TAP TO GO BACK TO KIOSK MENU

SAS®
GLOBAL FORUM 2020

MARCH 29 – APRIL 1
WASHINGTON, DC

USERS PROGRAM

SAS® VIYA: AUTOSCALING MAS NODES USING COMPUTE VIRTUAL MACHINES ON GOOGLE CLOUD PLATFORM

TAP TO GO BACK TO KIOSK MENU

Suvadeep Chatterjee and Sounak Nag

Core Compete

Abstract
Introduction
Methods
Architecture and Considerations
Changes to ACLFiles and Considerations While Taking the MAS Image
Considerations for the start and stop script
Conclusion

With the advent of the cloud-ready SAS® Viya 3.4, there is a demand for scalability for the Viya services on Cloud. SAS® Viya includes a number of different microservices to support a scalable and elastic architecture. This e-poster deals specifically on how to separate SAS® Micro Analytic Service (MAS) on a separate node and then auto-scale based on the CPU usage on that node using Google Cloud Platform's managed instance groups and other GCP native features. We have tested this and validated the functionality by publishing models with auto scaled MAS nodes. This presentation guides you through the process and how to use Google cloud-native features to achieve this.

Suvadeep Chatterjee

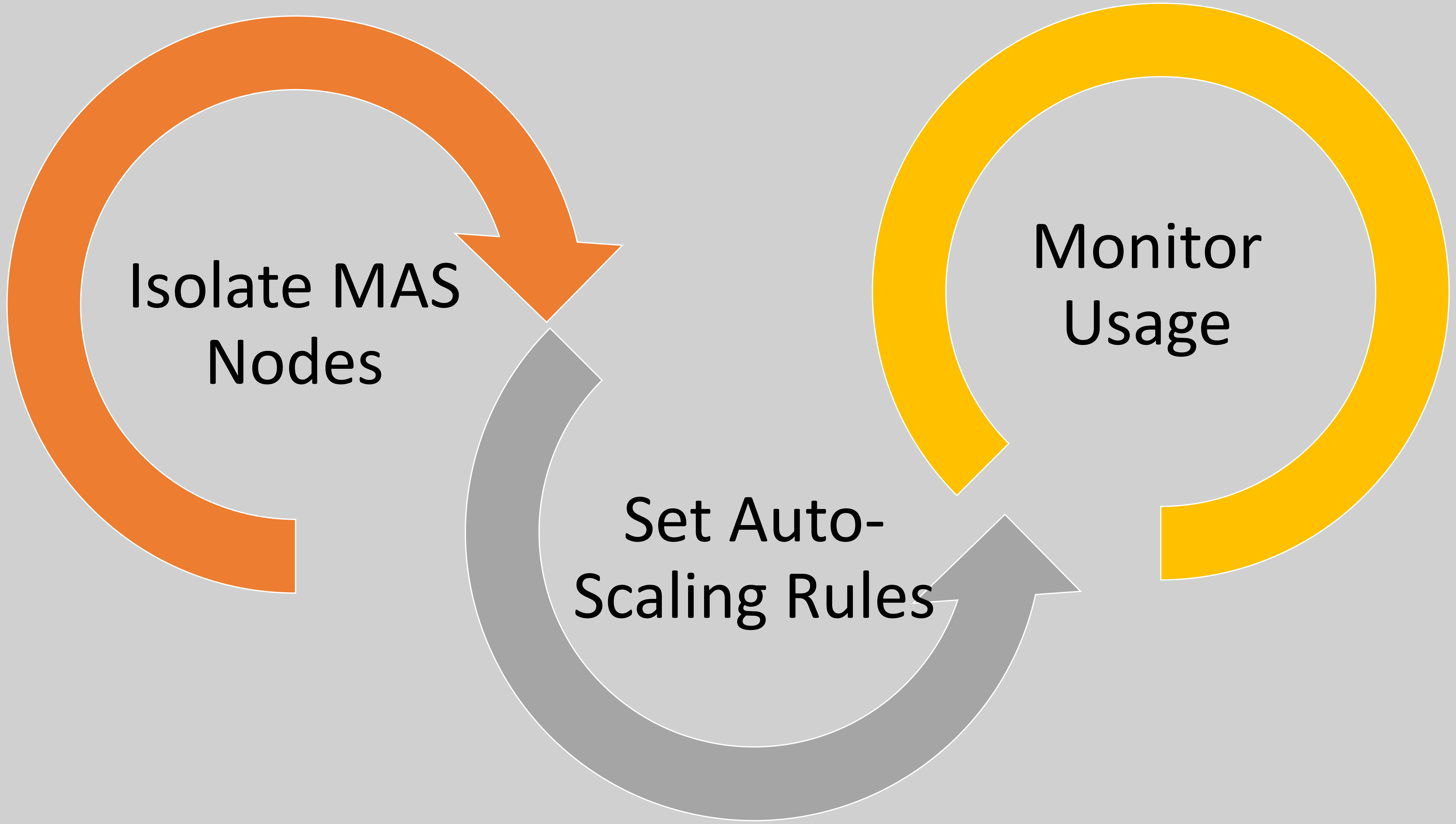
SAS® VIYA: AUTOSCALING MAS NODES USING COMPUTE VIRTUAL MACHINES ON GOOGLE CLOUD PLATFORM

Suvadeep Chatterjee and Sounak Nag

Core Compete

Introduction

For Google Cloud Platform (GCP), we will be using GCP native features like Managed Instance Groups, Instance Templates with start and stop scripts, Images, Snapshots, Cloud Storage, Cloud Filestore to auto-scale SAS® Viya 3.4 MAS nodes. SAS® Micro Analytic Service is a memory-resident, high-performance program execution service. Auto-scaling can be provided for execution of SAS models that require high memory utilization on MAS nodes in order to ensure consistent performance.



SAS® VIYA: AUTOSCALING MAS NODES USING COMPUTE VIRTUAL MACHINES ON GOOGLE CLOUD PLATFORM

Suvadeep Chatterjee and Sounak Nag

Core Compete

Abstract

Introduction

Methods

Architecture and Considerations

Changes to ACLFiles and Considerations While Taking the MAS Image

Considerations for the start and stop script

Conclusion

Google Cloud

Agenda

- Architecture and Considerations.
- Changes to ACL files and considerations while taking the MAS image.
- Things to consider for the start-up script
- Things to consider for the stop script

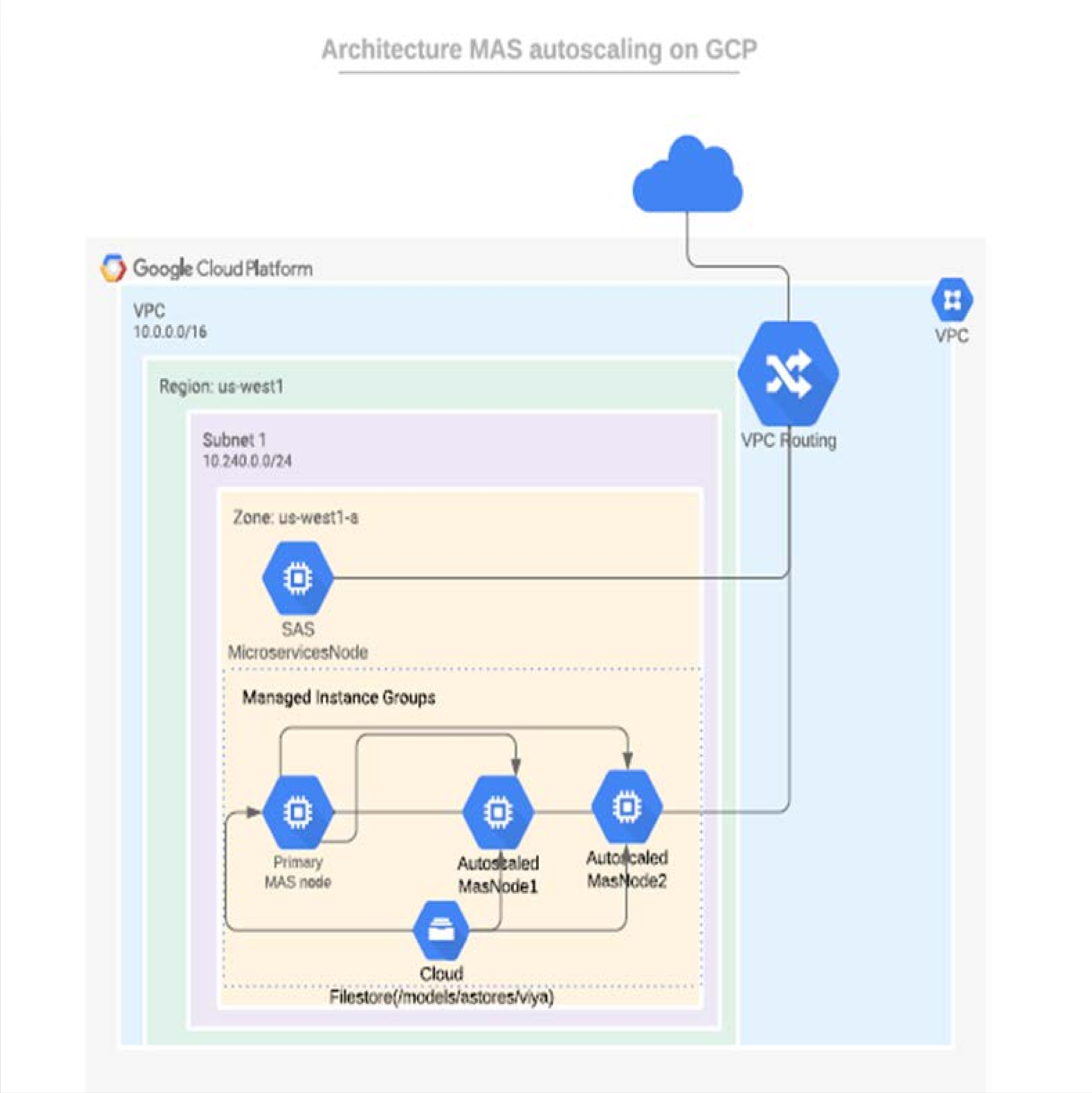
Components Used

- Google Compute Engine VMs
- SSD Persistent Disks
- Filestore
- Images
- Google Managed Instance Groups
- Instance templates
- Cloud Storage

SAS Viya

ADAPTIVE UNIFIED OPEN POWERFUL

High Level Architecture Diagram



Suvadeep Chatterjee and Sounak Nag

Core Compete

Abstract
Introduction
Methods
Architecture and Considerations
Changes to ACLFiles and Considerations While Taking the MAS Image
Considerations for the start and stop script
Conclusion

Architecture and Considerations

In this scenario, we have placed all the stateful microservices of SAS® Viya on one machine, the stateless microservices except MAS on another machine and just separated MAS on a separate node making it a 3-machine deployment. Note here that we can also separate the other microservices on other machines as per the requirement.

The major GCP components used are:

- Google Compute Engine for the SAS® Viya VMs.
- Managed Instance group that auto-scales MAS nodes based on CPU utilization and minimum number of instances at 1.
- A Cloud Filestore for the /models/astores/viya. Alternatively, Cloud Storage can be used but then the authentication will not be from the server level but maintained from the GCP side.

A few considerations for this architecture:

- An instance of Cloud Filestore is available in only one Google Cloud zone and does not include any way to failover if the zone where it resides becomes unavailable. That means, should there be an outage, users can expect downtime. Backups would need to be performed by the user, as Cloud Filestore has no snapshot feature currently. Back up must be done manually by periodically copying content to a bucket using gsutil.
- In the case of a shared VPC, the host project must be involved in the creation of Cloud Filestore instances. Costs for Cloud Filestore instances are charged to the host project instead of the service projects that use them. As an alternative, a Google SSD disk can be used, which can be multi-regional.

Abstract
Introduction
Methods
Architecture and Considerations
Changes to ACLFiles and Considerations While Taking the MAS Image
Considerations for the start and stop script
Conclusion

Changes to ACL Files and Considerations While Taking the MAS Image

Post deployment, we are making a few modifications to some of the ACL files. The purpose of doing so is to allow new nodes to be added during scale-up and then leave during the scale-down process. Now by default, this is blocked in SAS® Viya. So, we must edit the ACLs to suit our requirements. Along with those, there are considerations for taking the MAS node image. Here we discuss the challenges and how to proceed with those changes to mitigate those challenges.

Challenges we faced while trying to spin up new MAS images manually during the testing process:

- Every MAS node has a unique node-id which prevents the consul to start and be picked up by the consul head node. You will need to change this.
- It is better to stop SAS Services on the MAS node as there may be some leftover services files that will cause issues while starting the services.
- ACL files on the consul head node must be changed to allow the addition of new nodes. This file is named v1-consul-acl-rules-anonymous.txt on consul controller.
- Configuration files on MAS nodes must be changed to allow the MAS nodes to leave the cluster during the scale-down process. This file is named config-consul.json.

Abstract
Introduction
Methods
Architecture and Considerations
Changes to ACLFiles and Considerations While Taking the MAS Image
Considerations for the start and stop script
Conclusion

Considerations for the Start and Stop Script

Things to consider for the start-up script:

- Mount the required disks, cloud filestore and make the required permission changes. Create the folder path for /models/astores/viya.
- Replace the node-id from the image with a new one using the uuid feature, so the new consul node gets registered.
- Add the private ip, fqdn and alias name of the new node to the configuration server.
- Kill the existing start command of services which is executed automatically on system startup.

Things to consider for the stop script:

- Remove the entry from /etc/hosts file once the nodes are scaled-down.
- Stop services on the nodes which will be scaled down.

Note here that the start-up and stop script are both being set using the feature provided in instance templates in GCP. Hence this solution is entirely a GCP native one.

SAS ® VIYA: AUTOSCALING MAS NODES USING COMPUTE VIRTUAL MACHINES ON GOOGLE CLOUD PLATFORM

Suvadeep Chatterjee and Sounak Nag

Core Compete

Abstract
Introduction
Methods
Architecture and Considerations
Changes to ACLFiles and Considerations While Taking the MAS Image
Considerations for the start and stop script
Conclusion

Conclusion

- After the set up was done, we tested it in our architecture where there was one node with the microservices and the other one with MAS.
- This MAS node was part of the managed instance group. The scaling policy was based on CPU utilization of above 85%.
- We sent multiple requests to the MAS node using Apache JMETER to publish a sample decision flow to MAS to simulate the actual workload.
- We observed that based on the CPU utilization, new instances were launched as MAS nodes and the load was redistributed. We verified it using NMON.
- Once the load from JMETER decreased, the application scaled out.
- We have verified this for both SAS® Viya 3.4 and SAS® Viya 3.5.

References

- https://cloud.google.com/compute/docs/instance-groups/creating-groups-of-managed-instances
- https://cloud.google.com/compute/docs/autoscaler

SAS® GLOBAL FORUM 2020

USERS PROGRAM

MARCH 29 – APRIL 1 | WASHINGTON, DC | #SASGF