



KubeCon



CloudNativeCon

Europe 2026

API is the New SSH: Forging a Zero - Trust VM Platform on Kubernetes

Replacing Virtualized Environment with KubeVirt in 2026

Feature | Hard Limits | Kube-OVN vs Cilium

ABOUT ME



Evangelista Tragni

Lead Engineer



SCHEDULE

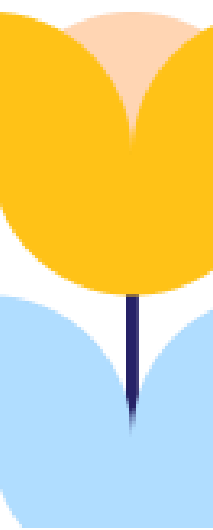
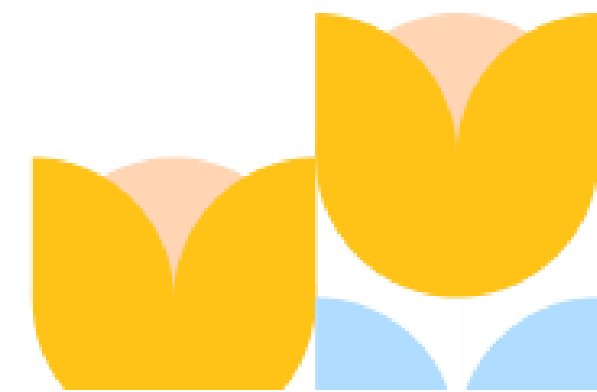
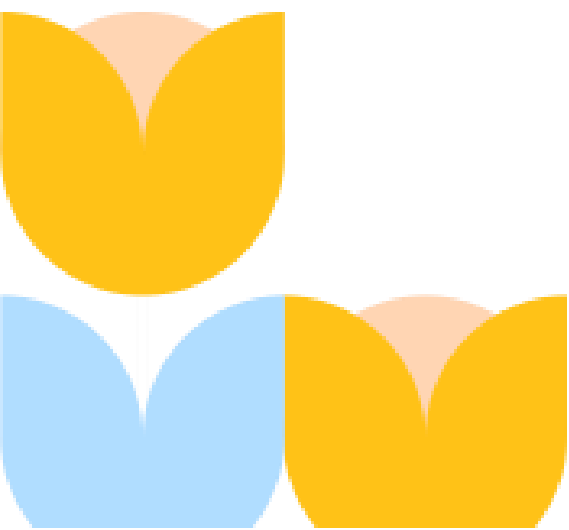
Market Landscape: The shift toward OSS and unified infrastructure.

Core Capabilities: Current KubeVirt features and native Kubernetes integration.

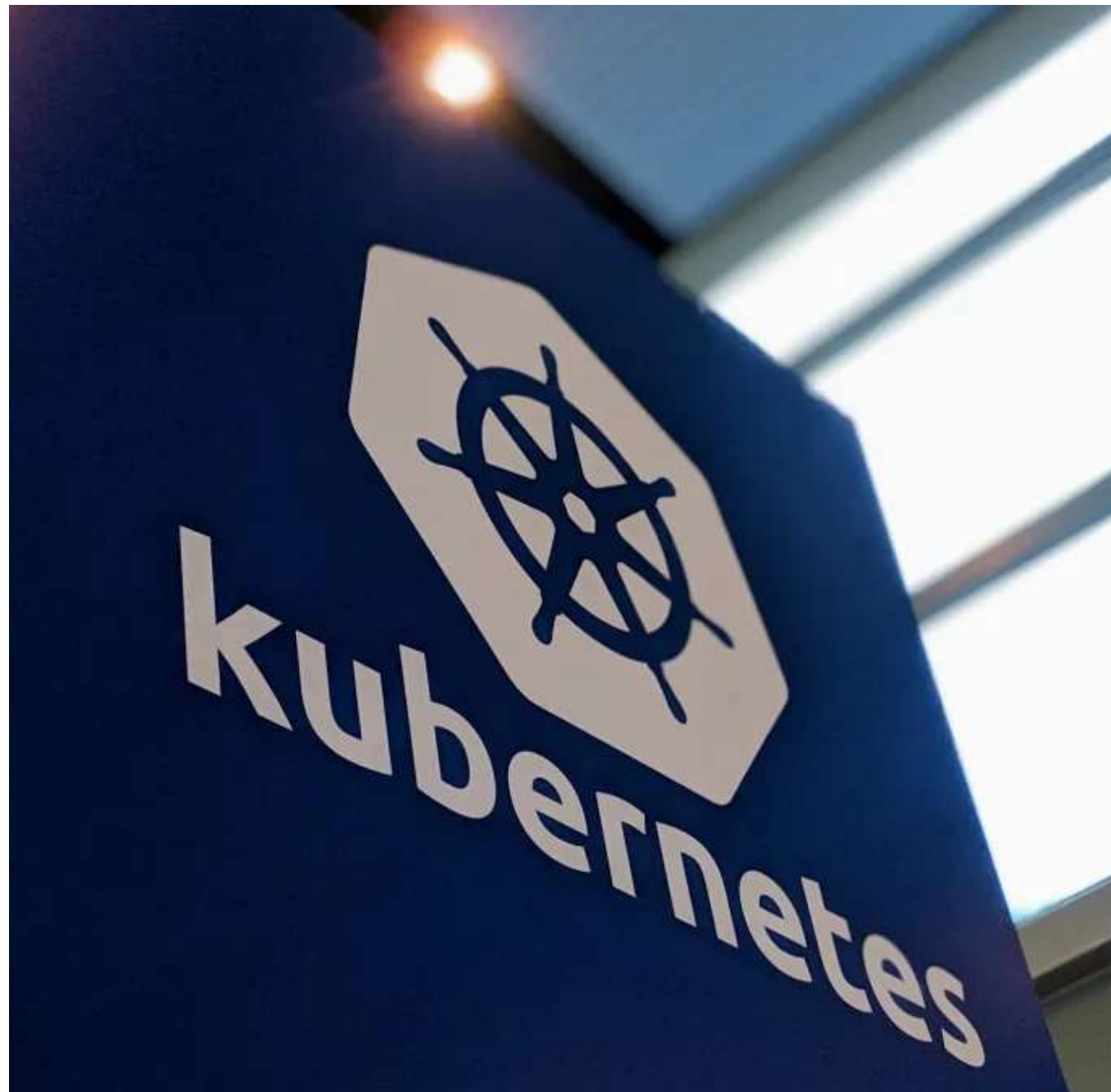
Networking Possibilities: CNI options, Multus, and connectivity strategies.

Operational Constraints: Identifying current limitations and technical hurdles.

Conclusion: Future developments and the path toward full maturity.



THE 2026 LANDSCAPE



The Open Source Wave: Open Source Software (OSS) has moved from the fringes to the core of every industry, becoming the default standard for modern enterprise architecture.

The Economics of Virtualization: Legacy virtualization costs have skyrocketed. The financial burden of maintaining traditional hypervisors has become unsustainable for most IT budgets.

Kubernetes is Everywhere: Regardless of the vendor, nearly every enterprise now operates a Kubernetes environment within their infrastructure it is the new universal baseline.

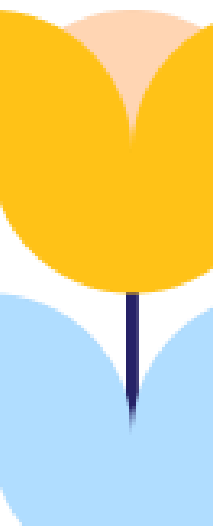
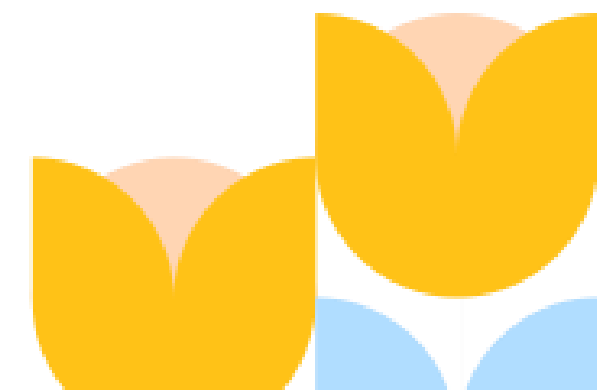
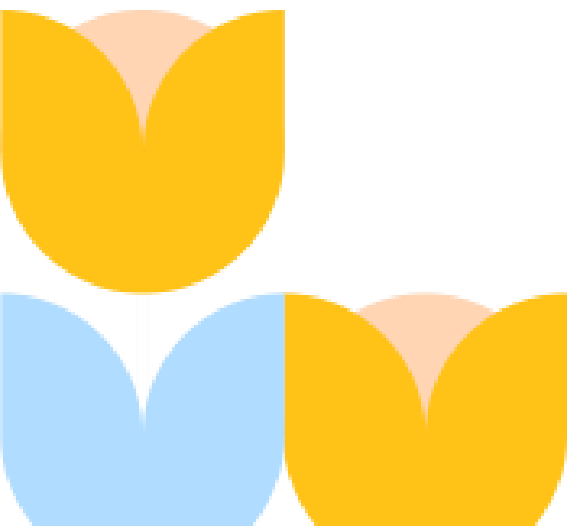
THE 2026 LANDSCAPE

Converged Infrastructure: The C-level mandate is "One Platform." There is no longer a strategic reason to manage two separate control planes (vCenter + K8s).

A Maturing Ecosystem: A wealth of new Kubernetes features and native tools now make it easier than ever to manage traditional workloads alongside containers.

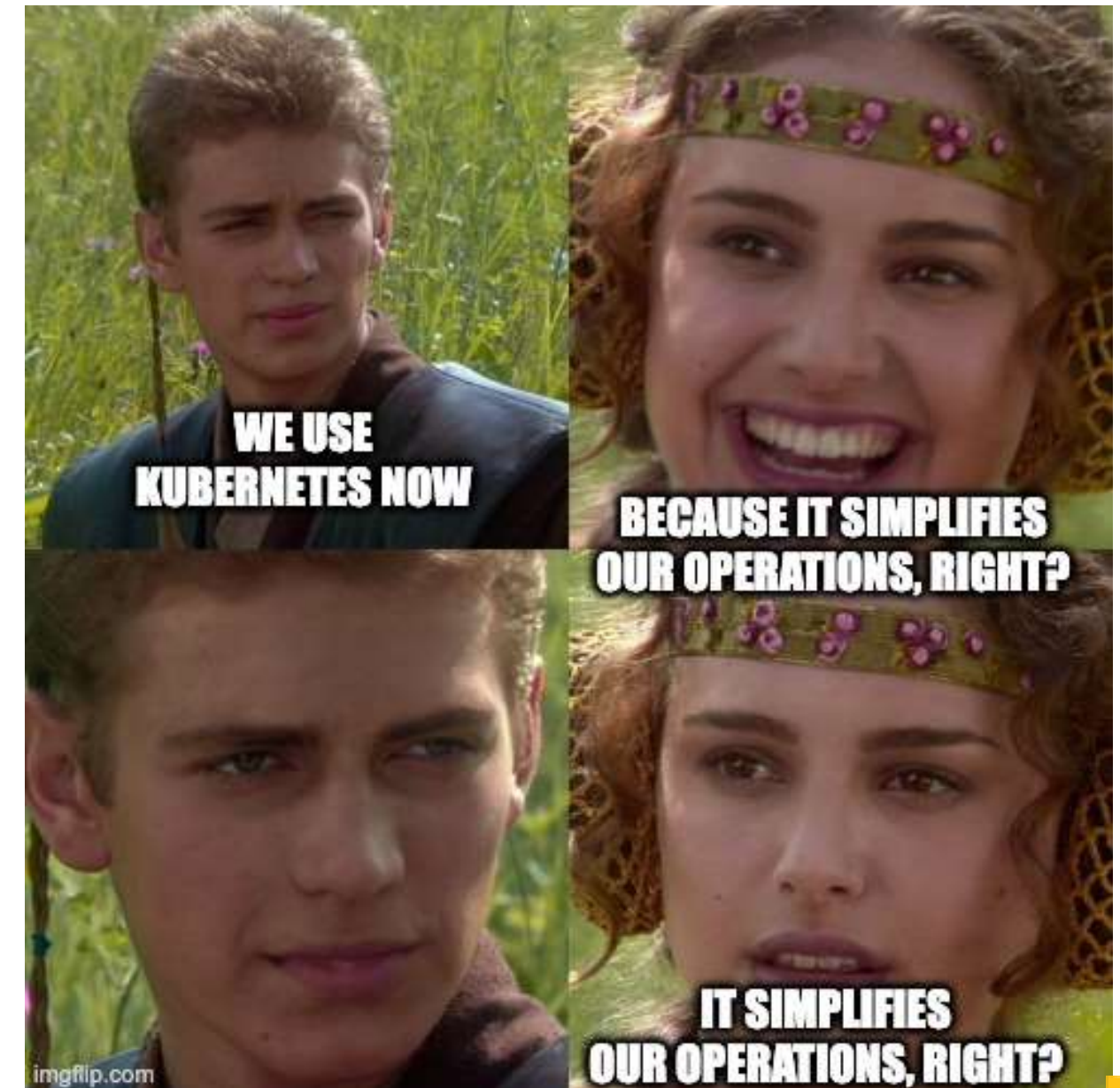
Adoption Surge: 28% of production clusters now host persistent VMs, up from just 8% in 2023.

The Ultimate Goal: Treat VMs as "Legacy Pods" within a single, unified orchestration layer.



VIRTUALIZATION IS A FEATURE. NOT A PLACE.

KubeVirt decouples Virtualization from the Infrastructure



credits: David Chan

ONE CONTROL PLANE TO RULE THEM ALL

THE PROMISE

Imagine deploying your legacy Windows SQL Server 2019 using the exact same pipeline (ArgoCD/Flux) that deploys your Golang microservices.

Unified RBAC

Unified Monitoring

Unified Storage (CSI)



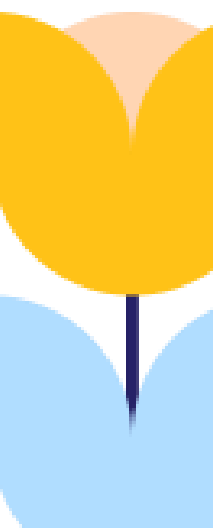
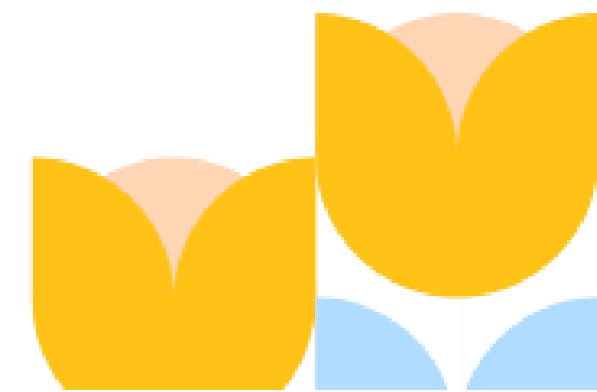
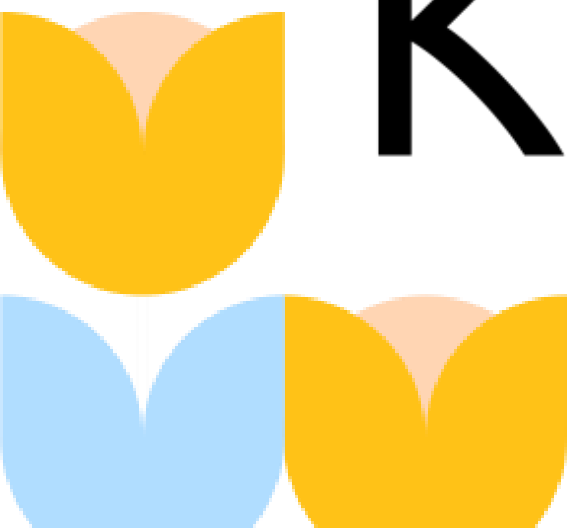


KUBEVIRT

KubeVirt extends Kubernetes by adding additional virtualization resource types through Custom Resource Definitions (**CRDs**).

This allows VM workloads to be treated as first-class citizens, sharing the same networking, storage, and monitoring pipelines as your containerized microservices.

KubeVirt



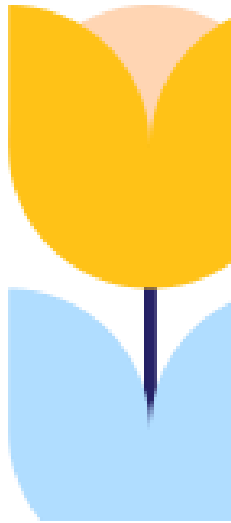
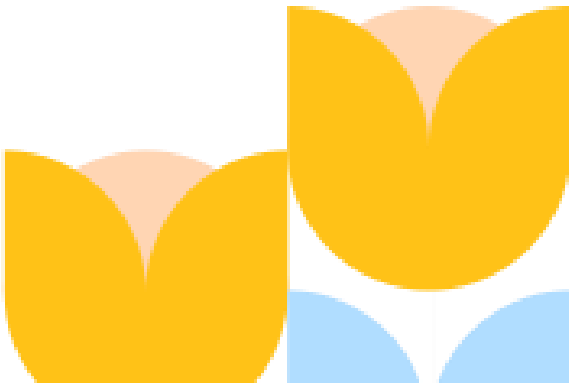
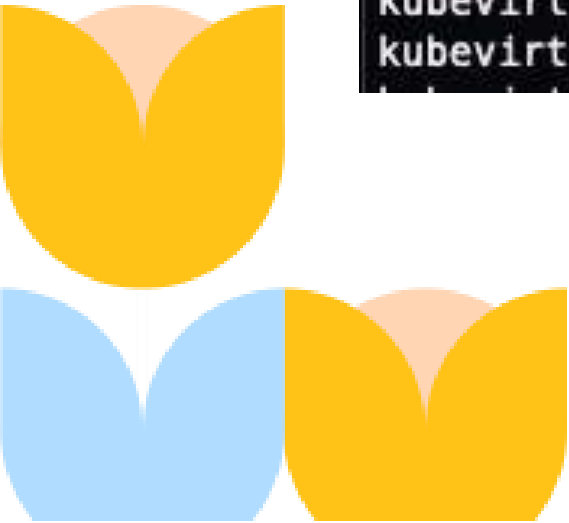
KUBEVIRT ARCHITECTURE

Virt-API: The entry point. Serves as the K8s API extension, validating and mutating VM objects.

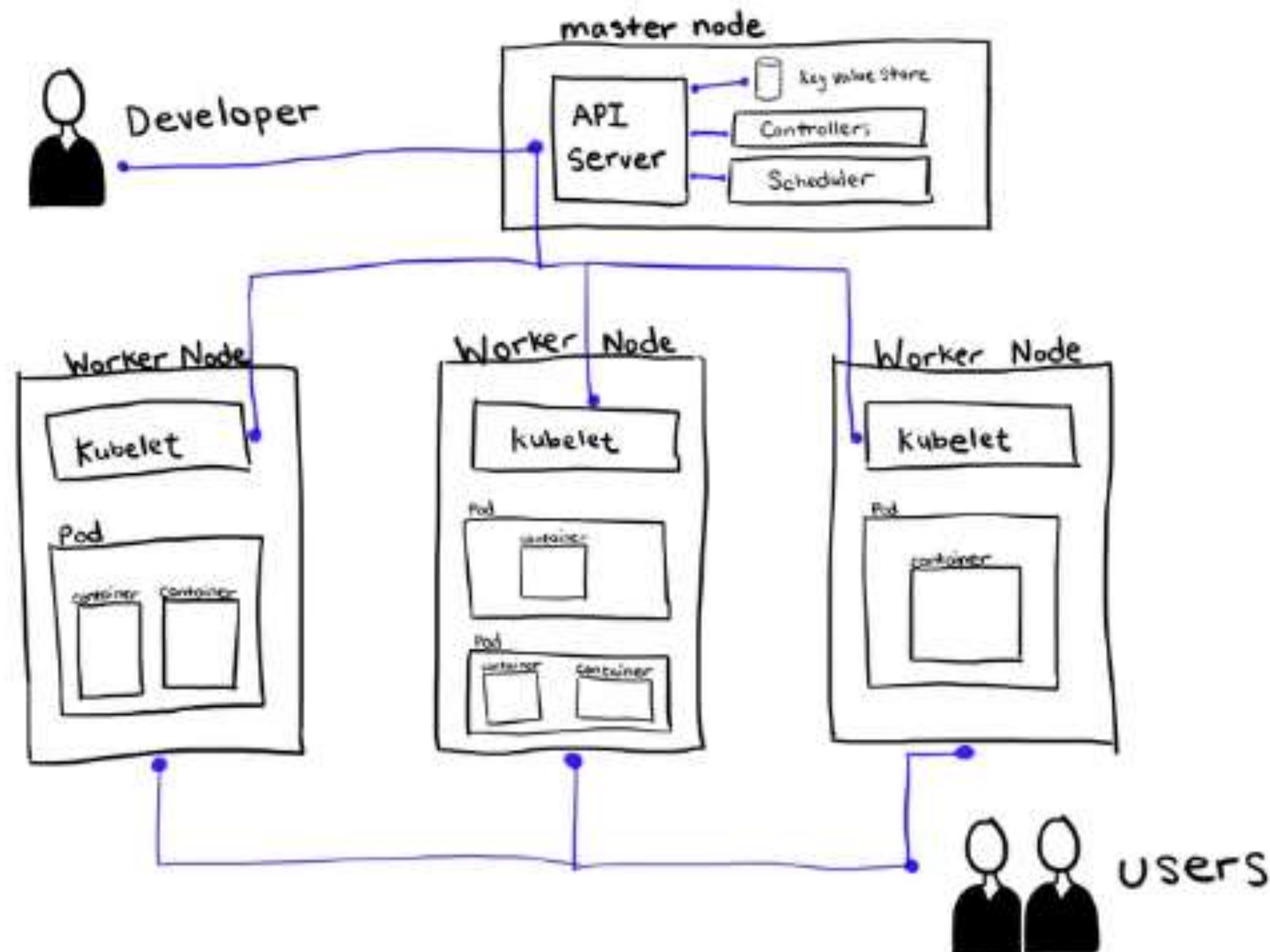
Virt-Controller: The "Brain". Watches for VM objects and manages the lifecycle of associated Pods.

Virt-Handler: The "Brawn". A DaemonSet running on every node. It communicates with Libvirt/QEMU and configures the host environment.

kubevirt	virt-api-564b56798c-57w9w	1/1	Running	1 (22d ago)	22d
kubevirt	virt-api-564b56798c-6bw4z	1/1	Running	1 (22d ago)	22d
kubevirt	virt-controller-8469995b94-9wxvv	1/1	Running	3 (25h ago)	22d
kubevirt	virt-controller-8469995b94-wmh6r	1/1	Running	4 (27h ago)	22d
kubevirt	virt-handler-2gjsq	1/1	Running	0	22d
kubevirt	virt-handler-8k9vd	1/1	Running	0	22d
kubevirt	virt-handler-dbmcp	1/1	Running	0	22d
kubevirt	virt-handler-fl284	1/1	Running	0	22d
kubevirt	virt-handler-hv8hw	1/1	Running	0	22d
kubevirt	virt-handler-jhfsz	1/1	Running	0	22d
kubevirt	virt-handler-m9mnc	1/1	Running	0	22d
kubevirt	virt-handler-pgc6q	1/1	Running	0	22d



THE VIRT-LAUNCHER POD



The Secret Sauce

KubeVirt runs VMs inside Pods. This pod is called **virt-launcher**.

- Contains a lightweight Libvirt daemon and QEMU process.
- The VM process is a child of the container process.
- If the Pod dies, the VM dies. If the Pod is scheduled, the VM is scheduled.
- Leverages standard CNI for networking and CSI for storage.

THE BASIC INFO: IT'S JUST A POD



VIRT-CONTROLLER

The "Brains." It watches for VirtualMachine CRDs and creates the corresponding Pods on the cluster.



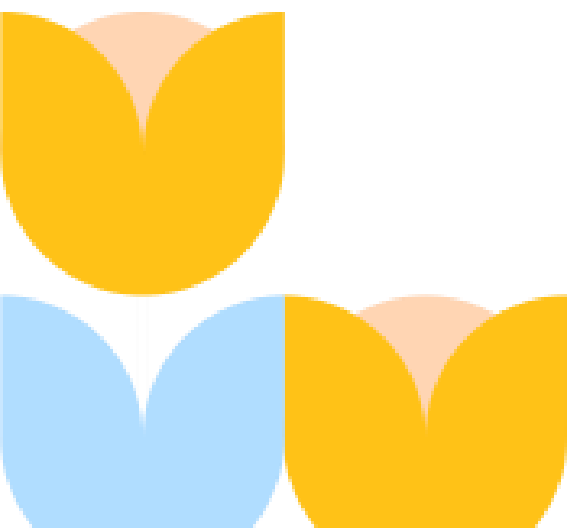
VIRT-HANDLER

The "Brawn." A DaemonSet on every node that configures the host environment (networking/storage) for the VM.



VIRT-LAUNCHER

A specialized Pod that wraps the Libvirt and QEMU processes. This is where your VM actually lives.



THE VIRTUALMACHINE MANIFEST

A declarative definition of the VM state, hardware, and boot configuration.

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata: { name: my-vm }
spec:
  running: true                # 1. State
  template:
    spec:
      domain:                  # 2. Hardware
        cpu: { cores: 2 }
        resources:
          requests: { memory: 4Gi }
        devices:
          disks:
            - name: rootdisk
              disk: { bus: virtio }
              bootOrder: 1      # 3. Boot
      volumes:                 # 4. Backend
        - name: rootdisk
          persistentVolumeClaim:
            claimName: my-vm-pvc
```

1

State Management

Declarative Power: Set running: true to boot. If the VM crashes, K8s restarts it. Set to false to shut down while keeping the config.

2

Domain (Hardware)

Defines the "Shape" of the VM. CPU topology (sockets/cores/threads), RAM requests/limits, and specific CPU features (e.g., host-passthrough).

3

Boot Configuration

Controls the BIOS/UEFI sequence. Here we explicitly set the rootdisk as the primary boot device (Order: 1).

4

Volumes (Backend)

The Data Source: This example maps the virtual disk to a PersistentVolumeClaim (PVC). You can also use containerDisk (for ephemeral images), dataVolume (auto-provisioning), or configMap (cloud-init).

API PRIMITIVES: VM VS VMI

VirtualMachine (VM)

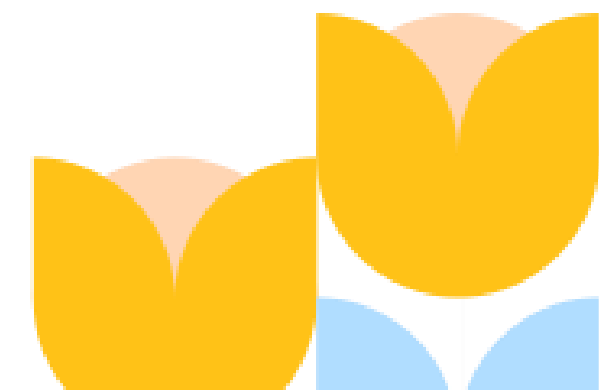
The Definition. Think of this like a Deployment or StatefulSet.

- Defines desired state (Running/Stopped).
- Manages restarts and updates.
- Generally, users interact with this object.

VirtualMachineInstance (VMI)

The Running Instance. Think of this like a Pod.

- Represents a single execution of a VM.
- Ephemeral. If you delete a VMI, it's gone.
- Managed by the VM controller.

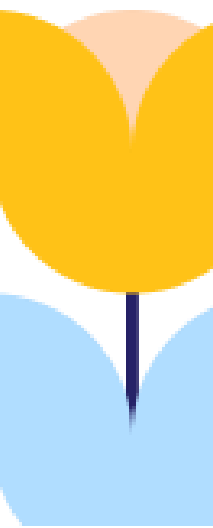
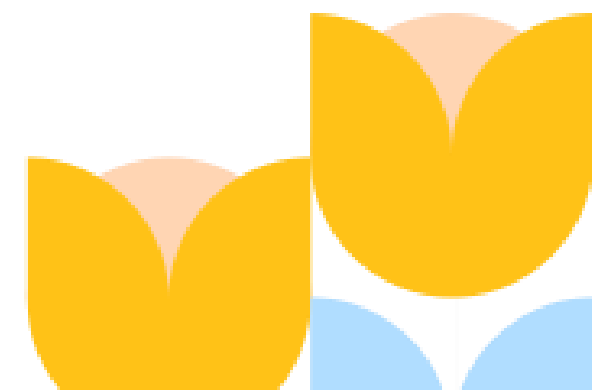
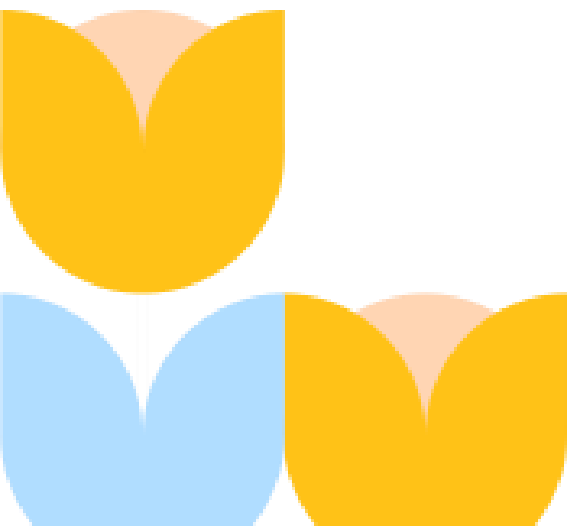


DEVICE PASSTHROUGH

Beyond Software Emulation

For high-performance workloads, KubeVirt supports hardware passthrough.

- GPU Passthrough: Assign NVIDIA/AMD cards directly to the guest for AI/ML or VDI.
- SR-IOV: High-performance networking that bypasses the kernel bridge.
- Host Devices: Pass specific USB or PCI devices (like dongles or accelerators) to the VM.



STORAGE & DATA INTEGRATION

Persistent Volume Claims (PVC)

KubeVirt VMs use standard PVCs for disks.
This means you can use any CSI driver (Ceph, EBS, Longhorn).

Containerized Data Importer (CDI)

A critical add-on. CDI provides a declarative way to import VM images (QCOW2, ISO) from HTTP endpoints or S3 buckets directly into PVCs.

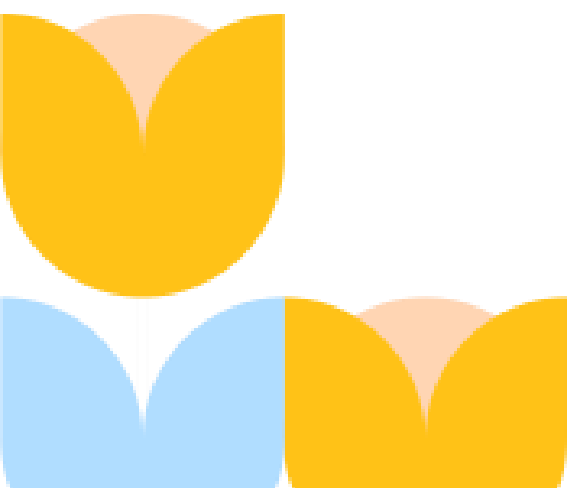
NETWORKING DEEP DIVE

Default: Masquerade (NAT)

The VM sits behind the Pod's eth0.
Traffic is NAT'd. The VM uses the Pod IP for egress.

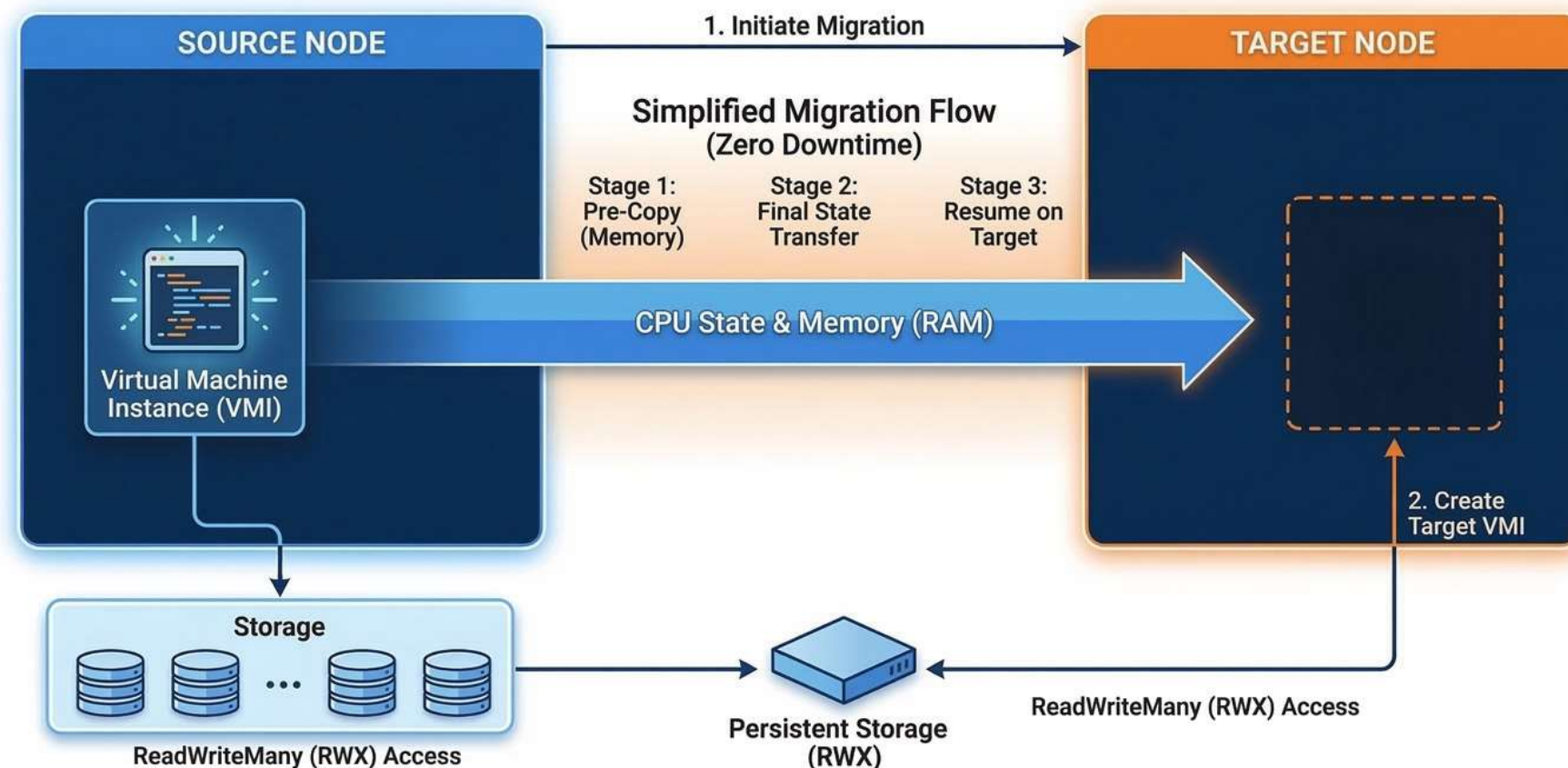
Secondary Interfaces (Multus)

Attach the VM directly to physical networks (VLANs) or secondary bridges. Essential for legacy applications requiring L2 adjacency.



LIVE MIGRATION MECHANICS

KubeVirt Live Migration Simplified Workflow



Moving without Downtime

KubeVirt leverages the native K8s API to facilitate live migration.

- **Target Pod Creation:** A new Launcher Pod is created on the destination node.
- **Memory Copy:** QEMU streams memory pages to the target.
- **Cutover:** CPU is paused briefly, final state transferred, and execution resumes on the new node.
- **Network:** Pod IP changes (unless using Multus), but Services handle routing updates.

THE CLOUD NATIVE WORLD INTEGRATION



ACCESS & OPERATIONS



VIRTCTL

The CLI companion. Use it for power management (`start`, `stop`) and console access.



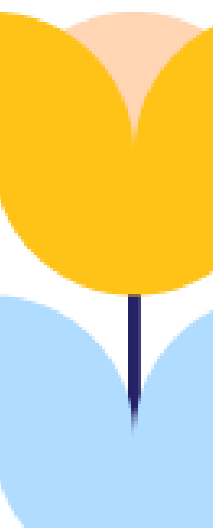
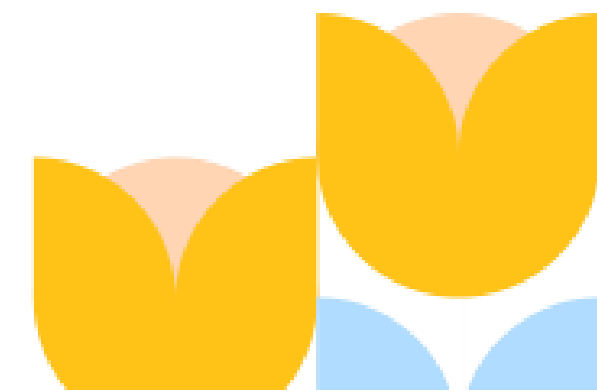
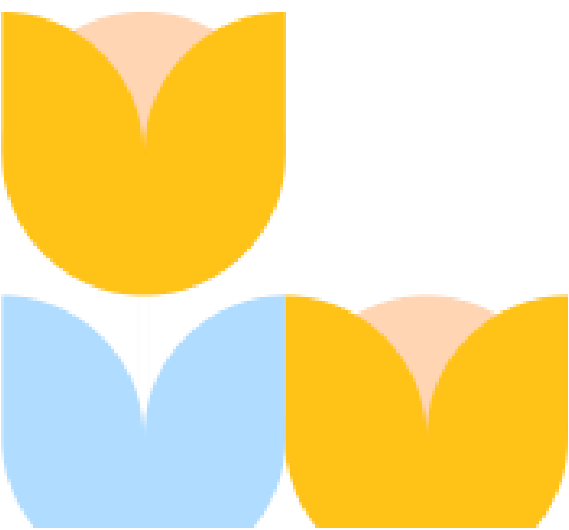
VNC & Console

Access the graphical or serial console of the VM directly through the Kubernetes API server.



SSH Service

Expose standard SSH via a K8s `Service` (NodePort or LoadBalancer) for traditional access.

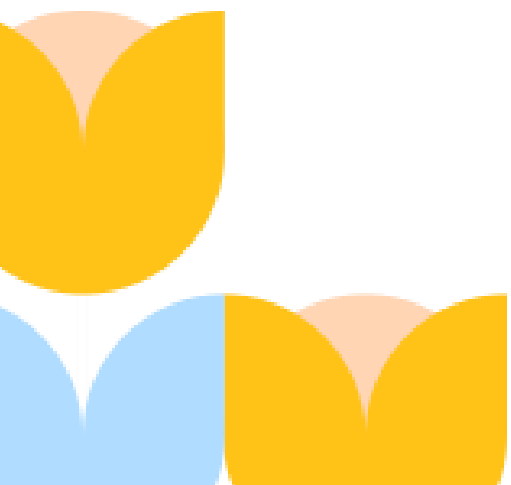
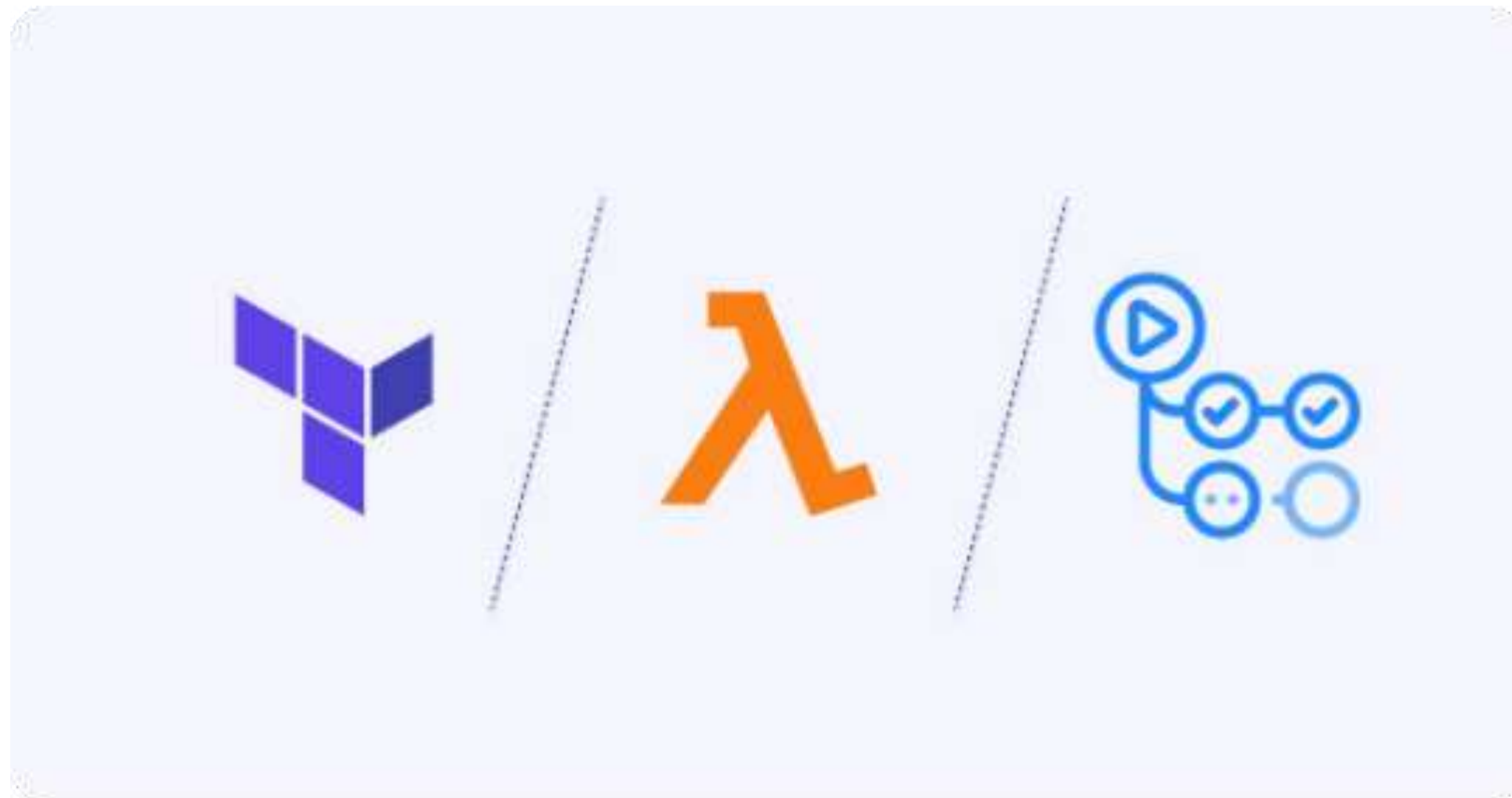


ECOSYSTEM INTEGRATION

GitOps Ready

Since VMs are just YAML manifests, they integrate perfectly with the cloud-native ecosystem.

- **ArgoCD / Flux:** Manage VM state via Git.
- **Tekton:** Build VM images and run VM-based tests in CI pipelines.
- **Cluster API:** Use KubeVirt as an infrastructure provider (CAPK) to run K8s clusters inside KubeVirt.



UNDER KUBERNETES & KUBEVIRT OS-LEVEL HARDENING

CHOOSE A CLOUD-NATIVE, OPEN-SOURCE SECURE OS AS YOUR FOUNDATION.



TALOS OS

Immutable, API-only OS with no shell or SSH.
Purpose-built for Kubernetes — minimal attack
surface by design.



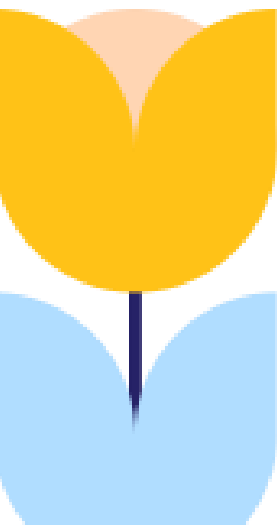
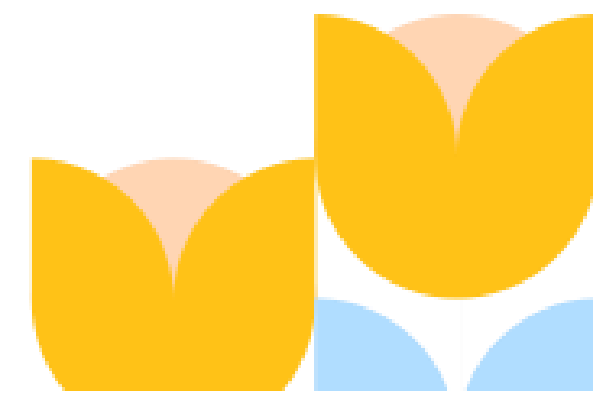
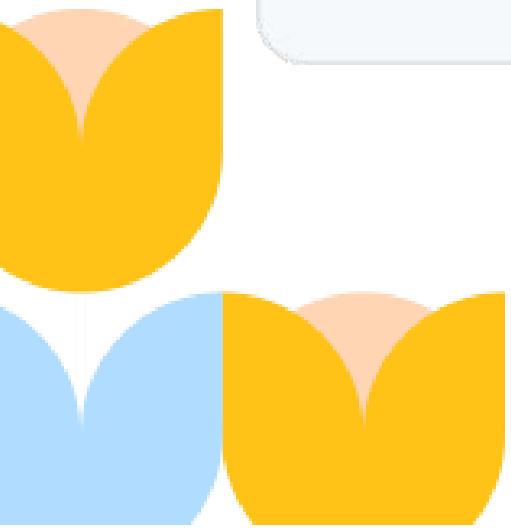
KAIROS

Build signed, hardened OS images for the edge.
Atomic A/B updates and air-gap ready
deployments.



OPEN SOURCE

Full transparency, community-driven security.
Declarative config, GitOps-friendly and fully
auditable.





THE CNI SHOWDOWN

Cilium vs. Kube-OVN

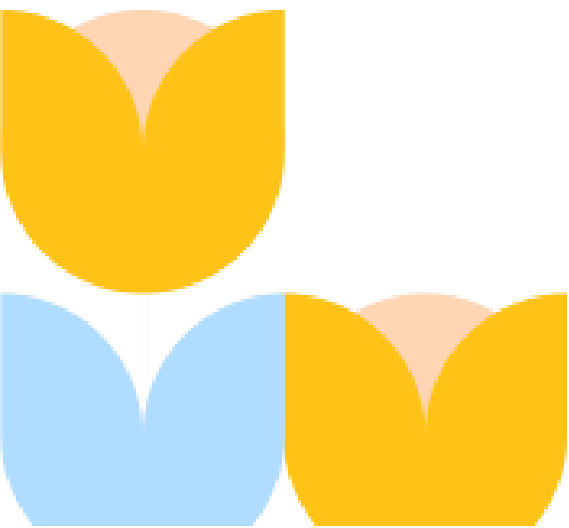
CONTENDER 1: CILIUM

THE SPEEDSTER

- > **Tech:** eBPF (Kernel Bypass).
- > **Focus:** Cloud Native, massive scale, deep observability (Hubble).
- > **Pros:** Insane throughput, low latency for Pod-to-Pod traffic.
- > **The Vibe:** "The Future of Networking."



cilium



CILIUM'S "LEGACY" PROBLEM

SQUARE PEG, ROUND HOLE

To give a VM a static IP on a VLAN with Cilium, you typically need to chain Multus + Macvlan.

This creates a split brain: Cilium manages the default network (eth0), but the VM's actual traffic goes out a side door that Cilium's policies often can't see or control.

THE COMPLEXITY

High

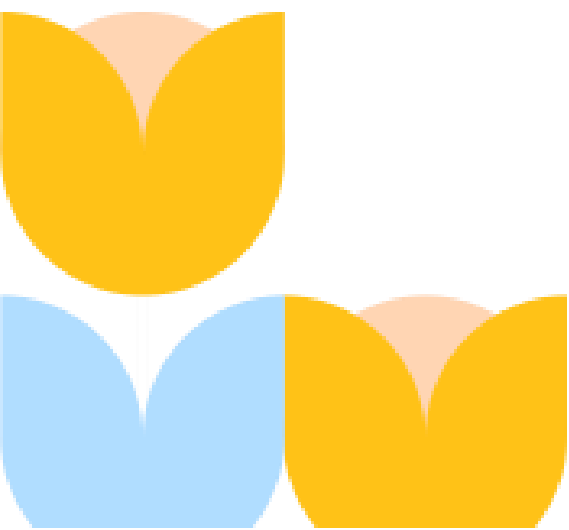
Requires managing IPAM separately from the CNI. Debugging connectivity issues across Multus interfaces is painful.

CONTENDER 2: KUBE-OVN



THE VETERAN

- > **Tech:** Open vSwitch (OVS) + OVN.
- > **Focus:** Network Virtualization (SDN).
- > **Heritage:** The engine that powered OpenStack.
- > **The Vibe:** "Bring the Physical Switch to Kubernetes."



WHY KUBE-OVN WINS FOR VMS

It speaks the language of your Network Engineer.



SUBNETS

Define ``192.168.10.0/24`` and assign it to a Namespace. Just like a VPC.



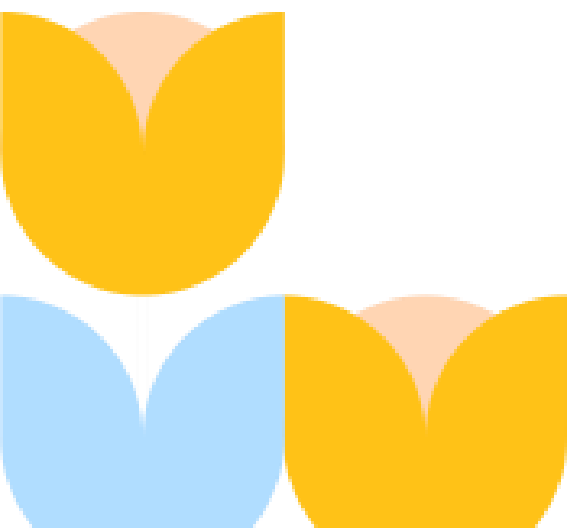
STATIC IPS

Assign a specific IP directly in the VM YAML. No DHCP hacks required.



VLANS

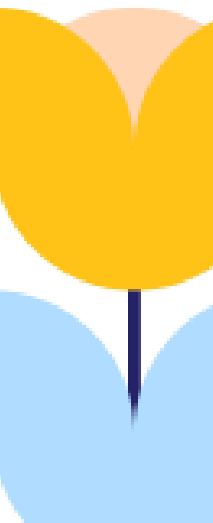
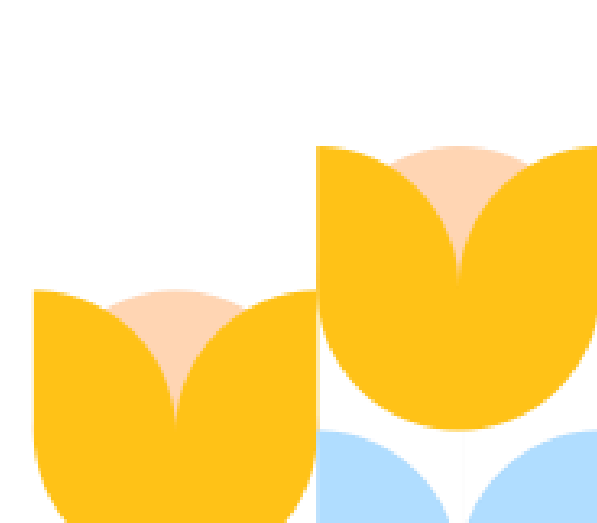
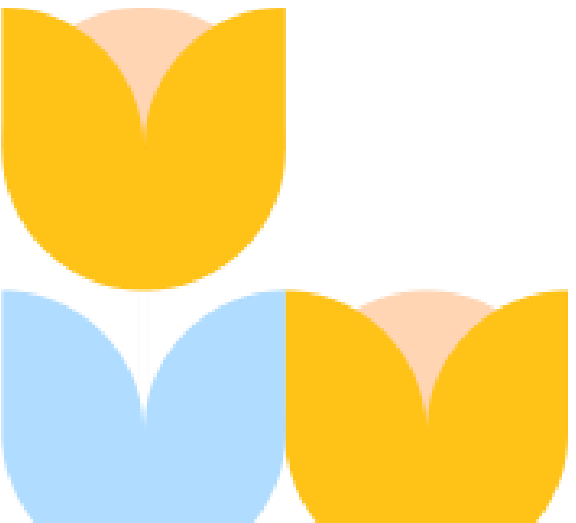
Native ``ProviderNetwork`` support bridges VMs directly to physical switch VLANs.



PERFORMANCE: SOFTWARE VS HARDWARE



While Cilium wins in pure software mode, Kube-OVN supports Hardware Offload (NVIDIA/Mellanox) to offload the SDN to the NIC, achieving zero-CPU overhead.



THE LIVE MIGRATION TEST

THE SCENARIO

Ping a VM continuously while migrating it to another node.

- > **Kube-OVN:** Pass. OVN maintains the logical switch port. The IP floats seamlessly.
- > **Cilium:** Fail / Complex. Often requires BGP peering or Beta L2 announcement features to prevent dropped packets during the switchover.



OPERATIONAL COMPLEXITY

DEVOPS ENGINEER

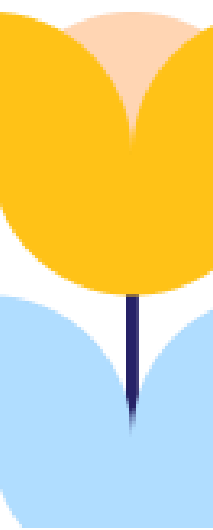
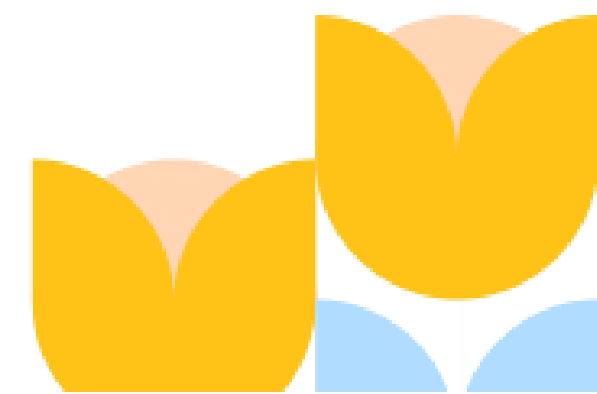
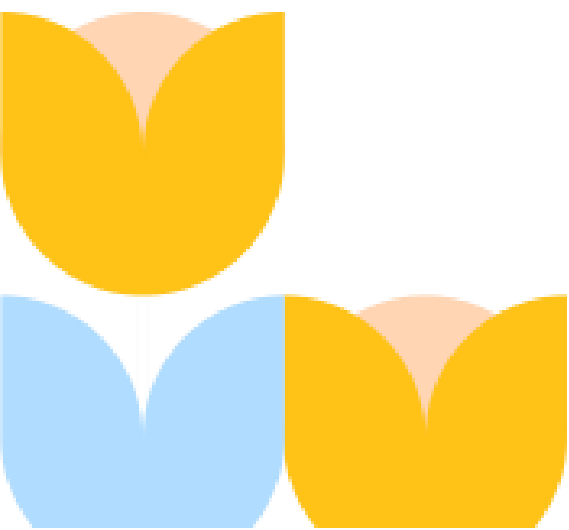
Prefers Cilium

Uses Hubble UI, thinks in Services and NetworkPolicies. Hates VLANs.

NETOPS ENGINEER

Prefers Kube-OVN

Understands Subnets, Gateways, and Routes.
Wants to debug with `tcpdump` and OVS flows.



THE CNI VERDICT

USE CILIUM IF...

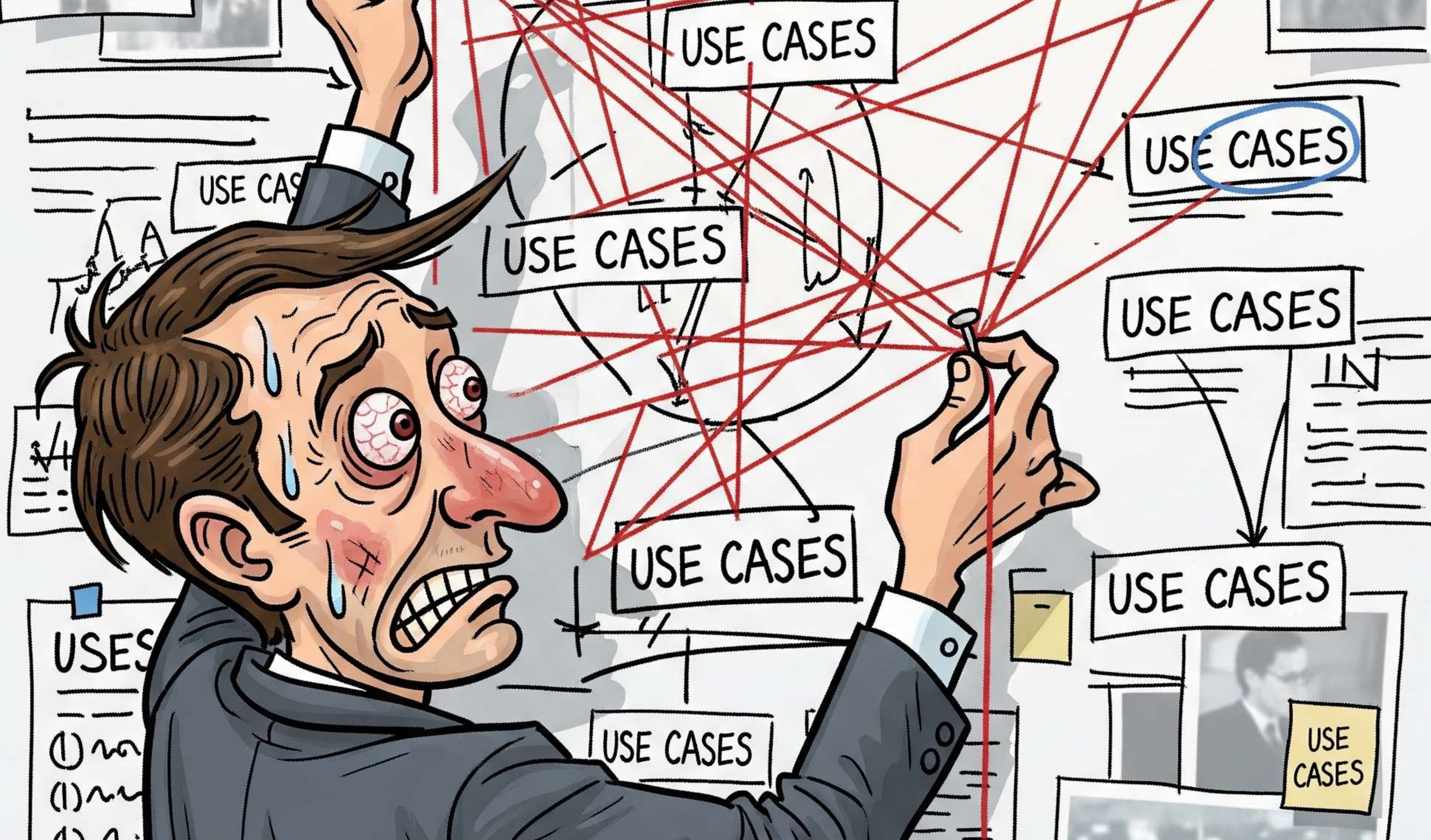
Your workload is 90% Containers, 10% VMs. You don't need legacy Layer 2 features.

USE KUBE-OVN IF...

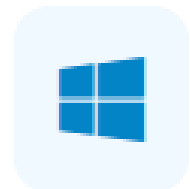
You are replacing VMware. You need VLANs, Static IPs, and a topology that mimics physical networking.

Recommendation:

For a dedicated "Virtualization Replacement" cluster, Kube-OVN is the path of least resistance.

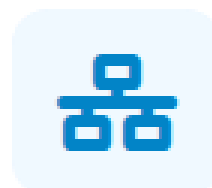


KEY USE CASES



LEGACY WINDOWS

Retain old Windows Server apps that cannot be containerized, but manage them with K8s.



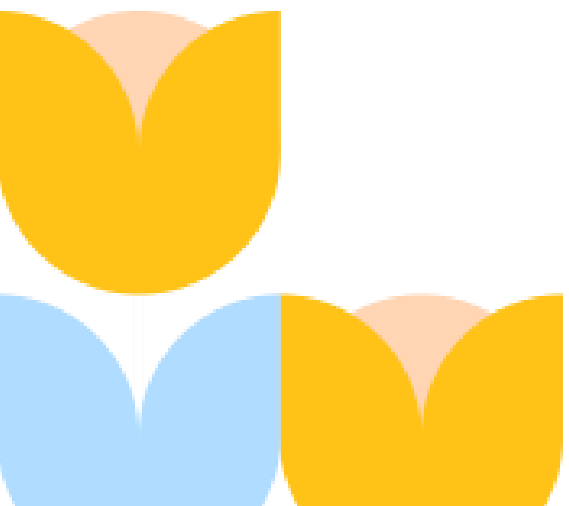
NFV / TELCO

Network Function
Virtualization requiring custom kernels and complex networking.



CONTROL PLANES

Run ephemeral Kubernetes clusters (Nodes = VMs) for development or testing.



HOW DID I DISCOVER?

STACKIT is the European sovereign cloud of the Schwarz Group.

STACKIT is adopting and contributing to KubeVirt, e.g. for a Distributed Cloud offering.

STACKIT offers scalable, innovative cloud and colocation services in accordance with European data protection standards.



schwarz digits



The sovereign cloud for Europe

IS KUBEVIRT THE BEST CHOICE?

KubeVirt decouples virtualization from the infrastructure.



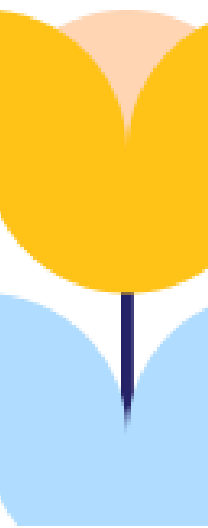
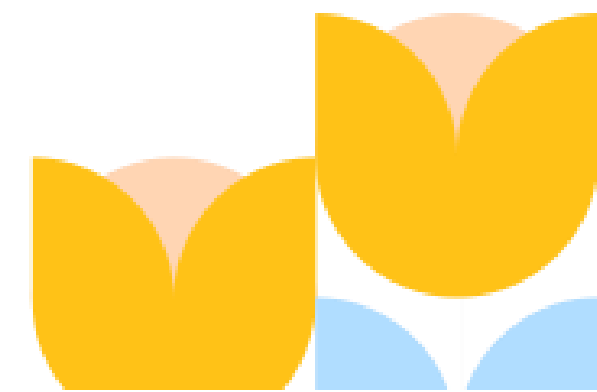
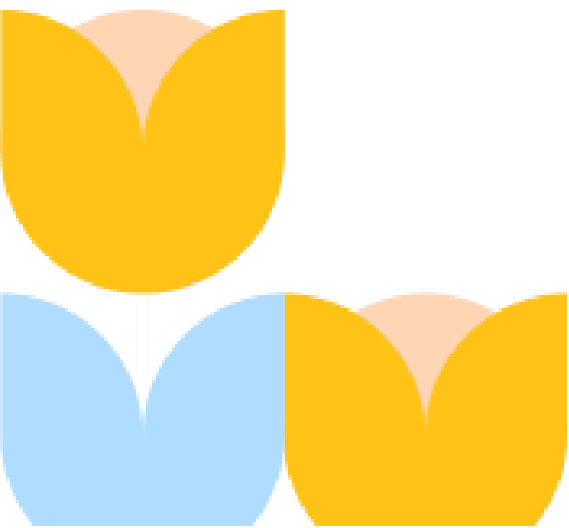
EXPECTATION VS. LIMITATIONS

LIMITATION #1 FRAGILE LIVE MIGRATION

"LIVE" IS A STRONG WORD.

In VMware, vMotion works even under heavy load. In KubeVirt, a VM with a high "dirty memory rate" (like a busy Database) will often fail to converge and time out.

Furthermore, migrating VMs connected to Bridge interfaces requires complex network orchestration to prevent MAC address conflicts.



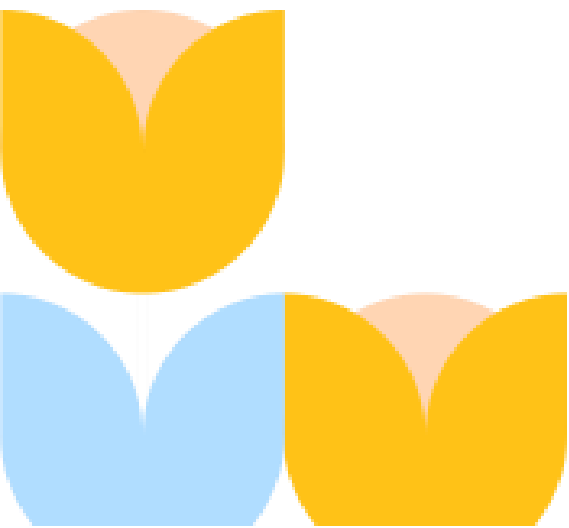
EXPECTATION VS. LIMITATIONS

LIMITATION #2: THE STORAGE LOTTERY

RWO VS RWX

To use Live Migration, you generally need ReadWriteMany (RWX) or shared Block storage. Many default cloud storage classes are ReadWriteOnce (RWO).

If you pick the wrong CSI driver, your VMs are effectively pinned to a single node forever. Provisioning time can also jump from seconds (Cloning) to minutes (Copying).



EXPECTATION VS. LIMITATIONS

LIMITATION #3: THE "DEATH & REBIRTH" CYCLE

40s

NODE NOTREADY



300s

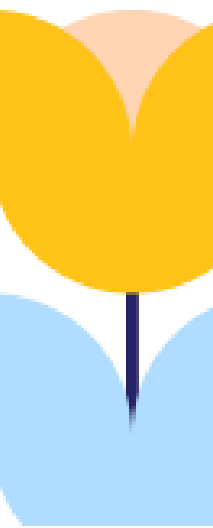
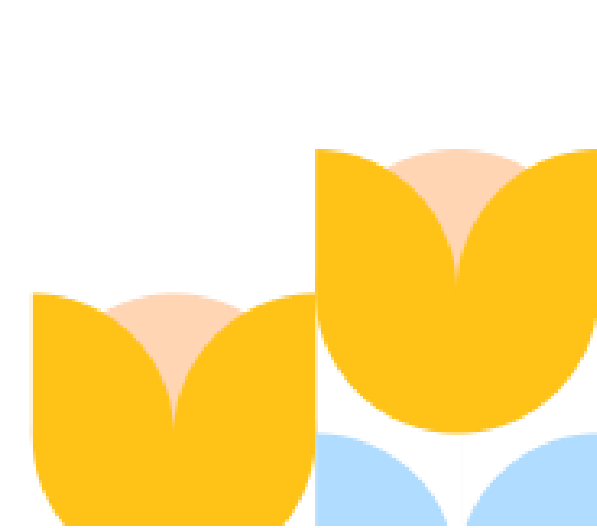
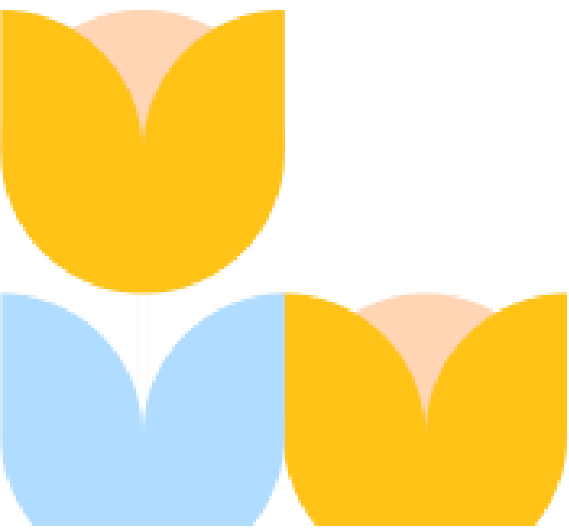
EVICTON TIMEOUT

The HA Gap: By default, Kubernetes waits 5 minutes before accepting a node is dead. You must tune `pod-eviction-timeout` aggressively or use external health checkers.

EXPECTATION VS. LIMITATIONS

LIMITATION #4: WINDOWS IS A SECOND-CLASS CITIZEN

- > **VirtIO Drivers:** Windows doesn't see KubeVirt disks/networks natively. You must inject drivers via container disks.
- > **Sysprep Hell:** No "Customization Wizard." You are writing XML into ConfigMaps to automate the OOBE experience.
- > **Performance:** Windows emulation overhead is significantly higher than Linux guests.



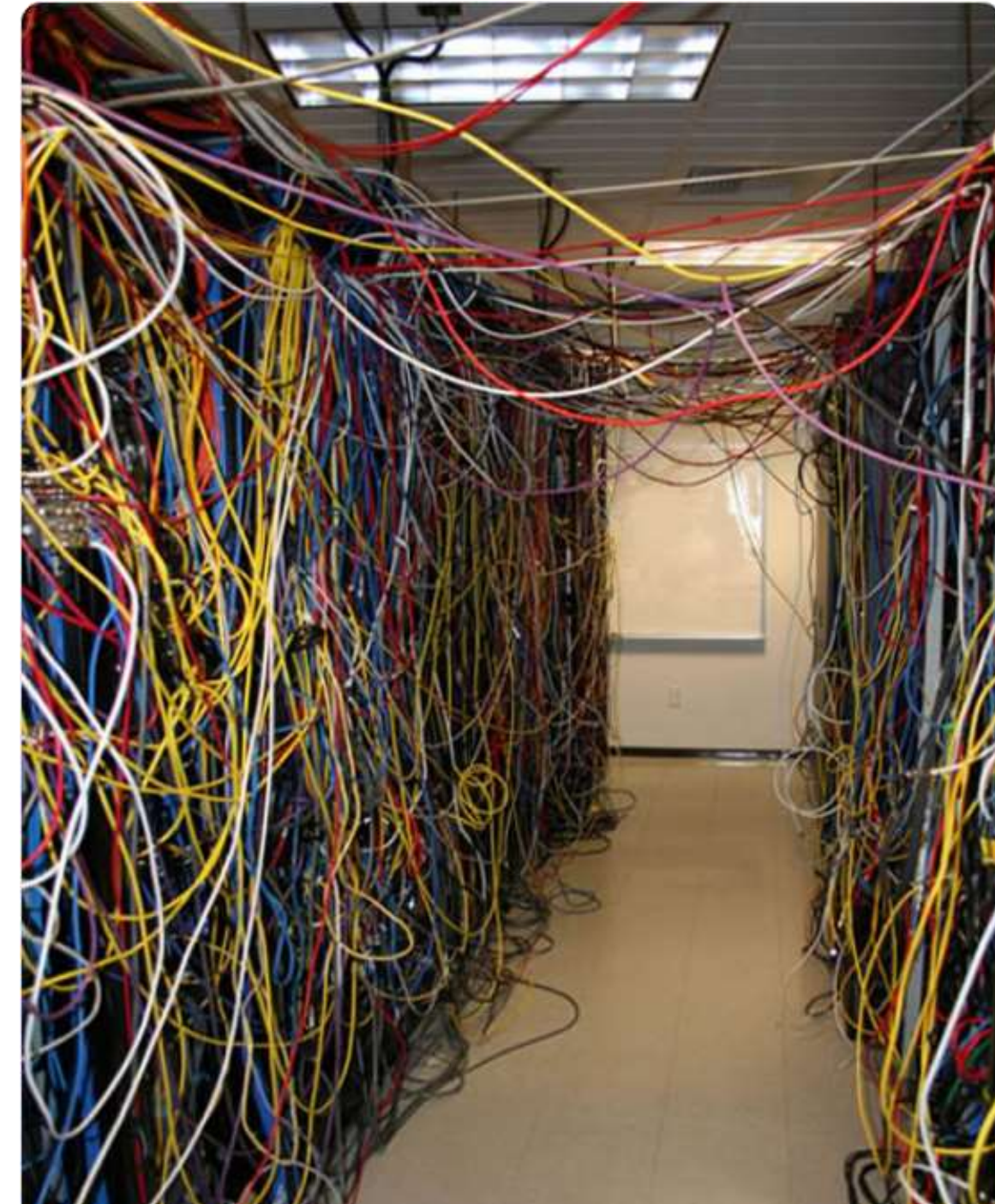
EXPECTATION VS. LIMITATIONS

LIMITATION #5: HARDWARE PASSTHROUGH

THE USB & GPU TRAP

Does your legacy app rely on a physical USB license dongle? Prepare for pain. USB passthrough is limited and buggy.

PCI/GPU Passthrough: While possible, it pins the VM to a specific node. You instantly lose Live Migration capabilities, turning your VM into a static pet.



EXPECTATION VS. LIMITATIONS

LIMITATION #6: ROLL YOUR OWN DR

NO "SITE RECOVERY MANAGER"

There is no magic "Failover" button in vanilla Kubernetes.

You must assemble your DR strategy from opensource blocks:

- > **VolSync:** For async data replication.
- > **Velero:** For backup and restore.
- > **GitOps:** To ensure config drift doesn't kill your recovery site.



EXPECTATION VS. LIMITATIONS

LIMITATION #7: WHERE IS MY DASHBOARD?



THE "ADMIN" GAP

Your admins are used to a rich GUI. Vanilla KubeVirt offers... `kubectl get vmi`.

The Fix: You essentially need a management wrapper to make this usable for Day 2 Ops:

- > OpenShift Virtualization
- > Rancher / Harvester
- > KubeVirt Manager

EXPECTATION VS. LIMITATIONS

LIMITATION #8 THE LAYER 2 GAP

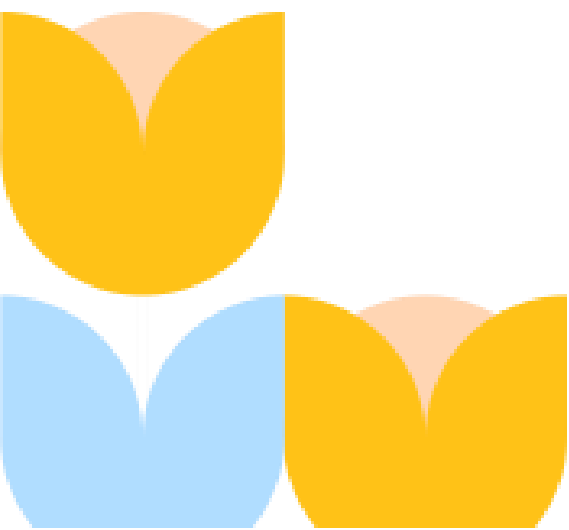
CONTAINERS CARE ABOUT IPS.

VMS CARE ABOUT MACS.

Most CNIs are built for Layer 3 overlay networking. They don't natively support the features legacy VMs require: Static MAC addresses, VLAN tags, and Multicast.

This leads to the "Layer Cake" of double encapsulation and performance penalties.

Sometimes I wonder,
why there are Layer 3 switches,
but no Layer 2 routers



EXPECTATION VS. LIMITATIONS

LIMITATION #9: SECURITY CONSTRUCTS



NETWORKPOLICIES

K8s NetworkPolicies apply to Pod IPs. If your VM uses a bridge to the physical network, these policies are bypassed.



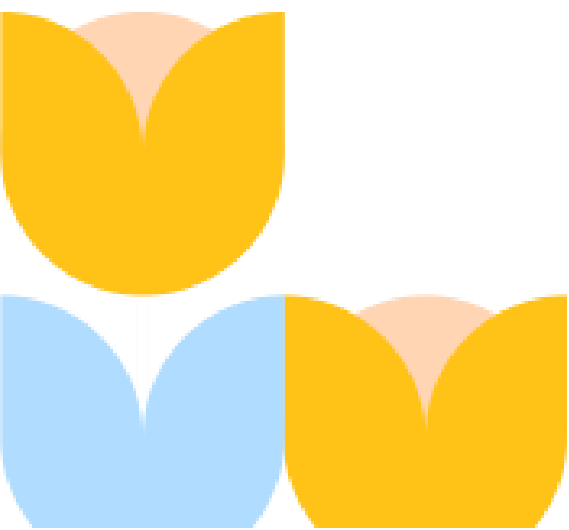
IDENTITY

VMs often lack the "Identity" (ServiceAccount) integration that makes mTLS and Service Mesh easy for containers.



THE GHOST

A bridged VM is effectively a ghost on the network. The K8s control plane knows very little about its traffic.



SUMMARY: IS IT PRODUCTION READY?



GREEN LIGHT

Linux Web Servers, Stateless Apps, Dev/Test Environments, CI/CD runners.



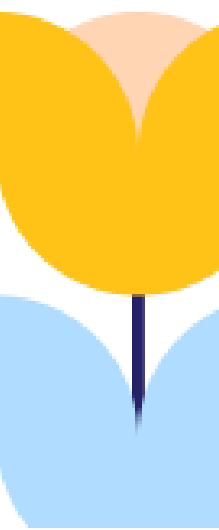
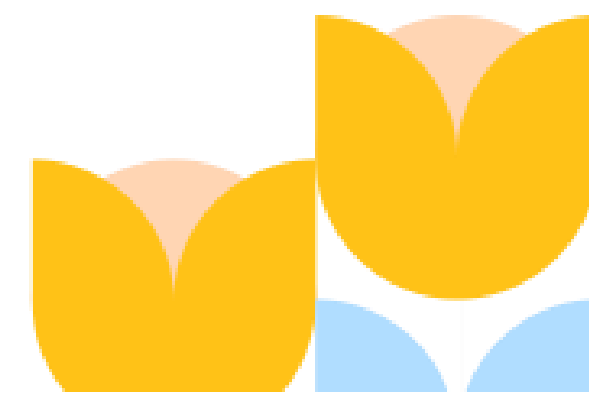
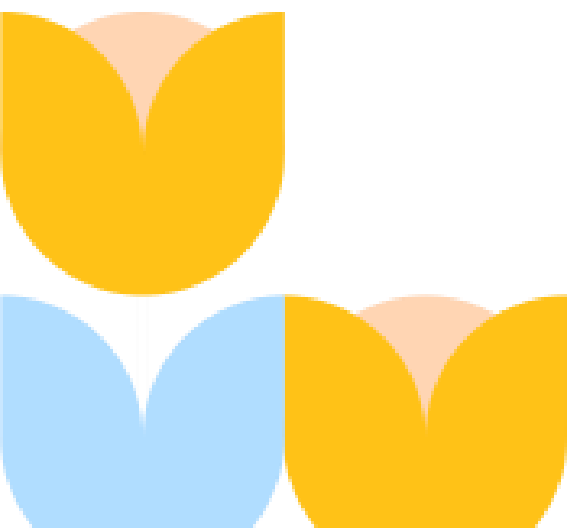
YELLOW LIGHT

Standard SQL Databases, Internal Monoliths. Requires tuning and correct Storage Class.



RED LIGHT

Oracle RAC, Multicast-heavy apps, USB-dependent legacy software.



THE TOOLBOX



KONVEYOR FORKLIFT

The "vMotion" from ESXi to KubeVirt. Handles conversion and data sync.



ROOK-CEPH

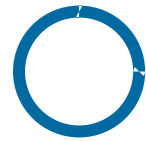
Open Source HCI storage. Essential for Block storage replication.



VELERO

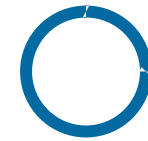
Backup and Restore. Don't go to production without CSI snapshot integration.

REALISTIC TIMELINE



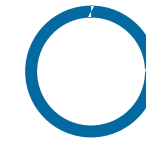
MONTH 1-6

Network & Storage Validation. Do not skip this.



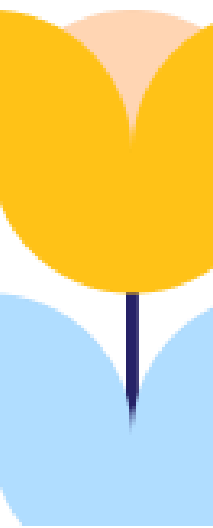
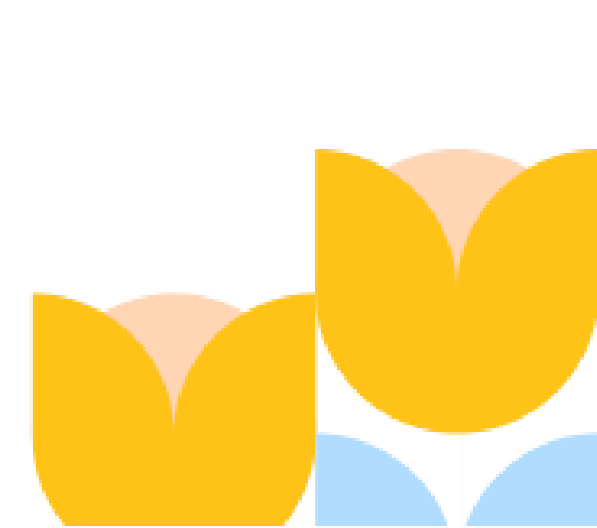
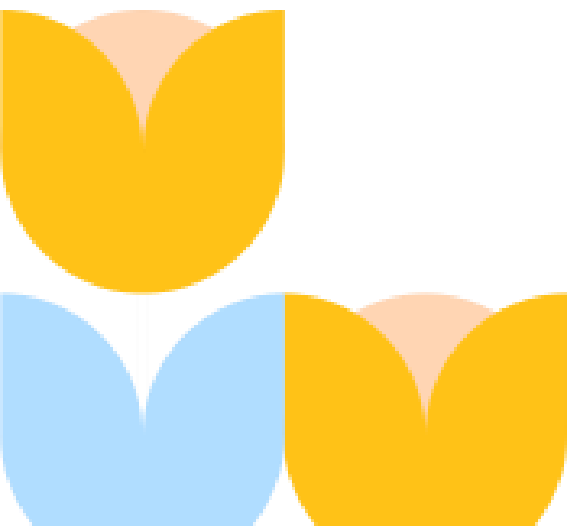
MONTH 6-12

"Pet" VM Migration. The hard manual work.



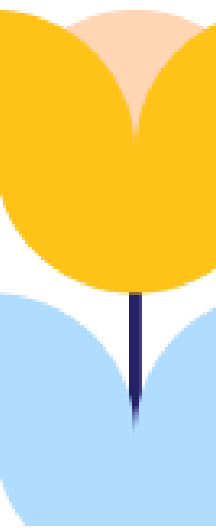
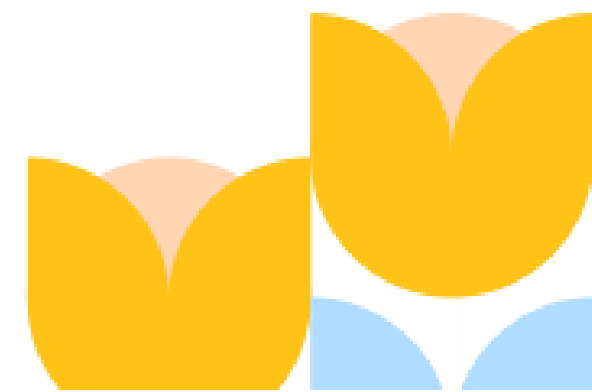
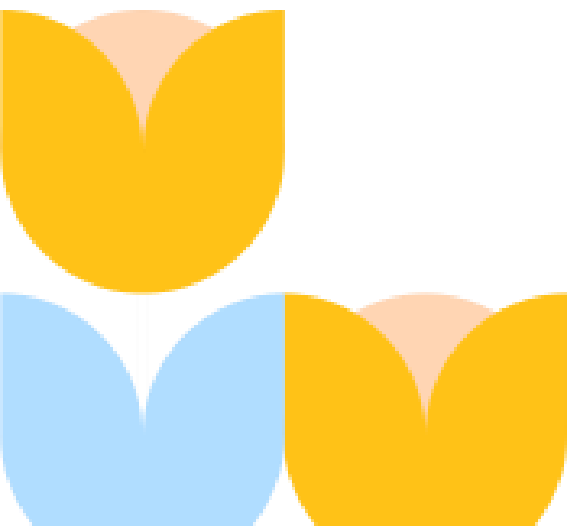
MONTH 12-18

Operational Tuning. Right-sizing and monitoring.



THE FUTURE OUTLOOK

IT'S NOT ABOUT RUNNING VMS. IT'S ABOUT RUNNING APPLICATIONS.





KubeCon



CloudNativeCon

Europe 2026

QUESTIONS?

Grab the Slides from Github



BIG THANKS FROM EVAN (ME)