

ASSIGNMENT 2

Himanshu Ranjan

4NI19IS037

Section - B

1) Comparison between hypervisor and docker

Docker is an open platform for **developing, shipping, and running applications**.

Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications.

A hypervisor, also known as a virtual machine monitor or VMM, is **software that creates and runs virtual machines (VMs)**. A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, such as memory and processing.

Comparison on different functionalities

Functioning Mechanism

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.

Number of Application Instances Supported

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.

Memory Requirement

Hypervisors enable users to run multiple instances of complete operating systems. This makes them resource hungry. They need dedicated resources for any particular 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.

| | HYPERVISOR | DOCKER |
|----------------------|---|--|
| OS SUPPORT | Hypervisors are OS agnostic. | Docker supports only Linux. |
| BOOT TIME | Consumes upto 1 min to boot up. | Boots within seconds. |
| SECURITY | Dual OS layers provide extra data security. | Dependent on supporting Linux kernel. |
| RESOURCE CONSUMPTION | Consumes gigabytes of space. | Docker containers are lightweight. |
| APPLICATION SUPPORT | Can run multiple OS instances simultaneously. | Supports multiple application instances. |

Boot time

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.

A hypervisor might consume up to a minute to boot the OS and get up and running.

Docker can create containers in seconds, and users can get started in no time.

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. |