

In a Matter of Seconds...

By: Shaswat Priyadarshi and Gopal Yadav, Stratacent

SAS Software as it is used in a traditional on-premise environment usually includes a shared server environment with multiple users/groups accessing and submitting jobs. As a shared SAS environment, this creates multiple problems both for operations, development and modeling community.

Some of the problems are as follows:

1. Initial Setup: Initial Setup takes weeks of capacity planning, procurement, software installation, and testing. This effort can take weeks/months to complete
2. Upgrades and Change management: Any changes to this environment must go through careful planning with all users involved in order to avoid disruption
3. As a shared environment, a single rogue job has the capability to completely hog the system resources and render it useless for everyone else.
4. The shared environment does not lend themselves for any chargeback model for IT

There is a more efficient way to provide platforms for analytical uses to the right individuals. With the use of SAS in containers, now IT can provide SAS POAC or SAS 9.4 in containers in public or private cloud or even in on-prem. Containers make it easier to deploy anywhere, scale up or down and can use automation to spin up and down. Customers can use platforms such as Kubernetes, Pivotal, OpenShift, Dockers, etc for this deployment. In this demo, we show how we can achieve the same use case mentioned above combining SAS's cloud offering, SAS Viya with containers and GCP in a few minutes.

Introduction:

The scope of this demo is to showcase how a Programming-only SAS Viya container image can be deployed on a GCP within minutes for ad hoc research, for temporary developmental purposes or batch production workloads. This demo also shows how we can scale up the resources for the deployed environment on the fly to manage increased workloads. The audience for this demo is assumed to have knowledge of docker container commands, and some level of exposure to SAS Viya and Google Cloud Platform.

As part of the demo, the following will be covered:

1. Spin up a standard GCP compute instance with required CPUs, Memory and storage.
2. Set up a docker environment for containers
3. Download and deploy the predefined SAS Viya container image to the GCP instance.
4. Launch SAS Studio and run sample programs.
5. Demonstrate how to increase the resources of GCP, without deleting the existing container image and building a new one.

Pre-requisites:

- Virtual hardware requirements for Docker host machine (can only be a Linux machine. Windows is not supported)
 - Minimum 30 GB disk space
 - Minimum of 4 vCPUs
 - Minimum 64 GB RAM



- Runtime environment (Docker)
 - Minimum docker version - 17.05.0
 - JRE should be available in the deployment environment (SAS provides OpenJDKJRE in the container.
 - For the basic POAC container, no additional data source is required however we can add them as an additional layer on top of our basic image.4. All the docker processes run as a user with root privileges. The user performing the deployment should, therefore, have sudo privileges.
 - For the basic POAC container, initially, we just get the sasdemo user account for both CAS Admin and SAS Studio, however, we can add more users as per the requirement.
 - There should be appropriate access to launch a GCP instance.

Pre-Deployment tasks:

1. Obtain the SAS Deployment Data zip file. You should be able to request one from your SAS Account Executive.
2. Download the mirror repository tool and use it to get a list of SAS POAC images (<https://support.sas.com/en/documentation/install-center/viya/deployment-tools/35/mirror-manager.html>)
3. Pull the latest POAC image to the Docker host from the SAS hosted repository (use docker pull command).
4. Create the required directories
 - sasinside
 - sasdemo
 - cas

The required user account “sasdemo” is pre-configured with the SAS provided Viya container image having appropriate privileges.

Deployment Steps:

1. (Optional) - This image can be pushed to google provided container registry/ public registry for re-use and can be used for multiple deployments.
2. Use SAS provided launcher script (launchsas.sh) which comes with the necessary arguments for docker to deploy the SAS Viya container image.

Validate Deployment:

1. Check if the container is running as expected (using docker ps command). Launch SAS Studio with the appropriate URL.

```
[sas@poachost sasdempoc]$ docker images
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
sasdempoc           v1          fc51d598fd5e  3 months ago  13.8GB
sas.sas.download/va-125-x64_redhat_linux_7-docker/sas-viya-programming  3.5.5-20191107.1573147361701  fc51d598fd5e  3 months ago  13.8GB
hello-world         latest      fce289e99eb9  14 months ago  1.84kB
gcr.io/stratacentlabs-dev/hello-world  latest      fce289e99eb9  14 months ago  1.84kB
[sas@poachost sasdempoc]$ ./launchsas.sh
e284fb5c8ad3df1e1df0601c0365a531a0b5ead041911f091c986f5a36031ad6
[sas@poachost sasdempoc]$ docker ps
CONTAINER ID        IMAGE          COMMAND                  CREATED           STATUS          PORTS
e284fb5c8ad3      sasdempoc:v1  "/usr/local/bin/tini..."  3 seconds ago    Up 3 seconds   0.0.0.0:5570->5570/tcp, 0.0.0.0:8081->80/tcp, 0.0.0.0:32791->0.0.0.0:32789->8777/tcp
0.0.0.0:32789->8777/tcp  sas-programming
```

Fig 1: List of docker images available on the host machine.

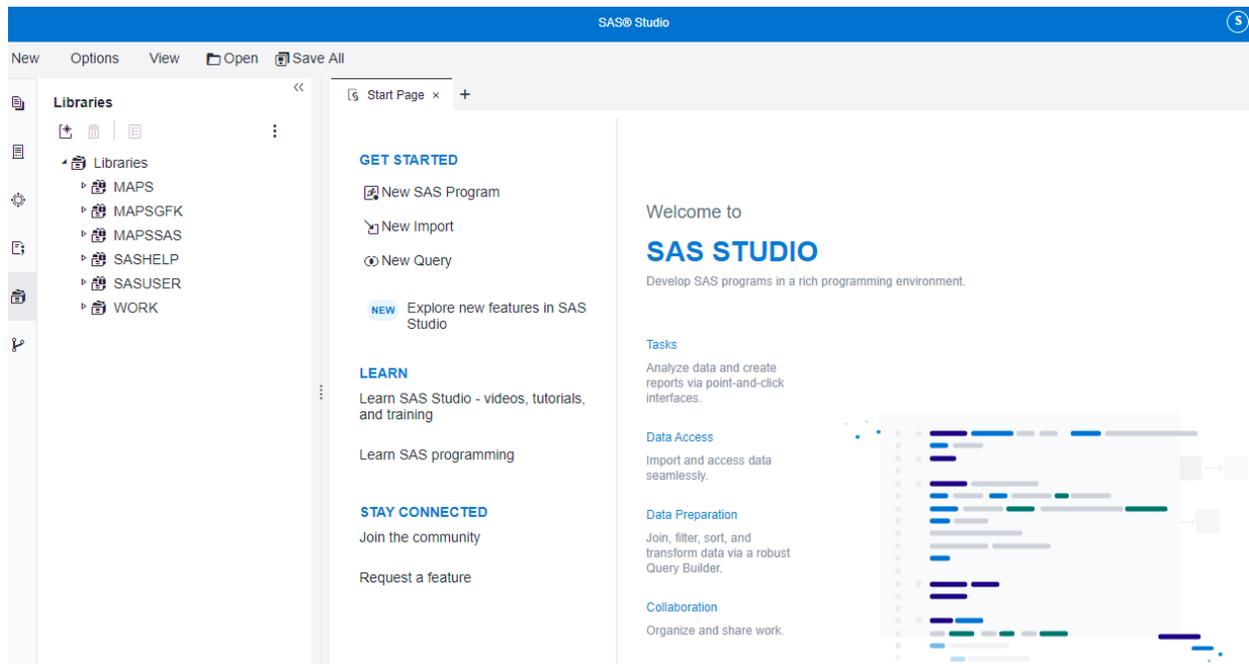


Fig 2: SAS Studio Home page as launched from SAS Viya Container.

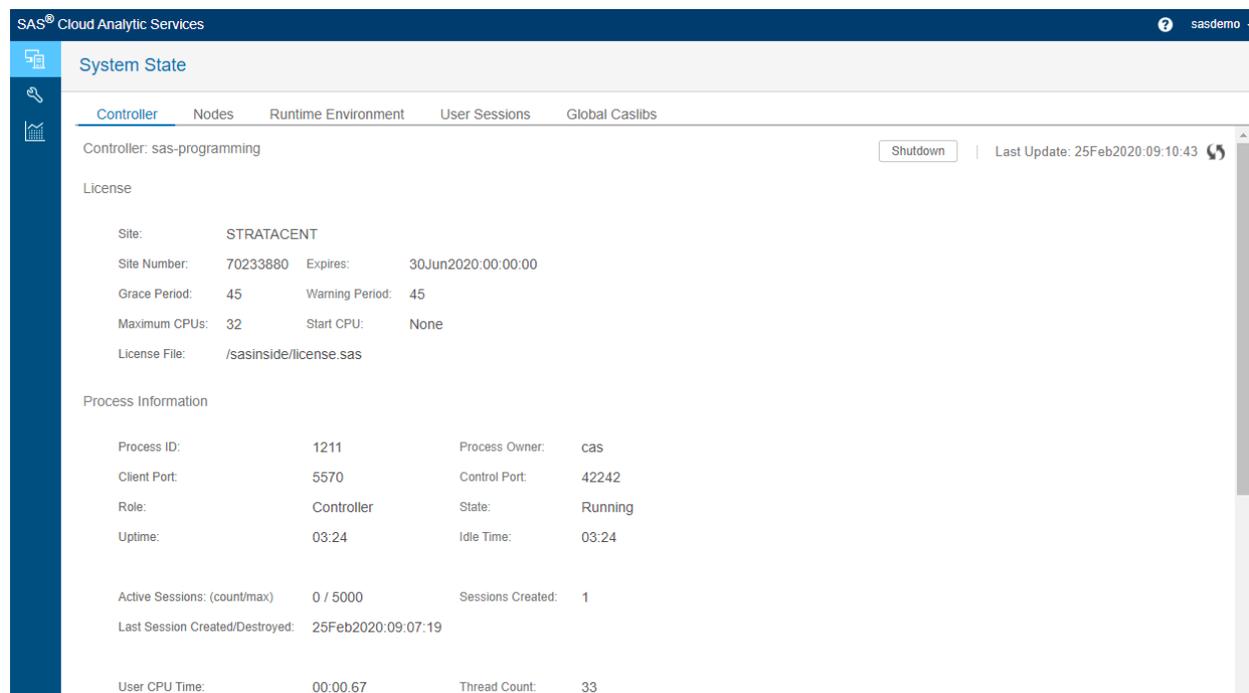


Fig 3: CAS Administrator page

Steps to Scale Up:

The existing container's resources can be scaled up very easily by following the below steps.

1. Using SSH terminal, stop the container using **docker stop <container name>** command
2. Go to GCP console —> Select VM instance —> Stop it —> Click Edit
3. Select the desired machine type from the drop-down list (Highlighted instance type in the screenshot below has been selected as a reference only).
4. Re-launch the instance.
5. Using SSH terminal, start the container image as mentioned in step 2 of the **deployment steps** section (launchsas.sh).

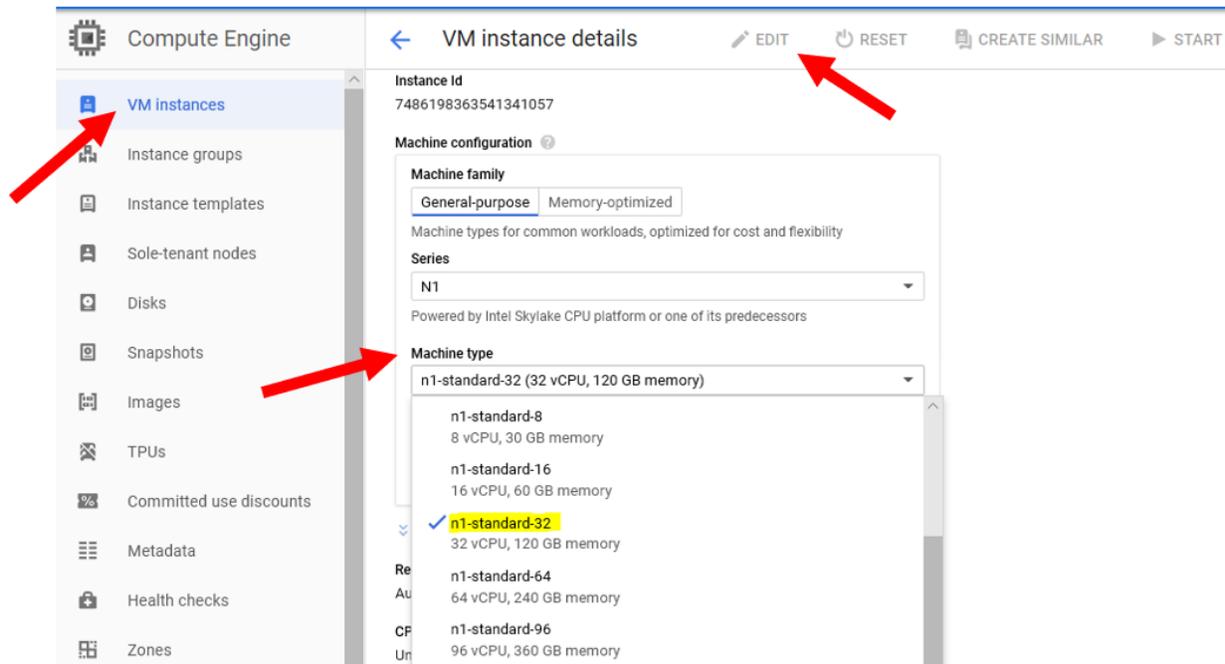


Fig 4: GCP Console to scale up the existing container instance

Conclusion:

SAS in containers provides an excellent way to provide an environment to the user without any sharing, with the ability to scale up or down for capacity and with the latest features. When combined with CI/CD pipelines, the changes made by the users can be easily captured for future reference. Public cloud platforms such as GCP, AWS, and Azure provide excellent platforms to allow IT organizations to manage these containers, create self-service automation to start and stop these containers. Proper organization and Billing structure can allow IT organizations to even provide the chargeback reports to the end-users for resource usage. The same container recipe can also be used with other analytic platforms such as Domino Data and Data Bricks.

References:

[SAS® Viya® 3.5 for Containers: Deployment Guide](#)