

## Creation of a RedShift Cluster

1. Redshift cluster named 'spark-redshift-etl' is created using dc2.large as Node Type.
2. No. of nodes selected are 2.
3. A parameter group is created for workload management for all the queries of spark\_redshift\_etl\_group and default queries from other group.

### Screenshots of the configuration of the RedShift cluster that you have created:

Services ▾
 [Alt+S]
upgradshardulrajhans @ 9902-9864-3140 ▾
N. Virginia ▾
Support ▾

Amazon Redshift > Clusters > spark-redshift-etl

**spark-redshift-etl**
Actions ▾
Edit
Add partner integration
Query cluster

**General information**

Cluster identifier spark-redshift-etl	Status Paused	Node type dc2.large	Endpoint spark-redshift-etl.czaviv6dqd8x.us-east-1.red...
Cluster namespace 10eb3589-040a-473b-b9f3-543296a94a42	Date created April 28, 2021, 10:13(UTC+05:30)	Number of nodes 2	JDBC URL jdbc:redshift://spark-redshift-etl.czaviv6dqd8...
	Storage used -	AQUA Not available	ODBC URL Driver={Amazon Redshift (x64)}; Server=spar...

**Database configurations**
Change admin user password
Rotate encryption keys
Edit ▾

Database name masterdb	Parameter group Defines database parameter and query queues for all the databases. spark-redshift-etl-paratemer-group	Encryption Disabled AWS KMS key ID -	Audit logging Disabled
Port 5439	SSH ingestion setting (cluster public key) ssh-rsa AAAAB3NzaC1yc2EAAAADAQAB...		
Admin user name awsuser			

**Network and security settings**
Edit publicly accessible
Edit

Virtual private cloud (VPC) vpc-0e29ace62785321c1	Availability Zone us-east-1d	VPC security group Specify which instances and devices can connect to the cluster. sg-03d4dc927be5b94b9	Publicly accessible Allow instances and devices outside the VPC to connect to your database through the cluster endpoint. Disabled
Subnet redshift-cluster	Enhanced VPC routing Disabled		
Endpoint URL -			

Setting up a database in the RedShift cluster and running queries to create the dimension and fact tables

**Queries to create the various dimension and fact tables with appropriate primary and foreign keys:**

1. **Creating the group and adding the default user in it for the workload management for better performance of the queries.**

```
create group spark_redshift_etl_group with user awsuser;
```

2. **Creating the schema for the user to store all the tables and related information.**

```
create schema if not exists etl_bank_schema;
```

3. **Creating 'dim\_location' table with 'location\_id' as primary key. 'location\_id' is also used as the distkey and sortkey for better performance.**

```
create table if not exists etl_bank_schema.dim_location(  
  location_id varchar(50) not null distkey sortkey primary key,  
  location varchar(50),  
  streetname varchar(255),  
  street_number integer,  
  zipcode integer,  
  lat decimal(10,3),  
  lon decimal(10,3)  
);
```

4. Creating 'dim\_card\_type' table with 'card\_type\_id' as primary key. 'card\_type\_id' is also used as sortkey and distkey for better performance and tuning.

```
create table if not exists etl_bank_schema.dim_card_type(  
    card_type_id varchar(50) not null distkey sortkey primary key,  
    card_type varchar(40)  
);
```

5. Creating 'dim\_date' table with 'date\_id' as primary key. 'date\_id' is also used as sortkey and distkey for better performance and tuning.

```
create table if not exists etl_bank_schema.dim_date(  
    date_id varchar(50) not null distkey sortkey primary key,  
    full_date_time timestamp,  
    year integer,  
    month varchar(20),  
    day integer,  
    hour integer,  
    weekday varchar(20)  
);
```

6. Creating 'dim\_atm' table with 'atm\_id' as primary key. 'atm\_location\_id' is used as the foreign key from 'dim\_location' table for 'location\_id' column. 'atm\_id' is also used as sortkey and distkey for better performance and tuning.

```
create table if not exists etl_bank_schema.dim_atm(  
    atm_id varchar(50) not null distkey sortkey primary key,  
    atm_number varchar(20),  
    atm_manufacturer varchar(50),  
    atm_location_id varchar(50) references etl_bank_schema.dim_location(location_id)  
);
```

7. Creating 'fact\_atm\_trans' table with 'trans\_id' as primary key. 'atm\_id', 'weather\_location\_id', 'date\_id', 'card\_type\_id' are used as the foreign keys from above corresponding defined dimension tables. 'trans\_id' is also used as sortkey and distkey for better performance and tuning.

```
create table if not exists etl_bank_schema.fact_atm_trans(  
    trans_id varchar(50) not null distkey sortkey primary key,  
    atm_id varchar(50) references etl_bank_schema.dim_atm(atm_id),  
    weather_loc_id varchar(50) references etl_bank_schema.dim_location(location_id),  
    date_id varchar(50) references etl_bank_schema.dim_date(date_id),  
    card_type_id varchar(50) references  
    etl_bank_schema.dim_card_type(card_type_id),  
    atm_status varchar(20),  
    currency varchar(10),  
    service varchar(20),  
    transaction_amount integer,  
    message_code varchar(255),  
    message_text varchar(255),  
    rain_3h decimal(10,3),  
    clouds_all integer,  
    weather_id integer,  
    weather_main varchar(50),  
    weather_description varchar(255)  
);
```

## Loading data into a RedShift cluster from Amazon S3 bucket

Queries to copy the data from S3 buckets to the RedShift cluster in the appropriate tables

1. Copying 'DIM\_LOCATION' table from S3 location stored in the CSV format.

```
copy etl_bank_schema.dim_location from
's3://shardul-etl-bank-data/dim_location/dim_location.csv'
iam_role 'arn:aws:iam::990298643140:role/redshift-s3-full-access'
csv region 'us-east-1';
```

2. Copying 'DIM\_ATM' table from S3 location stored in the CSV format.

```
copy etl_bank_schema.dim_atm from
's3://shardul-etl-bank-data/dim_atm/dim_atm.csv'
iam_role 'arn:aws:iam::990298643140:role/redshift-s3-full-access'
csv region 'us-east-1';
```

3. Copying 'DIM\_CARD\_TYPE' table from S3 location stored in the CSV format.

```
copy etl_bank_schema.dim_card_type from
's3://shardul-etl-bank-data/dim_card_type/dim_card_type.csv'
iam_role 'arn:aws:iam::990298643140:role/redshift-s3-full-access'
acceptinvchars csv region 'us-east-1';
```

4. Copying 'DIM\_DATE' table from S3 location stored in the Parquet format.

```
copy etl_bank_schema.dim_date from
's3://shardul-etl-bank-data/dim_date/dim_date.parquet'
iam_role 'arn:aws:iam::990298643140:role/redshift-s3-full-access'
format as parquet;
```

5. Copying 'FACT\_ATM\_TRANS' table from S3 location stored in the CSV format

```
copy etl_bank_schema.fact_atm_trans from  
's3://shardul-etl-bank-data/fact_transaction/fact_transaction.csv'  
iam_role 'arn:aws:iam::990298643140:role/redshift-s3-full-access'  
csv region 'us-east-1';
```