lab

lab title

AWS Relational Database Service (RDS)
V1.02

Course title

BackSpace Academy
AWS Certified Associate

BackSpace

Table of Contents

Contents

About the Lab

Please note that not all AWS services are supported in all regions. Please use the US-East-1 (North Virginia) region for this lab.

These lab notes are to support the hands on instructional videos of the AWS Relational Database Service (RDS) section of the AWS Certified Associate Course.

Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the lastest version with any updates or corrections.

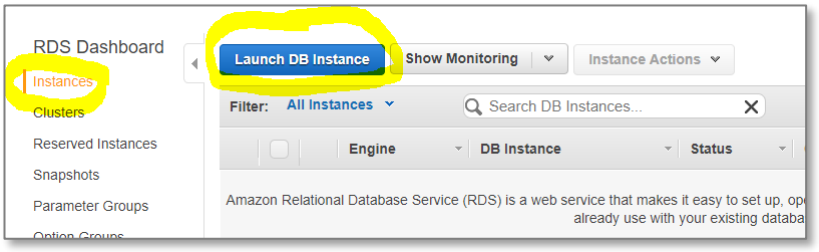
Deploying & Connecting to a MySQL Database Server

In this section, we will use the Relational Database Service to create a database. We will also connect in to the database.

From the AWS console select “RDS” from the Database services.

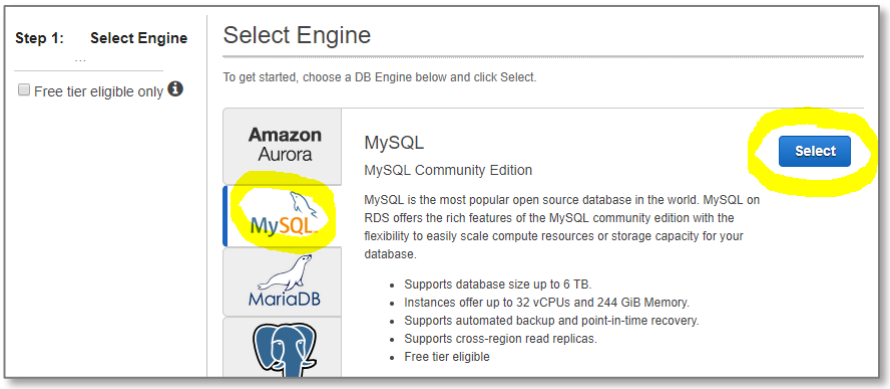
Select “instances”

Select “Launch DB Instance”

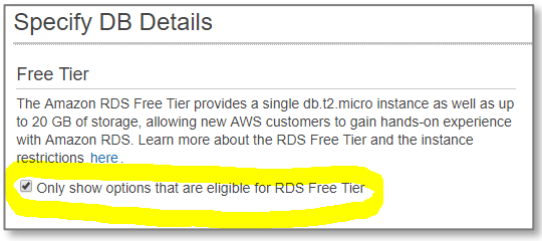


Select “Free tier eligible only”

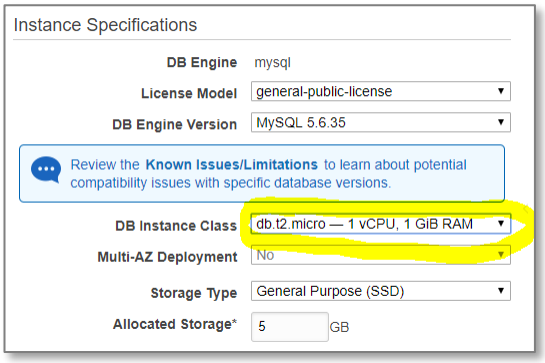
Select the MySQL Community Edition



Make sure “Only show options that are eligible for RDS Free Tier” is selected



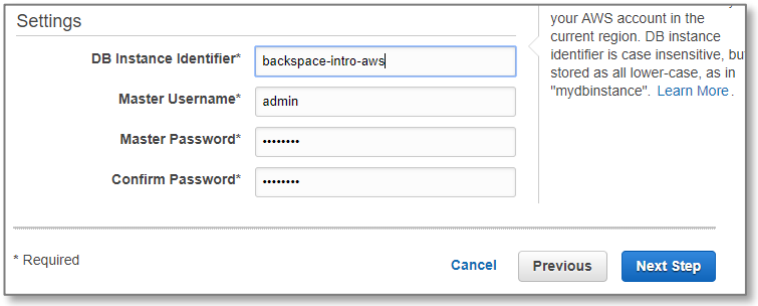
Select db.t2.micro instance class



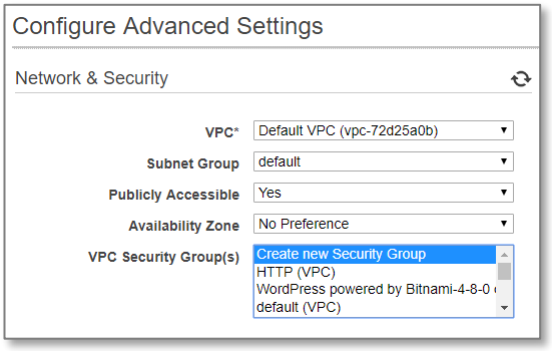
Give your instance a name/identifier.

Fill in a master username and password

Click “Next Step”

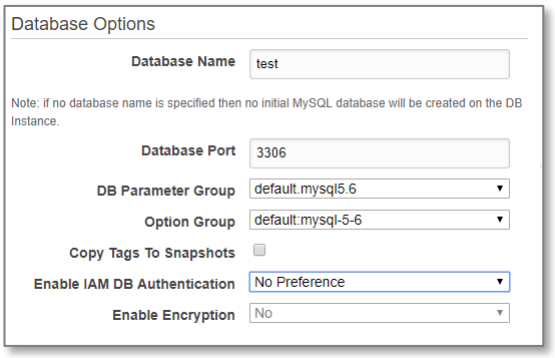


Leave settings for Network and Security as below.



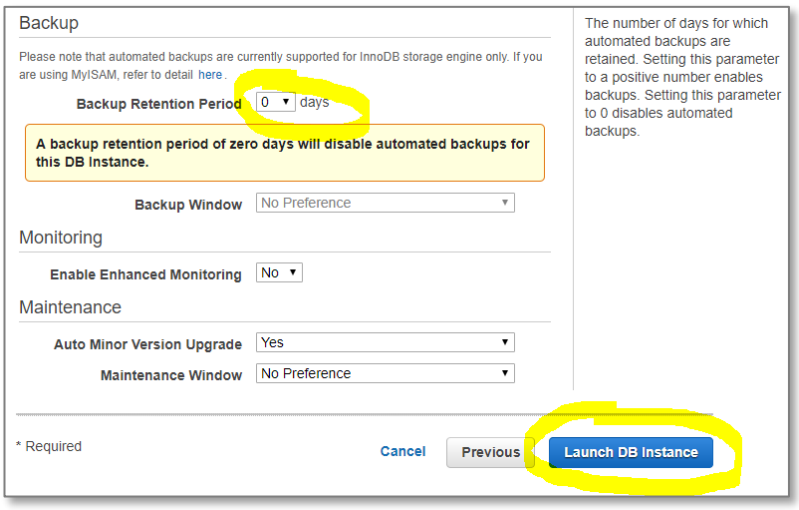
Enter a database name.

Leave other options default as below.

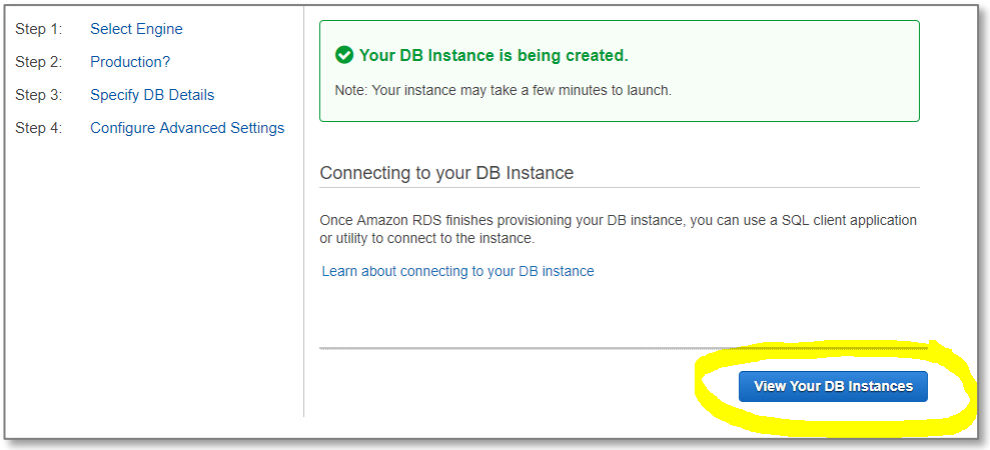


Change “Backup Retention Period” to disable automated backups.

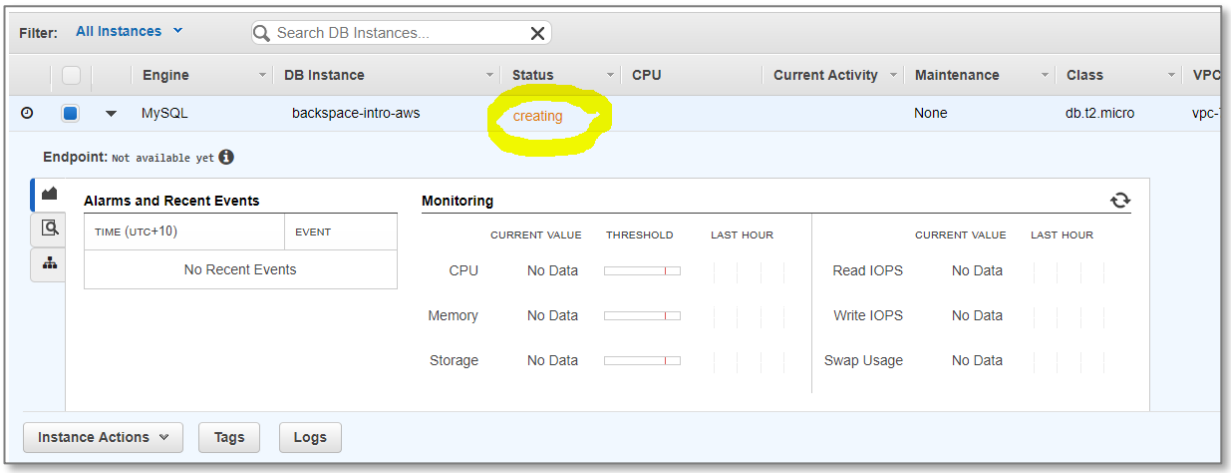
Click “Launch DB Instance”



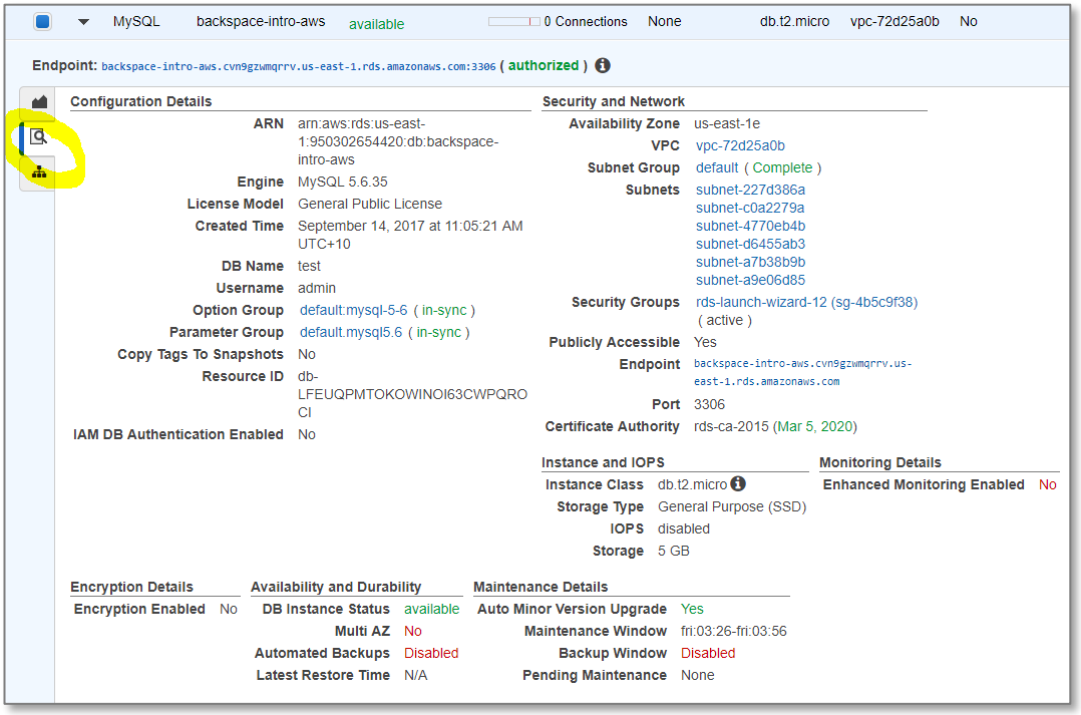
Click “View your DB Instances



Your instance will show status “creating”.



When you instance status changes to “available” you can view your configuration details which should look similar to below:

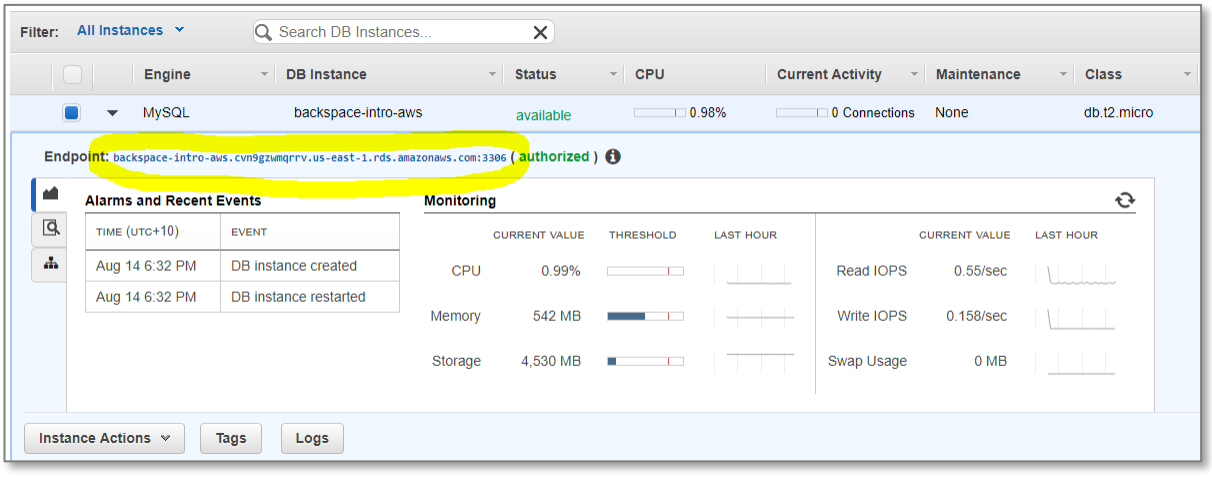


Connecting to your RDS Instance using the MySQL WorkBench

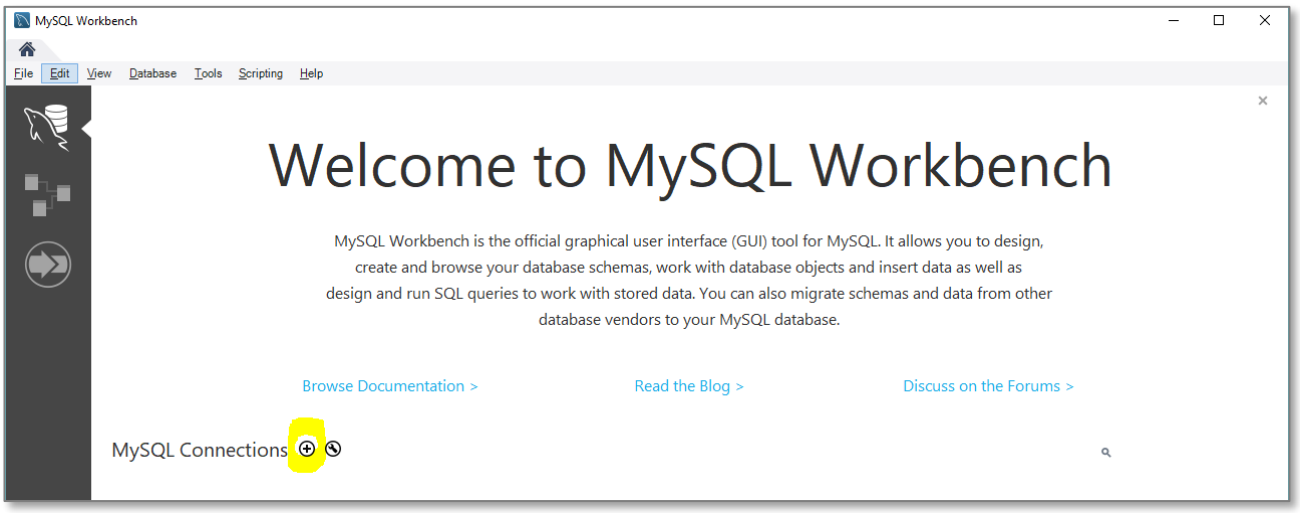
To connect to your MySQL Database you will need to download and install the MySQL Workbench from

https://dev.mysql.com/downloads/workbench/

When your instance status is “available”, copy the database server endpoint



Open the MySQL Workbench application click to add a new connection



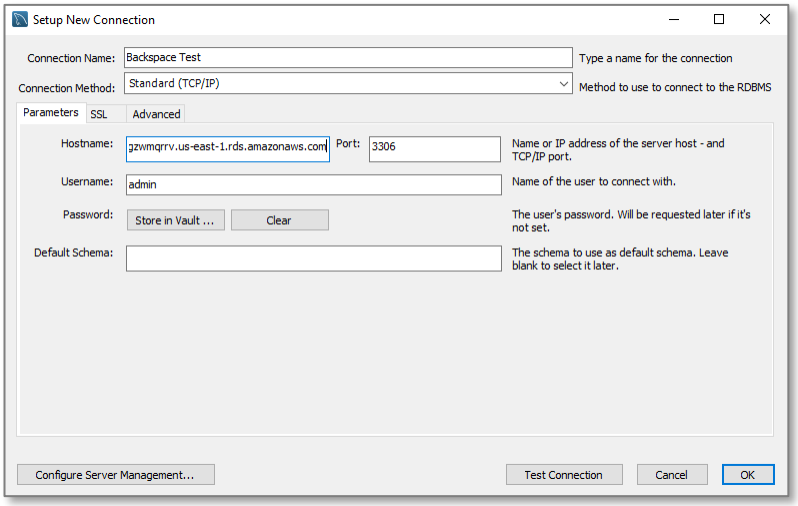
Give the connection a name.

The Hostname will be the RDS server endpoint with the ":3306" removed from the end.

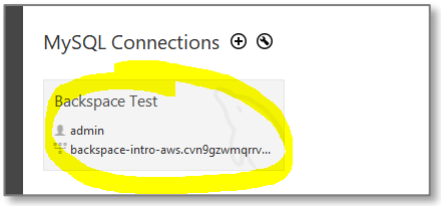
The port will be 3306.

The Username will be the master username we created in RDS (i.e. admin)

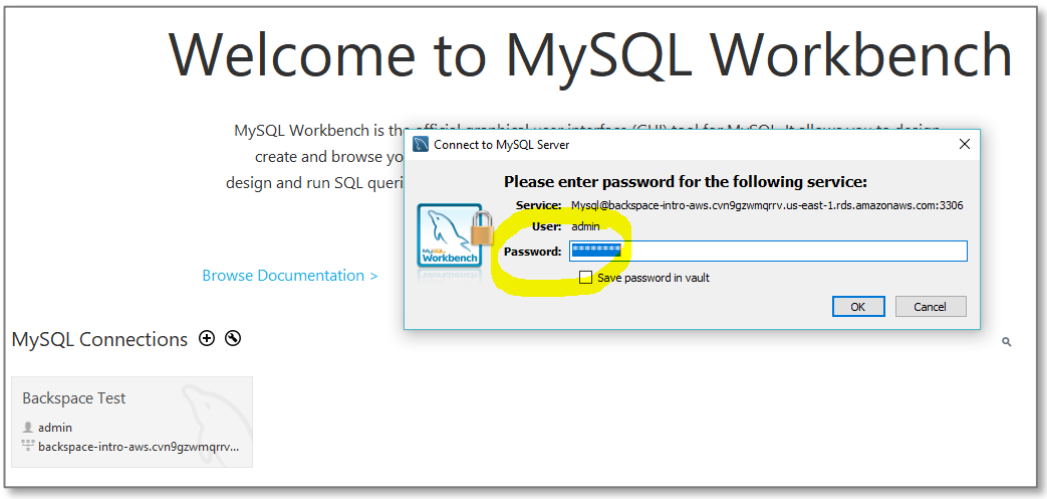
Click OK



Click on the Connection

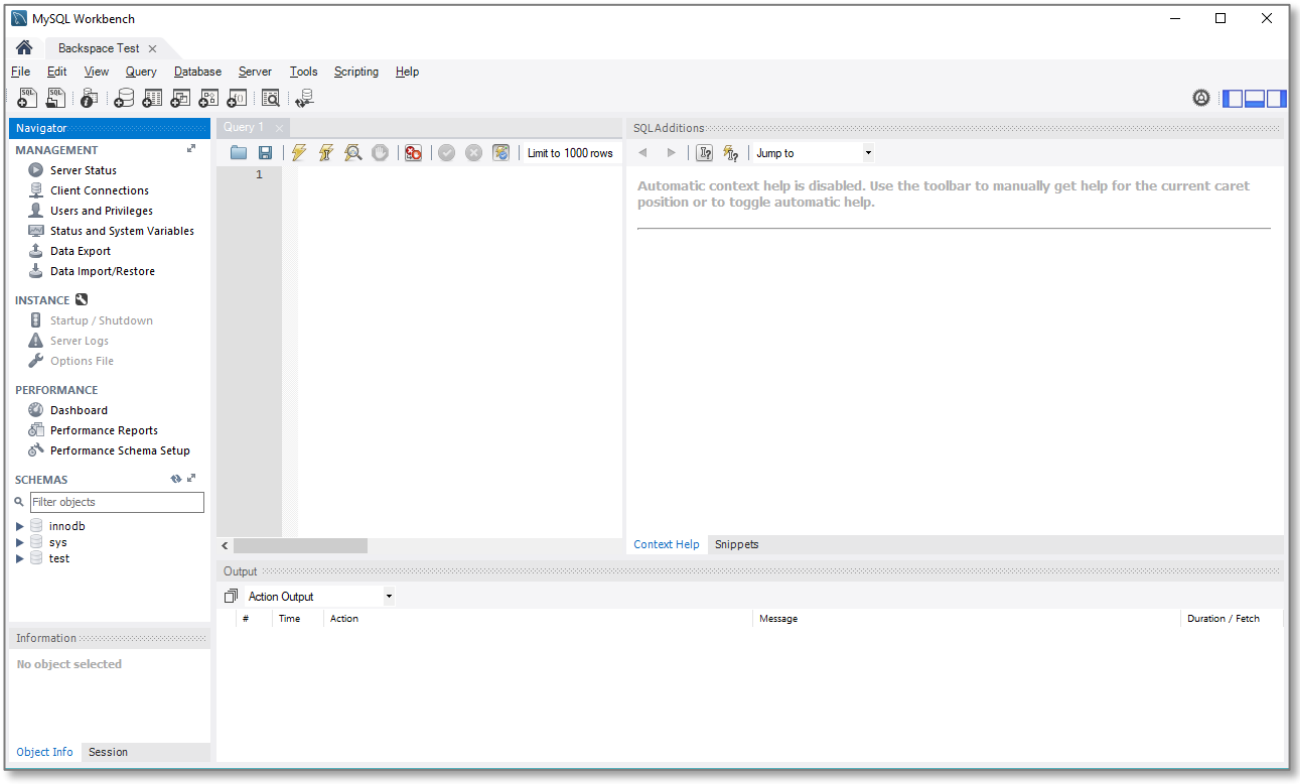


Enter the password you created in RDS for your master username

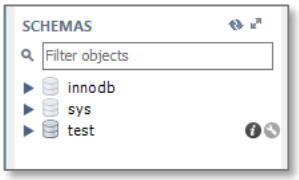


You will soon be connected to your database server

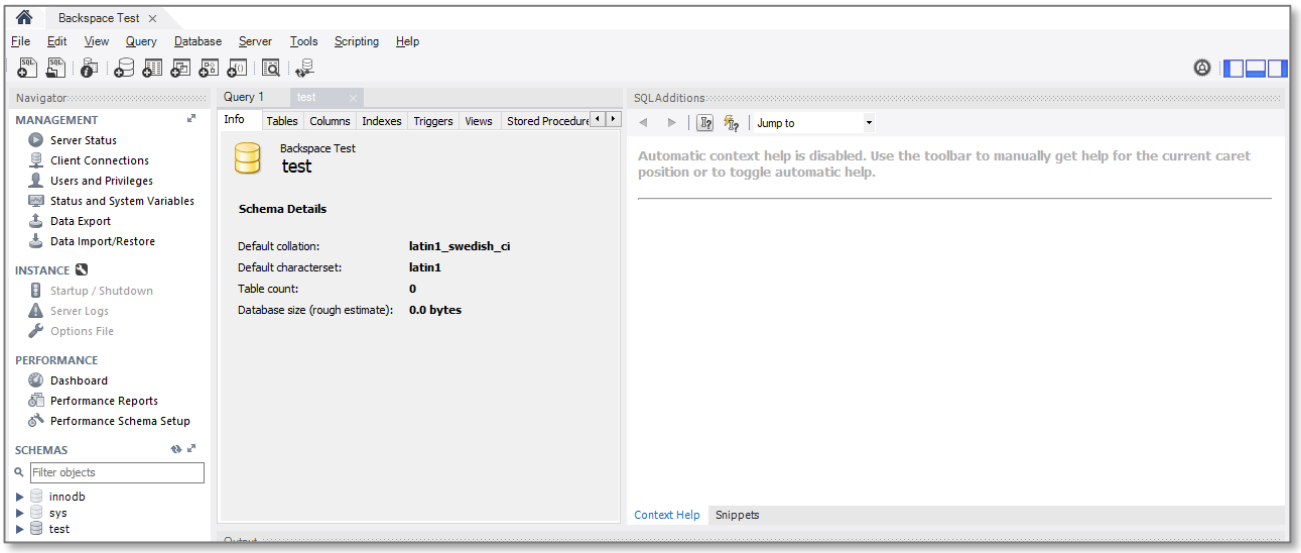
If have any problems connecting go to [Troubleshooting Connection Issues]



Hover over the “test” database under “SCHEMAS” and click the information icon to get information about the database that was created by us in RDS.



You then get an information screen for the database.

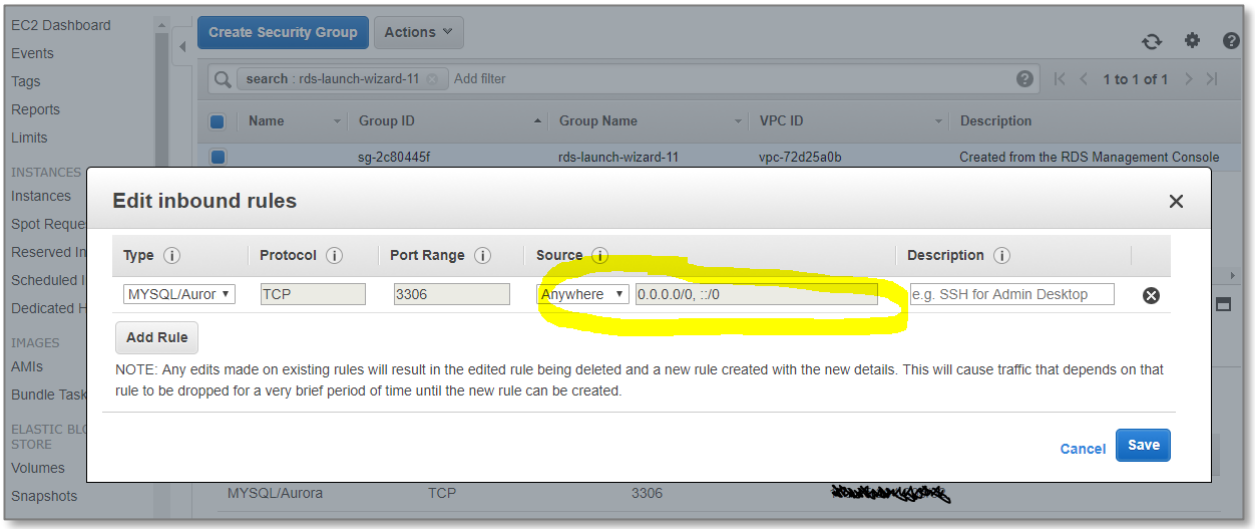


Troubleshooting Connection Issues

If you are getting connection errors then check the following:

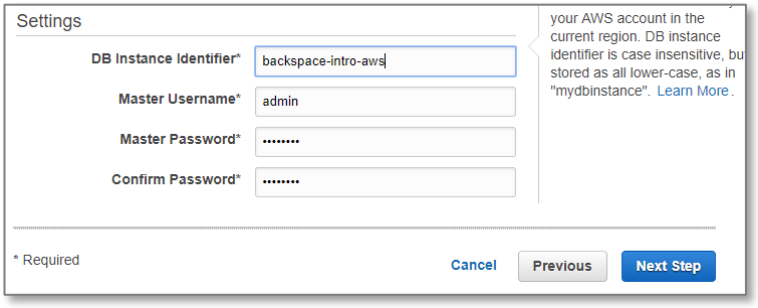
Security Group Inbound Rules

The security group should have an inbound rule for your IP address. If you are using a dynamic IP address or you are connecting from different networks then this will need to be changed to “anywhere” for the lab.



Database Username and Password

The username and password must be the one created when the RDS instance was created.



Hostname

The hostname will be the RDS Instance Connection Endpoint without :3306 on the end.

Connecting to your RDS Instance using the Command Line

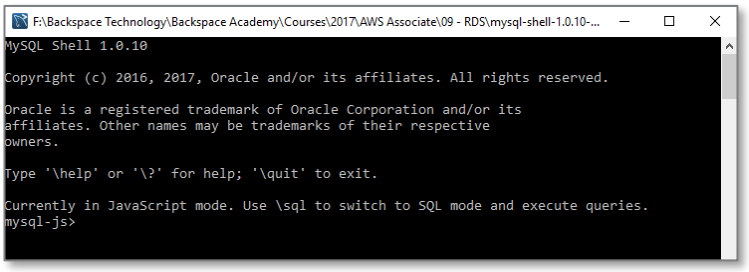
To connect to your MySQL Database using the command line you will need to download and install the MySQL Shell from

https://dev.mysql.com/downloads/shell/

Download and Unzip the file.

Go to the bin folder and run mysqlsh.exe

This will open the MySQL Shell

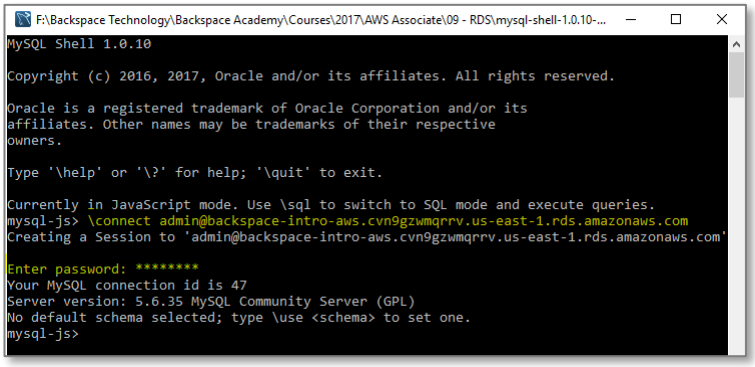


Connect your database using the following command (if the username is admin):

\connect admin@your-connection-hostname-goes-here

Enter your password when requested.

After a while you will be connected to your RDS instance.

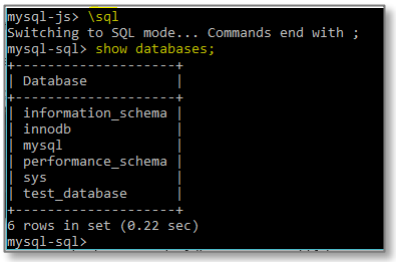


Enter SQL mode with the following command:

```
\sql
```

Enter the SQL command to list databases (don't forget the ';' on the end):

```
show databases;
```



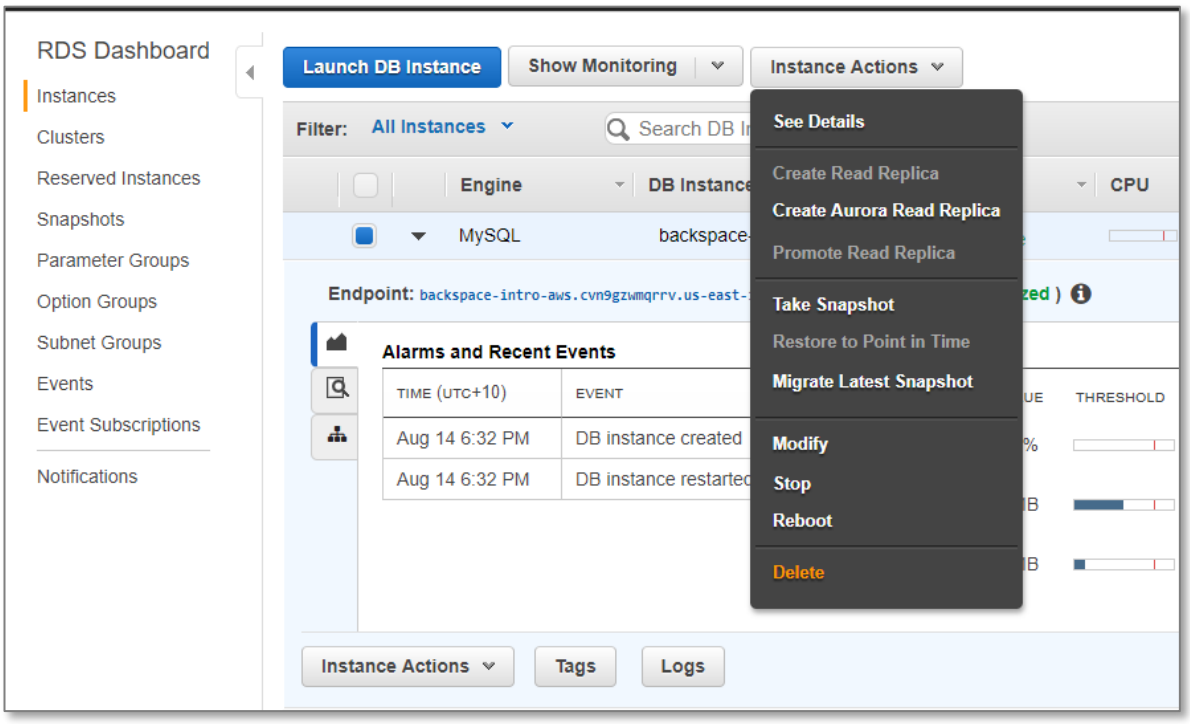
Type \quit to leave the command line

Clean Up

To avoid incurring charges from AWS we will terminate the instance.

Go back to the RDS console.

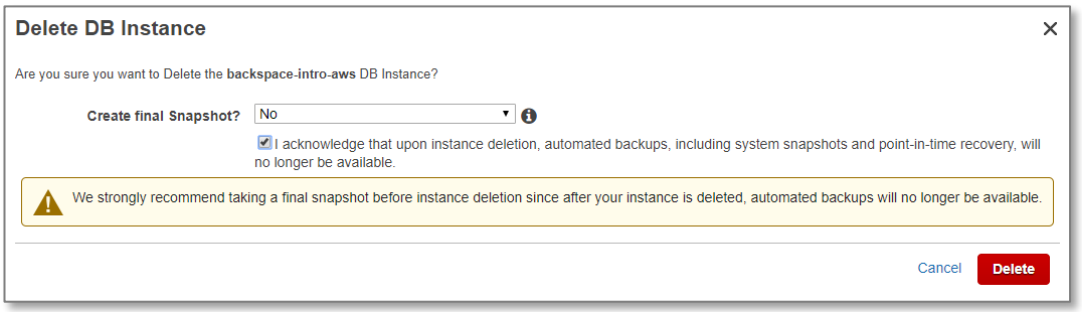
Click "Instance Actions", "Delete" to terminate the instance



Select “No” for “Create final snapshot”

Check “I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available.”

Click “Delete”



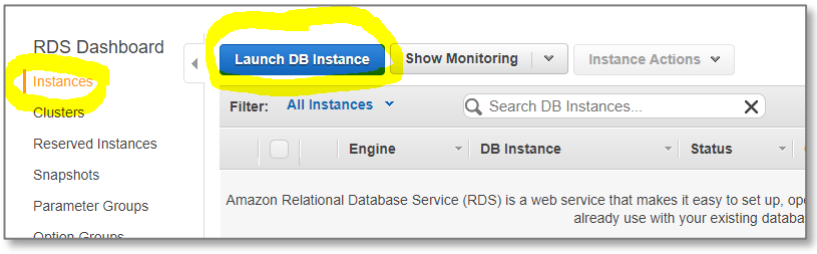
Deploying & Connecting to a PostgreSQL Database Server

In this section, we will learn how to deploy and connect to a PostgreSQL RDS instance.

From the AWS console select “RDS” from the Database services.

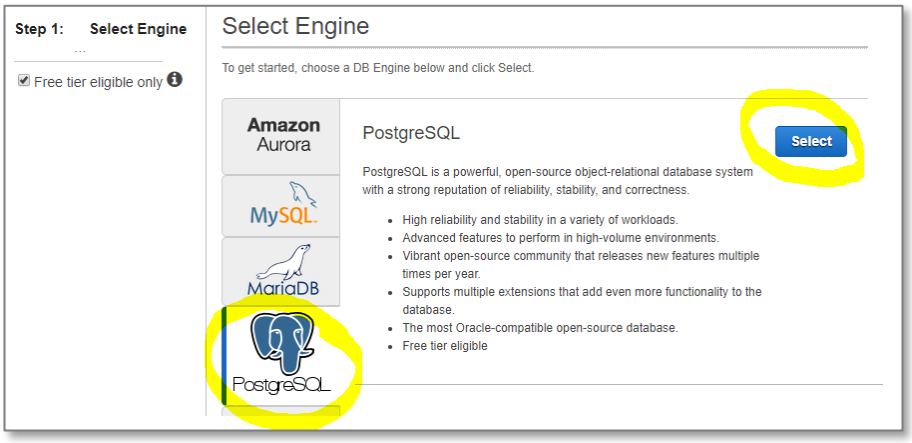
Select “instances”

Select “Launch DB Instance”



Select “Free tier eligible only”

Select the PostgreSQL engine



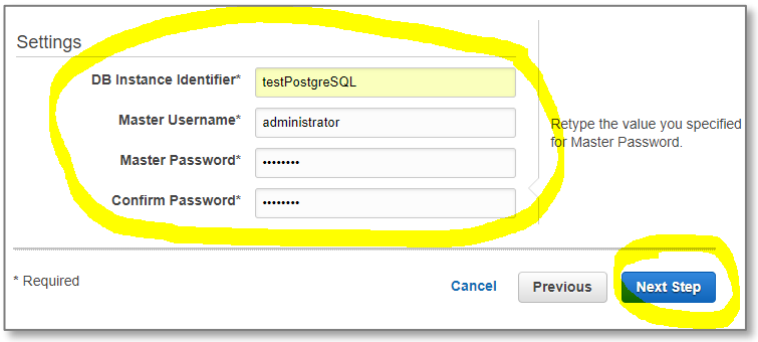
Make sure “Only show options that are eligible for RDS Free Tier” is selected

Select db.t2.micro instance class

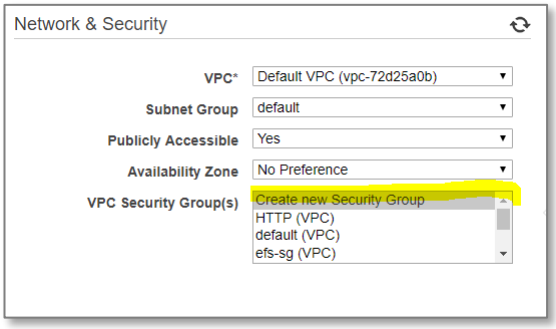
Give your instance a name/identifier “testPostgreSQL”.

Fill in a master username administrator (username “admin” can’t be used with PostgreSQL) and password

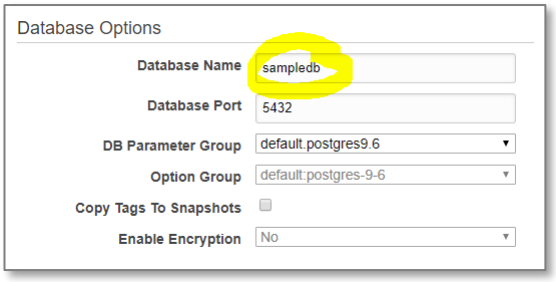
Click “Next Step”



Select “create new security group”

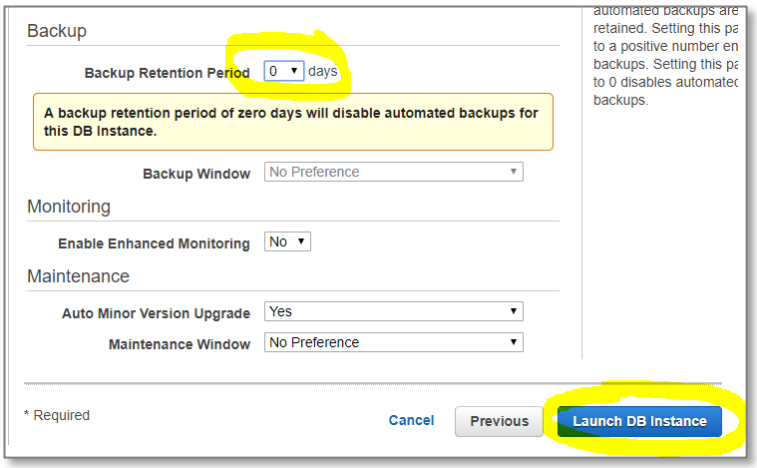


Call the database sampledb

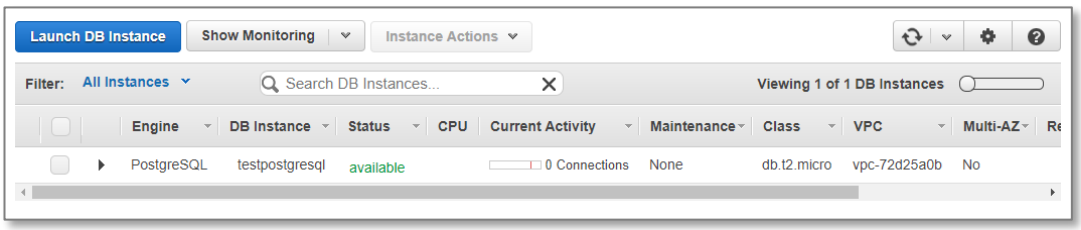


Set backup retention period to zero.

Click “Launch DB Instance”

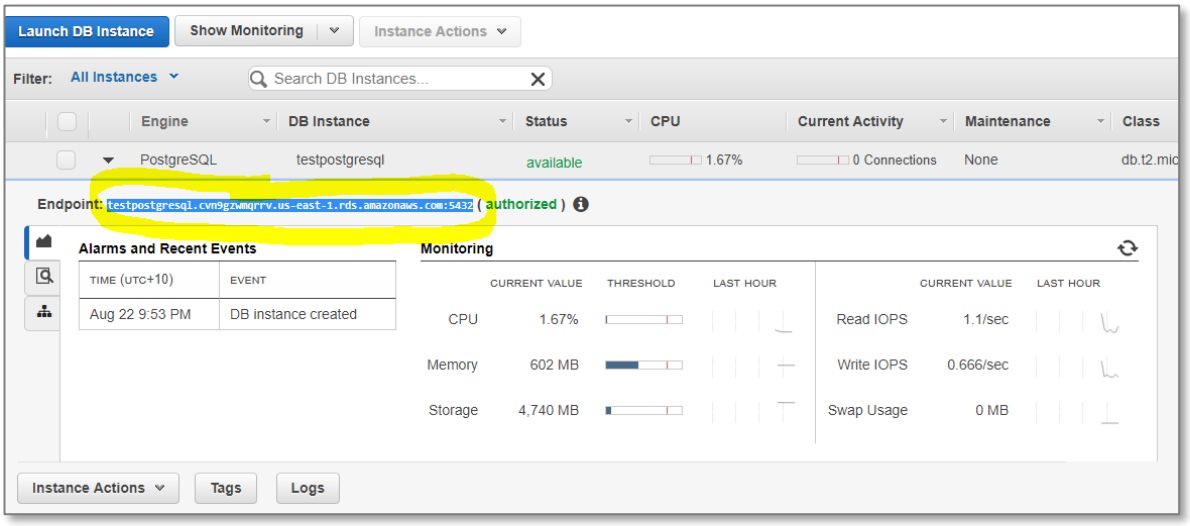


Click the refresh icon.
Wait for your instance state to change to available.



Connect to the Server using pgAdmin

Copy the server endpoint

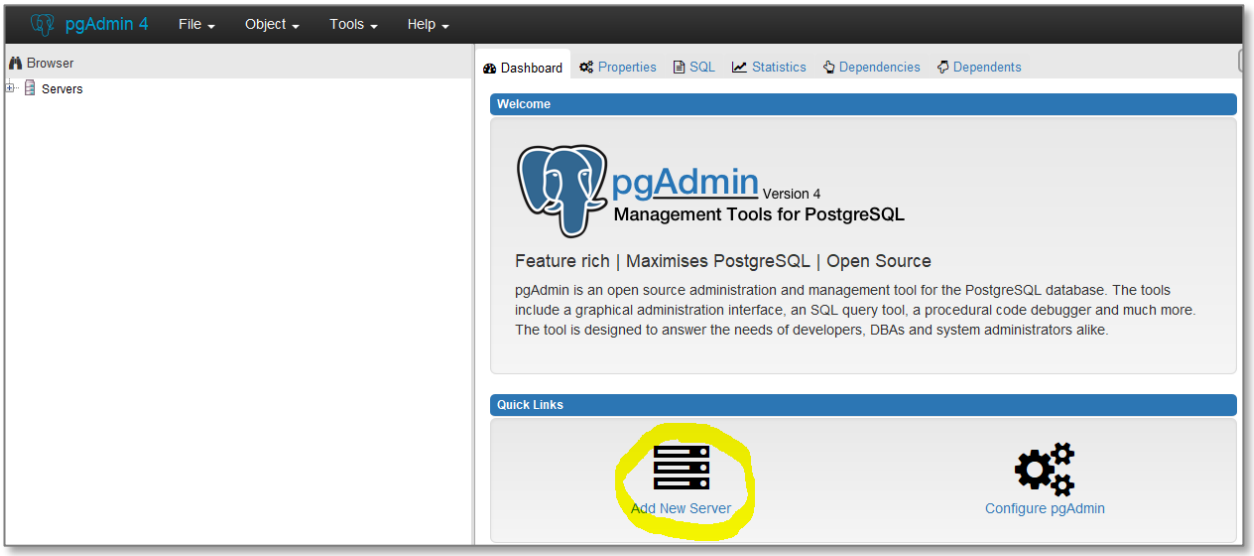


Download and install pgAdmin from:

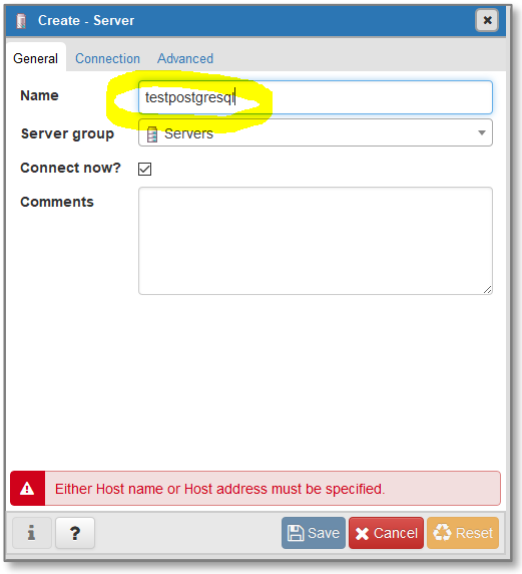
https://www.pgadmin.org/download/

Open pgAdmin

Select “Add new server”



Enter connection name “testpostgresql”



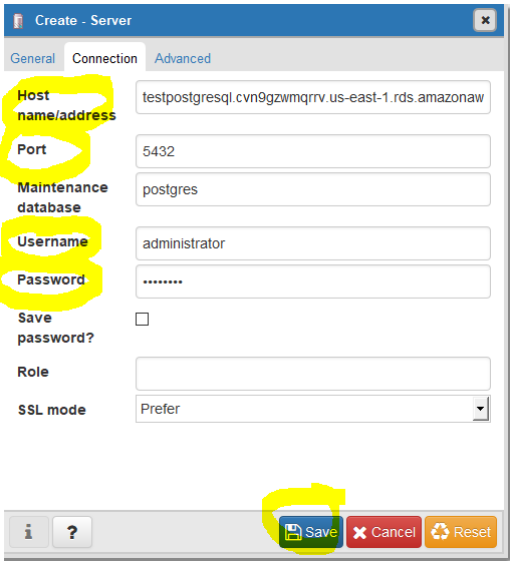
Click on the “Connection” tab

Enter the RDS instance endpoint without :5432 on the end

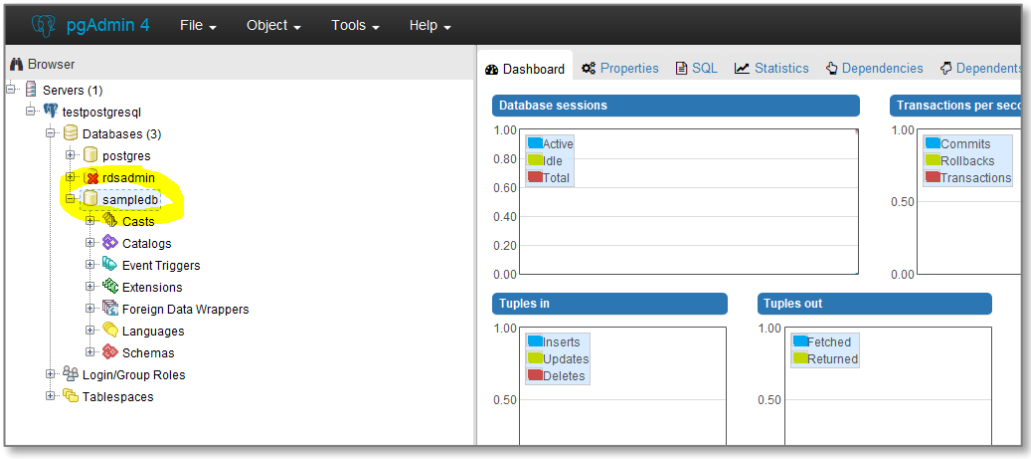
Enter Port 5432

Enter the username “administrator” and password

Click “Save”



You will now be connected to your database and will be able to navigate to see the sampledb database



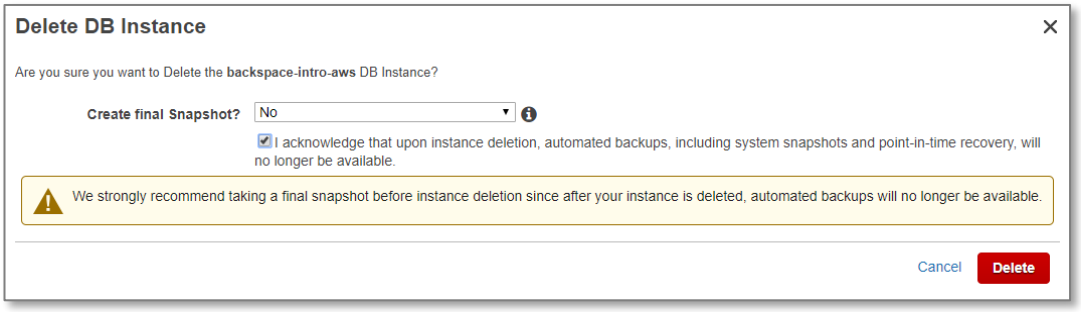
Clean Up

Go back to the RDS console and delete the instance

Select “No” for “Create final snapshot”

Check “I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available.”

Click “Delete”



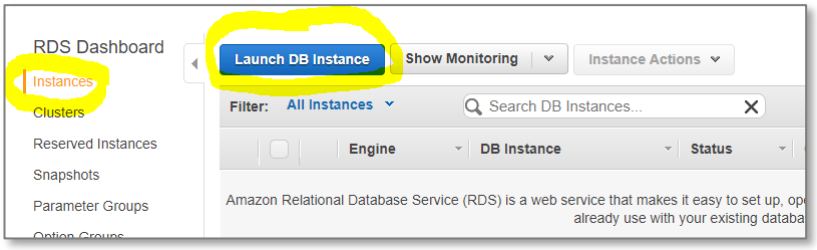
Deploying & Connecting to an Oracle Database Server

In this section, we will learn how to deploy and connect to an Oracle RDS instance.

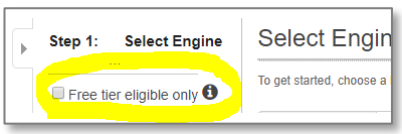
From the AWS console select “RDS” from the Database services.

Select “instances”

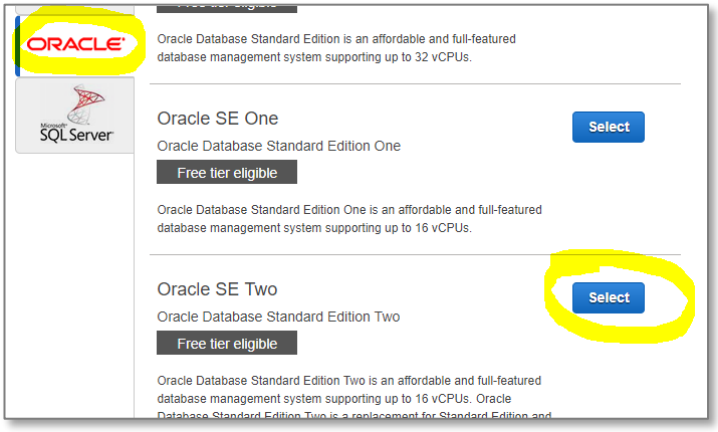
Select “Launch DB Instance”



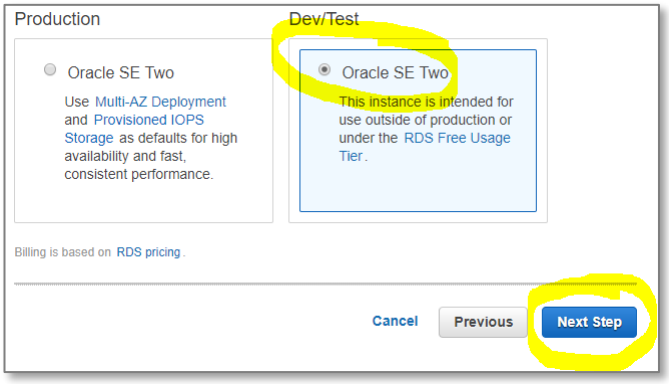
Licensed Oracle is not available on the free tier so uncheck “Free tier eligible only”



Select “Oracle SE Two”



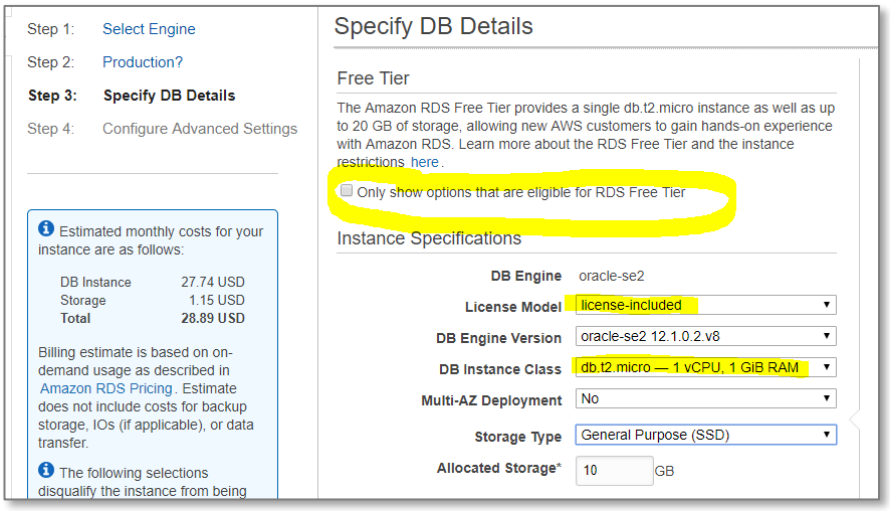
Select Dev/Test



Make sure Free tier is unchecked

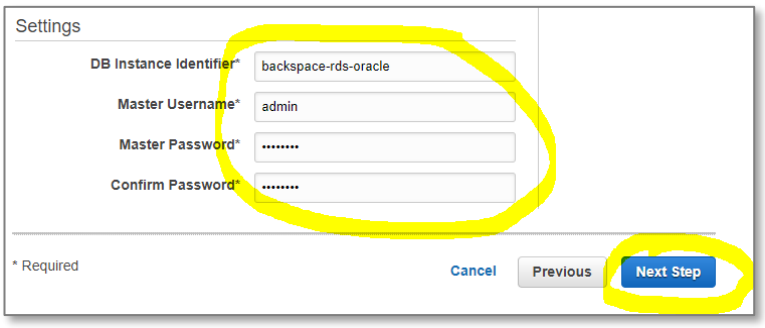
Select “license-included” for License Model

Select db.t2.micro instance type

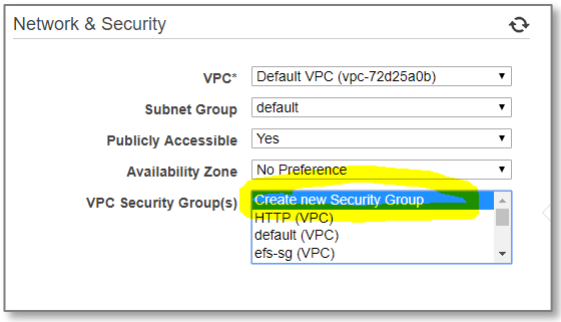


Give the server a name

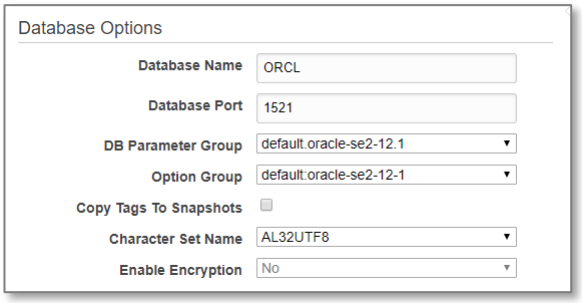
Enter a username “admin” and create a password. Be sure to remember this.



Select “Create new security group”

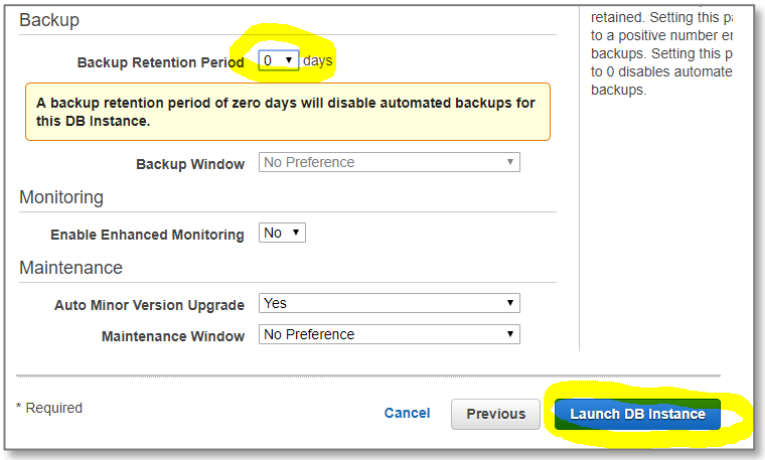


Leave Database Options as is

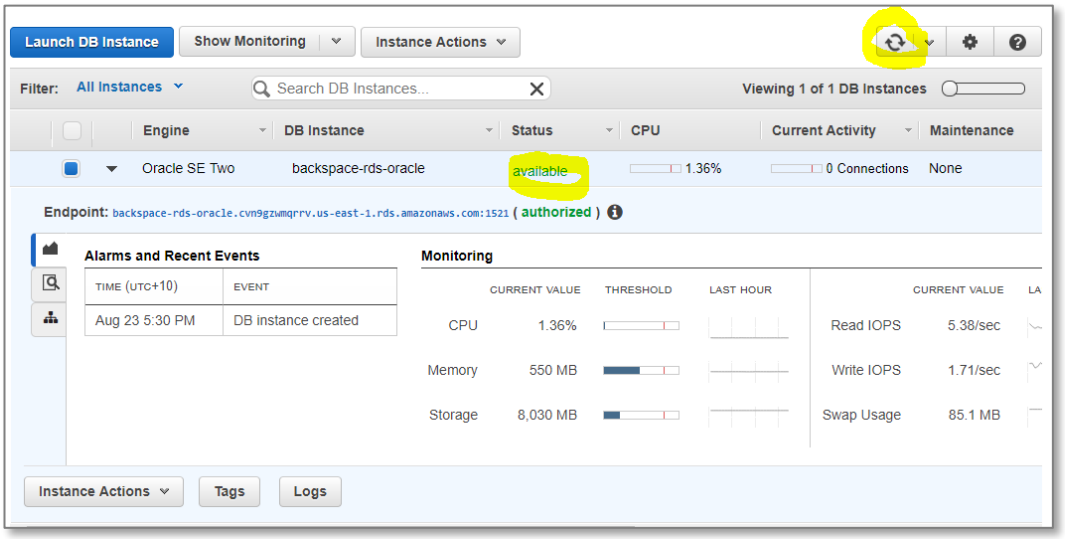


Disable automated backups

Click “Launch DB instance”



Refresh screen and wait for status to be “available”



Connecting to the Oracle RDS Instance

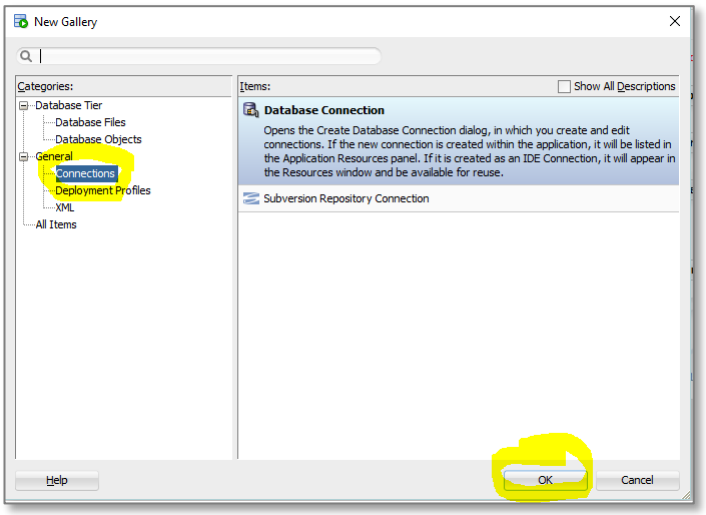
Download and install Oracle SQL Developer from:
http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html

Open SQL Developer

Select “File” – “New”

Select “Connections”

Click OK



Give the connection a name

Enter the database username and password

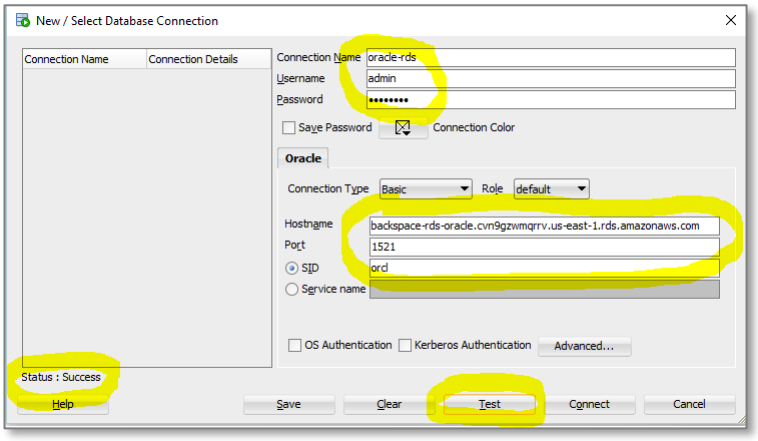
Enter the RDS Endpoint without :1521 on the end

Enter port as 1521

Enter SID as ORCL

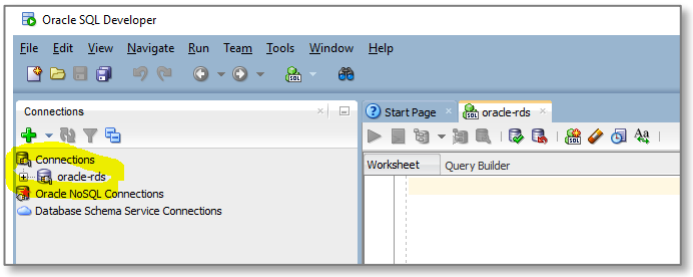
Click “Test”

Status should eventually be “success”



Click “Connect”

You will now see the new connection to the ORCL database in the treeview



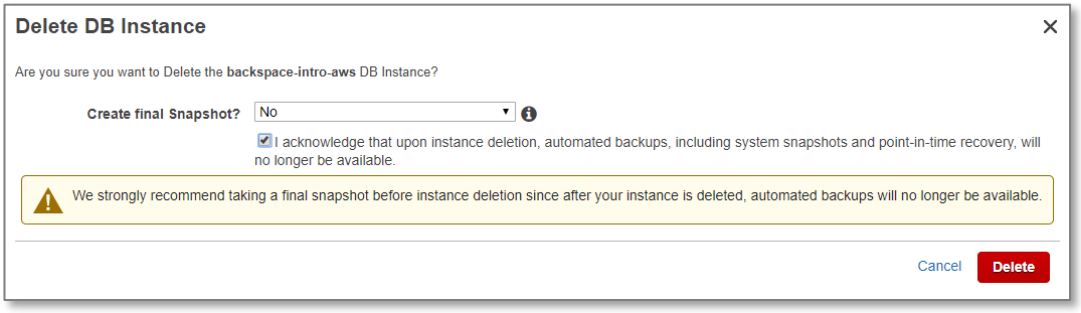
Clean Up

Go back to the RDS console and delete the instance

Select “No” for “Create final snapshot”

Check “I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available.”

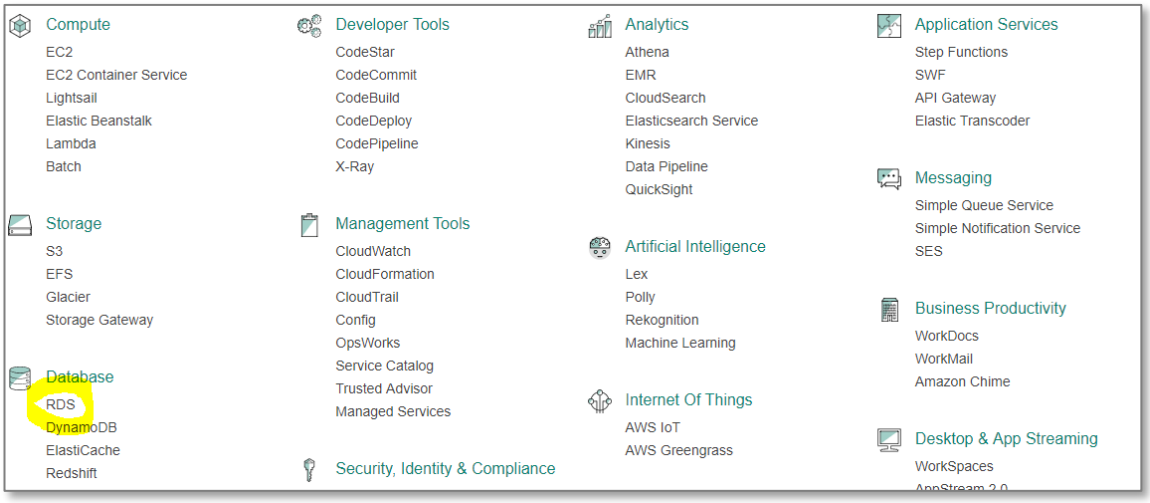
Click “Delete”



Deploying & Connecting to a Microsoft SQL Server Database

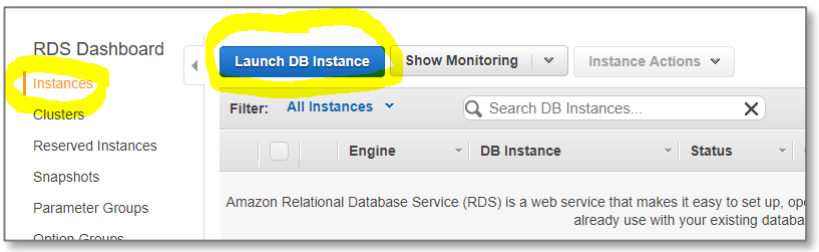
In this section, we will learn how to deploy and connect to an Microsoft SQL Server RDS instance.

From the AWS console select “RDS” from the Database services.



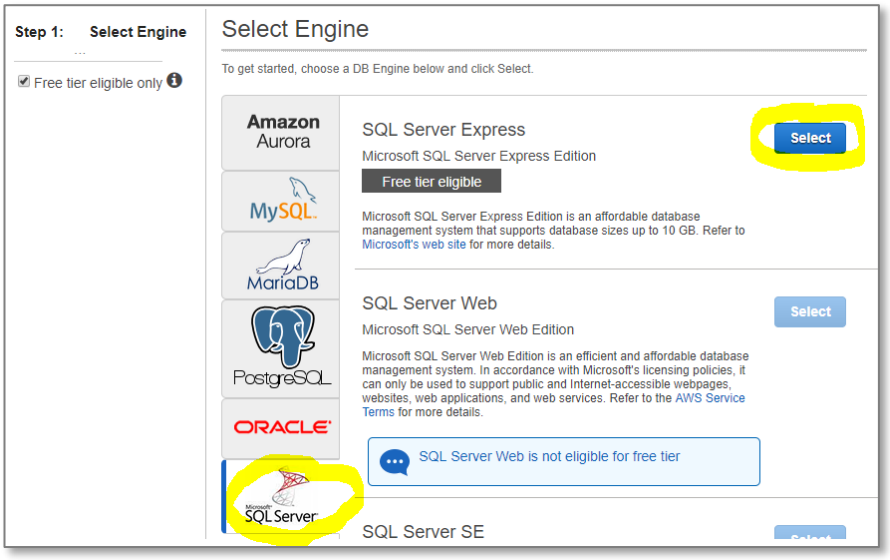
Select “instances”

Select “Launch DB Instance”



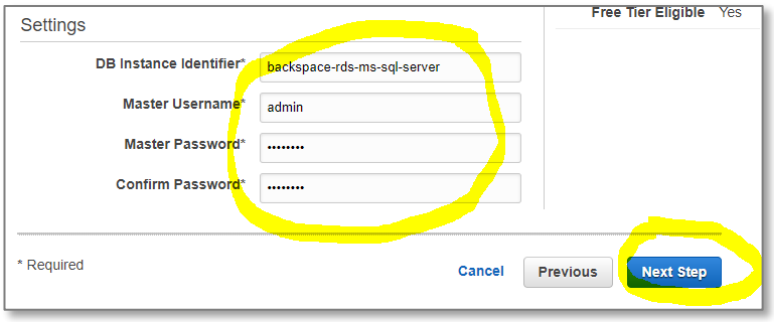
Select “Free tier eligible only”

Select SQL Server Express

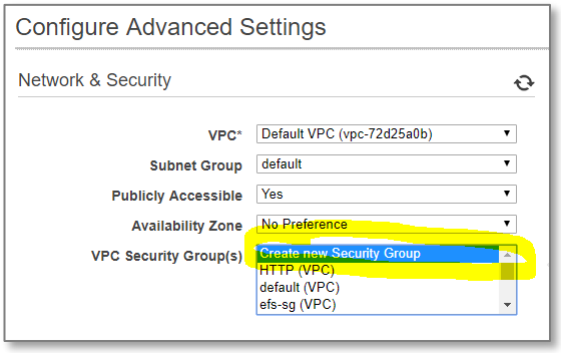


Give your DB Instance a name

Create a username and password for the database

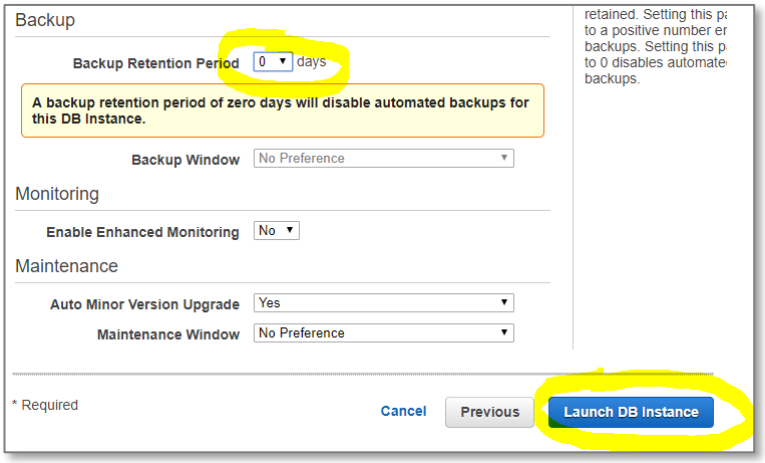


Select create “new security group”

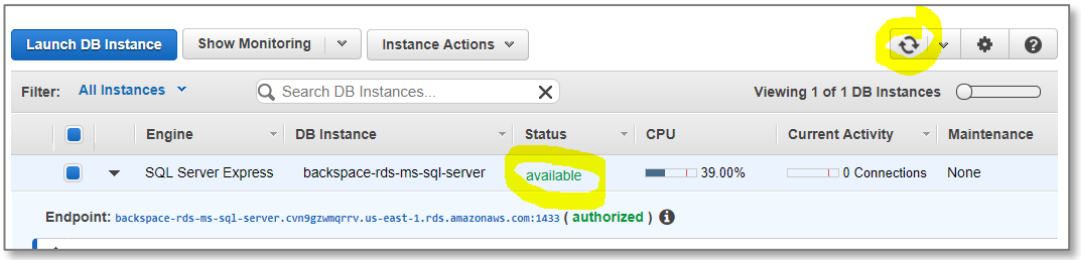


Change Backup retention period to zero.

Click “Launch DB Instance”



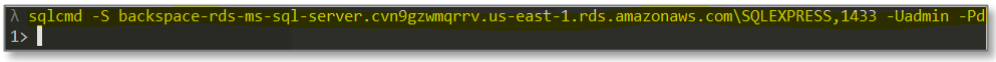
Refresh and wait for the instance state to change from “creating” to “available”



Connecting to the Database Using the Windows Command Line

Use sqlcmd from the Windows command line to connect to your database with the following command:

sqlcmd -S *rds_endpoint_goes_here*\SQLEXPRESS,1433 -U*username_goes_here* -P*password_goes_here*



Connecting to the Database Using SQL Management Studio

Download and install SQL Server Management Studio from:

https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms

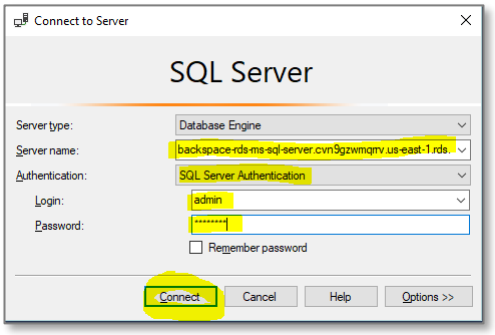
Open the app after installing

Enter in “Server name” the RDS endpoint without the :1433

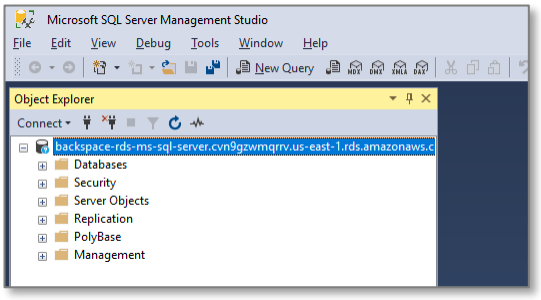
Select “SQLServer Authentication”

Enter the username and password

Click “Connect”



You will eventually be connected to your RDS database server



Clean up

To avoid incurring charges from AWS we will terminate the instance.

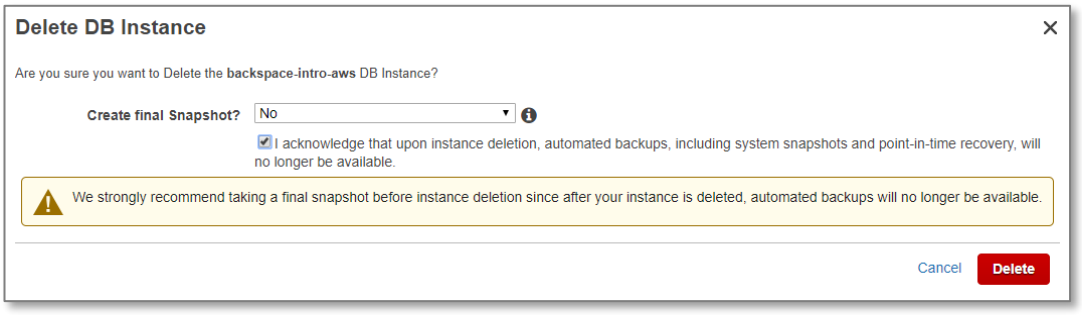
Go back to the RDS console.

Select the instance and delete.

Select “No” for “Create final snapshot”

Check “I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available.”

Click “Delete”

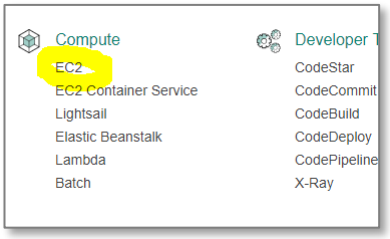


Migrating from Oracle to Aurora Using AWS Database Migration Service

In this section, we will learn how to migrate an Oracle database to AWS Aurora using the AWS Database Migration Service.

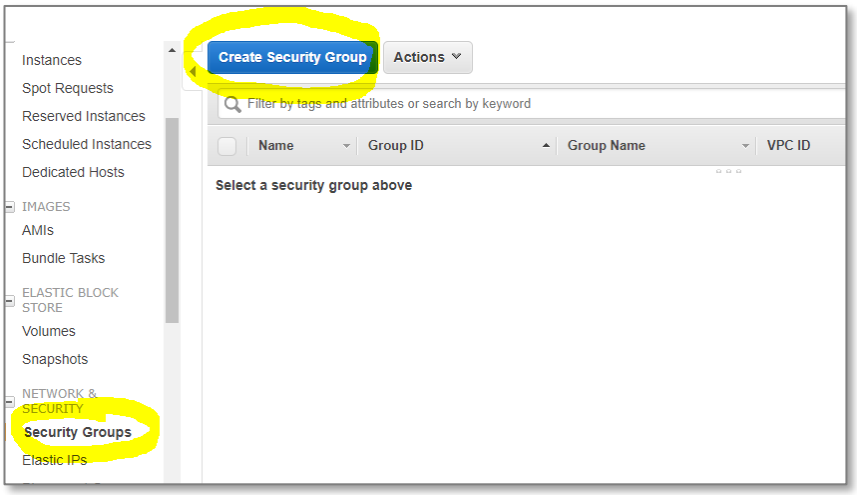
Create a Security Group

From the AWS console select “EC2” from the Compute services.



Select “Security Groups” from the side menu

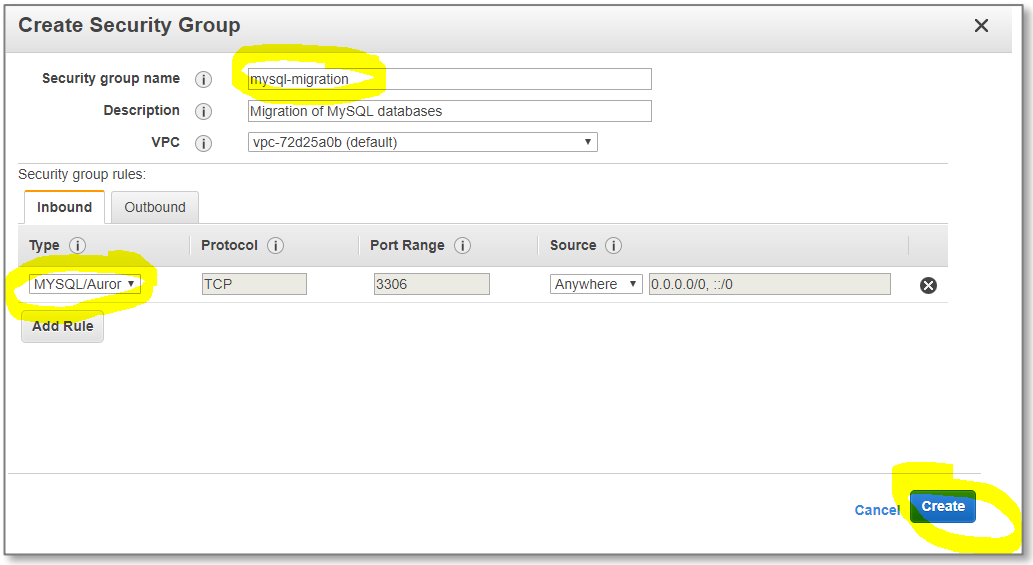
Click “Create Security Group”



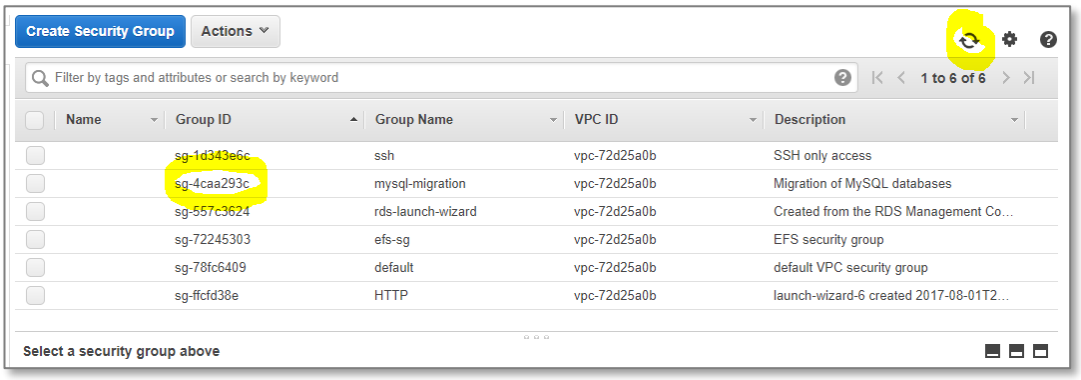
Give the Security the name “oracle-aurora-migration” and a description

Add an inbound rule for MySQL access. You can use “anywhere” for source although in a production environment you should select “My IP”

Click “Create”

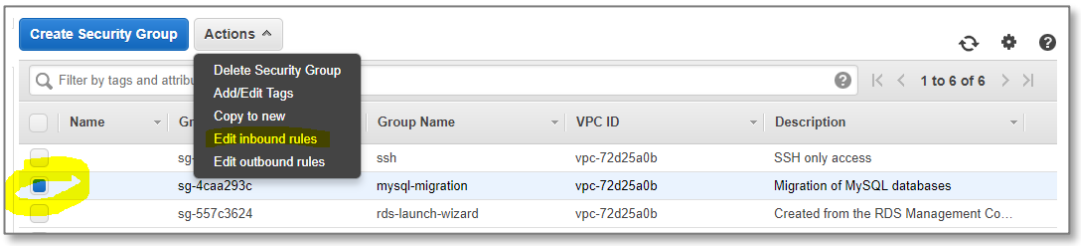


Click the refresh icon to see your security group

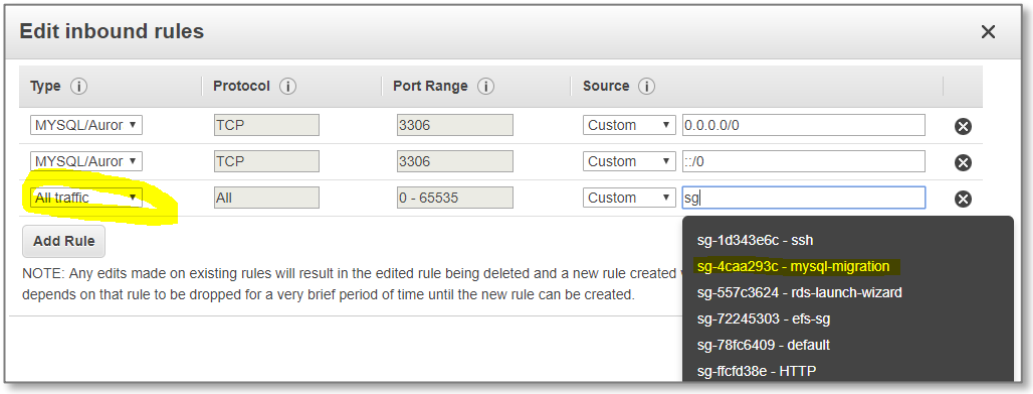


Copy the Security Group ID

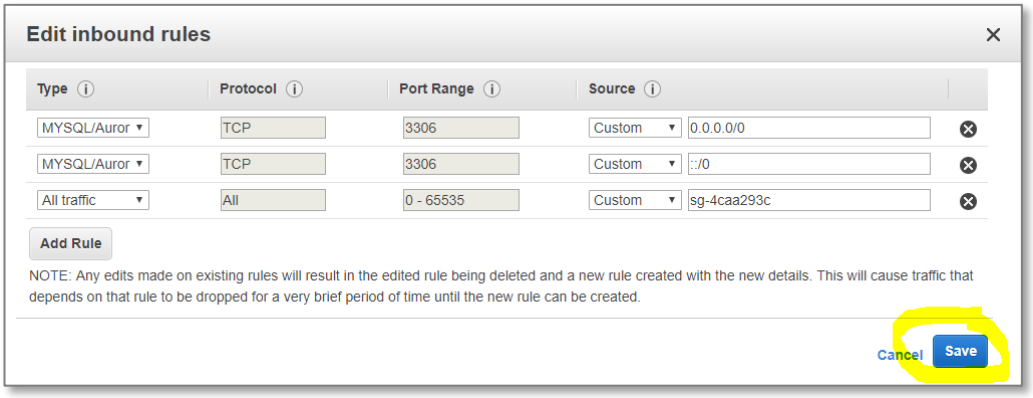
Select “Actions” – “Edit inbound rules”



Create an inbound rule for type “All traffic” and source the Security group ID (type “sg” for a list)

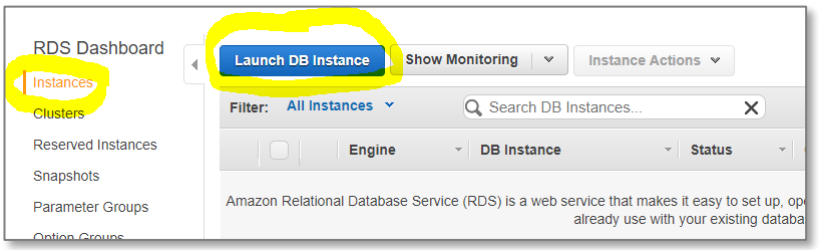


Click Save



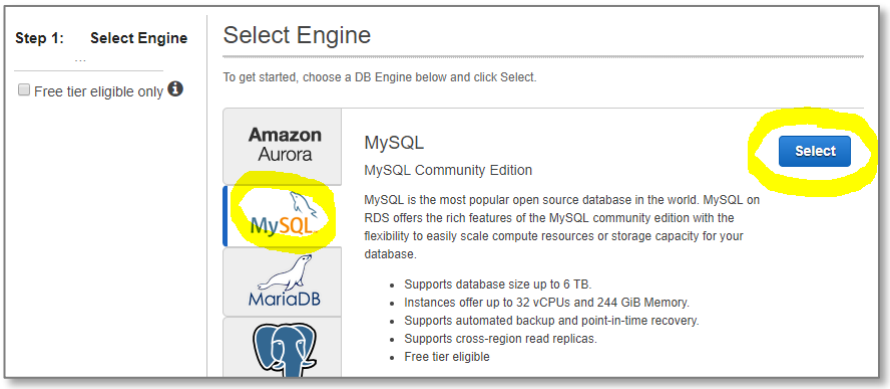
Create Source MySQL database

Select “Launch DB Instance”

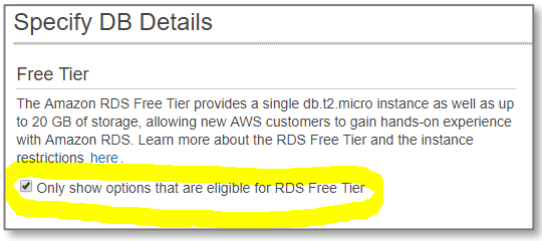


Select “Free tier eligible only”

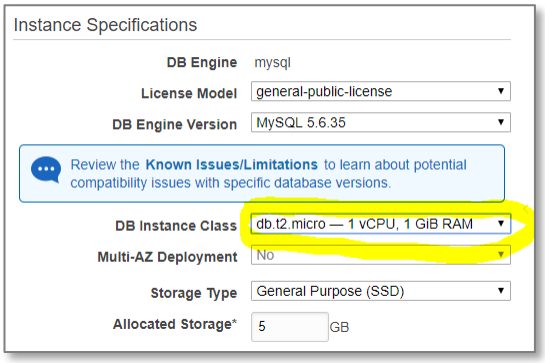
Select the MySQL Community Edition



Make sure “Only show options that are eligible for RDS Free Tier” is selected



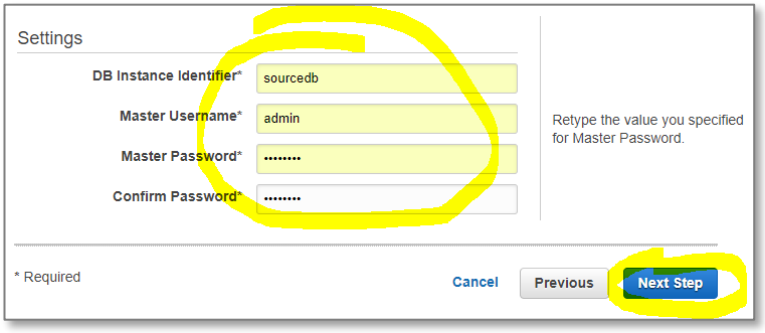
Select db.t2.micro instance class



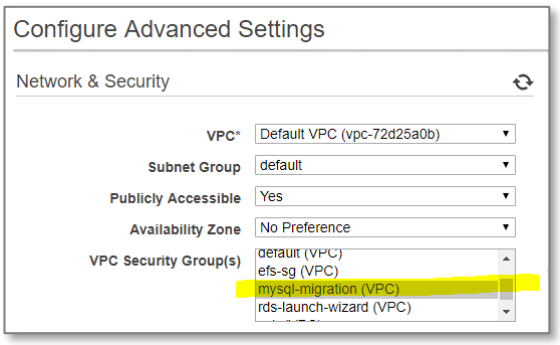
Give your instance a name/identifier.

Fill in a master username and password

Click “Next Step”

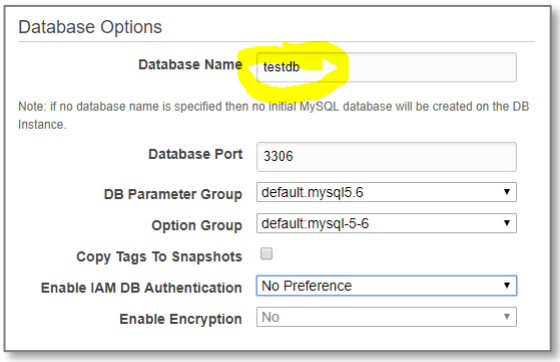


Select the security group



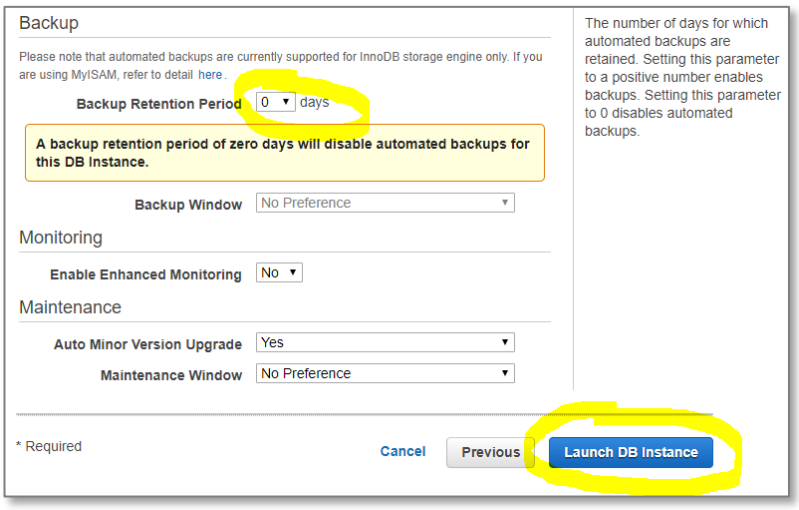
Enter database name “testdb”

Leave other options default as below.

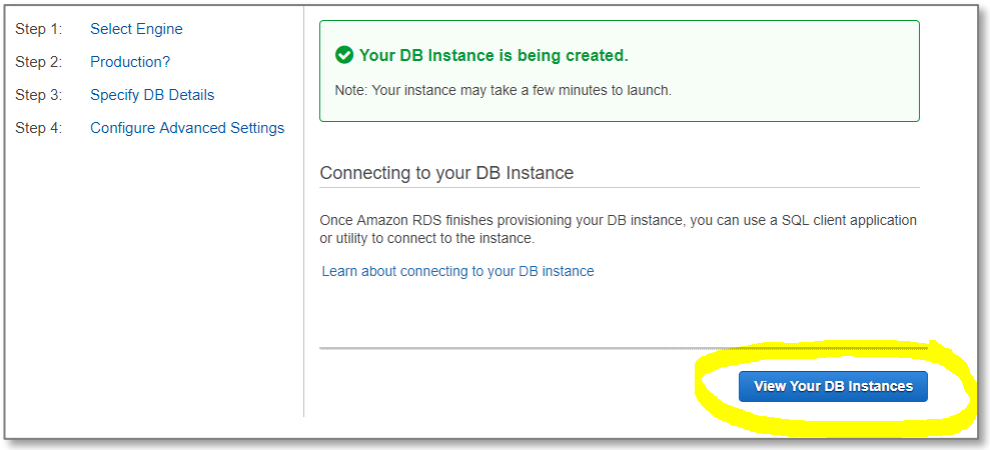


Change “Backup Retention Period” to disable automated backups.

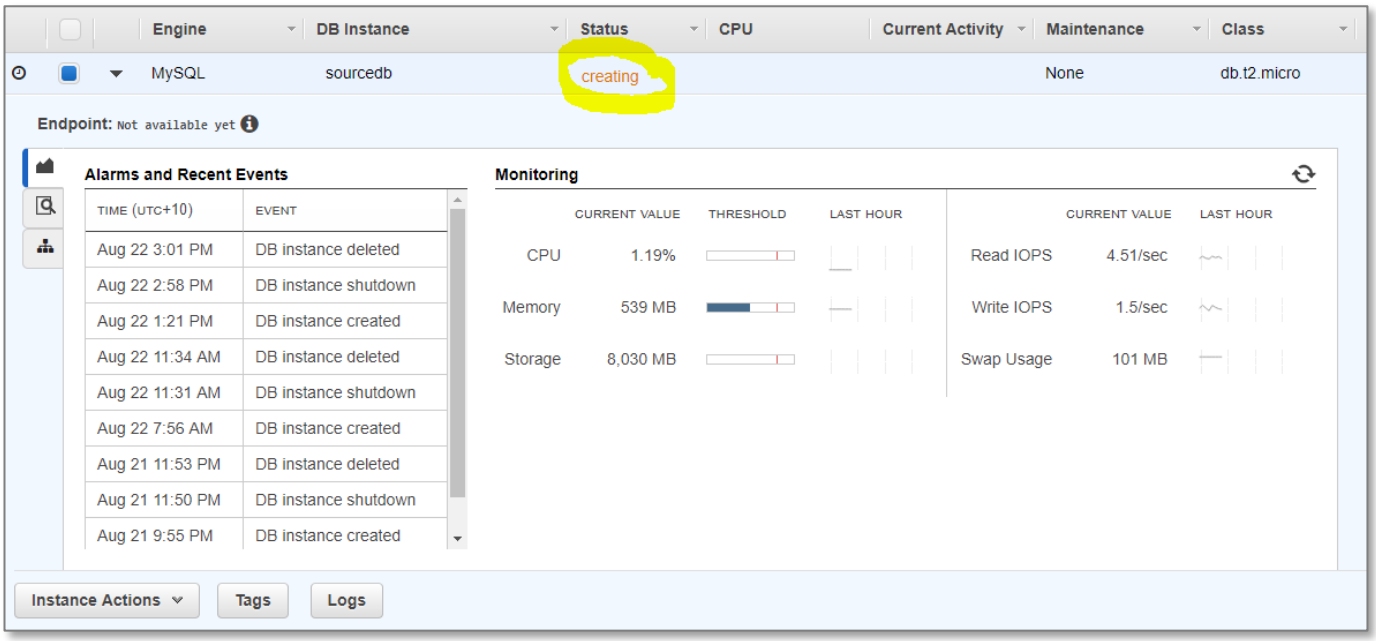
Click “Launch DB Instance”



Click “View your DB Instances

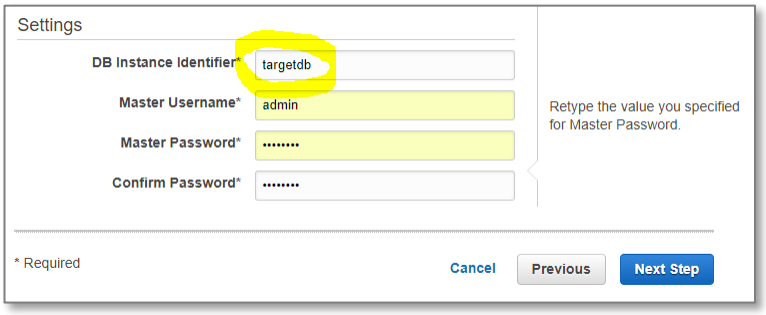


Your instance will show status “creating”.

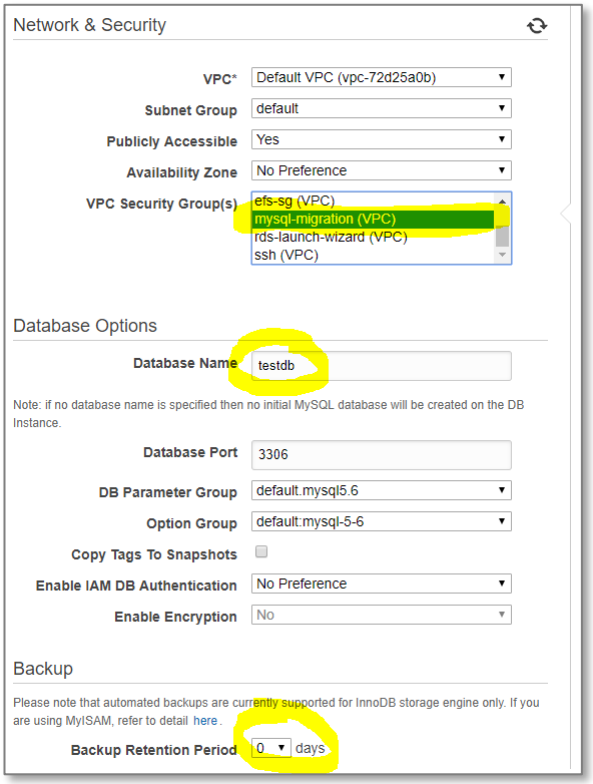


Create Target Database

Create another RDS instance with exactly the same settings as before but with a DB Instance Identifier “targetdb”



Use the same settings as before for security group, Database Name and Backup retention period:



Connecting to the Source Database

Connect to the source database using the MySQL Shell:

```
\connect admin@your-connection-hostname-goes-here
```

```
λ mysqlsh.exe
MySQL Shell 1.0.10

Copyright (c) 2016, 2017, Oracle and/or its affiliates. All rights reserved.

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

Type '\help' or '\?' for help; '\quit' to exit.

Currently in JavaScript mode. Use \sql to switch to SQL mode and execute queries.
mysql-js> \connect admin@sourcedb.cvn9gzwmqrrv.us-east-1.rds.amazonaws.com:3306
Creating a Session to 'admin@sourcedb.cvn9gzwmqrrv.us-east-1.rds.amazonaws.com:3306'
Enter password: ********
Your MySQL connection id is 26
Server version: 5.6.35 MySQL Community Server (GPL)
No default schema selected; type \use <schema> to set one.
mysql-js>
```

Set active database as testdb:

\use testdb

```
mysql-js> \use testdb
Schema set to `testdb`.
mysql-js>
```

Change to SQL mode

\sql

```
mysql-js> \sql
Switching to SQL mode... Commands end with ;
mysql-sql>
```

Create a table called migrate (don't forget the ";" at the end):

create table sample (name varchar(20));

```
mysql-sql> create table sample  (name varchar(20));
Query OK, 0 rows affected (0.27 sec)
mysql-sql>
```

Check it is there:

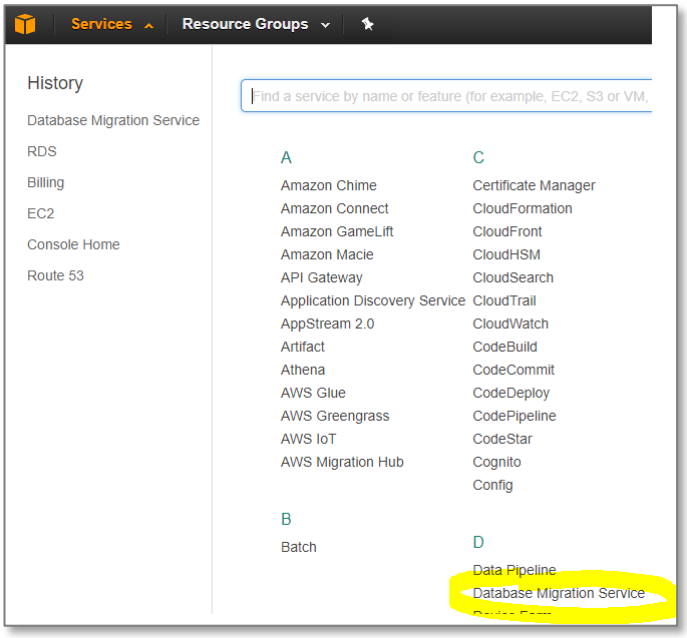
show tables;

```
mysql-sql> show tables;
+-----------------+
| Tables_in_testdb |
+-----------------+
| sample          |
+-----------------+
1 row in set (0.25 sec)
mysql-sql>
```

DO NOT change the target database as this table will be migrated across with the AWS Database Migration Service

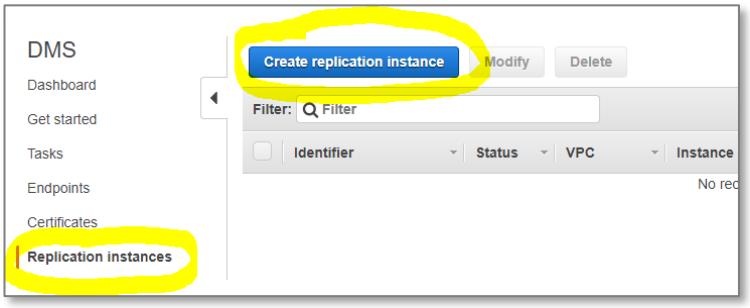
Creating an AWS Database Migration Service Job

Select “AWS Database Migration Service” from the services menu.

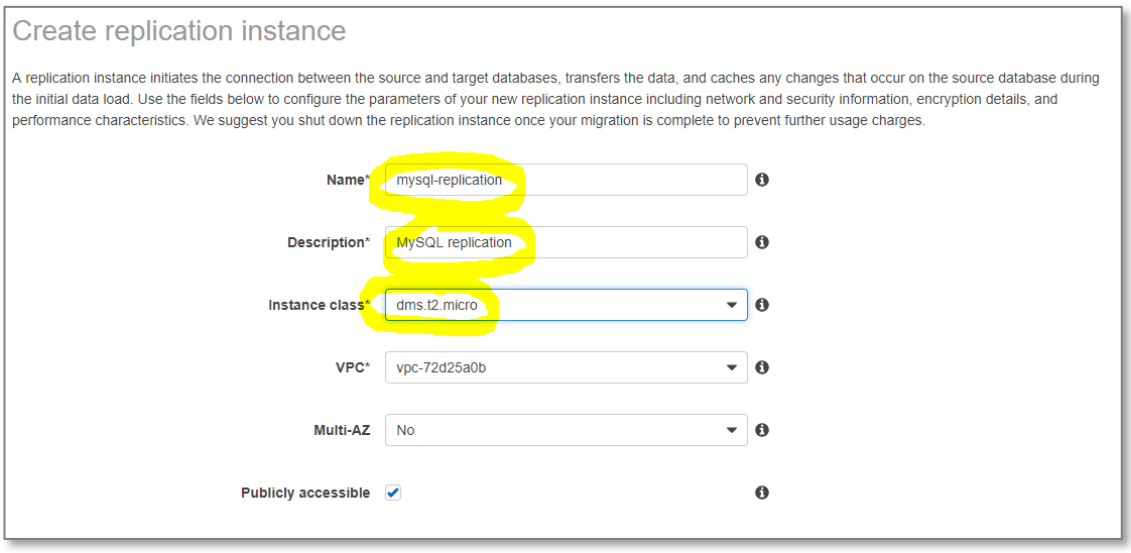


Create Replication Instance

Select “Replication Instances” from the side menu
Click “Create Replication Instance”

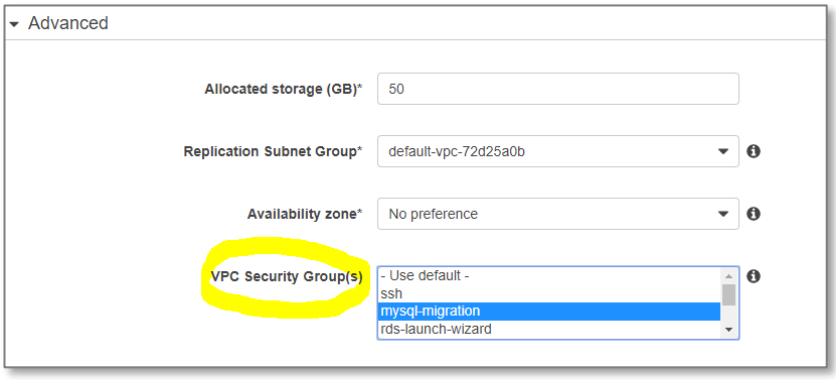


Give it a name and description
Set the instance class to t2.micro

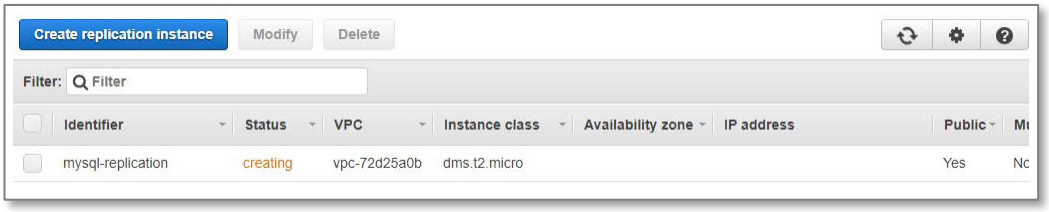


Select “Advanced”

Select your security group

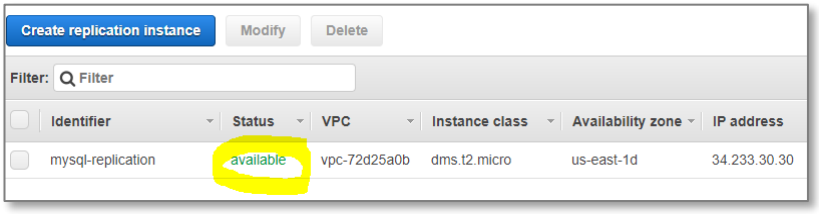


Click “Create replication instance”

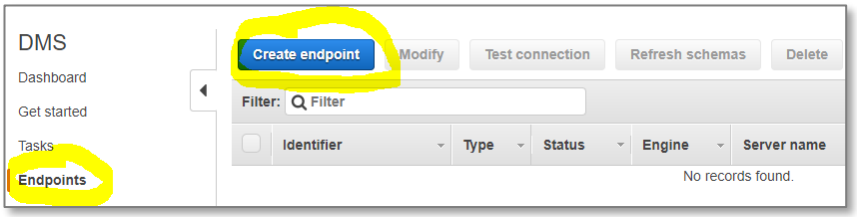


Create Source and Target Database Endpoints

Wait until the replication instance status is available.

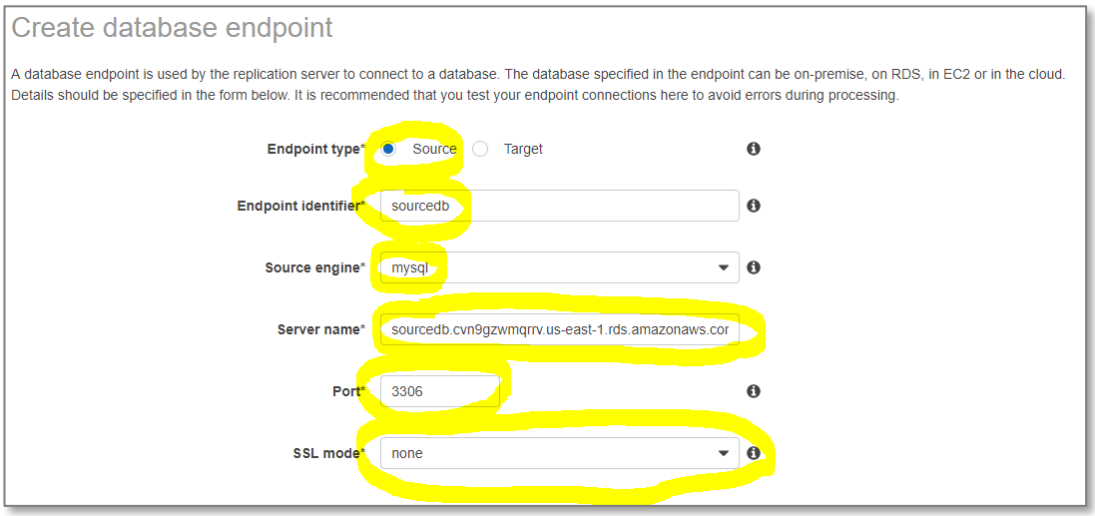


Select “Endpoints” from the side menu
Click “Create endpoint”



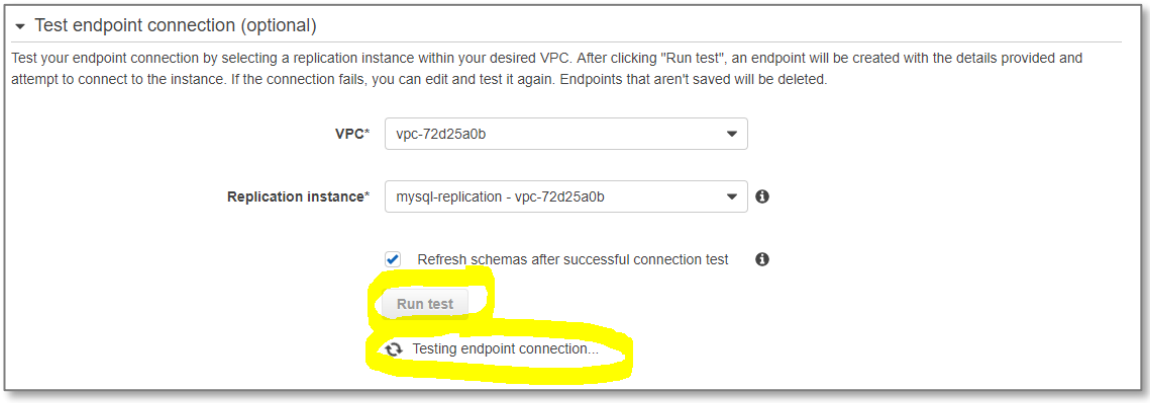
Enter the following:

Endpoint type – source
Endpoint identifier – sourcedb
Source engine – mysql
Server name – RDS source database endpoint without :3306 on the end
Port – 3306
SSL mode - None



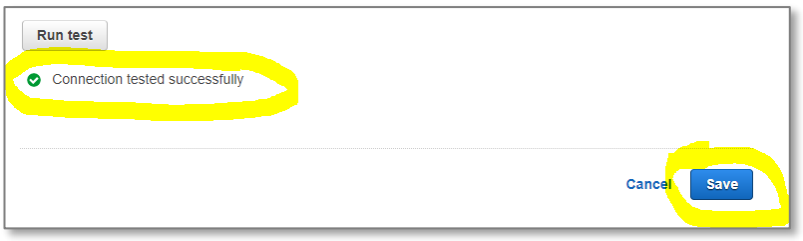
Enter the username and password for the instance

Now test the endpoint connection to the replication instance

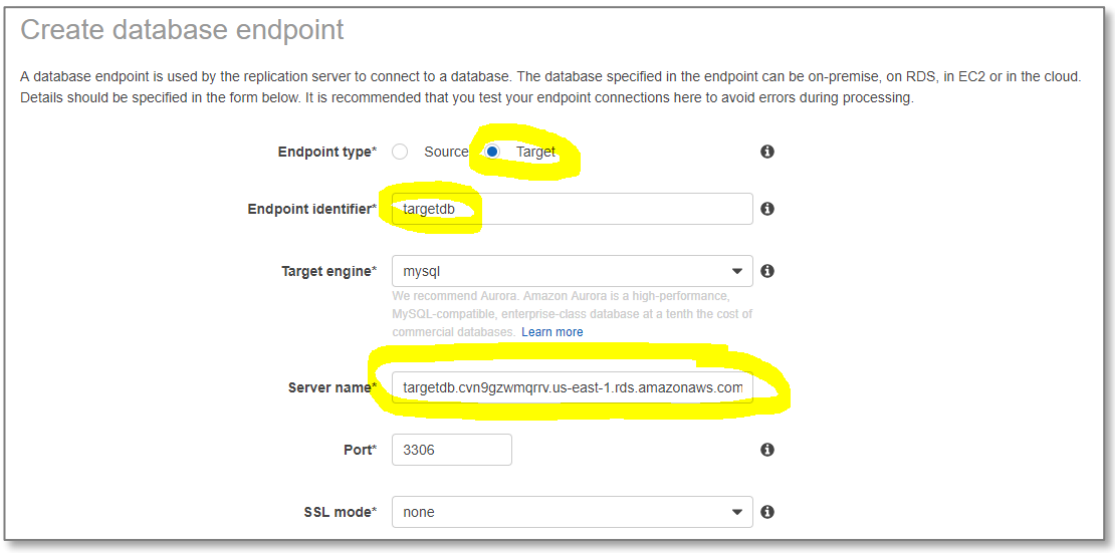


If connection test is successful click save:

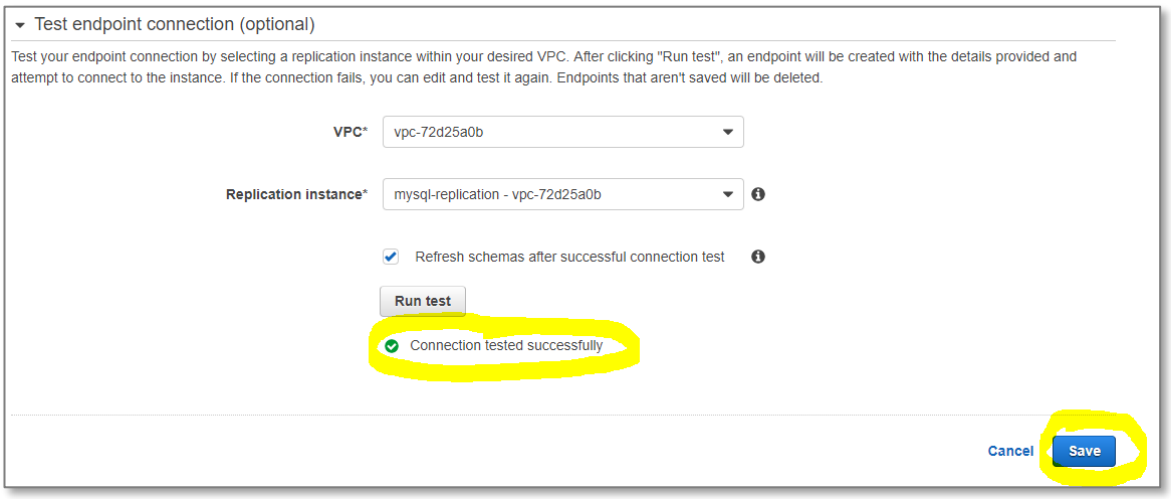
(If not successful check your RDS security groups)



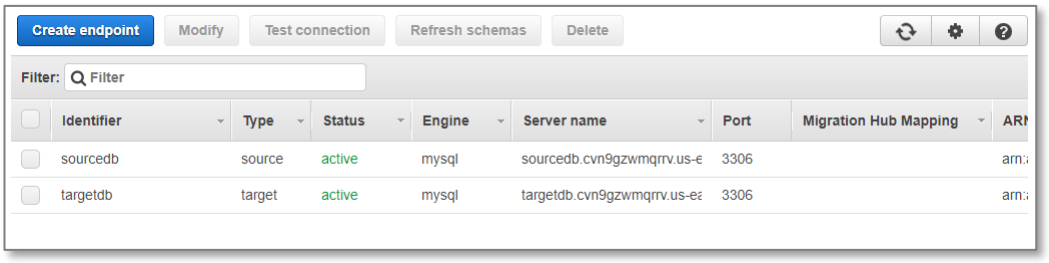
Select “Create endpoint and do the same process for the target database:



Test the target connection and if successful click save:

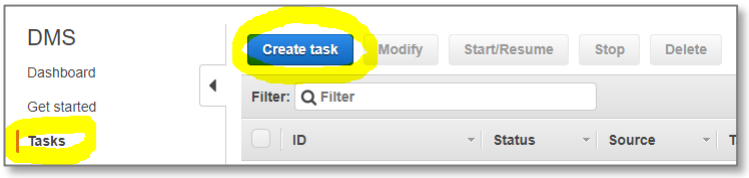


Both your endpoints should now be active

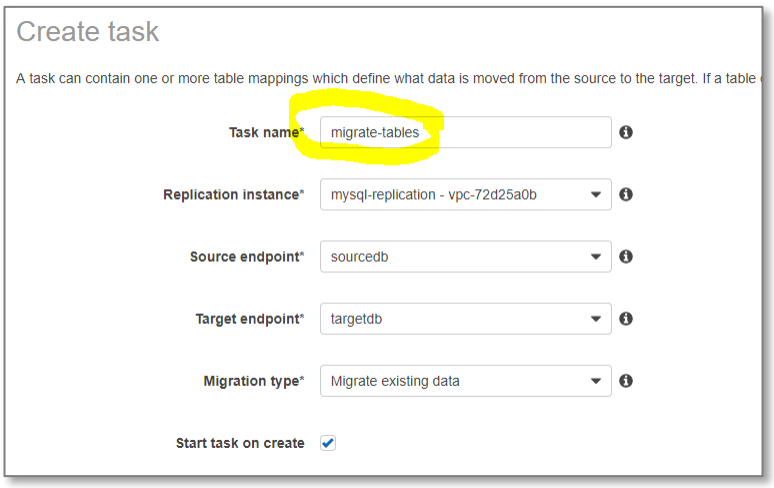


Create a Migration Task

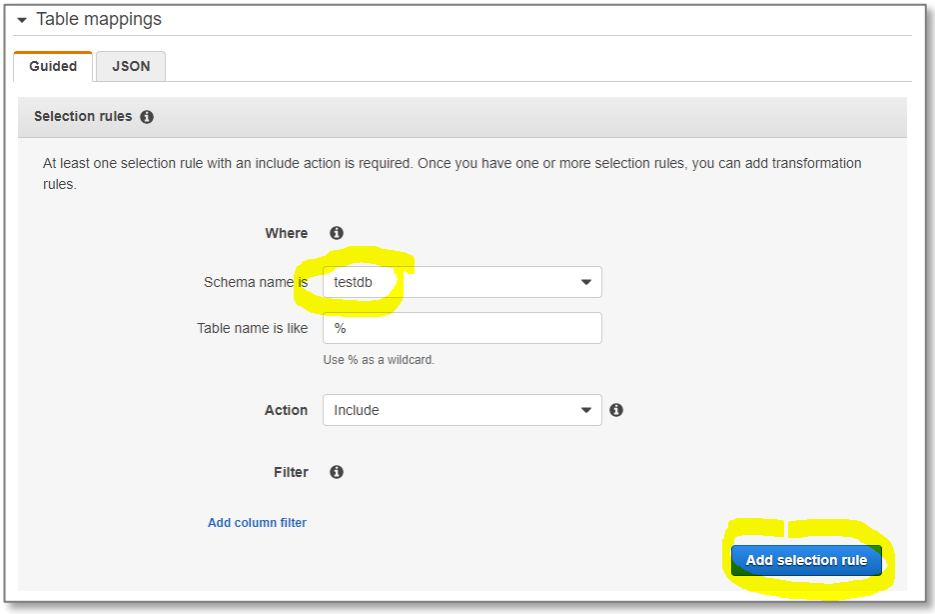
Select “tasks” from the side menu and click “create task”



Give the task a name

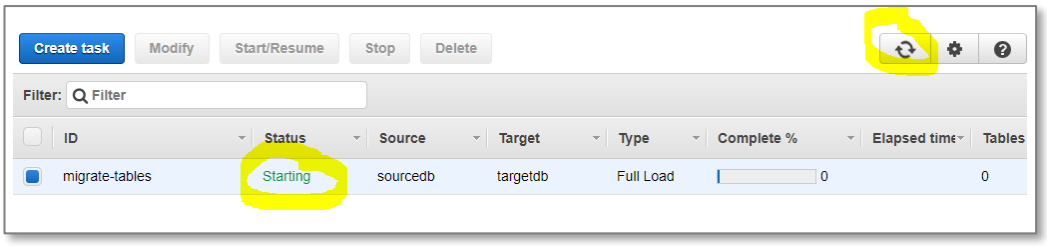


Select the testdb schema for migration in “Table mappings”
Click “Add selection rule”

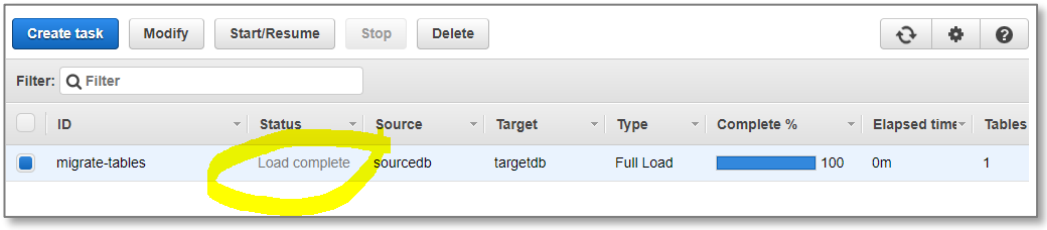


Click “Create Task”

The Status will eventually change from “Creating” to “Starting” to “Running”



When finished status will be “Load complete”



Now connect to the target database using the MySQL Shell to see if the table was migrated.

Connect to the source database using the MySQL Shell:

\connect admin@your-connection-hostname-goes-here

```
mysql-sql> \connect admin@targetdb.cvn9gzwmqrrv.us-east-1.rds.amazonaws.com:3306
Creating a Session to 'admin@targetdb.cvn9gzwmqrrv.us-east-1.rds.amazonaws.com:3306'
Enter password: ********
Closing old connection...
Your MySQL connection id is 54
Server version: 5.6.35 MySQL Community Server (GPL)
No default schema selected; type \use <schema> to set one.
mysql-sql>
```

Set active database as testdb:

\use testdb

```
mysql-js> \use testdb
Schema set to `testdb`.
mysql-js>
```

Change to SQL mode

\sql

```
mysql-js> \sql
Switching to SQL mode... Commands end with ;
mysql-sql>
```

Check the sample table is there:

show tables;

```
mysql-sql> show tables;
+-----------------+
| Tables_in_testdb |
+-----------------+
| sample          |
+-----------------+
1 row in set (0.25 sec)
mysql-sql>
```

\quit to end connection

Clean Up

Go back to the AWS Database Migration Service console delete the task, terminate the replication instance and delete the endpoints.

Go back to the RDS console and delete the RDS instances.