KubeCon
CloudNativeCon
North America 2019

KubeVirt Intro:

Virtual Machine Management on Kubernetes

Stephen Gordon - @xsgordon

Chandrakanth Jakkidi

Who are we?

- Senior Manager, Product Management, Cloud Platforms @ Red Hat
- Based in Toronto, Canada
- Supporting the team responsible for “Kubernetes-native infrastructure” related efforts, incl. KubeVirt.
- Previously OpenStack, Red Hat Virtualization product management and technical writer roles...
- ...and once upon a time COBOL developer.

Chandrakanth Reddy Jakkidi

- Senior Software Engineer @ F5 Networks

  Product Development , Ecosystems Group

- Supporting and Leading Containerization Efforts

  https://github.com/F5Networks/k8s-bigip-ctlr (Container Ingress Services)
  https://github.com/F5Networks/terraform-provider-bigip

- Active User Contributor to Open Source Communities

  (past - Openstack , Present - CNCF projects like k8s/kubevirt)

- Has around 14 years of experience in

  Networking/Virtualization/Cloud Technologies

- Previously worked with Cisco Systems ,

  Starent Networks , Emerson , NXP/Freescale ,Artesyn ….

KubeVirt Introduction

- Goal:
  - **Turn Kubernetes into a single orchestrator for containers and virtual machines.**
- Started in 2016 at Red Hat
- Open sourced in January 2017:
  - https://github.com/kubevirt/kubevirt
- Accepted into CNCF Sandbox 2019
- Apache 2.0 License

KubeVirt Community

- CNCF Sandbox
- 1,600+ GitHub Stars
- 76 Code Contributors (Red Hat)
  38 Code Contributors (non-Red Hat)
- 1,900+ Pull Requests
- 320+ GitHub Forks
- 20+ releases (close to stable release)
- Weekly Community Meeting
  and #virtualization on slack

(Some) Existing users and contributors*

Akamai	✓	
Apple		✓
Cloudflare	✓	✓
Cisco	✓	
Loodse	✓	(✓)
OSI	✓	(✓)
Red Hat	✓	✓
SAP	✓	(✓)
StackPath	✓	✓

* In the last 6 months +++ (…) In other repos

Why KubeVirt?

- Growing velocity behind Kubernetes and surrounding ecosystem for new applications.
- Reality that users will be dealing with virtual machine workloads for many years to come.
- Focus on building transition paths for users with workloads that will either never be containerized:
  - Technical reasons (e.g. older operating system or kernel)
  - Business reasons (e.g. time to market, cost of conversion)
- ...or will be decomposed over a longer time horizon.

KubeVirt Use Cases

- ...to run Virtual Machines to support new development
  - Build new applications relying on existing VM-based applications and APIs.
  - Leverage Kubernetes-based developer flows while bringing in these VM-based dependencies.
- ...to run Virtual Machines to support applications that can’t lift and shift
  - Users with very old applications who are not in a position to change them significantly.
  - Vendors with appliances (customer kernels, custom kmods, optimized workflows to build appliances, …) they want to bring to the cloud-native ecosystem.

KubeVirt Use Cases

- ...to run Kubernetes (!)
  - KubeVirt as a Cluster API provider
    - Hard Multi-Tenancy
  - Community provided cloud-provider-kubevirt

KubeVirt Use Cases

- ...to run Virtual Network Functions (VNFs) and other virtual appliances
  - VNFs in the context of Kubernetes are of continued interest, in parallel to Cloud-Native Network Function exploration.
    - Kubernetes is an attractive target for VNFs.
      - Compute features and management approach is appealing.
      - **But:** VNFs are hard to containerize!

KubeVirt Features

- Comprehensive API to run Virtual Machines on Kubernetes
- Kubernetes-native approach to virtualization
  - Implemented using **CustomResourceDefinitions**
  - Integration with **cluster** level features:
    - Storage, network, services, etc.
  - Integration with **node** level features:
    - CPUManager, multi-network, huge pages, etc.
- Focus on ease of use and a kubernetes-native look, feel, and behavior.

Getting started

KubeCon CloudNativeCon North America 2019

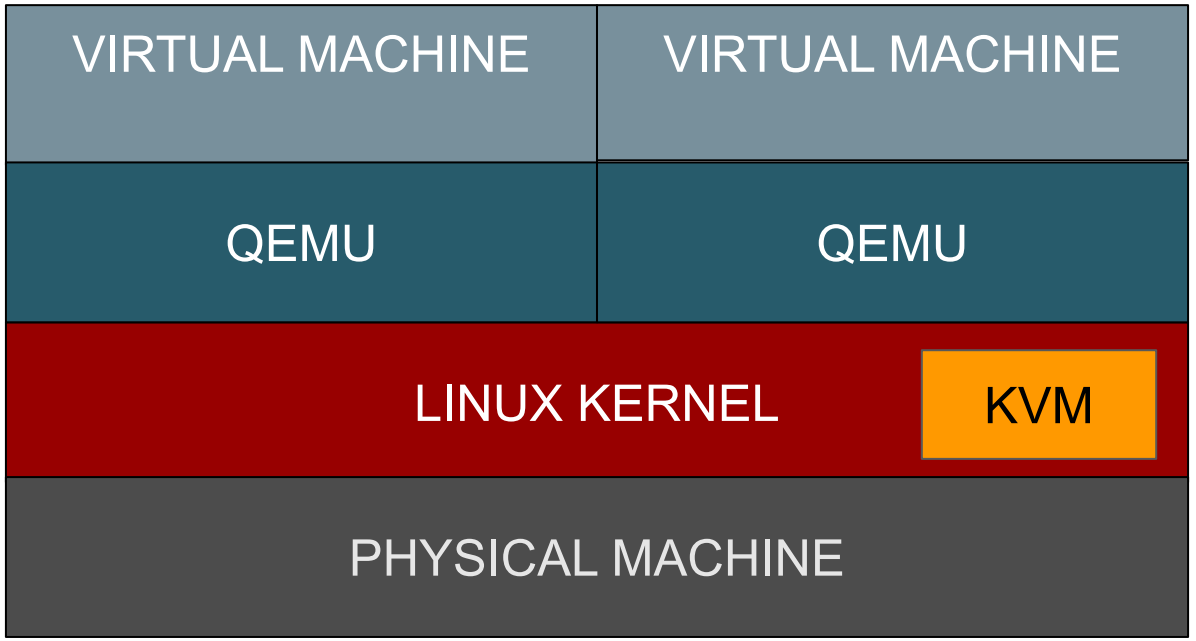
https://kubevirt.io

KubeCon
CloudNativeCon
North America 2019
Technical
Architecture

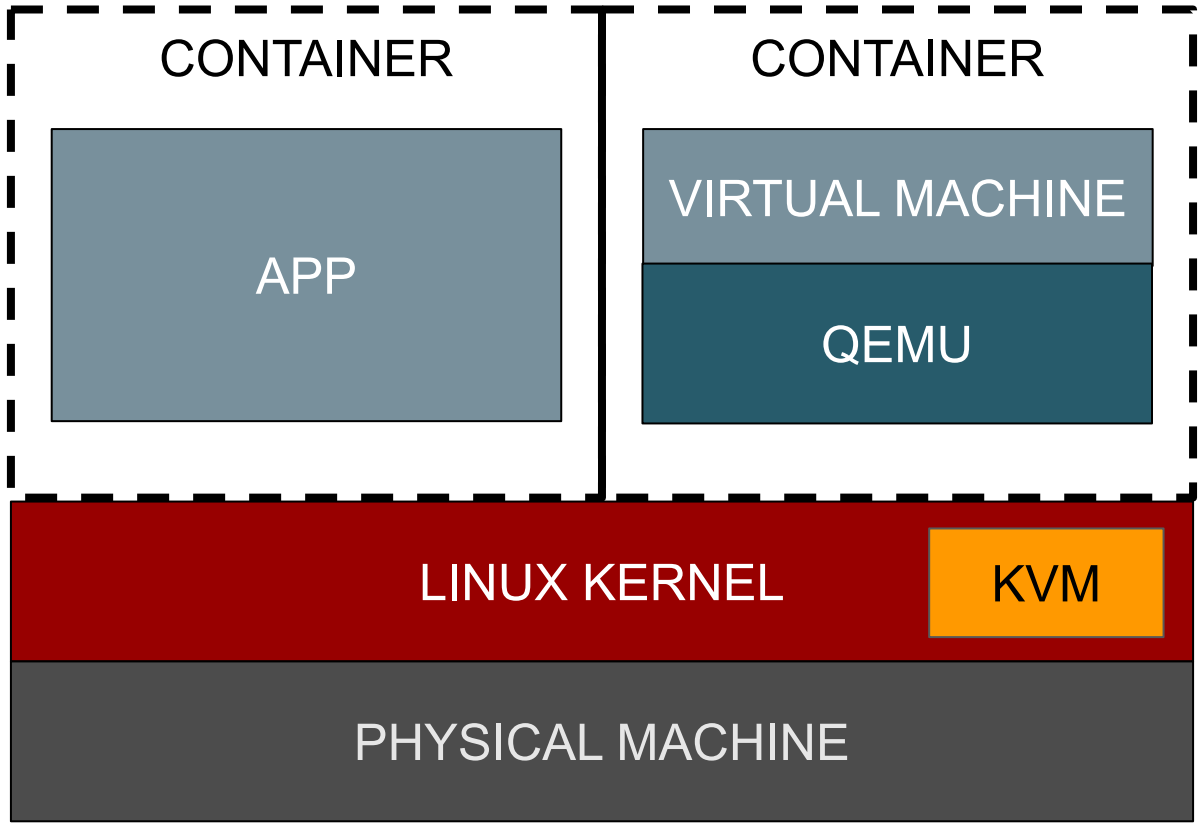
KubeVirt Technical Intro

- Virtualization Stack
- Ideal Environment
- Kubevirt Technical Intro
- Kubevirt Architecture
- Kubevirt Components
- Short Demo

Linux virtualization stack

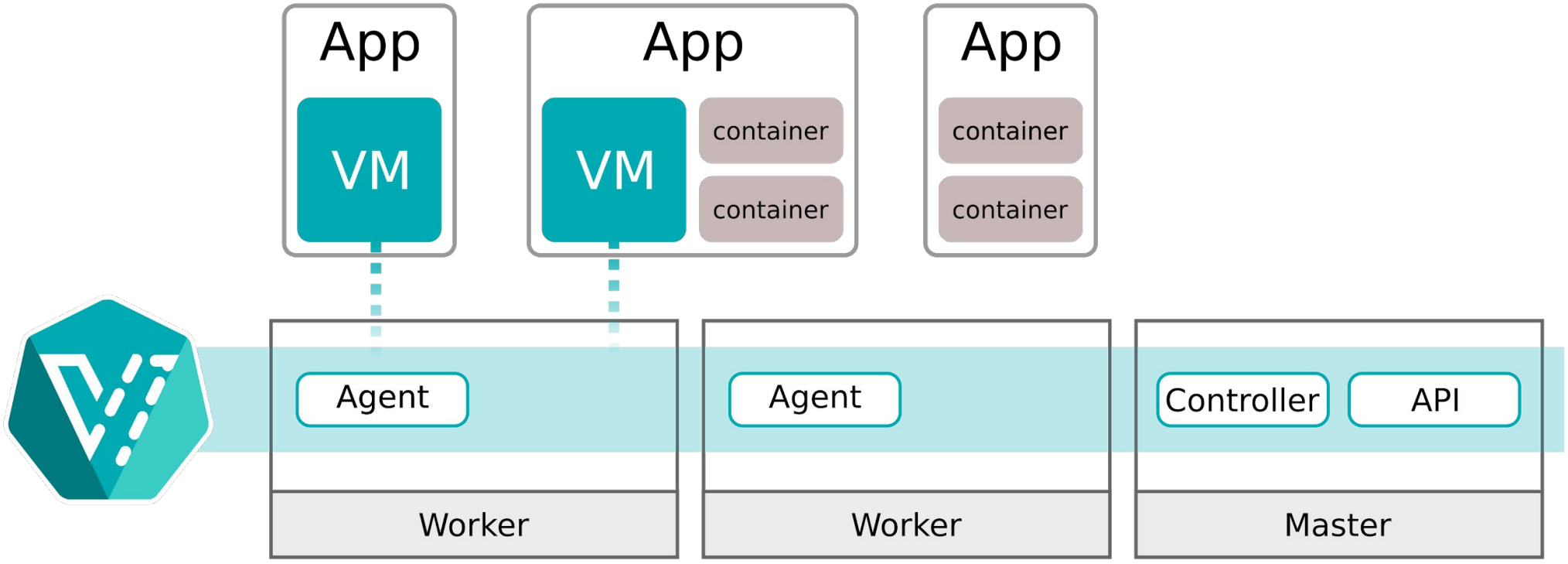


How to run Containers and VMs side by side?



Resultant VMs are able to run side by side directly on the same Kubernetes nodes as application containers.

Ideal Environment with KubeVirt



KubeVirt Components

- virt-api-server
- virt-Launcher
- virt-Controller
- virt-Handler
- libvirtd

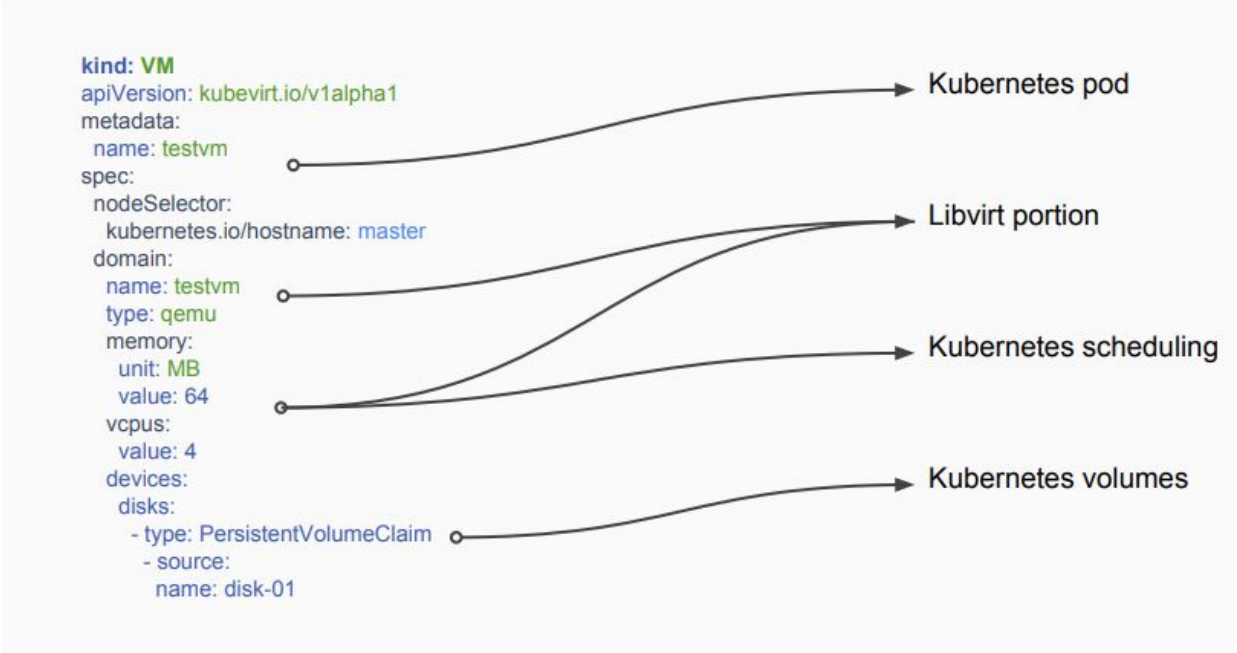
Virt-api-server

- **virt-api-server** serves as the entry point to kubevirt for all virtualisation related flows and takes care to update the virtualisation related custom resource definition (CRD)

- Dedicated API for virtualization
  - CRDs now, working on User API Server for custom (sub) resource types
  - Allows to define a VM resources & actions

- Responsible for the defaulting and validation of the provided VMs.

API EXAMPLE

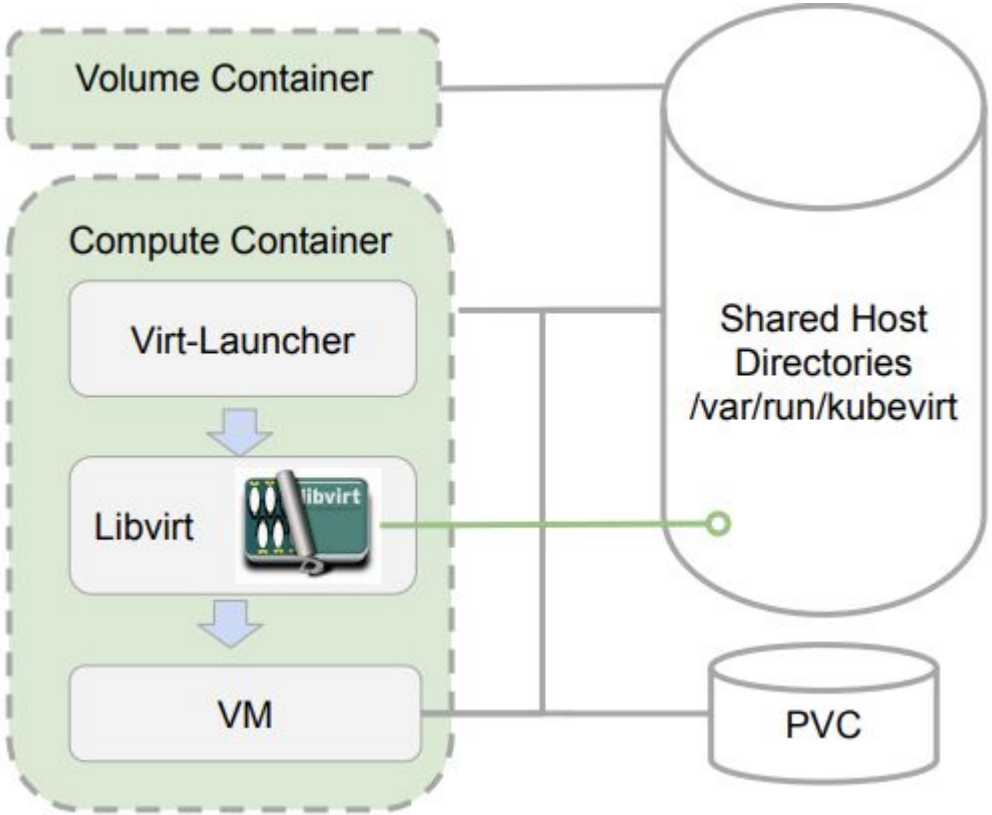


libvirtd

- is a toolkit to manage virtualization platforms
- is accessible from C, Python, Perl, Java and more
- is licensed under open source licenses
- supports KVM, QEMU, Xen, Virtuozzo, VMWare ESX, LXC, BHyve and more
- targets Linux, FreeBSD, Windows and OS-X
- An instance of libvirtd is present in every VM pod
- virt-launcher uses libvirtd to manage the life-cycle of the VM process.

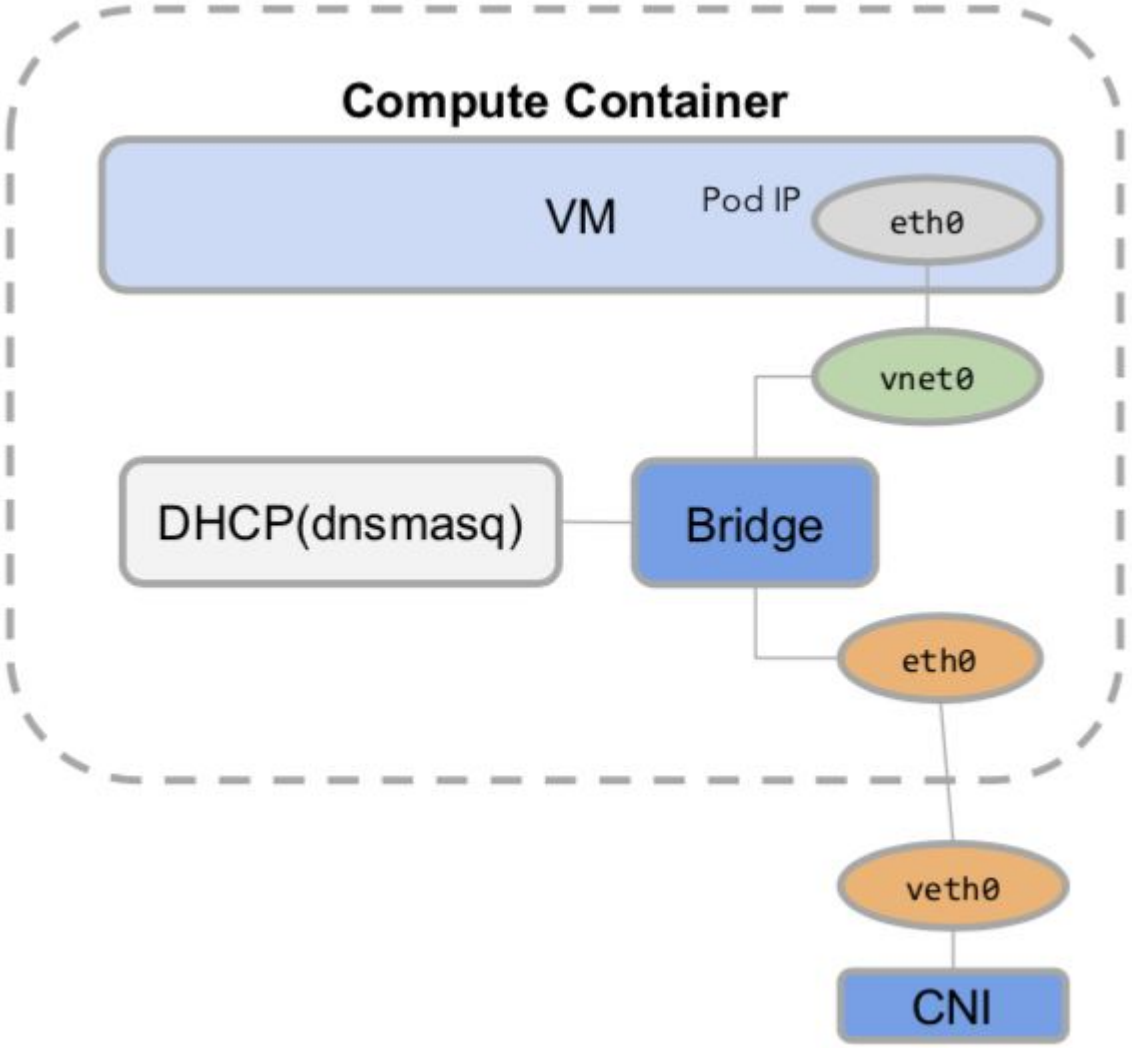
Virt-Launcher Pod Virtualization

- VM is inside a POD
- Launched by Virt-Launcher
- Uses Libvirt
  - KVM where available
  - Emulation where not
    - AWS, GCP
    - Nested VM
- Volume container
  - Unwrapping docker images to VM images
- Other containers
  - Sidecars as required
  - Infra container: liveness check



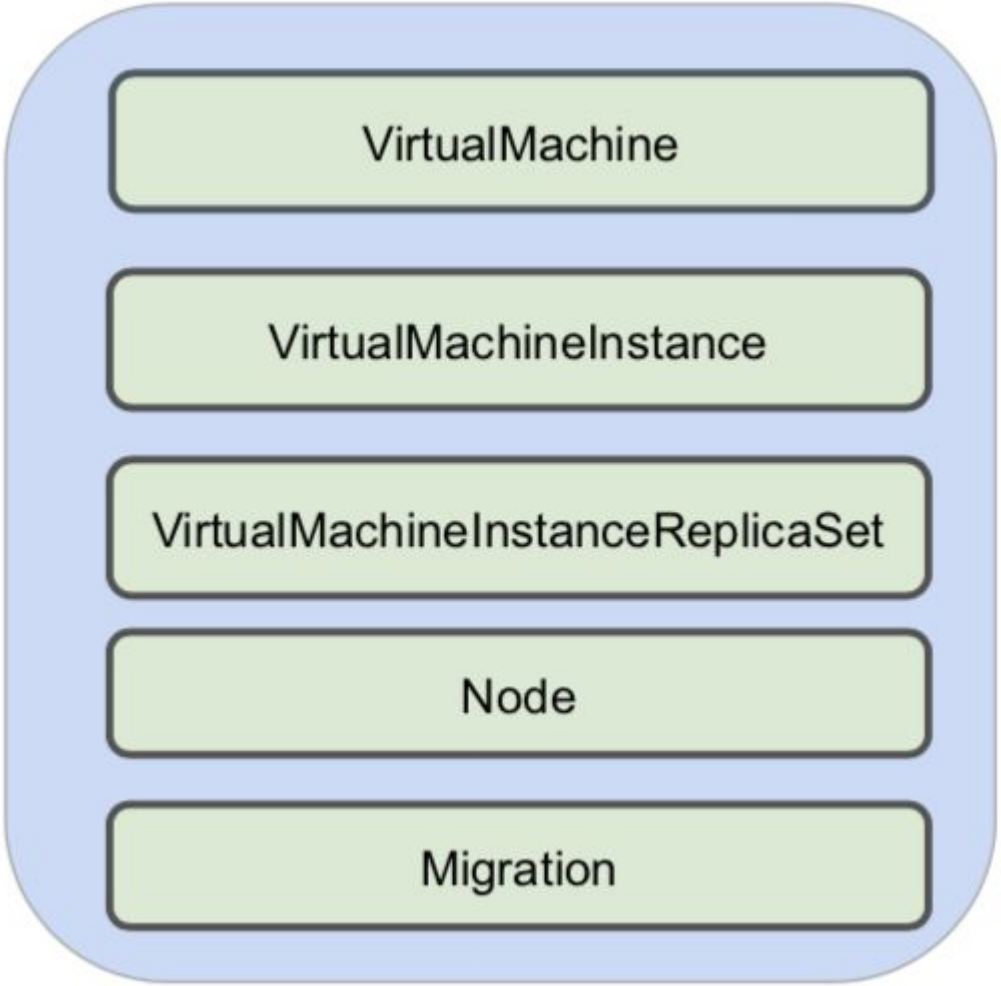
Virt-Launcher Pod Networking

- Virt-Launcher creates a dnsmasq on a link-local address
- Transfers the IP to the VM
- The Pod itself is without networking!!



Virt-Controller

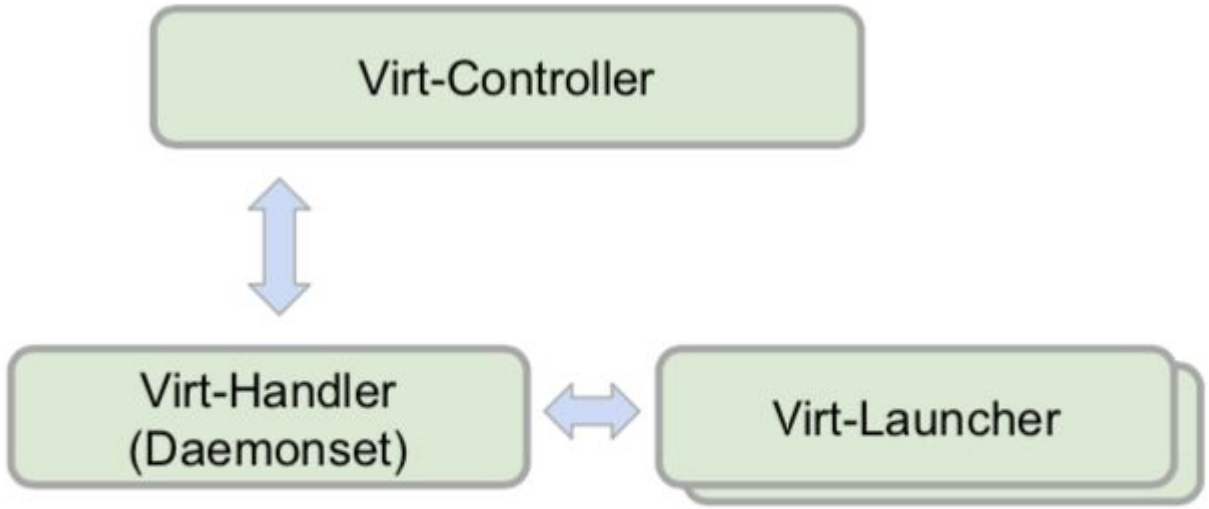
- Each Object has a corresponding controller
- VirtualMachine controller delegates most to VirtualMachineInstance
- Fairly comprehensive set of objects and more being discussed
  - VMGroups



Virt-Handler

- Is a Daemonset
- Acts as a minion
- Responsible for:
  - Stop
  - Update
  - Status
  - Restart
- Communicates to Libvirt via socket

  /var/run/kubevirt host mount



Demo

- Basic virtual machine launch

Demo

- CentOS virtual machine with Networking

Booting Options

Ephemeral Disks

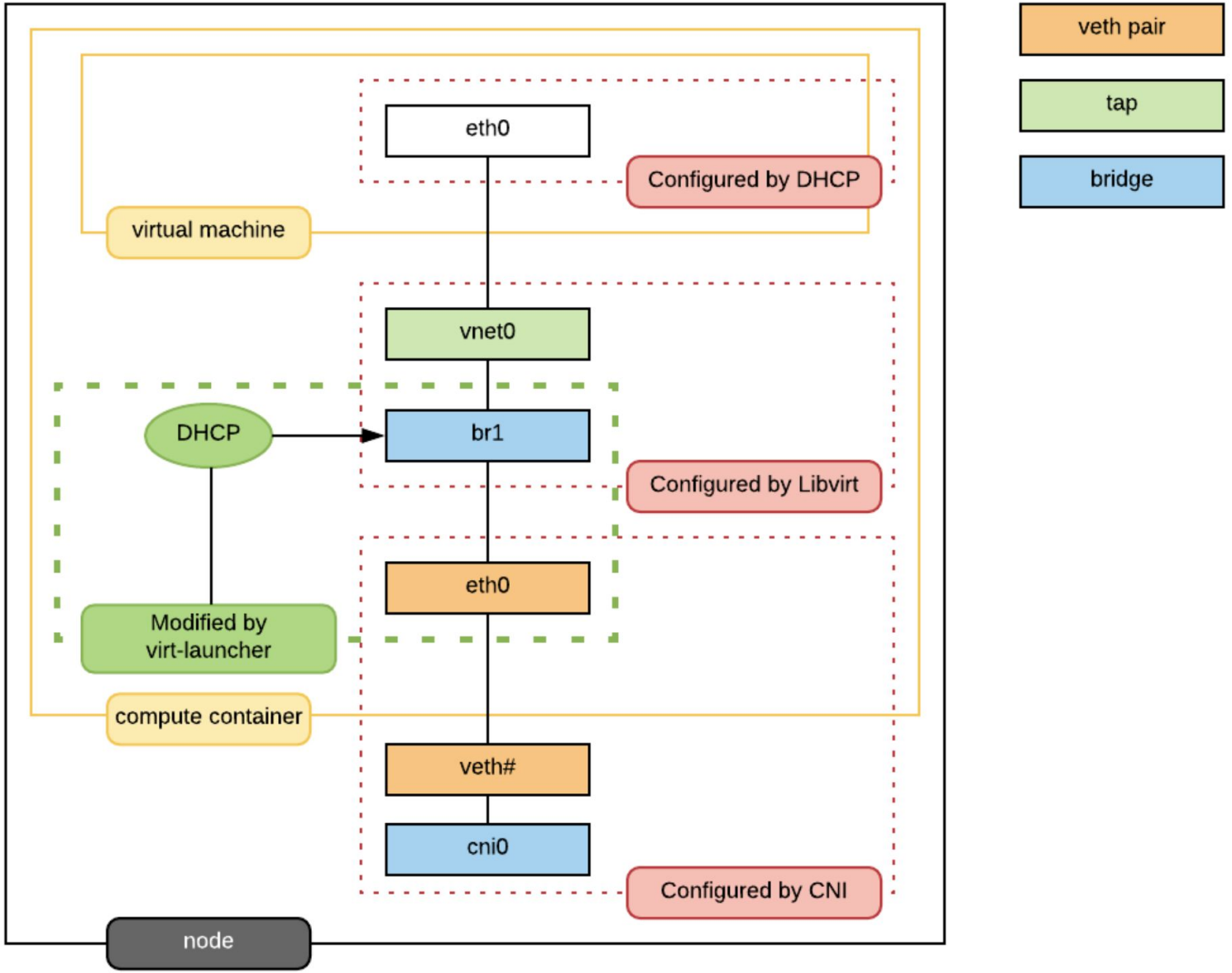
- **Immutable VMIs**
- **Loose changes across reboots**
- **Container Image embed VM images under /disk directory**

Persistent Disk

- **Data Volume**
- **Copy registry disk into a Data Volume**

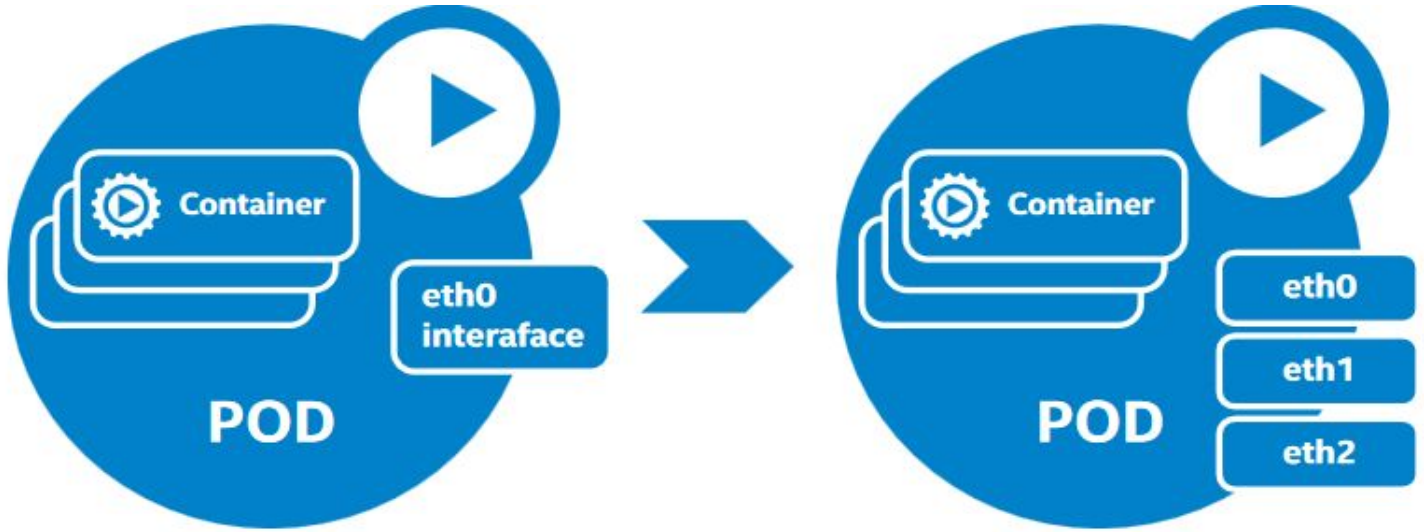
```
cat << END > Dockerfile
FROM scratch
ADD centos7.qcow2  /disk
END

docker build -t vmdisks/centos:latest .
docker push vmdisks/centos:latest
```

KubeVirt Networking



Multus CNI

- Multiple networking interfaces
- Multus + other CNI plugins
- Multus: behaves as a broker and arbiter of other CNI plugins
- Other CNI plugins: as master plugin, is used to configure and manage the primary network interface (eth0)

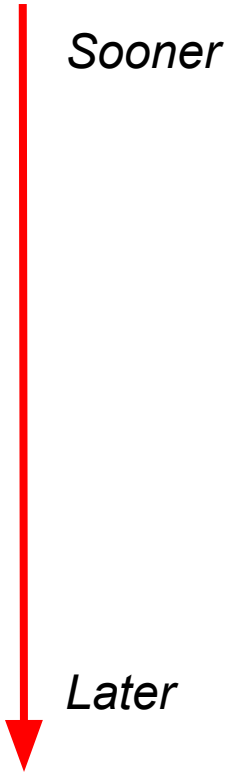


KubeCon
CloudNativeCon
North America 2019
What Next?

Future Plans

Highlights (not an exhaustive list!):

- Better support for deterministic workloads:
  - CPU Pinning
  - NUMA Topology Alignment
  - IO Thread pinning
- Storage-assisted snapshot and cloning.
- Forensic virtual machine capture
- GPU passthrough
- Policy-based live migration and additional migration modes.
- Hotplugging of CPUs, RAM, disks, and NICs (not necessarily in that order!).



Many of these features rely on enhancing Kubernetes itself!

Want to go deeper?

- **KubeVirt Deep Dive: Virtualized GPU Workloads on KubeVirt**
  - Wednesday 20th November - 10:55 AM - 11:30 AM
  - Room 1AB - San Diego Convention Center
  - https://sched.co/VnjX

References

- https://kubevirt.io
- https://github.com/intel/multus-cni
- https://www.cncf.io/wp-content/uploads/2019/09/KubeVirt-CNCF-Webinar

KubeCon
CloudNativeCon
North America 2019