Red Hat
OpenShift

Red Hat OpenShift Virtualization

An Overview

Brian Dumont
Lt Principal Solution Architect

Red Hat

What we'll discuss today

- Hybrid Cloud Evolution
- What is OpenShift Virtualization
- OpenShift Virtualization Architectural Overview
- Product and Customer Journeys

Hybrid Cloud

Reality of enterprise IT environments

Mixed infrastructure environments, diverse app portfolios, & limited automation

Infrastructure


Bare metal


Virtualization


Edge


Private cloud


Public cloud

Applications


AI/ML


Analytics


Serverless


Cloud-native and microservices


Java™


.Net


ISV

People & Processes


Developer tools

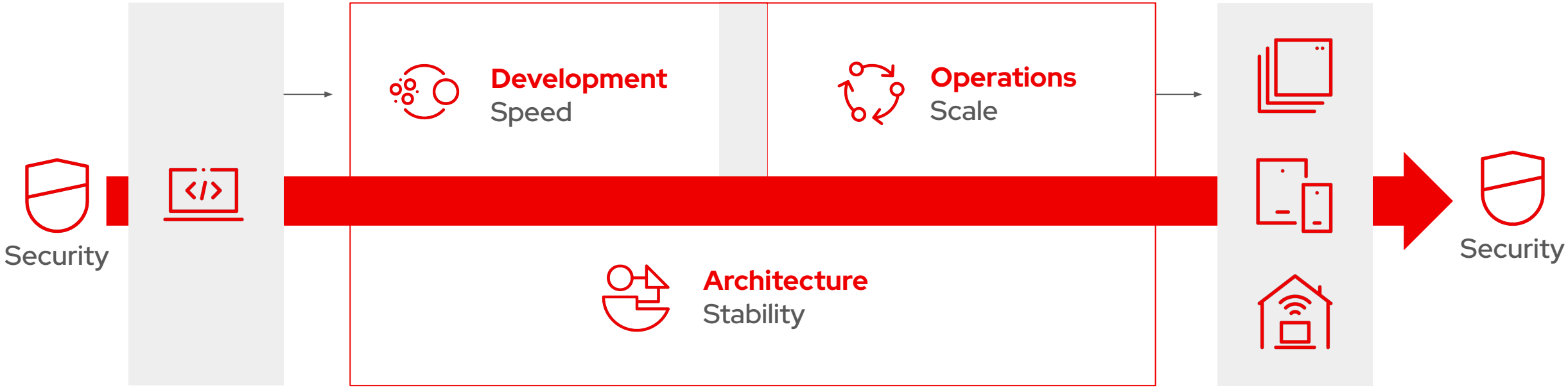

Pipeline and processes


People and policies


The right skills

Source: Red Hat detail. "The State of Enterprise Open Source," Feb. 2021.

How do you deliver business innovation faster in a complex world?



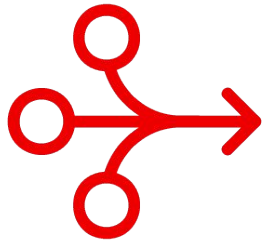
Flexibility
Consistency
Efficiency

Bare metal | Virtual | Private cloud | Public cloud | Edge

Red Hat

Challenges of traditional VMs

- Slow Evolution
- Supporting Growth
- Increasing cost
- Risk
- Developer Productivity



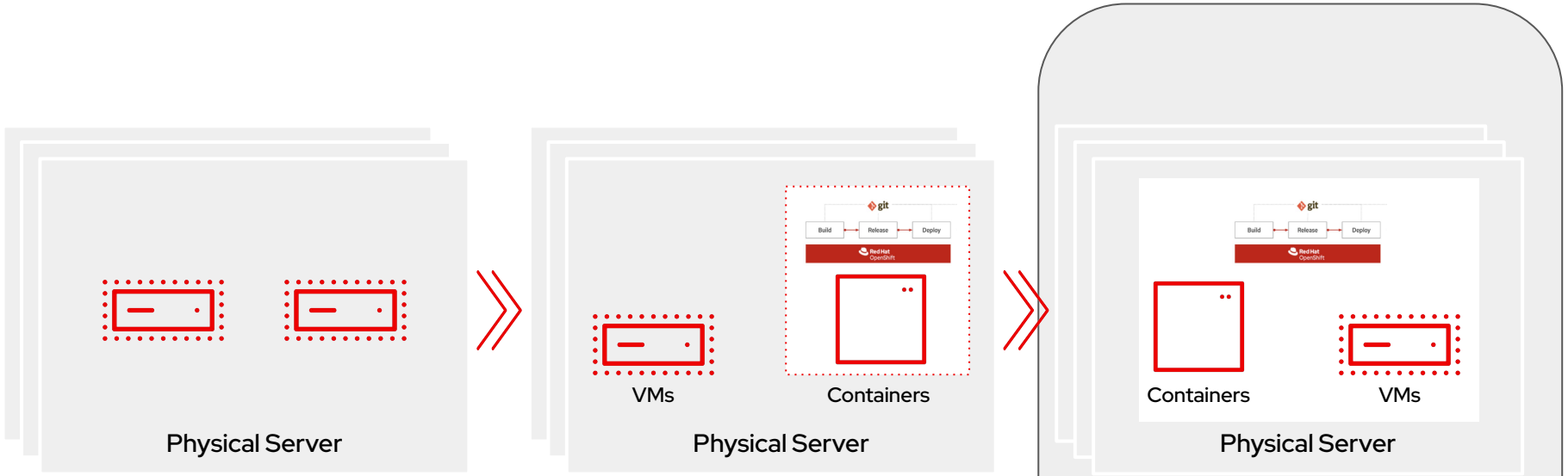
Kubernetes solves these challenges

- Innovate at speed
- Scalability
- Cost reduction for operating infrastructure
- Security focused
- Integrated development tools

Red Hat

Bring Cloud-native functionality to virtual machines with Red Hat innovation

The benefits of k8s without containerizing



Service uptime / easier maintenance

Segregated platforms to run modern applications

Unified application platform

OpenShift Virtualization

What is OpenShift Virtualization?

Leverage the Entire CNCF Project Ecosystem

KubeVirt*

133

Contributing Companies
CNCF Incubating project

CNCF Ecosystem Projects**

157

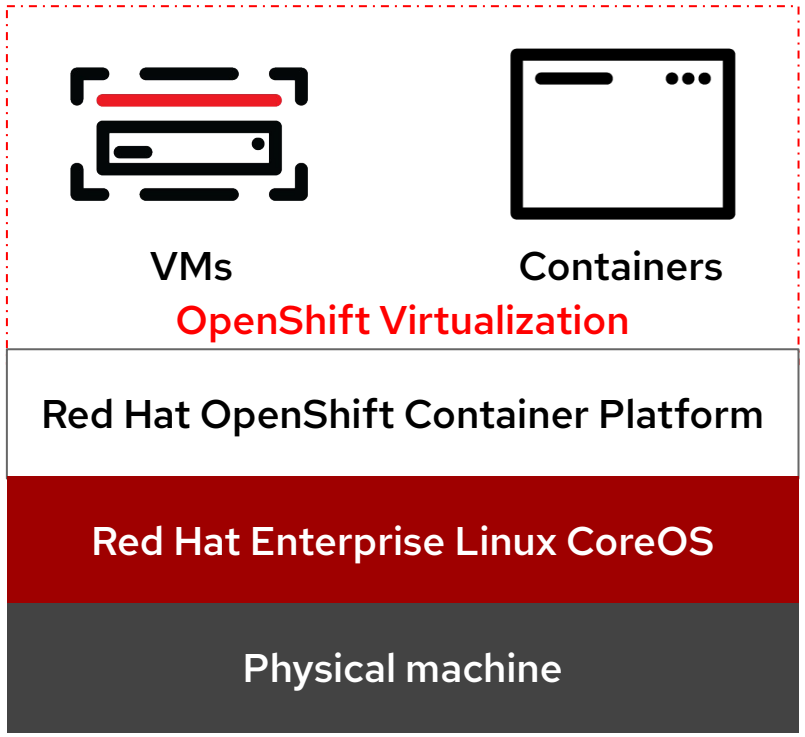
Observability & Alerting (Prometheus)
L2 Networking (Multus, Kube-OVN)
Service Meshes (Istio)
Automation (Tekton, ArgoCD)
Workload migration (Konveyor)

*https://www.cncf.io/blog/2022/04/19/kubevirt-becomes-a-cncf-incubating-project/
** https://www.cncf.io/reports/cncf-annual-report-2022/

Red Hat

OpenShift Virtualization is included with OpenShift

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

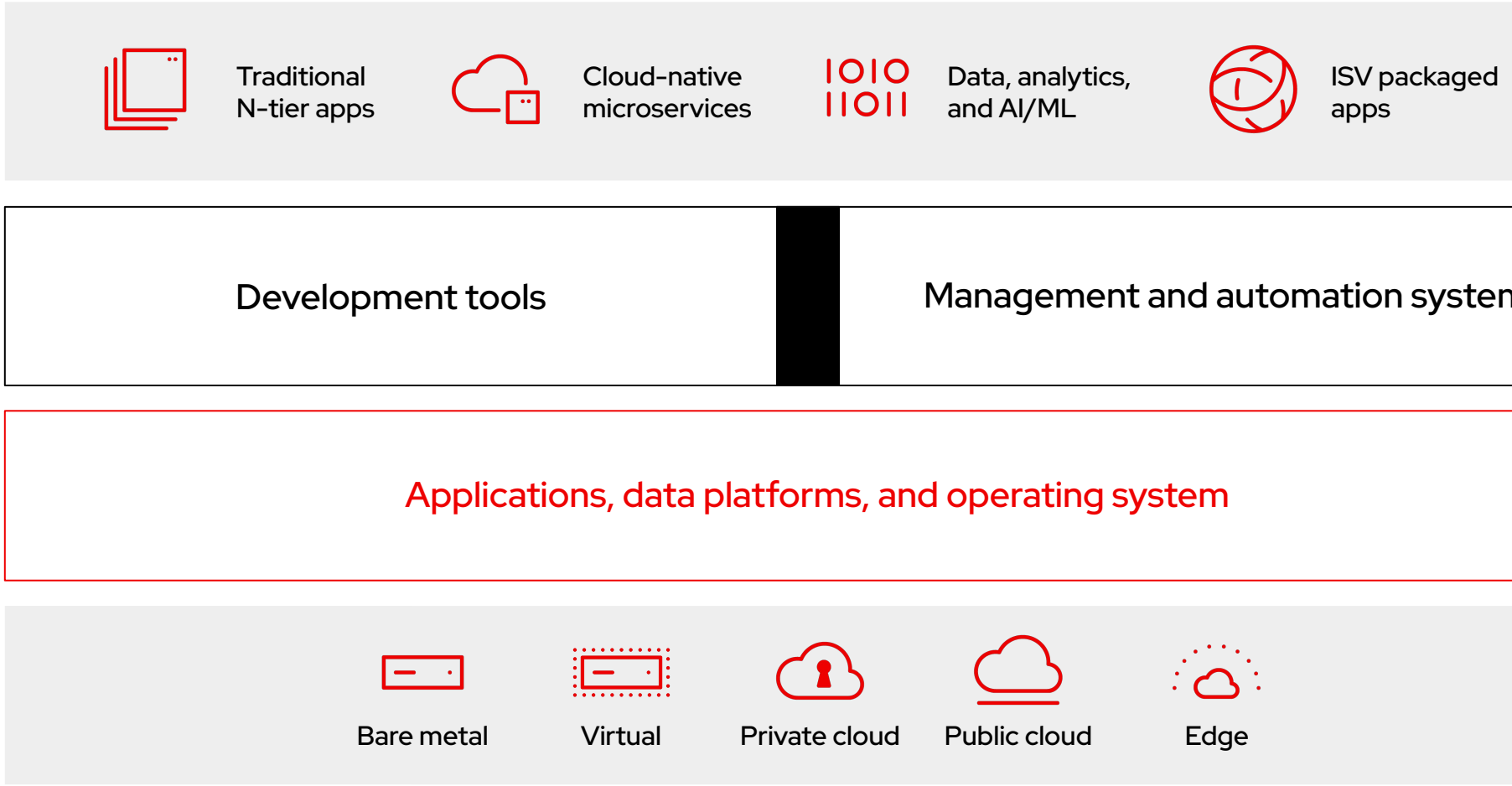


- Add VMs to new and existing applications
- Accelerates application delivery with a single platform that manages applications with same tools and teams
- Modernize legacy VM based applications over time, or maintain them as VMs
- OpenShift Virtualization is a feature of OpenShift and included in OCP/OPP/OKE SKU.

OpenShift 4.5+ includes OpenShift Virtualization

Red Hat

Remove Complexity by adopting a single platform

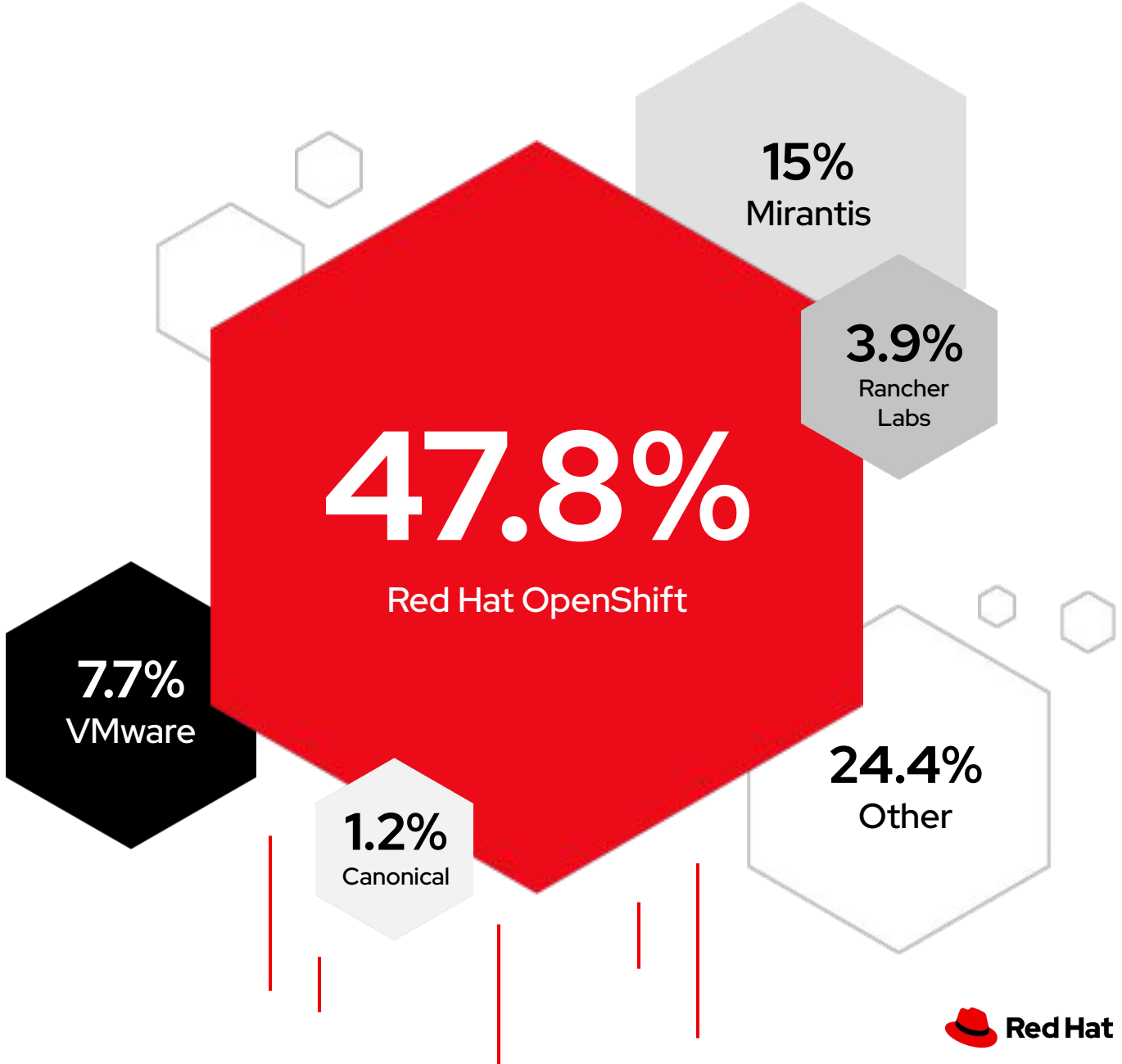


Red Hat

Red Hat OpenShift

Container platform

market share leader



Source: Who's Winning in the Container Software Market, IT Pro Today, Jun 29, 2021.

Red Hat

Realizing business value from adopting Red Hat OpenShift

636%
return on investment over 5 years

20%
higher application developer productivity

10 months
to payback

US$21.6 million
higher revenue per year per organization

54%
lower 5-year cost of operations

71%
less unplanned downtime

3x
more new features per year

21%
more efficient IT infrastructure teams

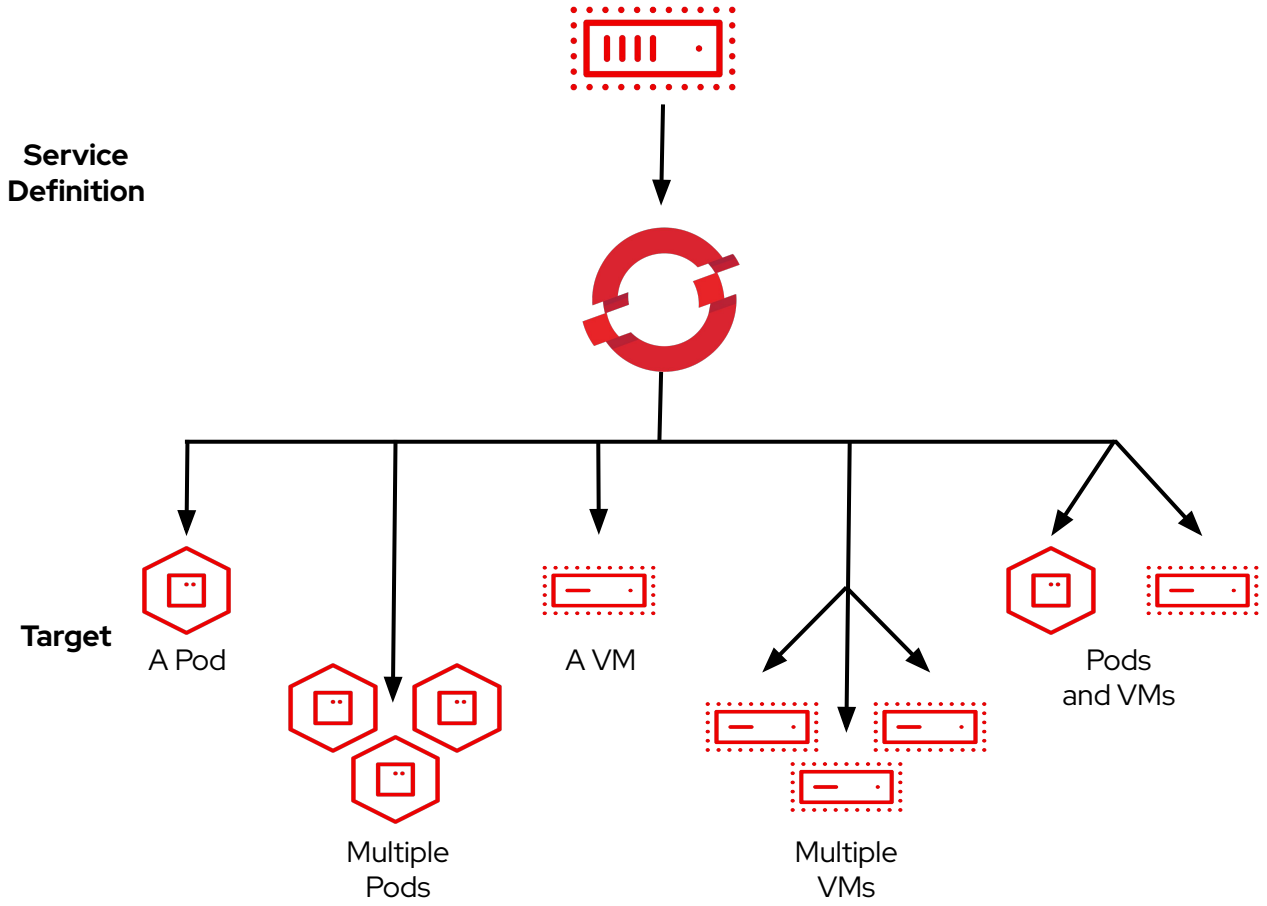
Source:

IDC White Paper, sponsored by Red Hat, "The Business Value of Red Hat OpenShift", doc # US47539121, February 2021.

Red Hat

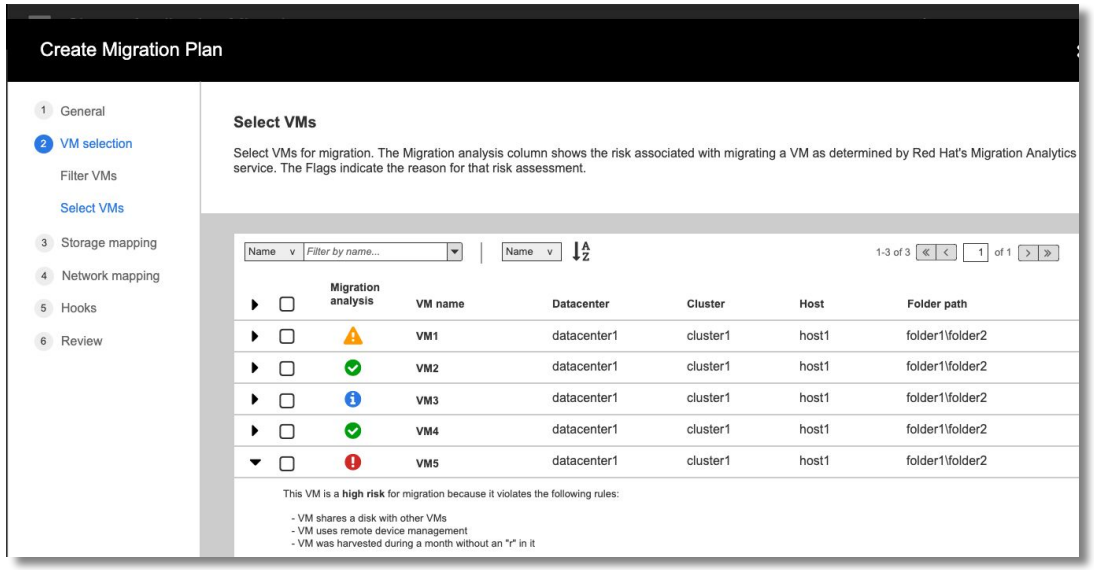
OpenShift Application-centric Technologies with Virtual Machines

- Virtual machines utilize OpenShift and Kubernetes functionality natively:
  - Service, Route
  - GitOps
  - Pipelines / Tekton
  - and others
- Containerized and virtualized app components don't know whether the other is virtual or containerized
- OpenShift Virtualization brings the benefits of Kubernetes without containerizing the application



Red Hat

OpenShift Virtualization: Bring traditional VMs to a modern Kubernetes platform



Create a migration plan with Red Hat Migration Toolkit for Virtualization (MTV)

Keep traditional VM behavior in a modern Kubernetes platform

- Administrator concepts and actions
- Network connectivity
- Live migration

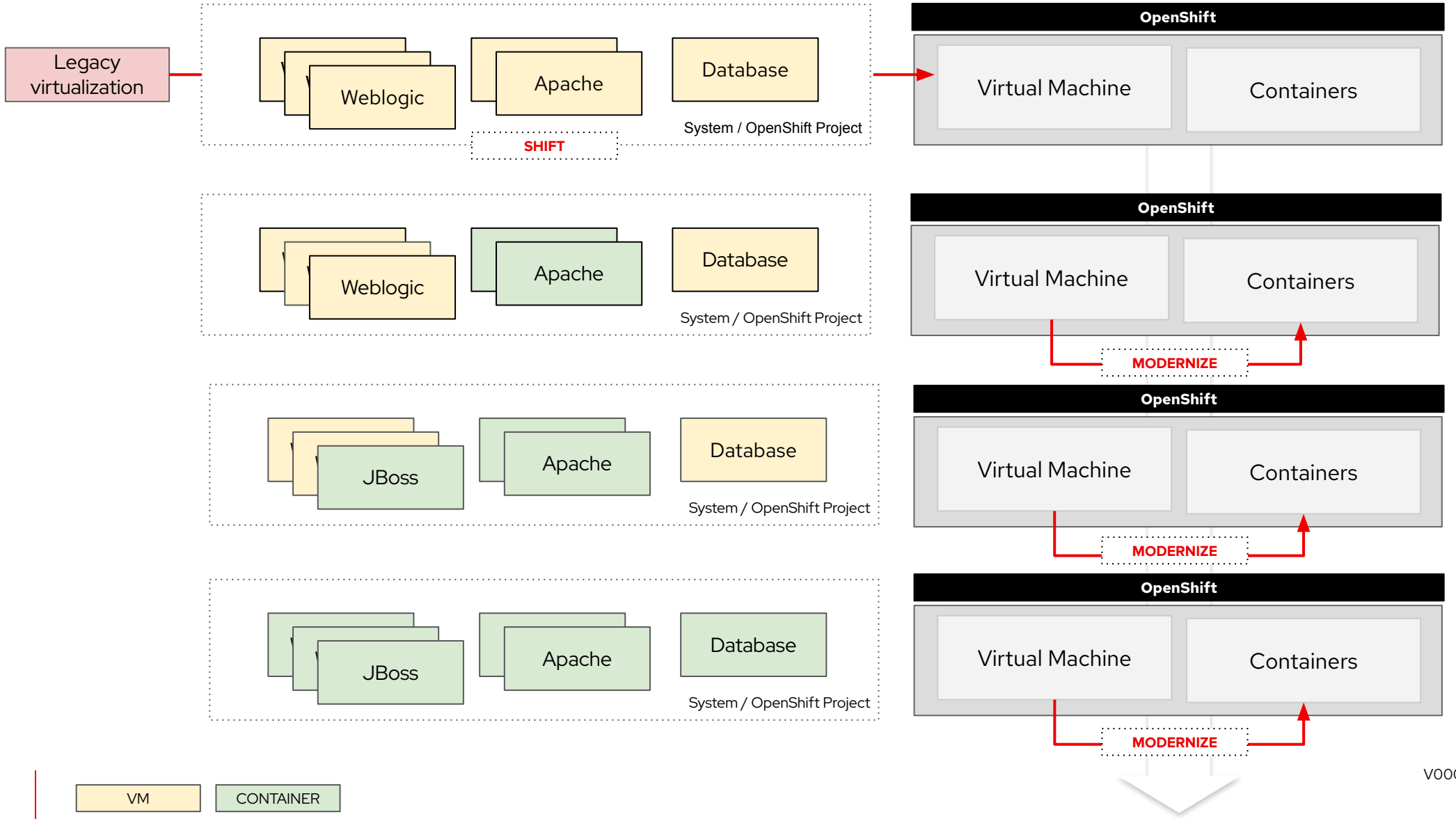
Keep existing roles and responsibilities intact

- Modernize skill sets over time, maintain business critical application components

Mass migration of VMs

- Simplified migration of VMs at scale with Migration Toolkit for Virtualization (MTV)

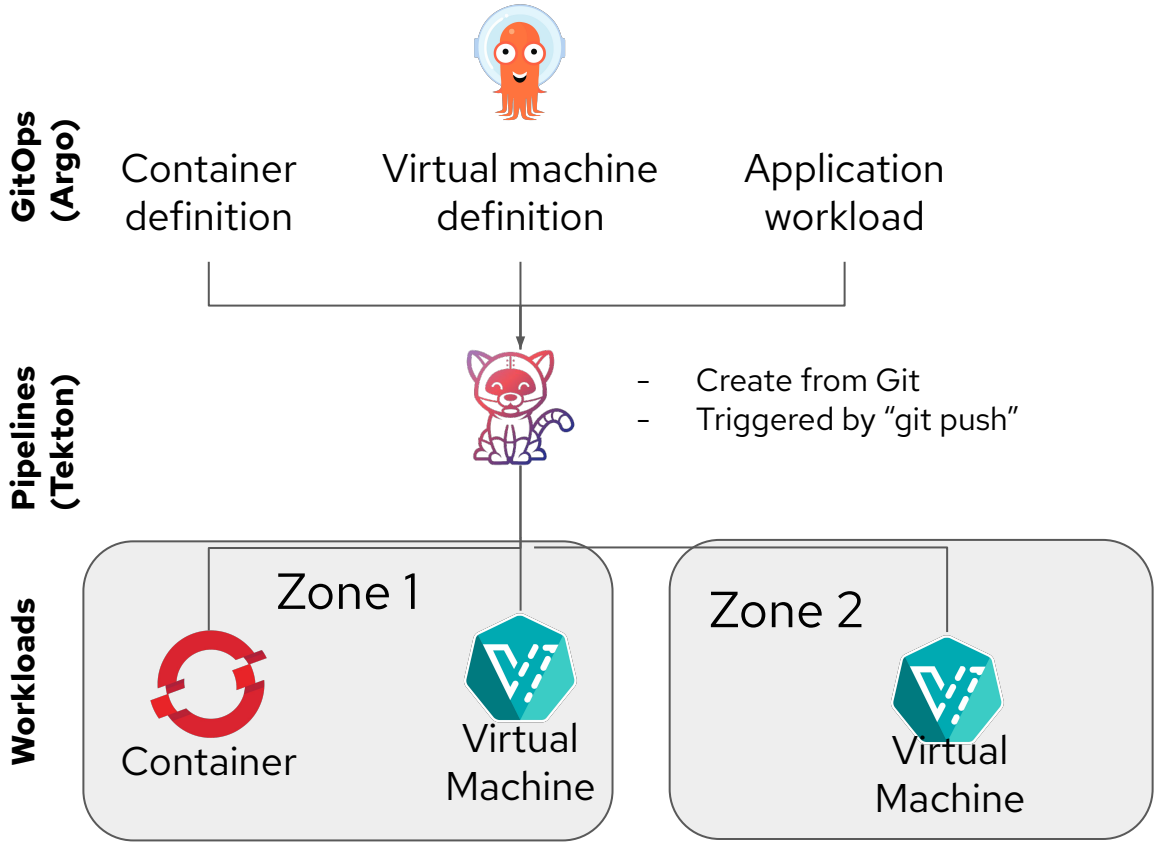
Enterprises Modernizing Applications Iteratively



Red Hat

OpenShift Virtualization:Build Cloud-native VMs

Deploy VMs as Code with CI/CD



Integrate legacy VMs with a modern GitOps framework

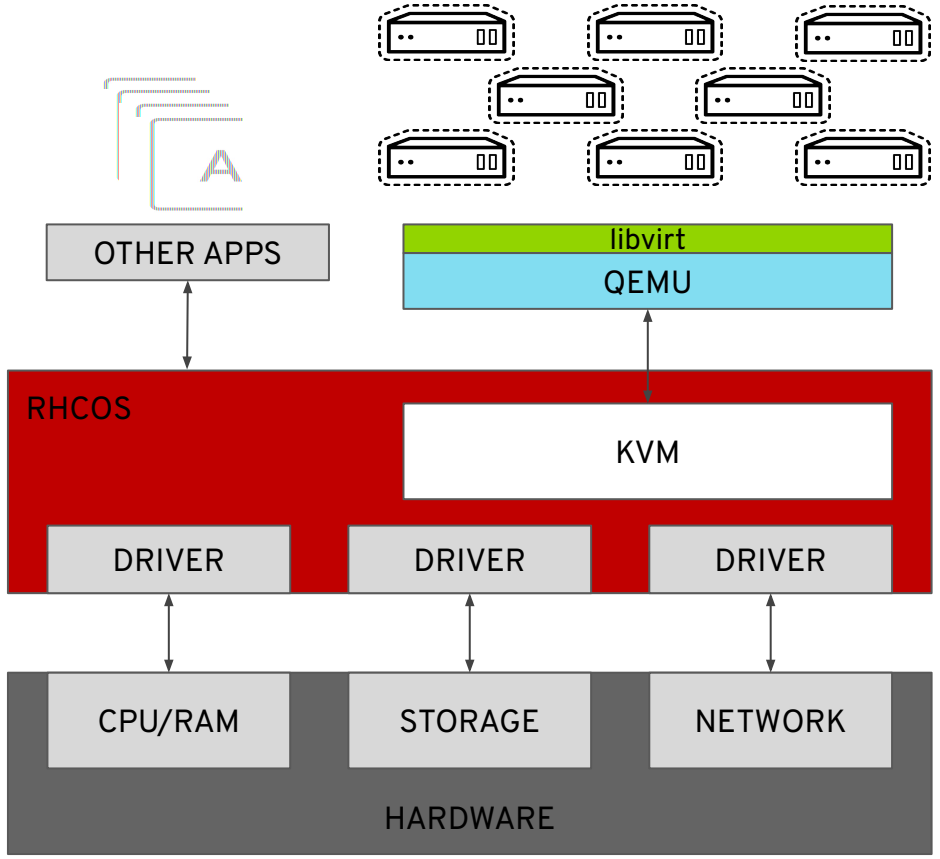
- Deploy different security zones to run both composite applications of pods/VMs as well as traditional VM workloads
- Deploy and automate Virtual Machines as Code with GitOps

Red Hat

OpenShift Virtualization Architectural Overview

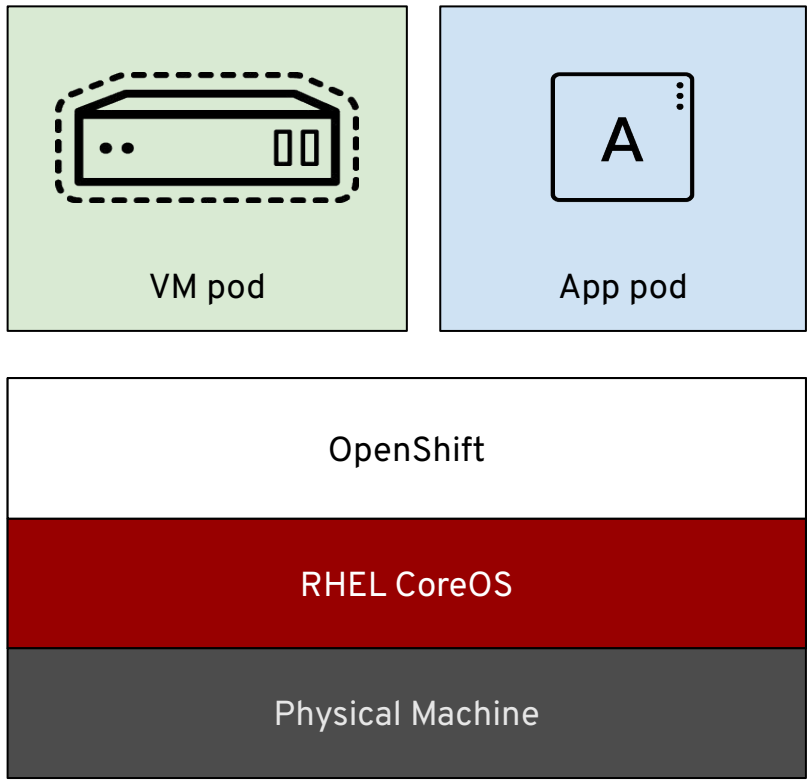
VM containers use KVM

- OpenShift Virtualization uses KVM, the Linux kernel hypervisor
- KVM is a core component of the Red Hat Enterprise Linux kernel
  - KVM has 10+ years of production use: Red Hat Virtualization, Red Hat OpenStack Platform, and RHEL all leverage KVM, QEMU, and libvirt
- QEMU uses KVM to execute virtual machines
- `libvirt` provides a management abstraction layer



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

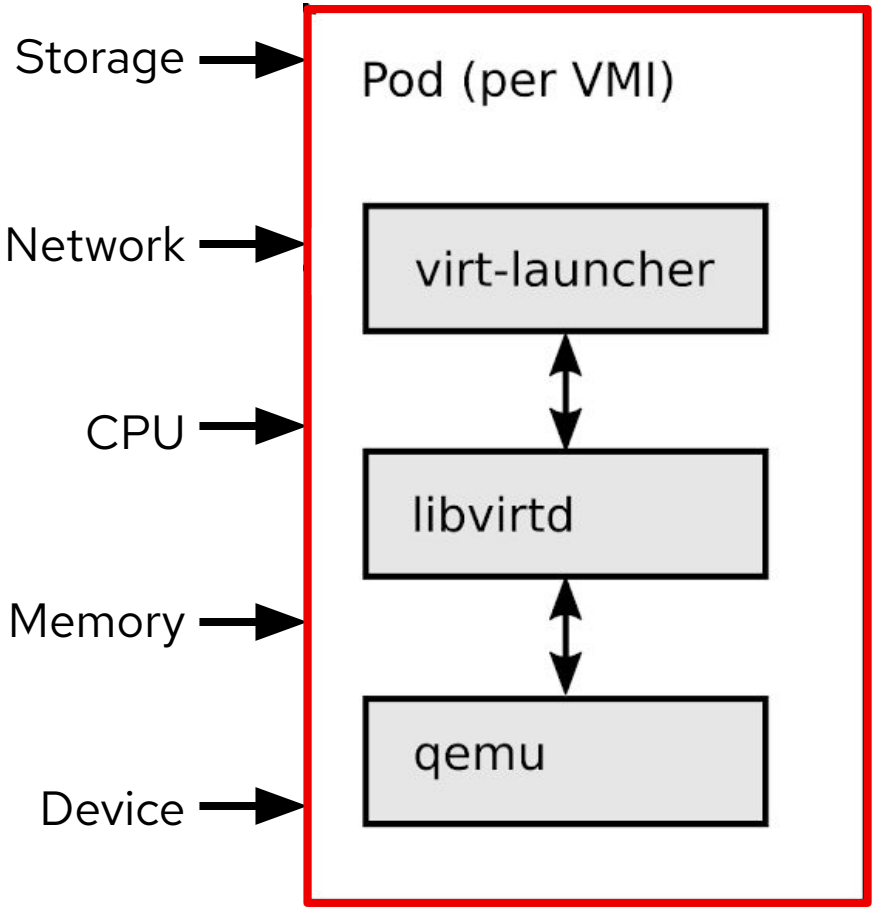


Virtualization native to Kubernetes

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

```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
```

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

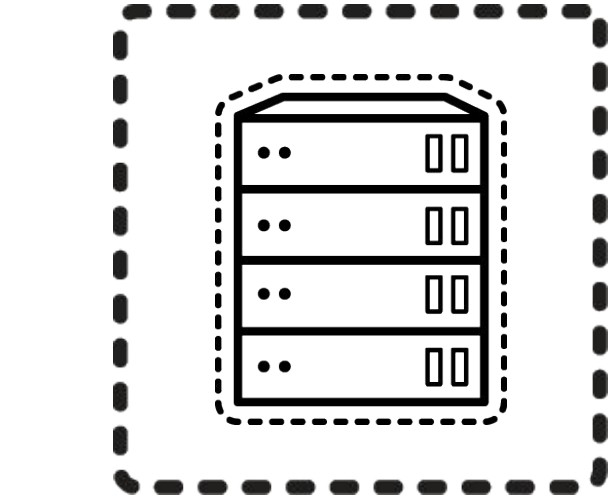
- libvirt and qemu from RHEL are mature, have high performance, provide stable abstractions, and have a minimal overhead.

Security - Defense in depth

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

Containerized virtual machines

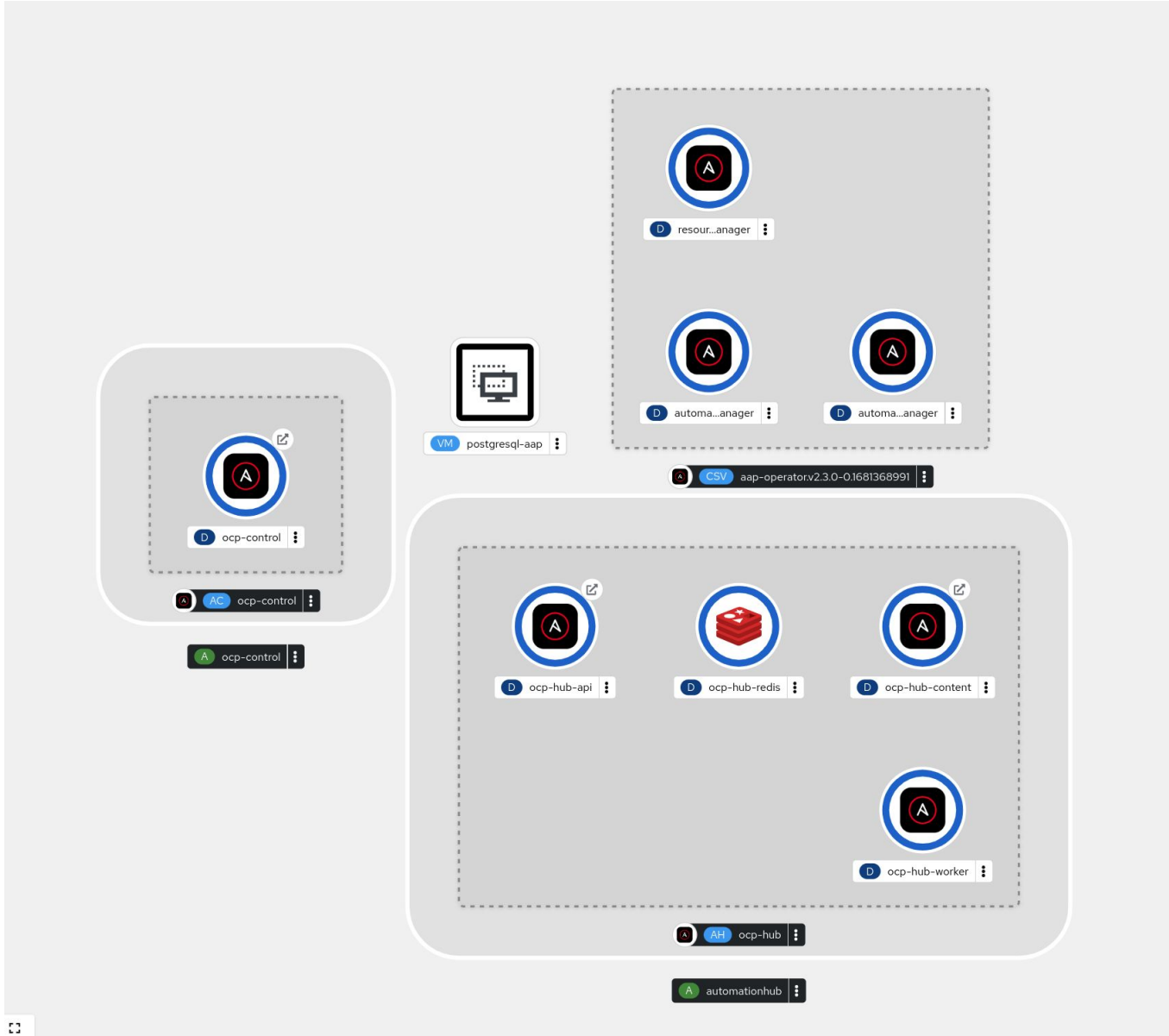
- Inherit many features and functions from Kubernetes
  - Scheduling, high availability, attach/detach resources
- Containerized virtual machines have the same characteristics as non-containerized
  - CPU, RAM, etc. limitations dictated by libvirt and QEMU
  - Linux and Windows guest operating systems
- 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



Red Hat OpenShift

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



Red Hat

Product and Customer Journey

Red Hat

Use Case and Product Journey

Financial Services

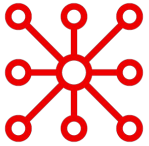
Adopt internal private cloud

Host multiple OpenShift Clusters

Persistent desktops

E-commerce

Modernize 3-tier applications - Adopt Kubernetes to adopt rapid lifecycle for .Net, Java, Linux applications



Telco

Roll out new mixed applications

VNF & CNF consolidation at the edge

Manufacturing/ Retail

Manufacturing floor automation

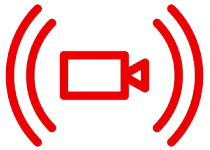
Store Infrastructure

Government/ Military

Adopt internal private cloud

Host multiple OpenShift Clusters

Tactical Edge with mix of Container/VM applications



Media/ Technology

Technical Workstations

Adopt internal private cloud

Red Hat

Retail: EMEA

sahibinden.com delivers reliable retail services faster with Red Hat OpenShift

"Red Hat OpenShift is the clear leader in enterprise Kubernetes. And while the virtualization market leaders can run Kubernetes on their virtualized infrastructure, only Red Hat OpenShift can run our whole virtualization environment within its Kubernetes container platform."

Hayri Yalçınkaya
Director of Infrastructure Management, sahibinden.com

Challenge

Sahibinden.com sought to modernize its IT infrastructure and work approaches to maintain its market-leading position against competition from start-ups and global retailers.

Solution

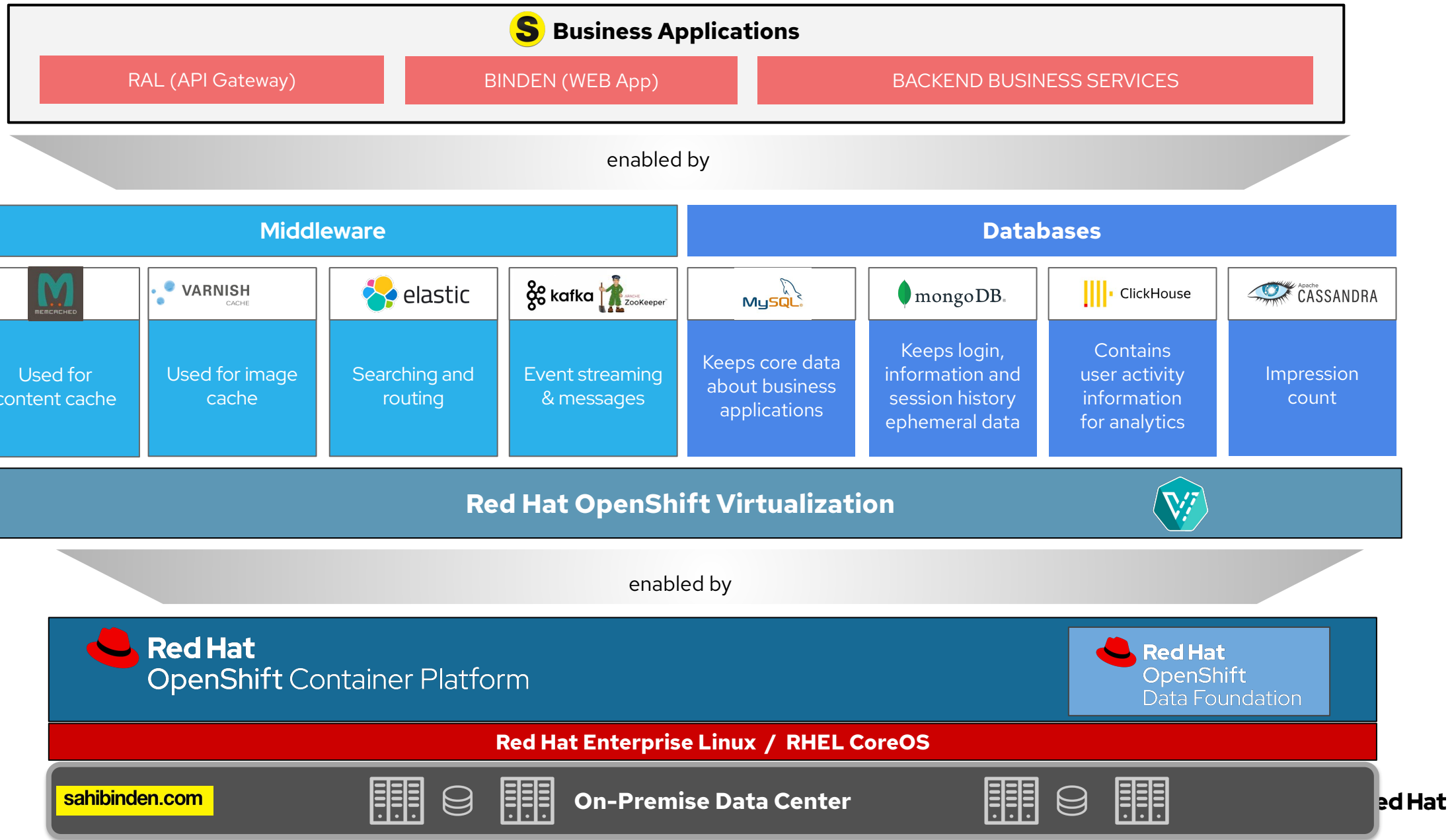
The company began a three-phase project to migrate its existing virtual machines (VMs) into container workflows with Red Hat® OpenShift®, running in a private cloud environment across its two datacenters.

Results

- **Decreased system reliability incidents by 97%** with active-active datacenter configuration and improved scalability
- **Improved DevOps experience** with combined VM and container management
- **Enhanced talent attraction and retention**
- Optimized container adoption with expert support and guidance

Source: Red Hat case study. "sahibinden.com delivers reliable retail services faster with Red Hat OpenShift," March 2022.

Applications running on OpenShift



Customer Journey

Assess workloads and devise migration strategy	Create Pilot Migration	Scale Migration	Results
2,400 VM's	Migrate initial batch of VM's	Migrate VM's at scale	Decreased system reliability incidents by 97%
Two datacenters, active-passive	Establish microservices development environment	Train staff and build skills	Combined VM and container management
		Modernize selected applications	Two datacenters, active-active

Red Hat

Edge

Challenge

Remote management
Space and Power constraints
Specialized workloads may require VMs e.g. running VNFs,
Sensors based on Windows, retail stores, remote sites

Solution

Utilize OpenShift to host both virtual machines (VMs) and containers utilizing compact cluster or Single Node OpenShift (SNO)

Results

100s of sites consolidated to run on minimum amount of hardware
Accelerate solution by several years using non-containerized workloads

Red Hat

Global Telecommunications / 5G Provider

Rolled out a greenfield large scale application at the core where some components had to be in VMs. Architect, develop, test, and scale *production-like* deployments many months before a containerized version of applications being available.

- MySQL Cluster Carrier Grade Edition from Oracle
- F5 BIG-IP , initially deployed as VM, refactoring to a container

“Significantly accelerate time to market, while ensuring reliability and quality of the overall solution.”

Red Hat

OpenShift Virtualization two years of demonstrated ability

Enterprise Virtualization Capabilities

- Live Migration
- Infrastructure fencing
- Application fencing
- DRS
- OADP for backup
- Crash consistent snapshots
- Hot pluggable disks
- Storage profiles
- CPU overcommit
- VM disk resize
- DPDK, SRI-IOV, IPv6
- GPU passthrough, vGPU
- UI for VM admins
- VM export

Enterprise Scale

- Performance and limits parity
- Scale whitepaper 3K VMs/21K containers in a single cluster

Public Cloud

- AWS tech preview
- IBM Cloud Bare Metal Tech Preview
- Equinix validation

Telco/ Edge Integrations

- SNO support
- VNF validation
- Latency self-check
- Compact Cluster Infrastructure fencing* (4.12)

High Level Roadmap Themes*

Enterprise Virtualization Capabilities

- Metro (Sync) DR with ODF
- Regional (Async) DR with ODF
- Ecosystem DR integrations
- Overlay Secondary network
- Network Hotplug
- CPU Hotplug
- Memory Overcommit

Enterprise Scale

- Scale to 10K VMs in a single cluster
- vNUMA
- Workload aware scheduler

Multi-Cluster Scaling

- OCP Virt as Hosted Control Plane (HyperShift) Provider
- ACM VM lifecycle and workflow

OpenShift Developer Services

- Tekton and ArgoCD integration for VMs
- Windows 11 and Windows Server 2022 examples.
- Ansible integration
- Gateway API for load balancing

* Roadmap items subject to change

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

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat

Comparing with traditional virtualization platforms

Red Hat

Terminology comparison

Feature	RHV	OpenShift Virtualization	vSphere
Where VM disks are stored	Storage Domain	PVC	datastore
Policy based storage	None	StorageClass	SPBM
Non-disruptive VM migration	Live migration	Live migration	vMotion
Non-disruptive VM storage migration	Storage live migration	N/A	Storage vMotion
Active resource balancing	Cluster scheduling policy	Pod eviction policy, descheduler	Dynamic Resource Scheduling (DRS)
Physical network configuration	Host network config (via nmstate w/4.4)	nmstate Operator, Multus	vSwitch / DvSwitch
Overlay network configuration	OVN	OCP SDN (OpenShiftSDN, OVNKubernetes, and partners), Multus	NSX-T
Host / VM metrics	Data warehouse + Grafana (RHV 4.4)	OpenShift Metrics, health checks	vCenter, vROps

Live Migration

- Live migration moves a virtual machine from one node to another in the OpenShift cluster
- Can be triggered via GUI, CLI, API, or automatically
- **RWX storage is required**
- Live migration is cancellable by deleting the API object
- Default maximum of five (5) simultaneous live migrations
  - Maximum of two (2) outbound migrations per node, 64MiB/s throughput each

Migration Reason	vSphere	OpenShift Virtualization
Resource contention	DRS	Pod eviction policy, pod descheduler
Node maintenance	Maintenance mode	Maintenance mode, node drain

Automated live migration

- OpenShift / Kubernetes triggers Pod rebalance actions based on multiple factors
  - Soft / hard eviction policies
  - Pod descheduler
  - Pod disruption policy
  - Node resource contention resulting in evictions
    - Pods are `Burstable` QoS class by default
    - All memory is requested in Pod definition, only CPU overhead is requested
- Pod rebalance applies to VM pods equally
- VMs will behave according to the eviction strategy
  - `LiveMigrate` - use live migration to move the VM to a different node
  - No definition - terminate the VM if the node is drained or Pod evicted

VM scheduling

- VM scheduling follows pod scheduling rules
  - Node selectors
  - Taints / tolerations
  - Pod and node affinity / anti-affinity
- Kubernetes scheduler takes into account many additional factors
  - Resource load balancing - requests and reservations
  - Large / Huge page support for VM memory
  - Use scheduler profiles to provide additional hints (for all Pods)
- Resources are managed by Kubernetes
  - CPU and RAM `requests` less than `limit` - `Burstable` QoS by default
  - K8s QoS policy determines scheduling priority: `BestEffort` class is evicted before `Burstable` class, which is evicted before `Guaranteed` class

Node Resource Management

- VM density is determined by multiple factors controlled at the cluster, OpenShift Virtualization, Pod, and VM levels
- Pod QoS policy
  - Burstable (limit > request) allows more overcommit, but may lead to more frequent migrations
  - Guaranteed (limit = request) allows less overcommitment, but may have less physical resource utilization on the hosts
- Cluster Resource Override Operator provides global overcommit policy, can be customized per project for additional control
- Pods request full amount of VM memory and approx. 10% of VM CPU
  - VM pods request a small amount of additional memory, used for libvirt/QEMU overhead
    - Administrator can set this to be overcommitted

High availability

- Node failure is detected by Kubernetes and results in the Pods from the lost node being rescheduled to the surviving nodes
- VMs are not scheduled to nodes which have not had a heartbeat from `virt-handler`, regardless of Kubernetes node state
- Additional monitoring may trigger automated action to force stop the VM pods, resulting in rescheduling
  - May take up to 5 minutes for `virt-handler` and/or Kubernetes to detect failure
  - Liveness and Readiness probes may be configured for VM-hosted applications
  - Machine health checks can decrease failure detection time