

# Virtualização no Kubernetes com **KubeVirt**

[Online] Cloud Native São Paulo – Meetup #16

**Davi Garcia**

Cloud Solutions Architect



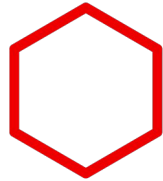
## KubeVirt

Sandbox Project na Cloud Native  
Computing Foundation (CNCF)

Através de APIs e runtimes,  
habilitar o Kubernetes a executar  
e gerenciar máquinas virtuais

**<https://kubevirt.io>**

# Princípios Orientadores



## **VMs ficam em pods**

VMs consomem recursos de onde o Kubernetes está fornecendo – pods.



## **Sem sobrecarga ou modificação de APIs**

VMs têm sua funcionalidade específica, portanto, uma API dedicada para expô-las através do K8S

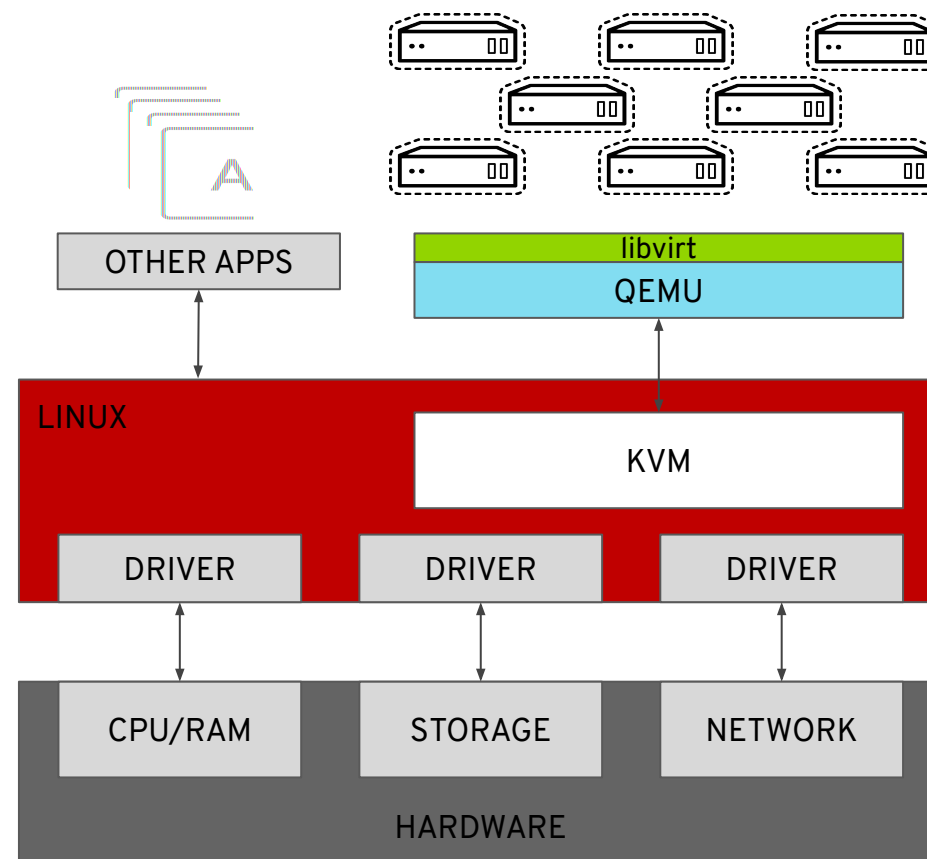


## **Prioridade na experiência Kubernetes-native**

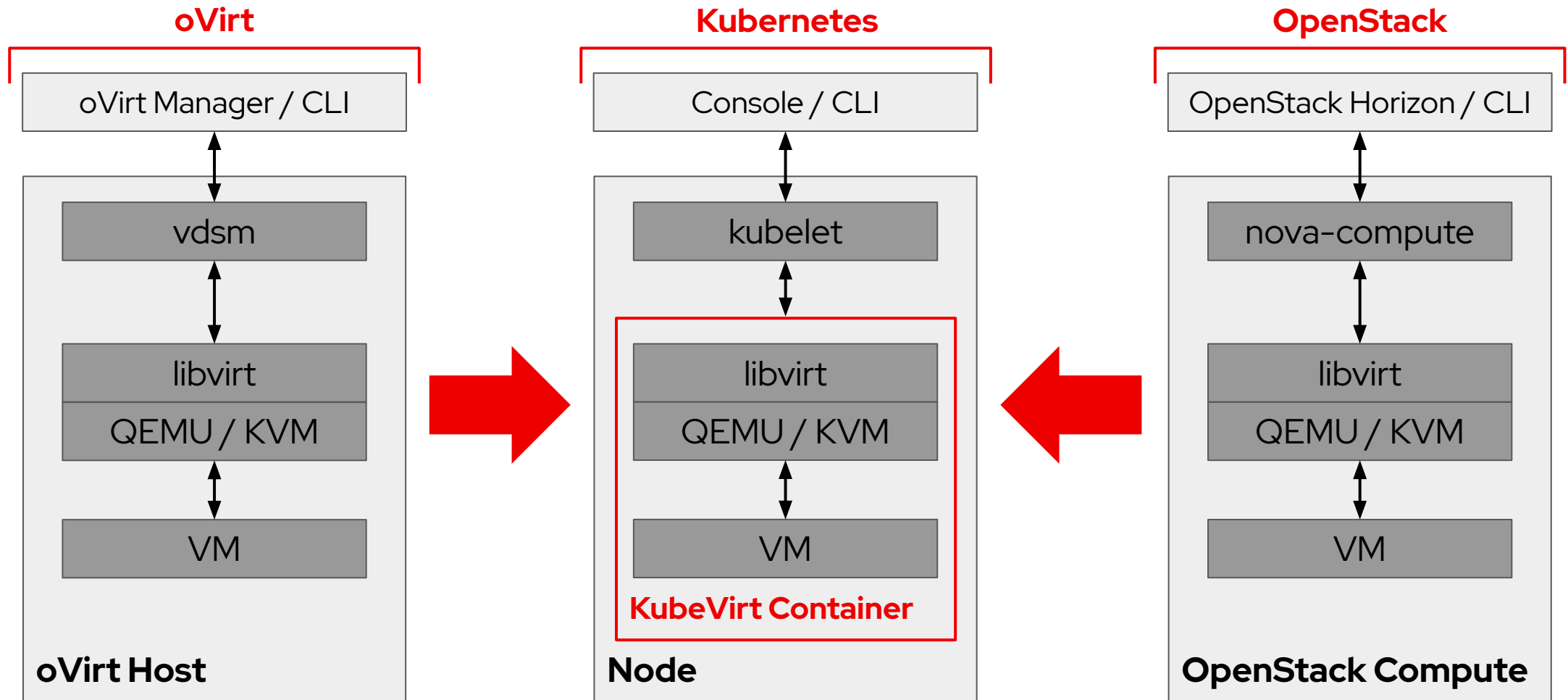
Para usabilidade, as funcionalidades da virtualização devem ser feitas seguindo filosofia do Kubernetes.

# VMs baseadas em KVM (em container)

- **KVM** é parte do kernel Linux
- **QEMU** usa **KVM** para executar máquinas virtuais (VMs)
- **libvirt** provê camada de abstração para gerenciamento
- Arquitetura que reutiliza experiência e componentes de outros projetos, como **oVirt** e **OpenStack**

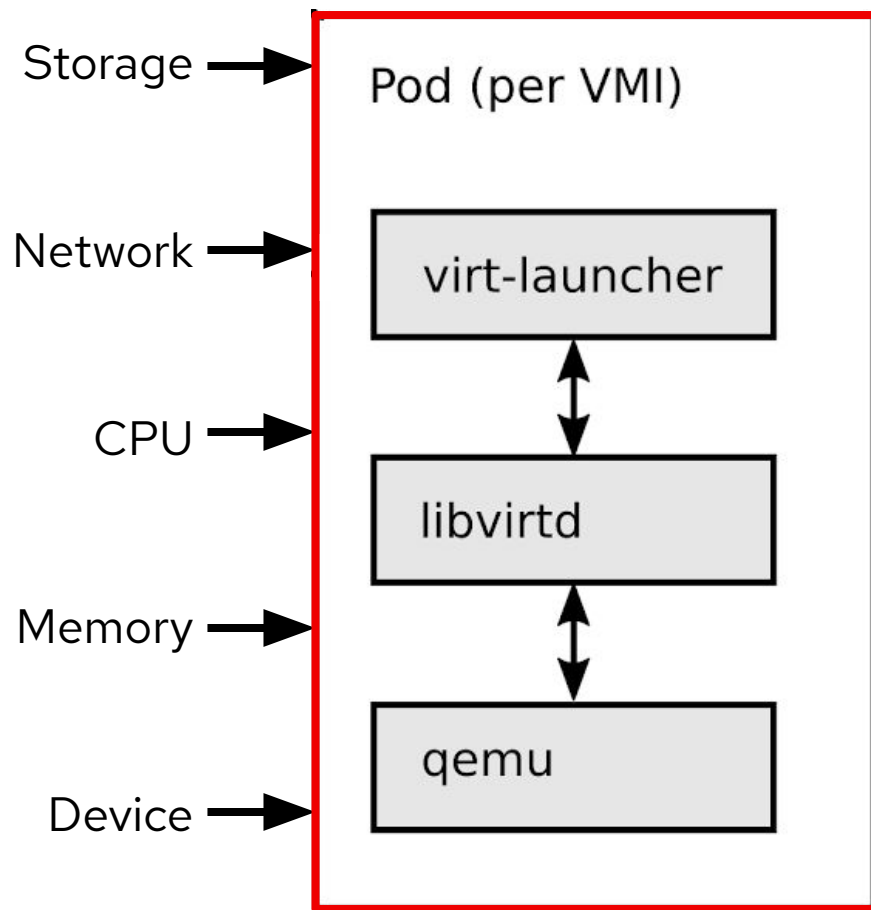


# Containerização do KVM



# VM em um pod

Containers para um modelo de recursos unificado



## Recursos Kubernetes

Toda VM é executada em um “launcher pod”. O launcher vai supervisionar, usando `libvirt`, e provendo integração com o ambiente do pod.

## Componentes Maduros

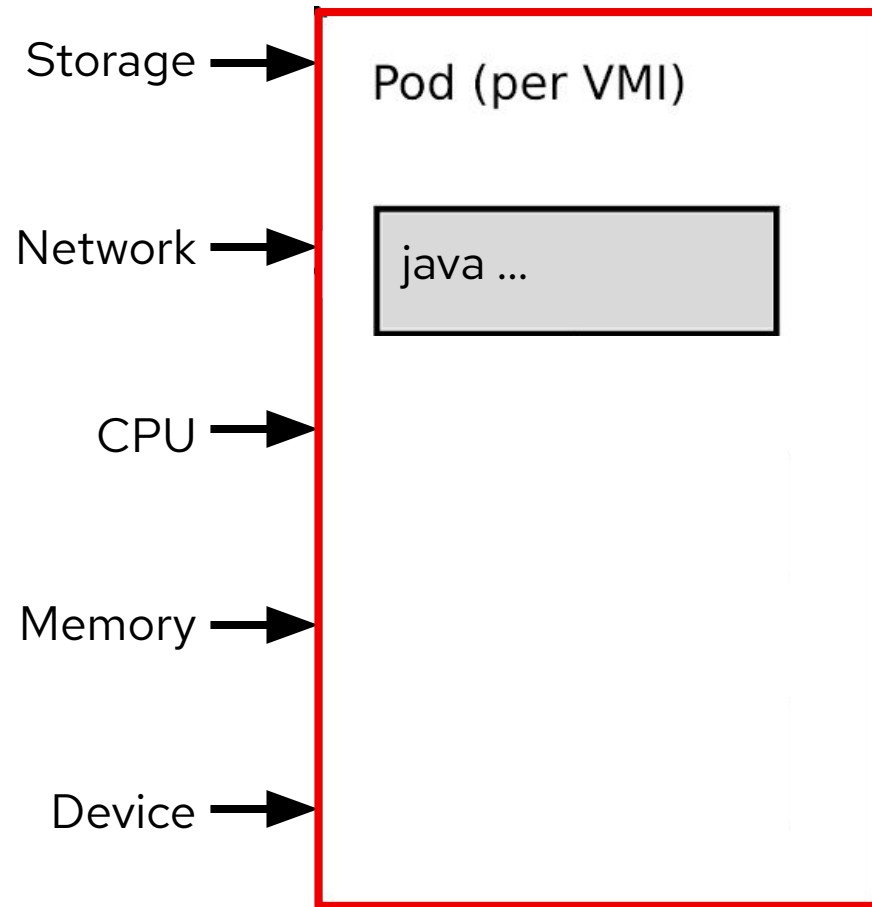
`Libvirt`, `qemu` e `KVM` são maduros, performáticos, proveem abstrações estáveis, e possuem overhead mínimo.

## Segurança

Pode aproveitar boas práticas do mundo Linux para proteger as cargas de trabalho virtualizadas (`SELinux` MCS, `SO Imutáveis`, `Isolamento KVM`, etc).

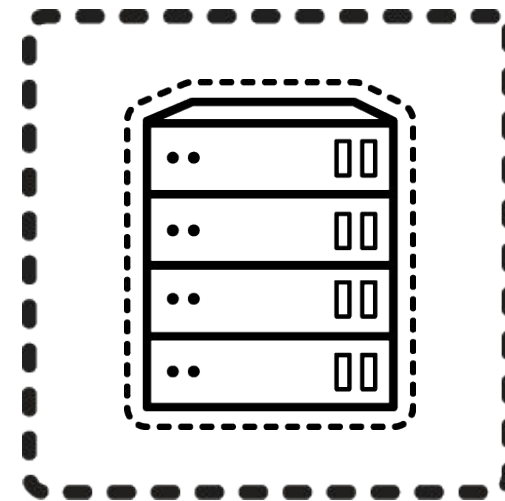
# VM em um pod

... é como qualquer outro processo para o Kubernetes



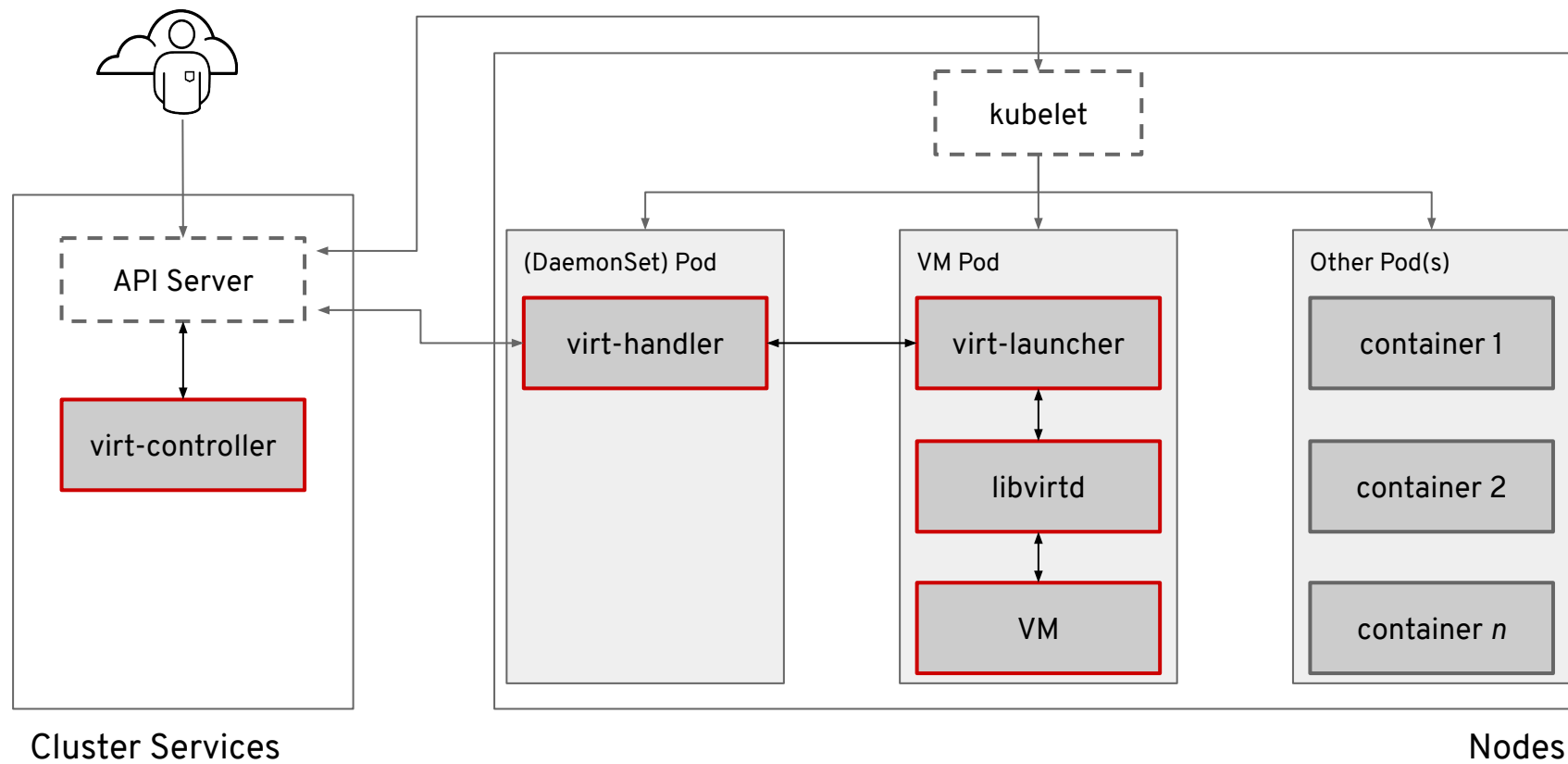
# Máquinas Virtuais Containerizadas

- **Herdam funções e funcionalidades do Kubernetes**
  - Escalonamento, alta disponibilidade, mapeamento de recursos (attach/detach)
- **Possuem as mesmas limitações das não-containerizadas**
  - CPU, RAM, etc. limitações ditadas pelo libvirt e QEMU
  - Sistemas operacionais guests como Linux e Windows
- **Armazenamento**
  - Usam Persistent Volumes Claims (PVCs) para os discos
  - Containerized Data Importer (CDI)
- **Conectividade**
  - Herda a rede do pod por padrão
  - Multus (CNI) habilita conexão externa direta



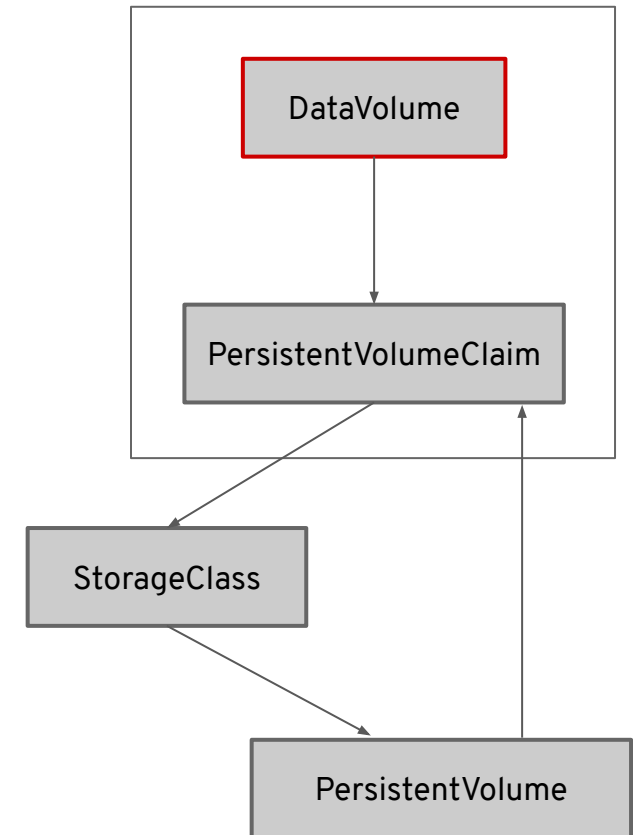


# Visão Arquitetural de Alto Nível



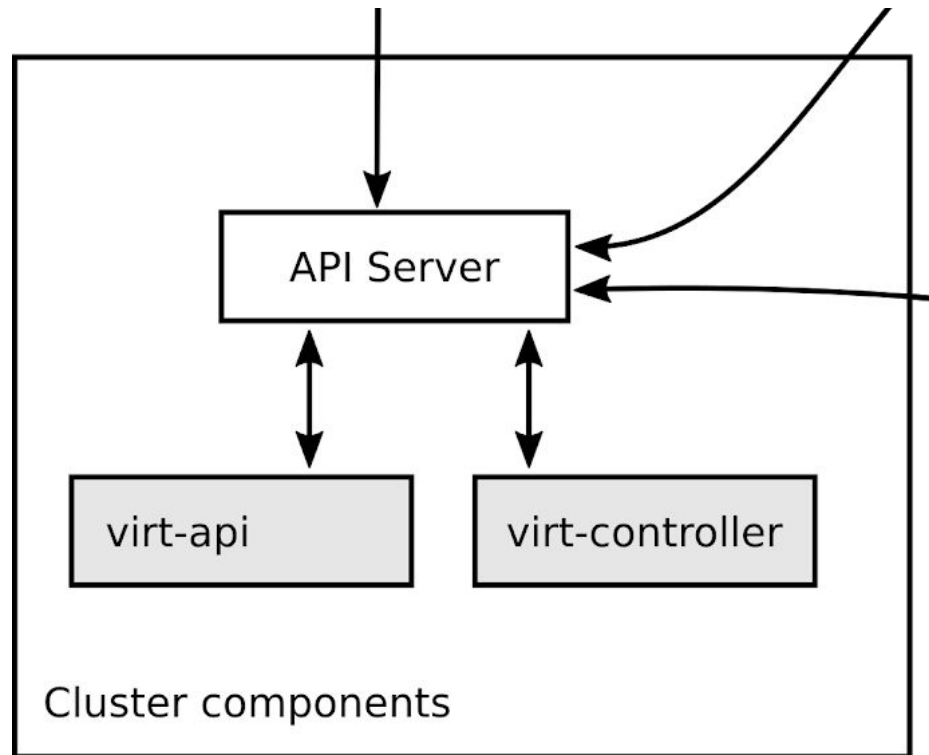
# Containerized Data Importer

- **Forma preferencial para mapear os discos das VMs**
  - Fornece APIs mais explícitas e automações complementares
- **Componente usado para importar, carregar e clonar imagens para o KubeVirt (KVM)**
  - Abstração em cima de PersistentVolumeClaim (PVC)
  - Fornece o recurso DataVolume
- **Capaz de manipular diversos formatos e convertê-los em IMG:**
  - tar, gzip, xz, raw, iso e qcow2
- **E de importar de diversas fontes:**
  - http(s), docker, local



# Adicionando virtualização ao Kubernetes

Como criar novas APIs no Kubernetes



## CRD e Aggregated API Servers

Estas são as maneiras usadas para estender a API Kubernetes para oferecer suporte a novas entidades.

Para os usuários, as novas entidades são indistinguíveis dos recursos nativos.

## Única API para todas as cargas de trabalho

Todas as cargas de trabalho (contêineres, VMs e funções) são gerenciadas por meio de uma única API.

## Operators

Simplificam a instalação, utilização e manutenção dos componentes envolvidos.

 vm.yaml

```
1  apiVersion: kubevirt.io/v1alpha3
2  kind: VirtualMachine
3  metadata:
4    name: testvm
5  spec:
6    running: false
7    template:
8      metadata:
9        labels:
10         team: Tiger
11      spec:
12        domain:
13          devices:
14            disks:
15              - disk:
16                 bus: virtio
17                 name: rootfs
18            interfaces:
19              - name: default
20          resources:
21            requests:
22              memory: 1GB
```

# Uma API Dedicada à Virtualização

Para reconhecer as diferenças e atender às expectativas

## Declarativa

Como tudo no Kubernetes, a API do KubeVirt é declarativa e segue as convenções da API Kubernetes.

## Específica de Domínio

As VMs são definitivamente diferente dos contêineres. Reutilizar a API do pod não seria explícito o suficiente para todos os detalhes necessários.

## Dividir e Conquistar

Devido à API dedicada, é simples adicionar funcionalidades específicas de virtualização, como migração ao vivo e acesso ao console gráfico.

# Demonstração!

Ou pelo menos tentativa de uma...

<https://github.com/davivcgarcia/demo-kubevirt>



# Obrigado!

**GitHub/LinkedIn/Twitter:** @davivcgarcia

 [linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)

 [youtube.com/user/RedHatVideos](https://youtube.com/user/RedHatVideos)

 [facebook.com/redhatinc](https://facebook.com/redhatinc)

 [twitter.com/RedHat](https://twitter.com/RedHat)