

Containerization as an Alternative to Virtualization

Having several containers running separate services can help with security protection and flexibility

By [Scott Gerenser & Andy Calvanese](#) .

Published: May 15, 2020

Scott Gerenser is a senior software engineer at Wheatstone Corp. and Andy Calvanese is vice president of technology.



Scott Gerenser

One term popping up more and more in the cloud space is “containerization.” If you’re paying attention to the trends in cloud computing, you’ve probably heard about it, or at least about the most popular container platform, Docker.

Containerization is becoming a popular alternative to virtualization for running many different applications on a single machine or cloud instance. It has many of the benefits of virtualization but without some of the downsides, which makes it useful for transitioning from a fixed-location studio to a virtual operation.

[\[Read: "Don't Be Afraid of AoIP"\]](#)

Whereas virtualization involves emulating an entire machine, including the hardware and operating system, containerization involves encapsulating one or more applications and supporting files (so called "userspace" in Linux lingo) into containers that can then run on top of a single common operating system (usually Linux).

BENEFITS

For example, in a virtualization scenario, you might have a server running VMWare ESXi hypervisor software, upon which are four Ubuntu Linux virtual machines for Service X, two Red Hat Linux VMs for Service Y, and a couple of Windows 10 Server VMs to handle any Windows applications you have.

Using virtualization this way still provides big benefits over maintaining multiple physical machines. Administration is easier. Spinning up a new server or changing configurations of the individual VMs is much easier than tinkering with hardware. Communication between the VMs is very fast and efficient.

The downside, as compared to containerization, is the relatively large overhead associated with virtualization. This is because each VM is running a complete OS kernel, each

with their own dedicated memory and each using up a percentage of your CPU to mostly do a lot of the same things.

Containerization, by comparison, also allows the running of a number of different isolated services on one machine, but within containers rather than full virtual machines.

Conceptually, a container can be thought of as a very lightweight, resource efficient VM. One container could host WheatNet-IP audio drivers and audio playback software, while another could host the station automation system, each totally isolated yet run off the same OS kernel.

Because each container operates independently of the others, you can avoid unintended interactions between software components and eliminate a single point of failure. Each application or container communicates with the others only through their defined APIs.



The container virtualization layer is extremely flexible and can scale up to meet rising demand for any of the services. Once you define what services are running in one or more containers, it's possible to move those containerized services between on-premise machines and the public cloud. This allows you to more easily scale services locally at your regional studio or in a cloud provider such as AWS or Azure.

And unlike with virtualization, there is no extra supervisory overhead to contend for resources, and containerization platforms are even able to run on top of virtualization platforms.

This last point is critical for long-term planning, since Amazon, Microsoft and other public cloud providers are

already running hypervisor software on their cloud instances. Attempting to add your own VM hypervisor on top of a cloud provider's machine instance may work poorly, or not at all. Containers, by contrast, work well on just about all the cloud providers and instance types. Most providers even offer tools to make it easy to manage and coordinate your containers running in their cloud.

Fundamentally, containerization and virtualization are two different ways of doing the same thing. Having several containers running separate services pushed up to a cloud won't solve issues such as communication latency over the internet, but it will offer some added security protection and flexibility, and let you allocate resources more efficiently, which is the point of the cloud, after all.