

Kubernetes and OpenStack

Competing or complementing?



kubernetes



Kubernetes and OpenStack together in one single presentation.

Last week I was busy finalizing a presentation for an event in South Africa. To see if the organizer of the event agreed to the scheduled topics, we setup a conference call and ran through the presentation. At one point I got to the part in the presentation where I explained that Kubernetes is a perfect fit with OpenStack. There a discussion started because he had the impression that Kubernetes and OpenStack were competing solutions. Then I thought to myself; “how nice would it be to write a short article about Kubernetes and OpenStack: competing or complementing?”

What is Kubernetes?

Kubernetes is an orchestration system for automating the deployment, scaling and management of applications in containers. It was originally designed by Google, and is now maintained by the Cloud Native Computing Foundation. It aims to move applications easily across clusters of hosts, independent from the cloud platform or container technology that is used, including Docker.

What problem do containers solve?

Think of moving a new application component from the developer's laptop to a test environment, from a staging environment into production, and perhaps from a physical machine in a data center to a virtual machine in a private or public cloud. If you are used to using virtual machines, you know that no matter what hypervisor you choose, when you move a virtual machine from one computing environment to another, stranger things will happen. The reason being that the supporting software environments are not completely identical. And it's not just different software that can cause problems, for example the network topology, security policies, the storage solution, basically every component in the infrastructure, might be different and therefore cause an issue with the application. Basically containers are a solution to this problem. How to get software to run reliably when moved from one computing environment to another.

How does this work?

Put simply, a container consists of an entire runtime environment: an application, plus all its dependencies, libraries and other binaries and configuration files needed to run it, bundled into one package. By containerizing the application platform and its dependencies, differences in Operating System distributions and underlying infrastructure are abstracted away.



Fairbanks
Postbus 255
3800 AG Amersfoort

Basicweg 12e
3821 BR Amersfoort

tel. +31(0)33 450 85 85
e-mail info@fairbanks.nl

bank NL 29 ABNA 0244 6430 59
NL 09 RABO 0123 7445 39

kvk 32102827
btw NL813226193B01

Fairbanks.nl

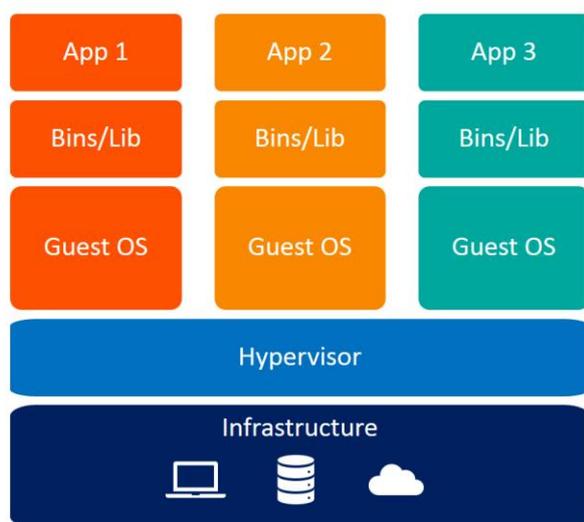
Kubernetes and OpenStack

Competing or complementing?

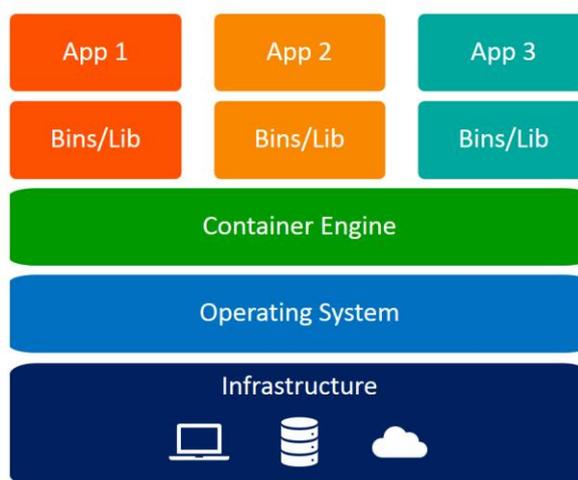
How do containers differ from virtual machines?

With virtualization technology, the package that can be passed around is a virtual machine, and it includes an entire operating system as well as the needed applications. A physical server running three virtual machines can have a hypervisor and for example three separate operating systems running on top of it.

By contrast a server running three containerized applications with Docker runs a single operating system, and each container shares the operating system kernel with the other containers. Shared parts of the operating system are read only, while each container has its way to access the container for writing. That means the containers are much more lightweight and use far fewer resources than virtual machines.



Virtual Machines



Containers

Do containers have any other benefits over virtualization?

A container may be only tens of megabytes in size, whereas a virtual machine with its own entire operating system may be several gigabytes in size. Because of this, a single host can run far more containers than virtual machines.



Kubernetes and OpenStack

Competing or complementing?

Another major benefit of containers is that virtual machines may take several minutes to boot up their operating systems before they begin running the applications they host, while containerized applications can be started almost instantly. That means containers can be installed in a "just in time" fashion when they are needed and can disappear when they are no longer required, freeing up resources on their hosts.

A third benefit is that containerization allows for greater modularity. Rather than run an entire complex application inside a single container, the application can be split in to modules, not only separating the database from the application front end and back end but also isolating individual application functions in separated service components. This is called the microservices approach. Applications built in this way are easier to manage because each component is relatively simple, and changes can be made to modules without having to rebuild or reboot the entire application. Because containers are so lightweight, individual modules (or microservices) can be spawned as they are needed and are available almost immediately.



What are the downsides?

Containers depend on the kernel, so they are not as isolated from each other as virtual machines; if there is a vulnerability in the kernel it can jeopardize the security of all containers as well. Also the persistent data storage is complicated; by design all of the data inside a container disappears forever when the container shuts down, unless you save it outside of the container first. Furthermore graphical applications don't work well, for example Docker was designed as a solution for deploying server applications that don't require a graphical interface. While creative strategies exist which are used to run a GUI application inside a container, these solutions typically don't work optimal at this time yet. So not all applications will benefit from container technology.



Fairbanks
Postbus 255
3800 AG Amersfoort

Basicweg 12e
3821 BR Amersfoort

tel. +31(0)33 450 85 85
e-mail info@fairbanks.nl

bank NL 29 ABNA 0244 6430 59
NL 09 RABO 0123 7445 39

kvk 32102827
btw NL813226193B01

Fairbanks.nl

Kubernetes and OpenStack

Competing or complementing?

A second downside of containers is that the management can be overwhelming. The management tools that are available to orchestrate large numbers of containers are not yet as comprehensive as software for managing virtualized infrastructures and clouds.

So as virtualization and containers have their own use cases and orchestration tools, they complement more than compete each other. Combining them in 1 platform enables you to run some applications in lightweight virtual machines to increase isolation and therefore security, and others in full virtual machines. While managing the shared resources of networking, compute and storage in an easy and familiar way.



Competing? No, complementing!

Sometimes OpenStack and Kubernetes are seen as competitive platforms, because they both manage and orchestrate cloud resources and services on top of that. One way to combine them is to run Kubernetes clusters within VMs, on OpenStack. This adds the ease of management with strong security segregation of VM technology to the containers in your Kubernetes clusters. As well as the reliability and resilience of VMs. Of course, these benefits come at a price: the overhead of the hypervisor. In our experience though this typically grows to approximately 4 percent of peak system performance. So if 4 percent is a price you can afford on your Kubernetes cluster, the combination of Kubernetes and OpenStack provide you with the ideal flexible cloud platform that allows you to deploy bare metal, virtual machines and container resources, all on a single network within a single pane of glass.

OpenStack supports the largest number of enterprise storage and networking systems today, and running Kubernetes with OpenStack allows you to seamlessly integrate containers into your environment. Whether you want to run containerized applications bare metal or VMs, OpenStack lets you run containers the best way for your business.

So OpenStack gives the perfect framework for exploring and implementing containers within any organization. Please let us know if you want to explore how OpenStack, Docker, Kubernetes or their combination can thrive business value in your organization.



Fairbanks
Postbus 255
3800 AG Amersfoort

Basicweg 12e
3821 BR Amersfoort

tel. +31(0)33 450 85 85
e-mail info@fairbanks.nl

bank NL 29 ABNA 0244 6430 59
NL 09 RABO 0123 7445 39

kvk 32102827
btw NL813226193B01

Fairbanks.nl