

# Containers and Virtualization for the HPC cluster

**Nicolas Kowski**

Kubernetes as the single API.

Virtualization in the context of sensitive data

20 May 2021



# Intro.


## User side:

- singularity containers.
- Vm's as login on isolated medical research tenants.

## **Previous situation.**

- Mix of technologies.**
- Classic security concept.**
- K8s POC and for Web - Apps.**
- Thinking about improve all this.**

## Motivation & transition.

- HPC sites attacked !? 
- recreate the whole infrastructure.
- re invent the security concept
- Hands on, under pressure.

# The New concept. What and how we did it?

## *Kubernetes as the brain.*



kubernetes

### Base Stack

- Netboot
- Repository
- DHCP
- DNS
- Mirror
- Syslog (fwd to SEC)
- Consul
- ...

### Core services:

- Vault (in the way to 0 Trust)
- Admin nodes, netboot (mother of chickens)

### Multi tenancy

- LSF Master (for now is a K8's VM)
- Login nodes for secure tenants (K8's VM)
- Netapp integration (at VM and at k8s level)
- Lustre Integration (at vm)
- Compute nodes (K8's VMs)

## Where we “applied” this recipes?

Re deployed this stack in the 3 clusters we have.

- Euler + Leonhard Open.
- Leomed - Medical research secure cluster.



## Leomed, Special case.

### *Coexisting user facing VMs with Microservices.*

**How do we provide, beside all the micro services, VM's?**

How do we make them “live” in the isolated specific Vans (VM's and containers)?

- Automate it
- Integrate the with the ETHz network.
- Operate with limited human resources (Ideally don't run yet another complex platform for this)

# The answer KubeVirt *Why?*



# KubeVirt

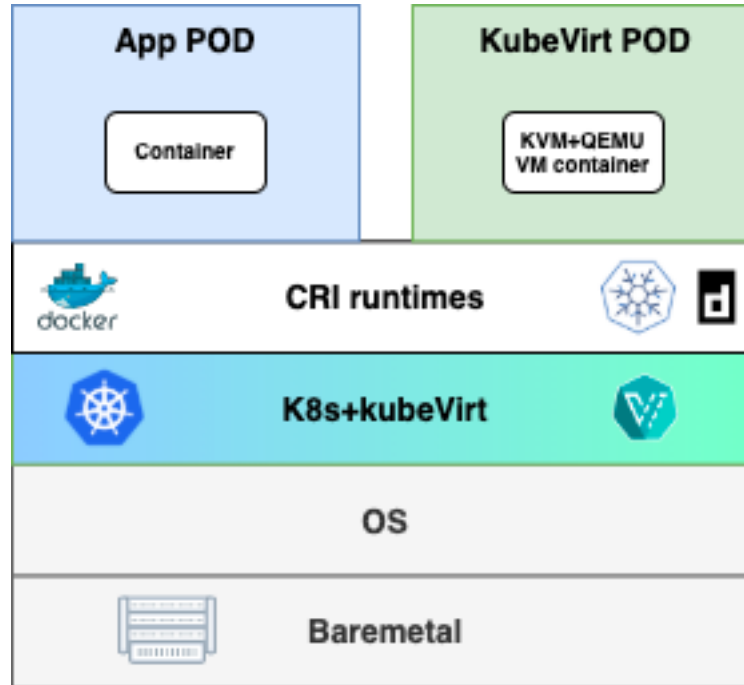


KubeVirt technology addresses the needs of development teams that have adopted or want to adopt [Kubernetes](#) but possess existing Virtual Machine-based workloads that cannot be easily containerized.

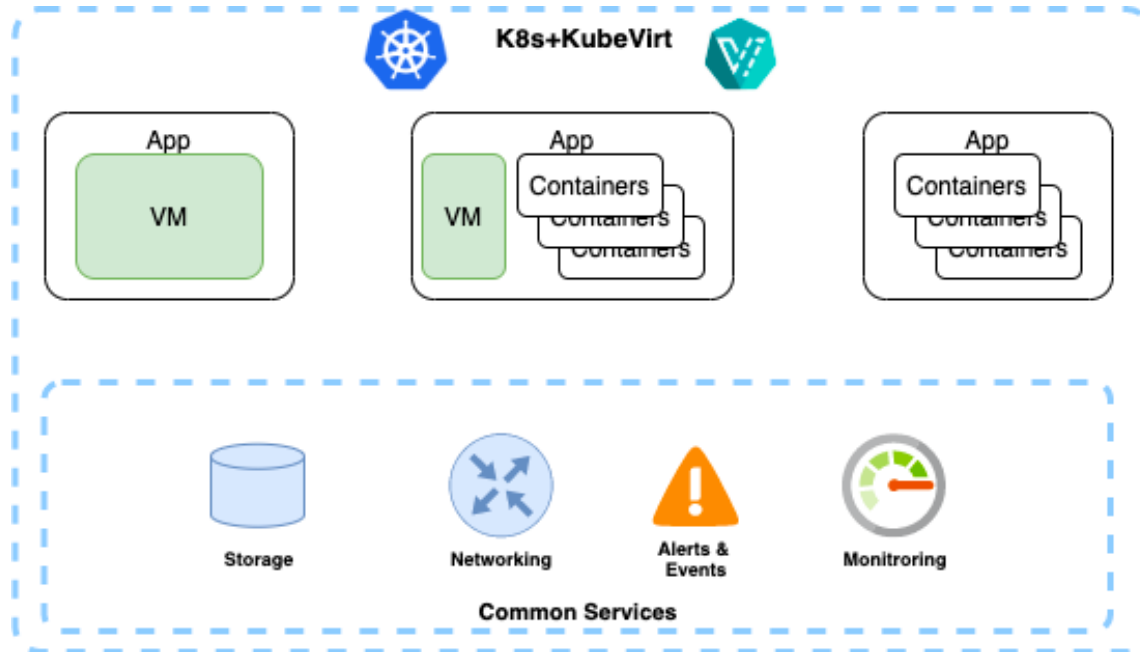
More specifically, the technology provides a **unified development platform where developers can build, modify, and deploy applications residing in both Application Containers as well as Virtual Machines in a common, shared environment.**



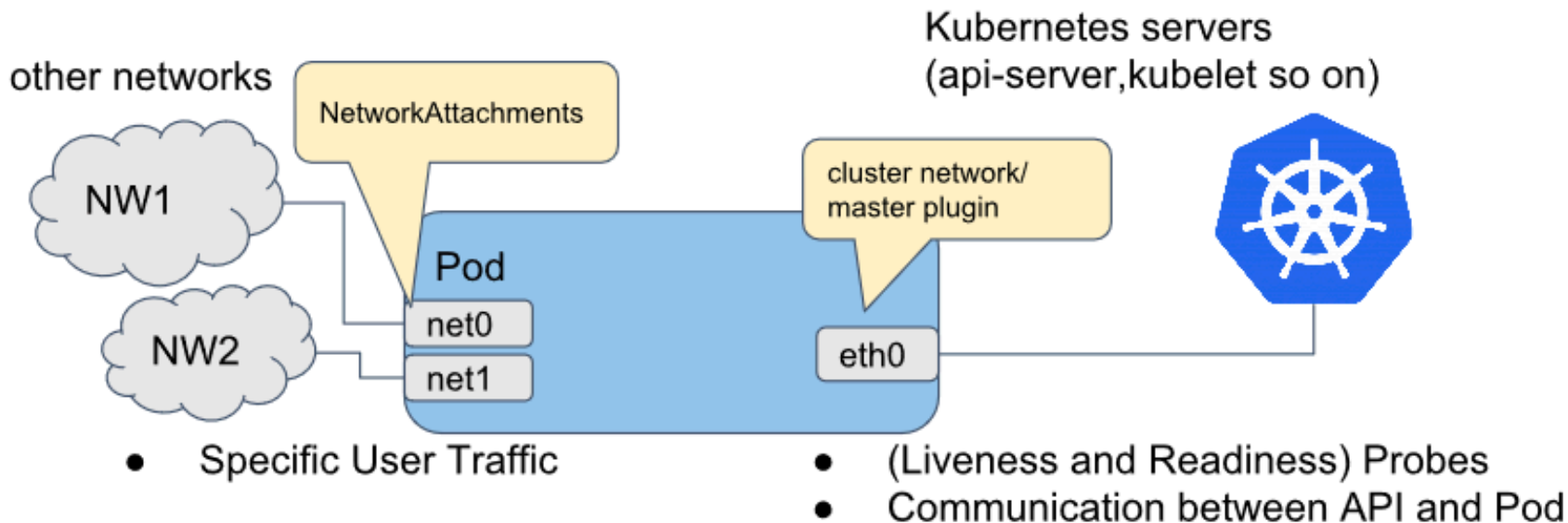
# How can a VM run in K8s?



# What about the other pods/ services?



# How a pod / Vm its deployed in a specific VLAN.



## Key benefits of k8s + Multus + KubeVirt:

- Inherit all the benefits of running something in k8s.
- Extremely powerful and innovative.
- Easy to automate and operate.
- Flexible, programable.
- Live Migrations.
- Same API.
- Biggest Community behind and growing. (vs ON or OS)
- We can implement with the knowhow and resources we have \o/

**Adopters:** Apple - Cisco - CloudFlare - Nvidia ...

# Automation - Tenantctl (Python)

## Creates new tenants

- K8s namespace.
- Hypervisor Bridges (clush + remote code execution)
- Network Attach definition (multus CNI & which bridge the Vm pod will live)
- DHCP server for Access and Fabric (Infiniband network). With leases (from db)
- Ops and Login nodes Vms (kubevirt with a disk based on Netapp)
- Point dns
- Run configuration for OPs and Login nodes. (Cdist conf - starts LSF master)

## Retrieve information and status

- Network information
- Vm status
- Db (which compute nodes this tenant have assigned)

## On development ( On pause. It being picked up by a dedicated team).

- CD (continues deploy). In case we need to modify something. Just git push and ArgoCD will do the rest.
- k8s tenant cluster (overcluster) (suspended)

# Lesson learned

**Don't change too much (except if you have to).**

- Complexity and the learning curve. Shift culture.**
- Human resources needed. Lots of things unfinished.**
- Plus day 2 operations.**

**We are on a mission.**

## Ideas under consideration:

- LSF / Slurm masters as containers
- Login nodes, not a vm, just another container.
- Workers runs a “LSF worker” container, so they are yet another k8s node (meaning a 3k+ kubernetes cluster). Abstraction from the underlying infra.
- Falco - keep improving security.
- K8s on demand - Users want to run something in k8s.

# Questions?

Contact us:

**Nicolas Kowenski:**

<https://www.linkedin.com/in/nicokowenski/>

[nicolas.kowenski@id.ethz.ch](mailto:nicolas.kowenski@id.ethz.ch)

<https://github.com/zakkg3/>

**Steven Armstrong:**

[steven.armstrong@id.ethz.ch](mailto:steven.armstrong@id.ethz.ch)