

Virtualization

Virtualization is a fancy word, but what it means is that you can encapsulate entire operating systems or parts thereof in a bubble of virtual hardware and software so that it appears like a complete, separate and real physical host to the software contained inside this bubble. Think Matrix.

Virtualization has two dimensions: interaction between the physical host and the virtual host and the interaction between the virtualized operating system and the underlying hardware.

Virtualization software is normally run as an application or a service on top of an operating system, like Windows, Linux, Mac OSX, etc. In technical terms, the physical machine is referred to as the host. Inside the virtual software, an instance of an operating system is executed, which can be similar or identical to the one powering the host. In technical terms, the virtualized environment is known as the virtual machine and the operating system running inside as the guest.

Furthermore, there are different ways the virtualization is brought to realization. There's simple emulation, where hardware and software calls are passed through a translation layer. There's para-virtualization, where parts of the activity inside virtual machines are passed onto real hardware while others are emulated. There's operating system level virtualization, where you boot into a dedicated kernel that permits only same instances of the operating system to run.

Some of these methods can be performed on the fly, without significant changes to the host and its operating system. Others require that you reboot your host into a dedicated instance of your operating system that supports virtualization. Others yet require a complete machine to be used for virtualization only, also known as bare-metal virtualization.

Virtualization software that manages the creation and running of virtual machines, the allocation of resources and the containment is often known as the hypervisor. Some virtualization products can also use the CPU technology virtualization extensions to boost the performance of virtual machines. This is known as hardware-assisted virtualization and make use of technologies such as VT-x (Intel) and AMD-V (AMD).

What is not virtualization?

Some people like to bundle any software that creates an abstraction layer between the operating system and some of the running processes as virtualization. For instance, there's Sanboxie, which can isolate your browsers in discardable containers. Some programs let you freeze your system, so it cannot be modified. Others also allow the use of a so-called shadow mode, where you run normally, however all and any changes are discarded on reboot.

All of these programs have their merits, but they are not considered virtualization technologies, because they do not simulate system calls, nor they allow running complete new instances of

other operating systems on top of your host; they only create additional separation layers, mainly for security. Speaking of security ...

Why should you use virtualization?

If security comes first and foremost to your mind - wait just there. We will discuss security in a separate article. But I do want to emphasize that virtualization was not primarily designed for security. It's mainly used for testing, consolidation of hardware and cost reduction, running legacy software, and education. Security is a nice bonus, but there are caveats, which we will elaborate separately.

What do you need to run virtualization technologies?

You will need a physical host of some kind. Depending on the type of virtualization software you choose, it will allow running virtualization software on top of your operating system alongside everything else - or it might require the hardware entirely to itself.

Since you are going to be running additional operating systems on top of your hardware, you will need additional resources, mostly CPU and RAM, to enable that. For example, if your machine has only 2GB RAM and it's already running Windows 7, spawning another instance of some operating system on top of it is going to be a close call, since you will have to dedicate some portion of your memory to the virtual machine. However, if you have 16GB RAM, then you can comfortably spawn four virtual machines with 2GB RAM each, which still leaves you with some 8GB RAM for your host.

Likewise, disk space and disk performance and CPU power are also an important factor. For example, running a MySQL database virtual machine from a 5,400rpm laptop disk is probably not a good idea. Or if you only have one CPU core and you plan on multimedia processing.

Ample resources are important for fast and efficient virtualization. You also want modern processors that support virtualization extensions, so you can enjoy additional benefits, including better performance, but also 2D and 3D graphics and other fancy stuff.

Free software

Now, finally, let's take a look at a few of the available technologies. For each, I will write down a brief explanation, the difficulty in use and setup, the intended audience, pros and cons, and other important details. I will also link to some of my very lengthy tutorials that teach how to setup and use these programs.

VMware Player

VMware Player is a cross-platform solution. It runs as another program on your system. You can use it to create virtual machines and install operating systems as guests. You can also run virtual appliances - pre-created virtual machines with specific uses.

<http://www.vmware.com/products/player/>



Audience: new and intermediate users.

Pros: Simple to install and use.

Cons: Limited functionality; no snapshots or folder sharing.

VirtualBox

VirtualBox is a cross-platform technology, currently owned by Oracle, formerly Sun. VirtualBox is similar to VMware Player, but it has a wider subset of features, including a more advanced network stack, unlimited snapshots, some OpenGL and DirectX support, and more. It is quite simple to install and use. You can also use command line for automated deployments. One sore thing is that you cannot take screenshots of your guests. Disk management is somewhat convoluted. You can install Guest Additions to gain USB 2.0 support, shared folders and to boost performance. VirtualBox also has a portable edition.

<http://www.virtualbox.org/>



Audience: new and intermediate users.

Pros: Simple to install and use, lots of features.

Cons: No screenshots, importing existing machines is difficult, non-intuitive disk management.

You may also want to read:

[How to install VirtualBox Guest Additions](#)

<http://www.dedoimedo.com/computers/virtualbox-guest-addons.html>

[VirtualBox networking & sharing tutorial](#)

<http://www.dedoimedo.com/computers/virtualbox-network-sharing.html>

VMware ESXi

ESXi is a bare-metal hypervisor with a reduced subset of features compared to ESX. It requires its own host to run and has a locked-down console that you can manually enable. You cannot take screenshots or movies of your virtual machines and migration/cloning is by hand only. You do get memory sharing for improved RAM utilization, a powerful monitoring and management, and command line with ssh, when unlocked. You can install VMware Tools to boost the performance of virtual machines. In some cases, the installation is automatic. Para-virtualization mode is also supported.

<http://www.vmware.com/products/vsphere-hypervisor/overview.html>



Audience: advanced users or professionals.

Pros: Powerful, extended virtual machine options.

Cons: Dedicated hardware, lots of resources, not easy to setup or run.

You may also want to read:

[How to clone virtual machines in ESXi](#)

<http://www.dedoimedo.com/computers/vmware-esxi-clone-machine.html>

[VMware ESXi tips and tricks](#)

<http://www.dedoimedo.com/computers/vmware-esxi-tips-tricks.html>

Other solutions

There are many other solutions, which I have not listed here, but we will see them in the future, like for instance Parallels Virtuozzo and OpenVZ and VirtualBox-based VMLite. Several image building and remastering solutions also exist, which sort of lean into the gray area between virtualization and thin clients. Linux also allows spawning operating system containers directly on modern kernels. And let's not forget cloud technology and cloud-like virtualization. We will talk about all these in the future, too.