

Exercise – Loading Data from S3 to Redshift

Step 1: Create a free tier Redshift cluster with a sample database.

Amazon Redshift > Clusters > Create cluster

Create cluster [Info](#)

Cluster configuration

Cluster identifier
This is the unique key that identifies a cluster.

The identifier must be from 1-63 characters. Valid characters are a-z (lowercase only) and - (hyphen).

What are you planning to use this cluster for?

☐ **Production**
Configure for fast and consistent performance at the best price.

☒ **Free trial**
Configure for learning about Amazon Redshift. This configuration is free for a limited time if your organization has never created an Amazon Redshift cluster.

? If your organization has never created an Amazon Redshift cluster, you're eligible for a free limited time trial of our dc2.Large node. The following are estimated charges if you are not eligible for a free trial.
[Learn more](#)

Drop all the created tables from the query editor

Step 2: Upload sample tickitdb data into S3

Step 3: Create a role in IAM to grant Redshift to access S3

Role ARN	arn:aws:iam::074367899248:role/Redshift-S3Role Copy
Role description	Allows Redshift clusters to call AWS services on your behalf. Edit
Instance Profile ARNs	Copy
Path	/
Creation time	2021-09-04 08:32 UTC+0530
Last activity	Not accessed in the tracking period
Maximum session duration	1 hour Edit

Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

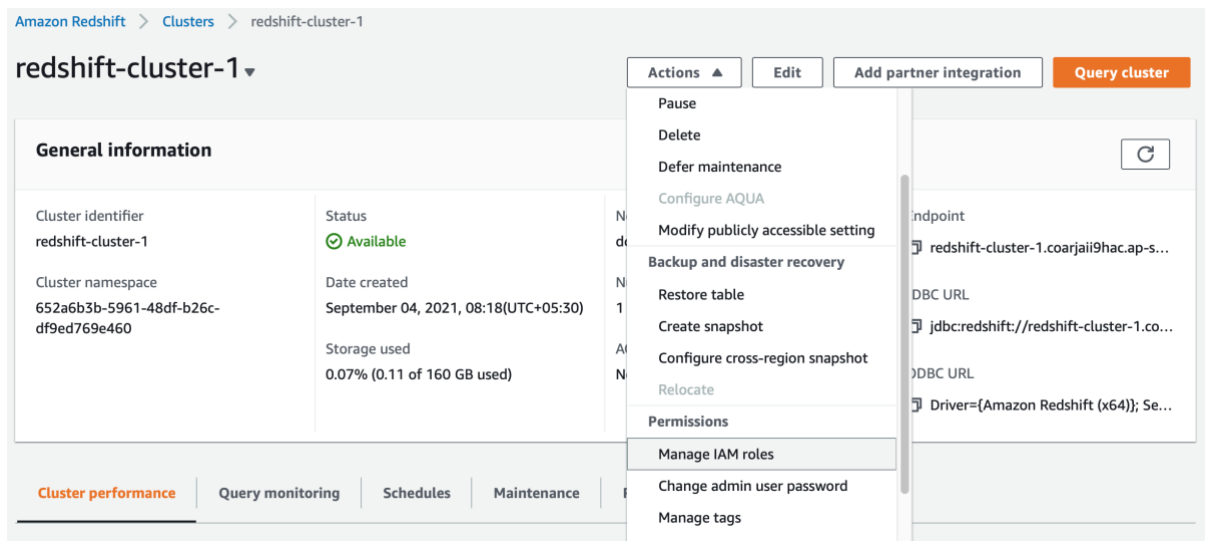
▼ Permissions policies (1 policy applied)

[Attach policies](#)

Policy name ▼	Policy type ▼
▶ AmazonS3FullAccess	AWS managed policy

▶ Permissions boundary (not set)

Step 3.1: Select the Redshift cluster and click Actions and select Modify IAM roles. Select the role created and click Associate IAM role



Step 4: Run the following commands in the query editor to create sample tables

```
create table users(
    userid integer not null distkey sortkey,
    username char(8),
    firstname varchar(30),
    lastname varchar(30),
    city varchar(30),
    state char(2),
    email varchar(100),
    phone char(14),
    likesports boolean,
    liketheatre boolean,
    likeconcerts boolean,
    likejazz boolean,
    likeclassical boolean,
    likeopera boolean,
    likerock boolean,
    likevegas boolean,
    likebroadway boolean,
    likemusicals boolean);

create table venue(
    venueid smallint not null distkey sortkey,
    venue name varchar(100),
    venue city varchar(30),
    venue state char(2),
    venue seats integer);

create table category(
    catid smallint not null distkey sortkey,
```

```

catgroup varchar(10),
catname varchar(10),
catdesc varchar(50));
create table date(
    dateid smallint not null distkey sortkey,
    caldate date not null,
    day character(3) not null,
    week smallint not null,
    month character(5) not null,
    qtr character(5) not null,
    year smallint not null,
    holiday boolean default('N'));

create table event(
    eventid integer not null distkey,
    venueid smallint not null,
    catid smallint not null,
    dateid smallint not null sortkey,
    eventname varchar(200),
    starttime timestamp);
create table listing(
    listid integer not null distkey,
    sellerid integer not null,
    eventid integer not null,
    dateid smallint not null sortkey,
    numtickets smallint not null,
    priceperticket decimal(8,2),
    totalprice decimal(8,2),
    listtime timestamp);

```

Step 5: Run the following COPY commands to load your data in from S3 in to redshift.

```

copy users from 's3://glueblueprintbucket/tickitdb/allusers_pipe.txt'
credentials 'aws_iam_role=arn:aws:iam::137817137639:role/redshifts3role'
delimiter '|' region 'ap-south-1';

copy venue
from 's3://glueblueprintbucket/tickitdb/venue_pipe.txt'
credentials 'aws_iam_role=arn:aws:iam::137817137639:role/redshifts3role'
delimiter '|' region 'ap-south-1';

copy category
from 's3://glueblueprintbucket/tickitdb/category_pipe.txt'
credentials 'aws_iam_role=arn:aws:iam::137817137639:role/redshifts3role'
delimiter '|' region 'ap-south-1';

```

```
copy date from 's3://glueblueprintbucket/tickitdb/date2008_pipe.txt'  
credentials 'aws_iam_role=arn:aws:iam::137817137639:role/redshifts3role'  
delimiter '|' region 'ap-south-1';
```

```
copy event  
from 's3://glueblueprintbucket/tickitdb/allevnts_pipe.txt'  
credentials 'aws_iam_role=arn:aws:iam::137817137639:role/redshifts3role'  
delimiter '|' region 'ap-south-1';
```

```
copy listing  
from 's3://glueblueprintbucket/tickitdb/listings_pipe.txt'  
credentials 'aws_iam_role=arn:aws:iam::137817137639:role/redshifts3role'  
delimiter '|' region 'ap-south-1';
```

Query Using Redshift Spectrum

Step 6: Create a role in IAM which grants Redshift permission for Athena, Glue and S3.

Create role

Review

Provide the required information below and review this role before you create it.

Role name* Redshift-Athena-Glue-S3-FullAccess







Use alphanumeric and '+,=, @, _' characters. Maximum 64 characters.

Role description Allows Redshift clusters to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+,=, @, _' characters.

Trusted entities AWS service: redshift.amazonaws.com

Policies

-  [AmazonAthenaFullAccess](#) 
-  [AWSGlueConsoleFullAccess](#) 
-  [AmazonS3FullAccess](#) 

Permissions boundary Permissions boundary is not set

Step 7: Modify the cluster to add the above role

Amazon Redshift > Clusters > Manage IAM roles

Manage IAM roles: redshift-cluster-1

Cluster permissions

Info Your cluster needs permissions to access other AWS services on your behalf. For the required permissions, add IAM roles with the principal "redshift.amazonaws.com". You can associate up to 10 IAM roles with this cluster. [Learn more](#)

Available IAM roles [Info](#)

Redshift-Athena-Glue-S3-FullAccess

Associated IAM roles	Status	
AWSServiceRoleForRedshift arn:aws:iam::074367899248:role/aws-service-role/redshift.amazonaws.com/AWSServiceRoleForRedshift	in-sync	<input type="button" value="Remove"/>
Redshift-S3Role arn:aws:iam::074367899248:role/Redshift-S3Role	in-sync	<input type="button" value="Remove"/>

Step 8: Create a schema in Redshift

Create external schema salesschema from data catalog database 'dev'

```
iam_role 'arn:aws:iam::137817137639:role/redshifts3role'
```

Create external database if not exists;

Step 9: Create an external table pointing to S3 data

```
create external table salesschema.sales(  
    salesid integer ,  
    listid integer ,  
    sellerid integer ,  
    buyerid integer ,  
    eventid integer ,  
    dateid smallint ,  
    qtysold smallint ,  
    pricepaid decimal(8,2),  
    commission decimal(8,2),  
    saletime timestamp)  
row format delimited fields terminated by '\t'  
stored as textfile  
location 's3://glueblueprintbucket/ticketdb/sales/';
```

```
select * from salesschema.sales;
```