前言: 问题描述

当前方案: 在多租户场景下，交付以裸机 (目前主要指 x86) 为主要算力，KubeVirt VM为弹性算力的 k8s clusters。

问题描述: 由Management k8s cluster 管理和维护所有租户的 k8s clusters 的生命周期。

- 优点：租户与租户之间是真实物理 k8s cluster 级别的隔离。
- 缺点：1）每个租户各自维护自己一套或者多套的 k8s clusters，资源开销大；2）当租户数量达到一定数量后，管理其 k8s clusters 会非常复杂。

解决方案 v1: 引入vCluster

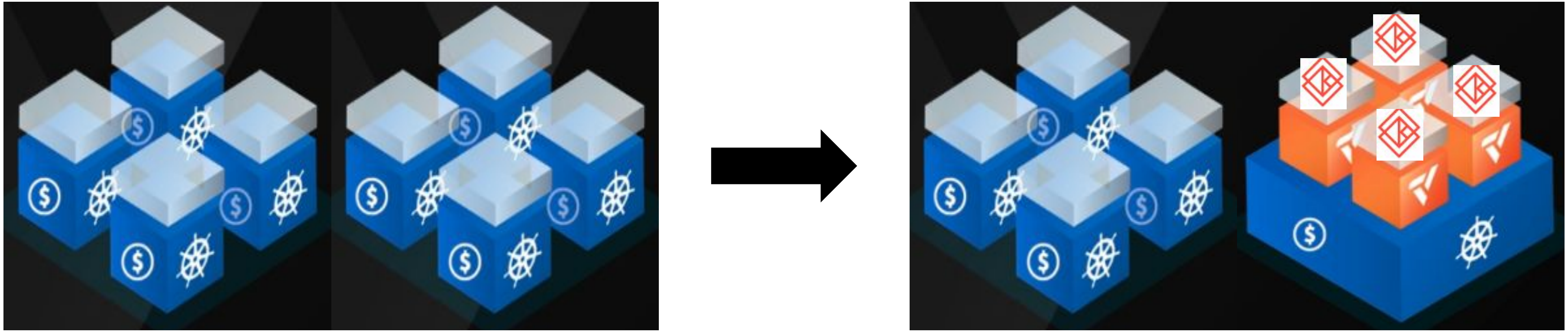
前言: 解决方案 v2

kubernetes

vCluster

katacontainers

愿景: 在vCluster的方案上继续提升隔离。

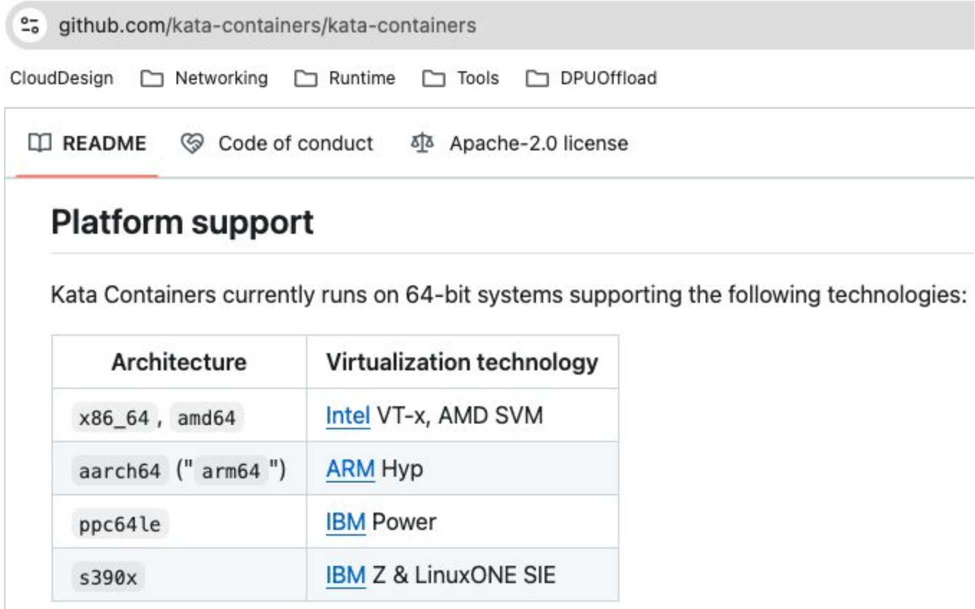


Agenda

- Kata Containers: Introduction & Features
- Kata Containers vs. Traditional Containers
- Demo: Install/Enable/Test Kata Containers in k8s
- Kata Containers’ Key Components: Kata Containers Runtime & Kata Agent
- Kata Containers: Networking
- Kata Containers: Workflow
- Integrate Kata Containers with vCluster
- Reference

Kata Containers: Introduction

Kata Containers is an open source container runtime, building lightweight virtual machines that seamlessly plug into the containers ecosystem [1].



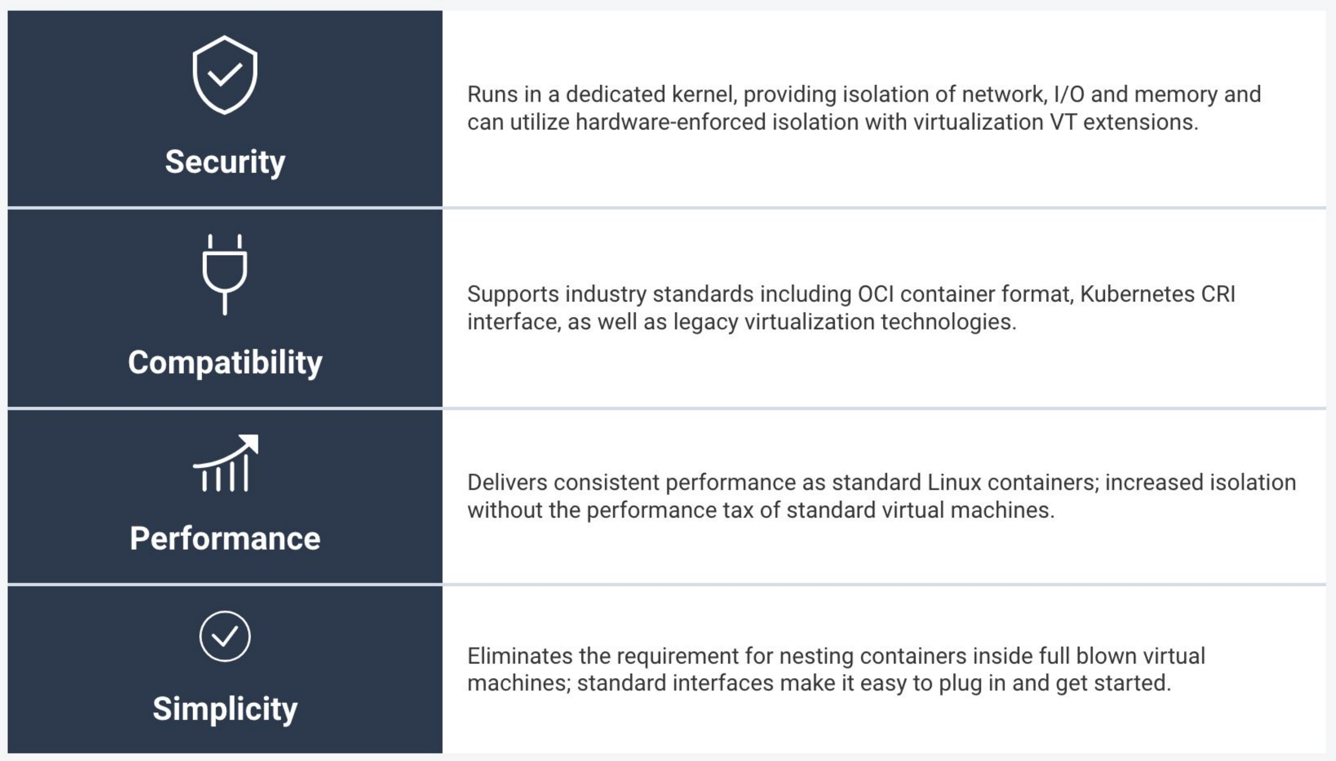
Platform support

Kata Containers currently runs on 64-bit systems supporting the following technologies:

Architecture	Virtualization technology
`x86_64`, `amd64`	Intel VT-x, AMD SVM
`aarch64` ("`arm64`")	ARM Hyp
`ppc64le`	IBM Power
`s390x`	IBM Z & LinuxONE SIE

Note: Kata Containers is not supported on RISC-V yet.

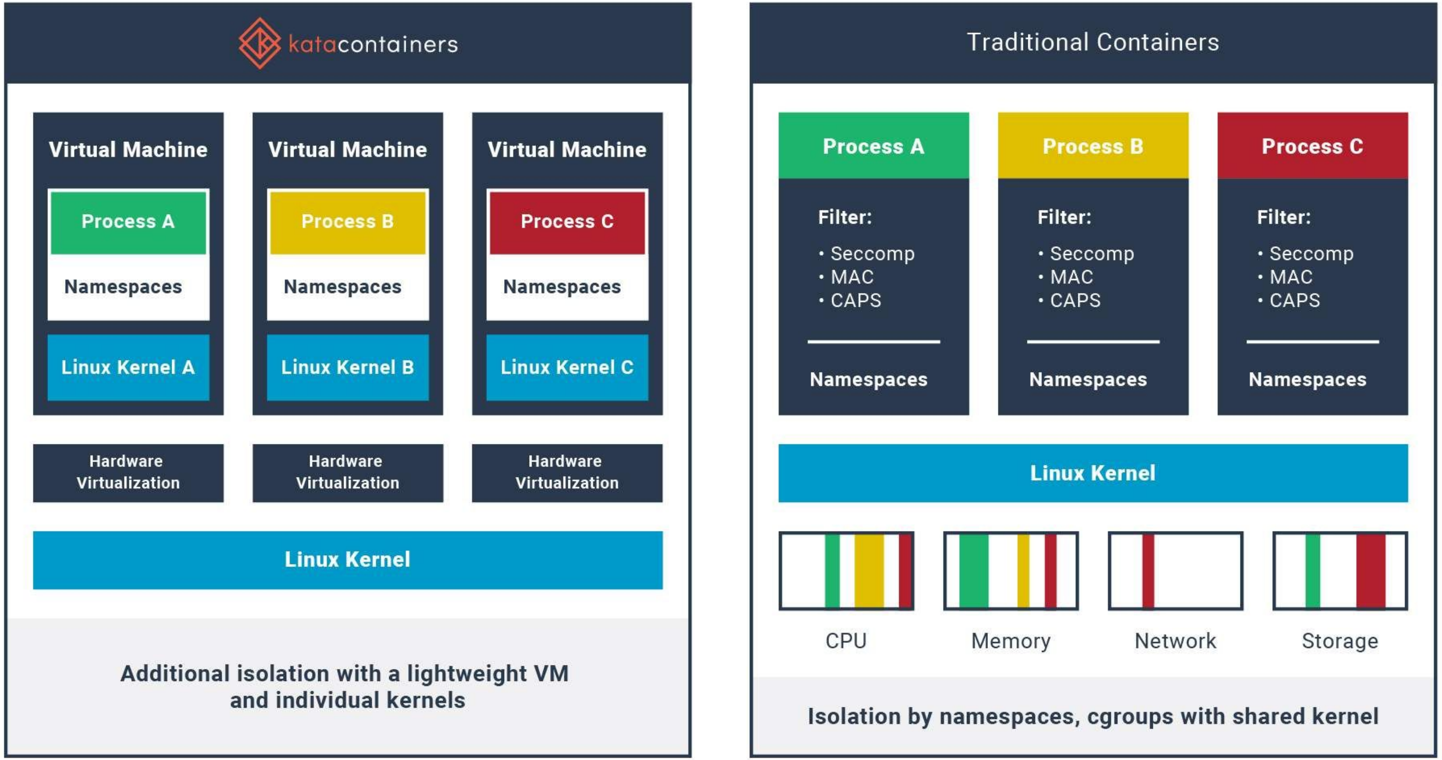
Kata Containers: Features



Security	Runs in a dedicated kernel, providing isolation of network, I/O and memory and can utilize hardware-enforced isolation with virtualization VT extensions.
Compatibility	Supports industry standards including OCI container format, Kubernetes CRI interface, as well as legacy virtualization technologies.
Performance	Delivers consistent performance as standard Linux containers; increased isolation without the performance tax of standard virtual machines.
Simplicity	Eliminates the requirement for nesting containers inside full blown virtual machines; standard interfaces make it easy to plug in and get started.

Figure: Kata Containers' Features [2]

Kata Containers vs. Traditional Containers



katacontainers

Virtual Machine | Virtual Machine | Virtual Machine

Process A | Process B | Process C

Namespaces | Namespaces | Namespaces

Linux Kernel A | Linux Kernel B | Linux Kernel C

Hardware Virtualization | Hardware Virtualization | Hardware Virtualization

Linux Kernel

Additional isolation with a lightweight VM and individual kernels

Traditional Containers

Process A | Process B | Process C

Filter:
- Seccomp
- MAC
- CAPS

Namespaces

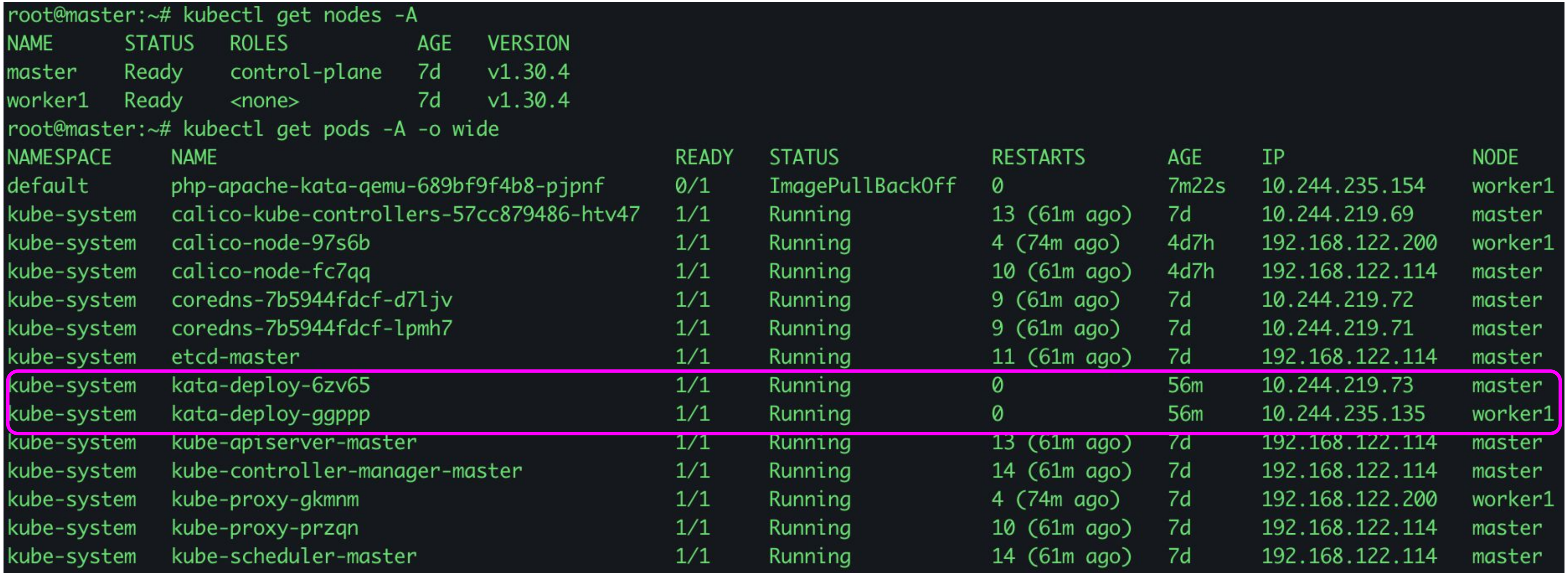
Linux Kernel

CPU | Memory | Network | Storage

Isolation by namespaces, cgroups with shared kernel

Figure: Kata Containers vs. Traditional Containers [2]

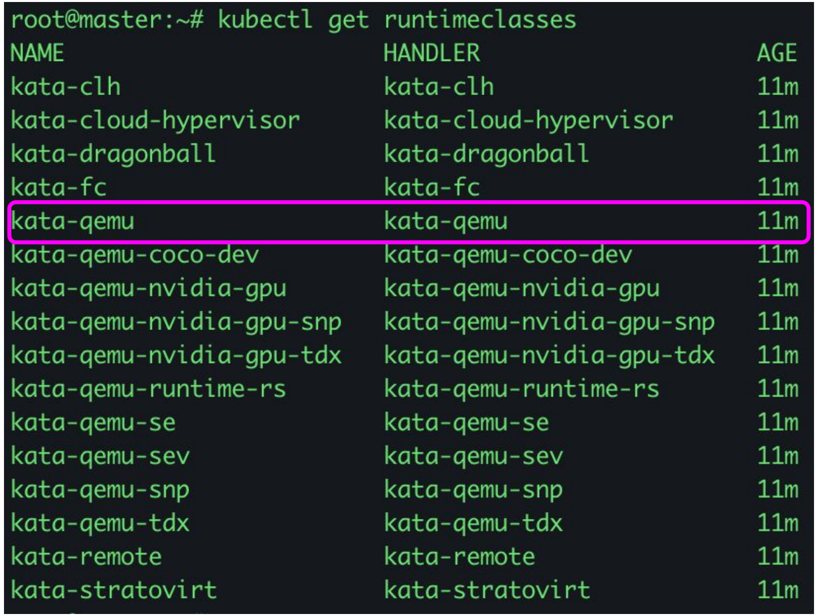
Demo: Install Kata Containers



```
root@master:~# kubectl get nodes -A
NAME      STATUS   ROLES           AGE   VERSION
master    Ready    control-plane   7d    v1.30.4
worker1   Ready    <none>          7d    v1.30.4
root@master:~# kubectl get pods -A -o wide
NAMESPACE     NAME                                       READY   STATUS             RESTARTS       AGE     IP                NODE
default       php-apache-kata-qemu-689bf9f4b8-pjpnf      0/1     ImagePullBackOff   0              7m22s   10.244.235.154    worker1
kube-system   calico-kube-controllers-57cc879486-htv47   1/1     Running            13 (61m ago)   7d      10.244.219.69     master
kube-system   calico-node-97s6b                          1/1     Running            4 (74m ago)    4d7h    192.168.122.200   worker1
kube-system   calico-node-fc7qq                          1/1     Running            10 (61m ago)   4d7h    192.168.122.114   master
kube-system   coredns-7b5944fdcf-d7ljv                   1/1     Running            9 (61m ago)    7d      10.244.219.72     master
kube-system   coredns-7b5944fdcf-lpmh7                   1/1     Running            9 (61m ago)    7d      10.244.219.71     master
kube-system   etcd-master                                1/1     Running            11 (61m ago)   7d      192.168.122.114   master
kube-system   kata-deploy-6zv65                          1/1     Running            0              56m     10.244.219.73     master
kube-system   kata-deploy-ggppp                          1/1     Running            0              56m     10.244.235.135    worker1
kube-system   kube-apiserver-master                      1/1     Running            13 (61m ago)   7d      192.168.122.114   master
kube-system   kube-controller-manager-master             1/1     Running            14 (61m ago)   7d      192.168.122.114   master
kube-system   kube-proxy-gkmnm                           1/1     Running            4 (74m ago)    7d      192.168.122.200   worker1
kube-system   kube-proxy-przqn                           1/1     Running            10 (61m ago)   7d      192.168.122.114   master
kube-system   kube-scheduler-master                      1/1     Running            14 (61m ago)   7d      192.168.122.114   master
```

```
# kubectl apply -f  https://raw.githubusercontent.com/kata-containers/kata-containers/main/tools/packaging/kata-deploy/kata-deploy/base/kata-deploy.yaml
# kubectl apply -f  https://github.com/kata-containers/kata-containers/blob/main/tools/packaging/kata-deploy/kata-rbac/base/kata-rbac.yaml
```

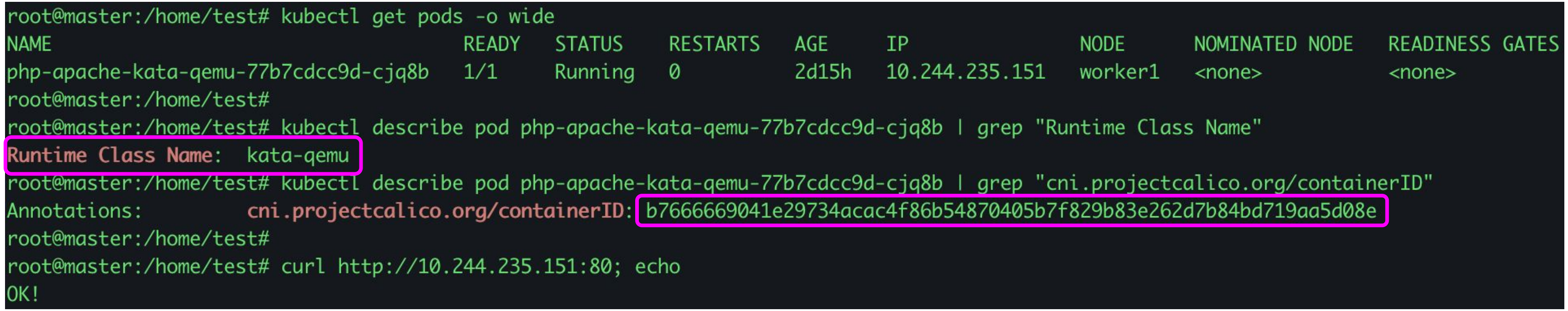
Demo: Enable Kata Container RuntimeClass



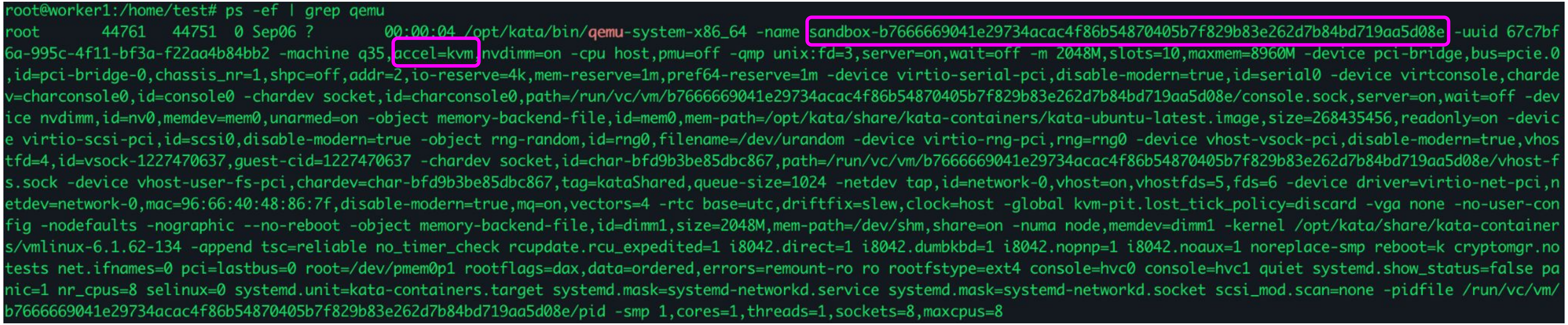
```
root@master:~# kubectl get runtimeclasses
NAME                        HANDLER                     AGE
kata-clh                    kata-clh                    11m
kata-cloud-hypervisor       kata-cloud-hypervisor       11m
kata-dragonball             kata-dragonball             11m
kata-fc                     kata-fc                     11m
kata-qemu                   kata-qemu                   11m
kata-qemu-coco-dev          kata-qemu-coco-dev          11m
kata-qemu-nvidia-gpu        kata-qemu-nvidia-gpu        11m
kata-qemu-nvidia-gpu-snp    kata-qemu-nvidia-gpu-snp    11m
kata-qemu-nvidia-gpu-tdx    kata-qemu-nvidia-gpu-tdx    11m
kata-qemu-runtime-rs        kata-qemu-runtime-rs        11m
kata-qemu-se                kata-qemu-se                11m
kata-qemu-sev               kata-qemu-sev               11m
kata-qemu-snp               kata-qemu-snp               11m
kata-qemu-tdx               kata-qemu-tdx               11m
kata-remote                 kata-remote                 11m
kata-stratovirt             kata-stratovirt             11m
```

```
# kubectl apply -f  https://raw.githubusercontent.com/kata-containers/kata-containers/main/tools/packaging/kata-deploy/runtimeclasses/kata-runtimeClasses.yaml
```

Demo: Start a Pod with kata-qemu RuntimeClass



```
root@master:/home/test# kubectl get pods -o wide
NAME                                 READY   STATUS    RESTARTS   AGE    IP               NODE      NOMINATED NODE   READINESS GATES
php-apache-kata-qemu-77b7cdcc9d-cjq8b   1/1     Running   0          2d15h  10.244.235.151   worker1   <none>           <none>
root@master:/home/test#
root@master:/home/test# kubectl describe pod php-apache-kata-qemu-77b7cdcc9d-cjq8b | grep "Runtime Class Name"
Runtime Class Name:   kata-qemu
root@master:/home/test# kubectl describe pod php-apache-kata-qemu-77b7cdcc9d-cjq8b | grep "cni.projectcalico.org/containerID"
Annotations:          cni.projectcalico.org/containerID: b7666669041e29734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e
root@master:/home/test#
root@master:/home/test# curl http://10.244.235.151:80; echo
OK!
```



```
root@worker1:/home/test# ps -ef | grep qemu
root       44761   44751  0 Sep06 ?        00:00:04 /opt/kata/bin/qemu-system-x86_64 -name sandbox-b7666669041e29734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e -uuid 67c7bf
6a-995c-4f11-bf3a-f22aa4b84bb2 -machine q35,accel=kvm,nvdimm=on -cpu host,pmu=off -qmp unix:fd=3,server=on,wait=off -m 2048M,slots=10,maxmem=8960M -device pci-bridge,bus=pcie.0
,id=pci-bridge-0,chassis_nr=1,shpc=off,addr=2,io-reserve=4k,mem-reserve=1m,pref64-reserve=1m -device virtio-serial-pci,disable-modern=true,id=serial0 -device virtconsole,charde
v=charconsole0,id=console0 -chardev socket,id=charconsole0,path=/run/vc/vm/b7666669041e29734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e/console.sock,server=on,wait=off -dev
ice nvdimm,id=nv0,memdev=mem0,unarmed=on -object memory-backend-file,id=mem0,mem-path=/opt/kata/share/kata-containers/kata-ubuntu-latest.image,size=268435456,readonly=on -devic
e virtio-scsi-pci,id=scsi0,disable-modern=true -object rng-random,id=rng0,filename=/dev/urandom -device virtio-rng-pci,rng=rng0 -device vhost-vsock-pci,disable-modern=true,vhos
tfd=4,id=vsock-1227470637,guest-cid=1227470637 -chardev socket,id=char-bfd9b3be85dbc867,path=/run/vc/vm/b7666669041e29734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e/vhost-f
s.sock -device vhost-user-fs-pci,chardev=char-bfd9b3be85dbc867,tag=kataShared,queue-size=1024 -netdev tap,id=network-0,vhost=on,vhostfds=5,fds=6 -device driver=virtio-net-pci,n
etdev=network-0,mac=96:66:40:48:86:7f,disable-modern=true,mq=on,vectors=4 -rtc base=utc,driftfix=slew,clock=host -global kvm-pit.lost_tick_policy=discard -vga none -no-user-con
fig -nodefaults -nographic --no-reboot -object memory-backend-file,id=dimm1,size=2048M,mem-path=/dev/shm,share=on -numa node,memdev=dimm1 -kernel /opt/kata/share/kata-container
s/vmlinux-6.1.62-134 -append tsc=reliable no_timer_check rcupdate.rcu_expedited=1 i8042.direct=1 i8042.dumbkbd=1 i8042.nopnp=1 i8042.noaux=1 noreplace-smp reboot=k cryptomgr.no
tests net.ifnames=0 pci=lastbus=0 root=/dev/pmem0p1 rootflags=dax,data=ordered,errors=remount-ro ro rootfstype=ext4 console=hvc0 console=hvc1 quiet systemd.show_status=false pa
nic=1 nr_cpus=8 selinux=0 systemd.unit=kata-containers.target systemd.mask=systemd-networkd.service systemd.mask=systemd-networkd.socket scsi_mod.scan=none -pidfile /run/vc/vm/
b7666669041e29734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e/pid -smp 1,cores=1,threads=1,sockets=8,maxcpus=8
```

# kubectl apply -f https://raw.githubusercontent.com/kata-containers/kata-containers/main/tools/packaging/kata-deploy/examples/test-deploy-kata-qemu.yaml

Kata Containers Runtime

The Kata Containers runtime is compatible with the OCI runtime specification and therefore works seamlessly with the Kubernetes Container Runtime Interface (CRI) through the CRI-O and containerd implementations.

Kata Containers provides a "shimv2" compatible runtime.

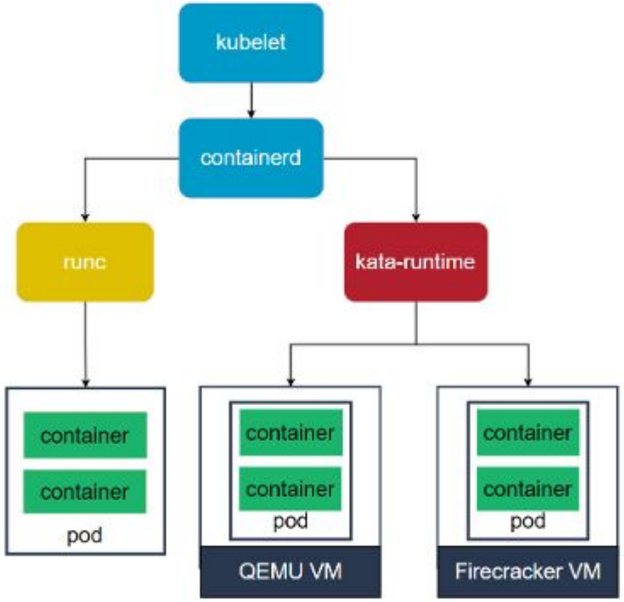
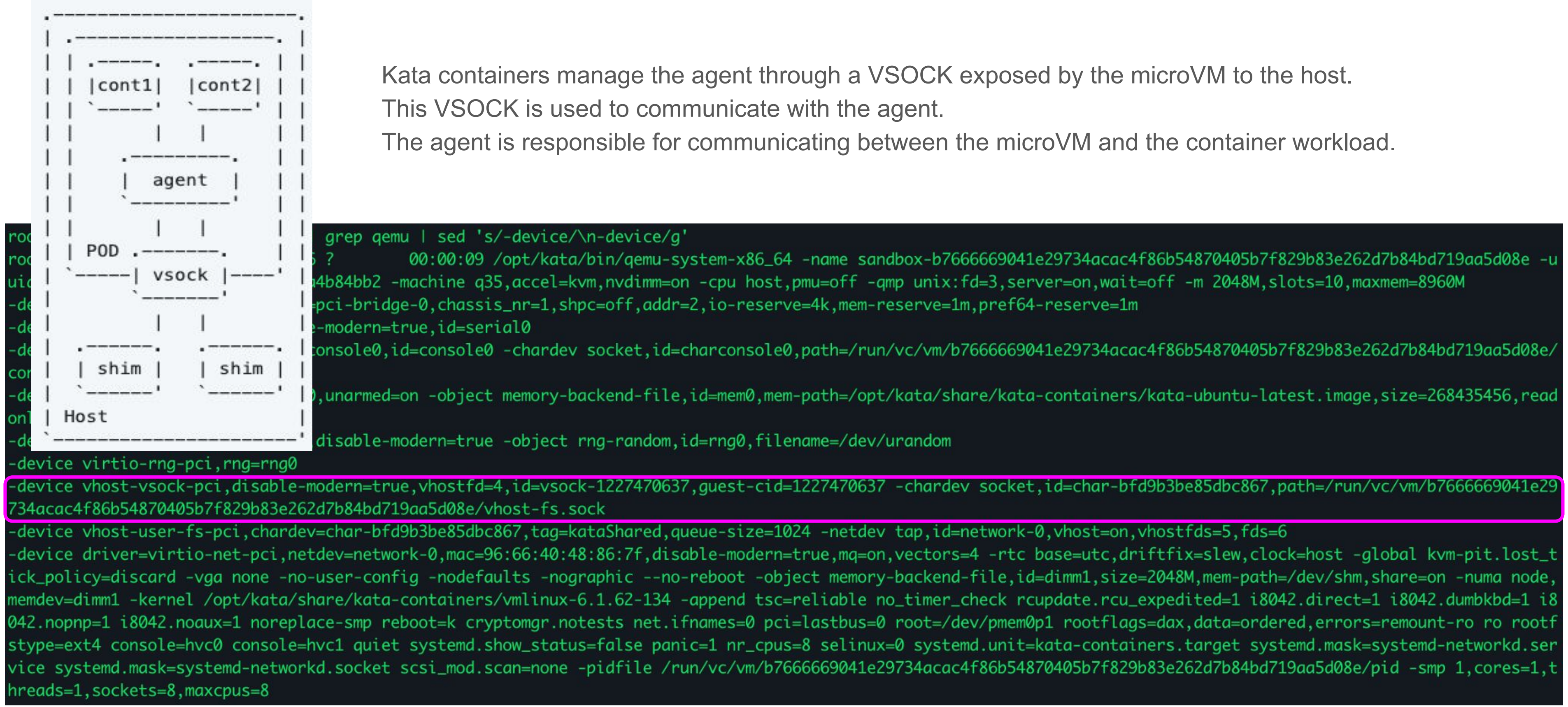


Figure: Integrate Kata Containers with k8s [3]

Details: https://raw.githubusercontent.com/kata-containers/kata-containers/main/docs/design/arch-images/katacontainers-e2e.svg

Kata Agent

Kata containers manage the agent through a VSOCK exposed by the microVM to the host.
This VSOCK is used to communicate with the agent.
The agent is responsible for communicating between the microVM and the container workload.



```
roo                           grep qemu | sed 's/-device/\n-device/g'
roo                          5 ?        00:00:09 /opt/kata/bin/qemu-system-x86_64 -name sandbox-b7666669041e29734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e -u
uid                          a4b84bb2 -machine q35,accel=kvm,nvdimm=on -cpu host,pmu=off -qmp unix:fd=3,server=on,wait=off -m 2048M,slots=10,maxmem=8960M
-de                          =pci-bridge-0,chassis_nr=1,shpc=off,addr=2,io-reserve=4k,mem-reserve=1m,pref64-reserve=1m
-de                          e-modern=true,id=serial0
-de                          console0,id=console0 -chardev socket,id=charconsole0,path=/run/vc/vm/b7666669041e29734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e/
con
-de                          0,unarmed=on -object memory-backend-file,id=mem0,mem-path=/opt/kata/share/kata-containers/kata-ubuntu-latest.image,size=268435456,read
onl
-de                           disable-modern=true -object rng-random,id=rng0,filename=/dev/urandom
-device virtio-rng-pci,rng=rng0
-device vhost-vsock-pci,disable-modern=true,vhostfd=4,id=vsock-1227470637,guest-cid=1227470637 -chardev socket,id=char-bfd9b3be85dbc867,path=/run/vc/vm/b7666669041e29
734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e/vhost-fs.sock
-device vhost-user-fs-pci,chardev=char-bfd9b3be85dbc867,tag=kataShared,queue-size=1024 -netdev tap,id=network-0,vhost=on,vhostfds=5,fds=6
-device driver=virtio-net-pci,netdev=network-0,mac=96:66:40:48:86:7f,disable-modern=true,mq=on,vectors=4 -rtc base=utc,driftfix=slew,clock=host -global kvm-pit.lost_t
ick_policy=discard -vga none -no-user-config -nodefaults -nographic --no-reboot -object memory-backend-file,id=dimm1,size=2048M,mem-path=/dev/shm,share=on -numa node,
memdev=dimm1 -kernel /opt/kata/share/kata-containers/vmlinux-6.1.62-134 -append tsc=reliable no_timer_check rcupdate.rcu_expedited=1 i8042.direct=1 i8042.dumbkbd=1 i8
042.nopnp=1 i8042.noaux=1 noreplace-smp reboot=k cryptomgr.notests net.ifnames=0 pci=lastbus=0 root=/dev/pmem0p1 rootflags=dax,data=ordered,errors=remount-ro ro rootf
stype=ext4 console=hvc0 console=hvc1 quiet systemd.show_status=false panic=1 nr_cpus=8 selinux=0 systemd.unit=kata-containers.target systemd.mask=systemd-networkd.ser
vice systemd.mask=systemd-networkd.socket scsi_mod.scan=none -pidfile /run/vc/vm/b7666669041e29734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e/pid -smp 1,cores=1,t
hreads=1,sockets=8,maxcpus=8
```

Kata Containers: Networking

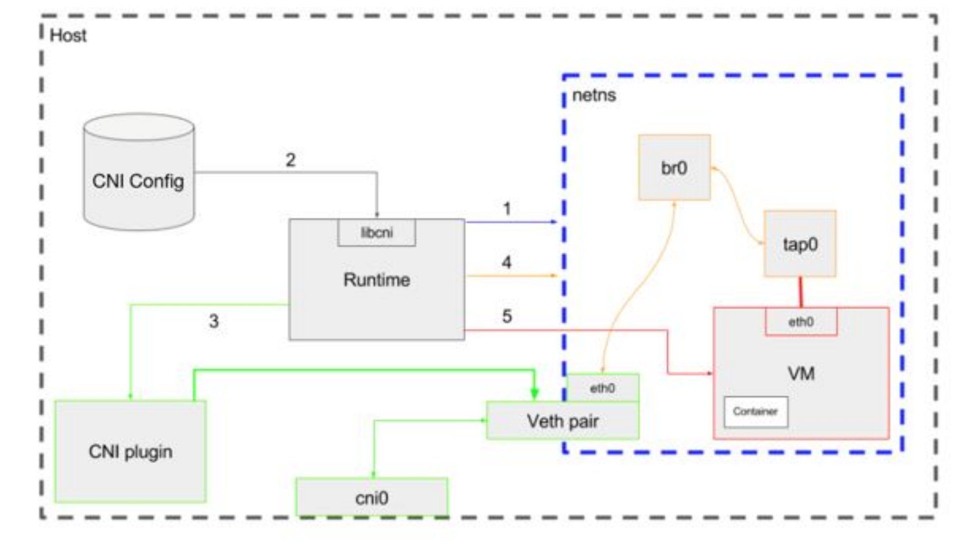
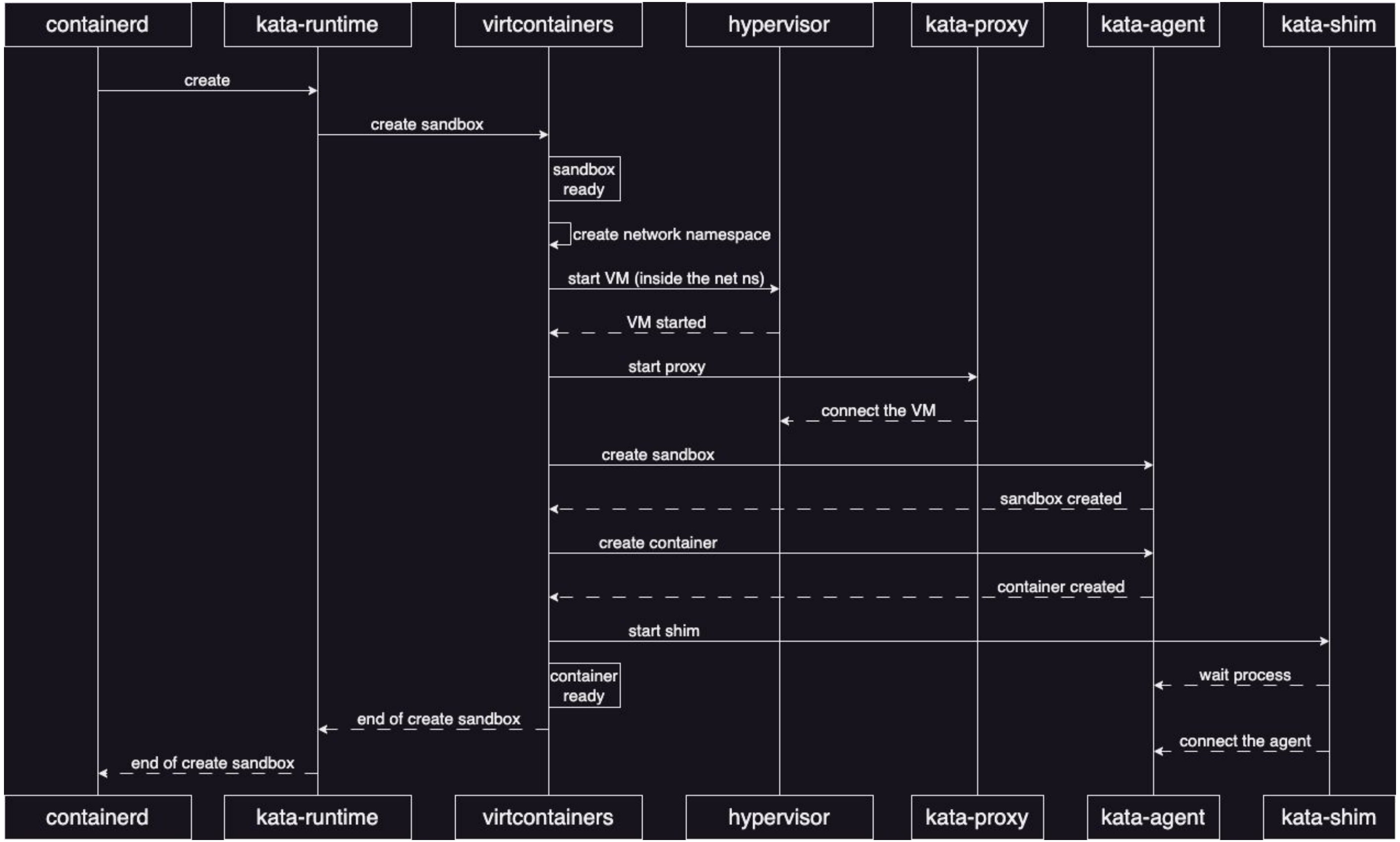


Figure: Container Network Initiative (CNI) implementation for virtual-machine-based containers.

1. The runtime creates the blue-bordered network namespace, which contains all devices associated with the VM.
2. The runtime reads the required configuration from the CNI configuration files for the containers.
3. The runtime will communicate with the configured plug-in to start network service for the container. A veth pair is set up between cni0 and the container's network namespace.
4. A bridge inside the namespace is created, a tap device is plumbed to the bridge for the VM to use with standard virtio drivers, and the previous veth pair endpoint is plumbed to the bridge as well.
5. The VM and container workload are started inside the network namespace.

```
root(
root                                                             sandbox-b7666669041e29734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e -u
uid (                                                            f -qmp unix:fd=3,server=on,wait=off -m 2048M,slots=10,maxmem=8960M
-dev                                                             em-reserve=1m,pref64-reserve=1m
-dev
-dev                                                             /run/vc/vm/b7666669041e29734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e/
cons
-dev                                                             /opt/kata/share/kata-containers/kata-ubuntu-latest.image,size=268435456,read
only
-dev                                                             /dev/urandom
-dev
-dev                                                             0637 -chardev socket,id=char-bfd9b3be85dbc867,path=/run/vc/vm/b7666669041e29
734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e/vhost-fs.sock
-device vhost-user-fs-pci,chardev=char-bfd9b3be85dbc867,tag=kataShared,queue-size=1024 -netdev tap,id=network-0,vhost=on,vhostfds=5,fds=6
-device driver=virtio-net-pci,netdev=network-0,mac=96:66:40:48:86:7f,disable-modern=true,mq=on,vectors=4 -rtc base=utc,driftfix=slew,clock=host -global kvm-pit.lost_t
ick_policy=discard -vga none -no-user-config -nodefaults -nographic --no-reboot -object memory-backend-file,id=dimm1,size=2048M,mem-path=/dev/shm,share=on -numa node,
memdev=dimm1 -kernel /opt/kata/share/kata-containers/vmlinux-6.1.62-134 -append tsc=reliable no_timer_check rcupdate.rcu_expedited=1 i8042.direct=1 i8042.dumbkbd=1 i8
042.nopnp=1 i8042.noaux=1 noreplace-smp reboot=k cryptomgr.notests net.ifnames=0 pci=lastbus=0 root=/dev/pmem0p1 rootflags=dax,data=ordered,errors=remount-ro ro rootf
stype=ext4 console=hvc0 console=hvc1 quiet systemd.show_status=false panic=1 nr_cpus=8 selinux=0 systemd.unit=kata-containers.target systemd.mask=systemd-networkd.ser
vice systemd.mask=systemd-networkd.socket scsi_mod.scan=none -pidfile /run/vc/vm/b7666669041e29734acac4f86b54870405b7f829b83e262d7b84bd719aa5d08e/pid -smp 1,cores=1,t
hreads=1,sockets=8,maxcpus=8
```

Kata Containers: Workflow



containerd

kata-runtime

virtcontainers

hypervisor

kata-proxy

kata-agent

kata-shim

create

create sandbox

sandbox ready

create network namespace

start VM (inside the net ns)

VM started

start proxy

connect the VM

create sandbox

sandbox created

create container

container created

start shim

container ready

wait process

end of create sandbox

connect the agent

end of create sandbox

containerd

kata-runtime

virtcontainers

hypervisor

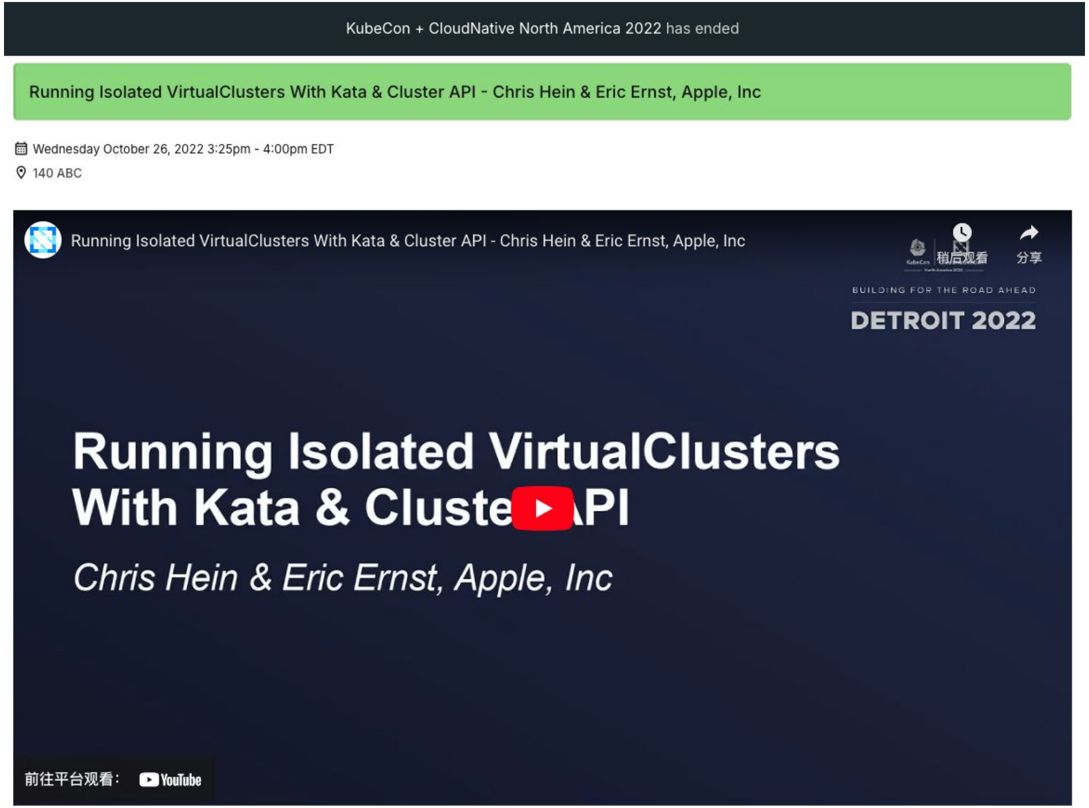
kata-proxy

kata-agent

kata-shim

Demo: Kata Containers + vCluster

https://github.com/loft-sh/vcluster/issues/2125



KubeCon + CloudNative North America 2022 has ended

Running Isolated VirtualClusters With Kata & Cluster API - Chris Hein & Eric Ernst, Apple, Inc

Wednesday October 26, 2022 3:25pm - 4:00pm EDT

140 ABC

Running Isolated VirtualClusters With Kata & Cluster API - Chris Hein & Eric Ernst, Apple, Inc

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Running Isolated VirtualClusters
With Kata & Cluste ▶ PI

Chris Hein & Eric Ernst, Apple, Inc

References

[1] Li, Guoqing & Takahashi, Keichi & Ichikawa, Kohei & Iida, Hajimu & Nakasan, Chawanat & Leelaprute, Pattara & Thiengburanathum, Pree & Phannachitta, Passakorn. (2023). The Convergence of Container and Traditional Virtualization: Strengths and Limitations. SN Computer Science. 4. 10.1007/s42979-023-01827-9.
[2] https://katacontainers.io/
[3] https://github.com/kata-containers/documentation/blob/master/design/architecture.md