13.1.1 Cloud and Virtualization Overview: Part 1

Click one of the buttons to take you to that part of the video.

Cloud and Virtualization Overview: Part 1 0:00-0:51

If you look online, you'll find cartoons where a young child asks, "What are clouds made of?" and his parent replies, "Linux servers, mostly."

The truth is, a lot of the IT services available in the cloud are provided by servers that run a Linux operating system. To provide these services, Linux might be running on bare metal hardware. It might be a host for virtual machines. Or it might run as a virtual machine. The services provided by these servers can vary from a specific application to a single server dedicated to a single customer to a whole enterprise infrastructure that includes a computing framework and a network framework.

As a Linux administrator, you should be able to describe cloud and virtualization concepts and compare different cloud and virtualization technologies.

In this video, we're going to talk about virtualization. We'll cover cloud concepts and technologies in another video.

Traditional Computing Model 0:52-1:49

If you have ever purchased your own desktop or laptop computer, you understand the traditional computing model. Your computer runs an operating system, like Linux or Windows. Your OS interacts with the computer hardware, takes care of the disks and file system, and does other housekeeping tasks. The software applications installed on the computer let you perform your day-to-day work or provide entertainment. The key idea is that a single operating system has full control over all the hardware resources in the computer.

While you may enjoy the lightning speed of your new computer, this computing model is inefficient. Unless you're using some hardware-intensive applications, such as a video editor, it's likely that much of your computer's hardware sits idle, even when you're multitasking between several applications. To further illustrate this inefficiency problem, suppose you purchase some costly servers. You wouldn't want these servers to sit idly when they could be doing more computing work, offering more services, and responding a larger number of users.

Virtualization 1:50-2:18

One technology that you can use to improve efficiency, especially on a server, is virtualization. With virtualization, you can pool multiple operating systems on the same physical hardware. Each operating system runs independently and runs its own set of applications. These independent operating systems are called virtual machines, or VMs. You'll also hear them called guest operating systems. When you run multiple VMs, the demand for hardware resources increases, resulting in better hardware utilization.

Type 1 Hypervisor 2:19-2:39

You can implement virtualization using special software called a hypervisor. Hypervisors come in two types. This image shows Type 1 hypervisor. It's often called a native or bare-metal hypervisor. It runs directly on the computer's hardware to manage the guest VMs. If you're familiar with Microsoft Hyper-V or VMware ESX, both are Type 1 hypervisors.

Type 2 Hypervisor 2:40-3:05

A Type 2 hypervisor is a software application that runs on top of a conventional operating system to manage the guest VMs. When a guest VM needs to interact with the host hardware, it must do so through the hypervisor and then through the host operating system. VMware Workstation, Oracle VM VirtualBox, Parallels Desktop for Mac, and the open source QEMU product are Type 2 hypervisors.

Linux-Based Hypervisors 3:06-3:40

Here's where it gets tricky. Two of the popular Linux-based hypervisors, Xen and KVM, are advertised as Type 1 hypervisors. However, both are installed on a normal Linux distribution, and both tap into the open-source QEMU emulation features. And when you reboot them after installation, they both have the look and feel of a Linux operating system. In reality, both Xen and KVM modify the Linux kernel so that,

while they have the look and feel of Linux, the hypervisor functionality is part of the kernel. This kernel manages guest VMs and interfaces directly with the host hardware, making them Type 1 hypervisors.

Types of Virtualization 3:41-5:04

All VMs are not virtualized in the same in the way. There are three types of virtualization environments: full virtualization, paravirtualization, and hardware-assisted virtualization. In full virtualization, the hypervisor presents a virtual hardware environment, which emulates a physical hardware environment. The guest operating system interacts with this virtual hardware as if it's physical hardware. It doesn't see any difference. Full virtualization provides the best isolation and security for VMs, but it comes at a cost: it takes extra time and processing power to provide virtual hardware emulation.

Paravirtualization improves performance by allowing the guest operating system to communicate directly with some of the host's physical hardware. You can implement paravirtualization by modifying the guest operating system with special device drivers. Both Xen and KVM offer recompiled paravirtual guest operating images that are more efficient than a fully virtualized equivalent.

Hardware vendors have developed features to make virtualization even more efficient. Various features allow the hardware to handle some of the virtualization processing the hypervisor would normally perform. Application requests for hardware resources can be sent directly to the hardware layer instead of being translated through both the guest operating system and the hypervisor. Hardware-assisted virtualization greatly improves a full virtualization environment.

Container Virtualization 5:05-6:04

There's another category of virtualization that has gained popularity recently, container virtualization. You may have noticed that each virtual machine requires a guest operating system. This requirement keeps each VM isolated and prevents interactions between applications running on different virtual machines. These multiple operating systems not only introduce computational overhead, but also require you to purchase additional licenses.

A better solution to this problem is container virtualization. Containers are defined as consistent runtime environments. You create containers using special software called a container engine. In each container, you install an application and all of its dependencies, including libraries, other binaries, and configuration files that are required to run the application. The benefit of this container is that the application is isolated from other applications running in other containers. This is good for security and portability. A container can be saved and easily ported to other container environments.

Containers vs. Virtual Machines 6:05-7:15

If you compare containers with virtual machines, you'll see that each has its benefits and shortcomings. First, containers are more lightweight and use fewer resources than virtual machines. Virtual machines are portable between different hypervisors; containers are only portable between different families of operating systems. Generally, you can save a virtual machine running in one hypervisor environment and run it in another hypervisor environment, even if the two hypervisors are from different software vendors. Conversely, an application that's containerized in a Windows environment can't be run by a Linux container engine.

A container may only occupy a few megabytes, whereas a virtual machine with its own operating system might take up several gigabytes. That means a single server can host many more containers than virtual machines. Also, a virtual machine may take several minutes to boot, while a containerized application only takes a few seconds to initialize. It's common for a container to be instantiated at the last second and then removed when it's no longer needed.

Containers also make it easy to modularize applications. You can create containers with simple applications that are built into a larger system. Then you can change one application without rebuilding the entire system.

Summary 7:16-7:43

So, let's review. In this video, we described virtualization and how virtual machines are implemented by a hypervisor. We identified two types of hypervisors that manage virtual machines. A type 1 hypervisor runs on bare metal hardware. A type 2 hypervisor runs on an operating system. We discussed the three types of virtualization: full virtualization, paravirtualization, and hardware-assisted virtualization. We ended by illustrating container virtualization and compared containers with virtual machines.
