A hypervisor is a process that separates a computer’s operating system and applications from the underlying physical hardware. Usually done as software although embedded hypervisors can be created for things like mobile devices.

The hypervisor drives the concept of virtualization by allowing the physical host machine to operate multiple virtual machines as guests to help maximize the effective use of computing resources such as memory, network bandwidth and CPU cycles.

Containers vs. hypervisors

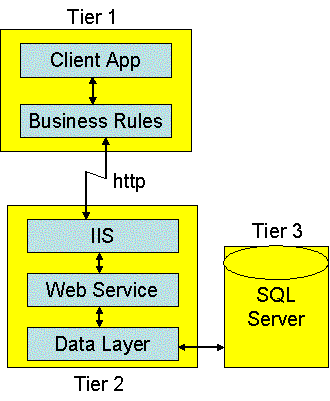
In recent years, container technology has grown in popularity as a possible replacement for hypervisors, as they can place more applications into a single physical server than a virtual machine can.

VMs take up a lot of system resources. Each VM runs not just a full copy of an operating system, but a virtual copy of all the hardware that the operating system needs to run. This quickly adds up to a lot of RAM and CPU cycles. In contrast, all that a container requires is enough of an operating system, supporting programs and libraries, and system resources to run a specific program,” says Steven J. Vaughan-Nichols in this 2016 [Network World article](https://www.networkworld.com/article/3068392/cloud-storage/containers-vs-virtual-machines-how-to-tell-which-is-the-right-choice-for-your-enterprise.html).

Security concerns and practical uses of VMs, however, mean that containers won’t necessarily replace hypervisors/VMs, but rather companies will use a combination of both, according to Vaughan-Nichols. On the security issue, some feel that containers are less secure than hypervisors, due to containers only having one OS that applications share, while VMs isolate not only the application, but the OS as well. If an application gets compromised, it could attack the single OS in a container, affecting other applications. If an application in a VM gets compromised, only one OS on that server would be affected, not other applications or OSes on the VM. Marvin Waschke writes about this issue in this [InfoWorld blog post](https://www.infoworld.com/article/3197444/cloud-computing/are-vms-more-secure-than-containers.html).

Virtual Machine and Container :

Both Containers and Virtual Machines have been around for quite some time. In very high level terms, a Virtual Machine or Hypervisor (such as [VMWare](http://vmware.com/), [Hyper-V](http://www.microsoft.com/en-ca/server-cloud/solutions/virtualization.aspx), [KVM](http://www.linux-kvm.org/page/Main_Page), [VirtualBox](http://www.virtualbox.org/), and [Xen](http://www.xenproject.org/)) is designed to emulate an entire physical computer including the various hardware abstraction required for networking, video, audio, etc...Container on the other hand ( [Docker](http://www.docker.com/), [Parallels](http://www.parallels.com/) , [CoreOS](http://en.wikipedia.org/wiki/CoreOS),[chroot](https://help.ubuntu.com/community/BasicChroot), ...)  runs on top of an existing kernel, leveraging resources form the kernel, and merely presents a virtual [userspace](http://en.wikipedia.org/wiki/User_space" \t "_blank) with separate filesystem, CPU, memory and protected processes.    
  
Without having to emulate the underlying hardware, you can pack 3-4 times as many containers into the same resource pool as a single Virtual Machine.  
  
  
So why would I use Virtual Machines, if Containers are just as good?  Well, because a Virtual Machine abstracts the ENTIRE hardware platform, there's evidence that it is better suited to defined network segregation.  

[](https://i-msdn.sec.s-msft.com/dynimg/IC36655.gif)You could, for instance, define a Virtual Machine to represent your web application in it's entirety, then within that VM, create containers for the[web, app, and database tiers](http://en.wikipedia.org/wiki/Multitier_architecture).  The containers would provide logical segregation between the tiers, and the VM would protect the entire application from other apps in the DMZ.

Virtual Machines also allow you to run completely different Operating Systems simultaneously on the same hardware.  For instance, on your [Ubuntu](http://ubuntu.com/) Laptop, you could use [Virtualbox](http://virtualbox.org/" \t "_blank), to simultaneously run [Windows 8.1](http://windows.microsoft.com/en-ca/windows-8/whats-new) and [OSX](https://www.apple.com/ca/osx).      
  
Or, on your server, you could simultaneously run [Redhat Linux](http://www.redhat.com/promo/rhev3/sysreq.html" \t "_blank), [Windows Server 2008](https://technet.microsoft.com/en-ca/windowsserver/bb414778.aspx), and [Windows Server 2012](https://technet.microsoft.com/en-ca/library/dn303418.aspx).     
  
A containerized system, as mentioned above, runs all containers off of the same Operating System Kernel.  
  
And by far the biggest benefit of Containers over Virtual Machines is speed of launch. A Virtual Machine is, for all intents and purposes, a complete computer Operating System.  On boot, it has to run through all of the legacy boot processes...

Mainframe virtualization:

Now, virtualization is primarily about disconnecting the application (and infrastructure software) from the hardware platform -- and, paradoxically, this has made the mainframe a more valuable and more applicable platform than ever.

From its inception, "virtual" computing has had two distinct meanings. At the hardware level, virtual memory or virtual disk allows systems to deliver the speed of a faster, higher level of storage and the capacity of a lower level of storage. Above the hardware level, a virtual machine (VM) allows one system to split into multiple "machines," each with its own operating system (OS). Thus, for example, a System z9 can contain a z/OS VM and a Linux VM, each operating as if it was a separate system. When vendors talk about virtualization today, most of the time they are talking about the second definition. n the mainframe's case, the reason for lower (system) administrative costs is the relative simplicity of the mainframe environment:

- It can run many applications on many virtual machines.

- The same administrative tool applies to all applications within a single mainframe, and

- No network management or systems management costs apply.

The core of IBM's mainframe virtualization offering is z/VM -- the virtual machine OS from which other virtualization solutions, such as VMWare (now owned by EMC Corp.), sprang. Z/VM's Virtual Machine Resource Manager provides not only automatic allocation of system resources between virtual machines but also user control over resource limits, prioritization and SLA (service level agreement) support. At present, z/VM can support thousands of z/OS, z/OS.e, TPF and Linux virtual machines -- all operating on the same single mainframe.

This allows intermachine communication at the speed of placing a message in main memory and picking it up from there and also adds security, since such messages never leave the mainframe. In addition, IBM's security suite protects the data within the mainframe and within each virtual machine.

It is also worth noting that IBM mainframes have been offering virtual memory for longer than most if not all other platforms, and IBM also provides a virtual tape library solution.

Virtualization has always been a strength of the mainframe, but with enterprises increasingly facing the costs of x86-based server farms, mainframe virtualization via virtual machines is more attractive than ever. Its benefits include simplification of the enterprise architecture, lower total cost of ownership, greater scalability and robustness, better security, and more flexibility to optimize the overall enterprise architecture. While the mainframe does not yet support Windows VMs, enough Windows and Unix firepower has been added to Linux to make most applications on any platform easy to move into a mainframe Linux VM.

https://searchdatacenter.techtarget.com/tip/Mainframe-virtualization-Old-friends-are-often-best