SAS® GLOBAL FORUM 2021

Paper 1062-2021

SAS Deployment on Google Cloud Tips/Tricks Including BigQuery Integration

Ande Stelk, Google

ABSTRACT

A technical discussion on how to best deploy SAS in GCP touching on Google Compute Engine, Storage and other configuration tips and tricks would be included. We will also cover integration with BigQuery using SAS Access to BigQuery. Viya v3 and v9 platforms would be referenced. This would be a specific presentation to GCP

INTRODUCTION

More and more businesses are moving to the cloud and one of the many reasons they choose Google Cloud are the industry leading data platforms such as Biq Query. The purpose of this paper is to provide a high-level overview of how to migrate SAS onto Google Cloud Platform (GCP) from on-premise. As these efforts usually include data platform migrations, general best practices for connecting to BigQuery (BQ) are also discussed.

Typically GCP is leveraged as Infrastructure As A Service (IAAS) for SAS (version 9.4 or Viya version 3.5) using Google Compute Engine (GCE) server instances. For SAS Viya 4.0, Google Kubernetes Engine (GKE) and Docker containers can be leveraged however those deployments are not in the scope of this paper.

Moving SAS into the cloud has many advantages such as allowing the application to be closer to the data source(s), avoids any applicable cloud egress charges, and supports the general IT strategy to move away from proprietary data centers.

CONSIDERATIONS

Moving to GCP has many many advantages however there are some considerations to be made.

PERFORMANCE

The highly virtualized infrastructure of any public cloud can impact query performance. If SQL pass-through or SAS Accelerator solutions are currently leveraged on premise that will no longer be in GCP. I highly recommend defining specific Service Level Agreements (SLA's) between analyst teams and IT related to the business reasons queries must complete in a specific timeframe, conduct testing and contingency plans to ensure they do. Don't wait until there are issues at the end of the migration but rather start these discussions proactively. Different storage configurations, use of in-memory technologies or a blend of SAS an GCP services such as Tensor Flow or ML on BQ are all options to alleviate any performance bottlenecks.

LICENSING AND ENVIRONMENT CONSIDERATIONS

Ensure you have contacted both your SAS and Google Cloud representatives. They will be happy to partner with you to design the best architecture. Some recommended discussion points would be;

- Are there SAS licensing implications as a result of a move to GCP?
  - SAS would be installed in a bring-your-own-license (byol) capacity
- SAS software upgrades/re-configurations may be required to be compatible with specific GCP products (e.g., SAS access engines/connectors to GCP data sources)
  - Refer to support documentation at SAS Install Center
- Will the technology, data and SAS use cases change as a result of the move to GCP driving an upgrade or need for larger/smaller SAS footprint?
- If you are planning to do a "lift and shift" of an existing on premise environment to GCE without changing SAS platforms or data volumes, this is a straightforward mapping exercise to replicate the existing infrastructure in Google Cloud
- If the data volumes, structures or SAS platforms (e.g. v9.x to Viya) then request SAS generate an Enterprise Excellence Center (EEC) environment sizing to be completed. SAS offers this to all customers at no cost and SAS will then make GCE instance recommendations based on variables defined from a questionnaire
  - Storage volumes needed (local and persistent) vary and using the existing SAS environment storage sizes to model from is recommended
  - Don't forget to include storage for any long term sas data set (.dat) files that are intended for long term storage. These can be stored on persistent volumes or in Google Cloud Storage (GCS) buckets depending on the specific requirements

Notes:

SAS recommends the data it consumes (e.g., BigQuery, Cloud Storage, DataProc) be located in the same region/zone(s) as the GCE instances SAS is installed upon.

SAS GRID is not generally deployed on GCP - SAS Grid is from the v9.x platform and allows worloads to be distributed among multiple compute servers leveraging a shared file system. More information below

SAS PERFORMANCE REQUIREMENTS VARY WITH EACH DEPLOYMENT

- SAS publishes a guideline of 125 MB/s per physical core of IO throughput. However, this is highly variable depending on the specific deployments and workloads (for example, basic queries vs. advanced analytics)
- Work internally and with SAS and GCP representatives to define SLAs, types of SAS workloads being deployed, and potential test cases.
- *This guide references high-level architectures for SAS Viya and Version 9 platforms. Please partner with your GCP and SAS engineers for detailed deployment frameworks.*
- *Specific system requirements for SAS Viya and Version 9 platforms can be found at https://support.sas.com/en/documentation/system-requirements.html*.

SOME ARCHITECTURE SPECIFICATIONS/CONSIDERATIONS

GENERAL INFORMATION

- As of the time of this writing, SAS Viya requires fixed host names *and* fixed IP addresses which can be tricky in a cloud environment
    - These requirements vary for specific configurations so check with your SAS engineer to confirm
- SAS Version 9 and Viya Version 3 are supported on a variety of Linux and Windows o/s.
    - RHEL is commonly used operating system for SAS Viya and Version 9 in GCP
    - CentOS and Ubuntu may be used but could be considered "alternate operating systems" by SAS for virtual machine deployments https://support.sas.com/kb/43/233.html.

INSTANCE TYPES

- N# Standard series on GCE for Version 9 Compute and Metadata.
- N# High Memory for SAS Version 9 Miditier node(s).
- N# High Memory series on GCE for SAS Viya (Controller Worker and Microservices nodes).
- Larger sizes, such as 32 vCPU, are generally recommended because they have higher network bandwidth (10GB).
- Skylake or higher (Cascade Lake) processors.
- Different instance types and sizes may be acceptable - the above are general recommendations.

STORAGE

- SAS has 3 considerations as it relates to data storage:

1. Source data can come from a variety of GCP services (for example, BigQuery) These should be defined and part of the infrastructure planning
2. SAS data sets (.dat or .hdat) extensions are analytical data files and persist over time and are typically contained in the SASDATA file system. These can be stored within Cloud Storage buckets and/or Persistent Dis Volumes. Local storage should not be used as it is erased upon instance restart and these files will be lost
3. SAS also utilizes temporary storage caches for SAS WORK, and UTILOC file systems for SAS Version 9 and CAS_DISK_CACHE for SAS Viya.Due to the heavy read/write requirements for this, as it relates to computation, SSD volumes are required.
    1. Persistent SSD can be utilized
    2. Local SSD volumes will result in highest performance. However, use of these in GCP add complexity as these instances cannot be stopped and started. Refer to https://cloud.google.com/compute/docs/disks/local-ssd
    3. Additional scripting (example below) is required
    4. The above additional complexity, cost, and performance requirements need to be made based on the specific installation
    5. A general recommendation is to start with persistent SSD volumes and if performance improvements are needed convert to Local SSD

ADDITIONAL STORAGE CONSIDERATIONS

- XFS is recommended for all SAS storage configurations
- Due to GCP Persistent Storage configurations, multiple volumes, striping, and RAID settings do not apply as it relates to performance. Choose the highest performing and volume size required by the project
- For optimal performance of Local SSD, a minimum of 4 volumes (striped and set as RAID 0) are recommended. Another performance increase can be found with quantities 16 and 24 drives (see comparisons at https://cloud.google.com/compute/docs/disks/local-ssd).
- Local SSD usage require scripting to allow for key files to persist and an instance to be restarted in an outage situation similar to the below:

```
#!/usr/bin/env bash
PROJECT_ID=<YOUR PROJECT ID>
CONF_SCRIPT=gs://<YOUR BUCKET>/install.sh
REGION=us-central1
ZONE=${REGION}-a
SUBNET=default
IP_ADDRESS=<my internal IP>

MACHINE_TYPE=n1-highmem-32
INSTANCE_NAME=sas-perf-testing-${MACHINE_TYPE}-local24

# reserve private/internal IP address
gcloud compute addresses create my-vm-ip-address \
 --region ${REGION} --subnet ${SUBNET} --addresses ${IP_ADDRESS}

gcloud beta compute --project=${PROJECT_ID} instances create
${INSTANCE_NAME} \
--zone=${ZONE} --machine-type=${MACHINE_TYPE} \
--scopes=storage-ro \
--metadata startup-script-url=${CONF_SCRIPT} \
--image=rhel-7-v20200403 --image-project=rhel-cloud \
--boot-disk-size=20GB --boot-disk-type=pd-ssd \
--boot-disk-device-name=${INSTANCE_NAME}-boot-disk \
--private-network-ip=my-vm-ip-address \
--min-cpu-platform="Intel Skylake" \
--local-ssd=interface=NVME --local-ssd=interface=NVME --local-
ssd=interface=NVME --local-ssd=interface=NVME --local-ssd=interface=NVME --
local-ssd=interface=NVME
--local-ssd=interface=NVME --local-ssd=interface=NVME --local-
ssd=interface=NVME --local-ssd=interface=NVME --local-ssd=interface=NVME --
local-ssd=interface=NVME --local-ssd=interface=NVME --local-ssd=interface=NVME
--local-ssd=interface=NVME --local-ssd=interface=NVME --local-
ssd=interface=NVME --local-ssd=interface=NVME --local-ssd=interface=NVME --
local-ssd=interface=NVME --local-ssd=interface=NVME --local-ssd=interface=NVME
--local-ssd=interface=NVME --local-ssd=interface=NVME --reservation-affinity=any \
```

Here is a sample startup script:

```
#!/usr/bin/env bash
CONF_BUCKET=gs://<YOUR BUCKET>
```

```
cd /tmp/

# get all required packages
sudo yum install bc time wget expect -y
sudo wget http://ftp.sas.com/techsup/download/ts-tools/external/SASTSST_UNIX_installation.sh -O /tmp/SASTSST_UNIX_installation.sh
sudo wget http://mirror.centos.org/centos/6/os/x86_64/Packages/xfsprogs-3.1.1-20.el6.x86_64.rpm -O /tmp/xfsprogs-3.1.1-20.el6.x86_64.rpm
sudo chmod +x /tmp/SASTSST_UNIX_installation.sh
sudo yum localinstall /tmp/xfsprogs-3.1.1-20.el6.x86_64.rpm  -y
sudo gsutil cp ${CONF_BUCKET}/sas_tools.tar /tmp/
sudo tar xvf sas_tools.tar

echo "Check for nvme"
lsblk|grep nvme
if [[ $? -eq 0 ]]
 then
      echo "Setting up nvm array"
      nvm_arr_size=`lsblk|grep nvme|wc -l`
      nvm_arr=`for drive in $(seq 1 $nvm_arr_size); do printf "/dev/nvme0n$drive " ; done`
      sudo mdadm --create /dev/md0 --level=0 --raid-devices=$nvm_arr_size $nvm_arr

      echo "Creating xfs fs for local-ssd"
      sudo mkfs.xfs /dev/md0 ; sudo mkdir /mnt/sasfs ; sudo mount /dev/md0 /mnt/sasfs
   else
      echo "Creating xfs fs for pd-ssd"
      sudo mkfs.xfs /dev/sdb ; sudo mkdir /mnt/sasfs ; sudo mount /dev/sdb /mnt/sasfs

fi

if [[ -d /mnt/sasfs ]]
  then
      sudo chmod a+w /mnt/sasfs
      sudo /tmp/sas_tools/rhel_iotest.sh -t /mnt/sasfs &
fi
```

SAS GRID DEPLOYMENTS

- SAS Grid is a Version 9 technology designed to distribute workloads among multiple compute instances
- SAS Grid deployments would follow the instance and storage recommendations in this paper
- Unlike the more modern SAS Viya platform, SAS Grid requires a robust shared file system outside of the SAS platform to distribute workloads to multiple server instances
    - At this time, GCP does not offer a shared file system service that meets SAS Grid requirements

    - Google Cloud Filestore may be utilized, depending on specific performance requirements for a specific configuration but this would be outside the normal Grid deployment needs
    - A 3rd party solution is generally required, such as DDN Lustre or IBM Spectrum Scale
    - Please partner with your SAS engineering team and SAS partners to identify the appropriate shared file deployment for your specific project
    - A good reference can be found at Shared File Systems: Determining the Best Choice for Your Distributed SAS® Foundation Applications
- Finally, it is this author's opinion any SAS v9 platform migration to GCP should involve consideration of a move to SAS Viya platform due to its ability to natively distribute workloads, in-memory computation, overall cloud friendly architecture and REST API capabilities

DISASTER RECOVERY AND HIGH AVAILABILITY

- Work with your SAS and GCP technical team to define the best solutions for your specific deployment needs
- SAS metadata and midtier for Version 9, as well as microservices for Viya, can be clustered
- SAS Version 9 Compute tier cannot distribute among multiple instances without the deployment of specific SAS Grid solutions
- SAS Viya can have multiple worker nodes and be deployed with a primary (active) and secondary (passive) controller configuration
    - This will require the additions of a shared file system for both primary and secondary controllers to utilize
    - NFS or Cloud Filestore are appropriate for single-zone deployments
    - For multi-zone deployments, partner with your Google and SAS teams for recommendations
- For highest performance, SAS recommends deployments be contained within a single zone. While a multi-zone deployment is possible in GCP specifically, and testing has shown little impact on overall performance, It is important to note that in many scenarios, any running SAS jobs will be lost and need to be restarted. This may negate the benefit of a multi-zone deployment. Please partner with GCP and SAS teams to discuss options for your specific deployment
- SAS does NOT recommend a single environment installed across multiple regions
- For multi-zone/multi-region environments, SAS recommends duplication of the entire SAS environment in each zone/region with a load balancer added. However, with specific SAS solutions having varying requirements (for example, fixed IP and/or host names, LDAP integrations) these installations can become increasingly complex. A detailed review of recovery requirements and architecture options should be considered with customer, Google, and SAS engineering teams

SAS INTEGRATION WITH BIGQUERY

SAS Access Engine / Connectors for data sources may be licensed individually and to connect to BQ, SAS Access for ODBC or SAS Access for BQ is required. For most customers, the latest version of SAS Access for BigQuery is preferred over SAS Access for ODBC. The latter requires additional steps to install and configure the ODBC driver BigQuery, and this

driver is available for download from the GCP ODBC and JDBC drivers for BigQuery page. However, if other databases such as Cloud SQL will be used with SAS in addition to BigQuery, using the SAS Access for ODBC connector might be a better fit.

Write versus read requirements, table width (e.g., number of columns), and column width (string length in particular) may necessitate different CAS configuration optimizations. The more data analyzed in memory, the more robust the cluster and larger the CAS cache needs to be. In general, the following are recommended best practices when using SAS Studio or SAS Visual Analytics with BigQuery, but specific customer requirements should always be taken into consideration as well.

READING FROM BIGQUERY

- If BigQuery is being used with a new SAS deployment, consider using explicit queries to make the most of BigQuery features and push-down processing
- If an existing SAS deployment wants to reuse past caslibs and SAS code, consider using implicit queries, but be aware that certain use cases might not be able to leverage all BigQuery features or push-down the most processing to BigQuery
- A "response too large to return" error from BigQuery means that a resultset size exceeded 128MB (compressed), and the connector needs to have large BigQuery results support enabled. Earlier versions of SAS Access for BigQuery do not have such functionality, but the ODBC driver supports this feature by setting **AllowLargeResults=1** in the ODBC configuration
- Add partitions to BigQuery tables for better performance and reduced bytes scanned, but note that SAS tools do not visually denote the difference between a partitioned table and an unpartitioned table
- Flatten nested BigQuery columns using the BigQuery **UNNEST** function before importing a table
- Referencing partitioned table pseudo columns in a query will fail unless used in an explicit sql pass-through query.
- Use the caslib **READBUFF** option to increase read performance
- Use the caslib **compress=yes** option when importing data with to conserve source disk space while gaining modest performance improvements
- *SAS Visual Analytics:* Use a data filter instead of a report filter when working with in-memory tables that have a larger number of rows
- *SAS Visual Analytics:* For numeric data types, double-quote values in a table import filter (e.g., "geo_id"="69") instead of single-quoting or leaving unquoted to ensure all expected data is included

WRITING TO BIGQUERY

- Use the caslib **INSERTBUFF** option to avoid BigQuery DML-related limits
- Use the caslib **bulkload=yes** option to increase write performance of large data transfers into BigQuery

IN CONCLUSION

I hope you found this information useful and many thanks to all the SAS and GCP teams who contributed to it. SAS and GCP technologies and services are fast changing so always verify information in this article with the most current documentation available from SAS and Google Cloud

CONTACT INFORMATION

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

Ande Stelk
andestelk@google.com