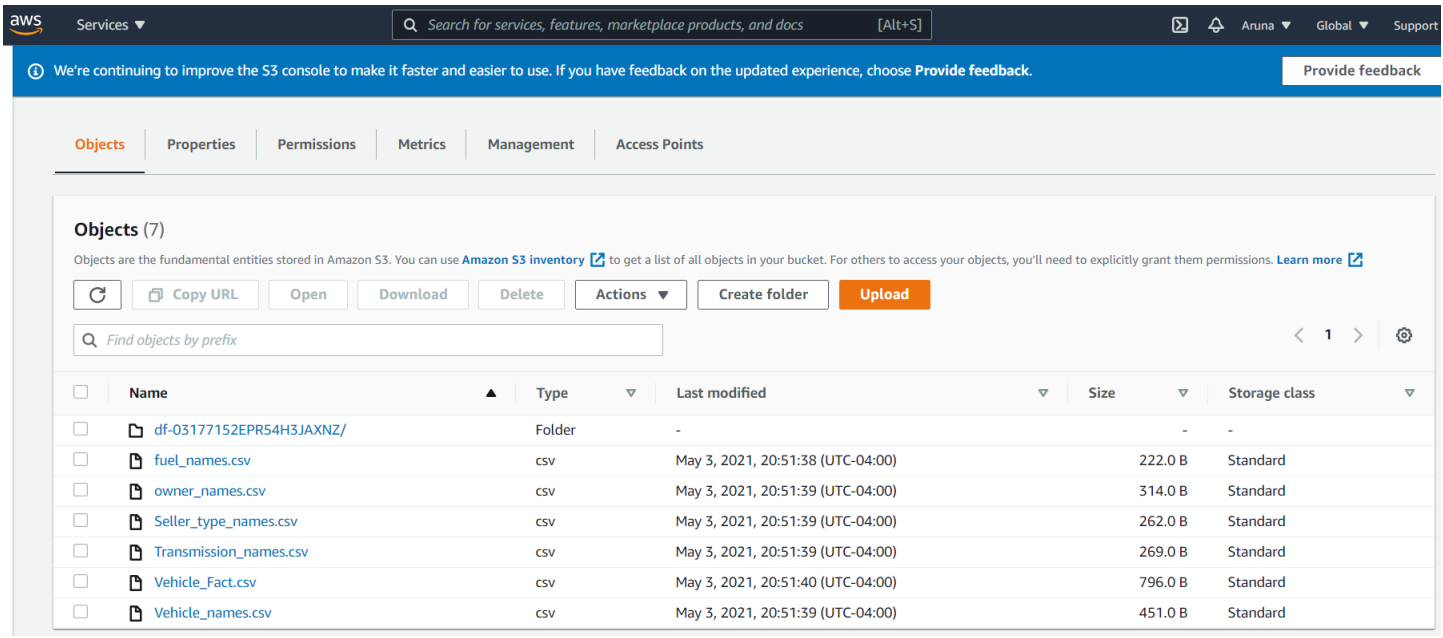
Datawarehouse Engineer take home

With the given problem statement, I have taken Vehicles dataset from Kaggle and decided to Use **Amazon S3** as **Storage layer** and **Amazon redshift** for **Datawarehouse layer** and Amazon Redshift can be connected to **Tableau** for **Reporting Purpose**.

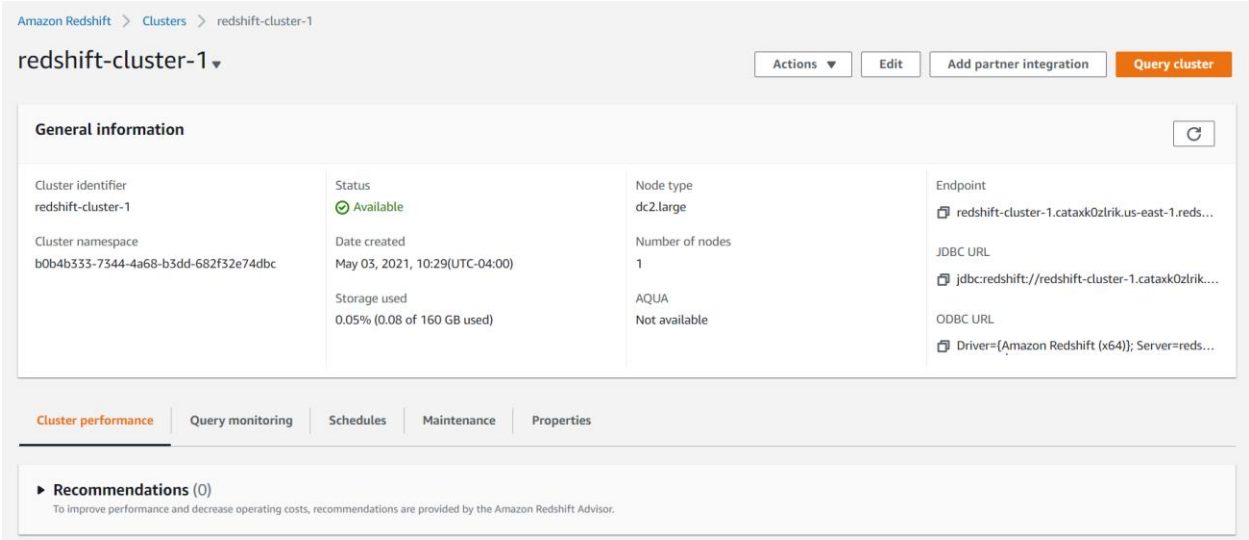
And for scheduling this entire flow I used **Amazon Data Pipeline**.

I separated the data from Vehicles dataset (source : https://www.kaggle.com/nehalbirla/vehicle-dataset-from-cardekho)into Fact Table and Dimension Tables and uploaded to Amazon S3 as shown below.

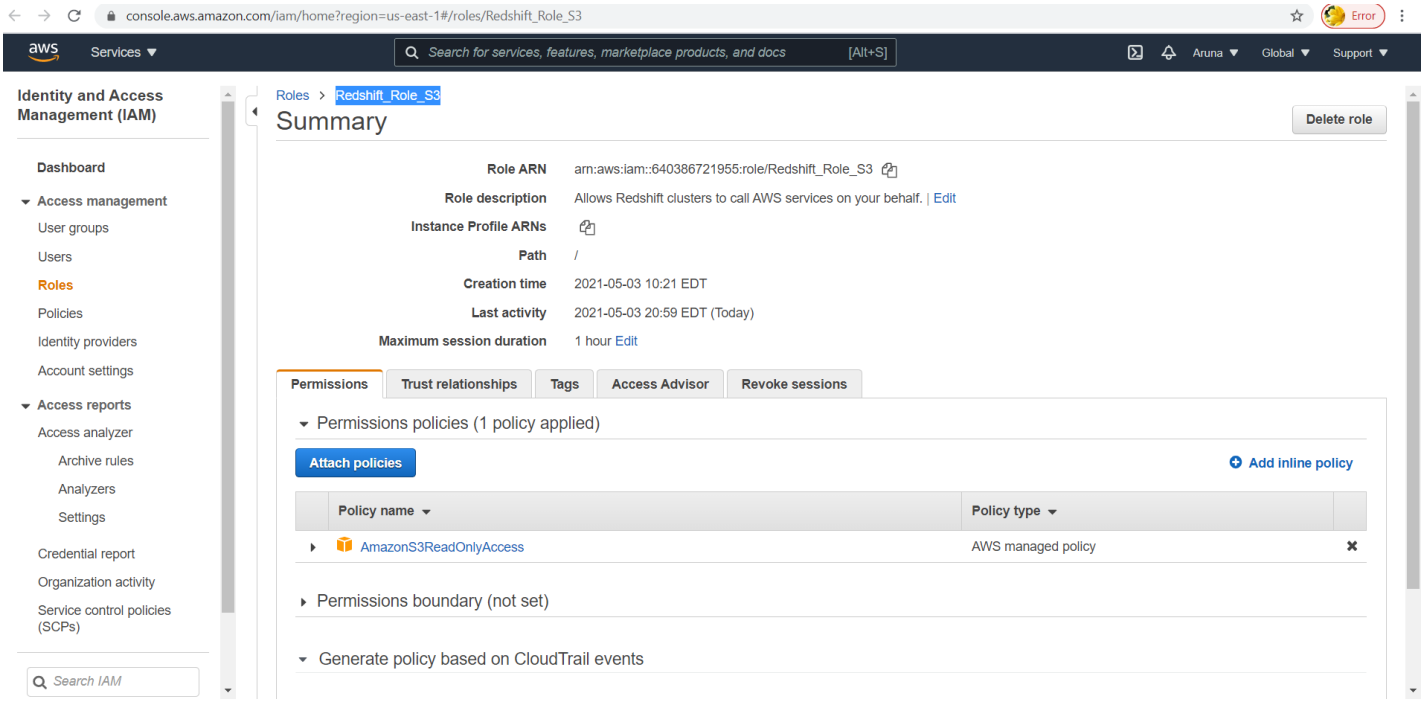
The list of tables are as follows



From the Amazon S3 as the input layer, I created a cluster in Amazon Redshift **redshift-cluster-1** with Node type as dc2.large and Number of nodes as 1.



Created a IAM Cluster role Redshift_Role_S3 which contains AmazonS3ReadOnlyAccess



And I associated this IAM role to the cluster I created.

After creating the cluster in Redshift, I started creating Tables in Redshift with the same structure of the files that I loaded in S3 and I loaded data into tables using the copy command.

While loading the data to tables using copy command , I gave the input path as S3 Bucket and also given Aws_Iam_Role and ignored the header

Tables that I created: one Fact Table and 5 Dimension Tables:

```
create table if not exists vehicle_fact(Vehicle_id nvarchar(100),
fuel_id nvarchar(20),
seller_type_id nvarchar(20),
transmission_id nvarchar(20),
owner_id nvarchar(50),
year nvarchar(10),
selling_price decimal(20,2),
km_driven decimal(20,2));
```

==========

```
copy vehicle_fact from 's3://redshiftfilestorage/Vehicle_Fact.csv'
credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'
delimiter ',' region 'us-east-1'
IGNOREHEADER 1;
```

==========

```
create table if not exists Vehicle_names(Vehicle_id nvarchar(10),
Vehicle_name nvarchar(200));
```

==========

```
copy Vehicle_names from 's3://redshiftfilestorage/Vehicle_names.csv'
credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'
delimiter ',' region 'us-east-1'
IGNOREHEADER 1;
```

==========

```
create table if not exists Transmission_names(transmission_id
 nvarchar(10),
transmission_name nvarchar(200));
```

==========

```
copy Transmission_names from 's3://redshiftfilestorage/Transmission_names.csv'
credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'
delimiter ',' region 'us-east-1'
IGNOREHEADER 1;
```

==========

```
create table if not exists Seller_type_names(seller_type_id nvarchar(10),
seller_type_names nvarchar(200));
```

==========

```
copy Seller_type_names from 's3://redshiftfilestorage/Seller_type_names.csv'
credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'
delimiter ',' region 'us-east-1'
IGNOREHEADER 1;
```

==========

```
create table if not exists owner_names(owner_id nvarchar(10),
owner_name nvarchar(200));
```

==========

```
copy owner_names from 's3://redshiftfilestorage/owner_names.csv'
credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'
delimiter ',' region 'us-east-1'
IGNOREHEADER 1;
```

==========

```
create table if not exists fuel_names(fuel_id nvarchar(10),
```

fuel_names nvarchar(200));

==========

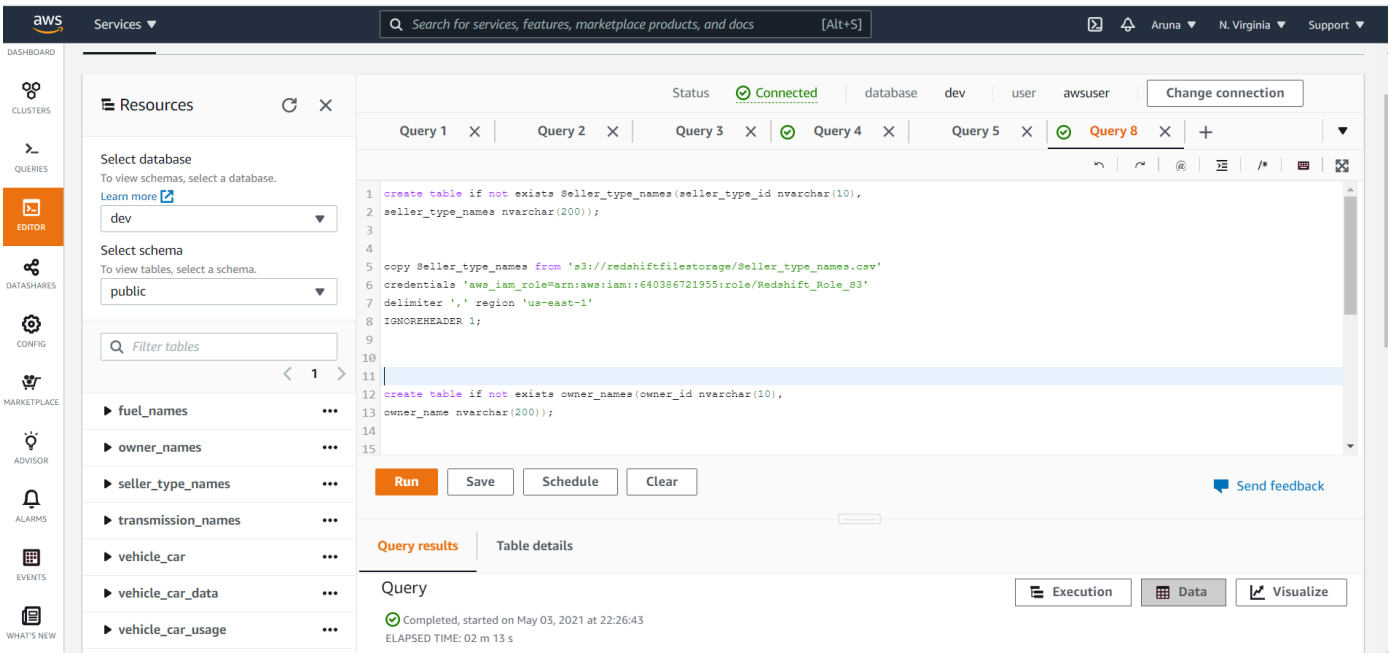
copy fuel_names from 's3://redshiftfilestorage/fuel_names.csv'

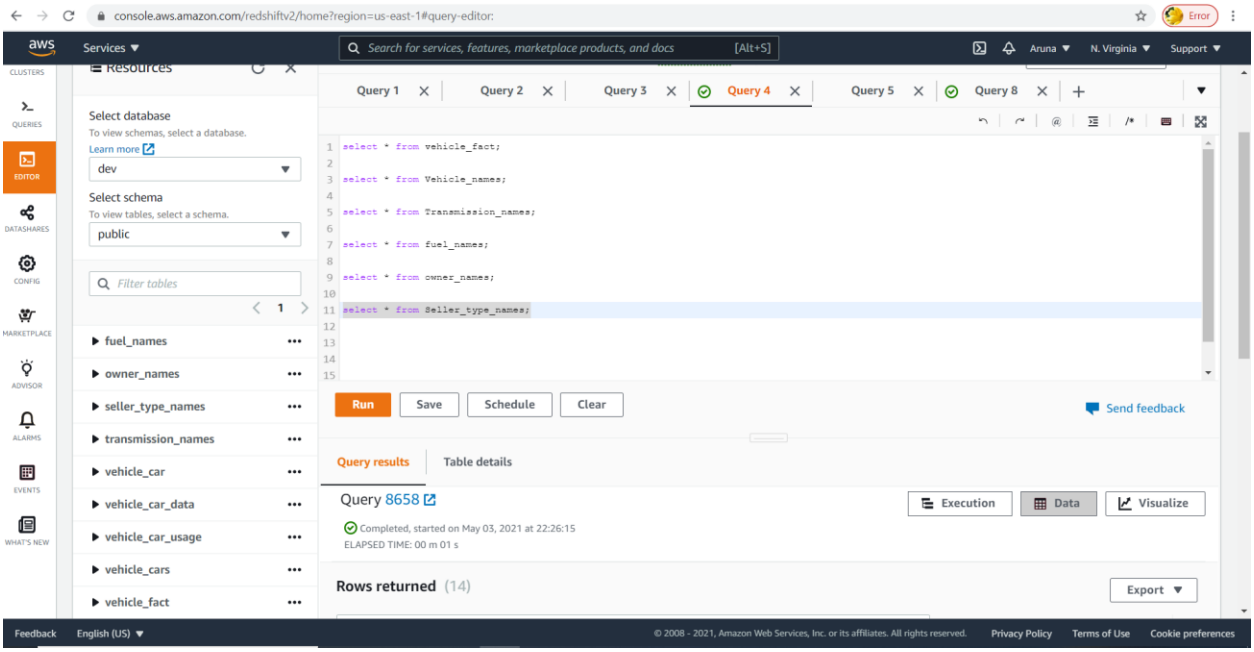
credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'

delimiter ',' region 'us-east-1'

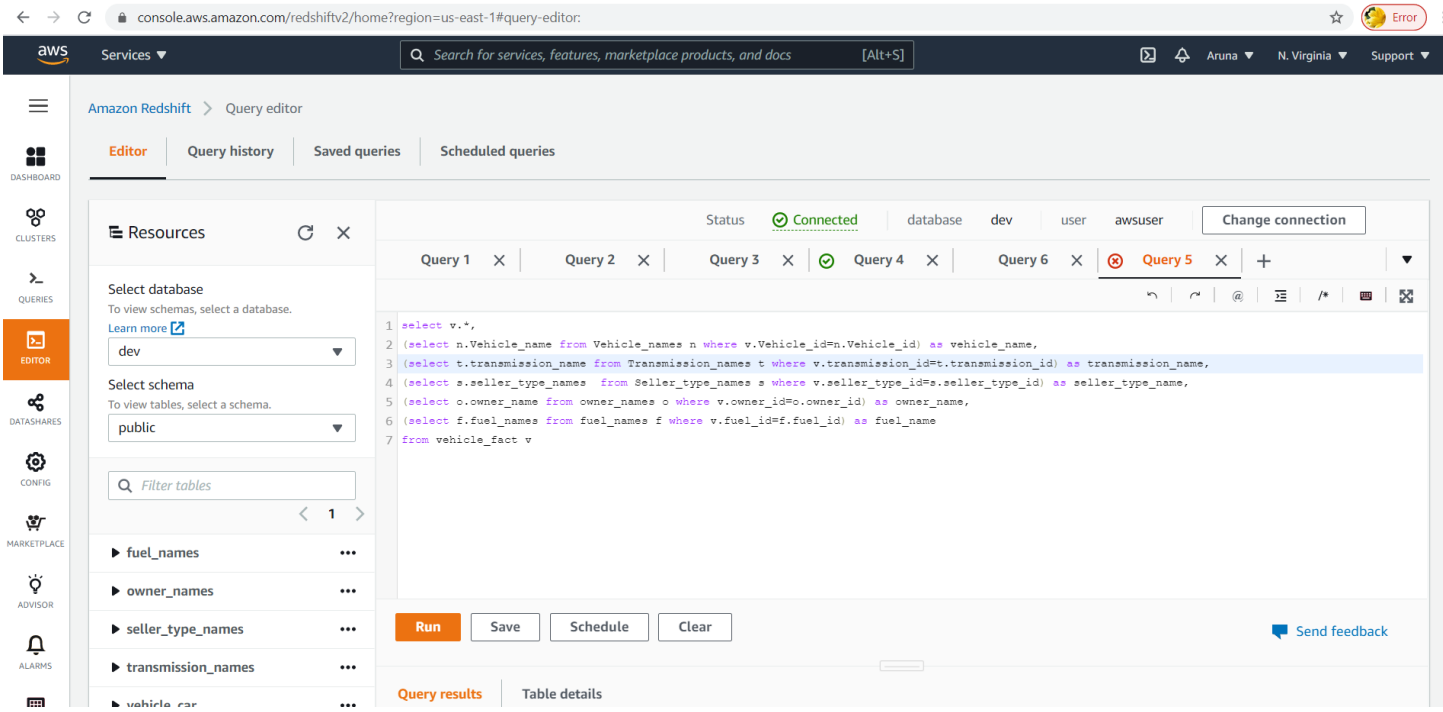
IGNOREHEADER 1;

==========





Using the Star Schema concept, I joined the fact table to the dimension tables to get the dimension names.



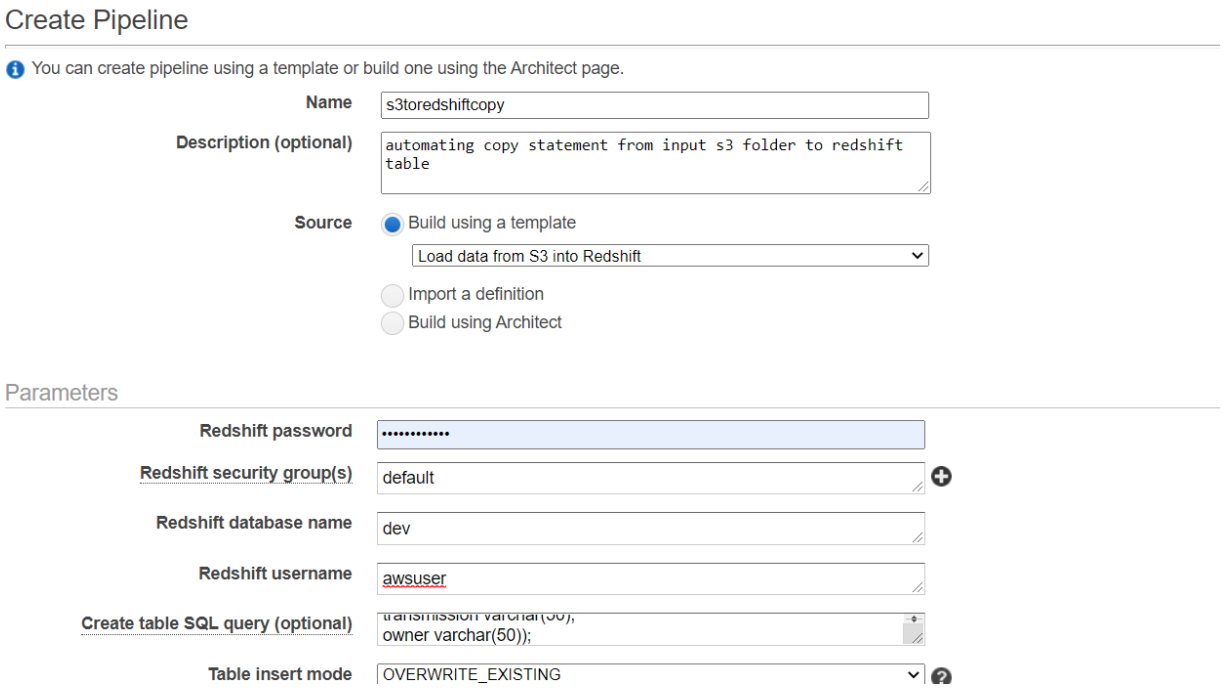
Now, this data after Dimensional Modelling is available for reporting.

Data Pipeline scheduling:

==========

I scheduled this entire data pipeline that is loading the data from Amazon s3 to Amazon redshift using the copy statement daily using Amazon Data Pipeline.

I created a pipeline called s3toredshiftcopy which takes the data from S3 Bucket and load into Redshift tables everyday.



Redshift table name vehicle_car_usage

Input S3 folder s3://redshiftfilestorage/

Redshift JDBC connection string jdbc:redshift://redshift-cluster-1.cataxk0zlnk.us-east-1.redshift.amazonaws.com:5439/dev

Copy options (optional)
- delimiter ','
- region 'us-east-1'
- IGNOREHEADER 1

Primary keys (optional)

Schedule

You can run your pipeline once or specify a schedule. More

Run
- on pipeline activation
- on a schedule

Pipeline Configuration

Logging
- Enabled
- Disabled

Copy execution logs to S3. More

S3 location for logs
s3://redshiftfilestorage/

Pipeline Configuration

Logging
- Enabled
- Disabled

Copy execution logs to S3. More

S3 location for logs
s3://redshiftfilestorage/

Security/Access

IAM roles
- Default
- Custom

IAM Roles let you control permissions for AWS Data Pipeline and your EC2 applications. More

Tags

Add up to 10 tags to your pipeline. These tags will be applied to the pipeline as well as any resources created by the pipeline. A tag consists of a case-sensitive key-value pair. Learn more

Key	Value (Optional)
Add key to create	

This pipeline launches an Amazon EC2 instance (t1.micro) in your account on every scheduled execution of the pipeline. Normal service charges for this resource will apply in addition to charges for other AWS services used by this pipeline.

Cancel | Edit in Architect | Activate

How would you rate your experience with this service console?

aws Services Search for services, features, marketplace products, and docs [Alt+S] Aruna N. Virginia Support

Data Pipeline > List Pipelines > Execution Details: s3toredshiftcopy (df-03177152EPR54H3JAXNZ) Data Pipeline Help

Edit Pipeline | Rerun | Cancel | Mark Finished

Show all components in any state with Schedule Interval between 1821-05-03 16:58:24 UTC and 2021-05-03 16:58:24 UTC Apply

Filter: Activities Filter instances ... 1 instances (all loaded)

Component Name	Schedule Interval (UTC)	Type	Status	Execution Start (UTC)	Execution End (UTC)	Attempt
RedshiftLoadActivity	2021-05-03 16:58:22 – 2021-05-03 16:58:22	RedshiftCopyActivity	WAITING_FOR_RUNNER	2021-05-03 16:58:25	–	1 of 3