



OKD on oVirt

Gal Zaidman
Team Lead OpenShift on RHV,
Senior Software Engineer
09/2021

About me

I'm a Senior Software Engineer at Red Hat, working as part of the integration team and currently the team lead of the Openshift on oVirt effort.

I've started working at Red Hat 3.5 years ago as an intern and since then I have worked on:

- Openshift on oVirt.
- Virtio-win drivers installer.
- Python 2 to 3 porting effort.

[Blog](#)

[Linkedin](#)

[Github](#)



What we are going to talk about today

- What is OKD?
- Why OKD on oVirt?
- Demos



What is OKD?

OKD is a sibling Kubernetes distribution to Red Hat OpenShift.

OKD is an open source distribution of Kubernetes that adds features that bring a robust, flexible, and scalable container orchestration platform that is enterprise-ready.

OKD enables building, deploying, and managing container-based applications in any public or private data center.



What is OKD?

OKD can be installed on many environments such as:

Cloud Environments



Virtualization Platforms



Bare-Metal



OKD Features

OKD offers many features to automate, scale, and maintain your applications:

- ▶ [High Availability](#) (Cluster and Applications)
- ▶ [Lightweight Operating System](#) (Fedora Coreos)
- ▶ [Load Balancing](#)
- ▶ [Autoscaling](#) (Pods and Worker nodes)
- ▶ [Logging and Monitoring](#) (Prometheus & Elasticsearch)
- ▶ [Services Discovery](#)
- ▶ [Storage](#) (oVirt CSI)
- ▶ [Application Management](#) (S2I, Image registry)
- ▶ [Cluster Extensibility](#) (OLM, Operatorhub)



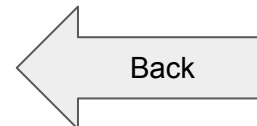
High Availability

Kubernetes has been designed with high availability in mind, for both internal components and user applications.

A highly available etcd cluster stores the state of the OKD cluster and its applications.

Resources stored in etcd, such as deployment configurations, provide automatic restarting of containers to ensure that your application is always running and that faulty containers are terminated.

This applies not only to applications, but also to containerized services that make up the cluster, such as the web console and the internal image registry.

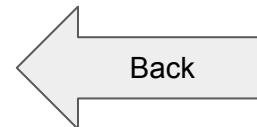


Lightweight Operating System

OKD runs on Red Hat Enterprise Linux CoreOS, Red Hat's lightweight operating system that focuses on agility, portability, and security.

RHEL CoreOS is an immutable operating system that is optimized for running containerized applications. The entire operating system is updated as a single image, instead of on a package-by-package basis, and both user applications and system components such as network services run as containers.

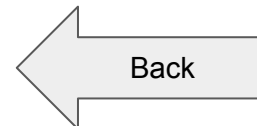
OKD controls updates to RHEL CoreOS and its configurations, and so managing an OKD cluster includes managing the operating system on cluster nodes, freeing system administrators from these tasks and reducing the risk of human error.



Load Balancing

Clusters provide three types of load balancers:

- External load balancer, which manages access to the OKD API. The technology that external load balancers use is dependent on the cloud provider that runs the OKD cluster.
- HAProxy load balancer, used by Route resources for external access to applications.
- Internal load balancer, which uses Netfilter rules to manage traffic from inside the cluster. Used by applications and services.

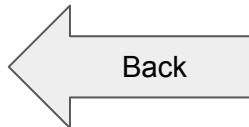


Autoscaling

OKD support scaling in two types of scaling:

- **Pod scaling**: adapt to traffic amount in real time by automatically starting new containers when load increases, and terminating containers when the load decreases.
- **Worker nodes scaling**: adapt to traffic amount in real time by automatically adding or removing worker nodes from the cluster according to the aggregated load from running applications, keeping responsiveness and costs down on public and private clouds.

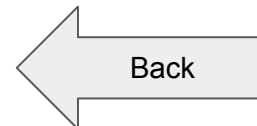
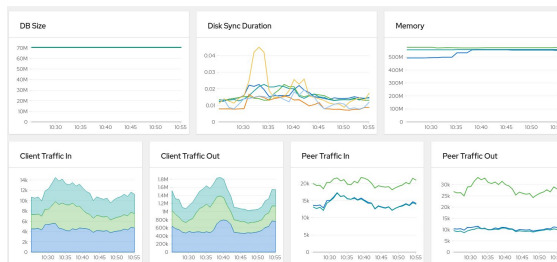
This features ensures that application's access time remains optimal regardless of the number of concurrent connections or activity.



Logging and Monitoring

OKD ships with an advanced aggregated logging and monitoring solution:

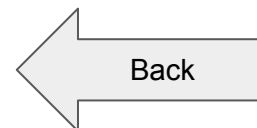
- **Monitoring:** based on Prometheus, which gathers hundreds of metrics about your cluster. This solution interacts with an alerting system that allows you to obtain detailed information about your cluster activity and health.
- **Aggregated Logging:** OKD ships with an advanced aggregated logging solution, based on Elasticsearch, which allows long-term retention of logs from cluster nodes and containers.



Services Discovery

OKD runs an internal DNS service on the cluster and configures all containers to use that internal DNS for name resolution.

This means that applications can rely on friendly names to find other applications and services, without the overhead of an external services catalog.



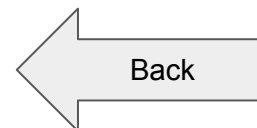
Storage

Kubernetes adds an abstraction layer between the storage back end and the storage consumption.

As such, applications can consume long-lived, short-lived, block, and file-based storage using unified storage definitions that are independent of the storage back end.

This way applications are not dependent on particular cloud provider storage APIs.

OKD embeds a number of storage providers that allow for automatic provisioning of storage on popular cloud providers and virtualization platforms, and so cluster administrators do not need to manage the fine details of proprietary storage arrays.



Application Management

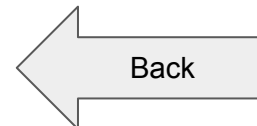
OKD empowers developers to automate the development and deployment of their applications.

Use the OKD Source-to-Image (S2I) feature to automatically build containers based on your source code and run them in OKD.

The internal registry stores application container images, which can be reused. This decreases the time it takes to publish your applications.

The developer catalog, accessible from the web console, is a place for publishing and accessing application templates. It supports many runtime languages, such as Python, Ruby, Java, and Node.js, and also database and messaging servers.

You can expand the catalog by installing new operators, which are prepackaged applications and services that embed operational intelligence for deploying, updating, and monitoring their applications.



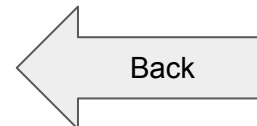
Cluster Extensibility

OKD adds features that are not provided by upstream Kubernetes.

OKD packages these extensions as operators for ease of installation, update, and management.

OKD also includes the Operator Lifecycle Manager (OLM), which facilitates the discovery, installation/update of applications and infrastructure components packaged as operators.

Red Hat, in collaboration with AWS, Google Cloud and Microsoft, launched the OperatorHub, accessible at <https://operatorhub.io> a repository and marketplace for operators compatible with OKD and other distributions of Kubernetes that include the OLM.



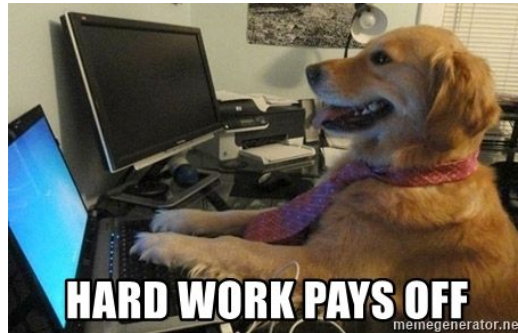
Why oVirt?

- Supports multiple installations methods:
 - Installer Provisioned Installation w/o customisations (IPI) - Extremely easy installation, just ran the installer on your existing oVirt environment and you are good to go.



Why oVirt?

- Supports multiple installations methods:
 - Installer Provisioned Installation w/o customisations (IPI) – Extremely easy installation, just ran the installer on your existing oVirt environment and you are good to go.
 - User Provisioned Installation(UPI) – manual work is required but allows for more customizations then IPI.



Installation methods

A light gray rounded square containing the text 'IPI' in bold black font.

IPI

vs

A light gray rounded square containing the text 'UPI' in bold black font.

UPI

Installation methods

Installer Provisioned w/o customisations (IPI):

- Simplified opinionated “Best Practices” for cluster provisioning.
- Installer creates all the VMs for the OKD cluster on the oVirt environment.
- Allows for some customizations through the OKD installer, for details see [Docs](#).

Installation methods

User Provisioned infrastructure (UPI):

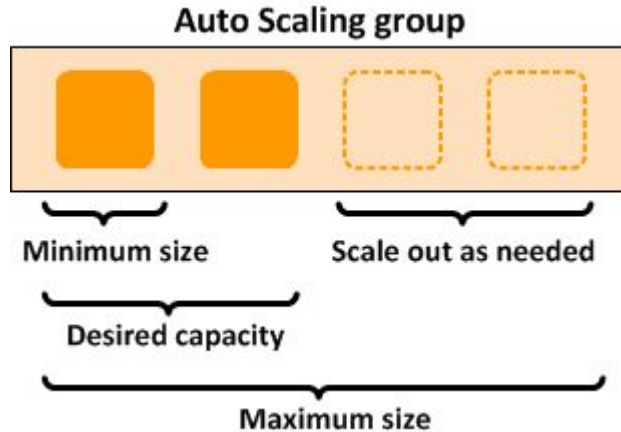
- User managed resources & infrastructure provisioning
- User creates all the VMs for the OKD cluster on the oVirt environment.
- Allows for the most customizations but comes with the price.
- Currently documented only without platform provider which means no machine management, scaling and CSI support - this will change in the next version.

Why oVirt?

- Supports multiple installations methods (IPI and UPI).
- Easy to maintain:
 - Maintaining virtual machines is easier than bare metals.
 - oVirt provider allows managing the OKD machine lifecycle from OKD CLI/UI without going through the oVirt engine.

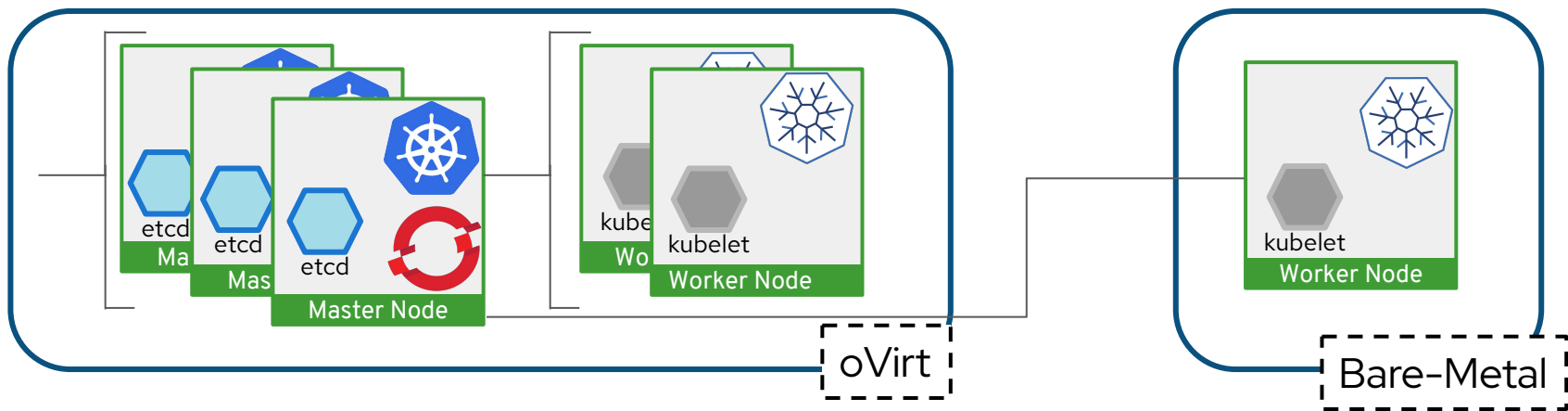
Why oVirt?

- Supports multiple installations methods (IPI and UPI).
- Easy to maintain.
- Supports scaling and autoscaling of oVirt VMs.



Why oVirt?

- Supports multiple installations methods (IPI and UPI).
- Easy to maintain.
- Supports scaling and autoscaling of oVirt VMs.
- Support mixed env - an OKD control plain can manage oVirt workers and bare metal workers on the same cluster.



Why oVirt?

- Supports multiple installations methods (IPI and UPI).
- Easy to maintain.
- Supports scaling and autoscaling of oVirt VMs.
- Support mixed env (oVirt/bare metal workers on the same OKD cluster).
- Take advantage of virtualization features such as:
 - Resource Overcommitment (mostly for dev/test environments).
 - Thin provisioning.
 - And all the great features oVirt offers.

Why oVirt?

- Supports multiple installations methods (IPI and UPI).
- Easy to maintain.
- Supports scaling and autoscaling of oVirt VMs.
- Support mixed env (oVirt/bare metal workers on the same OKD cluster).
- Take advantage of virtualization features.
- Fully utilize your hardware.

Installation demo

Prerequisites

- All the Prerequisites listed in the docs, [Link](#).
- Two static IP addresses for API/IngressVIPs, [Link](#).
- [DHCP lease time](#) of above 30m

Notes:

- Currently 4.9 release doesn't work on oVirt because of an fcos issue.

Autoscaler demo

Prerequisites

- Read docs at [Link](#) and create a ClusterAutoscaler and a MachineAutoscaler yamls according to your needs.



Thank you!

<https://ovirt.org/>

users@ovirt.org

 @ovirt