

## ASSIGNMENT 2

### 1) Comparison between hypervisor and docker

Hypervisor:

**The hypervisor** allows users to run multiple instances of the entire operating system. This makes them resource-intensive. However They need dedicated resources for any of the shared hardware that the hypervisor allocates at startup time.

Docker:

**Docker** can create as many containers as you need. Depending on application requirements and available processing capabilities, Docker provides it for containers.

Docker	Hypervisor
Dockers work only on software of the operating system and not on the hardware side. It takes the host kernel and works on the principle of virtualization.	Hypervisors can be made to work on software and hardware where it works on the operating system or on the CPU and storage services of the system.
Docker does not allow user to create multiple instances of operating system in the same computer but it makes virtualization by making container in the same system. Containers help users to work separately on different or same applications. The same operations are carried out by containers in the system	In a single system, we can use multiple operating systems with the help of Hypervisor. This makes the system to work with multiple users with different methods even for the same program. Hence the same operation is done by different operating systems.

Resource requirement is low as containers are working on the same operating system and this makes the system to share resources within the containers.	More power and resources are required by the systems using hypervisors as different programs are being run on the same system with different operating systems.
Boot time is low for dockers as all the containers work on the same machine. User can start the system in seconds and can start working on the same machine.	Boot time is high for hypervisors as different operating systems are used. It may take some minutes to start the system and users can resume their work only after booting the machine.
If the same application needs to be tested in the system with different instances, we can use containers as different parameters can be given to the application in the same container and can be tested at the same time. Dockers support this method of working which is called agile model.	We cannot test the same application with different parameters in hypervisor as there is no container method available. This application needs to be developed and tested in the system. If the parameters must be changed. It should be modified in the same operating system itself.
Docker does not have an OS for itself and thus it creates instances and parameters by sitting on the top of OS. This helps in modifying the instances if needed. It works solely on host OS and does not control hardware of the system.	Hypervisor works with host OS and guest OS which creates layers that run the hardware. We cannot create different instances for the same application in the system. But we can control the hardware and make the system work with both OS.
Docker work only on the software of the operating system and no on the hardware side. It takes the host kernel and works on the principle of virtualization.	Hypervisors can be made to work on software and hardware where it works on the operating system or on the CPU and storage services of the system

## 2) Comparison between Containers and Virtual machines

### Container:

It sits on the top of a physical server and its host operating system. They share a common operating system that requires care and feeding for bug fixes and patches. They are more agile and have high portability than virtual machines.

### Virtual Machine:

It runs on top of an emulating software called the hypervisor which sit between the hardware and the virtual machine. The hypervisor is the key to enable virtualization. It manages the sharing of physical resources into virtual machines. Each virtual machine runs its own guest operating system. They are less agile and have low portability than containers.

Containers	Virtual Machines (VM)
A container is a software that allows different functionalities of an application independently.	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.
Applications running in a container environment share a single OS.	Applications running on VM system can run different OS.
Containers virtualize the operating system only.	VM virtualizes the computer system.
Containers take a few seconds to run.	VM takes minutes to run, due to large size.
Containers are useful when we are required to maximise the running applications using minimal servers.	VM's are useful when we require all of OS resources to run various applications.
Size of container is very light, i.e. a few megabytes.	VM size is very large.

Examples of containers are: Rancher OS, Photon OS, Containers by Docker.	Examples of VM are: KVM, Xen, VMware.
--	---------------------------------------