INTRODUCING CONTAINER-NATIVE VIRTUALIZATION

Cats and Dogs Living Together

Stephen Gordon  
Principal Product Manager  
Red Hat  
sgordon@redhat.com / @xsgordon  
10th May 2018

Fabian Deutsch  
Manager, Software Engineering  
Red Hat  
fdeutsch@redhat.com / @dummdida
FIRST A LITTLE HISTORY

Virtual Machines
FIRST A LITTLE HISTORY

Virtual Machines

Containers
VIRTUAL MACHINES AND CONTAINERS

VIRTUAL MACHINES

VM virtualizes the hardware

CONTAINERS

Container virtualizes isolates the process
Each of these attributes can be a positive or a negative for a given workload.

Increasingly organizations have a mix of both.
EXISTING SYSTEMS TREAT THESE SEPARATELY
WHAT ABOUT EXISTING WORKLOADS?

CONTAINER INFRASTRUCTURE AND ORCHESTRATION

Container Application and Kubernetes orchestration as provided by OpenShift are becoming the standard for new applications.
WHAT ABOUT EXISTING WORKLOADS?

**CONTAINER INFRASTRUCTURE AND ORCHESTRATION**

Container Application and Kubernetes orchestration as provided by OpenShift are becoming the standard for new applications.

**VIRTUALIZED WORKLOADS**

Virtualized Workloads are not going anywhere fast! Business reasons (cost, time to market) and technical reasons (older/different operating system)
WHAT ABOUT EXISTING WORKLOADS?

**CONTAINER INFRASTRUCTURE AND ORCHESTRATION**
Container Application and Kubernetes orchestration as provided by OpenShift are becoming the standard for new applications.

**VIRTUALIZED WORKLOADS**
Virtualized Workloads are not going anywhere fast! Business reasons (cost, time to market) and technical reasons (older/different operating system)

**CONVERGING INFRASTRUCTURE**
How can we bring these two worlds closer together?
OPENSHIFT CONTAINER PLATFORM

Benefit from:
- Placement
- Network security
- Isolation
- Quota
- etc.
WHAT IS CONTAINER-NATIVE VIRTUALIZATION?

Technology enabling developer use of OpenShift Container Platform as a unified platform for building, modifying, and deploying applications residing in both containers and virtual machines in a common, shared environment.

Add virtual machines to your OpenShift projects directly from the service catalog!
CONTAINER-NATIVE VIRTUALIZATION

Leverages tried and trusted RHEL (KVM) virtualization capabilities. Technology Preview access in an upcoming release of Red Hat Cloud Suite.
EXAMPLE USE CASE - START WITH A VM
EXAMPLE USE CASE - BUILD NEW SERVICES
EXAMPLE USE CASE - START DECOMPOSING
Resultant virtual machines are able to run side by side directly on the same OpenShift nodes as application containers.
LEVERAGE EXISTING ECOSYSTEMS

Use native OpenShift Primed storage solutions or Red Hat OpenStack Platform storage solutions via OpenStack Cinder interface (cloning support recommended).
DEEP DIVE
Upstream Projects

- **KubeVirt**
  The virtual machine operator

- **Containerized Data Importer (CDI)**
  Importing disks

- **Virt-v2v (APB)**
  Importing a whole virtual machine

- **OpenShift Web Console**
  With UI extensions
Import using virt-v2v

- Not shown in the Keynote demo for CNV

- Virtual Machines are imported into CNV using virt-v2v
  - From VMware, libvirt, and ova

- Initial work with upstream to get it right
OpenShift Web Console

- Additions to OpenShift Web Console to display virtual machines alongside pods
  - Metrics from pods
  - Direct reference to the underlying pod

- Creation of virtual machines from the UI via (if the specific service broker is enabled)
  - Templates* (also work from the CLI)
  - APBs (Ansible Playbook Bundles)
Templates & Service Catalog

- Virtual Machines can be created from OpenShift Templates like any other object
  - OpenShift Console was showing a new “Virtualization” category
  - Parameters, metadata, ...

- Template Service was used to make the Template discoverable in the Service Catalog

- Ansible Playbook Bundles (APBs) can be used for more complex setups
Disks - Import and storage

- Virtual Machine disks are mapped to and stored on **PersistentVolumes**
  - 1:1 mapping of disk to PV - Alignment with Kubernetes concepts
  - Mutable and immutable
  - Benefit from the Kubernetes/OpenShift ecosystem and thus indirectly from OpenStack Cinder and it’s own ecosystem

- Virtual Machine disks are imported using **CDI (Containerized Disk Importer)**
  - Ability to fetch disk images via http, ...
  - Upload is in the works
Offline and running Virtual Machine

- Offline Virtual Machine
- Define a virtual machine without running it
- Comparable to idling deployment

- Running Virtual Machine
- Created from the template section of an offline virtual machine
- Comparable to how pods are created from deployments
Scheduling

- Virtual Machine are scheduled as pods
  - Same set of features (affinity/anti-affinity, labels and selectors, taints and tolerations)
  - Custom scheduler as needed

- Applications within virtual machines are exported using Service and Routes
  - Selection using labels and selectors
Virtual Machine Operator and API

- Virtual Machines have their own kind
  - Ability to express all common virtual machine parameters and actions
  - Targeted feature set is comparable to libvirt

- Implemented as a CRD
  - Inheriting authn, authz, client tool support, ...

```yaml
apiVersion: kubevirt.io/v1alpha1
kind: VirtualMachine
metadata:
  name: vm-fedora
spec:
  domain:
    devices:
      disks:
        ...
    resources:
      requests:
        memory: 1024M
    volumes:
      ...
status:
  interfaces:
    - ipAddress: 172.17.0.12
  nodeName: localhost
  phase: Running
```
Virtual Machines and Pods

- Virtual Machines live in pods
  - Transparent to higher-level management systems (monitoring, metrics, ...)
  - Technically: Not worse than today

- Virtual Machines leverage pods
  - Metadata - Labels and annotations, passed through and additional
  - CPU and memory resources
  - Affinity and anti-affinity
  - Storage and network

- Virtual Machines specifics
  - Specific events
Virtual machine client tool

- Additional server sided virtualization functional needs to be exposed
  - Connecting to consoles
  - Starting and stopping
    - Live migration
- Leveraging the existing kubectl plugin mechanism
  - Standalone or as a plugin

```
$ kubectl plugin virt start myvm
$ kubectl plugin virt vnc myvm
```
Network - Connectivity and service integration

- Virtual Machine are connected to the regular pod network
  - From the outside there is no difference between a VM and pod

- Applications within virtual machines are exported using Service and Routes
  - Selection using labels and selectors

- Advanced networking (SR-IOC, L2, infiniband, ...) is under discussion in upstream Kubernetes
KataContainers? gVisor?

- Another use of KVM in the Kubernetes context

- Different use-case: Improve security of container workloads
  - Adding an isolation layer to pods

- Not suited or aimed to run general purpose virtual machines
DEMO (link)
FUTURE PLANS
FUTURE PLANS

Technology Preview enabled with OpenShift Container Platform in an upcoming release of Red Hat Cloud Suite.

We are looking for early adopters now!

Current “wish list”:

- Additional VM life-cycle actions
- Turn-Key storage solutions
- Additional networking options (Multi-net? SR-IOV?)
- VM ReplicaSets, DaemonSets, etc.
- Istio
We Want to Hear From You!

Provide your input on your infrastructure and use cases

Take this survey

Or visit the survey booth in the Feedback Zone located in:

Moscone West
L2 Lobby
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
facebook.com/redhatinc
twitter.com/RedHat
youtube.com/user/RedHatVideos
LINKS

- **KubeVirt**
  - Website
  - User Guide
  - API Documentation
- **Containerized Data Importer**
- **OpenShift Web Console**
- **Ansible Playbooks**