

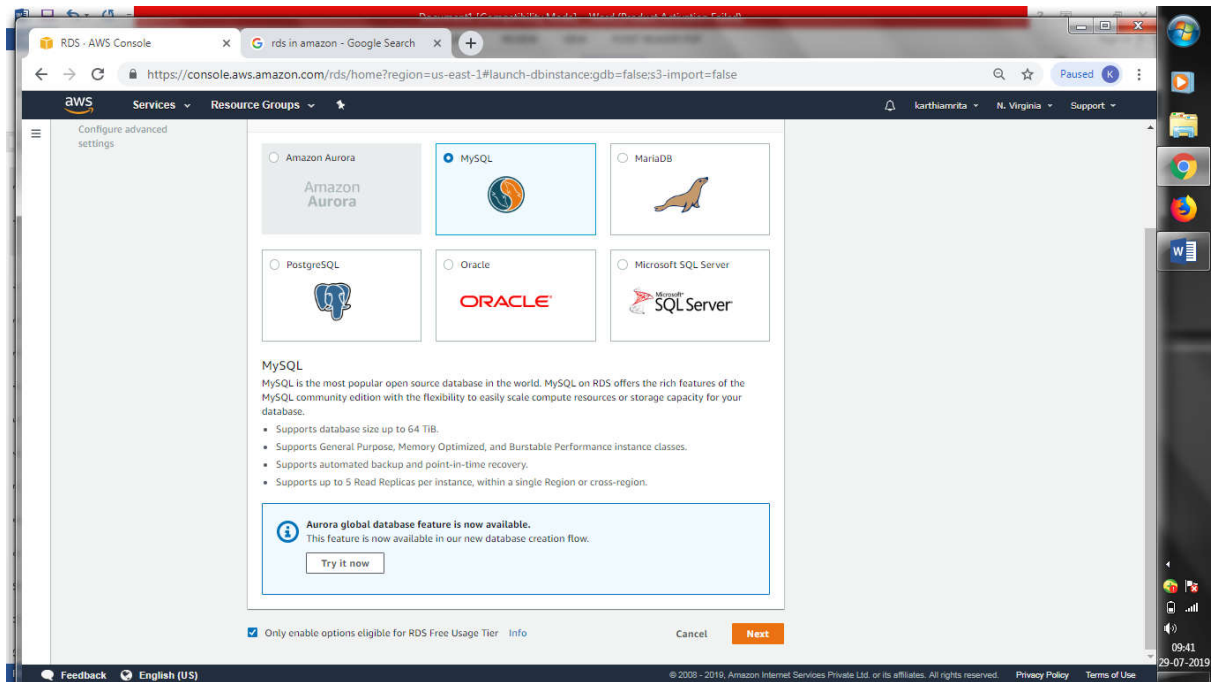
Create a Relational Database Service (RDS) in AWS and connect to RDS

Step 1:

Choose RDS from AWS service

Step 2:

Choose create database – MYSQL and enable free tier option



Step 3:

In the specify DB details tab: < Store the instance name , username and password details as they are needed to connect to the database >

Settings

DB instance identifier [Info](#)
Specify a name that is unique for all DB instances owned by your AWS account in the current region.

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.

Master Username must start with a letter. Must contain 1 to 16 alphanumeric characters.

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

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

[Cancel](#) [Previous](#) [Next](#)

Step 4: network settings

Network & Security

Virtual Private Cloud (VPC) [Info](#)
VPC defines the virtual networking environment for this DB instance.

Create new VPC ▼ ↺

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.

Create new DB Subnet Group ▼

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

Database option

Database options

Database name [Info](#)

PRODUCTDB

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

Leave all other settings to default

Step 5:

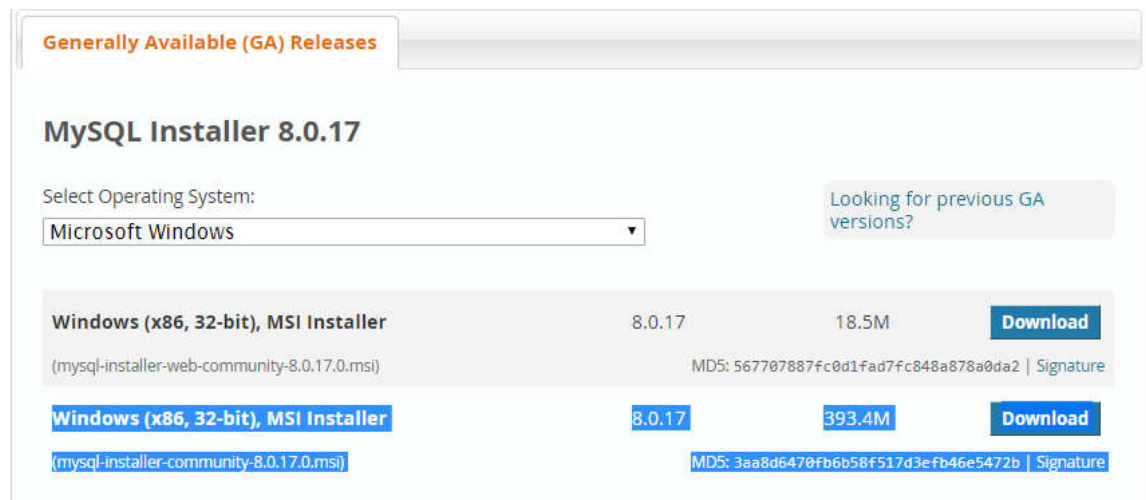
Create and view the database instance -productinstance- is available

Step 6:

Connect to the database using SQL-Workbench

Install MYSQL and MYSQL work bench

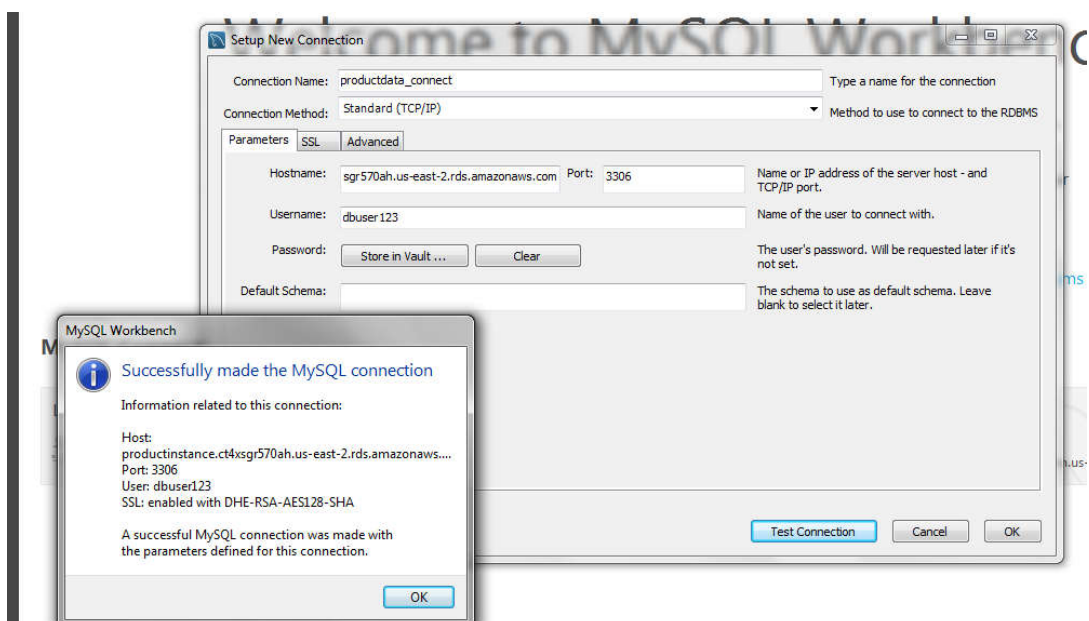
<https://dev.mysql.com/downloads/windows/installer/8.0.html>



Refer link for installation <https://www.youtube.com/watch?v=WuBcTJnluzo>

Step 7:

Launch MYSQL work bench and go to menu - Databases – Connect to databases.



Fill the details

1. Stored connection – productdata_connect
2. Hostname: endpoint details from dbinstance in AWS
3. Username: username – username entered during RDS creation on aws (step3)
4. Password: password entered during RDS creation on aws (step3)

Test the connection and connect to database.

Step 8:

Write queries in SQL workbench and update Database on AWS

Step 9:

Create a node program to connect to the mysql server

Step 1. create_db_inaws1.js

```
var mysql = require('mysql');

var con = mysql.createConnection({
  host: "productinstance.ct4xsgr570ah.us-east-2.rds.amazonaws.com",
  user: "dbuser123",
  password: "dbuser123",
  port: "3306"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  con.query("CREATE DATABASE mydb12", function (err, result) {
    if (err) throw err;
    console.log("Database created");
  });
});
```

Step 2 . install mysql in the folder using

```
npm install mysql --save
```

Step 3: To run in cmd using nodejs :

```
>> node create_db_inaws1.js
```

The screenshot shows a Windows Command Prompt window titled "C:\Windows\system32\cmd.exe - node create_db_inaws1.js". The window displays the output of running a Node.js script. At the top, there are two error messages from MySQL's protocol parser:

```
mysql\lib\protocol\Protocol.js:291:23)
    at Parser.parsePacket (E:\Cloud Computing\nodepgms-example\node_modules\mysql\lib\protocol\Parser.js:433:10)
    at Parser.write (E:\Cloud Computing\nodepgms-example\node_modules\mysql\lib\protocol\Parser.js:43:10)
```

Following these errors, the prompt shows a series of directory changes to "E:\Cloud Computing\nodepgms-example>". After several repetitions, the user enters the command "node create_db_inaws1.js". The script responds with "Connected!" and "Database created".

File Name	Date and Time	Type
Homegroup		
sql-first	29-07-2019 18:27	JScript Script File
table create	29-07-2019 18:38	JScript Script File