

Paper 4871-2020

## *Lessons Learned from Managing SAS® Viya® at Scale in the Cloud*

Jelle Daemen, Tom Dogger, SAASNOW

### ABSTRACT

Starting in 2015, SaasNow has been managing SAS® products in a private cloud in a unique monthly, fully self-service cloud model. Starting with SAS® vApp technology, SaasNow was one of the first providers to offer a true cloud experience for SAS® Visual Analytics. With the release of SAS Viya in 2016, SaasNow has been able to support many more SAS products, including SAS® Visual Data Mining and Machine Learning. Currently, all SAS Viya products and bundles can be automatically deployed on SaasNow within a few hours.

Over the years, we have deployed and managed hundreds of environments, ranging from small four core demo environments, to multi-server production setups of up to 128 cores with a wide variety of customer-specific customizations and integrations, like customized authentication (Kerberos or SSO integrations) and external data source connections (SQL and Apache Hadoop). We have developed best practices for sizing, deploying, updating, upgrading, and monitoring many instances of SAS Viya at the same time. This enables us to make SAS Viya available to our customers and partners within several hours and have new releases available within days after they have been released by SAS. In this way, we offer a true cloud experience for SAS Viya.

In this paper, we share the lessons learned from deploying and managing SAS Viya at scale.

### INTRODUCTION

Hosting multiple servers in the cloud having different kind of SAS Viya products at the same time is quite a complex thing to do, as the most important thing to do is making sure actions taken are as automated as possible. To be able to do this, you might need to use multiple tools or programming languages.

To explain the complexity and differences in lessons we learned and tools we use, we divided the activities in four different kinds of activities being: deployment, monitoring, upgrading and maintenance.

Whilst quick starts are available for deploying SAS Viya on multiple (public) cloud providers, these do not take into account the other aspects of managing a SAS Viya environment. After reading / attending this talk, we hope you have learned what kind of issues we encounter and how we dealt with a number of them.

## MAIN ACTIVITIES

When a customer requests a new server, an orchestration process is started through an application programming interface (API). A customer logs in to his SAASNOW environment and can request a new server. There are 3 important questions: the size of a server (small, medium or big usage), the SAS Viya products required (i.e. Visual Analytics, Visual Statistics, Visual Data Mining and Machine Learning) on the server. The third question is the preferred location if applicable, since for some data or in some countries data is required to stay in the same country as the company its address.

When the orchestration is done, the server will be added to the monitoring system. It will be updated or upgraded on a regular basis and we will perform maintenance to the server.

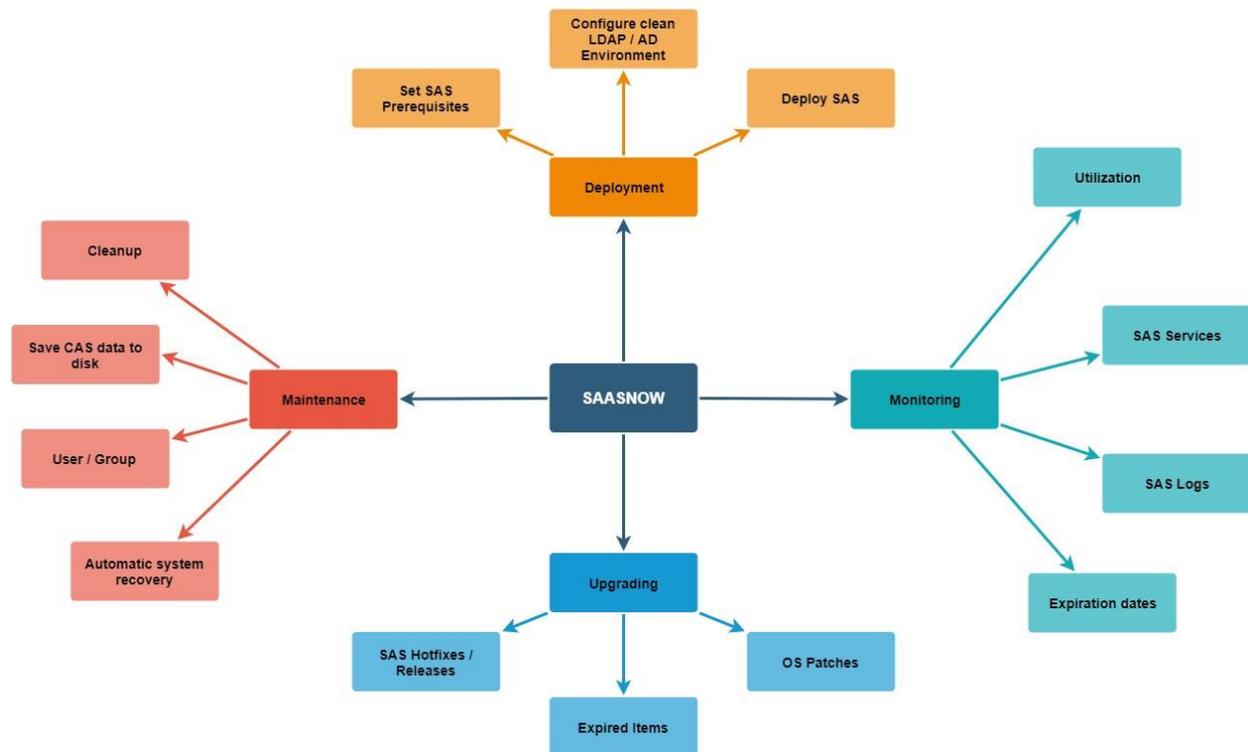


Figure 1: The main activities in a high over picture.

## DEPLOYMENT

When the questions required to create a new server (size, which SAS products and location) are answered, the API starts the process in creating a machine. It also makes sure the correct size of server is created and a Linux operating system is being installed.

After that there are some packages installed, and settings set, to make the SAS installation executes as smooth as possible. During this stage some playbook files are adjusted (i.e. sitedefault.yml). Playbook files are files used by ansible, during the installation of SAS Viya.

By default we also create an empty OpenLDAP <sup>(1)</sup> (Open Lightweight Directory Access Protocol) server at the same machine as the SAS installation. If the customer wants to use a different LDAP / AD, this is possible.

When the deployment is done, the customer gets an email that his server is ready to use. When this is done, the server will be joined to the monitoring system. It will be updated or upgraded periodically and our default maintenance steps will be executed.

## MONITORING

Our monitoring system is used to measure a number of components, for example metrics like CPU, memory or hard disk usage. Next to this, our monitoring system continuously monitors if all the SAS services are running. Furthermore checks are in place to monitor expiration dates in the SAS License or the SSL certificates. The monitoring tool can also check SAS logs. Although not all logs need to be processed, there are some standard logs being checked for specific fatal or error messages.

For basically all kind of errors or service down messages the system reports the issue to the support team and if a service is down and the cause of the issue is known, the services will automatically be tried to restored.

## UPDATING / UPGRADING

On a periodic basis OS patches are applied to the servers we manage. In some cases (i.e. security patches) patches are applied more often to prevent security issues. Also new hotfixes or complete new releases of the deployed SAS products are usually installed as soon as possible. For hotfixes there might be a delay, but new SAS releases are being tested and installed within a week. These patches are applied in consult with the customer at a day of their preference.

Furthermore if the monitoring tool reported issues with expiration dates, our scripts detect them, and try to fix the issue (if new licenses are provided).

## MAINTENANCE

In consult with customers, we can set some scripts to make sure their machine is functioning as smoothly as possible. We do have scripts to save CAS data to disk for backup reasons and we can run cleanup scripts on temporary SAS or CAS directories.

Another much requested feature is a synchronization between the different authorization systems in SAS Viya and match these to the operation system. We introduced the SaasNow Viya Authorization Sync Mechanism (VASM) to make sure the authorization for SAS Content, CAS and the OS are in sync.

## LESSONS LEARNED

### MANAGING MULTIPLE SERVERS

Managing multiple servers, whether it is local or in the cloud can be challenging, while trying to keep all servers at the same base, you also want to provide freedom to the users.

For our servers we want to be able to execute a number of things simultaneously, regardless of the SAS Products that are installed. All servers need to be monitored and they all need to be updated periodically.

The most important thing we discovered is to write scripts in a way to automate as much as possible, regarding the data on the server itself. You can check what SAS Services are installed on a specific server. Use that check, to check to see whether all services are up and running.

Obviously not everything can be the same on all servers, things like database connections are server specific. We chose to grant customers a specific location where they can upload their database information, so we can grant the SAS Viya environment to connect to these databases.

### MULTIPLE PROGRAMMING LANGUAGES

The four different activities described above, all have their own set of tools or programming languages to get the required actions done. Links to the tools described below can be found in the reference list at the bottom of this paper.

For the deployment activities the tools used are Red Hat Ansible<sup>(2)</sup>, Puppet<sup>(3)</sup> and GNU Bash (Unix shell) <sup>(4)</sup>.

For monitoring the tools used are Nagios<sup>(5)</sup> and Red Hat Ansible.

For upgrading use Red Hat Ansible and GNU Bash (Unix shell) scripting.

For maintenance the main tool is GNU Bash (Unix shell).

The main advantage of using different tools, is you are allowed to use the strengths of these tools to what you want to do for a specific action.

The main disadvantage of using multiple tools is the diversity of the development team. Since the members of the development team need to have at least a basic level of knowledge of all tools, and they need to have some specific skills for the specific tools they use in their daily tasks.

New scripts that are created, need to be documented well and certain choices made might need to be explained, so other people can understand why a certain choice was made, and how to proceed from that decision onward.

### NEW SAS VIYA RELEASES

When SAS provides a new release of Viya, whether this is a complete upgrade to a new maintenance level of Viya, a more simple update of the current Viya release or just hotfixes, the installation needs to be thoroughly tested. Most common changes are documented in the SAS Deployment Guides <sup>(6)</sup>, Occasionally some new features are not very well documented, resulting in faulty installations or unstable runtime environments. Especially the combination of updates combined with OS updates or third-party tools like Ansible are a bit tricky sometimes.

Next to this, we always have to check internally when communicating with SAS Technical Support. If advises are given for a specific customer, these recommendations might be applicable to other instances too, in some cases requiring our scripts to be adjusted or extended.

## RESOURCES USED BY MONITORING SYSTEMS

When using monitoring tools, it is important to keep track of the resources used by the monitoring tool. The monitoring does need to run continually, and as real time as possible, so potential issues are traced before they become a problem. However when running both tools we were using (Nagios and Red Hat Ansible), programming your tools inefficiently means high resource usage on the system. Since SAS Software uses a lot of CPU, you do not want your monitoring tools to take too much of these resources. Which means you need to program your tools efficiently and only check the resources you need to check at the moment you need to check them. An obvious example would be: you do not need to check the license expiration date every minute. Once a day, or once a week would be sufficient.

## CHECK IF A TASK IS REDUNDANT

Every time a script is written, one of the main questions is, do we need to run this for a single server, a single customer, or multiple servers?

When we know the answer to this question, we can define how we want to deploy the script to the correct servers. This can be done at installation time using Puppet, or can be done per server, using Red Hat Ansible.

Occasionally we can even decide to use a combination of a script and customer input. By giving a customer the availability of saving a file on a specific location, we can automatically set database connections, add users to a LDAP configuration or grant permissions to specific files.

## CONSISTENCY OF USERS, PERMISSIONS AND LOCATIONS

From a number of customers we received questions on how to make sure the correct people have access to their specific SAS Folders and CAS Libraries, and that these authorizations are also reflected in the file system. For this we introduced the SaasNow Viya Authorization Sync Mechanism (VASM). Using a combination of tools, VASM makes sure authorizations in SAS Content, CAS and the filesystem stay in sync. VASM is available for SaasNow customers as an add-on for their server.

## CONCLUSION

Managing multiple servers, whether it is local or in the cloud can be challenging. While trying to keep all servers to keep the same base, you also want to provide freedom to the users.

As you do not want to have too many manual jobs to be executed on a server, you will need to script as many tasks as possible, which needs to be done efficiently and clear. You also need to make sure to document your scripts, so other people can follow your scripts and see why you make a certain choice and how to proceed from there on.

## REFERENCES

- (1) OpenLDAP - <https://www.openldap.org/>
- (2) Red Hat Ansible - <https://www.redhat.com/en/technologies/management/ansible>
- (3) Puppet - <https://puppet.com/>
- (4) GNU Bash (Unix shell) - <https://www.gnu.org/software/bash/>
- (5) Nagios - <https://www.nagios.org/>
- (6) SAS Deployment Guides - <https://support.sas.com/en/documentation/install-center/viya/deployment-guides.html>

## ACKNOWLEDGMENTS

A special thank you to Paul Koot for his contribution to and reviewing of this paper.

## CONTACT INFORMATION

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

Jelle Daemen  
SAASNOW | powered by Notilyze  
[jelle@notilyze.com](mailto:jelle@notilyze.com)  
<https://www.saasnow.com/>

Tom Dogger  
SAASNOW | powered by Notilyze  
[tom@notilyze.com](mailto:tom@notilyze.com)  
<https://www.saasnow.com/>