

WHITE PAPER



VIRTUALIZATION
WITH
ZAPPCORE

 appiyo

Do we do it with Hypervisors or do we do it with Containers?

Internet-scale becomes a non-negotiable attribute for cloud based solutions, which turns the spotlight on **elastic computing** and security, not to overlook costs. This brings **virtualization** right into the Centre of technology considerations.

First, let us define virtualization. It is an apparent computing environment [apparent - seeming real or true, but not necessarily so] provided on a hardware. The natural question that follows this definition is 'What is the motive?' Fundamentally, it is to extract the maximum possible use of hardware by allowing multiple computing environments to prevail. That's it. Of course, the corollaries are cost reduction and lesser administration overheads. Now that the motive is clear, implementation poses two choices of virtualization – one which makes use of Containers and the other which makes use of Hypervisors. It is but natural for technology people to be drawn into a debate about which one is better. And there are many views, some even predicting the death of Hypervisor based virtualization! [Sample this: <http://www.forbes.com/sites/benkepess/2013/12/10/do-containers-mark-the-death-knell-for-virtualization/>].



Speed versus Choice

What gives will depend on where the principal player comes from

Such debates pitch the protagonists of Container based virtualization arguing that an additional layer of Hypervisor is an overhead dragging down performance, against the antagonists who view Hypervisor as an enabler of choice for the customer to operate different complete Operating Systems.

Let us conclude this thought here with the following three observations:

1



If the principal player is a cloud or datacenter service provider, he may go with Hypervisor based virtualization

2



If the principal player is a captive IT, they may lean towards Hypervisor based virtualization

3



If the principal player is a platform provider, the choice may be Container based virtualization

Let us come back to our question which we asked first.

- If we do not need many different complete OS to run on, then Containers are a preferred choice;
- High concentration of Containers in a single node is possible – typically, in the order of hundreds;
- Containers can be extremely light-weight, efficient and can secure processes; Any application can be packaged and run as light-weight Containers;
- Elastic resource allocation - resources can be added any time without any shutdown or volume resizing (some services might need to be restarted to consider new resources);
- Minimizes license and other costs.

Instead of trying to run an entire guest OS, container virtualization isolates the guests, but doesn't try to virtualize the hardware. Instead, you have containers (hence the name) for each virtual environment. The kernel provides process isolation and performs resource management. This means that all virtual machines have their own file system, processes, memory, devices, etc. Container-based solutions should be able to address as many CPUs and as much RAM as the host kernel.

**For us, the jury is out
and the debate ends.**

**Appiyo uses
Container based
virtualization. We
call them
"Zappcore"**

Reference

For a detailed treatment on Containerization in High Performance Computing (HPC) environment, refer <http://marceloneves.org/papers/pdp2013-containers.pdf>;

Containers, not Virtual Machines, are the future cloud:

<http://www.linuxjournal.com/content/containers%E2%80%94not-virtual-machines%E2%80%94are-future-cloud>