Kubevirt Internals

Sergi Jimenez <sjr@redhat.com>

Juan Manuel Parrilla <juanma@redhat.com>

OpenSouthCode 2019

Whoami

Kubevirt Community

```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: sergis.alpha.redhat.com
spec:
  group: alpha.redhat.com
  versions:
    - name: v1alpha1
      served: true
      storage: true
  scope: Cluster
  names:
    plural: sergis
    singular: sergi
    kind: Engineer
    shortNames:
    - sjr
    - tripledes
```

Github: @tripledes
IRC: freenode/TripleDES
Twitter: @tripledes

Kubevirt Community

```
---
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: juanpas.alpha.redhat.com
  description: 'Juan Manuel Parrilla'
  labels:
  -   python_golang_lover
spec:
  group: alpha.redhat.com
  versions:
    - name: v1alpha1
      served: true
      storage: true
  scope: Cluster
  names:
    plural: juanpas
    singular: juanpa
    kind: Engineer
    shortNames:
    - Parri
```

Github: github.com/jparrill
IRC: @jparrill
Twitter: @Kerbeross

Prologue

What is KubeVirt?

“Virtual Machine management addon to Kubernetes that extends Kubernetes in a way that allows it to schedule VM workloads side by side with container workloads.”

What is Kubevirt

- Why do we do KubeVirt?
- Is there something wrong with RHV/oVirt?
- Is there something wrong with OpenStack/RDO?

???

What is Kubevirt

- Why do we do KubeVirt?
- Is there something wrong with RHV/oVirt?
- Is there something wrong with OpenStack/RDO?

There is **nothing** wrong with RHV or OpenStack. What we try to solve is providing a unified stack for hybrid cloud operations based on OpenShift/Kubernetes.

Virtual Machines as just another Workload

- KubeVirt is delivered via Pods, like any other k8s application
- KubeVirt is managed by Operators:
  - Installations only happen on the k8s workload level
  - Updates only happen on the k8s workload level
  - Deinstallation simply means deleting our components from the cluster
- KubeVirt is fully integrated into k8s control plane:
  - We are an aggregated API service
  - Normal kubectl/oc commands work with VMs too
- KubeVirt fully relies on provided extension mechanisms from k8s:
  - Custom Resource Definitions
  - Webhooks
  - ...

Architecture

What are Containers?

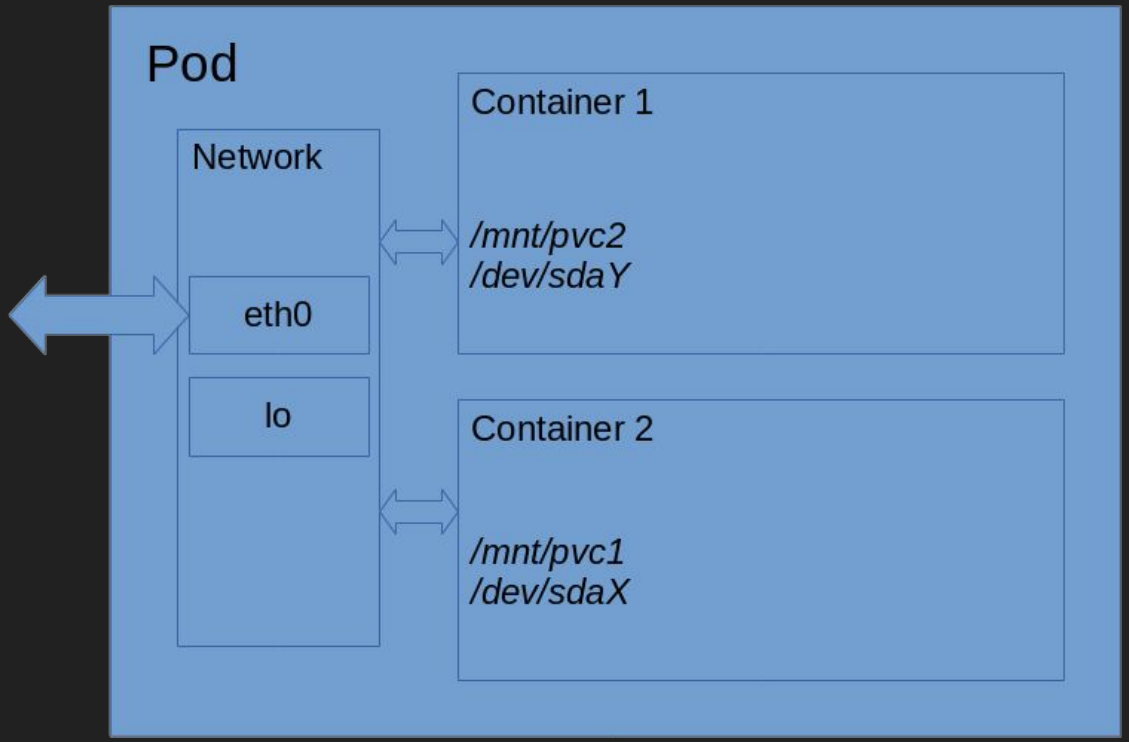
Containers mostly use two isolation techniques provided by the kernel:

- cgroups
- namespaces

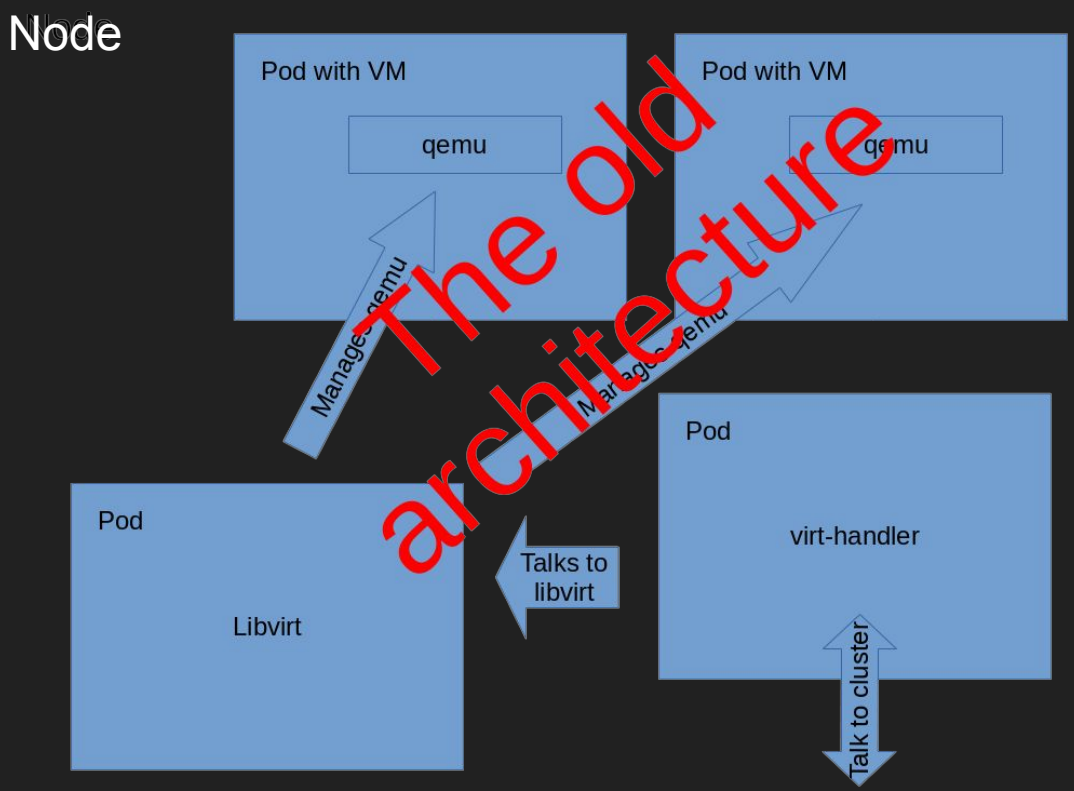
What are Pods?

- Consists of one or more containers
- Each container has its own cgroups
- All containers of a pod share the network namespace
- Each container has for the rest its own namespace:
  - PID namespace
  - IPC namespace
  - Mount namespace

What are Pods?



How we started with libvirt



Simplified world with containers

- What if we don’t fight the kubelet?
- What if we could simplify our lives by welcoming the fact that containers do a lot of security and isolation work for us?

How we use libvirt

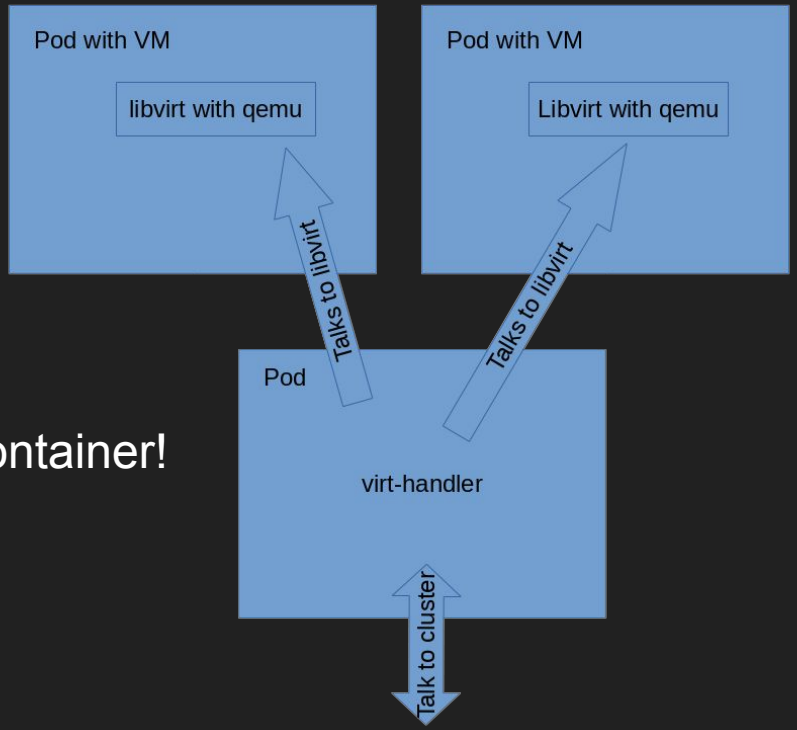
How about starting one libvirt instance inside the container where the Virtual Machine is supposed to run in?

We don’t have to trick libvirt or Kubernetes then in any way:

- Kubernetes provides the resources
- Libvirt runs inside the target pod and sees what it is supposed to see and what it can use for defining the Virtual Machine

How we use libvirt

node



libvirt runs inside a container!

Components

CDI - Containerized Data Importer

Containerized-Data-Importer (CDI) is a persistent storage management add-on for Kubernetes. It's primary goal is to provide a declarative way to build Virtual Machine Disks on PVCs for Kubevirt VMs

CDI works with standard core Kubernetes resources and is storage device agnostic, while its primary focus is to build disk images for Kubevirt, it's also useful outside of a Kubevirt context to use for initializing your Kubernetes Volumes with data.

Networking

Multus - which is a “meta” CNI (Container Network Interface), allowing multiple CNIs to coexist, and allow for a pod to use the right ones for its networking needs.

Types:

- Linux Bridge
- OVS
- SRIOV
- ....

Features

What libvirt provides for Storage

libvirt provides Storage management. Here some nice features:

- Directory pool
- Filesystem pool
- Network filesystem pool
- Logical volume pool
- Disk pool
- iSCSI pool
- iSCSI direct pool
- SCSI pool

- Multipath pool
- RBD pool
- Sheepdog pool
- Gluster pool
- ZFS pool
- Vstorage pool

How we use libvirt for Storage

Here what k8s provides:

- Support for arbitrary storage providers via CSI (Container Storage Interface)
- Filesystem support (kubelet provides volumes mounted in specific folders)
- Block device support (kubelet provides block devices in /dev/ inside the container)

We only attach images in folders or block devices in /dev/ in the container to qemu.

What libvirt provides for Networking

libvirt provides nice Network Management Features. Here some:

- Firewall rules
- OVS integration
- Attachment to Bridges with TAP devices
- Attachment to interfaces with macvtap
- SRIOV
- Host Device passthrough
- DHCP server
- ...

How we use libvirt for Networking

Here what k8s does:

- Support for arbitrary network providers via CNI (Container Network Interface)
- Provides one NIC in an isolated network namespace for the pod
- With multus multiple NICs are possible
- With multus and a device plugin SRIOV can be passed through

The kubelet hides most of the host from libvirt. At the end we have two scenarios:

- Kubelet provides a fully prepared NIC (default)
- Kubelet provides a fully prepared NIC and a PCI device (sriov)

We attach to the final NIC or forward the PCI device to qemu.

What libvirt provides for Isolation

libvirt provides cgroup management and to a degree also namespace isolation.

How we use libvirt for Isolation

We just let the kubelet set up the whole isolation and just start libvirt and qemu in the resulting, already isolated container.

Trimmed down libvirt

libvirt is currently redesign into smaller independent components. We mostly only need core functionality from libvirt. This means for us:

- Less dependencies means less CVEs
- Less dependencies means smaller base images
- Smaller memory footprint

Developer Demo

Developing with KubeVirt

Bring up an OpenShift cluster:

- Prerequisites:
  - Nested virtualization enabled
  - Docker, rsync and make
- Start an openshift cluster with two nodes:

```
$ git clone https://github.com/kubevirt/kubevirt
$ cd kubevirt
$ export KUBEVIRT_PROVIDER=os-3.11.0-crio # k8s-1.13.3 is a more lightweight one
$ export KUBEVIRT_NUM_NODES=2
$ make cluster-up
$ cluster/kubectl.sh get nodes
```

Developing with KubeVirt

- Deploy KubeVirt on the cluster:

```
$ make cluster-sync
```

- Start your first VirtualMachineInstance:

```
$ cluster/kubectl.sh create -f cluster/examples/vmi-nocloud.yaml
```

- Dump the domain xml:

```
$ cluster/kubectl.sh get pods -l kubevirt.io
NAME                             READY   STATUS    RESTARTS   AGE
virt-launcher-vmi-nocloud-rcwgq   2/2     Running   0          56s
$ cluster/kubectl.sh exec virt-launcher-vmi-nocloud-rcwgq -c compute virsh dumpxml \
     default_vmi-nocloud
```

Resources

Kubevirt Resources

- Kubevirt.io
- Kubevirt VM Creation Flow
- CDI
- CDI - How to
- Multus - How to
- KubevirtCI
- Kubevirt Operator

Thank You & Enjoy the Lab :)!

Further Questions?

#virtualization in slack.k8s.io

#Container-Native Virtualization in chat.google.com

kubevirt-dev@googlegroups.com

cnv-devel@redhat.com