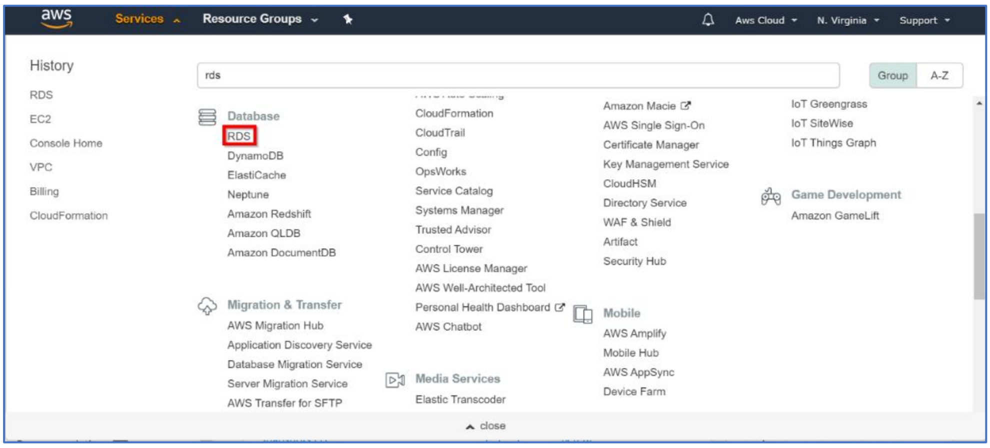
Getting Started with AWS - Amazon Relational Database Service (RDS)

Set up, operate, and scale a relational database in the cloud with just a few clicks.

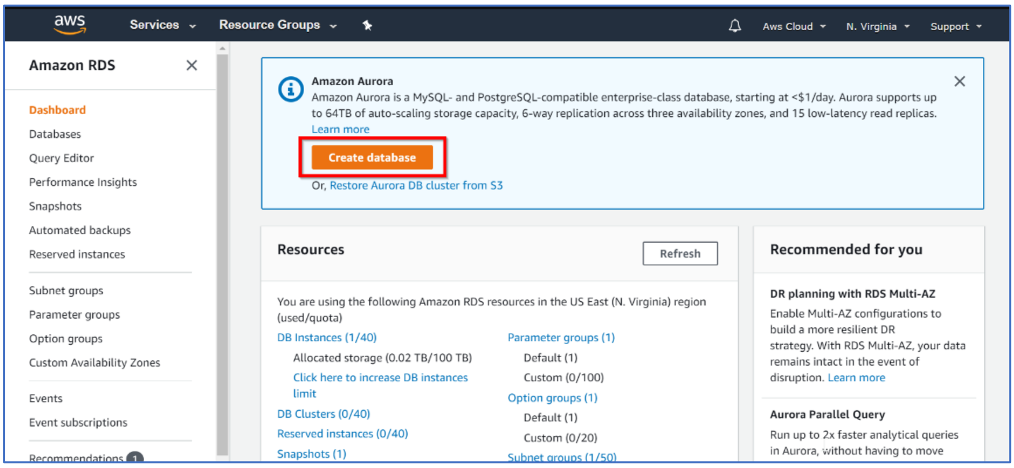
Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

Amazon RDS is available on several database instance types - optimized for memory, performance or I/O - and provides you with six familiar database engines to choose from, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server. You can use the AWS Database Migration Service to easily migrate or replicate your existing databases to Amazon RDS.

Step 1: Click on RDS service in AWS to Create Your Amazon Relational Database Service



Step 2: Click on Create database



Amazon Aurora

Amazon Aurora is a MySQL- and PostgreSQL-compatible enterprise-class database, starting at <$1/day. Aurora supports up to 64TB of auto-scaling storage capacity, 6-way replication across three availability zones, and 15 low-latency read replicas. Learn more

Or, Restore Aurora DB cluster from S3

Resources

You are using the following Amazon RDS resources in the US East (N. Virginia) region (used/quota)

DB Instances (1/40)
Allocated storage (0.02 TB/100 TB)
Click here to increase DB instances limit
DB Clusters (0/40)
Reserved instances (0/40)
Snapshots (1)

Parameter groups (1)
Default (1)
Custom (0/100)
Option groups (1)
Default (1)
Custom (0/20)
Subnet groups (1/50)

Recommended for you

DR planning with RDS Multi-AZ

Enable Multi-AZ configurations to build a more resilient DR strategy. With RDS Multi-AZ, your data remains intact in the event of disruption. Learn more

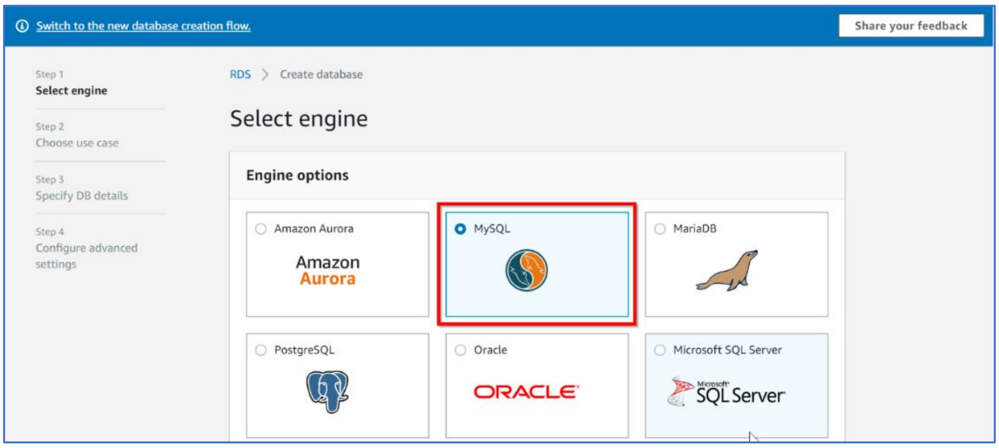
Aurora Parallel Query

Run up to 2x faster analytical queries in Aurora, without having to move

Step 3: Select Engine

Select your desired engine for relational database.

For ex: - In this case we are selecting MySQL database.



ⓘ Switch to the new database creation flow.

Share your feedback

Step 1
Select engine

Step 2
Choose use case

Step 3
Specify DB details

Step 4
Configure advanced settings

RDS > Create database

Select engine

Engine options

○ Amazon Aurora

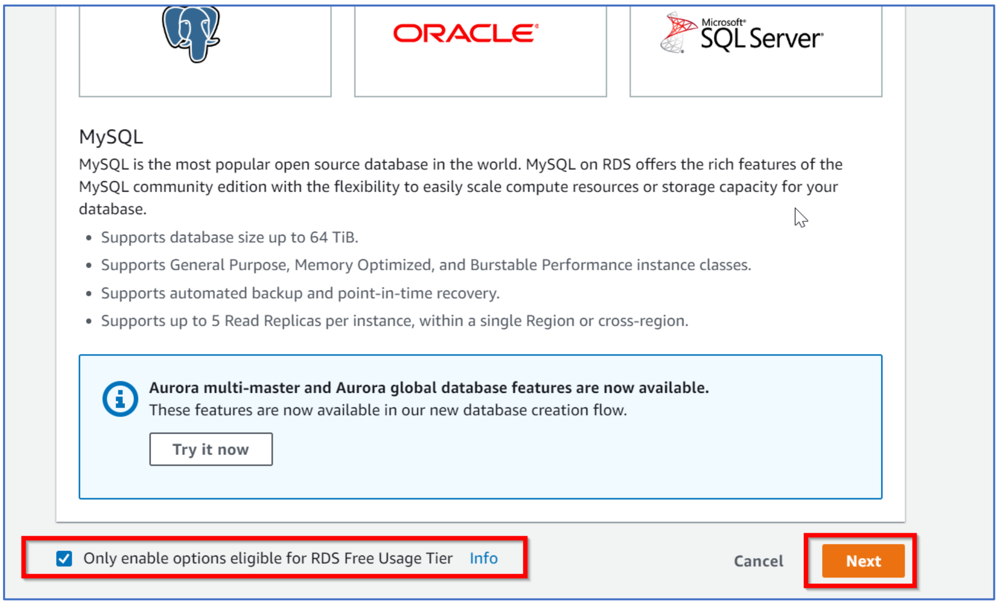
● MySQL

○ 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 5 Read Replicas per instance, within a single Region or cross-region.

Aurora multi-master and Aurora global database features are now available.
These features are now available in our new database creation flow.

Try it now

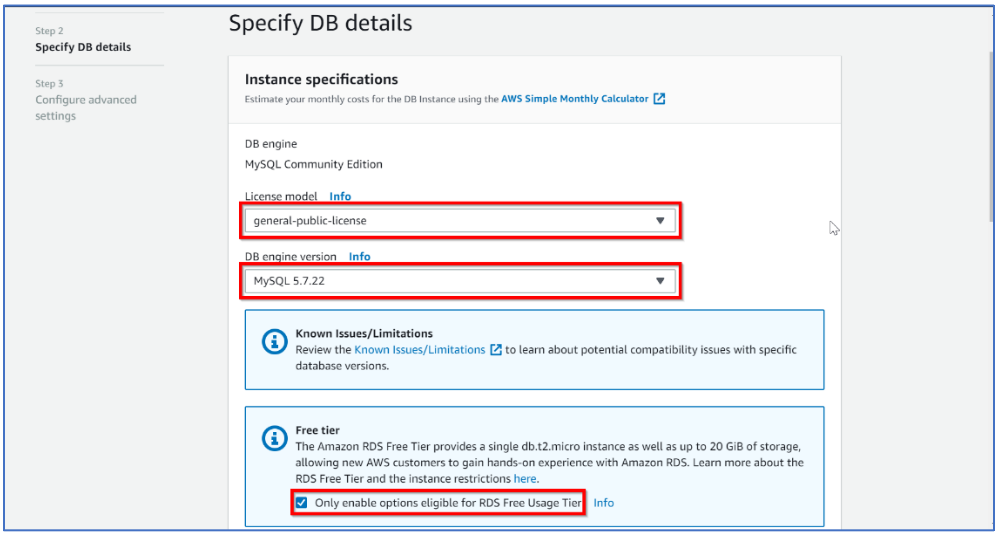
☑ Only enable options eligible for RDS Free Usage Tier Info

Cancel Next

(*Make sure you’ve checked Only enable options eligible for RDS Free Usage Tier*)

And then Hit NEXT

Step 4: Specify DB Details



Step 2
Specify DB details

Step 3
Configure advanced settings

Specify DB details

Instance specifications

Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator

DB engine
MySQL Community Edition

License model Info
general-public-license

DB engine version Info
MySQL 5.7.22

Known Issues/Limitations
Review the Known Issues/Limitations to learn about potential compatibility issues with specific database versions.

Free tier
The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GiB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions here.

☑ Only enable options eligible for RDS Free Usage Tier Info

Step 5: Configure settings of DB instance

Here we’ve to name the

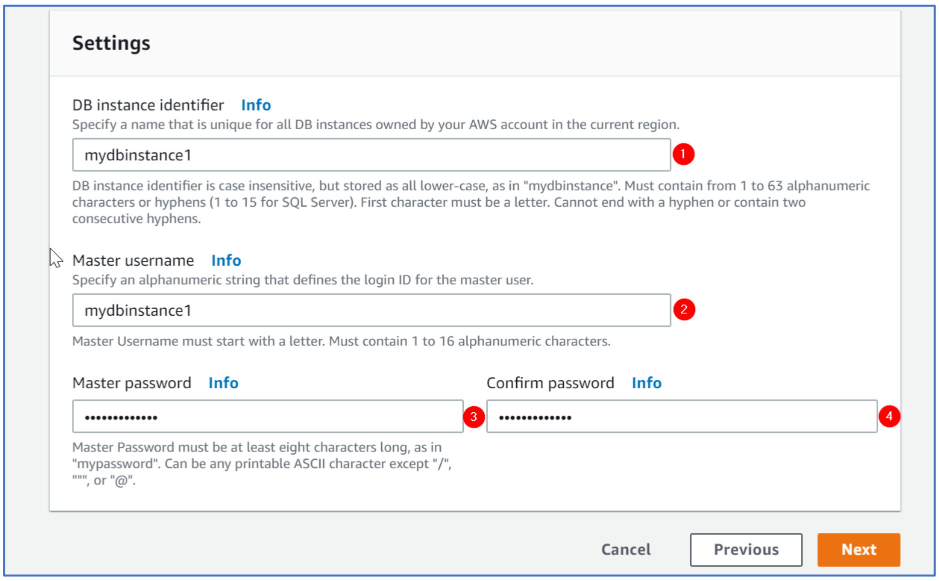
1) DB instance identifier

2) Master username

3)Master password

4)Confirm password

And then Hit NEXT



Settings

DB instance identifier Info
Specify a name that is unique for all DB instances owned by your AWS account in the current region.
mydbinstance1 1
DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance". Must contain from 1 to 63 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Cannot end with a hyphen or contain two consecutive hyphens.

Master username Info
Specify an alphanumeric string that defines the login ID for the master user.
mydbinstance1 2
Master Username must start with a letter. Must contain 1 to 16 alphanumeric characters.

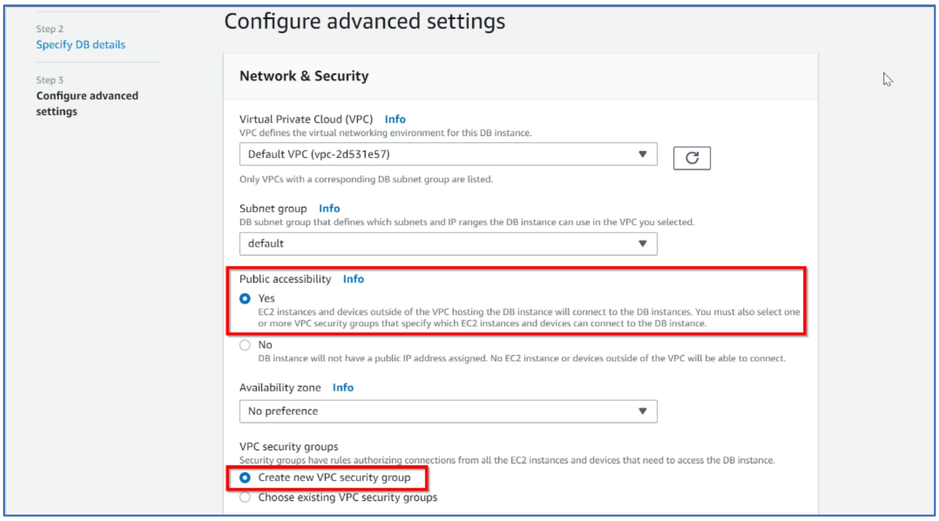
Master password Info
••••••••••••• 3
Master Password must be at least eight characters long, as in "mypassword". Can be any printable ASCII character except "/", """, or "@".

Confirm password Info
••••••••••••• 4

Step 6: Configure advanced settings

Make sure Public accessibility is enabled Yes.

We’ll create new VPC security group for our DB



Step 2
Specify DB details

Step 3
Configure advanced settings

Configure advanced settings

Network & Security

Virtual Private Cloud (VPC) Info
VPC defines the virtual networking environment for this DB instance.
Default VPC (vpc-2d531e57)
Only VPCs with a corresponding DB subnet group are listed.

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

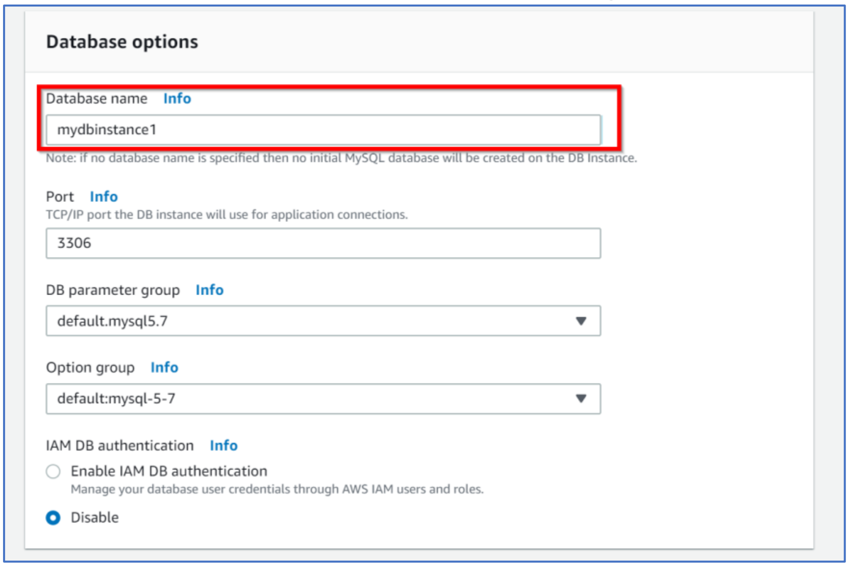
Public accessibility Info
● Yes
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.
○ No
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

Availability zone Info
No preference

VPC security groups
Security groups have rules authorizing connections from all the EC2 instances and devices that need to access the DB instance.
● Create new VPC security group
○ Choose existing VPC security groups

Step 7: Configure Database options

Name the Database and leave the rest other settings in default.



Database options

Database name Info
mydbinstance1
Note: if no database name is specified then no initial MySQL database will be created on the DB Instance.

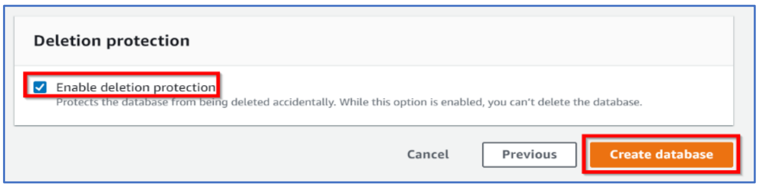
Port Info
TCP/IP port the DB instance will use for application connections.
3306

DB parameter group Info
default.mysql5.7

Option group Info
default:mysql-5-7

IAM DB authentication Info
○ Enable IAM DB authentication
Manage your database user credentials through AWS IAM users and roles.
● Disable

Enable Deletion protection for database



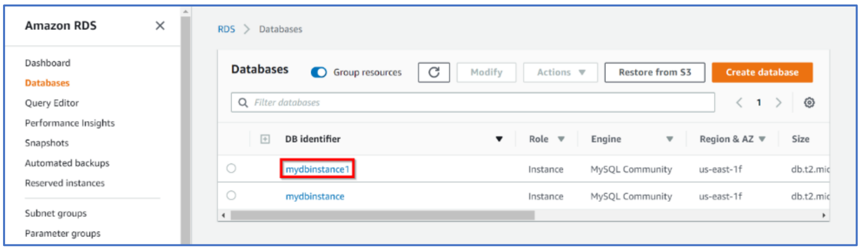
Deletion protection

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

Step 8: Amazon RDS Databases

Here we can see two databases are enlisted.

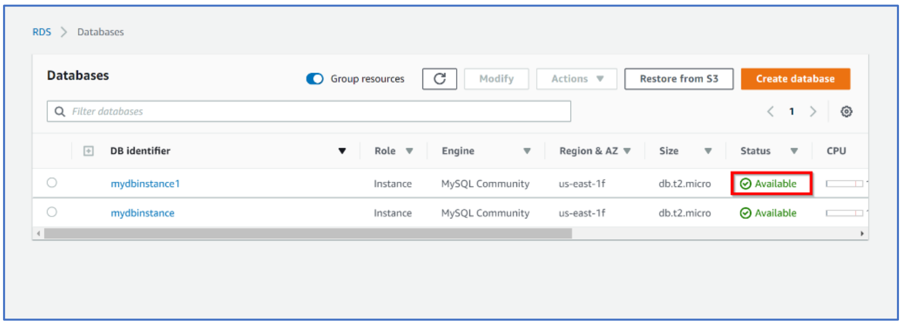
mydbinstance1 will start spinning.



	DB identifier	Role	Engine	Region & AZ	Size
○	mydbinstance1	Instance	MySQL Community	us-east-1f	db.t2.mic
○	mydbinstance	Instance	MySQL Community	us-east-1f	db.t2.mic

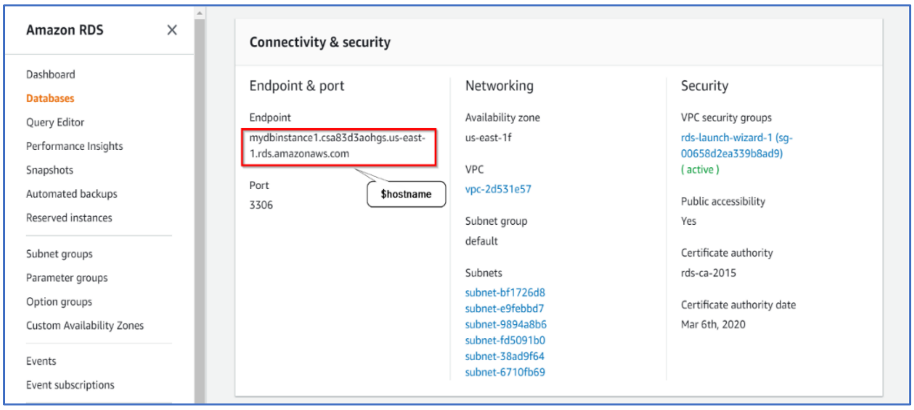
AWS says it might take 6 seconds - 15 mints to come into effect

After certain time interval it becomes Available



	DB identifier	Role	Engine	Region & AZ	Size	Status	CPU
○	mydbinstance1	Instance	MySQL Community	us-east-1f	db.t2.micro	✓ Available	
○	mydbinstance	Instance	MySQL Community	us-east-1f	db.t2.micro	✓ Available	

After clicking on mydbinstance1 it'll show the info regarding the DB



Connectivity & security

Endpoint & port

Endpoint
mydbinstance1.csa83d3aohgs.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability zone
us-east-1f

VPC
vpc-2d531e57

Subnet group
default

Subnets
subnet-bf1726d8
subnet-e9febbd7
subnet-9894a8b6
subnet-fd5091b0
subnet-38ad9f64
subnet-6710fb69

Security

VPC security groups
rds-launch-wizard-1 (sg-00658d2ea339b8ad9)
(active)

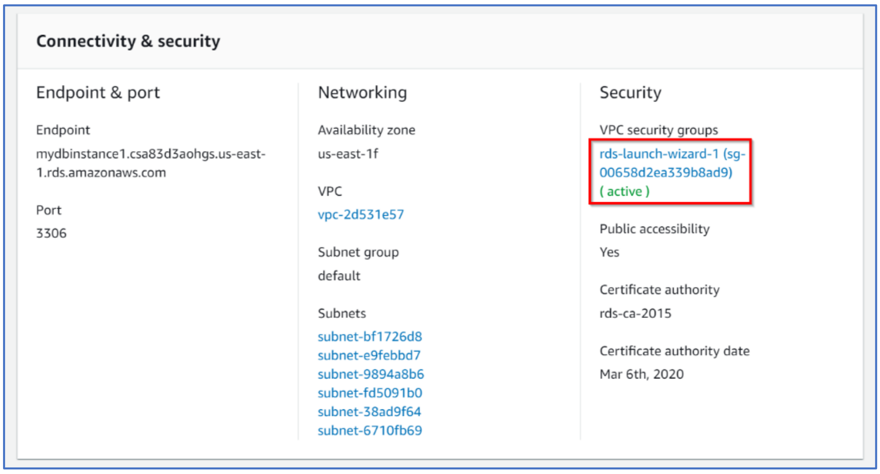
Public accessibility
Yes

Certificate authority
rds-ca-2015

Certificate authority date
Mar 6th, 2020

Step 9: Configuring mydbinstance1 database VPC Security groups

Click on VPC Security groups



Connectivity & security

Endpoint & port

Endpoint
mydbinstance1.csa83d3aohgs.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability zone
us-east-1f

VPC
vpc-2d531e57

Subnet group
default

Subnets
subnet-bf1726d8
subnet-e9febbd7
subnet-9894a8b6
subnet-fd5091b0
subnet-38ad9f64
subnet-6710fb69

Security

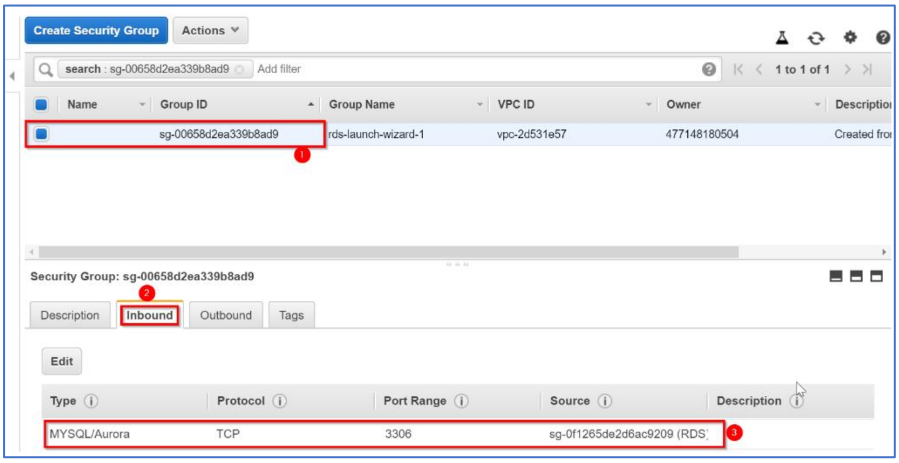
VPC security groups
rds-launch-wizard-1 (sg-00658d2ea339b8ad9)
(active)

Public accessibility
Yes

Certificate authority
rds-ca-2015

Certificate authority date
Mar 6th, 2020

Select security group and make sure MYSQL/Aurora and port range 3306 is present from Inbound



Name	Group ID	Group Name	VPC ID	Owner	Descriptio
	sg-00658d2ea339b8ad9	rds-launch-wizard-1	vpc-2d531e57	477148180504	Created fro

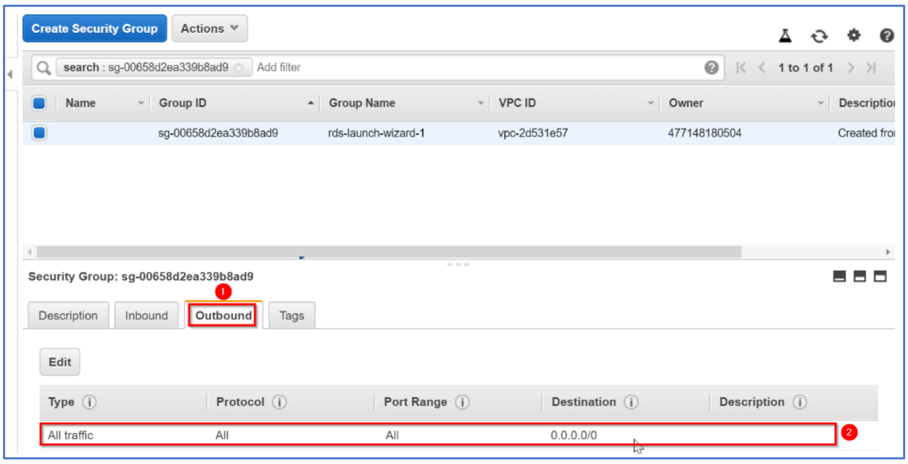
Security Group: sg-00658d2ea339b8ad9

Description | Inbound | Outbound | Tags

Edit

Type	Protocol	Port Range	Source	Description
MYSQL/Aurora	TCP	3306	sg-0f1265de2d6ac9209 (RDS)	

From Outbound section All traffic should be enabled



Name	Group ID	Group Name	VPC ID	Owner	Descriptio
	sg-00658d2ea339b8ad9	rds-launch-wizard-1	vpc-2d531e57	477148180504	Created fro

Security Group: sg-00658d2ea339b8ad9

Description | Inbound | Outbound | Tags

Edit

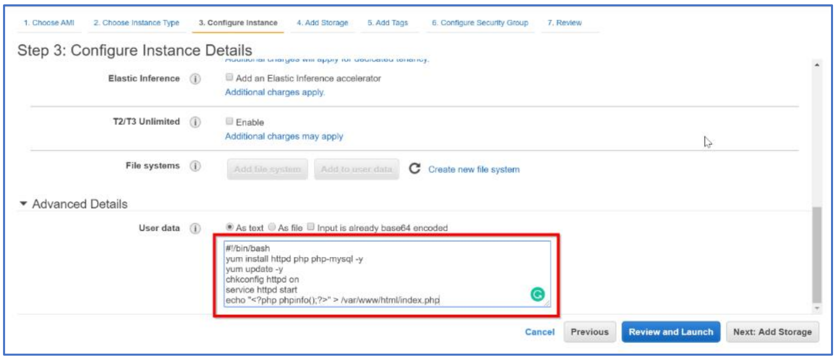
Type	Protocol	Port Range	Destination	Description
All traffic	All	All	0.0.0.0/0	

Step 10: Launch RDS instance

Launch an AMI instance

Configure the user data using text script

```
#!/bin/bash

yum install httpd php php-mysql -y

yum update -y

chkconfig httpd on

service httpd start

echo "<?php phpinfo();?>" > /var/www/html/index.php
```



Step 3: Configure Instance Details

Elastic Inference: □ Add an Elastic Inference accelerator
Additional charges apply.

T2/T3 Unlimited: □ Enable
Additional charges may apply

File systems: Add file system | Add to user data | Create new file system

Advanced Details

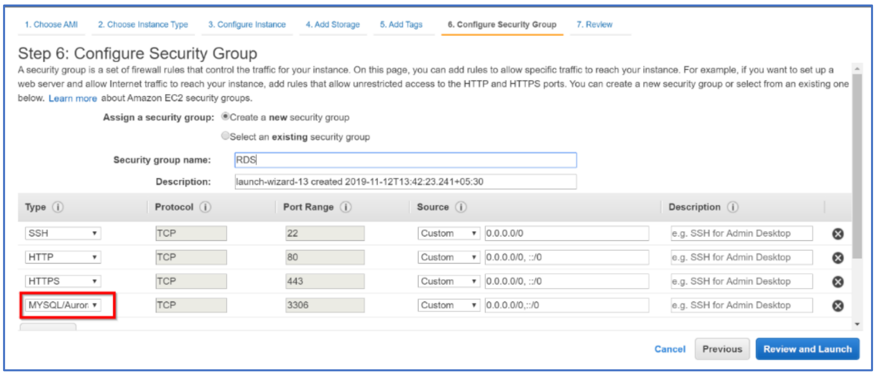
User data: ● As text ○ As file □ Input is already base64 encoded

```
#!/bin/bash
yum install httpd php php-mysql -y
yum update -y
chkconfig httpd on
service httpd start
echo "<?php phpinfo();?>" > /var/www/html/index.php
```

Step 11: Configure Security Group for RDS instance

Here we’ve to Create a new security group name RDS

Make Sure to add port range **(3306,80,22)** and Launch



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

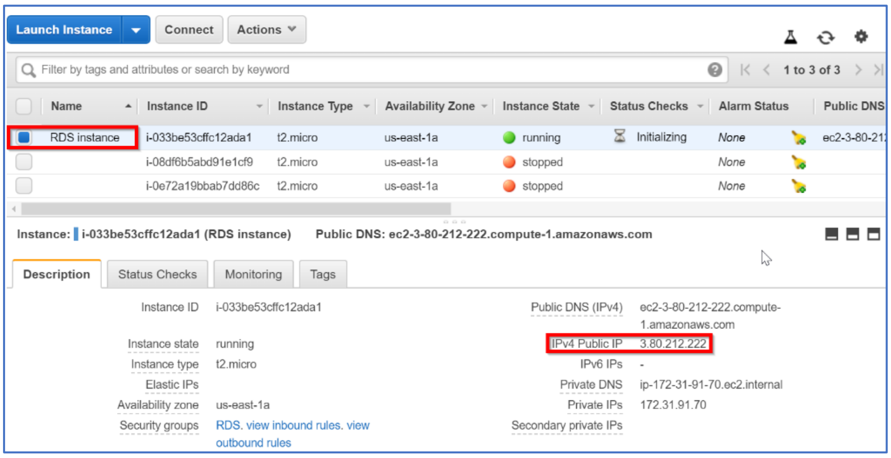
Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: RDS

Description: launch-wizard-13 created 2019-11-12T13:42:23.241+05:30

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
MYSQL/Auror	TCP	3306	Custom	0.0.0.0/0,::/0	e.g. SSH for Admin Desktop

Step 12: Log in to RDS instance using Putty Client



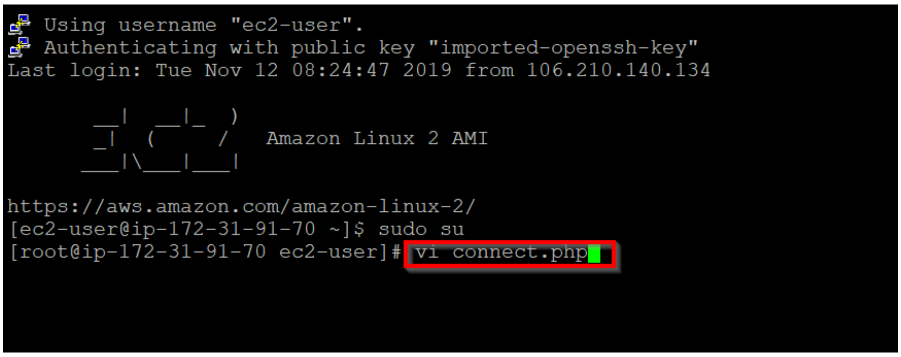
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
RDS instance	i-033be53cffc12ada1	t2.micro	us-east-1a	running	Initializing	None	ec2-3-80-21
	i-08df6b5abd91e1cf9	t2.micro	us-east-1a	stopped		None	
	i-0e72a19bbab7dd86c	t2.micro	us-east-1a	stopped		None	

Instance: i-033be53cffc12ada1 (RDS instance) Public DNS: ec2-3-80-212-222.compute-1.amazonaws.com

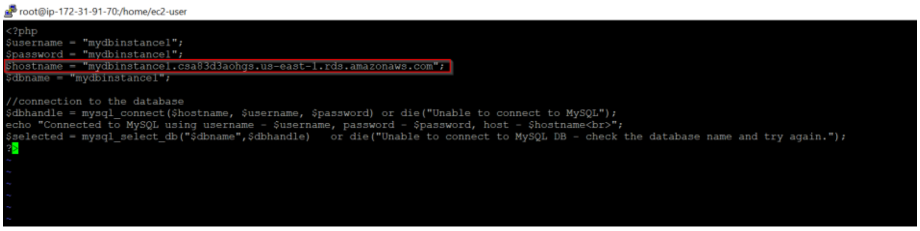
Instance ID	i-033be53cffc12ada1	Public DNS (IPv4)	ec2-3-80-212-222.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	3.80.212.222
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-91-70.ec2.internal
Availability zone	us-east-1a	Private IPs	172.31.91.70
Security groups	RDS. view inbound rules. view outbound rules	Secondary private IPs	

Open EC2 System and go to the path below and create connect.php file

(/var/www/html/) and copy the content



```
 Using username "ec2-user".
 Authenticating with public key "imported-openssh-key"
Last login: Tue Nov 12 08:24:47 2019 from 106.210.140.134

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-91-70 ~]$ sudo su
[root@ip-172-31-91-70 ec2-user]# vi connect.php
```



root@ip-172-31-91-70:/home/ec2-user

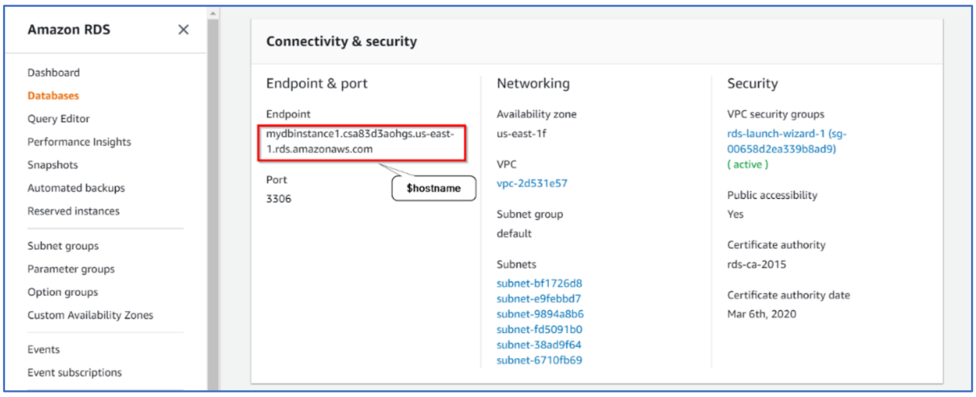
```
<?php
$username = "mydbinstance1";
$password = "mydbinstance1";
$hostname = "mydbinstance1.csa83d3aohgs.us-east-1.rds.amazonaws.com";
$dbname = "mydbinstance1";

//connection to the database
$dbhandle = mysql_connect($hostname, $username, $password) or die("Unable to connect to MySQL");
echo "Connected to MySQL using username - $username, password - $password, host - $hostname<br>";
$selected = mysql_select_db("$dbname",$dbhandle)   or die("Unable to connect to MySQL DB - check the database name and try again.");
?>
~
~
~
~
~
~
```

a) Enable Super user using **sudo su**

b) Create a php file named **connect.php** using vi editor and enter Db connection code

```
<?php

$username = "mydbinstance1"; // Master username

$password = "mydbinstance1"; // Master Password

$hostname = "mydbinstance1.csa83d3aohgs.us-east-1.rds.amazonaws.com"; // Endpoint

$dbname = "mydbinstance1";//Database Name


//connection to the database

$dbhandle = mysql_connect($hostname, $username, $password) or die("Unable to connect to MySQL");

echo "Connected to MySQL using username - $username, password - $password, host - $hostname<br>";

$selected = mysql_select_db("$dbname",$dbhandle)   or die("Unable to connect to MySQL DB - check the database name and try again.");

?>

:wq
```

Host Name is **Endpoint** of RDS Database



Connectivity & security

Endpoint & port

Endpoint
mydbinstance1.csa83d3aohgs.us-east-1.rds.amazonaws.com

Port
3306

$hostname

Networking

Availability zone
us-east-1f

VPC
vpc-2d531e57

Subnet group
default

Subnets
subnet-bf1726d8
subnet-e9febbd7
subnet-9894a8b6
subnet-fd5091b0
subnet-38ad9f64
subnet-6710fb69

Security

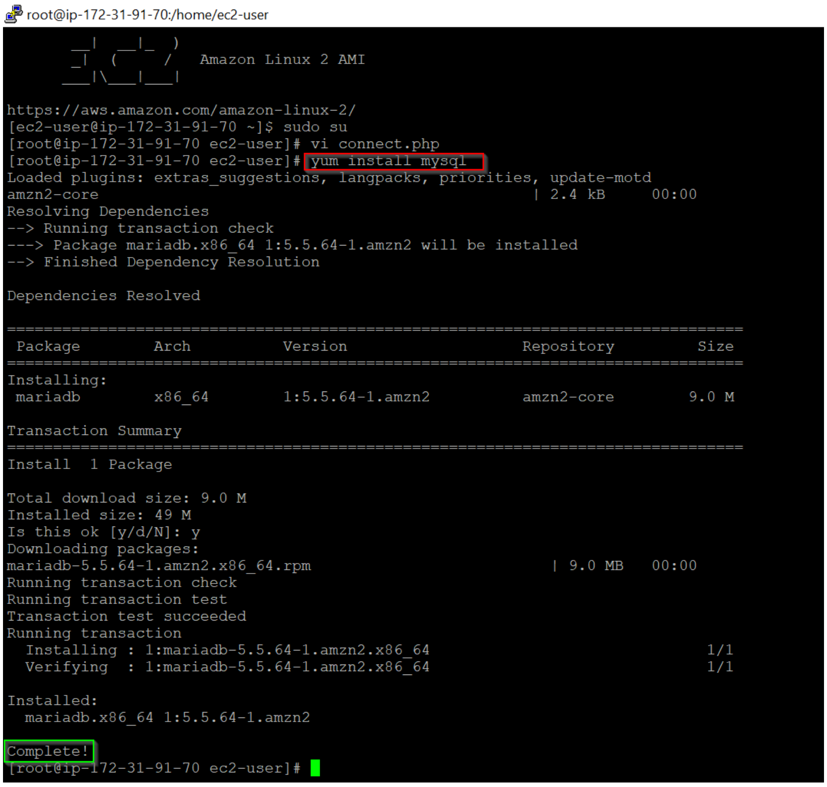
VPC security groups
rds-launch-wizard-1 (sg-00658d2ea339b8ad9)
(active)

Public accessibility
Yes

Certificate authority
rds-ca-2015

Certificate authority date
Mar 6th, 2020

Step 13: Install MySQL on AMI



root@ip-172-31-91-70:/home/ec2-user

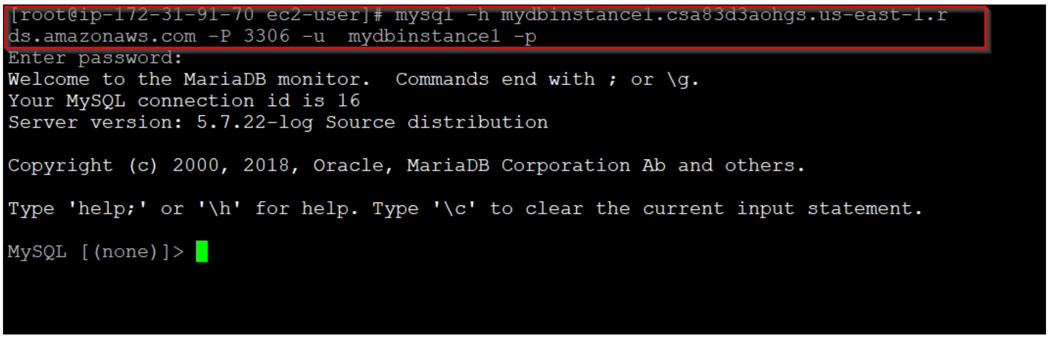
```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-91-70 ~]$ sudo su
[root@ip-172-31-91-70 ec2-user]# vi connect.php
[root@ip-172-31-91-70 ec2-user]# yum install mysql
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                               | 2.4 kB     00:00
Resolving Dependencies
--> Running transaction check
---> Package mariadb.x86_64 1:5.5.64-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package         Arch          Version                    Repository        Size
================================================================================
Installing:
 mariadb         x86_64        1:5.5.64-1.amzn2           amzn2-core       9.0 M

Transaction Summary
================================================================================
Install  1 Package

Total download size: 9.0 M
Installed size: 49 M
Is this ok [y/d/N]: y
Downloading packages:
mariadb-5.5.64-1.amzn2.x86_64.rpm                          | 9.0 MB   00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:mariadb-5.5.64-1.amzn2.x86_64                              1/1
  Verifying  : 1:mariadb-5.5.64-1.amzn2.x86_64                              1/1

Installed:
  mariadb.x86_64 1:5.5.64-1.amzn2

Complete!
[root@ip-172-31-91-70 ec2-user]#
```

Step 14: Connect to RDS Database

Syntax: mysql -h "Host Name"-P 3306 -u "databasename" -p

For ex: mysql -h mydbinstance1.csa83d3aohgs.us-east-1.rds.amazonaws.com -P 3306 -u mydbinstance1 -p

Enter password: •••••••••



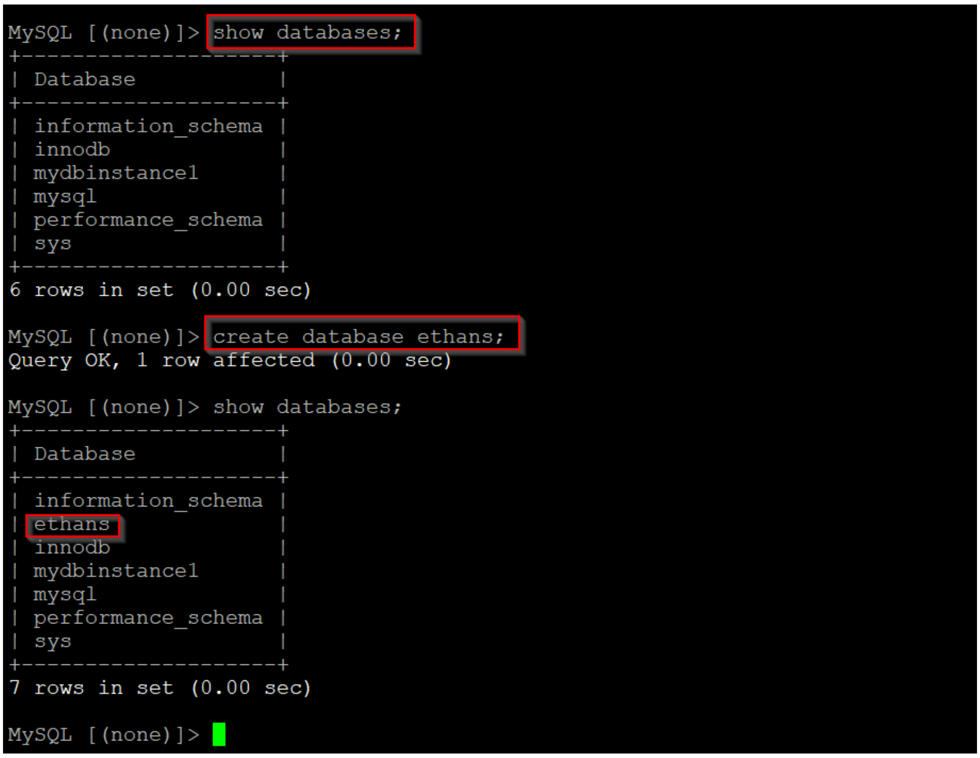
```
[root@ip-172-31-91-70 ec2-user]# mysql -h mydbinstance1.csa83d3aohgs.us-east-1.r
ds.amazonaws.com -P 3306 -u  mydbinstance1 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.7.22-log Source distribution

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

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

MySQL [(none)]>
```

Query: show databases;

This command will enlist the Database

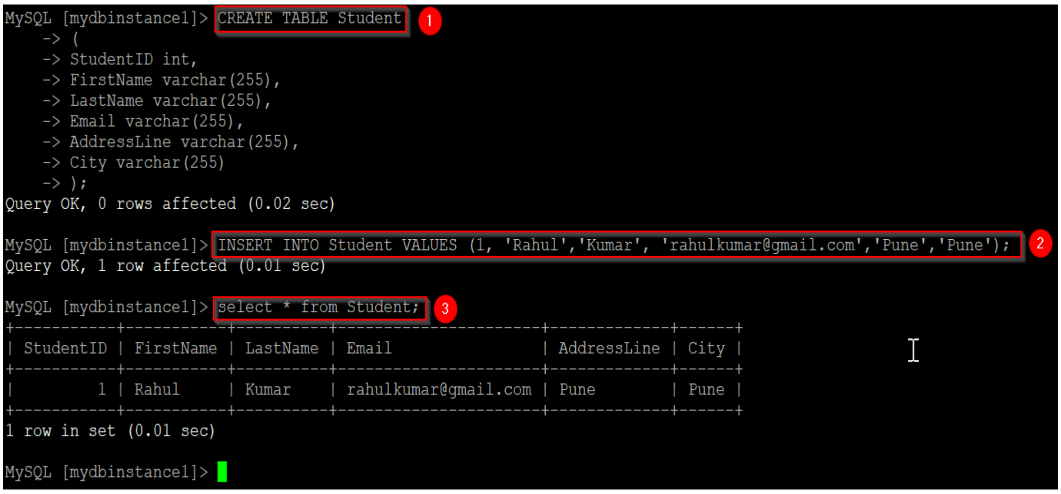
Query: create database ethans;

This command will create new database ethans



```
MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| mydbinstance1      |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)

MySQL [(none)]> create database ethans;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| ethans             |
| innodb             |
| mydbinstance1      |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
7 rows in set (0.00 sec)

MySQL [(none)]>
```

Step 15: Insert data into mydbinstance1 Database



```
MySQL [mydbinstance1]> CREATE TABLE Student
    -> (
    -> StudentID int,
    -> FirstName varchar(255),
    -> LastName varchar(255),
    -> Email varchar(255),
    -> AddressLine varchar(255),
    -> City varchar(255)
    -> );
Query OK, 0 rows affected (0.02 sec)

MySQL [mydbinstance1]> INSERT INTO Student VALUES (1, 'Rahul','Kumar', 'rahulkumar@gmail.com','Pune','Pune');
Query OK, 1 row affected (0.01 sec)

MySQL [mydbinstance1]> select * from Student;
+-----------+-----------+----------+----------------------+-------------+------+
| StudentID | FirstName | LastName | Email                | AddressLine | City |
+-----------+-----------+----------+----------------------+-------------+------+
|         1 | Rahul     | Kumar    | rahulkumar@gmail.com | Pune        | Pune |
+-----------+-----------+----------+----------------------+-------------+------+
1 row in set (0.01 sec)

MySQL [mydbinstance1]>
```

Done