Paper 2987-2019

Integration of Google BigQuery with SAS® Analytics Pro in Docker Container on Cloud Environment

Sanket Mitra, Srivalli Avadhanula and Fahad Ali, Core Compete

ABSTRACT

SAS® Analytics for Containers provides the option to deploy SAS® Analytics within container-enabled infrastructures, including Docker and Kubernetes, which are often run in the cloud. Aiming to analyze massively large data from Google BigQuery through SAS® in containerized environment, we have integrated Google BigQuery with SAS® 9.4 Analytics Pro in Docker Container on Google Cloud Environment. This paper guides you through the process of configuring SAS® Access to BigQuery in containerized SAS®® Application and validation steps for the same.

INTRODUCTION

BigQuery is a RESTful web service that enables interactive analysis of massively large datasets working in conjunction with Google Storage. It is an Infrastructure as a Service (IaaS) that may be used complementarily with MapReduce. For the benefits of SAS® programmers, who wants to use the analytics feature of SAS® with Google BigQuery in cloud native environment this type of configuration will be useful.

One should have a working knowledge of the following to integrate Google BigQuery with SAS® 9.4 Analytics Pro in Docker container.

- SAS® installation and configuration
- Linux operating system and commands
- Docker Installation and CLI
- Google cloud platform on which the SAS® container can run.

DOCKER INSTALLATION GUIDE

DOWNLOAD THE PACKAGE

- Go to https://download.docker.com/linux/centos/7/x86_64/stable/Packages/ and download the .rpm file for the Docker version docker-ce-18.03.1.ce-1.el7.centos.x86_64.rpm to install.
- Place the rpm in the target path of the server before install.

UNINSTALL OLD VERSIONS (IF ANY)

Older versions of Docker were called docker or docker-engine. If these are installed, uninstall them, along with associated dependencies.

```
yum remove docker \
                  docker-client \
                  docker-client-latest \
                  docker-common \
                  docker-latest \
                  docker-latest-logrotate \
                  docker-logrotate \
                  docker-selinux \
                  docker-engine-selinux \
                  docker-engine
```

INSTALL DOCKER

Install Docker CE, changing the path below to the path where you downloaded the Docker package.

```
$ sudo yum install /path/to/docker-ce-18.03.1.ce-1.el7.centos.x86_64.rpm
```

START DOCKER

Start the docker container by below mentioned command

```
$ sudo systemctl start docker
```

VALIDATION

Verify that docker is installed correctly by running the hello-world image.

```
$ sudo docker run hello-world
```

INTERGRATION OF SAS® WITH GOOGLE BIGQUERY

DOWNLOAD THE SIMBA AND UNIX ODBC DRIVERS

- Download the Simba drivers for Google BigQuery https://cloud.google.com/bigquery/partners/simba-drivers/ (ODBC driver releases 2.1.20.1025
- Download the UnixODBC (http://www.unixodbc.org/download.html)
- Once downloaded, put the ODBC driver in any path.

UNTAR BOTH THE ODBC DRIVER

```
mkdir /usr/local/SASDocker
```

```
tar -xvzf SimbaODBCDriverforGoogleBigQuery64_2.1.11.1011.tar.gz -C /opt

chown root:root simba/ -R
```

```
tar -xvzf unixODBC-2.3.7.tar.gz -C /opt

chown root:root unixODBC-2.3.7 -R
```

INSTALL UNIX ODBC AND COPY SIMBA ODBC DRIVER

Go to unixODBC directory

```
cd /opt/unixODBC-2.3.7
```

Install unixODBC

```
./configure --prefix=/usr/local/SASDocker/unixODBC --
sysconfdir=/usr/local/SASDocker/unixODBC/etc

make

make install
```

Export the following paths

```
export ODBCINI=/opt/simba/googlebigqueryodbc/Setup/odbc.ini

export ODBCINSTINI=/opt/simba/googlebigqueryodbc/Setup/odbcinst.ini
```

```
export ODBCHOME=/opt/simba/googlebigqueryodbc/Setup

export
LD_LIBRARY_PATH=/opt/simba/googlebigqueryodbc/lib/64/:/usr/lib:/usr:/lib:/usr
/local/lib:/usr/lib64:/opt/unixODBC/lib

export
SIMBAGOOGLEBIGQUERYODBCINI=${LD_LIBRARY_PATH}:/opt/simba/googlebigqueryodbc/l
ib/64/simba.googlebigqueryodbc.ini

export
SIMBAINI=/opt/simba/googlebigqueryodbc/lib/64/simba.googlebigqueryodbc.ini

export ODBCSYSINI=$ODBCHOME
```

Get the Refresh Token

Open /opt/simba/googlebigqueryodbc/Setup/odbc.ini file and go to [Google BigQuery 64-bit] or [Google BigQuery 32-bit] based on your requirement.

```
[ODBC Data Sources]
Google BigQuery 32-bit=Simba ODBC Driver for Google BigQuery 32-bit
Google BigQuery 64-bit=Simba ODBC Driver for Google BigQuery 64-bit
```

Figure 1. Display for odbc.ini

Copy the URL and login with credentials to get the token

https://accounts.google.com/o/oauth2/auth?scope=https://www.googleapis.com/auth/bigquery&response_type=code&redirect_uri=urn:ietf:wg:oauth:2.0:oob&client_id=977385342095.apps.googleusercontent.com&hl=en&from_login=1&as=76356ac9e8ce640b&pli=1&authuser=0

```
# RefreshToken: The Refresh Token used. This can be generated from the Windows connection dialog.
# It can also be generated by executing the following steps:
# 1. Get an Authentication by logging into Google from the following URL:
# https://accounts.google.com/o/oauth2/auth?scope=https://www.googleapis.com/auth/bigquery&response_type=code&redirect_uri=urn:ietf:wg:oauth:2.0:oob&client_id=977385342095.apps.googleusercontent.com&hl=
en&from_login=1&as=76356ac9e8ce640b&pli=1&authuser=0
# 2. Run the get_refresh_token.sh shell script and pass in the Authentication Token received in step 1.
# 3. Copy the Refresh Token (the text on the right-side of the colon, without the trailing or leading spaces) from the output of the script.
# This is a required setting.
RefreshToken=1/vDUBagXA2YTRf14h1A4LUJxPD6U8t5Gb7gvyBGWqFAc,
```

Figure 2. Display for odbc.ini

Allow accessory to BigQuery tools

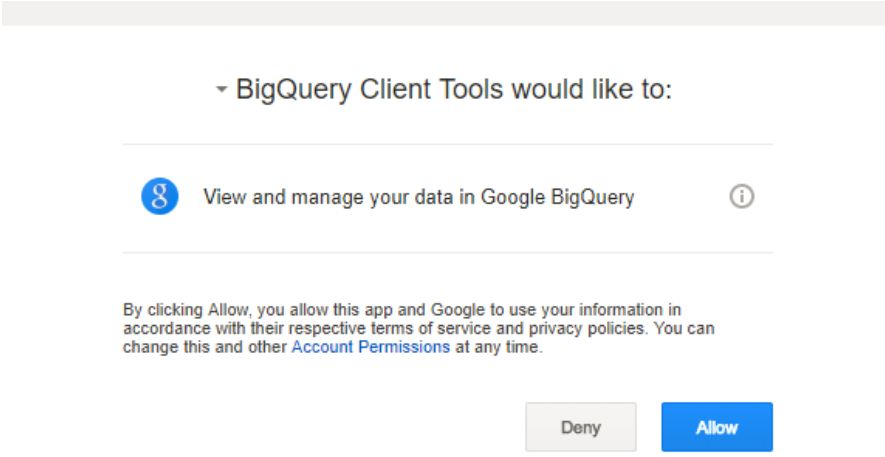


Figure 3. Access to BigQuery tools

Copy the code and take a note for the same

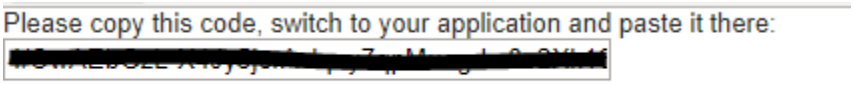


Figure 4. Copy the code

Go to /opt/simba/googlebigqueryodbc/Tools and run get_refresh_token.sh file

```
./get_refresh_token.sh <Place the google code here>
```

```
[sasinst@comp-prod Tools]$ ./get_refresh_token.sh 4/arq7pMyqSU033zz3jlfzdX9H8NhIEE7CffO_UWW-lb0
refresh_token : 1/38U1Xuwi7YCmvfuCjX1K516zBu8xjtYAFvI9cH_BOss,
[sasinst@comp-prod Tools]$
```

Figure 5. Run the get_refresh_token.sh script

Edit /opt/simba/googlebigqueryodbc/Setup/odbc.ini

Change the DSN Name

Rename [Google BigQuery 64-bit] to any short hostname to use as DSN.

```
[googlebq]

# Description: DSN Description.
# This key is not necessary and is only to give a description of the data source.
Description=Simba ODBC Driver for Google BigQuery (64-bit) DSN
```

Figure 6. Rename the DSN

Provide driver path

```
# Driver: The location where the ODBC driver is installed to.
Driver=/opt/simba/googlebigqueryodbc/lib/64/libgooglebigqueryodbc_sb64.so
```

Figure 7. Location of ODBC driver

Provide catalogue

You will get the project name in Project info of google console.

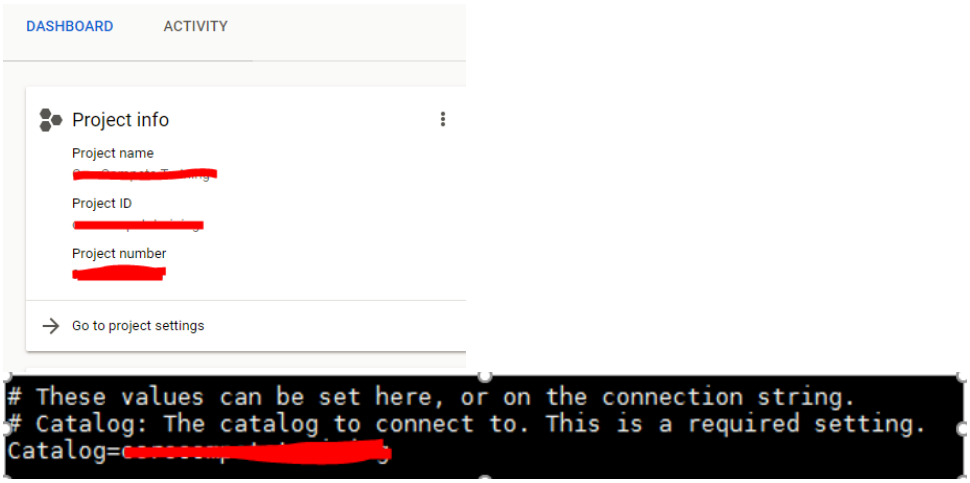


Figure 8. Change the catalog name as per project name

Provide RefreshToken

Figure 9. Edit RefreshToken field

Edit /opt/simba/googlebigqueryodbc/lib/64/simba.googlebigqueryodbc.ini

```
[Driver]

DriverManagerEncoding=UTF-32
```

```
ErrorMessagesPath=/opt/simba/googlebigqueryodbc/ErrorMessages

LogLevel=4

LogPath=/tmp


ODBCInstLib=/opt/unixODBC/lib/libodbcinst.so
```

Edit /opt/simba/googlebigqueryodbc/Setup/odbcinst.ini

Give the whole driver path in 64 bit block and save it

```
[sasinst@comp-prod Setup]$ cat odbcinst.ini
# To use this INI file, replace $(INSTALLDIR) with the
# directory the tarball was extracted to.

[ODBC Drivers]
Simba ODBC Driver for Google BigQuery 32-bit=Installed
Simba ODBC Driver for Google BigQuery 64-bit=Installed

[Simba ODBC Driver for Google BigQuery 32-bit]
Description=Simba ODBC Driver for Google BigQuery(32-bit)
Driver=$(INSTALLDIR)/lib/libgooglebigqueryodbc_sb32.so

[Simba ODBC Driver for Google BigQuery 64-bit]
Description=Simba ODBC Driver for Google BigQuery(64-bit)
Driver=/opt/simba/googlebigqueryodbc/lib/64/libgooglebigqueryodbc_sb64.so
[sasinst@comp-prod Setup]$
```

Figure 10. Edit odbcinst.ini

Copy simba on SASDocker directory

```
cp -Rp /opt/simba /usr/local/SASDocker
```

BUILDING A SAS® 9.4 CONTAINER

INSTALL/CONFIG SAS® STUDIO

Run SAS® Deployment Wizard to install SAS® Studio on a supported Linux 64-bit operating system. During the installation, change the default location for the SAS® Studio installation to */usr/local/SASHome. Steps as below…*

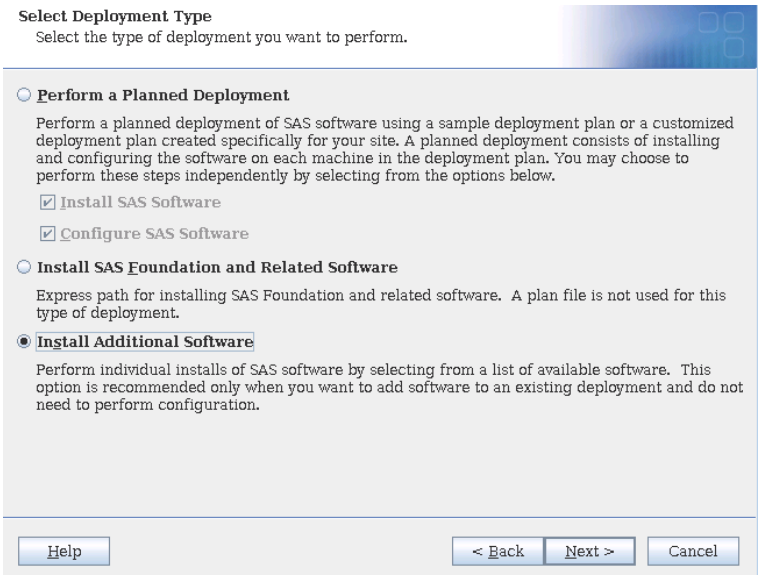


Figure 11. Install steps

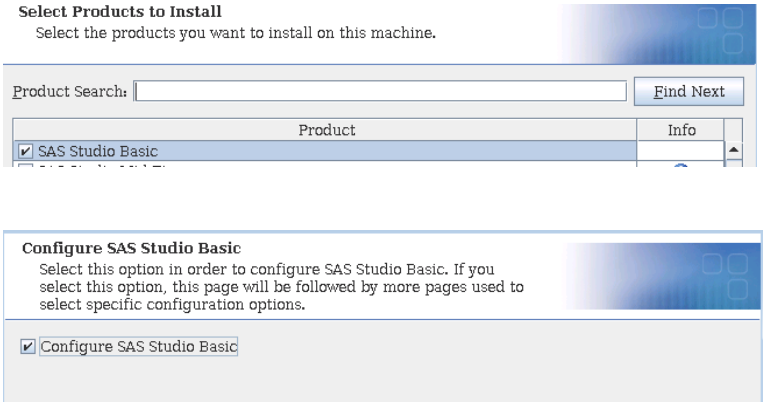


Figure 12. Install steps

- make sure you use PAM Authentication
- make sure you copy system-auth to sasauth

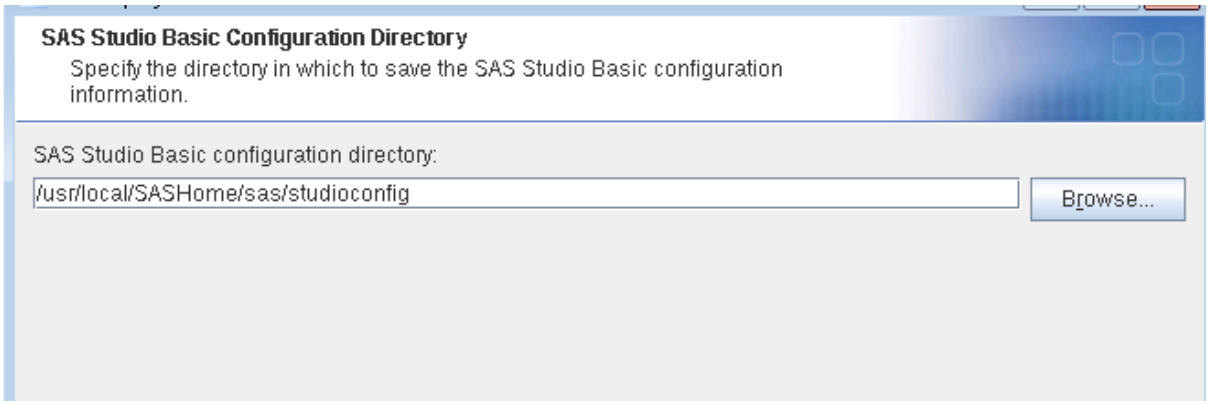


Figure 13. Install steps

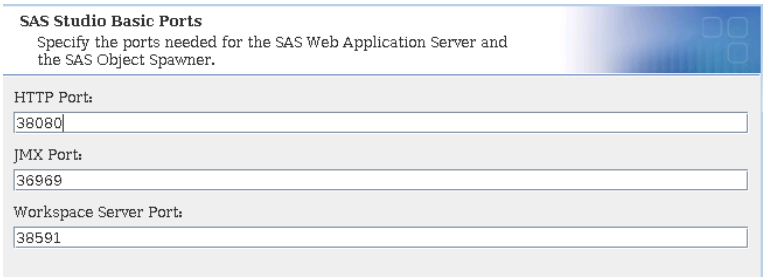


Figure 14. Install steps

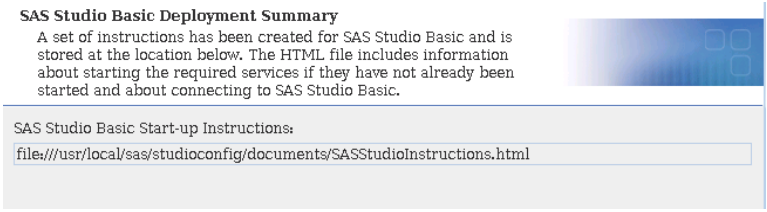


Figure 15. Install steps

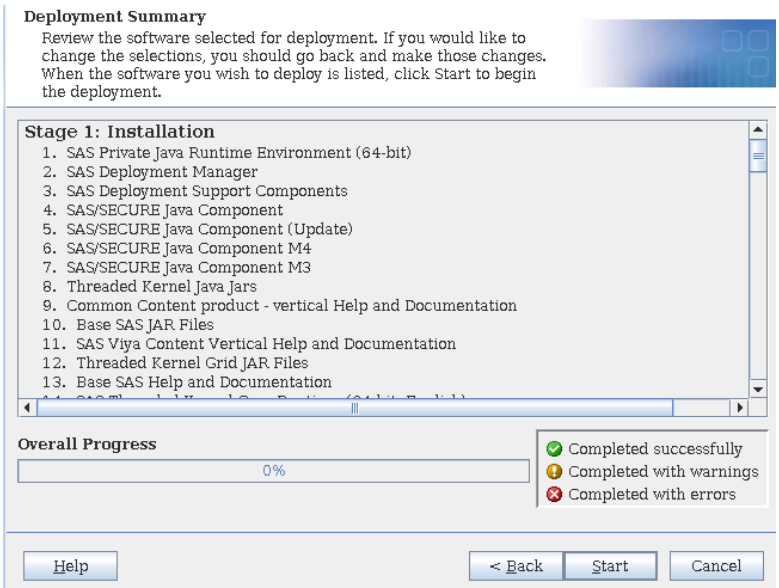


Figure 16. Install steps

ENVIORNMENT VARIABLES

Once validated, place all the environment variables in /usr/local/SASHome/SASFoundation/9.4/bin/sasenv_local

```
#!/bin/sh -p
##########################################################################
#
#
# This file is used to define local environment variables that are useds
# with SAS.  These values will override the default values in sasenv.
# User modifications made to this file will NOT be overwritten by the SAS
# installation program as the system default values file are.
#
# !Important Note:
#
#    Please make sure that this file does not replace the LD_LIBRARY_PATH
#    environment variable. If you need additional paths, make sure you
#    append or prepend the path to the existing  LD_LIBRARY_PATH .
#
#    Redefining the LD_LIBRARY_PATH completely will cause Java to stop working.
#
#    The example below shows a correct way to append to the LD_LIBRARY_PATH.
#
#    LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/local/product/lib
#    export LD_LIBRARY_PATH
#
#
##########################################################################
# This file is sourced by the sasenv script in !SASROOT/bin
##########################################################################

export ODBCINI=/opt/simba/googlebigqueryodbc/Setup/odbc.ini
export ODBCINSTINI=/opt/simba/googlebigqueryodbc/Setup/odbcinst.ini
export ODBCHOME=/opt/simba/googlebigqueryodbc/Setup
export LD_LIBRARY_PATH=/opt/simba/googlebigqueryodbc/lib/64/:/usr/lib:/usr:/lib:/usr/local/lib:/usr/lib64
export SIMBAGOOGLEBIGQUERYODBCINI=/opt/simba/googlebigqueryodbc/lib/64/simba.googlebigqueryodbc.ini
export SIMBAINI=/opt/simba/googlebigqueryodbc/lib/64/simba.googlebigqueryodbc.ini
export ODBCSYSINI=$ODBCHOME



##########################################################################
# End of sasenv_local
##########################################################################
```

Figure 17. Display for sasenv_local

CHANGE SAS_U8 TO SAS_EN

```
vi /usr/local/SASHome/sas/studioconfig/workspaceserver/workspaceserver.sh

# Set environment variables

SAS_COMMAND=/usr/local/SASHome/SASFoundation/9.4/bin/sas_en /*change sas_u8
to sas_en */
```

CREATE A TAR FILE FOR SASHOME

```
tar -cvf SASHomeTar.tar /usr/local/SASHome
```

CREATE A FILE WITH DOCKERFILE

```
FROM centos

MAINTAINER sanket sanket.mitra@corecompete.com

# Install libraries and clean all

RUN yum -y install numactl-libs.x86_64 \

  passwd \

  libXp \

  libpng12 \

  libXmu.x86_64 \

  && yum clean all


# Add group

RUN useradd -m svc_sasinst

RUN groupadd -g 1001 svc_sasgrp


# Add sas user

RUN usermod -a -G svc_sasgrp svc_sasinst


# Set default password by pointing to /etc/passwd

RUN echo -e "password" | /usr/bin/passwd --stdin svc_sasinst


# Make the SASHome directory and add the TAR file

RUN mkdir -p /usr/local/SASHome
```

```
ADD SASHomeTar.tar /
RUN chown -R svc_sasinst:svc_sasgrp /usr/local/SASHome
EXPOSE 38080

# copy system-auth in sasauth to configure pam authentication
RUN cp /etc/pam.d/system-auth /etc/pam.d/sasauth

# copy simba and unixODBC into docker container
COPY simba/ /opt/simba/
COPY unixODBC/ /opt/unixODBC/

# copy libodbc.so.2.0.0 and create softlink
COPY unixODBC/lib/libodbc.so.2.0.0 /lib64/
RUN ln -s /lib64/libodbc.so.2.0.0 /lib64/libodbc.so
RUN ln -s /lib64/libodbc.so.2.0.0 /lib64/libodbc.so.2

# Add startup script to start SAS Studio
ADD startup.sh /
ENTRYPOINT ["/startup.sh"]
```

SASSTUDIO STARTUP SCRIPT (STARTUP.SH)

```
#!/bin/bash
/usr/local/SASHome/SASFoundation/9.4/utilities/bin/setuid.sh
/usr/local/SASHome/sas/studioconfig/sasstudio.sh start
tail -f /dev/null
```

CHANGE PERMISSION OF STARTUP.SH

```
chmod 755 startup.sh
```

DEPLOYING A SAS® CONTAINER

ADD THE FOLLOWING FILES TO THE SASDOCKER DIRECTORY:

- the Dockerfile
- the TAR file that you created, which contains the SASHome directory
- the start-up script that starts SAS® Studio (startup.sh)
- simba driver
- unixODBC driver

BUILD THE DOCKER IMAGE

```
docker build -t sasabq:v1 .
docker images /*to check build images*/
docker run -d -p 38080:38080 sasabq:v1 /*run the container*/
```

TO VALIDATE RUNNING DOCKER

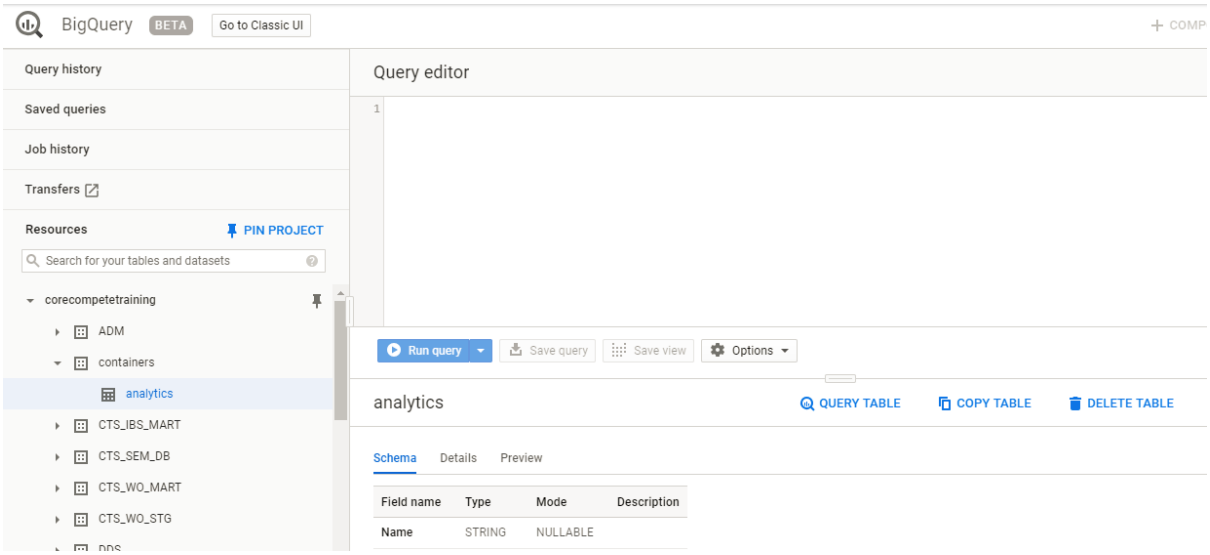
```
docker ps -a
```

TO ENTER INTO RUNNING DOCKER

```
docker exec -it <docker-container-id> /bin/bash
```

VALIDATION

- Go to https://cloud.google.com/ and Login with your google credentials
- Navigate to the project you are working on.
- Navigate to BigQuery
- Create a demo dataset and corresponding schema



Display 18. BigQuery Page in google console

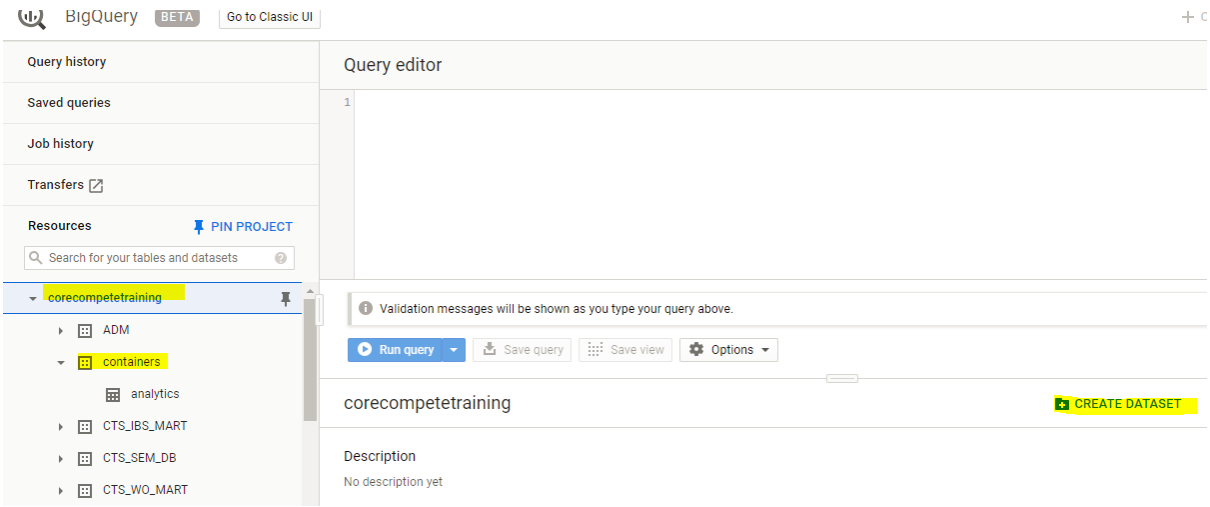


Figure 19. Create dataset in BigQuery

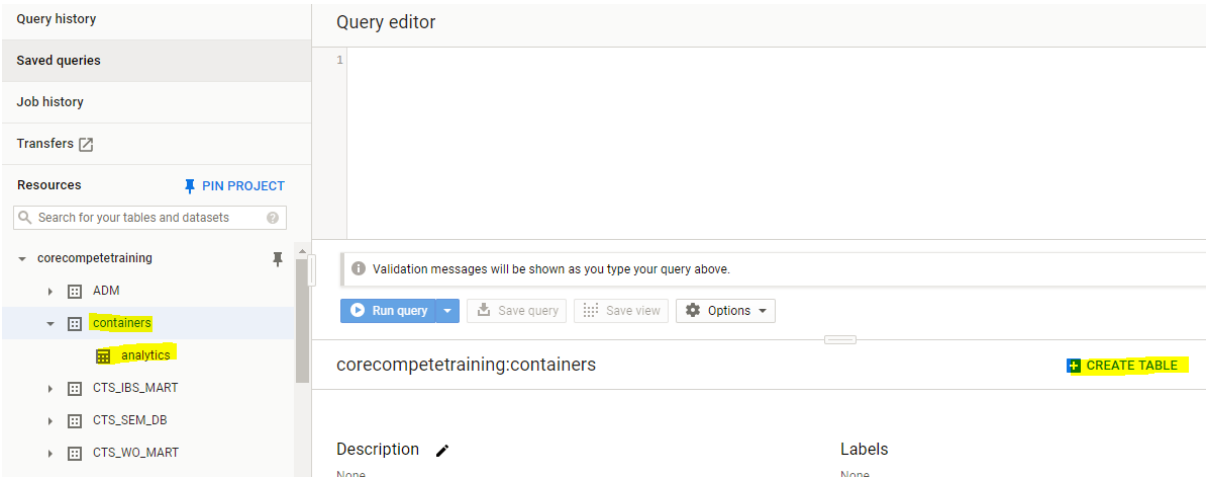


Figure 20. Check table in BigQuery

LOGIN TO SAS® STUDIO

- http://<hostname_or_ip>:38080 (e.g. http://sgf-2019.corecompete.com:38080)
- Login with user id and password provided while creating the docker image.

RUN THE BELOW QUERY IN SAS® STUDIO OR SAS® FOUNDATION TO GET THE DATA

```
proc sql;

connect to odbc (DATASRC="googlebq");

create table work.test as select * from connection to odbc (select
country.country_code AS Country,

       population.country_name AS CName,

       country.short_name AS SName


  FROM `bigquery-public-data.world_bank_health_population.country_summary` AS
country

  JOIN `bigquery-public-
data.world_bank_health_population.health_nutrition_population` AS population

  ON country.latest_trade_data = population.year where population.year=2015
order by population.year);

quit;


proc freq data=work.test;

   tables Country CName SName / out=FreqCount outexpect sparse;

   title 'World Bank health Population';
```

```
run;
```

```
LIBNAME gcplib ODBC DATASRC=googlebq user="****" password="****";

proc datasets lib=gcplib;run;
```

CONCLUSION

As demonstrated in this example, it is possible to configure SAS® with Google BigQuery in cloud native environment, which can be used as SAS® Access to BigQuery in containerized SAS Application to help SAS® programmers to use the analytics feature of SAS® with Google BigQuery.

REFERENCES

SAS® 9.4 and Container Technology: Build and Run a Container. SAS® Institute Inc., Cary, NC Available at https://documentation.sas.com/?docsetId=containers&docsetTarget=n133nr0ok71e5pn1oy96124cg1iz.htm&docsetVersion=9.4&locale=en

CONTACT INFORMATION

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

Sanket Mitra
sanket.mitra@corecompete.com

Srivalli Avadhanula
srivalli.avadhanula@corecompete.com

Fahad Ali
fahad.ali@corecompete.com