Red Hat Summit

Containers and VMs together

Openshift Virtualization

Artur Pająk
Solution Architect
RH CEE

Red Hat

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.

Red Hat

Virtualization continues to dominate

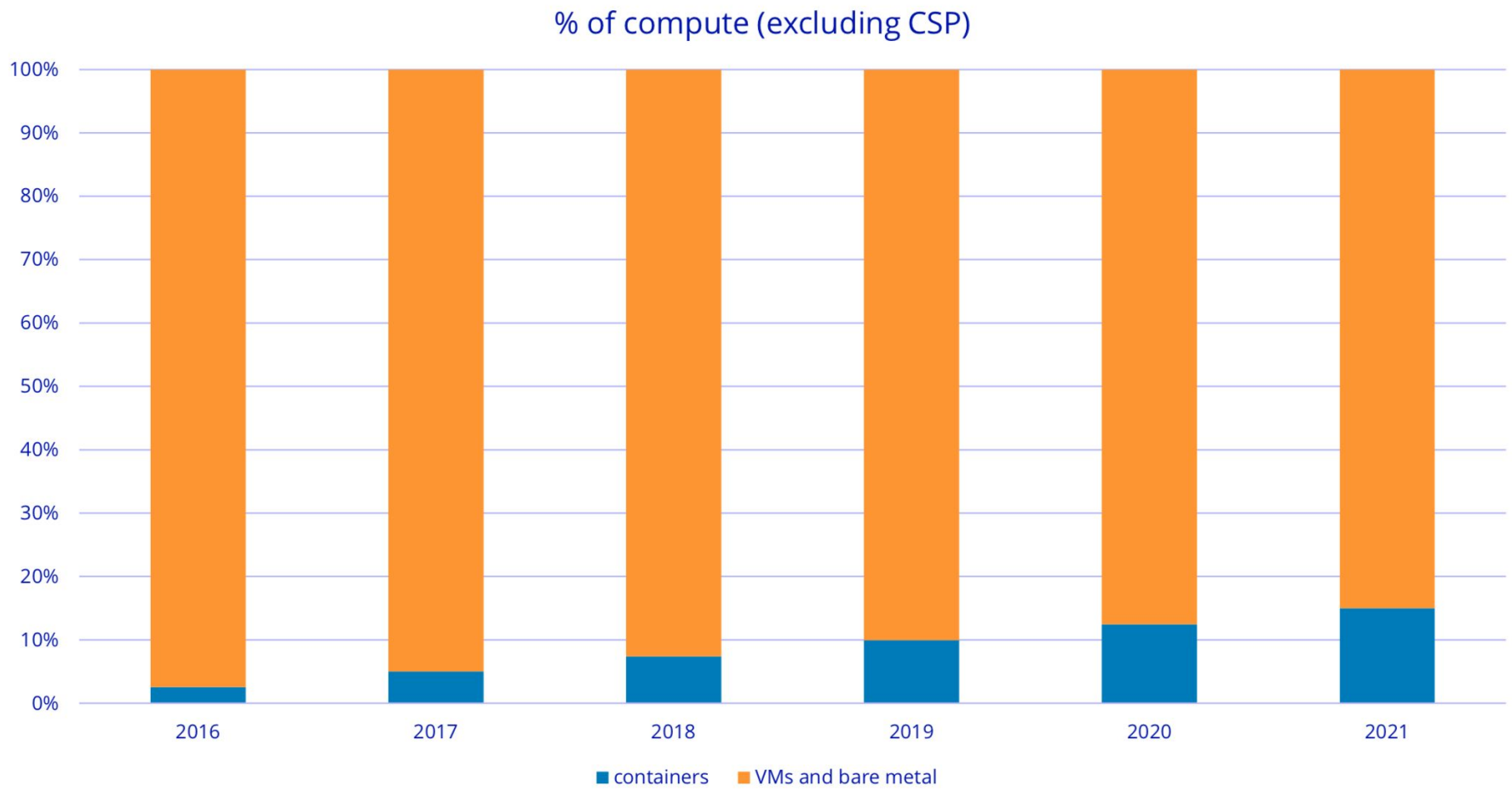


Image source: IDC Container Forecast, 2018

Red Hat

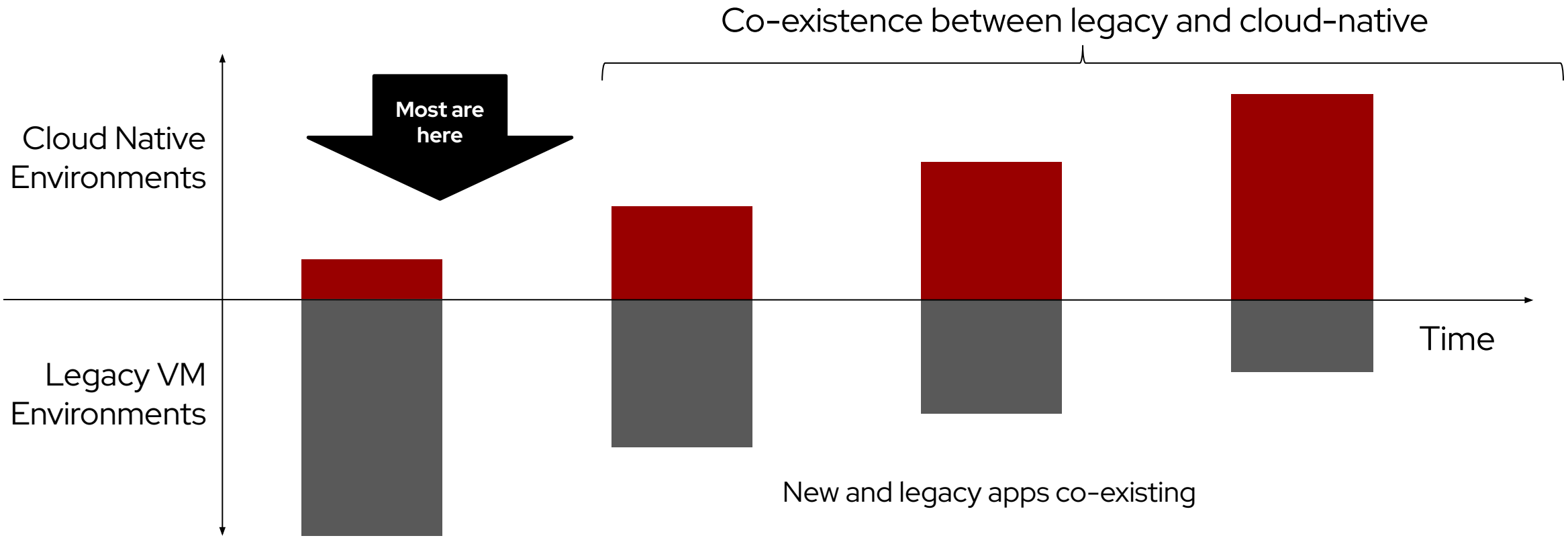
The move from Virtual Machines to Containers

Accelerate Digital Transformations

Improve Developer Productivity

Increase Operational Efficiency and Standardization

Traditional virtualization cannot support these cloud-native applications.

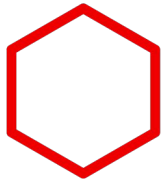


Red Hat

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.

Red Hat

The Paths to Application Modernization

Red Hat

The paths to Application Modernization

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

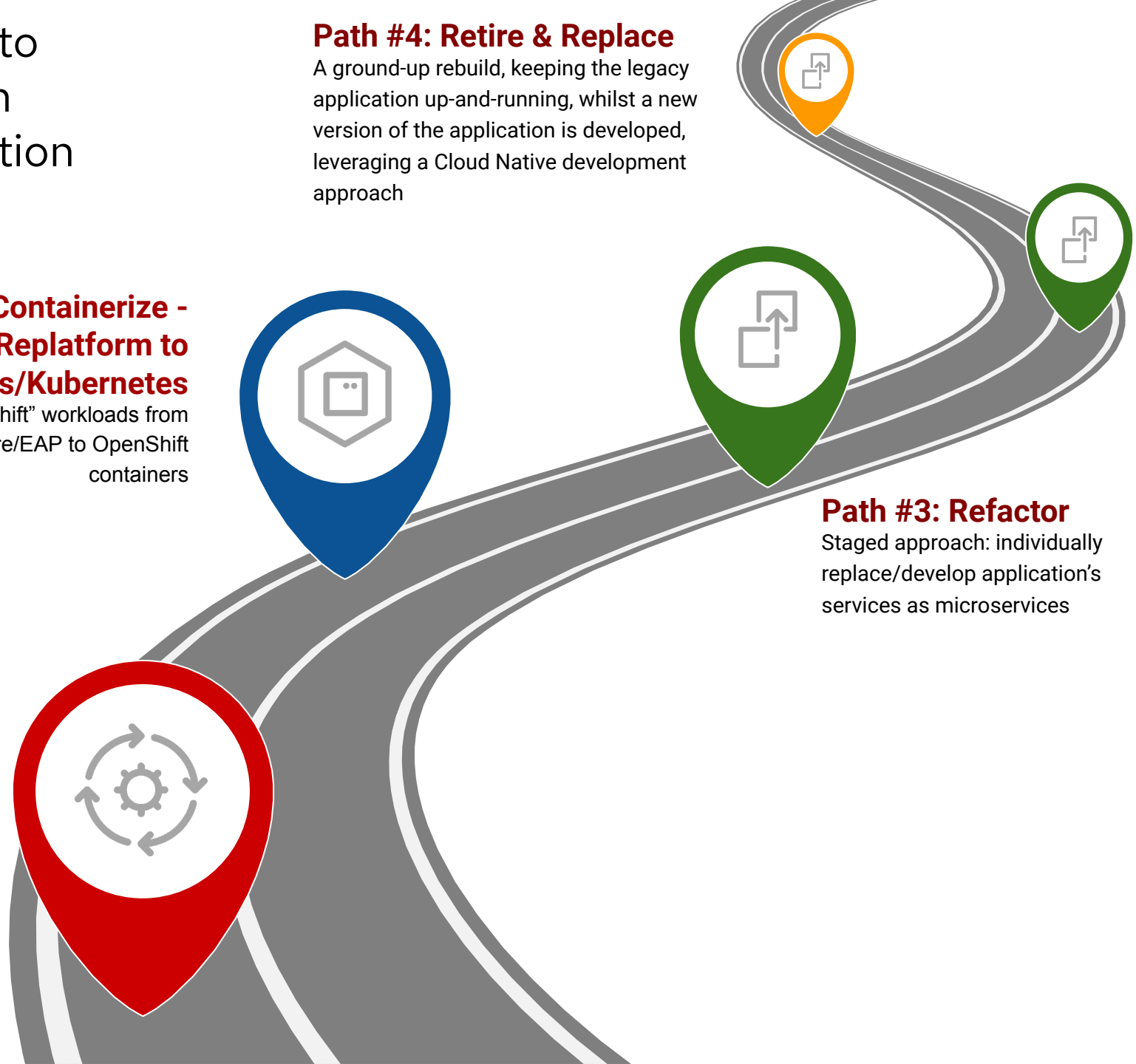
Staged approach: individually replace/develop application's services as microservices, incorporating more advanced capabilities into applications, such as AI/ML and event driven approaches

Path #2: Containerize - Replatform to Containers/Kubernetes

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

Path #3: Refactor

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



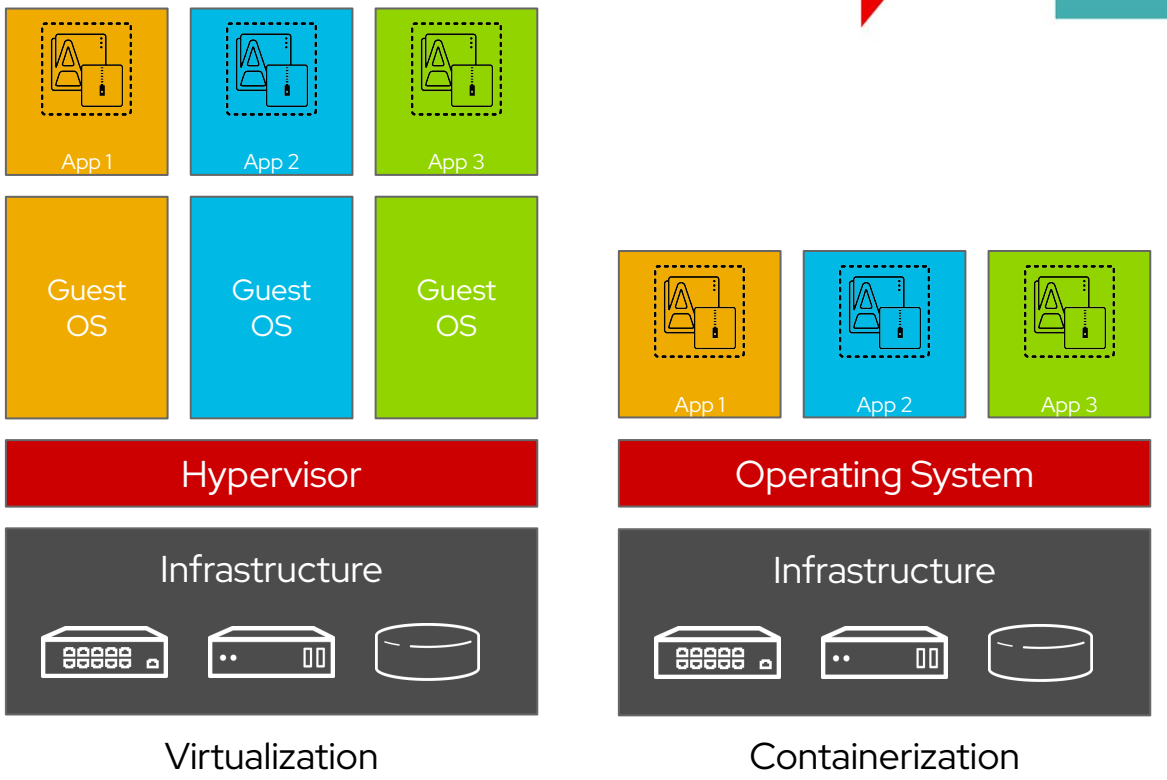
Path #1: Virtualize - Rehost Application Server to OpenShift

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

Red Hat

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



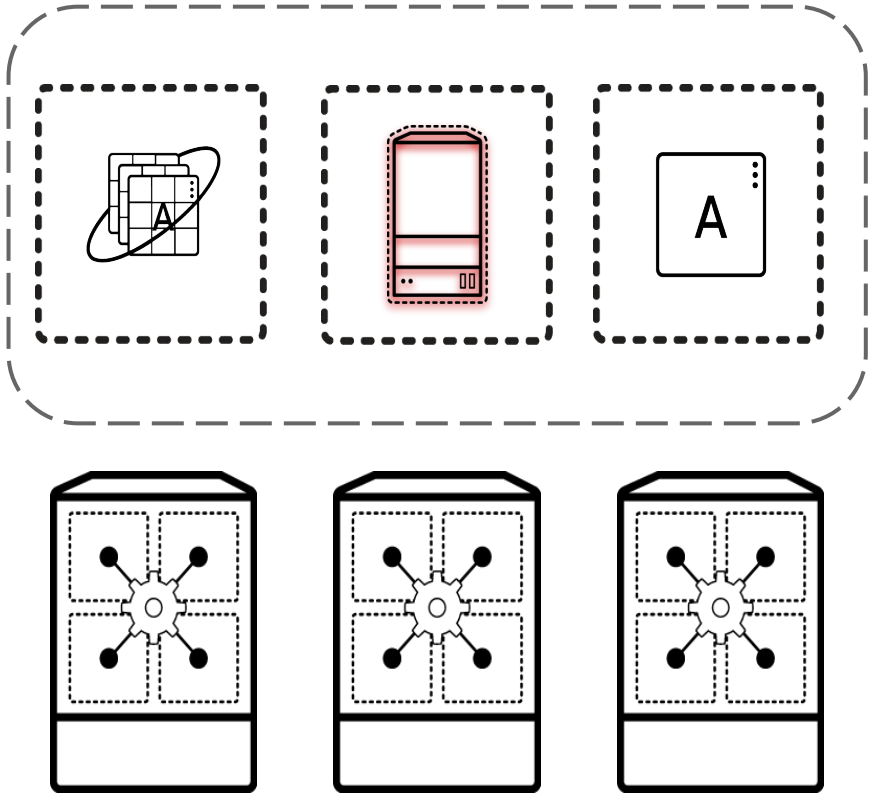
Virtualization

Containerization

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

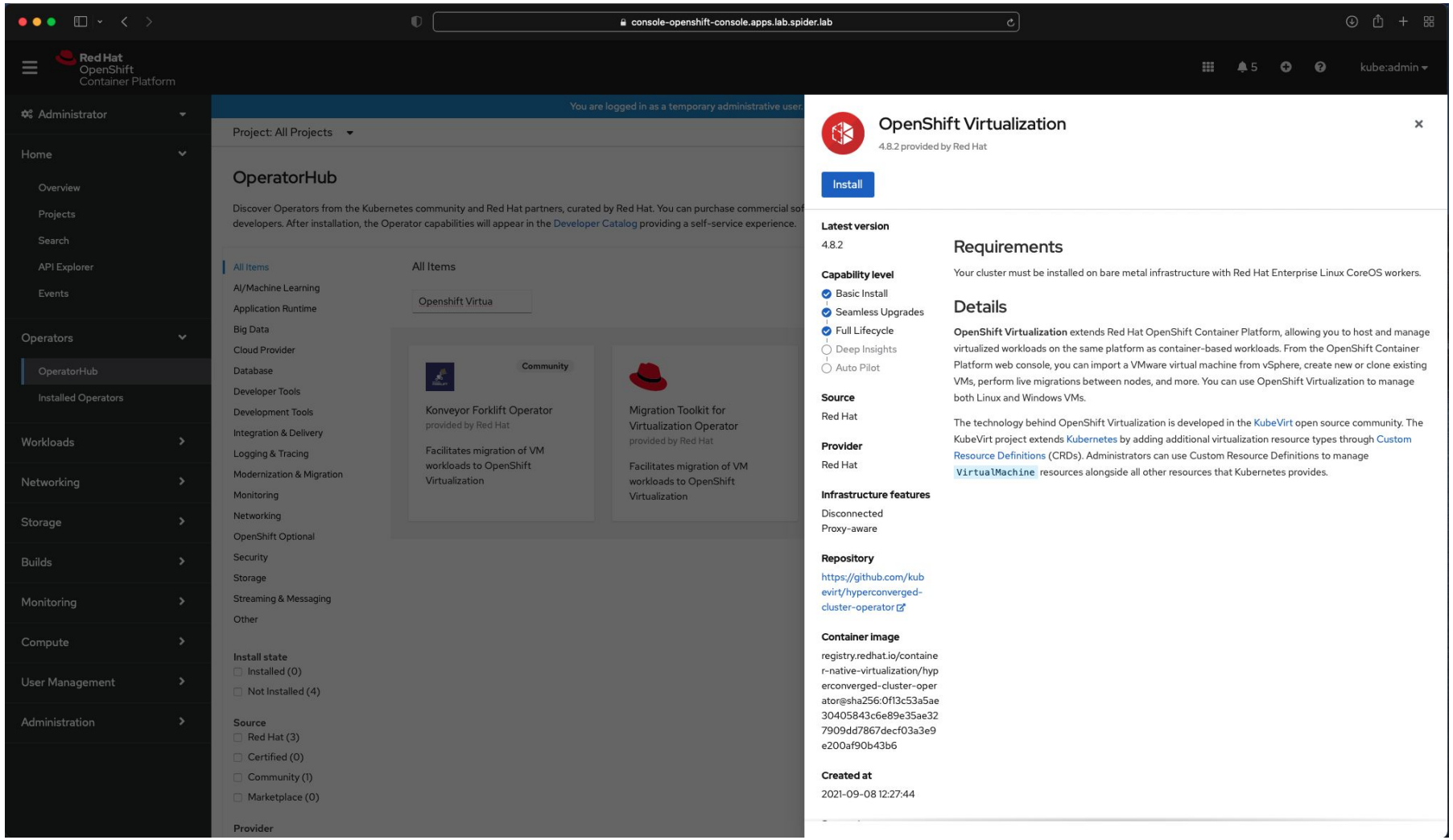


Red Hat

OpenShift Virtualization

Red Hat

How to install?



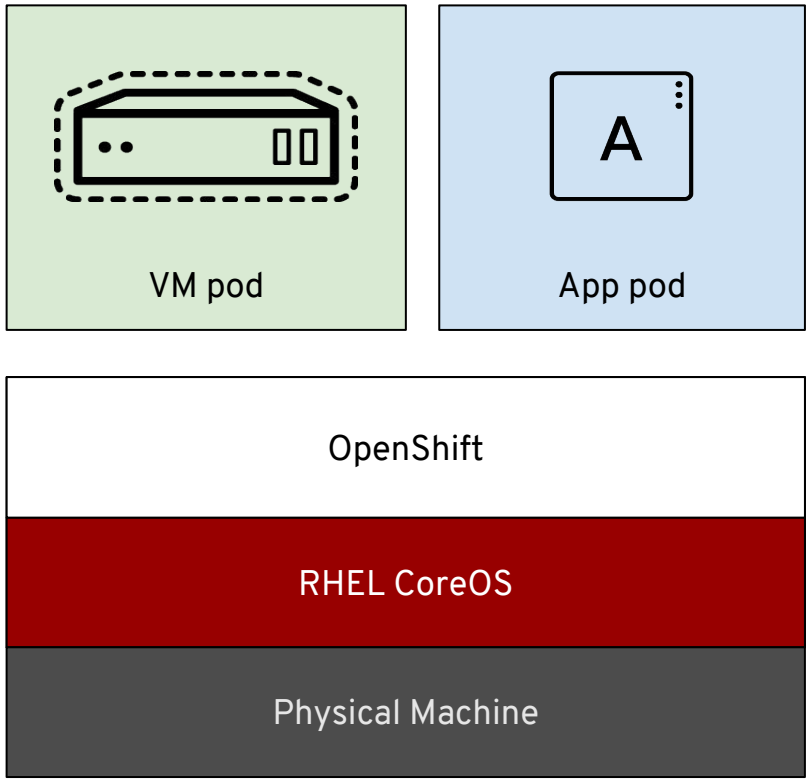
Red Hat

VMs in a container world

Red Hat

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



Red Hat

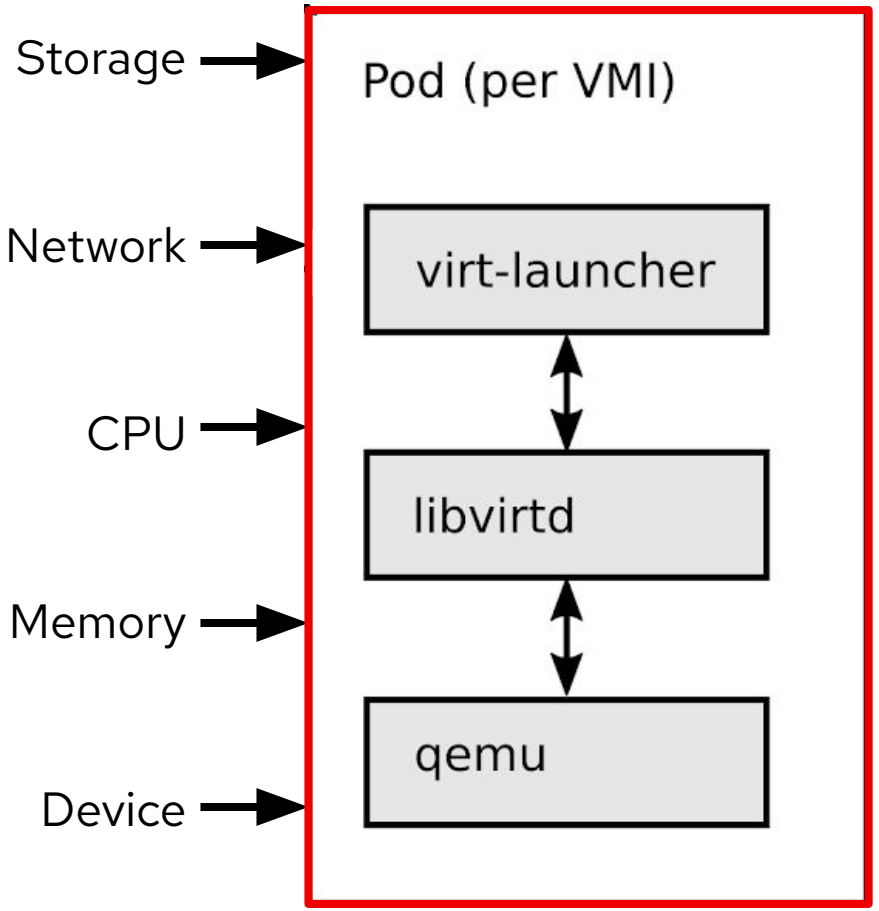
Virtualization native to Kubernetes

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

```yaml
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
```

Red Hat

Contenerized 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

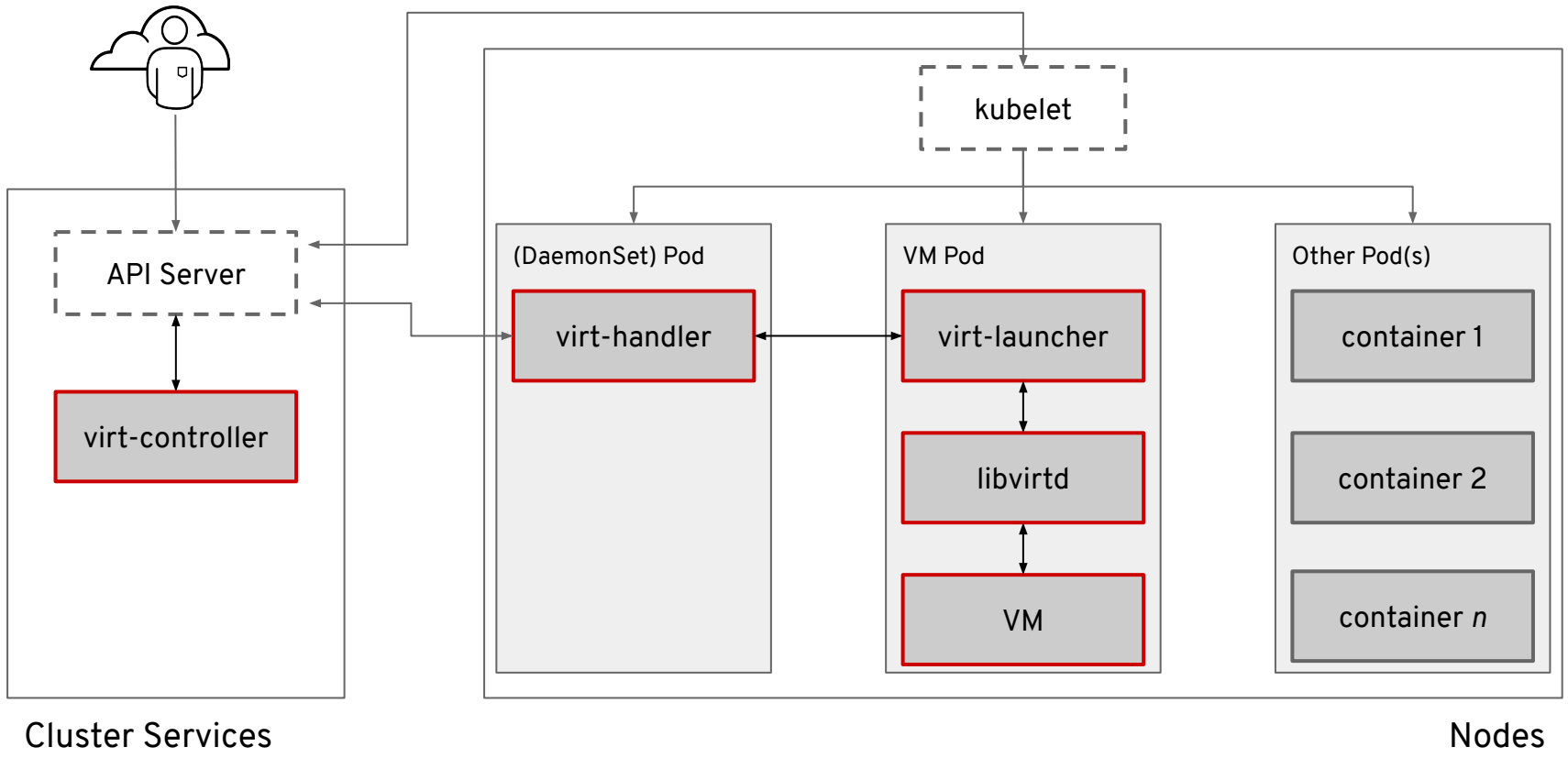
- libvirt and qemu from RHEL are mature, have high performance, provide stable abstractions, and have a minimal overhead.

Security - Defense in depth

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

Red Hat

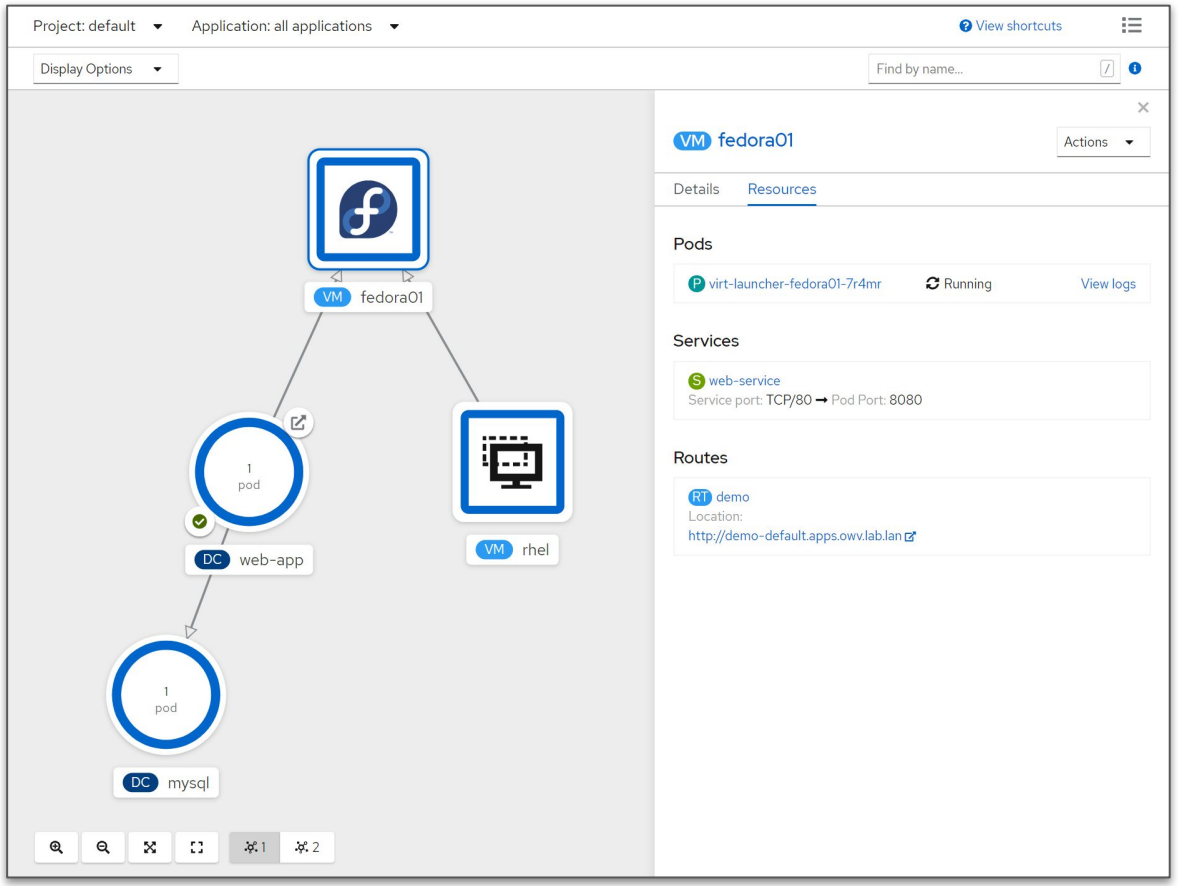
Architectural Overview



Red Hat

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

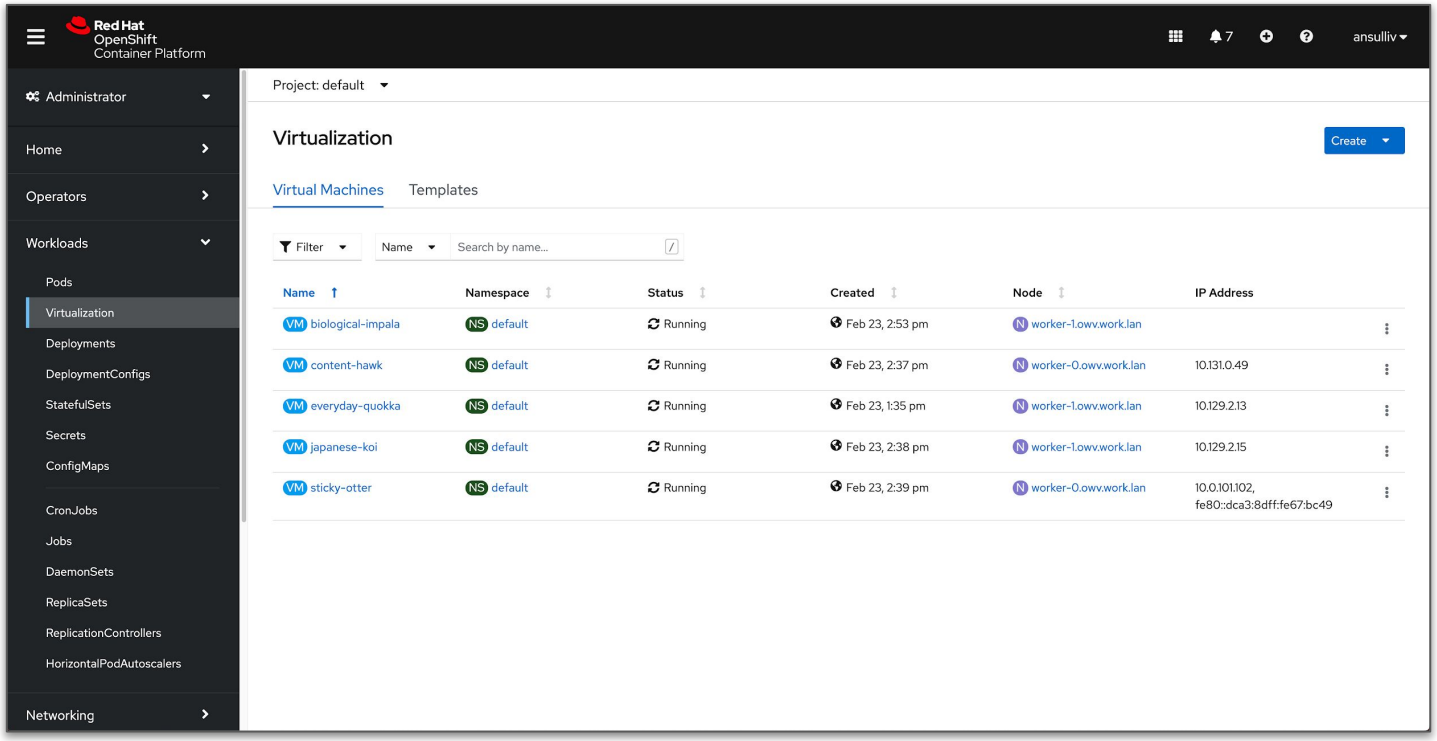


Managed VM with OpenShift

Red Hat

Virtual Machine Management

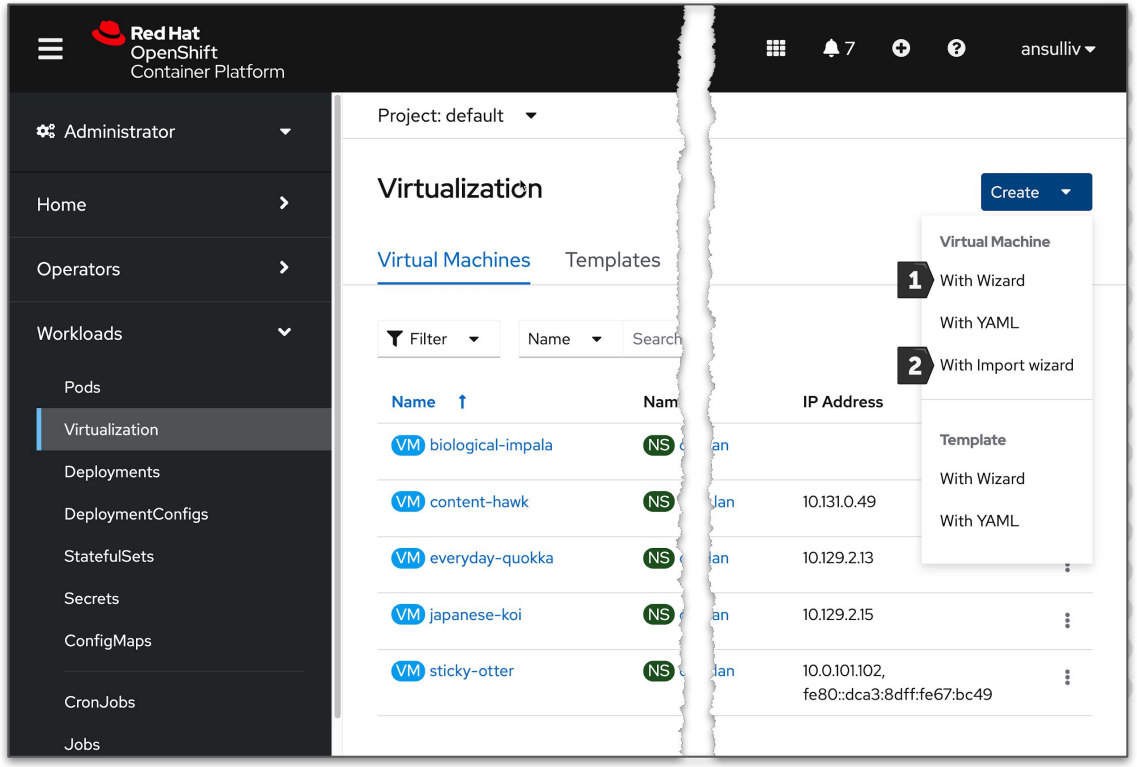
- Create, modify, and destroy virtual machines, and their resources, using the OpenShift web interface or CLI
- Use the `virtctl` command to simplify virtual machine interaction from the CLI



Red Hat

Virtual Machine creation

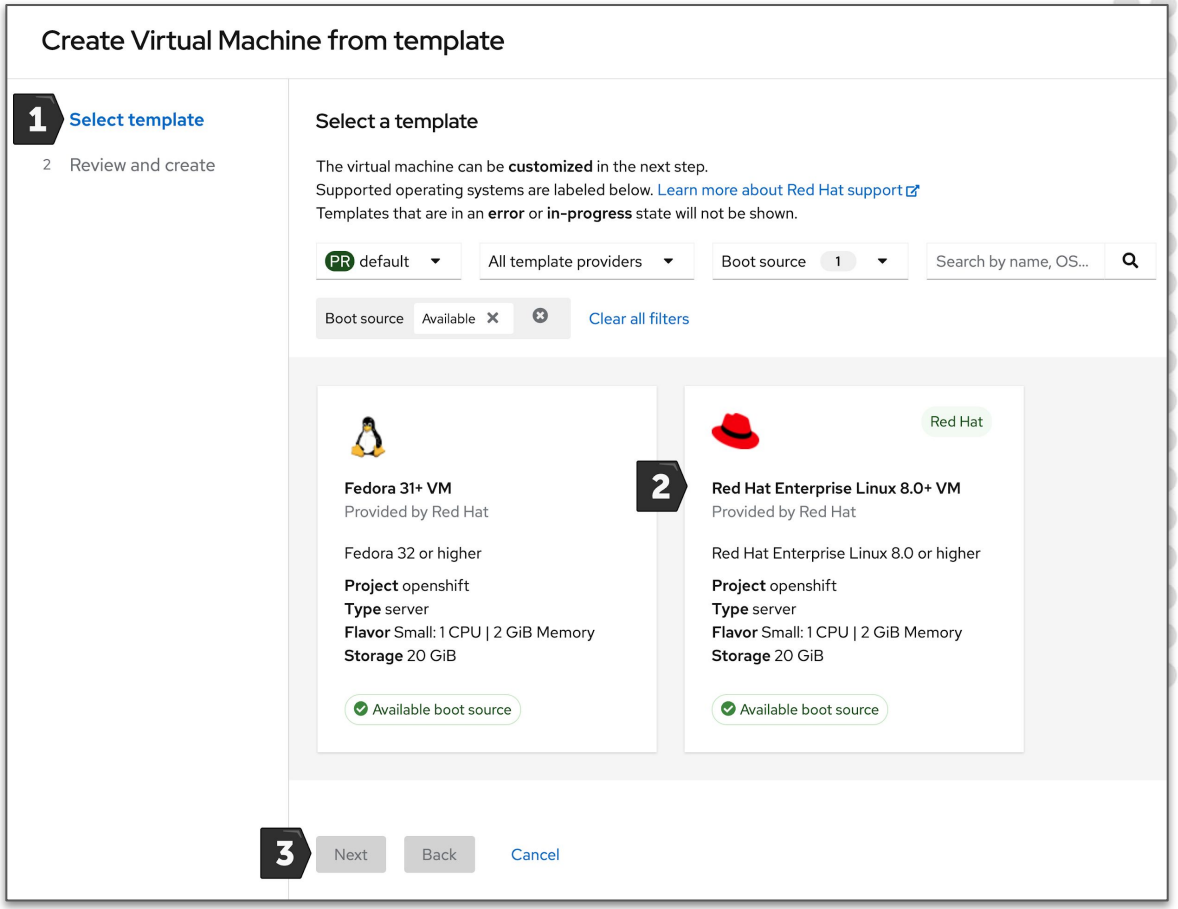
- Streamlined and simplified creation via the GUI or create VMs programmatically using YAML
- Full configuration options for compute, network, and storage resources
  - Clone VMs from templates or import disks using DataVolumes
  - Pre-defined and customizable presets for CPU/RAM allocations
  - Workload profile to tune KVM for expected behavior
- Import VMs from VMware vSphere or Red Hat Virtualization



Red Hat

Using templates for virtual machines

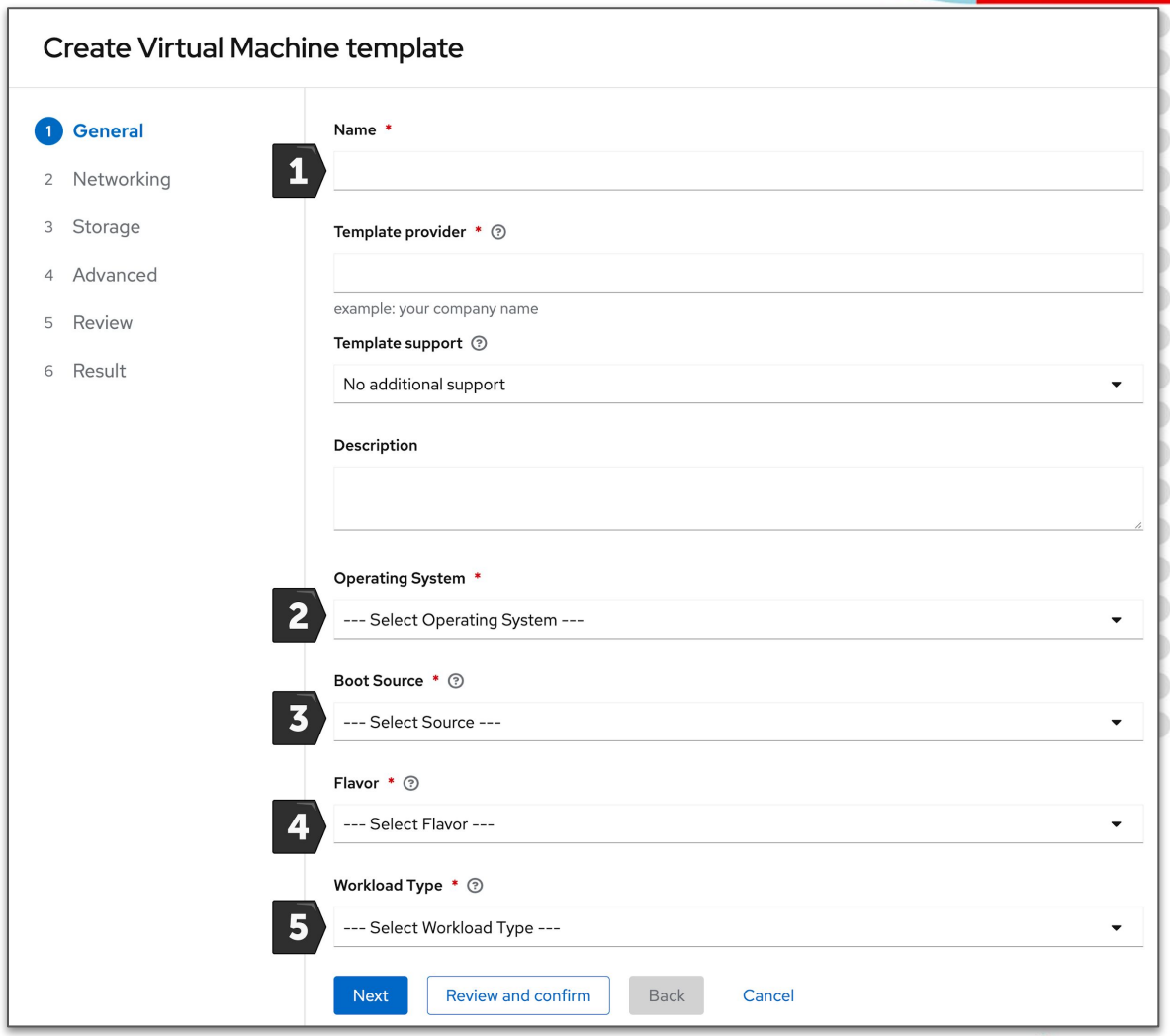
- Simplified and streamlined virtual machine creation experience for VM consumers
- Administrators configure templates with an OS disk, consumers select from the options



Red Hat

Create a template - General

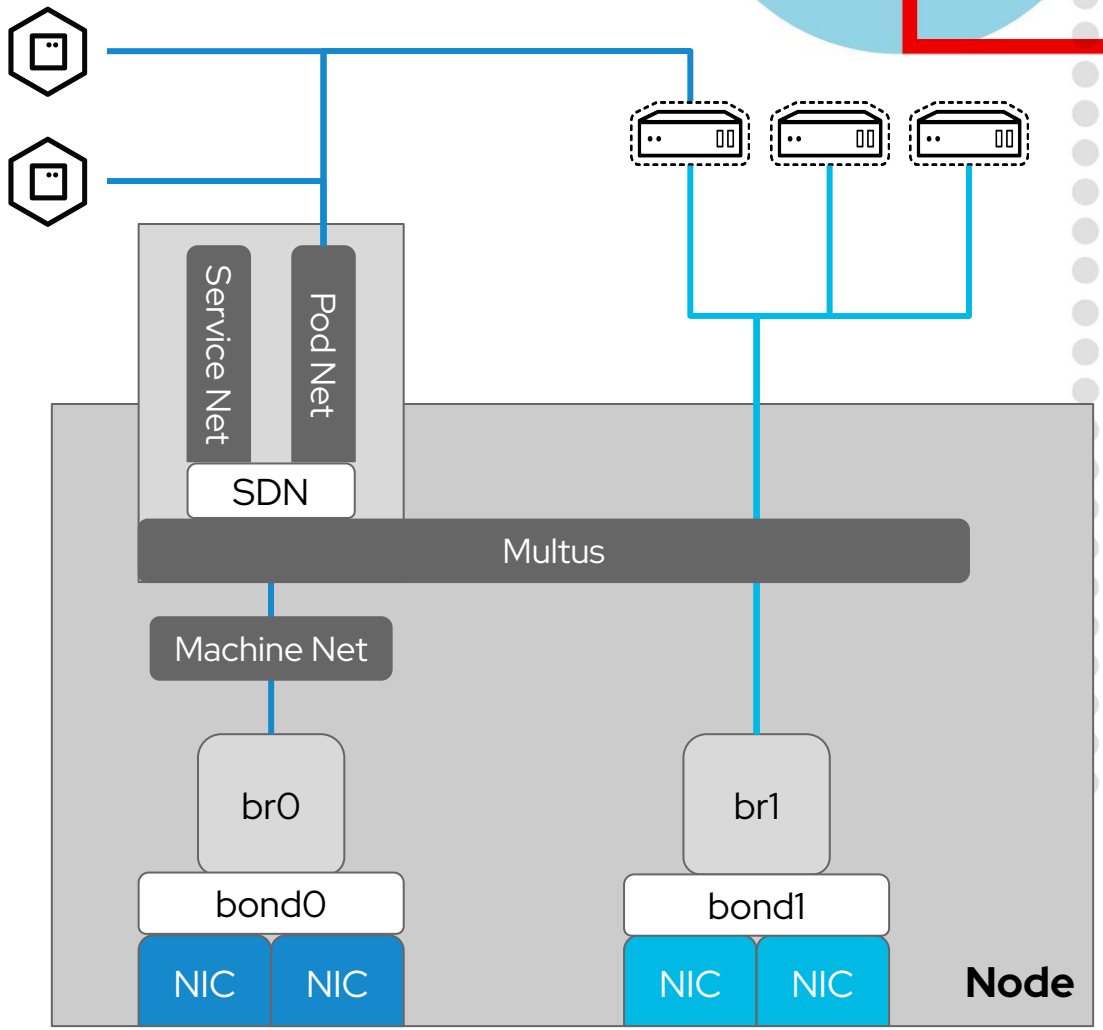
- In addition to unique names, each template is associated with a provider
  - Providers represent who created the template, with optional support information
- The guest operating system and source boot disk are provided. A boot disk can be imported during the process, or an ISO can be used to boot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



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

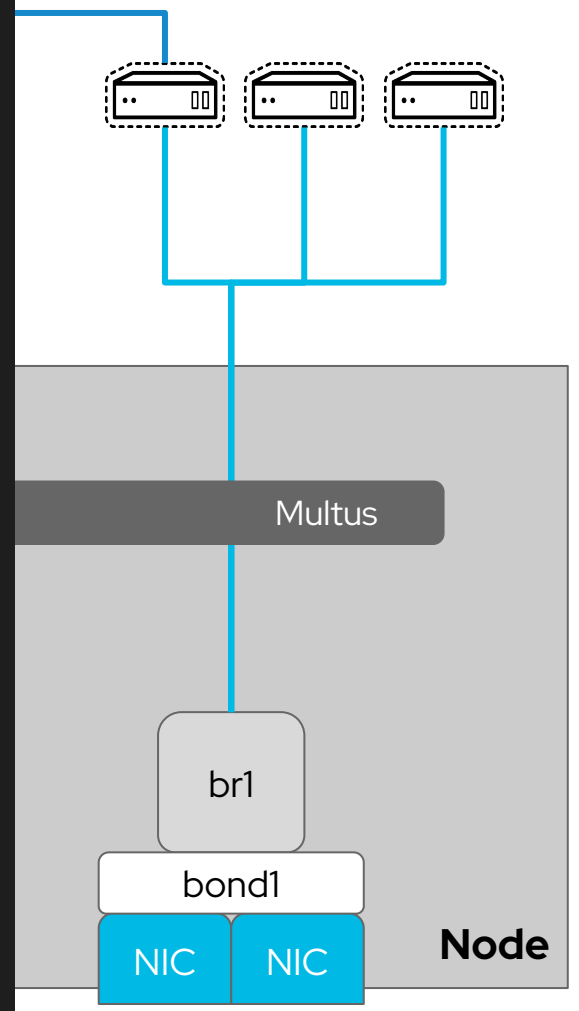


Red Hat

Host bond configuration

- NodeNetworkConfiguration-Policy (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

```yaml
apiVersion: nmstate.io/v1alpha1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: worker-bond1
spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ""
  desiredState:
    interfaces:
    - name: bond1
      type: bond
      state: up
      ipv4:
        enabled: false
      link-aggregation:
        mode: balance-alb
        options:
          miimon: '100'
        slaves:
        - eth2
        - eth3
      mtu: 1450
```



Red Hat

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

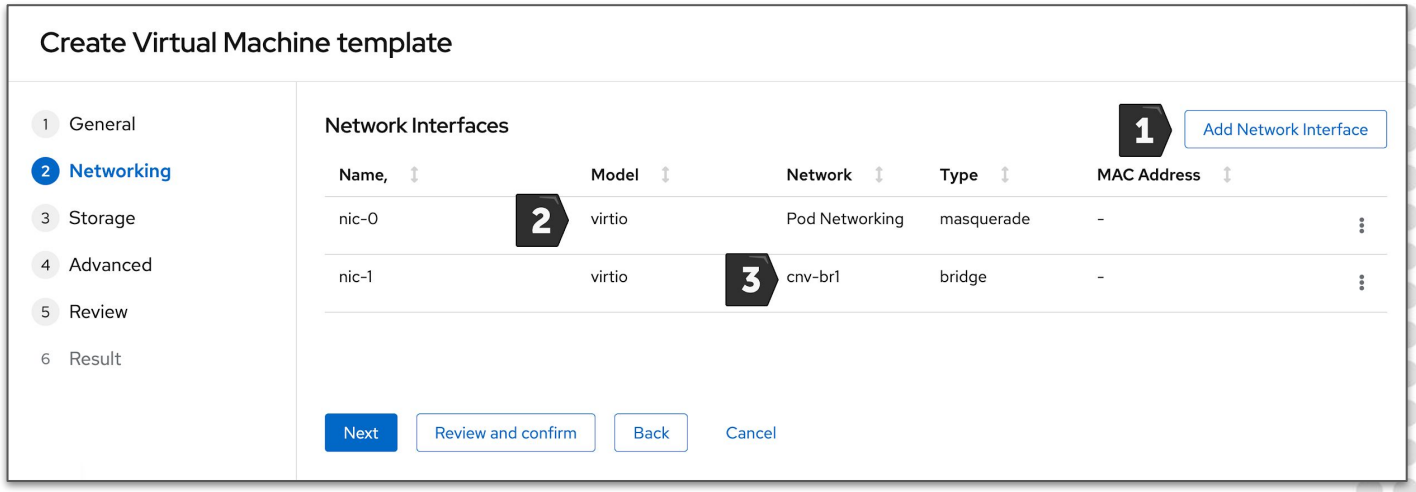
```yaml
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
  name: br1-public
  annotations:
    k8s.v1.cni.cncf.io/resourceName: bridge.network.kubevirt.io/br1
spec:
  config: '{
    "cniVersion": "0.3.1",
    "name": "br1-public",
    "plugins": [
      {
        "type": "cnv-bridge",
        "bridge": "br1"
      },
      {
        "type": "cnv-tuning"
      }
    ]
  }'
```

```yaml
kind: Pod
apiVersion: v1
metadata:
  name: application-pod
  annotations:
    k8s.v1.cni.cncf.io/networks: bond1-br1
```


Red Hat

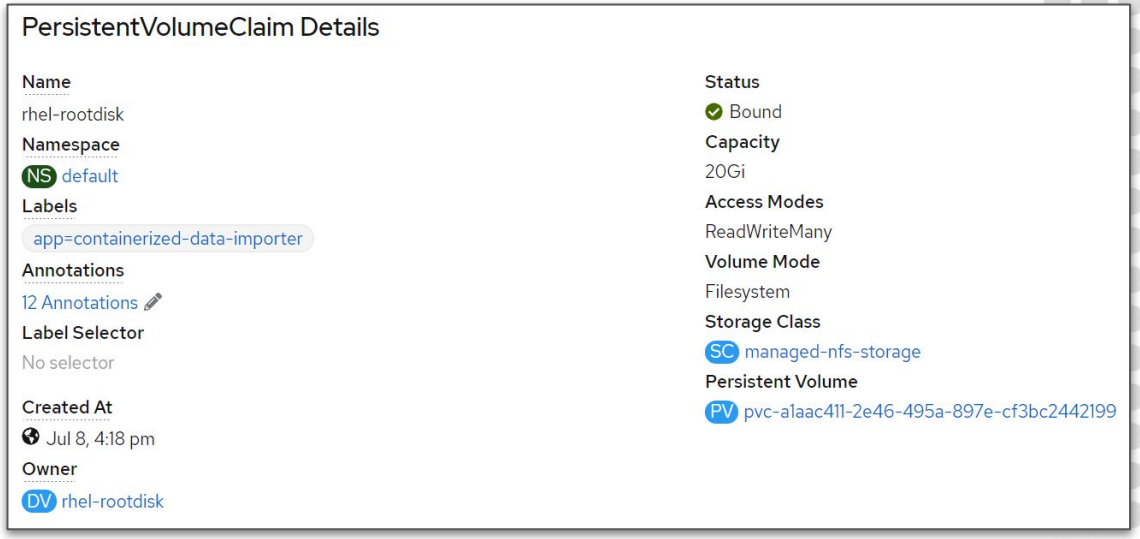
Create a template - Networks

- 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



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



Red Hat

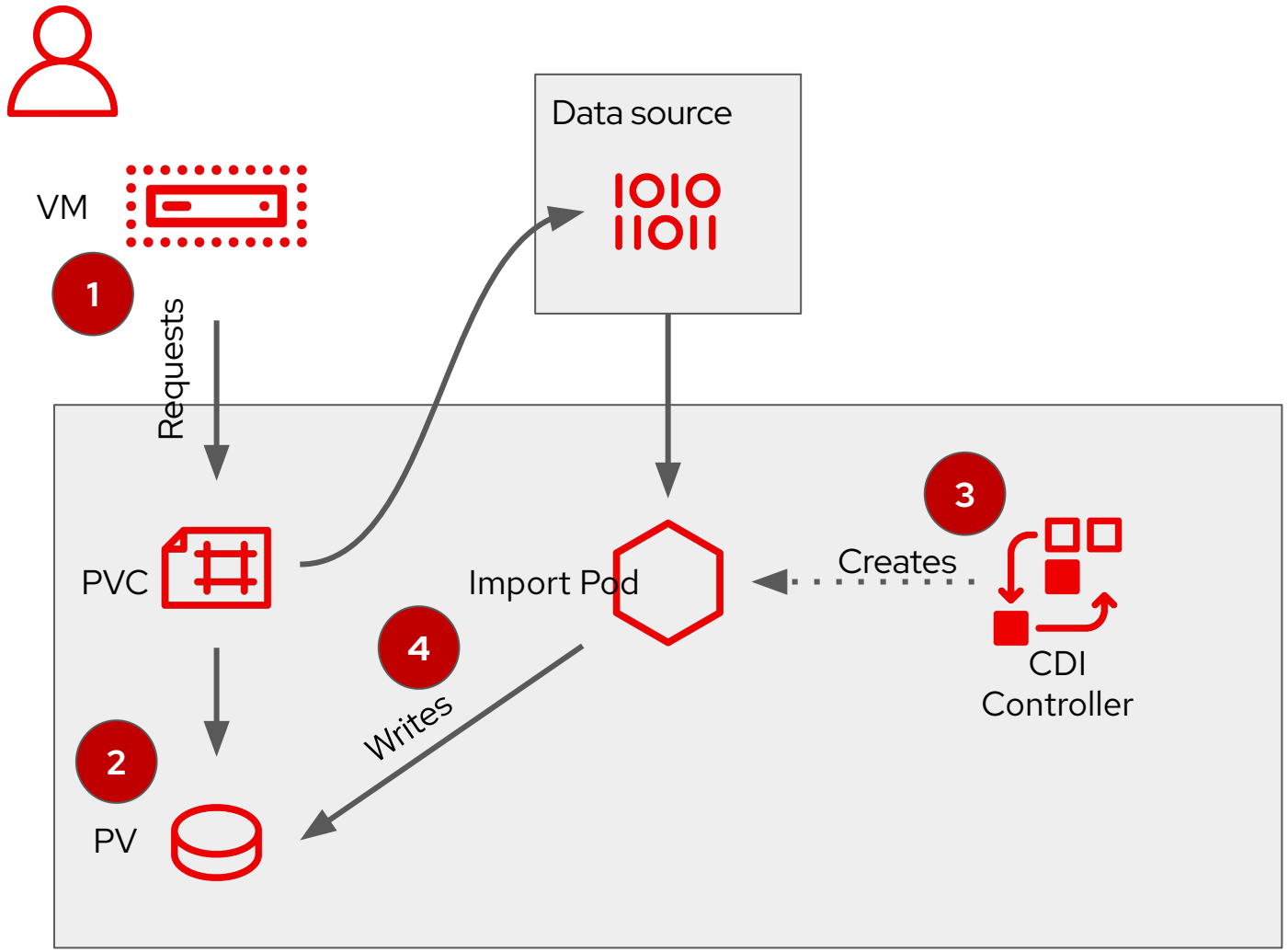
DataVolumes

- 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)

```yaml
dataVolumeTemplates:
  - apiVersion: cdi.kubevirt.io/v1alpha1
    kind: DataVolume
    metadata:
      creationTimestamp: null
      name: vm-rootdisk
    spec:
      pvc:
        accessModes:
          - ReadWriteMany
        resources:
          requests:
            storage: 20Gi
        storageClassName: my-storage-class
        volumeMode: Filesystem
      source:
        http:
          url: 'http://web.server/disk-image.qcow2'
```

Red Hat

Contenerized Data Importer

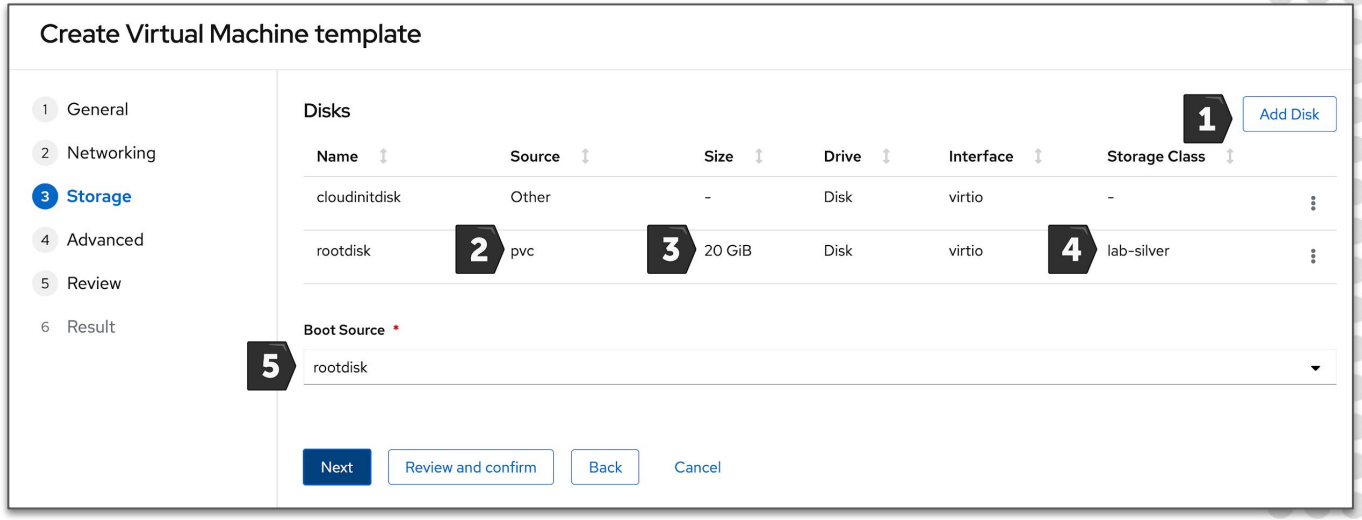


1. The user creates a virtual machine with a DataVolume
2. The StorageClass is used to satisfy the PVC request
3. The CDI controller creates an importer pod, which mounts the PVC and retrieves the disk image. The image could be sourced from S3, HTTP, or other accessible locations
4. After completing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

Red Hat

Create a template - Storage

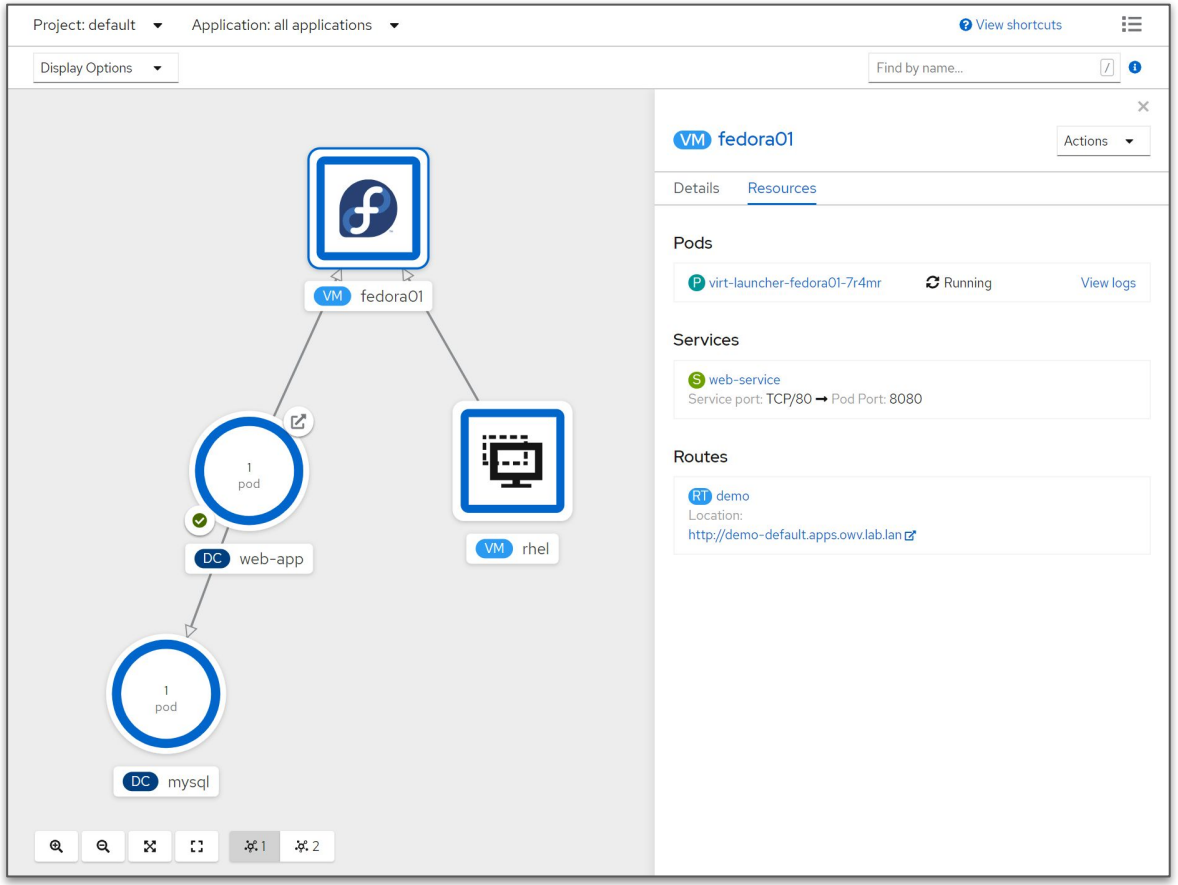
- Add or edit persistent storage
- Disks can be sourced from
  - Imported QCOW2 or raw images
  - New or existing PVCs
  - Clone existing PVCs
- Use SATA/SCSI interface for compatibility or VirtIO for paravirtual performance
- For new or cloned disks, select from available storage classes
  - Customize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



Red Hat

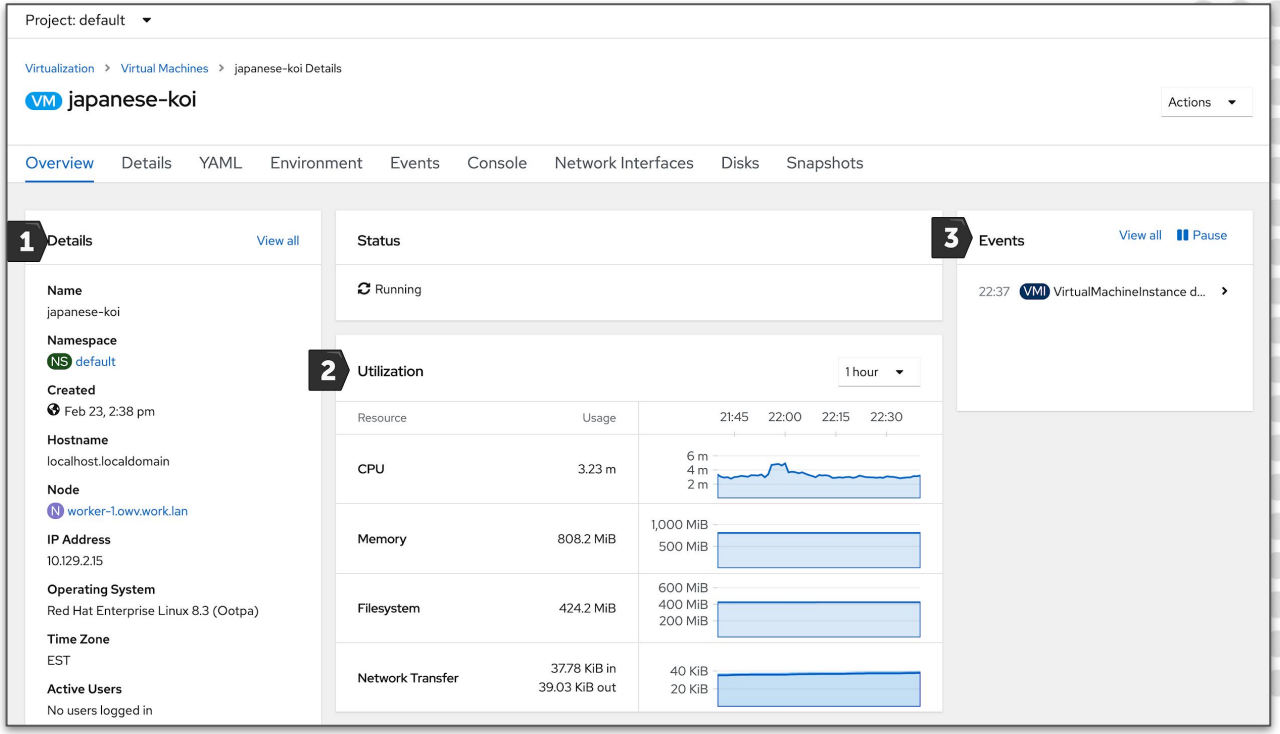
Using VMs and containers together

- 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

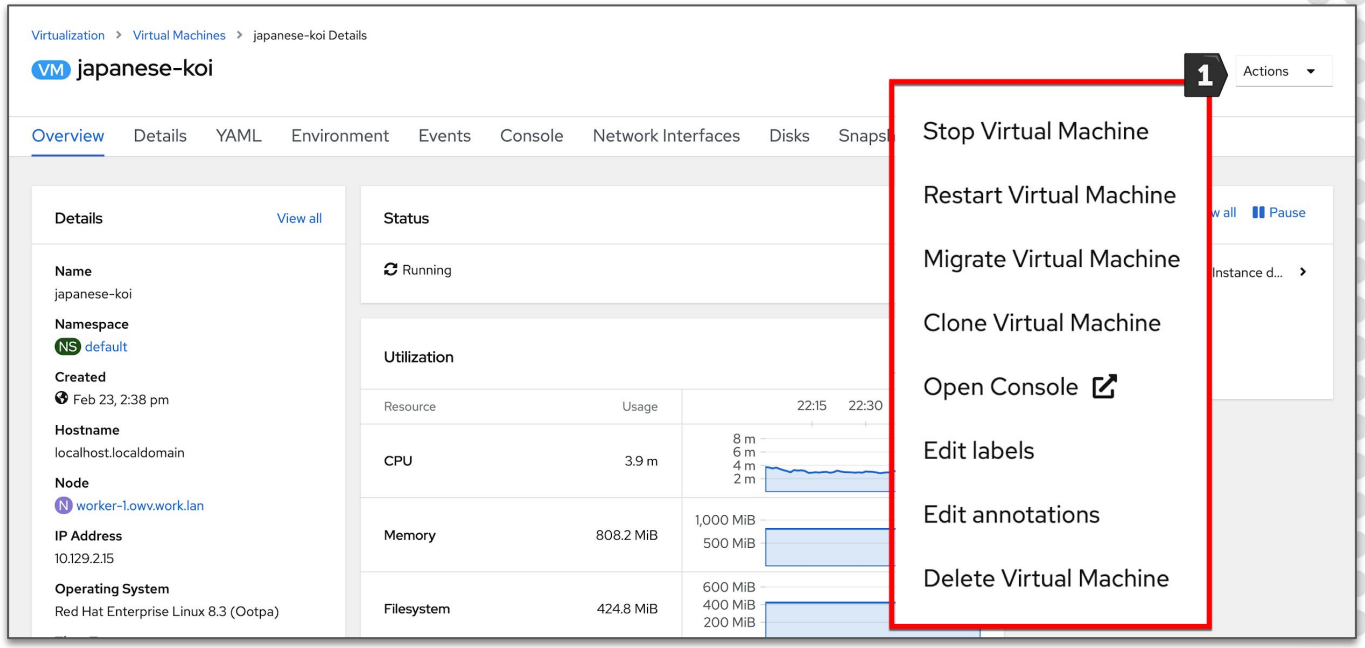
- General overview about the virtual machine
- Information populated from guest when integrations are available
  - IP address, etc.
- Inventory quickly shows configured hardware with access to view/manage
- Utilization reporting for CPU, RAM, disk, and network
- Events related to the Pod, scheduling,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



Red Hat

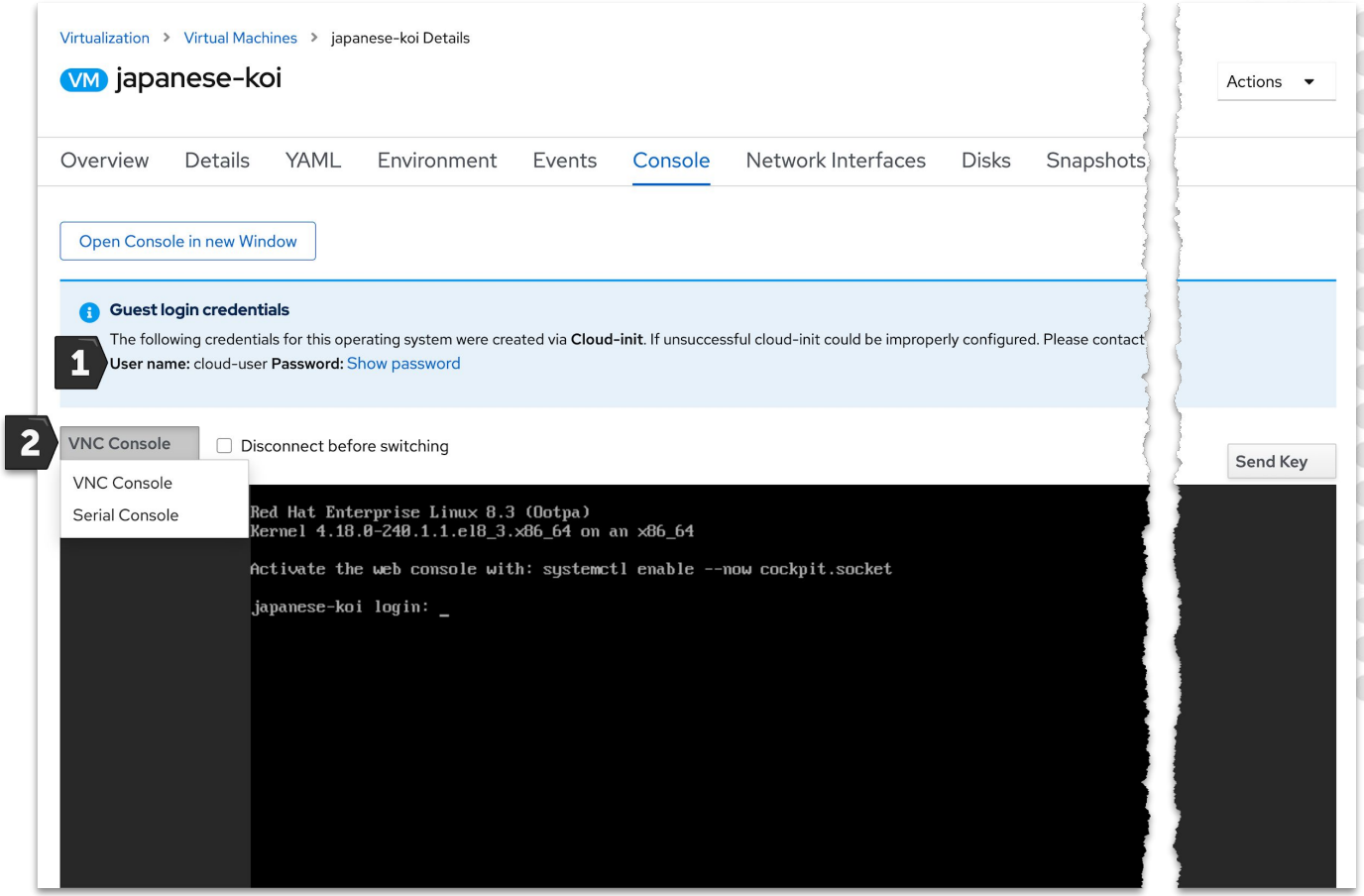
Virtual Machine - Actions

- 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

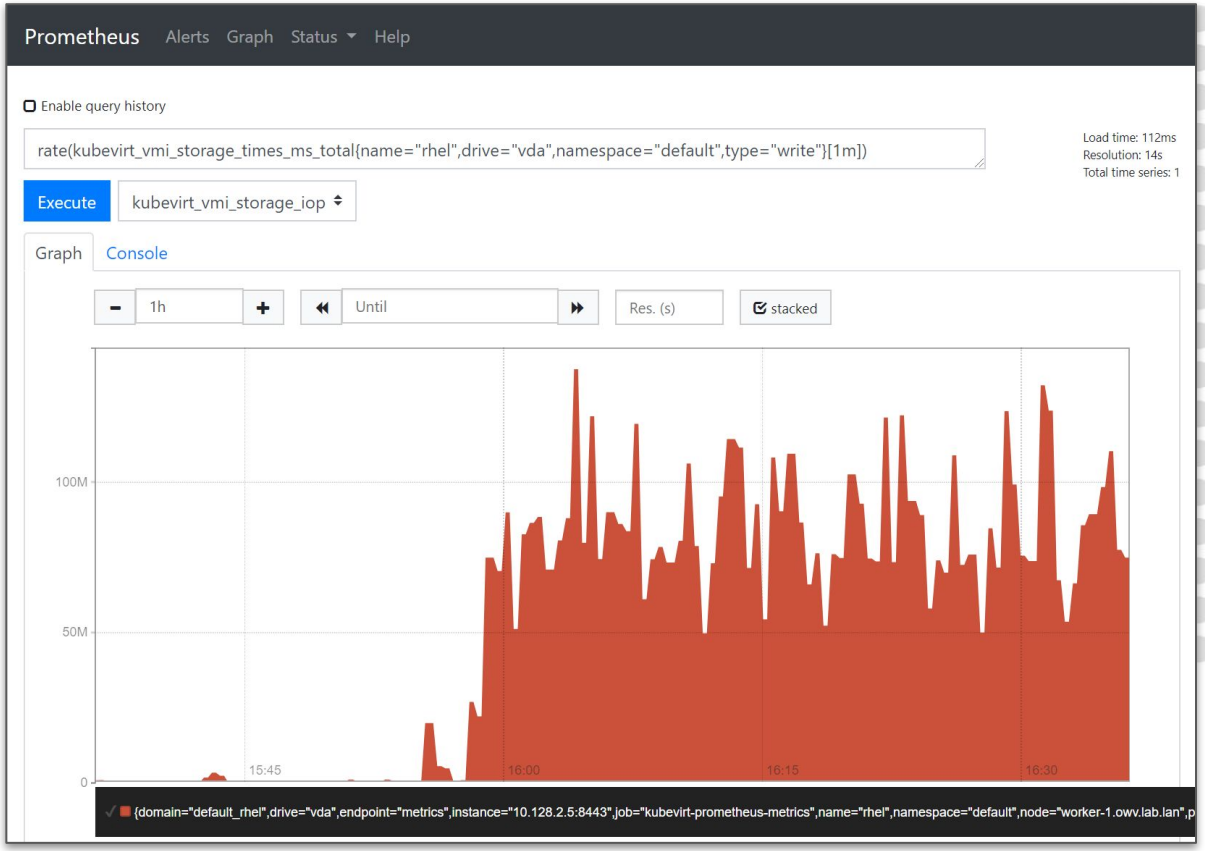
- Browser-based access to the serial and graphical console of the virtual machine
- Access the console using native OS tools, e.g. `virt-viewer`, using the `virtctl` CLI command
  - `virtctl console vmname`
  - `virtctl vnc vmname`



Red Hat

Detailed Virtual Machine metrics

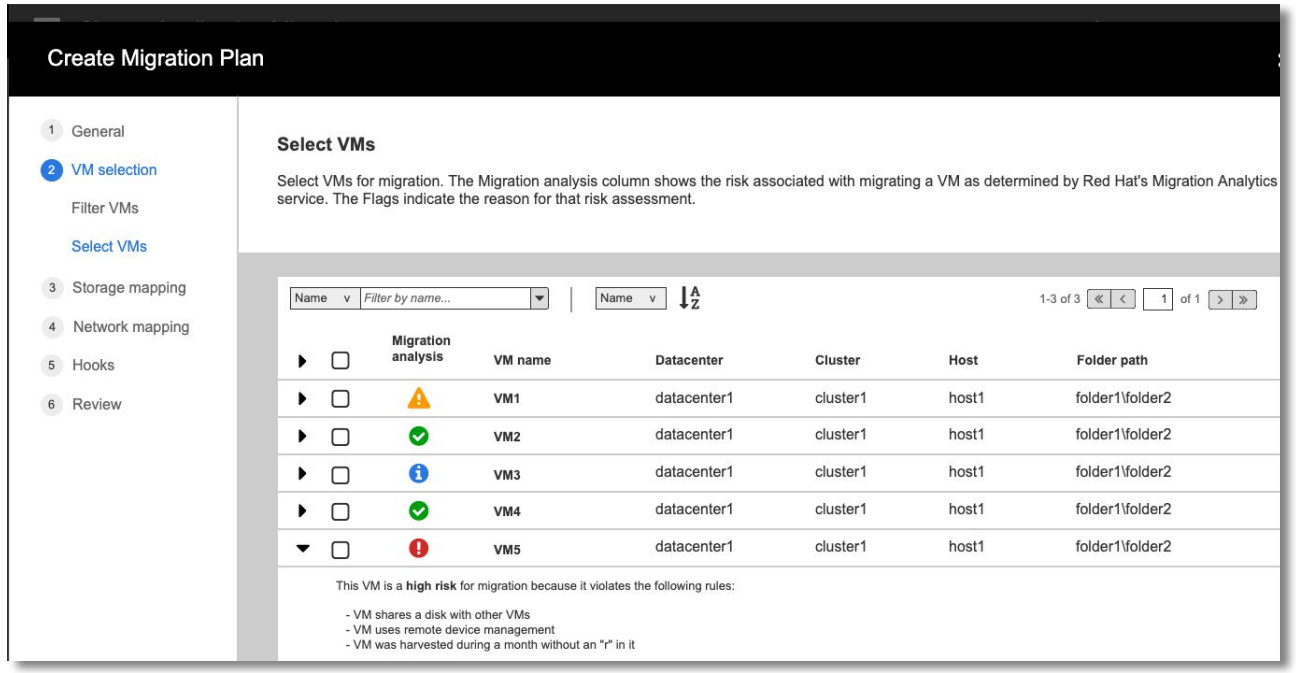
- Virtual machine, and VM pod, metrics are collected by the OpenShift metrics service
  - Available under the `kubevirt` namespace in Prometheus
- Available per-VM metrics include
  - Active memory
  - Active CPU time
  - Network in/out errors, packets, and bytes
  - Storage R/W IOPS, latency, and throughput
- 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



Red Hat

VM migration

Migration Toolkit for Virtualization (MTV)



Migration Analytics

Detect potential compatibility issues before migrating to ensure a successful migration

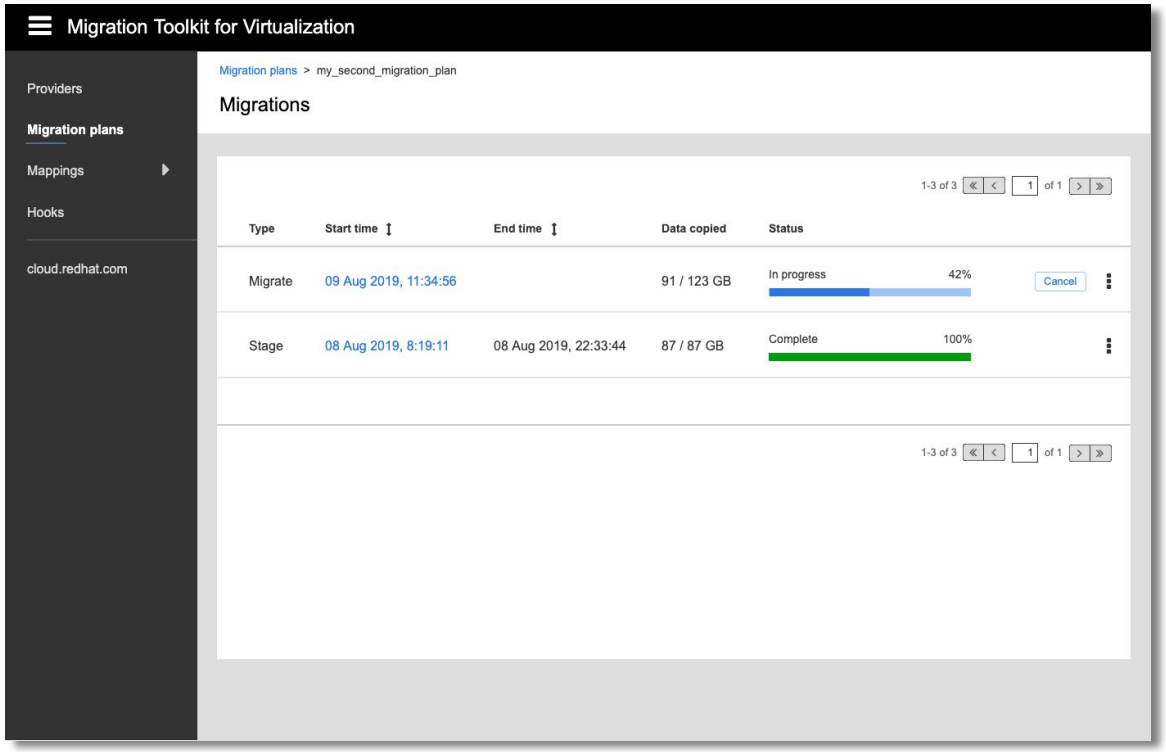
Mass Migration of VMs

Migrate workloads at scale to OpenShift

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

Red Hat

Migration Toolkit for Virtualization (MTV)



Migration Analytics

Detect potential compatibility issues before migrating to ensure a successful migration

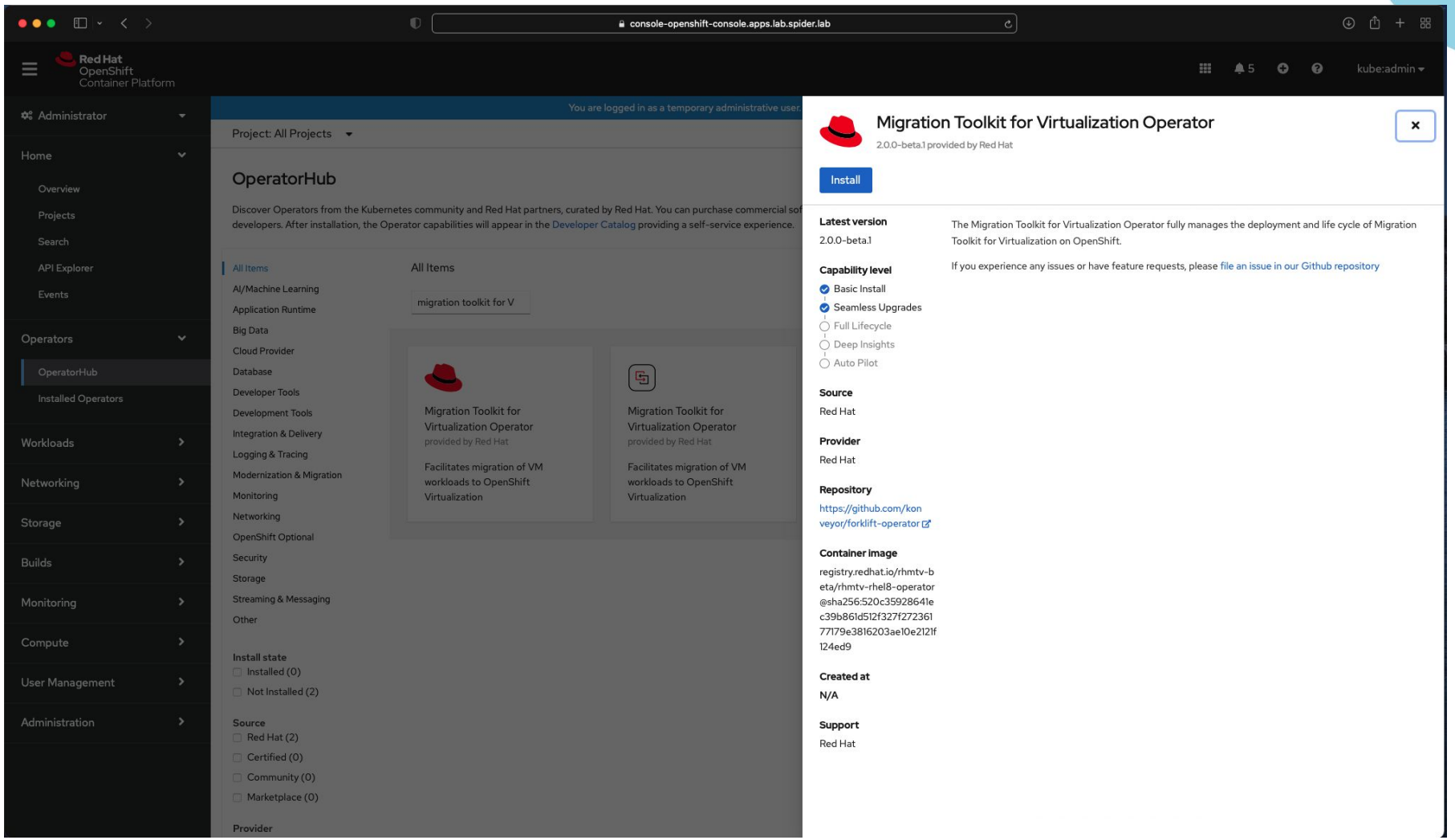
Mass Migration of VMs

Migrate workloads at scale to OpenShift

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

Red Hat

How to install?



Red Hat

OpenShift Virtualization conditions

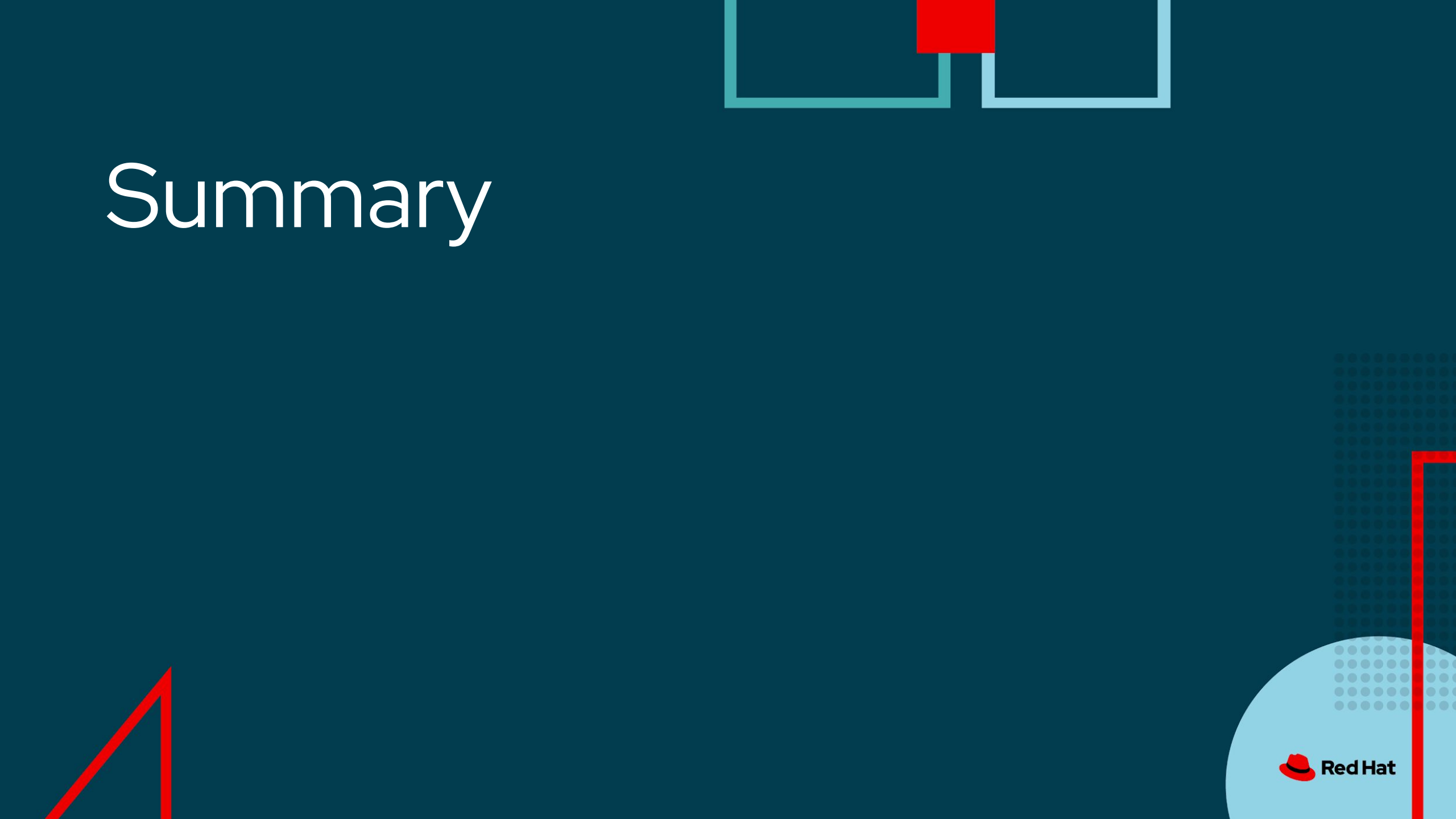
Red Hat

Supported configuration

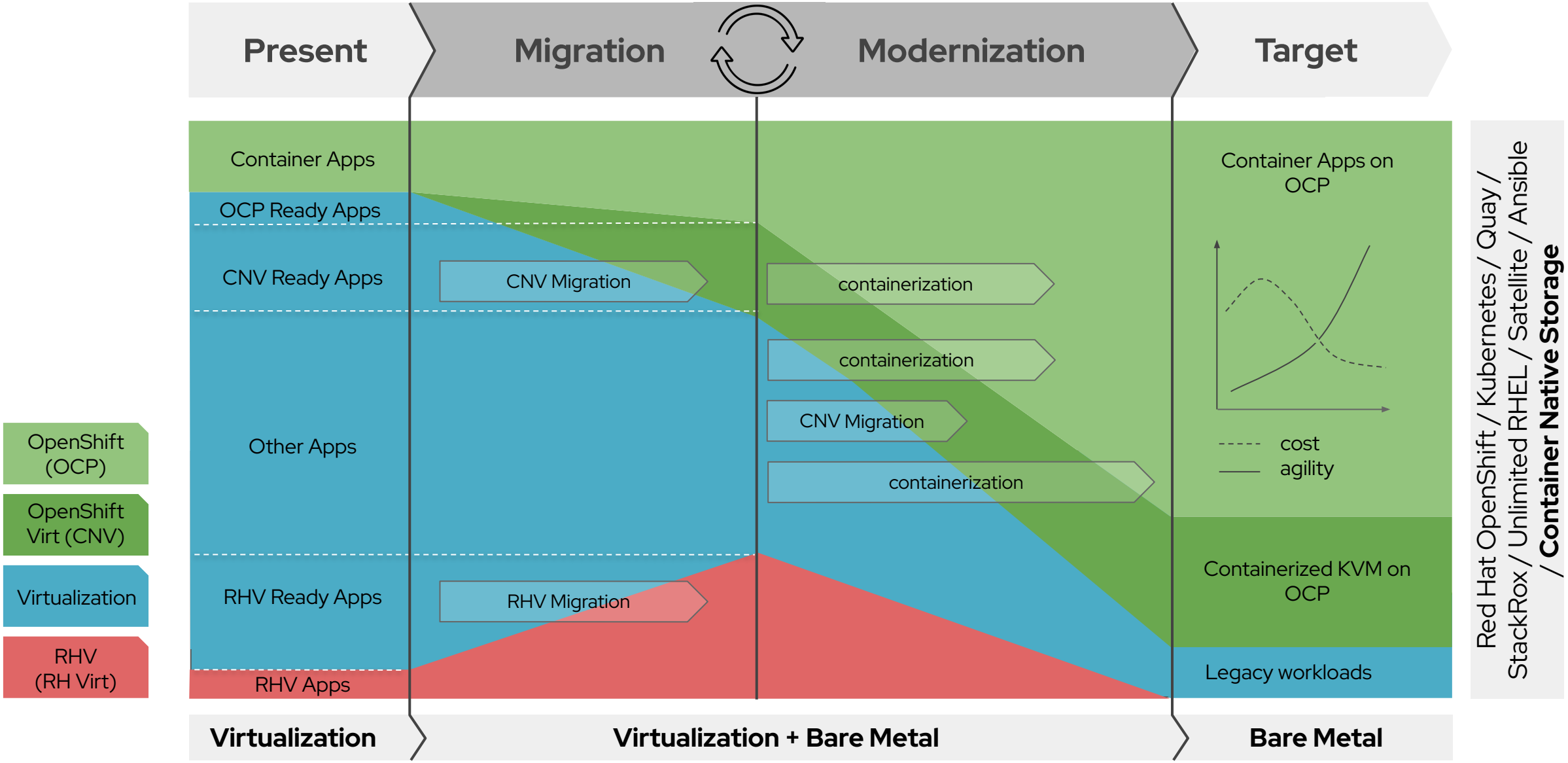
- VM nodes should be physical with CPU virtualization technology enabled in the BIOS
  - Nested virtualization *works*, but ***is not supported***
  - Emulation *works*, but ***is not supported*** (and is extremely slow)

Red Hat

Summary



Apps: Software Infrastructure landscape



OCP- Openshift Container Platform/CNV - Container Native Virtualization/RHV - Red Hat Virtualization

Red Hat

Red Hat Summit

THANK YOU

linkedin.com/company/Red-Hat

facebook.com/RedHatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat