MIRANTIS | LABS

Tech Talks

Managing VMs and Containers Together with KubeVirt

Kevin Ng

Sr. Solutions Architect

Kevin Ng

Sr. Solution Architect

kng@mirantis.com

www.linkedin.com/in/kevinkng/

Passionate about discovering and eliminating root causes of barriers to value, with over a decade of experience helping Fortune 500 companies in multiple industries including finance, retail, healthcare, logistics, and consumer staples achieve software delivery excellence

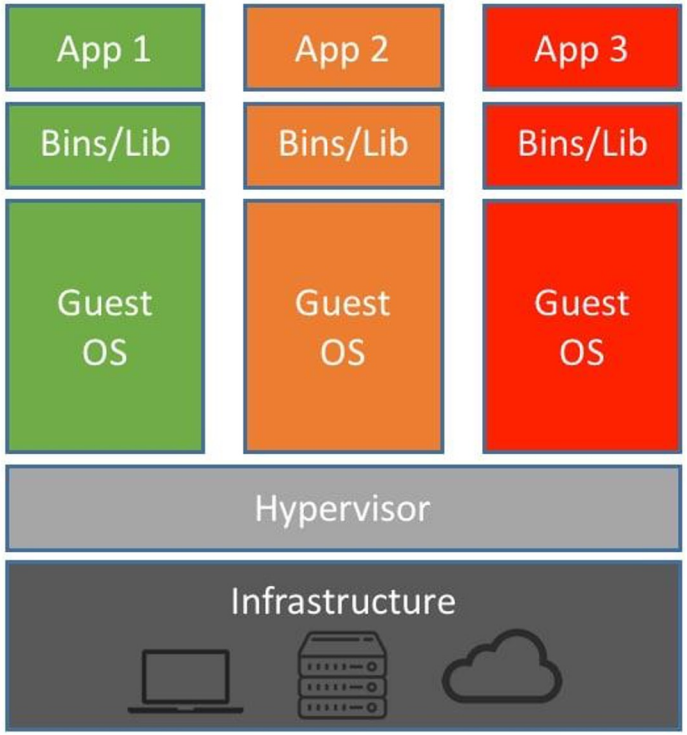
Disclaimer

The content contained herein is for informational purposes only, may not be referenced or added to any contract, and should not be relied upon to make purchasing decisions. It is not a commitment, promise, or legal obligation to provide any features, functionality, capabilities, code, etc. or to provide anything within any schedule, date, time, etc. All Mirantis product and service decisions remain at Mirantis sole and exclusive discretion.

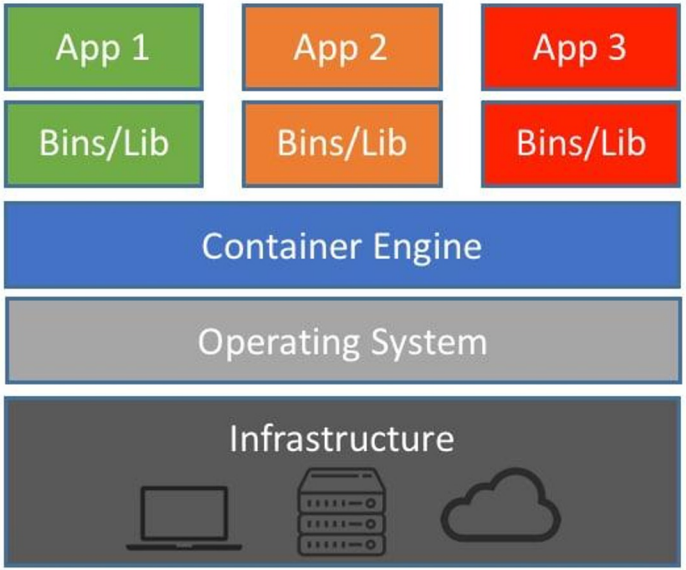
Agenda

- Use Cases
- Benefits
- Architecture Overview
- Installation
- Kubevirt in action

A tale of two workloads



Machine Virtualization



Containers

Kubernetes is everywhere

Use Cases

- Compliment VMWare footprint
- Workloads that are not container ready
- Edge cloud deployments supporting mixed workloads
- Mixed workloads in the datacenter
- Security Isolation of workloads into VMs

Benefits

- Reduce control plane footprint - Edge and Datacenter
- Reduced complexity - One control plane for both containers and VMs
- Single set of orchestration primitives - K8s for both (Containers and VMs)
- Consolidated orchestration

Projects used – k0s

- Fully **open source**, **supported, validated and enhanced distribution** of Kubernetes
- Security compliance
- Full stack **automated life-cycle management**

https://k0sproject.io/

Projects used - kubevirt

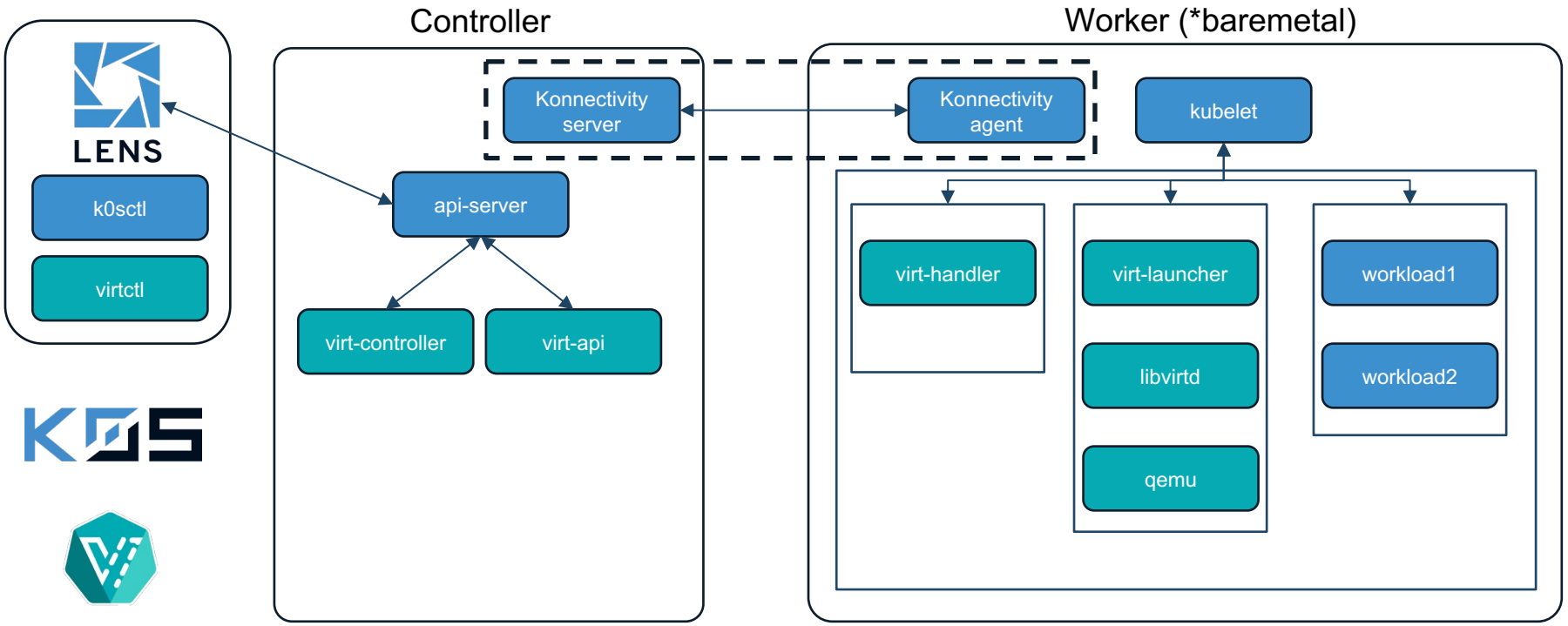
- Brings **virtual machine capabilities** into Kubernetes
- CNCF Incubating project

https://kubevirt.io/

Installation

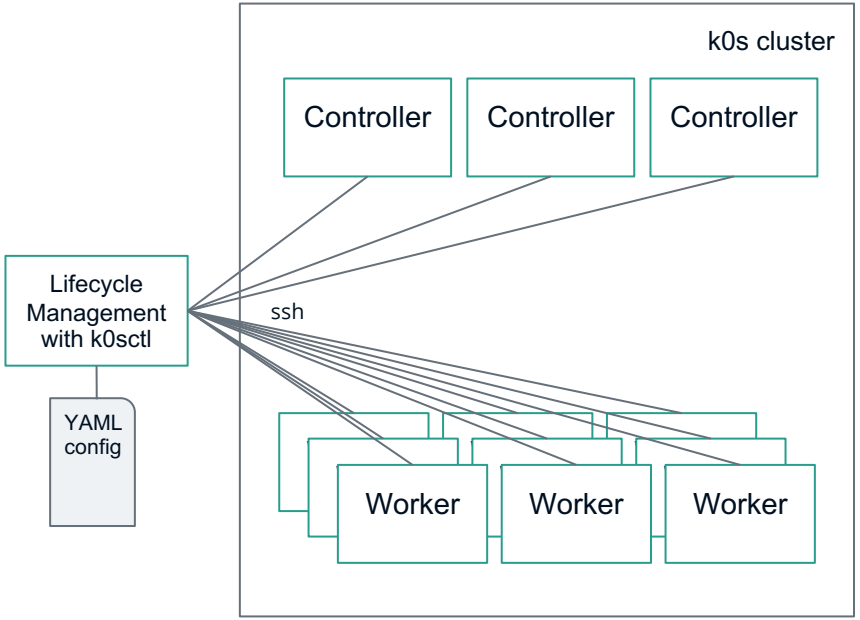
- k0s
- kubevirt operator
- virtctl
- libvirt-clients

Setup


LENS
k0sctl
virtctl
Controller
Konnectivity server
api-server
virt-controller
virt-api
Worker (*baremetal)
Konnectivity agent
kubelet
virt-handler
virt-launcher
libvirtd
qemu
workload1
workload2

Installation – k0s

- Cluster setup with k0sctl
- Need servers to be available and have ssh access
- Update kubelet location on worker node



Installation - kubevirt

- Deploy the KubeVirt operator

```
kubectl apply -f
https://github.com/kubevirt/kubevirt/releases/download/${RELEASE}/kubevirt-
operator.yaml
```

- Create KubeVirt CR

```
kubectl apply -f
https://github.com/kubevirt/kubevirt/releases/download/${RELEASE}/kubevirt-cr.yaml
```

Install virtctl

wget https://github.com/kubevirt/kubevirt/releases/download/${VERSION}/virtctl-${VERSION}-linux-amd64

 # available architectures
darwin-amd64
darwin-arm64
linux-amd64
linux-arm64
windows-amd64

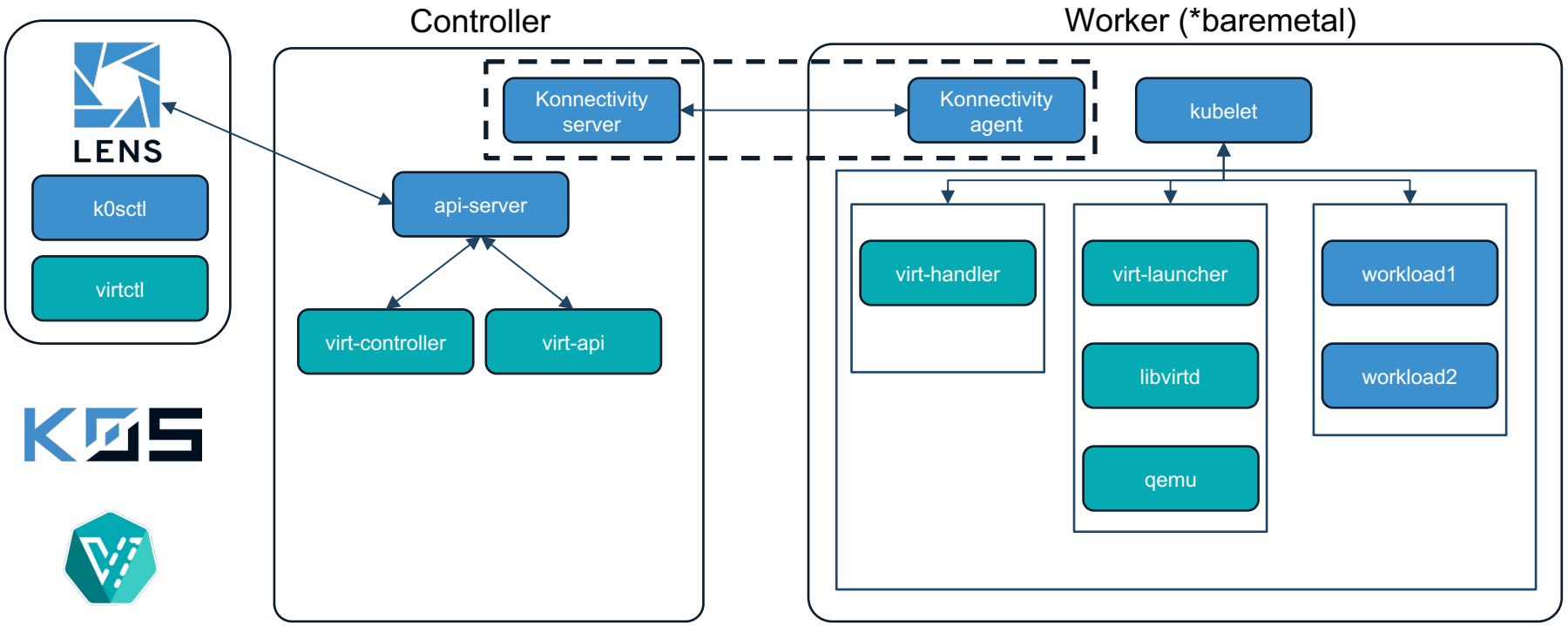
Running VMs

- kind: VirtualMachine

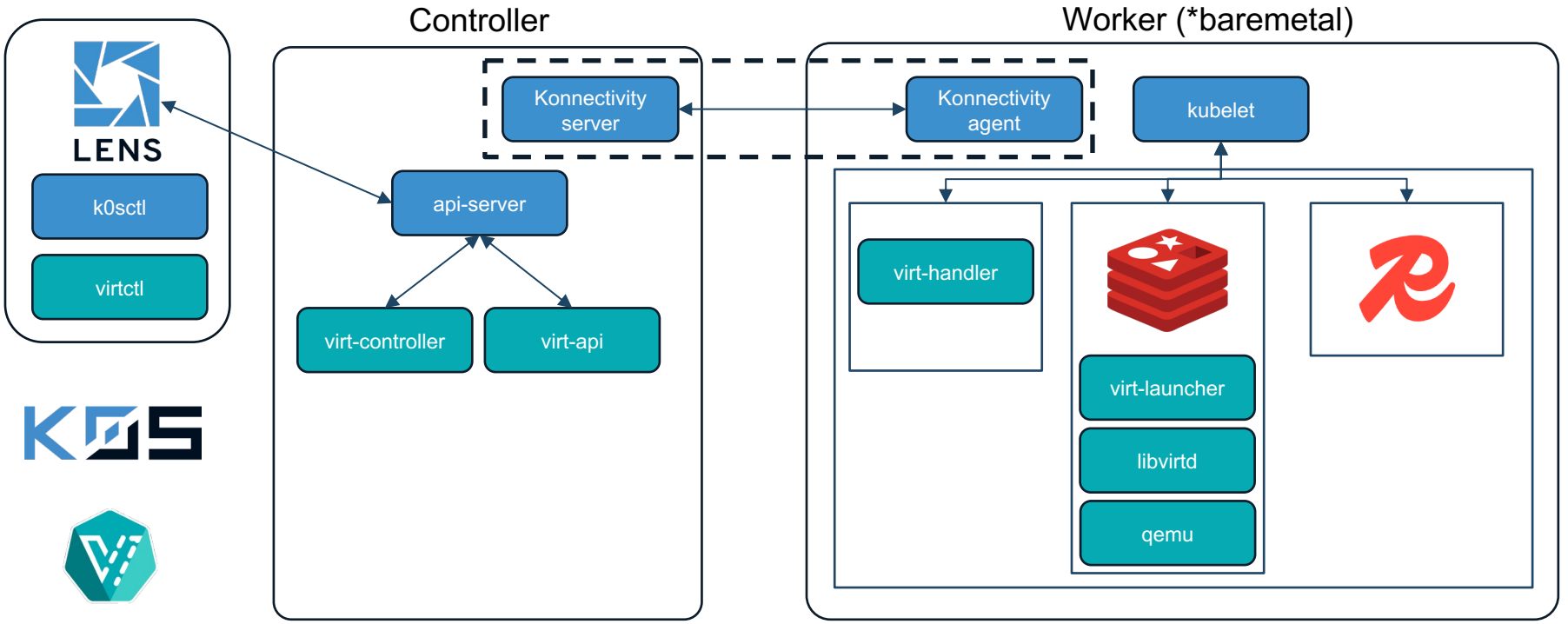
```
kubectl apply –f vm.yaml  #create vm instance
virtctl start ubuntu      #start vm
virtctl console ubuntu    #connect to vm
virtctl stop ubuntu       #stop vm
```

Demo

Setup


LENS
k0sctl
virtctl
Controller
Konnectivity server
api-server
virt-controller
virt-api
Worker (*baremetal)
Konnectivity agent
kubelet
virt-handler
virt-launcher
libvirtd
qemu
workload1
workload2

Setup


Controller
Worker (*baremetal)
LENS
k0sctl
virtctl
Konnectivity server
Konnectivity agent
kubelet
api-server
virt-handler
virt-controller
virt-api
virt-launcher
libvirtd
qemu

Try it out yourself

Projects used

k0sproject.io

kubevirt.io

k0smotron.io

Lab assets

MIRANTIS

mirantis.com/labs

kevng9/k0s_kubevirt_techtalk

kng@mirantis.com

in/kevinkng/