Red Hat OpenShift

Software defined compute

Domenico Piol

Senior Solution Architect

Red Hat

What we'll discuss today

- What is SDC?
- A history lesson - and what are VM's, what are containers?
- How does reality look like and what's the value of SDC?
- Demo
- What's next? - another small demo
- Q & A

Software-defined compute (subset of software-defined data center) is when compute resources are **virtualized and abstracted from the hardware**, creating an operation that can be **managed through a central interface**.

Virtualization is a mainstream technology in today's data centers, originally deployed for server consolidation. Today, customers are reformulating their virtualization strategies around private, hybrid, and multi-cloud architectures. In addition, **containers** have emerged, sparked by Kubernetes, as a modern application platform that can be deployed **on premises, in a public cloud, and on the edge**, often in conjunction with VMs and SaaS.

Source: IDC

Once upon a time in the datacenter

Dedicated, Bare Metal

Applications are deployed on dedicated bare metal servers. In consequence, the servers are often over dimensioned and have inefficient utilization rates.

Virtualized

2001 - VMWare releases ESXi
Machine virtualization is used to consolidate and isolate applications on virtual machines. This increases the application density on bare metal servers but the VM images are very large.

Container

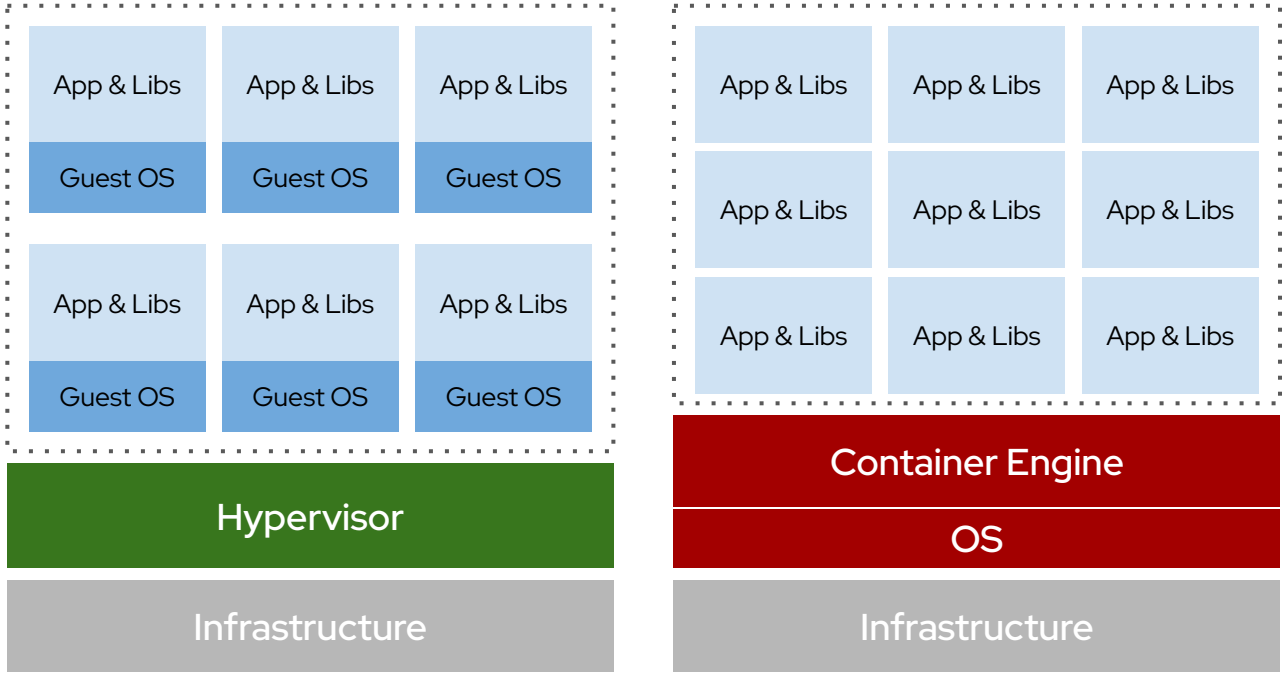
2013 - Docker launched, Kubernetes in 2014
Containerization is established as a trend. A container starts faster than a VM and shares the operating system with other containers, thus increasing application density again.

Red Hat

What are VM's, and what are containers?

VM's are virtualizing servers, whereas containers virtualize applications!

The packaging bundle of a VM contains the application, libraries and the full guest-os - which can be of considerable size.
On the other hand, a container image only contain the application, it's libraries and a very thin layer of system libraries (enabling the communication between the container and the system-kernel of the host/node-os).

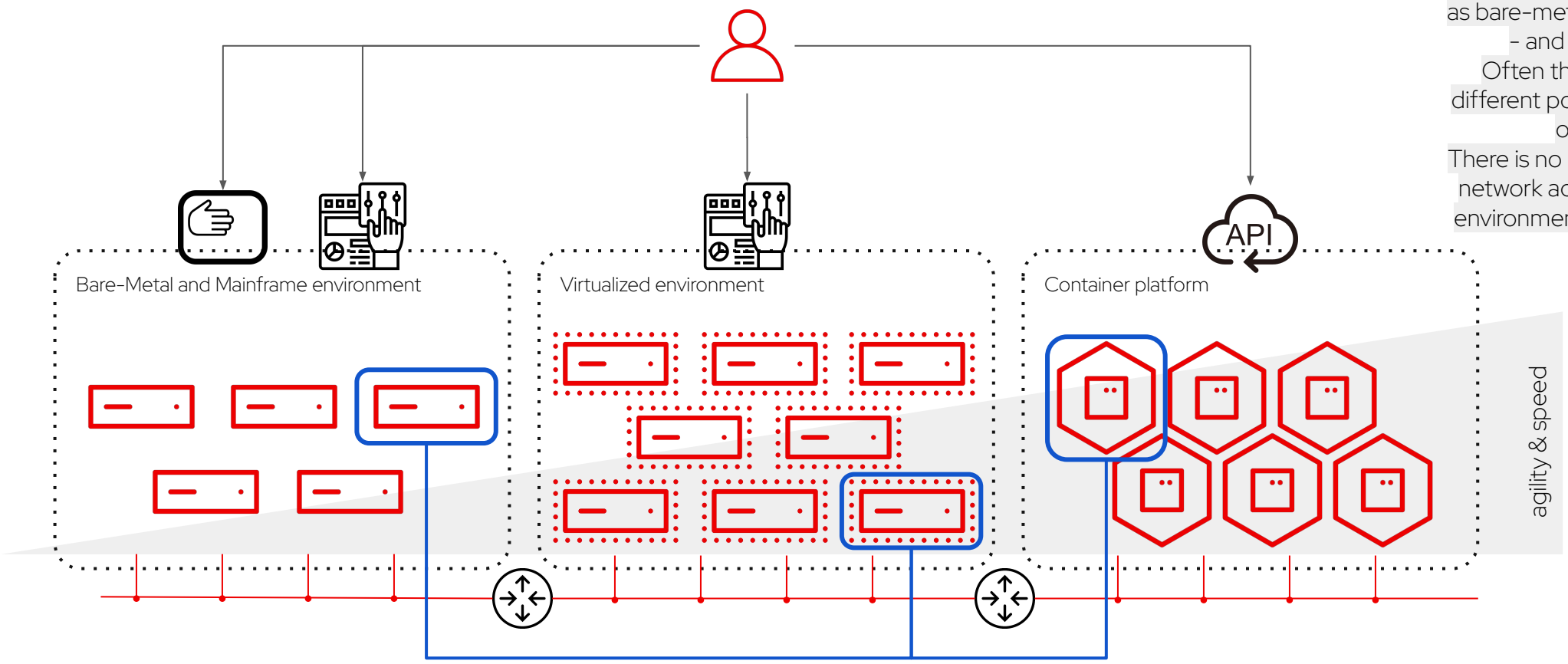


containers allow a much higher density, at the cost of a tighter binding to the node-os

Red Hat

How does reality look like today?

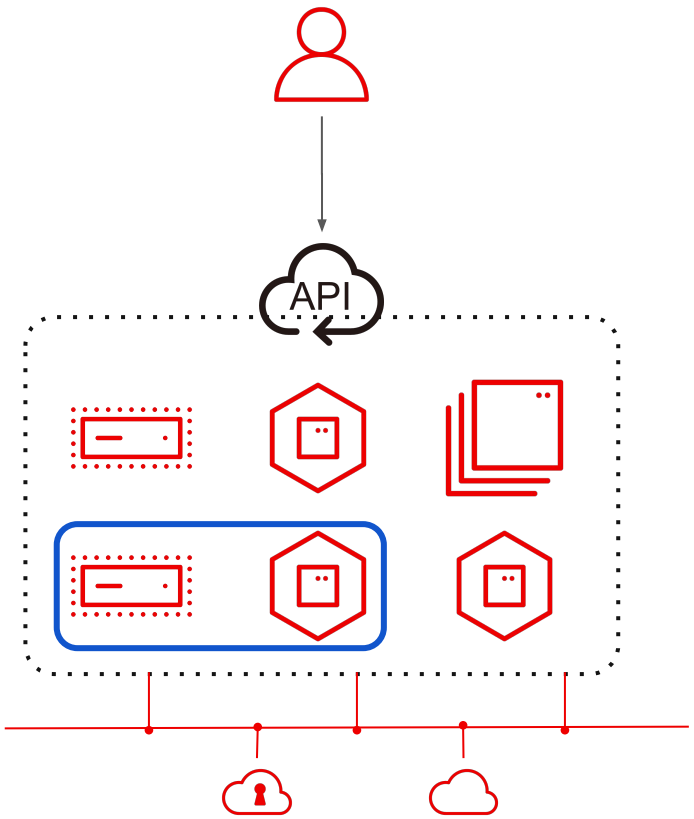
Large companies typically have a diverse landscape of target environments - such as bare-metal, virtualized and containers - and this on- and off-prem.
Often the interaction happens via different portals, deployment-tools and often manual tasks.
There is no consistent management and network across the environments. The environments often have very different maturity-levels.



most critical, large & complex applications are mixed!

Red Hat

The promise of “Software Defined Compute”



“... when a compute function is virtualized and abstracted from the hardware it’s on, creating an operation that can be managed through a central interface.”

In principle this means define you application’s needs as code and consume compute via an API... the same, everywhere!

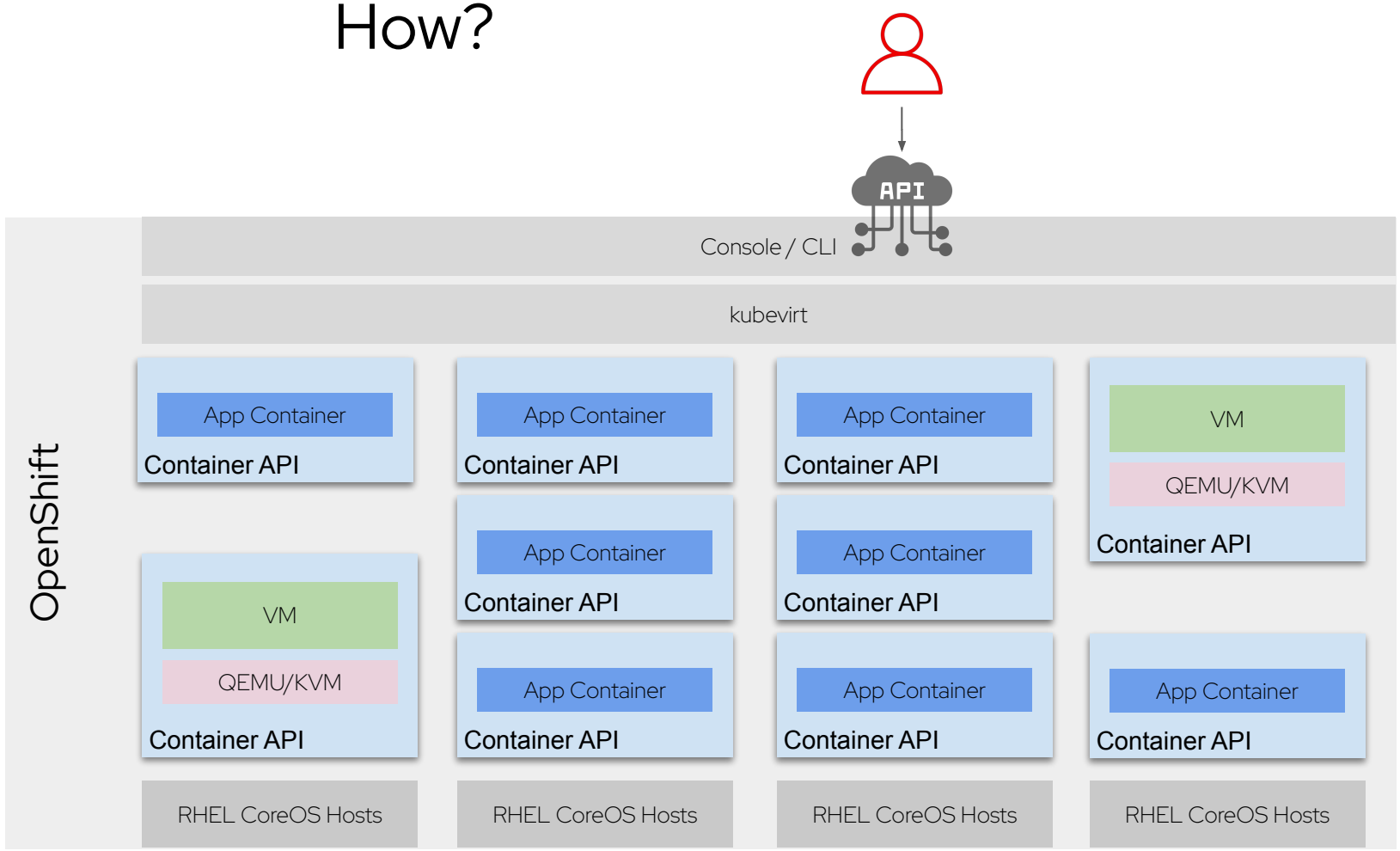
We want to offer a compute platform where the applications can be moved around to different pieces of virtual infrastructure, depending on the availability of resources. Software defined compute architecture address customer’s next generation IT requirements such as agility, speed and scalability.

Red Hat

How?

All objects in OpenShift - container and virtualized - are sharing the **same API**, **compute resources** and **management interface**. With that, all parts of an application as a whole (and the infrastructure) can be **managed consistently** by the same CI/CD pipeline!

This continuity - without a media break - decreases effort and boosts efficiency considerably!



Red Hat

Why?

Because compared to traditional compute, software-defined compute helps to gain the following:

- **Speed and Agility** - deliver new services faster and where suitable (special HW)
- **On the fly scaling** - a lot of flexibility w/o additional HW.
- **Operational benefits** - such as snapshots, immutability/reproducibility, return capacity if not required, et al
  → everything-as-code
- **Lower IT costs** - virtualization - and containerization in particular - leads to higher density of apps on the same compute resources. Everything-as-code enables extensive automation!
- **Be innovative!** - The higher efficiency allows app-dev organizations to explore instead of manage infrastructure.

And remember, it's an evolution! The first step being virtualization, the next containerization. And as in any evolution you will **see both of them for a long, long time** concurrently as applications can be complex and take time to evolve - or modernize. This leads to hybrid/mixed application topologies, and this calls for:

- **Consolidation and standardization** of compute environments (consistent API and central management interface)

Additional things to consider

- **It's cultural** - involve everybody!
- **Virtualize (and containerize) as much of your application as u can** - more is virtualized, bigger the benefit!
- **Everything as Code!** Define and manage all resources as code. → see following slides
- **Automate & Orchestrate** - no manual tasks, that is where the money goes.
- **Security** - micro-segmentation now becomes easy to implement! → see following slide
- **Service catalog** - needed for consuming for consuming services, enable developers to consume
- Think of **Disaster Recovery** (and BCP) and HA in general. Old paradigms might not be sufficient anymore.

Everything as code

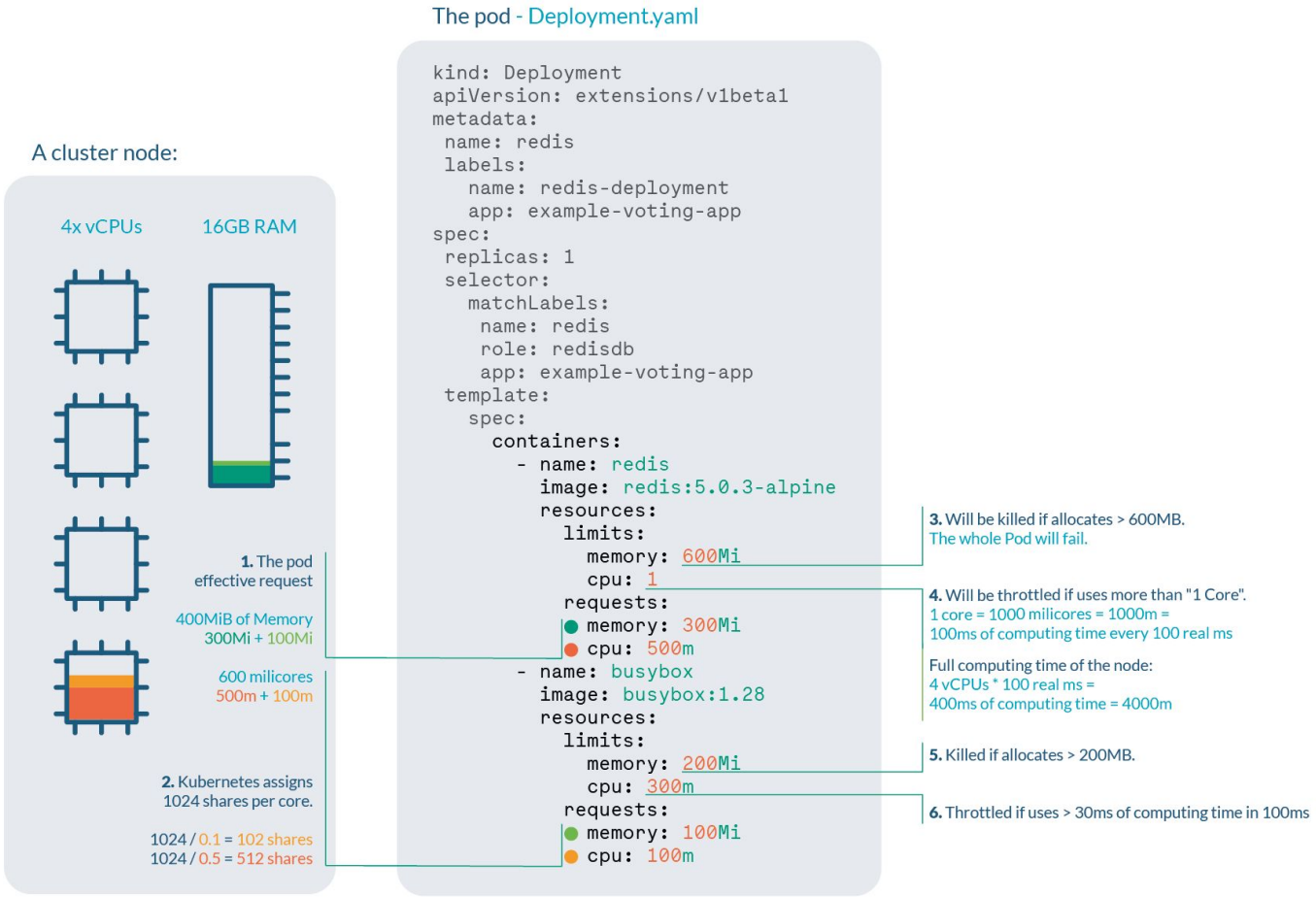
The key to automation, and with that to efficiency and speed, is everything-as-code. It means that every aspect of (any) compute resources is described as code and can so be consumed by an API.

So, before the age of SDC - when you bought a new physical server - you specified things like no of CPUs, Memory, Storage size and no of machines to buy in the online shop, right? So how does this work these days? Let's look at a container for example on the right:

This for containers... and for a VM or other compute resources within OpenShift? Precisely, the same! → Consistency

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: sccstore-ui
spec:
  replicas: 2
  strategy:
    spec:
      containers:
      - name: sccstore-ui
        image: sccstore-ui:latest
        volumeMounts:
        - name: sccstore-pv-claim
          mountPath: "/var/my-storage/"
        resources:
          requests:
            memory: "256Mi"
            cpu: "700m"
          limits:
            memory: "300Mi"
            cpu: "2000m"
      volumes:
      - name: mysql-persistent-storage
        persistentVolumeClaim:
          claimName: sccstore-pv-claim
```

Resource management



However, even if it looks like magic, it is not and the consumed resources must still be backed by real physical assets!

You cannot consume 32GB of RAM on a 16GB host… well, kind of you can but not at the same time for all containers or VMs on the host.

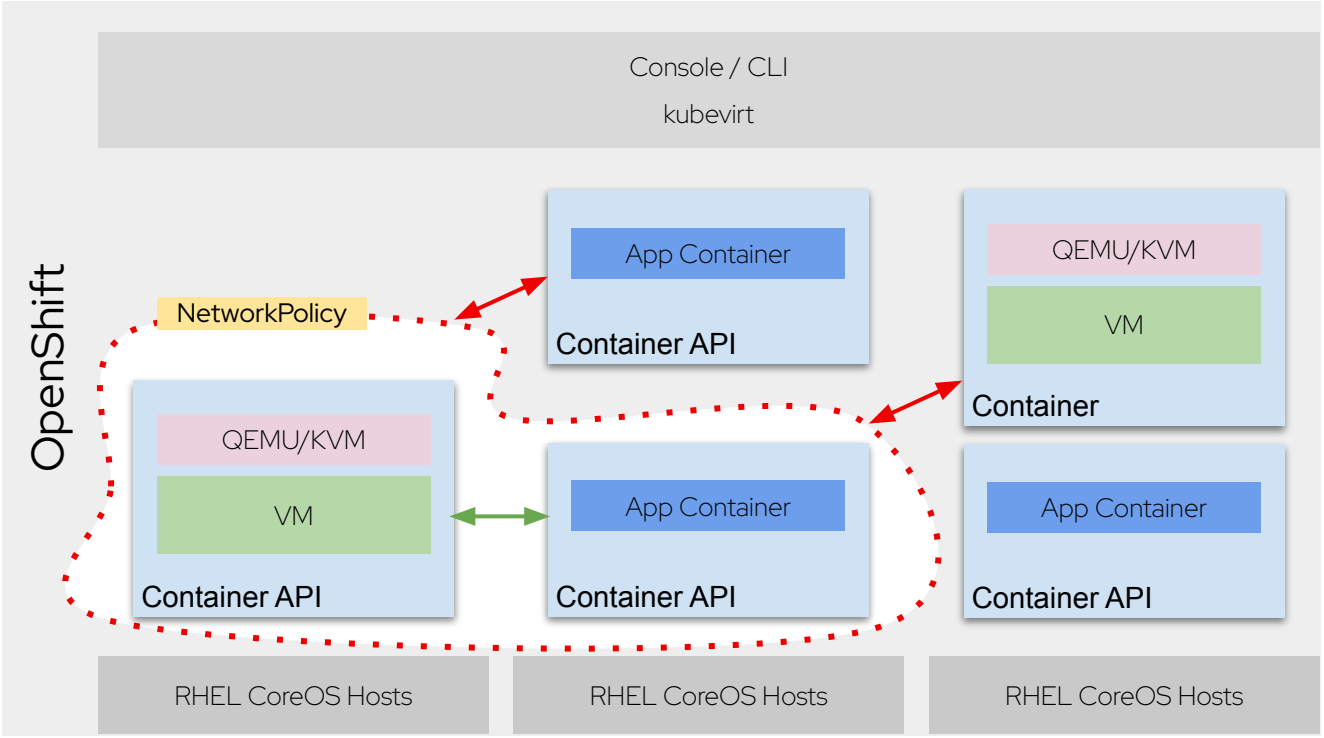
→ The agility of scaling on-demand has its (physical) limits!

Enhanced security with micro-segmentation

Microsegmentation is a security method of managing network access between workloads. With microsegmentation, administrators can manage security policies that limit traffic based on application identity. Microsegmentation is used to reduce the attack surface, improve breach containment and strengthen regulatory compliance.

As all application components are managed by the same platform and API, the NetworkPolicy can be easily used to control traffic.

In a heterogeneous environment this would be much more difficult and most likely involving additional tooling like GuardiCore for bare-metal, NSX-T/V for ESXi, et al, making it very difficult to manage.



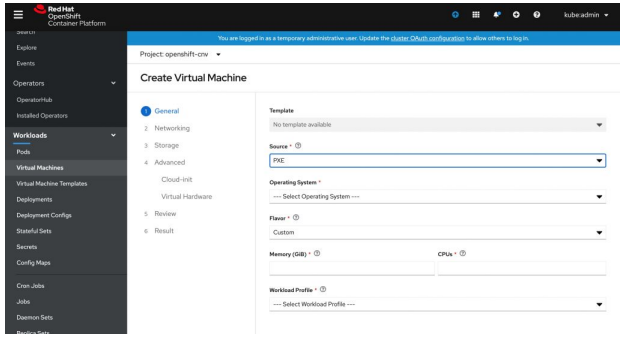
Red Hat

How to get a VM into OpenShift?

Create

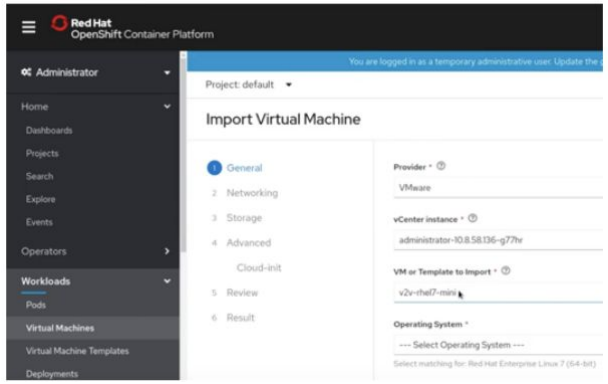
You can create a VM from the OpenShift CLI (from YAML), but it is also very easy to do from the Web Console.

You do not require cluster administrator privileges to create a VM with OpenShift Virtualization. Any OpenShift user who has the ability to run containers, that is, users with admin or edit roles in a project, are able to create VMs.

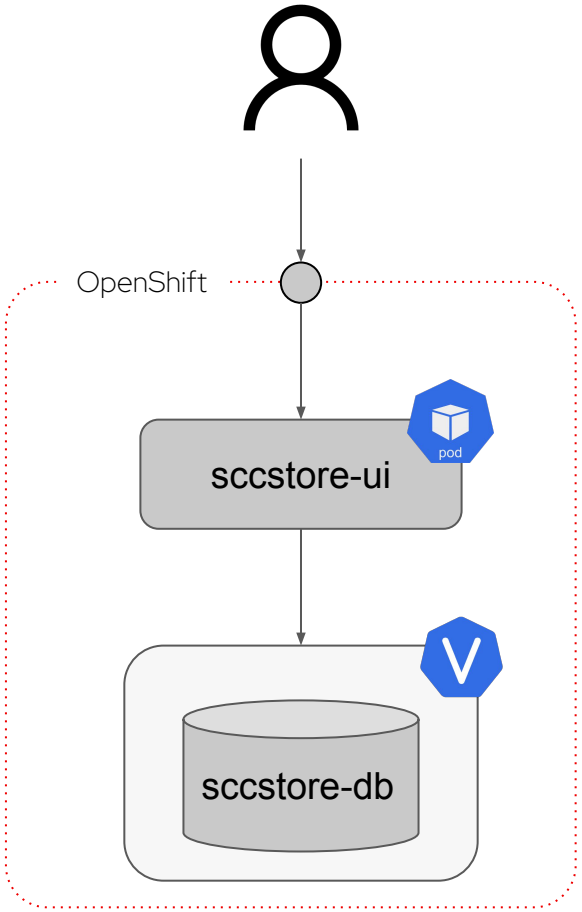


Import

OpenShift also provides an Import Virtual Machine wizard so VMs, and the applications running on them, can be migrated from vSphere, Red Hat virtualization, and Red Hat OpenStack Platform directly to OpenShift. You can also use the Import feature to import reusable templates for immediate use or when creating new VMs.



Red Hat



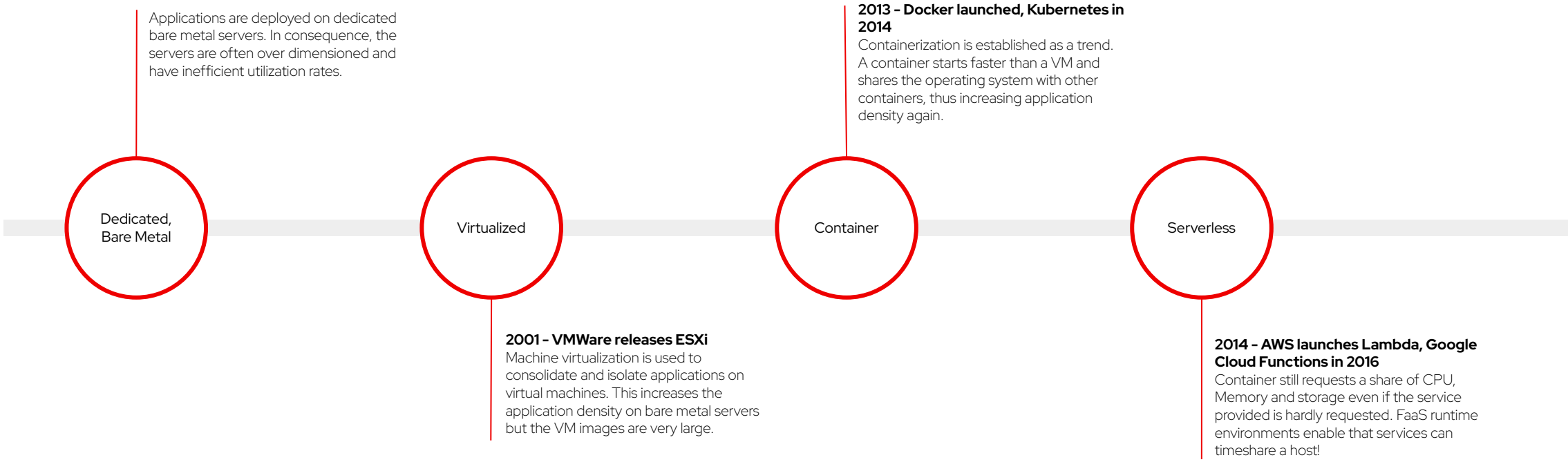
Demo

Consistent management of hybrid applications with virtualized and containerized components.

Red Hat

So, that was cool... but what's next?

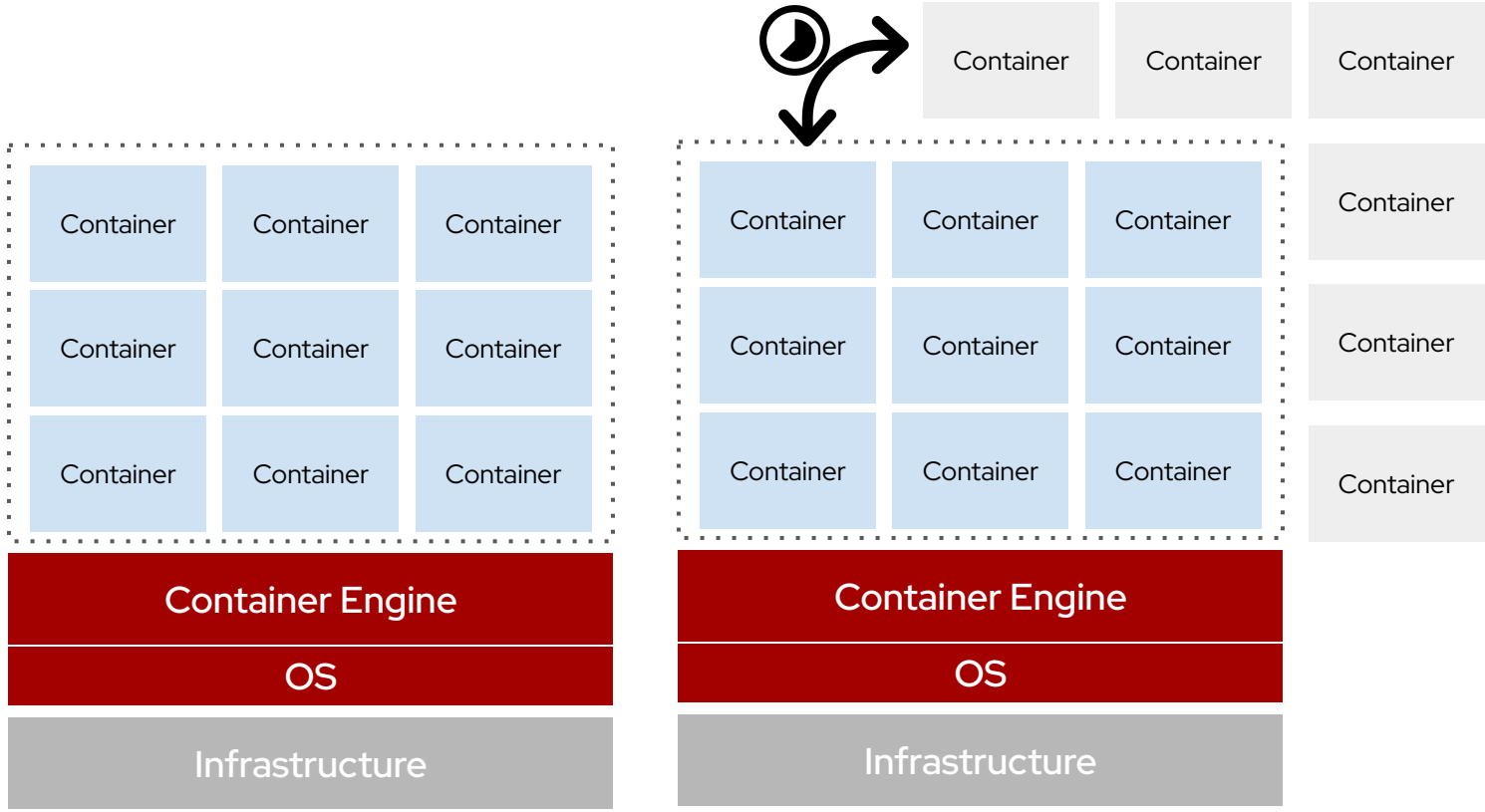
Once upon a time in the datacenter – part II



Red Hat

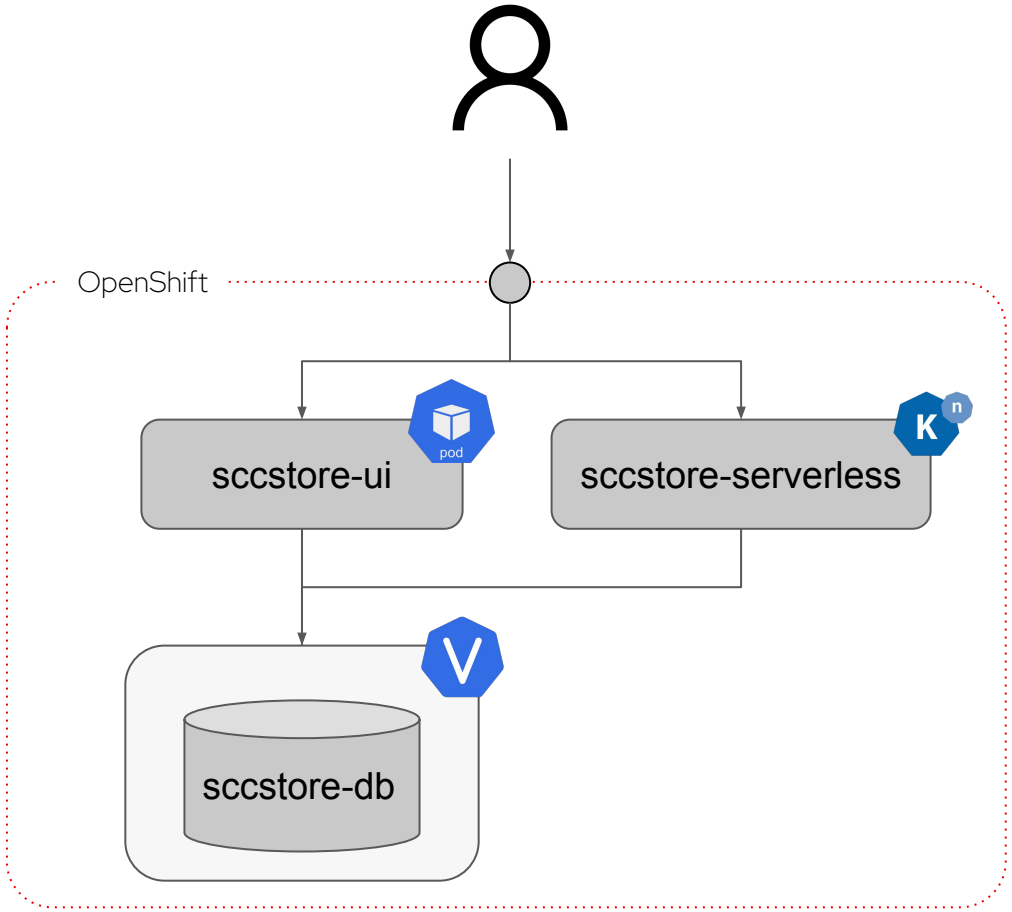
Serverless compared to traditional containers?

Serverless computing is an execution model in which a provider dynamically allocates only the compute resources and storage needed to execute a particular piece of code. This said, serverless applications are sharing hosts on a timely basis.
This means resource allocation is based on their computation and do not have to reserve and pay for a fixed amount of memory or CPU, as the service is auto-scaling.



by time sharing a host, the density can overall be increased over a period of time

Red Hat



Demo

Consistent management of hybrid applications with virtualized, containerized and serverless components.

Red Hat

What did we learn today

- SDC is about virtualization (vm, containers or serverless), automation and the consistent management of resources
- Today's reality is a heterogeneous landscape of compute resources, often managed within silos by different teams with different api's and tooling
- SDC helps reducing costs by increasing the easy-of-use of resources, the utilization of compute resources (density), automation and consistency
- SDC can increase app-security, management and agility because of the homogeneity of the environment (seamless API)
- **OpenShift** facilitates this journey as it covers all aspects, including migration from legacy environments


Let me in, I want to see with my own eyes!
No, it's no use...
Aha! I knew it! Servers!
Um...
And these? Aren't these hard drives? And these cables?
Yeah...
You told me I'd be getting SERVERLESS architecture!
You've been caught red handed! What a scam!


Q&A

Red Hat

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