



Connect

Exploring Latest Features in OpenShift Virtualization

Gökhan Göksu

Senior Solution Architect

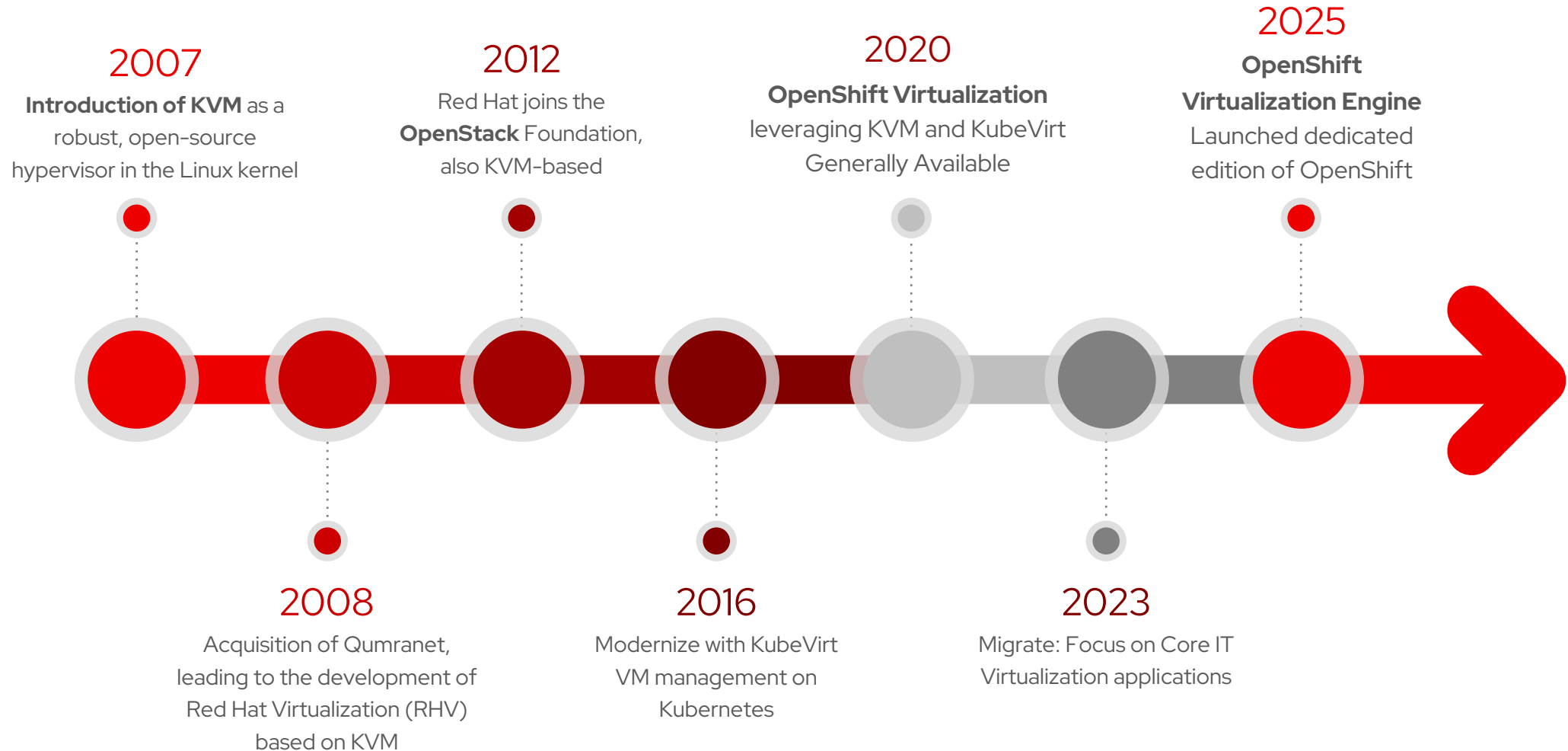
gokhan@redhat.com

www.linkedin.com/in/ggoksu



Red Hat has a long history with Virtualization

Open Source driving KVM, RHEL, OpenStack and now OpenShift Virtualization



What We Hear From the Field



“I want to modernize”

- Wants to modernize to containers, but also run **VMs in a more modern way**
- Stand up a **secondary** virtualization platform for new workloads
- Legacy and next-gen virtualization platforms **co-exist**



“I need to migrate”

- **Migrate** off their current traditional virtualization platform completely, as quickly and as safely as possible
- **Modernization is subordinate to migration**; containers and Kubernetes are implementation details
- Willing to take **calculated risk** with their production workloads

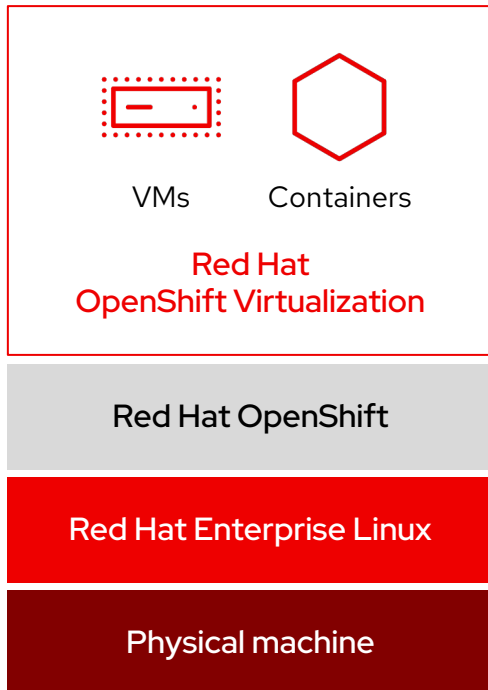
OpenShift Virtualization

- Virtual machines
 - Running in **containers**, managed as Pods
 - Using the **KVM** hypervisor
- Scheduled, deployed, and managed by **Kubernetes**
- Integrated with container orchestrator resources and services
 - Traditional Pod-like **network connectivity** and/or external VLAN
 - **Persistent storage** with PVC, PV, StorageClass



Red Hat OpenShift Virtualization

The modern option for general purpose virtualization

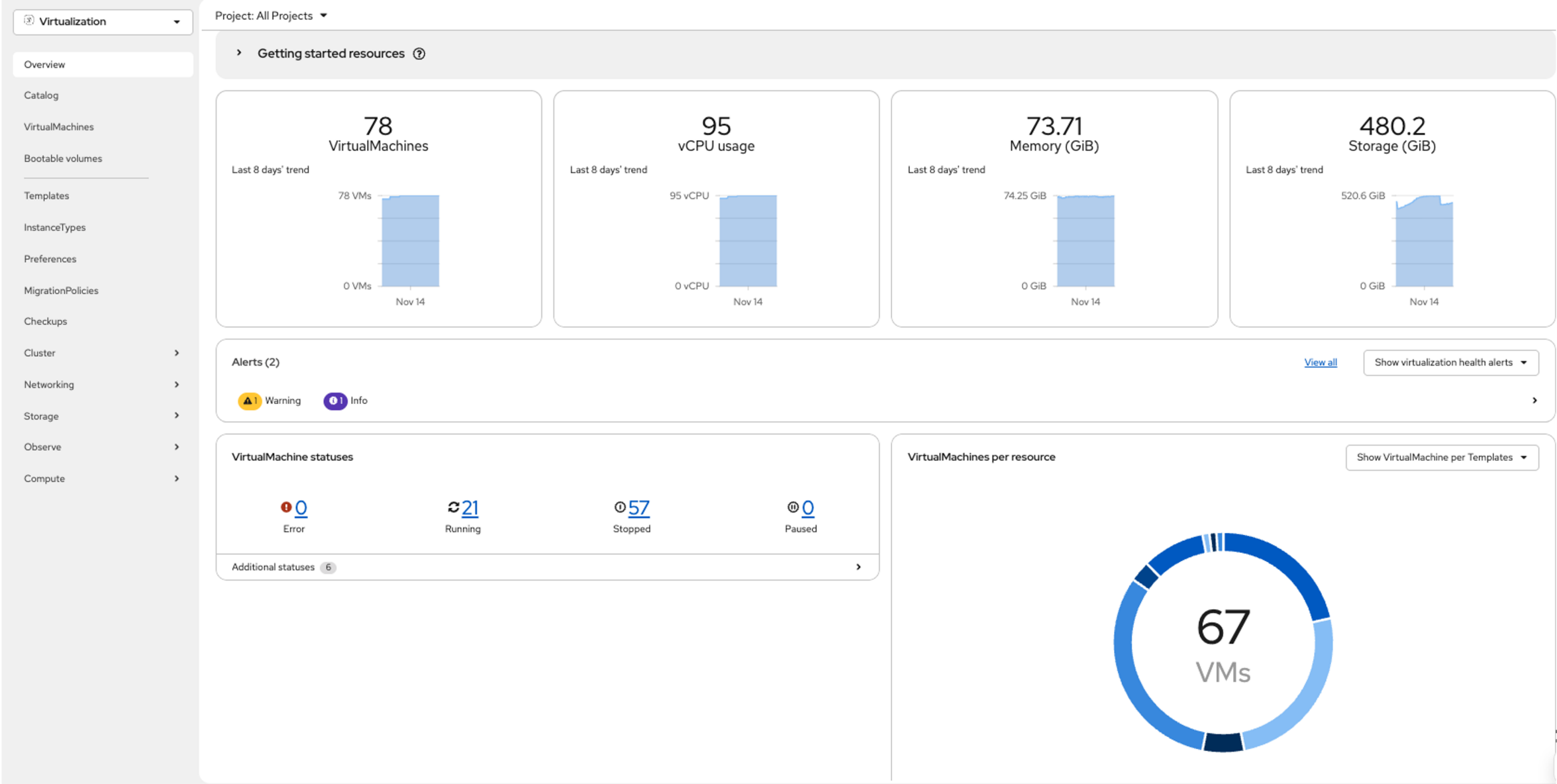


- ▶ **Unified platform**
for virtual machines and containers
- ▶ **Consistent management**
tools, interfaces, and APIs incl. ACM and AAP integrations
- ▶ **Performance and stability**
of Linux, KVM, and qemu
- ▶ **Healthy open source community**
the KubeVirt project is a top 10 CNCF active project, with 200+ contributing companies
- ▶ **Diverse ecosystem**
of Red Hat & partner operators
- ▶ **Included feature**
of all OpenShift subscriptions
- ▶ **Includes Red Hat Enterprise Linux**
guest entitlements
- ▶ **Supports Microsoft Windows**
guests through Microsoft SVVP
- ▶ **Guest VM migration**
using Ansible + Migration Toolkit for Virtualization, Training and Consulting
- ▶ **Virt admin focused training**

OpenShift Virtualization Features



Dedicated VM Admin View



Designed for the Virtual Infrastructure Admin

VirtualMachines

Show only projects with VirtualMachines ☒

Projects [+ Create Project](#)

All projects 5

alan-vms 1

- fedora-scarlet-shrimp-79

mark-dev1 2

- centos-stream9-brown-rattlesnake-79
- rhel9-aqua-blackbird-51

sully-vms 2

- rhel-9-amethyst-nightingale-28
- rhel9-indigo-sloth-11

All projects summary

Virtual Machines (5)

0 Error

5 Running

0 Stopped

0 Paused

Filter

Projects All

Name

Search by name...

	Name ↑	Namespace ↑	Status ↑
<input type="checkbox"/>	VM fedora-scarlet-shrimp-79	NS alan-vms	Running
<input type="checkbox"/>	VM centos-stream9-brown-rattlesnake-79	NS mark-dev1	Running
<input type="checkbox"/>	VM rhel9-aqua-blackbird-51	NS mark-dev1	Running
<input type="checkbox"/>	VM rhel-9-amethyst-nightingale-28	NS sully-vms	Running

Simplified Management

- Multi-cluster Tree-view
- Optimize cluster for Virtualization with recommended operators
- OpenShift Virtualization installation in Disconnected and Registry-less (TP)
- New VM metrics and alerts for CPU and storage latency

The screenshot displays the OpenShift Virtualization console interface. On the left, a tree view shows the project hierarchy: 'All clusters' > 'virt-hub' > 'ugo' > 'fedora-fuchsia-swan-76', 'radek', 'default' (containing 'example-1', 'fedora-jade-pike-13', 'rhel-10-lavender-butterfly-16', 'rhel7-apricot-anteater-91', 'rhel9-salmon-blackbird-17'), 'virt-spoke' > 'default' (containing 'example-1', 'example-2', 'rhel-8-jade-louse-14'). The main panel shows the 'virt-spoke (cluster)' view with a summary of 5 Virtual Machines: 0 Error, 1 Running, 4 Stopped, and 0 Paused. Below this, usage statistics are shown: CPU (4.12 m, Requested of 1), Memory (356.9 MiB, Used of 4 GiB), and Storage (2.87 GiB, Used of 29.99 GiB). A table lists the VMs with columns for Name, Namespace, Status, Conditions, Node, and IP address. The table shows three VMs: 'example-spoke-1' (Stopped), 'example-spoke-2' (Stopped), and 'rhel-8-jade-louse-14' (Running).

Name	Namespace	Status	Conditions	Node	IP address
example-spoke-1	default	Stopped	Ready=Failed	-	-
example-spoke-2	default	Stopped	Ready=Failed	-	-
rhel-8-jade-louse-14	default	Running	DataVolume=Ready	ip-10-0-14	10.128.3.209

Virtual Machine Details

- Control the state and status of the virtual machine
- Actions menu allows quick access to common VM tasks
 - Start/stop/restart
 - Live migration
 - Clone
 - VM Console
 - Delete
- Change VM Settings like CPU, Memory, Network and Disks

The screenshot displays the OpenShift VirtualMachines console interface. The top navigation bar shows 'VirtualMachines' and 'VirtualMachine details'. Below this, a header for the selected VM 'database' indicates it is 'Running'. A tabbed interface includes 'Overview' (selected), 'Details', 'Metrics', 'YAML', 'Configuration', and 'Events'.

The 'Details' tab contains a table with the following information:

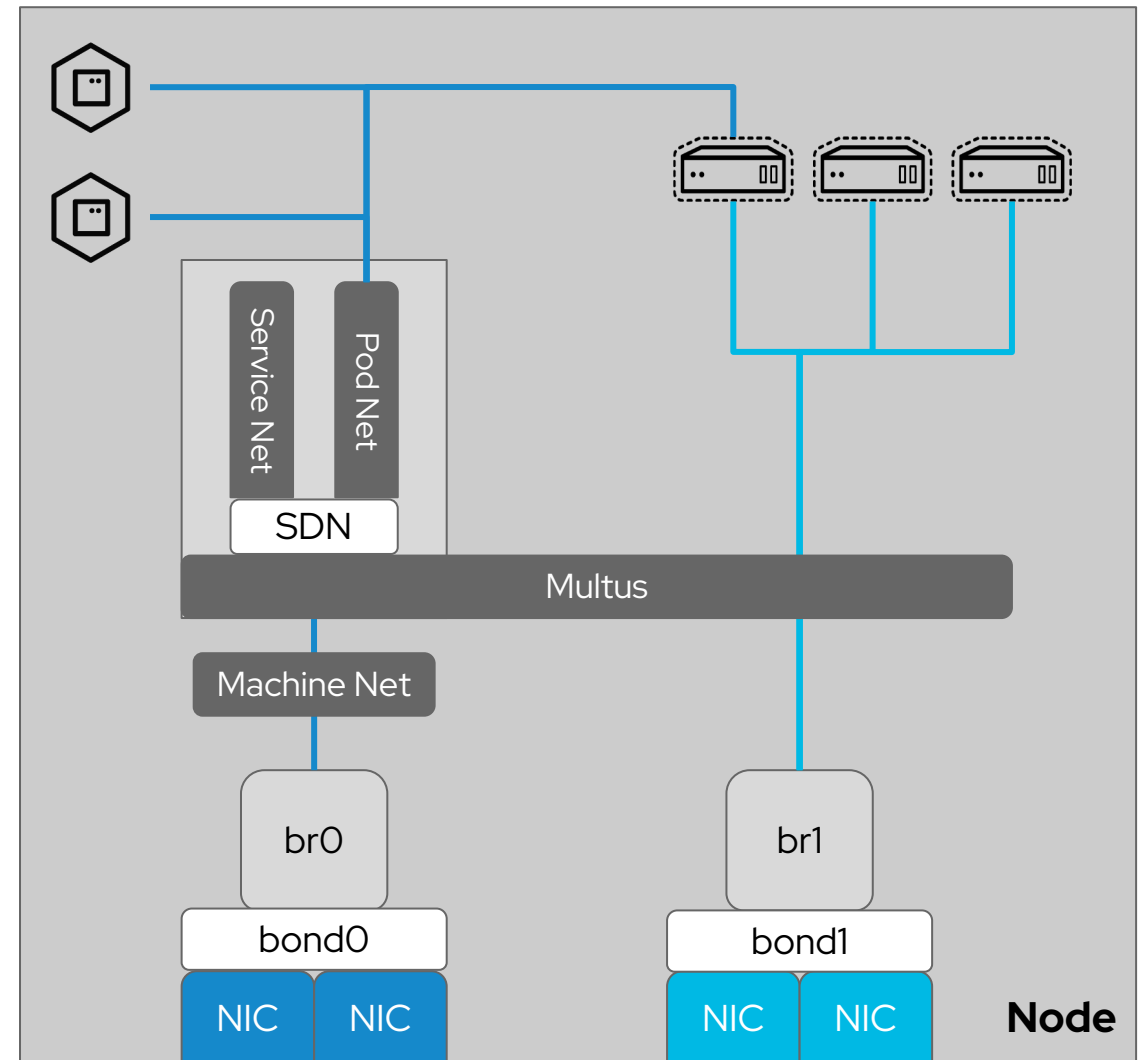
Name	database
Status	Running
Created	1 day ago
Operating system	CentOS Linux 7 (Core)
CPU Memory	1 CPU 2 GiB Memory
Hostname	database
Template	None

To the right of the table is a 'VNC console' window showing a terminal output. Below the table is a link 'Open web console' with an external link icon.

An 'Actions' dropdown menu is open, showing a list of operations: Stop, Restart, Pause, Clone, Migrate (with a sub-option 'Migrate to a different Node'), Copy SSH command (with a sub-option 'SSH using virtctl'), Edit labels, Edit annotations, and Delete. Below the menu, the 'IP address' is displayed as '10.128.2.43' with a copy icon.

Example Host Network Configuration

- Pod, service, and machine network are configured by OpenShift automatically
- Use the NMstate to configure additional host network interfaces
 - **bond1** and **br1** in the example to the right
- VMs and Pods connect to one or more networks simultaneously



GUI-based host network configuration

- Apply NMstate configuration using a **form** in the OpenShift admin console
- Create and configure
 - Ethernet interface IP (static, DHCP)
 - Bonds - mode 1-6 bonds with options, including IP configuration
 - Linux bridge configuration utilizing ethernet and/or bonds for “uplinks”
- Specify **node selectors** to have configuration automatically applied to matching nodes

The screenshot displays the Red Hat OpenShift Admin Console interface. On the left is a navigation sidebar with categories like Administrator, Home, Operators, Workloads, Virtualization, Migration, Networking (expanded), Storage, Builds, Observe, Compute, User Management, and Administration. The 'Networking' section is active, showing sub-items: Services, Routes, Ingresses, NetworkPolicies, NetworkAttachmentDefinitions, NodeNetworkConfigurationPolicy (selected), and NodeNetworkState.

The main panel is titled 'Create NodeNetworkConfigurationPolicy' with an 'Edit YAML' link. It contains the following fields and options:

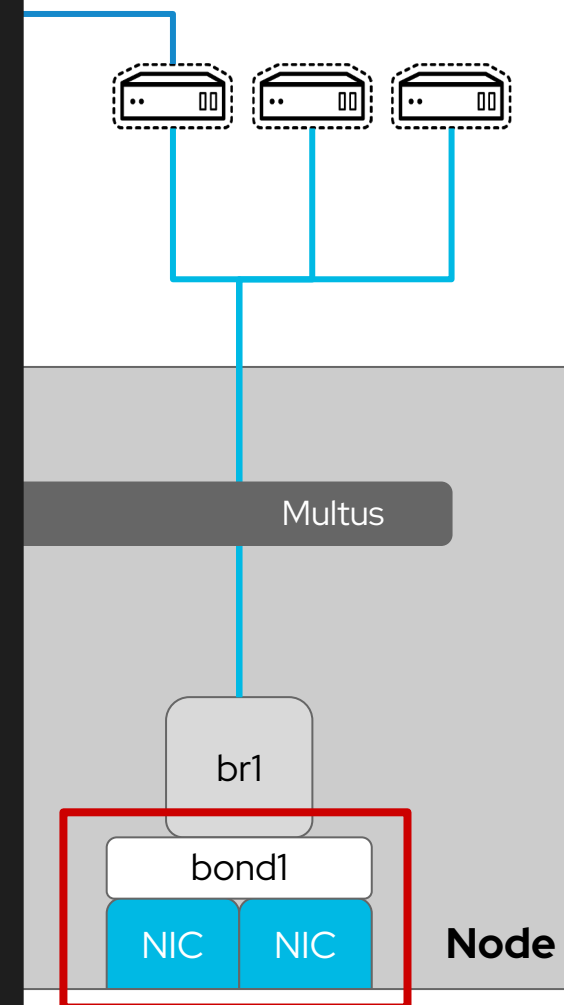
- Apply this NodeNetworkConfigurationPolicy only to specific subsets of nodes using the node selector:** A checkbox that is checked.
- Policy name:** A text input field containing 'worker-bond0-br1'.
- Description:** A large text area.
- Policy Interface(s):** A section with a link to 'Add another interface to the policy'. It contains two expandable sections:
 - Bonding bond0:** A section header.
 - Bridge br1:** An expandable section containing:
 - Interface name:** A text input field containing 'br1'.
 - Network state:** A dropdown menu set to 'Up'.
 - Type:** A dropdown menu set to 'Bridge'.
 - IP configuration:** A checkbox for 'IPv4' which is unchecked.
 - Port:** A text input field containing 'bond0'.
 - Enable STP:** An unchecked checkbox.

At the bottom of the form is a blue 'Create' button.

Host bond configuration

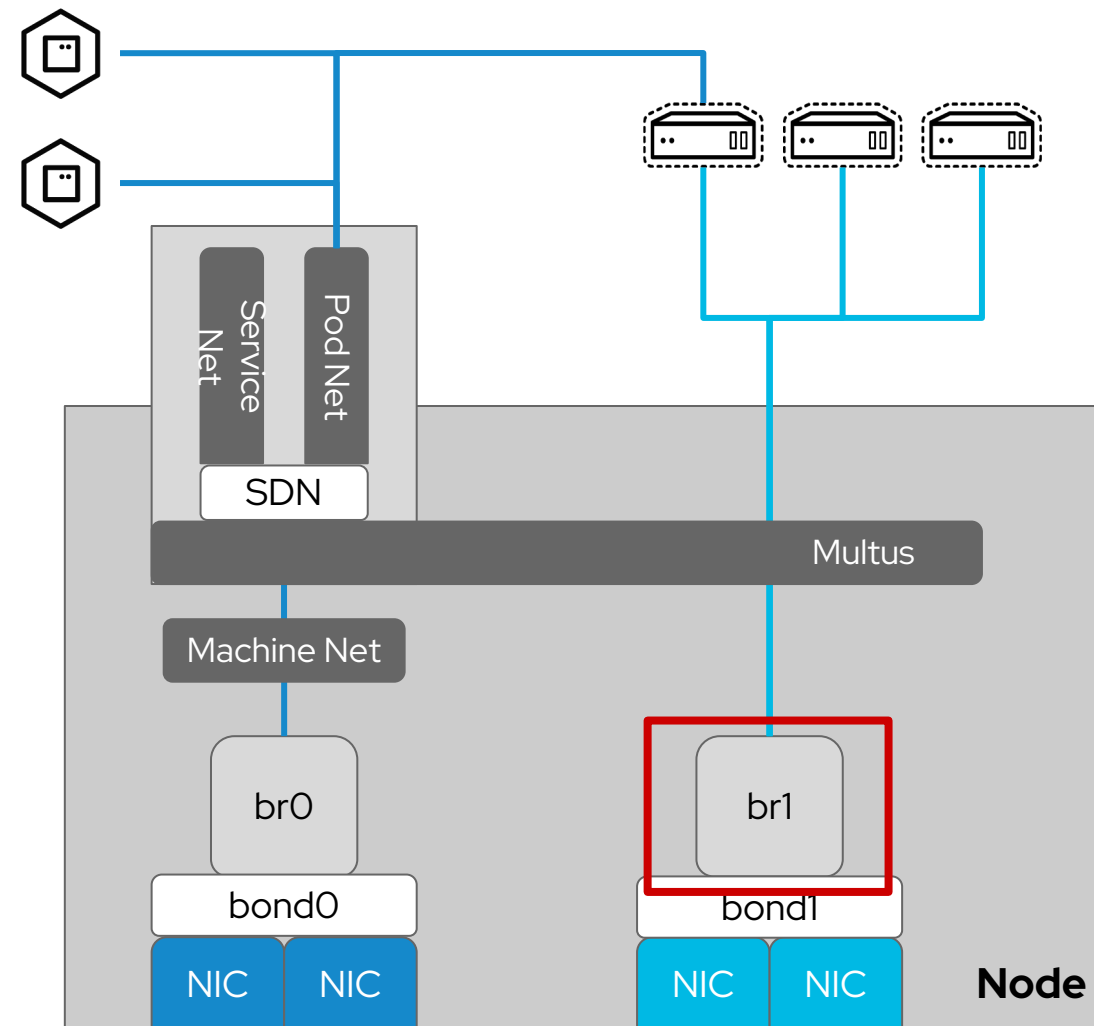
- NodeNetworkConfigurationPolicy (NNCP)
 - Nmstate operator CRD
 - Configure host network using declarative language
- Applies to all nodes specified in the nodeSelector, including newly added nodes automatically
- Update or add new NNCPs for additional host configs

```
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
```



Host bridge configuration

```
1  apiVersion: nmstate.io/v1alpha1
2  kind: NodeNetworkConfigurationPolicy
3  metadata:
4    name: worker-bond1-br1
5  spec:
6    nodeSelector:
7      node-role.kubernetes.io/worker: ""
8    desiredState:
9      interfaces:
10       - name: br1
11         description: br1 with bond1
12         type: linux-bridge
13         state: up
14         ipv4:
15           enabled: false
16         bridge:
17           options:
18             stp:
19               enabled: false
20           port:
21             - name: bond1
```



GUI-based Network Attachment Definition

Project: default ▼

Create NetworkAttachmentDefinition

Configure via: ☒ Form view ☐ YAML view

Name * ?

nad

Description

example-nad

Network Type *

Network Type

Linux bridge

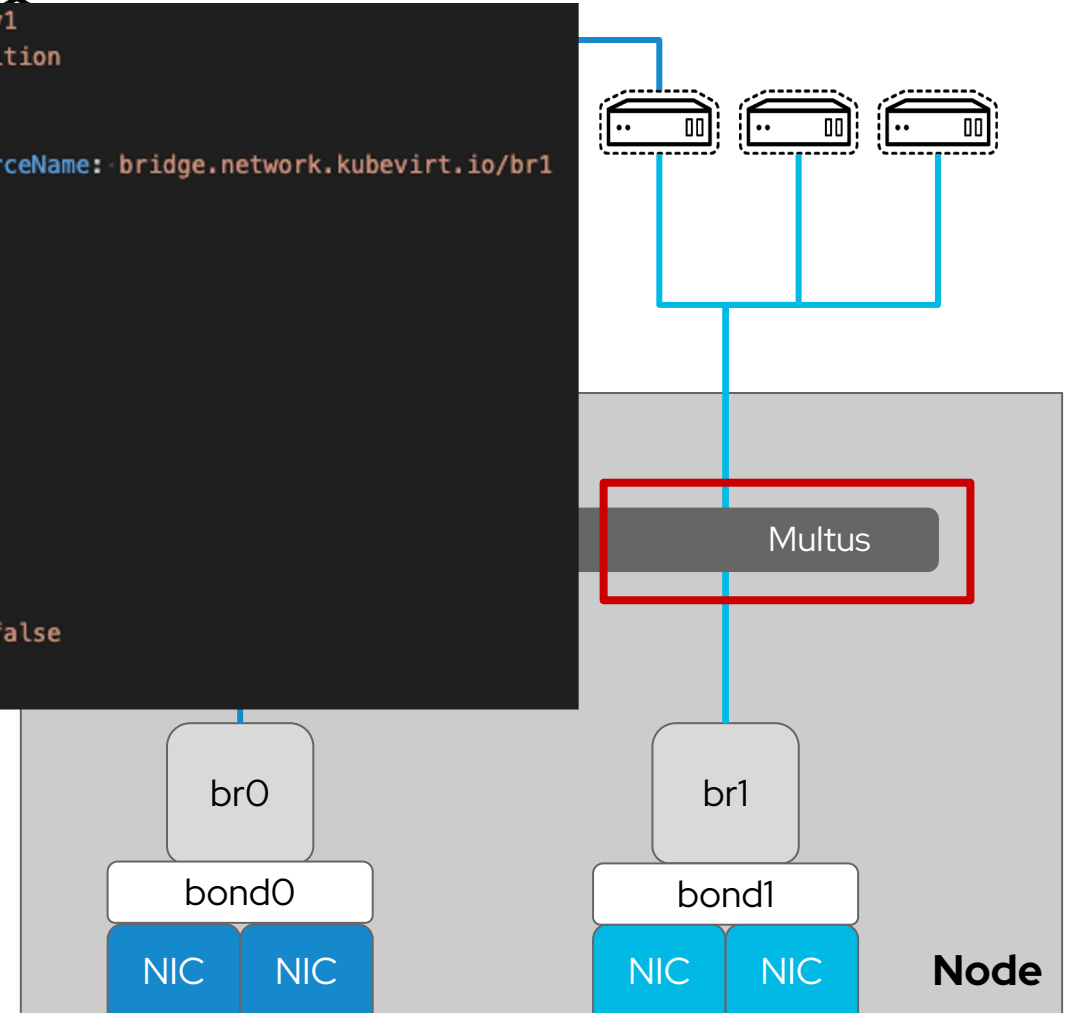
OVN Kubernetes L2 overlay network

OVN Kubernetes secondary localnet network

Network Attachment Definition configuration

- net-attach-def configures multus to allow the VM to access an underlying resource
 - Optionally define VLAN tags
- Limited to the namespace it's created in
 - Except the default namespace, which is available to all

```
1  apiVersion: k8s.cni.cncf.io/v1
2  kind: NetworkAttachmentDefinition
3  metadata:
4    annotations:
5      k8s.v1.cni.cncf.io/resourceName: bridge.network.kubevirt.io/br1
6    name: vlan-93
7    namespace: default
8  spec:
9    config: >-
10     {
11       "name": "vlan-93"
12       "type": "cnv-bridge",
13       "cniVersion": "0.3.1",
14       "bridge": "br1",
15       "vlan": 93,
16       "macspoofchk": true,
17       "ipam": {},
18       "preserveDefaultVlan": false
19     }
20
```



Host network configuration status

- Use the admin console to view the **NodeNetworkState**
- Detailed configuration information for host networking including
 - IP and MAC addresses
 - Bond configuration
 - Bridge configuration
- **Review** and **troubleshoot** host network configuration

The screenshot displays the Red Hat OpenShift Admin Console interface. The left sidebar contains a navigation menu with the following items: Administrator, Home, Operators, Workloads, Virtualization, Migration, Networking (expanded), Services, Routes, Ingresses, NetworkPolicies, NetworkAttachmentDefinitions, NodeNetworkConfigurationPolicy, and Storage. The 'NodeNetworkState' item is highlighted with a red '1'. The main content area is titled 'NodeNetworkState' and includes a search bar and a table of network interfaces. The table has columns for Name, IP address, Ports, and MAC address. The 'ethernet' interface is expanded, showing details for 'enp1s0', 'enp2s0', and 'enp3s0'. The 'linux-bridge' interface is also expanded, showing details for 'br-flat'. A red '2' highlights the 'ethernet' section, and a red '3' highlights the 'linux-bridge' section.

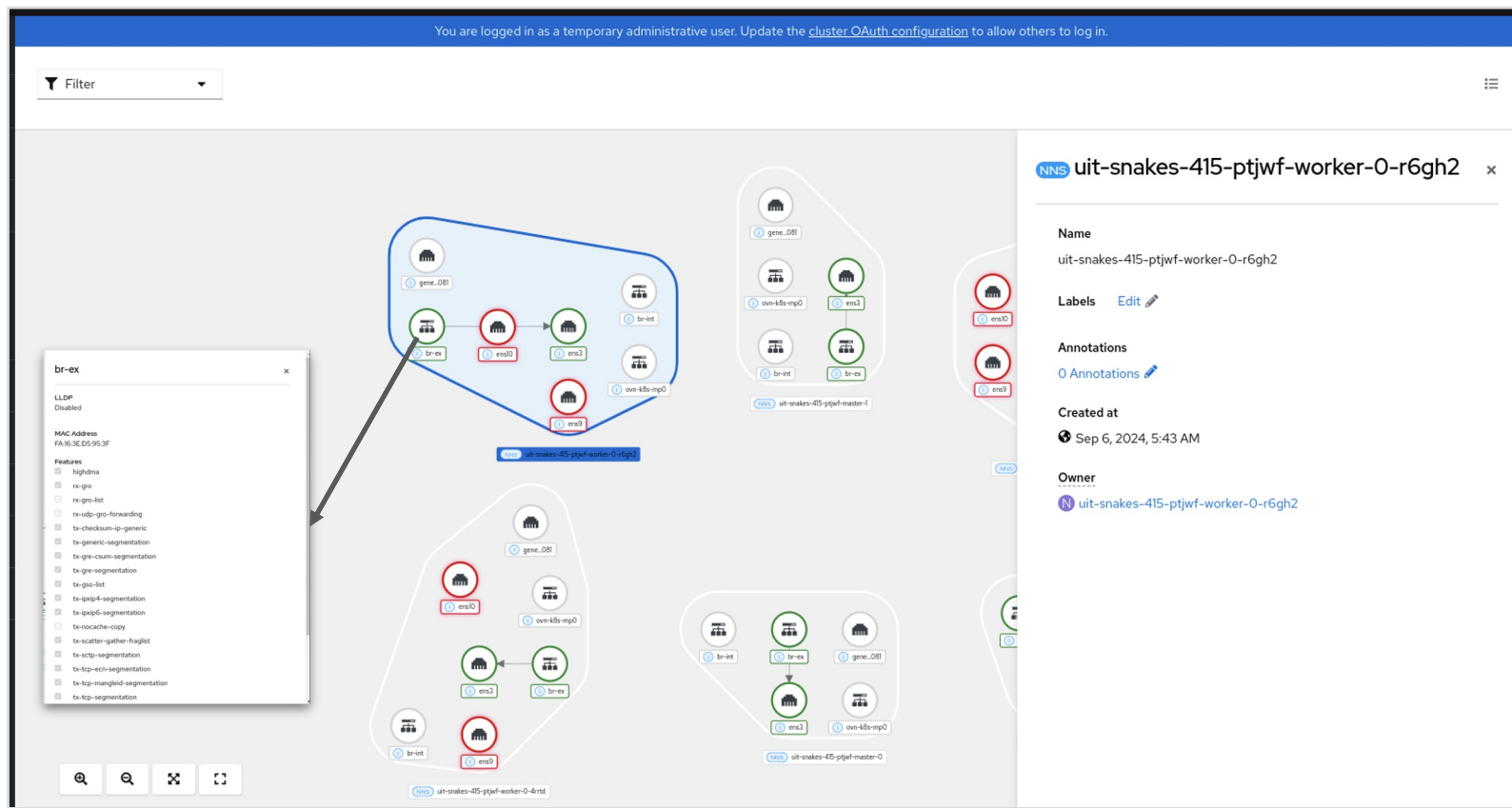
Name	IP address	Ports	MAC address
enp1s0 ↑	172.22.0.71/24	-	DE:AD:BE:EF:00:04
enp2s0 ↑	192.168.123.104/24	-	52:54:00:00:00:04
enp3s0 ↑	-	-	52:54:00:00:01:04
br-flat ↑	-	1	52:54:00:00:01:04

Node Network State (NNS) Topology View

Visualize node network configuration

A graphical representation of the nodes network configuration to provide admins a visual way to see and search the nodes configurations.

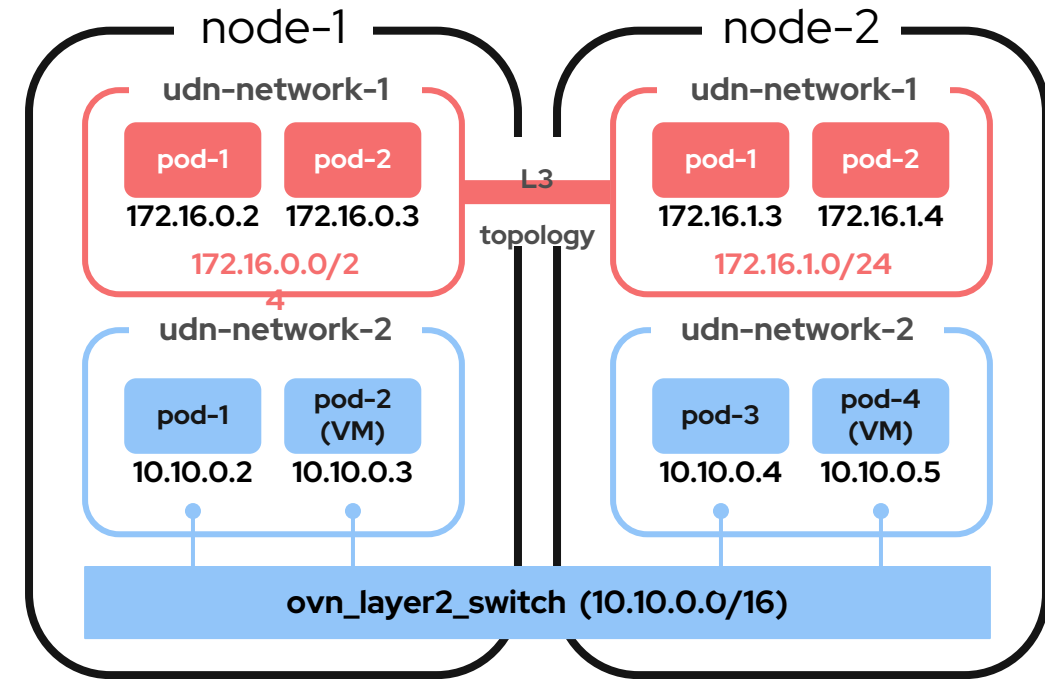
Also provides detailed view for each of the network devices



Native Network Isolation for Namespaces

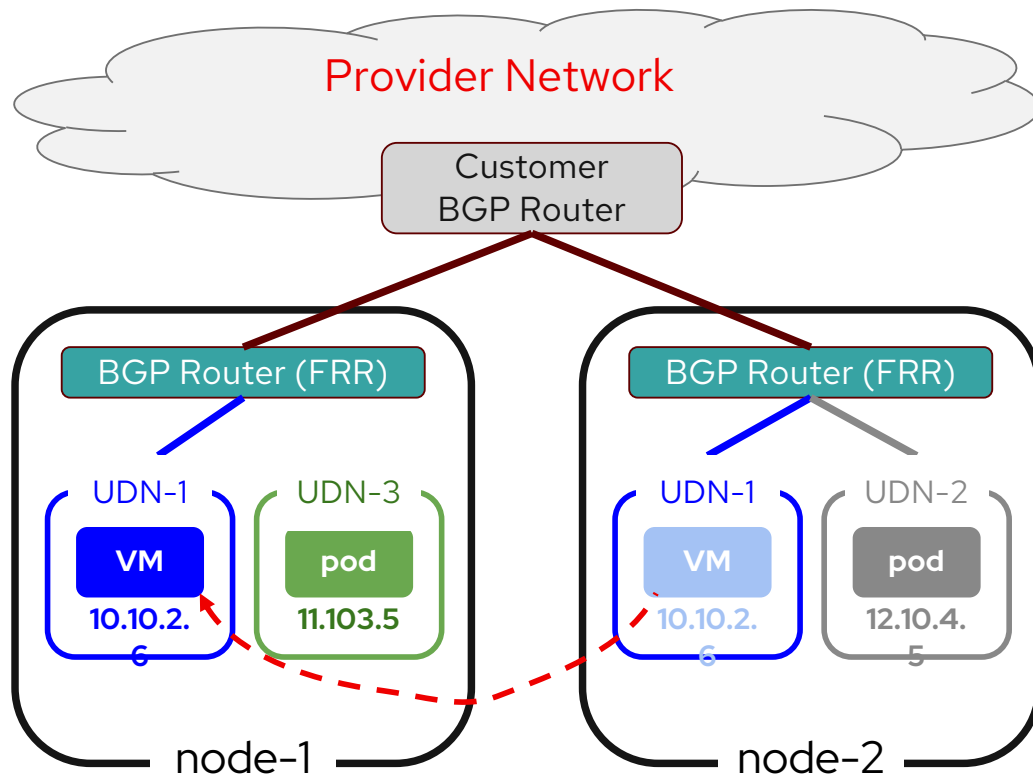
A better solution for the monolithic layer 3 Kubernetes pod network

- ▶ **User Defined Network (UDN)** support in OVN-Kubernetes
- ▶ One or more namespaces in each UDN (tenant)
- ▶ Support for:
 - **OpenShift Virtualization**
 - static IP assignments for the life of VMs (for OCP Virt)
 - L2, L3 & localnet UDN topologies
 - overlapping pod IPs across UDNs
 - Kubernetes Network Policy
 - clusterIP services and external services



Networking Enhancements

- Routed ingress (BGP) for L2 User Defined Network
- Ability to change the virtual network interface link state of a running VM



Network interfaces

[Add network interface](#)

Filter Name Search by name...

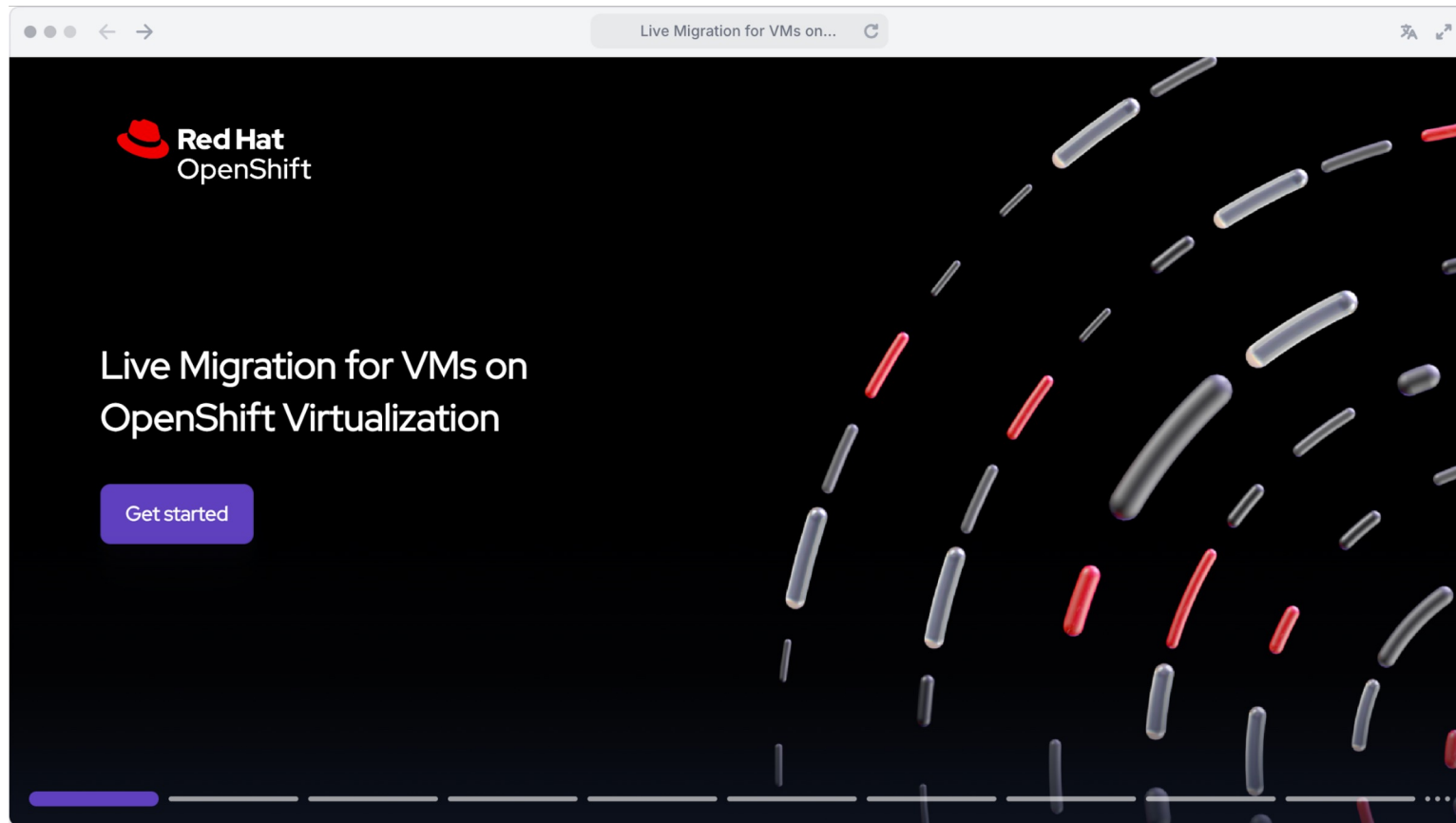
Name	Model	Net...	State	Type	MA...
nic-magenta-aphid-75	virtio	default/test1		Bridge	02:59:f4:00:00:00
nic-silver-cattle-53	virtio	default/localnet		Bridge	

Set link down
Edit
Delete

Live Migration

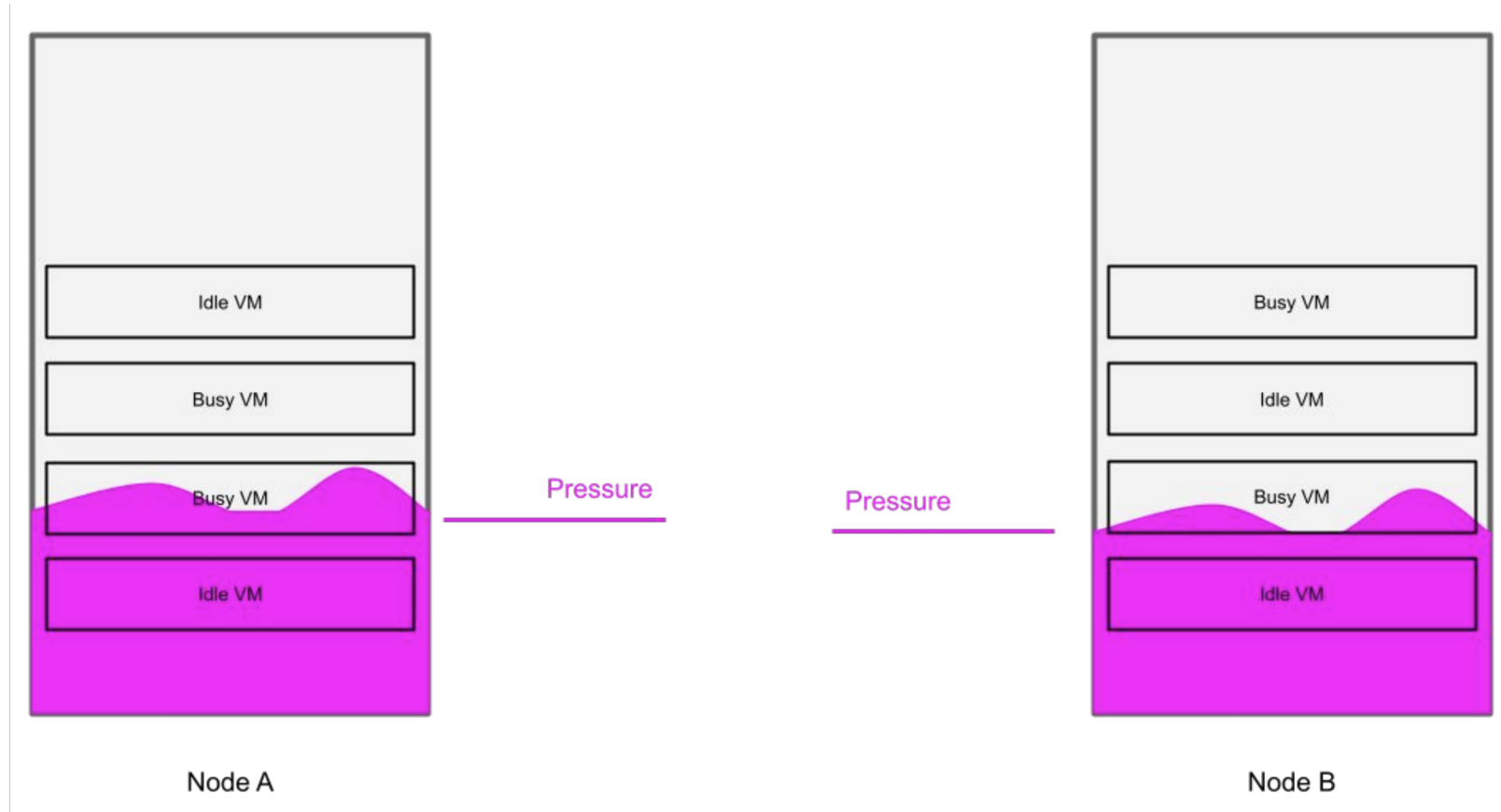
- Live migration moves a virtual machine from one node to another in the OpenShift cluster
- Can be triggered via GUI, CLI, API, or automatically
- Shared storage (RWX) is required
- New Features
 - Live migration to **specific node**
 - **Cross-cluster** VM live migration (TP)

Demo: Live Migration

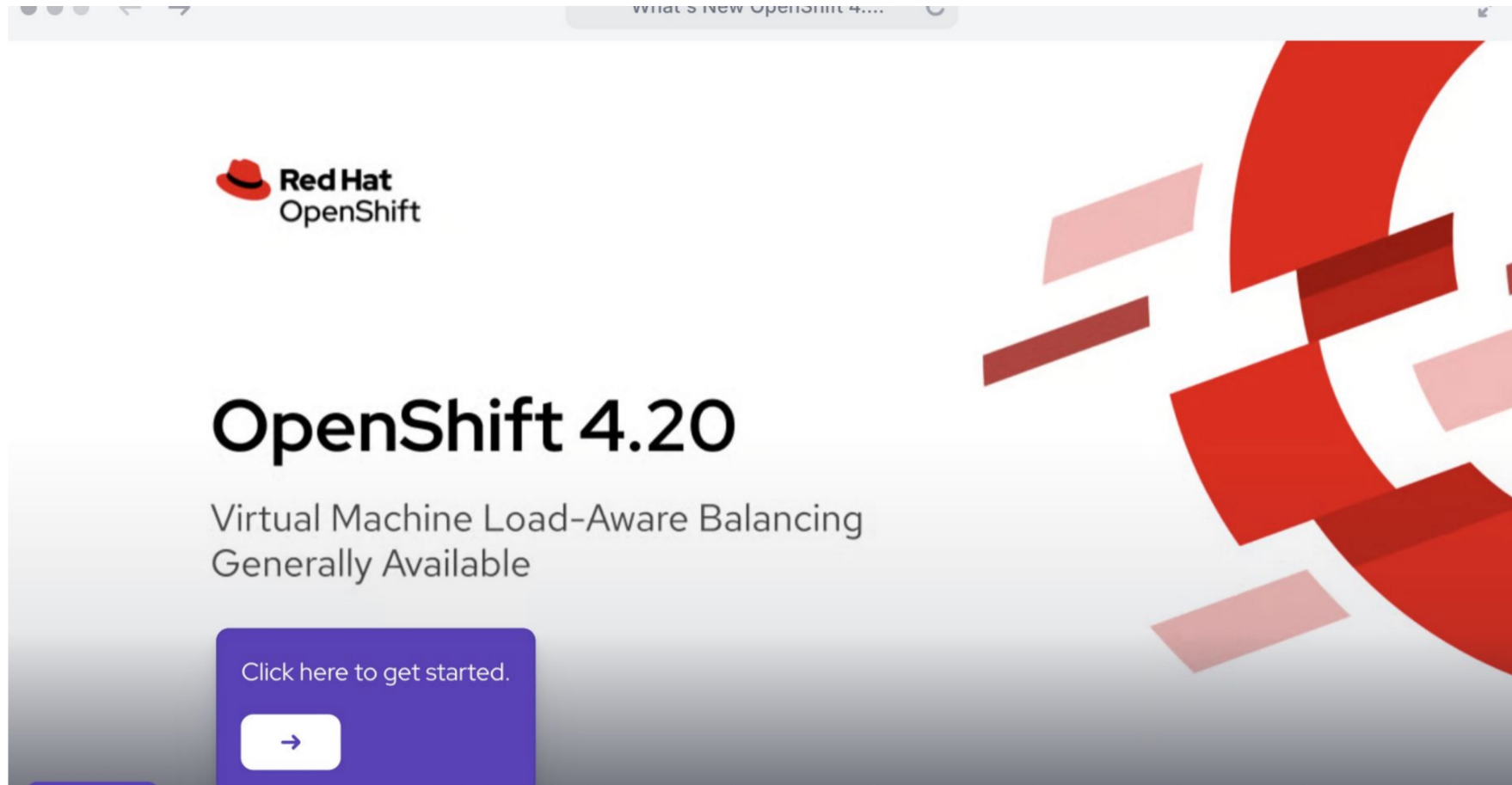


Load-aware Balancing (Descheduler)

- CPU utilization based Automatic VM workload balancing

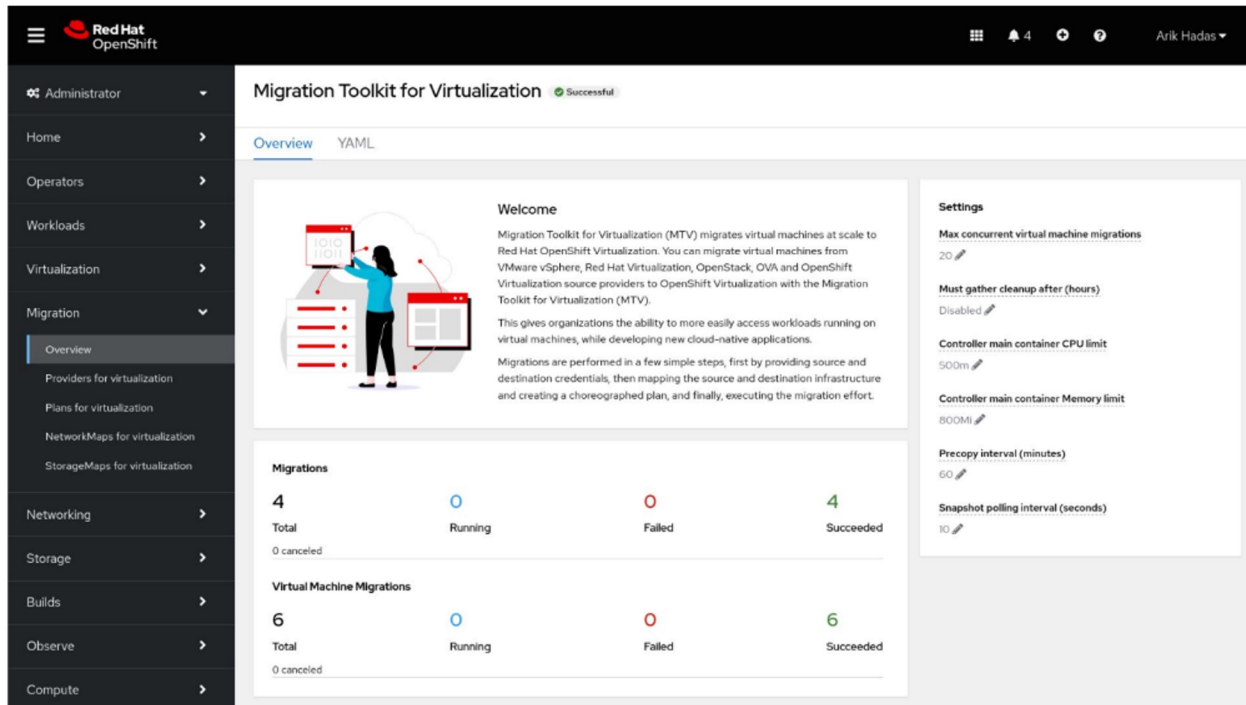


Demo: Load-aware Balancing (Descheduler)



Migrate your virtual machines with minimal disruption

 **Migration toolkit for virtualization (MTV)** included with OpenShift



Red Hat OpenShift

Administrator

Home

Operators

Workloads

Virtualization

Migration

Overview

Providers for virtualization

Plans for virtualization

NetworkMaps for virtualization

StorageMaps for virtualization

Networking

Storage

Builds

Observe

Compute

Migration Toolkit for Virtualization Successful

Overview YAML

Welcome

Migration Toolkit for Virtualization (MTV) migrates virtual machines at scale to Red Hat OpenShift Virtualization. You can migrate virtual machines from VMware vSphere, Red Hat Virtualization, OpenStack, OVA and OpenShift Virtualization source providers to OpenShift Virtualization with the Migration Toolkit for Virtualization (MTV).

This gives organizations the ability to more easily access workloads running on virtual machines, while developing new cloud-native applications.

Migrations are performed in a few simple steps, first by providing source and destination credentials, then mapping the source and destination infrastructure and creating a choreographed plan, and finally, executing the migration effort.

Settings

- Max concurrent virtual machine migrations: 20
- Must gather cleanup after (hours): Disabled
- Controller main container CPU limit: 500m
- Controller main container Memory limit: 800Mi
- Precopy interval (minutes): 60
- Snapshot polling interval (seconds): 10

Migrations

4	0	0	4
Total	Running	Failed	Succeeded
0 canceled			

Virtual Machine Migrations

6	0	0	6
Total	Running	Failed	Succeeded
0 canceled			

VM migrations made simple

- Migrate virtual machines to OpenShift Virtualization in a few steps
- Provide source and destination credentials, map infrastructure, and create migration plans

Migration toolkit for virtualization

Making mass migration possible



Easy to use UI

Simple, easy to use interface



Concurrent migrations

Easily migrate VMs concurrently from VMware, Red Hat Virtualization and OpenStack to OpenShift and between OpenShift clusters



Pre-copied data

VM data pre-copied before shutdown (Warm Migration) for VMware and RHV migrations



VM validation service

Run checks on VM configuration to avoid migration issues



Storage offloading

Accelerate migrations with underlying storage systems versus IP network (Tech Preview MTV 2.10)



Migration network selection

Avoid impact on other running workloads

Accelerate VM migrations with existing storage systems

Fast, low-risk migrations with storage offloading

Faster than traditional migrations

- Reduces IP network load
- Offloads compute resources to the storage system
- Migrations up to **10x faster***

Compatible with supported certified storage partners

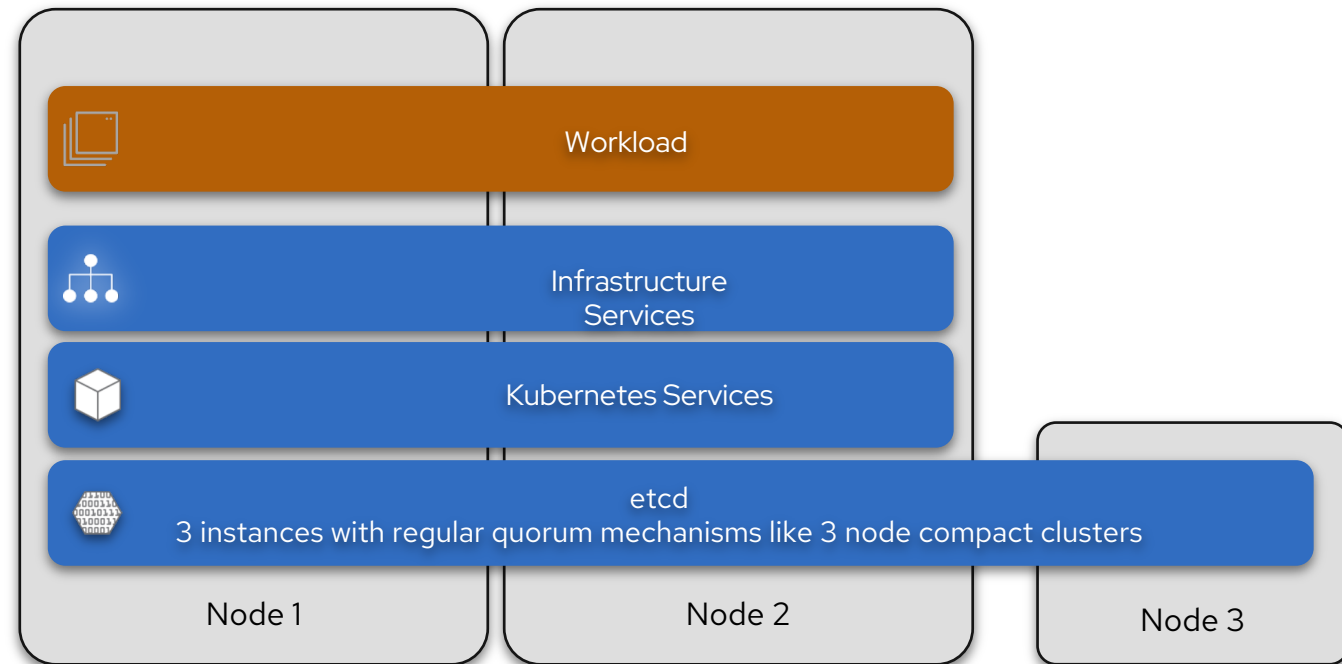
- Partner list growing



Two Node OpenShift with Arbiter (TNA)

New Topology Approach:

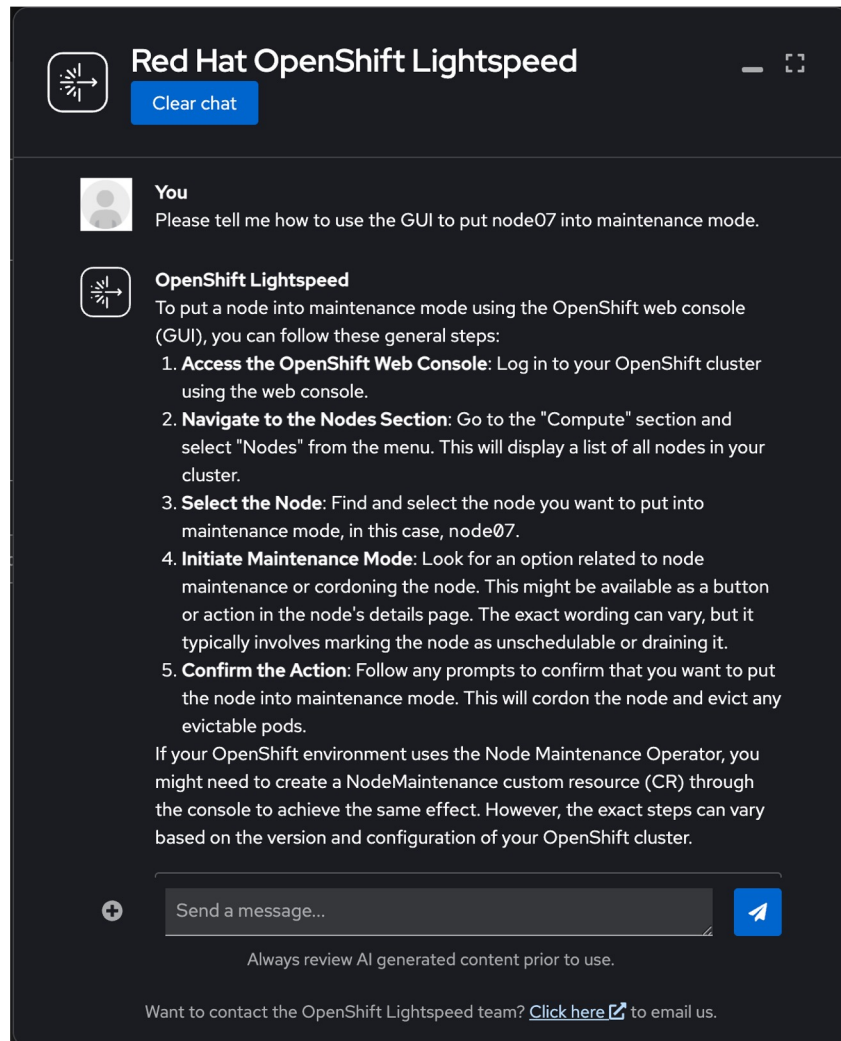
- **Two node** solution for cost sensitive customers who do not need a full third node for their workload
- Small arbiter node (2 vCPU, 8Gi), running only third etcd instance
- **Same HA characteristics** as a regular three node cluster - tolerates single node outage
- **OpenShift Virtualization** fully supported
- Hyperconverged Storage / Software Defined Storage (replica 2, disks only on node 1 and 2) via partners:
 - Pure / Portworx
 - Arctera / Infoscale
 - IBM / Fusion



```
% oc get nodes
NAME    STATUS    ROLES
node1   Ready    master,worker
node2   Ready    master,worker
node3   Ready    arbiter
```


Operational Guidance with OpenShift Lightspeed

Generative AI based conversational assistant for OpenShift



- ▶ Generative AI-based **conversational assistant** in OpenShift Console, accessible throughout the Management Console
- ▶ Assists traditional vAdmins transition to OpenShift Virtualization
- ▶ Get **product help** and helps guide daily tasks and troubleshooting
- ▶ Context Aware, and trained on VMware -> OpenShift product terms
- ▶ Bring your own LLM provider
- ▶ For both VMs and containers

Easy/Guided Install

Installation Wizard tuned for OpenShift Virtualization (Tech Preview)

- 1 Cluster details
- 2 Operators
- 3 Host discovery
- 4 Storage
- 5 Networking
- 6 Review and create

Operators

Find bundles or operators

Bundles

Virtualization ?



Run virtual machines alongside containers on one platform.

Technology Preview

OpenShift AI ?



Train, serve, monitor and manage AI/ML models and applications using GPUs.

Developer Preview

Single Operators (22 | 8 selected)

Storage

☒ Local Storage Operator ?

Allows provisioning of persistent storage by using local volumes. [Learn more](#)

☐ Logical Volume Manager Storage ?

Storage virtualization that offers a more flexible approach for disk space management. [Learn more](#)

☐ OpenShift Data Foundation ?

Persistent software-defined storage for hybrid applications. [Learn more](#)

Virtualization

☒ OpenShift Virtualization ?

Run virtual machines alongside containers on one platform. [Learn more](#)

☒ Migration Toolkit for Virtualization ?

This Toolkit (MTV) enables you to migrate virtual machines from VMware vSphere, Red Hat Virtualization, or OpenStack to OpenShift Virtualization running on Red Hat OpenShift. [Learn more](#)

Ease of Migration

- **Live migration of VMs and Storage across clusters**
- GA of Storage accelerated migration to OpenShift
- Migration from Hyper-V
- Self Guided Migration assistant



Networking

- **BGP and EVPN for stretching networks across clusters**



Simplified & Scaled Administration

- **Extend single cluster experience to multiple clusters**
- VM Right Sizing operations
- Guided networking configuration



Red Hat OpenShift Virtualization What's Next

Edge to Cloud

- **Two node HA Solution (no arbiter needed) for Edge deployments**
- GA support for ARM platforms
- GA of Oracle OCI, GCP, Azure ARO



Storage & Data Protection

- **Change block tracking for Incremental backup**



Security and Performance

- **Optimized defaults for multiple live migrations and databases**
- VM vulnerability reporting in RHACS
- Compliance Operator for Hardened OpenShift Virtualization





Connect

Thank you



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



twitter.com/RedHat

