Paper 3391-2019

Your Data Is in Amazon Web Services (AWS): How Do You Access It with SAS®?

Lou Galway, SAS Institute Inc.

ABSTRACT

This paper discusses three different types of storage services available from Amazon Web Services (AWS)—Amazon S3, Amazon Aurora, and Amazon Redshift—and how SAS® can access data from each type of storage service for analytical purposes. Amazon S3 stores data as objects, such as files. You can access these files by using the S3 procedure in SAS. Amazon Aurora is a relational database that is part of Amazon's Relational Database Service. The engine for Aurora is compatible with both MySQL or Postgres. Depending on whether the Aurora database is based on MySQL or Postgres, you can use SAS/ACCESS® Interface to MySQL or SAS/ACCESS® Interface to Postgres to access the data in Aurora. Amazon Redshift is a fully managed, scalable data warehouse in the cloud. You can use SAS/ACCESS® Interface to Amazon Redshift to access data stored in Amazon Redshift.

INTRODUCTION

Amazon Web Services (AWS) is one of the leading providers of cloud computing services, including storage and databases. The many services offered by AWS include Amazon Simple Storage Service (S3), Amazon Aurora, and Amazon Redshift. S3 is a file storage system that enables users to upload data to the AWS cloud. Aurora is a database system that can be used for applications. Redshift is a data warehousing service that can also be used for business applications. All three of these services can be data sources and can store the output of analytics from SAS. This paper does not discuss mass uploading of data to the cloud. It assumes that data is already there, or that you want to load data or results back to AWS from SAS.

DATA STORED IN S3

S3 is a cost-effective storage offering from Amazon Web Services. It is typically used for storing files. As Hadoop seems to be losing its newness appeal and companies are looking for ways to store data in the cloud cost effectively, S3 is a viable option. Customers of S3 can store text files, data files, image files, and other data from the web. More details about S3 can be found in a FAQ document on the Amazon site.

SAS users can access files stored in S3 either by using the S3 procedure or by creating a caslib. PROC S3 is used for object management, such as creating buckets or files. Table 1 shows a list of use cases:

PROC S3	Specifies the connection parameters to S3.	Ex. 1, Ex. 2, Ex. 3
BUCKET	Specifies whether to enable transfer acceleration for a bucket.	
COPY	Copies an S3 object to an S3 destination.	Ex. 2, Ex. 3
CREATE	Creates an S3 bucket.	Ex. 1

DELETE	Deletes an S3 location or object.	Ex. 2
DESTROY	Deletes an S3 bucket.	
ENCKEY	Enables you to work with encryption keys.	Ex. 3
GET	Retrieves an S3 object.	Ex. 3
GETACCEL	Retrieves the transfer acceleration status for a bucket.	
GETDIR	Retrieves the contents of an S3 directory.	Ex. 3
INFO	Lists information about an S3 location or object.	
LIST	Lists the contents of an S3 location.	Ex. 1
MKDIR	Specifies a directory to create in an S3 location.	Ex. 2
PUT	Specifies a local object to write to an S3 location.	Ex. 1, Ex. 3
PUTDIR	Specifies a local directory to write to an S3 location.	Ex. 3
RMDIR	Deletes a directory from an S3 location.	

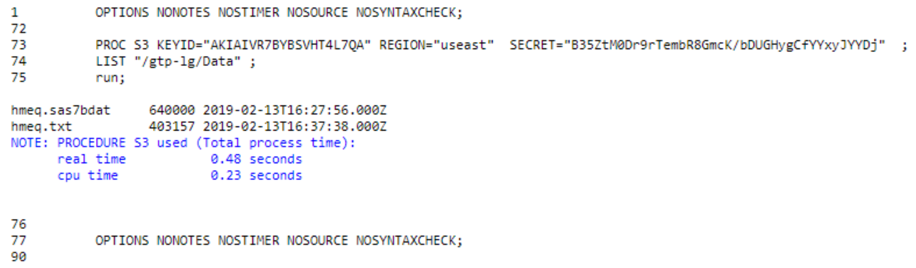
Table 1. PROC S3 Use Cases

To use PROC S3, you need an AWS bucket, folder with data files, region, key ID, and secret. For more information about keys and secrets, see the Amazon security documentation.

For a SAS user, S3 becomes another data source to support analytics. For example, a text file could be stored in S3 and be used for predictive modeling. The following is an example of using the PROC S3 LIST statement to list the contents of a bucket or location named gtp-lg/Data:

```
PROC S3 KEYID="XXXXX" REGION="useast"  SECRET="XXXXX"  ;
LIST "/gtp-lg/Data”;
run;
```

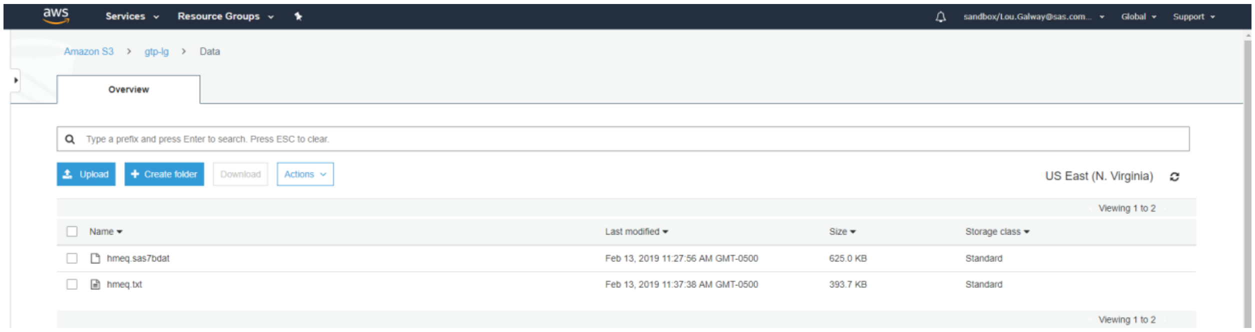
In Display 1, the log file shows two files in the folder “Data.”



```
1          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
72
73         PROC S3 KEYID="AKIAIVR7BYBSVHT4L7QA" REGION="useast"  SECRET="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"  ;
74         LIST "/gtp-lg/Data" ;
75         run;

hmeq.sas7bdat    640000 2019-02-13T16:27:56.000Z
hmeq.txt         403157 2019-02-13T16:37:38.000Z
NOTE: PROCEDURE S3 used (Total process time):
      real time           0.48 seconds
      cpu time            0.23 seconds


76
77         OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
90
```

Display 1. Log File

In Display 2, the view is from the AWS interface showing the same two files:

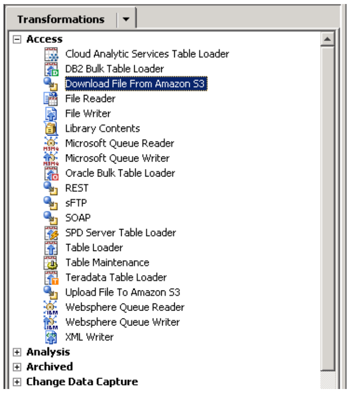


Name	Last modified	Size	Storage class
hmeq.sas7bdat	Feb 13, 2019 11:27:56 AM GMT-0500	625.0 KB	Standard
hmeq.txt	Feb 13, 2019 11:37:38 AM GMT-0500	393.7 KB	Standard

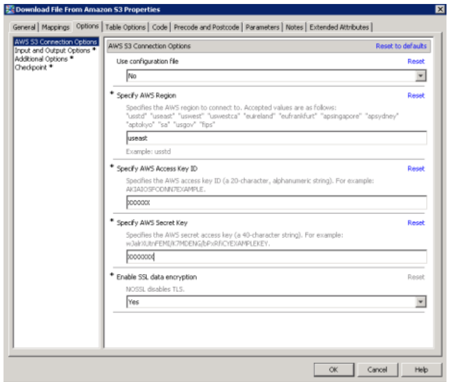
Viewing 1 to 2

Display 2. AWS Interface

SAS® Data Integration Studio has a data transformation to download a file from S3. The Download File From Amazon S3 transformation became available with the SAS Data Integration Studio release 9.402, as part of SAS9.4M4. In Display 3, the transformation is located under the **Access** group. The Download File From Amazon S3 transformation has options to configure PROC S3 behind the scenes, as shown in Display 4. In Display 4 and Display 5, the same information is required as in the preceding PROC S3 code example: the AWS access key, region, secret, and S3 location. The output file location is needed for the GET use case to define where to move the S3 data to the local file system.



Display 3. Download File From Amazon S3 Transformation



AWS S3 Connection Options

Use configuration file: No

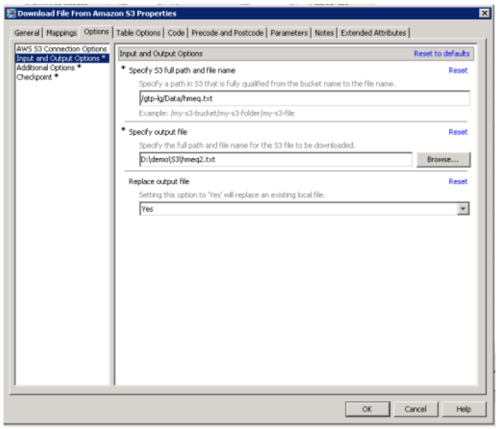
* Specify AWS Region
Specifies the AWS region to connect to. Accepted values are as follows:
"usstd" "useast" "uswest" "uswestca" "euireland" "eufrankfurt" "apsingapore" "apsydney" "aptokyo" "sa" "usgov" "fips"
useast
Example: usstd

* Specify AWS Access Key ID
Specifies the AWS access key ID (a 20-character, alphanumeric string). For example: AKIAIOSFODNN7EXAMPLE.
xxxxxx

* Specify AWS Secret Key
Specifies the AWS secret access key (a 40-character string). For example: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY.
xxxxxxx

* Enable SSL data encryption
NOSSL disables TLS.
Yes

Display 4. Connection Options for Download Transformation



Input and Output Options

* Specify S3 full path and file name

Specify a path in S3 that is fully qualified from the bucket name to the file name.

/gtp-lg/Data/hmeq.txt

Example: /my-s3-bucket/my-s3-folder/my-s3-file

* Specify output file

Specify the full path and file name for the S3 file to be downloaded.

D:\demo\S3\hmeq2.txt

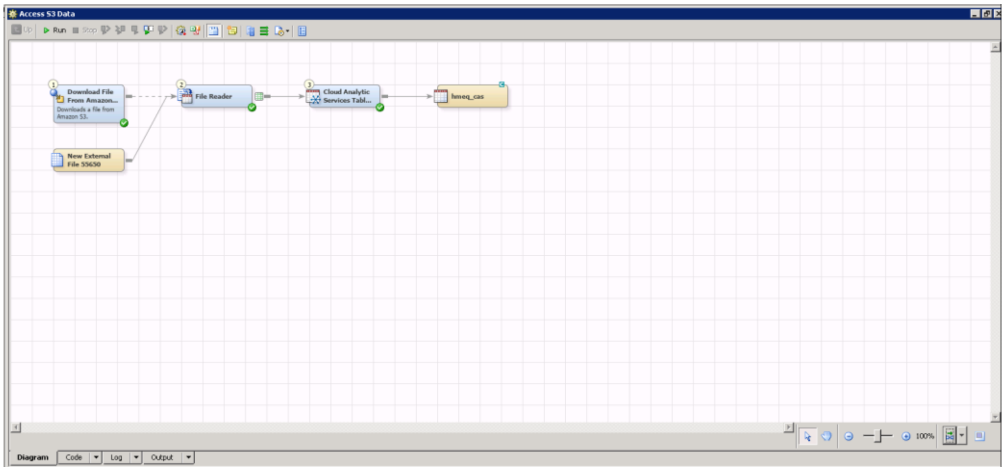
Replace output file

Setting this option to 'Yes' will replace an existing local file.

Yes

Display 5. Input and Output Options for Download Transformation

Display 6 is an example of using the Download File From Amazon S3 transformation in SAS Data Integration Studio to access data in S3 and then load the data into SAS® Cloud Analytic Services (CAS). CAS is a cloud-based run-time engine that is part of the SAS® Viya architecture. This job has two steps: The first is moving data from S3 to the local file system where SAS in installed, and the second is lifting the data into CAS.

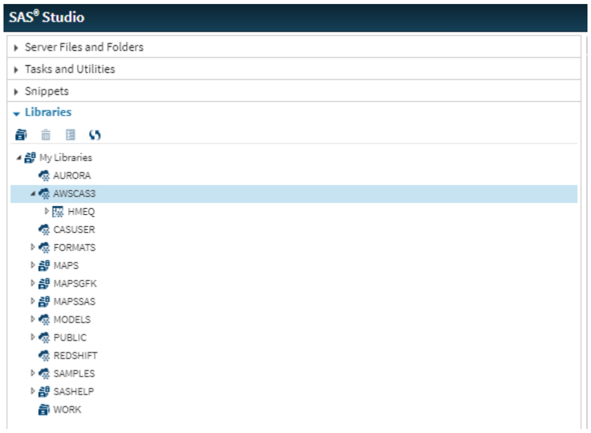


Display 6. Download Transformation in SAS Data Integration Studio

Another way to access data in S3 is with the use of a CASLIB statement. CASLIB statements are used when interacting with CAS. The steps to move data into CAS for processing are to use a CASLIB statement to define a connection to S3 and then use PROC CASUTIL to load data to CAS for use in-memory. PROC CASUTIL is a utility procedure to manage tables and caslibs in three main areas: transferring data, managing table and file information, and dropping and deleting files. The following is an example CASLIB statement and PROC CASUTIL:

```
caslib AWSCAS3 datasource=(srctype="s3",

accessKeyId="XXXX",

secretAccessKey="XXXX",

region="US_East",

bucket="gtp-lg",
objectpath="/Data/");

proc cas;
session mySession;

action loadtable submit /
caslib="AWSCAS3"
path="hmeq.txt" ;
run;
```

Executing this code creates a caslib and loads data to CAS. Display 7 shows the same table (HMEQ) in the previous example in a caslib in SAS® Studio.



Display 7. Tables in a Caslib named AWSCAS3

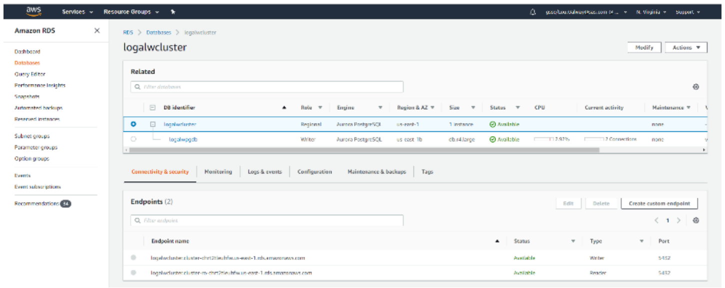
This section discussed two different methods to access data stored in Amazon S3: using PROC S3 and creating a caslib. Please note that PROC S3 is not a SAS/ACCESS® engine. The procedure performs object management functions as described in Table 1. The caslib described in the preceding example is not the same as a SAS library created with a SAS/ACCESS engine. A caslib is an in-memory space to hold tables, access control lists, and data source information when interacting with CAS. The reason that SAS/ACCESS technology is not used with S3 is because S3 doesn’t process data; it only stores the data.

DATA STORED IN AURORA

Aurora from Amazon Web Services is a high-performance, scalable, secure, fully managed relational database that is compatible with MySQL and Postgres SQL. Aurora customers have stated that its cost effectiveness, elasticity of capacity, database scalability, and billing for consumption are desired service qualities. The Amazon Aurora database is used for enterprise applications, software as a service applications, and web and gaming applications.

SAS can access data in an Aurora database with either SAS/ACCESS® Interface to MySQL or SAS/ACCESS® Interface to PostgreSQL, depending how the database was created. In the creation of the Aurora database, there is a choice to make the Aurora database compatible with MySQL or Postgres. In the example for this paper, the version compatible with Postgres was chosen during the database creation.

Display 8 shows an Aurora database in AWS named logalwpgdb.



DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maintenance
logalwcluster	Regional	Aurora PostgreSQL	us-east-1	1 instance	Available			none
logalwpgdb	Writer	Aurora PostgreSQL	us-east-1b	db.r4.large	Available	2.92%	2 Connections	none

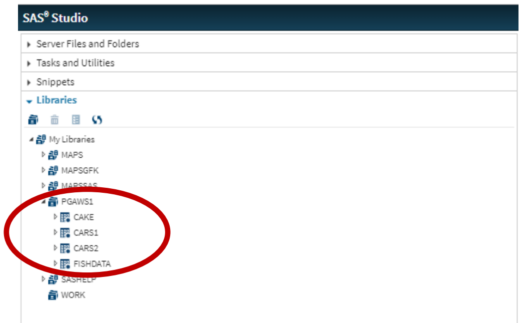
Endpoint name	Status	Type	Port
logalwcluster.cluster-chrt2tleuhfw.us-east-1.rds.amazonaws.com	Available	Writer	5432
logalwcluster.cluster-ro-chrt2tleuhfw.us-east-1.rds.amazonaws.com	Available	Reader	5432

Display 8. Aurora Database in AWS

The following is an example of using the SAS/ACCESS Interface to PostgreSQL to connect to the logalwpgdb database with a SAS LIBNAME statement:

```
LIBNAME pgaws1 POSTGRES SERVER= "logalwpgdb.chrt2tleuhfw.us-east-
1.rds.amazonaws.com" PORT=5432 DATABASE=logalwpgdb

USER=XXXX PW=XXXX ;
```

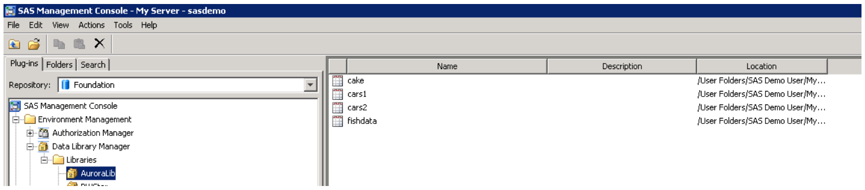
Required connection information for the LIBNAME statement are server, port, database, user, and password. Display 9 shows the resulting library named “pgaws1” with a few tables in the Aurora database.



Display 9. PGAWS1 Library

The same connection can be made with SAS® Management Console or SAS Data Integration Studio. Display 10 shows the use of SAS Management Console to create a SAS library. The connection requirements are the same as those for the LIBNAME statement. The engine (Postgres or MySQL), server, database, user, and password are needed. Display 11 shows the connection dialog box for the Postgres Server named Aurora Server. The user and password were supplied by creating the AuroraAuth, and the user ID and password are added to a SAS user account in the user manager. The last step is creating a library using the Postgres engine and selecting the server created in Display 11. Detailed instructions for creating a library can be found on the SAS support site:

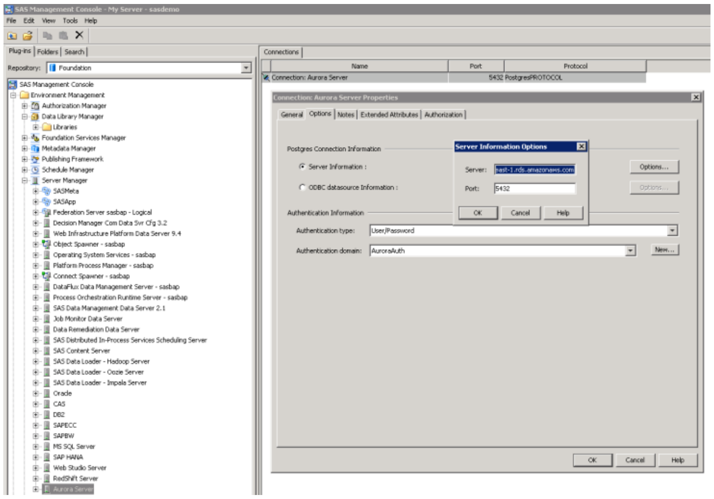
https://go.documentation.sas.com/?docsetId=bidsag&docsetTarget=p01ik9gejwwfhtn1ay82cgkg9p0c.htm&docsetVersion=9.4&locale=en



Repository: Foundation

Name	Description	Location
cake		/User Folders/SAS Demo User/My...
cars1		/User Folders/SAS Demo User/My...
cars2		/User Folders/SAS Demo User/My...
fishdata		/User Folders/SAS Demo User/My...

Display 10. Creating a SAS Library with SAS Management Console


SAS Management Console - My Server - sasdemo
File Edit View Tools Help
Plug-ins
Folders
Search
SAS Management Console
Environment Management
Authorization Manager
Data Library Manager
Libraries
Foundation Services Manager
Metadata Manager
Publishing Framework
Schedule Manager
Server Manager
SASMeta
SASApp
Federation Server sasbap - Logical
Decision Manager Com Data Svr Cfg 3.2
Web Infrastructure Platform Data Server 9.4
Object Spawner - sasbap
Operating System Services - sasbap
Platform Process Manager - sasbap
Connect Spawner - sasbap
DataFlux Data Management Server - sasbap
Process Orchestration Runtime Server - sasbap
SAS Data Management Data Server 2.1
Job Monitor Data Server
Data Remediation Data Server
SAS Distributed In-Process Services Scheduling Server
SAS Content Server
SAS Data Loader - Hadoop Server
SAS Data Loader - Oozie Server
SAS Data Loader - Impala Server
Oracle
CAS
DB2
SAPECC
SAPBW
MS SQL Server
SAP HANA
Web Studio Server
RedShift Server
Aurora Server
Connections
Connection: Aurora Server Properties
General
Options
Notes
Extended Attributes
Authorization
Server Information Options
OK
Cancel
Help
Options...
Options...
New...
OK
Cancel
Help

Repository: Foundation

Name	Port	Protocol
Connection: Aurora Server	5432	PostgresPROTOCOL

Postgres Connection Information

● Server Information :

○ ODBC datasource Information :

Server: east-1.rds.amazonaws.com

Port: 5432

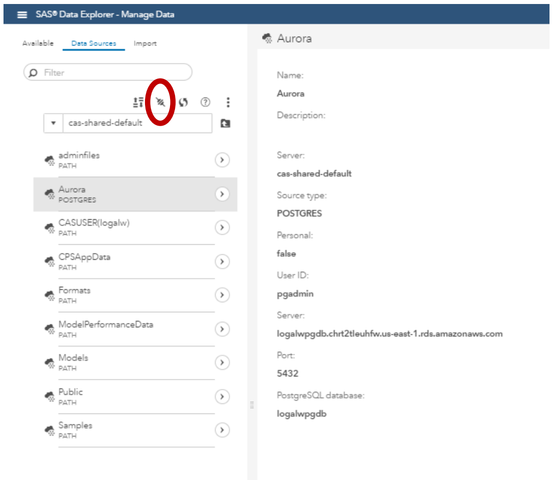
Authentication Information

Authentication type: User/Password

Authentication domain: AuroraAuth

Display 11. Postgres Server Information for Aurora Database

SAS® Data Explorer can also be used to make the connection to the Aurora database. In SAS Data Explorer, select **Data Sources** and click the new connection icon shown in Display 12 to invoke the Connection Settings dialog box as shown in Display 13. Define the connection with the same required information as you would use to create a LIBNAME statement: engine type, server, database, user, and password, as shown in Display 13.



Name:
Aurora

Description:

Server:
cas-shared-default

Source type:
POSTGRES

Personal:
false

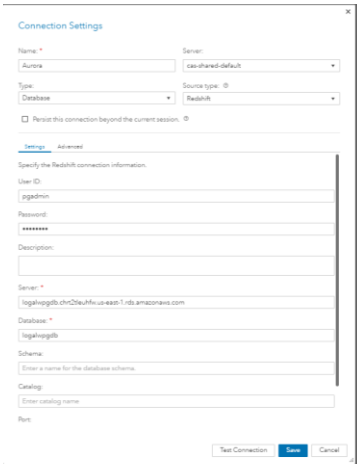
User ID:
pgadmin

Server:
logalwpgdb.chrt2tleuhfw.us-east-1.rds.amazonaws.com

Port:
5432

PostgreSQL database:
logalwpgdb

Display 12. SAS Data Explorer


Connection Settings

Name: * Aurora

Server: cas-shared-default

Type: Database

Source type: Redshift

□ Persist this connection beyond the current session.

Specify the Redshift connection information.

User ID: pgadmin

Password: ••••••••

Description:

Server: * logalwpgdb.chrt2tleuhfw.us-east-1.rds.amazonaws.com

Database: * logalwpgdb

Schema: Enter a name for the database schema.

Catalog: Enter catalog name

Port:

Display 13. Connection Settings in SAS Data Explorer

As companies are moving their data to cloud databases such as Amazon Aurora, there is still a need to analyze the data stored in the database as well as other data sources. Data scientists, data engineers, and business analysts can access data in the Aurora database using SAS access engines such as SAS/ACCESS Interface to PostgreSQL and SAS/ACCESS Interface to MySQL. These SAS access engines can be part of larger solutions that use different interfaces such as SAS Management Console, SAS Data Integration Studio, SAS® Visual Analytics, SAS® Visual Data Mining and Machine Learning, and others.

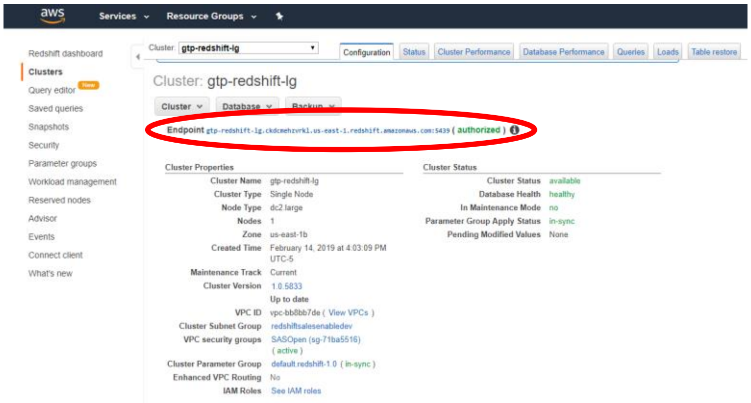
DATA STORED IN REDSHIFT

Amazon Redshift is a fast, cost-effective scalable data warehousing service offered by AWS. It is used to support reporting and dashboards, providing a repository for data spread between structured and unstructured data.

SAS connects to RedShift using SAS/ACCESS Interface to Amazon Redshift. This interface is specifically designed to access data effectively. Jeff Bailey and Chris DeHart wrote a 2016 SAS Global Forum paper describing the SAS/ACCESS Interface to Amazon Redshift in detail:

http://support.sas.com/resources/papers/proceedings16/SAS5200-2016.pdf

Display 14 shows a RedShift example, including the required information to connect from SAS: the endpoint (server), port, database, schema, user, and password.



Cluster: gtp-redshift-lg

Cluster: gtp-redshift-lg

Endpoint gtp-redshift-lg.ckdcmehzvrkl.us-east-1.redshift.amazonaws.com:5439 (authorized)

Cluster Properties

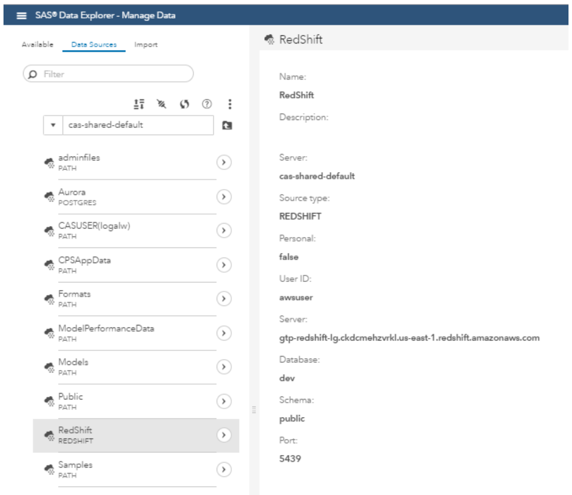
Cluster Name	gtp-redshift-lg
Cluster Type	Single Node
Node Type	dc2.large
Nodes	1
Zone	us-east-1b
Created Time	February 14, 2019 at 4:03:09 PM UTC-5
Maintenance Track	Current
Cluster Version	1.0.5833 Up to date
VPC ID	vpc-bb8bb7de (View VPCs)
Cluster Subnet Group	redshiftsalesenabledev
VPC security groups	SASOpen (sg-71ba5516) (active)
Cluster Parameter Group	default.redshift-1.0 (in-sync)
Enhanced VPC Routing	No
IAM Roles	See IAM roles

Cluster Status

Cluster Status	available
Database Health	healthy
In Maintenance Mode	no
Parameter Group Apply Status	in-sync
Pending Modified Values	None

Display 14. Redshift Interface

SAS interfaces such as SAS Data Explorer, SAS Management Console, SAS Data Integration Studio, and SAS Studio use the SAS/ACCESS Interface to Amazon Redshift. Display 15 shows a connection to a database (dev) in Redshift (gtp-redshift-lg) with SAS Data Explorer using the SAS/ACCESS Interface to Amazon Redshift and the Redshift data connector. The term *data connector* is specific to the SAS Viya architecture, and data connectors are part of the SAS/ACCESS engine bundle. For example, the SAS/ACCESS Interface for Redshift on SAS Viya includes the appropriate data connector. Display 16 shows the selection of a database server used in a Redshift SAS library in SAS Management Console. The process of creating a library in SAS Management Console for Redshift is similar to creating a library for Aurora as previously described, except for the selection of the engine type. The engine type to use is Redshift, as shown in Display 17.



cas-shared-default

adminfiles	PATH
Aurora	POSTGRES
CASUSER(logalw)	PATH
CPSAppData	PATH
Formats	PATH
ModelPerformanceData	PATH
Models	PATH
Public	PATH
RedShift	REDSHIFT
Samples	PATH

RedShift

Name:
RedShift

Description:

Server:
cas-shared-default

Source type:
REDSHIFT

Personal:
false

User ID:
awsuser

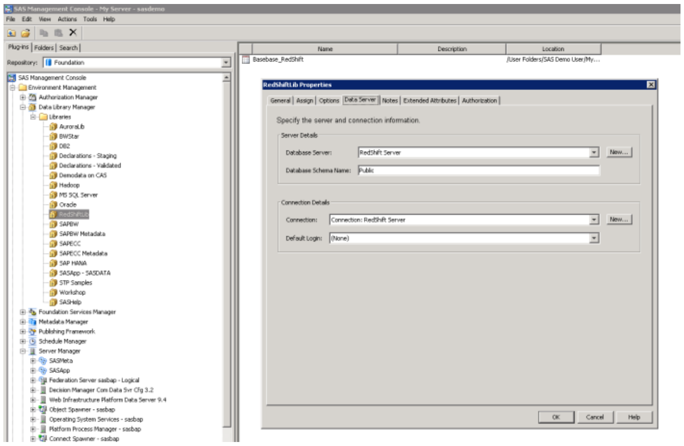
Server:
gtp-redshift-lg.ckdcmehzvrkl.us-east-1.redshift.amazonaws.com

Database:
dev

Schema:
public

Port:
5439

Display 15. Connection to a Database in Redshift



Specify the server and connection information.

Server Details

Database Server: RedShift Server

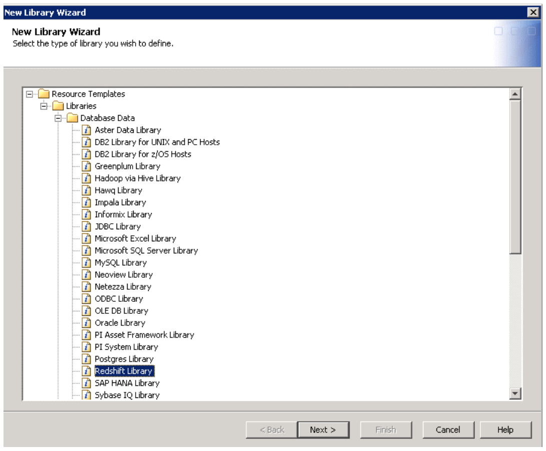
Database Schema Name: Public

Connection Details

Connection: Connection: RedShift Server

Default Login: (None)

Display 16. Selection of Database Server in SAS Management Console

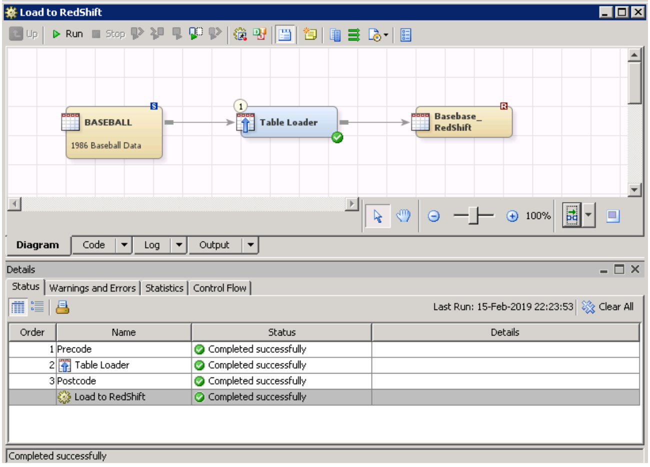

New Library Wizard

New Library Wizard

Select the type of library you wish to define.

Display 17. Selection of Redshift Engine Type

After the Redshift SAS library is configured, it can be used in SAS applications such as SAS Data Integration Studio to supply data for jobs to manipulate data, feed analytic models, and move data. Display 18 shows a simple SAS Data Integration Studio job to load a SAS table to Redshift.



Last Run: 15-Feb-2019 22:23:53

Order	Name	Status	Details
1	Precode	Completed successfully	
2	Table Loader	Completed successfully	
3	Postcode	Completed successfully	
	Load to RedShift	Completed successfully	

Completed successfully

Display 18. SAS Data Integration Studio Job

The Amazon Redshift service offers a powerful, scalable, cost-effective option for a data warehouse in the cloud. The data stored in Redshift can be valuable with the use of analytics, but it first must be accessed with an analytic platform. The SAS/ACCESS Interface for Redshift offers more than the capability to read and write data to Redshift. A few points from the SAS Global Forum paper It's raining data! Harnessing the Cloud with Amazon Redshift and SAS/ACCESS® are worth repeating. The installation and configuration process is made easier by bundling the DataDirect Amazon Redshift driver and ODBC Driver Manager instead of obtaining and setting it up yourself. More SQL pass-down functions enable you to let Redshift do some of the work prior to moving data to SAS for further processing. Finally, the DataDirect driver provides better write performance.

CONCLUSION

Companies are moving their data to the cloud as part of the Digital Transformation movement. AWS cloud services such as S3, Aurora, and Redshift are viable options to store data. To gain additional value and insights from this data, the first step is to access the data. This paper has shown multiple examples of how to access data in S3, Aurora, and Redshift using SAS/ACCESS technology and the PROC S3 procedure. Using SAS/ACCESS technology and PROC S3 procedure, you can easily access data stored in Amazon to fuel your analytic processes.

ACKNOWLEDGMENTS

I would like to thank the following for their support in writing this paper:

Jeff Bailey

Uttam Kumar

Peter McNeil

RECOMMENDED READING

- Bailey, Jeff. 2014. "An Insider's Guide to SAS/ACCESS® Interface to ODBC." *Proceedings of the SAS Global Forum 2014 Conference.* Cary, NC: SAS Institute Inc. Available https://support.sas.com/resources/papers/proceedings14/SAS039-2014.pdf.
- DeHart, Chris and Jeff Bailey. 2016. "It's raining data! Harnessing the Cloud with Amazon Redshift and SAS/ACCESS®." *Proceedings of the SAS Global Forum 2016 Conference.* Cary, NC: SAS Institute Inc. Available http://support.sas.com/resources/papers/proceedings16/SAS5200-2016.pdf.
- Amazon Web Services, Inc. 2019. "What is Amazon Aurora." In *User Guide for Aurora*. Available https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/CHAP_AuroraOverview.html.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Lou Galway
SAS Institute Inc.
919-531-0326
Lou.Galway@sas.com