



SAS® Club

Der Business Analytics Club für SAS User
27. November 2019 | Wien



SAS in der Cloud

SAS Club Wien, 27.11.2019

Agenda

- Überblick SAS & Cloud
- Container
 - Kurze Einführung
 - SAS for Containers Überblick
- Einsatz von Container-Technologie für Analytik

What does SAS do *in the Cloud*?



The **SAS Cloud** combines software, infrastructure, and services that are **designed and managed by SAS** for optimal performance and value.



SAS provides **cloud-deployable** options that allow our customers to run SAS on almost any private, public, or hybrid cloud infrastructure.



Software as a Service

Off-the-shelf offerings designed to scale and fit for purpose. Sign up, log in, and get to work. Can be modified to your future needs.



Results as a Service

Give us your data and problem, and we give you the answers on which you can take action.



Managed Services

Your software or infrastructure can be hosted or remotely managed by SAS experts 24/7.



SAS® Cloud

Example Offerings



Software as a Service (SaaS)

- SAS® Visual Analytics
- SAS® Visual Statistics
- SAS® VDMML
- SAS® 360 Plan, SAS® 360 Discover, SAS® 360 Engage



Results as a Service (RaaS)

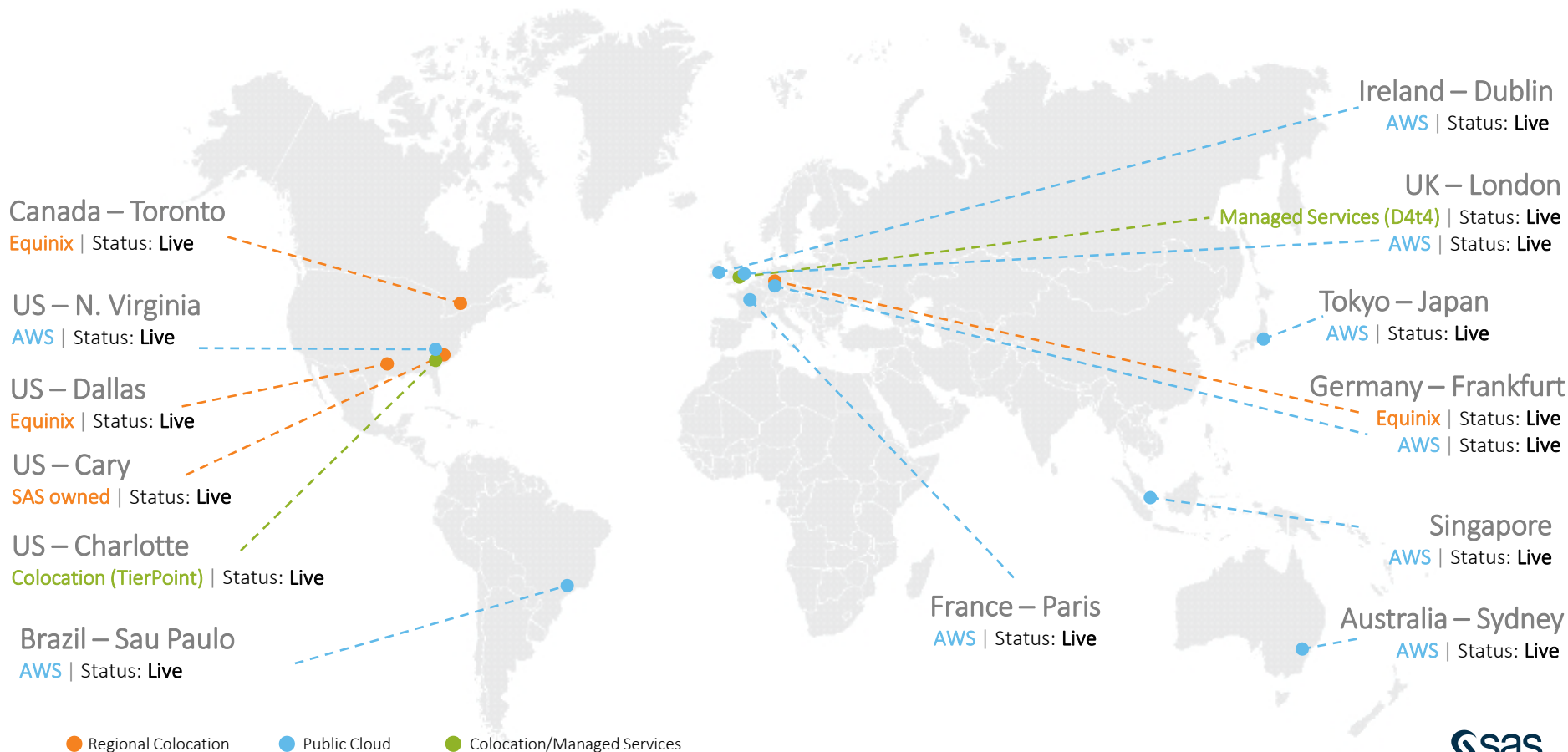
- SAS® Results: Analytic Insights
- SAS® Results: Fraud Insights
- SAS® Results: Customer Insights
- SAS® Results: Retail Insights



Managed Services

- Remote Managed Services
- Hosted Managed Services

Global Data Center Coverage





The **SAS® Cloud** combines software, infrastructure, and services that are *designed and managed by SAS* for optimal performance and value.



SAS provides *cloud-deployable* options that allow our customers to run SAS on almost any private, public, or hybrid cloud infrastructure.



General Support for Cloud Providers

Support to run SAS Products on cloud infrastructures, if customers follow **system requirements** and **policy on virtualized environments**.



Cloud Deployment Patterns

Support cloud **deployment patterns** (e.g. containers, orchestration) so that **our customers** can deploy SAS in a variety of cloud infrastructures.



Cloud-Specific Offerings

Specific offerings and deployment best practices for cloud infrastructure providers.

SAS® on Cloud Providers

Example Offerings



General Support for Cloud Providers

- General policy on virtualized environments



Cloud Deployment Patterns

- SAS Analytics for Containers
- SAS Analytics for Containers on SAS Viya
- SAS support of Docker Containers and Kubernetes Orchestration

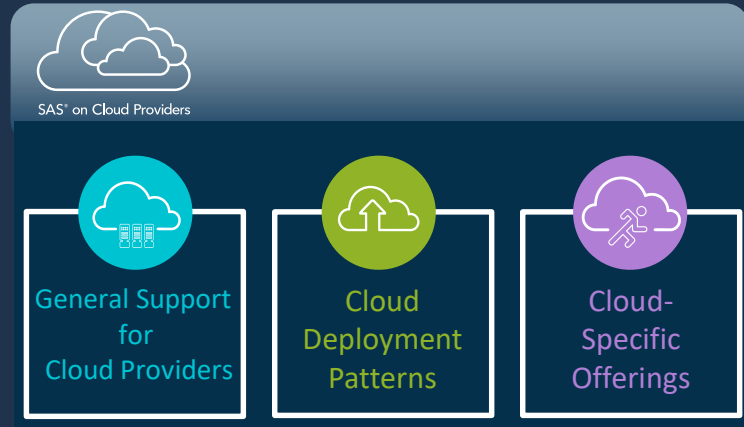
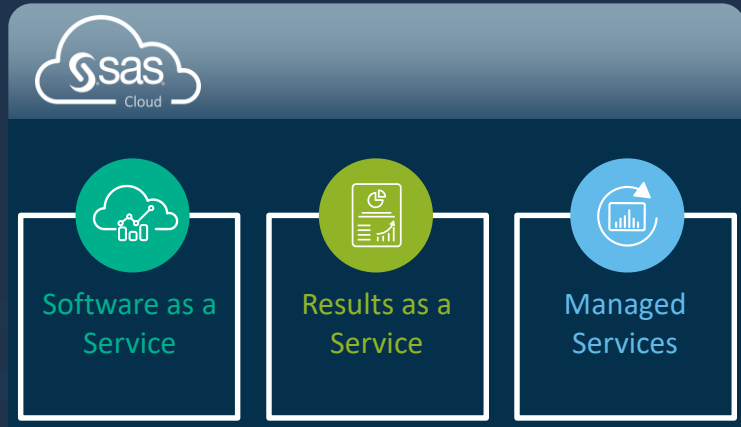


Cloud-Specific Offerings

- Quick Start for
 - AWS
 - Microsoft Azure
 - Google Cloud Platform
- SAS/ACCESS for
 - Amazon Redshift
 - BigQuery
- ...

SAS® and Cloud Computing

Our Overall Cloud Strategy





Containers

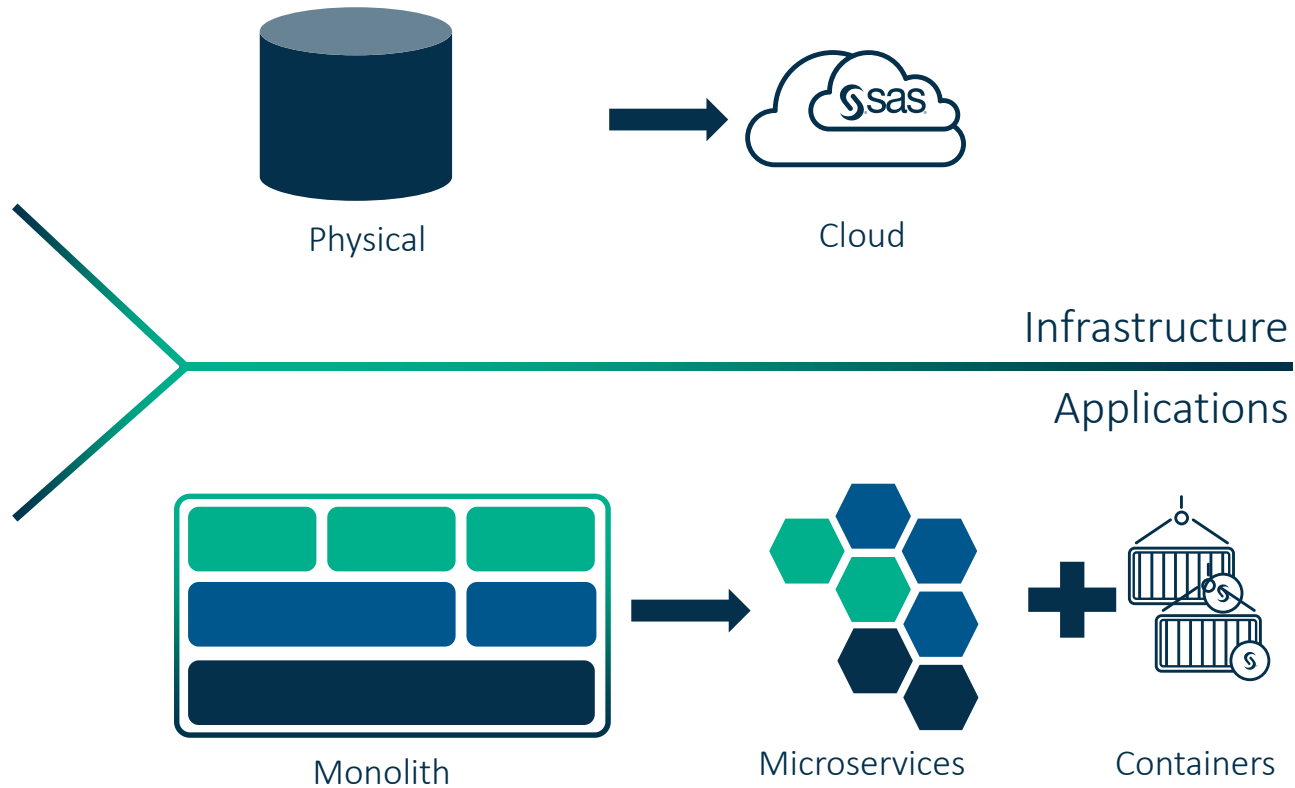
Current State

How we deliver software

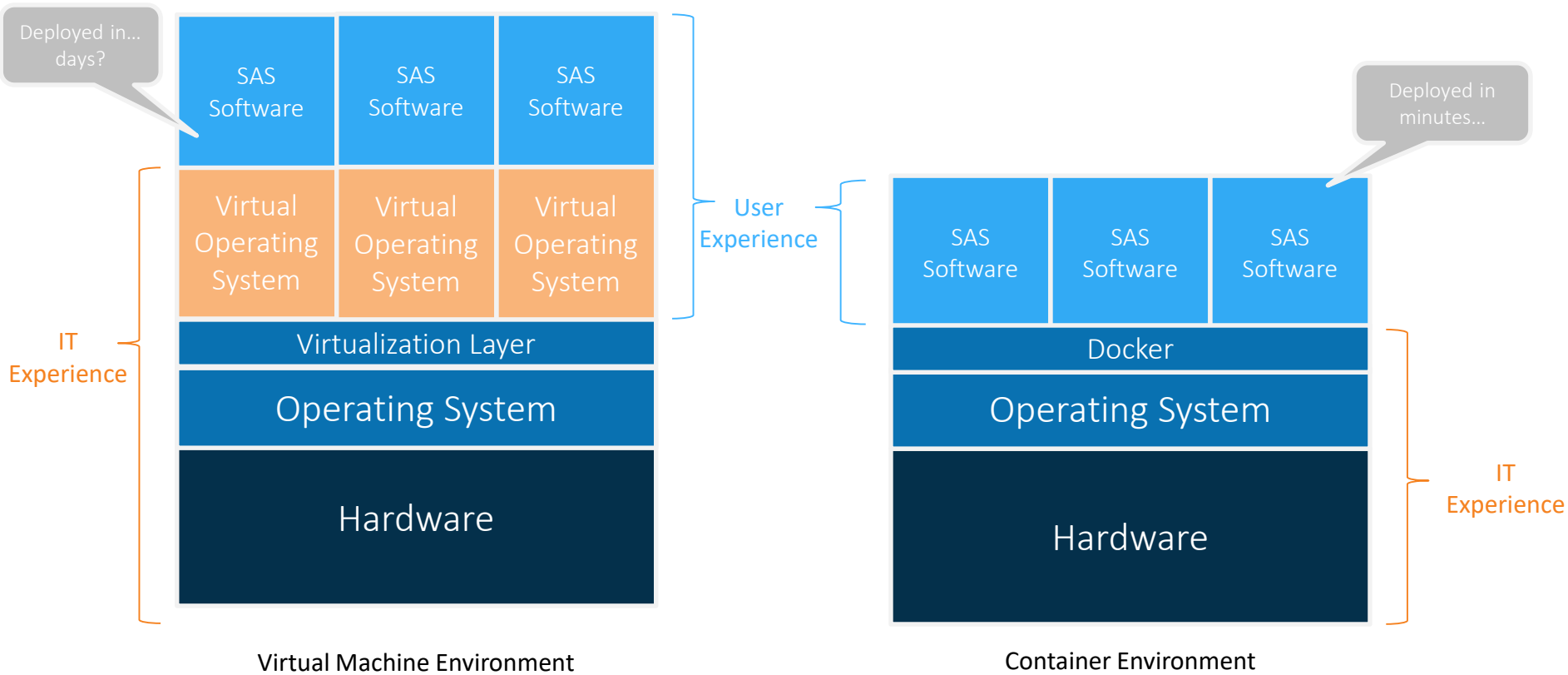
- Viya deployments have become highly automated
 - Cloud
 - Cloud tools such as CloudFormation, Terraform
 - Shell scripts
 - [SAS Viya ARK](#)
 - Not to mention ansible itself...
- But all Viya deployments are still done on dedicated infrastructure
 - “Always-on” Virtual Machines



Software Evolution



What is a Container?




```
1 ARG BASEIMAGE=centos
2 ARG BSETAG=7
3
4 FROM $BASEIMAGE:$BSETAG
5
6 ENV DOMINO_USER_NAME="centos"
7
8 # install pre-regs
9 RUN rpmdb --rebuilddb; yum -y install java-1.8.0-openjdk openssh-clients
  openssh-server glibc libpng12 libXp libXmu net-tools numactl sudo httpd mod_ssl rsync
  which initscripts iproute lsof git wget bzip2; yum clean all
10
11 # prerequisites
12 RUN sed -i "/keepcache=/c\keepcache=1" /etc/yum.conf; sh -c 'echo "* - nofile 20480"
  >> /etc/security/limits.conf'; sed -i.bak -e 's/4096/65536/g'
  /etc/security/limits.d/20-nproc.conf;
13
14 # # Configure users
15 RUN useradd --create-home --shell /bin/bash ${DOMINO_USER_NAME} && sh -c 'echo
  "SASpw1" | passwd "${DOMINO_USER_NAME}" --stdin' && echo "host localhost user
  ${DOMINO_USER_NAME} password SASpw1" > /home/${DOMINO_USER_NAME}/.authinfo &&
  useradd -M sas
16
17 # use chown to avoid separate run chown command, 1000=centos, 1001=sas
18 ADD --chown=1001:1001 sashome /nfsshare/sas9/sashome
```

```
1 ARG BASEIMAGE=centos
2 ARG BSETAG=7
3
4 FROM $BASEIMAGE:$BSETAG
5
6 ENV DOMINO_USER_NAME="centos"
7
8 # install pre-regs
9 RUN rpmdb --rebuilddb; yum -y install java-1.8.0-openjdk openssh-clients
  openssh-server glibc libpng12 libXp libXmu net-tools numactl sudo httpd mod_ssl rsync
  which initscripts iproute lsof git wget bzip2; yum clean all
10
11 # prerequisites
12 RUN sed -i "/keepcache=/c\keepcache=1" /etc/yum.conf; sh -c 'echo "* - nofile 20480"
  >> /etc/security/limits.conf'; sed -i.bak -e 's/4096/65536/g'
  /etc/security/limits.d/20-nproc.conf;
13
14 # # Configure users
15 RUN useradd --create-home --shell /bin/bash ${DOMINO_USER_NAME} && sh -c 'echo
  "SASpw1" | passwd "${DOMINO_USER_NAME}" --stdin' && echo "host localhost user
  ${DOMINO_USER_NAME} password SASpw1" > /home/${DOMINO_USER_NAME}/.authinfo &&
  useradd -M sas
16
17 # use chown to avoid separate run chown command, 1000=centos, 1001=sas
18 ADD --chown=1001:1001 sashome /nfsshare/sas9/sashome
```

Two different flavors

How to get started with containers

Pre-baked

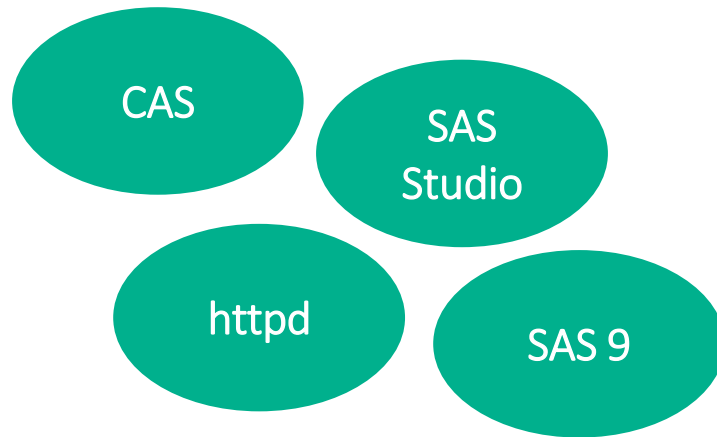
Build your own



“Single” Container (programming only)


“poac”

- This container image can run on any host where Docker is installed








http://github.com/sassoftware






SAS Software

Open Source from SAS Software

 Cary, North Carolina, USA  <https://www.sas.com/>  github@sas.com Verified


[Report abuse](#)

 **Repositories** 109  **People** 27  **Projects** 0

Pinned repositories


[sas_kernel](#)

A Jupyter kernel for SAS. This opens up all the data manipulation and analytics capabilities of your SAS system within a notebook interface. Use the Jupyter Notebook interface to execute SAS code a...

 Jupyter Notebook ★ 131 🍴 52


[sas-viya-programming](#)

Code samples and materials to help you learn to access SAS Viya services by writing programs in Python and other open-source languages

 Jupyter Notebook ★ 70 🍴 75


[sas-prog-for-r-users](#)

Teaching and lab materials for the "SAS Programming for R Users" course, including course notes, data, and code.

 SAS ★ 80 🍴 40


[saspy](#)

A Python interface module to the SAS System. It works with Linux, Windows, and mainframe SAS. It supports the sas_kernel project (a Jupyter Notebook kernel for SAS) or can be used on its own.

 Python ★ 154 🍴 67


[python-swat](#)

The SAS Scripting Wrapper for Analytics Transfer (SWAT) package is the Python client to SAS Cloud Analytic Services (CAS). It allows users to execute CAS actions and process the results all from Py...

 Python ★ 67 🍴 28

[python-dlpy](#)

The SAS Deep Learning Python (DLPy) package provides the high-level Python APIs to deep learning methods in SAS Visual Data Mining and Machine Learning. It allows users to build deep learning model...

 Python ★ 89 🍴 38

“Full” Containers (viya visuals)

Also can be smp or mpp cas

- These container images can run on any Kubernetes cluster



K8S Cluster

Container Recipe vs Image



SAS “stuff” in a Docker Container

- Build your own
- Whether you *should or not*, is up to you
- Not tested, limited support
- Our customers are *trying* this today



Container **Recipe** from SAS

- Instructions for building your own
- Tested and supported
- Can be modified to some extent
- 9.4 available now, SAS Viya available now



Container **Image** from SAS

- Already created
- Tested and supported
- Limited customizations
- SAS Viya

Workloads Suitable for Containers

Data Science Workbench

- Individual workspaces
- Spin up when needed
- Scalable compute
- Code in SAS, Python or R
- Domino Data Lab

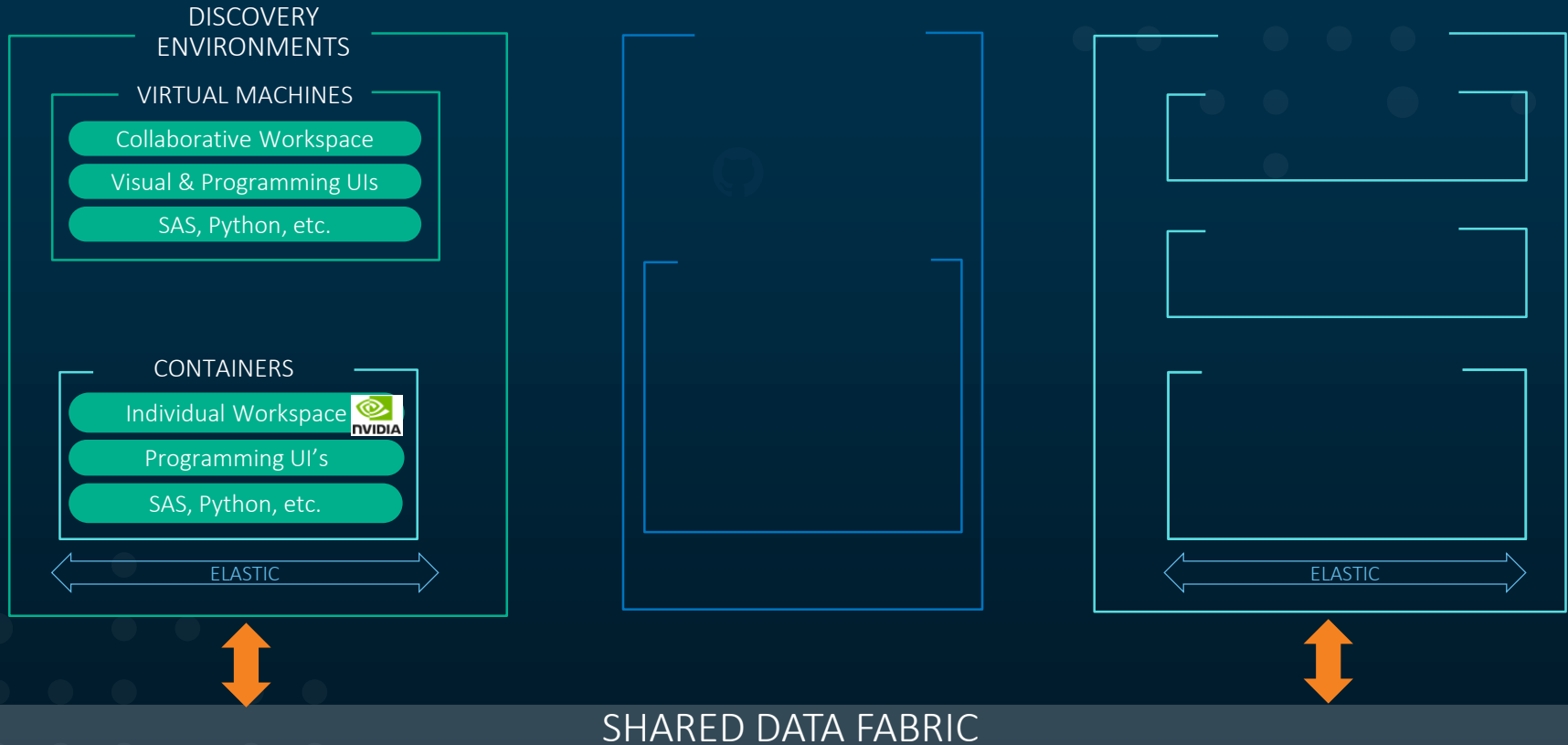
Runtime engine

- Single purpose
- Packaged with model and decision code
- Spin up multiple containers to scale out
- Integrate with DevOps
- Suitable for edge deployment

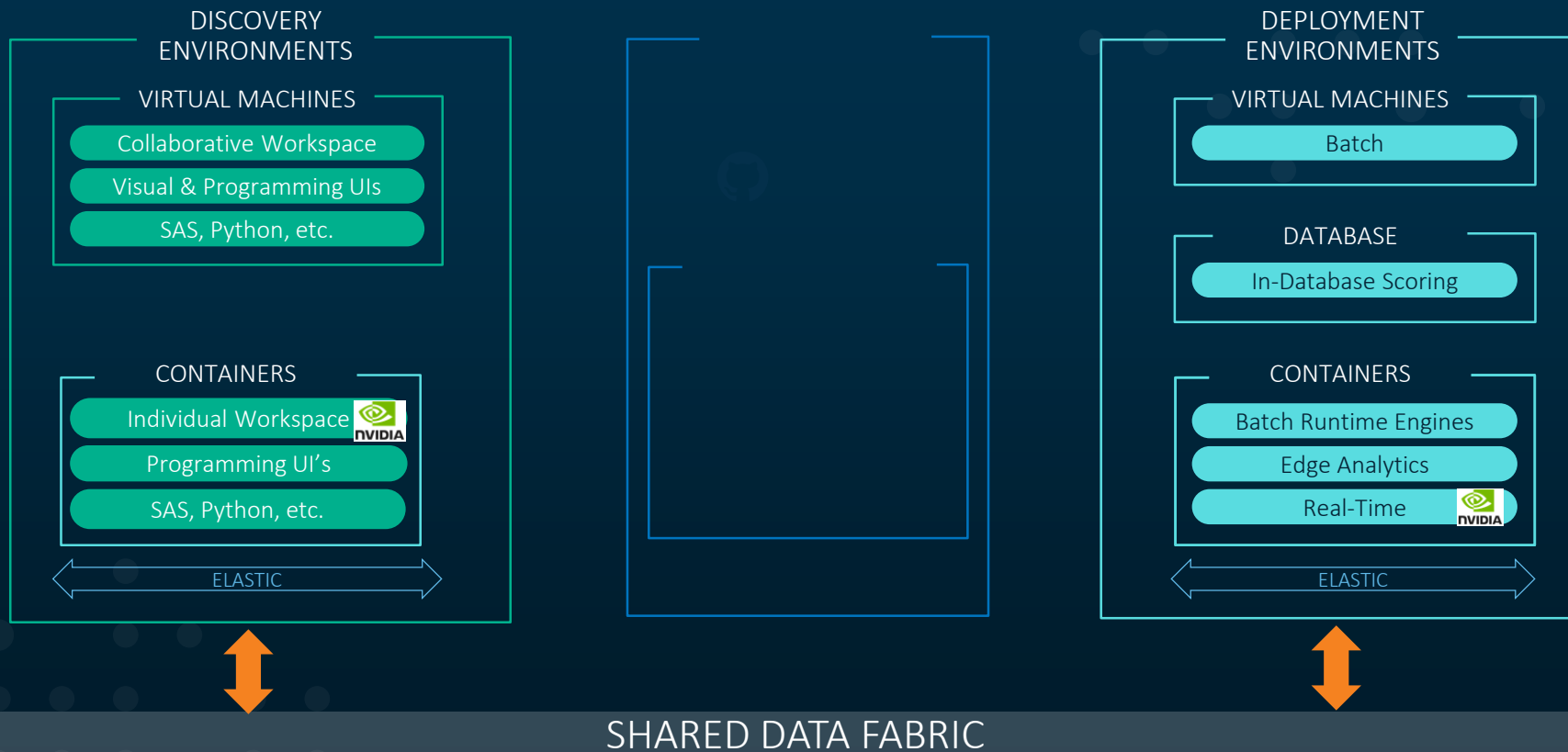
Full platform – work in progress

- Collaborative workspaces
- Visual interfaces

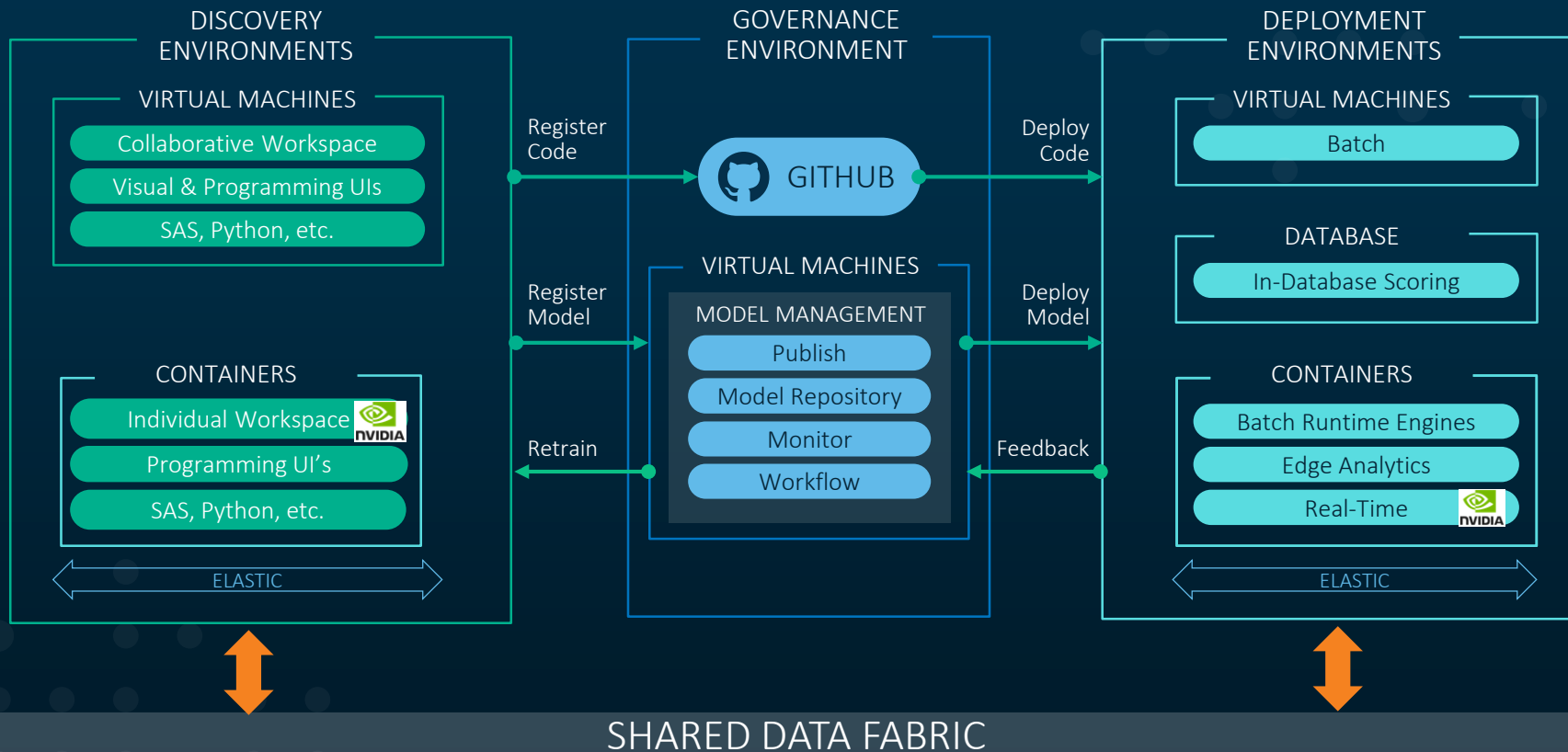
Enterprise KI Plattform: logische Architektur



Enterprise KI Plattform: logische Architektur

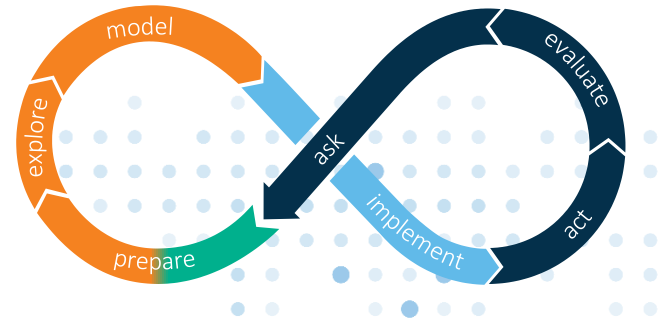
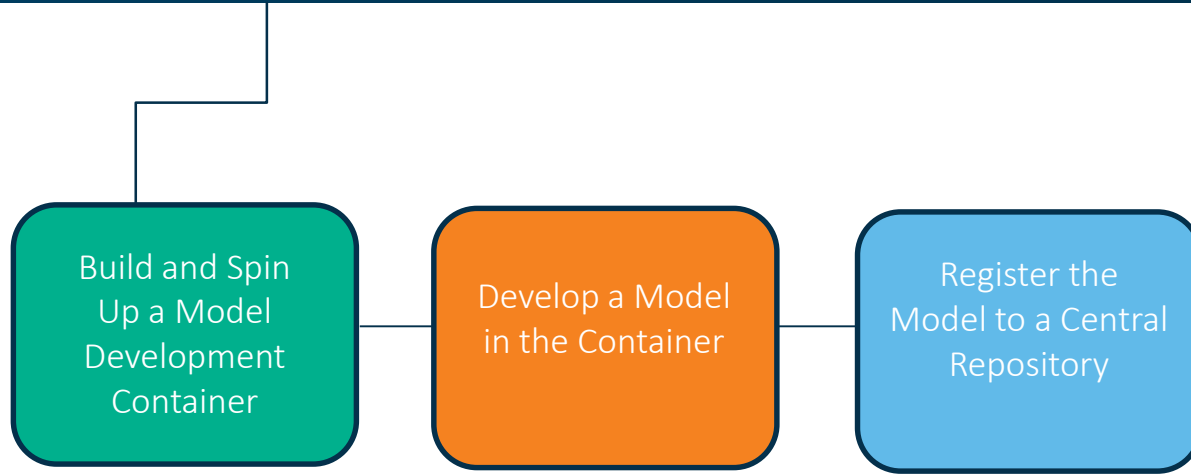


Enterprise KI Plattform: logische Architektur





SAS Model Developer





Build and Spin Up a Model Development Container

Identify what is
needed for
modeling

Spin up container
with software for
modeling

The screenshot displays the SAS Project MAP interface. At the top, the SAS logo and 'THE POWER TO KNOW.' tagline are visible, along with 'Project MAP' and 'AP MSSE'. Below the header, there are tabs for 'App', 'Repository', and 'Question?'. A green button labeled 'Switch User (User1)' is in the top right. The main area shows five container cards, each with a cloud icon and a Docker logo. The cards are labeled: 'viya-poac-development', 'viya-poac-development-user1', 'viya-poac-development-user2', 'viya-poac-ps', and 'viya-poac-scoring'. Each card lists the file path '/home/sasdemo/data', the Viya version (3.4), Python version (3.6), and R version (1.1.463). Below each card is a status bar with a Docker logo and a label: 'Up - Global', 'Up - Personal', 'Up - Personal', 'Up - Global', and 'Up - System'. Below the cards, there is a 'Selected container:' section with a text box containing 'viya-poac-development'. An 'Action:' section has a 'Switch Container' button. At the bottom, a 'Status:' section shows 'loading . . OK.'.



Develop a Model in the Container

Build Models in
Visual Interface

Compare
Development
Models

CONTINUE



Register the Model to a Central Repository

Click button to Register
Model in Model
Repository

Build – ship – run

