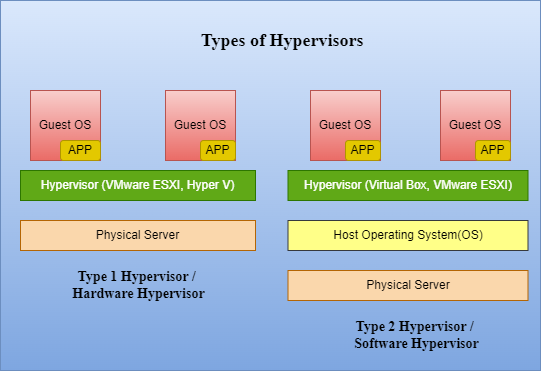
**Containers**

**1.1 Virtualization with Hypervisor**

Hypervisor is the software used to create Virtual machines/servers on top of Physical servers/ physical devices. Hypervisor is installed on a Physical device which in turn helps to create a Virtual Machine. Hypervisors, Virtual Machines, and Physical devices are interlinked together to implement the concept of virtualization.

Basically, Hypervisors are of two types  
1) Type 1 Hypervisor which is also called Hardware hypervisor  
2) Type 2 Hypervisor which is also called Software hypervisor



**Type 1 Hypervisor :**  
The Type1 Hypervisor is directly installed on the Hardware/ Physical device/Bare metal server. It doesn’t need the help of any operating system for installation. Once the Hypervisor is installed on the hardware, we can create a Virtual Machine using the installed Hypervisor. As we are creating a machine/server using software(Hypervisor) and it is virtual, we call it Virtual Machine. There are many Type1 Hypervisors available such as VMWare ESXi, Hyper V, and Xen.   
The Virtual Machine (VM) to be created in Hypervisor can have any Operation System (Guest OS) such as Windows, MAC, or Linux irrespective of the Host/Base Operating System (OS present in the Physical Machine). Any required application can be installed in the VM created.

**Type 2 Hypervisor :**

The Type2 Hypervisor needs to be installed on an Operating System (Windows/Linux/MAC). Without an OS, Type2 Hypervisor can’t be installed. So, in Type2 Hypervisor, the first layer is Physical Hardware, the second layer is Operation System, and the third layer is the preferred Type2 Hypervisor. There are many Type2 Hypervisors available such as Virtual Box, and VMWare Workstation. The Virtual Machine to be created using Type 2 Hypervisor can have any Operating System (Guest OS) irrespective of the Host/Base Operating System.   
The Disadvantage of using Type2 Hypervisor is – if there is any issue with the Base OS, the VMs in the Hypervisor and the Hypervisor both become inaccessible.

Type2 Hypervisors are used for local systems, R&D, POC, Testing, and replication of production. But, In Production (real-time), Type1 Hypervisors are used.

**1.2 Limitations of Virtual Machine Servers over Containers**

Let’s understand the cons of Virtual Machines and the need to opt for containers.

**Time took:** The time consumed to spin up/provision/create a Virtual Machine is much higher than creating a container. Container creation takes less than a minute based on the configuration specified.

**Resource (CPU, Memory) Allocation:** The Memory & CPU allocation is not dynamic for Virtual Machines i.e., Virtual Machines can’t allocate resources for themselves. The below example helps you to understand it clearly -  
Assume there is an application (App1) or a workload running in a Virtual Machine (VM1). The configuration of VM1 is 1 core CPU & 2GB RAM. If App1 is used by fewer people, then it will work fine in VM1. But if the no.of users increases for App1, then with VM1 configuration, the App1 doesn’t run fast. Hence, comes the necessity to increase the resource configuration of VM1, which should be done manually through the dashboard/CLI. Either we can turn off the running VM and allocate (increase/decrease) the resource size or without turning the VM, we can increase/decrease the resource size.   
Technically, the process of scaling in (decreasing the size) & Scale-out (increasing the size) of the resources should be done using scripts or commands (manually) in Virtual Machines. But Containers can resize their resources automatically.

**Resource utilization by OS:** The resources (CPU, Memory) allocated to the VM are not completely used by the Application running in it. As every VM has an Operating System that runs with some libraries, binaries, packages, and services which consume CPU, Memory, and Storage. So a part of the resources allocated to the VM is used by Operation System and the rest are used by the application running in it. But this limitation can be overcome using the container concept because containers don’t have kernel packages of the Operating System. They only have libraries and binaries of OS.

**No.of Machines created:** Compared to Physical servers, we can create more no.of Virtual Machines with different OS in each VM and run different/same applications in each VM. The no.of VMs that can be created in a Hypervisor is limited by the Physical device resources.   
Compared to VMs, the no.of VMs that can be created on a Physical device is less than the no.of containers that can be created as the container size is very less.

**1.3 Containers - Introduction**

**Sunitha comments – What is Container definition—**

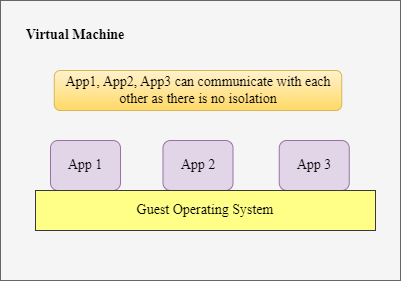
**1)** Containers only have libraries & binaries which are required for OS & Application to run. They don’t have Kernel Packages in them which makes containers lightweight.

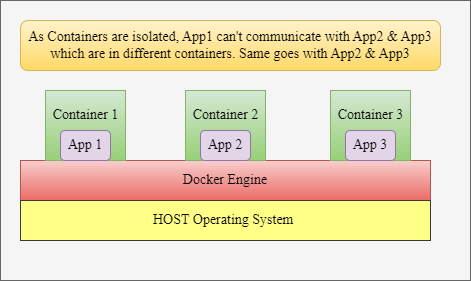
A kernel is the base of any Operating System used for communication between the Hardware and the Applications. Kernel converts the Machine Level Language to Human readable and vice versa. This kernel task is done by the Containerization tool for Containers.

**2)** In general, the application/s running in a Virtual Machine can communicate with each other using Process IDs as each application running is a process having a Process ID associated with it.

A container is also another simple process on your machine but is isolated from other processes i.eApplication running in one container can’t communicate with an Application running in another container though all the containers are running on the same Host.

Linux has a feature called Kernel Namespace & C groups which are also adopted by Docker (Containerization tool) for container isolation. So, the Process ID of the application running in one Container isn’t visible to any other Container running on the Docker Engine.



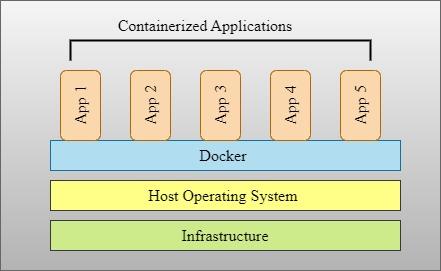


**3)** In Containers, the Operating System is virtualized whereas in Virtual Machines the hardware components of the Base Physical machine are virtualized by the Hypervisor.

**4)** Docker is one of the containerization tools which is portable and works well with all container capabilities. Docker makes the actions like pulling the images, creating a repository, pushing the images to a repository, and downloading the images easier compared to any other containerization tools. This makes the Docker tool easy to use.

**5)** Docker is installed on the Host Operating System. Without an Operating system, Docker can’t be installed. Using the installed Docker, containers are created. Docker can be installed on any Operating System (Windows, Linux, MAC).

**6)** Application running inside a container uses all the resources allocated to it as there are no kernel packages present in the container. This is the reason for containers to be light weighted and the provision time to be fast compared to bare metal servers and Virtual Machines.



**7)** The Container resource allocation is dynamic i.e. if the application running in a container and needs extra memory and CPU then the container immediately informs the docker about the requirement and allocates the resources based on the Hardware resources available. It doesn’t need any extra process/command to allocate the resources.

**8)** The above advantages of containers make running small-scale applications like cache servers and large-scale applications like Hadoop & Big data on containers easy and fast.

**Types of Containerization tools**

There are types of containerization tools – Open source & Enterprise

1. LXD/LXC - LXD (Daemon) is the latest version. LXD is only available for Linux Operating System and isn’t compatible with Windows or any other Operating System.
2. CRI-O
3. Rkt - rkt is an application container engine developed for modern production cloud-native environments
4. Containerd
5. Docker – There are two types of Docker – Community Edition (Docker Engine) & Enterprise Edition (Licensed)
6. Podman
7. Runc

All the tools mentioned above are open source except Docker which has both Enterprise and Community Edition. As they are open-source tools, they don’t need any license to be purchased. Just follow the official document of the tool and install it accordingly to create containers.