

McMaster
University



SEP 721 – Data Analytics, Machine Learning and AI on
Cloud Platforms

Assignment 2: Qwiklabs- 1,2 and 3

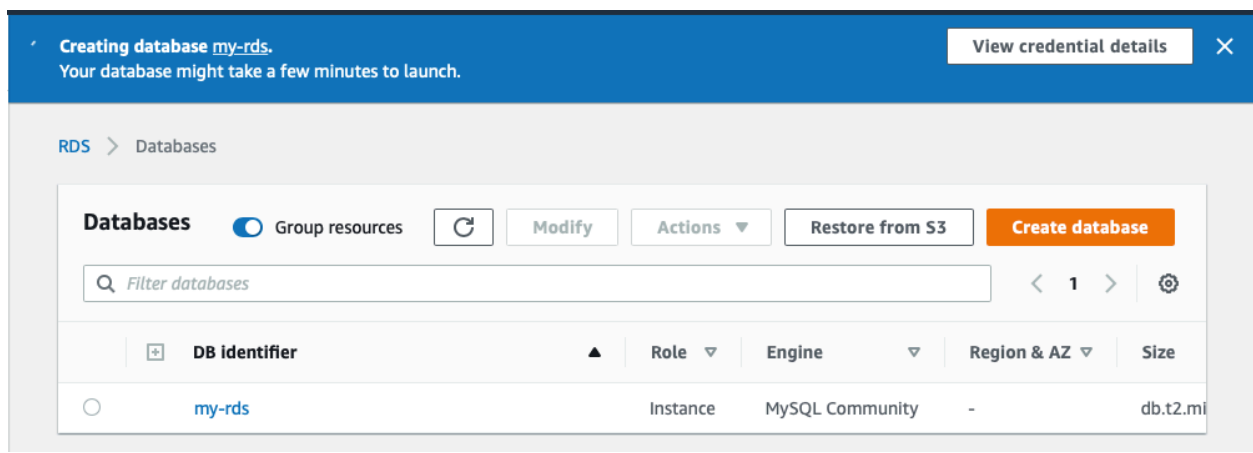
Submitted by,

Greeshma Gopal(gopalg)

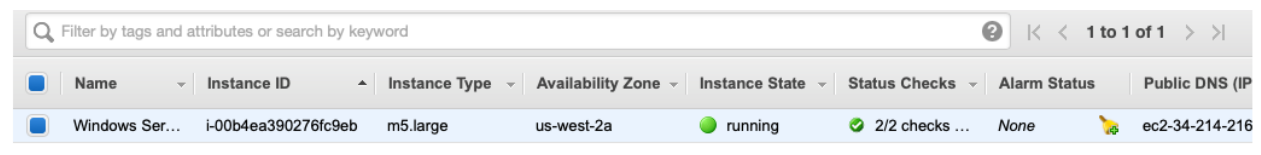
ID- 400245291

Lab 1: Introduction to Amazon Relational Database Service (RDS) (Windows)

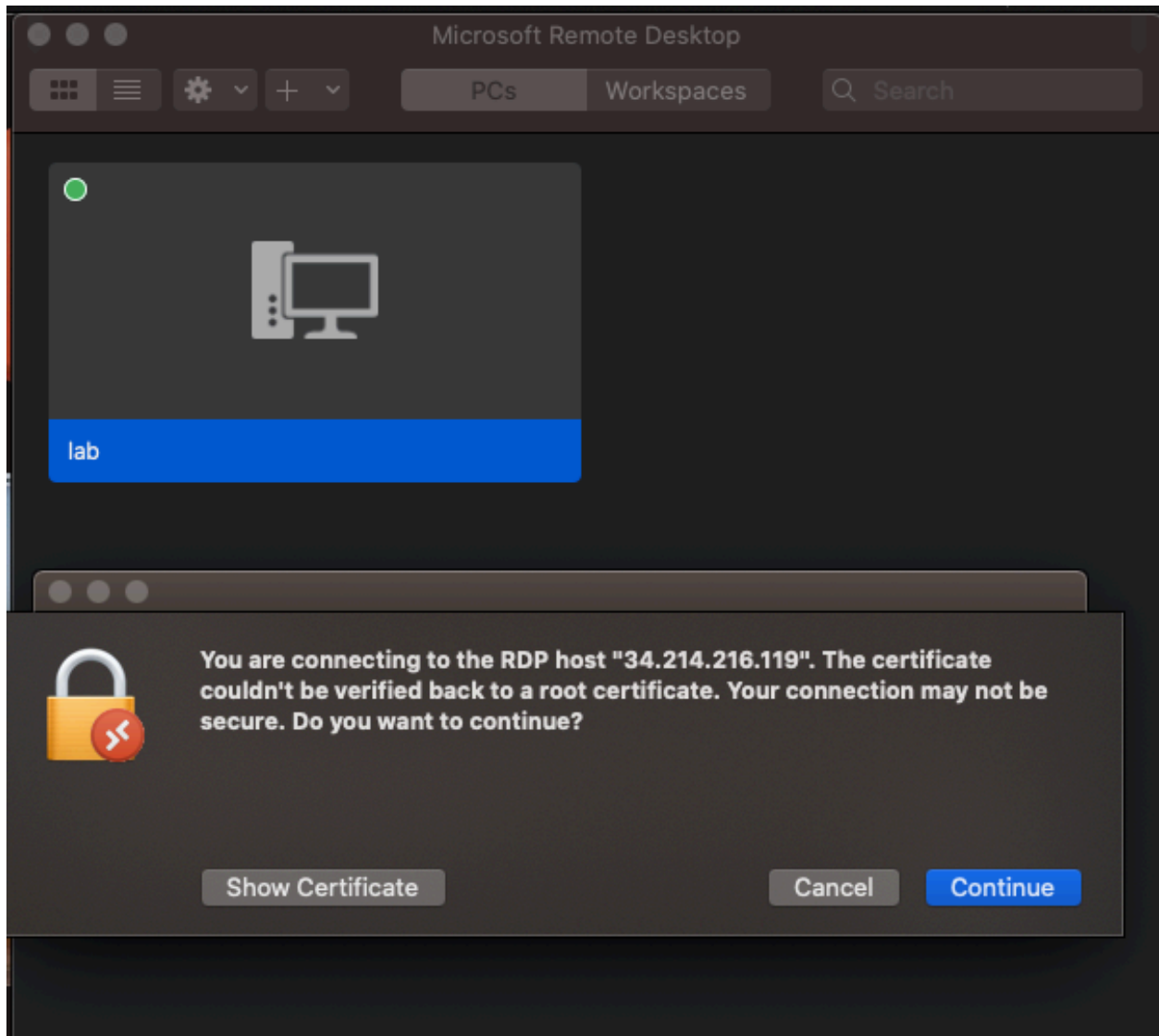
- I have used qwiklab credits to get the login credentials. The database was created with the configuration provided



- Logging in to EC2 instance. After the database was created, the EC2 instance which was created has to be logged in to.



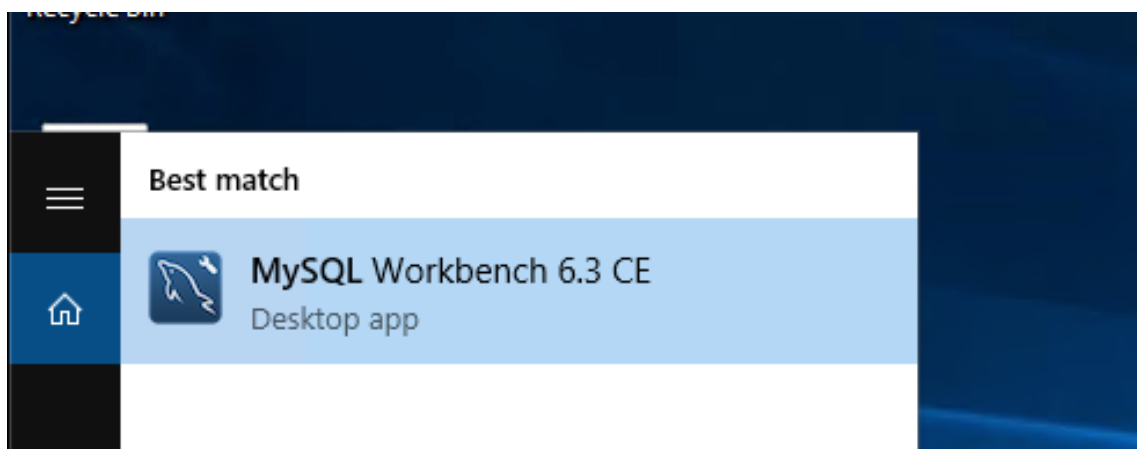
- Post this, I downloaded the Microsoft remote desktop, this is to access the MySQL database.



- The desktop was successfully logged in.



- We will have to access MySQL in the desktop



- A connection has to be setup with the credentials provided

The screenshot shows the 'Setup New Connection' dialog box. It has a title bar with a minus, maximize, and close button. The dialog is divided into sections. At the top, there's a 'Connection Name' field with a placeholder 'Type a name for the connection'. Below it is the 'Connection Method' dropdown, currently set to 'Standard (TCP/IP)' with a placeholder 'Method to use to connect to the RDBMS'. There are three tabs: 'Parameters', 'SSL', and 'Advanced'. The 'Parameters' tab is active. It contains several fields: 'Hostname' with the value 'ja25r8gg.us-west-2.rds.amazonaws.com', 'Port' with the value '3306', 'Username' with the value 'student', 'Password' with a 'Store in Vault ...' button and a 'Clear' button, and 'Default Schema' which is empty. To the right of these fields are descriptive labels: 'Name or IP address of the server host - and TCP/IP port.', 'Name of the user to connect with.', 'The user's password. Will be requested later if it's not set.', and 'The schema to use as default schema. Leave blank to select it later.' At the bottom of the dialog are four buttons: 'Configure Server Management...', 'Test Connection', 'Cancel', and 'OK'.

Setup New Connection

Connection Name: Type a name for the connection

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

Parameters SSL Advanced

Hostname: ja25r8gg.us-west-2.rds.amazonaws.com Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: student 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.

This screenshot shows the same 'Setup New Connection' dialog box as above, but with an additional 'Store Password For Connection' sub-dialog box overlaid on top. The sub-dialog has a title bar and a close button. It contains the text 'Please enter password for the following service:' followed by a list of service details: 'Service: Mysql@my-rds.cb6eba25r8gg.us-west-2.rds.amazonaws.com:3306', 'User: student', and 'Password: *****'. There are 'OK' and 'Cancel' buttons at the bottom of the sub-dialog. The 'Store Password For Connection' sub-dialog also features a DBeaver logo and a lock icon.

Setup New Connection

Connection Name: Type a name for the connection

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

Parameters SSL Advanced

Hostname: ja25r8gg.us-west-2.rds.a Username: student Password: Default Schema:

Store Password For Connection

Please enter password for the following service:

Service: Mysql@my-rds.cb6eba25r8gg.us-west-2.rds.amazonaws.com:3306

User: student

Password: *****

- The MySQL can be queried and utilized like a regular database

The screenshot displays the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar contains a Navigator pane with sections for MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup), and SCHEMAS (Filter objects, innodb, lab, sys). The main workspace is divided into three panes: a Query pane on the left showing a list of queries, a central SQL editor pane with the following SQL code:

```
1 CREATE TABLE lab.staff (firstname text, lastname text, phone text);
2
3 INSERT INTO lab.staff VALUES ("John", "Smith", "555-1234");
4
5 INSERT INTO lab.staff VALUES ("Sarah",
6 "Jones", "555-8866");
7
8 * lab.staff firstname = "Sarah";
```

The right pane shows the Output tab with the following results:

#	Time	Action	Message	Duration / Fetch
1	02:37:18	CREATE TABLE lab.staff (firstname text, lastname text, phone text)	0 row(s) affected	0.015 sec
2	02:37:18	INSERT INTO lab.staff VALUES ("John", "Smith", "555-1234")	1 row(s) affected	0.000 sec
3	02:37:18	INSERT INTO lab.staff VALUES ("Sarah", "Jones", "555-8866")	1 row(s) affected	0.000 sec


The bottom pane shows the Result Grid tab with the following data:

firstname	lastname	phone
Sarah	Jones	555-8866

The bottom right corner of the interface includes buttons for Result Grid and Form Editor.

Lab 2: Introduction to Amazon DynamoDB

- Create a new table in DynamoDB with a primary key and sort key for uniquely identifying each item




Services ▾ Resource Groups ▾

awsstudent @ 5029-0881-8785 ▾ Oregon ▾ Support ▾

web, gaming, ad-tech, IoT, and many other applications.

Create table


[Getting started guide](#)



Create tables

Create DynamoDB tables with a few clicks. Just specify the desired read and write throughput for your table, and DynamoDB handles the rest.


[More about DynamoDB throughput](#)



Add and query items

Once you have created a DynamoDB table, use the AWS SDKs to write, read, modify, and query items in DynamoDB.

[DynamoDB API reference](#)



Monitor and manage tables

Using the AWS Management Console, you can monitor performance and adjust the throughput of your tables, enabling you to scale seamlessly.

[Monitoring tables](#)

Create DynamoDB table

Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name*

Music

Primary key*

Partition key

Artist

String

Add sort key

Song

String

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type.

You do not have the required role to enable Auto Scaling by default.

Please refer to [documentation](#).

Name

Music

No CloudWatch alarms have been triggered for this table.

Stream details

Stream enabled

No

View type

-

Latest stream ARN

-

Manage Stream

Table details

Table name

Music

Primary partition key

Artist (String)

Primary sort key

Song (String)

Point-in-time recovery

DISABLED

Enable

Encryption Type

DEFAULT

Manage Encryption

KMS Master Key ARN

Not Applicable

Encryption Status

CloudWatch Contributor Insights

DISABLED

Manage Contributor Insights

PREVIEW

Time to live attribute

DISABLED

Manage TTL

Table status

Active

- Adding data to the table

Edit item

Tree

Item {2}

Artist String : Pink Floyd

Insert

String

Binary

Number

StringSet

NumberSet

BinarySet

Map

List

Boolean

Null

Remove

g : Money

- Editing to add more fields

Edit item

Tree ▾

↕

↕

▼ Item {3}

⊕ Album String : The Dark side of the Moon

⊕ Artist String : Pink Floyd

⊕ Song String : Money

- Creating multiple items

<input type="checkbox"/>	Artist ⓘ ▲	Song ▼	Album ▼	Year ▼	Genre ▼	LengthSeconds	
<input type="checkbox"/>	John Lennon	Imagine	Imagine	1971	Soft rock		
<input type="checkbox"/>	Pink Floyd	Money	The Dark sid...	1973			
<input type="checkbox"/>	Psy	Gagnam style	Psy 6 (Six Ru...	2011		219	

- Modifying existing item

Edit item

Tree ▾

Item {5}

- Album String : Psy 6 (Six Rules), Part 1
- Artist String : Psy
- LengthSeconds Number : 219
- Song String : Gagnam style
- Year Number : 2012

- Querying the table

Scan: [Table] Music: Artist, Song ^ Viewing 1 to 3 Items

Query ▾ [Table] Music: Artist, Song ▾ ^

Partition key Artist String = Psy

Sort key Song String = Gangnam Style

Filter Enter attribute String = Enter value ✕

+ Add filter

Sort ☒ Ascending ☐ Descending

Attributes ☒ All ☐ Projected

<input type="checkbox"/>	Artist ⓘ	Song	Album	Year	Genre	LengthSeconds
--------------------------	----------	------	-------	------	-------	---------------

Query: [Table] Music: Artist, Song ^ Viewing 1 to 1 items

Query [Table] Music: Artist, Song ^

Partition key Artist String = Psy

Sort key Song String = Gagnam style

+ Add filter

Sort ☒ Ascending ☐ Descending

Attributes ☒ All ☐ Projected

Start search

<input type="checkbox"/>	Artist	Song	Album	Year	LengthSeconds
<input type="checkbox"/>	Psy	Gagnam style	Psy 6 (Six Rules), Part 1	2012	219

- Scanning the table

Scan: [Table] Music: Artist, Song ^ Viewing 1 to 1 items

Scan [Table] Music: Artist, Song ^

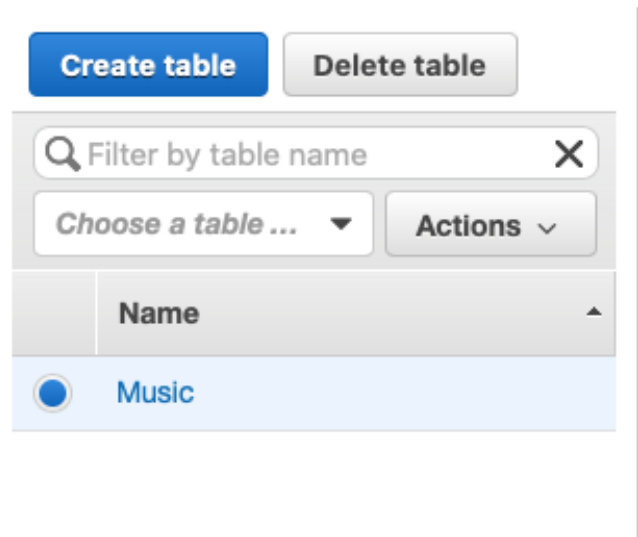
Filter Year Number = 1971 x

+ Add filter

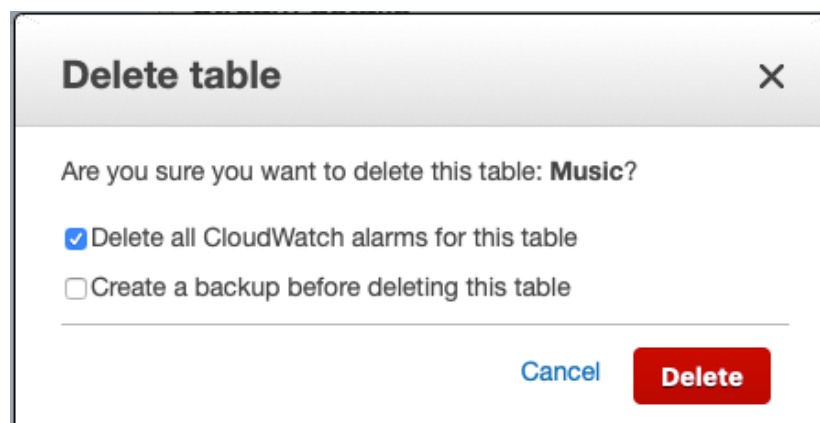
Start search

<input type="checkbox"/>	Artist	Song	Album	Year	Genre
<input type="checkbox"/>	John Lennon	Imagine	Imagine	1971	Soft rock

- Deleting the table



A screenshot of the AWS IAM console showing a table list. At the top, there are two buttons: "Create table" (blue) and "Delete table" (grey). Below these is a search bar labeled "Filter by table name" with a magnifying glass icon and a close button (X). Under the search bar, there is a dropdown menu labeled "Choose a table ..." and an "Actions" dropdown button. Below these elements is a table with a header row containing "Name" and a table body with one row labeled "Music". The "Music" row is highlighted in light blue.



A screenshot of the "Delete table" confirmation dialog. The dialog has a title bar "Delete table" with a close button (X). The main text asks: "Are you sure you want to delete this table: **Music**?". Below this text are two checkboxes: "Delete all CloudWatch alarms for this table" (checked) and "Create a backup before deleting this table" (unchecked). At the bottom right, there are two buttons: "Cancel" (blue) and "Delete" (red).

Lab 3 : Working with Amazon Redshift

- Creating clusters using amazon redshift

The screenshot displays the AWS Redshift console's 'Node type' selection page. The interface includes a top navigation bar with the AWS logo, 'Services', 'Resource Groups', and a user profile 'awsstude'. A left-hand sidebar contains navigation icons for 'QUERIES', 'EDITOR', 'CONFIG', 'MARKETPLACE', 'ADVISOR', 'ALARMS', 'EVENTS', and 'WHAT'S NEW'. The main content area is titled 'Node type' and instructs users to 'Choose a node type that meets your CPU, RAM, storage capacity, and drive type requirements.' It features two primary columns of node types. The first column, labeled 'Recommended' and 'RA3', describes 'High performance with scalable managed storage' and lists the 'ra3.16xlarge' node type with a price of '\$13.04/node/hour' and '\$0.024/GB/month' for managed storage up to 64 TB/node. It includes an icon showing multiple server racks. The second column, labeled 'DC2', describes 'High performance with fixed local SSD storage' and lists two options: 'dc2.large' at '\$0.25/node/hour' with 160 GB/node storage, and 'dc2.8xlarge' at '\$4.80/node/hour' with 2.6 TB/node storage. It includes an icon showing a server rack with an SSD. Below these columns is a link to 'Show legacy dense storage node types'. At the bottom, the 'Nodes' section allows users to 'Enter the number of nodes that you need', with a numeric input set to '2' and a slider, with a range of '(1-32)'. A 'Configuration summary' section is partially visible at the bottom.

aws

Services Resource Groups

awsstude

Node type

Choose a node type that meets your CPU, RAM, storage capacity, and drive type requirements.

Recommended

RA3

High performance with scalable managed storage

☐ ra3.16xlarge \$13.04/node/hour
Managed storage: \$0.024/GB/month
up to 64 TB/node

DC2

High performance with fixed local SSD storage

☒ dc2.large \$0.25/node/hour
Storage: 160 GB/node

☐ dc2.8xlarge \$4.80/node/hour
Storage: 2.6 TB/node

ra3.16xlarge
48 vCPU (gen 3)

dc2.large
2 vCPU (gen 2)

► Show legacy dense storage node types

Nodes

Enter the number of nodes that you need.

2

Range (1-32)

Configuration summary

Cluster details

Cluster identifier

This is the unique key that identifies a cluster.

The identifier must be from 1-63 characters. Valid characters are a-z (lowercase only) and - (hyphen).

Database port (optional)

Port number of the port where the database accepts inbound connections.

The port must be numeric (1150-65535).

Master user name

Enter a login ID for the master user of your DB instance.

The name must be 1-128 alphanumeric characters, and it can't be a **reserved word**.

Master user password

☐ Show password

▼ Network and security

Virtual private cloud (VPC)

This VPC defines the virtual networking environment for this cluster.

vpc-04d4845d28016ddef



You can't change the VPC associated with this cluster after the cluster has been created. [Learn more](#)

VPC security groups

This VPC security groups define which subnets and IP ranges the cluster can use in the VPC.

sg-09eb8761a0ffe4e0d

Cluster subnet group

Choose the Amazon Redshift subnet group to launch the cluster in.

Availability Zone

aws Services Resource Groups

Node type

Choose a node type that meets your CPU, RAM, storage capacity, and drive type requirements.

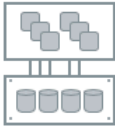
Recommended

RA3

High performance with scalable managed storage

☐ ra3.16xlarge \$13.04/node/hour
\$0.024/GB/month

Managed storage:
up to 64 TB/node


 ra3.16xlarge
48 vCPU (gen 3)

DC2

High performance with fixed local SSD storage

☒ dc2.large \$0.25/node/hour
Storage: 160 GB/node

☐ dc2.8xlarge \$4.80/node/hour
Storage: 2.6 TB/node

 dc2.large
2 vCPU (gen 2)

[Show legacy dense storage node types](#)

Nodes

Enter the number of nodes that you need.

2

Range (1-32)

Configuration summary

- Cluster has been created with the above configuration

Clusters (1)

All status

☐ **Cluster** ☐ Status ☐ Storage capacity us... ☐ CPU utilization ☐ Snapshots ☐ Notificati... ☐ Tags

☐ lab
dc2.large | 2 nodes | 320 GB Available 24% -

- Connecting to the cluster using pgweb IP which was provided

pgweb

Scheme

Standard

SSH

Host	<input type="text"/>
Username	<input type="text" value="master"/>
Password	<input type="password" value="....."/>
Database	<input type="text" value="lab"/>
Port	<input type="text" value="5432"/>
SSL	<div>require</div>

Connect

- The table flight was created post which the number of rows in the table is being fetched

lab3

Rows

Structure

Indexes

Constraints

SQL Query

History

Activity

Connection

public

Tables (1)

flights

1 SELECT COUNT(*) FROM flights;

Run Query

Explain Query

count

96825753

- The data has been copied to the new table which was created. The below query will fetch the data from the table

lab3

Rows

Structure

Indexes

Constraints

SQL Query

History

Activity

Connection

Connect

Disconnect

public

Tables (1)

flights

1 SELECT *

2 FROM flights

3 ORDER BY random()

4 LIMIT 10;

Run Query

Explain Query

JSON

CSV

XML

Table Information

year	month	day	carrier	origin	dest	aircraft_code	miles	departures	minutes	seats	passengers	freight_pounds
2003	11	7	Chautauqua Airlines Inc.	RDU	STL	676	667	3	341	132	99	0
1992	1	10	United Air Lines Inc.	BTV	IAD	619	442	1	75	128	50	124
2003	2	16	Continental Air Lines Inc.	MCI	EWB	616	1092	1	131	104	37	23
2003	3	26	Hageland Aviation Service	TLT	BET	35	36	1	20	6	3	0
1991	7	10	Southwest Airlines Co.	BNA	PHX	619	1448	2	381	274	193	580
1995	1	29	Simmons Airlines	SHV	DFW	442	190	2	105	128	52	10
2011	11	28	Delta Air Lines Inc.	JFK	SFO	622	2586	5	1789	869	796	3157
2006	12	14	Air Wisconsin Airlines Corp	HSV	CLT	629	333	1	59	50	34	0
1995	9	1	Delta Air Lines Inc.	MCI	ATL	715	693	2	189	298	213	1334
2008	7	5	Mesaba Airlines	PIR	MSP	456	350	1	76	34	19	0

- In the below query we are grouping the result based on the carrier. The top three carriers based on number of departures are Southwest Airlines Co., Delta Airlines Co., and American Airlines Inc.

```
1 SELECT
2   carrier,
3   SUM (departures)
4 FROM flights
5 GROUP BY carrier
6 ORDER BY 2 DESC
7 LIMIT 10;
```

Run Query

Explain Query

carrier	sum
Southwest Airlines Co.	19846786
Delta Air Lines Inc.	17480331
American Airlines Inc.	15468460
United Air Lines Inc.	13402737
Northwest Airlines Inc.	9522027
Continental Air Lines Inc.	7929064
US Airways Inc.	7865177
American Eagle Airlines Inc.	6687299
ExpressJet Airlines Inc.	6270410
USAir	6052546

- The top three carriers based on number of passengers are Delta Airlines Co., Southwest Airlines Co., and American Airlines Inc.

[Run Query](#) [Explain Query](#)

carrier	sum
Delta Air Lines Inc.	1894041955
Southwest Airlines Co.	1812013508
American Airlines Inc.	1599088928
United Air Lines Inc.	1396240286
Northwest Airlines Inc.	846994753
Continental Air Lines Inc.	751752616
US Airways Inc.	738513678
USAir	406251028
America West Airlines Inc.	327787337
Alaska Airlines Inc.	290615980

- The top three carriers based on miles flown are United airlines Inc., Southwest Airlines Co., and American Airlines Inc.

Rows
Structure
Indexes
Constraints
SQL Query
History
Activity
Connection

```

1 SELECT
2     carrier,
3     SUM(miles)
4 FROM flights
5 GROUP BY carrier
6 ORDER BY 2 DESC
7 LIMIT 10;

```

Run Query
Explain Query

carrier	sum
United Air Lines Inc.	7478582456
Delta Air Lines Inc.	7356255471
American Airlines Inc.	6033117399
Southwest Airlines Co.	5326561689
Continental Air Lines Inc.	4454443574
Northwest Airlines Inc.	4277614432
US Airways Inc.	3193773517
USAir	2178906790
America West Airlines Inc.	1713042483
Federal Express Corporation	1514604035

- The top three carriers based on passenger-miles flown are Southwest Airlines Co., Delta Airlines Co., and American Airlines Inc.

```

1 SELECT
2   carrier,
3   SUM (passengers * miles)
4 FROM flights
5 GROUP BY carrier
6 ORDER BY 2 DESC
7 LIMIT 10;

```

Run Query Explain Query

carrier	sum
American Airlines Inc.	1733245030322
Delta Air Lines Inc.	1618918706400
United Air Lines Inc.	1537952630588
Southwest Airlines Co.	1045280304865
Continental Air Lines Inc.	805455729207
Northwest Airlines Inc.	742067994834
US Airways Inc.	585542298085
America West Airlines Inc.	293772304506
Alaska Airlines Inc.	266206170484
USAir	240098568130

- The top three carriers based on freight pounds are Federal Express Corporation, United Parcel Service and Delta Airlines Inc.

```

1 SELECT
2   carrier,
3   SUM (freight_pounds)
4 FROM flights
5 GROUP BY carrier
6 ORDER BY 2 DESC
7 LIMIT 10;

```

Run Query Explain Query

carrier	sum
Federal Express Corporation	115908103479
United Parcel Service	70144943595
Delta Air Lines Inc.	12727490652
United Air Lines Inc.	11310081837
Korean Air Lines Co. Ltd.	11237313800
American Airlines Inc.	10391736851
China Airlines Ltd.	9654006204
Eva Airways Corporation	6378076357
Northwest Airlines Inc.	6310003281
Southwest Airlines Co.	5494931618

- Creating a new table aircraft for joining

public

Tables (2)

aircraft

flights

```
1 CREATE TABLE aircraft (  
2   aircraft_code CHAR(3) SORTKEY,  
3   aircraft      VARCHAR(100)  
4 );
```

- Copying data from the bucket to the aircraft table

public

Tables (2)

aircraft

flights

```
1 COPY aircraft  
2 FROM 's3://us-west-2-aws-training/awsu-spl/spl-17/4.2.5.prod/data/lookup_aircraft.csv'  
3 IAM_ROLE 'arn:aws:iam::723548672068:role/Redshift-Role'  
4 IGNOREHEADER 1  
5 DELIMITER ','  
6 REMOVEQUOTES  
7 TRUNCATECOLUMNS  
8 REGION 'us-west-2';
```

- Selecting the data from the newly created table aircraft

```

1 SELECT *
2 FROM aircraft
3 ORDER BY random()
4 LIMIT 10;

```

Run Query Explain Query

aircraft_code	aircraft
314	Bell B-47J2
039	Cessna 182 Skylane
467	Swearingen Metro III
085	Stinson SR-9
675	Embraer-145
050	Howard DGA-15P
175	Lockheed L-12A/L-10/10A
406	Beech 200 Super Kingair
040	De Havilland DHC2 Beaver
691	Airbus Industrie A300-600/RVCF/RCF

- Joining the tables aircraft and flight ordering by trips

```

1 SELECT
2   aircraft,
3   SUM(departures) AS trips
4 FROM flights
5 JOIN aircraft using (aircraft_code)
6 GROUP BY aircraft
7 ORDER BY trips DESC
8 LIMIT 10;

```

Run Query

Explain Query

JSON

aircraft	trips
Boeing 737-300	19632153
McDonnell Douglas DC9 Super 80/MD81/82/83/88	18608868
Canadair RJ-200ER /RJ-440	12062704
Boeing 757-200	10768257
Boeing 727-200/231A	9188041
Embraer-145	9184729
Boeing 737-100/200	8567467
Boeing 737-700/700LR	7550737
McDonnell Douglas DC-9-30	7105295
Airbus Industrie A320-100/200	6664803

- Analyzing the performance of the queries by fetching the query execution plan.

RowsStructureIndexesConstraintsSQL QueryHistoryActivityConnection

```
1 SET enable_result_cache_for_session TO OFF;
2
3 EXPLAIN
4 SELECT
5     aircraft,
6     SUM(departures) AS trips
7 FROM flights
8 JOIN aircraft using (aircraft_code)
9 GROUP BY aircraft
10 ORDER BY trips DESC
11 LIMIT 10;
```

Run QueryExplain Query

QUERY PLAN

XN Limit (cost=1000004618456.86..1000004618456.88 rows=10 width=29)

-> XN Merge (cost=1000004618456.86..1000004618457.82 rows=383 width=29)

Merge Key: sum(flights.departures)

-> XN Network (cost=1000004618456.86..1000004618457.82 rows=383 width=29)

Send to leader

-> XN Sort (cost=1000004618456.86..1000004618457.82 rows=383 width=29)

Sort Key: sum(flights.departures)

-> XN HashAggregate (cost=4618439.47..4618440.42 rows=383 width=29)

-> XN Hash Join DS_DIST_ALL_NONE (cost=4.79..4127906.89 rows=98106516 width=29)

Hash Cond: ("outer".aircraft_code = "inner".aircraft_code)

-> XN Seq Scan on flights (cost=0.00..968257.52 rows=96825752 width=11)

-> XN Hash (cost=3.83..3.83 rows=383 width=32)

- Analyzing the data on the table flight

```
1 ANALYZE COMPRESSION flights;
```

Run Query

Explain Query

Table	Column	Encoding	Est_reduction_pct
flights	year	zstd	90.86
flights	month	zstd	88.54
flights	day	zstd	69.61
flights	carrier	zstd	49.04
flights	origin	zstd	46.21
flights	dest	zstd	40.47
flights	aircraft_code	zstd	43.32
flights	miles	zstd	91.08
flights	departures	zstd	33.30
flights	minutes	zstd	65.37
flights	seats	zstd	68.08
flights	passengers	az64	0.00

- Creating tables from other table

public

Tables (3)

aircraft

airports

flights

```
1 CREATE TABLE airports (  
2   airport_code CHAR(3) SORTKEY,  
3   airport      varchar(100)  
4 );
```

- Copying the data from bucket to the airports table

The screenshot shows the AWS Redshift console interface. On the left, a sidebar displays the database schema under the 'public' schema, listing tables: 'aircraft', 'airports', and 'flights'. The main area contains a SQL query editor with the following text:

```
1 COPY airports
2 FROM 's3://us-west-2-aws-training/awsu-spl/spl-17/4.2.5.prod/data/lookup_airports.csv'
3 IAM_ROLE 'arn:aws:iam::723548672068:role/Redshift-Role'
4 IGNOREHEADER 1
5 DELIMITER ','
6 REMOVEQUOTES
7 TRUNCATECOLUMNS
8 REGION 'us-west-2';
```

Below the query editor are two buttons: 'Run Query' (highlighted in green) and 'Explain Query'. At the bottom, a status message reads 'No records found'.

- Creating a new table `vegas_flights` to load the data

The screenshot shows the AWS Redshift console interface. On the left, a sidebar displays the database schema under the 'public' schema, listing tables: 'aircraft', 'airports', 'flights', and 'vegas_flights'. The main area contains a SQL query editor with the following text:

```
1 CREATE TABLE vegas_flights
2     DISTKEY (origin)
3     SORTKEY (origin)
4 AS
5 SELECT
6     flights.*,
7     airport
8 FROM flights
9 JOIN airports ON origin = airport_code
10 WHERE dest = 'LAS';
```

- Selecting the data loaded from the airport table

```

1 SELECT
2   airport,
3   to_char(SUM(passengers), '999,999,999') as passengers
4 FROM vegas_flights
5 GROUP BY airport
6 ORDER BY SUM(passengers) desc
7 LIMIT 10;

```

Run Query

Explain Query

JSON

airport	passengers
Los Angeles, CA: Los Angeles International	29,403,292
Phoenix, AZ: Phoenix Sky Harbor International	24,160,227
Dallas/Fort Worth, TX: Dallas/Fort Worth International	15,377,974
Denver, CO: Denver International	14,937,489
Chicago, IL: Chicago O'Hare International	14,494,577
San Francisco, CA: San Francisco International	14,241,188
San Diego, CA: San Diego International	11,744,708
Salt Lake City, UT: Salt Lake City International	10,985,774
Atlanta, GA: Hartsfield-Jackson Atlanta International	10,279,586
Reno, NV: Reno/Tahoe International	10,166,655

- Verifying the disk space and Data distribution using the below query

```

1 SELECT
2   owner AS node,
3   diskno,
4   used,
5   capacity,
6   used/capacity::numeric * 100 as percent_used
7 FROM stv_partitions
8 WHERE host = node
9 ORDER BY 1, 2;

```

Run Query

Explain Query

node	diskno	used	capacity	percent_used
0	0	1039	190633	0.5450263070926859400
1	0	808	190633	0.4238510646110589400

```

1 SELECT
2     name,
3     count(*)
4 FROM stv_blocklist
5 JOIN (SELECT DISTINCT name, id as tbl from stv_tbl_perm) USING (tbl)
6 GROUP BY name;

```

Run Query

Explain Query

JSON

CSV

name	count
aircraft	20
airports	20
flights	1605
redshift_auto_health_check_1052096	16
vegas_flights	149

- Exploring the query performance using redshift console

Query runtime

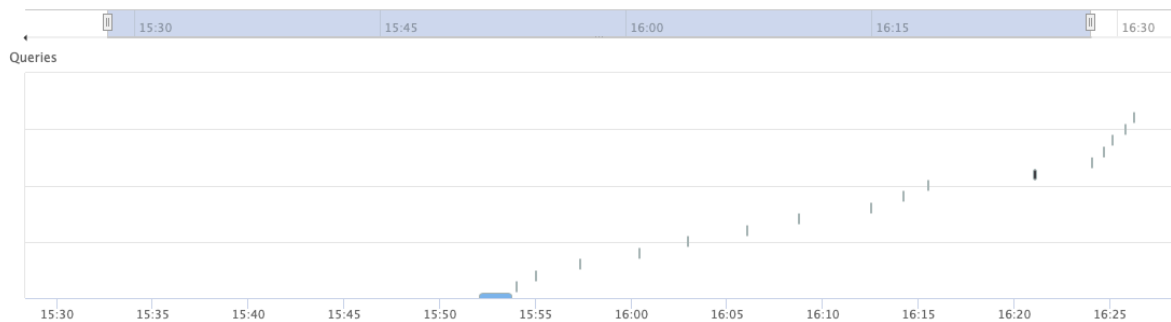
Total queries

Workload execution breakdown

Workload concurrency

Query ID	Duration	User	SQL	Cluster	Status
370,371,372...	10 s	master	ANALYZE COMPRESSION flights;	lab3	Completed

Query runtime



Cluster status

Available Unavailable



Queries and loads (17)

[Terminate query](#)

All queries ▼

Filter queries

< 1 > ⚙

<input type="checkbox"/>	Start time ▲	Query ▼	Status ▼	Duration ▼	SQL ▼	Copy SQL	PID ▼	Transaction ID ▼
<input type="checkbox"/>	Mar 17th, 2020 03:52:02 PM 36 minutes ago	12,37,42	Completed	2 m	COPY flights FROM 's3://us-west-2 ...	Copy	8710	1122
<input type="checkbox"/>	Mar 17th, 2020 03:54:01 PM 34 minutes ago	46	Completed	226 ms	SELECT COUNT(*) FROM flights;	Copy	8710	1199
<input type="checkbox"/>	Mar 17th, 2020 03:54:57 PM 33 minutes ago	54	Completed	5 s	SELECT * FROM flights ORDER BY r ...	Copy	8710	1215
<input type="checkbox"/>	Mar 17th, 2020 03:57:19 PM 31 minutes ago	108	Completed	1 s	SELECT carrier, SUM (departures) ...	Copy	8710	1342
<input type="checkbox"/>	Mar 17th, 2020 04:00:23 PM 28 minutes ago	134	Completed	1 s	SELECT carrier, SUM (passengers) ...	Copy	8710	1400
<input type="checkbox"/>	Mar 17th, 2020 04:02:54 PM 25 minutes ago	166	Completed	1 s	SELECT carrier, SUM (miles) FROM ...	Copy	8710	1470
<input type="checkbox"/>	Mar 17th, 2020 04:06:00 PM 22 minutes ago	198	Completed	2 s	SELECT carrier, SUM (passengers * ...	Copy	8710	1546
<input type="checkbox"/>	Mar 17th, 2020 04:08:44 PM 20 minutes ago	226	Completed	1 s	SELECT carrier, SUM (freight_pou ...	Copy	8710	1604

- Verifying the query execution details

☐ Mar 17th, 2020 03:53:40 PM
36 minutes ago

42

Completed

2 s

Main

Rewritten query

Query details

Query ID	Cluster	User	Type	Status
12	lab3	master	Rewritten query	Completed
From March 17, 2020 at 03:52:02 PM To March 17, 2020 at 03:53:29 PM				Total runtime
				1m

[Query details](#)[Query plan](#)

SQL

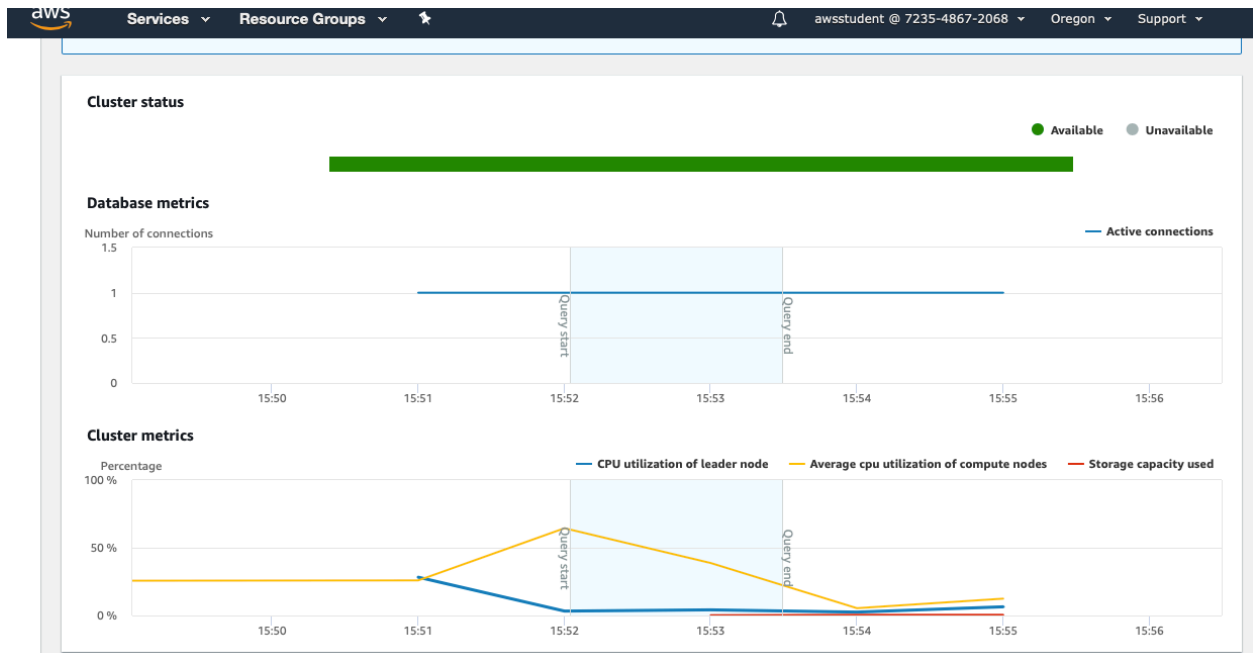
Copy

```
COPY flights
FROM
's3://us-west-2-aws-training/awsu-spl/spl-17/4.2.5.prod/data/flights-usa' IAM_ROLE '' GZIP
DELIMITER ',' REMOVEQUOTES REGION 'us-west-2';
```

Execution details

Total runtime	1m
Statistics	
Rows returned	484,128,783
Total data scanned	21.39 GB

- Verifying the database metrics using cluster console



Maintenance
Edit

Current cluster version 1.0.13448 🔗	Release status Up to date	Allow version upgrade Yes	Maintenance track Current
Maintenance window Every Saturday from 08:00 PM to 08:30 PM		Defer maintenance window You can defer maintenance on your cluster for a specified window. Disabled	
Next scheduled maintenance in 4 days			

Audit logging
Edit

Audit logging logs information about connections and user activities in your database to monitor for security and troubleshooting purposes. The logs are stored in Amazon S3 buckets for convenient access.

Audit logging
Disabled

- Verifying the database metrics using cluster console

🟢 Available			dc2.large	2 total nodes
Cluster performance Query monitoring Maintenance and monitoring Backup Properties Schedule				
Backup details				<button>Edit</button>
Automated snapshot schedule -		Cross-region snapshots Disabled		0 manual snapshots 1 automated snapshots
Snapshot retention period Automated (30 days) Manual (Indefinitely)				

- Deleting the cluster after use.

Amazon Redshift > Clusters								
Clusters (1)					<button>Refresh</button>	<button>Query cluster</button>	<button>Actions</button>	<button>Create cluster</button>
All status <input type="text" value="Search"/>								
<input checked="" type="checkbox"/>	Cluster	▲	Status ▼	Storage capacity us...	CPU utilization			
<input checked="" type="checkbox"/>	lab3 dc2.large 2 nodes 320 GB		🟢 Available	< 1%		...	Tags ▼	
					Resize	Modify	Reboot	Pause
					Delete	Restore table		