

## Assignment - 2

### 1. Comparison between hypervisor and docker

#### Differences between Hypervisor and Docker:

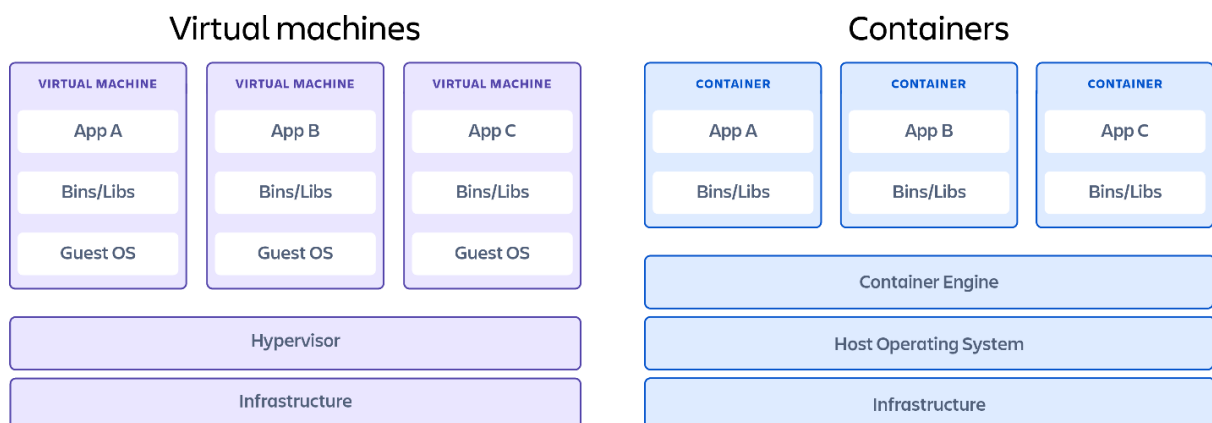
1. Functioning Mechanism
2. Number of Application Instances Supported
3. Memory Requirement
4. Boot Time
5. Architecture Structure
6. OS Support

Hypervisor	Docker
Hypervisors are of two types – the bare metal works directly on the hardware while type two hypervisor works on top of the operating system.	Docker, on the other hand, works on the host kernel itself. Hence, it does not allow the user to create multiple instances of operating systems.  Instead, they create containers that act as virtual application environments for the user to work on
A hypervisor allows the users to generate multiple instances of complete operating systems.	Dockers can run multiple applications or multiple instances of a single application. It does this with containers
Hypervisors enable users to run multiple instances of complete operating systems. This makes them resource hungry. They need dedicated resources for any instance among the shared hardware which the hypervisor allocates during boot.	Dockers, however, do not have any such requirements. One can create as many containers as needed. Based on the application requirement and availability of processing power, the Docker provides it to the containers.
A hypervisor might consume up to a minute to boot the OS and get up and running.	As Dockers do not require such resource allocations for creating containers, they can be created quickly to get started.  One of the primary reasons why the use of Dockers and containers is gaining traction is their capability to get started in seconds.

	Docker can create containers in seconds, and users can get started in no time.
Hypervisor though, has the host OS and then also has the guest OS further. This creates two layers of the OS that are running on the hardware.	Dockers help with the agile way of working. Within each container, different sections of the program can be developed and tested.
Hypervisors are OS agnostic. They can run across Windows, Mac, and Linux.	Dockers, on the other hand, are limited to Linux only. That, however, is not a deterrent for Dockers since Linux is a strong eco-system. Many major players are entering into the Dockers' fray

## 2. Comparison between Containers and Virtual machines?

Containers and virtual machines are very similar resource virtualization technologies. Virtualization is the process in which a system singular resource like RAM, CPU, Disk, or Networking can be 'virtualized' and represented as multiple resources. The key differentiator between containers and virtual machines is that virtual machines virtualize an entire machine down to the hardware layers and containers only virtualize software layers above the operating system level.



Virtual Machines(VMs)	Containers
VM is piece of software that allows you to install other software inside of it so you basically control it virtually as opposed to installing the software directly on the computer.	While a container is a software that allows different functionalities of an application independently.
While a container is a software that allows different functionalities of an application independently.	While applications running in a container environment share a single OS.
While applications running in a container environment share a single OS.	While containers virtualize the operating system only.
VM size is very large.	While the size of container is very light; i.e. a few megabytes.
VM takes minutes to run, due to large size	While containers take a few seconds to run.
VM uses a lot of system memory.	VM uses a lot of system memory.
VM is more secure.	While containers are less secure.
VM's are useful when we require all of OS resources to run various applications.	While containers are useful when we are required to maximise the running applications using minimal servers.
Examples of VM are: KVM, Xen, VMware.	While examples of containers are: RancherOS, PhotonOS, Containers by Docker.