McMaster University

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

Assignment 2: Qwiklabs- 1,2 and 3

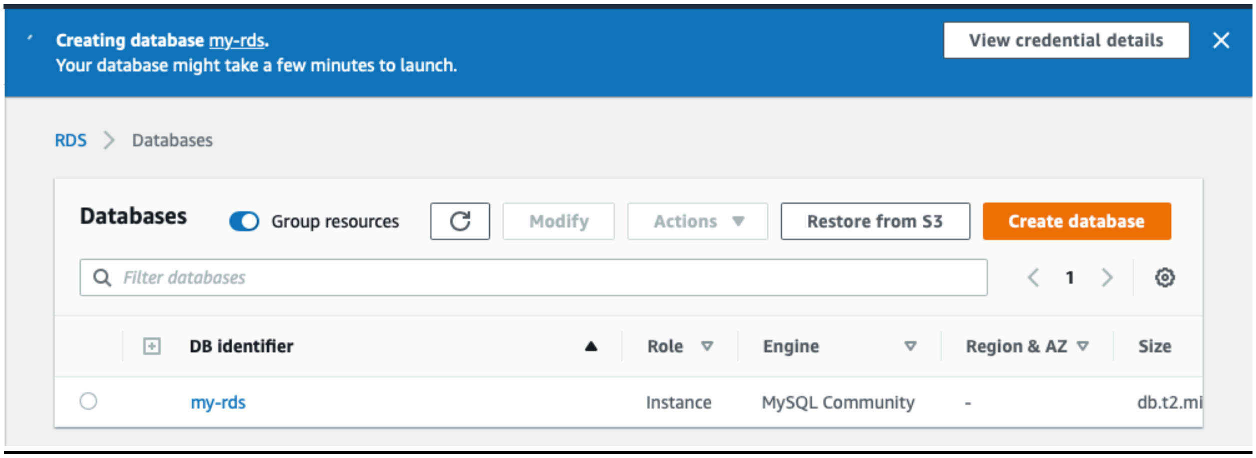
Submitted by,

Greeshma Gopal(gopalg)

ID- 400245291

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

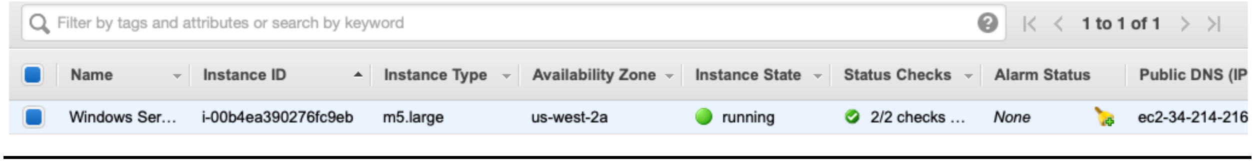
- I have used qwiklab credits to get the login credentials. The database was created with the configuration provided



Creating database my-rds.
Your database might take a few minutes to launch.

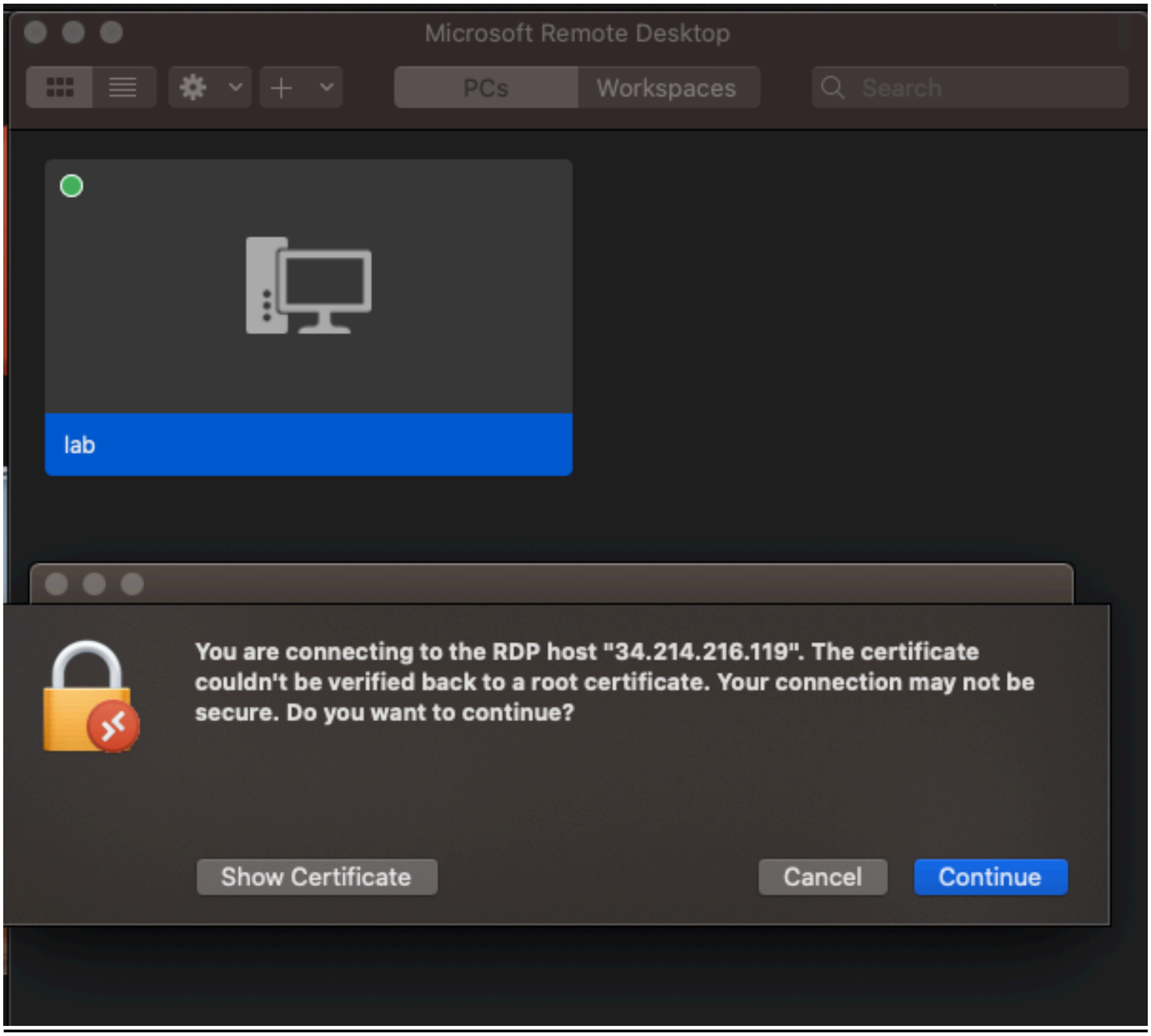
	DB identifier	Role	Engine	Region & AZ	Size
○	my-rds	Instance	MySQL Community	-	db.t2.mi

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



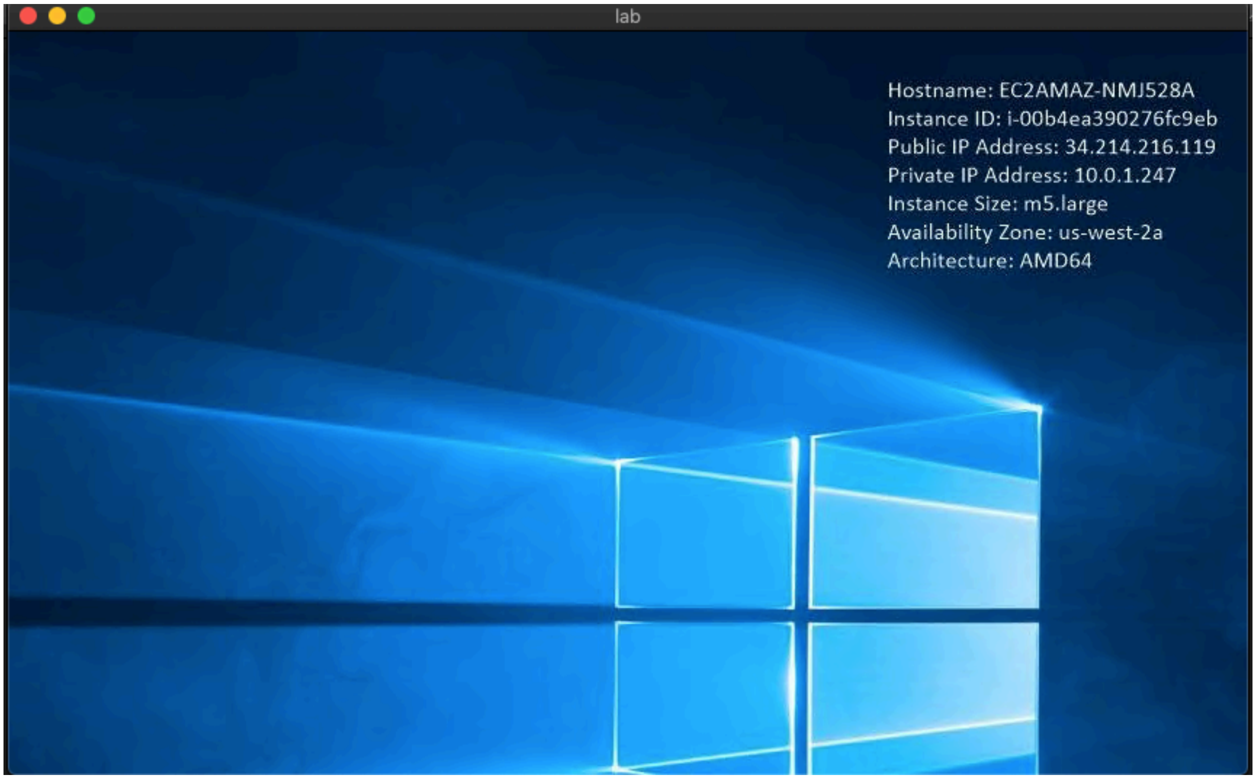
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IP
■	Windows Ser…	i-00b4ea390276fc9eb	m5.large	us-west-2a	● running	✓ 2/2 checks …	*None*	ec2-34-214-216

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



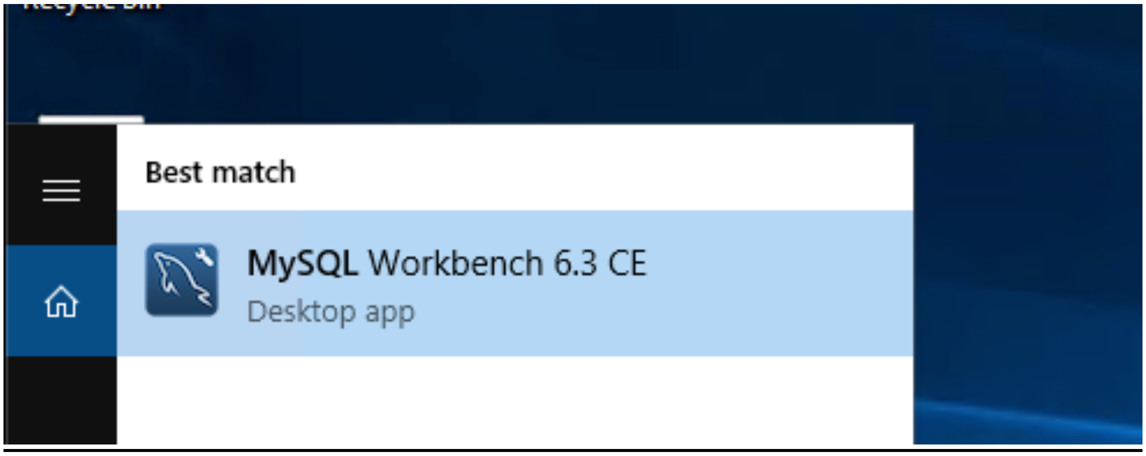
You are connecting to the RDP host "34.214.216.119". The certificate couldn't be verified back to a root certificate. Your connection may not be secure. Do you want to continue?

- The desktop was successfully logged in.

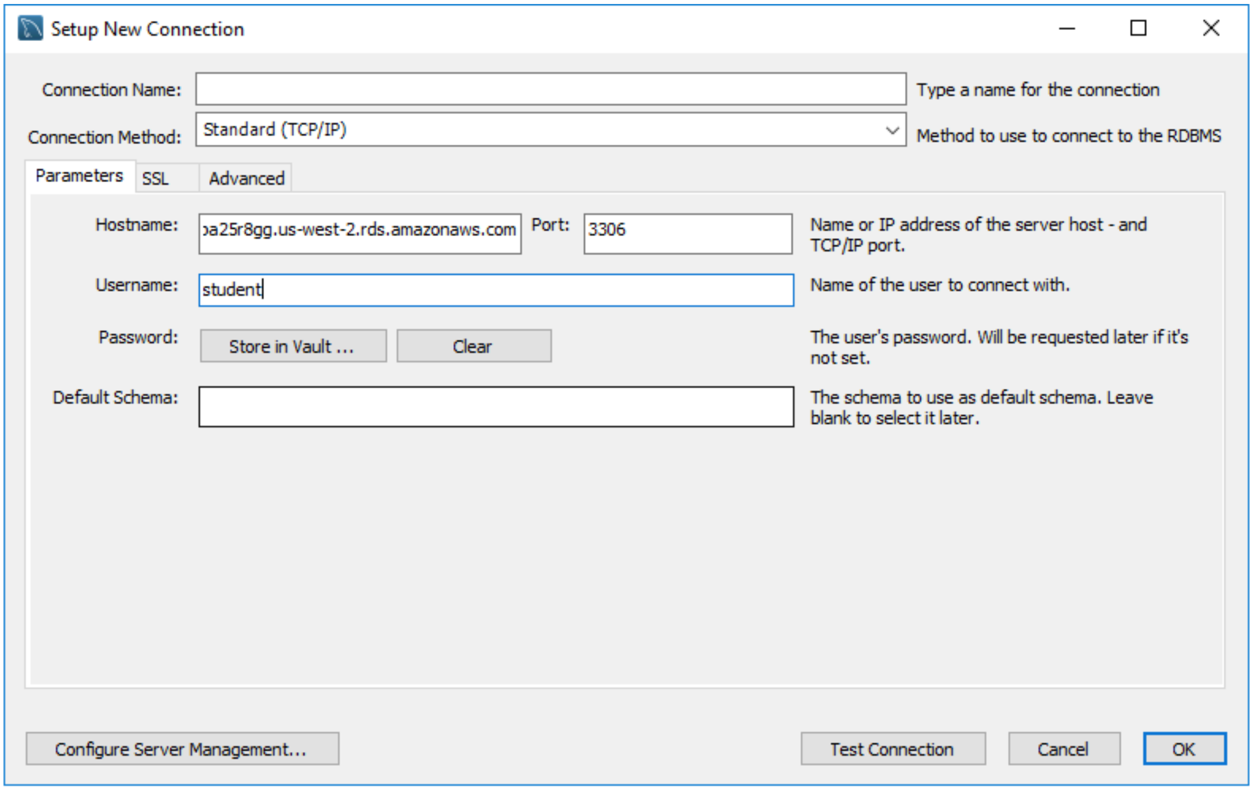


Hostname: EC2AMAZ-NMJ528A
Instance ID: i-00b4ea390276fc9eb
Public IP Address: 34.214.216.119
Private IP Address: 10.0.1.247
Instance Size: m5.large
Availability Zone: us-west-2a
Architecture: AMD64

- We will have to access MySQL in the desktop



- A connection has to be setup with the credentials provided



Setup New Connection

Connection Name: Type a name for the connection

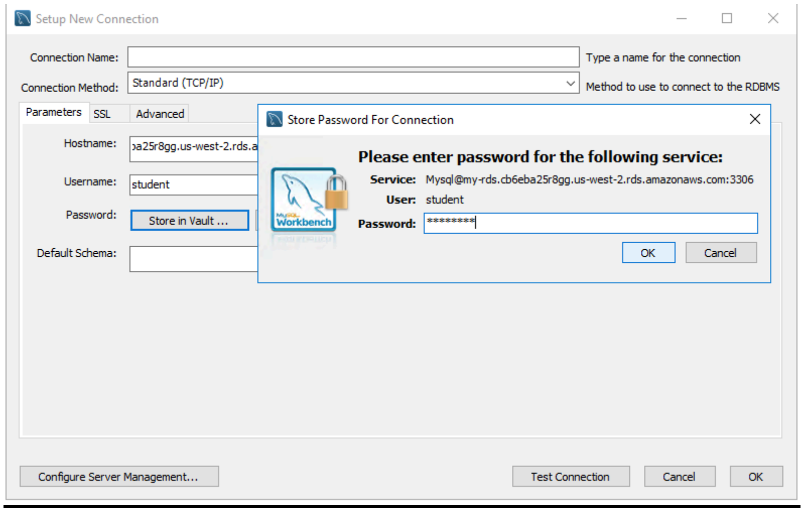
Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Hostname: ɔa25r8gg.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.



Setup New Connection

Connection Name: Type a name for the connection

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

Hostname: ɔa25r8gg.us-west-2.rds.a

Username: student

Password:

Default Schema:

Store Password For Connection

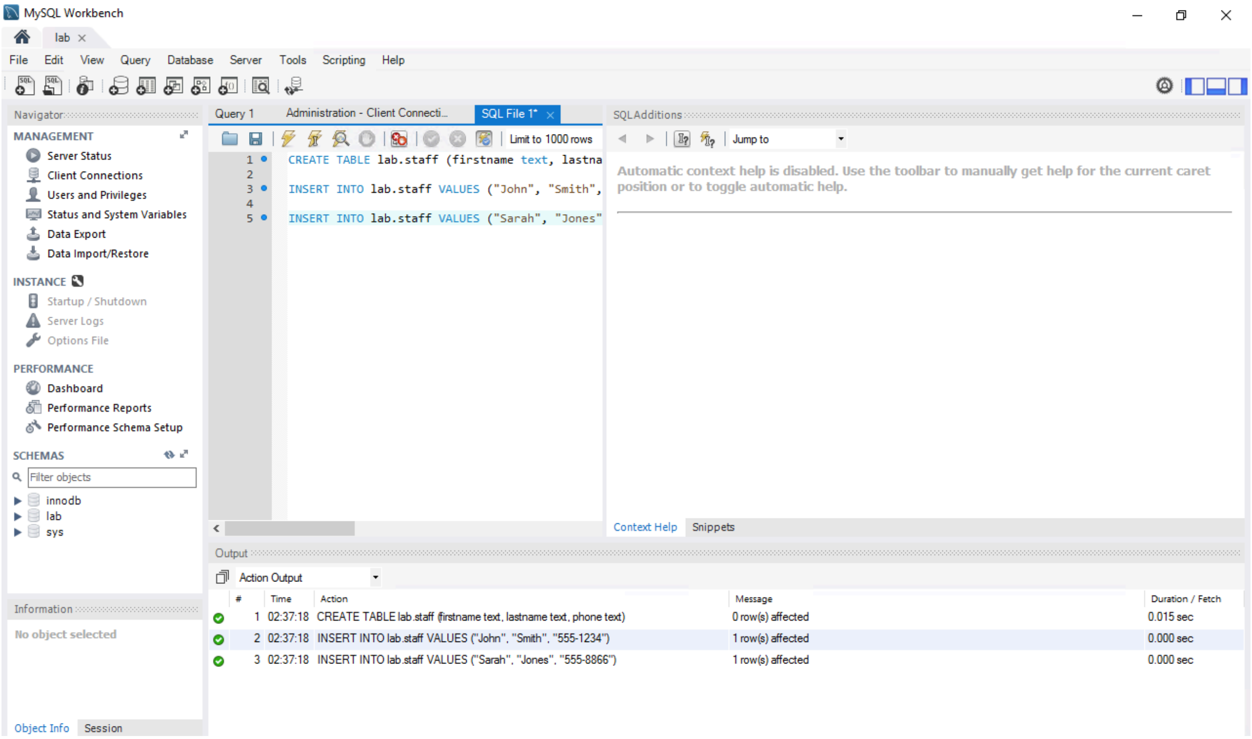
Please enter password for the following service:

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

User: student

Password: ********

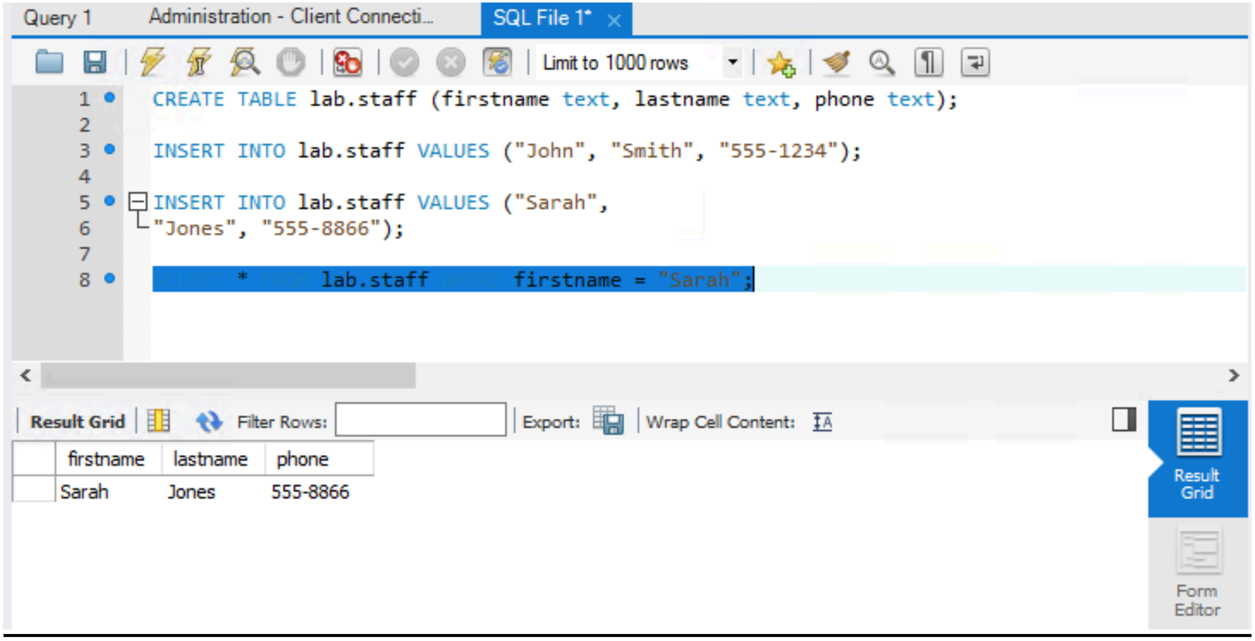
- The MySQL can be queried and utilized like a regular database



```
CREATE TABLE lab.staff (firstname text, lastna

INSERT INTO lab.staff VALUES ("John", "Smith",

INSERT INTO lab.staff VALUES ("Sarah", "Jones"
```

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  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
```

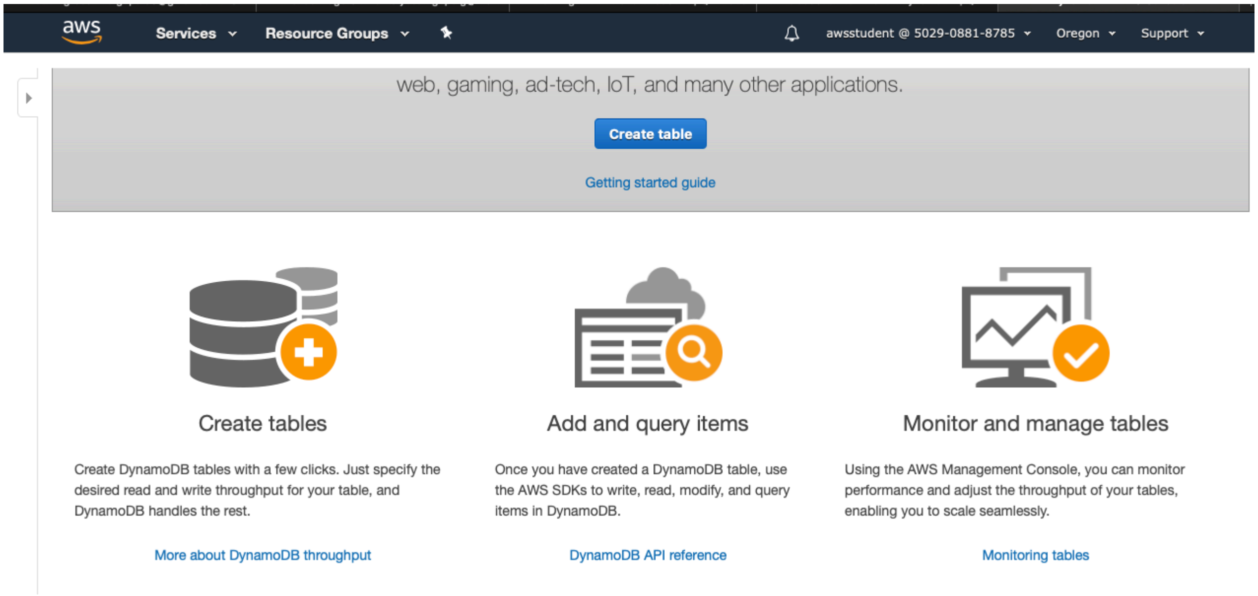


```
CREATE TABLE lab.staff (firstname text, lastname text, phone text);

INSERT INTO lab.staff VALUES ("John", "Smith", "555-1234");

INSERT INTO lab.staff VALUES ("Sarah",
"Jones", "555-8866");

SELECT * FROM lab.staff WHERE firstname = "Sarah";
```

firstname	lastname	phone
Sarah	Jones	555-8866

Lab 2: Introduction to Amazon DynamoDB

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



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

Create tables

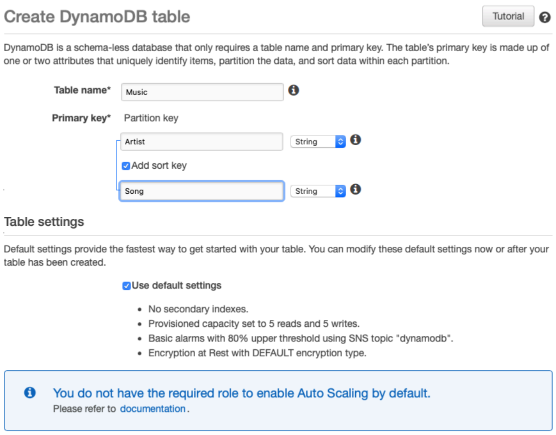
Create DynamoDB tables with a few clicks. Just specify the desired read and write throughput for your table, and DynamoDB handles the rest.

Add and query items

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

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.



Create DynamoDB table

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

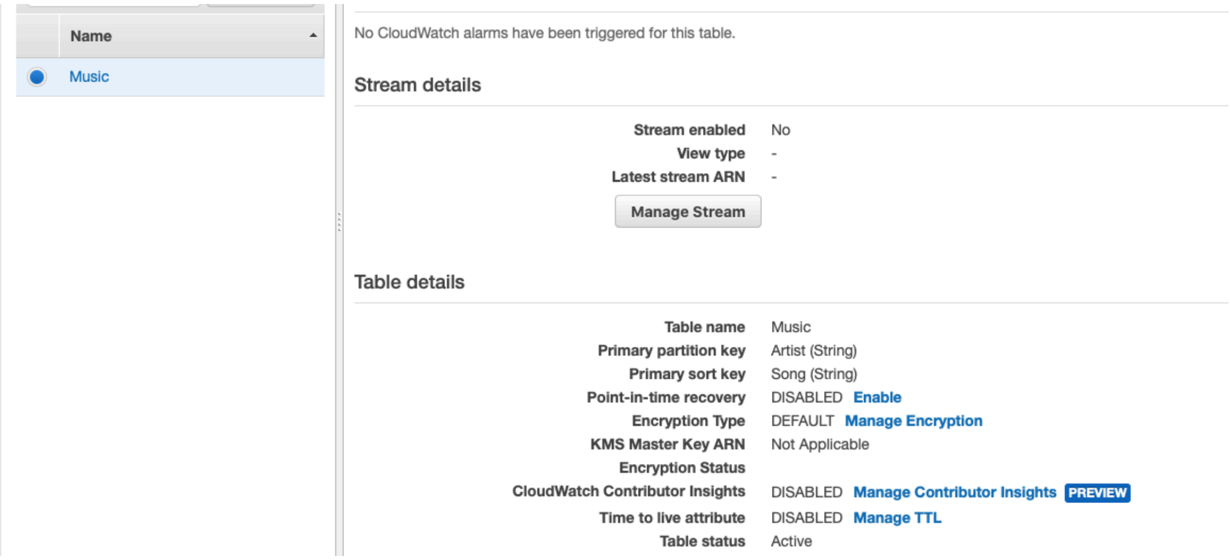
Table settings

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.



No CloudWatch alarms have been triggered for this table.

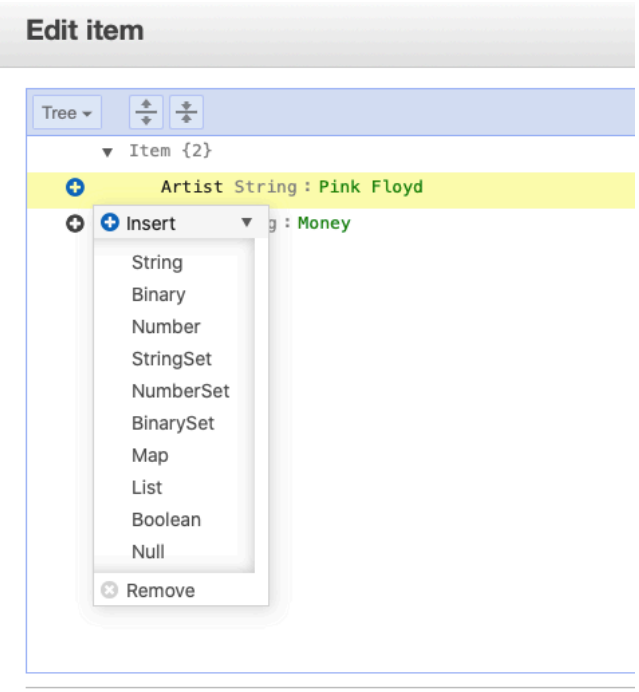
Stream details

Stream enabled	No
View type	-
Latest stream ARN	-

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

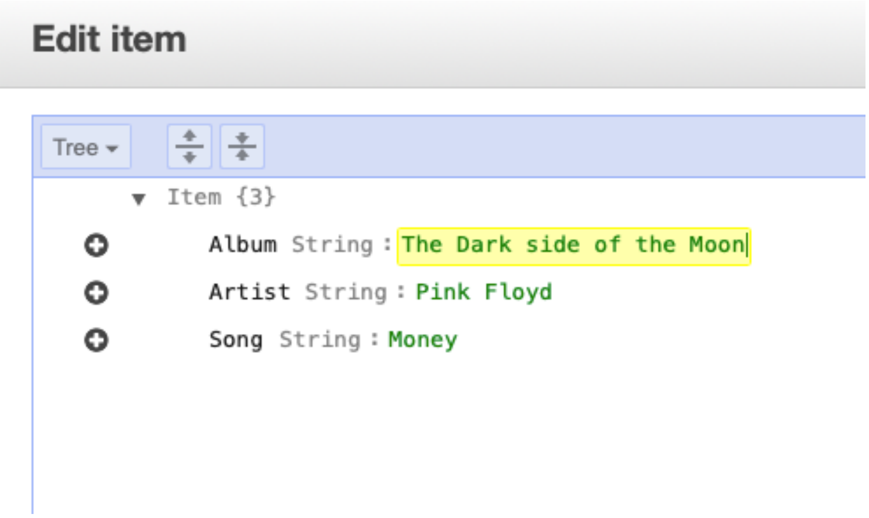
- Adding data to the table



Edit item

```
▼ Item {2}
    Artist String : Pink Floyd
              g : Money
```

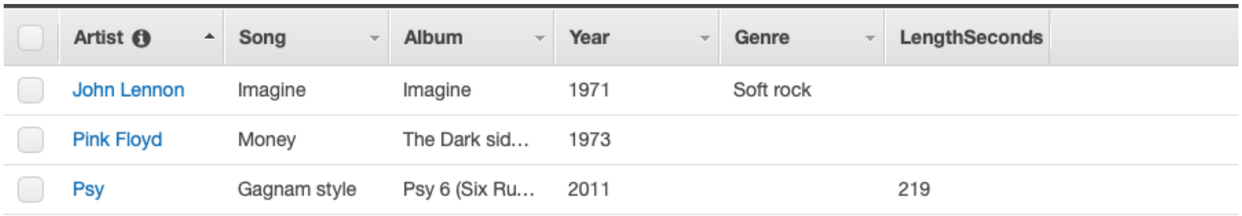
- Editing to add more fields



Edit item

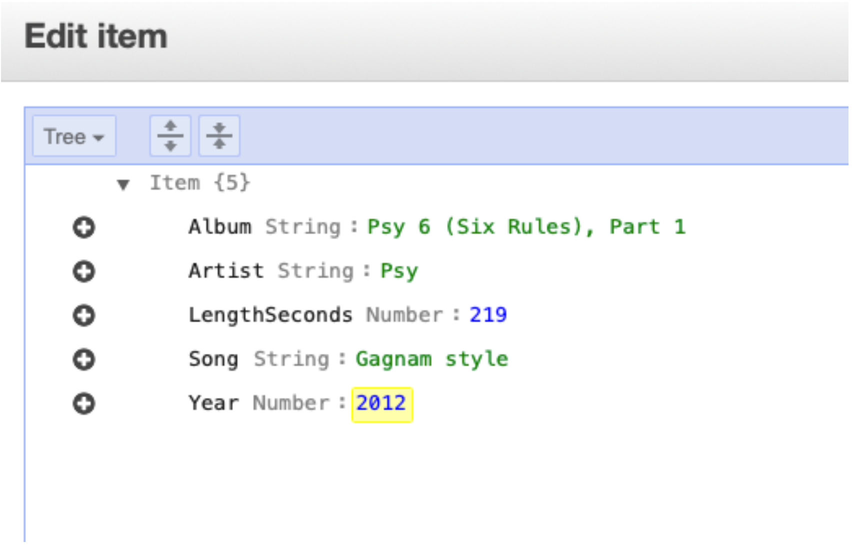
```
▼ Item {3}
    Album String : The Dark side of the Moon
    Artist String : Pink Floyd
    Song String : Money
```

- Creating multiple items



□	Artist	Song	Album	Year	Genre	LengthSeconds
□	John Lennon	Imagine	Imagine	1971	Soft rock	
□	Pink Floyd	Money	The Dark sid...	1973		
□	Psy	Gagnam style	Psy 6 (Six Ru...	2011		219

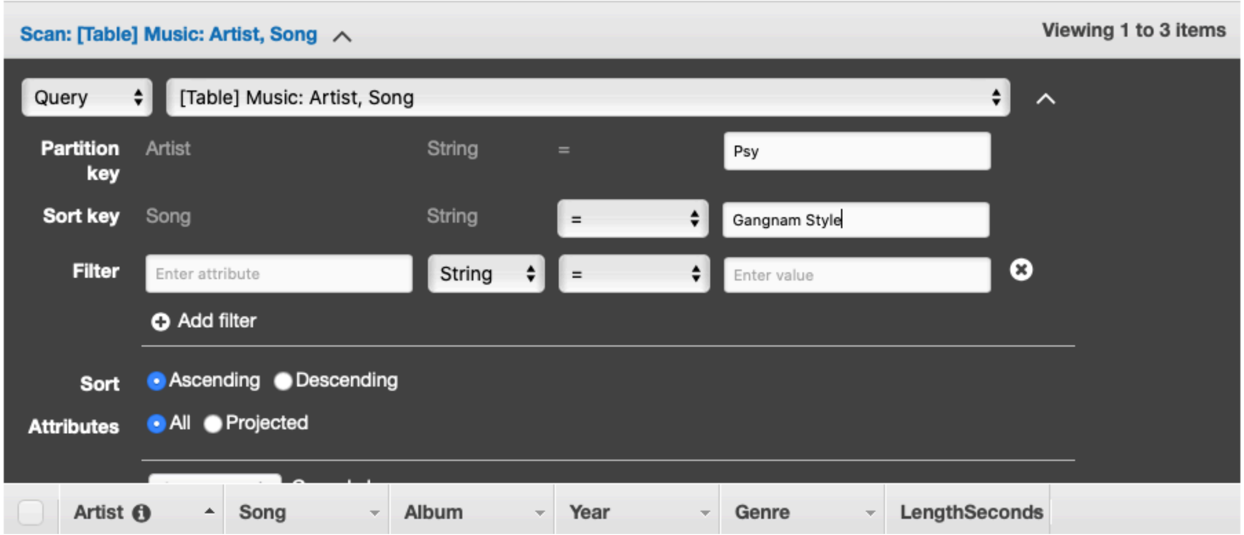
- Modifying existing item



Edit item

```
▼ 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

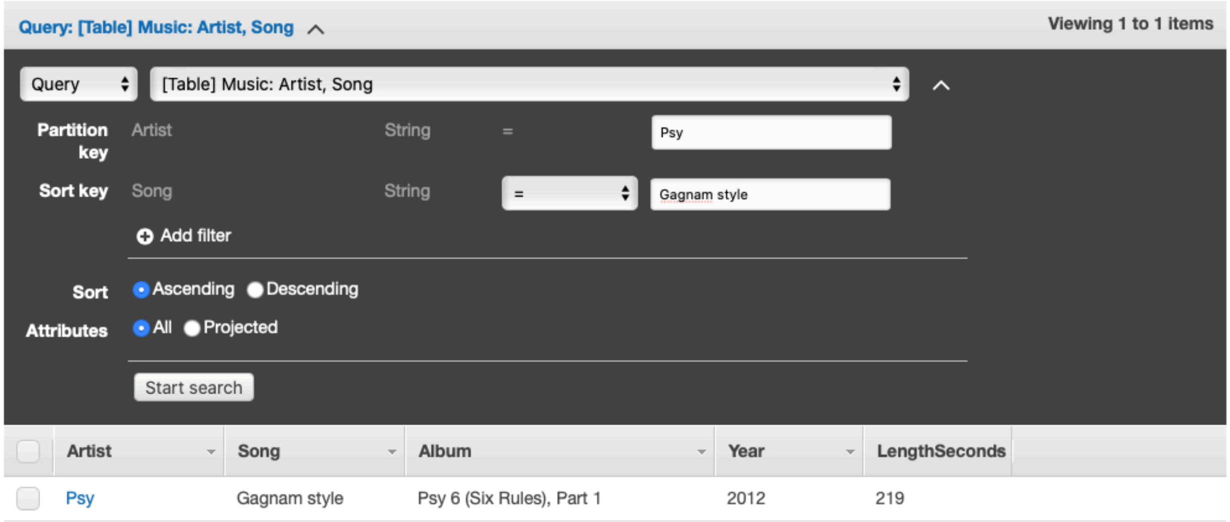
Sort key: Song String = Gangnam Style

Filter: Enter attribute | String | = | Enter value

Sort: ● Ascending ○ Descending

Attributes: ● All ○ Projected

□	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

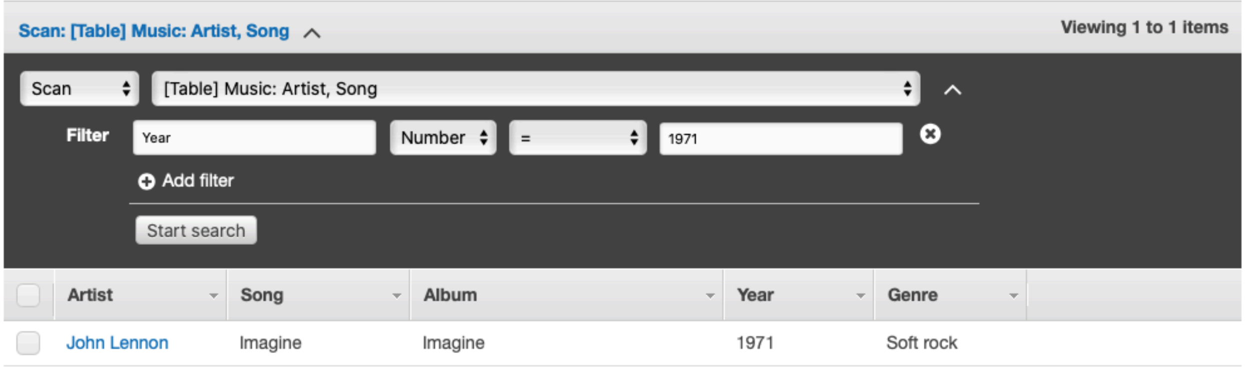
Sort: ● Ascending ○ Descending

Attributes: ● All ○ Projected

Start search

□	Artist	Song	Album	Year	LengthSeconds
□	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

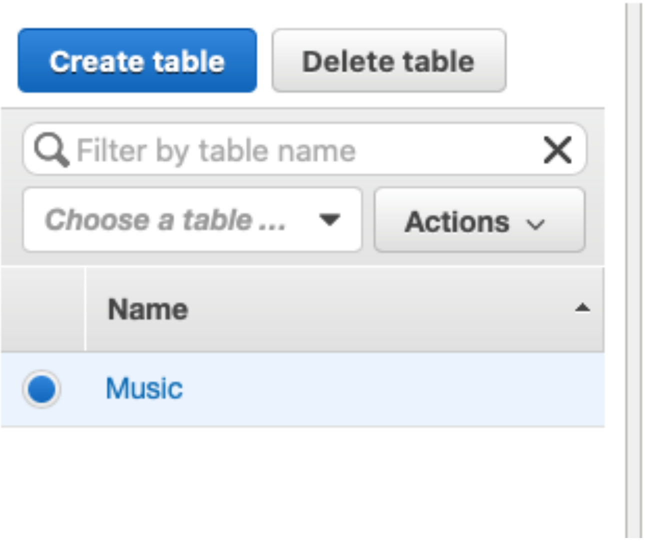
Filter: Year Number = 1971

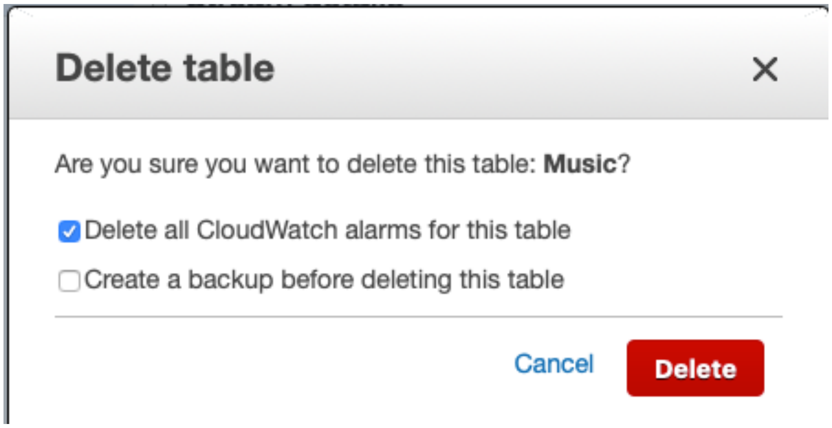
Add filter

Start search

□	Artist	Song	Album	Year	Genre
□	John Lennon	Imagine	Imagine	1971	Soft rock

- Deleting the table





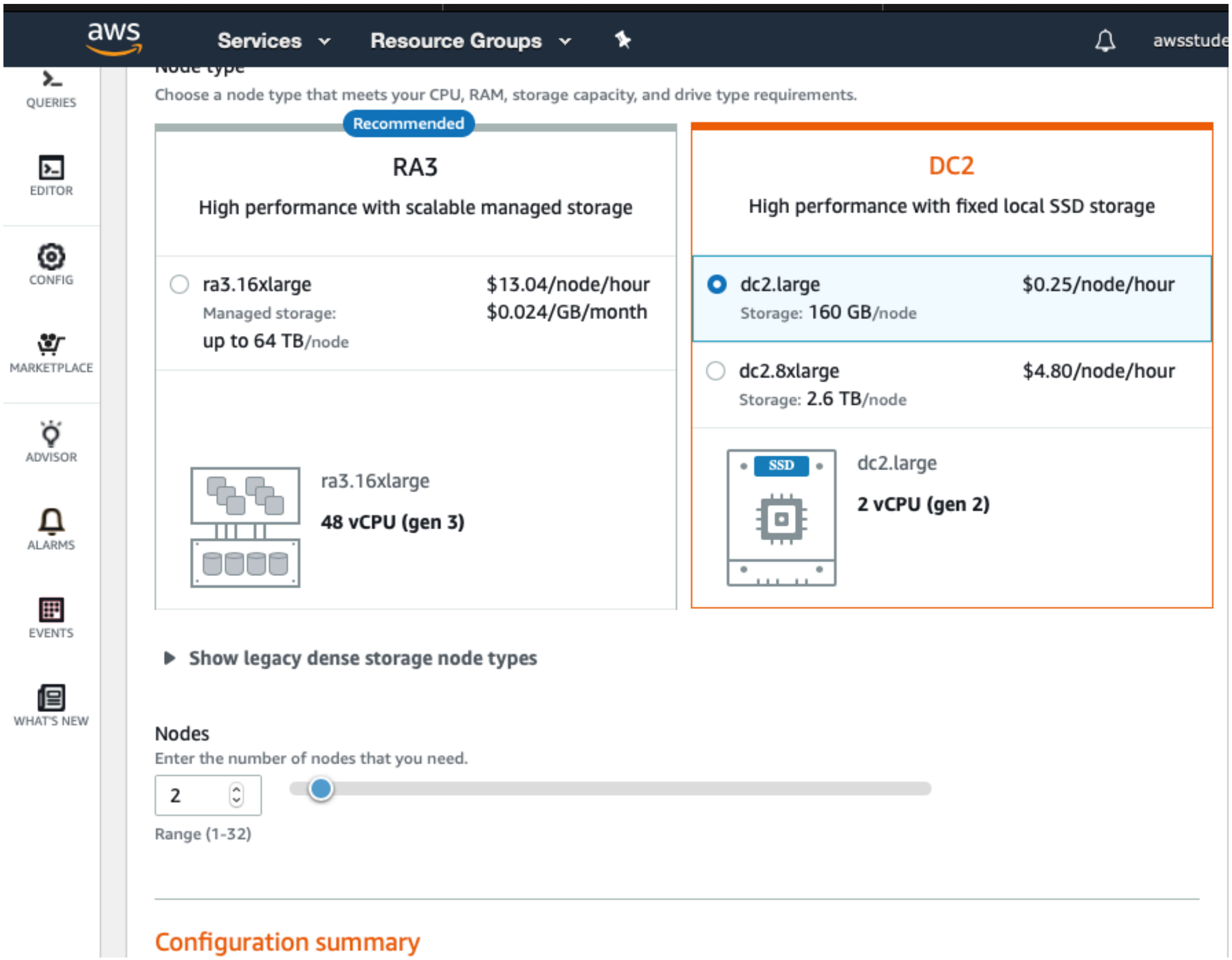
Are you sure you want to delete this table: **Music**?

☑ Delete all CloudWatch alarms for this table

☐ Create a backup before deleting this table

Lab 3 : Working with Amazon Redshift

- Creating clusters using amazon redshift



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: up to 64 TB/node — $0.024/GB/month

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

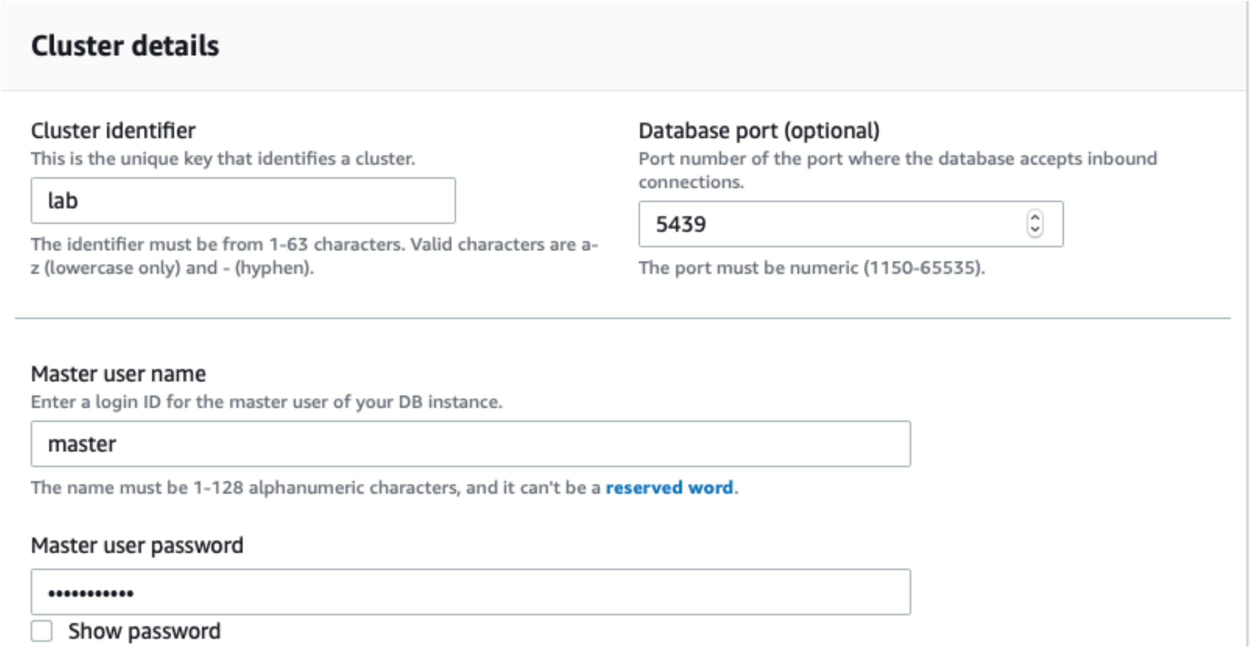
Nodes

Enter the number of nodes that you need.

2

Range (1-32)

Configuration summary



Cluster details

Cluster identifier

This is the unique key that identifies a cluster.

lab

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.

5439

The port must be numeric (1150-65535).

Master user name

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

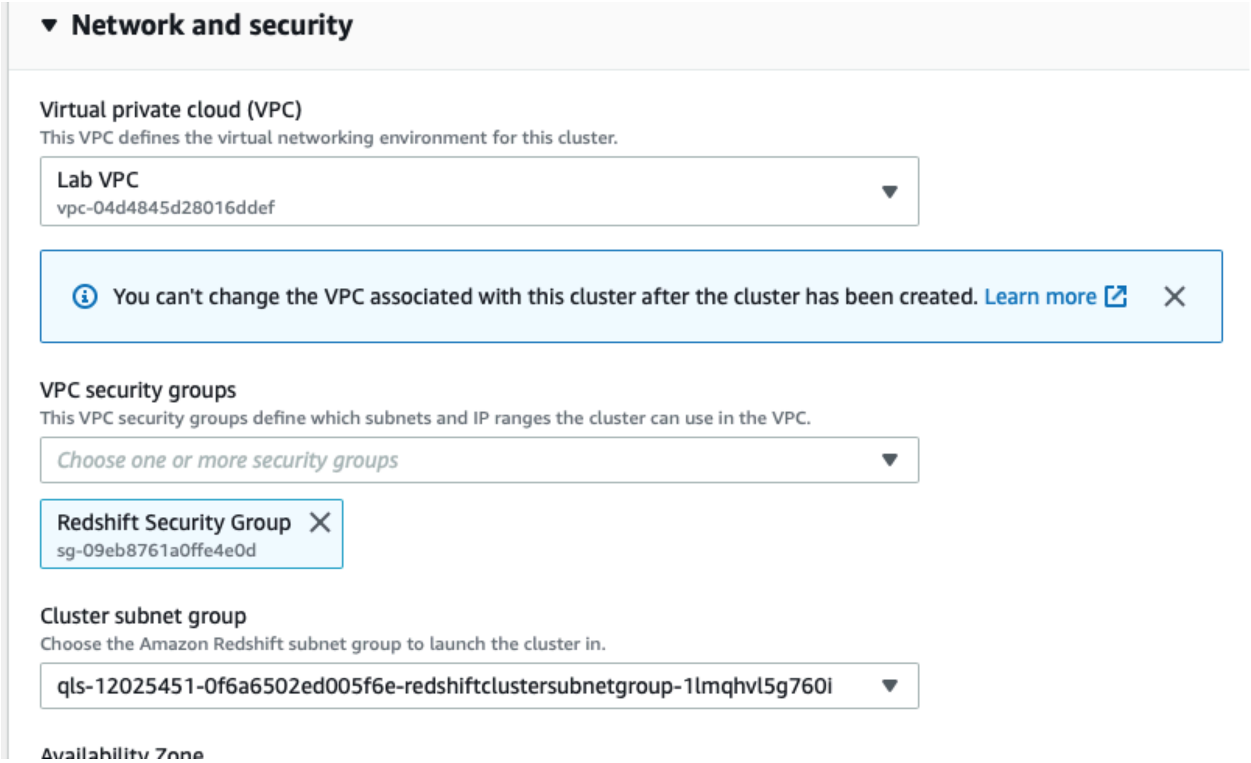
master

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.

Lab VPC
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.

Choose one or more security groups ▼

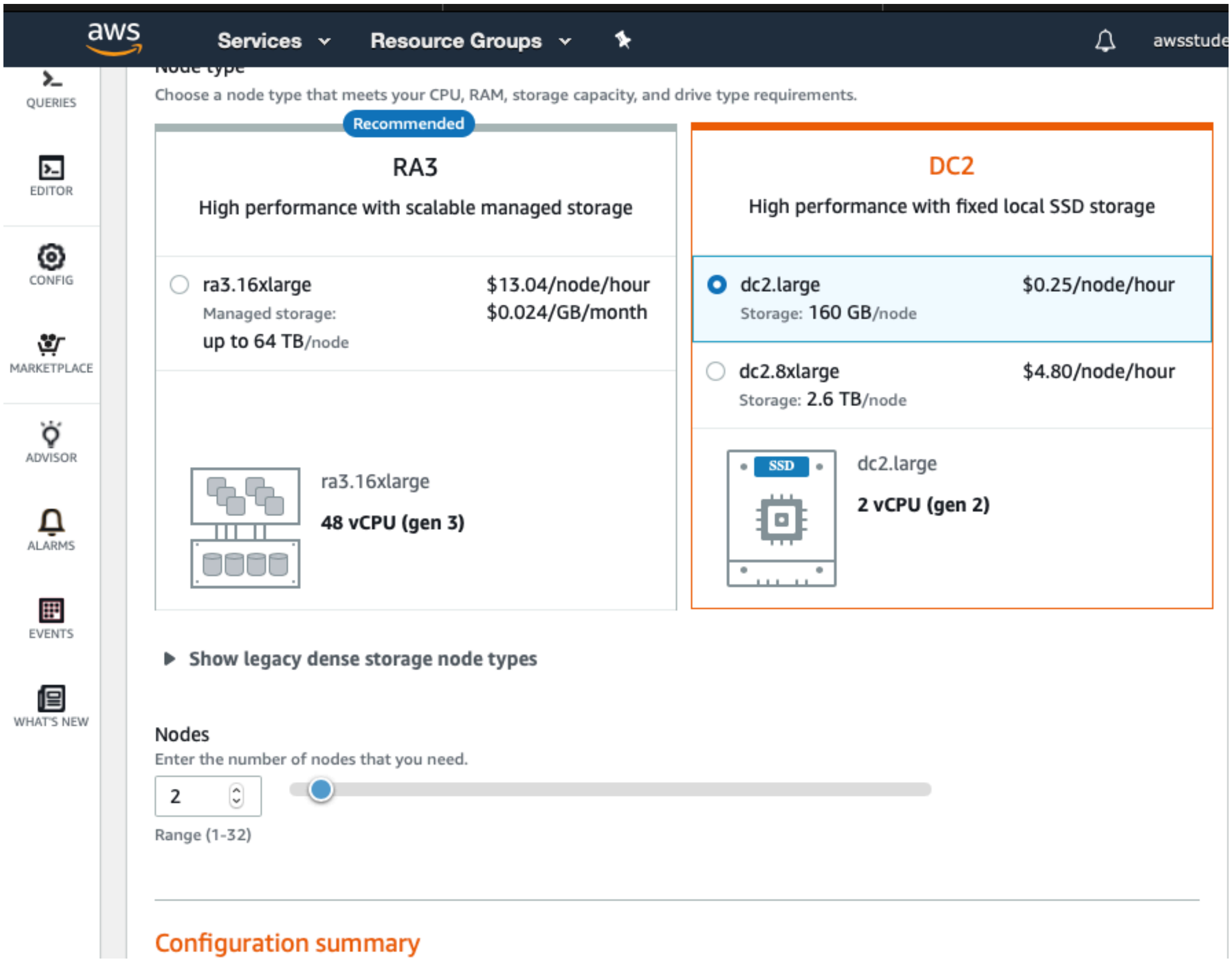
Redshift Security Group ✕
sg-09eb8761a0ffe4e0d

Cluster subnet group

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

qls-12025451-0f6a6502ed005f6e-redshiftclustersubnetgroup-1lmqhvl5g760i ▼

Availability Zone



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: up to 64 TB/node — $0.024/GB/month

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

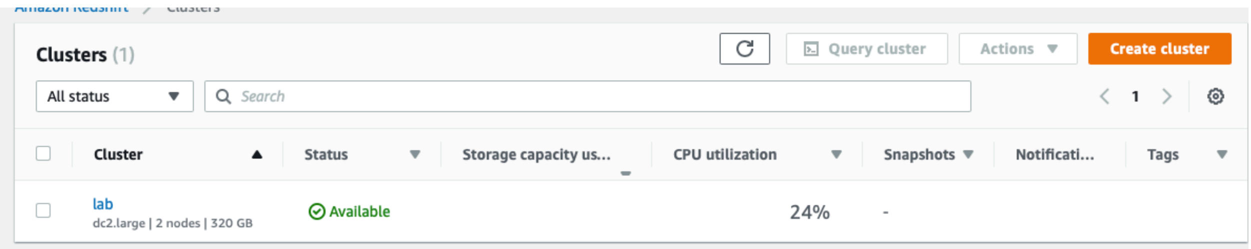
Enter the number of nodes that you need.

2

Range (1-32)

Configuration summary

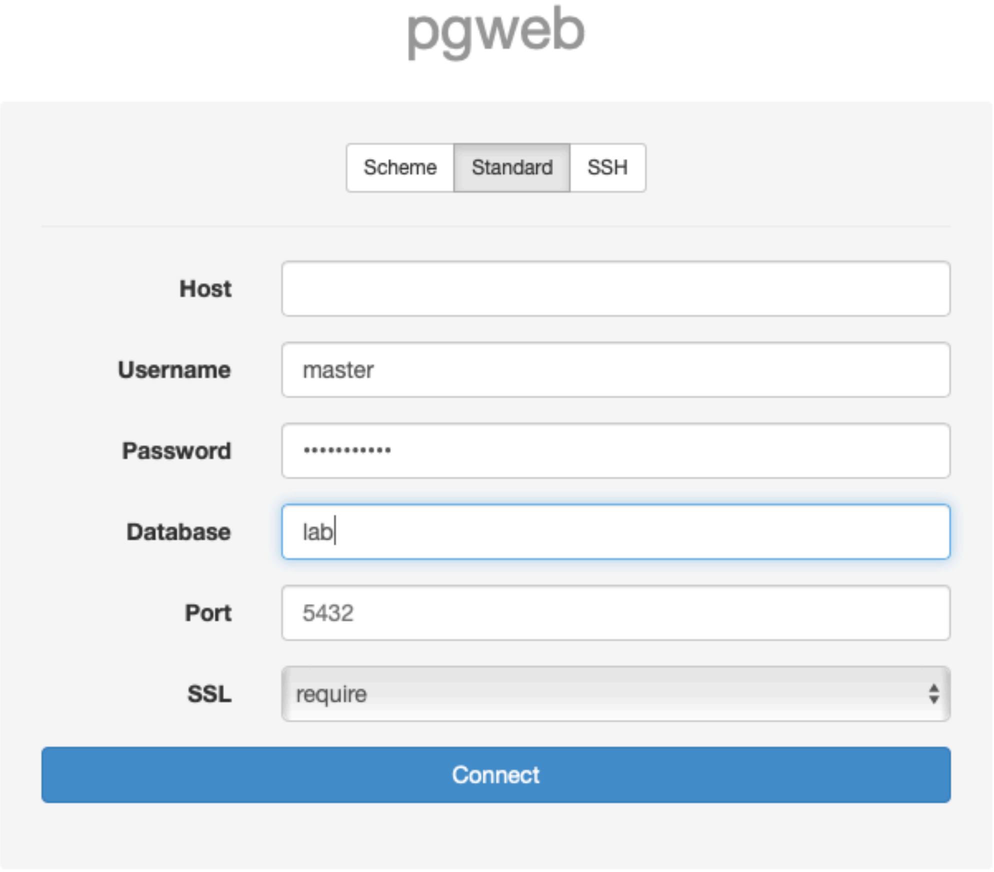
- Cluster has been created with the above configuration



Clusters (1)

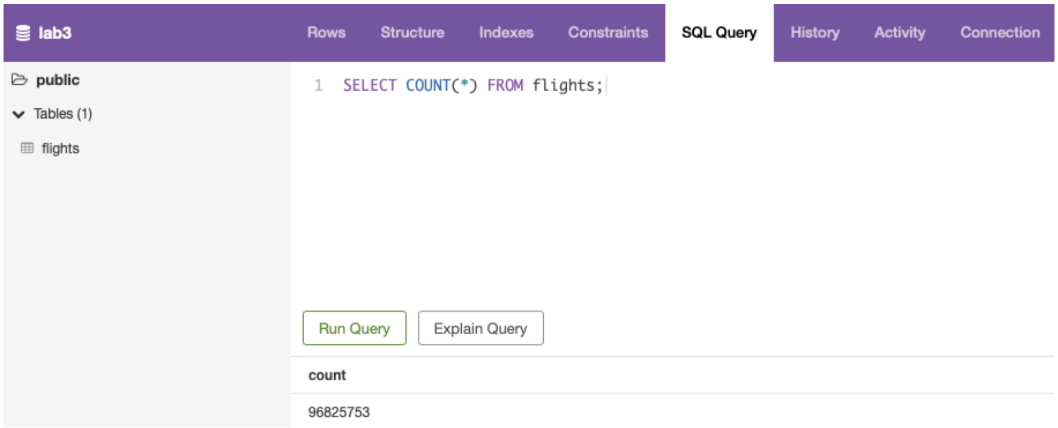
☐	Cluster	Status	Storage capacity us...	CPU utilization	Snapshots	Notificati...	Tags
☐	lab dc2.large | 2 nodes | 320 GB	✓ Available		24%	-		

- Connecting to the cluster using pgweb IP which was provided



Host	
Username	master
Password	•••••••••••
Database	lab
Port	5432
SSL	require

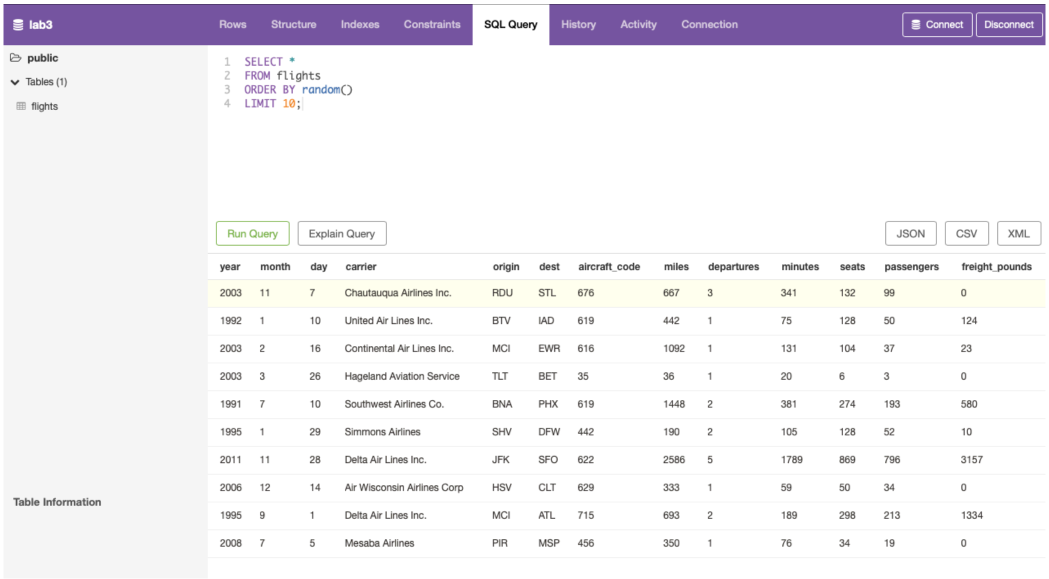
- The table flight was created post which the number of rows in the table is being fetched



```
SELECT COUNT(*) FROM flights;
```

count
96825753

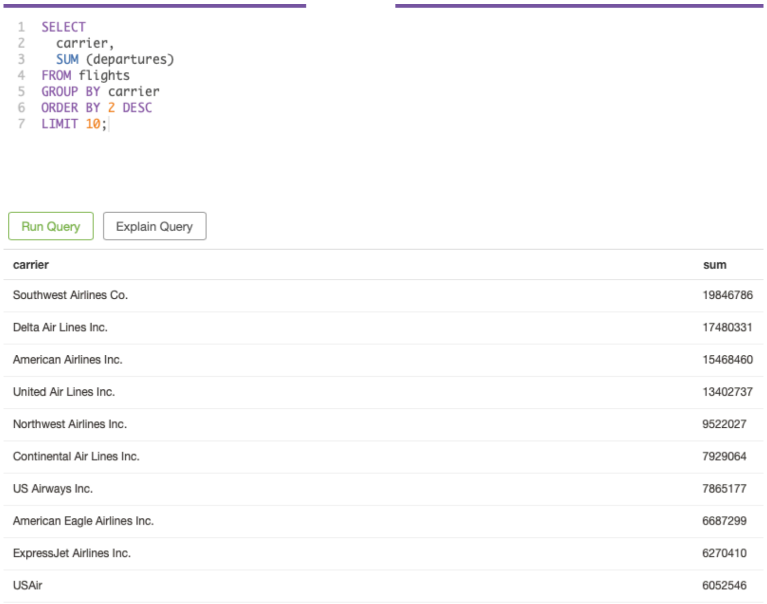
- The data has been copied to the new table which was created. The below query will fetch the data from the table



```
SELECT *
FROM flights
ORDER BY random()
LIMIT 10;
```

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	EWR	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

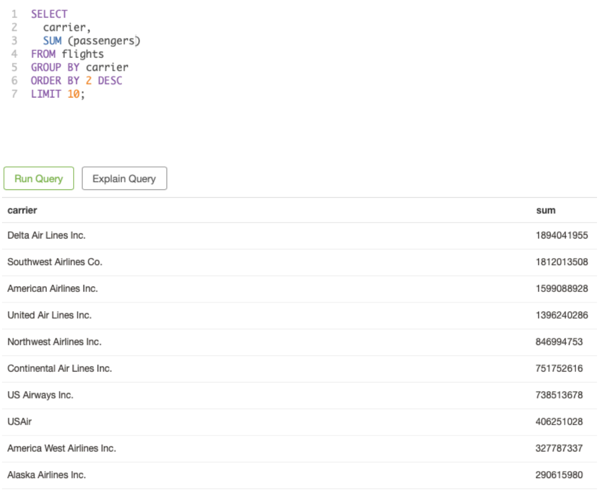
- In the below query we are grouping the result based on the carrier. The top three carriers based on number of departures are Southwest Airlines Co., Delta Airlines Co., and American Airlines Inc.



```
SELECT
  carrier,
  SUM (departures)
FROM flights
GROUP BY carrier
ORDER BY 2 DESC
LIMIT 10;
```

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

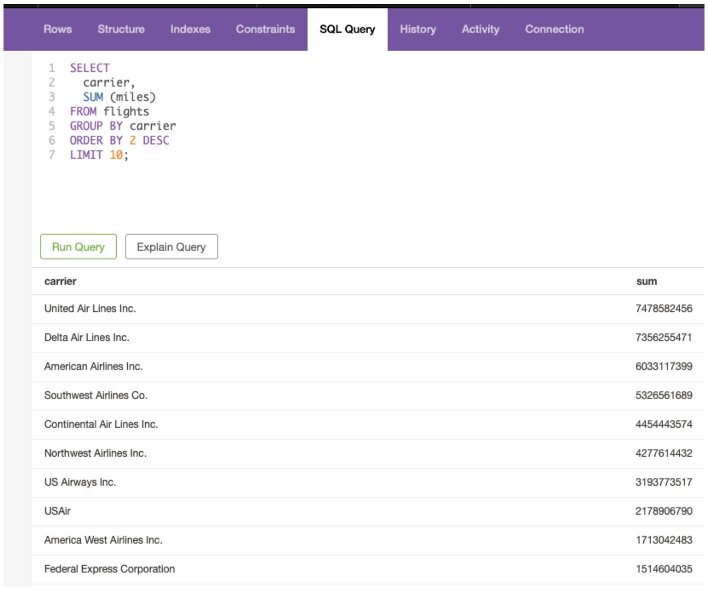
- The top three carriers based on number of passengers are Delta Airlines Co., Southwest Airlines Co., and American Airlines Inc.



```sql
SELECT
  carrier,
  SUM (passengers)
FROM flights
GROUP BY carrier
ORDER BY 2 DESC
LIMIT 10;
```

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

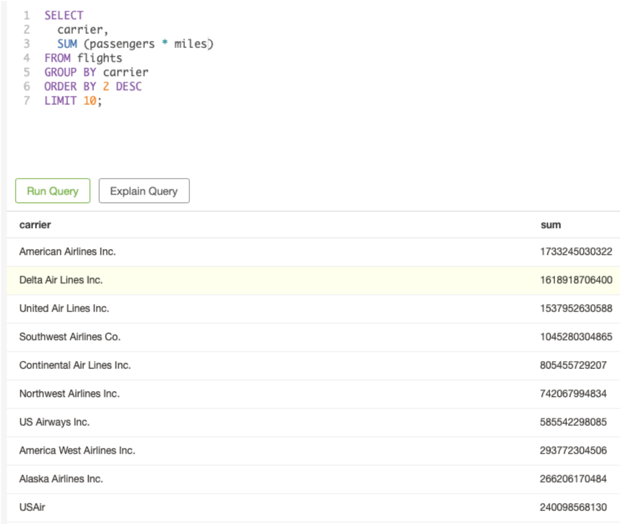
- The top three carriers based on miles flown are United airlines Inc., Southwest Airlines Co., and American Airlines Inc.



```sql
SELECT
  carrier,
  SUM (miles)
FROM flights
GROUP BY carrier
ORDER BY 2 DESC
LIMIT 10;
```

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

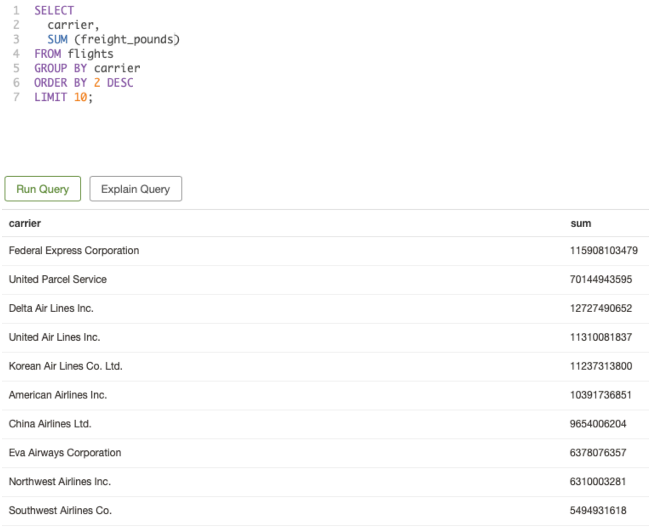
- The top three carriers based on passenger-miles flown are Southwest Airlines Co., Delta Airlines Co., and American Airlines Inc.



```
SELECT
  carrier,
  SUM (passengers * miles)
FROM flights
GROUP BY carrier
ORDER BY 2 DESC
LIMIT 10;
```

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

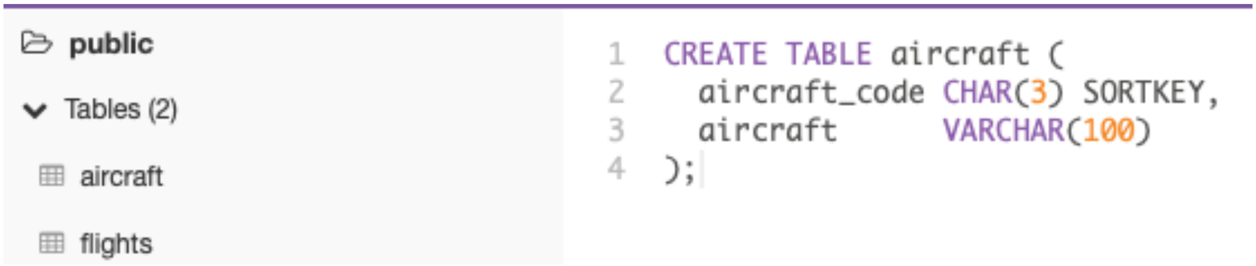
- The top three carriers based on freight pounds are Federal Express Corporation, United Parcel Service and Delta Airlines Inc.



```
SELECT
  carrier,
  SUM (freight_pounds)
FROM flights
GROUP BY carrier
ORDER BY 2 DESC
LIMIT 10;
```

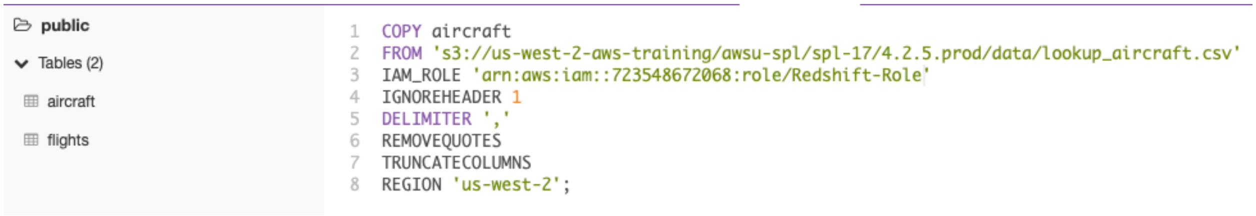
carrier	sum
Federal Express Corporation	115908103479
United Parcel Service	70144943595
Delta Air Lines Inc.	12727490652
United Air Lines Inc.	11310081837
Korean Air Lines Co. Ltd.	11237313800
American Airlines Inc.	10391736851
China Airlines Ltd.	9654006204
Eva Airways Corporation	6378076357
Northwest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

- Creating a new table aircraft for joining



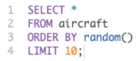
```sql
CREATE TABLE aircraft (
  aircraft_code CHAR(3) SORTKEY,
  aircraft      VARCHAR(100)
);
```

- Copying data from the bucket to the aircraft table

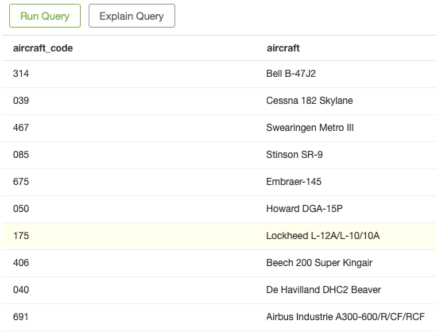


```sql
COPY aircraft
FROM 's3://us-west-2-aws-training/awsu-spl/spl-17/4.2.5.prod/data/lookup_aircraft.csv'
IAM_ROLE 'arn:aws:iam::723548672068:role/Redshift-Role'
IGNOREHEADER 1
DELIMITER ','
REMOVEQUOTES
TRUNCATECOLUMNS
REGION 'us-west-2';
```

- Selecting the data from the newly created table aircraft

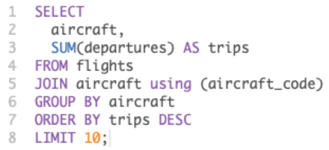


```sql
SELECT *
FROM aircraft
ORDER BY random()
LIMIT 10;
```

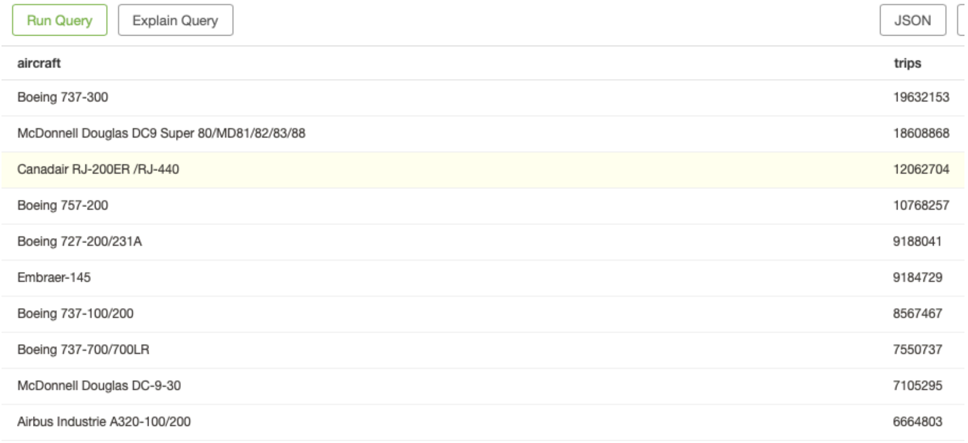


aircraft_code	aircraft
314	Bell B-47J2
039	Cessna 182 Skylane
467	Swearingen Metro III
085	Stinson SR-9
675	Embraer-145
050	Howard DGA-15P
175	Lockheed L-12A/L-10/10A
406	Beech 200 Super Kingair
040	De Havilland DHC2 Beaver
691	Airbus Industrie A300-600/R/CF/RCF

- Joining the tables aircraft and flight ordering by trips

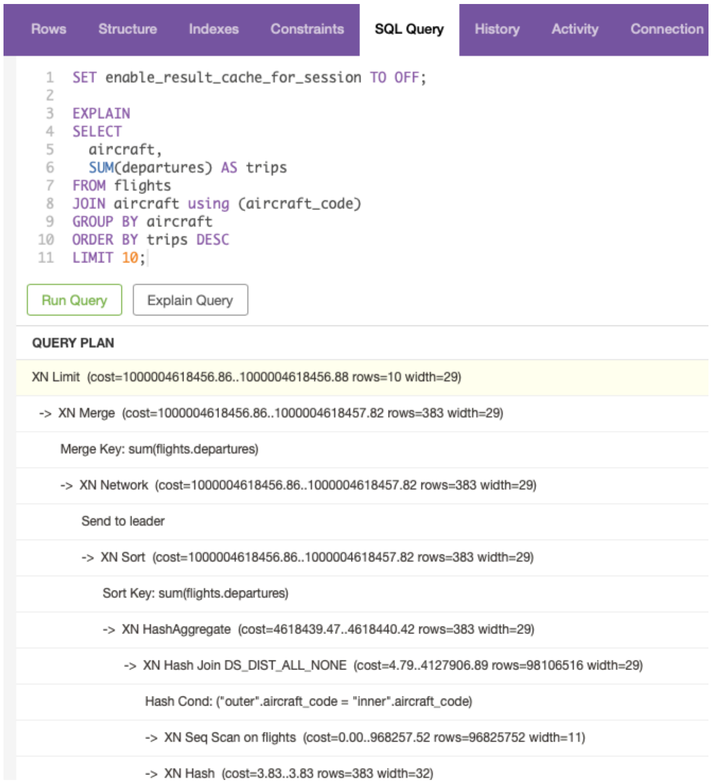


```sql
SELECT
  aircraft,
  SUM(departures) AS trips
FROM flights
JOIN aircraft using (aircraft_code)
GROUP BY aircraft
ORDER BY trips DESC
LIMIT 10;
```



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.

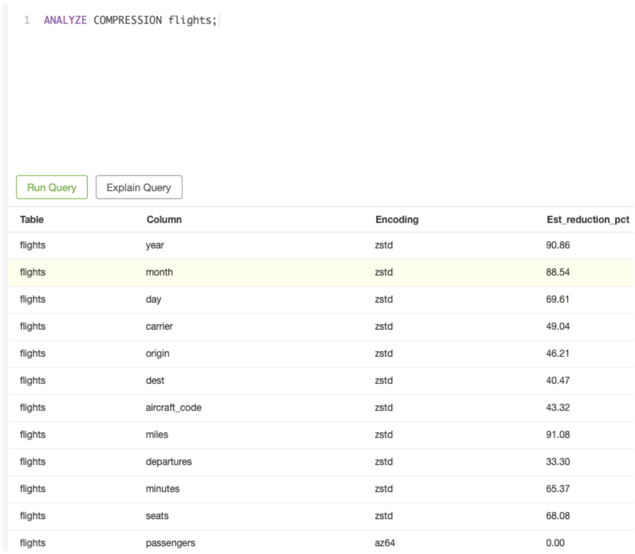


```
SET enable_result_cache_for_session TO OFF;

EXPLAIN
SELECT
  aircraft,
  SUM(departures) AS trips
FROM flights
JOIN aircraft using (aircraft_code)
GROUP BY aircraft
ORDER BY trips DESC
LIMIT 10;
```

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)
```

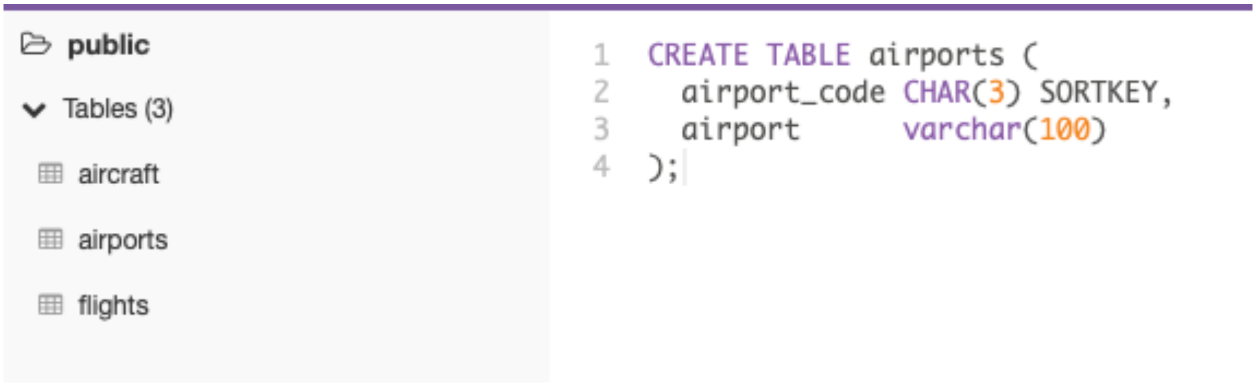
- Analyzing the data on the table flight



```sql
ANALYZE COMPRESSION flights;
```

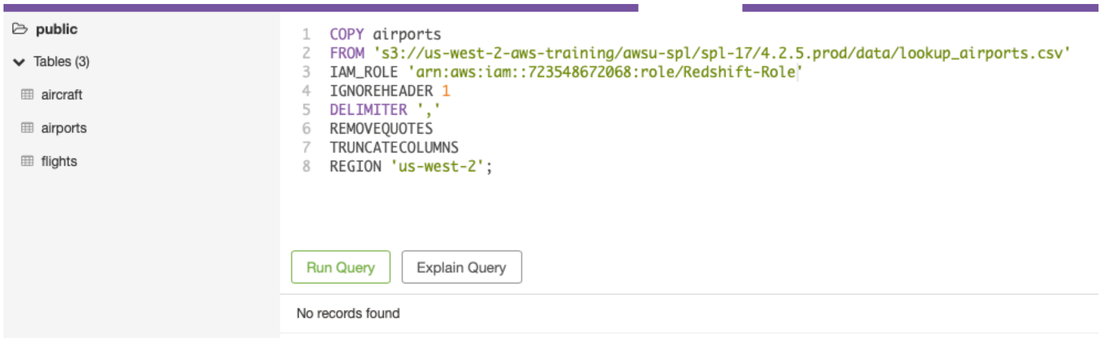
Table	Column	Encoding	Est_reduction_pct
flights	year	zstd	90.86
flights	month	zstd	88.54
flights	day	zstd	69.61
flights	carrier	zstd	49.04
flights	origin	zstd	46.21
flights	dest	zstd	40.47
flights	aircraft_code	zstd	43.32
flights	miles	zstd	91.08
flights	departures	zstd	33.30
flights	minutes	zstd	65.37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

- Creating tables from other table



```sql
CREATE TABLE airports (
  airport_code CHAR(3) SORTKEY,
  airport      varchar(100)
);
```

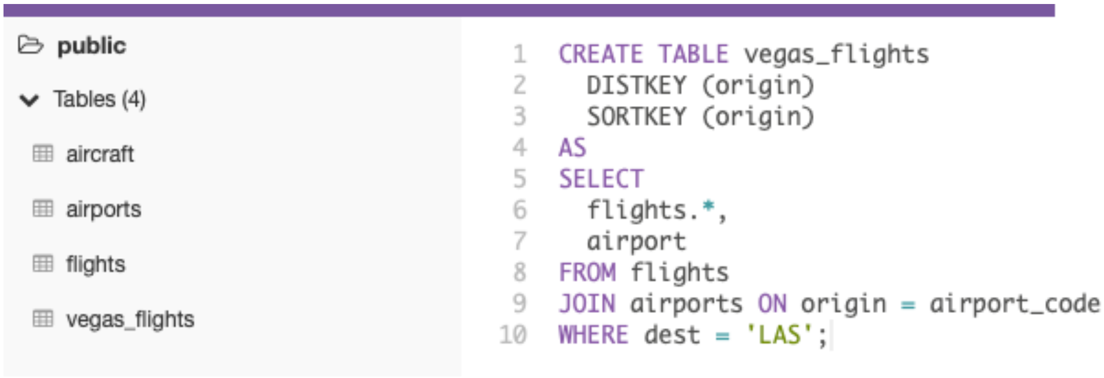
- Copying the data from bucket to the airports table



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

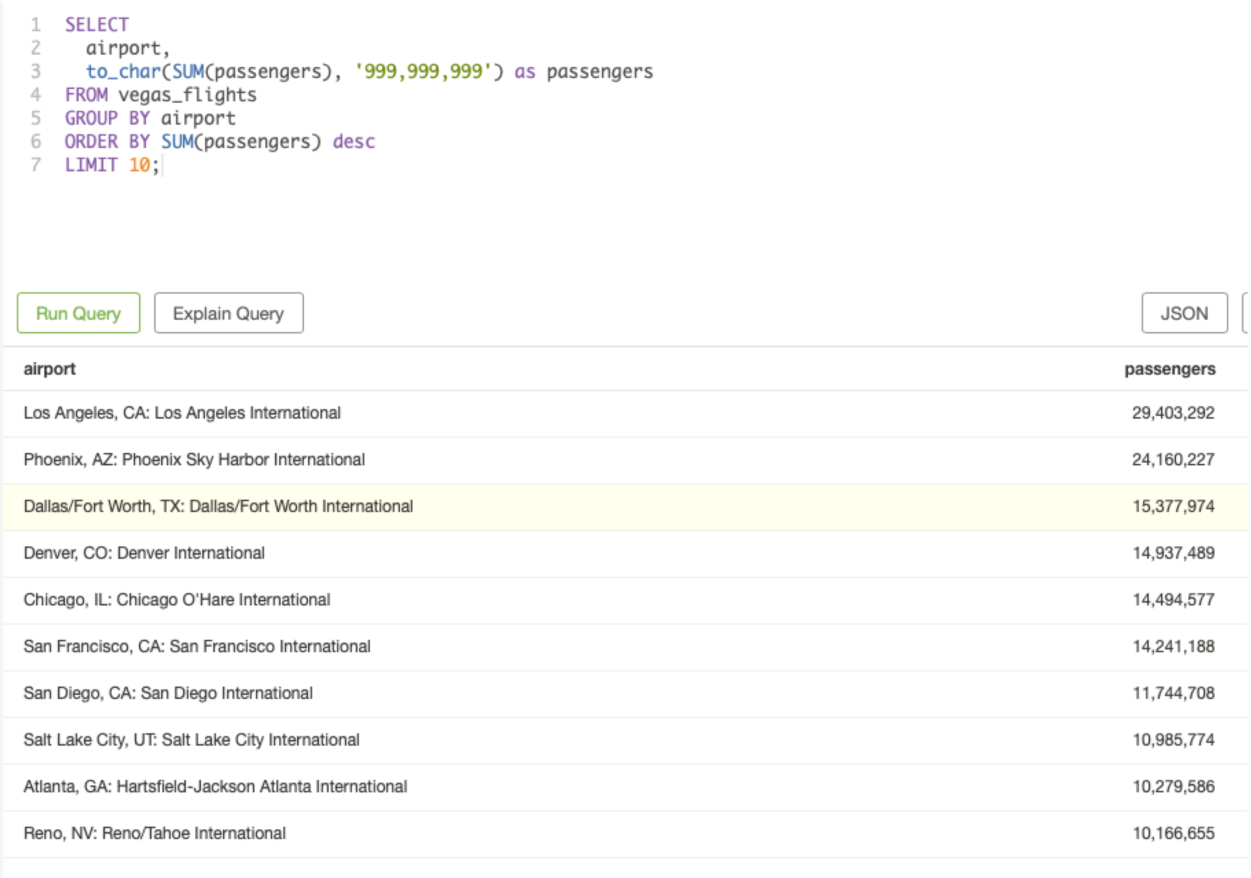
No records found

- Creating a new table vegas_flights to load the data



```
CREATE TABLE vegas_flights
  DISTKEY (origin)
  SORTKEY (origin)
AS
SELECT
  flights.*,
  airport
FROM flights
JOIN airports ON origin = airport_code
WHERE dest = 'LAS';
```

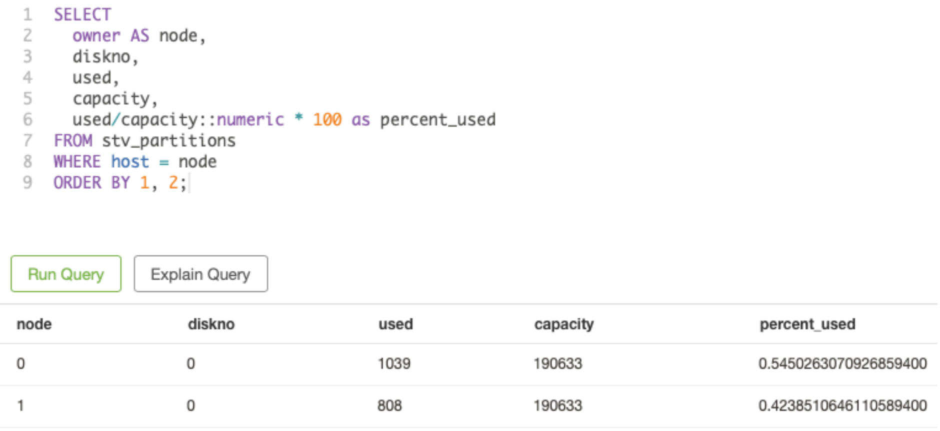
- Selecting the data loaded from the airport table



```sql
SELECT
  airport,
  to_char(SUM(passengers), '999,999,999') as passengers
FROM vegas_flights
GROUP BY airport
ORDER BY SUM(passengers) desc
LIMIT 10;
```

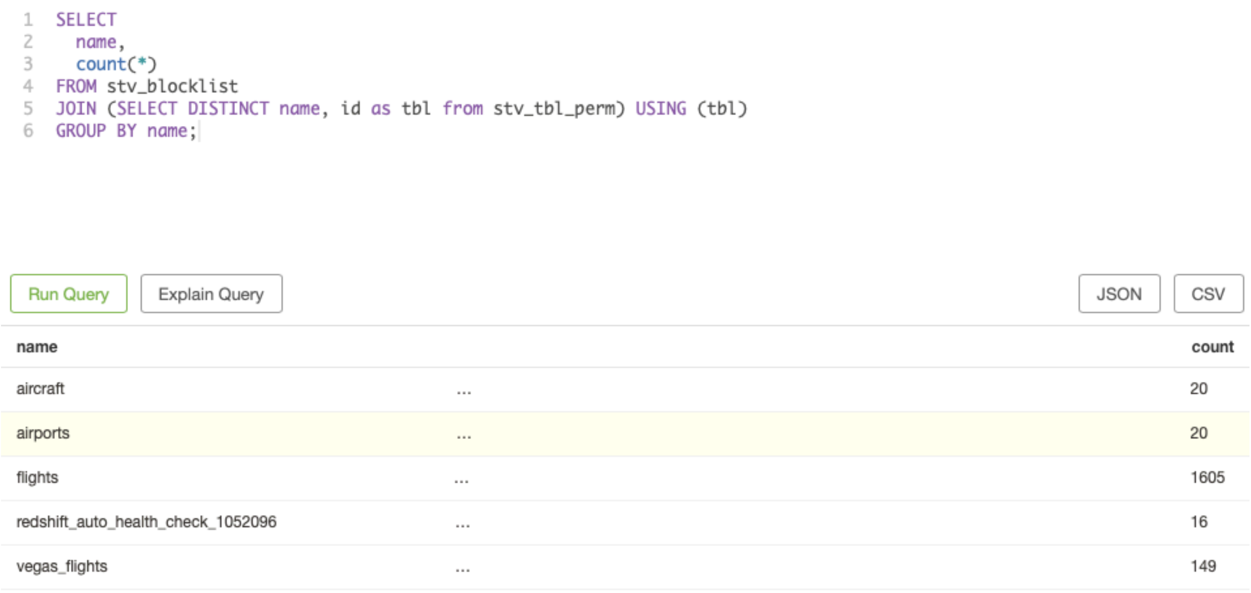
airport	passengers
Los Angeles, CA: Los Angeles International	29,403,292
Phoenix, AZ: Phoenix Sky Harbor International	24,160,227
Dallas/Fort Worth, TX: Dallas/Fort Worth International	15,377,974
Denver, CO: Denver International	14,937,489
Chicago, IL: Chicago O'Hare International	14,494,577
San Francisco, 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



```sql
SELECT
  owner AS node,
  diskno,
  used,
  capacity,
  used/capacity::numeric * 100 as percent_used
FROM stv_partitions
WHERE host = node
ORDER BY 1, 2;
```

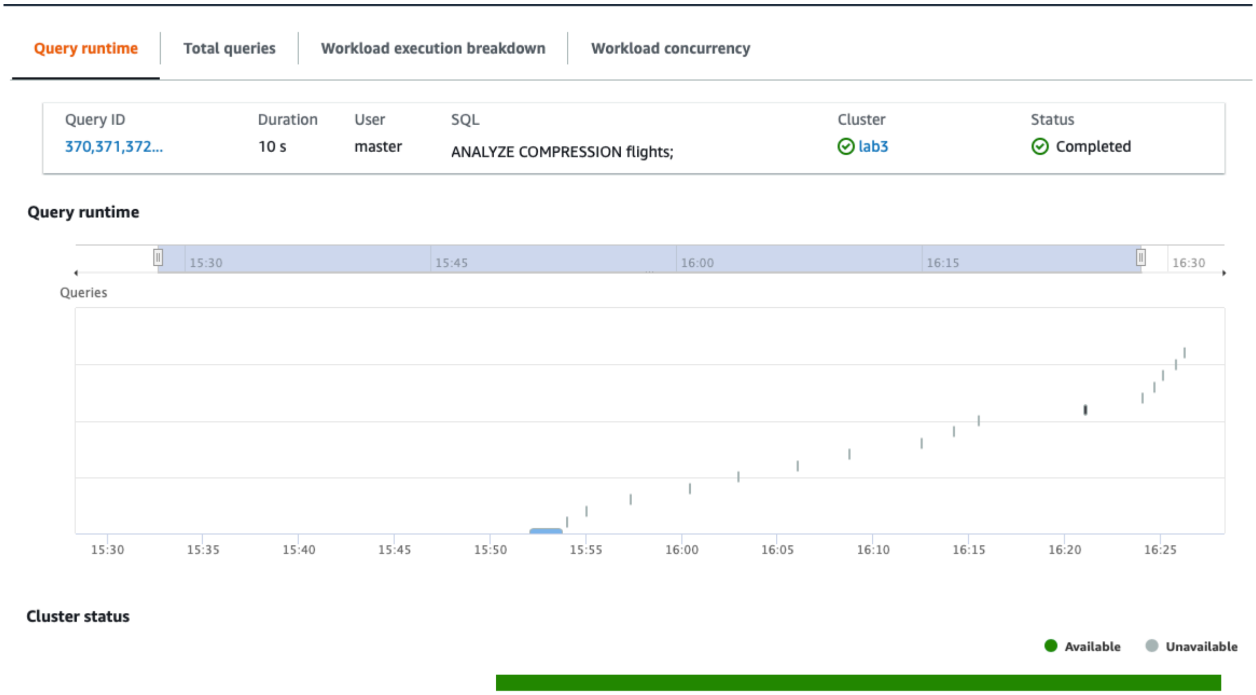
node	diskno	used	capacity	percent_used
0	0	1039	190633	0.5450263070926859400
1	0	808	190633	0.4238510646110589400



```
SELECT
  name,
  count(*)
FROM stv_blocklist
JOIN (SELECT DISTINCT name, id as tbl from stv_tbl_perm) USING (tbl)
GROUP BY name;
```

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

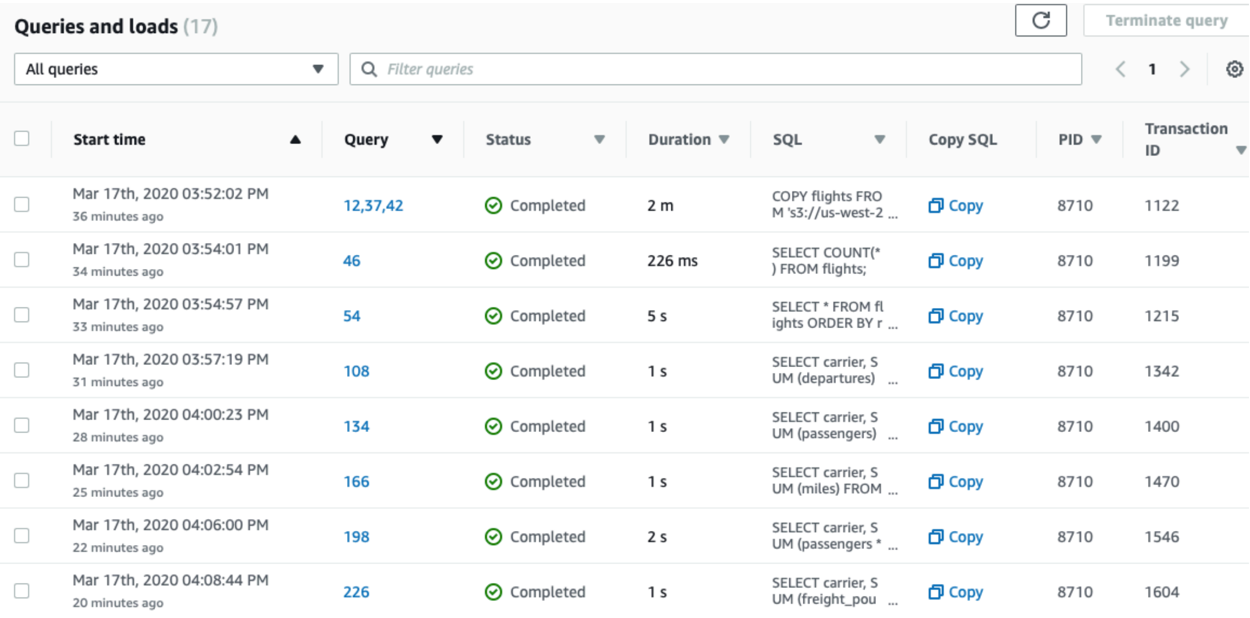
- Exploring the query performance using redshift console



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

Query runtime

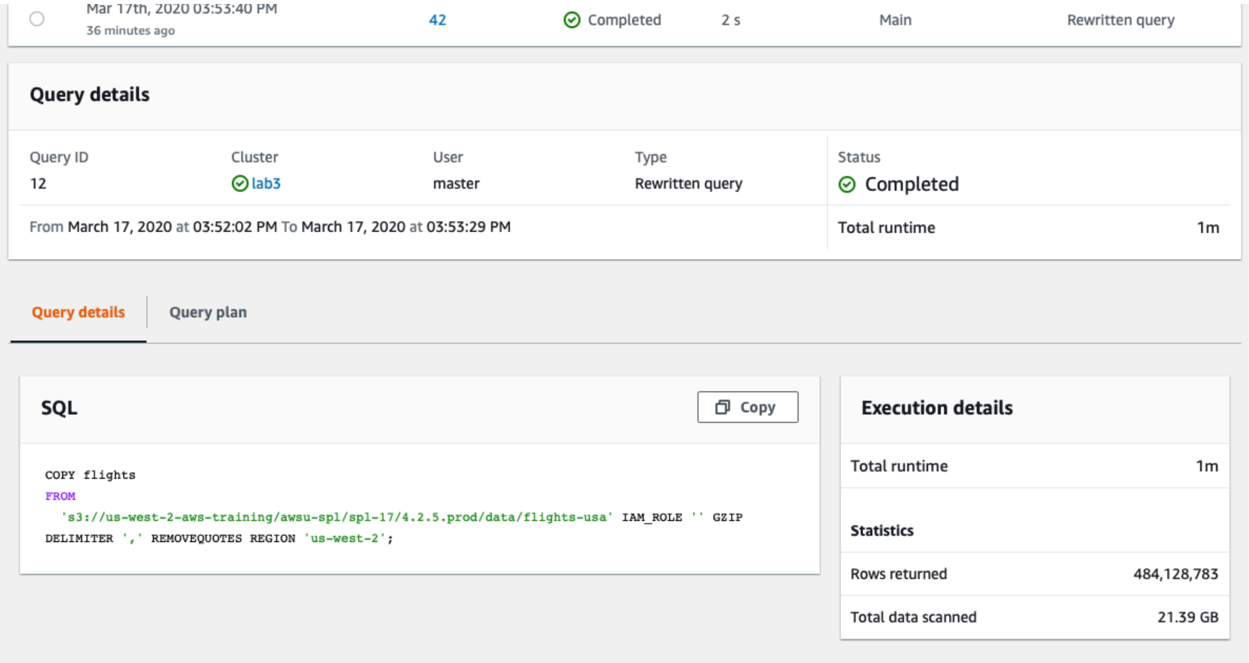
Cluster status



Queries and loads (17)

□	Start time ▲	Query ▼	Status ▼	Duration ▼	SQL ▼	Copy SQL	PID ▼	Transaction ID ▼
□	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
□	Mar 17th, 2020 03:54:01 PM 34 minutes ago	46	✓ Completed	226 ms	SELECT COUNT(*) FROM flights;	Copy	8710	1199
□	Mar 17th, 2020 03:54:57 PM 33 minutes ago	54	✓ Completed	5 s	SELECT * FROM flights ORDER BY r ...	Copy	8710	1215
□	Mar 17th, 2020 03:57:19 PM 31 minutes ago	108	✓ Completed	1 s	SELECT carrier, SUM (departures) ...	Copy	8710	1342
□	Mar 17th, 2020 04:00:23 PM 28 minutes ago	134	✓ Completed	1 s	SELECT carrier, SUM (passengers) ...	Copy	8710	1400
□	Mar 17th, 2020 04:02:54 PM 25 minutes ago	166	✓ Completed	1 s	SELECT carrier, SUM (miles) FROM ...	Copy	8710	1470
□	Mar 17th, 2020 04:06:00 PM 22 minutes ago	198	✓ Completed	2 s	SELECT carrier, SUM (passengers * ...	Copy	8710	1546
□	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

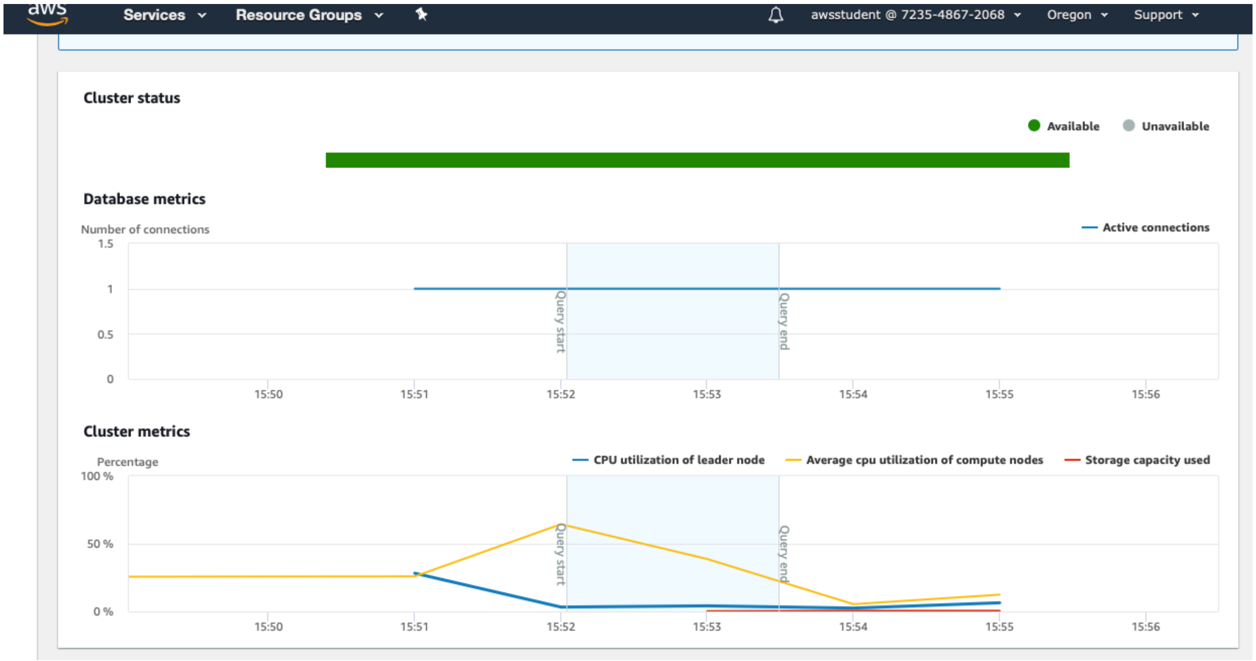
SQL

```
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

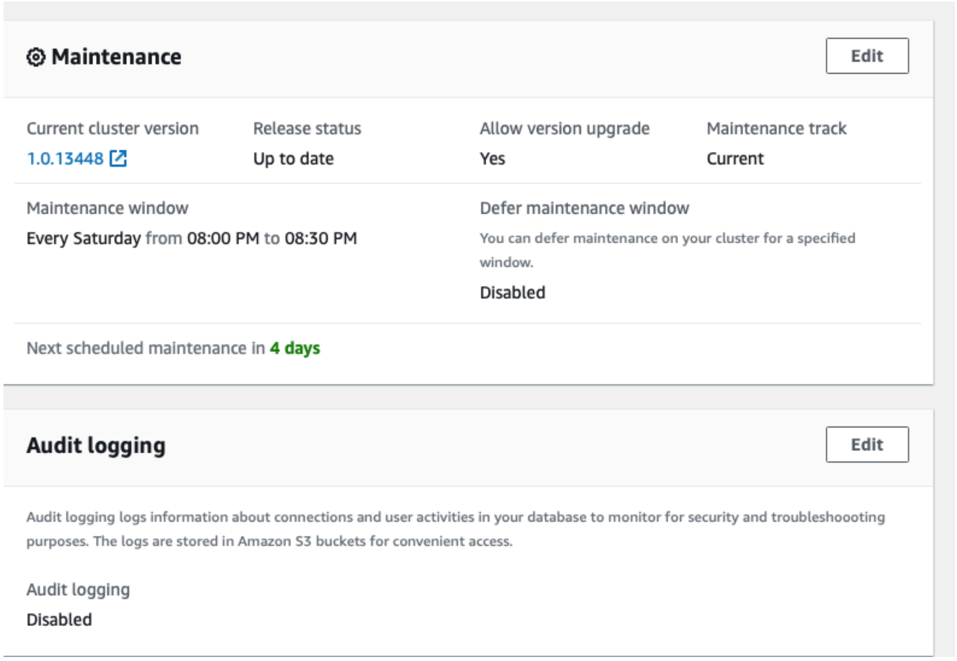
- Verifying the database metrics using cluster console



Cluster status

Database metrics

Cluster metrics



Maintenance

Current cluster version	Release status	Allow version upgrade	Maintenance track
1.0.13448	Up to date	Yes	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**

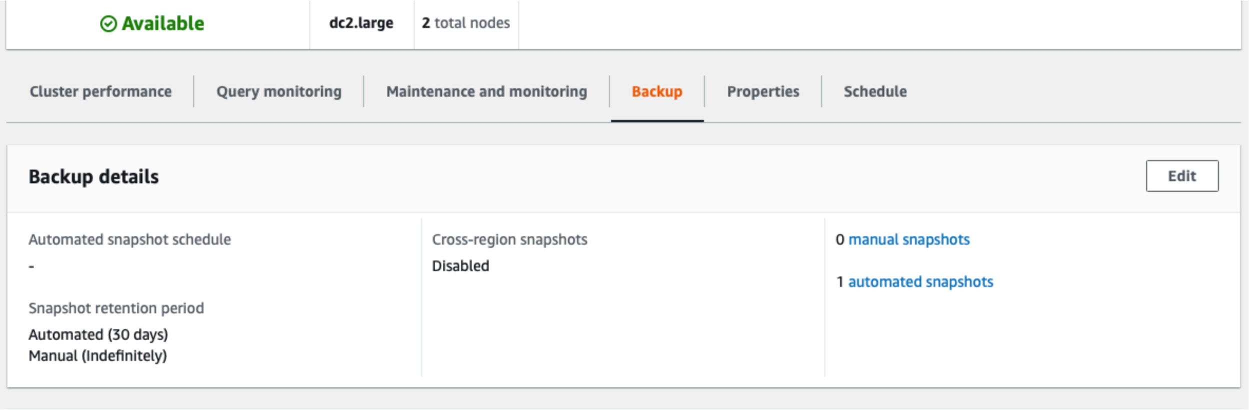
Audit logging

Audit logging logs information about connections and user activities in your database to monitor for security and troubleshoooting 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

Backup details

Automated snapshot schedule
-

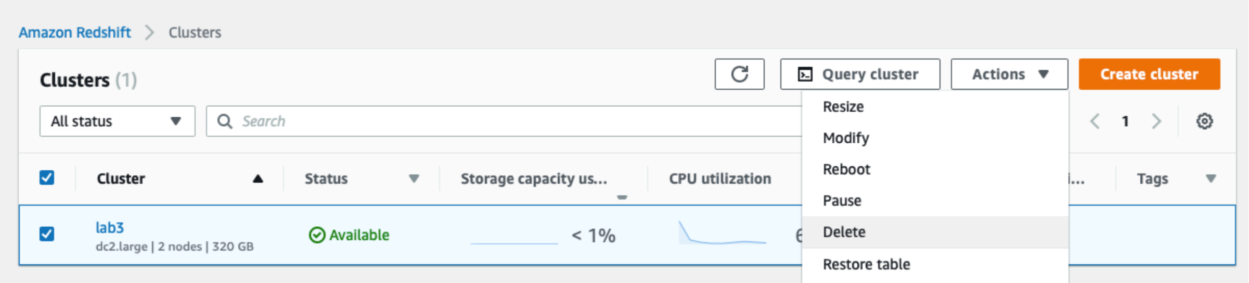
Snapshot retention period
Automated (30 days)
Manual (Indefinitely)

Cross-region snapshots
Disabled

0 manual snapshots

1 automated snapshots

- Deleting the cluster after use.



Clusters (1)

☑	Cluster ▲	Status ▼	Storage capacity us...	CPU utilization	...	Tags ▼
☑	lab3 dc2.large | 2 nodes | 320 GB	Available	< 1%			