OKD Virtualization: what’s new, what’s next

Simone Tiraboschi
Principal Software Engineer - Red Hat
OKD
Virtualization
Agenda

▷ Intro: what are we talking about?
▷ CRC + OKD Virtualization: the simplest way to try OKD Virtualization @home
▷ New features:
  • Golden images
  • Kubevirt Tekton tasks
▷ What's next
What’s OKD?

OKD is a distribution of Kubernetes
OKD embeds Kubernetes and extends it with security and other integrated concepts
OKD adds developer and operations-centric tools
OKD is a sibling Kubernetes distribution to Red Hat OpenShift Container Platform (OCP for short)

Governance:

https://flickr.com/photos/6471971n08/6918935136/ “OKD & Kubernetes”
What’s OKD?

APPLICATIONS AND SERVICES
from Red Hat and community operators

PLATFORM AND CLUSTER MANAGEMENT
Kubernetes, security, monitoring, registry. etc

LINUX HOST with Fedora CoreOS*

FOR HYBRID / MULTI-CLOUD DEPLOYMENTS
What’s KubeVirt?

Kubernetes Virtualization API and runtime in order to define and manage virtual machines:

- Virtual machines
  - Running in containers
  - Using the KVM hypervisor
- Scheduled, deployed, and managed by Kubernetes
- Integrated with container orchestrator resources and services
  - Traditional Pod-like SDN connectivity and/or connectivity to external VLAN and other networks via multus
  - Persistent storage paradigm (PVC, PV, StorageClass)

https://flickr.com/photos/linein/2946303389/  
“A Container & VM”
Why KubeVirt? Using VMs and containers together

- Follows Kubernetes paradigms:
  - Container Networking Interface (CNI)
  - Container Storage Interface (CSI)
  - Custom Resource Definitions (CRD, CR)
- Schedule, connect, and consume VM resources as container-native
- Virtual Machines connected to pod networks are accessible using standard Kubernetes methods:
  - Service
  - Route
  - Ingress
- VM-to-pod, and vice-versa, communication happens over SDN or ingress depending on network connectivity
CRC +
OKD
Virtualization
How can I try it @home? CRC + OKD Virtualization

CRC is the quickest way to get started building OpenShift (OCP/OKD) clusters. It is designed to run on a local computer to simplify setup and testing, and to emulate the cloud development environment locally with all of the tools needed to develop container-based applications.

It's not intended for production use!!!

- Single node cluster which behaves as both a control plane and worker node
- It's an ephemeral cluster inside a VM
- The OpenShift cluster runs in a virtual machine known as an instance on your laptop => you have to use nested virtualization to play with OKD Virtualization
How to: 1. Setup CRC

Enable nested virtualization on your laptop:

```
$ sudo modprobe -r kvm_intel
$ sudo modprobe kvm_intel nested=1
```

Download CRC binaries from [https://github.com/crc-org/crc](https://github.com/crc-org/crc)

Tune CRC configuration:

```
$ crc config set disk-size 64
$ crc config set memory 20480
$ crc config set enable-cluster-monitoring true
$ crc config set preset okd
```

Configure prerequisites and download the bundle:

```
$ crc setup
```

Start your CRC instance:

```
$ crc start
```

sed 's/intel/amd/g' on AMD machines

add 'options kvm_intel nested=1' to /etc/modprobe.d/kvm.conf to make it persistent

Optionally enable the monitoring stack

Switch from OCP -> OKD

CRC is already pre-configured to use kubevirt-hostpath-provisioner as a dynamic provisioner for PVs backed by CRC VM’s filesystem.

No other manual actions are required on the CRC VM in order to be able to store and execute nested VMs inside the CRC VM.
After a few minutes...

```
Starting openshift instance... [waiting for the cluster to stabilize]
INFO Operator authentication is not yet available
INFO Operator deployment is not yet available
INFO Operator ingress is progressing
INFO Operator ingress is progressing
INFO Operator deployment is not yet available
INFO Operator authentication is not yet available
INFO Operator authentication is not yet available
INFO Operator authentication is not yet available
INFO Operator authentication is not yet available
INFO Operator network is progressing
INFO All operators are available. Ensuring stability...
INFO Operators are stable (2/3)...
INFO Operators are stable (3/3)...
INFO Adding crc-admin and crc-developer contexts to kubeconfig...
Started the OpenShift cluster.

The server is accessible via web console at:
https://console-openshift-console.apps-crc.testing

Log in as administrator:
Username: kubeadm
Password: Gb34l-761ct-W4d4f-SDucS

Log in as user:
Username: developer
Password: developer

Use the 'oc' command line interface:
$ eval $(oc oc-env)
$ oc login -u developer https://api.crc.testing:6443
```

stixabes@tk4s:~$
Deploy **KubeVirt HyperConverged Cluster Operator** from the OperatorHub.
Deploy **KubeVirt HyperConverged Cluster Operator**

Not needed on CRC since HPP is already preconfigured.
HCO got deployed
Create a CR to trigger HCO

Create HyperConverged

Configure via: Form view □ | YAML view □

Note: Some fields may not be represented in this form view. Please select "YAML view" for full control.

Name
kubevirt-hyperconverged

Labels
app=frontend

Infra
Infra HyperConvergedConfig influences the pod configuration (currently only placement) for all the infra components needed on the virtualization enabled cluster but not necessarily directly on each node running VMs/VMs.

Workloads
Workloads HyperConvergedConfig influences the pod configuration (currently only placement) of components which need to be running on a node where virtualization workloads should be able to run. Changes to Workloads HyperConvergedConfig can be applied only without existing workload.

Storages
Storages contains configuration for importing containerized data

defaultCPUModel
Deployment options

Enable this, we will use it later (you can do it also at day-2)
After a few minutes...

This was not there before...
Golden images
Golden Images: motivation

Hyperscalers (like AWS, GCP, Azure, IBM Cloud...) provide:

▸ root disk images for commonly used operating systems
▸ continuous updates of those images

Starting a fresh VM is really really quick and simple!

the KubeVirt ecosystem should provide the tools necessary for supporting a similar pattern of disk image availability within the cluster
Golden Images: design

Container Registry

Polls registry for updates

Newly detected container image SHA or TAG results in new PVC import

- PVC: fedora-32-03-01-2021-10
- PVC: fedora-32-04-11-2021-04
- PVC: fedora-32-05-04-2021-16
- PVC: fedora-33-06-11-2021-08
- PVC: fedora-33-07-21-2021-05
- PVC: fedora-33-08-09-2021-11

DataSource:
- Name: fedora
- PVC: fedora-33-08-09-2021-11

DataImportCron updates a managed DataSource to point to the corresponding latest import PVC
How they look like for cluster admins
How can I use them
On the storage side...
My nice VM is there...
And it has been customized from the template
DataImportCrons
Custom DatalImportCrons
Kubevirt
Tekton
Pipelines
Kubevirt Tekton Pipelines: motivation

How can I automate the execution of complex, long and error prone tasks?

Like creating a custom golden image and installing the OS, Configuring it, Updating...
Tekton

- Tekton AKA OpenShift Pipelines is a cloud-native, continuous integration and continuous delivery (CI/CD) solution based on Kubernetes resources.

- It uses Tekton building blocks to automate deployments across multiple platforms by abstracting away the underlying implementation details.

- Tekton introduces a number of standard custom resource definitions (CRDs) for defining CI/CD pipelines that are portable across Kubernetes distributions.

Still not available out of the box in community-operators.

You can install it with:

```bash
$ export TEKTON_VERSION=v0.64.0
$ oc apply -f "https://github.com/tektoncd/operator/releases/download/${TEKTON_VERSION}/openshift-release.yaml"
```
How does Tekton work?

- **Task**: Defines a set of build steps, such as compiling code, running tests, and building and deploying images.
- **Pipeline**: Defines the set of tasks that compose a pipeline.
- **PipelineResource**: Defines an object that is an input (such as a Git repository) or an output (such as a Docker image) of the pipeline.
- **PipelineRun**: Instantiates a Pipeline for execution with specific inputs, outputs, and execution parameters.
KubeVirt Tekton tasks provide OpenShift Virtualization - specific Tekton tasks, which focus on:

- Creating, updating and managing resources of KubeVirt (VMs, DataVolumes, DataSources, Templates)
- Manipulating disk images with libguestfs tools

https://github.com/kubevirt/kubevirt-tekton-tasks
Create Virtual Machines
- create-vm-from-manifest - create VM from yaml manifest
- create-vm-from-template - create VM from template

Utilize Templates
- copy-template - copy template
- modify-vm-template - update template metadata

Create DataVolumes/DataSources
- create-datavolume-from-manifest (4.11) - create dataVolume
- modify-data-object (>= 4.12) - create/delete dataVolume/dataSource

Generate SSH Keys
- generate-ssh-keys - generate SSH keys and store them in cluster

Execute Commands in Virtual Machines
- execute-in-vm: execute commands over SSH
- cleanup-vm: execute commands and/or stop/delete VMs

Manipulate PVCs with libguestfs tools
- disk-virt-customize: execute virt-customize commands in PVCs
- disk-virt-sysprep: execute virt-sysprep commands in PVCs

Wait for Virtual Machine Instance Status
- wait-for-vmi-status - wait for VM status
Tekton Task operator

- The Tekton Tasks Operator (TTO) is a operator that takes care of deploying Kubevirt tasks and example pipelines.

- Starting from OKD Virtualization 4.11, TTO is deployed by default, but does not deploy any resources until explicitly enabled.

- To enable resource creation - spec.featureGates.deployTektonTaskResources needs to be updated in the HCO CR:

```bash
$ oc patch hco kubevirt-hyperconverged --type=merge -p '
{"spec":
"featureGates":
{"deployTektonTaskResources": true}}'
```

https://github.com/kubevirt/tekton-tasks-operator
Tekton Task operator - Example pipelines

- windows10-installer - Populates golden windows 10 image in openshift-virtualization-os-images namespace
- windows10-customize - Installs sql server in golden windows image and creates new image and template

<table>
<thead>
<tr>
<th>Pipelines</th>
<th>PipelineRuns</th>
<th>PipelineResources</th>
<th>Conditions</th>
<th>Repositories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Name</td>
<td>Search by name...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Last run</td>
<td>Task status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>windows10-installer</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>windows10-customize</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example pipelines - windows10-installer

- Populates golden windows image in kubevirt-os-images namespace
- Installs virtio drivers
- The minimal required input is the URL to the Windows 10 ISO
- The default Windows 10 template will detect the golden image as default disk image automatically
- Pipeline definition:
Example pipelines - windows10-installer demo
Example pipelines - windows10-customize

- Creates golden images with customizations applied on top of a basic Windows installation
- Uses image created by windows10-installer pipeline
- Example pipeline installs MS SQL Server in Windows 10
Example pipelines - windows10-customize demo
What's next
What's next

OKD: New Patterns, New CI/CD Pipelines and a new CoreOS

- What we call OKD is now "OKD running on the latest stable release of Fedora CoreOS (FCOS for short)"
- OKD Streams built using Tekton pipelines

OKD CentOS Streams CoreOS ('SCOS' for short)
(a real upstream for OCP on RHEL9)


OKD Virtualization

- Enhance support for Tekton pipelines
- Additional metrics for monitoring and alerting
- ARM support
- Backup/Restore
- Non-privileged controller
- Real time virtual machine
- ....
OKD Working Group

Website
okd.io

Twitter
twitter.com/okd_io

Slack
#openshift-dev on kubernetes.slack.com

Google Group
groups.google.com/forum/#!forum/okd-wg

Bi-weekly Video Conference Meetings
apps.fedoraproject.org/calendar/okd

Repositories
github.com/openshift/community
github.com/openshift/okd
OKD Virtualization SIG

Reddit
www.reddit.com/r/OKD_Virtualization

Twitter
twitter.com/OKD_Virt_SIG

Slack
#virtualization on kubernetes.slack.com

Website
okd-virtualization.github.io

GitHub
github.com/okd-virtualization
Thank you