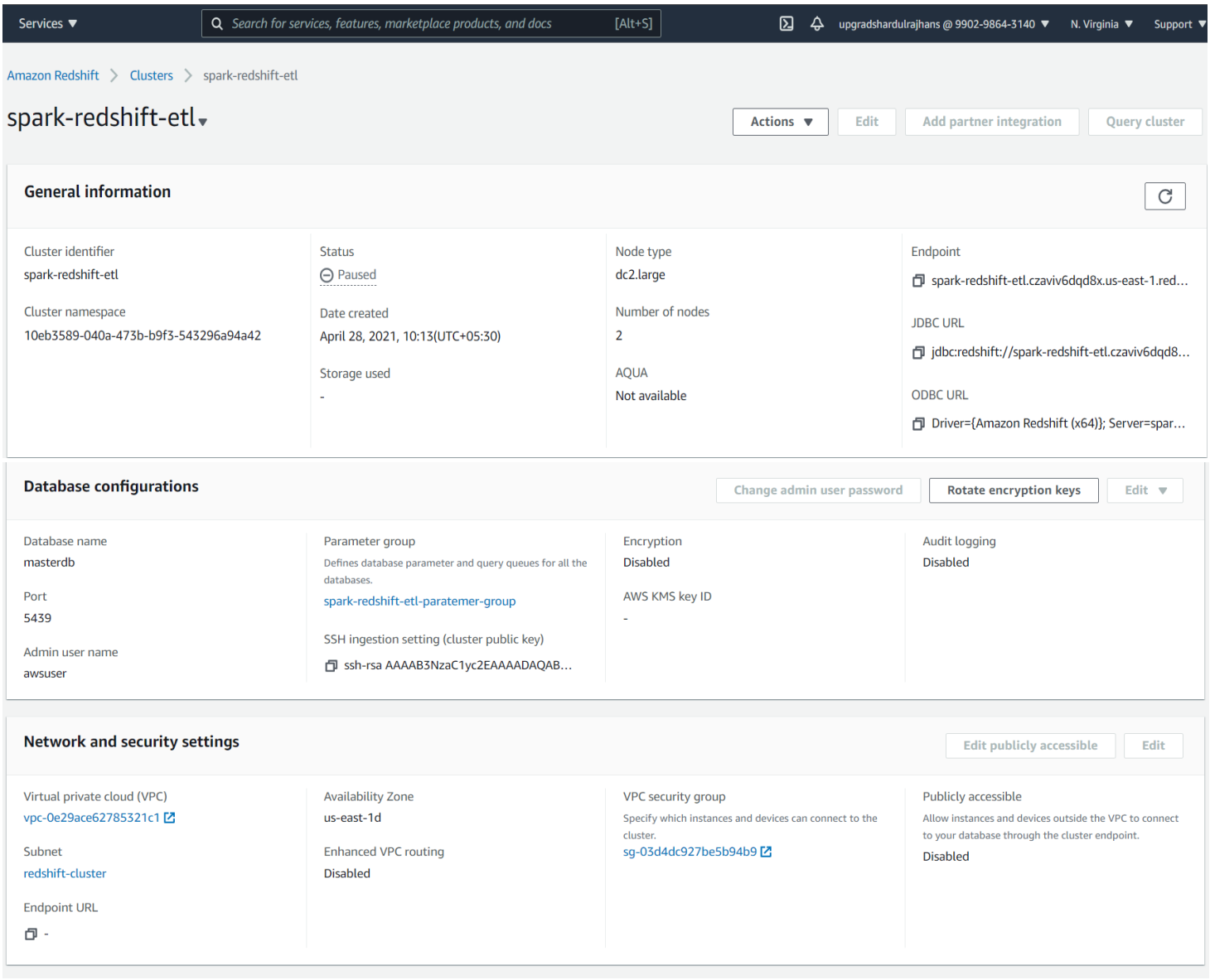
Creation of a RedShift Cluster

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:



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 foriegn 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';
```