

Realtime
publishers

The Essentials Series:
Containers Virtualization for Linux

What is Containers Virtualization?

sponsored by



by Greg Shields

What Is Containers Virtualization?.....	1
All About the Differences	2
Containers Enjoy Unique Benefits.....	3
Multiple Distribution Support.....	4
Containers Virtualization Is Fast Virtualization.....	4

Copyright Statement

© 2009 Realtime Publishers, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers, Inc or its web site sponsors. In no event shall Realtime Publishers, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at info@realtimepublishers.com.

What Is Containers Virtualization?

In the minds of many IT professionals, the idea of virtualization conjures up thoughts of hypervisors and the consolidation of operating systems (OSs) onto virtual hosts. For many, virtualization embodies technologies that abstract OSs from their hardware. Physical resources are emulated in one fashion or another to isolate virtual machines from the hosts they reside upon. At the same time, those virtual hosts provide resources for the processing of virtual machine needs. Requests for resources are proxied through that hypervisor layer, incurring an overhead cost in the translation from virtual to physical and back again.

But this definition of virtualization is only one possibility, designed to support OS heterogeneity at the same time as server consolidation. The type of virtualization described here is commonly called *Hardware Virtualization*. This architecture effectively decouples a virtual workload from the technology used to virtualize it. With Hardware Virtualization, the “layer of abstraction” between the virtual and the physical is considered to exist “below” the level of the virtualized OS.

Now consider how the environment changes when that layer of abstraction moves from a position “beneath” the OS to a new location “within” it. In this configuration, the OS that is installed to the physical hardware along with OS templates that reside in that core OS become a core part of the virtual workloads that are hosted on top. As you can see in Figure 1, with the “layer” existing both atop and within a full instance of an OS, the files and configurations that make up that instance become the basis by which all virtual machines are based.

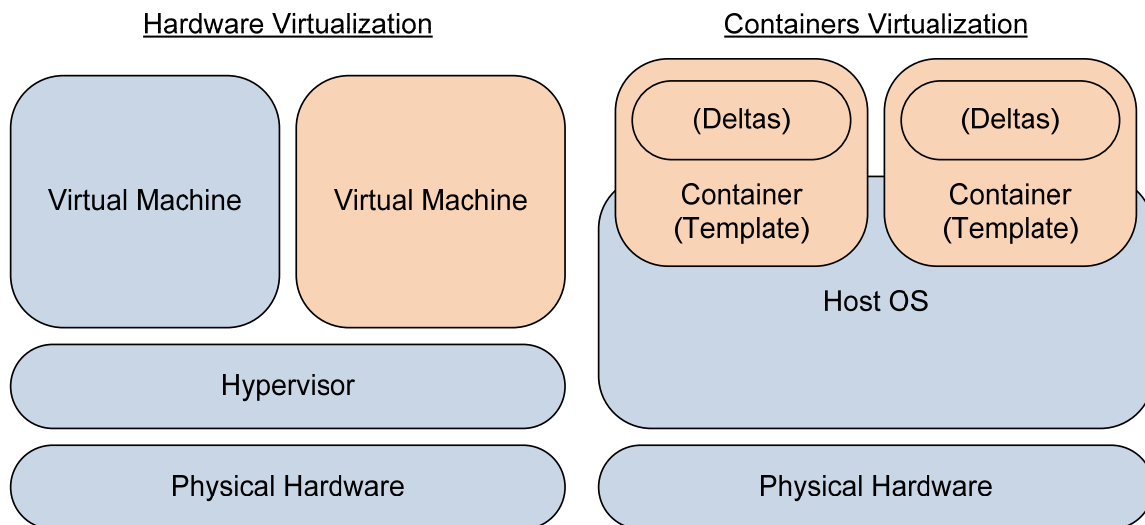


Figure 1: Shown in red, Hardware Virtualization on the left forces each virtual machine to be fully atomic. Containers Virtualization for Linux merges host OS templates with the individual personality of each container.

All About the Differences

The architecture being described here represents another way to virtualize called *Containers Virtualization*. With Containers Virtualization, each individual virtual workload or “container” is made up of two portions. The first is involved with the OS template housed within the core OS. This is shown in red on the right side of Figure 1. The second portion is comprised of the differences between that template and the individual configuration of the container itself. The ultimate result is that OS virtualization for Linux enables multiple environments to run simultaneously from the same core OS.

To illustrate, think for a second about the configuration of two sample Linux servers. In this example, these two servers have been freshly installed with the same packages and the same configuration. The only differences between these two newly created computers are their names. In this situation, how different is the configuration between these two computers? Arguably, those differences can be little more than a few characters in a few files.

With Hardware Virtualization, the hosting of two servers such as these requires the creation of two entirely segregated virtual machines. Each virtual machine requires its own separate installation and configuration, and no connection exists between the two computers. Separate hard drive space, buffers, cache, and other hardware resources must be carved out of the virtual host for the processing of each virtual machine.

In contrast, when virtualized with Containers Virtualization, the disks of these two sample computers are made up of the differences between each computer and the template. Buffers, cache, and hardware resources need not be emulated because each collocated virtual machine is in fact a combination of the template and its individual deltas. Hard drive space is conserved because the personalization that requires storage is made up of only those differences.

Containers Enjoy Unique Benefits

Because of this mechanism, Containers Virtualization enjoys a set of benefits not seen by Hardware Virtualization architectures. These benefits align with improved performance, scalability, and density:

- *Performance.* With Hardware Virtualization, devices on a virtual machine require special drivers that integrate with its hypervisor layer. These drivers receive virtual machine resource requests and translate them into calls that can ultimately be processed by the real device driver for the hardware. Whether using emulated or synthetic drivers, this translation process involves a resource cost that impacts virtual machine performance. With Containers Virtualization, no emulated or synthetic drivers are used. Each container makes use of the real driver installed to the system, which eliminates the overhead cost and improves performance across all collocated containers.
- *Scalability.* The same emulation activity that impacts driver usage in Hardware Virtualization also has a tendency to limit the scaling of virtual machines. An emulated driver is by nature developed with a hard-coded upper limit on resources that it can support. This is a function of the emulation process itself. Containers Virtualization's use of real resource calls—rather than emulated or synthetic—means that any container can scale to the full resources of the host server. This concept of scalability works in another way as well. With Containers Virtualization, because resources are not locked-in at each virtual machine's boot time, assigned resources such as processor and memory can be adjusted on the fly. This enables administrators to rapidly adjust assigned resources as necessary to meet workload demands.
- *Density.* With the increase in performance gained through low virtualization overhead, more simultaneous virtual machines can be collocated on a single server. This provides gains in either of two ways: It can mean that fewer physical resources are needed for the processing of an individual container, or it can mean that certain high-use workloads are now candidates for virtualization that would otherwise not be. Environments that leverage Containers Virtualization can balance the requirements of high-use workloads with the desire for high levels of consolidation.

Multiple Distribution Support

Specific to Linux and its various distributions, there are other unique benefits that can be seen in environments that leverage Containers Virtualization. These benefits relate to the function of how a Linux container's configuration is based on the combination of its configuration and the files installed to the host.

Consider the situation where your IT environment requires the use of multiple distributions for various processing needs. Although each individual distribution has unique characteristics and areas where it excels, many are based on a common kernel. That common kernel is equivalent to the Host OS on the right side of Figure 1. It provides a central location where multiple, simultaneous distributions can be virtualized and collocated atop a single host.

As an example, using Containers Virtualization, one or more CentOS instances can process their workloads at the same time and on the same host as one or more Red Hat or SuSE instances. As before, the individual differences between the distribution and the core template are what make up the container.

Making this capability even more compelling is the concept of simultaneous patching. With the file composition of each container being first based on the core OS installation, any patch or update to that core OS has the ability to automatically and immediately update the configuration of every collocated container on the same host. For environments that may support dozens or hundreds of containers per host, this ability to quickly update large numbers of virtual workloads with a small number of actions reduces the operations costs while improving IT's level of response time.

Containers Virtualization Is Fast Virtualization

All these capabilities add up to make Containers Virtualization a compelling add-on for Linux environments that need to support high performance or high virtual workload density. These improvements come as a part of Containers Virtualization's relocation of the layer of abstraction away from its traditional location directly on top of physical hardware. Yet this virtualization architecture also provides benefits in the realm of systems management. The next article in this series will discuss those changes to how virtual workloads are managed and talk about differences in the management toolsets commonly seen between the different classes of virtualization today.