The Intersection of Workload Placement Between Kubernetes and the SAS Viya Platform

by Rob Collum

Advisory Technical Architect, SAS

rob.collum@sas.com

SBK704813

SAS EXPLORE

Infrastructure

Planning, provisioning, and managing infrastructure (that is, servers, storage, networking, databases, etc.) to provide the physical resources for SAS Viya.

Kubernetes

Kubernetes (Greek for "helmsman") offers an automated and extensible platform for managing containerized workloads to provide automation, resiliency, load balancing, and security.

SAS Viya

SAS Viya is tightly integrated with Kubernetes for workload placement. SAS Workload Management extends that integration to provide additional capabilities

The Intersection of Workload Placement Between Kubernetes and the SAS Viya Platform

Infrastructure

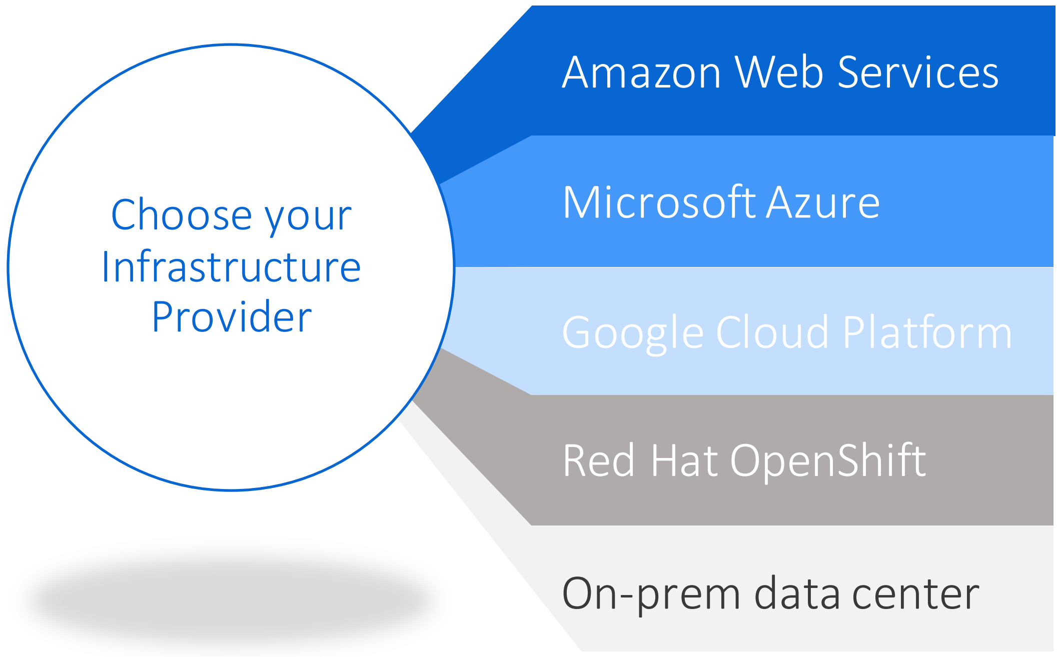
Infrastructure

Provisioning infrastructure that’s ideal for SAS Viya workloads is critical to ensuring efficient use and optimized costs.

Machine virtualization and software containerization now offer incredible levels of flexibility and adjustment. SAS Viya can enjoy these benefits as the environment is right-sized over time to meet ever-changing usage and requirements.

Infrastructure

SAS Viya works with a variety of infrastructure providers



1. Elastic Kubernetes Service (EKS)
   Amazon Web Services
2. Azure Kubernetes Service (AKS)
   Microsoft Azure
3. Google Kubernetes Engine (GKE)
   Google Cloud Platform
4. OpenShift Container Platform (OCP)
   Red Hat OpenShift
5. Upstream, open-source Kubernetes
   Your on-premise data center
   (or self-managed on a cloud provider)

SAS EXPLORE

sas

Infrastructure

Partner with your SAS account representative to engage the

SAS World-Wide Sizings team

for help with estimating infrastructure requirements.

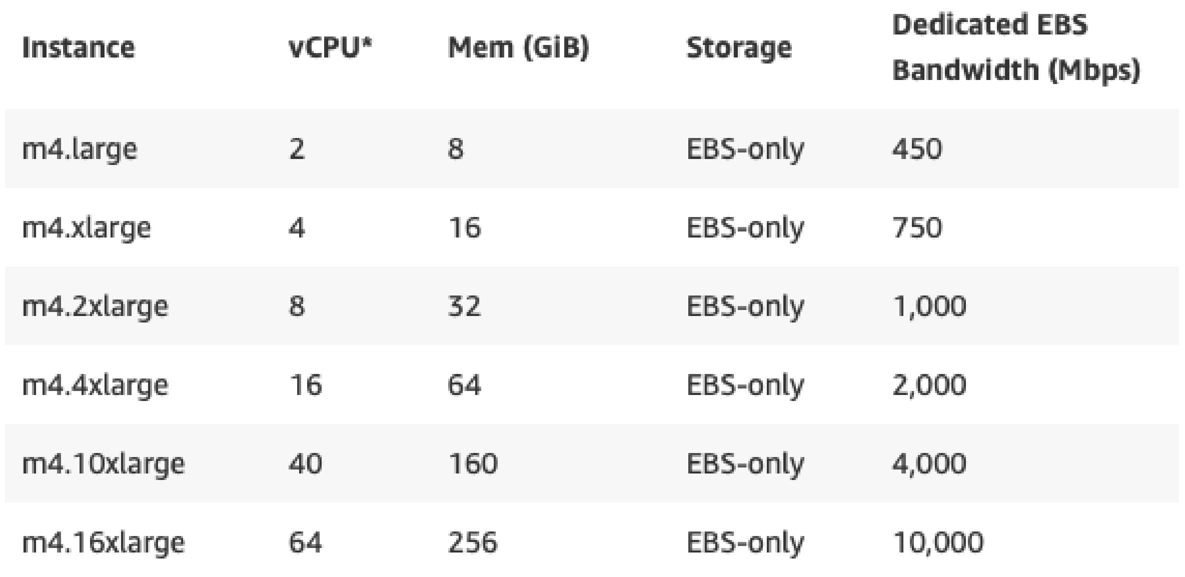
Infrastructure

Nodes

In Kubernetes, a node is typically a virtual machine provisioned by selecting an instance type to run. Instances offer combinations of CPU, memory, storage, and networking capacity to give an appropriate mix of resources for your applications.

Select instance type(s) as needed for specific workloads.

For most providers, the size (and/or type) can be changed after initial provisioning.



Instance	vCPU*	Mem (GiB)	Storage	Dedicated EBS Bandwidth (Mbps)
m4.large	2	8	EBS-only	450
m4.xlarge	4	16	EBS-only	750
m4.2xlarge	8	32	EBS-only	1,000
m4.4xlarge	16	64	EBS-only	2,000
m4.10xlarge	40	160	EBS-only	4,000
m4.16xlarge	64	256	EBS-only	10,000

SAS EXPLORE

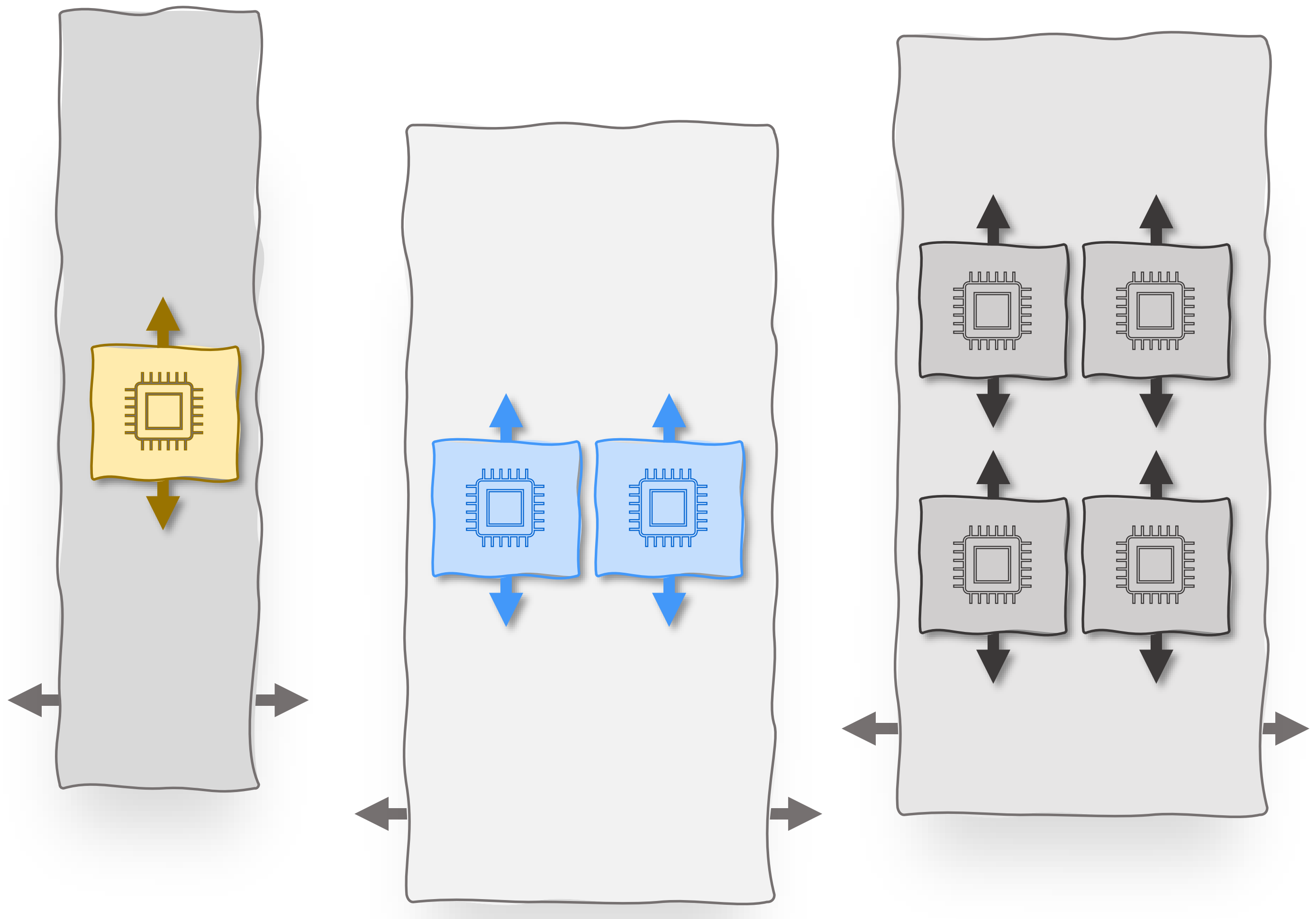
Infrastructure

Node pools

In Kubernetes, the node pool concept refers to a group of nodes within a cluster that all have the same configuration. Typically, this means all the nodes in a node pool are the same instance type.

Node pools can scale up (or down) the number of nodes running.

Establish node pools to independently manage optimized resources.



SAS EXPLORE

sas

Infrastructure

Node pools > **SAS Workload Classes**

SAS identifies four major workload classes to consider when supplying infrastructure for the SAS Viya platform. This allows for optimized hardware for efficient operation and cost.

Additional nodes pools (for other SAS Viya workload classes or for third-party services hosted in the cluster) can be provisioned per the site's requirements.

CAS

high-performance, in-memory analytics

Compute

SAS Programming Runtime Environment

Stateful

critical infrastructure services

Stateless

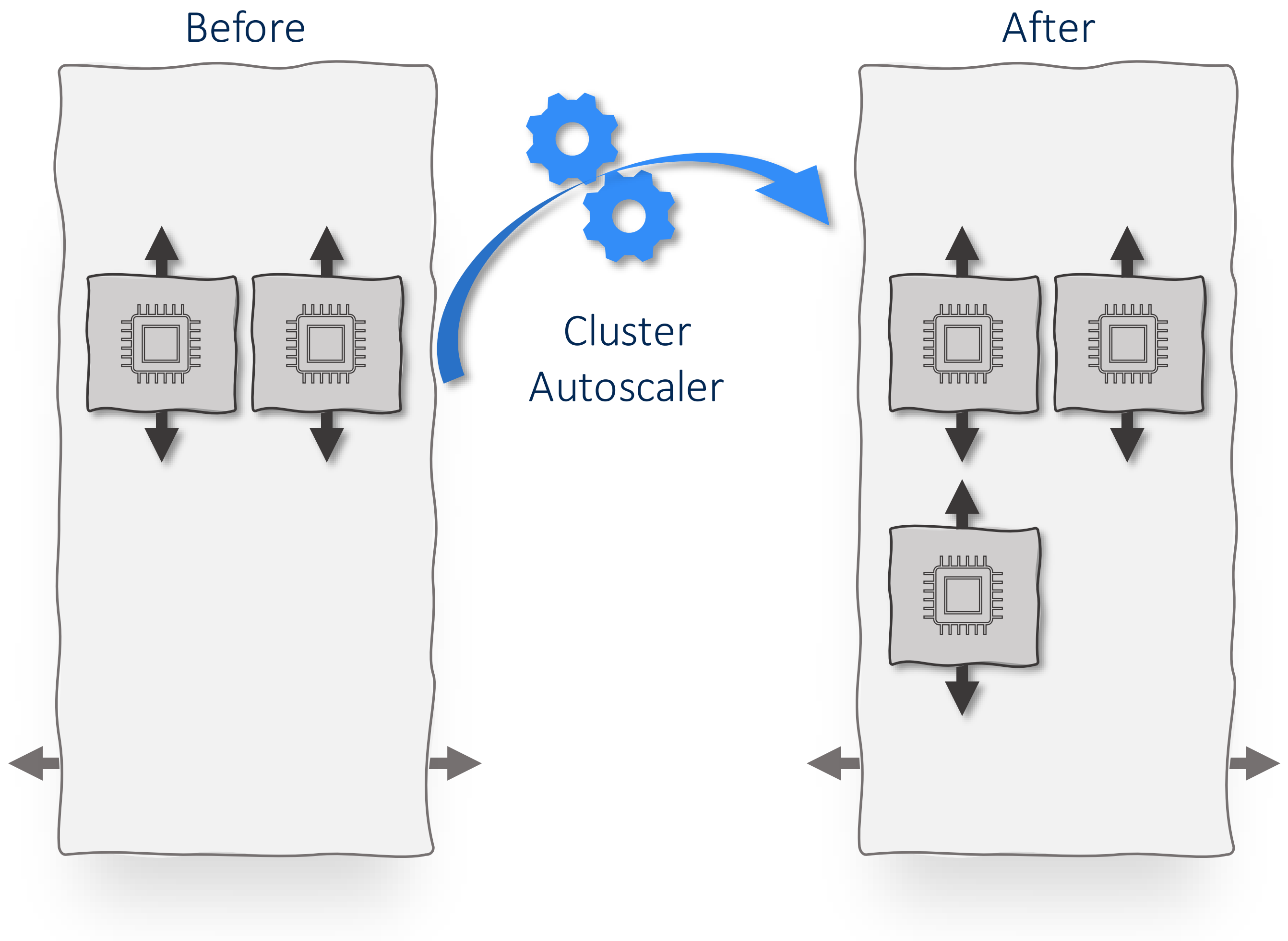
components for specific functionality

Infrastructure

Cluster Autoscaler

A cluster autoscaler operates outside of Kubernetes to adjust the size of the cluster as needed. It does this by monitoring the cluster to ensure every pod has a place to run, scaling up (or down) as configured.

Node pools are a natural complement to allow for automatic and independent scaling of resources per workload demands.



SAS EXPLORE

Infrastructure

There’s a lot more we’re not looking at in this presentation:

Networking
- Default or custom container network interface (CNI)
- External access
- Quality of service

Storage
- Default or custom container storage interface (CSI)
- Backups, disaster recovery
- Quality of service

External database(s)
- Custom or cloud-managed
- Use by SAS Viya for operations and/or analytics data mart
- Quality of service

-
-
-

SAS EXPLORE

sas

Infrastructure

Infrastructure provisioning is where workload management begins.

Choices made regarding size, throughput, availability, location, price, and more will impact where jobs run, how quickly they take to complete, and how much they benefit the business.

Kubernetes

Kubernetes

SAS Viya is designed to rely on Kubernetes as a platform for administering its containerized software across multiple hosts for availability and scalability.

Kubernetes offers a variety of features and capabilities to automatically monitor and manage software. It is highly configurable as well as extensible. Some aspects of the Kubernetes cluster are setup for SAS Viya when it is deployed whereas others are realized as the software executes.

Kubernetes

When implementing a cloud provider's managed Kubernetes service, integration between the cluster and the physical infrastructure is included. For example:

- An Ingress Controller deployed to the Kubernetes cluster works with the physical load balancer seamlessly.
- Storage classes defined along with PVC can automatically provision additional disk volumes.
- The cluster autoscaler can change the number of nodes running in a node pool.

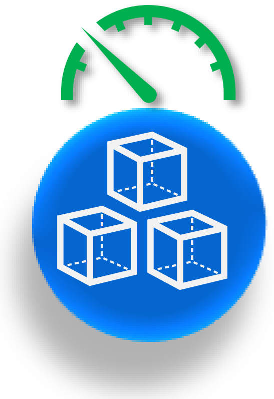
Kubernetes

Pods

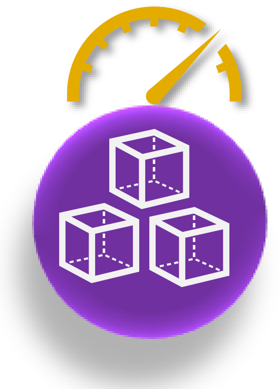
A pod is the smallest unit of a Kubernetes application. A single pod could be comprised of one or more containers. In general, pods are ephemeral such that if one fails, Kubernetes can automatically start up a replica to replace it.

SAS Viya software is deployed to Kubernetes as a variety of different pods. Most SAS Viya pods can operate statelessly and can act ephemerally. Some SAS Viya pods - notably for analytic engines and some stateful infrastructure services – require more deliberate care when starting and stopping.

Kubernetes



```
resources:
  limits:
    cpu: 2
  requests:
    cpu: 50m
```



```
resources:
  limits:
    memory: 3Gi
  requests:
    memory: 1500Mi
```

Resource Requests and Limits

Each container in a pod can notify Kubernetes of boundaries to set on the amount of CPU and/or memory they will use.

A container can establish a request for CPU and/or memory which Kubernetes essentially treats as a minimum that it'll try to reserve in the cluster. Actual usage could be more or less.

We can also establish limits for CPU and/or memory usage of containers. This means the runtime prevents the container from using more than the configured limit.

SAS EXPLORE

Kubernetes

Quality of Service

Quality of Service (QoS) is a classification system which determines the scheduling (and eviction) priority of pods in a k8s cluster. QoS is not something set explicitly – instead it's determined based on requests and limits defined in the deployment.

There are 3 classes of QoS…

SAS EXPLORE

Kubernetes

Quality of Service = Guaranteed

A pod's QoS class will be Guaranteed when:

- All containers in the pod specify both a request and a limit
- Every container's memory request equals its limit
- Every container's CPU request equals its limit.

Kubernetes will only schedule the pod to nodes which have sufficient memory and CPU resources to satisfy their limits.

Kubernetes

Quality of Service = Burstable

A pod’s QoS class will be Burstable when:

- It isn’t Guaranteed QoS
- At least 1 container in the pod has either a CPU or memory request or limit.

Kubernetes will schedule the pod to any node that has available resources. Limits per pod are enforced, but it’s possible the limit values could combine over time to exceed that node’s capacity.

SAS EXPLORE

sas

Kubernetes

Quality of Service = BestEffort

A pod's QoS class will be BestEffort when:

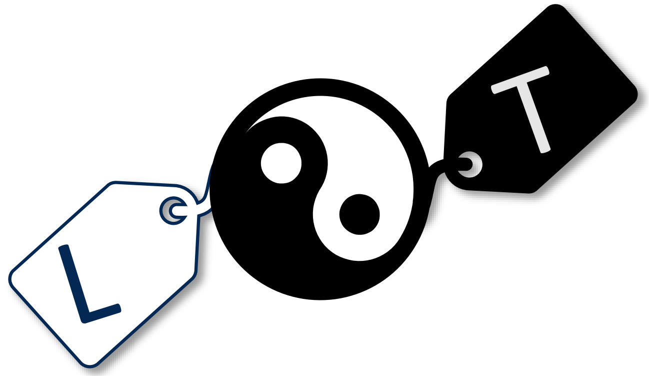
- It isn't Guaranteed nor Burstable QoS
- No container in the pod has defined CPU or memory requests or limits.

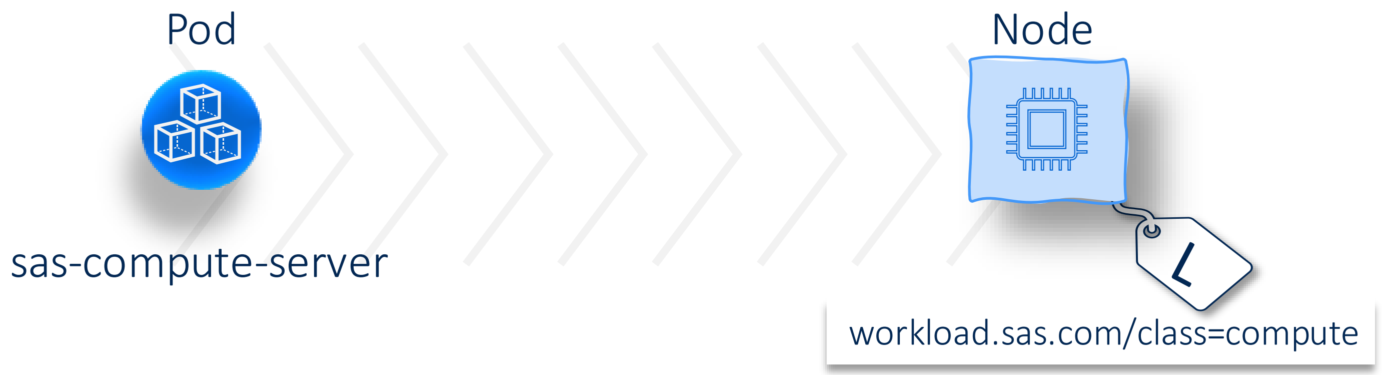
The pod is scheduled to run on any node that has available resources. It can use any amount of free CPU and/or memory on the node. While flexible, take care to ensure that pods aren't resource hogs which contend with other pods, degrading service.

Kubernetes

Labels

In Kubernetes, a label applied to a node can be used to attract pods with specific tasks. Those pods are defined with an associated node selector to match the label.





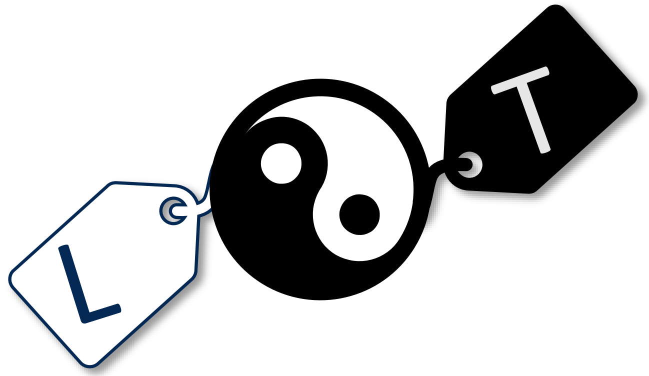
SAS EXPLORE

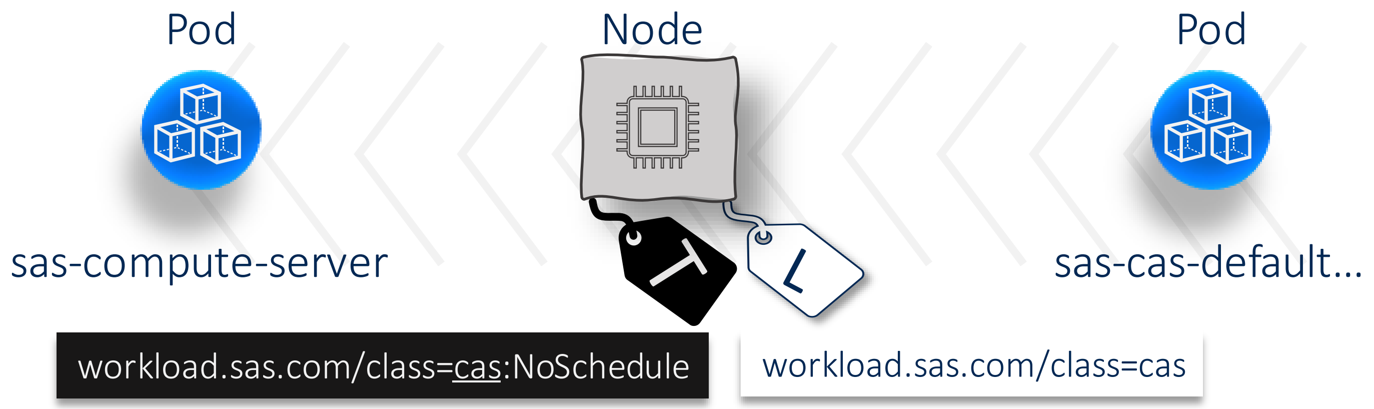
sas

Kubernetes

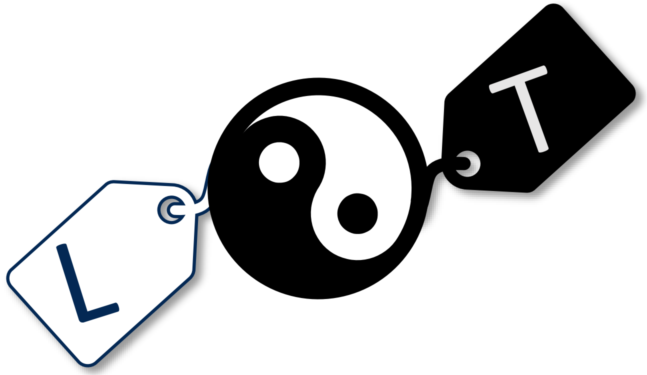
Taints

A taint is a special label used to repel unwanted pods away from the node. Pods that have the appropriate toleration defined can run on those nodes.




Pod
Node
Pod
sas-compute-server
T
L
sas-cas-default...
workload.sas.com/class=cas:NoSchedule
workload.sas.com/class=cas

Infrastructure



Labels and Taints

By judiciously using labels and taints together, nodes with specialized features (or otherwise limited use) can be reserved for the desired workloads.

Pod Affinities

Pod affinities (and anti-affinities) are similar in concept to the node affinities defined by Labels and Taints except they are are used to attract pods to (or repel pods away from) each other.

Kubernetes

Kubernetes is a rich environment with innumerable configuration possibilities. It’s scalable, robust, resilient, and extensible. The pace of innovation to improve Kubernetes continues to make it better every day.

There are many approaches to configuring how work executes in a Kubernetes cluster. SAS Viya utilizes many of these and also extends some areas with its own operators to best utilize the environment.

SAS Viya

SAS Viya Platform

The SAS Viya platform is made up of numerous software components which work together to deliver analytics, reporting, data management, and more. SAS Viya integrates with Kubernetes to manage the different kinds of workloads that those software components deliver.

The major analytic engines in SAS Viya can make their own accommodations for workload management beyond what Kubernetes can see or manage.

SAS Viya Platform

Improving Concurrency Performance

SAS Viya is designed to accommodate running analytic tasks of nearly any size. This means for larger jobs, it requires more resources, more CPU, more RAM, more disk, and more hosts.

On the other hand, not all analytic tasks need so much. So out of the box, SAS Viya is configured to “play nice” in what might be a shared environment.

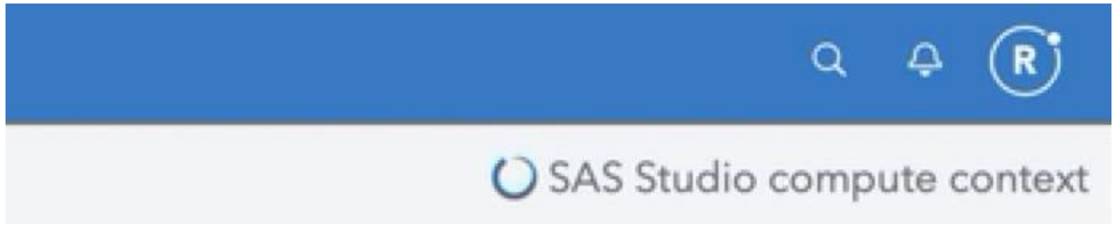
Let’s look at some configuration changes to make SAS Viya handle more concurrent jobs.

SAS Viya Platform

Improving Concurrency Performance

Compute

Have you waited for this in SAS Studio?



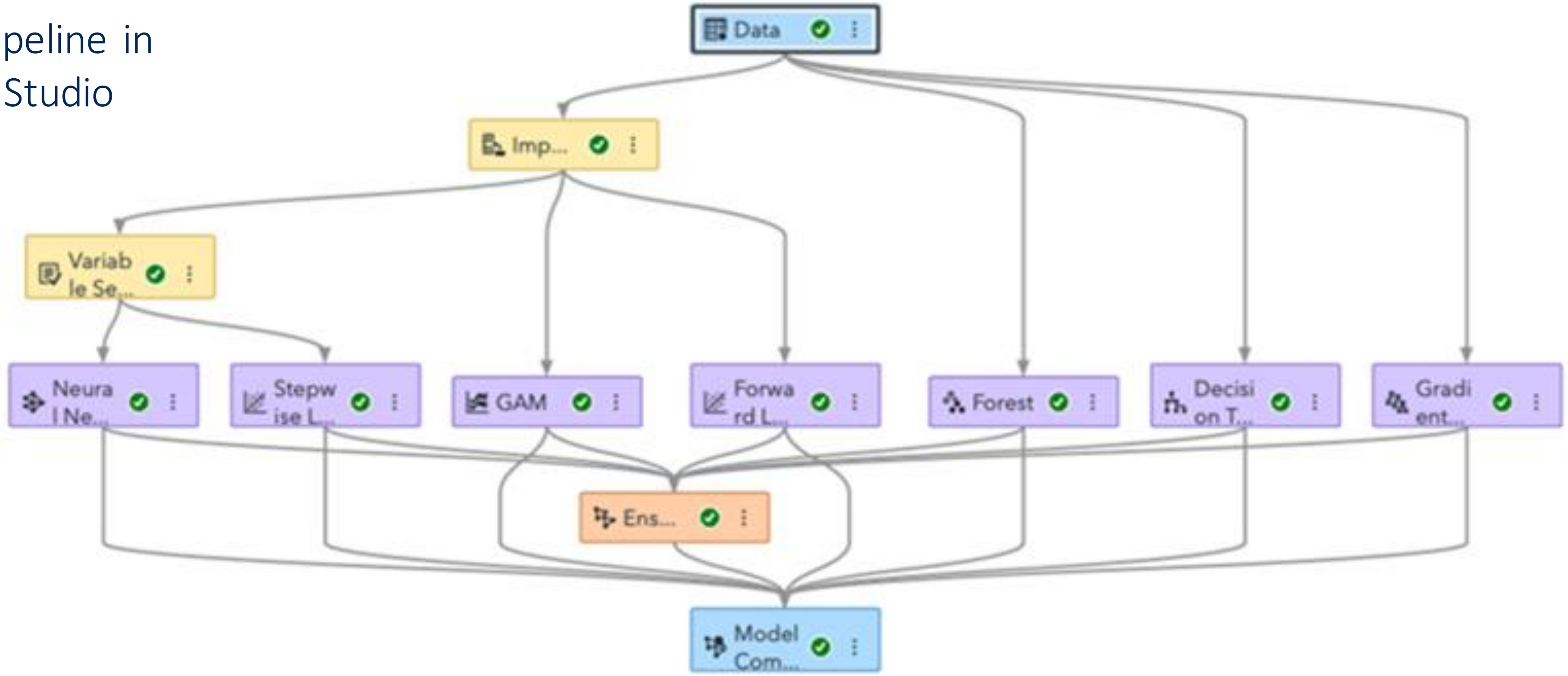
By default, SAS Viya will trigger the instantiation of a new SAS Compute Server pod in Kubernetes when the user logs in. Kubernetes needs time to schedule the pod, pull the containers, and execute the SAS Compute Server software so that it can then connect back to the SAS Studio app.

SAS Viya Platform

Improving Concurrency Performance

Compute

Analytics Pipeline in SAS Model Studio

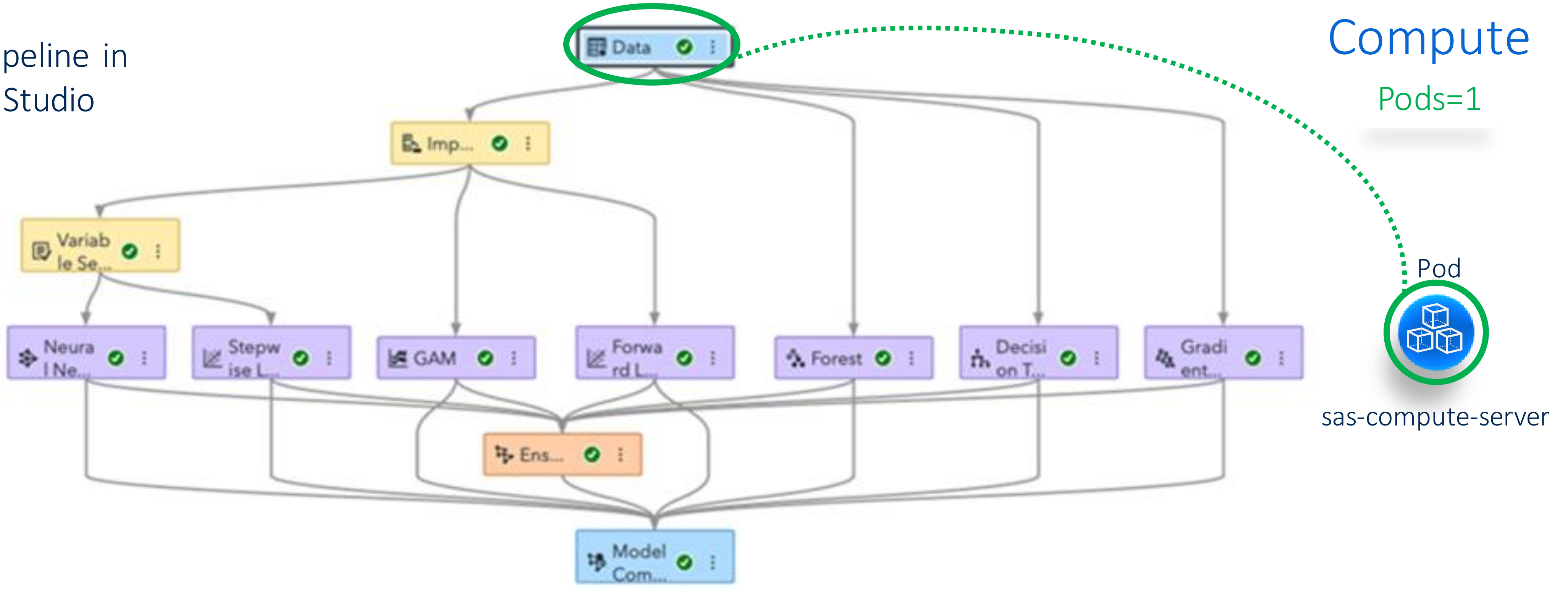


SAS EXPLORE

SAS Viya Platform

Improving Concurrency Performance

Analytics Pipeline in SAS Model Studio



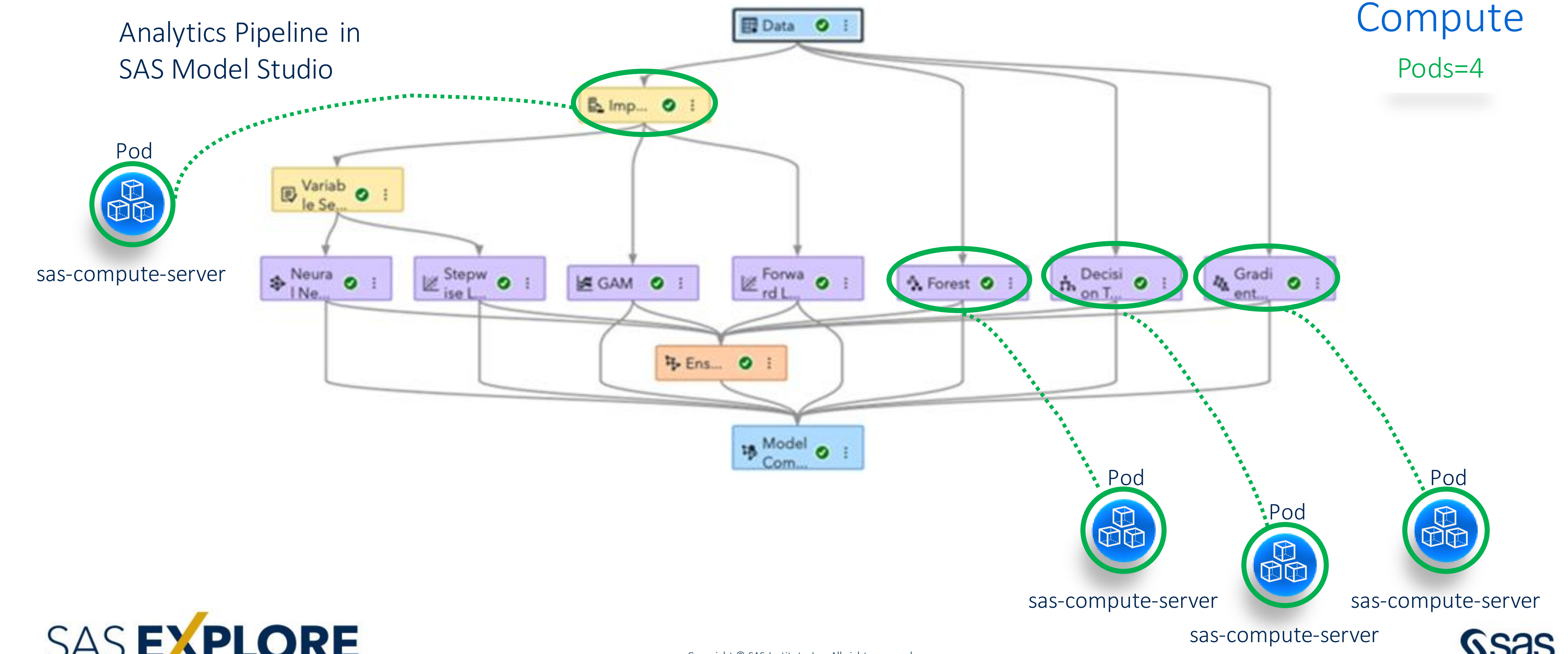
SAS Viya Platform

Improving Concurrency Performance

Compute

Pods=4

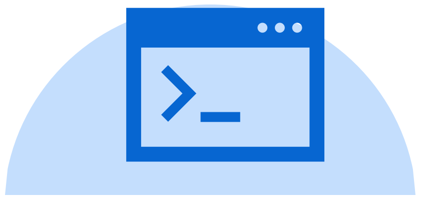
Analytics Pipeline in
SAS Model Studio



SAS Viya Platform

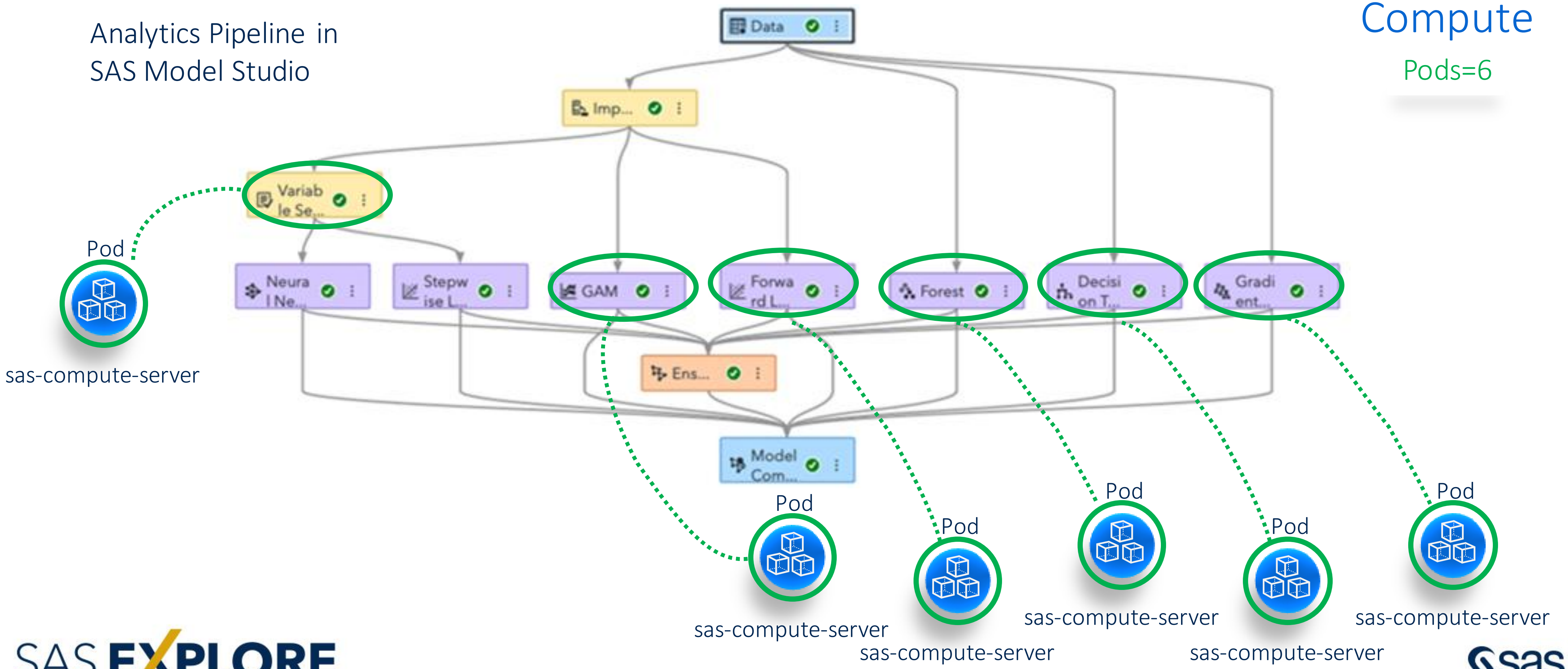
Improving Concurrency Performance

Analytics Pipeline in SAS Model Studio



Compute

Pods=6



SAS EXPLORE

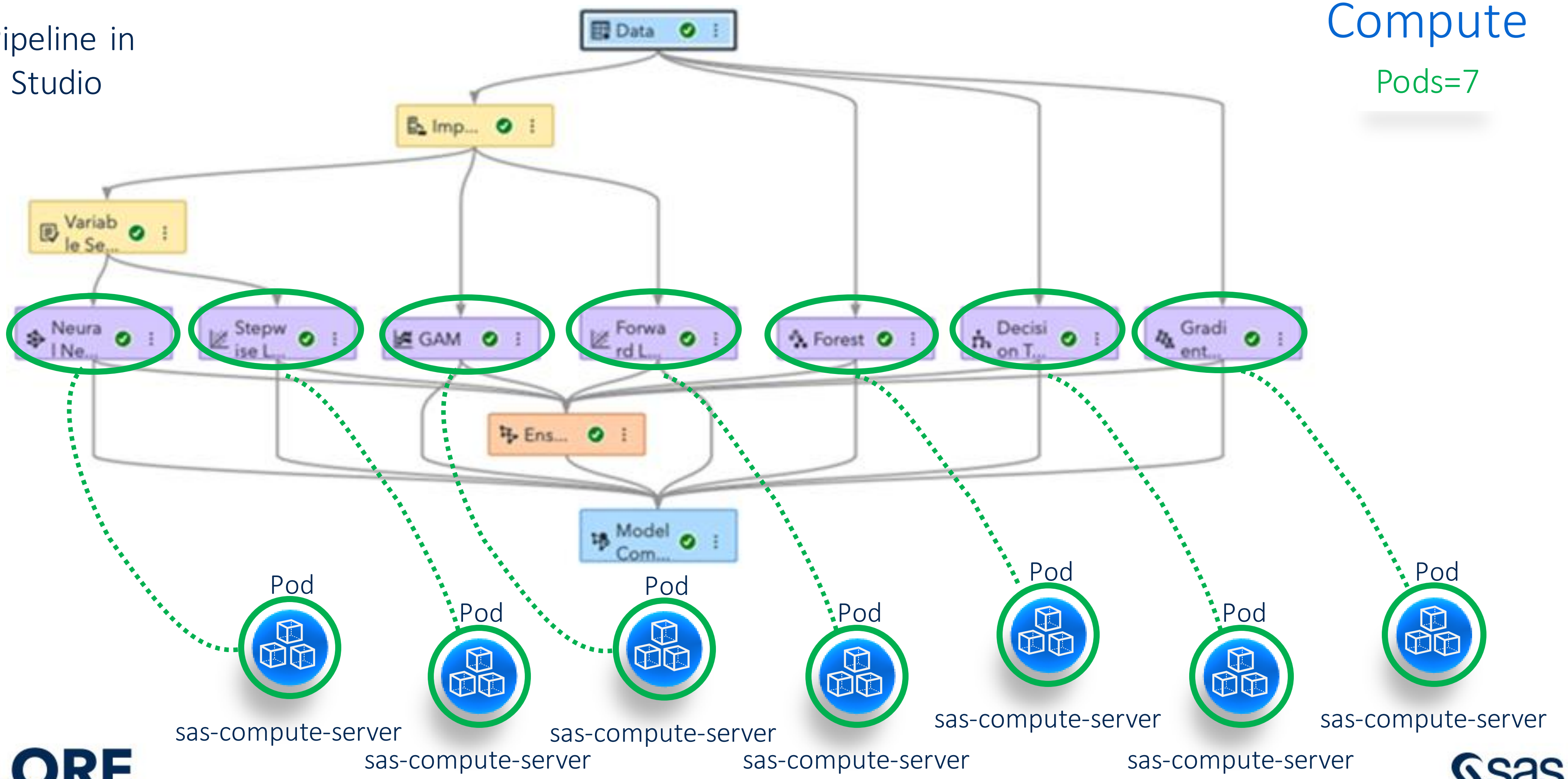
SAS Viya Platform

Improving Concurrency Performance

Analytics Pipeline in SAS Model Studio

Compute

Pods=7



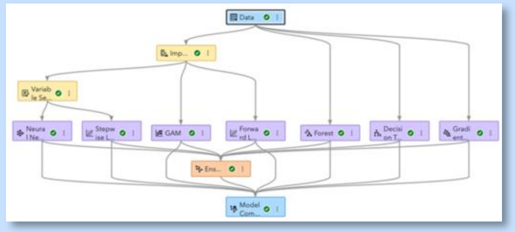
SAS EXPLORE

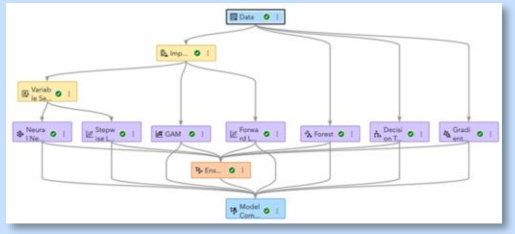
SAS Viya Platform

Improving Concurrency Performance

Compute

user1

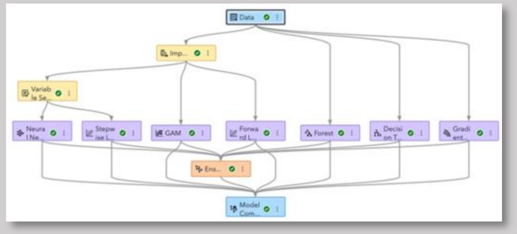


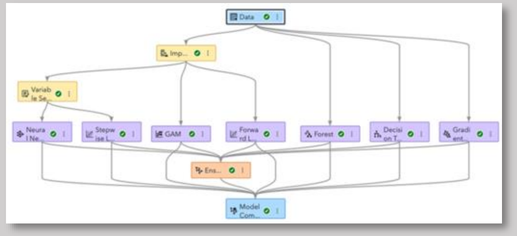


2 pipelines
26 jobs total
14 jobs concurrent

+

user2

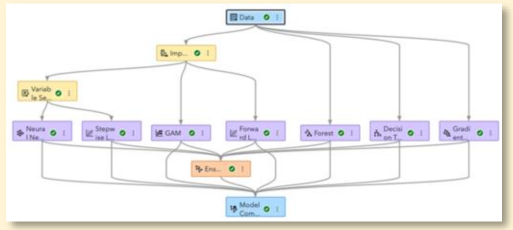


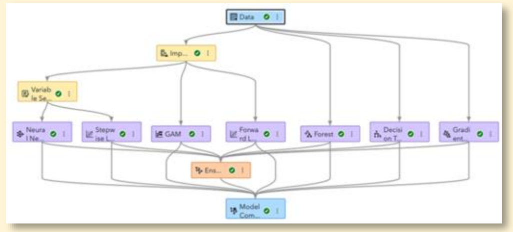


2 pipelines
26 jobs total
14 jobs concurrent

+

user3





2 pipelines
26 jobs total
14 jobs concurrent

=

6 pipelines
78 jobs total
Up to 42 jobs concurrent

SAS EXPLORE

sas

SAS Viya Platform

Improving Concurrency Performance

Compute

How to improve the startup time of SAS Compute Server?

Answer 1: ***Reduce*** the time it takes for Kubernetes to start a new SAS Compute Server pod

SAS R&D has done iterative work on this and achieved nominal gains, but Kubernetes still has its overhead.

SAS Viya Platform

Improving Concurrency Performance

Compute

How to improve the startup time of SAS Compute Server?

Answer 2: *Eliminate* the time it takes for Kubernetes to start a new SAS Compute Server pod

Well, not really "eliminate" – but instead "pre-start" by defining Reusable Compute Servers.

SAS EXPLORE

SAS Viya Platform

Improving Concurrency Performance

Compute

1. Enable Resuable Compute Servers:
   - Run using a shared service account (not as the individual user)
   - Can specify a minimum number to keep running at all times (and more can be started ad-hoc as needed)
   - Will auto-terminate the pod after a set period of inactivity

   - Eliminate the perceived start-up time for SAS Compute Server so that processes can get started on their work more quickly

SAS Viya Platform

Improving Concurrency Performance

Compute

But wait, there's more. Other parts of SAS Viya need tweaking to take advantage of this increased power.

2. Increase the maximum number of processes for SAS Launcher:
   - With SAS Workload Management, adjust the Maximum Jobs Allowed for SAS Workload Orchestrator
   - Without SAS Workload Management, adjust the SAS_LAUNCHER_USER_PROCESS_LIMIT environment variable

SAS EXPLORE

SAS Viya Platform

Improving Concurrency Performance

Compute

and

3. Increase the maximum number of concurrent nodes for SAS analytics tools (fyi: “nodes” of the SAS Model Studio pipeline)
   – SAS Environment Manager > Configuration > Definitions > sas.analytics.execution > Maximum Concurrent Nodes
   – Specify a value less than the maximum number of processes previously

SAS EXPLORE

SAS Viya Platform

Improving Concurrency Performance

For details and more, see:

communities.sas.com > SAS Communities Library >

Improving Concurrency Performance in SAS Viya

https://communities.sas.com/t5/SAS-Communities-Library/Improving-Concurrency-Performance-in-SAS-Viya/ta-p/784542

SAS Viya

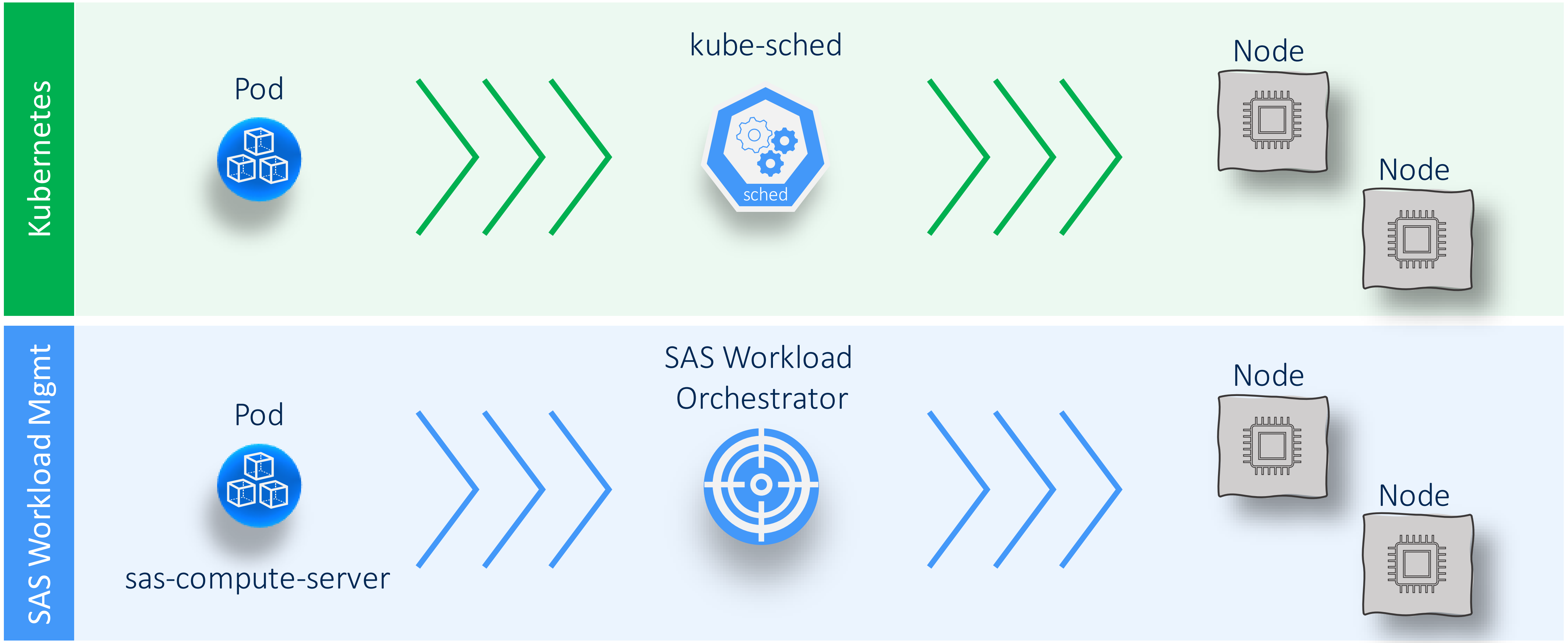
SAS Workload Management

Beyond the SAS Viya platform’s nominal configuration, SAS Workload Management provides additional advanced controls to direct SAS Programming Runtime Environment workloads.

Backend SAS Compute Servers as well as discrete programming tasks are run as Jobs (k8s pods) which are assigned to Queues and scheduled with appropriate Priority to Hosts (k8s nodes) that have the required Resources (for example, k8s labels).

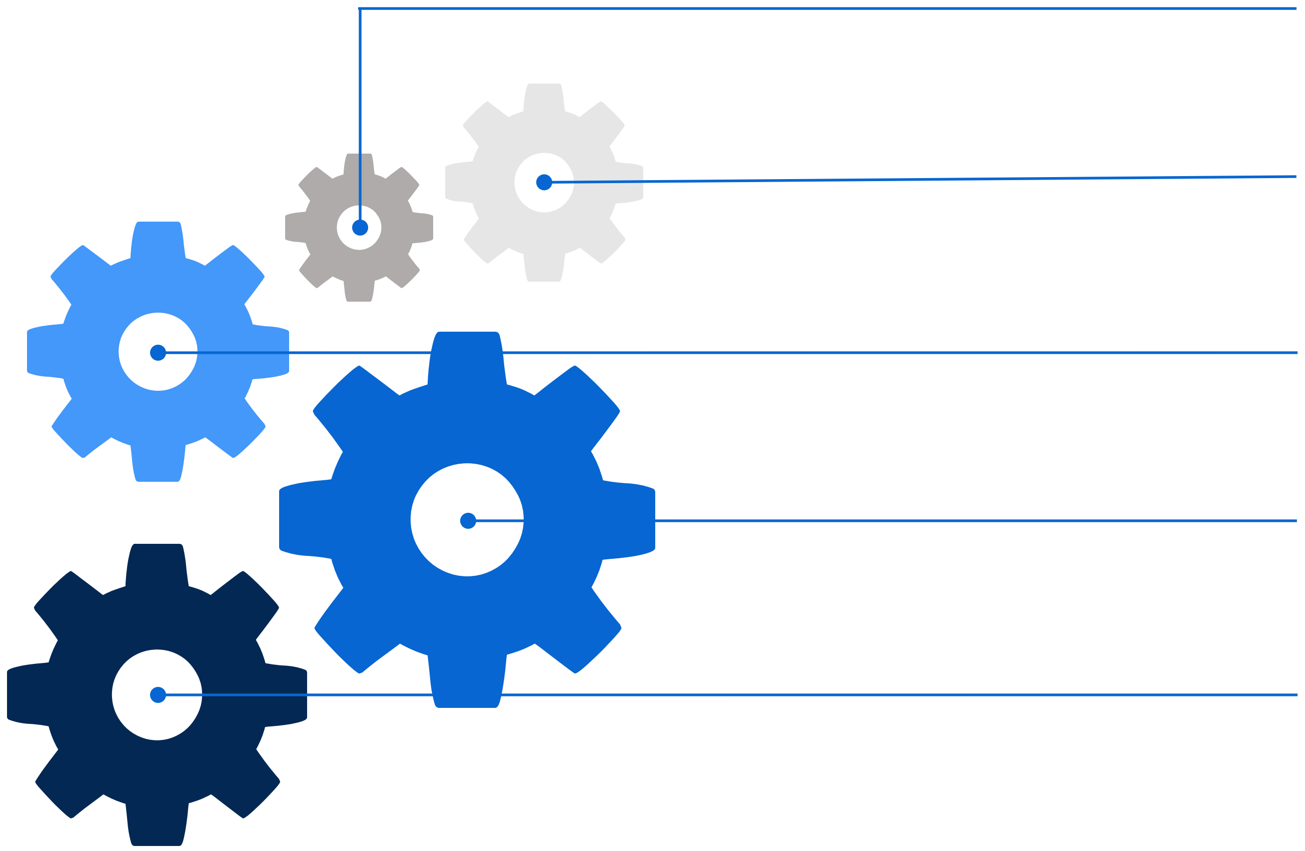
SAS Workload Management

Not using the Kubernetes scheduler for SAS Compute Servers



SAS Workload Management

SWO monitors numerous factors to schedule jobs



CPU and memory utilization

IO rate and available storage

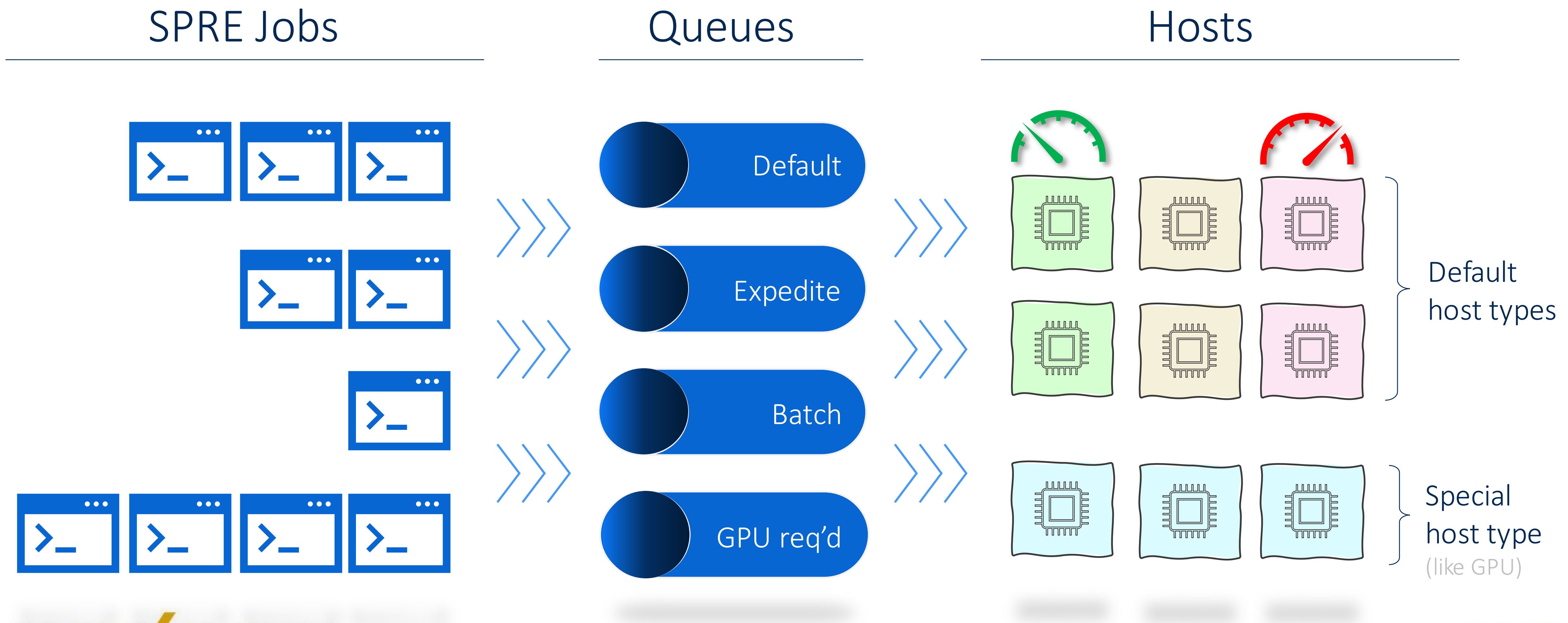
Resources used by other pods not started by SWO

Host attributes (job slots, GPU, or other features)

Other user-defined criteria

SAS Workload Management

SWO manages jobs with queues to select hosts



SAS Viya

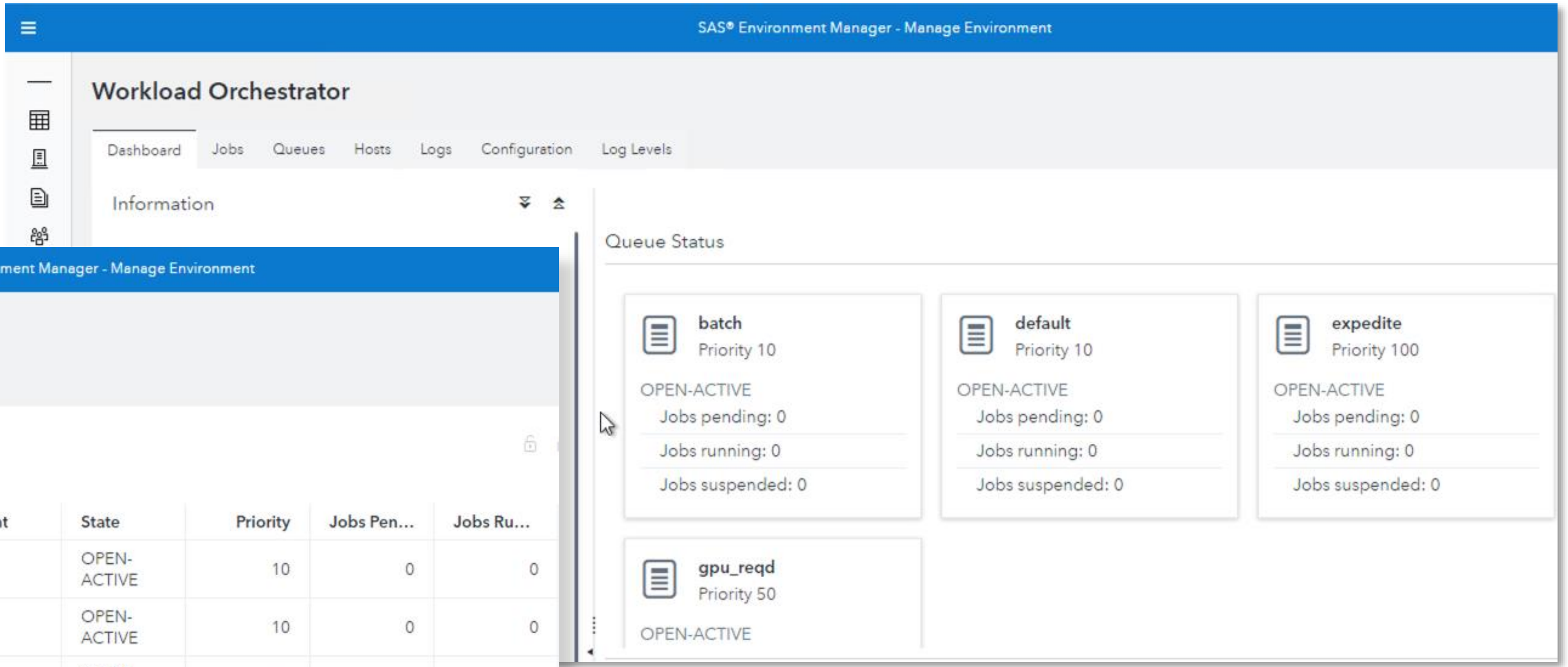
SAS Workload Management

Furthermore, SAS Workload Management provides a shift in ability to manage the workload from the Kubernetes administrator to the SAS Viya administrator and SAS Viya users.

This allows for the people who use the software and understand the processing objectives at the lowest level to manage the most effective use of the environment.

SAS Workload Management

SWO brings cluster resource administration to users



SAS® Environment Manager - Manage Environment

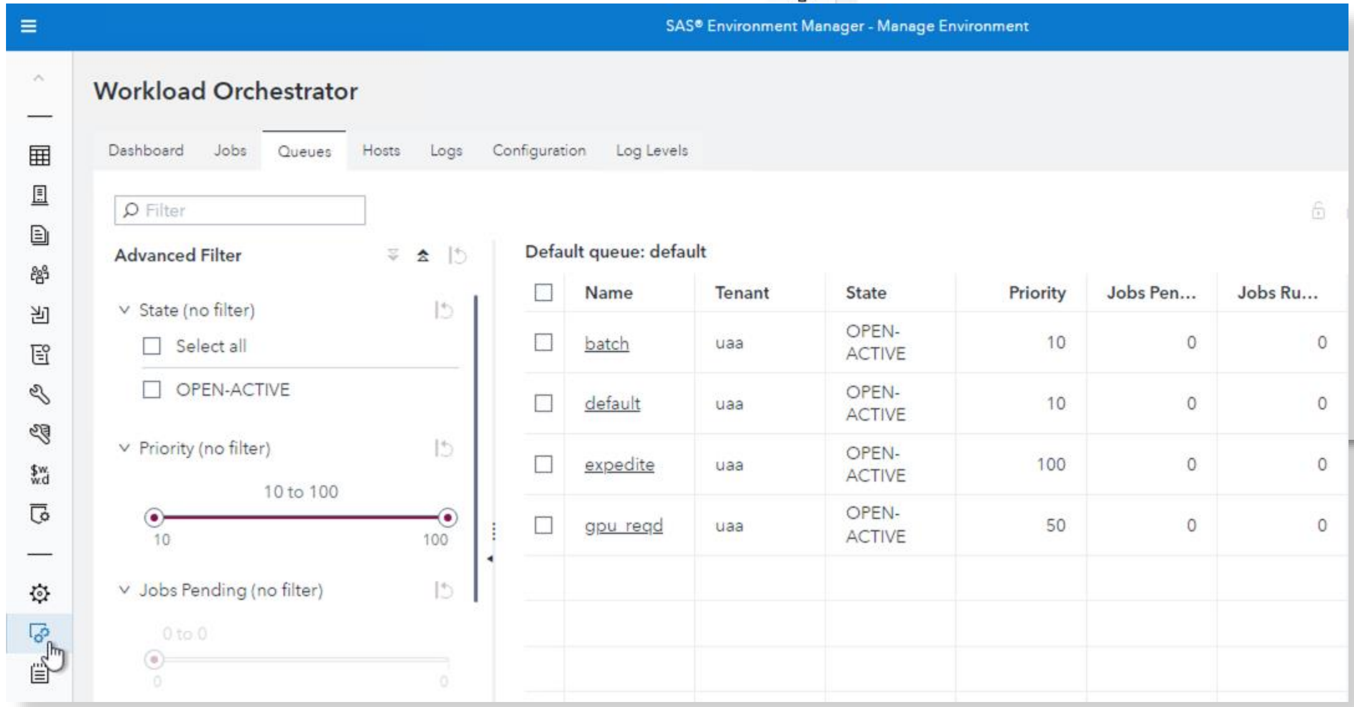
Workload Orchestrator

batch
Priority 10
OPEN-ACTIVE
Jobs pending: 0
Jobs running: 0
Jobs suspended: 0

default
Priority 10
OPEN-ACTIVE
Jobs pending: 0
Jobs running: 0
Jobs suspended: 0

expedite
Priority 100
OPEN-ACTIVE
Jobs pending: 0
Jobs running: 0
Jobs suspended: 0

gpu_reqd
Priority 50
OPEN-ACTIVE



SAS® Environment Manager - Manage Environment

Workload Orchestrator

State (no filter)
Select all
OPEN-ACTIVE

Priority (no filter)
10 to 100
10 — 100

Jobs Pending (no filter)
0 to 0
0 — 0

Default queue: default

Name	Tenant	State	Priority	Jobs Pen...	Jobs Ru...
batch	uaa	OPEN-ACTIVE	10	0	0
default	uaa	OPEN-ACTIVE	10	0	0
expedite	uaa	OPEN-ACTIVE	100	0	0
gpu_reqd	uaa	OPEN-ACTIVE	50	0	0

SAS EXPLORE

SAS Viya Platform

and SAS Workload Management

Compute

SAS Viya integrates most of its components with Kubernetes for workload management and service availability.

SAS Viya provides additional controls for managing backend SPRE servers (for example, sas-compute-server) to ensure it has necessary resources.

SAS Workload Management delivers advanced capabilities to optimize the backend SPRE servers for best efficiency and performance.

SAS Viya

CAS

CAS

The SAS Cloud Analytic Service (CAS) is a high-performance in-memory analytics engine. Designed to work with the largest volumes of data while maintaining availability and responsiveness, CAS doesn’t conform to the ephemeral pod approach that’s typical in Kubernetes environments.

A CAS server’s workload is managed primarily by data distribution across CAS nodes. Additional CAS servers can be provisioned for more capacity and workload separation.

CAS

Deployment Options

CAS

The SAS Viya platform runs with at least one CAS server. There are two modes of deployment for CAS:

1. SMP = single node CAS server

   runs as a single pod in the Kubernetes cluster, often deployed to a dedicated node to fully utilize CPU and memory.

CAS

Deployment Options

CAS

Two modes of deployment:

2. MPP = multi-machine CAS server
   runs as multiple pods in the Kubernetes cluster with controller and worker roles. Often deployed to a dedicated node pool to fully utilize CPU and memory.

   Data loaded to CAS is evenly distributed across the workers which acts as the primary form of workload management.

CAS

Deployment Options

CAS

Multiple CAS servers can run as well:

SMP CAS

sas-cas-server-default-controller

MPP CAS

sas-cas-server-shared-controller
sas-cas-server-shared-worker-0
sas-cas-server-shared-worker-1
… … …

Personal CAS

sandboxed inside
sas-compute-server…

SAS EXPLORE

sas

CAS

Deployment Options

CAS

By default, “auto-resourcing” for CAS is employed…

Each CAS server pod (controllers and workers) will request:

- 70% of the node’s total memory
- Number of cores = 1+floor(total node cores/2)
  for a node with 8 cores, then 5 are requested

This effectively reserves 1 node per CAS server pod.

From cas-server-default-worker pod spec

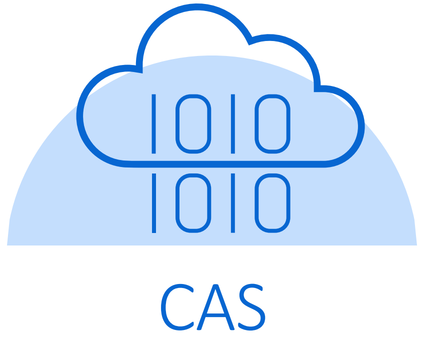
```
resources:
   limits:
     cpu: 7910m
     memory: 30922732Ki
   requests:
     cpu: '5'
     memory: '21979814297'
```

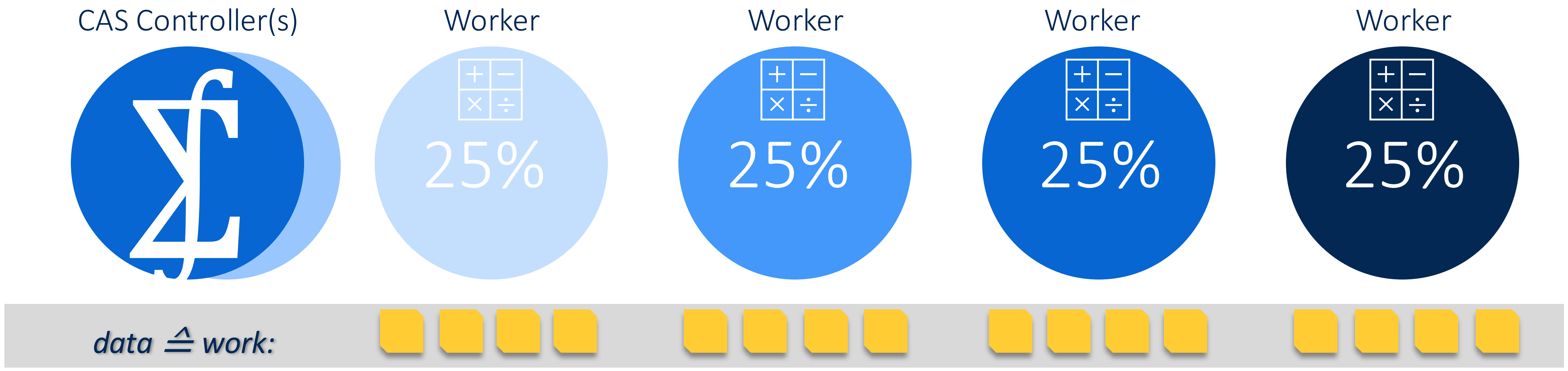
SAS EXPLORE

sas

MPP CAS

Data distribution ≙ Workload distribution





- Each CAS worker gets approximately the same amount of data (over-simplification of ideal).
- So, when an analytics action is performed, each CAS worker has the same amount of work to do.
- Final results are collated by the CAS controller and returned to the client

CAS

Workload design

CAS

By design:

- All nodes in a node pool are the same instance type with the same resources (CPU, memory, storage, I/O)
- CAS server pods are scheduled to run on nodes labeled and tainted for the CAS workload class exclusively
- Each CAS pod effectively isolates itself to a dedicated host by requesting a majority of node resources (and with pod anti-affinity)
- Data (and hence, work) is distributed evenly across CAS workers

CAS

Workload design

CAS

In other words:

- The MPP CAS server is designed on the basis that each worker has the same amount of work to do on an equal quantity of data utilizing the same number of resources.

SAS EXPLORE

sas

CAS

Workload design

CAS

Workload *inside* an MPP CAS server is managed by data distribution and resource assignment.

Workload *across* multiple CAS servers is managed by users and administrators.

Kubernetes "keeps the lights on", initially scheduling CAS pods to their intended nodes per labels, taints, requests, limits, and so on.

SAS EXPLORE

Coda

More information

Want to reduce the deployment and/or administration overhead of your SAS Viya solution? Try SAS Cloud:

- Software as a Service
  You don't have to manage infrastructure, operating systems or software. You simply sign up, log in and get to work, focusing on your analytic challenges.

- Managed Application Services
  We'll help you design and manage your cloud services and solutions – in our cloud, your cloud, or on-site.

More information

While at SAS Explore, be sure to visit us in the Innovation Hub:

Platform

- Cloud
- Administration
- Move to Viya

Programming

- Open-source programming
- Clients

Customer Success & Engagement

- Technical Support
- SAS Communities
- Customer feedback

All Things Data

- Data engineering
- Data connectivity

SAS EXPLORE

sas

More information

Try it for yourself!

Come to my hands-on workshop

Delivering Powerful Analytics Capabilities with the SAS Viya Platform and Kubernetes

5pm, Wednesday in Juniper 4

SAS EXPLORE

End

Thank you!

SAS EXPLORE