

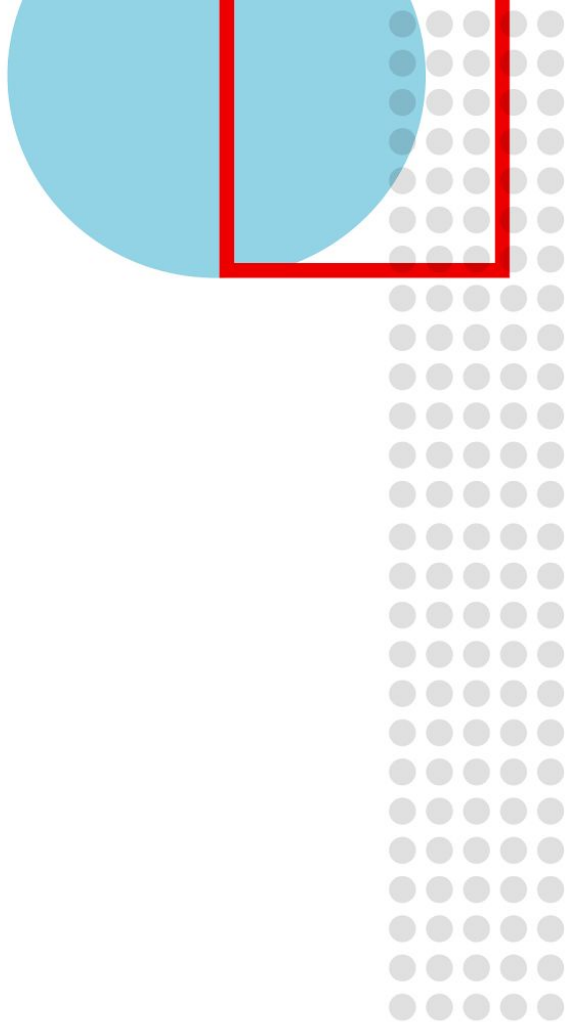


Red Hat
Summit

Containers and VMs together

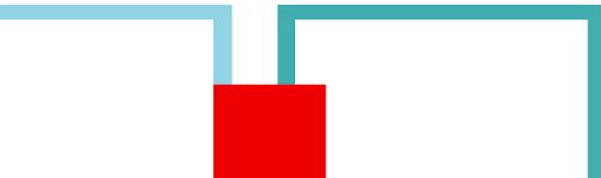
Openshift Virtualization

Artur Pająk
Solution Architect
RH CEE



 Gartner predicts that, by 2022, more than 75% of global organizations will be running containerized applications in production, which is a significant increase from fewer than 30% in 2019.

Gartner, Assessing Kubernetes for Hybrid and Multicloud Application Portability, June 2020.



Virtualization continues to dominate

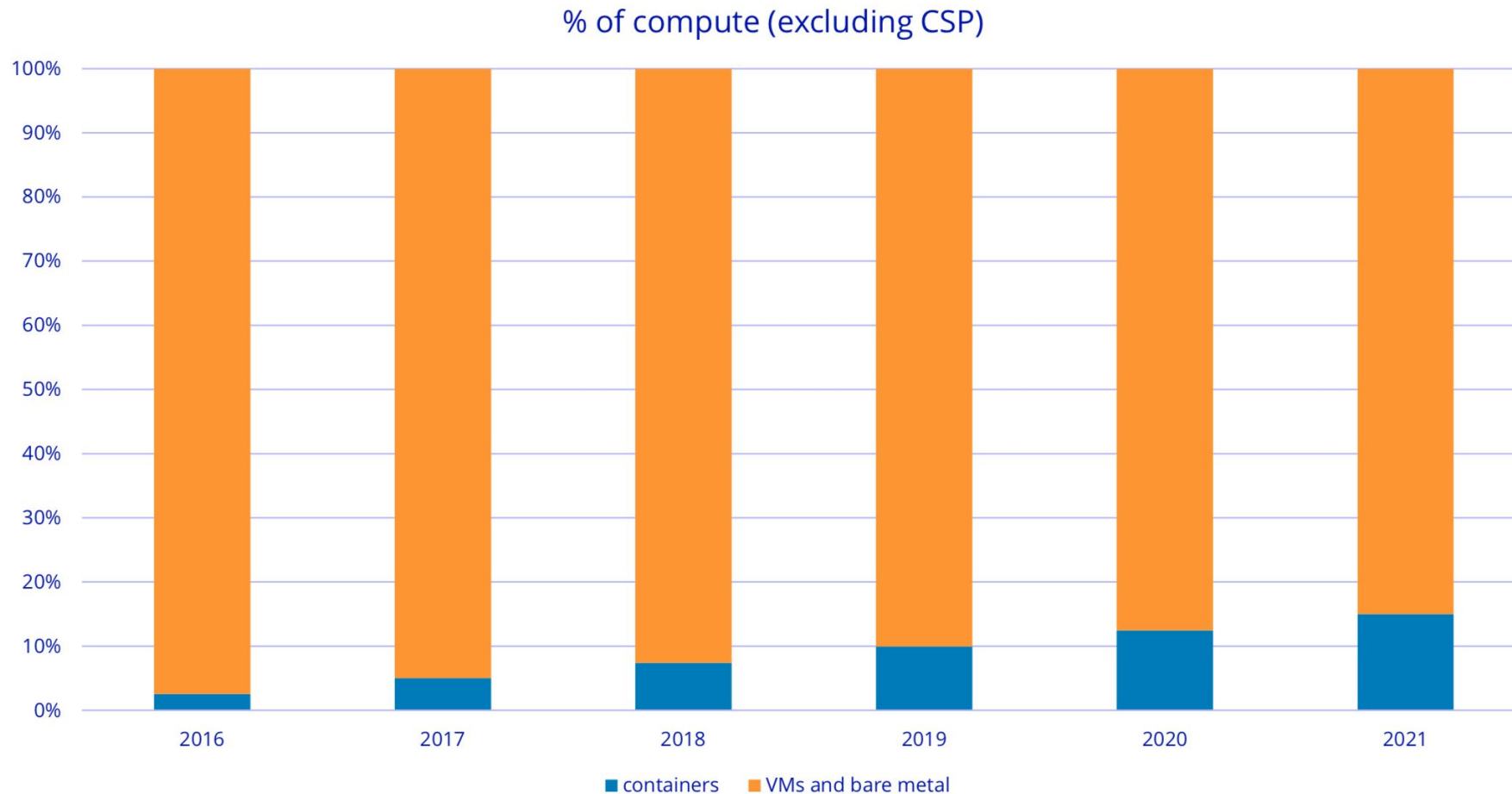


Image source: IDC Container Forecast, 2018

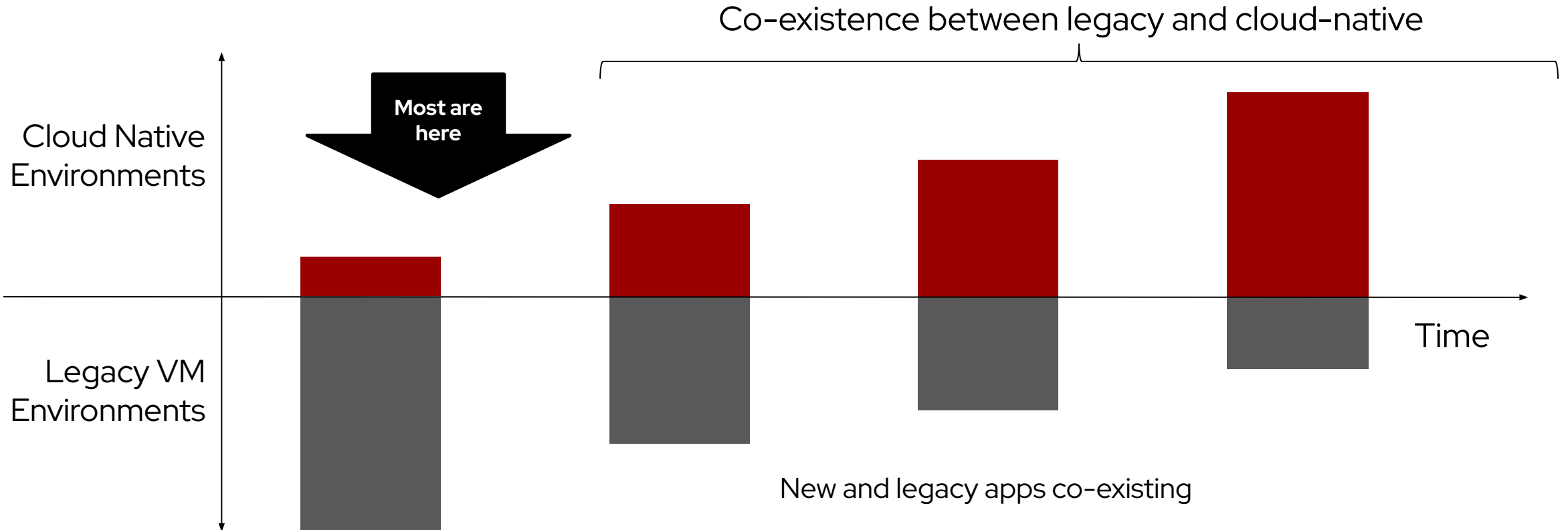
The move from Virtual Machines to Containers

Accelerate Digital Transformations

Improve Developer Productivity

Increase Operational Efficiency and Standardization

Traditional virtualization cannot support these cloud-native applications.

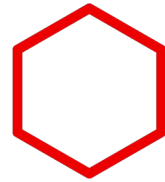


It is about managing both VMs and containers



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.

The Paths to Application Modernization

The paths to Application Modernization

Path #2: Containerize - Replatform to Containers/Kubernetes

"Lift, tinker, and shift" workloads from WebSphere/EAP to OpenShift containers

Path #4: Retire & Replace

A ground-up rebuild, keeping the legacy application up-and-running, whilst a new version of the application is developed, leveraging a Cloud Native development approach

Path #3+: Refactor Plus

Staged approach: individually replace/develop application's services as microservices, incorporating more advanced capabilities into applications, such as AI/ML and event driven approaches

Path #3: Refactor

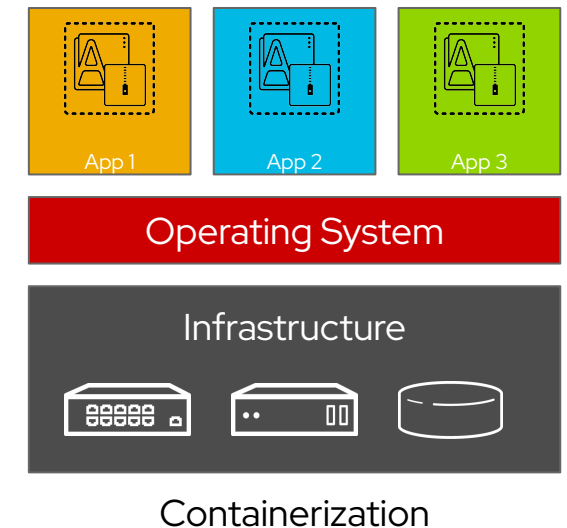
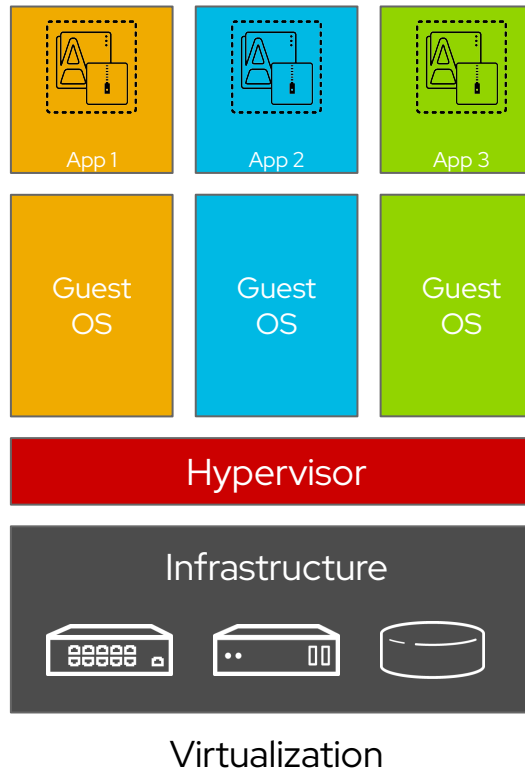
Staged approach: individually replace/develop application's services as microservices

Path #1: Virtualize - Rehost Application Server to OpenShift

"Lift, and shift" Java application on App Server to JBoss or WebSphere on OpenShift

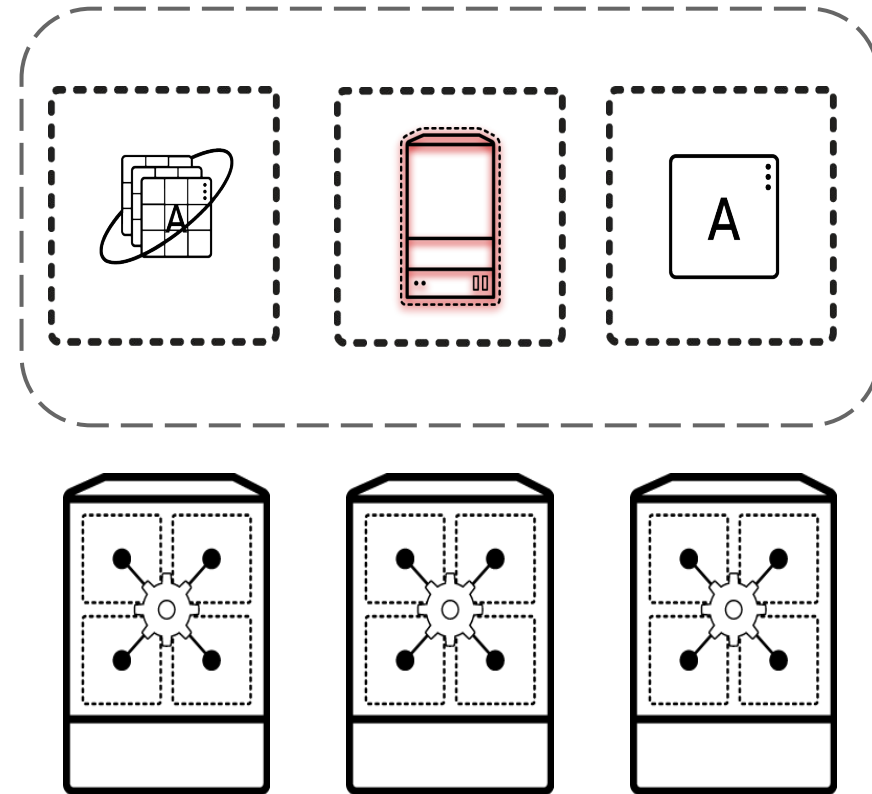
Containers are not virtual machines

- Containers are process isolation
- Kernel namespaces provide isolation and cgroups provide resource controls
- No hypervisor needed for containers
- Contain only binaries, libraries, and tools which are needed by the application
- Ephemeral



Virtual machines can be put into containers

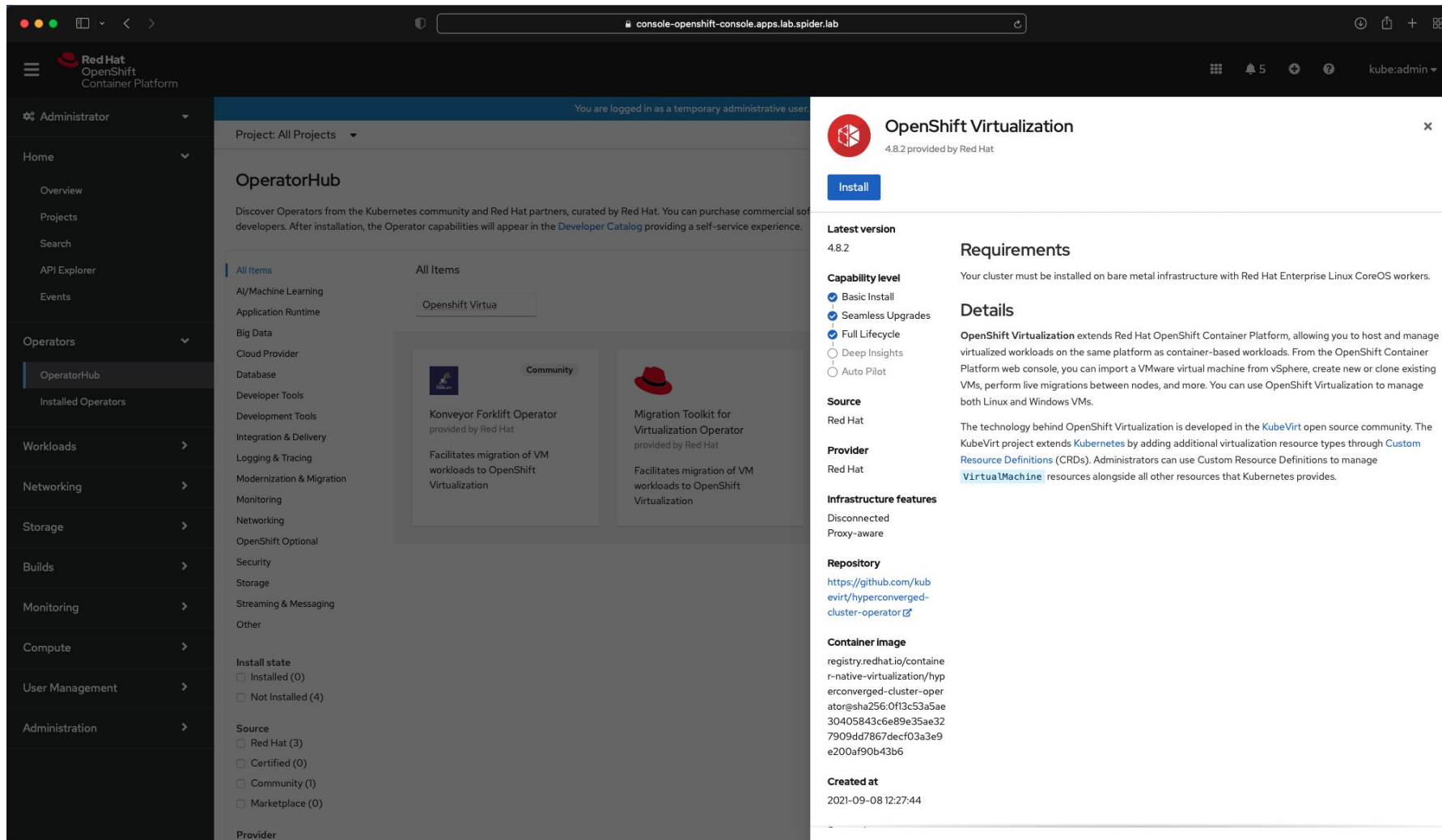
- A KVM virtual machine is a process
- Containers encapsulate processes
- Both have the same underlying resource needs:
 - Compute
 - Network
 - (sometimes) Storage



OpenShift Virtualization



How to install?

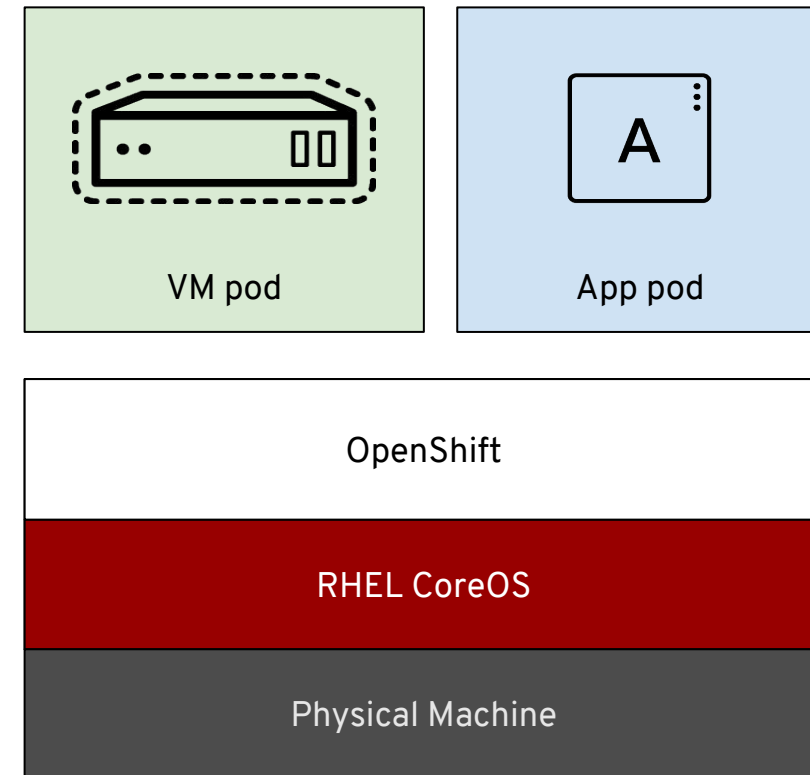


The screenshot displays the OpenShift Virtualization console interface. On the left, a sidebar menu shows navigation options: Administrator, Home, Overview, Projects, Search, API Explorer, Events, Operators (selected), Installed Operators, Workloads, Networking, Storage, Builds, Monitoring, Compute, User Management, and Administration. The main content area is titled 'OperatorHub' and shows a list of operators. Two operators are visible: 'Konveyor Forklift Operator' (provided by Red Hat) and 'Migration Toolkit for Virtualization Operator' (provided by Red Hat). The 'Migration Toolkit for Virtualization Operator' is highlighted. On the right, a details panel for 'OpenShift Virtualization' (version 4.8.2) is shown. It includes an 'Install' button, a 'Latest version' section, a 'Requirements' section stating that the cluster must be installed on bare metal infrastructure with Red Hat Enterprise Linux CoreOS workers, and a 'Details' section explaining that OpenShift Virtualization extends Red Hat OpenShift Container Platform to host and manage virtualized workloads. The details section also mentions that the technology is developed in the KubeVirt open source community and that administrators can use Custom Resource Definitions (CRDs) to manage VirtualMachine resources. The details panel also lists the source (Red Hat), provider (Red Hat), infrastructure features (Disconnected, Proxy-aware), repository (https://github.com/kubevirt/hyperconverged-cluster-operator), container image (registry.redhat.io/containers-native-virtualization/hyperconverged-cluster-operator@sha256:0f13c53a5ae30405843c6e89e35ae327909dd7867decf03a3e9e200af90b43b6), and creation time (2021-09-08 12:27:44).

VMs in a container world

Virtual machines in a container world

- Provides a way to transition application components which can't be directly containerized into a Kubernetes system
 - Integrates directly into existing k8s clusters
 - Follows Kubernetes paradigms:
 - Container Networking Interface (CNI)
 - Container Storage Interface (CSI)
 - Custom Resource Definitions (CRD, CR)
- Schedule, connect, and consume VM resources as container-native

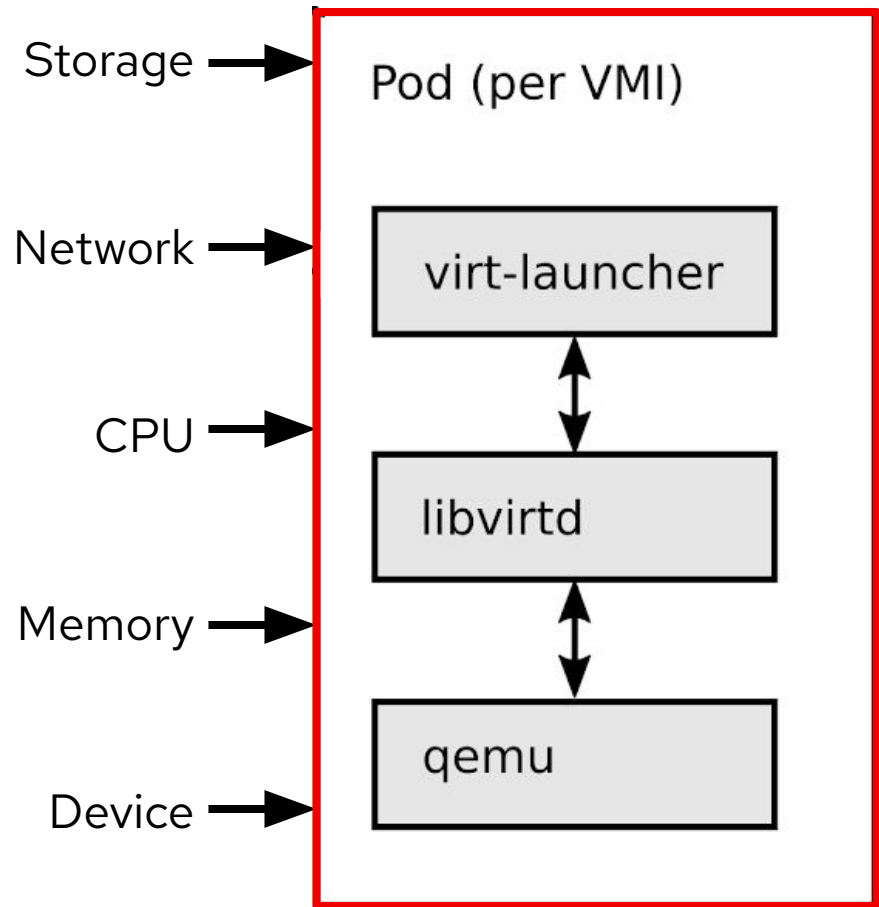


Virtualization native to Kubernetes

- Operators are a Kubernetes-native way to introduce new capabilities
- New CustomResourceDefinitions (CRDs) for native VM integration, for example:
 - VirtualMachine
 - VirtualMachineInstance
 - VirtualMachineInstanceMigration
 - VirtualMachineSnapshot
 - DataVolume

```
apiVersion: kubevirt.io/v1alpha3
kind: VirtualMachine
metadata:
  labels:
    app: demo
    flavor.template.kubevirt.io/small: "true"
    name: rhel
spec:
  dataVolumeTemplates:
  - apiVersion: cdi.kubevirt.io/v1alpha1
    kind: DataVolume
    metadata:
      creationTimestamp: null
      name: rhel-rootdisk
    spec:
      pvc:
        accessModes:
        - ReadWriteMany
        resources:
          requests:
            storage: 20Gi
        storageClassName: managed-nfs-storage
        volumeMode: Filesystem
```

Containerized virtual machines



Kubernetes resources

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

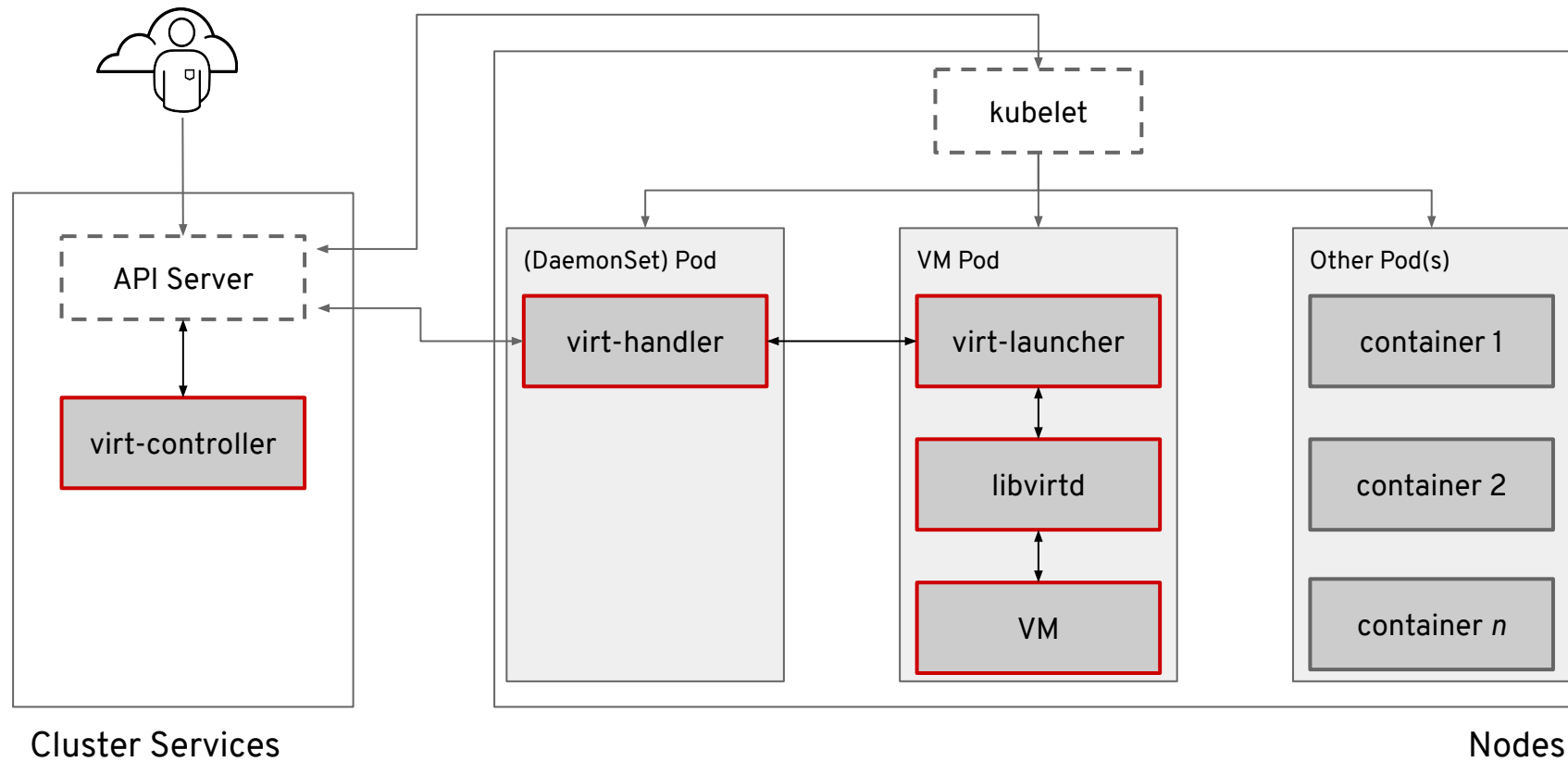
Red Hat Enterprise Linux

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

Security - Defense in depth

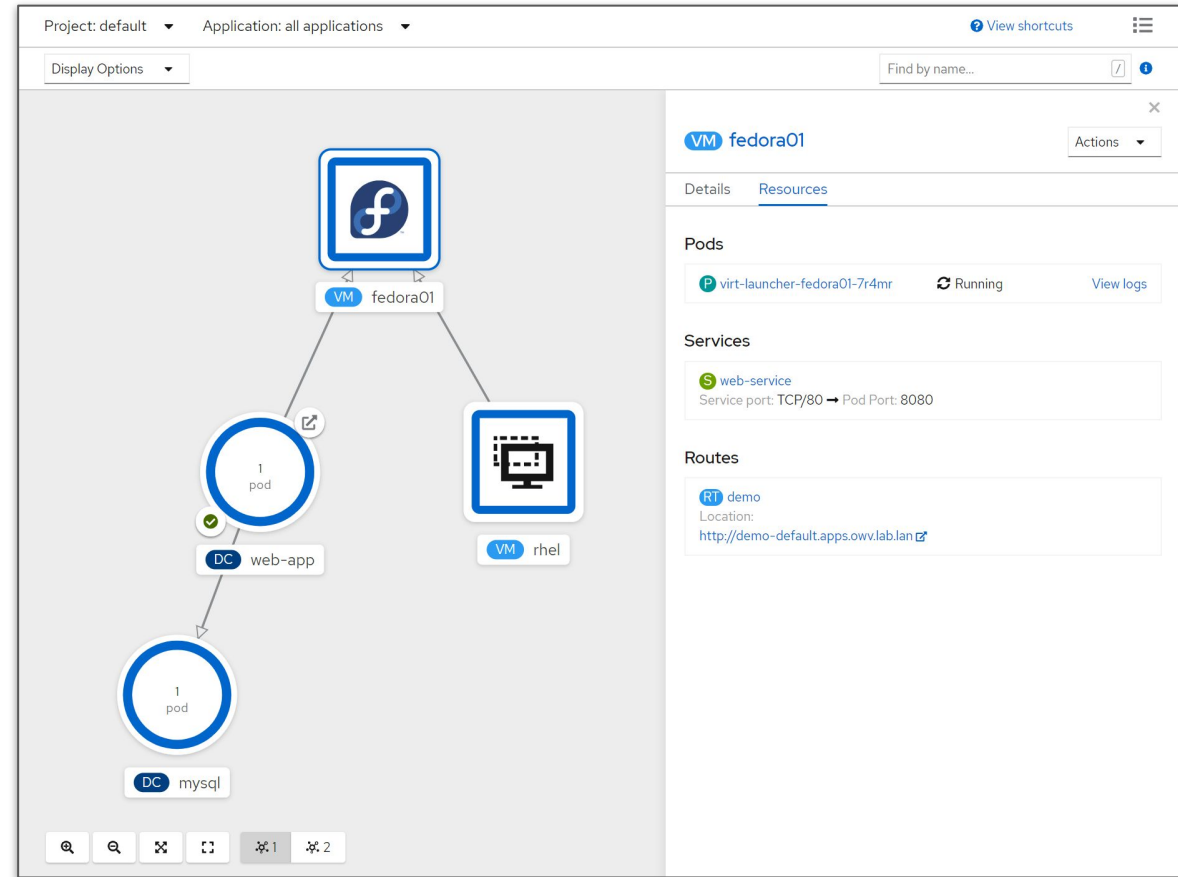
- Immutable RHCOS by default, SELinux MCS, plus KVM isolation - inherited from the Red Hat Portfolio stack

Architectural Overview



Using VMs and containers together

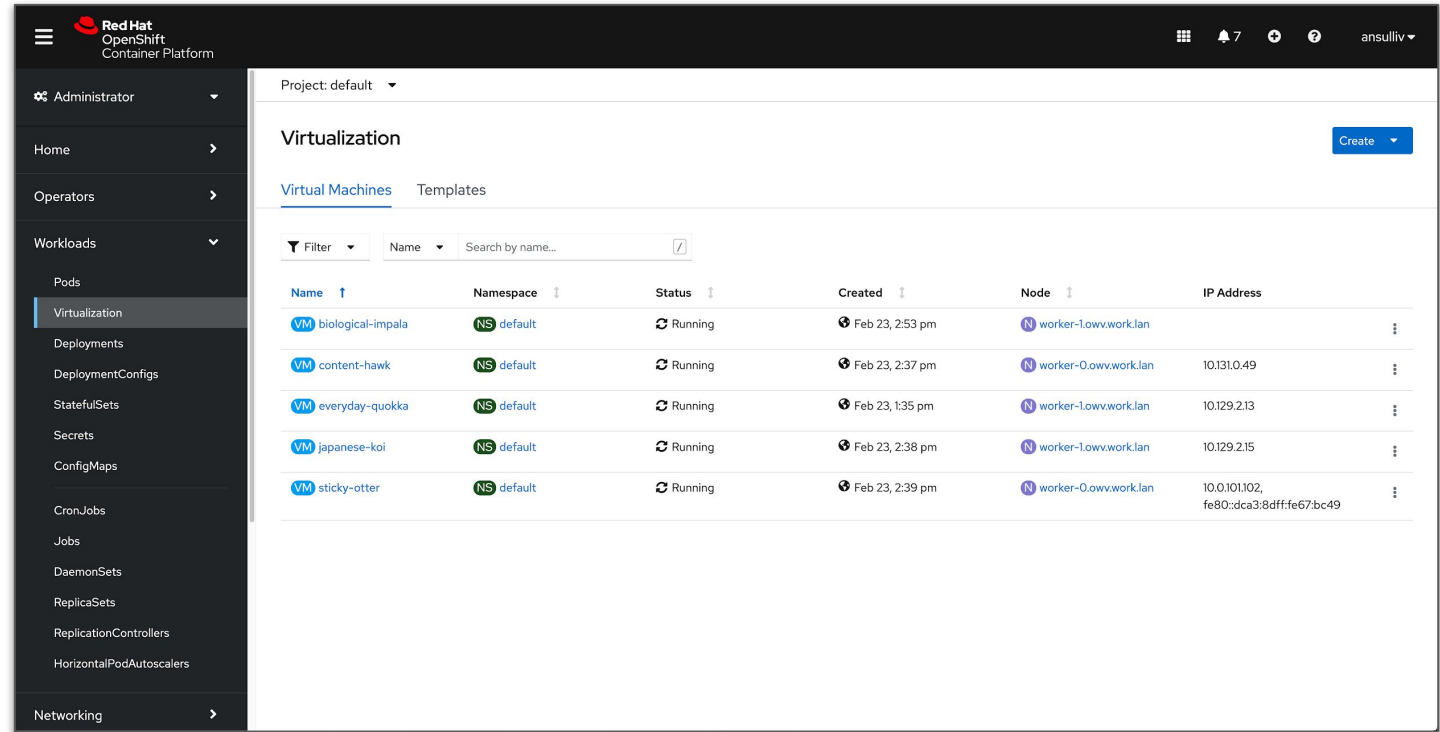
- Virtual machines connected to pod networks are accessible using standard Kubernetes methods:
 - Service
 - Route
 - Ingress
- Network policies apply to VM pods the same as application pods
- VM-to-pod, and vice-versa, communication happens over SDN or ingress depending on network connectivity



Managed VM with OpenShift

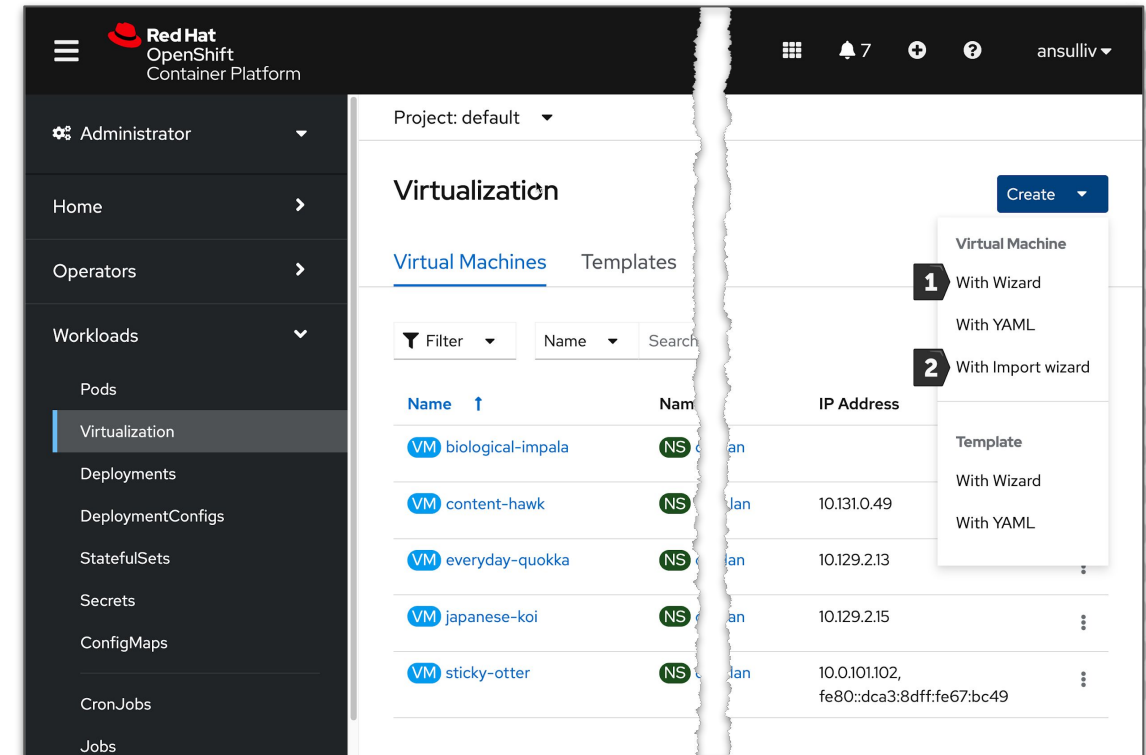
Virtual Machine Management

- Create, modify, and destroy virtual machines, and their resources, using the OpenShift web interface or CLI
- Use the `virtctl` command to simplify virtual machine interaction from the CLI



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
- Import VMs from VMware vSphere or Red Hat Virtualization



Using templates for virtual machines

- Simplified and streamlined virtual machine creation experience for VM consumers
- Administrators configure templates with an OS disk, consumers select from the options

Create Virtual Machine from template

1

Select template

2 Review and create

Select a template

The virtual machine can be **customized** in the next step.
Supported operating systems are labeled below. [Learn more about Red Hat support](#)
Templates that are in an **error** or **in-progress** state will not be shown.

PR default

All template providers


Boot source 1

Search by name, OS...

Boot source

Available

Clear all filters



Fedora 31+ VM

Provided by Red Hat

Fedora 32 or higher


Project openshift

Type server

Flavor Small: 1 CPU | 2 GiB Memory

Storage 20 GiB

Available boot source



Red Hat Enterprise Linux 8.0+ VM

Provided by Red Hat

Red Hat Enterprise Linux 8.0 or higher

Project openshift

Type server

Flavor Small: 1 CPU | 2 GiB Memory

Storage 20 GiB


Available boot source

3

Next

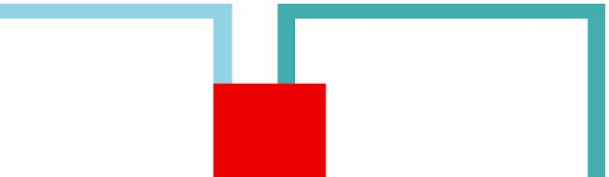
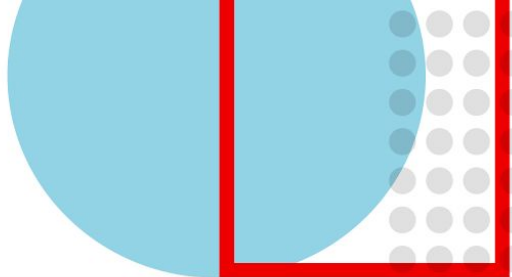
Back

Cancel



Create a template - General

- In addition to unique names, each template is associated with a provider
 - Providers represent who created the template, with optional support information
- The guest operating system and source boot disk are provided. A boot disk can be imported during the process, or an ISO can be used to boot and install the OS
- A default flavor, representing CPU and memory allotments, is assigned
- Workload type determines optimizations to balance between performance and efficiency



Create Virtual Machine template

1 General

2 Networking

3 Storage

4 Advanced

5 Review

6 Result

1

Name *

Template provider * ⓘ

example: your company name

Template support ⓘ

No additional support ▾

Description

2

Operating System *

--- Select Operating System --- ▾

3

Boot Source * ⓘ

--- Select Source --- ▾

4

Flavor * ⓘ

--- Select Flavor --- ▾

5

Workload Type * ⓘ

--- Select Workload Type --- ▾

Next

Review and confirm

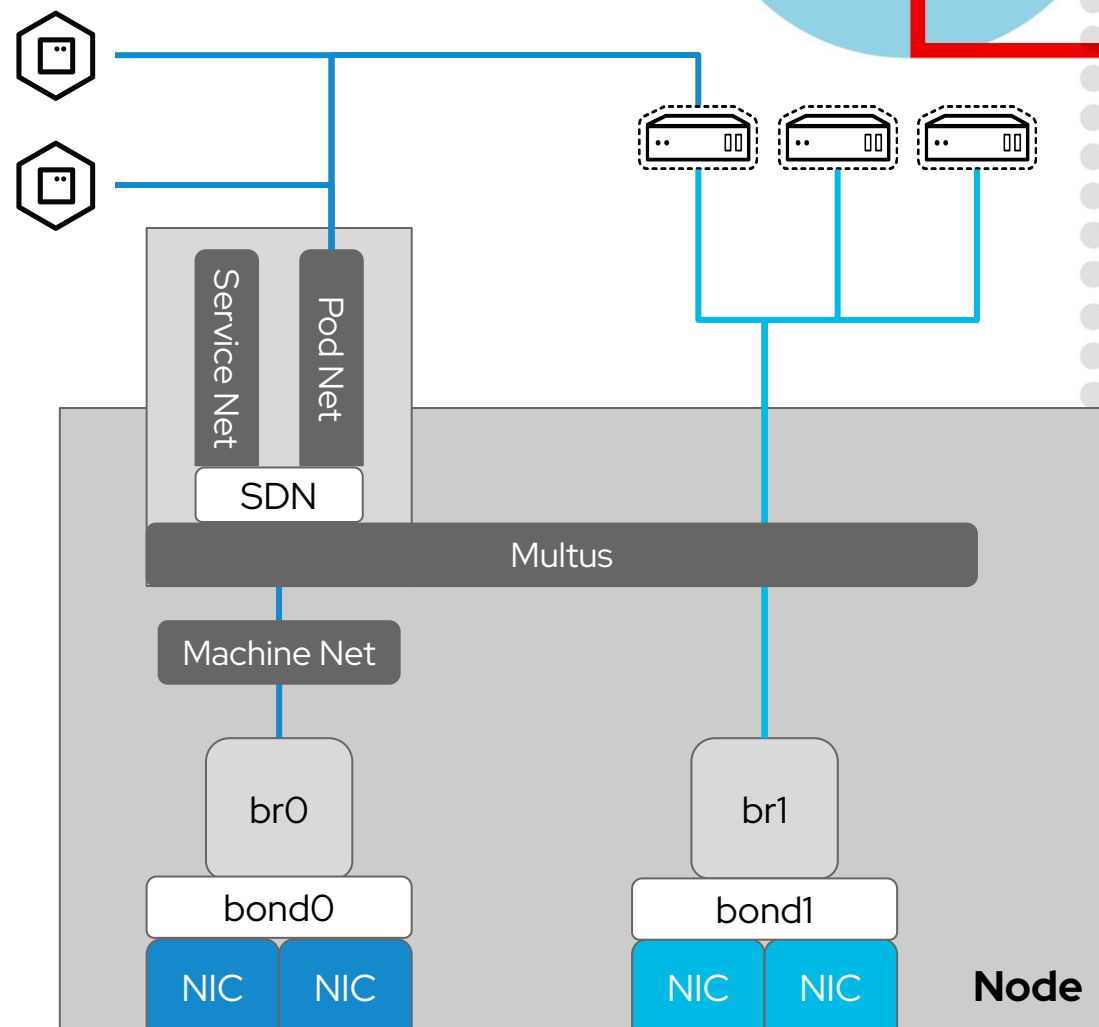
Back

Cancel

Example host network configuration

- Pod, service, and machine network are configured by OpenShift automatically
 - Use kernel parameters (dracut) for configuration at install - `bond0` in the example to the right
- Use `kubernetes-nmstate`, via the NMstate Operator, to configure additional host network interfaces
 - `bond1` and `br1` in the example to the right
- VM pods connect to one or more networks simultaneously

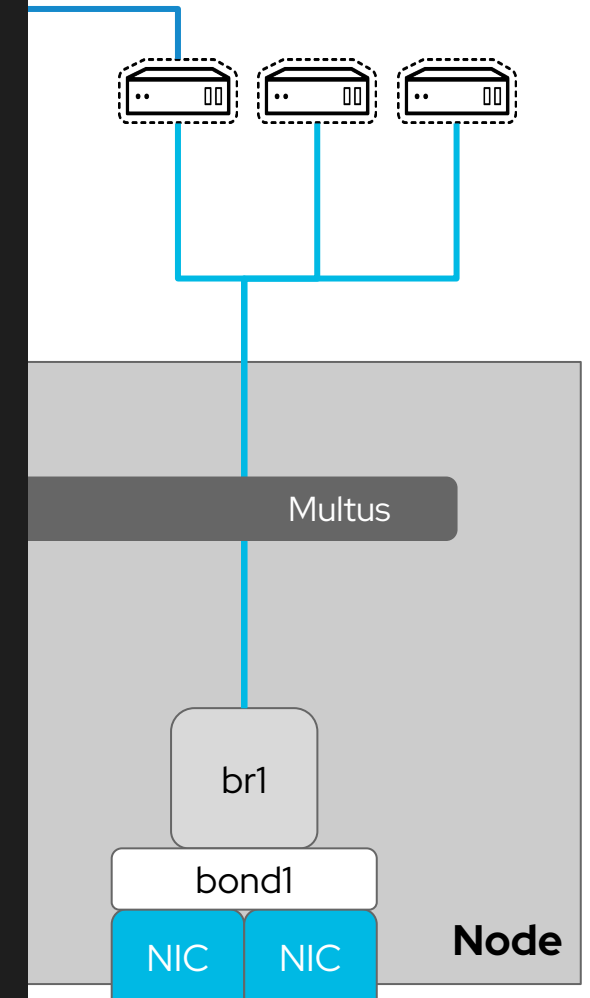
The following slides show an example of how this setup is configured



Host bond configuration

- NodeNetworkConfigurationPolicy (NNCP)
 - Nmstate operator CRD
 - Configure host network using declarative language
- Applies to all nodes specified in the nodeSelector, including newly added nodes automatically
- Update or add new NNCPs for additional host configs

```
1  apiVersion: nmstate.io/v1alpha1
2  kind: NodeNetworkConfigurationPolicy
3  metadata:
4    name: worker-bond1
5  spec:
6    nodeSelector:
7      node-role.kubernetes.io/worker: ""
8    desiredState:
9      interfaces:
10       - name: bond1
11         type: bond
12         state: up
13         ipv4:
14           enabled: false
15         link-aggregation:
16           mode: balance-alb
17           options:
18             miimon: '100'
19           slaves:
20             - eth2
21             - eth3
22           mtu: 1450
```



Connecting Pods to networks

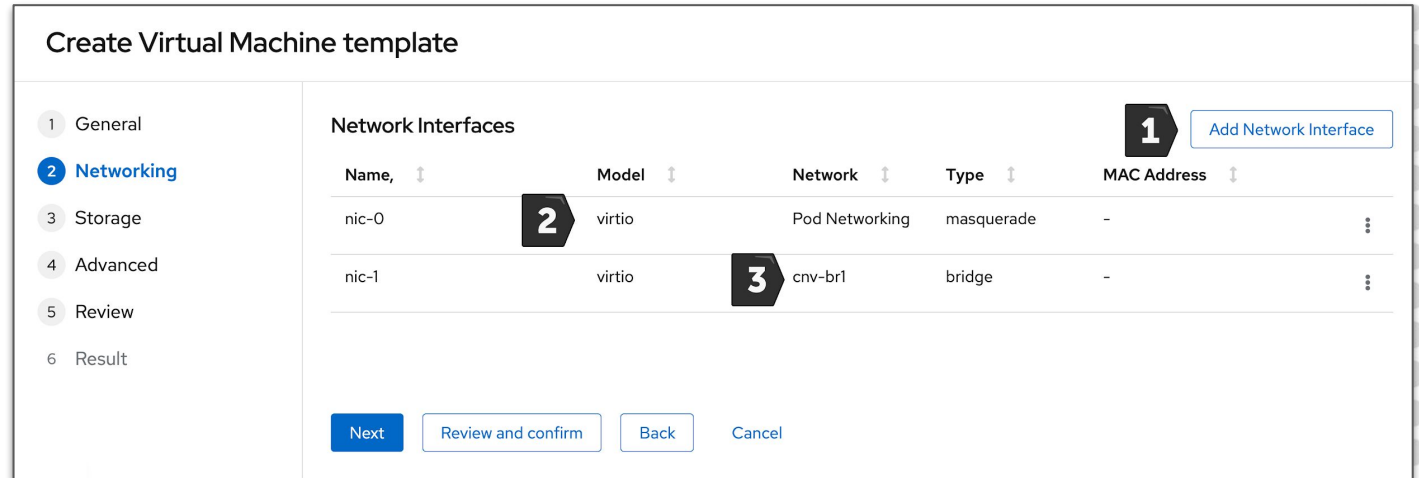
- Multus uses CNI network definitions in the NetworkAttachmentDefinition to allow access
 - `net-attach-def` are namespaced
 - Pods cannot connect to a `net-attach-def` in a different namespace
- `cnv-bridge` and `cnv-tuning` types are used to enable VM specific functions
 - MAC address customization
 - MTU and promiscuous mode
 - `sysctls`, if needed
- Pod connections are defined using an annotation
 - Pods can have many connections to many networks

```
1  apiVersion: k8s.cni.cncf.io/v1
2  kind: NetworkAttachmentDefinition
3  metadata:
4    name: br1-public
5    annotations:
6      k8s.v1.cni.cncf.io/resourceName: bridge.network.kubevirt.io/br1
7  spec:
8    config: '{
9      "cniVersion": "0.3.1",
10     "name": "br1-public",
11     "plugins": [
12       {
13         "type": "cnv-bridge",
14         "bridge": "br1"
15       },
16       {
17         "type": "cnv-tuning"
18       }
19     ]
20   }'
```

```
1  kind: Pod
2  apiVersion: v1
3  metadata:
4    name: application-pod
5    annotations:
6      k8s.v1.cni.cncf.io/networks: bond1-br1
```

Create a template - Networks

- Add or edit network adapters
- One or more network connections
 - Pod network for the default SDN
 - Additional multus-based interfaces for specific connectivity
- Multiple NIC models for guest OS compatibility or paravirtualized performance with VirtIO
- Masquerade, bridge, or SR-IOV connection types
- MAC address customization if desired



Create Virtual Machine template

1 General
2 **Networking**
3 Storage
4 Advanced
5 Review
6 Result

Network Interfaces

Name, ↑	Model ↑	Network ↑	Type ↑	MAC Address ↑
nic-0	virtio	Pod Networking	masquerade	-
nic-1	virtio	cnv-br1	bridge	-

1 Add Network Interface

2

3

Next Review and confirm Back Cancel

Virtual Machine Storage

- OpenShift Virtualization uses the Kubernetes PersistentVolume (PV) paradigm
- PVs can be backed by
 - In-tree iSCSI, NFS, etc.
 - CSI drivers
 - Local storage using host path provisioner
 - OpenShift Container Storage
- Use dynamically or statically provisioned PVs
- RWX is required for live migration
- Disks are attached using VirtIO or SCSI controllers
 - Connection order specified in the VM definition
- Boot order customized via VM definition

PersistentVolumeClaim Details

Name

rhel-rootdisk

Namespace

 default

Labels

 app=containerized-data-importer


Annotations

[12 Annotations](#) 

Label Selector

No selector

Created At

 Jul 8, 4:18 pm

Owner

 rhel-rootdisk

Status

 Bound

Capacity

20Gi

Access Modes

ReadWriteMany

Volume Mode

Filesystem

Storage Class

 managed-nfs-storage

Persistent Volume

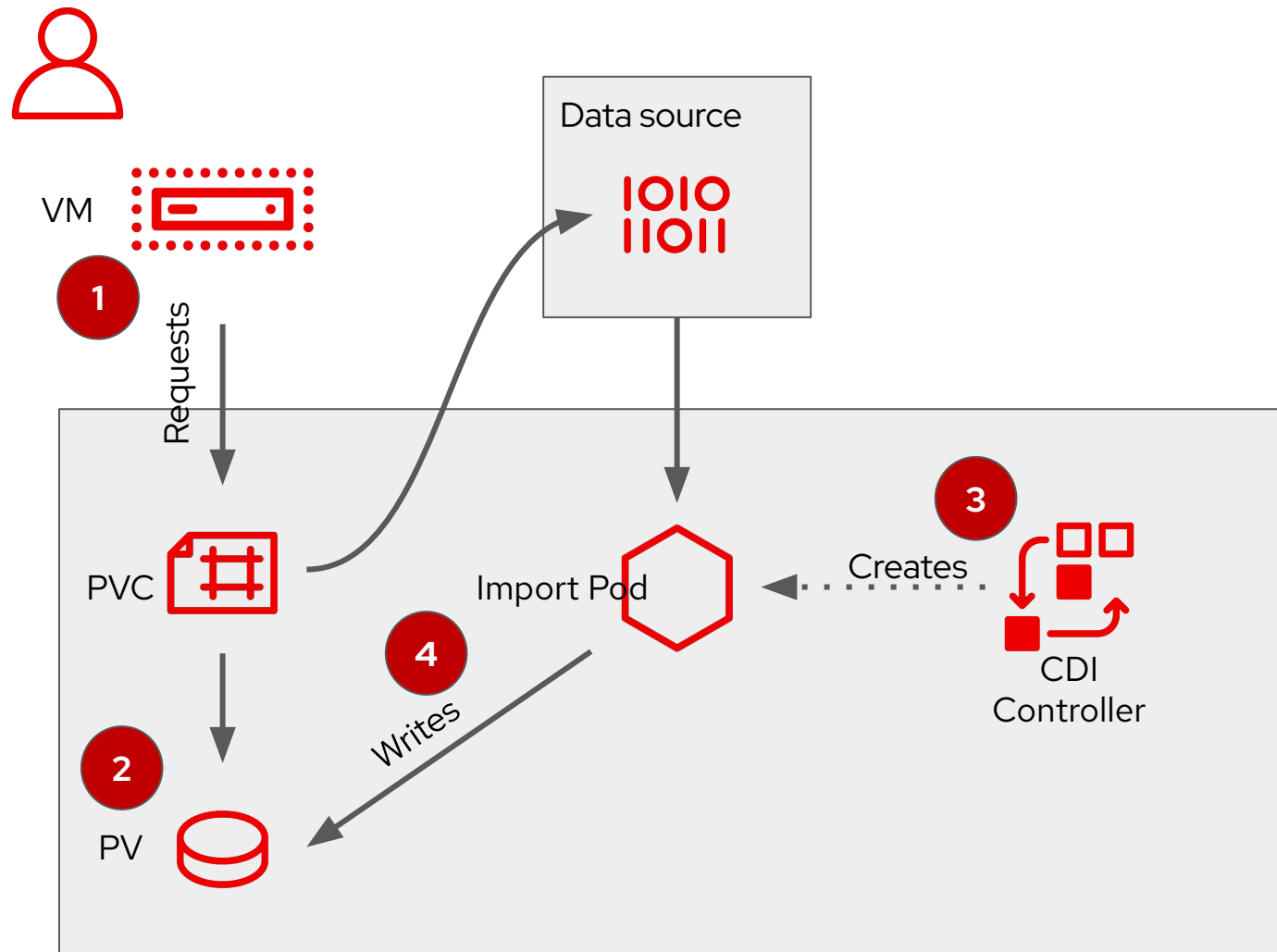
 pvc-alaac411-2e46-495a-897e-cf3bc2442199

DataVolumes

- VM disks can be imported from multiple sources using DataVolumes, e.g. an HTTP(S) or S3 URL for a QCOW2 or raw disk image, optionally compressed
- VM disks can be cloned / copied from existing PVCs
- DataVolumes are created as distinct objects or as a part of the VM definition as a `dataVolumeTemplate`
- DataVolumes use the `ContainerizedDataImporter` to connect, download, and prepare the disk image
- DataVolumes create PVCs based on defaults defined in the `kubevirt-storage-class-defaults` ConfigMap or according to the profile (as of version 4.8)

```
1  dataVolumeTemplates:
2    - apiVersion: cdi.kubevirt.io/v1alpha1
3      kind: DataVolume
4      metadata:
5        creationTimestamp: null
6        name: vm-rootdisk
7      spec:
8        pvc:
9          accessModes:
10            - ReadWriteMany
11          resources:
12            requests:
13              storage: 20Gi
14          storageClassName: my-storage-class
15          volumeMode: Filesystem
16        source:
17          http:
18            url: 'http://web.server/disk-image.qcow2'
```

Containerized Data Importer



1. The user creates a virtual machine with a DataVolume
2. The StorageClass is used to satisfy the PVC request
3. The CDI controller creates an importer pod, which mounts the PVC and retrieves the disk image. The image could be sourced from S3, HTTP, or other accessible locations
4. After completing the import, the import pod is destroyed and the PVC is available for the VM

Create a template - Storage

- Add or edit persistent storage
- Disks can be sourced from
 - Imported QCOW2 or raw images
 - New or existing PVCs
 - Clone existing PVCs
- Use SATA/SCSI interface for compatibility or VirtIO for paravirtual performance
- For new or cloned disks, select from available storage classes
 - Customize volume and access mode as needed
 - RWX PVCs are required for live migration

Create Virtual Machine template

1 General
2 Networking
3 **Storage**
4 Advanced
5 Review
6 Result

Disks

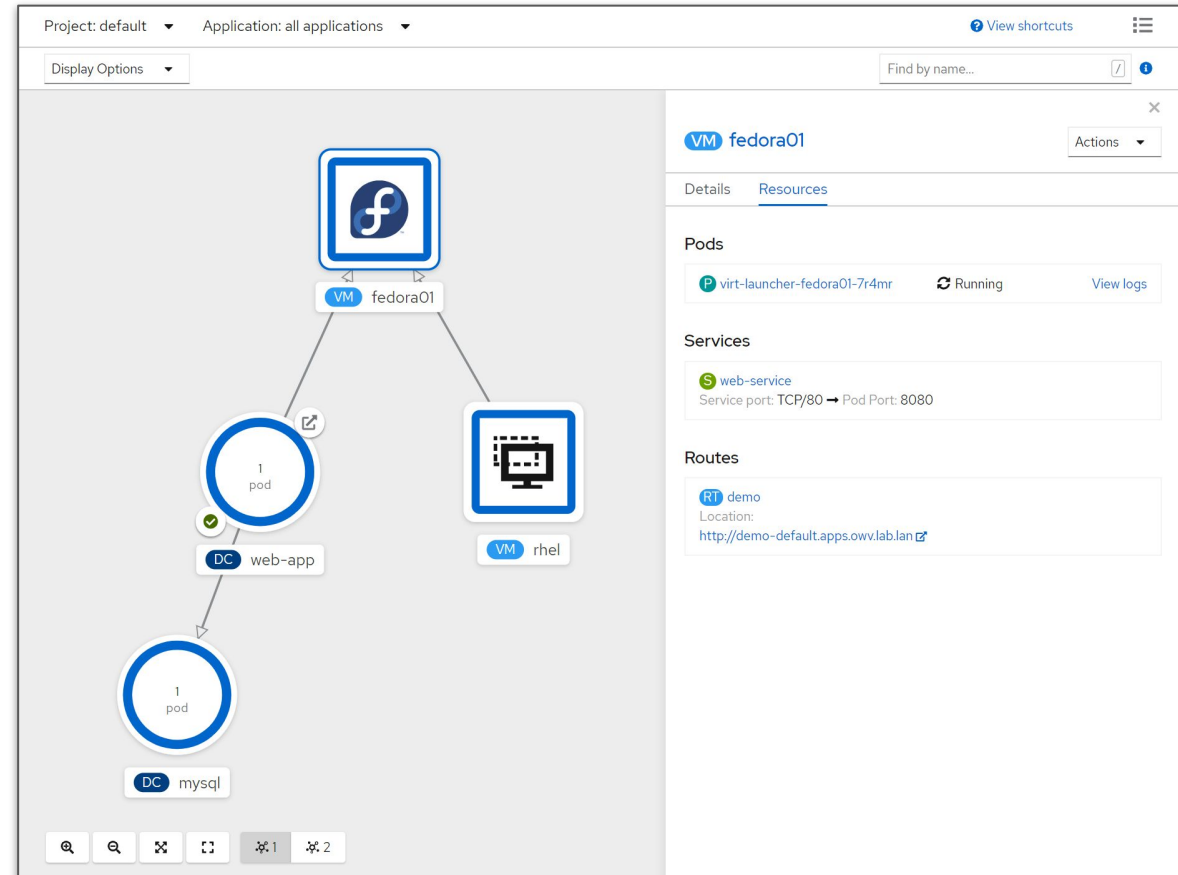
Name	Source	Size	Drive	Interface	Storage Class
cloudinitdisk	Other	-	Disk	virtio	-
rootdisk	pvc	20 GiB	Disk	virtio	lab-silver

Boot Source
rootdisk

Next Review and confirm Back Cancel

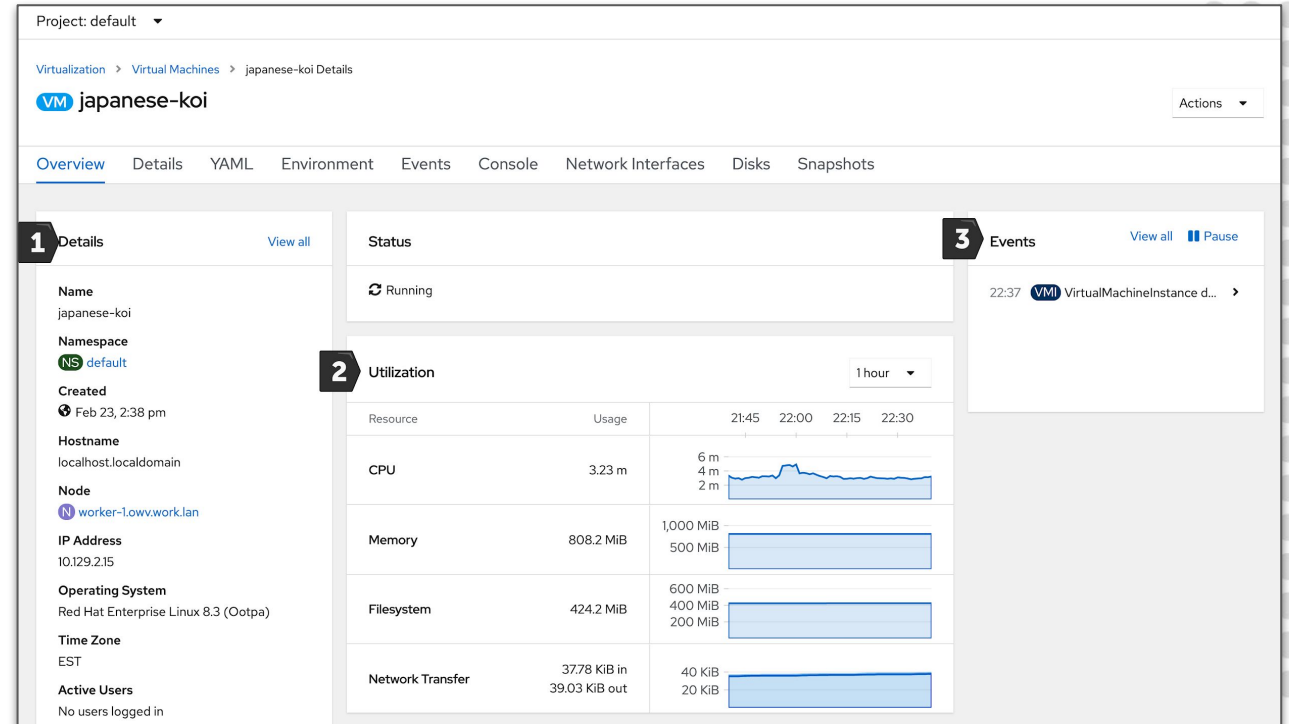
Using VMs and containers together

- Virtual machines connected to pod networks are accessible using standard Kubernetes methods:
 - Service
 - Route
 - Ingress
- 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



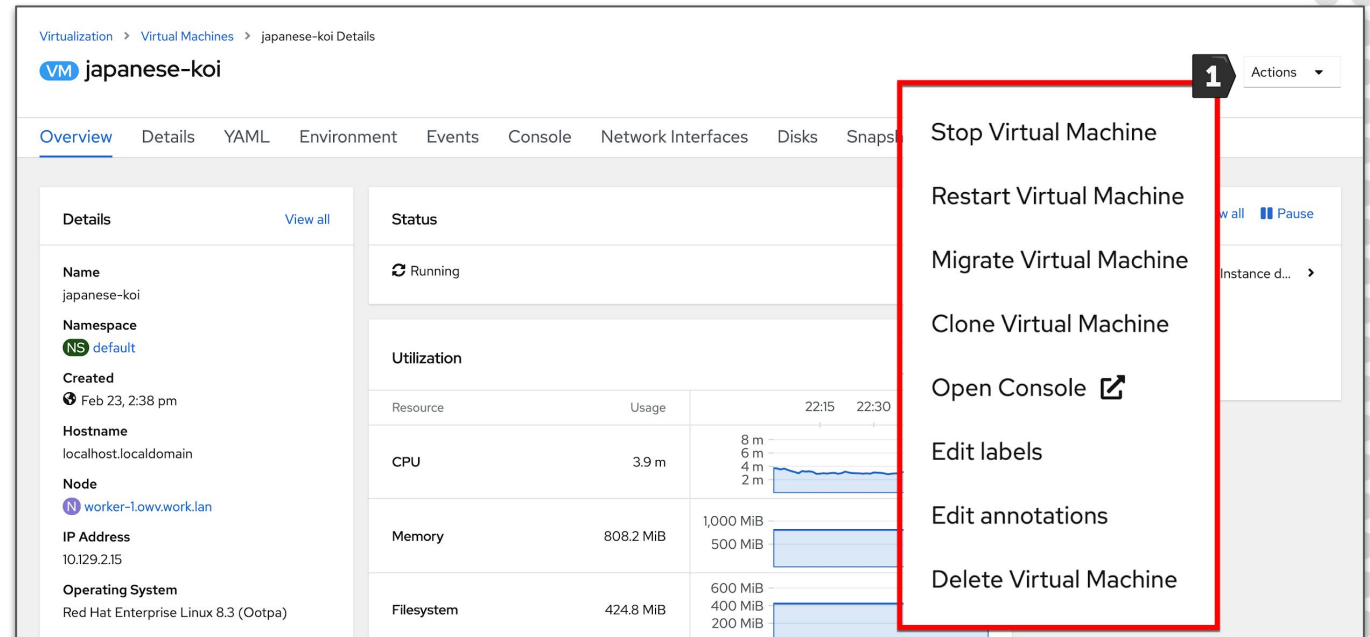
Virtual Machine - Overview

- 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



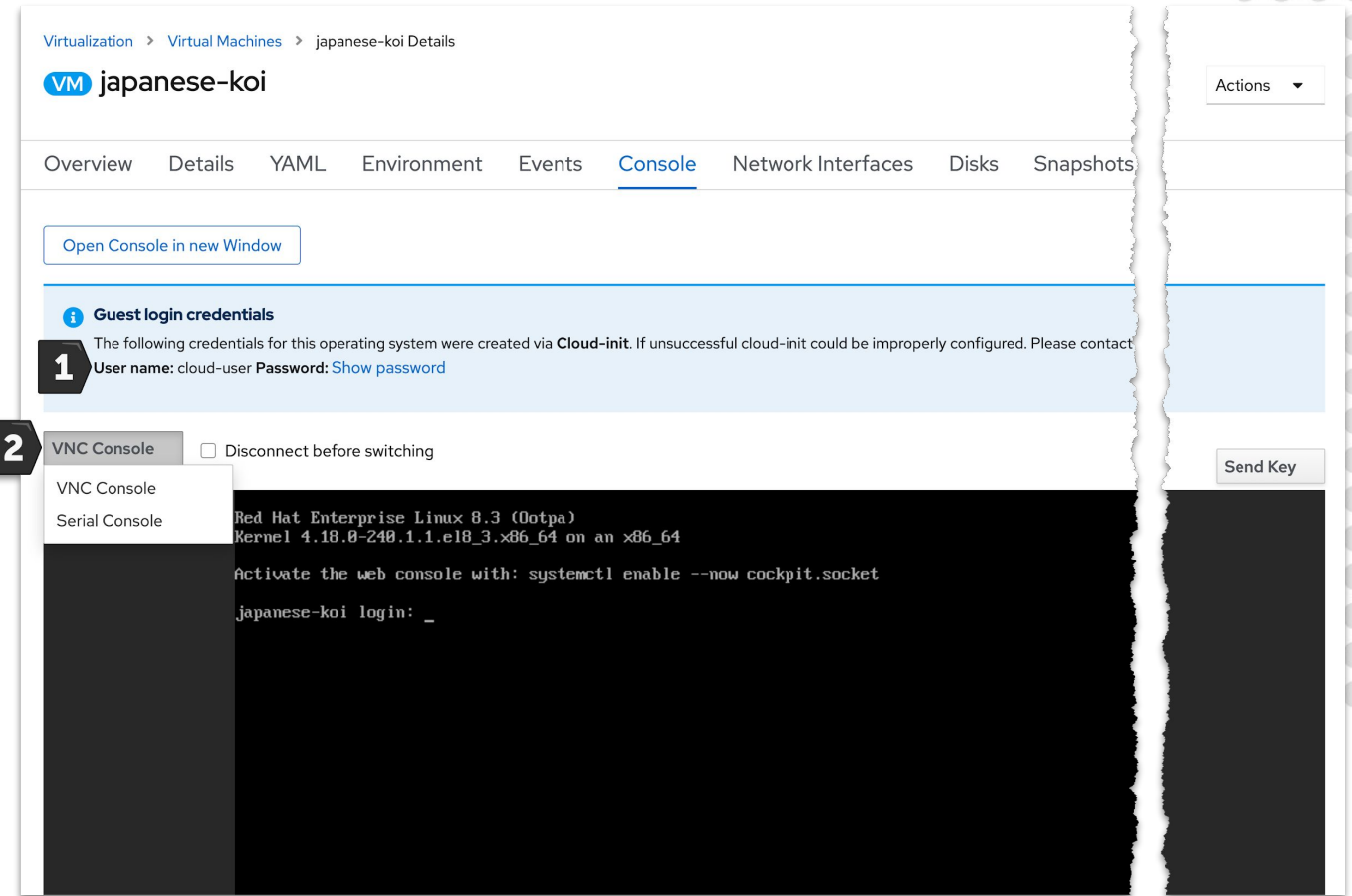
Virtual Machine - Actions

- 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 VM list



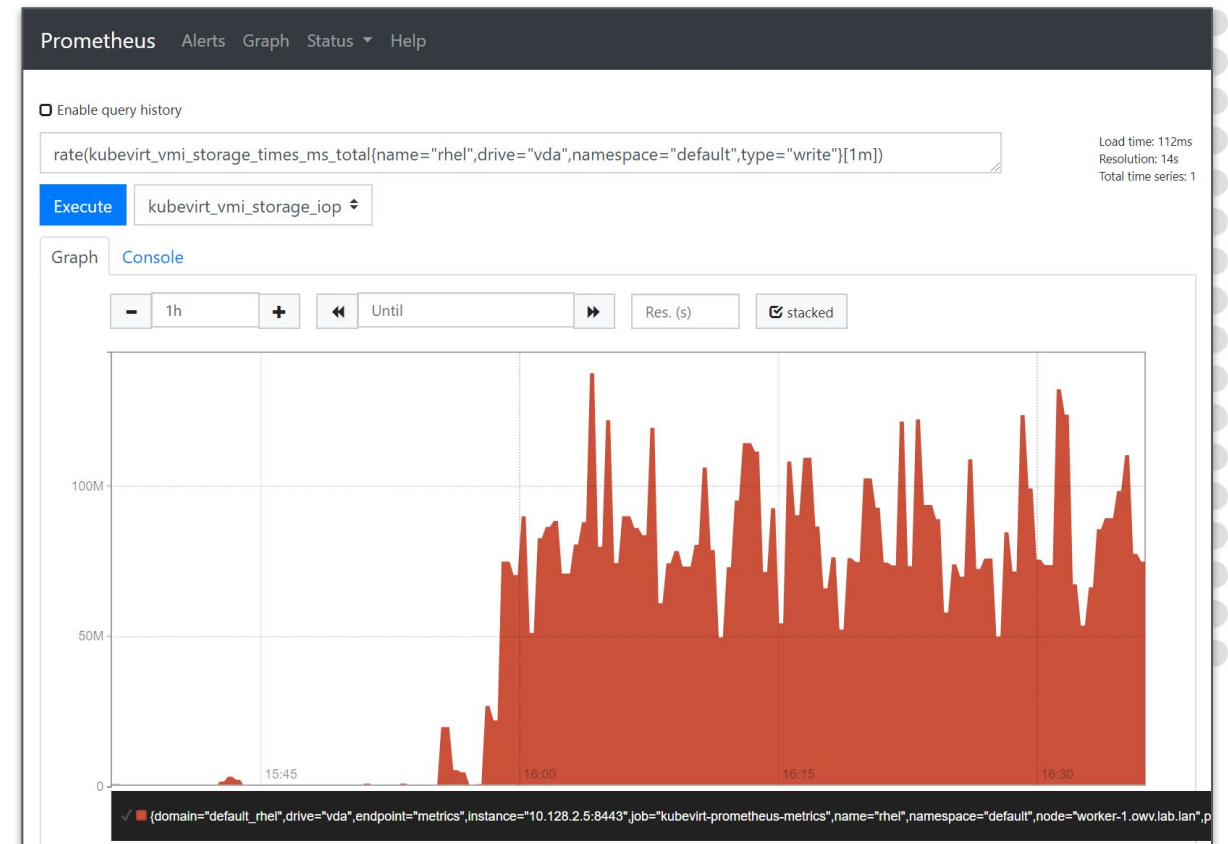
Virtual Machine - Console

- Browser-based access to the serial and graphical console of the virtual machine
- Access the console using native OS tools, e.g. `virt-viewer`, using the `virtctl` CLI command
 - `virtctl console vmname`
 - `virtctl vnc vmname`



Detailed Virtual Machine metrics

- Virtual machine, and VM pod, metrics are collected by the OpenShift metrics service
 - Available under the `kubevirt` namespace in Prometheus
- Available per-VM metrics include
 - Active memory
 - Active CPU time
 - Network in/out errors, packets, and bytes
 - Storage R/W IOPS, latency, and throughput
- VM metrics are for VMs, not for VM pods
 - Management overhead not included in output
 - Look at virt-launcher pod metrics for
- No preexisting Grafana dashboards



VM migration

Migration Toolkit for Virtualization (MTV)

Create Migration Plan

1 General

2 VM selection

3 Storage mapping

4 Network mapping

5 Hooks

6 Review

Select VMs

Select VMs for migration. The Migration analysis column shows the risk associated with migrating a VM as determined by Red Hat's Migration Analytics service. The Flags indicate the reason for that risk assessment.

Name v Filter by name...

Name v ↓ A Z

1-3 of 3 << < 1 of 1 > >>

	Migration analysis	VM name	Datacenter	Cluster	Host	Folder path
▶ <input type="checkbox"/>	⚠	VM1	datacenter1	cluster1	host1	folder1\folder2
▶ <input type="checkbox"/>	✓	VM2	datacenter1	cluster1	host1	folder1\folder2
▶ <input type="checkbox"/>	i	VM3	datacenter1	cluster1	host1	folder1\folder2
▶ <input type="checkbox"/>	✓	VM4	datacenter1	cluster1	host1	folder1\folder2
▼ <input type="checkbox"/>	!	VM5	datacenter1	cluster1	host1	folder1\folder2

This VM is a **high risk** for migration because it violates the following rules:

- VM shares a disk with other VMs
- VM uses remote device management
- VM was harvested during a month without an "r" in it


Migration Analytics

Detect potential compatibility issues before migrating to ensure a successful migration

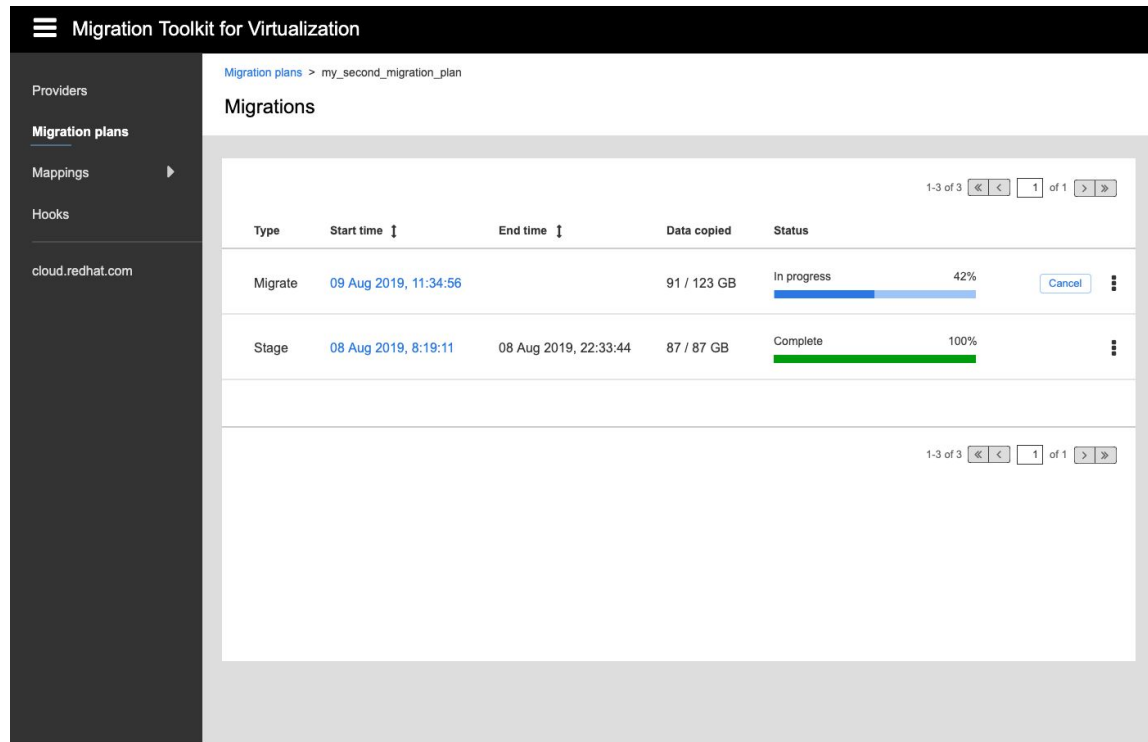
Mass Migration of VMs

Migrate workloads at scale to OpenShift

- Provide source and destination credentials
- Map infrastructure
- Create migration plans



Migration Toolkit for Virtualization (MTV)



Migration Analytics

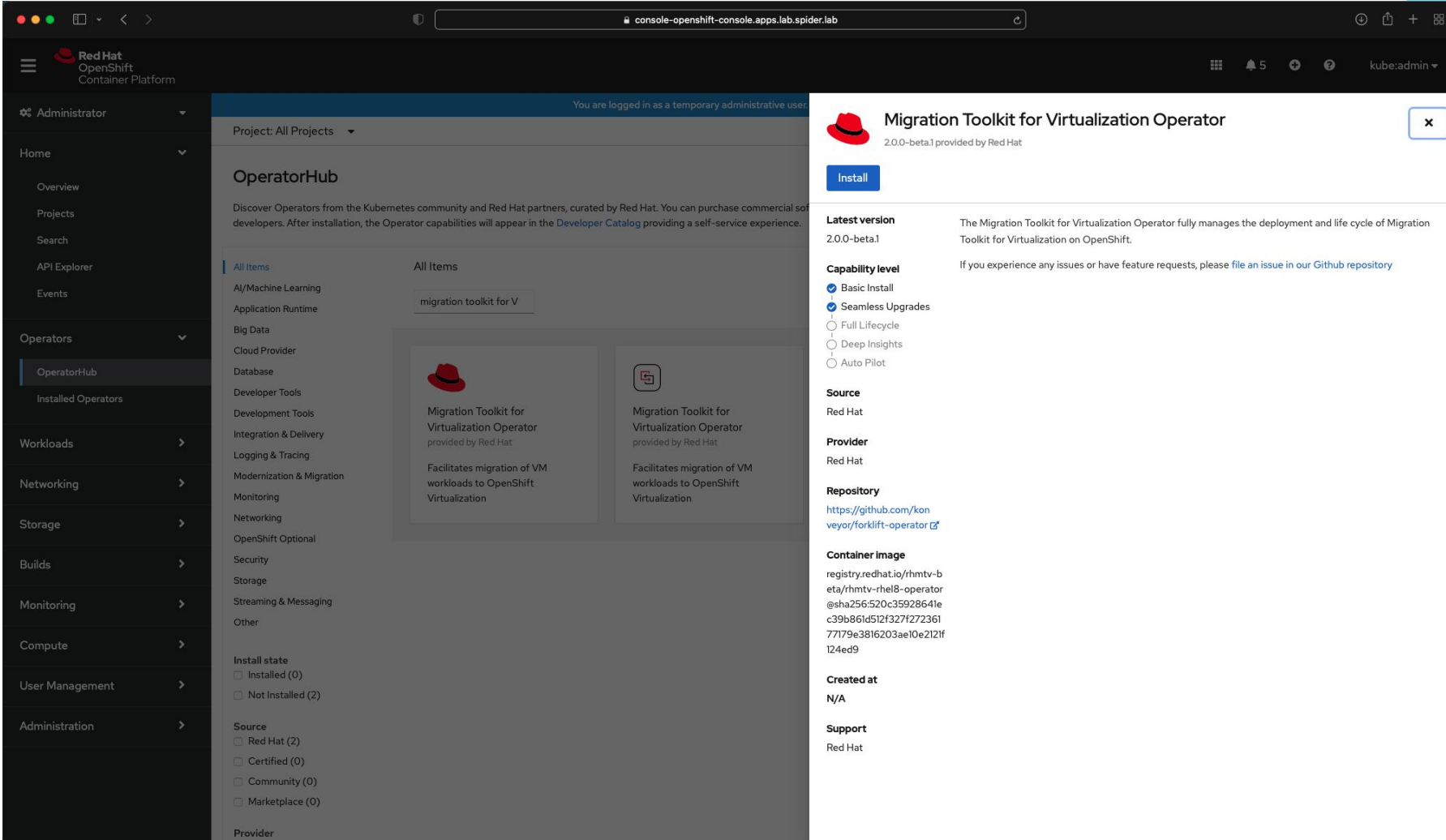
Detect potential compatibility issues before migrating to ensure a successful migration

Mass Migration of VMs

Migrate workloads at scale to OpenShift

- Provide source and destination credentials
- Map infrastructure
- Create migration plans

How to install?



The screenshot displays the Red Hat OpenShift Container Platform console. The left sidebar shows the navigation menu with categories like Administrator, Home, Operators, Workloads, Networking, Storage, Builds, Monitoring, Compute, User Management, and Administration. The main content area is titled "OperatorHub" and shows a list of operators. The "Migration Toolkit for Virtualization Operator" is highlighted, and its details are shown on the right. The details include the latest version (2.0.0-beta.1), capability level (Basic Install, Seamless Upgrades), source (Red Hat), provider (Red Hat), repository (https://github.com/konveyor/forklift-operator), container image (registry.redhat.io/rhmtv-beta/rhmtv-rhel8-operator), and support (Red Hat).

Red Hat OpenShift Container Platform

Administrator

Home

Overview

Projects

Search

API Explorer

Events

Operators

OperatorHub

Installed Operators

Workloads

Networking

Storage

Builds

Monitoring

Compute

User Management

Administration

You are logged in as a temporary administrative user

Project: All Projects

OperatorHub

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase commercial software from Red Hat partners. After installation, the Operator capabilities will appear in the [Developer Catalog](#) providing a self-service experience.

All Items

AI/Machine Learning

Application Runtime

Big Data

Cloud Provider

Database

Developer Tools

Development Tools

Integration & Delivery

Logging & Tracing

Modernization & Migration

Monitoring

Networking

OpenShift Optional

Security

Storage

Streaming & Messaging

Other

Install state

☐ Installed (0)

☐ Not Installed (2)

Source

☐ Red Hat (2)

☐ Certified (0)

☐ Community (0)

☐ Marketplace (0)

Provider

migration toolkit for V

Migration Toolkit for Virtualization Operator provided by Red Hat

Facilitates migration of VM workloads to OpenShift Virtualization

Migration Toolkit for Virtualization Operator provided by Red Hat

Facilitates migration of VM workloads to OpenShift Virtualization

Migration Toolkit for Virtualization Operator

2.0.0-beta.1 provided by Red Hat

Install

Latest version

2.0.0-beta.1

Capability level

☒ Basic Install

☒ Seamless Upgrades

☐ Full Lifecycle

☐ Deep Insights

☐ Auto Pilot

Source

Red Hat

Provider

Red Hat

Repository

<https://github.com/konveyor/forklift-operator>

Container image

registry.redhat.io/rhmtv-beta/rhmtv-rhel8-operator@sha256:520c35928641ec39b861d512f327f27236177179e3816203ae10e2121f124ed9

Created at

N/A

Support

Red Hat

The Migration Toolkit for Virtualization Operator fully manages the deployment and life cycle of Migration Toolkit for Virtualization on OpenShift.

If you experience any issues or have feature requests, please [file an issue in our Github repository](#)

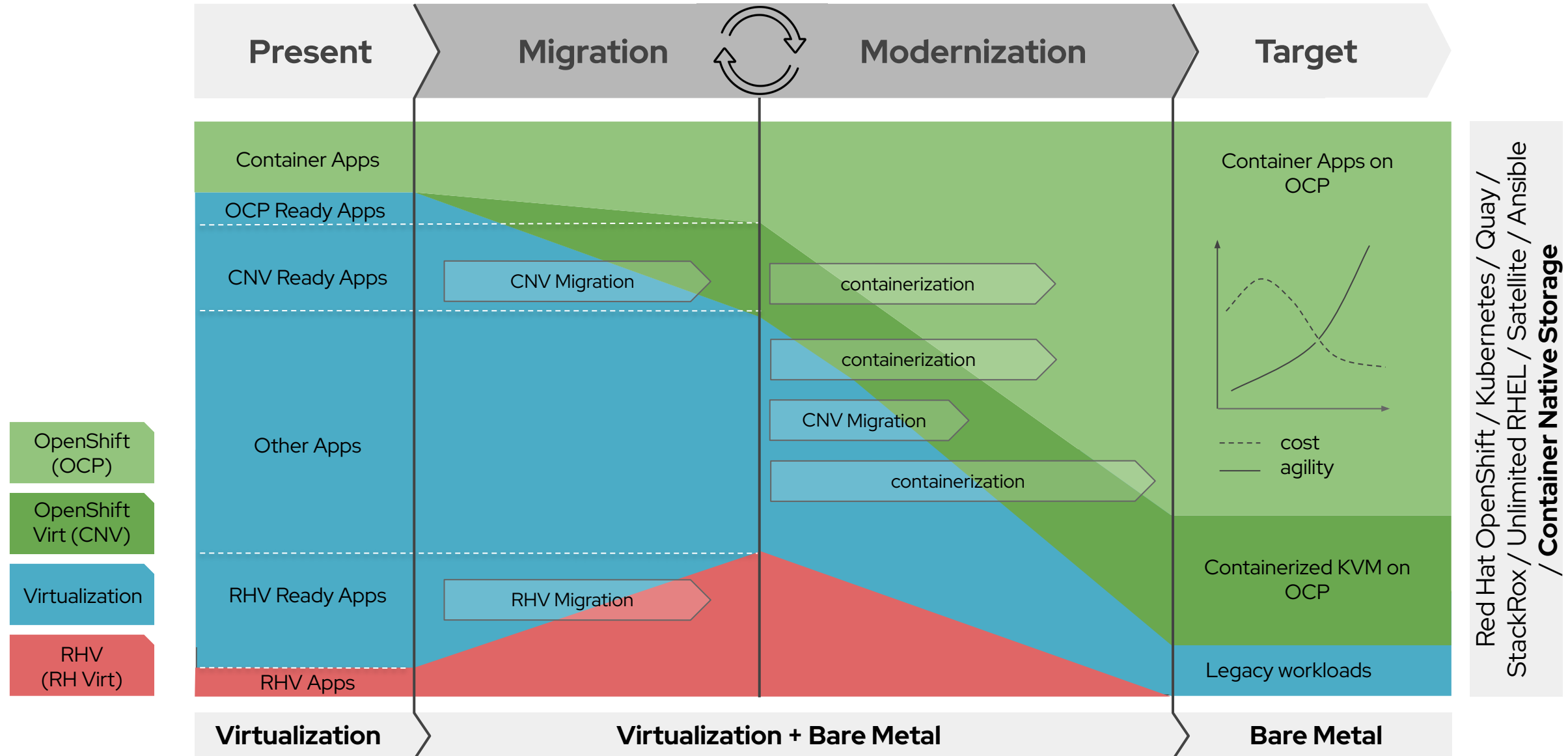
OpenShift Virtualization conditions

Supported configuration

- VM nodes should be physical with CPU virtualization technology enabled in the BIOS
 - Nested virtualization *works*, but ***is not supported***
 - Emulation *works*, but ***is not supported*** (and is extremely slow)

Summary

Apps: Software Infrastructure landscape



Red Hat
Summit

THANK
YOU



linkedin.com/company/Red-Hat



facebook.com/RedHatinc



youtube.com/user/RedHatVideos



twitter.com/RedHat