1.croqmrmqlu3z.ap-south-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
ap-south-1a

VPC
vpc-05456272489289152

Subnet group
dbsubnetgroup01

Subnets
subnet-0c2abec7fbda25886
subnet-0fd6c182ffec0a349
subnet-0a796f41252c8eefd

Network type
IPv4

Security

VPC security groups
default (sg-0f4abb0085b3eab83)
Active

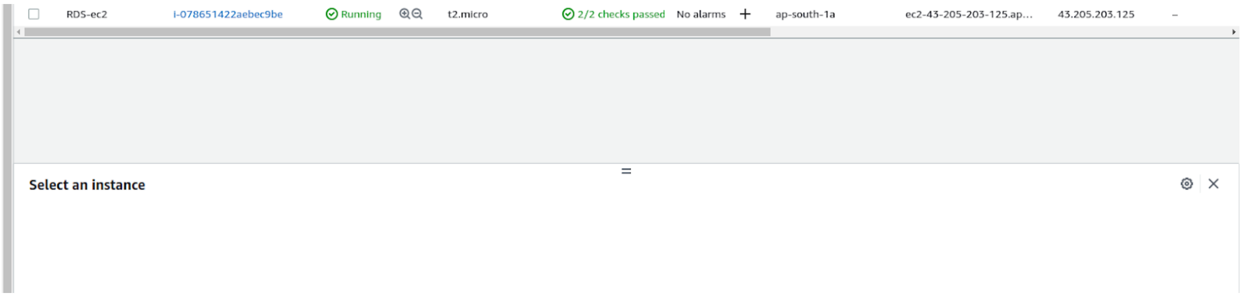
Publicly accessible
No

Certificate authority Info
rds-ca-2019

Certificate authority date
August 22, 2024, 22:38 (UTC+05:30)

DB instance certificate expiration date
August 22, 2024, 22:38 (UTC+05:30)

,now you need to create a EC@ instance



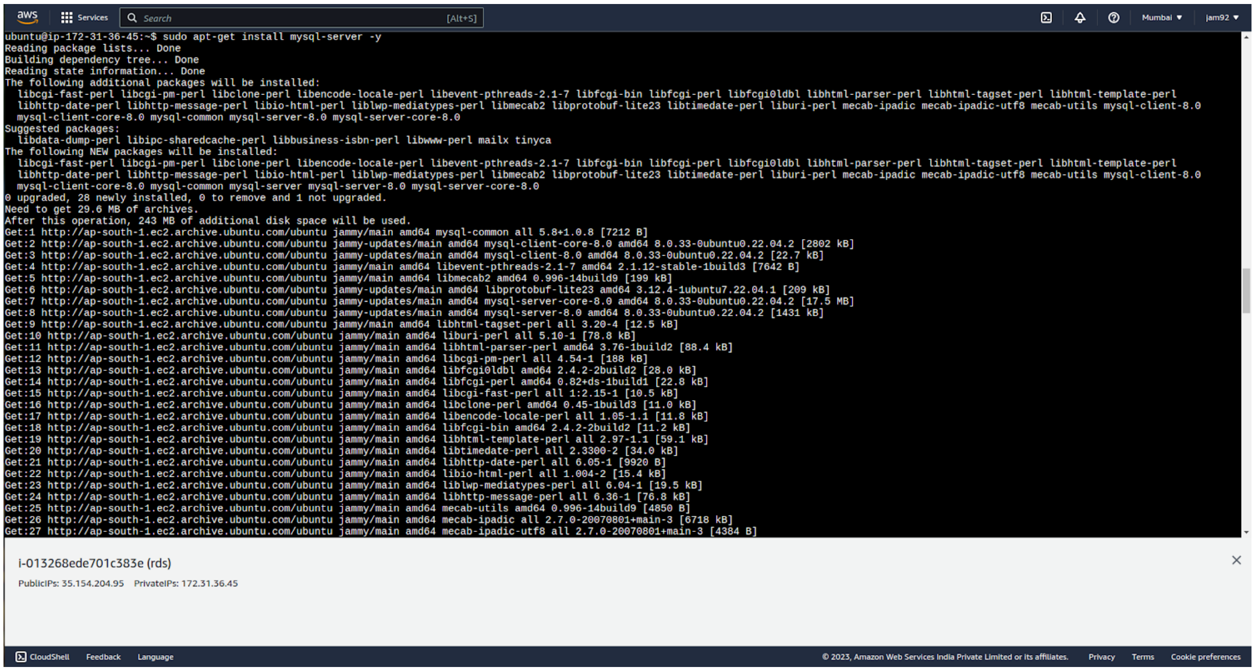
□	RDS-ec2	i-078651422aebec9be	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms	+	ap-south-1a	ec2-43-205-203-125.ap...	43.205.203.125	–

After connect EC2 instance you need to install the mysql on instance.

sudo apt update

sudo apt-get install mysql-server -y

mysql --version



```
ubuntu@ip-172-31-36-45:~$ sudo apt-get install mysql-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi0ldbl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite23 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi0ldbl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite23 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-common mysql-server mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 28 newly installed, 0 to remove and 1 not upgraded.
Need to get 29.6 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+1.0.8 [7212 B]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-core-8.0 amd64 8.0.33-0ubuntu0.22.04.2 [2802 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-8.0 amd64 8.0.33-0ubuntu0.22.04.2 [22.7 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libevent-pthreads-2.1-7 amd64 2.1.12-stable-1build3 [7642 B]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libmecab2 amd64 0.996-14build9 [199 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libprotobuf-lite23 amd64 3.12.4-1ubuntu7.22.04.1 [209 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-server-core-8.0 amd64 8.0.33-0ubuntu0.22.04.2 [17.5 MB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-server-8.0 amd64 8.0.33-0ubuntu0.22.04.2 [1431 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-tagset-perl all 3.20-4 [12.5 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liburi-perl all 5.10-1 [78.8 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-parser-perl amd64 3.76-1build2 [88.4 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libcgi-pm-perl all 4.54-1 [188 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libfcgi0ldbl amd64 2.4.2-2build2 [28.0 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libfcgi-perl amd64 0.82+ds-1build1 [22.8 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libcgi-fast-perl all 1:2.15-1 [10.5 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libclone-perl amd64 0.45-1build3 [11.0 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libencode-locale-perl all 1.05-1.1 [11.8 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libfcgi-bin amd64 2.4.2-2build2 [11.2 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-template-perl all 2.97-1.1 [59.1 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libtimedate-perl all 2.3300-2 [34.0 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhttp-date-perl all 6.05-1 [9920 B]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libio-html-perl all 1.004-2 [15.4 kB]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liblwp-mediatypes-perl all 6.04-1 [19.5 kB]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhttp-message-perl all 6.36-1 [76.8 kB]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mecab-utils amd64 0.996-14build9 [4850 B]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mecab-ipadic all 2.7.0-20070801+main-3 [6718 kB]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mecab-ipadic-utf8 all 2.7.0-20070801+main-3 [4384 B]
```

i-013268ede701c383e (rds)

PublicIPs: 35.154.204.95   PrivateIPs: 172.31.36.45

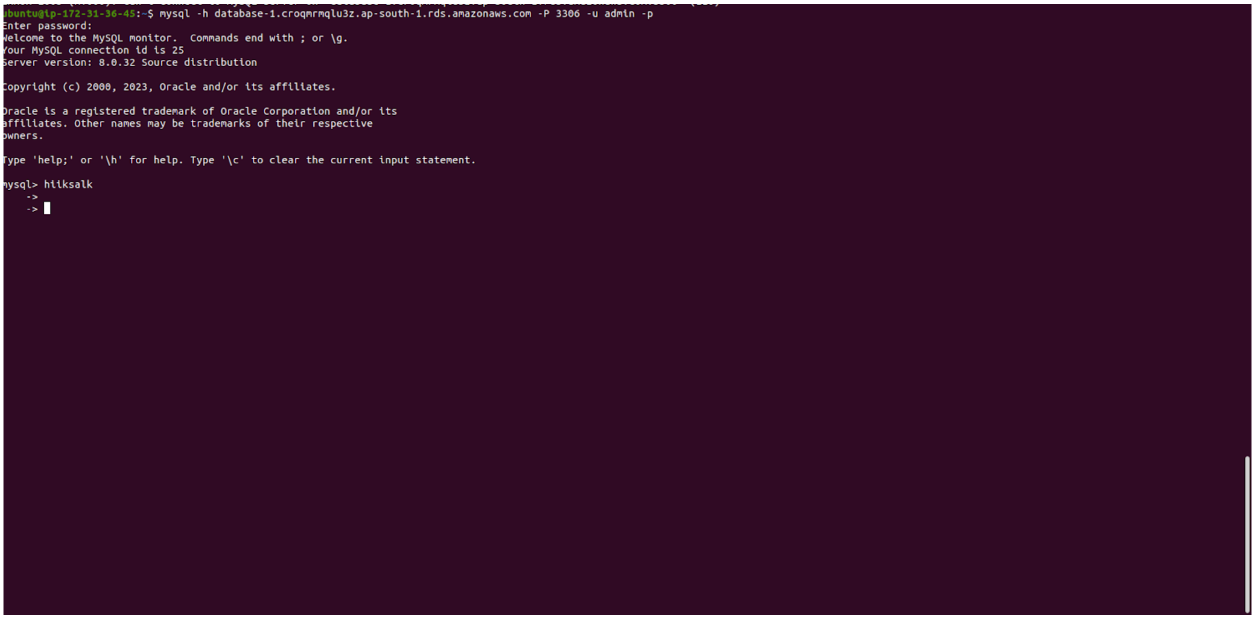
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

MySQL -h {endpoint} -P 3306 -u admin -p

mysql -h database-1.croqmrmqlu3z.ap-south-1.rds.amazonaws.com -P 3306 -u admin -p
If some times the output wont come, please check the connection of the EC2, connection in RDS > ACtions > setup EC2 connection

Then run the output

Now you can see the output in the below image,



```
ubuntu@ip-172-31-36-45:~$ mysql -h database-1.croqmrmqlu3z.ap-south-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 8.0.32 Source distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> hiiksalk
    ->
    ->
```