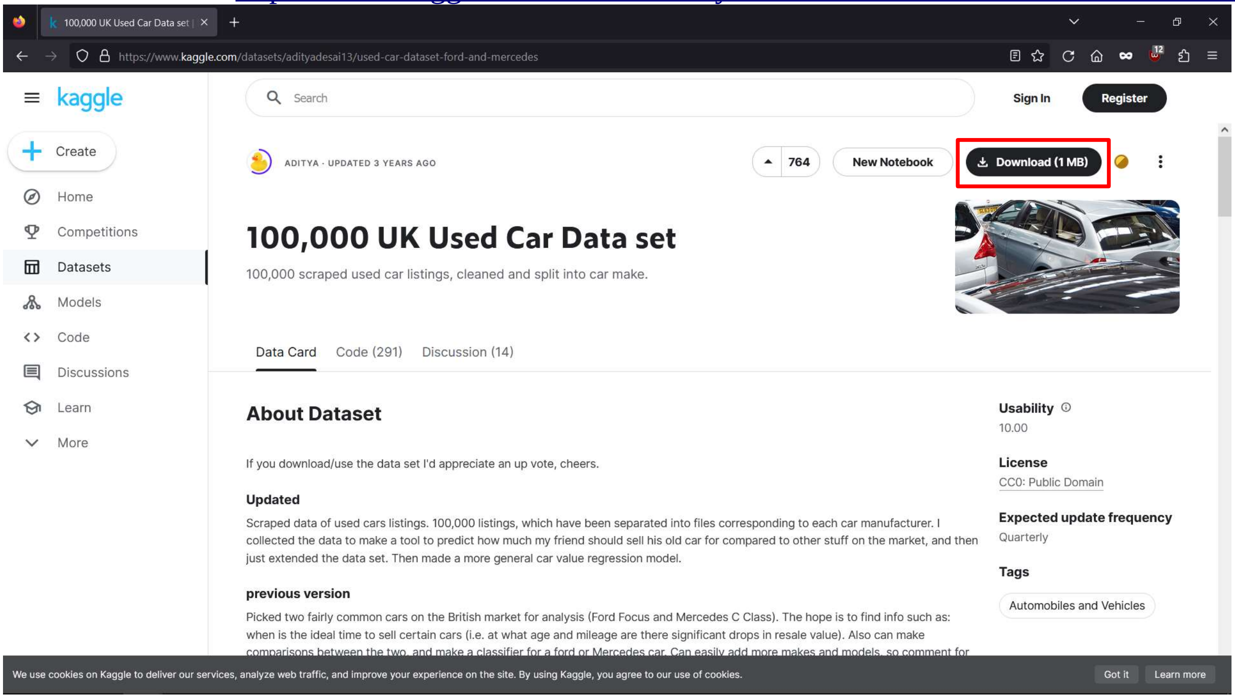
ETL to AWS RDS Project by Daniel Lee

Objective:

1. Create a python script to merge csv datasets that are pre-cleansed & have a common schema into a single file inclusive of some light automated transformation.
2. Host the dataset in AWS RDS MySQL database.
3. Using MySQL Workbench to load the data to the RDS.

Process:

i. Download the dataset at https://www.kaggle.com/datasets/adityadesai13/used-car-dataset-ford-and-mercedes



100,000 UK Used Car Data set

100,000 scraped used car listings, cleaned and split into car make.

About Dataset

If you download/use the data set I'd appreciate an up vote, cheers.

Updated

Scraped data of used cars listings. 100,000 listings, which have been separated into files corresponding to each car manufacturer. I collected the data to make a tool to predict how much my friend should sell his old car for compared to other stuff on the market, and then just extended the data set. Then made a more general car value regression model.

previous version

Picked two fairly common cars on the British market for analysis (Ford Focus and Mercedes C Class). The hope is to find info such as: when is the ideal time to sell certain cars (i.e. at what age and mileage are there significant drops in resale value). Also can make comparisons between the two, and make a classifier for a ford or Mercedes car. Can easily add more makes and models, so comment for

Usability 10.00

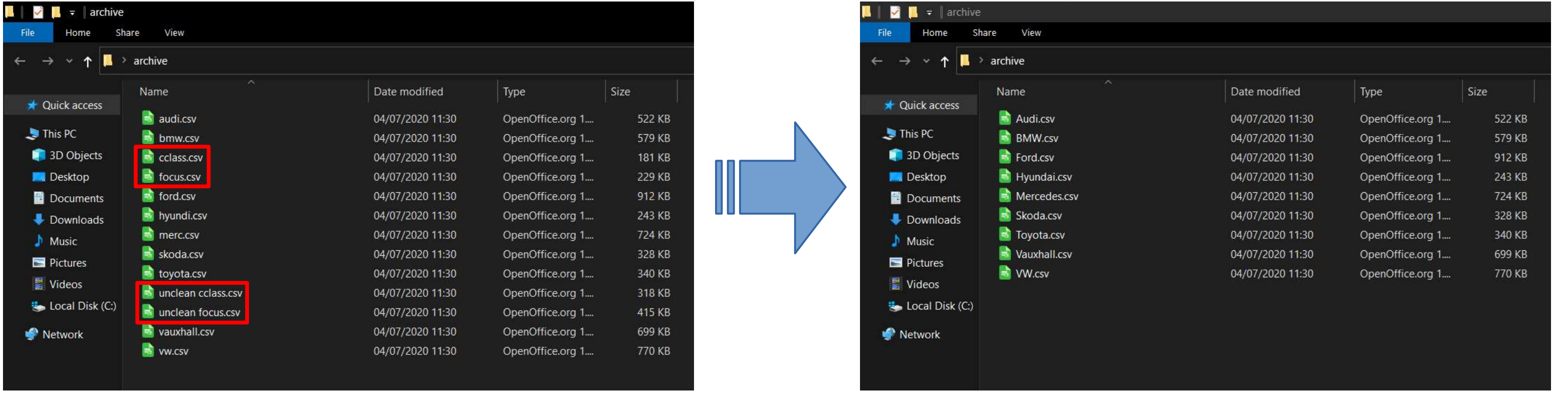
License CC0: Public Domain

Expected update frequency Quarterly

Tags Automobiles and Vehicles

We use cookies on Kaggle to deliver our services, analyze web traffic, and improve your experience on the site. By using Kaggle, you agree to our use of cookies.

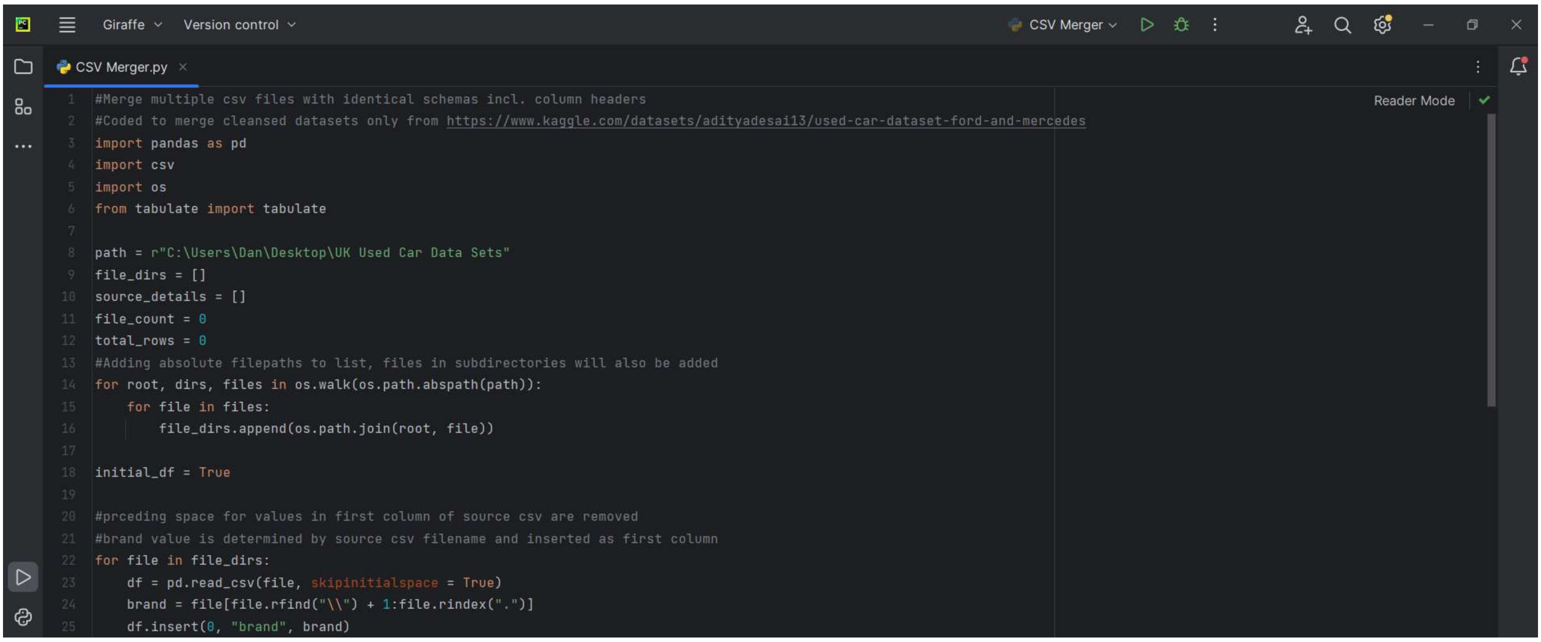
ii. Extract the contents. Delete the unclean and single model only files. Then tidy up the files names.



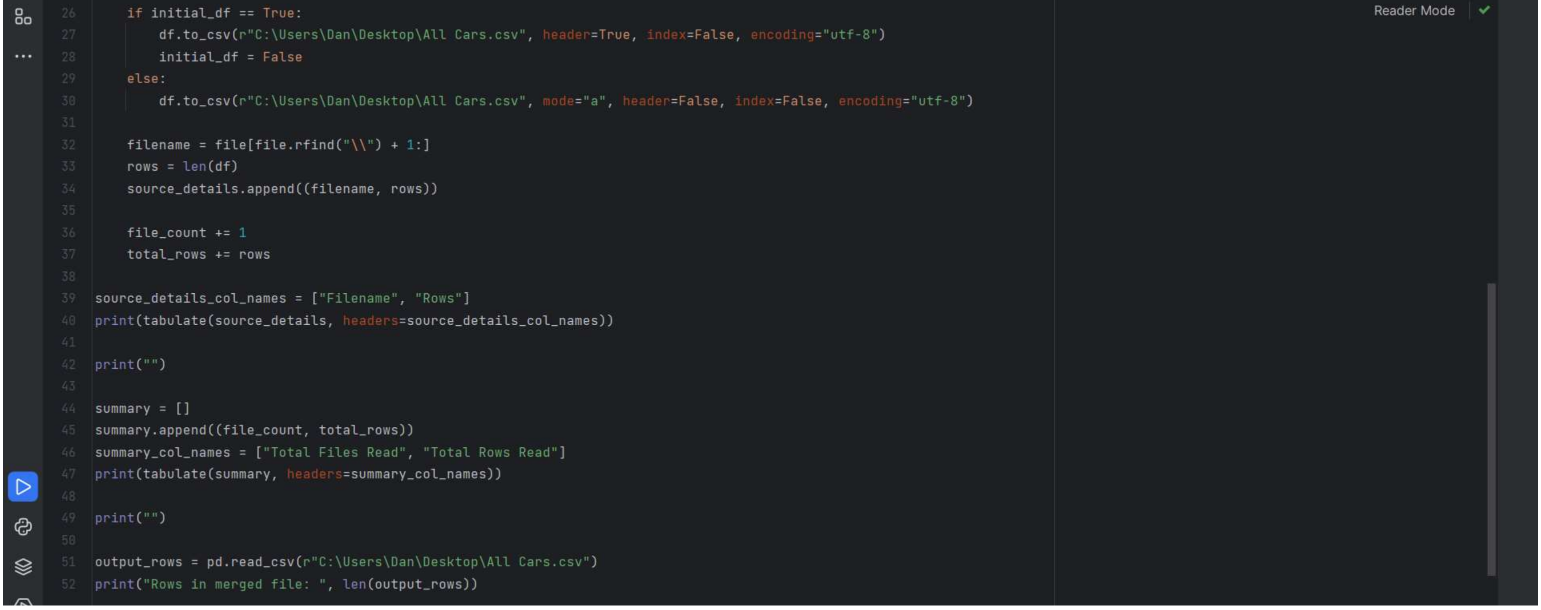
Name	Date modified	Type	Size
audi.csv	04/07/2020 11:30	OpenOffice.org 1...	522 KB
bmw.csv	04/07/2020 11:30	OpenOffice.org 1...	579 KB
cclass.csv	04/07/2020 11:30	OpenOffice.org 1...	181 KB
focus.csv	04/07/2020 11:30	OpenOffice.org 1...	229 KB
ford.csv	04/07/2020 11:30	OpenOffice.org 1...	912 KB
hyundi.csv	04/07/2020 11:30	OpenOffice.org 1...	243 KB
merc.csv	04/07/2020 11:30	OpenOffice.org 1...	724 KB
skoda.csv	04/07/2020 11:30	OpenOffice.org 1...	328 KB
toyota.csv	04/07/2020 11:30	OpenOffice.org 1...	340 KB
unclean cclass.csv	04/07/2020 11:30	OpenOffice.org 1...	318 KB
unclean focus.csv	04/07/2020 11:30	OpenOffice.org 1...	415 KB
vauxhall.csv	04/07/2020 11:30	OpenOffice.org 1...	699 KB
vw.csv	04/07/2020 11:30	OpenOffice.org 1...	770 KB

Name	Date modified	Type	Size
Audi.csv	04/07/2020 11:30	OpenOffice.org 1...	522 KB
BMW.csv	04/07/2020 11:30	OpenOffice.org 1...	579 KB
Ford.csv	04/07/2020 11:30	OpenOffice.org 1...	912 KB
Hyundai.csv	04/07/2020 11:30	OpenOffice.org 1...	243 KB
Mercedes.csv	04/07/2020 11:30	OpenOffice.org 1...	724 KB
Skoda.csv	04/07/2020 11:30	OpenOffice.org 1...	328 KB
Toyota.csv	04/07/2020 11:30	OpenOffice.org 1...	340 KB
Vauxhall.csv	04/07/2020 11:30	OpenOffice.org 1...	699 KB
VW.csv	04/07/2020 11:30	OpenOffice.org 1...	770 KB

iii. Create & run the python script to merge the files. This will also insert a new column for the brand which references the file name.

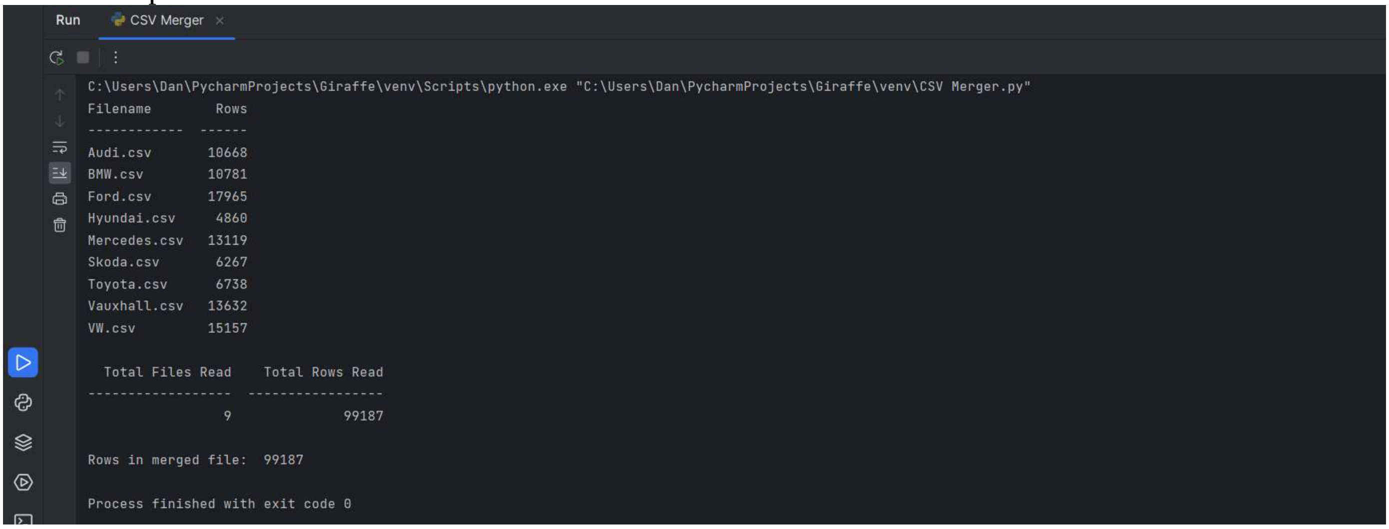


```python
#Merge multiple csv files with identical schemas incl. column headers
#Coded to merge cleansed datasets only from https://www.kaggle.com/datasets/adityadesai13/used-car-dataset-ford-and-mercedes
import pandas as pd
import csv
import os
from tabulate import tabulate

path = r"C:\Users\Dan\Desktop\UK Used Car Data Sets"
file_dirs = []
source_details = []
file_count = 0
total_rows = 0
#Adding absolute filepaths to list, files in subdirectories will also be added
for root, dirs, files in os.walk(os.path.abspath(path)):
    for file in files:
        file_dirs.append(os.path.join(root, file))

initial_df = True

#prceding space for values in first column of source csv are removed
#brand value is determined by source csv filename and inserted as first column
for file in file_dirs:
    df = pd.read_csv(file, skipinitialspace = True)
    brand = file[file.rfind("\\") + 1:file.rindex(".")]
    df.insert(0, "brand", brand)
```



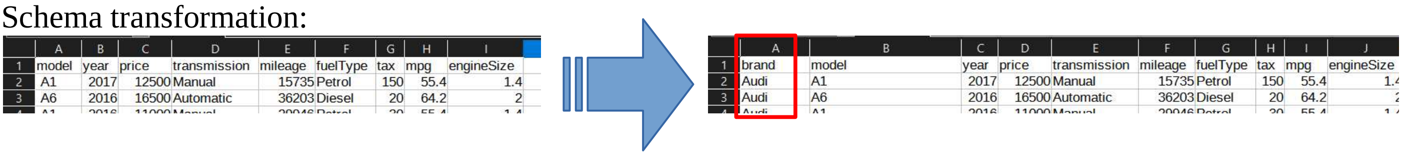
```
    if initial_df == True:
        df.to_csv(r"C:\Users\Dan\Desktop\All Cars.csv", header=True, index=False, encoding="utf-8")
        initial_df = False
    else:
        df.to_csv(r"C:\Users\Dan\Desktop\All Cars.csv", mode="a", header=False, index=False, encoding="utf-8")

    filename = file[file.rfind("\\") + 1:]
    rows = len(df)
    source_details.append((filename, rows))

    file_count += 1
    total_rows += rows

source_details_col_names = ["Filename", "Rows"]
print(tabulate(source_details, headers=source_details_col_names))

print("")

summary = []
summary.append((file_count, total_rows))
summary_col_names = ["Total Files Read", "Total Rows Read"]
print(tabulate(summary, headers=summary_col_names))

print("")

output_rows = pd.read_csv(r"C:\Users\Dan\Desktop\All Cars.csv")
print("Rows in merged file: ", len(output_rows))
```

Console output:



```
C:\Users\Dan\PycharmProjects\Giraffe\venv\Scripts\python.exe "C:\Users\Dan\PycharmProjects\Giraffe\venv\CSV Merger.py"
Filename         Rows
------------  ------
Audi.csv       10668
BMW.csv        10781
Ford.csv       17965
Hyundai.csv     4860
Mercedes.csv   13119
Skoda.csv       6267
Toyota.csv      6738
Vauxhall.csv   13632
VW.csv         15157

  Total Files Read    Total Rows Read
------------------  -----------------
                 9              99187

Rows in merged file:  99187

Process finished with exit code 0
```

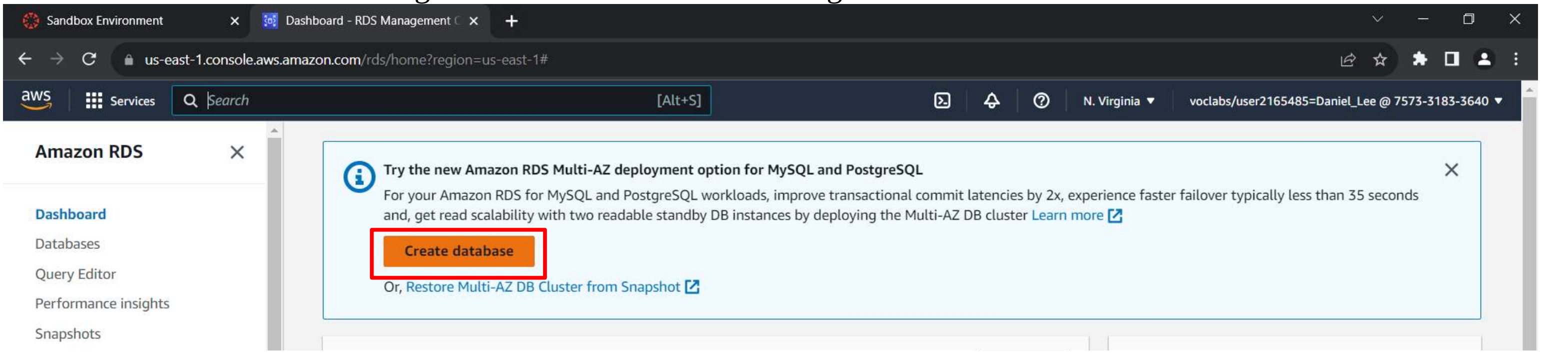
Schema transformation:



	A	B	C	D	E	F	G	H	I
1	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
2	A1	2017	12500	Manual	15735	Petrol	150	55.4	1.4
3	A6	2016	16500	Automatic	36203	Diesel	20	64.2	2

	A	B	C	D	E	F	G	H	I	J
1	brand	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
2	Audi	A1	2017	12500	Manual	15735	Petrol	150	55.4	1.4
3	Audi	A6	2016	16500	Automatic	36203	Diesel	20	64.2	2

iv. Go to the AWS RDS management console and start creating the database


Sandbox Environment
Dashboard - RDS Management C
us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#
aws
Services
Search
[Alt+S]
N. Virginia ▼
voclabs/user2165485=Daniel_Lee @ 7573-3183-3640 ▼
Amazon RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots

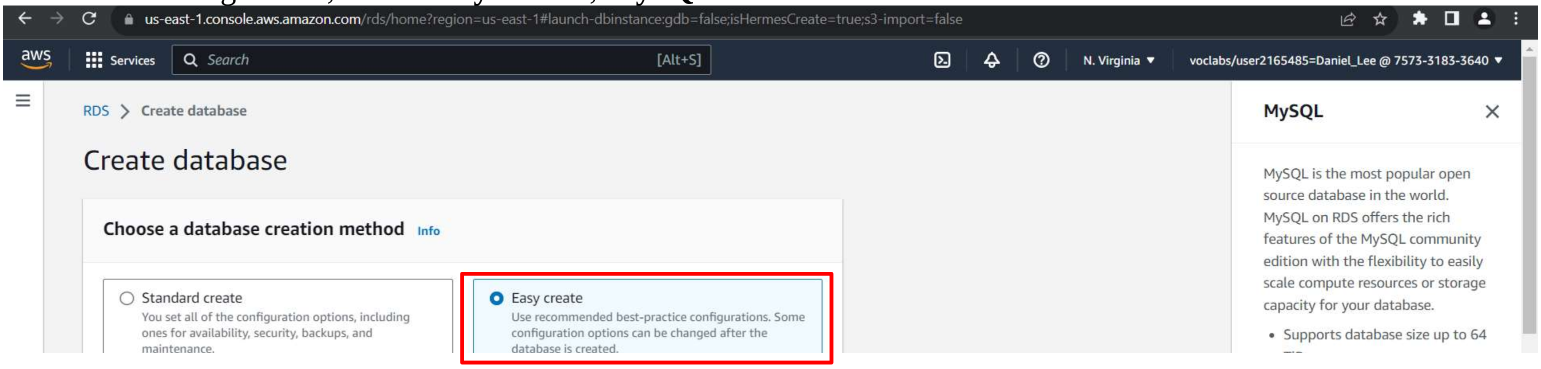
Try the new Amazon RDS Multi-AZ deployment option for MySQL and PostgreSQL

For your Amazon RDS for MySQL and PostgreSQL workloads, improve transactional commit latencies by 2x, experience faster failover typically less than 35 seconds and, get read scalability with two readable standby DB instances by deploying the Multi-AZ DB cluster Learn more

Create database

Or, Restore Multi-AZ DB Cluster from Snapshot

v. In the configuration, select **Easy Create**, **MySQL** & **Free Tier**


us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:gdb=false;isHermesCreate=true;s3-import=false
aws
Services
Search
[Alt+S]
N. Virginia ▼
voclabs/user2165485=Daniel_Lee @ 7573-3183-3640 ▼
RDS
Create database

Create database

Choose a database creation method Info

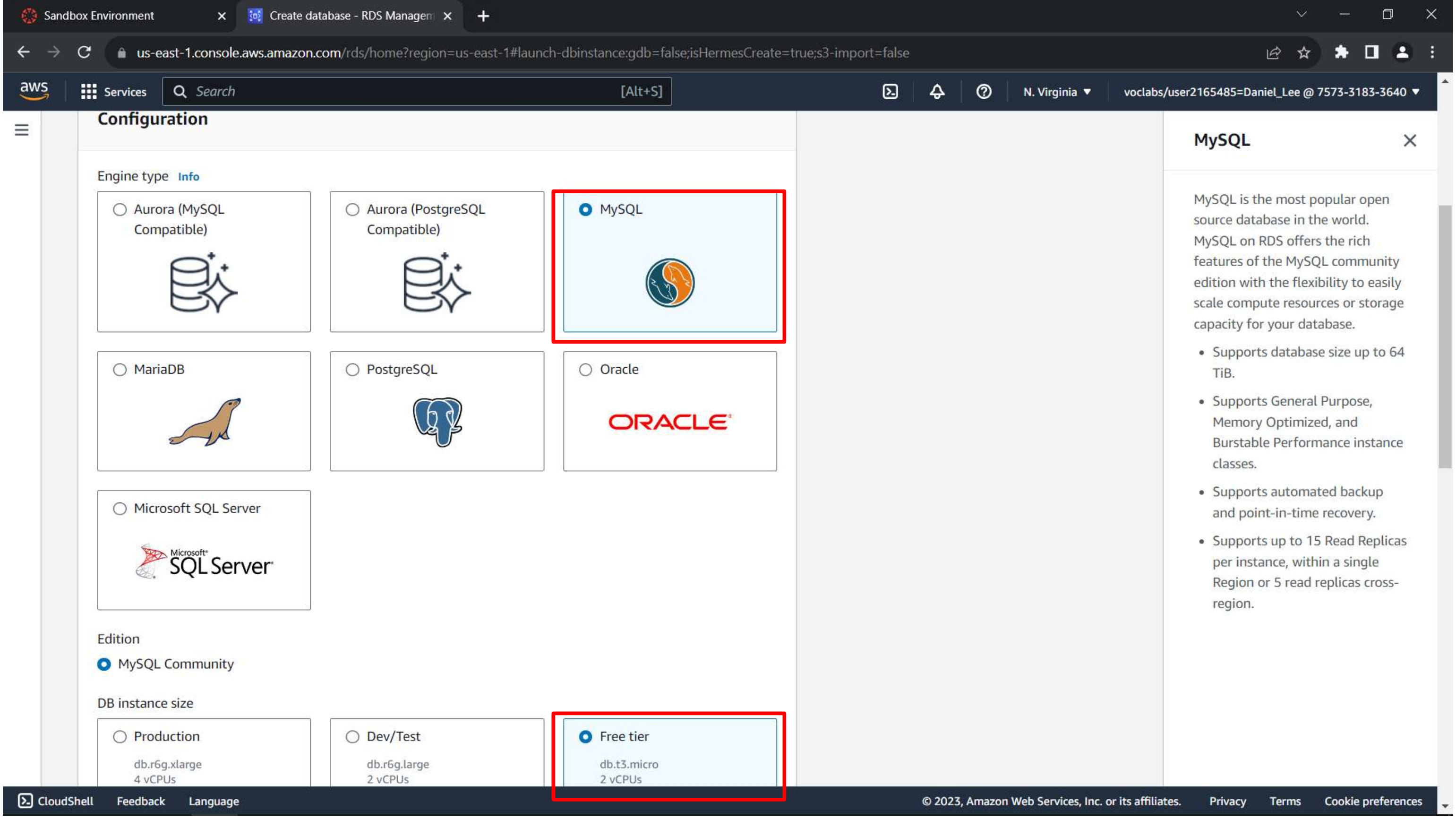
○ Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

● Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

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


Configuration

Engine type Info

○ Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

● MySQL

○ MariaDB

○ PostgreSQL

○ Oracle

○ Microsoft SQL Server

Edition

● MySQL Community

DB instance size

○ Production
db.r6g.xlarge
4 vCPUs

○ Dev/Test
db.r6g.large
2 vCPUs

● Free tier
db.t3.micro
2 vCPUs

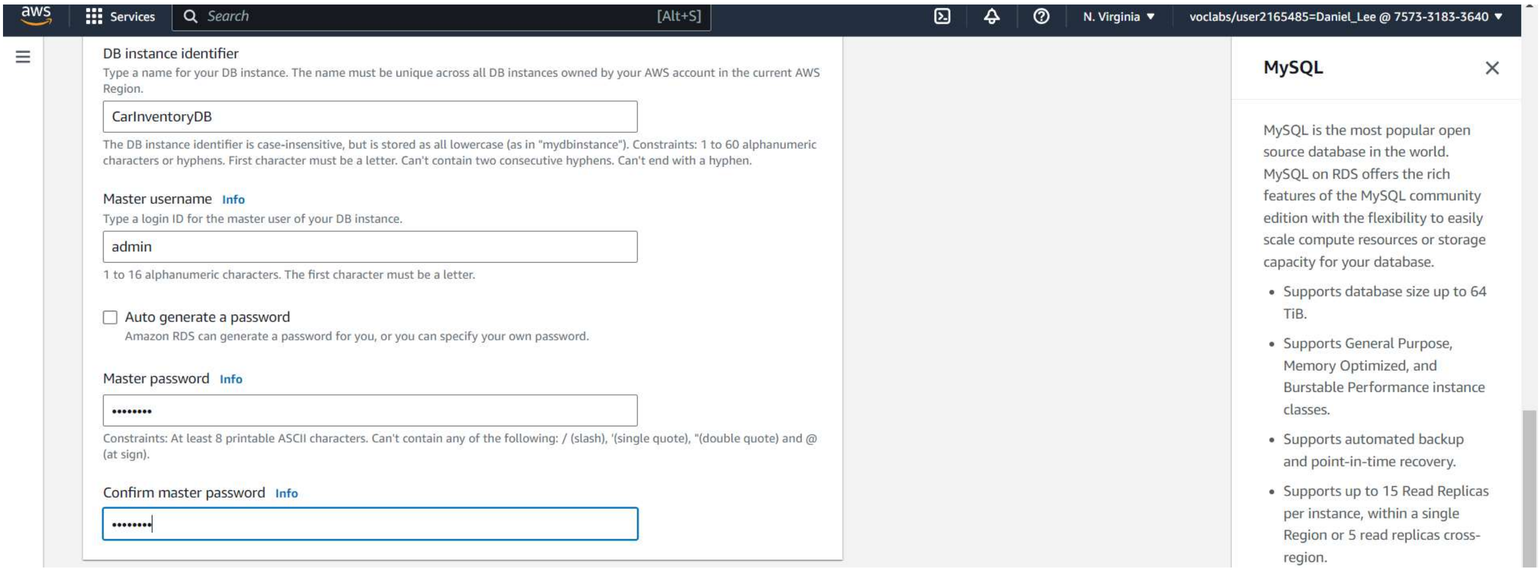
MySQL

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.

Sandbox Environment
Create database - RDS Managem
us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:gdb=false;isHermesCreate=true;s3-import=false
Services
Search
[Alt+S]
N. Virginia
voclabs/user2165485=Daniel_Lee @ 7573-3183-3640
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

vi. Set a database name, then the master username and password to *admin* & *password* respectively. Select the **Create Database** button at the bottom.


DB instance identifier
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

CarInventoryDB

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username Info
Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. The first character must be a letter.

□ 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

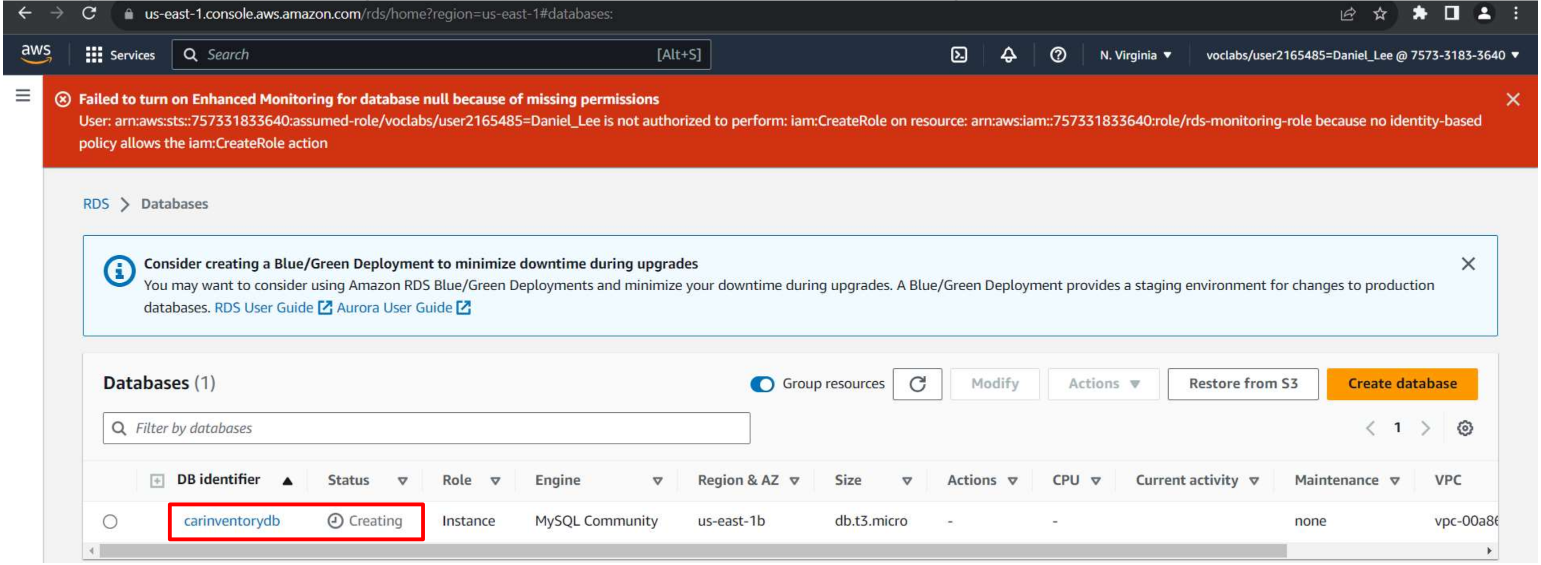
••••••••

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.

vii. Ignore the error message at the top in red. Wait for the status to change to available then click on the database name


Failed to turn on Enhanced Monitoring for database null because of missing permissions
User: arn:aws:sts::757331833640:assumed-role/voclabs/user2165485=Daniel_Lee is not authorized to perform: iam:CreateRole on resource: arn:aws:iam::757331833640:role/rds-monitoring-role because no identity-based policy allows the iam:CreateRole action

RDS > Databases

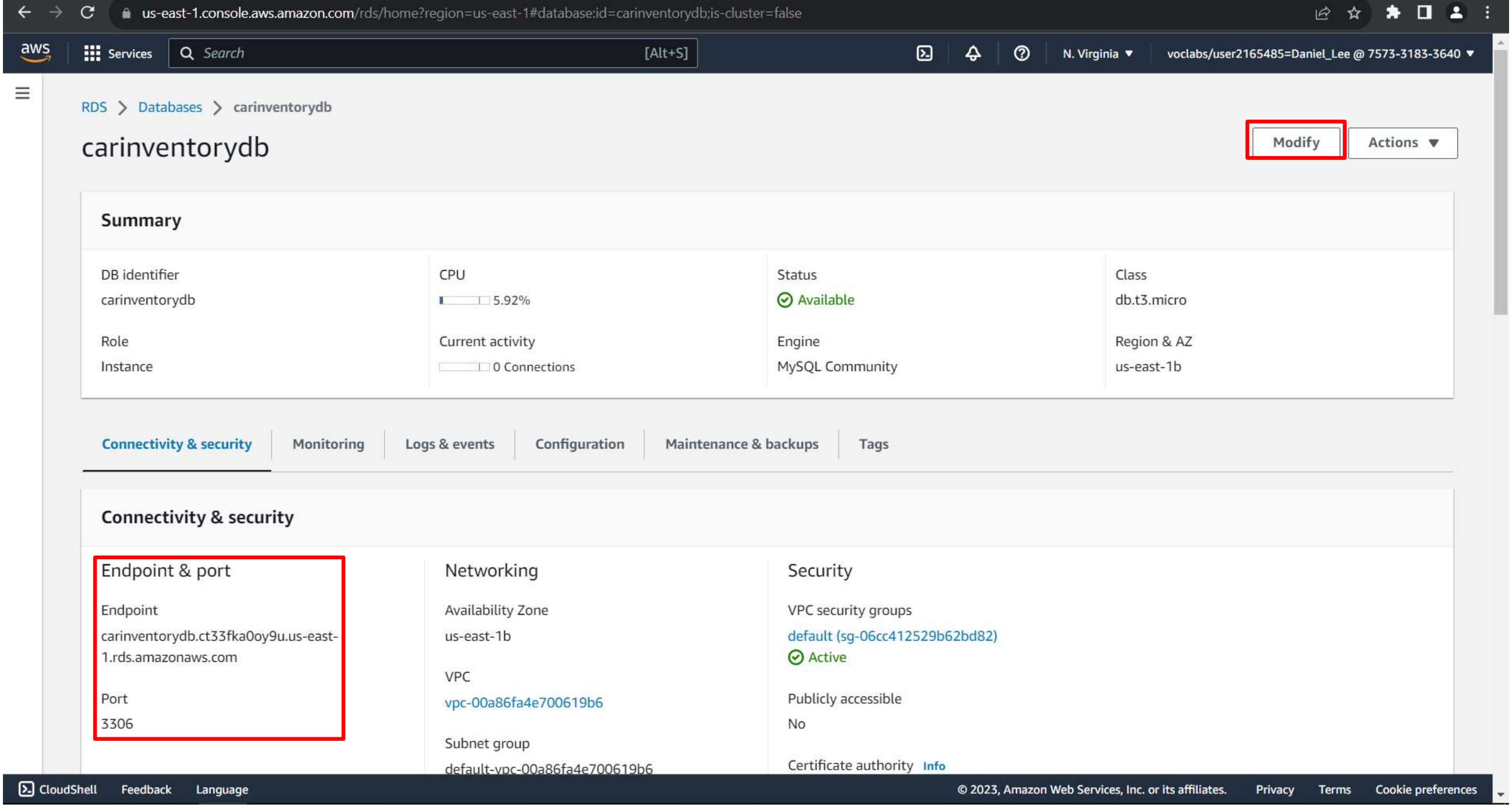
Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora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

Databases (1)

	DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU	Current activity	Maintenance	VPC
○	carinventorydb	Creating	Instance	MySQL Community	us-east-1b	db.t3.micro	-	-		none	vpc-00a86

us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases:
Services
Search
[Alt+S]
N. Virginia
voclabs/user2165485=Daniel_Lee @ 7573-3183-3640
Group resources
Modify
Actions
Restore from S3
Create database
Filter by databases
1

viii. You should now see an Endpoint. Copy and paste into Notepad for future reference. Click on Modify



us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#database:id=carinventorydb;is-cluster=false
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user2165485=Daniel_Lee @ 7573-3183-3640
RDS > Databases > carinventorydb
Modify
Actions
Connectivity & security
Monitoring
Logs & events
Configuration
Maintenance & backups
Tags
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

carinventorydb

Summary

DB identifier carinventorydb	CPU 5.92%	Status Available	Class db.t3.micro
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ us-east-1b

Connectivity & security

Endpoint & port

Endpoint
carinventorydb.ct33fka0oy9u.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1b

VPC
vpc-00a86fa4e700619b6

Subnet group
default-vpc-00a86fa4e700619b6

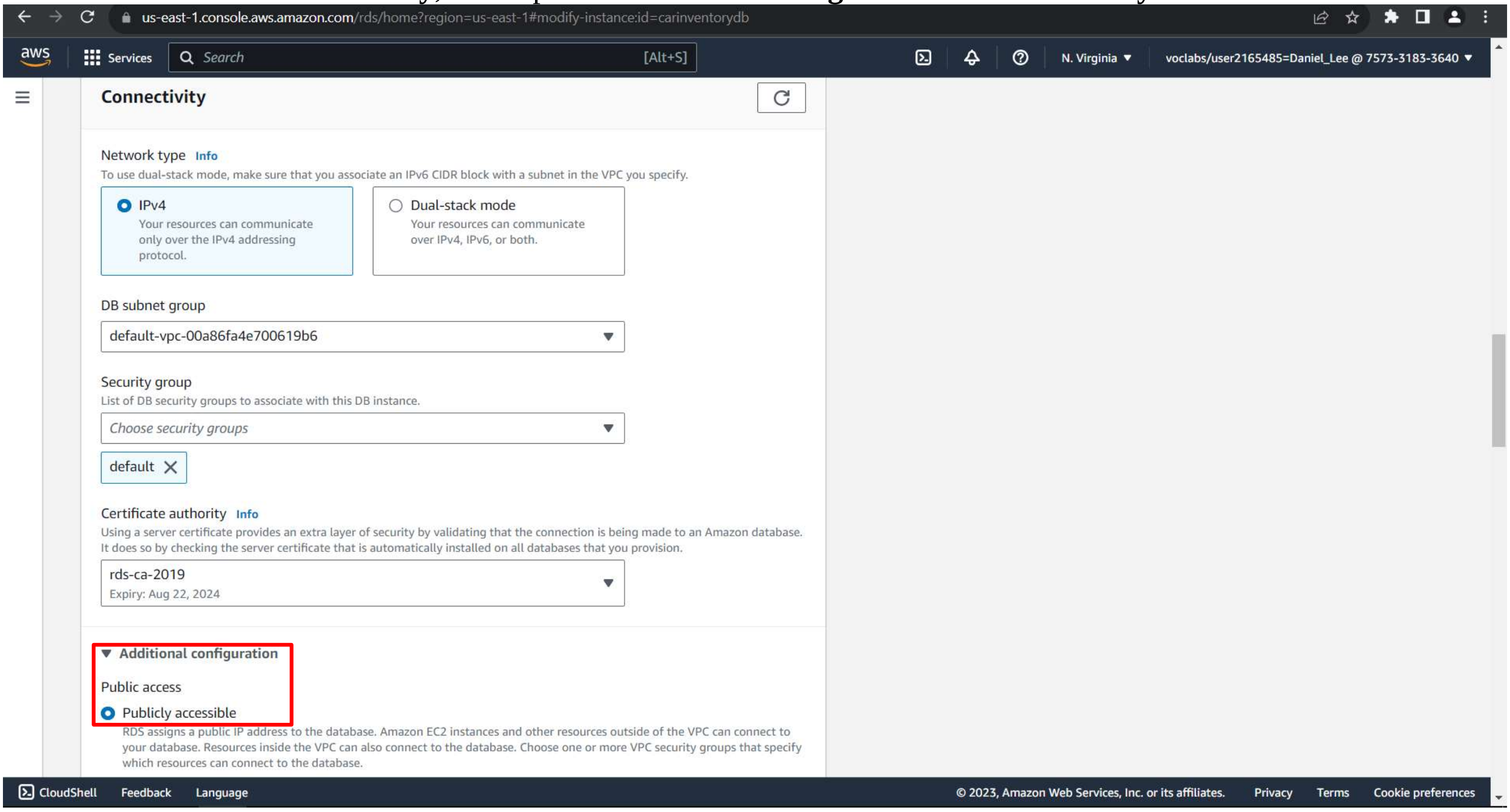
Security

VPC security groups
default (sg-06cc412529b62bd82)
Active

Publicly accessible
No

Certificate authority Info

ix. Scroll down to **Connectivity**, and expand **Additional Configuration**. Select **Publicly accessible**



us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#modify-instance:id=carinventorydb
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user2165485=Daniel_Lee @ 7573-3183-3640
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Connectivity

Network type Info
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

● IPv4
Your resources can communicate only over the IPv4 addressing protocol.

○ Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

DB subnet group
default-vpc-00a86fa4e700619b6

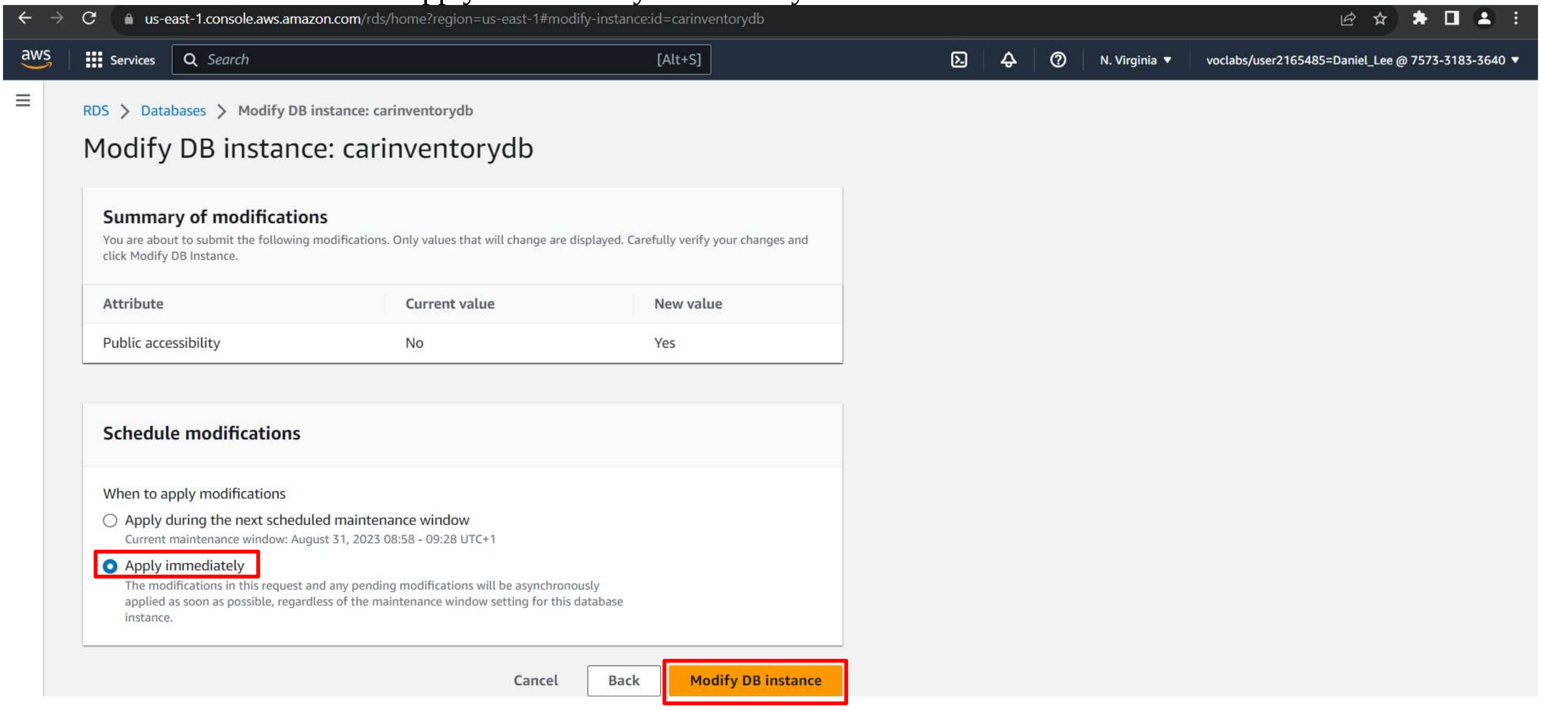
Security group
List of DB security groups to associate with this DB instance.
Choose security groups
default ✕

Certificate authority Info
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.
rds-ca-2019
Expiry: Aug 22, 2024

▼ Additional configuration

Public access
● Publicly accessible
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

x. On the next screen select Apply Immediately and Modify DB Instance


us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#modify-instance:id=carinventorydb
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user2165485=Daniel_Lee @ 7573-3183-3640
RDS
Databases
Modify DB instance: carinventorydb
Cancel
Back
Modify DB instance

Modify DB instance: carinventorydb

Summary of modifications

You are about to submit the following modifications. Only values that will change are displayed. Carefully verify your changes and click Modify DB Instance.

Attribute	Current value	New value
Public accessibility	No	Yes

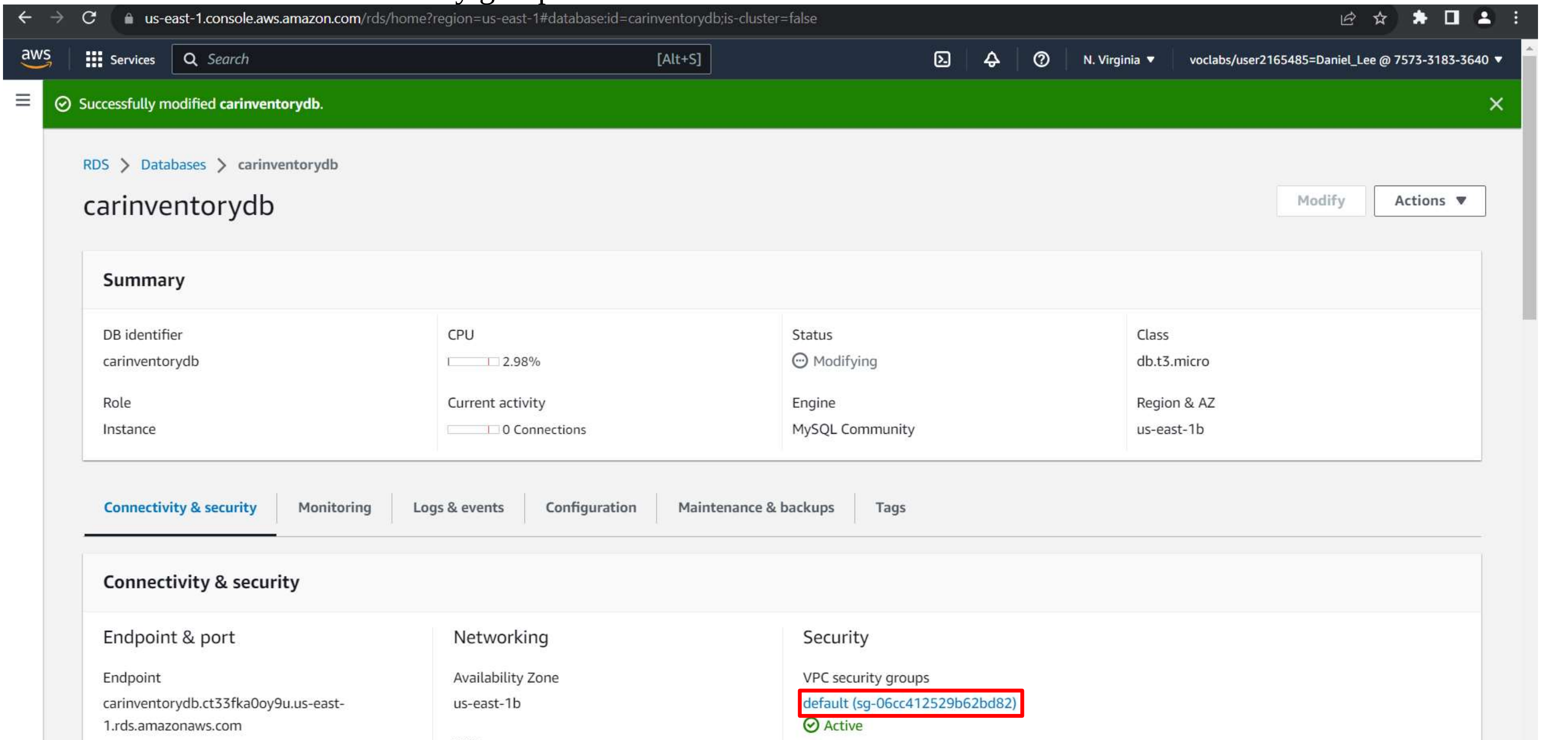
Schedule modifications

When to apply modifications

○ Apply during the next scheduled maintenance window
Current maintenance window: August 31, 2023 08:58 - 09:28 UTC+1

● Apply immediately
The modifications in this request and any pending modifications will be asynchronously applied as soon as possible, regardless of the maintenance window setting for this database instance.

xi. Now click on the VPC security group



Successfully modified **carinventorydb**.

carinventorydb

Summary

DB identifier	CPU	Status	Class
carinventorydb	2.98%	Modifying	db.t3.micro
Role	**Current activity**	**Engine**	**Region & AZ**
Instance	0 Connections	MySQL Community	us-east-1b

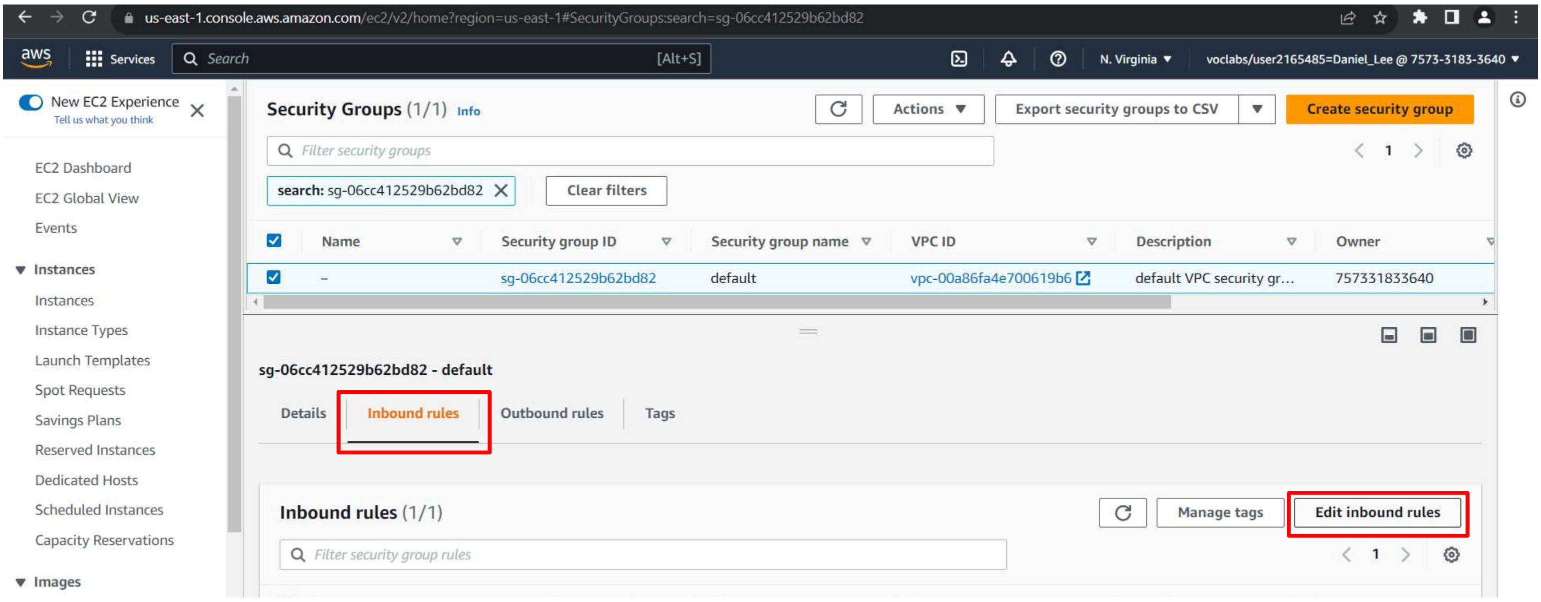
Connectivity & security

Endpoint & port
Endpoint
carinventorydb.ct33fka0oy9u.us-east-1.rds.amazonaws.com

Networking
Availability Zone
us-east-1b

Security
VPC security groups
default (sg-06cc412529b62bd82)
Active

xii. Select the inbound rules tab then Edit Inbound Rules



Security Groups (1/1) Info

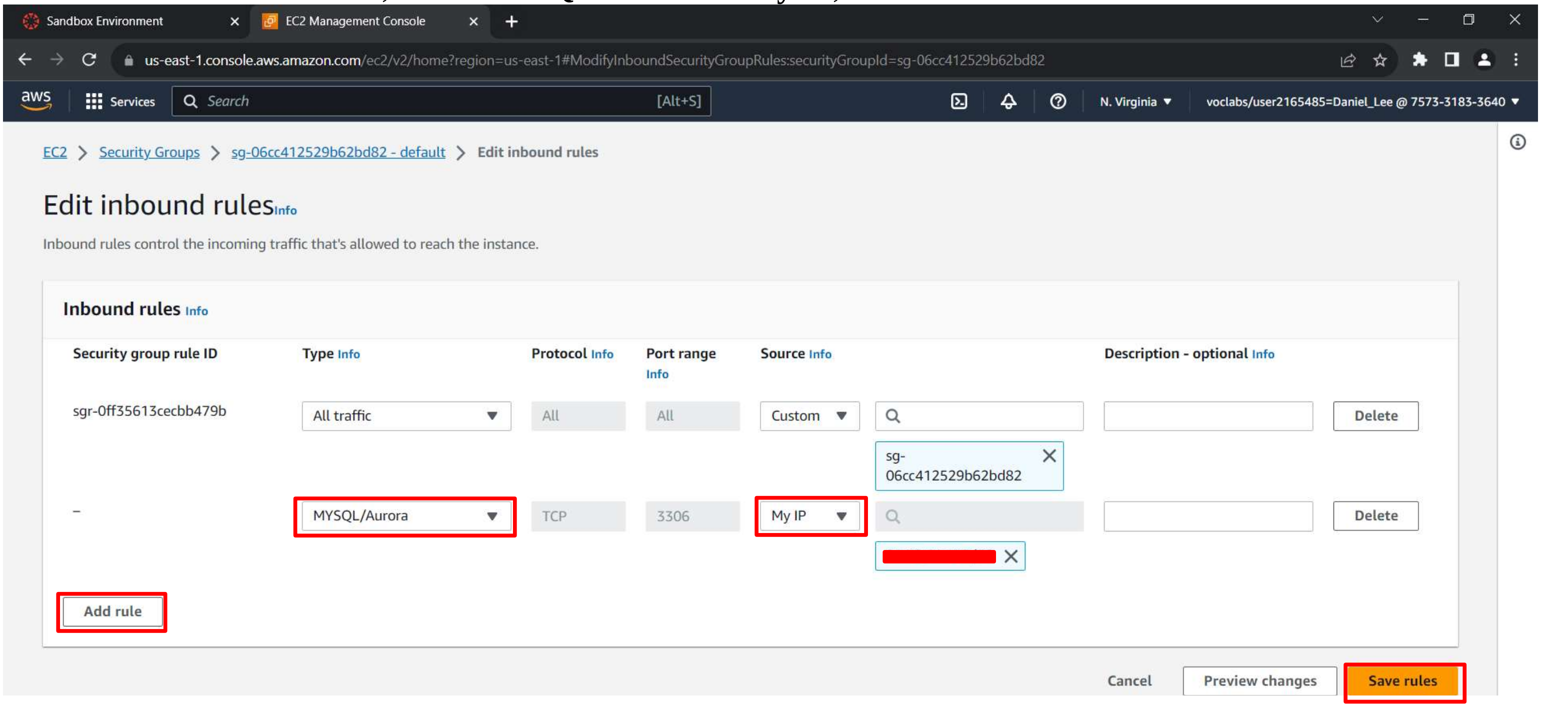
search: sg-06cc412529b62bd82

☑	Name	Security group ID	Security group name	VPC ID	Description	Owner
☑	–	sg-06cc412529b62bd82	default	vpc-00a86fa4e700619b6	default VPC security gr...	757331833640

sg-06cc412529b62bd82 - default

Inbound rules (1/1)

xiii. Click **Add Rule**, select **MYSQL/Aurora** & **My IP**, then **Save rules**


Sandbox Environment
EC2 Management Console
us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-06cc412529b62bd82
Services
Search [Alt+S]
N. Virginia
voclabs/user2165485=Daniel_Lee @ 7573-3183-3640
EC2 > Security Groups > sg-06cc412529b62bd82 - default > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-0ff35613cecbb479b	All traffic	All	All	Custom	sg-06cc412529b62bd82		Delete
–	MYSQL/Aurora	TCP	3306	My IP	[redacted]		Delete

Add rule
Cancel
Preview changes
Save rules

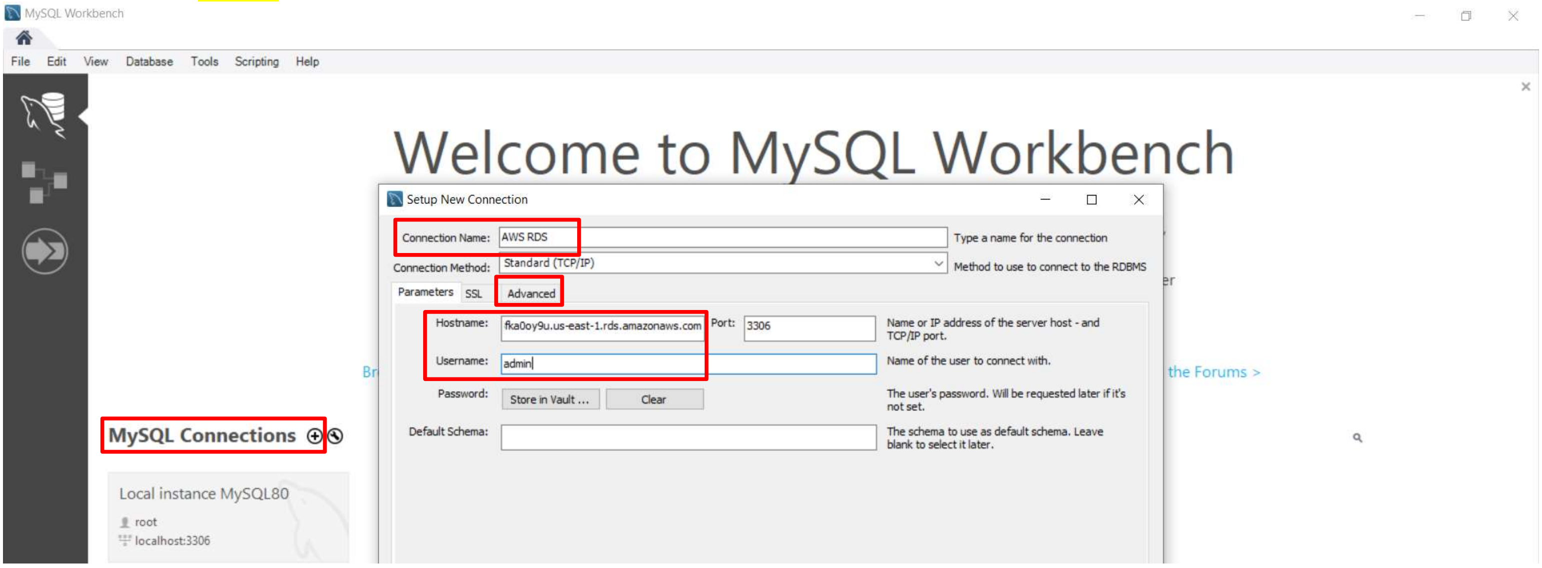
xiv. Open MySQL Workbench

i. Add a new MySQL connection

ii. Enter a Connection Name of your choice

iii. Retrieve the Endpoint from Notepad and paste as the Hostname

iv. Enter admin for Username



Connection Name: AWS RDS — Type a name for the connection

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Hostname: fka0oy9u.us-east-1.rds.amazonaws.com Port: 3306 — Name or IP address of the server host - and TCP/IP port.

Username: admin — Name of the user to connect with.

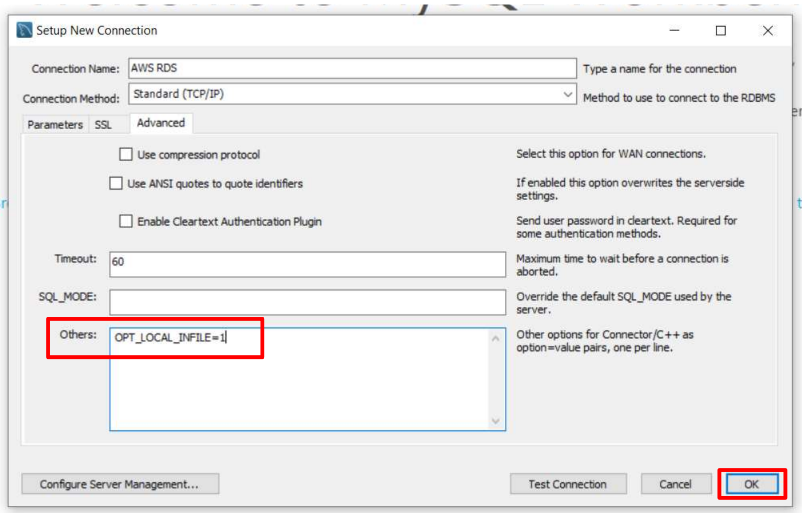
Password: — The user's password. Will be requested later if it's not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.

v. Select the Advanced tab

vi. enter OPT_LOCAL_INFILE=1 into the Others box

vii. Click OK



Connection Name: AWS RDS — Type a name for the connection

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

☐ Use compression protocol — Select this option for WAN connections.

☐ Use ANSI quotes to quote identifiers — If enabled this option overwrites the serverside settings.

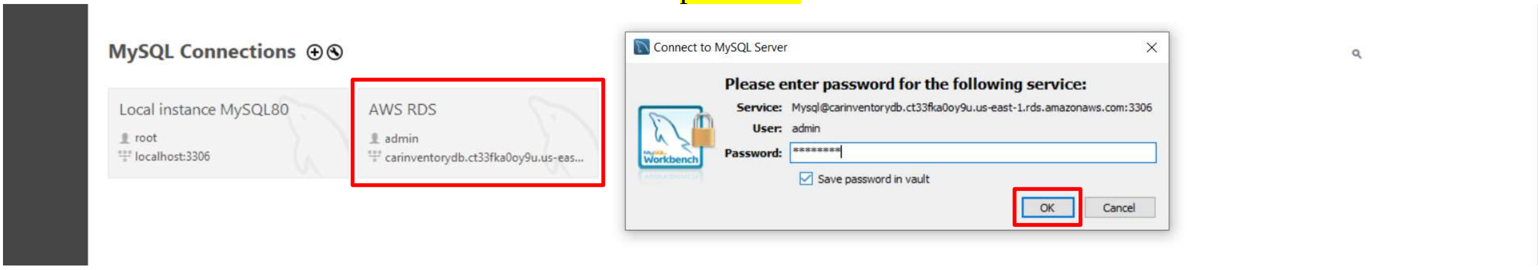
☐ Enable Cleartext Authentication Plugin — Send user password in cleartext. Required for some authentication methods.

Timeout: 60 — Maximum time to wait before a connection is aborted.

SQL_MODE: — Override the default SQL_MODE used by the server.

Others: OPT_LOCAL_INFILE=1 — Other options for Connector/C++ as option=value pairs, one per line.

xv. Double-click on the new connection then enter password as the Password and click OK

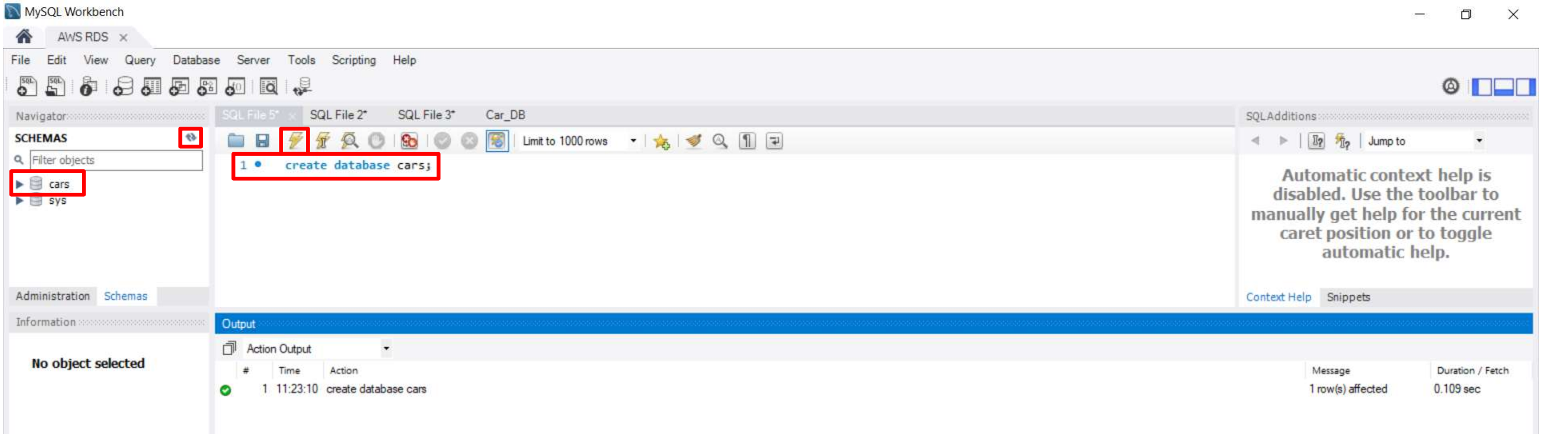


Please enter password for the following service:

Service: Mysql@carinventorydb.ct33fka0oy9u.us-east-1.rds.amazonaws.com:3306
User: admin
Password: ********

☑ Save password in vault

xvi. Create a new database by running the following query

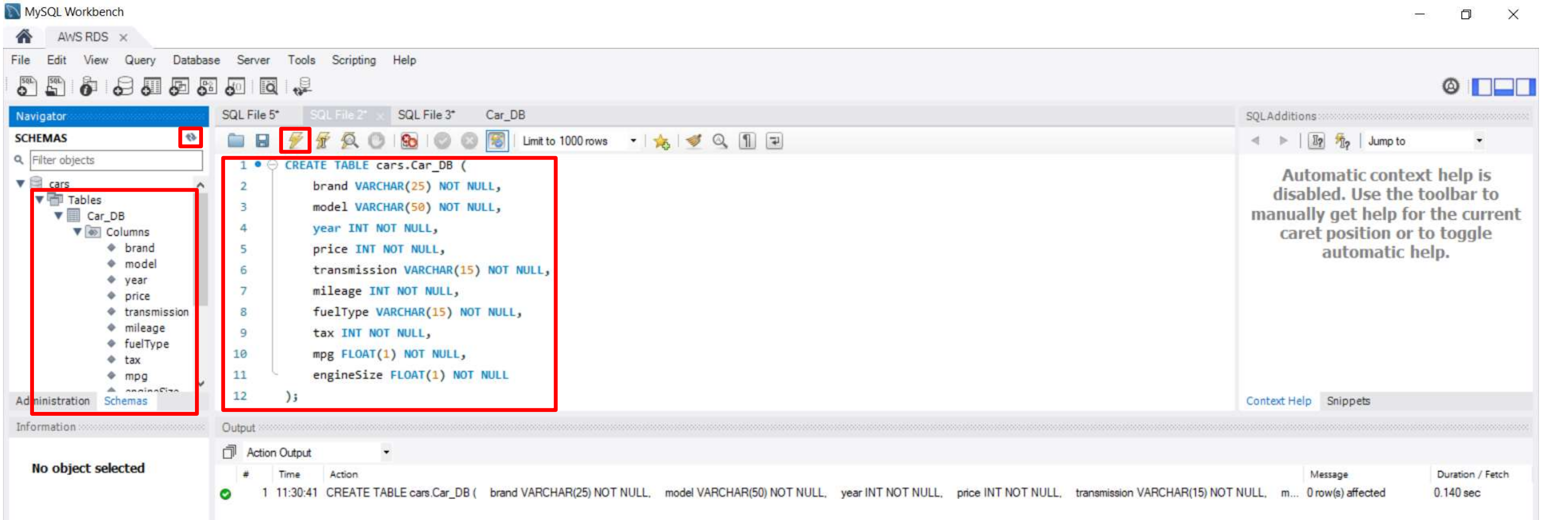


```
create database cars;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
#   Time      Action                  Message            Duration / Fetch
1   11:23:10  create database cars    1 row(s) affected  0.109 sec
```

xvii. Create a new table in the new database by running the following query

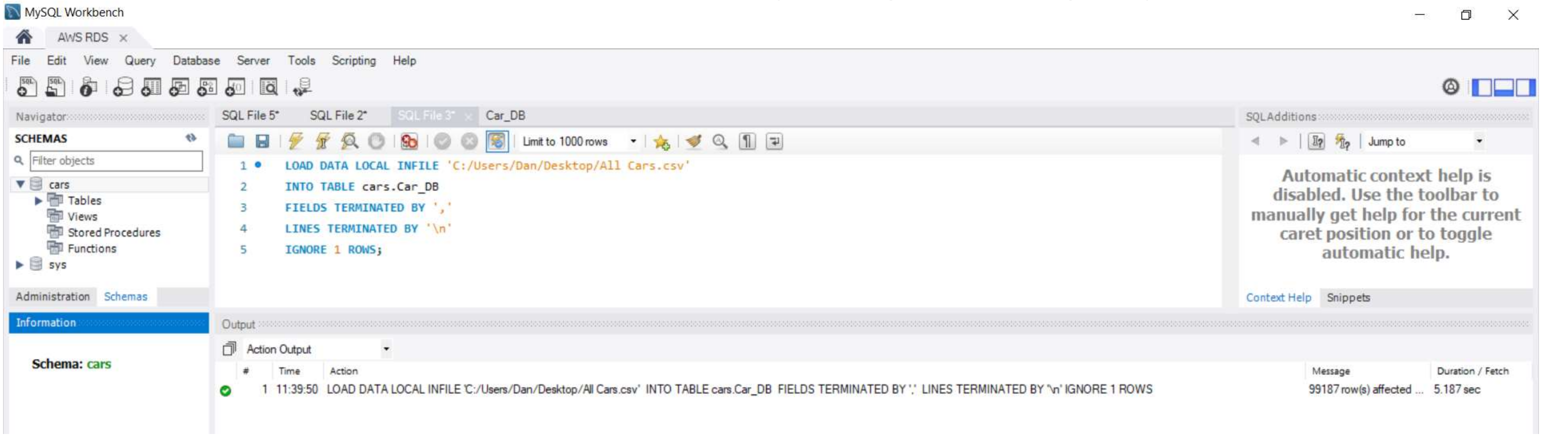


```
CREATE TABLE cars.Car_DB (
    brand VARCHAR(25) NOT NULL,
    model VARCHAR(50) NOT NULL,
    year INT NOT NULL,
    price INT NOT NULL,
    transmission VARCHAR(15) NOT NULL,
    mileage INT NOT NULL,
    fuelType VARCHAR(15) NOT NULL,
    tax INT NOT NULL,
    mpg FLOAT(1) NOT NULL,
    engineSize FLOAT(1) NOT NULL
);
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
#   Time      Action                                                                                                                                              Message            Duration / Fetch
1   11:30:41  CREATE TABLE cars.Car_DB (   brand VARCHAR(25) NOT NULL,   model VARCHAR(50) NOT NULL,   year INT NOT NULL,   price INT NOT NULL,   transmission VARCHAR(15) NOT NULL,   m...  0 row(s) affected  0.140 sec
```

xviii. Load the new table with the data from the csv by running the following query



```
LOAD DATA LOCAL INFILE 'C:/Users/Dan/Desktop/All Cars.csv'
INTO TABLE cars.Car_DB
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

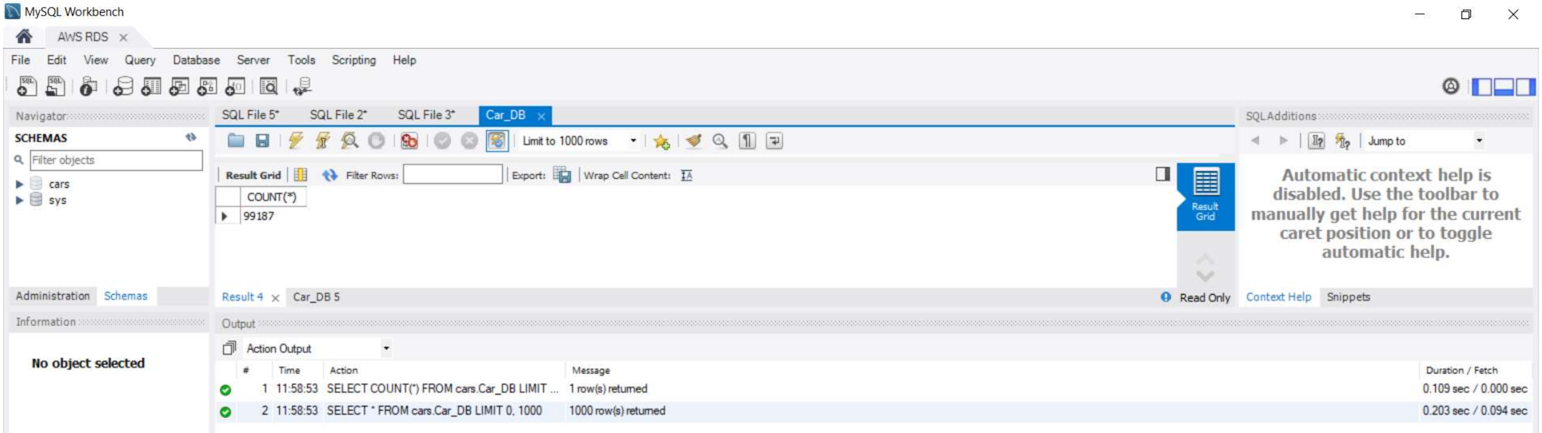
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Schema: cars

```
#   Time      Action                                                                                                                                Message                  Duration / Fetch
1   11:39:50  LOAD DATA LOCAL INFILE 'C:/Users/Dan/Desktop/All Cars.csv'  INTO TABLE cars.Car_DB  FIELDS TERMINATED BY ','  LINES TERMINATED BY '\n' IGNORE 1 ROWS   99187 row(s) affected ...  5.187 sec
```

xix. Run a couple of checks:

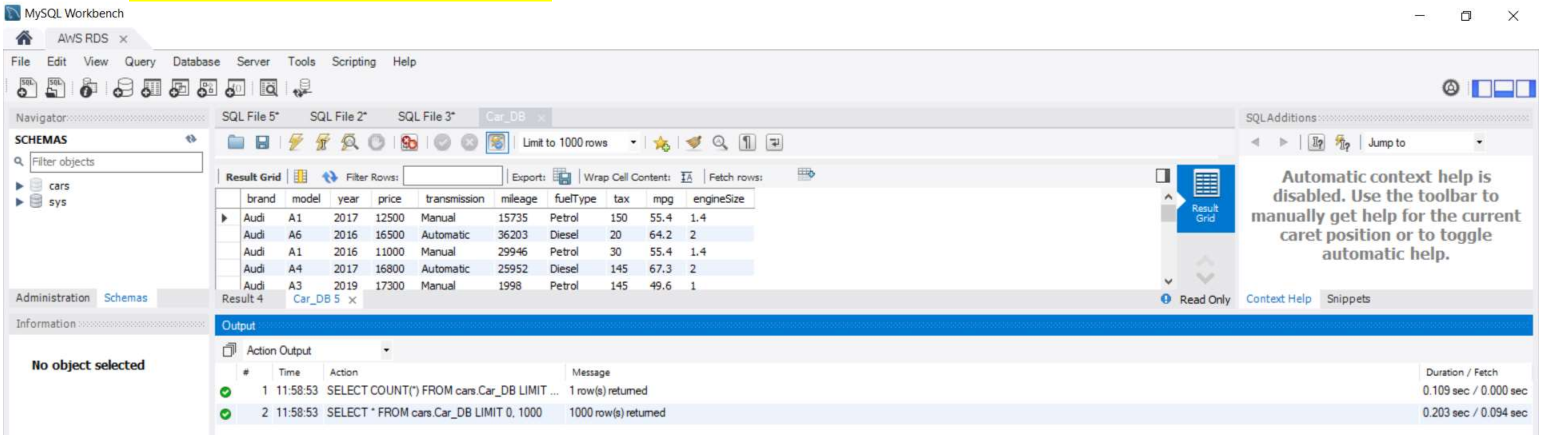
i. SELECT COUNT(*) FROM cars.Car_DB;



	COUNT(*)
▶	99187

```
#  Time      Action                                    Message            Duration / Fetch
1  11:58:53  SELECT COUNT(*) FROM cars.Car_DB LIMIT ...  1 row(s) returned  0.109 sec / 0.000 sec
2  11:58:53  SELECT * FROM cars.Car_DB LIMIT 0, 1000     1000 row(s) returned  0.203 sec / 0.094 sec
```

ii. SELECT * FROM cars.Car_DB;



brand	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
Audi	A1	2017	12500	Manual	15735	Petrol	150	55.4	1.4
Audi	A6	2016	16500	Automatic	36203	Diesel	20	64.2	2
Audi	A1	2016	11000	Manual	29946	Petrol	30	55.4	1.4
Audi	A4	2017	16800	Automatic	25952	Diesel	145	67.3	2
Audi	A3	2019	17300	Manual	1998	Petrol	145	49.6	1

```
#  Time      Action                                    Message            Duration / Fetch
1  11:58:53  SELECT COUNT(*) FROM cars.Car_DB LIMIT ...  1 row(s) returned  0.109 sec / 0.000 sec
2  11:58:53  SELECT * FROM cars.Car_DB LIMIT 0, 1000     1000 row(s) returned  0.203 sec / 0.094 sec
```