**VIRTUAL MACHINE**

A virtual machine (VM) is a computing environment that functions as an isolated system with its own CPU, memory, network interface, and storage, created from a pool of hardware resources. Software called a hypervisor isolates the necessary computing resources and enables the creation and management of VMs.

The physical machine that runs the VMs is called the host machine, host computer, host operating system, or simply host. The many VMs that use its resources are guest machines, guest computers, guest operating systems, or simply guests. The hypervisor treats compute resources—like CPU, memory, and storage—as a pool of resources that can easily be relocated between existing guests or to new virtual machines.

VMs allow multiple different operating systems to run simultaneously on a single computer—like running a Linux on a MacOS or Windows system. Each operating system runs in the same way an operating system or application normally would on the host hardware, so the end user experience emulated within the VM is nearly identical to a real-time operating system experience running on a physical machine.

**How does a hypervisor work?**

A VM can be defined in a single data file, which can be moved from one computer to another, opened in either one, or be expected to work the same. A hypervisor manages the hardware and separates the physical resources from the virtual environments. Resources are partitioned as needed from the physical environment to the VMs. The physical hardware still does the execution, so the CPU is still executing CPU instructions as requested by the VMs, for example, while the hypervisor manages the schedule.

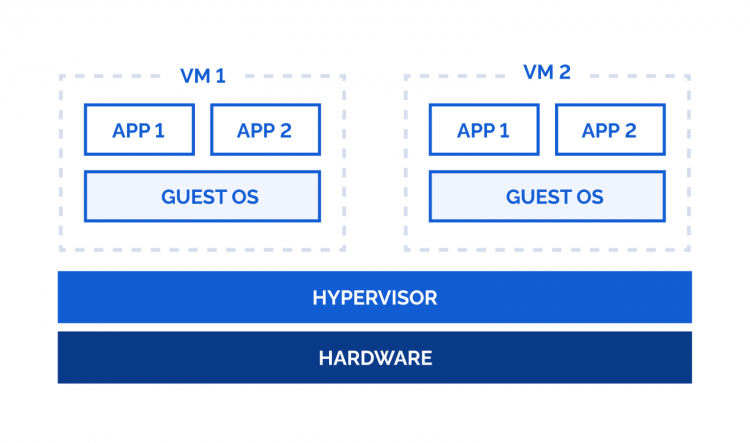
When the VM is running and a user or program issues an instruction that requires additional resources from the physical environment, the hypervisor schedules the request to the physical system’s resources so that the virtual machine’s operating system and applications can access the shared pool of physical resources.

In Linux® environments, the built-in hypervisor is called the Kernal Based Virtual Machine. Other options include Xen, which is open source, and Microsoft Hyper-V.

There are 2 different types of hypervisors that can be used for virtualization.

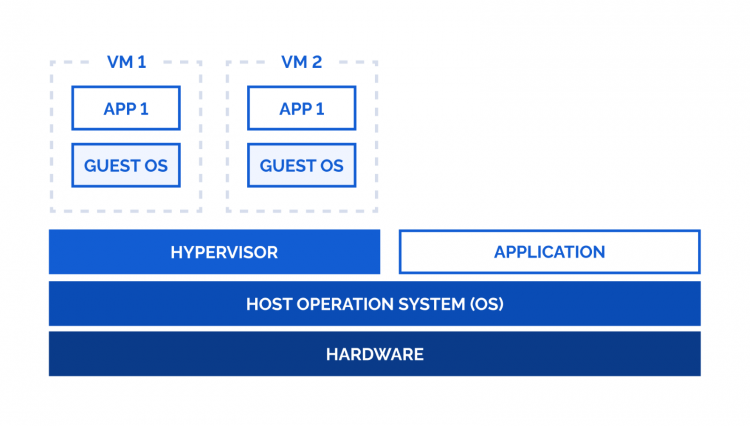
**Type 1**

A type 1 hypervisor is on bare metal. VM resources are scheduled directly to the hardware by the hypervisor. KVM is an example of a type 1 hypervisor.



**Type 2**

A type 2 hypervisor is hosted. VM resources are scheduled against a host operating system, which is then executed against the hardware. VMware Workstation and Oracle VirtualBox are examples of type 2 hypervisors.



**Advantages of VMs**

With virtualization, multiple operating systems can run alongside each other and share the same virtualized hardware resources. Without virtualization, you can only run 1 operating system on the hardware.

Server consolidation is a top reason to use VMs. Most operating system and application deployments only use a small amount of the physical resources available when deployed to bare metal. By virtualizing your servers, you can place many virtual servers onto each physical server to improve hardware utilization.

This keeps you from needing to purchase additional physical resources, like hard drives or hard disks, as well as reducing the need for power, space, and cooling in the datacentre. VMs provide additional disaster recovery options by enabling failover and redundancy that could previously only be achieved through additional hardware.

A VM provides an environment that is isolated from the rest of a system, so whatever is running inside a VM won’t interfere with anything else running on the host hardware.

Because VMs are isolated, they are a good option for testing new applications or setting up a production environment. You can also run a single purpose VM to support a specific process.

**How VMs relate to the cloud**

Virtualization is one of the technologies that makes cloud computing possible. Public and private clouds virtualize resources into shared pools, add a layer of administrative control, and deliver those resources with automated self-service functions.

The virtualization, management, and automation software that creates clouds all sit on top of the operating system, which maintains the connections among physical resources, virtual data pools, management software, automation scripts, and customers.