



OpenShift virtualization

Aka Container-native virtualization (CNV)

Robert Bohne

SR. SPECIALIST SOLUTION ARCHITECT | OPENSIFT

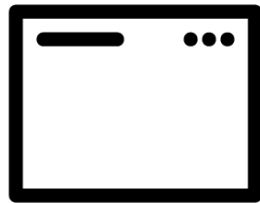
Twitter: @RobertBohne

Red Hat OpenShift and OpenShift virtualization

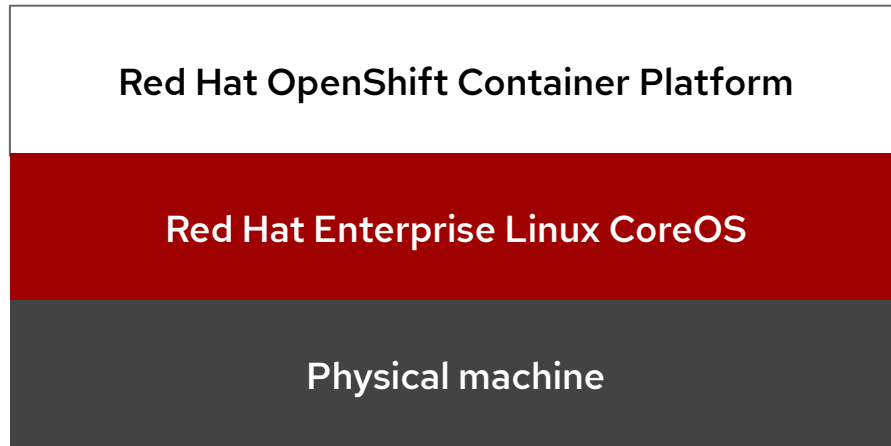
Modernize workloads and support mixed applications consisting of VMs, containers, and serverless



VMs



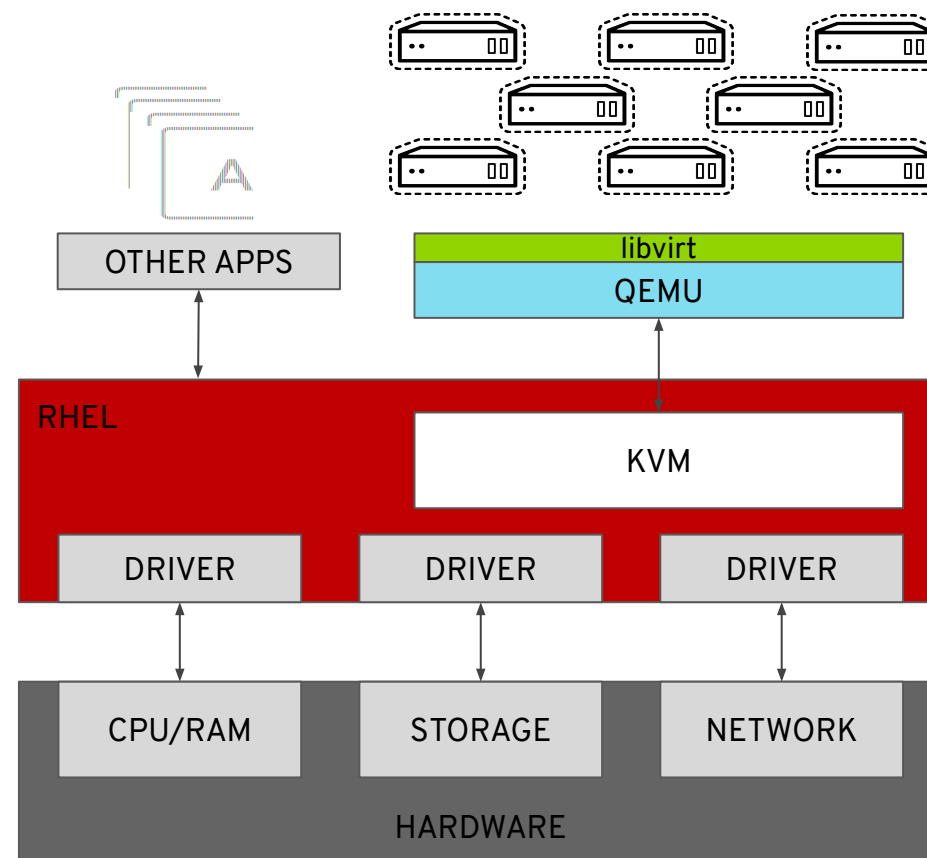
Containers



- Accelerate 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 applications over time, or maintain them as VMs

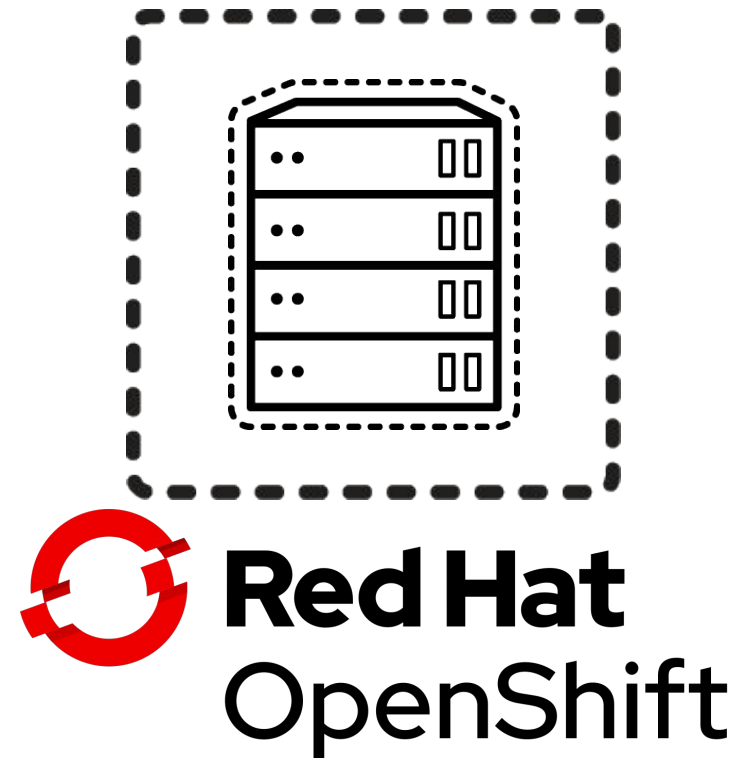
KVM-based (containerized) VMs

- 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`

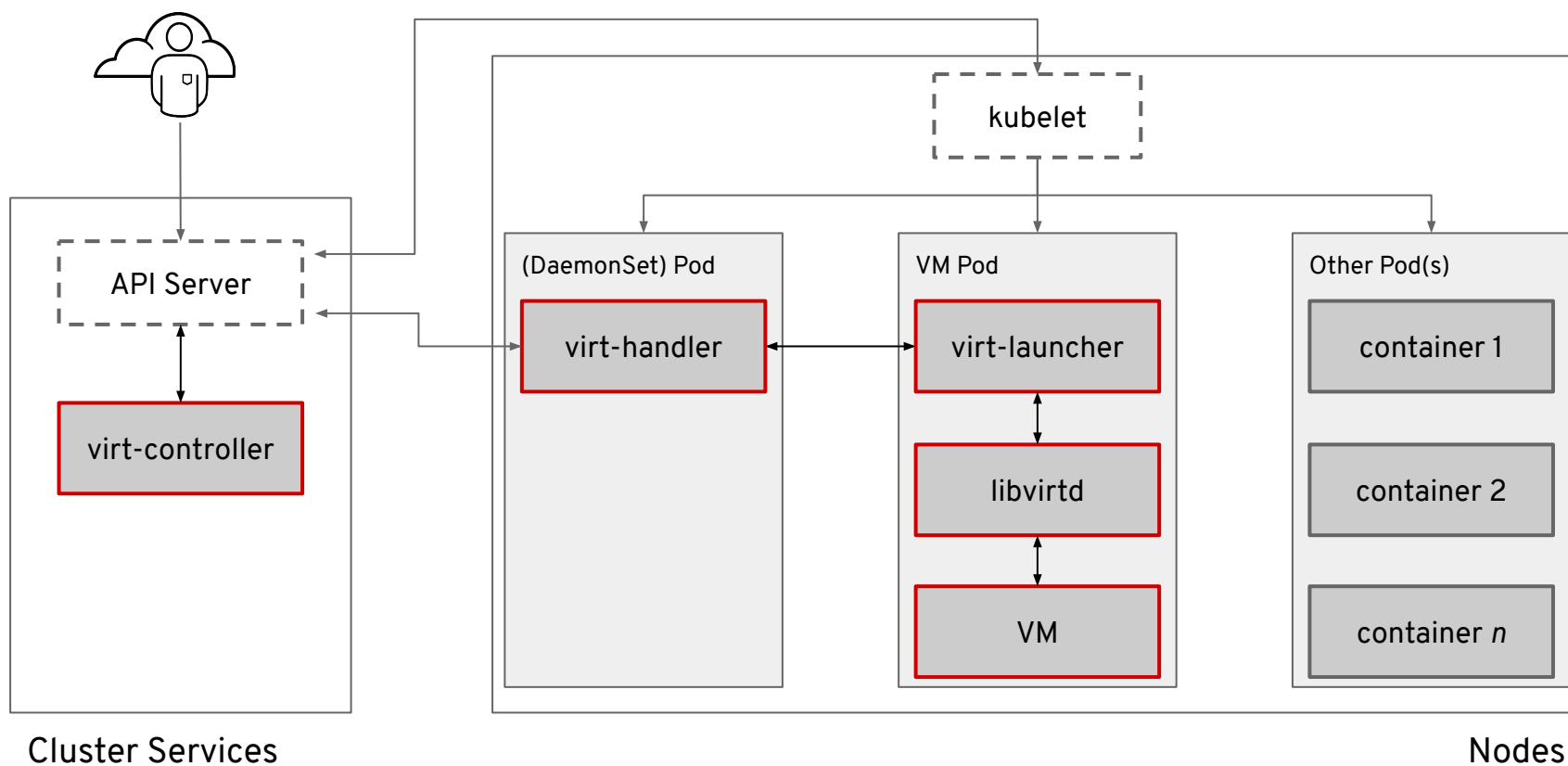


Containerized virtual machines

- 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



KubeVirt

- ▶ Open Source, Go
- ▶ Initiated in 2016 by Red Hat
- ▶ Contributions by other companies
e.g (v)GPU support by Nvidia
- ▶ CNCF sandbox project since 2019
- ▶ Provides an API for running KVM based virtual machines in Kubernetes
- ▶ Goal: run those VMs alongside with containerized workloads, using the same networks / storage etc.



KubeVirt resources

KubeVirt introduces several CRDs for managing virtual machines.

The most important ones are:

- ▶ **VirtualMachine (VM)**: represents a virtual machine, which can be started and stopped
- ▶ **VirtualMachineInstance (VMI)**: when a VirtualMachine is started, a VirtualMachineInstance is created, which represents the *running* virtual machine

```
apiVersion: kubevirt.io/v1alpha3
kind: VirtualMachine
metadata:
  name: vm-fedora
  labels:
    kubevirt.io/vm: vm-fedora
spec:
  running: false
  template:
    <vmi template>
```

```
apiVersion: kubevirt.io/v1alpha3
kind: VirtualMachineInstance
metadata:
  name: vmi-fedora
  labels:
    kubevirt.io/vm: vm-fedora
spec:
  domain:
    devices:
      disks:
        - name: containerdisk
          disk:
            bus: virtio
    resources:
      requests:
        memory: 2G
  volumes:
    - name: containerdisk
      containerDisk:
        image:
          kubevirt/fedora-container-disk-demo
```


Import and Virtual Machine

- ▶ Importing a VMware virtual machine or template
- ▶ Importing virtual machine images with DataVolumes
- ▶ Importing virtual machine images to block storage with DataVolumes
- ▶ Uploading a local disk image to a block storage DataVolume

Disk packet into a Container image

```
qemu-img convert -f raw -O qcow2 disk.img iis.qcow2
```

```
cat - > Dockerfile <<EOF
FROM scratch
LABEL maintainer="Robert Bohne <robert.bohne@redhat.com>"
ADD iis.qcow2 /disk/rhel.qcow2
EOF
```

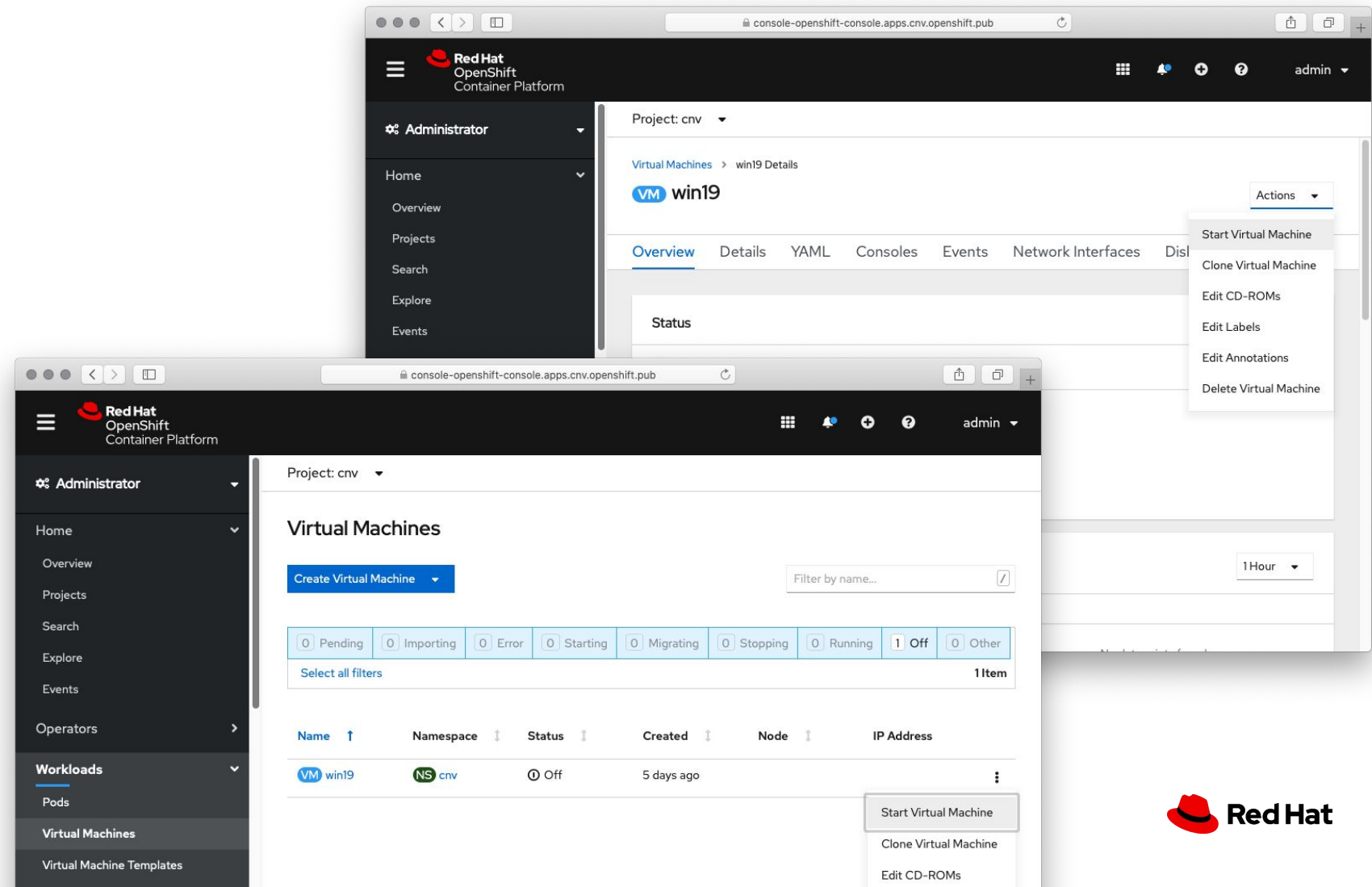
```
oc create is iis -n cnv
```

```
export REGISTRY=$(oc get route default-route -n openshift-image-registry --template='{{ .spec.host }}')
export REGISTRY_TOKEN=$(oc whoami -t)
podman login -u $(oc whoami) -p $REGISTRY_TOKEN --tls-verify=false $HOST
```

```
podman build -t ${REGISTRY}/cnv/iis:latest .
podman push ${REGISTRY}/cnv/iis:latest
```

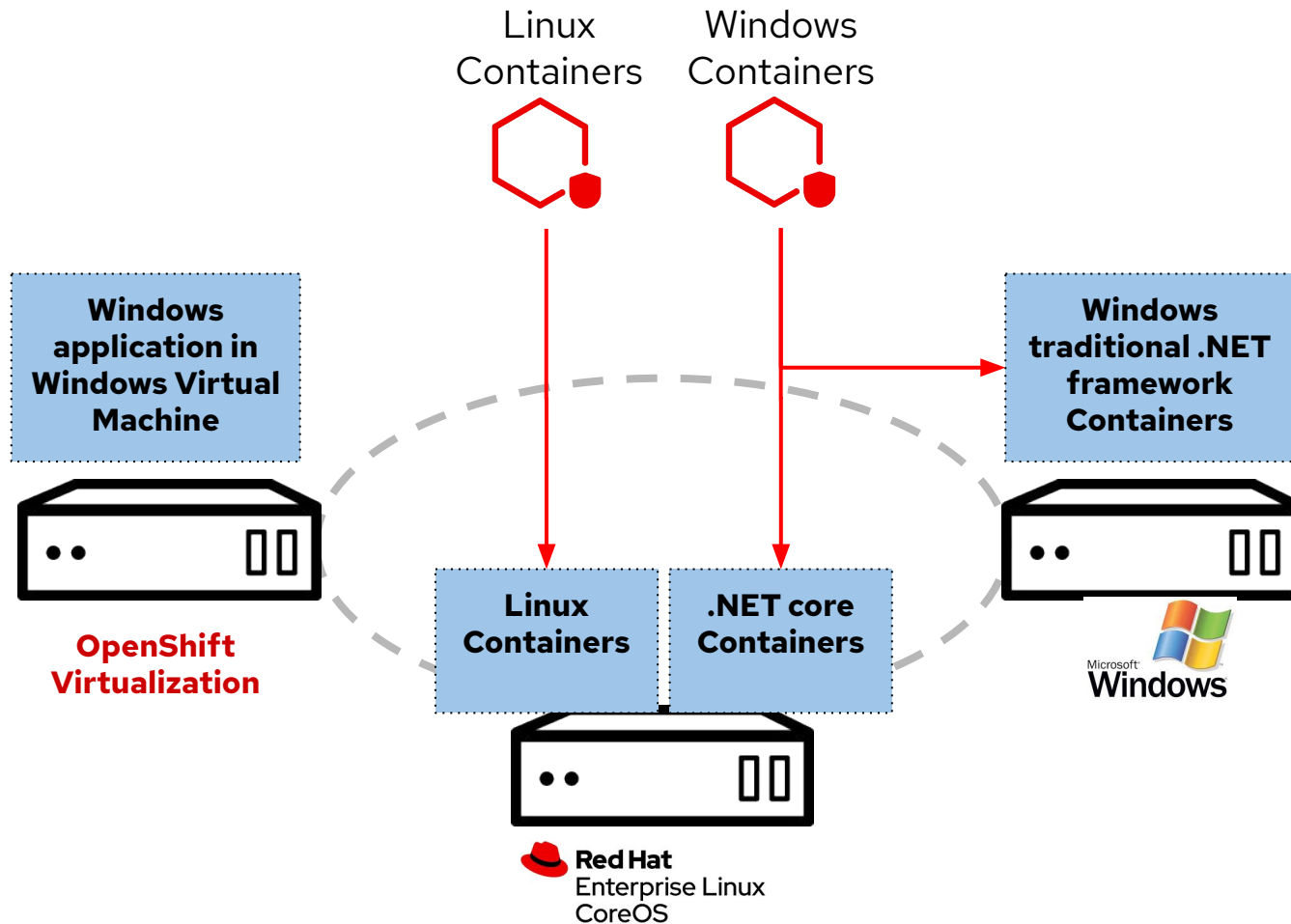
Start & Stop a Virtual Machine

- ▶ Use virtctl CLI
- ▶ WebUI



Mixed Windows and Linux Workloads

12



- Run Linux containers on RHEL
- Run .NET core containers on RHEL
- Run traditional **.NET framework containers on Windows**
- Run **Windows VMs with CNV** (Container Native Virtualization)
- All scheduled and managed by OpenShift

