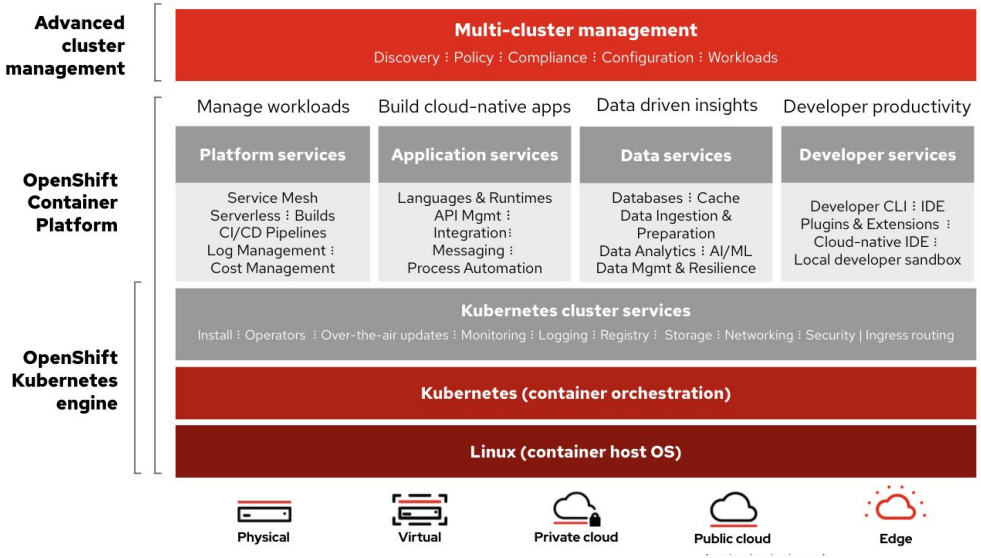
Kata Containers on OpenShift

张家驹
Red Hat

Red Hat

What is OpenShift?

A smarter Kubernetes platform



Automated, full-stack installation from the container host to application services

Seamless Kubernetes deployment to any cloud or on-premises environment

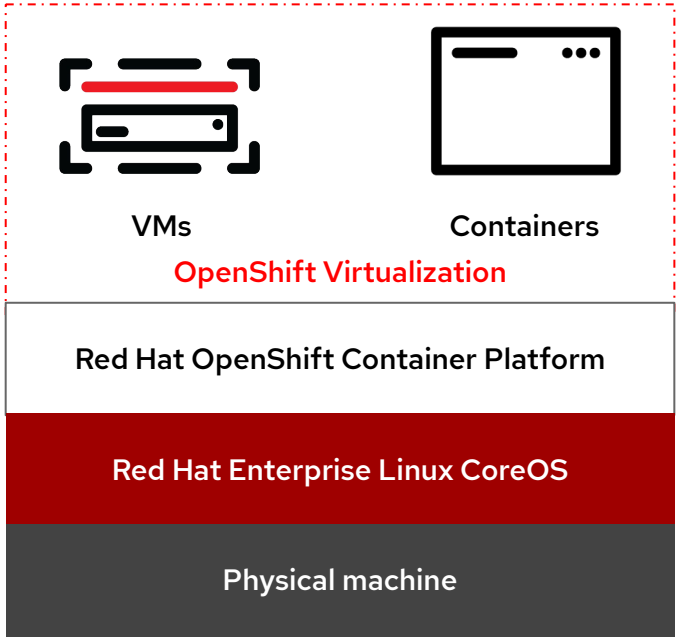
Autoscaling of cloud resources

One-click updates for platform, services, and applications

Red Hat

Virtual Machines Run and Managed in OpenShift

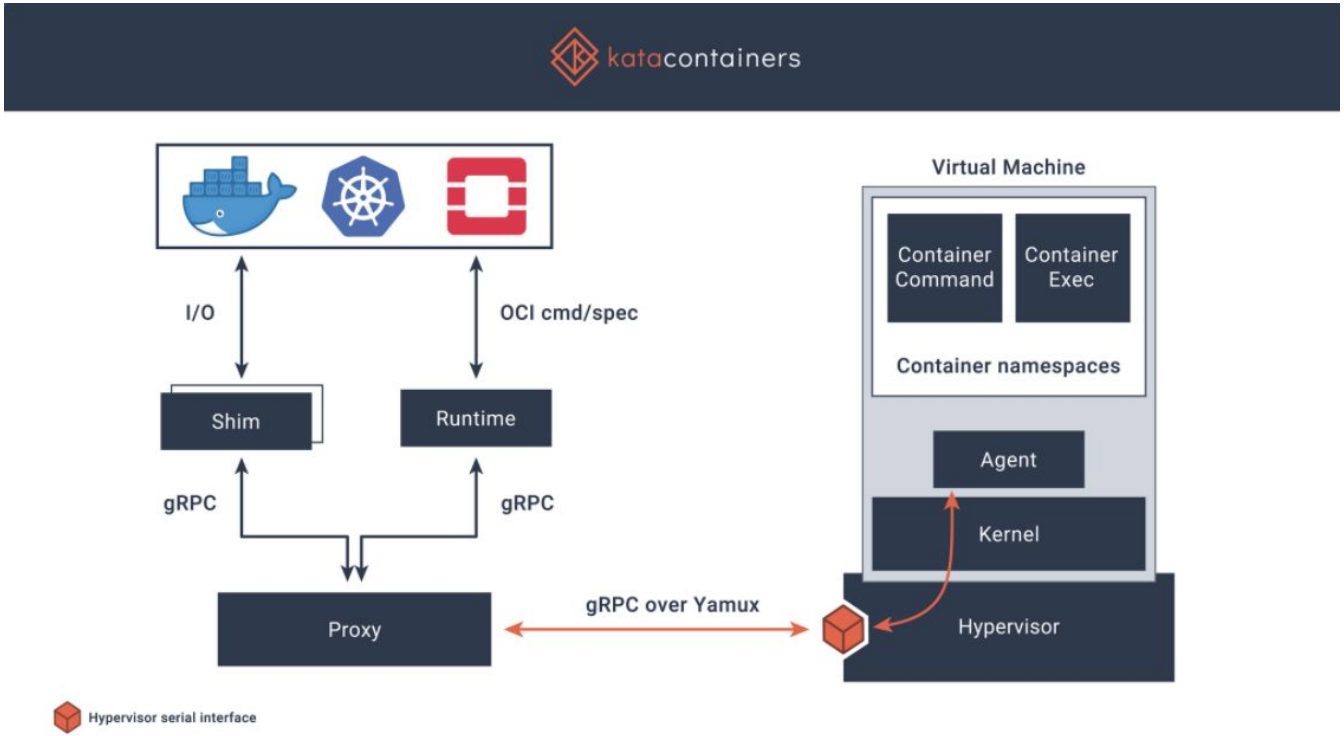
Modernize workloads and support mixed applications consisting of VMs, containers, and serverless



- OpenShift Virtualization accelerates application delivery with a single platform that can manage “mixed applications” with the same tools and teams
- Add VMs to new and existing applications
- Modernize legacy VM applications over time, or maintain them as VMs

Red Hat

Kata Containers



Red Hat

Why we integrate Kata into OpenShift?

Kata containers is an open source project developing a container runtime using virtual machines and providing the same look and feel as with vanilla containers.
By leveraging hardware virtualization technologies kata containers provides powerful workload isolation compared to existing container solutions.

We will be integrating kata containers into Openshift to provide the ability to run kernel isolated containers for any workload which requires custom kernel tuning (sysctl, scheduler changes, cache tuning, etc), custom kernel modules (out of tree, special arguments, etc), exclusive access to hardware, root privileges, or other administrative privileges above and beyond what is secure in a shared kernel environment (regular runc). .

Difference between Kubevirt and Kata

• Kubevirt aims to run **VM images** on Openshift providing the VM with the look and feel of a virtual machine running as a legacy VM (CRD defining a Virtual Machine in Kubernetes)	• Kata containers aim to run **container images** on Openshift in an isolated manner using virtualization tools (uses Runtime Class resource to choose Kata only for specific workloads which require kernel level isolation)

Our goals

- Provide a way to enable running user workloads on kata containers on an OpenShift cluster.
- Clusters with kata container workloads should support upgrades.
- Add security policy to control which users can run what workloads on kata in an OpenShift cluster.

2 fundamental problems we need to address

Getting the kata artifacts on to an RHCOS node ● Short term: Use container images to deliver RPMs and install using rpm-ostree ● Long term: Use RHCOS extensions (qemu-kvm-core and dependencies only)	Deploying kata and configuring CRI-O to use it as the runtime ● We will use an operator to deploy and configure CRI-O.

What is RHCOS extension?

- This will add additional software onto the host - but this software will still be versioned with the host (included as part of the OpenShift release payload) and upgraded with the cluster.
- The `rpm-ostree` project has had as its goal from the start to re-cast RPMs as "operating system extensions" - much like how e.g. Firefox and its extensions work. By default it operates as a pure image system, but packages can be layered on (and overridden) client side.

Kata Operator Development

- In order to use Kata Runtime, administrators need to login to the worker nodes of the OpenShift/Kubernetes cluster and manually install/configure the Kata Runtime as well as CRI runtime, such as CRI-O. `Kata Operator` aims to help administrators install, upgrade, and uninstall the Kata runtime (and it's dependencies) by extending OpenShift/Kubernetes API using Custom Resources and an Operator.
- Create an API which supports:
  - Installation of Kata Runtime on all or selected worker nodes
  - Configure the CRI Runtime, such as CRI-O, to use Kata Runtime on those worker nodes
  - Installation of the runtimeClass on the cluster
  - Perform updates to the Kata Runtime
  - Uninstall Kata Runtime and reconfigure CRI-O to not use it

The problems we are still working on

- Missing important features such as SR-IOV to support efficient network functions, or IPv6, …
- Concerns about security or supportability, i.e. issues that arise from turning Kata into a Red Hat product. These include SELinux support, better sandboxing of the qemu process using libvirt, …
- Product shaping concerns, either downstream (e.g. which kernels do we support in the guest?) or caused by upstream (e.g. change of the agent from Go to Rust)
- Usability concerns, e.g. documentation, configuration defaults, performance, …

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat