



OpenShift Virtualization

Technical Review
NYRHUG Sep 2020

Andrew Sullivan
Technical Marketing Manager
andrew.sullivan@redhat.com

Peter Lauterbach
Product Manager - Cloud Platforms
pelauter@redhat.com

OpenShift and Kubernetes crossing to early majority in IT adoption

Two emerging trends

1. Need for a better modernization strategy for virtual machine (VM)-based workloads

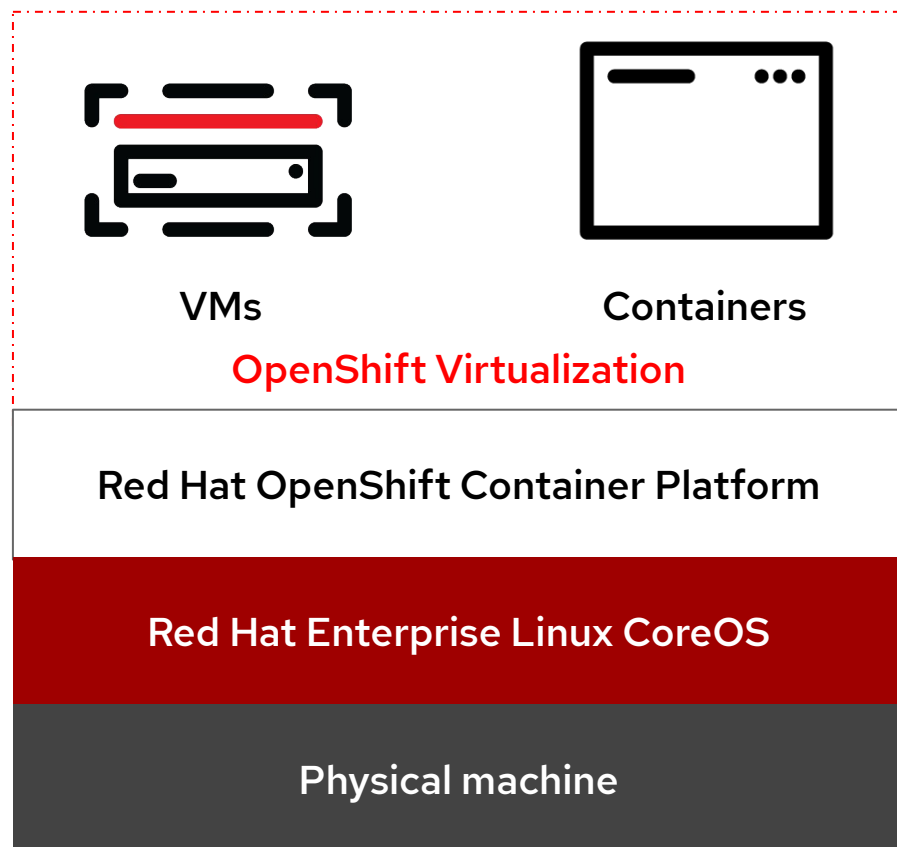
An all-or-nothing approach to containerization is too slow, so organizations have a large investment in virtual machines.

2. Desire for a single architecture for all workloads

Kubernetes supports stateful applications, and organizations desire to reduce costs by adopting a single cloud-native platform.

OpenShift and OpenShift Virtualization

Modernized workloads, support mixed applications consisting of VMs, containers, and serverless



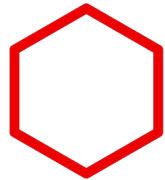
- Bring VMs to OpenShift 4.5+
- Accelerates application delivery with a single platform that can manage “mixed applications” with the same tools and teams
- Add VMs to new and existing applications
- Modernize legacy VM based applications over time, or maintain them as VMs
 - [SAP’s open source project “Gardener” leveraged Red Hat OpenShift Virtualization](#)
 - [Goldman Sachs Revamps Virtualization Infrastructure](#)

It is about managing both VMs and containers



Virtual machines

VMs have been built for decades, and they will not go away overnight.



Containers

Containers solve certain use cases and will continue to rise, but some VMs will remain.



Applications

VMs and containers will be used to build applications, and some might even build on both.

For existing applications

Vintage applications in VMs can be taken to OpenShift as well



Lift and shift

By using infrastructure migration solutions, existing VM-based applications can be migrated as-is to OpenShift from VMware and other platforms.



Windows apps in OpenShift now

Existing – a decade old–Microsoft Windows VMs can be brought into OpenShift as-is today, and be refactored over time as needed.

For application developers

Consistent developer experience (across VMs, containers, serverless)

Unified tools, process and pipelines for all apps

Application environments consist of VMs, containers, serverless and more. Development teams can now leverage the same tools, pipelines, and platform for building, managing and diagnosing issues with all apps.

Windows apps in OpenShift

Windows VMs can be brought into OpenShift as-is (and maintained if they're older), or refactored to use Windows Containers and Windows Server 2019

Refactor VMs on your schedule

All applications - old and new - can benefit from a unified control and DevOps pipeline. Teams can choose the applications they want to refactor or containerize when the time is right.

For infrastructure owners

Modernize and simplify your datacenter



Consistency of management

Align your system administrators on one simpler architecture to manage and expose

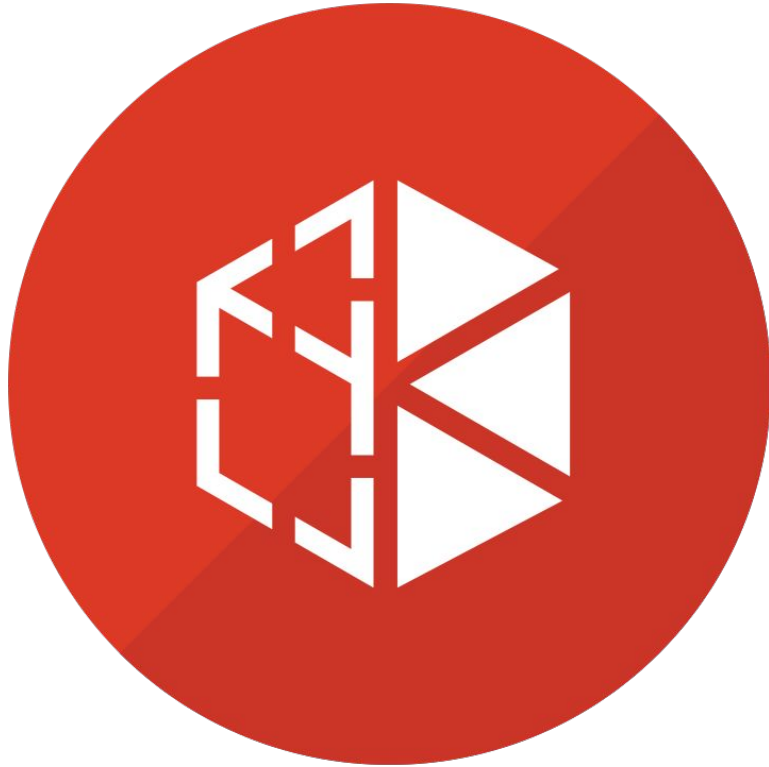


Focus on infrastructure and cluster operation

Focus your attention on maintaining the platform and it's infrastructure. Workload related aspects are covered by developer self-service. Security is intrinsic and inherited from KVM.

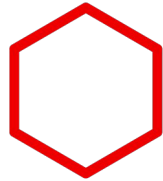
One modern platform
based on Kubernetes

OpenShift Virtualization



A virtualization API and runtime for OpenShift, built on KubeVirt, to run and manage virtual machines using a Kubernetes-native way

Guiding principles



VMs live in pods

VMs consume resources from where Kubernetes is providing them—pods.



Dedicated API to acknowledge differences

VMs have their specific functionality, thus a dedicated API to expose them.

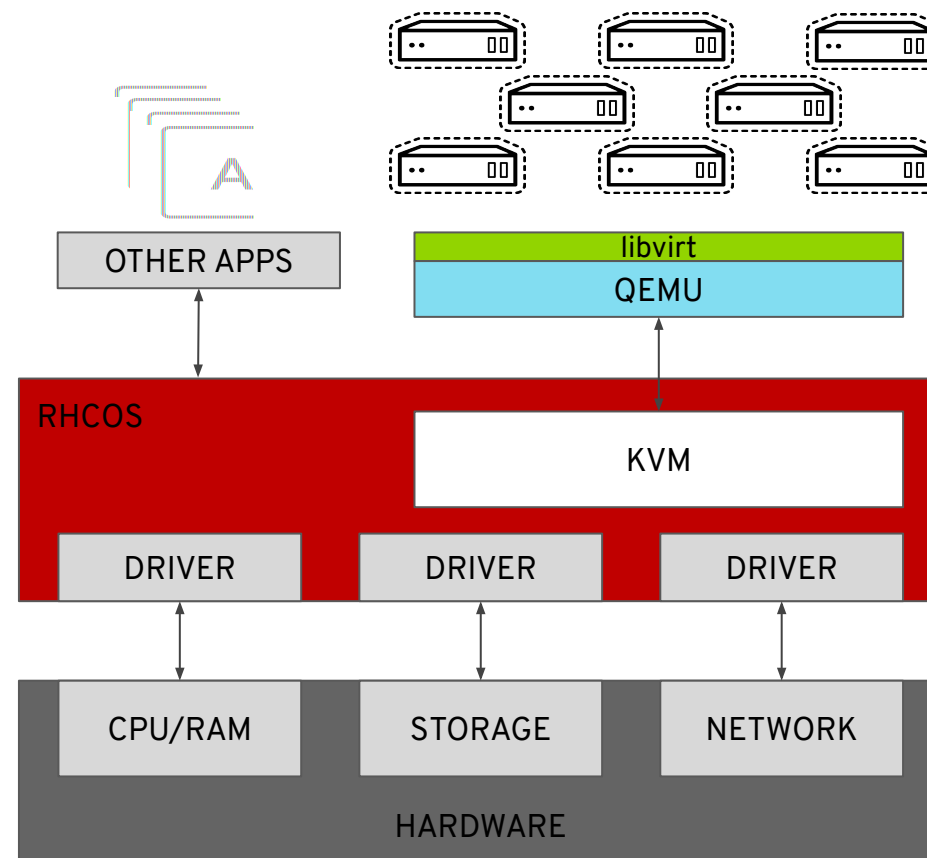


Kubernetes-native before virtualization features

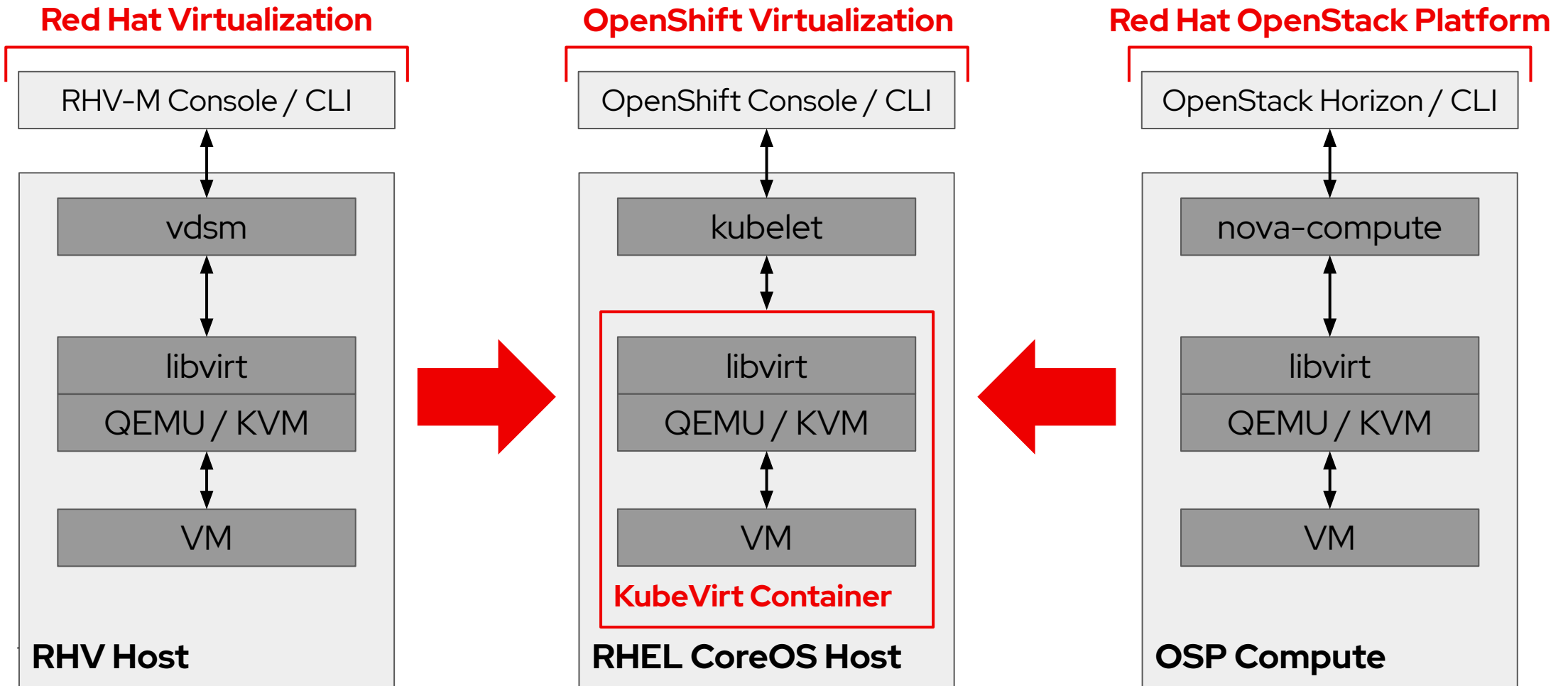
For usability virtualization features have to be solved in a Kubernetes-native way.

KVM-based (containerized) VMs

- KVM is a part of the Red Hat Enterprise Linux kernel
- QEMU uses KVM to execute virtual machines
- libvirt provides a management abstraction layer
- Red Hat Virtualization, Red Hat OpenStack Platform, and OpenShift Virtualization all leverage KVM, QEMU, and libvirt

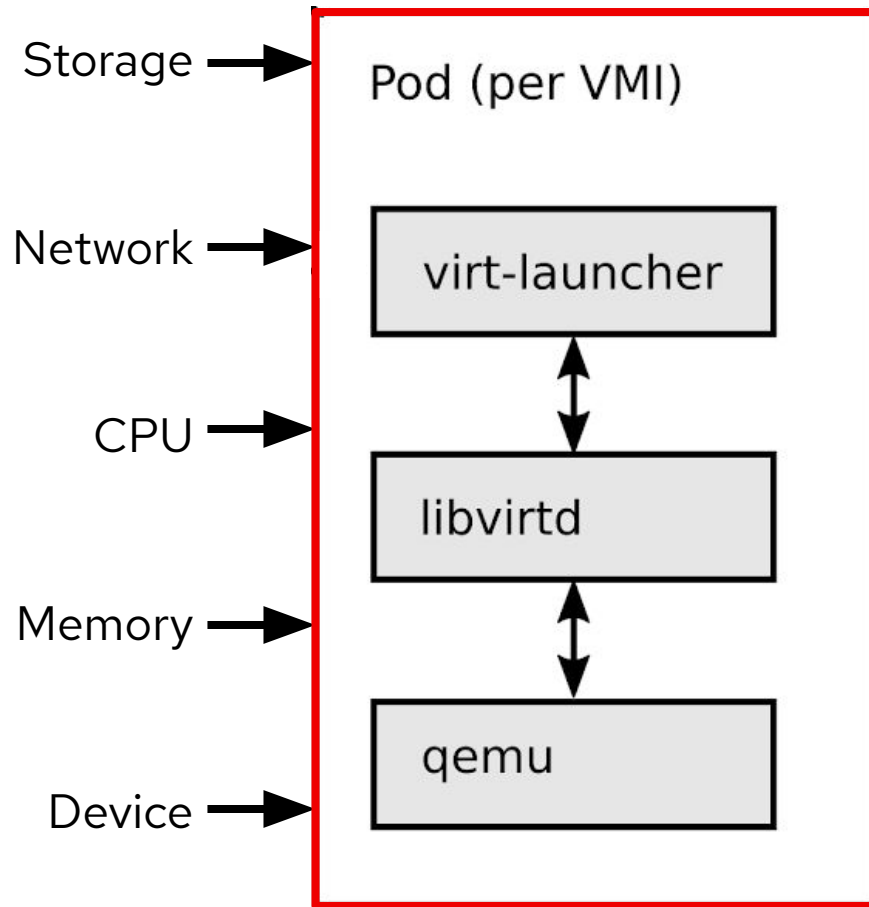


Containerizing KVM



VM in a pod

Containers for a unified resource model



Kubernetes resources

Every VM runs in a launcher pod. The launcher process will supervise, using `libvirt`, and provide pod integration.

RHEL-AV

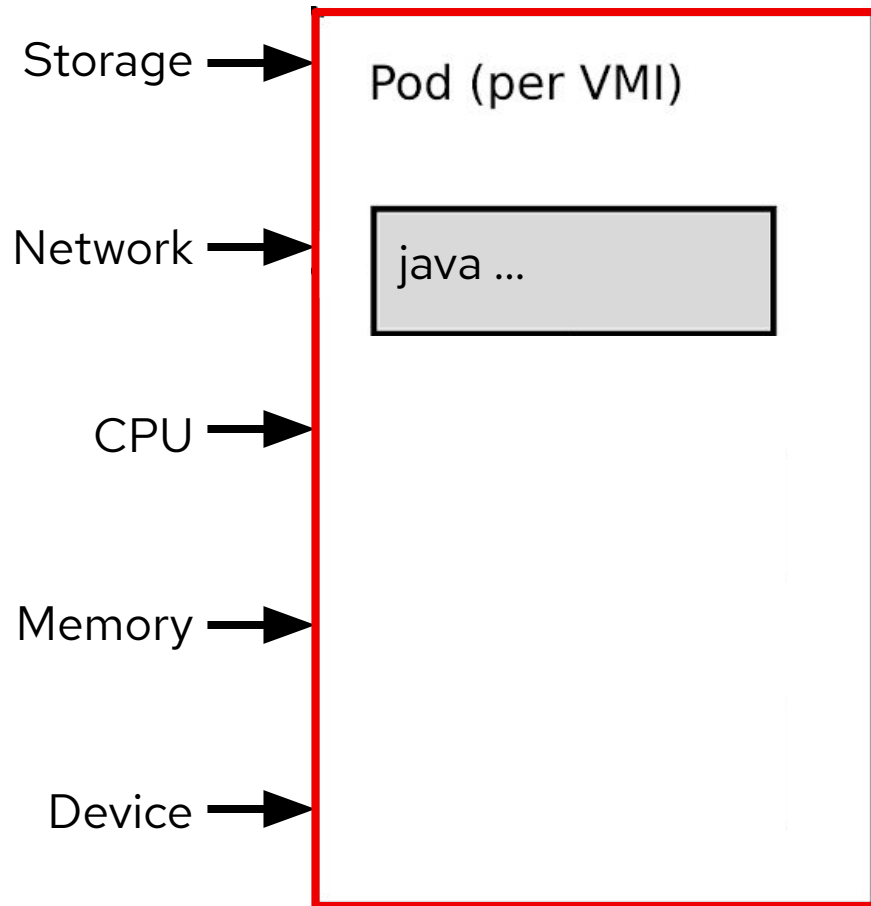
`libvirt` and `qemu` from RHEL-AV are mature, have high performance, provide stable abstractions, and have a minimal overhead.

Security - Defense in depth

Immutable RHCOS by default, SELinux MCS, plus KVM isolation - inherited from the Red Hat Portfolio stack

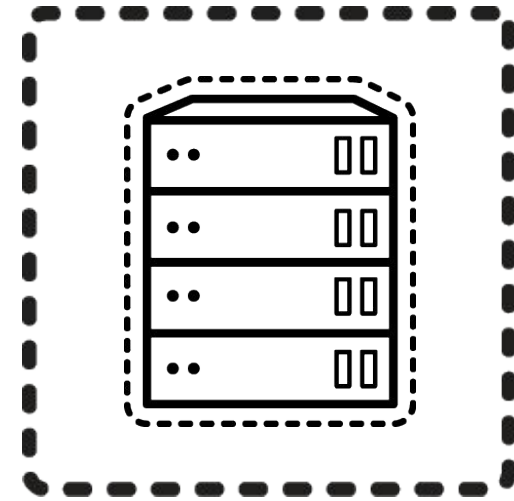
VM in a pod

... is just like any other process to Kubernetes

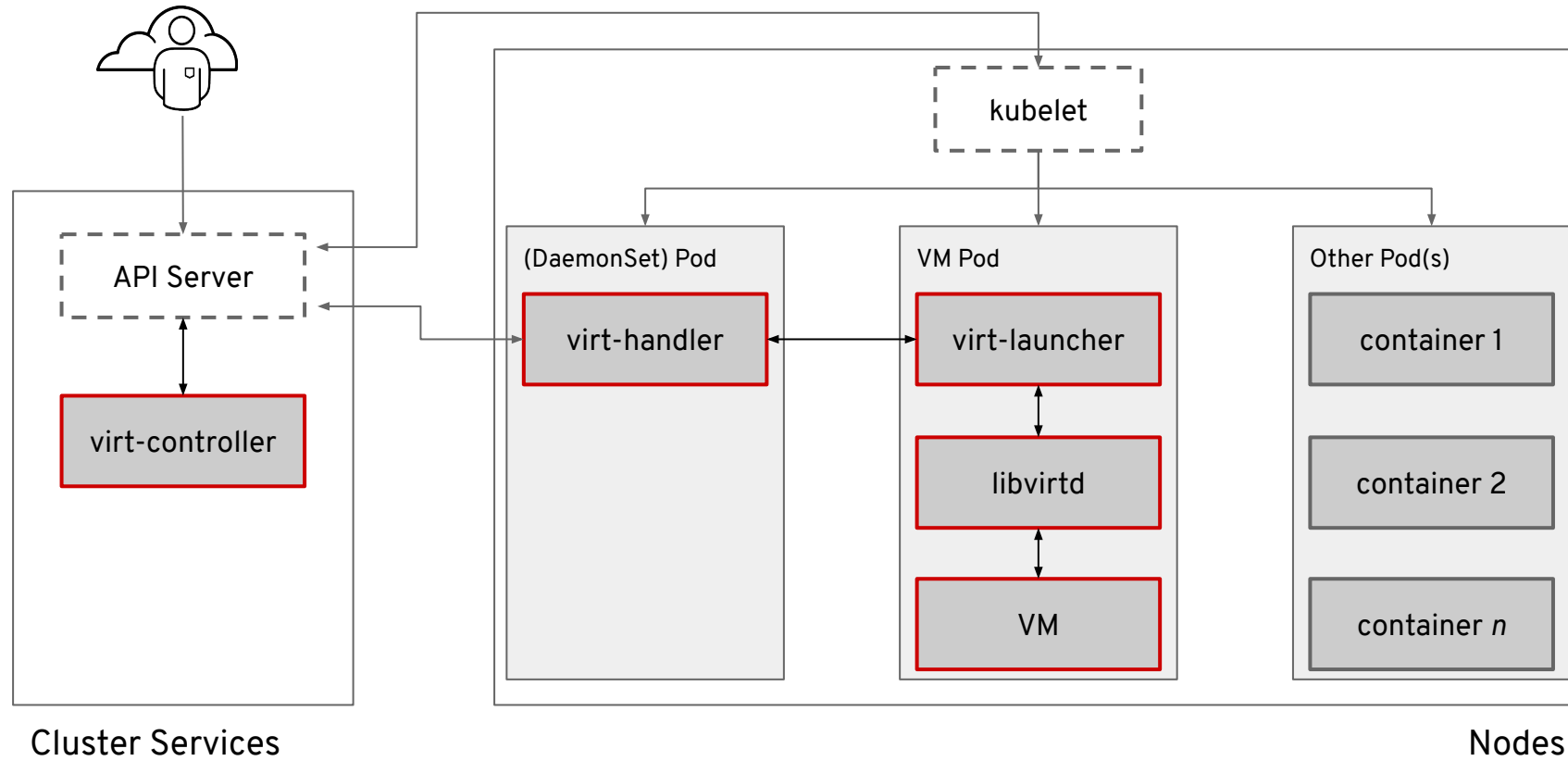


Containerized virtual machines

- Inherit many features and functions from Kubernetes
 - Scheduling, high availability, attach/detach resources
- Containerized virtual machines have the same limitations as non-containerized
 - CPU, RAM, etc. limitations dictated by libvirt and QEMU
 - Linux and Windows guest operating systems
- Storage
 - Use Persistent Volumes Claims (PVCs) for VM disks
 - Containerized Data Importer (CDI) import VM images
- Network
 - Inherit pod network by default
 - Multus enables direct connection to external network

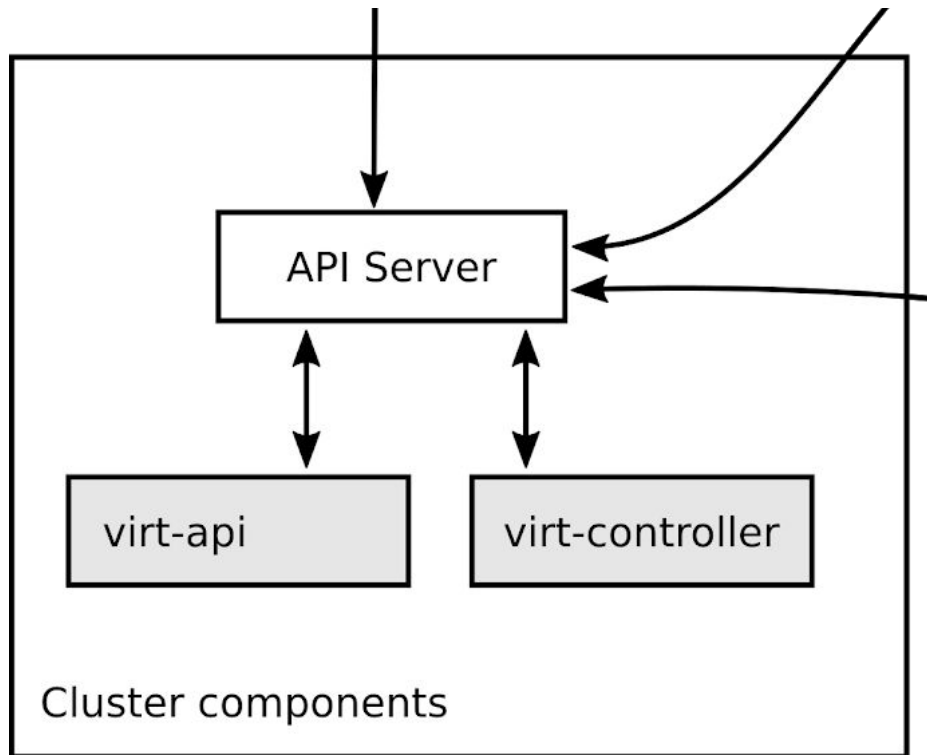


Architectural Overview



Adding virtualization to the Kubernetes APIs

How to build new APIs into Kubernetes



CRD and aggregated API servers

These are the ways to extend the Kubernetes API in order to support new entities.

For users, the new entities are indistinguishable from native resources.

Single API entry point for all workloads

All workloads (containers, VMs, and serverless) are managed through a single API.

<> vm.yaml

```
1  apiVersion: kubevirt.io/v1alpha3
2  kind: VirtualMachine
3  metadata:
4    name: testvm
5  spec:
6    running: false
7    template:
8      metadata:
9        labels:
10         team: Tiger
11     spec:
12       domain:
13         devices:
14           disks:
15             - disk:
16                 bus: virtio
17                 name: rootfs
18           interfaces:
19             - name: default
20       resources:
21         requests:
22           memory: 1GB
```

A dedicated virtualization API

To acknowledge differences and meet expectations

Declarative

Like anything in Kubernetes, the KubeVirt API is declarative, and follows Kubernetes API conventions.

Domain-specific

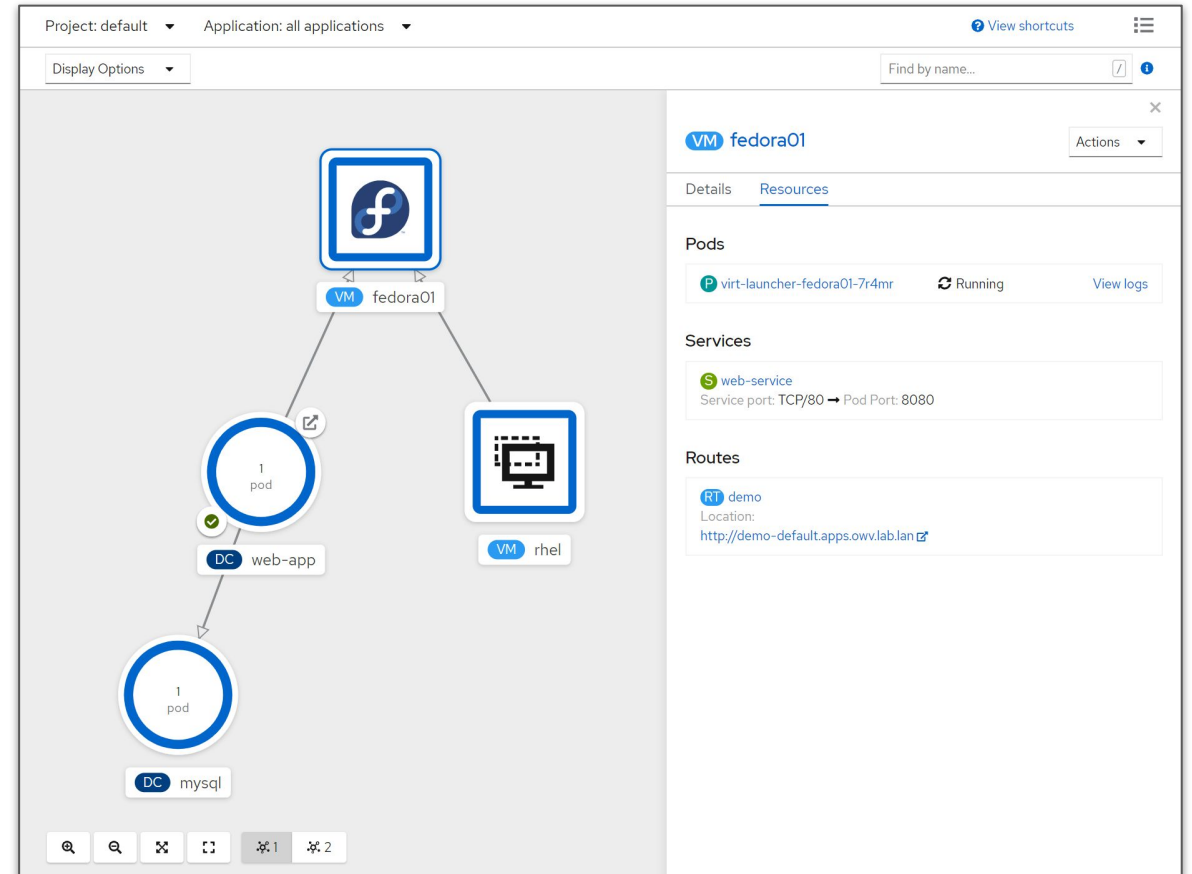
VMs are inherently differently defined than containers. Reusing the pod API is not explicit enough for all the necessary details—and due to differences.

Divide and conquer

Due to the dedicated API, it is straightforward to add virtualization-specific functionality, like live migration and graphical console access.

Using VMs and containers together

- Virtual Machines connected to pod networks are accessible using standard Kubernetes methods:
 - Service
 - Route
 - Pipelines
 - etc.
- Network policies apply to VM pods the same as application pods
- VM-to-pod, and vice-versa, communication happens over SDN or ingress depending on network connectivity



Demo

VMs ❤️ OpenShift

OpenShift Virtualization Roadmap

Near Term

(3-6 months)

CORE

- Streamline 'oc drain'
- Improve performance with Hyper-V enlightenments
- Improve golden image template workflow
- Sizing guidance for field

NETWORK

IPv6

- Dual stack
- Multicast
- IPv6 multi-address
- Enhance CNI test suite for VMs

STORAGE

- Offline snapshots
- Hotplug disk PoC
- Performance and scale of VMs on OCS
- Enhance CNI test suite for VMs

Mid Term

(6-9 months)

CORE

- **Virt gaps** - vCPU & vNUMA pinning
- **Security** - FIPS
- **Developer**
 - Using VMs naturally in Developer Pipelines

NETWORK

- Hotplug - NIC
- **Ecosystem CNI certification**
 - Tigera - Calico

STORAGE

- Application consistent snapshots and cloning
- Improve default performance with io=native
- **Backup and DR**
 - via OCS data protection
- Ecosystem CSI certification

Long Term

(9+ months)

CORE

- OCP as on-prem infrastructure
- **Cloud** - Bare Metal where supported by public cloud provider
- **Mass migration**- VMw to OCP & RHV to OCP

NETWORK

- OVS HW offload
- **Ecosystem CNI certification**
 - Nuage
 - Juniper

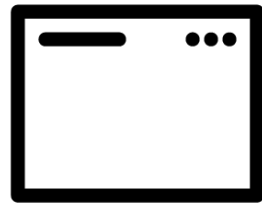
STORAGE

- **High Availability**

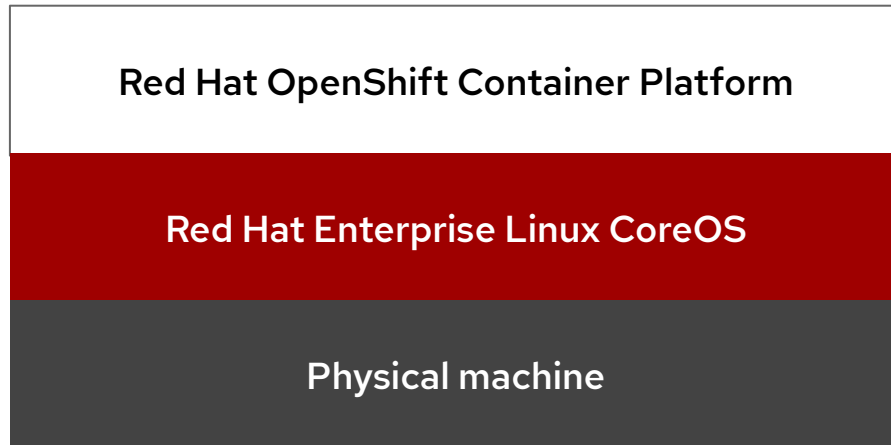
Modern virtualization using OpenShift



VMs



Containers



- Modernize on your timeline
 - Strongly aligned with Kubernetes
 - Get started quickly, expand as needed
- Migrate existing workloads easily using integrated tools
- Get started
 - Align your teams to a single platform
 - Existing skill sets still apply!

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

 [facebook.com/redhatinc](https://www.facebook.com/redhatinc)

 twitter.com/RedHat