PLATFORM9

KubeVirt -
Beyond Containers
Back to VMs !!

Roopak Parikh | @roopak_parikh | Platform9

Josh Hurt | Kubernetes Engineer | Platform9

Agenda

- Introduction to KubeVirt
- Use Cases
- How To
- Architecture
- Demo
- Q&A

What is KubeVirt - An Introduction

What is KubeVirt

- KubeVirt is a set of CRDs and controllers (Operator)
- A way to run Virtual Machines on Kubernetes!
- Uses the same Kubernetes native bits:
  - Scheduling
  - Storage
  - Networking
  - Monitoring
  - Tooling - Kubectl

KubeVirt - About

- Started at RedHat in 2016
- Open sourced in 2017
- Apache 2.0 License
- KubeVirt - A candidate for CNCF Sandbox

- https://github.com/kubevirt
- #virtualization on Slack
- 1400+ Stars
- Contributions from: Akamai, Apple, Cisco, Cloudflare, Loodse and others

KubeVirt - User Voice

gonzolino commented on Jul 31 • edited

At SAP we use KubeVirt to run K8s worker nodes. Together with https://github.com/kubevirt/cloud-provider-kubevirt we use Kubevirt as a "cloud provider" to run K8s clusters.

By putting KubeVirt on a bare metal cluster and running additional K8s clusters on KubeVirt VMs we give our developers many of the features they know from cloud providers (e.g. LoadBalancers, cluster autoscaling) in an on-prem environment.

3

bharatnc commented 29 days ago

I strongly recommend adding KubeVirt to the CNCF Sandbox.

At Cloudflare, we use Kubevirt VMs for test environments that are closer to production parity than containers would be - and because these VMs can run on Kubernetes, engineers are able to utilize unused capacity to maximize test efficiency, rather than being restricted to a set of costly, dedicated hardware.

We also use VMs to run the majority of our production CI/CD build workloads and the KubeVirt has been essential for moving our build workloads from traditional dedicated hardware to auto-scaled Kubernetes VMs.

It has been exciting to work with the project. Along the way we have contributed to it and KubeVirt maintainers have been really cooperative and helped us a lot with feature requests and fixes.

2

visheshtanksale commented 28 days ago

We at Nvidia are using Kubevirt to develop a new VM management platform. We have some VM workloads which cannot be containerized. Kubevirt's ability to orchestrate VMs within Kubernetes framework, helps us leverage the same stack for VMs and containers. This makes Kubevirt a perfect solution for our use case.
The Kubevirt community is very active and helpful. We are actively looking to contribute GPU support for Kubevirt VM.
I highly recommend Kubevirt to CNCF sandbox

3

KubeVirt - Use Cases

- One Orchestration platform
  - Standardization on operational model, processes, and tooling
- Application Modernization
  - Applications that are in transition from being Monolithic to Microservices
- Virtual Network Function Modernization
  - Network Functions will be running in VMs: Custom kernel modules, specific kernel version, specific network drivers
  - Other applications in the *NFV* stack can easily run on containers
  - Strong desire to move microservices

KubeVirt - Use Cases Contd.

- Turtles all the way down: Kubernetes on Kubernetes
  - Using VMs running on KubeVirt as the building-block for ‘workload’ Kubernetes clusters.
  - To allow self-service
  - Using KubeVirt cloud-provider

- DevTest Cloud
  - Immutable VMs
  - Self-Service
  - Increase velocity

KubeVirt - Concepts

Kubevirt - Compute

- VirtualMachine
  - The immortal VM object
  - Just an object, there are no associated pods/processes
- VirtualMachineInstance
  - Instantiation of a VM when it is modified/started
- VMI Preset
  - Same idea as a “flavor” but includes ability to set storage/network params too
  - Individual VMIs can override specific values, accepting the rest of the preset values as defaults

```
apiVersion: kubevirt.io/v1alpha3
kind: VirtualMachine
metadata:
  name: testvm
spec:
  running: false
  template:
    metadata:
      labels:
        kubevirt.io/size: small
        kubevirt.io/domain: testvm
    spec:
      domain:
        devices:
          disks:
            - name: containerdisk
              disk:
                bus: virtio
            - name: cloudinitdisk
              disk:
                bus: virtio
          interfaces:
          - name: default
            bridge: {}
        resources:
          requests:
            memory: 64M
```

KubeVirt - Images

VM booting options

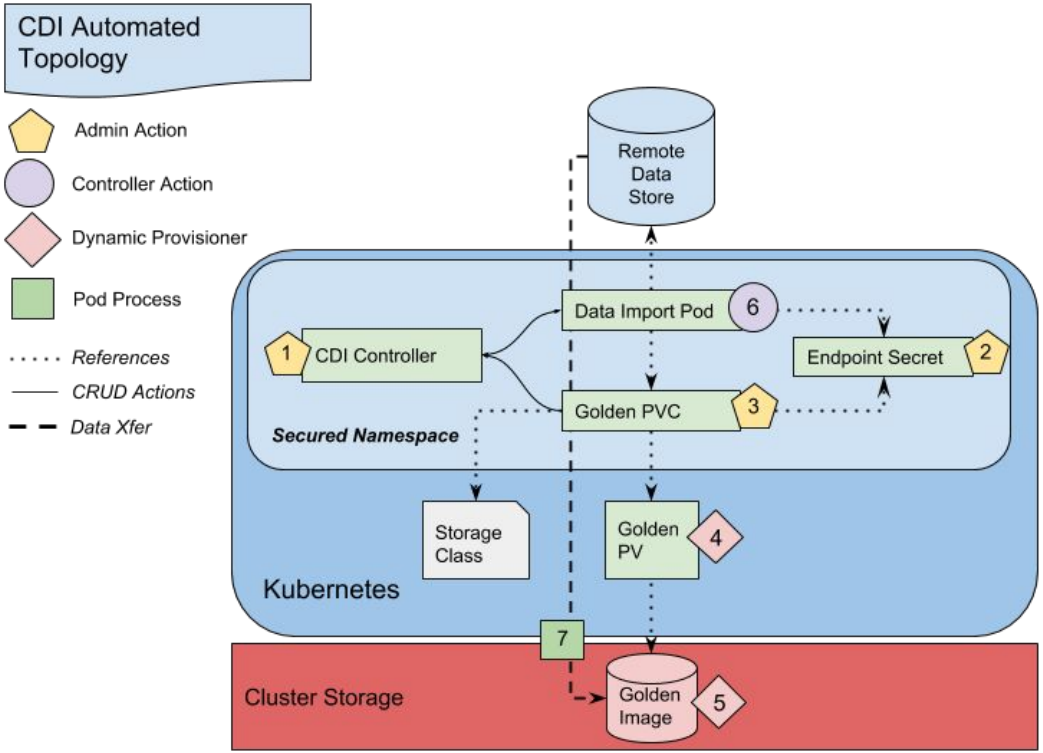
- Ephemeral Disk
  - Immutable VMIs
  - Lose changes across reboots
  - Container image embed VM images under /disk directory
- Persistent Disk
  - DataVolume
  - Copy Registry disk into a DataVolume

```
cat << END > Dockerfile
FROM scratch
ADD fedora25.qcow2 /disk
END
docker build -t vmdisks/fedora25:latest .
docker push vmdisks/fedora25:latest
```

```
kind: VirtualMachineInstance
spec:
  domain:
    devices:
      disks:
      - type: ContainerDisk:v1alpha
        - source:
          name: vmdisks/fedora25:latest
        - target:
            device: sda
```

Kubevirt - CDI

- 2nd project under KubeVirt org: containerized-data-importer
  - solves problem of “how do I load in compatible images for my VMs?”
- CRD/Controller which sits on top of PVCs



Kubevirt - Storage

- Cloud-init, emptyDisks, hostDisks, DataVolume…
  - also k8s primitives such as ConfigMap, Secret, ServiceAccount
    Note: updates to these are not seen by the VM
- Otherwise nothing special - uses k8s-native Storage concepts & tools
- Enables live migration if setup correctly
  - ReadWriteMany AccessMode
  - Also dependent on networking (ex. bridge disallows LiveMigration)

Kubevirt - Networking

- By default uses Pod networking
  - Makes interoperability possible
- CNI (extra)
  - Multus
  - Genie
- SR-IOV
  - NFV use cases

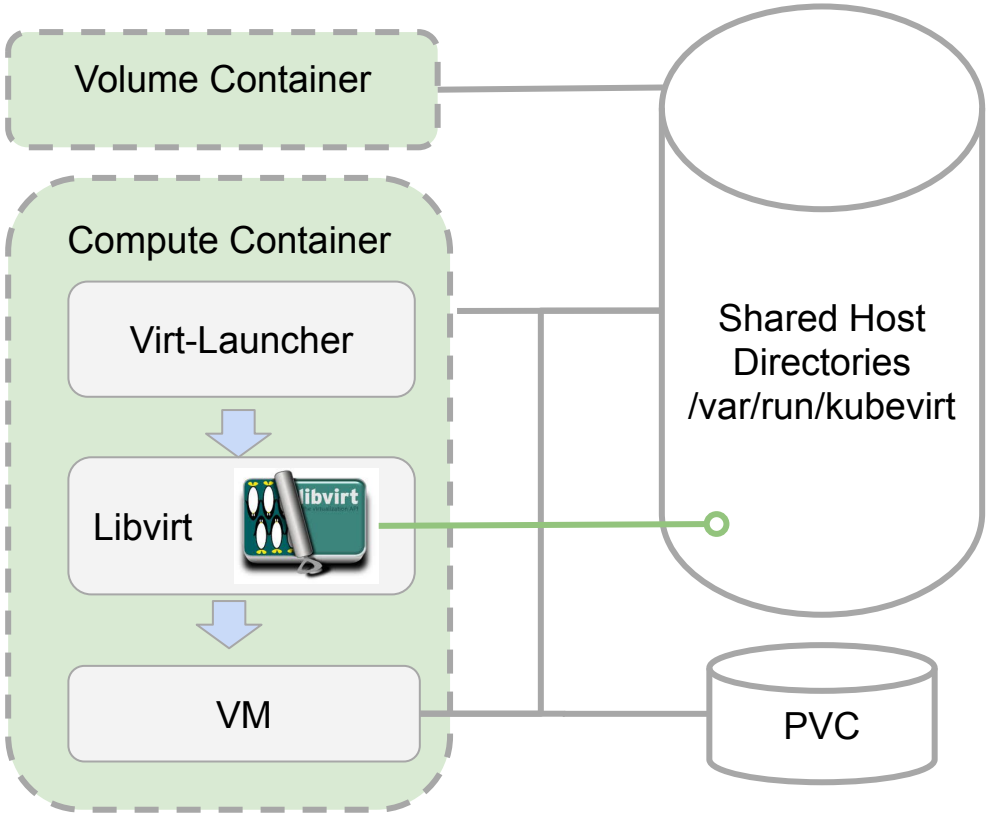
```
kubectl get pod -o wide
NAME                            READY     STATUS    IP
my-nginx-6fbb694897-v9gfl       1/1       Running   10.20.58.8
virt-launcher-testvm-jfkx9      2/2       Running   10.20.46.11
```

```
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8973 qdisc
pfifo_fast qlen 1000
    link/ether 1e:bf:4d:2c:01:84 brd ff:ff:ff:ff:ff:ff
    inet 10.20.46.11/24 brd 10.20.46.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::1cbf:4dff:fe2c:184/64 scope link tentative flags
08
       valid_lft forever preferred_lft forever
```

Kubevirt - Architecture

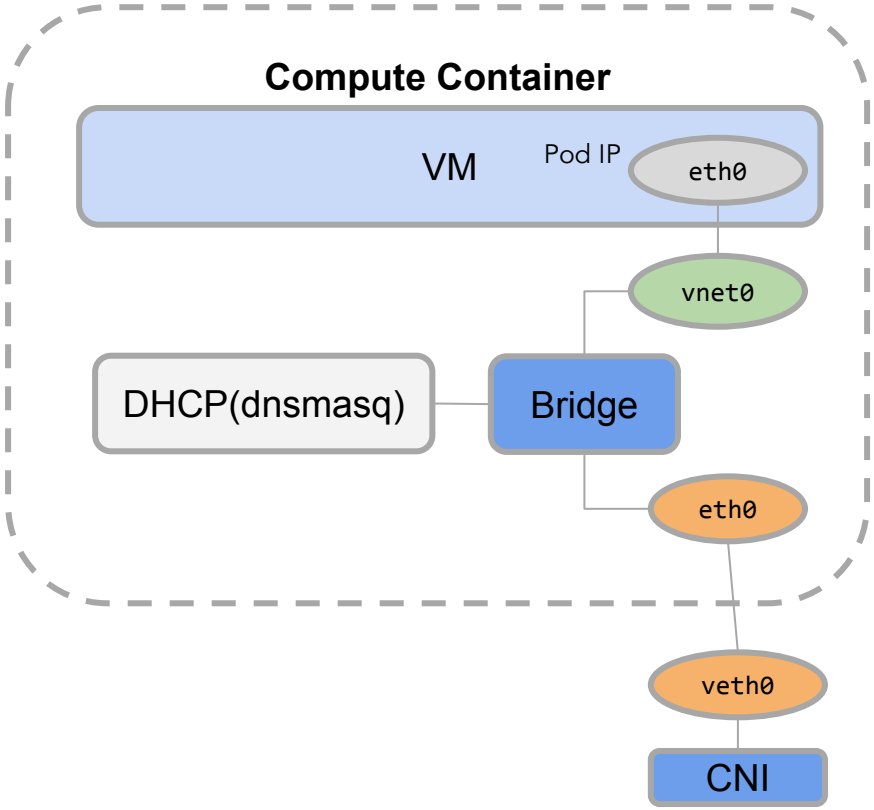
Virt-Launcher Pod - Virtualization

- 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



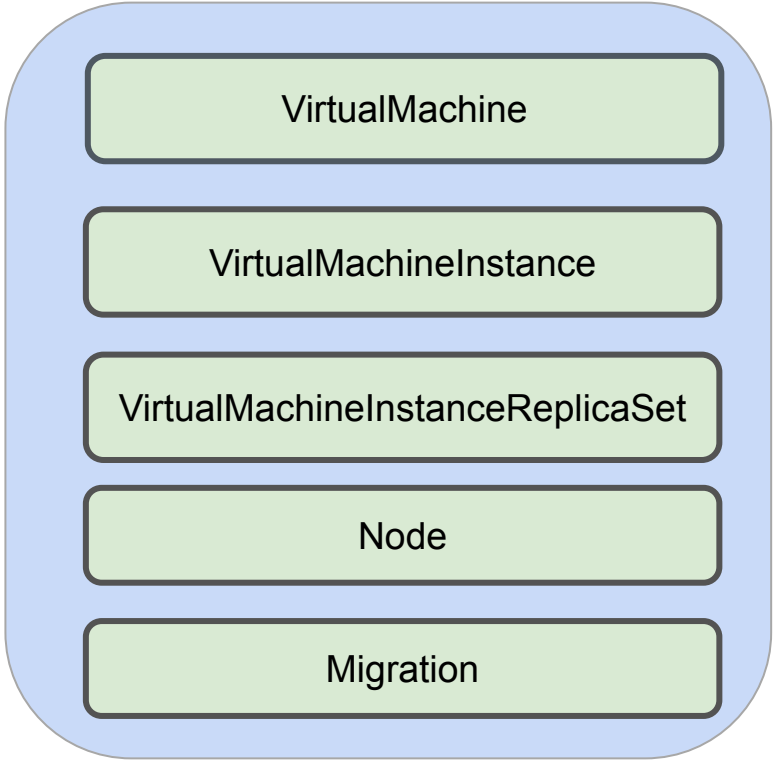
Virt-Launcher Pod - Networking

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



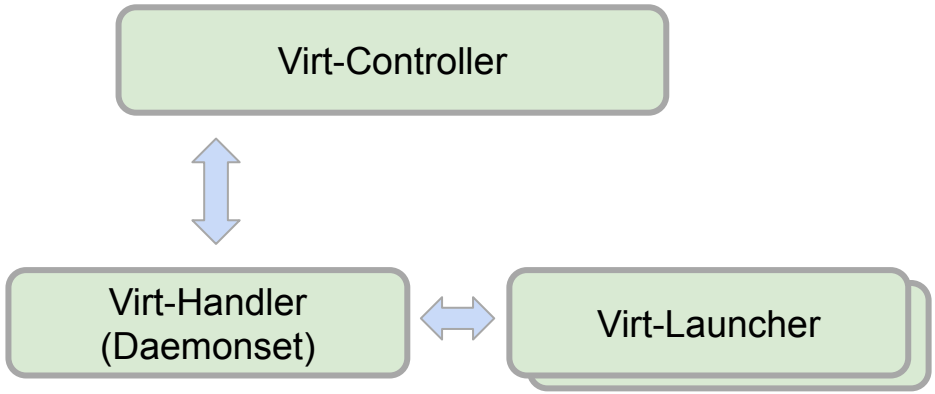
Virt-Controller - All the Controller(s)

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

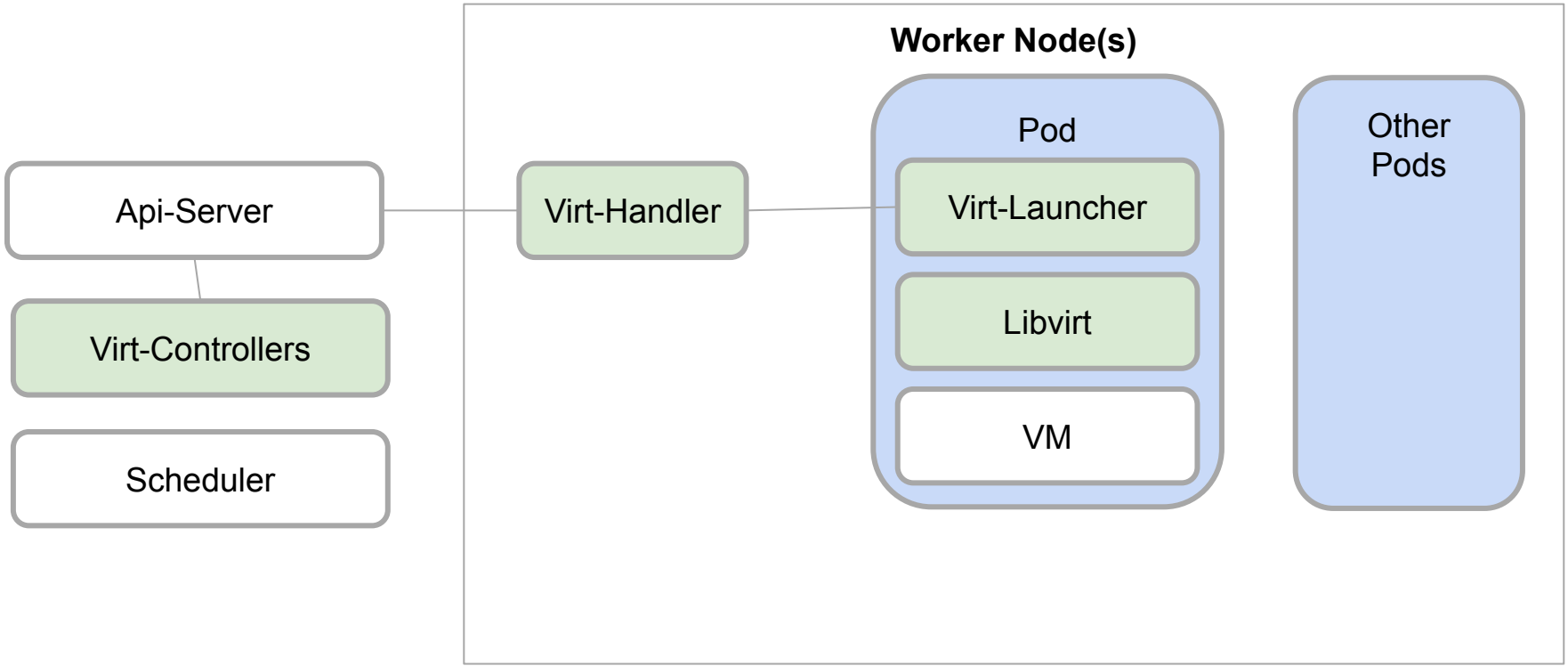


Virt-Handler

- Is a Daemonset
- Acts as a minion
- Responsible for:
  - Stop
  - Update
  - Status
  - Restart
- Communicates to Libvirt via socket */var/run/kubevirt* host mount



The complete picture



KubeVirt - References

- The website: https://kubevirt.io/
- Examples: https://github.com/kubevirt/kubevirt/tree/master/examples
- Web-UI: https://github.com/kubevirt/web-ui-operator

Demo:
Container - VM Connected

Q&A