KubeVirt

Making running virtual machines in a Kubernetes cluster a mainstream activity

containerdays.io 2022

Daniel Hiller - Red Hat

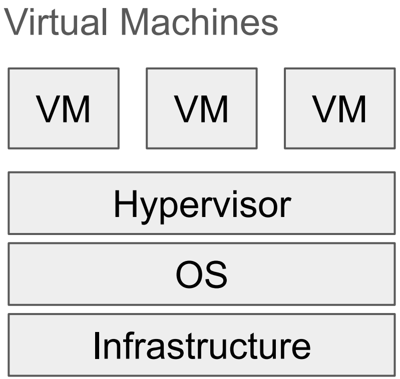
Who am I?

- Daniel Hiller
- Software Engineer @ Red Hat
- Maintaining CI infra and automation for KubeVirt org

- www.dhiller.de
- twitter.com/dhill3r
- github.com/dhiller

What is this about?

What is this about?



Virtual Machines

VM VM VM

Hypervisor

OS

Infrastructure

What is this about?

Virtual Machines

*“**Kubernetes***

is a** portable, extensible, open source **platform

***for managing containerized workloads** and services,*

that facilitates both declarative configuration and automation.”

source

What is this about?

“Kubernetes

is a *portable, extensible, open source* ***platform***

for managing containerized workloads *and services,*

that facilitates both declarative configuration and automation.”

source

Virtual Machines

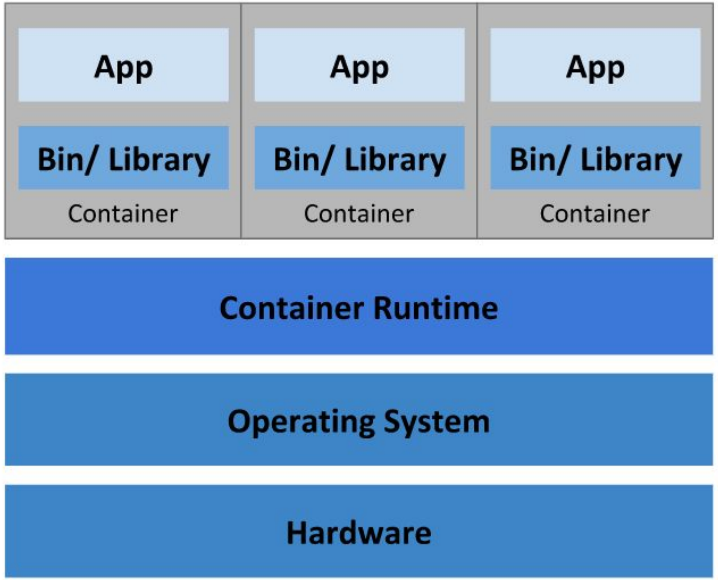
“a virtual machine (VM) is **the** virtualization/**emulation of a computer system**.

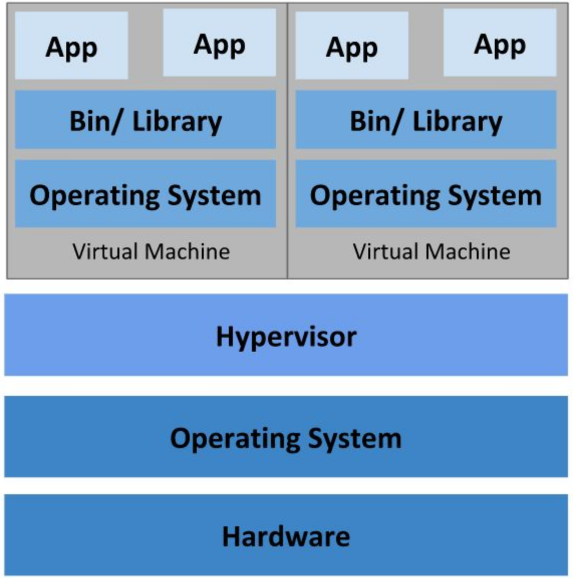
Virtual machines are **based on computer architectures** and

provide functionality **of a physical computer**.”

source

Containers vs. Virtual Machines





source

The stories (imagined)

What if I could get rid of that computer sitting in the closet?

What if I could put that VM that is sitting on an old computer somewhere else?

The stories (imagined)

“It works on that machine! - Then let’s ship that”

:)

What is KubeVirt?

What is KubeVirt?

“**KubeVirt** technology

addresses the needs of development teams that have adopted or want to adopt Kubernetes

but **possess existing Virtual Machine-based workloads that cannot be easily containerized**.”

source: https://kubevirt.io/

What is KubeVirt?

More specifically,

the technology **provides a unified development platform** where

developers can build, modify, and deploy applications

residing in **both Application Containers as well as Virtual Machines in a common, shared environment**.

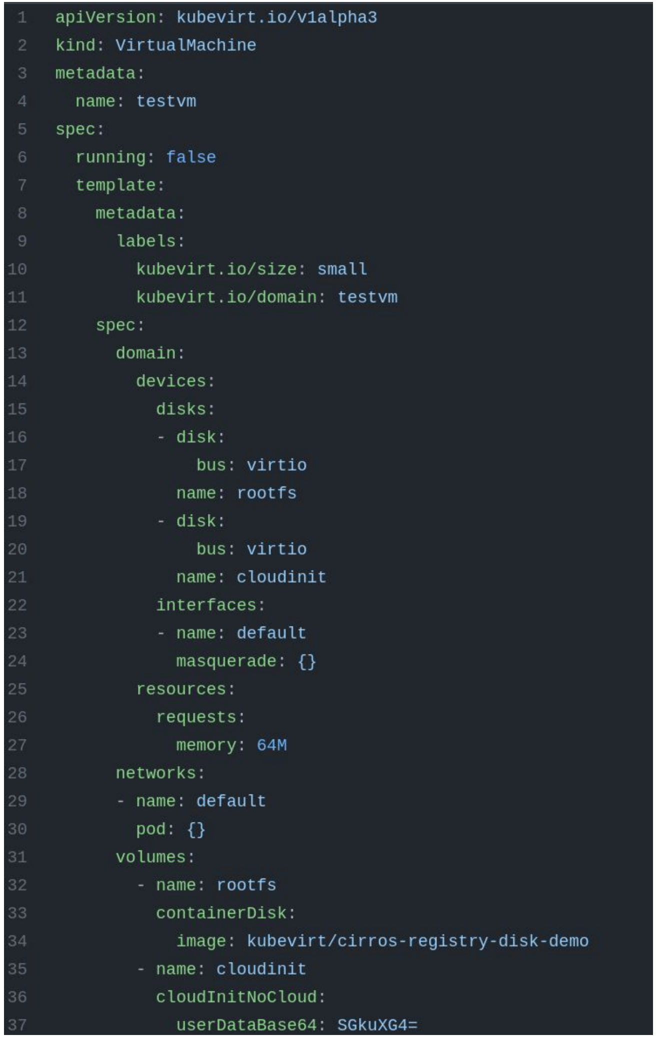
source: https://kubevirt.io/

What is KubeVirt?

KubeVirt makes it possible to

- Manage VMs in the same environment as containers
- Manage VMs as k8s objects

What is KubeVirt?



```yaml
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:
          - disk:
              bus: virtio
            name: rootfs
          - disk:
              bus: virtio
            name: cloudinit
          interfaces:
          - name: default
            masquerade: {}
        resources:
          requests:
            memory: 64M
      networks:
      - name: default
        pod: {}
      volumes:
        - name: rootfs
          containerDisk:
            image: kubevirt/cirros-registry-disk-demo
        - name: cloudinit
          cloudInitNoCloud:
            userDataBase64: SGkuXG4=
```

So much for theory…

The story (imagined)

- Prepare the VM for import
- Convert the disk image to an importable format
- Import the image into kubevirt
- Create the VM with the image
- Success

The story (reality)

- **Prepare the VM to be imported inside VirtualBox**
- Convert the disk image to an importable format
- Import the image into kubevirt using cdi-uploadproxy
- Create the vm with the image
- Success

The story (reality)

- Prepare the VM to be imported inside VirtualBox
- **Convert the disk image to an importable format**
- Import the image into kubevirt using cdi-uploadproxy
- Create the vm with the image
- Success

The story (reality)

- Prepare the VM to be imported inside VirtualBox
- Convert the disk image to an importable format
- **Import the image into kubevirt using cdi-uploadproxy**
- **Create the vm with the image**
- Success

The story (reality)

- Prepare the VM to be imported inside VirtualBox
- Convert the disk image to an importable format
- Import the image into kubevirt using cdi-uploadproxy
- Create the vm with the image
- **Success**

Demo time!

Demo environment

- kubevirt/kubevirtci
- Dockerized cluster nodes
- Pre-pulled images to reduce component spin up time
- Enabled components:
  - containerized-data-importer to enable uploading images
  - rook-ceph as storage provider
  - Prometheus and Grafana for monitoring

Live demo

- VM Import
- Monitoring
- Snapshot and Restore
- Live Migration

About the project

KubeVirt

- Open source (APL2.0) CNCF project in incubation state
- Adopted by several vendors and end users
- 3.4k GitHub stars, 230+ contributors, 800+ forks, 5k+ PRs
- Monthly release schedule
- Tested on latest three minor Kubernetes releases
- Contributions from companies like SuSe, Nvidia, ARM

An incomplete list of features

Shown

- VM image import (i.e. vbox)
- Live migration
- Snapshot and restore
- Monitoring
- Serial console and graphical console access (vnc, rdp? source)

Also

- HotPlug Volumes
- vGPU and Mediated Devices
- Node assignment
- Cloud-init and sysprep
- Zero downtime rolling updates

Note: Some features need to get enabled through a feature gate, here’s a list

Outlook to features

- Memory encryption
- Multi-arch clusters
- VM export

Easy to install

```
# Point at latest release
$ export RELEASE=$(curl https://storage.googleapis.com/kubevirt-prow/release/kubevirt/kubevirt/stable.txt)

# Deploy the KubeVirt operator
$ kubectl apply -f https://github.com/kubevirt/kubevirt/releases/download/${RELEASE}/kubevirt-operator.yaml

# Create the KubeVirt CR (instance deployment request) which triggers the actual installation
$ kubectl apply -f https://github.com/kubevirt/kubevirt/releases/download/${RELEASE}/kubevirt-cr.yaml

# wait until all KubeVirt components are up
$ kubectl -n kubevirt wait kv kubevirt --for condition=Available
```

source

Live demo

- VM Import
- Monitoring
- Snapshot and Restore
- Live Migration

Findings

- Preparation of an existing VM for import is the most effort
- Very old VMs (i.e. EOL OS such as Windows XP) will make trouble when trying to install virtio drivers
- Use `qemu-system-x86_64` to run the VM before importing (MAGIC happens)
- Need to better understand what `qemu-system-x86_64` is actually doing in order to specify the KubeVirt VM better
- Get sound working so my kids can enjoy the VM again ;-)

Q&A

Thank you for attending!

Have questions?

Doc links:

- **VM Import**
- **Monitoring**
- **Snapshot and Restore**
- **Live Migration**

KubeVirt welcomes all kinds of contributions!

- Weekly community meeting happening every Wednesday 3PM CET
- Links:
  - KubeVirt website
  - KubeVirt user guide
  - KubeVirt Contribution Guide
  - GitHub
  - Kubernetes Slack channels
    - #virtualization
    - #kubevirt-dev
  - Scripts and vm definition for this presentation: https://github.com/dhiller/containerdays.io-2022