

1



VMs + Containers = The Perfect Wedding

Sreejith Anujan sreejith@redhat.com

What is KubeVirt?

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



KubeVirt

• Virtual machines

- Running in containers
- Using the KVM hypervisor
- Scheduled, deployed, and managed by Kubernetes
- Integrated with container orchestrator resources and services
 - Traditional Pod-like SDN connectivity and/or connectivity to external VLAN and other networks via multus
 - Persistent storage paradigm (PVC, PV, StorageClass)



VM containers use KVM

- OpenShift Virtualization uses KVM, the Linux kernel hypervisor
- KVM is a core component of the 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

6

• libvirt provides a management abstraction layer



Built with Kubernetes

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:

8

- Container Networking Interface (CNI)
- Container Storage Interface (CSI)
- Custom Resource Definitions (CRD, CR)
- Schedule, connect, and consume VM resources as container-native





V0000000

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
 - 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 / Fedora / CentOS Stream

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

Security - Defense in depth

• Immutable CoreOS by default, SELinux MCS, plus KVM isolation

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 with Kubernetes

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

Bed Hat OpenShift Container Platform					III 🔶	•••	ansulliv 🔻
🕈 Administrator 🗸	Project: default 🔹						
Home >	Virtualization						
Operators >	Virtual Machines	Virtual Machine 7	Femplates				
Workloads 🗸							
Pods	Create Virtual Machine	e 🔻					
Virtualization	_						
Deployments	▼ Filter ▼ Na	ame 🝷 Search by na	ame				
Deployment Configs	Name 1	Namespace 1	Status 🗍	Created 1	Node 1	IP Address	
Stateful Sets	VM fedora01	NS default	C Running	🚱 Jul 9, 5:00 pm	N worker-	10.131.0.74	:
Secrets					0.owv.lab.lan		
Config Maps	VM rhel	NS default	2 Running	🚱 Jul 8, 4:18 pm	N worker- 0.owv.lab.lan	192.168.14.163/24, fe80::87cc:48e:1e2 9d23/64	: ::
Cron Jobs	VM rhelO1	NS default	O Off	🚱 Jul 9, 4:58 pm			:
Jobs			-	2			•
Daemon Sets	VM windows2019	NS default	C Running	🚱 Jul 9, 5:01 pm	N worker- 1.owv.lab.lan	10.128.2.52	
Replica Sets							
Replication Controllers							

Create VMs



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



Create Virtual Machine - General

- Source represents how the VM will boot
 - Boot via PXE, optionally diskless
 - URL will import a QCOW2 or raw disk image using a DataVolume
 - Container uses a container image, pulled from a registry, for the disk
 - \circ $\,$ Disk uses an existing PVC $\,$

- Flavor represents the preconfigured CPU and RAM assignments
 - Tiny = 1 vCPU and 1GB RAM, Small = 1 vCPU and 2GB RAM, etc.
- Workload profile defines the category of workload expected and is used to set KVM performance flags

Project: default 🔻	
Create Virtual Machine	
1 General 2 Networking	Name *
3 Storage4 AdvancedCloud-init	Description
Virtual Hardware 5 Review	Template No template available
6 Result	Source * Select Source PXE URL Container Disk
2	Flavor * [®] Custom Select Flavor Tiny Small Medium Large Custom Workload Profile * [®] Select Workload Profile desktop highperformance server

Create Virtual Machine - 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

Project: default 🔻						
Create Virtual Machine						
 General Networking 	Network Interfac	Ces Model 1	Network 🗍	Туре 💲	Add Netwo	ork Interface
4 Advanced Cloud-init	nic-0	VirtlO	Pod Networking	masquerade		:
Virtual Hardware		Add Network	Interface			
6 Result		nic-1				
	2	VirtIO			•	
	3	Network * host-br1			•	
	4	Type * bridge			•	
		MAC Address				
	l			Cancel	Add	
	Next Review	v and create Ba	ack Cancel			

Create Virtual Machine - 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

						ect: default 🔻	Proj
						eate Virtual Machine	Cr
Disk	Add D				Disks	General	1
	Storage Class	Interface 1 St	Size 1	Source 1	Name 1	Networking	2
0	-	VirtlO -	10 GiB	URL	rootdisk	Storage	3
				_		Advanced	4
			Add Disk			Cloud-init	
			Source *			Virtual Hardware	
			Blank	ے ا		Review	5
			disk-0			Result	6
		010	Size *				
	GIB	GIB	20				
			VirtlO	3			
			Storage Class				
			Advanced	4			
			Volume Mode				
	-		Filesystem				
	-		Access Mode Single User (RWO)	5			
	Add	Cancel		L			
		el	Back Cancel	view and create	Next Revi		
	GiB V V V Add	GiB	Interface * VirtIO Storage Class managed-nfs-storage Advanced Volume Mode Filesystem Access Mode Single User (RWO) Back Cancel	3 4 5	Next	Kesult	6

Create Virtual Machine - Advanced

- Customize the operating system deployment using cloud-init scripts
 - Guest OS must have cloud-init installed
 - RHEL, Fedora, etc. cloud images
- Attach ISOs to the VM CD/DVD drive
 - ISOs stored in container images (registry), existing PVC, or imported from URL

Project: default 🔻	
Create Virtual Machine	
 General Networking Storage Advanced Cloud-init Virtual Hardware Review Result 	• Form Outstand script Hostname Authenticated SSH Keys • Add SSH Key • Base-64 encoded 2 / Image: / Image:
	Next Review and create Back Cancel

Create Virtual Machine - Review

- A summary of the decisions made
- Warnings and other important information about the configuration of the VM are displayed
- Choose to automatically power on the VM after creation

Red Hat OpenShift Container Platform							* • (9	ansulliv 👻
🗘 Administrator	Project: default 🔻								
Home >	Create Virtual Machine								
Operators >	1 General	Review and co	onfirm y	our settings	5				
Workloads >	2 Networking	Goporal							
Networking >	3 Storage	Name	rhel(12					
Storage >	4 Advanced	Description	No c	lescription					
Builds >	Cloud-init	Source	URL		1 X X 8000.00				
Monitoring >	Virtual Hardware	Operating System Flavor	Red Sma	Hat Enterprise L II: 1 vCPU, 2 GiB I	inux 8.0 or higher Memory				
Compute >	6 Result	Workload Profile	desk	top					
User Management >		Networking							
Administration >		Name	,	Model	Netwo	rk	1	4AC Addr	ess
		nic-0	1	/irtlO	Pod Ne	etworking			
		Storage							
Sonage Sonage									
		Name S	ource	Size	Interface	Storage Cla	ss Acce	ss Mode	Volume Mode
		rootdisk U	IRL	10 GiB	VirtIO		Sing (RW	e User D)	Filesystem
		Advapcod							
			ot Enabled						
	2	 Start virtual mac 	thine on crea	ation					
		Create Virtual Ma	achine	Back C.	ancel				

Virtual machines

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



Containerizing KVM



Architectural Overview



Cluster Services

Nodes

Adding virtualization to the Kubernetes API

CRD and aggregated API servers

- These are the ways to extend the Kubernetes API in order to support new entities
- For users, the new entities are indistinguishable from native resources

Single API entry point for all workloads

• All workloads (containers, VMs, and serverless) are managed through a single API

