

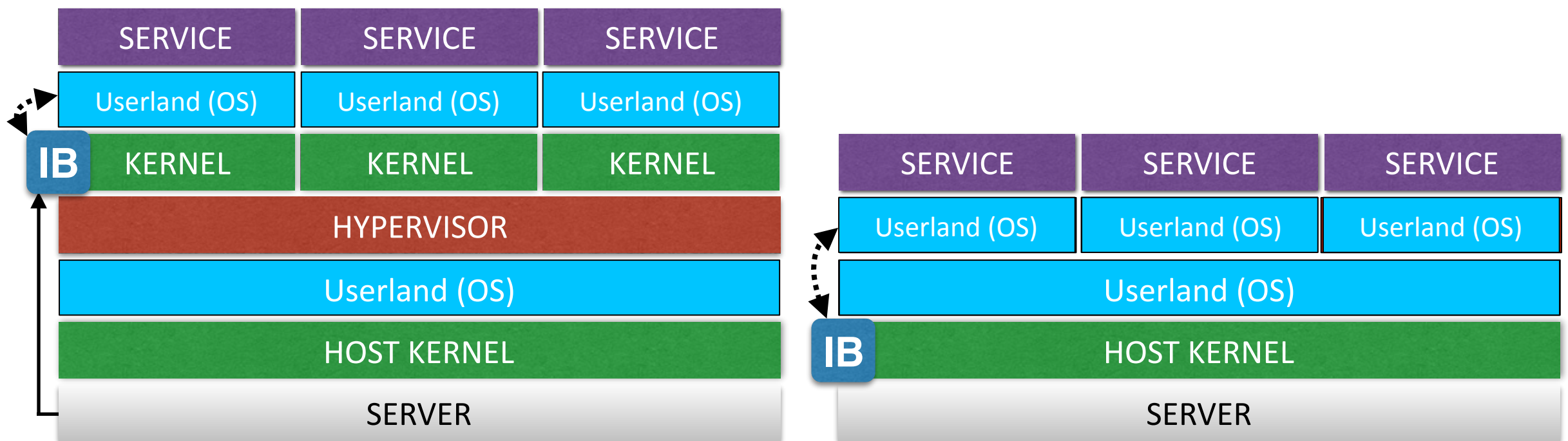
Multi-host containerised HPC cluster

The new Docker networking put into action to spin up a SLURM cluster

The Bits and Pieces...

Docker

- Containers do not spin up their own kernel
 - ▶ All containers share the same host kernel
 - ▶ they are separated via Kernel Namespaces
 - ▶ constrained using CGroups

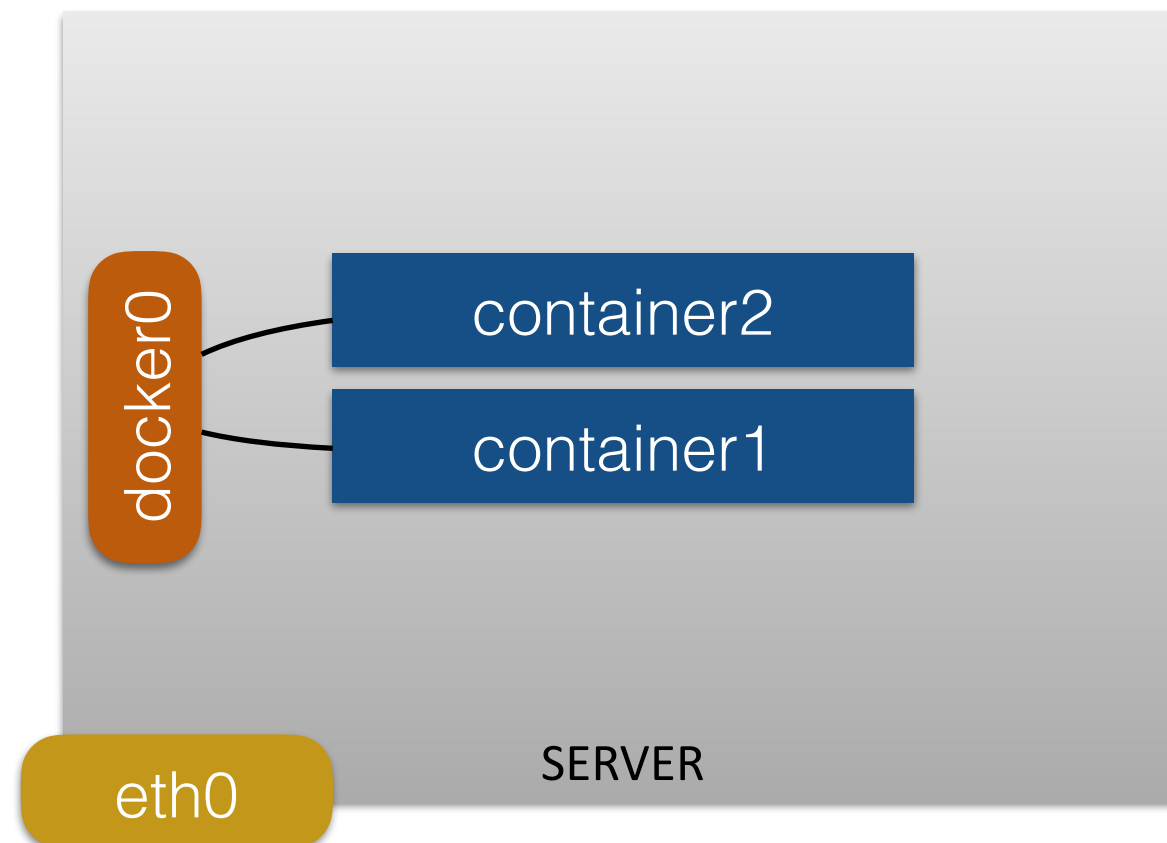


Traditional Virtualisation

Containerisation

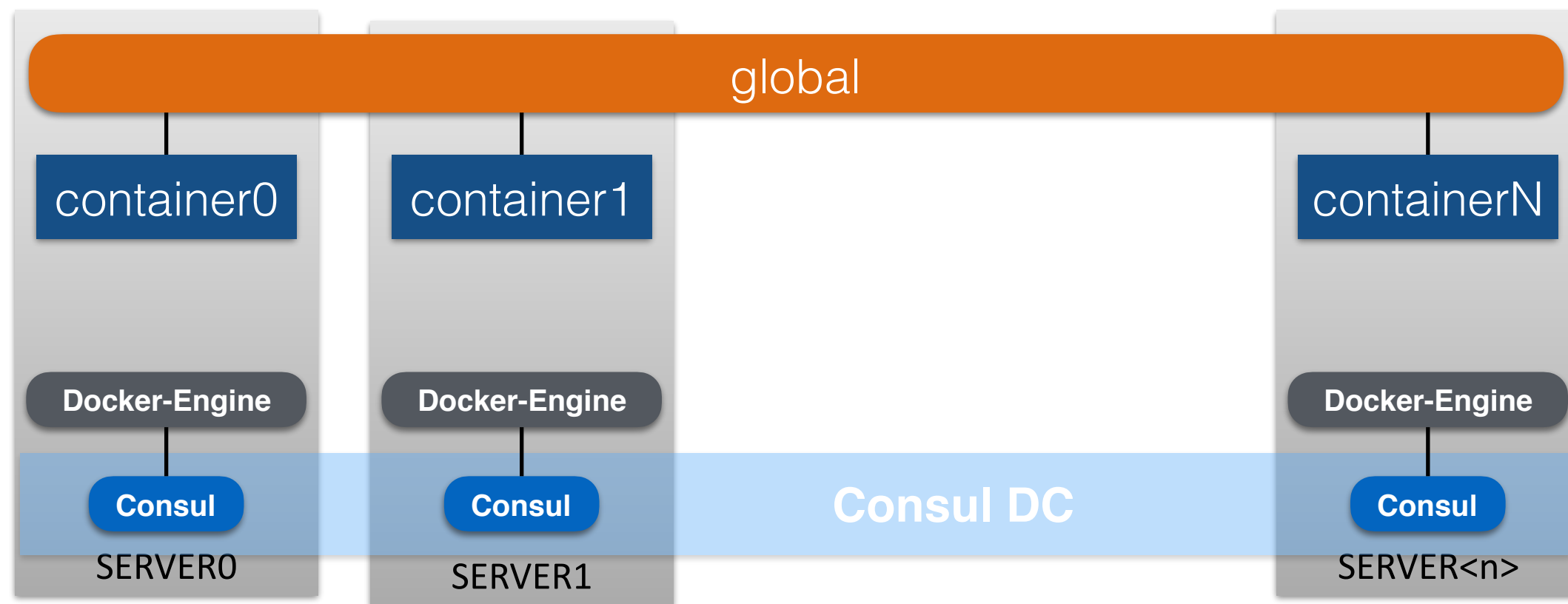
Docker

- Docker-Engine as a Container Runtime
 - ▶ creates/starts/stops/manipulates/removes containers - all through RESTful API
 - ▶ handles Namespaces, CGroups, IP connectivity, bind-mounts, etc.



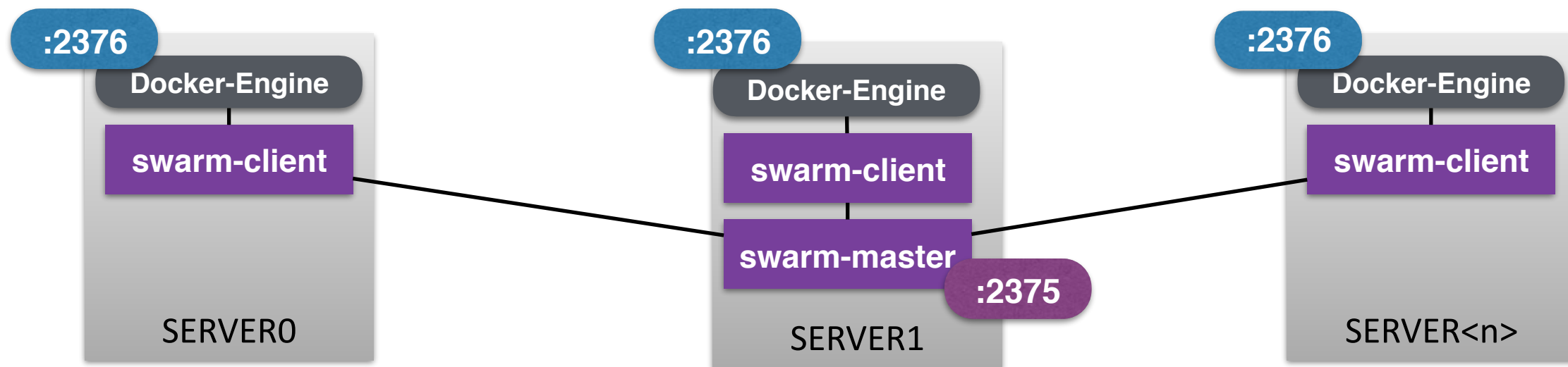
Docker Networking

- Docker Networking spans networks across engines
 - ▶ KV-store to synchronise (Zookeeper, etcd, Consul)
 - ▶ VXLAN to pass messages along (macvlan might take over)



Docker Swarm

- Docker Swarm proxies docker-engines
 - ▶ serves an API endpoint in front of multiple docker-engines
 - ▶ does (simple) placement decisions.



Docker Swarm [cont]

```
[root@venus001 ~]# docker -H docker1:2375 info | grep --color=never "(192|Container|^\\w+)"
Containers: 50
Images: 374
docker1: 192.168.12.11:2376
  └ Containers: 5
venus001: 192.168.12.181:2376
  └ Containers: 12
venus002: 192.168.12.182:2376
  └ Containers: 6
venus003: 192.168.12.183:2376
  └ Containers: 4
venus004: 192.168.12.184:2376
  └ Containers: 4
venus005: 192.168.12.185:2376
  └ Containers: 5
venus006: 192.168.12.186:2376
  └ Containers: 5
venus007: 192.168.12.187:2376
  └ Containers: 3
venus008: 192.168.12.188:2376
  └ Containers: 6
CPUs: 76
Total Memory: 266.5 GiB
Name: a7d177accc81
[root@venus001 ~]#
```

query docker-swarm

Introduce new Technologies

Introducing new Tech



Introducing new Tech

Self-perception when introducing new tech...



Introducing new Tech

... not always the same as the perception of others.



Docker Buzzword Chaos!



Pitfalls

[incomplete list]

- You say Docker - perception is 'basically VMware'
 - ▶ VMs were easy to shoehorn in legacy workflow, containers might break it
 - ▶ spans environments: laptop, dev-cluster, staging, prod
- Not everyone is a unicorn, I do
 - ▶ NOT want special distributions
 - ♦ useful for elasticity (AWS) and green-flied deployment, not so much for a on-premise datacenter w/ legacy in it.
 - ▶ want to leverage existing <stuff>
 - ♦ security (ssh infrastructure), user authentication
 - ♦ installation workflow, monitoring, logging
 - ▶ want to keep up with upstream docker ecosystem
 - ♦ networking, volumes
 - ♦ features of docker-engine, -compose, -swarm


Reduce to the max!

Put a Docker **on** it!

- Leverage existing install/configuration workflow
 - ▶ Kickstart + ansible-run
- Don't focus on corner cases
 - ▶ postpone snowflake-container (e.g. needs multi-tenant IB usage)
 - ▶ User namespace: we get there (eventually)
- HPC environments assumptions
 - ▶ single-tenant
 - ▶ focus on performance

The Setup

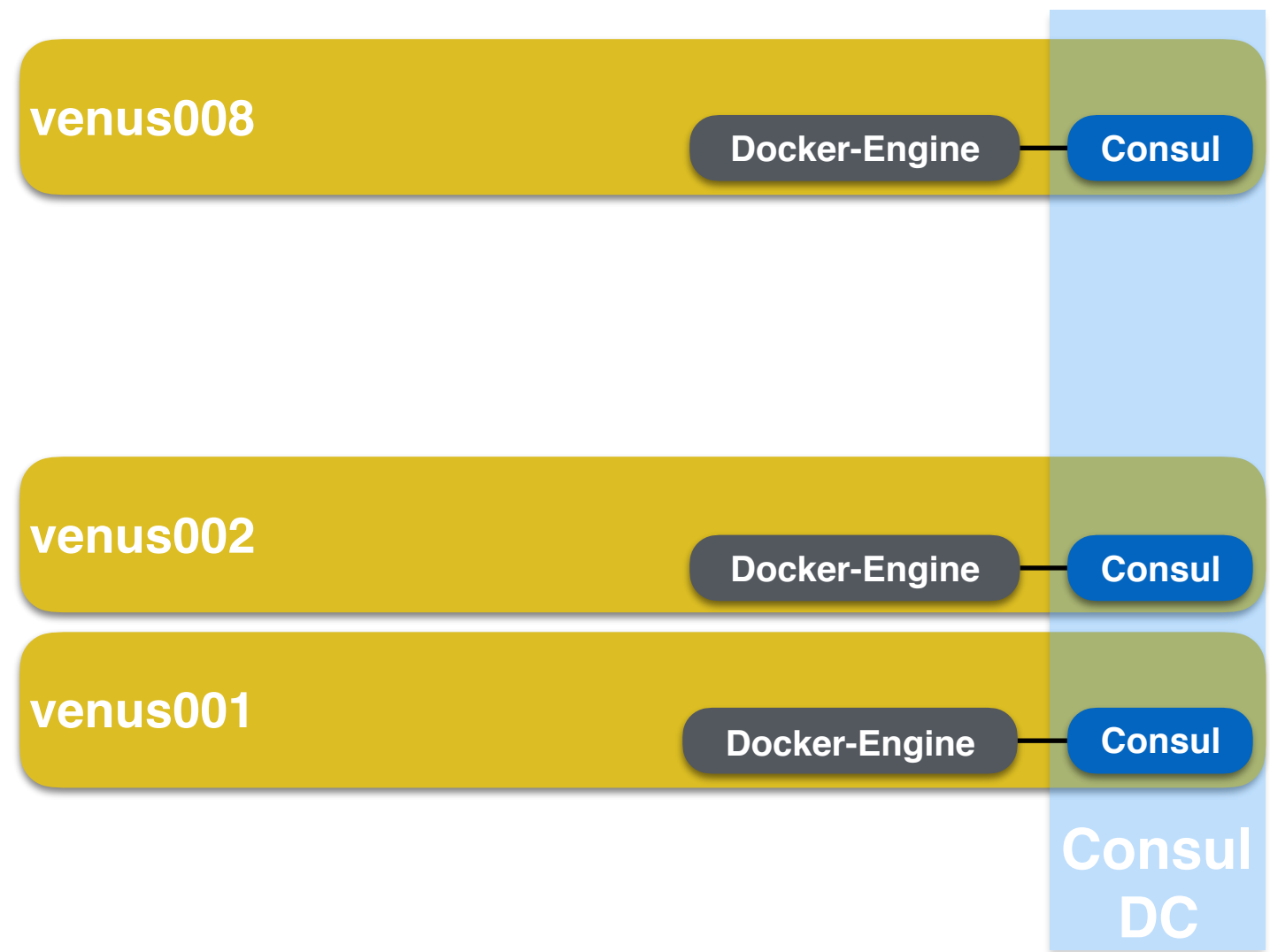
Testbed

- Hardware (courtesy of )
 - ▶ 8x Sun Fire x2250, 2x 4core XEON, 32GB, Mellanox ConnectX-2)
- Software
 - ▶ Base installation
 - CentOS 7.2 base installation (updated from 7-alpha)
 - ▶ Ansible
 - consul, sensu
 - docker v1.9.1, docker-compose
 - docker SWARM



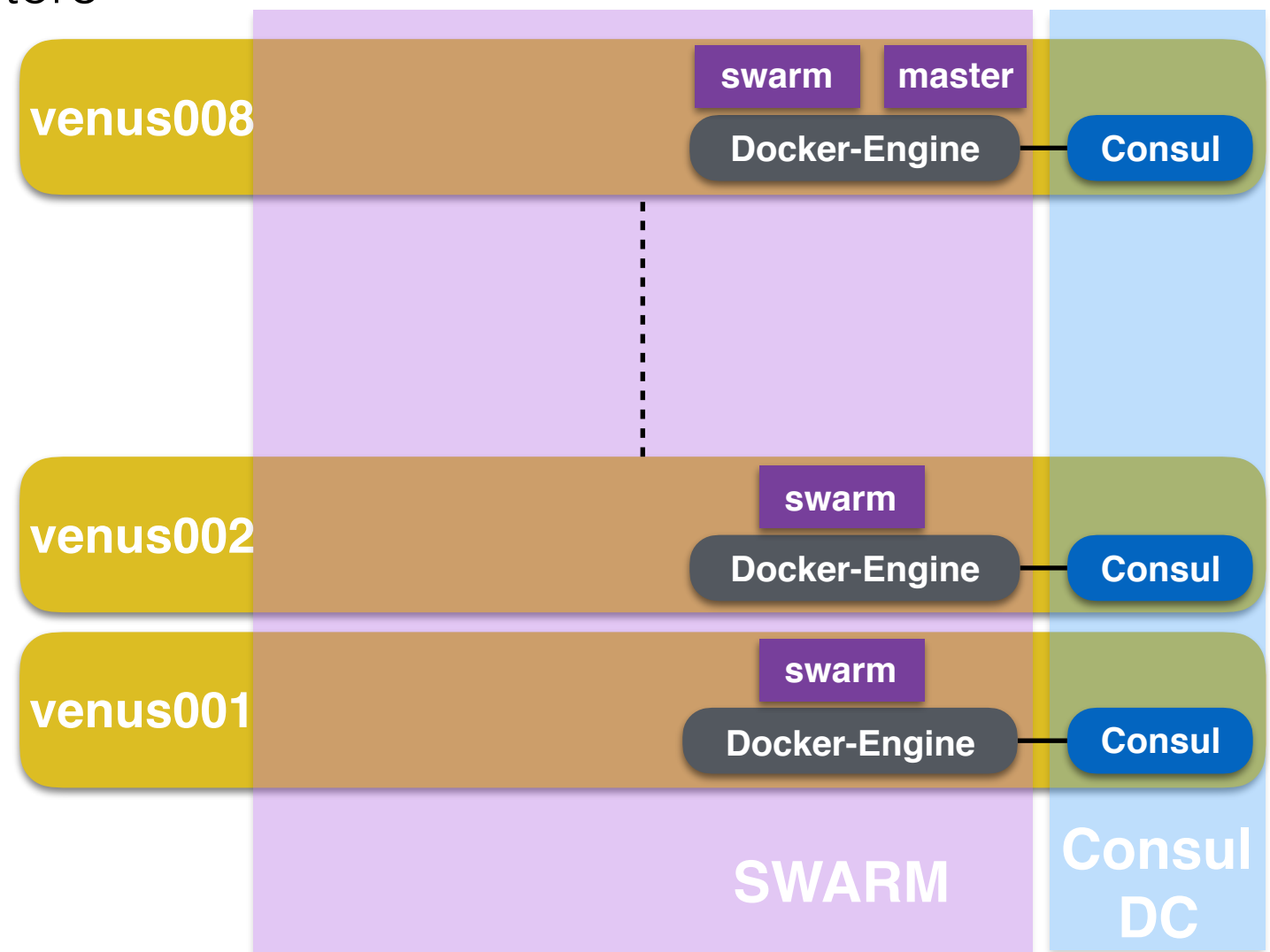
Docker Networking

- Synchronised by Consul



Docker SWARM

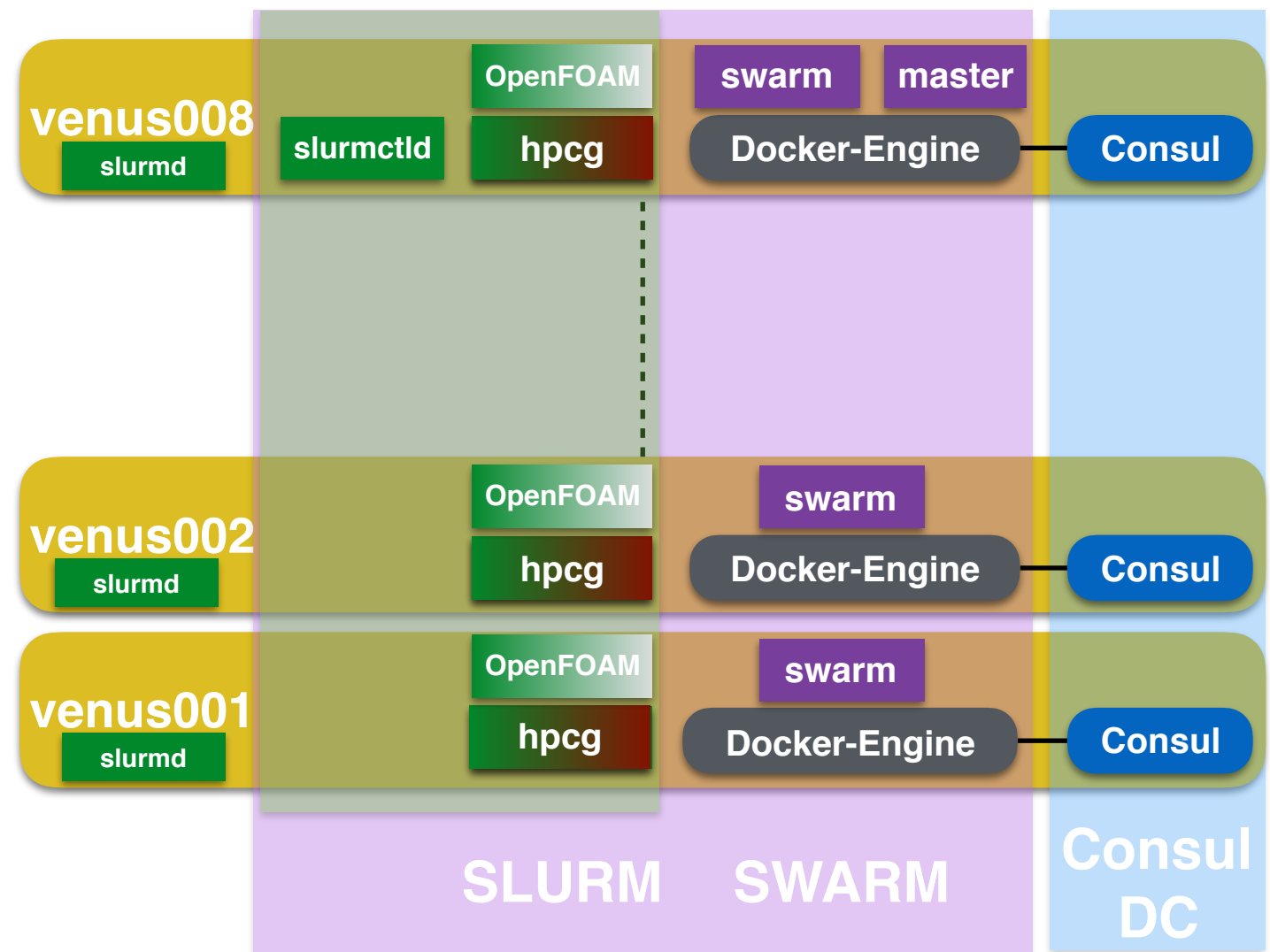
- Docker SWARM
 - ▶ Synchronised by Consul KV-store



SLURM Cluster

- SLURM within SWARM

- ▶ slurmd within app-container
- ▶ pre-stage multiple container
- ▶ spawn at job-start (pre-hook)

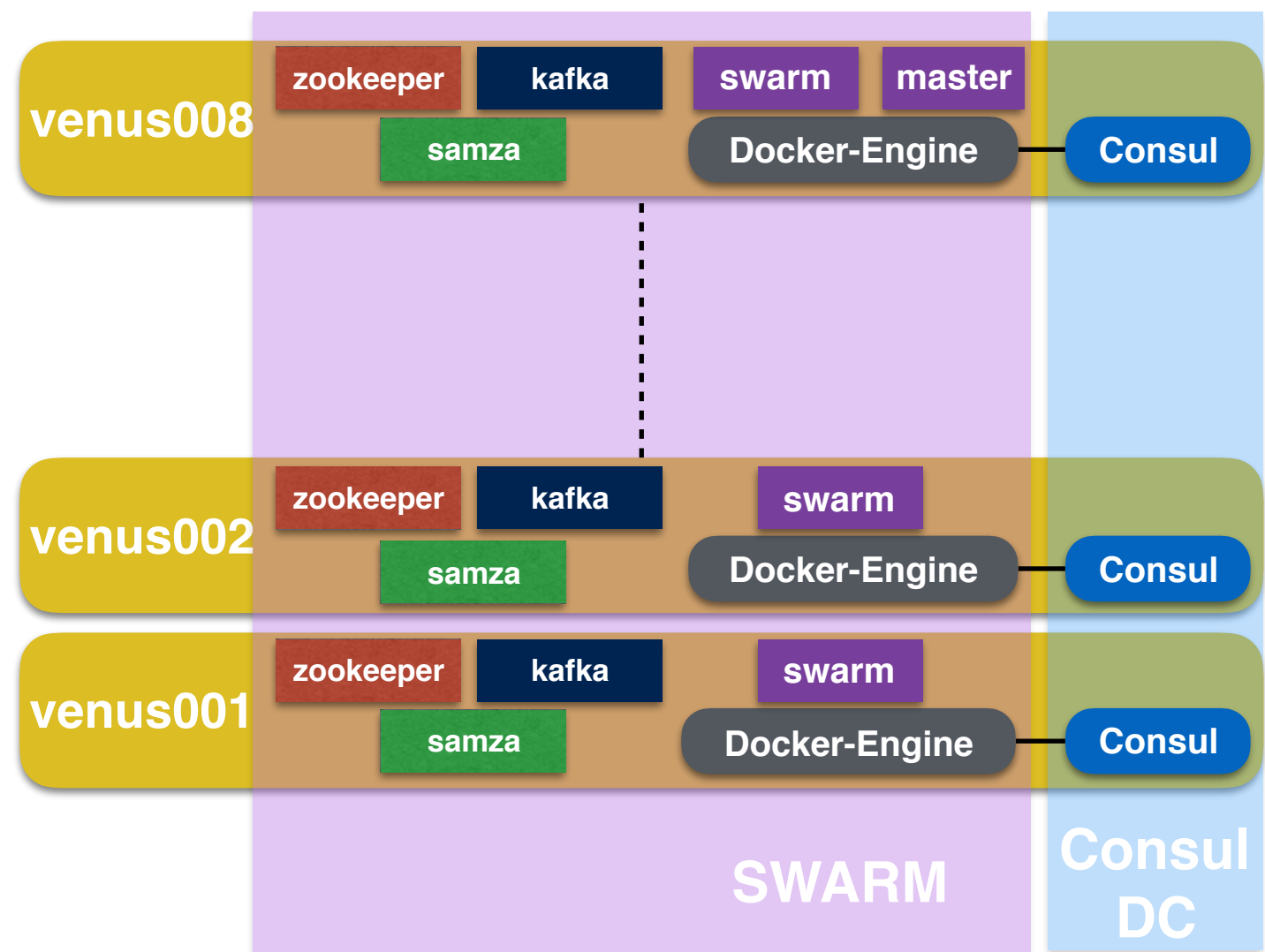
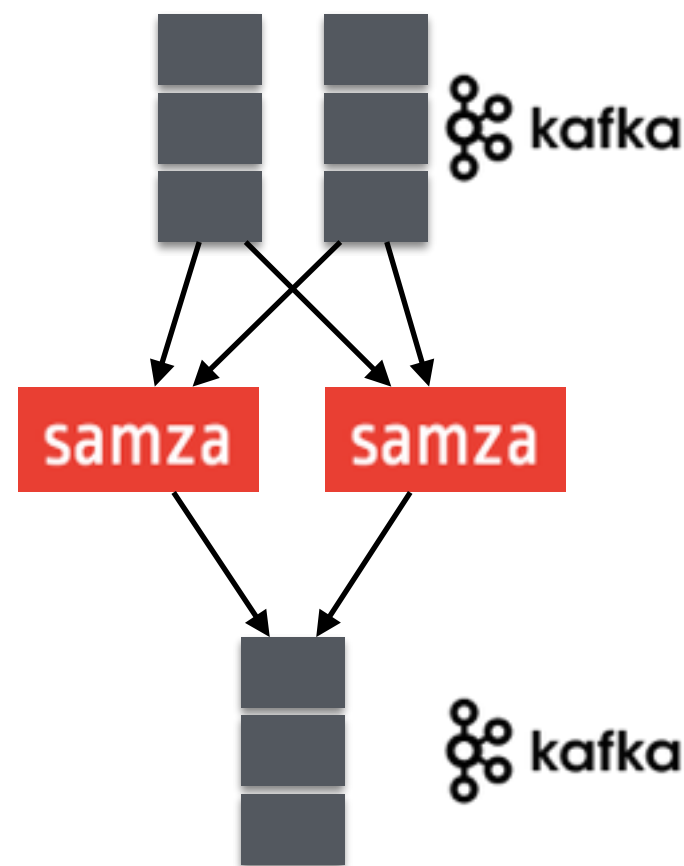


SLURM Cluster [cont]

```
venus001 rc=0 docker1:2375 provisioning (master) # docker exec -ti venus001/hpcg1 sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
all*      up     infinite    8    idle hpcg[1-8]
odd       up     infinite    4    idle hpcg[1,3,5,7]
even      up     infinite    4    idle hpcg[2,4,6,8]
venus001 rc=0 docker1:2375 provisioning (master) # docker exec -ti venus001/hpcg1 srun -N8 hostname
hpcg7
hpcg3
hpcg2
hpcg6
hpcg4
hpcg5
hpcg8
hpcg1
venus001 rc=0 docker1:2375 provisioning (master) #
```

Samza Cluster

- Distributed Samza
 - ▶ Zookeeper and Kafka cluster
 - ▶ Samza instances to run jobs



Outta time!

SENSU

The image shows the Uchima SENSU dashboard and a Slack notification window. The dashboard displays a list of clients and detailed information for a specific client, VENUS001. The Slack window shows a series of status messages from the sensu bot.

Uchima SENSU Dashboard - Clients List

Name	IP	Events
docker1	10.0.2.15	
sensu	172.17.0.7	
venus001	192.168.12.181	
venus002	192.168.12.182	

Uchima SENSU Dashboard - Client Details (VENUS001)

Client Information:

- VENUS001** (a few seconds ago)
- Default**
- _id:** Default/venus001
- address:** 192.168.12.181
- name:** venus001
- subscriptions:** infiniband, physical
- timestamp:** 2016-01-17 17:30:03
- version:** 0.21.0

Checks and Outputs:

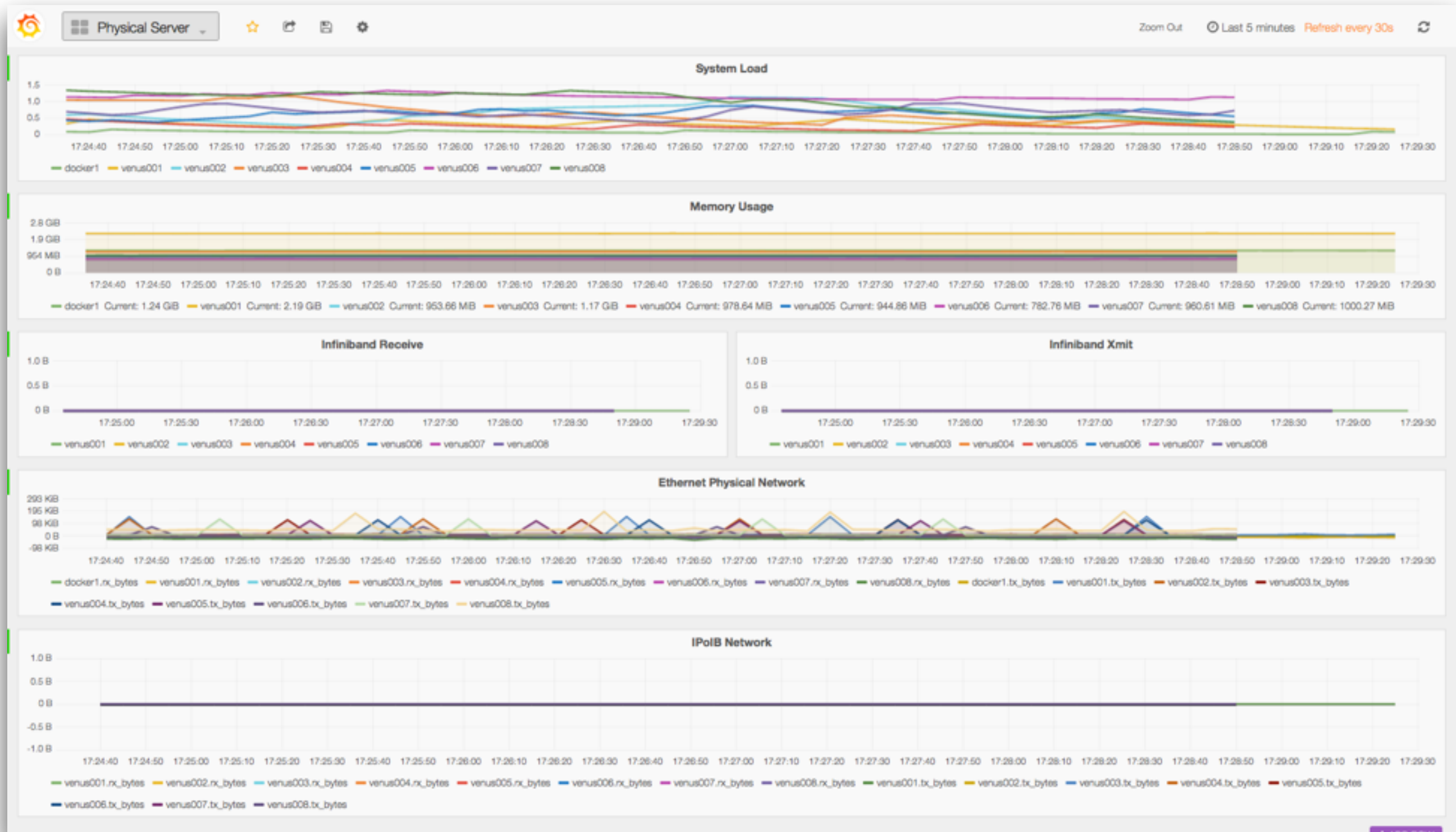
Check	Output	Time
net_metrics	venus001.net.veth2a24fb8.tx_packets 0 1453048230 ...	a few seconds ago
keepalive	Keepalive sent from client 4 seconds ago	a few seconds ago
cpu_metrics	venus001.cpu.total.user 21678627 1453048230 ...	a few seconds ago
load_metrics	venus001.load_avg.one 0.39 1453048230 ...	a few seconds ago
ib_metrics	venus001.infiniband.port1.SymbolErrorCounter 0 14530482...	a few seconds ago
mem_metrics	venus001.memory.total 29340794880 1453048230 ...	a few seconds ago

Slack #sensu Channel

Today

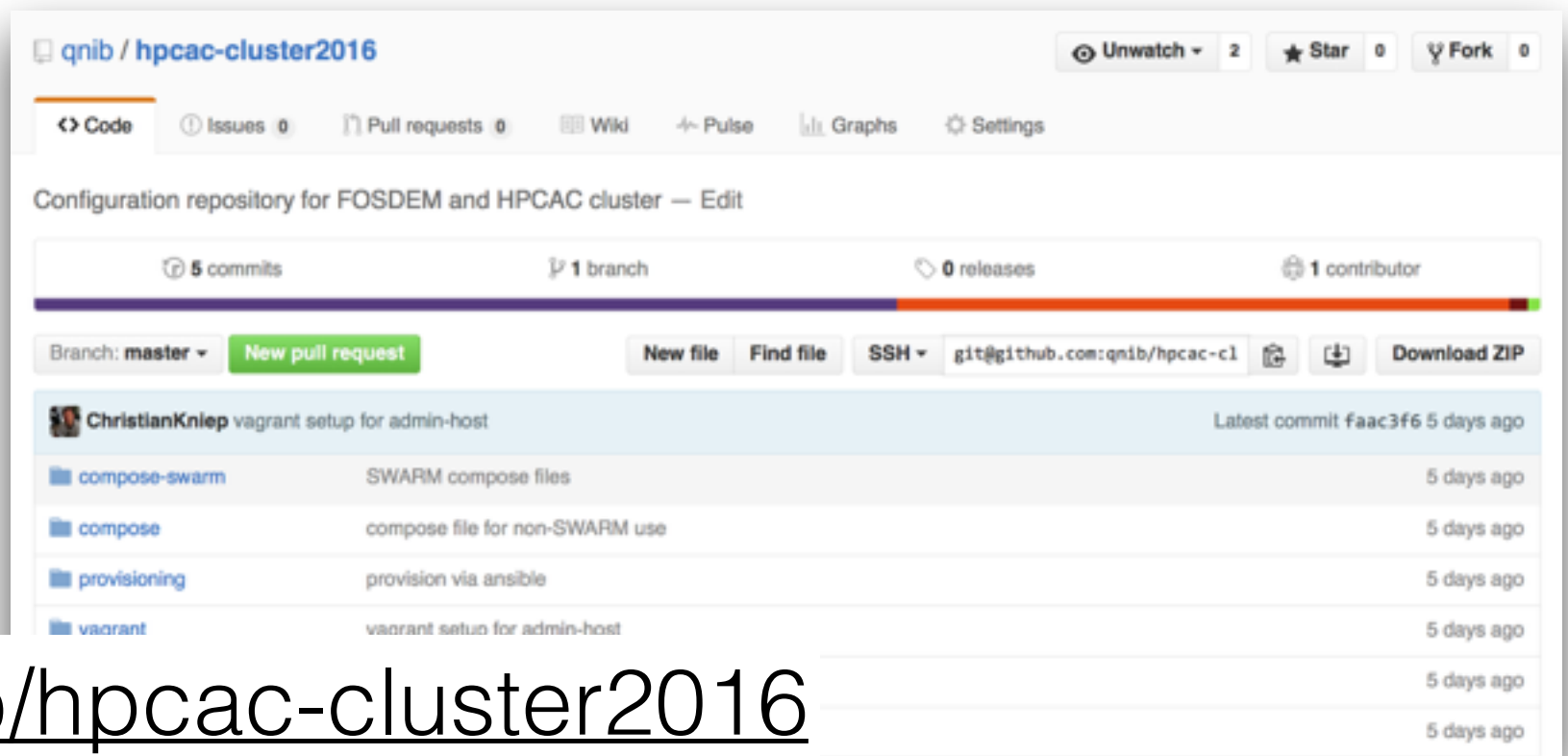
- sensu BOT 8:01 PM**
 - 10.0.2.15 - WARNING**
docker1/keepalive: No keepalive sent from client for 148 seconds (≥ 120): 10.0.2.15 : physical
 - 10.0.2.15 - CRITICAL**
docker1/keepalive: No keepalive sent from client for 208 seconds (≥ 180): 10.0.2.15 : physical
- sensu BOT 8:31 PM**
 - 10.0.2.15 - CRITICAL**
docker1/keepalive: No keepalive sent from client for 1958 seconds (≥ 180): 10.0.2.15 : physical
- sensu BOT 8:45 PM**
 - 10.0.2.15 - OK**
docker1/keepalive: Keepalive sent from client 5 seconds ago : 10.0.2.15 : physical

Metrics



Recap aka. IMHO

- Using vanilla docker tech on-top of any distribution
 - ▶ keep up with the ecosystem and prevent vendor/ecosystem lock-in
- 80/20 rule
 - ▶ have caveats on the radar but don't bother too much
 - ▶ everything is so fast moving - it's hard to predict
- Don't scare away stakeholders
 - ▶ KISS
 - ▶ reuse workflow and infrastructure
 - ▶ solution and not problem driven



<https://github.com/qnib/hpcac-cluster2016>

Q&A



eGalea Conference (Pisa)
20/21 April 2016



<http://qnib.org>