Load data from MySQL in local to snowflake using RDS (and matillion)

Official documentations:

https://www.matillion.com/blog/using-the-rds-query-component-in-matillion-etl-for-snowflake-to-load-data-from-postgresql

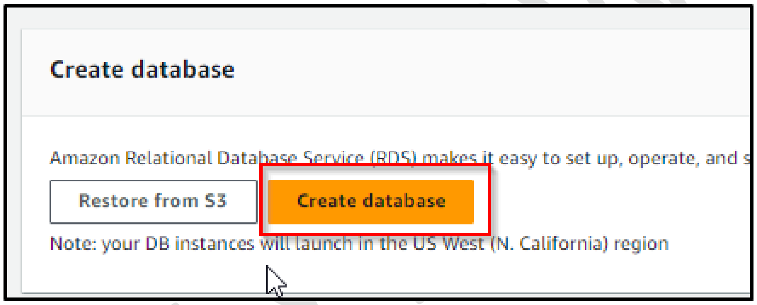
https://docs.matillion.com/metl/docs/2103740/#video

To load data from MySQL in local to snowflake using RDS (and matillion) involves the below steps:

1. Configure an Amazon RDS environment for MySQL
2. In matillion:
   a. Prepare snowflake – create a table
   b. Use RDS component to load into snowflake
3. Run Job and verify results

STEP 1: Create database and Configure an Amazon RDS environment for MySQL

a) Login to AWS console
b) Serach for RDS and load RDS
c) Click on create database



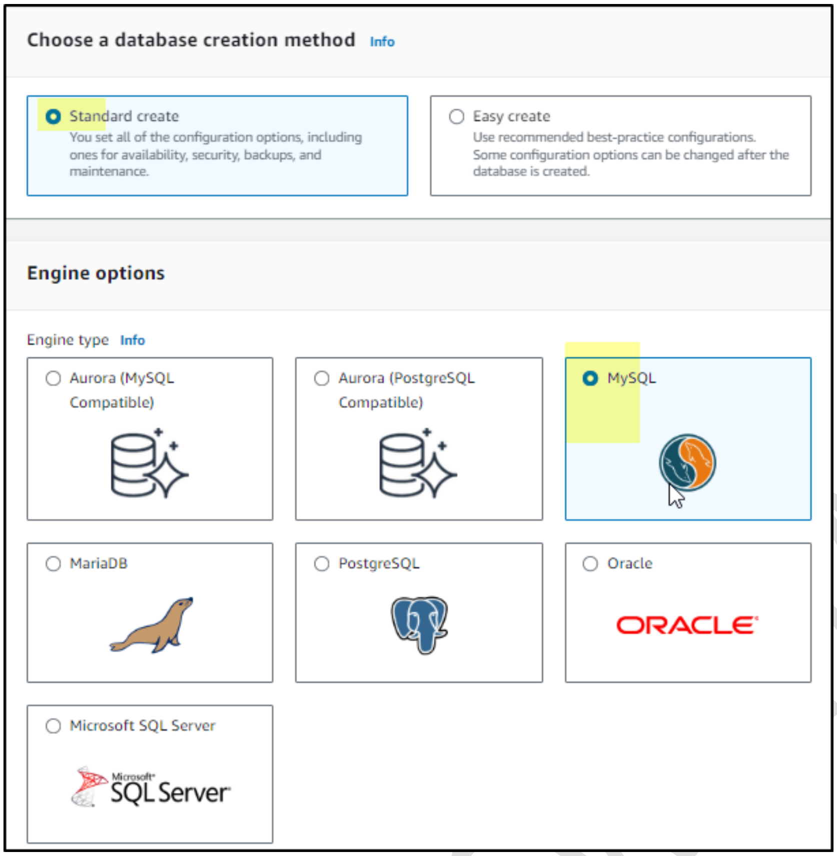
Create database

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and s

Note: your DB instances will launch in the US West (N. California) region

d) Select database creation method and Engine options

Load data from MySQL in local to snowflake using RDS (and matillion)



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.

Engine options

Engine type Info

○ Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

● MySQL

○ MariaDB

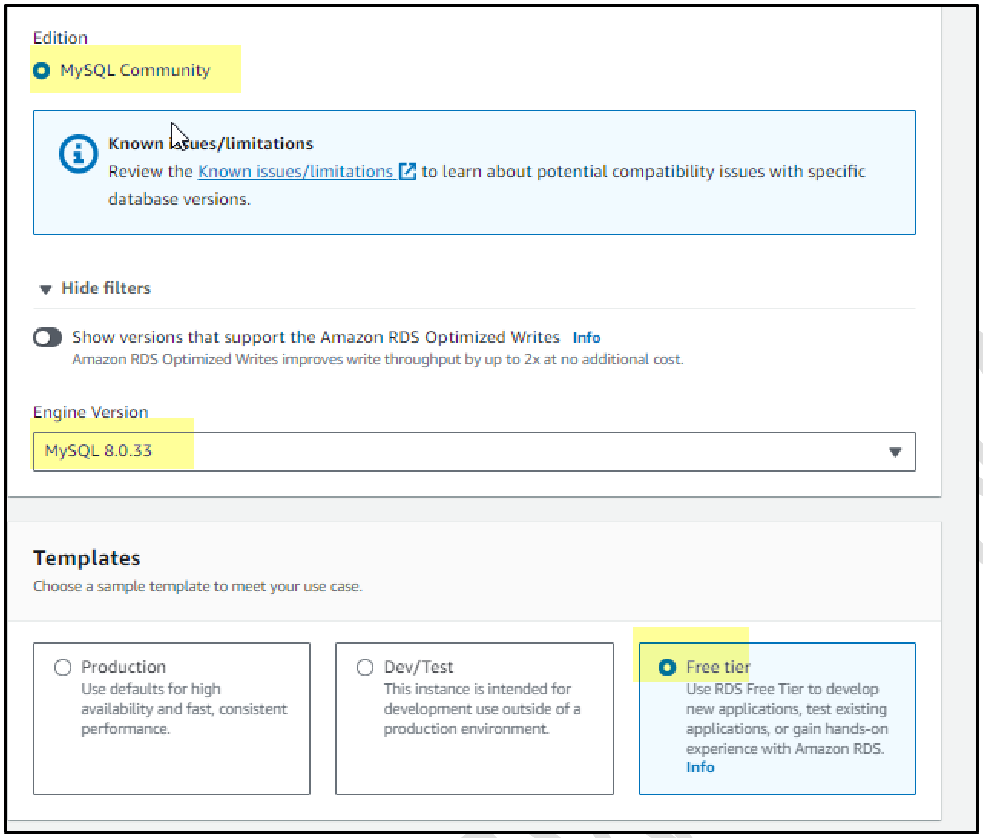
○ PostgreSQL

○ Oracle

○ Microsoft SQL Server

e) Select Edition, engine version and template

Load data from MySQL in local to snowflake using RDS (and matillion)



Edition

● MySQL Community

Known issues/limitations

Review the Known issues/limitations to learn about potential compatibility issues with specific database versions.

▼ Hide filters

Show versions that support the Amazon RDS Optimized Writes Info

Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine Version

MySQL 8.0.33

Templates

Choose a sample template to meet your use case.

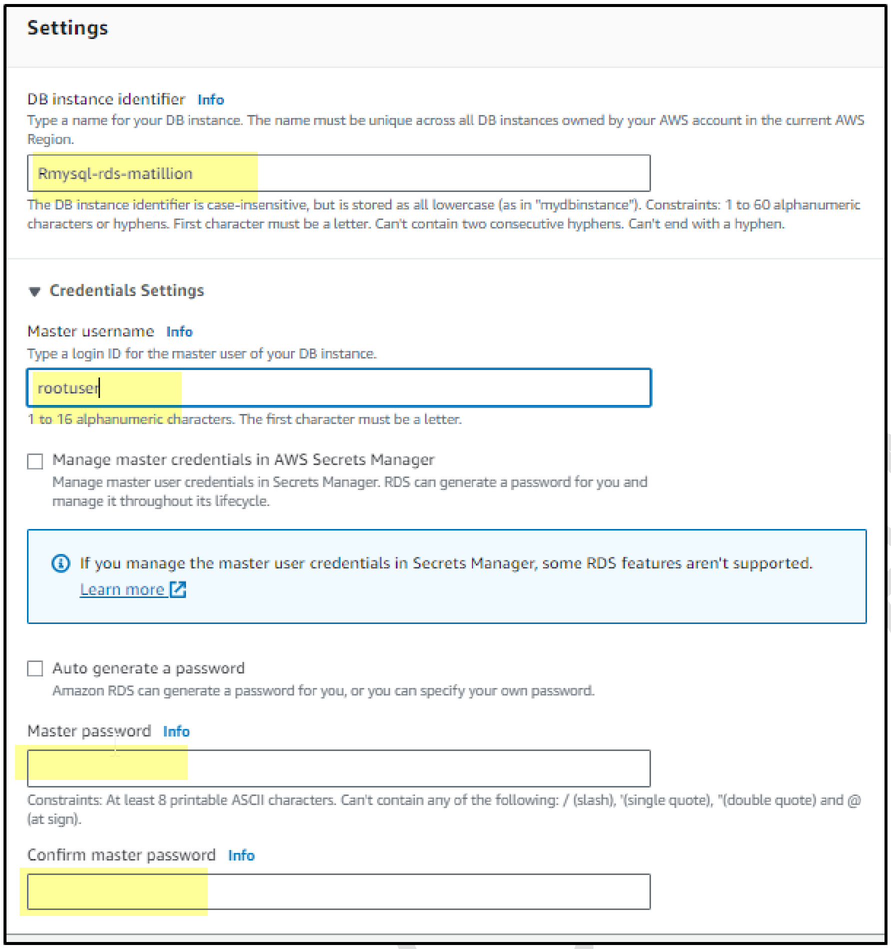
○ Production
Use defaults for high availability and fast, consistent performance.

○ Dev/Test
This instance is intended for development use outside of a production environment.

● Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Info

f) Update settings and instance configuration : You can change the db instance identifier and master username for future convienience. Update the passwords

Load data from MySQL in local to snowflake using RDS (and matillion)



Settings

DB instance identifier Info

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.

Rmysql-rds-matillion

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.

▼ Credentials Settings

Master username Info

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

rootuser

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

□ Manage master credentials in AWS Secrets Manager

Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

ⓘ If you manage the master user credentials in Secrets Manager, some RDS features aren't supported. Learn more

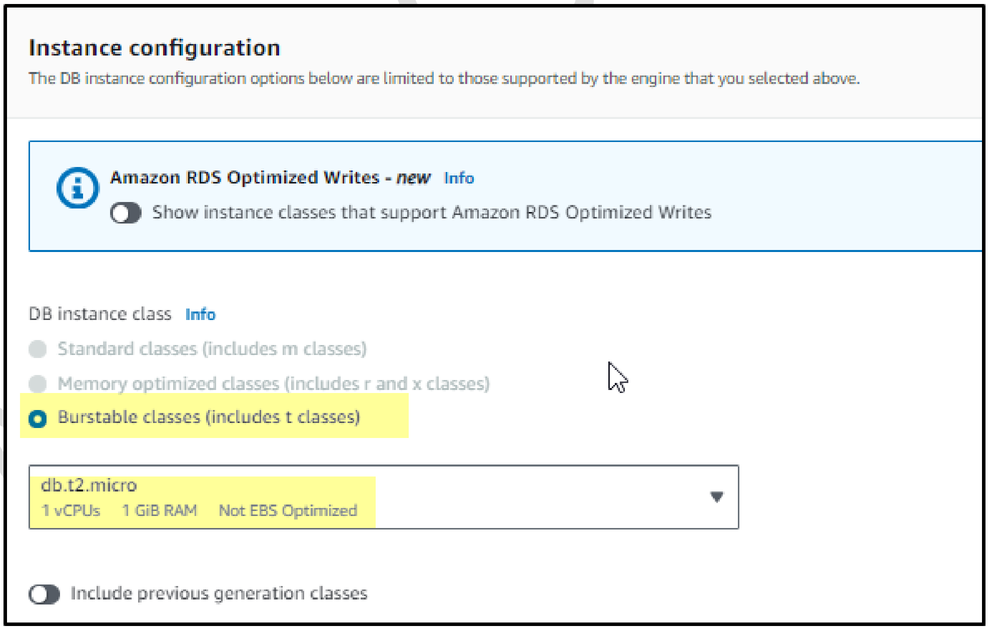
□ Auto generate a password

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



Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

ⓘ **Amazon RDS Optimized Writes - *new*** Info

Show instance classes that support Amazon RDS Optimized Writes

DB instance class Info

○ Standard classes (includes m classes)

○ Memory optimized classes (includes r and x classes)

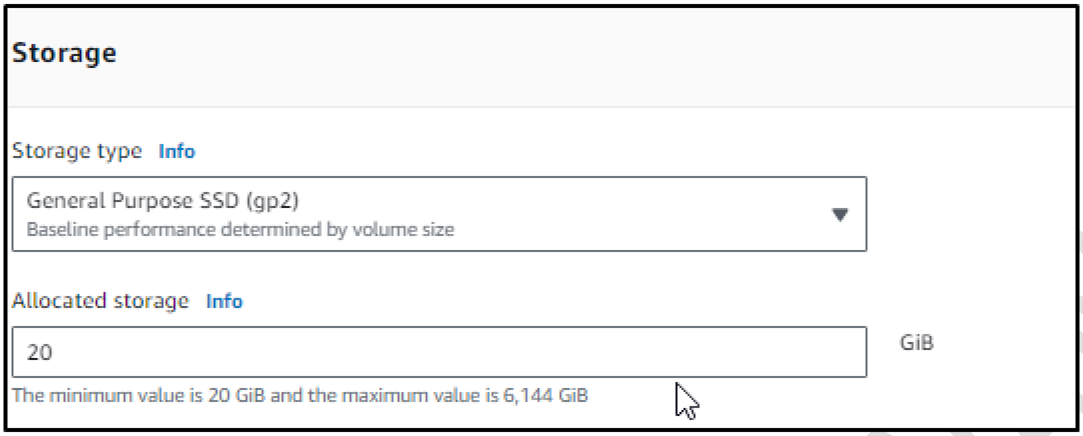
● Burstable classes (includes t classes)

db.t2.micro
1 vCPUs 1 GiB RAM Not EBS Optimized

Include previous generation classes

g) Storage – let the defaults be

Load data from MySQL in local to snowflake using RDS (and matillion)



Storage

Storage type Info

General Purpose SSD (gp2)
Baseline performance determined by volume size

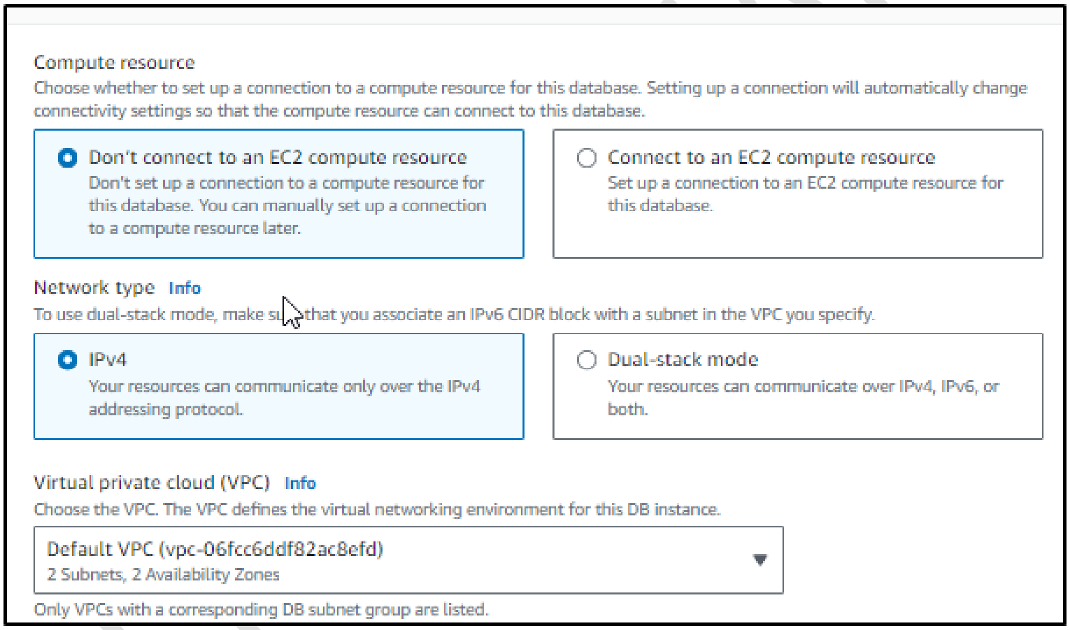
Allocated storage Info

20 GiB

The minimum value is 20 GiB and the maximum value is 6,144 GiB

h) Connectivity :

a. For compute resource and VPC - let the defaults be



Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

● Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

○ Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Network type Info

To use dual-stack mode, make su[re] 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.

Virtual private cloud (VPC) Info

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

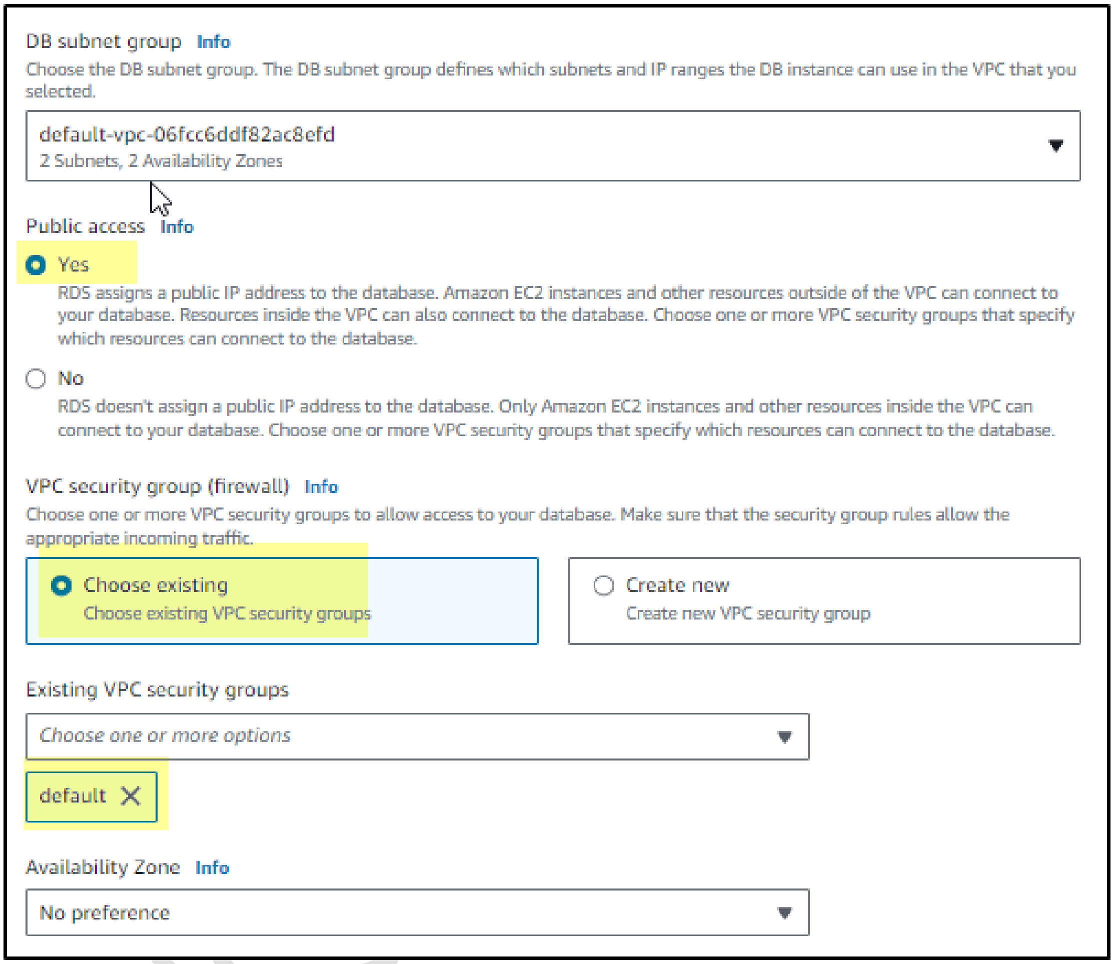
Default VPC (vpc-06fcc6ddf82ac8efd)
2 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

b. Public access:

Since RDS has to access the Mysql from our local, we need public access. Hence Public access has to be set to yes

Load data from MySQL in local to snowflake using RDS (and matillion)



DB subnet group Info

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

default-vpc-06fcc6ddf82ac8efd
2 Subnets, 2 Availability Zones

Public access Info

● Yes
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.

○ No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall) Info

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

● Choose existing
Choose existing VPC security groups

○ Create new
Create new VPC security group

Existing VPC security groups

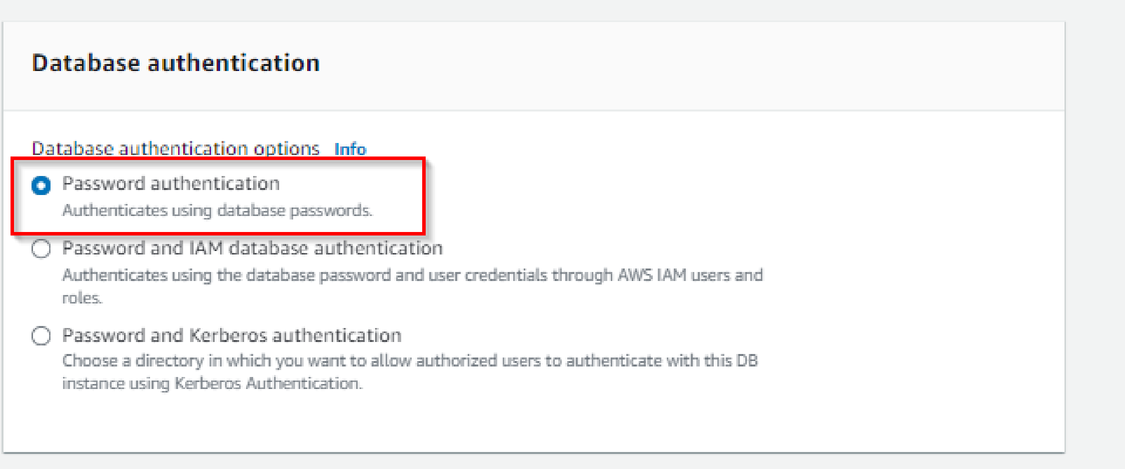
Choose one or more options

default ✕

Availability Zone Info

No preference

i) Let the database authentication remain – password authentication



Database authentication

Database authentication options Info

● Password authentication
Authenticates using database passwords.

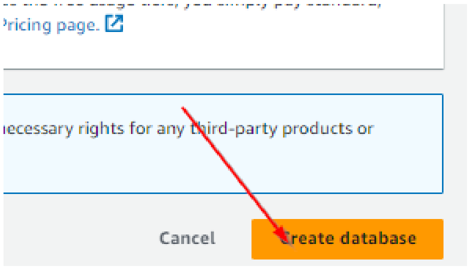
○ Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

○ Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

j) Make sure to read the monthly costs sections

Load data from MySQL in local to snowflake using RDS (and matillion)

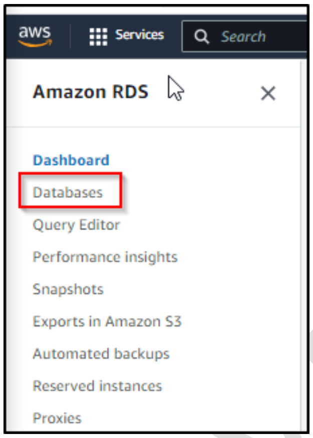
k) Click Create database

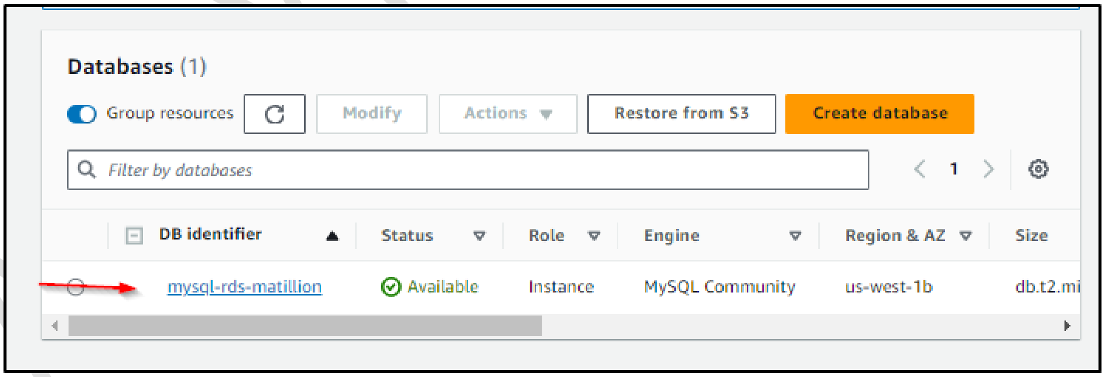


l) It may take few minutes for AWS to create RDS instance. You will be notifies that database is created successfully.

Step 1b) Configuring RDS Mysql Environment:

a) In RDS, click Databases; it will list the db instance you just created.

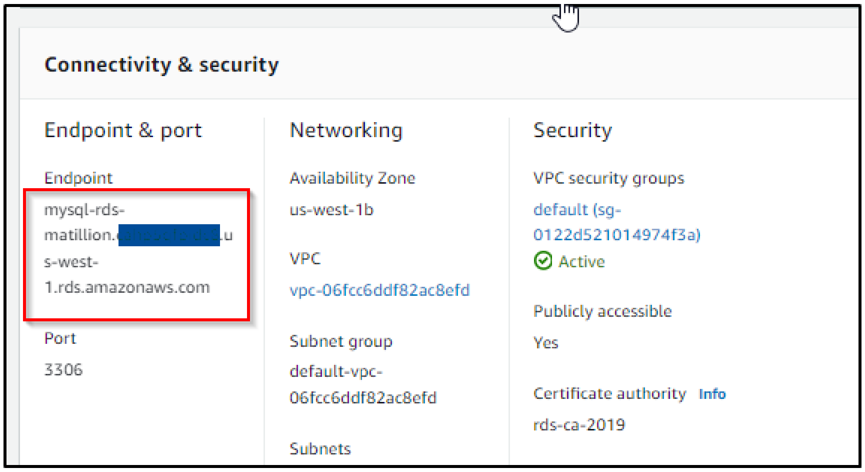




DB identifier	Status	Role	Engine	Region & AZ	Size
mysql-rds-matillion	Available	Instance	MySQL Community	us-west-1b	db.t2.mi

b) Click on the Db identifier. A new page will open . Please make note of the endpoint which will be used later when configuring MySQL using MySQL workbench.

Load data from MySQL in local to snowflake using RDS (and matillion)



Connectivity & security

Endpoint & port

Endpoint
mysql-rds-matillion.[redacted]us-west-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-west-1b

VPC
vpc-06fcc6ddf82ac8efd

Subnet group
default-vpc-06fcc6ddf82ac8efd

Subnets

Security

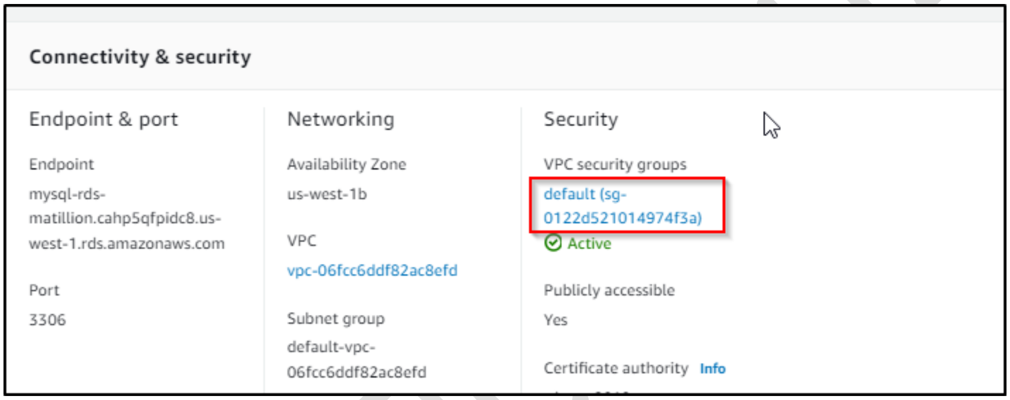
VPC security groups
default (sg-0122d521014974f3a)
✓ Active

Publicly accessible
Yes

Certificate authority Info
rds-ca-2019

c) Next step s to create a ‘inbound rule ‘, so that MySQL in local can access the RDS thru post 3306. This is the port that is used to usually to configure MySQL.

To do so, click on VPC security groups – it will open up a new page.



Connectivity & security

Endpoint & port

Endpoint
mysql-rds-matillion.cahp5qfpidc8.us-west-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-west-1b

VPC
vpc-06fcc6ddf82ac8efd

Subnet group
default-vpc-06fcc6ddf82ac8efd

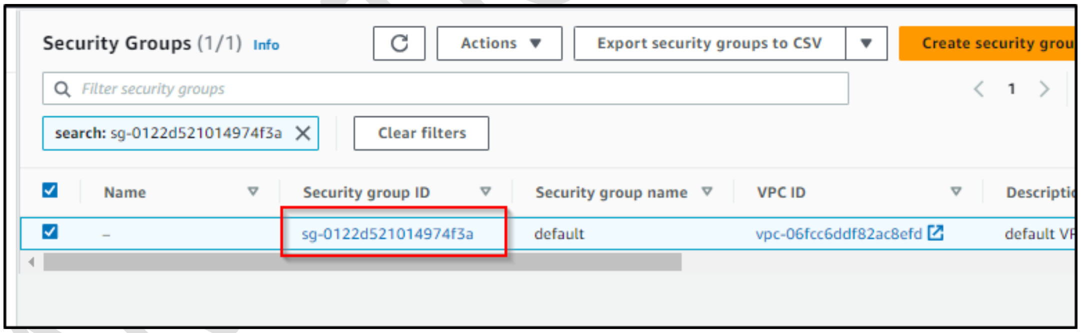
Security

VPC security groups
default (sg-0122d521014974f3a)
✓ Active

Publicly accessible
Yes

Certificate authority Info

Select the security group ID



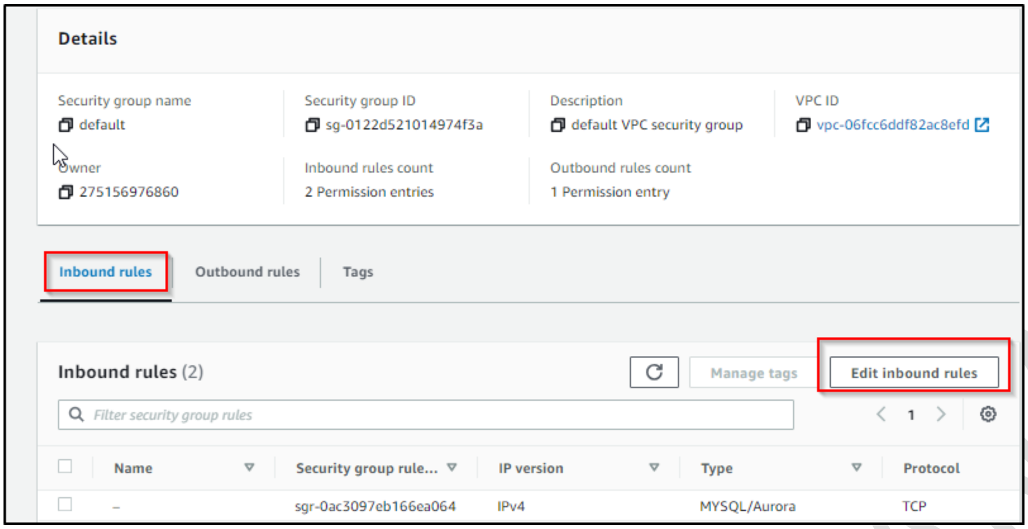
Security Groups (1/1) Info

search: sg-0122d521014974f3a ✕

☑	Name	Security group ID	Security group name	VPC ID	Descripti
☑	–	sg-0122d521014974f3a	default	vpc-06fcc6ddf82ac8efd	default VF

In this page, click on Edit inbound rules.

Load data from MySQL in local to snowflake using RDS (and matillion)



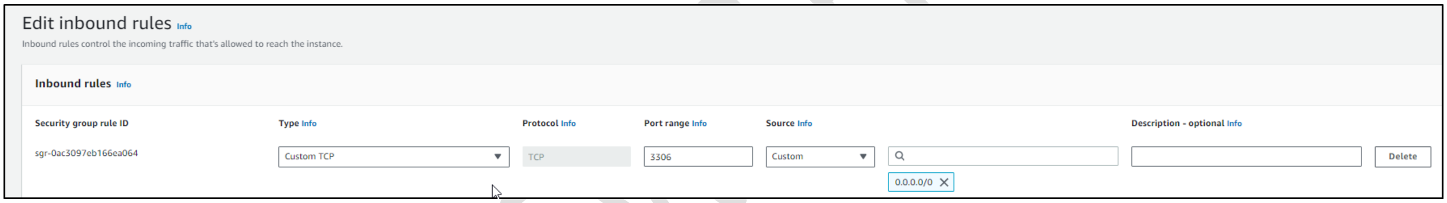
Details

Security group name	Security group ID	Description	VPC ID
default	sg-0122d521014974f3a	default VPC security group	vpc-06fcc6ddf82ac8efd
Owner	Inbound rules count	Outbound rules count	
275156976860	2 Permission entries	1 Permission entry	

Inbound rules (2)

Name	Security group rule...	IP version	Type	Protocol
–	sgr-0ac3097eb166ea064	IPv4	MYSQL/Aurora	TCP

Edit inbound rules page will appear. We need to create a custom rule. Click on 'add rule' button and select source – anywhere. Click on save rule



Edit inbound rules Info

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

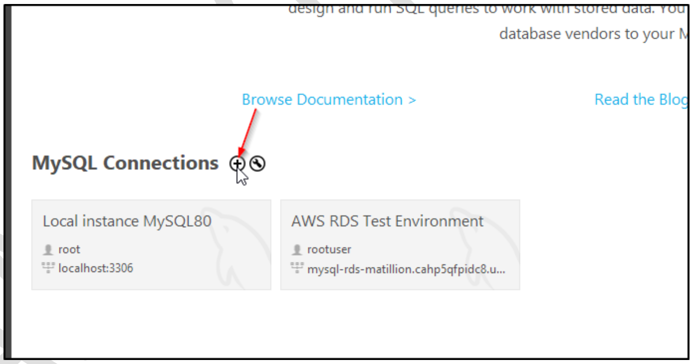
Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0ac3097eb166ea064	Custom TCP	TCP	3306	Custom	0.0.0.0/0		Delete

The inbound we just created should be listed.

d) Next , we need to connect the RDS environment using SQL Workbench

i. Go to MySQL workbench, click on + to create a new connection.



design and run SQL queries to work with stored data. You

database vendors to your M

MySQL Connections

Local instance MySQL80
root
localhost:3306

AWS RDS Test Environment
rootuser
mysql-rds-matillion.cahp5qfpidc8.u...

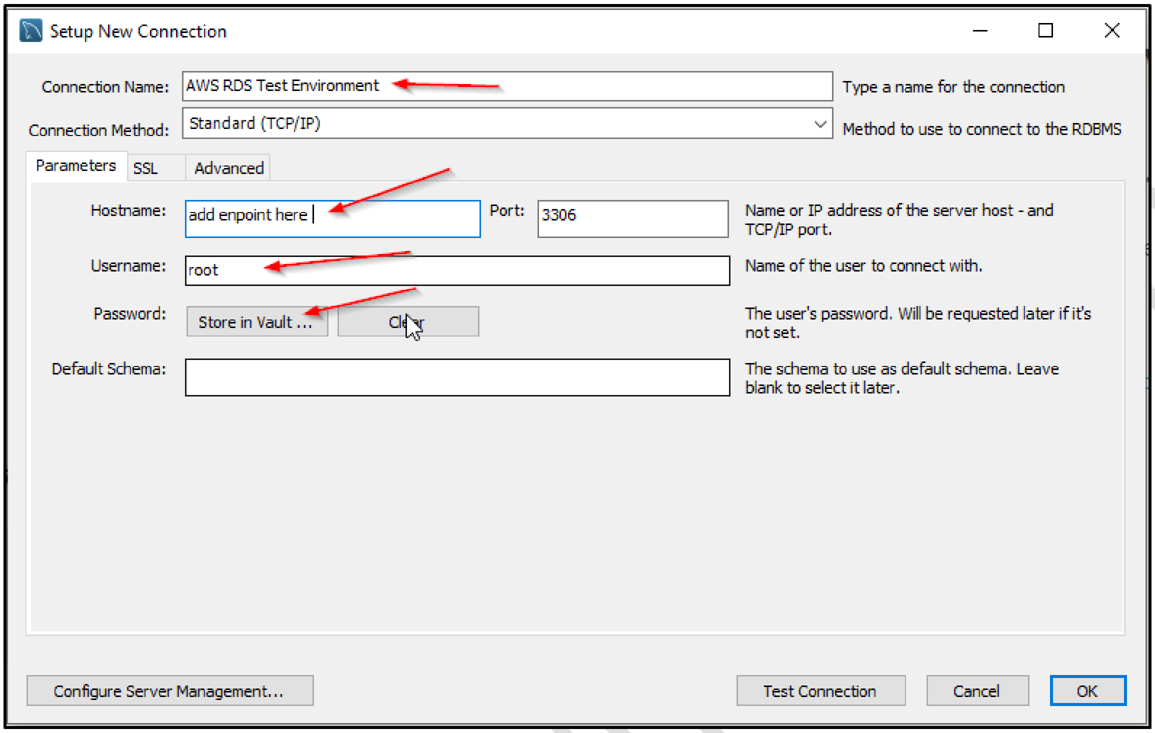
ii. Update details :

a. Name – name of your choice

b. Hostname – the end point that we copied after creating the database in RDS

c. Username & Password – credentials provided when creating the database in RDS

Load data from MySQL in local to snowflake using RDS (and matillion)



Setup New Connection

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

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

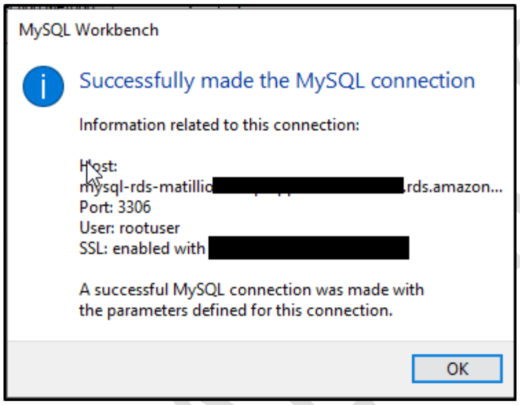
Hostname: add enpoint here — Port: 3306 — Name or IP address of the server host - and TCP/IP port.

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

Password: Store in Vault ... Clear — 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.

Once done, click on **Test connection**



MySQL Workbench

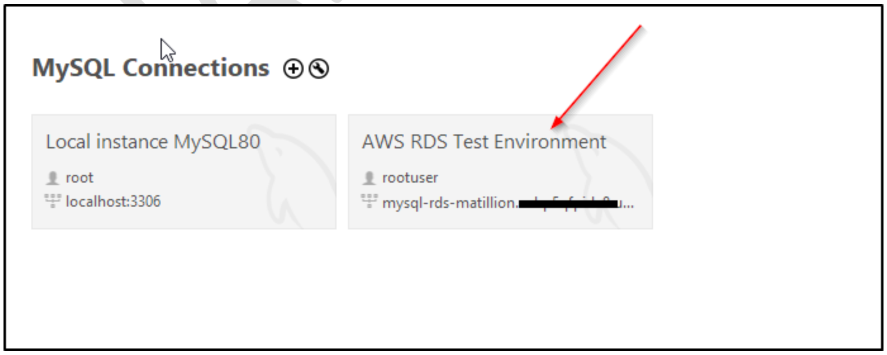
Successfully made the MySQL connection

Information related to this connection:

Host:
mysql-rds-matillio[redacted]rds.amazon...
Port: 3306
User: rootuser
SSL: enabled with [redacted]

A successful MySQL connection was made with the parameters defined for this connection.

e) Click on the new connection , you just created



MySQL Connections ⊕ ⊗

Local instance MySQL80 root localhost:3306	AWS RDS Test Environment rootuser mysql-rds-matillion.[redacted]u...

Load data from MySQL in local to snowflake using RDS (and matillion)

d) Create a database. Create a table and insert data using the below script

```
-- create database

        CREATE DATABASE `HOSPITAL`;

        -- create table
        CREATE TABLE `RN_HOSPITAL_TABLE` (
          `PATIENT_ID` int DEFAULT NULL,
          `PATIENT_NAME` varchar(30) DEFAULT NULL,
          `BILLING_ADDRESS` varchar(50) DEFAULT NULL,
          `DIAGNOSIS` varchar(20) DEFAULT NULL,
          `TREATMENT` varchar(50) DEFAULT NULL,
          `COST` float DEFAULT NULL
        ) ;

        -- insert records
        insert into RN_hospital_table
         (patient_id, patient_name, billing_address, diagnosis, treatment, cost)
         values
         (1, 'Mark Knopfler', '1982 Telegraph Road', 'Industrial Disease', 'a week of peace and quiet', 2000.00),
         (2, 'Guido van Rossum', '37 Florida St.', 'python bite', 'anti-venom', 70000.00),
         (3, 'Devin', '197 Brigade Road Texas', 'dog bite', 'Rabies Injection', 40000.00),
         (4, 'Mark', '38 denver St Chicago', 'Dengue', 'Malaria', 50000.00),
         (5, 'Peter', '78 New Yor City', 'Accident', 'Operation', 340000.00);
```

NOTE: Please use your initials for table creation

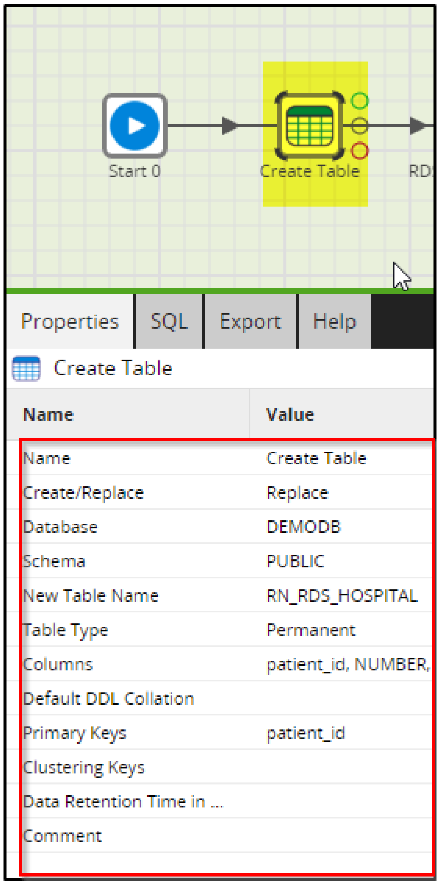
STEP 2 a) in matillion: - Prepare snowflake – create a table

a) Load matillion

b) Create an orchestration job

c) Use ‘create table’ component – to create table in snowflake where data will be loaded.

Load data from MySQL in local to snowflake using RDS (and matillion)



Create Table

Name	Value
Name	Create Table
Create/Replace	Replace
Database	DEMODB
Schema	PUBLIC
New Table Name	RN_RDS_HOSPITAL
Table Type	Permanent
Columns	patient_id, NUMBER,
Default DDL Collation	
Primary Keys	patient_id
Clustering Keys	
Data Retention Time in ...	
Comment	

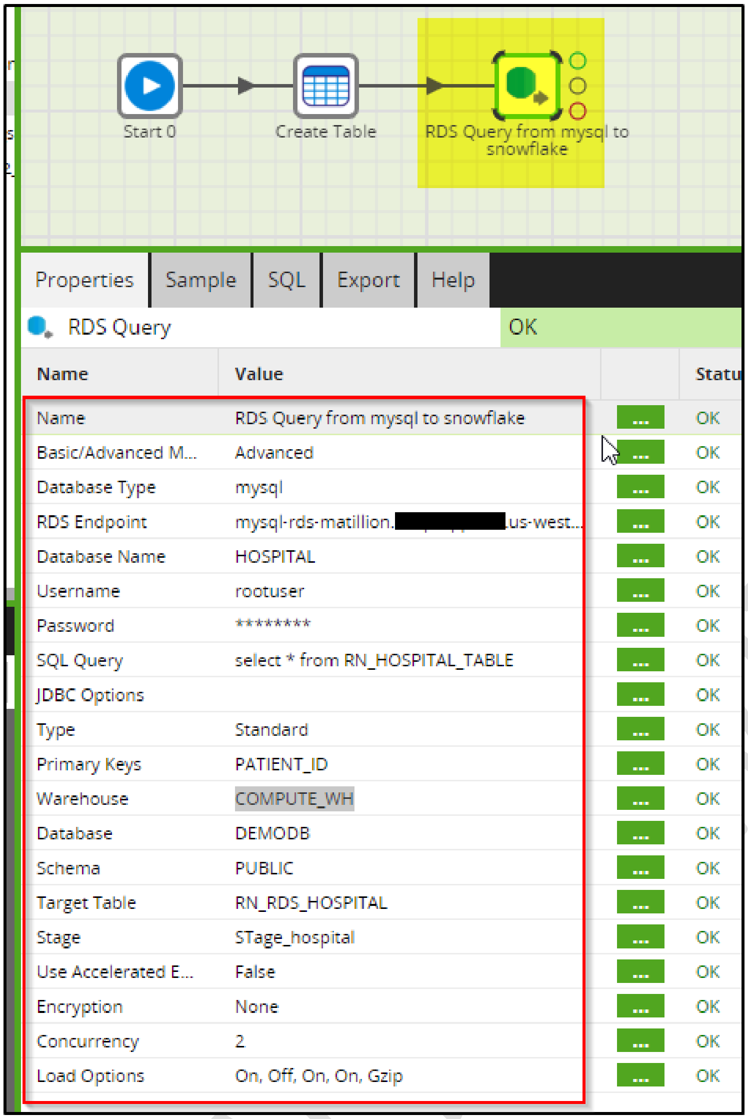
STEP 2 b) in matillion: - Use RDS component to load into snowflake

Drag and Drop ‘RDS load’ component and fill in the properties.

Please refer the image, Below I have explained on few properties

Property	
Name	Give a name of your choice
Database Type	mysql
RDS Endpoint	This is the endpoint that we copied after creation of RDS db.
Database Name	Db name of source (here MySQL)
Username	Credential provided @ RDS
Password	Credential provided @ RDS
SQL Query	select * from RN_HOSPITAL_TABLE
Warehouse	Snowflake details
Database	
Schema Target Table	
Stage	AWS S3 stage (create a new one or point to excising one)

Load data from MySQL in local to snowflake using RDS (and matillion)



RDS Query OK

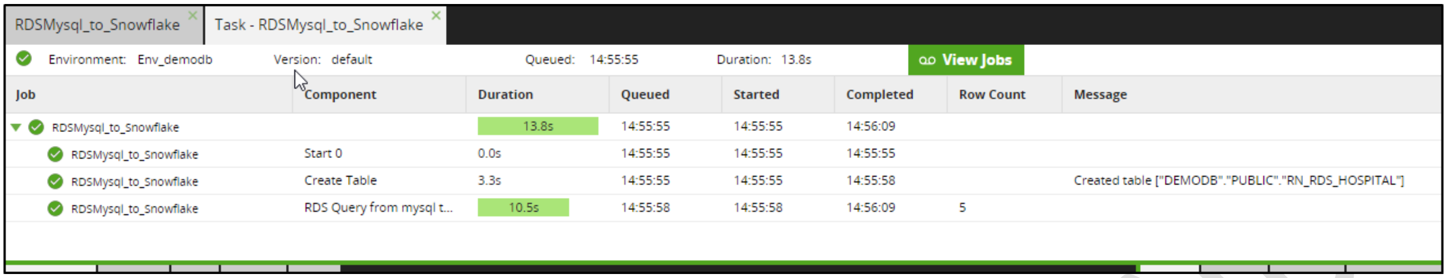
Name	Value		Statu
Name	RDS Query from mysql to snowflake	...	OK
Basic/Advanced M...	Advanced	...	OK
Database Type	mysql	...	OK
RDS Endpoint	mysql-rds-matillion.[redacted]us-west...	...	OK
Database Name	HOSPITAL	...	OK
Username	rootuser	...	OK
Password	********	...	OK
SQL Query	select * from RN_HOSPITAL_TABLE	...	OK
JDBC Options		...	OK
Type	Standard	...	OK
Primary Keys	PATIENT_ID	...	OK
Warehouse	COMPUTE_WH	...	OK
Database	DEMODB	...	OK
Schema	PUBLIC	...	OK
Target Table	RN_RDS_HOSPITAL	...	OK
Stage	STage_hospital	...	OK
Use Accelerated E...	False	...	OK
Encryption	None	...	OK
Concurrency	2	...	OK
Load Options	On, Off, On, On, Gzip	...	OK

STEP 3: Run Job and verify results:

a) Run the job

b) Verifying successful job completion in matillion

Load data from MySQL in local to snowflake using RDS (and matillion)

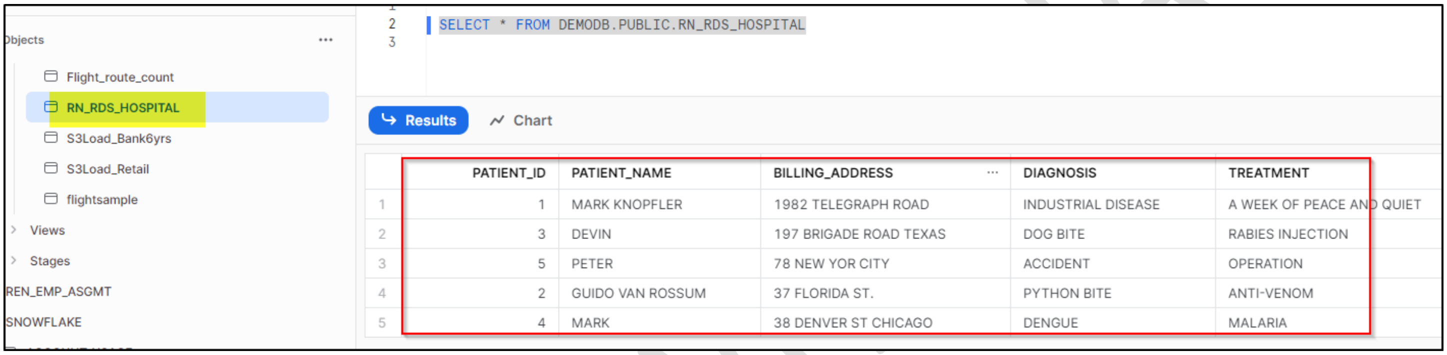


Environment: Env_demodb Version: default Queued: 14:55:55 Duration: 13.8s

Job	Component	Duration	Queued	Started	Completed	Row Count	Message
RDSMysql_to_Snowflake		13.8s	14:55:55	14:55:55	14:56:09		
RDSMysql_to_Snowflake	Start 0	0.0s	14:55:55	14:55:55	14:55:55		
RDSMysql_to_Snowflake	Create Table	3.3s	14:55:55	14:55:55	14:55:58		Created table ["DEMODB"."PUBLIC"."RN_RDS_HOSPITAL"]
RDSMysql_to_Snowflake	RDS Query from mysql t...	10.5s	14:55:58	14:55:58	14:56:09	5	

c) Verifying if data is actually created. This can be done in 2 ways:

a. Checking it in snowflake directly :



```
SELECT * FROM DEMODB.PUBLIC.RN_RDS_HOSPITAL
```

	PATIENT_ID	PATIENT_NAME	BILLING_ADDRESS	DIAGNOSIS	TREATMENT
1	1	MARK KNOPFLER	1982 TELEGRAPH ROAD	INDUSTRIAL DISEASE	A WEEK OF PEACE AND QUIET
2	3	DEVIN	197 BRIGADE ROAD TEXAS	DOG BITE	RABIES INJECTION
3	5	PETER	78 NEW YOR CITY	ACCIDENT	OPERATION
4	2	GUIDO VAN ROSSUM	37 FLORIDA ST.	PYTHON BITE	ANTI-VENOM
5	4	MARK	38 DENVER ST CHICAGO	DENGUE	MALARIA

b. Create a transformation job , use 'table input' component and verify @ sample tab.