Red Hat Summit

Connect

Crash Course in OpenShift Virtualization

Running VMs Natively on Kubernetes with KubeVirt

James Force
Principal Consultant
Red Hat

James Force

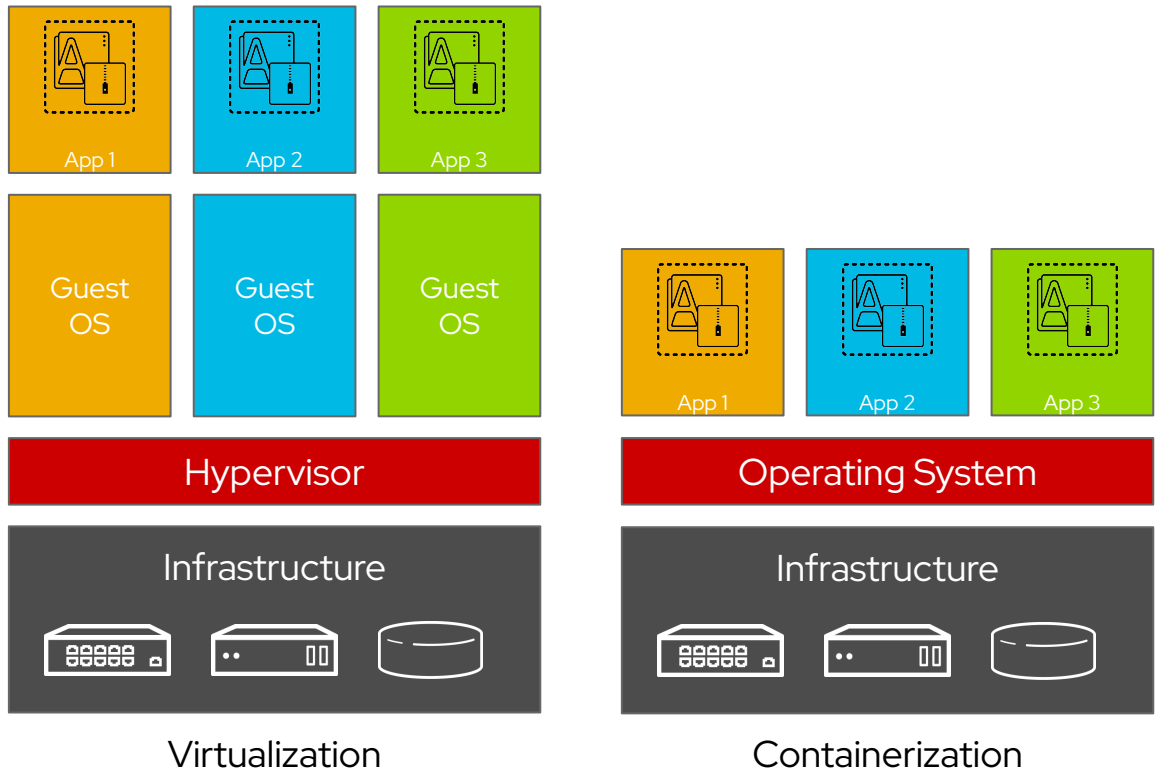
Principal Consultant
Red Hat

What & Why OpenShift Virtualization?

- Kubernetes has become the platform for running container workloads.
- The VM ecosystem has never been as unified:
  - Open-source: KVM, Proxmox, OpenStack
  - Proprietary: VMware, Nutanix
  - Public cloud: AWS EC2, Azure, GCP
  - No single solution dominates the industry.
- But think about it: Kubernetes is essentially a workload orchestrator.
- KubeVirt brings VMs into Kubernetes, preserving the principles of Kubernetes.

Containers are not virtual machines

- Containers are process isolation
- Kernel namespaces provide isolation and cgroups provide resource controls
- Contain only binaries, libraries, and tools which are needed by the application
- Ephemeral



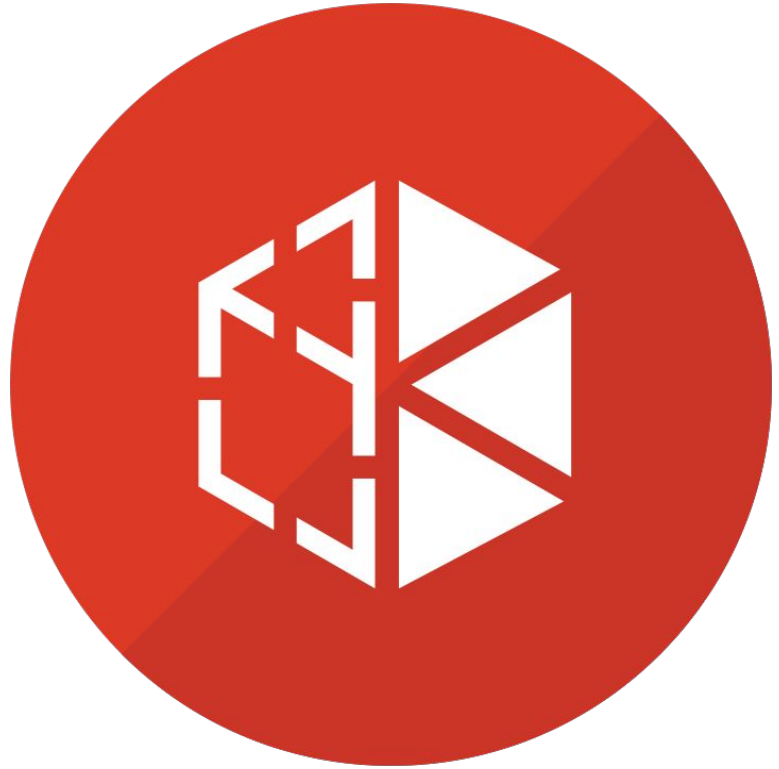
Red Hat

Virtual machines can be put into containers

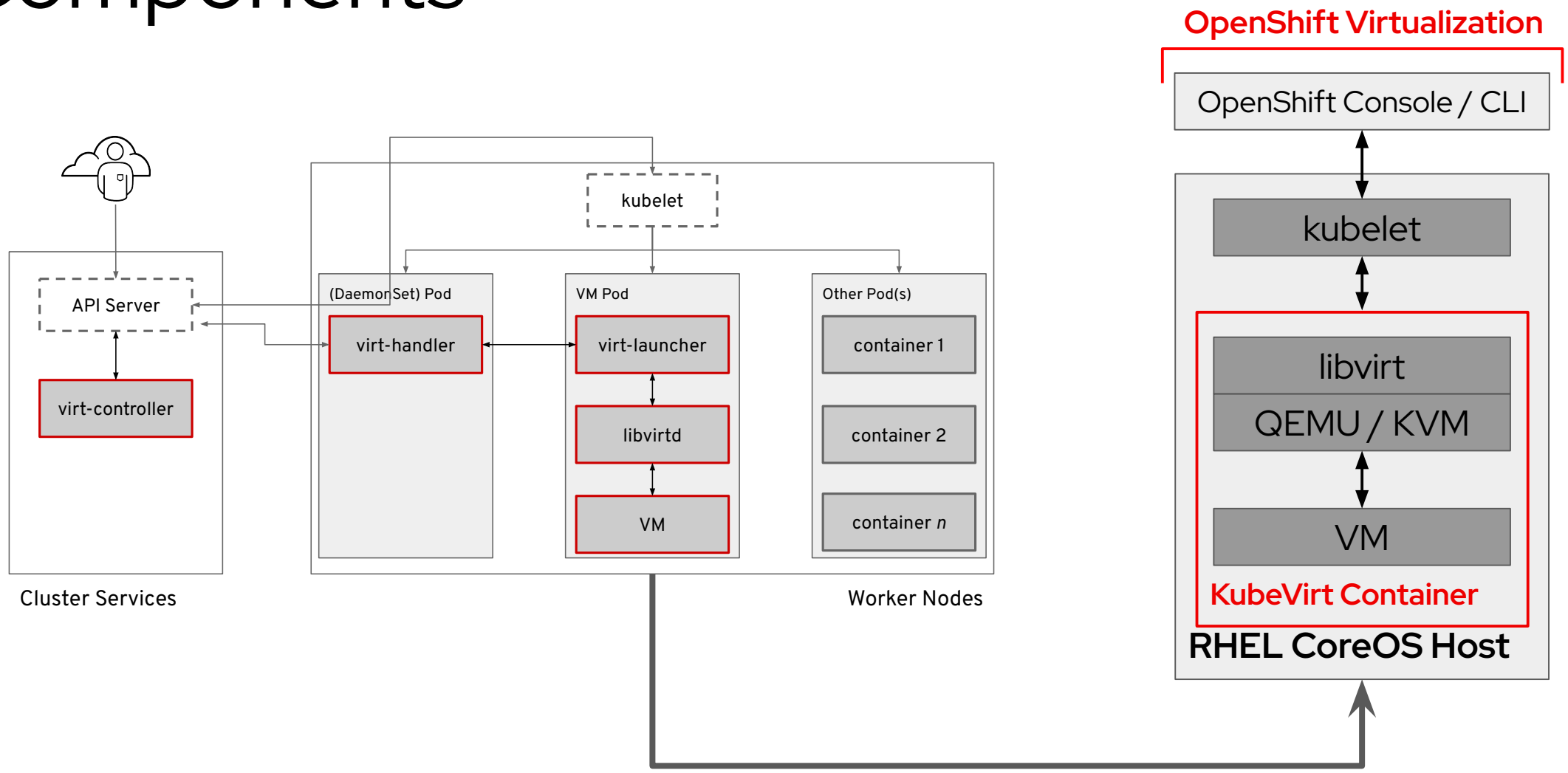
- A KVM virtual machine is a process
- Containers encapsulate processes
- Both have the same underlying resource needs:
  - Compute
  - Network
  - (sometimes) Storage

OpenShift Virtualization

- Virtual machines
  - Running in containers, managed as Pods
  - Using the KVM hypervisor
- Scheduled, deployed, and managed by Kubernetes
- Integrated with container orchestrator resources and services
  - Traditional Pod-like SDN connectivity and/or connectivity to external VLAN and other networks via multus
  - Persistent storage paradigm (PVC, PV, StorageClass)

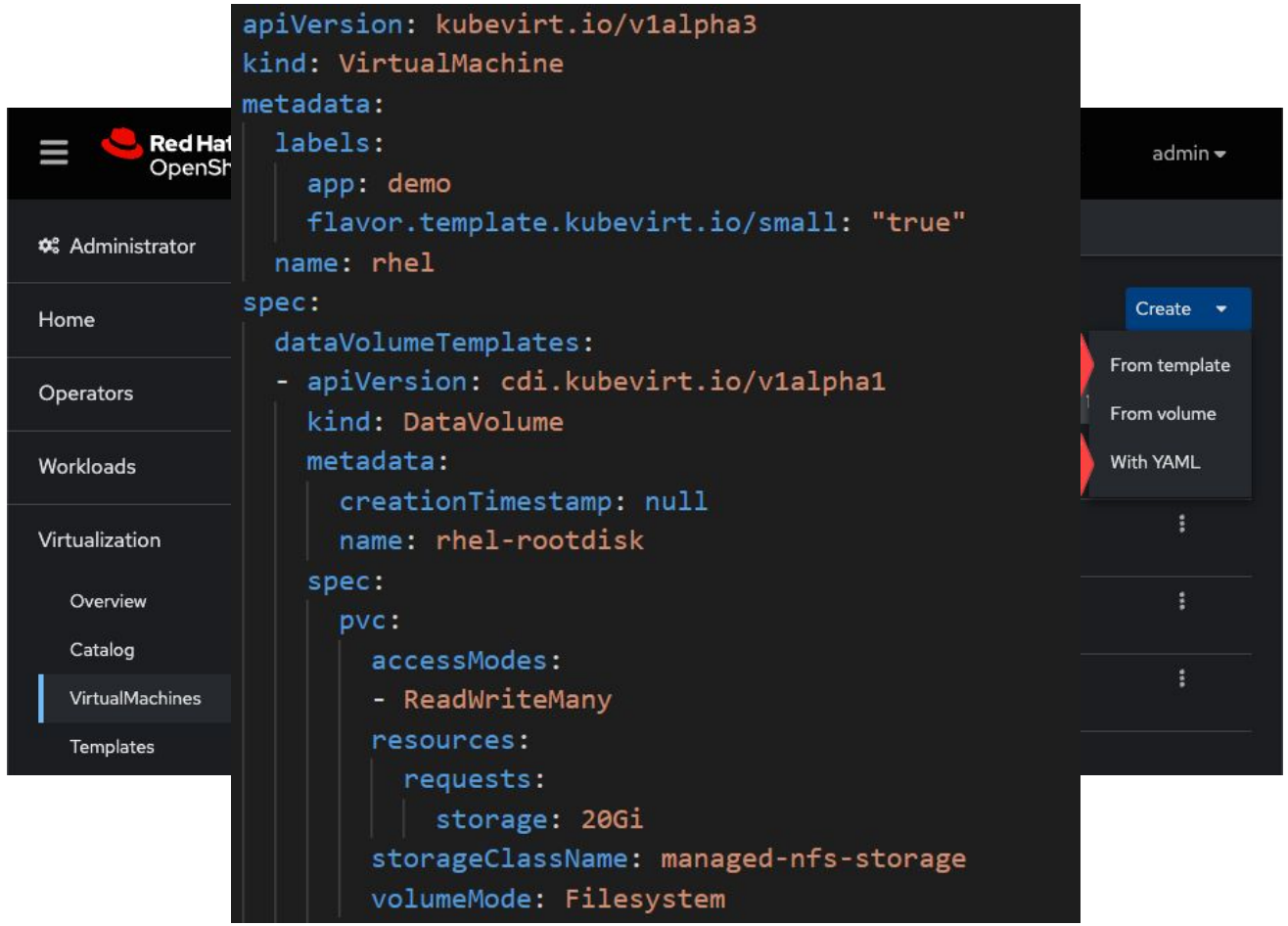


Architecture & Key Components



Virtual Machine Creation

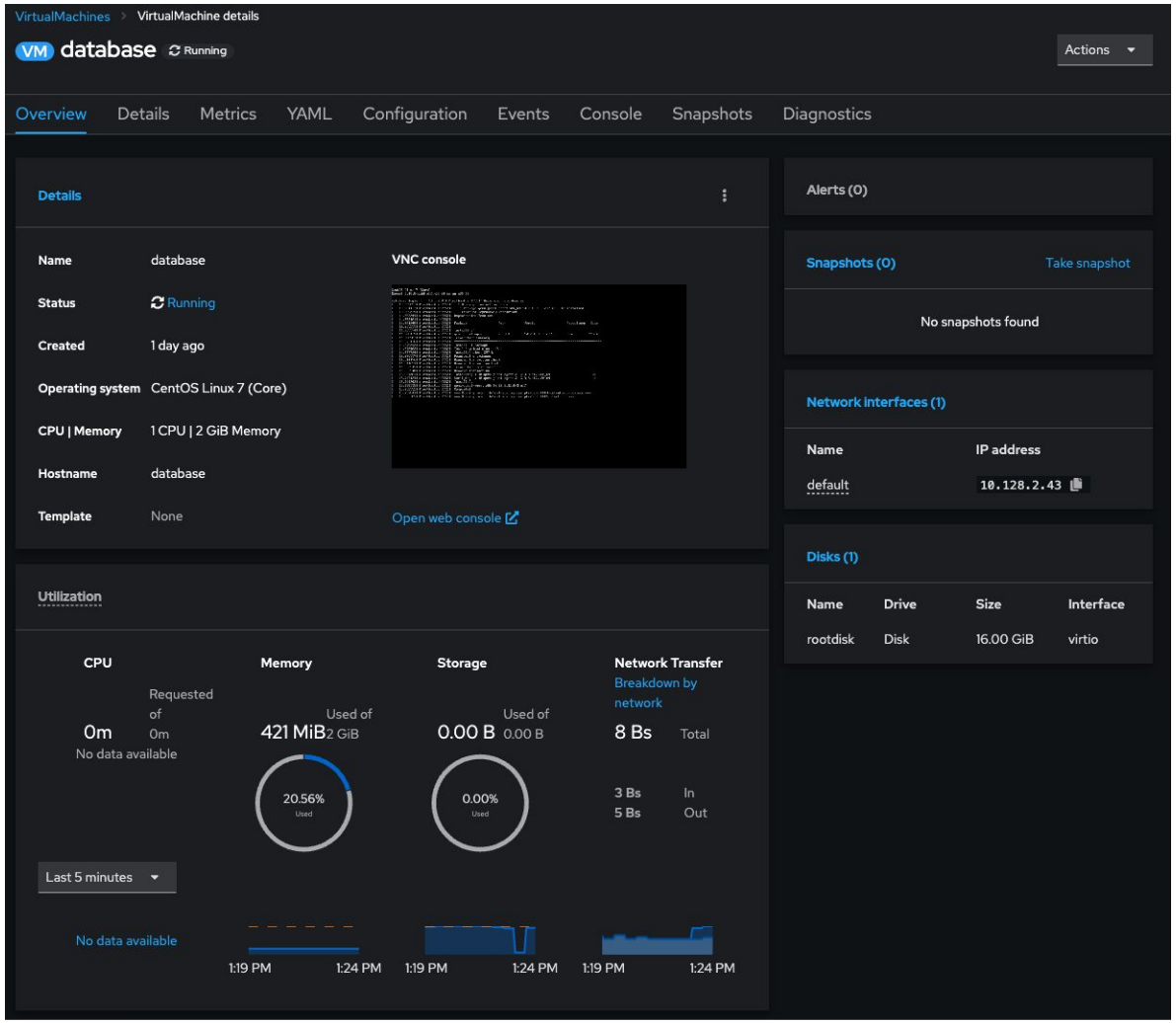
- Streamlined and simplified creation via the GUI or create VMs programmatically using YAML
- Full configuration options for compute, network, and storage resources
  - Clone VMs from templates or import disks using DataVolumes
  - Pre-defined and customizable presets for CPU/RAM allocations
  - Workload profile to tune KVM for expected behavior
- **TLDR**: Everything you'd expect



Red Hat

Virtual Machine - Overview

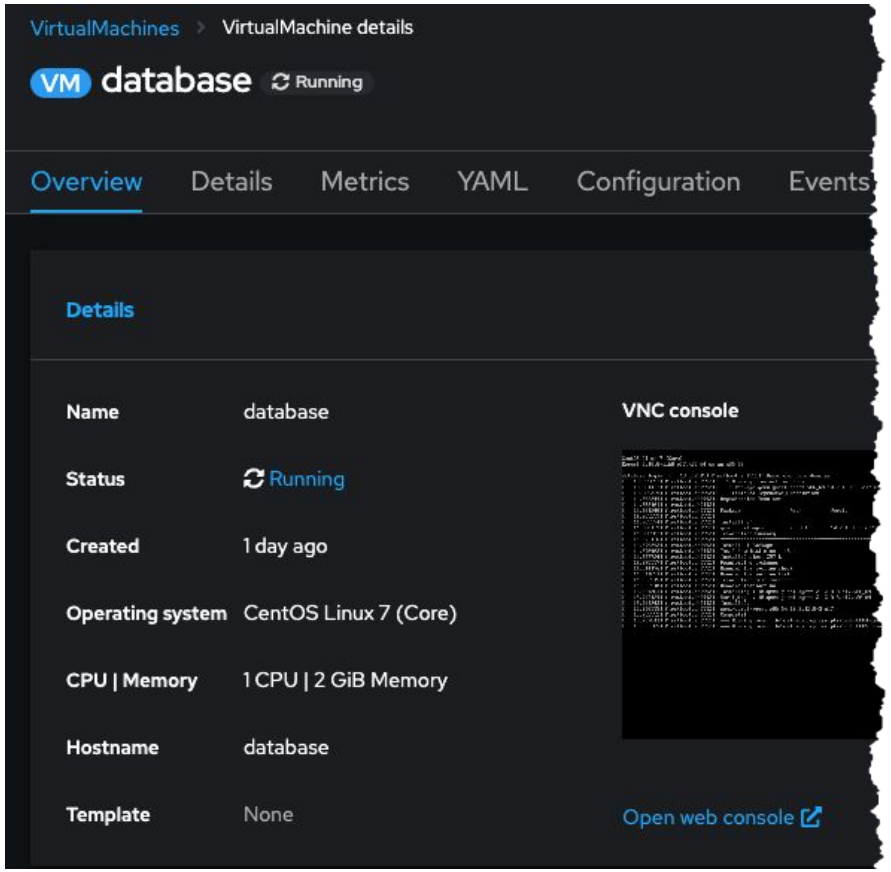
- General overview about the virtual machine
- Information populated from guest when integrations are available
  - IP address, etc.
- Inventory quickly shows configured hardware with access to view/manage
- Utilization reporting for CPU, RAM, disk, and network
- Events related to the Pod, scheduling, and resources are displayed

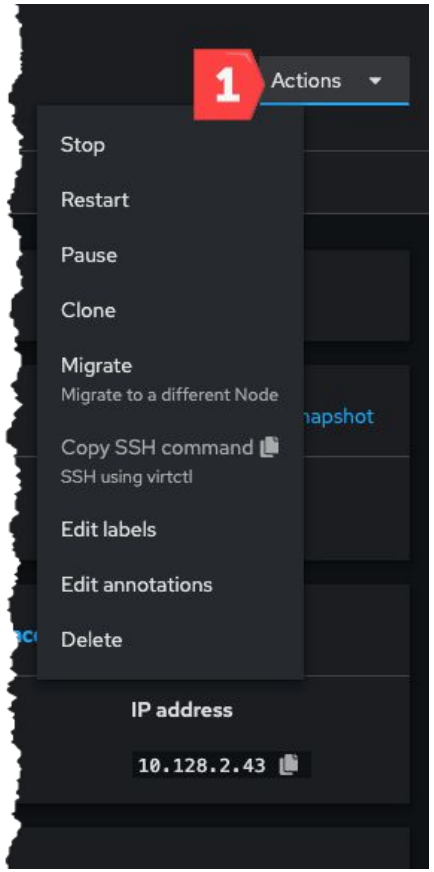


Red Hat

Virtual Machine - Actions

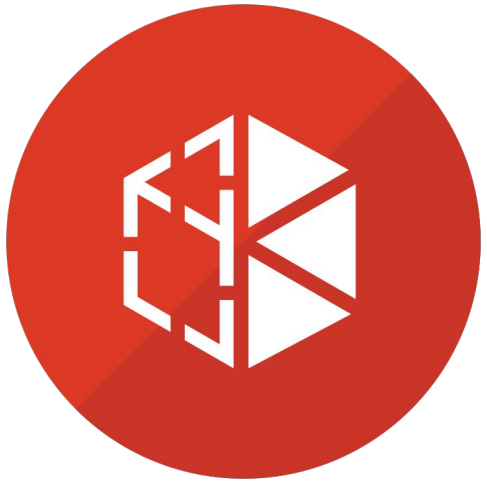
- Control the state and status of the virtual machine
- Actions menu allows quick access to common VM tasks
  - Start/stop/restart
  - Live migration
  - Clone
  - Edit application group, labels, and annotations
  - Delete
- Accessible from all tabs of VM details screen and the primary VM list





Red Hat

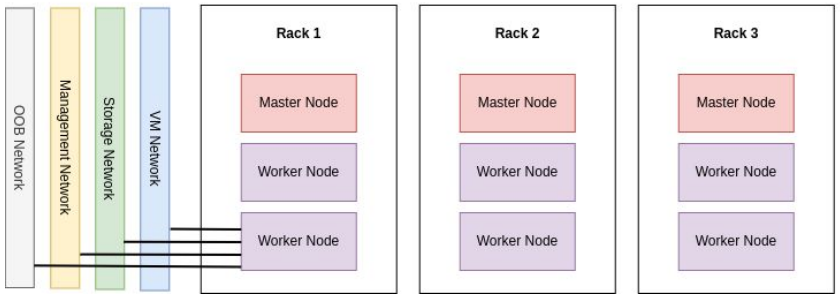
Considerations for OpenShift-Virt



Compute

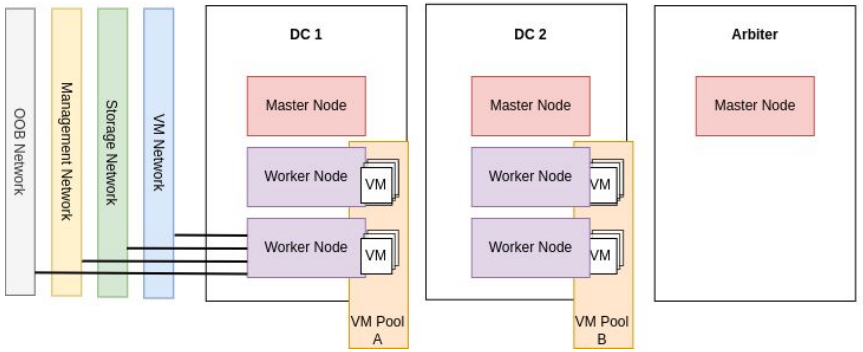
Topology

Single DC



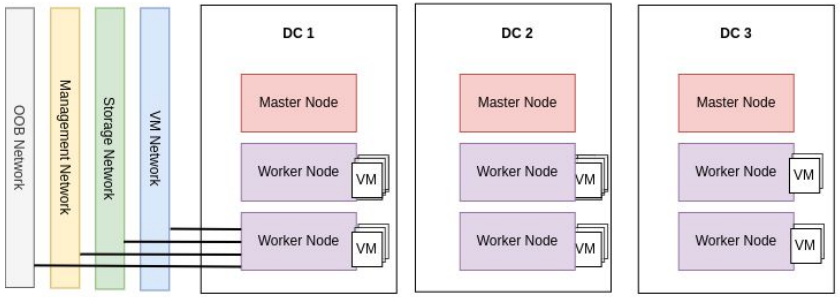
Metro (stretched)

2 DC + Arbiter



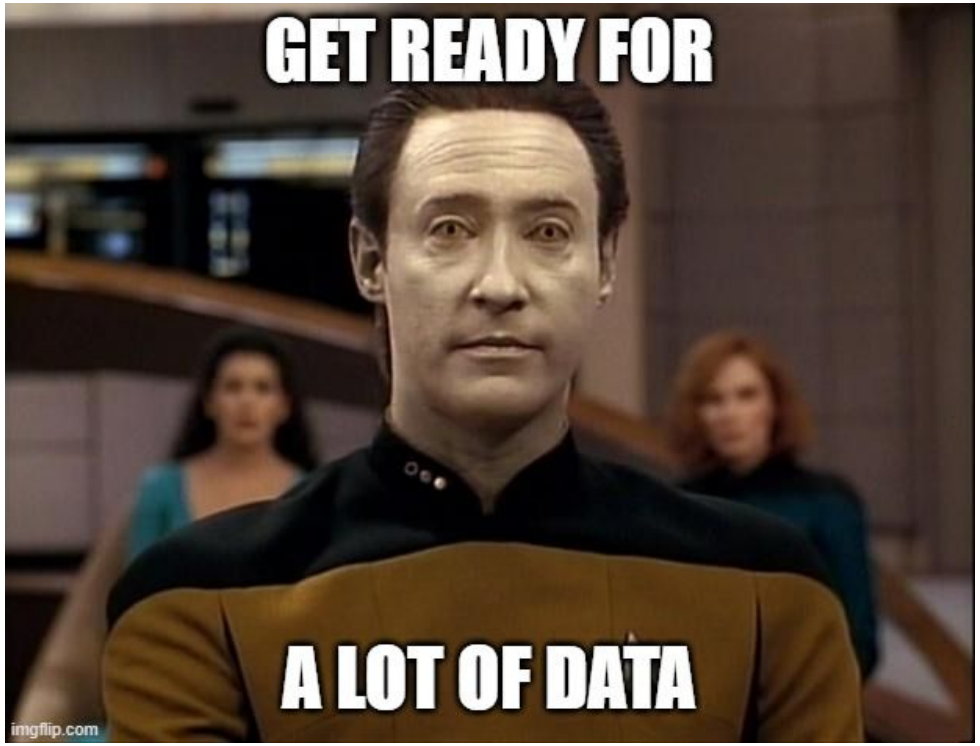
Advanced Cluster Management (ACM) Clusters?

Metro (stretched)



Hosted Control Planes (HCP)?

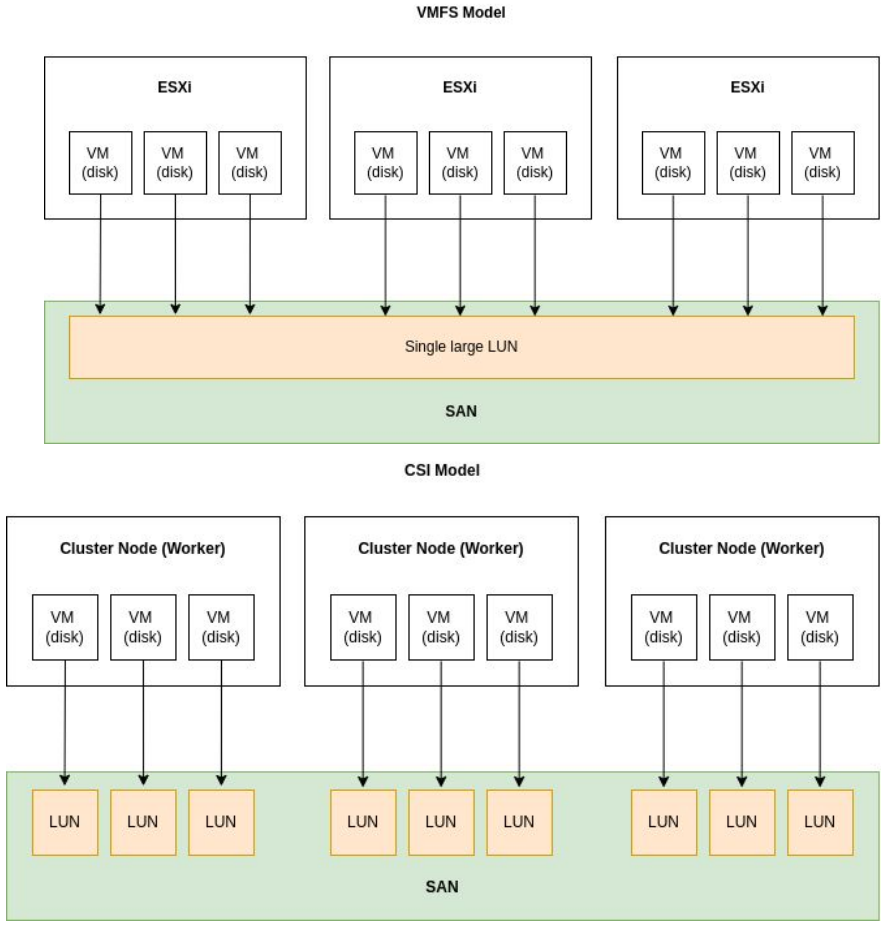
Storage



Storage

External Storage with a good CSI = :)

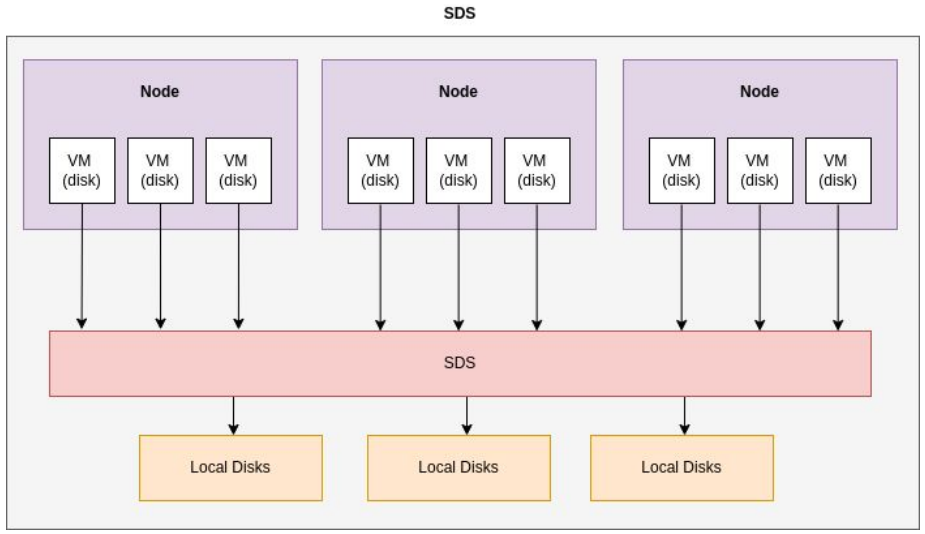
- **External storage via CSI** (e.g. NetApp Trident, Dell PowerStore)
- **Requires RWX support** from the CSI driver
- **Each volume maps to a LUN** (unlike other approaches e.g. VMware's VMFS model of carving VMDKs from larger LUNs)
- Must be comfortable with dynamic LUN provisioning
- Typically offers low latency and is often the preferred approach
- Success depends on the quality of the CSI driver and any LUN limits of the array



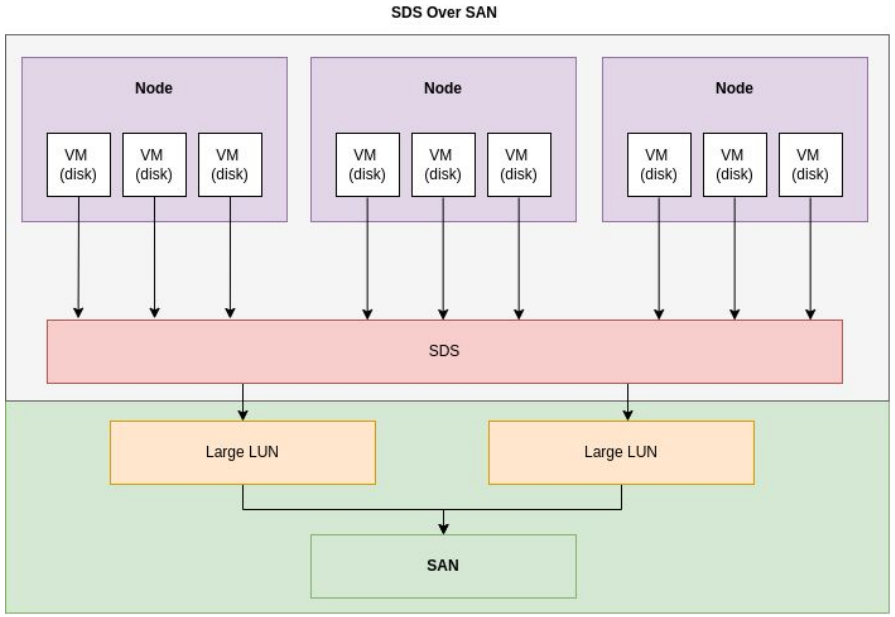
Storage

SDS

- Kubernetes can use “proper” SDS solutions like Rook (ODF) or Portworx with local disks, where replication is handled by the SDS itself. This approach is efficient and Kubernetes-native, but it puts the burden of managing storage directly on the platform team.



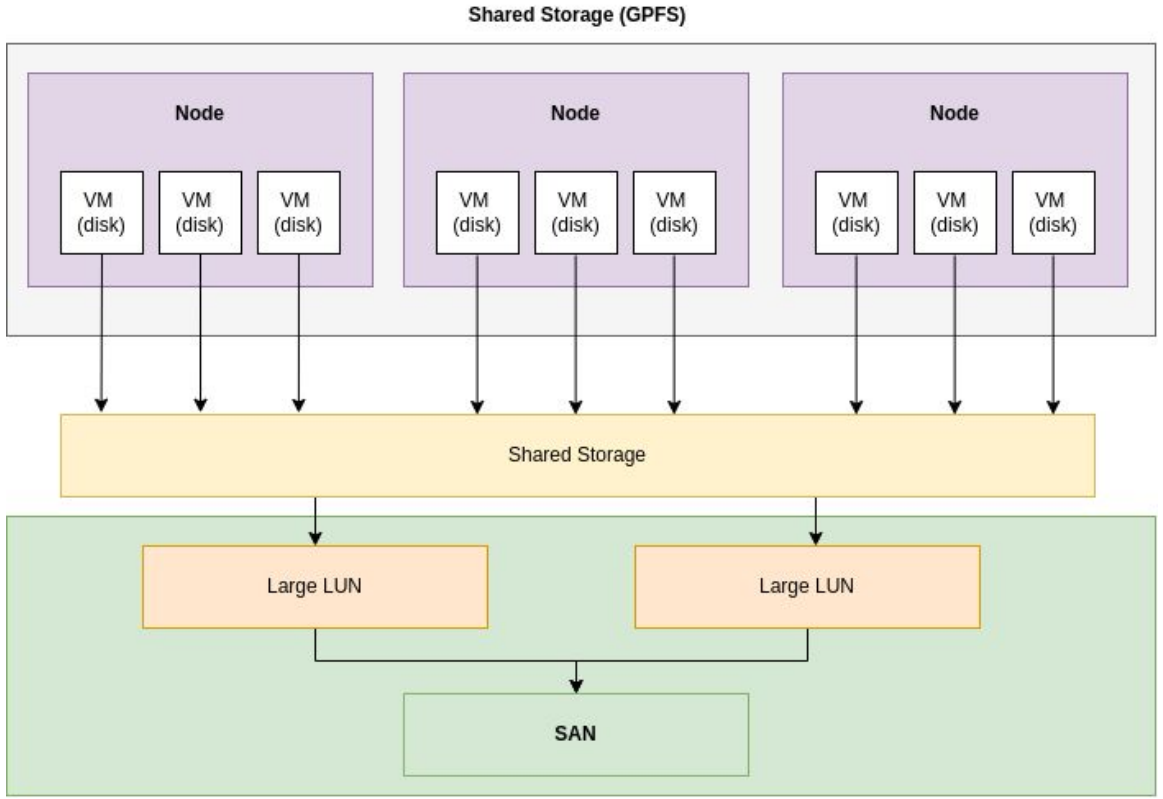
- The alternative is SDS over SAN, presenting LUNs to the SDS. This could add latency and write amplification while still creating operational overhead



Storage

Shared-disk File Systems

- Works with any SAN, including older models
- without direct CSI support
- Pretty new and pretty old... IBM GPFS

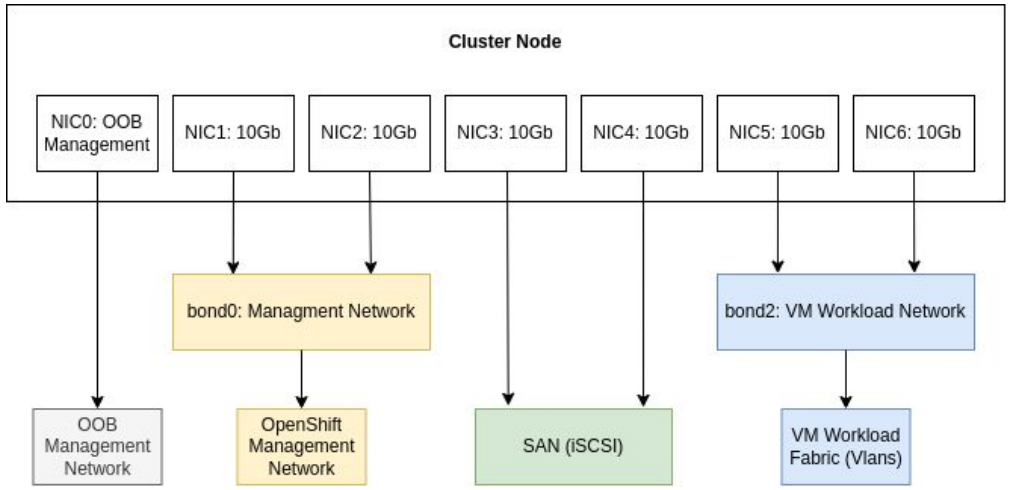


Networking

Networking

At the node level

- Bare metal networking considerations
  - VMs typically connect to external networks/VLANs, not the Kubernetes cluster network
  - External networking simplifies VM migration
  - May require BGP or advanced routing
  - Consider UDNs (User Defined Networks) for flexibility

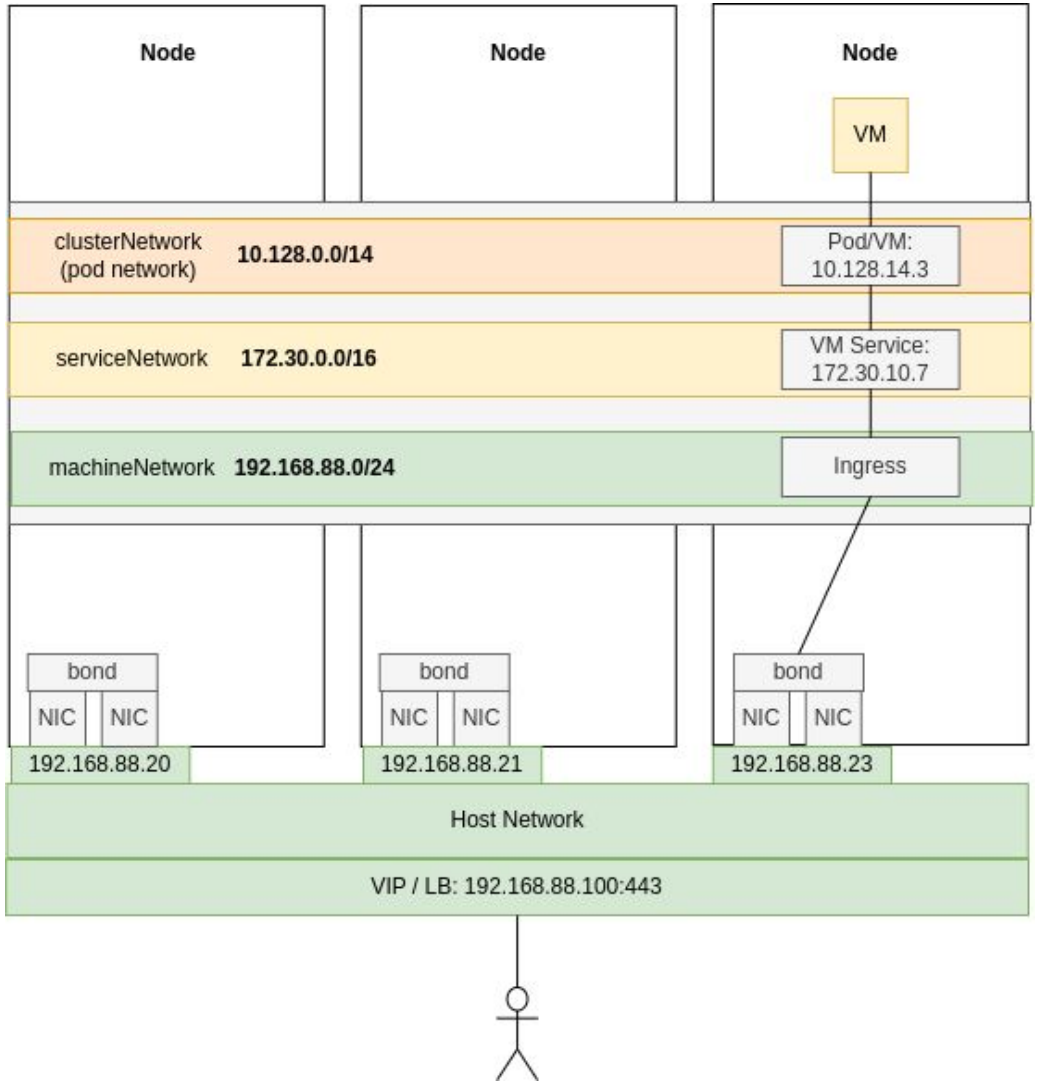


- Example NIC layout
  - 1 × BMC / OOB management
  - 2 × bonded for Kubernetes network & management
  - 2 × bonded for VM network
  - 2 × unbonded for multipath storage network

Networking

The OpenShift Default Network

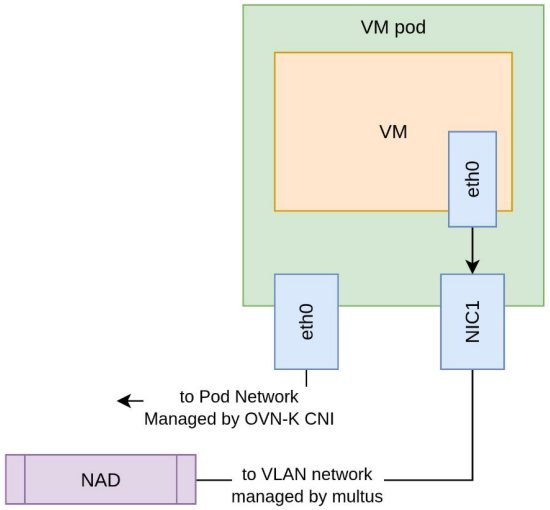
- The OG cluster k8s network
- Micro segmentation via Network Policies for tight security.
- Pods and VMs (though VMs need a little network thought).
- Pre-wired for ‘traditional’ Kubernetes networking you already know - Ingress, NodePort, Load Balancers etc

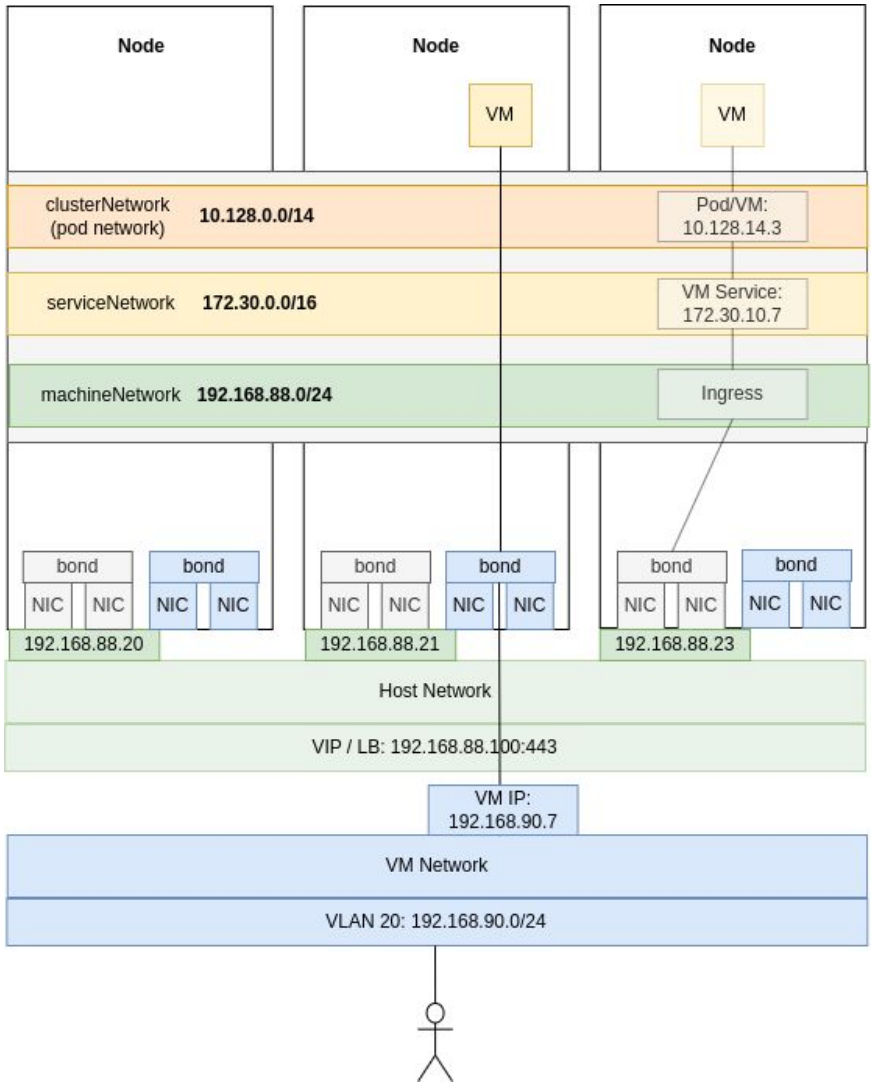


Networking

Bridge / VLAN

- bridge that connects your cluster's workloads to the outside world's existing VLANs (like vlan1, vlan2, etc.).
- Perfect for VMs that absolutely *must* sit on a pre existing enterprise network.
- OpenShift acts as the friendly middleman linking your internal apps to external networks.

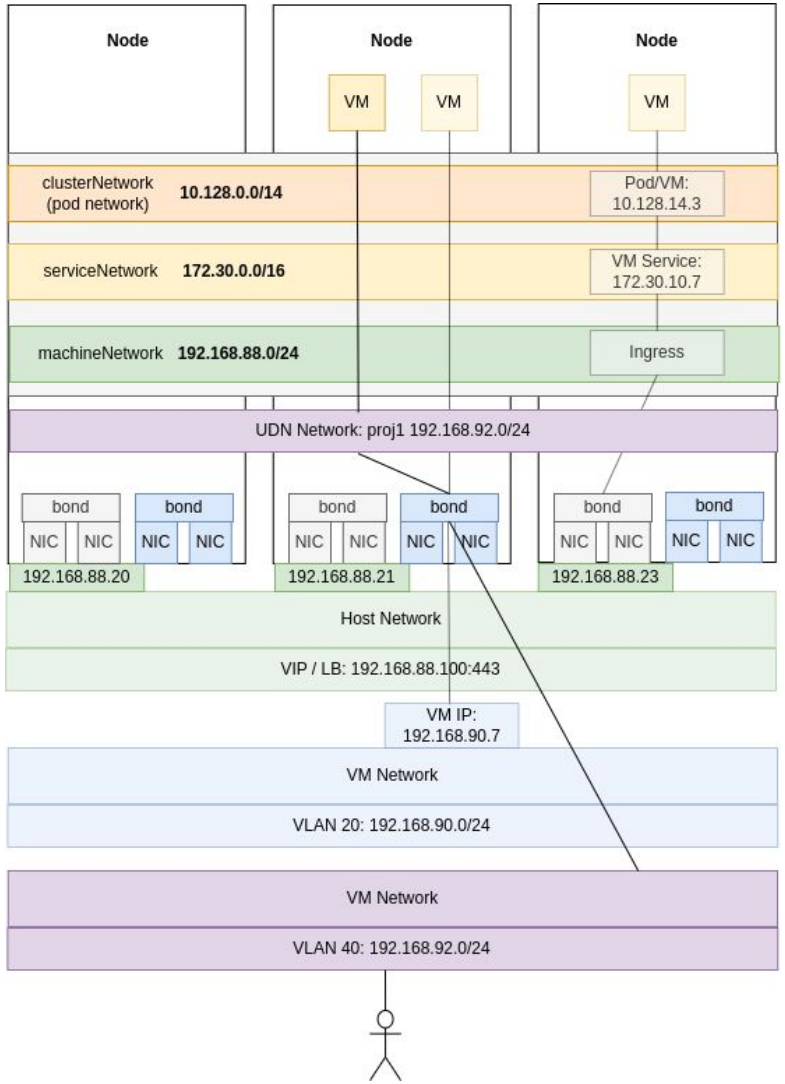




Networking

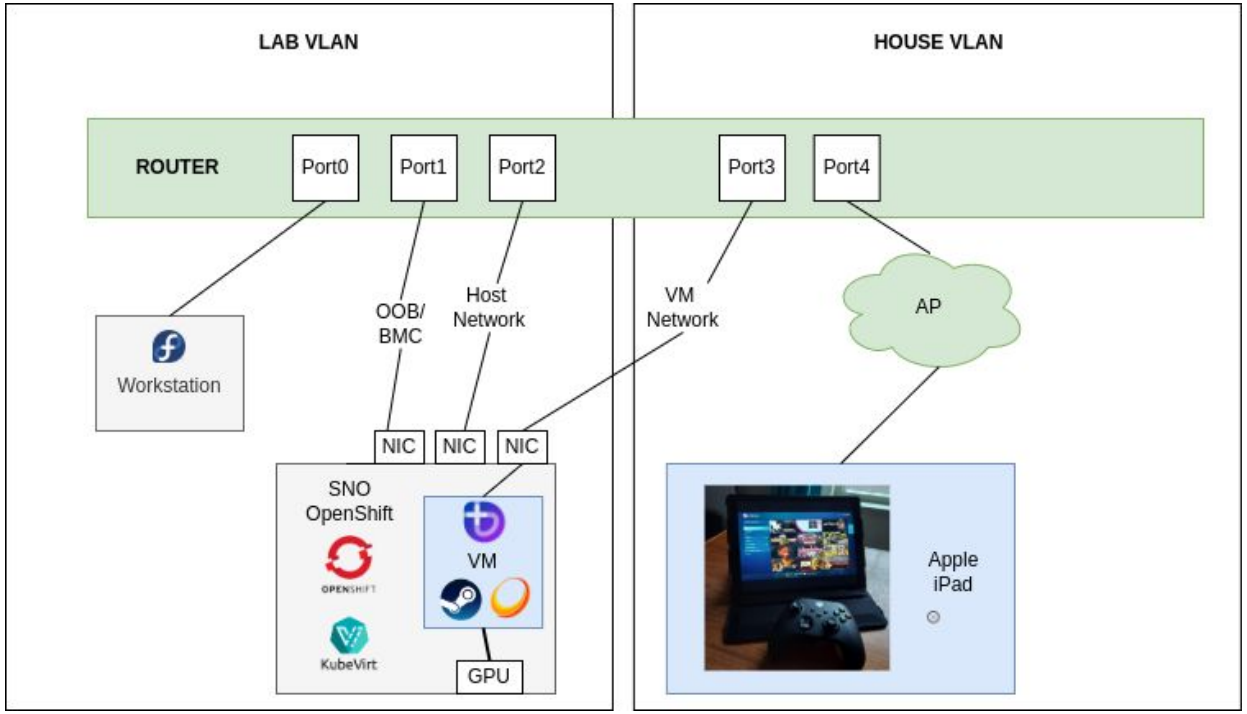
The User Defined Network (UDN) - NEW

- Network as a Service (NaaS) basically - with power comes responsibility!
- Total flexibility! Define your own VLANs and subnet.
- Best for when your VMs have complex, specific, or demanding networking requirements. It's very custom-tailored!



KubeVirt in Action

GEFORCE NOW™



Red Hat Summit

Connect

Thank you


linkedin.com/company/red-hat


facebook.com/redhatinc


youtube.com/user/RedHatVideos


twitter.com/RedHat