

# **DevOps Assignment-2**

**Name:** VISHAKHA V

**USN:** 4NI18IS111

**Section:** B

## **1) Comparison between hypervisor and docker**

An open platform for creating, distributing, and running programs is Docker. You may divide your apps from your infrastructure with the help of Docker, allowing for rapid software delivery. You can manage your infrastructure using Docker in the same manner that you manage your applications.

Software that builds and manages virtual machines is called a hypervisor, also referred to as a virtual machine monitor or VMM (VMs). By essentially sharing its resources, such as memory and computation, a hypervisor enables a single host computer to handle numerous guest virtual machines (VMs).

## **Comparison on different functionalities**

### **Functioning Mechanism**

There are two sorts of hypervisors: type one works on the hardware itself, while type two runs on top of the operating system.

Conversely, Docker operates directly on the host kernel. As a result, the user cannot build several instances of the same operating system. Instead, they build containers that serve as user-interactive virtual application environments.

## 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

Users can run several instances of whole operating systems thanks to hypervisors. They become resource poor as a result. On the shared hardware that the hypervisor allots at boot, they require specific resources for each instance.

However, there are no similar requirements for Dockers. As many containers as are required can be made. The amount of processing power made available to the containers by Docker depends on the requirements of the application and its availability.

	HYPERVERSOR	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**

Containers can be created rapidly to get started thanks to Dockers' lack of need for such resource allocations.

The fact that Dockers and containers may be set up in a matter of seconds is one of the main factors contributing to their growing popularity.

It could take a hypervisor up to a minute to start the OS and function.

Users can start using Docker right away and containers can be created in just a few seconds.

## **2) Comparison between Containers and Virtual machines**

### **Container:**

It is supported by its host operating system and a physical server. They share an operating system that needs maintenance and upkeep in order to receive bug updates and patches. When compared to virtual machines, they are more agile and very portable.

### **Virtual Machine:**

It functions on top of the hypervisor, a piece of software that serves as an intermediary between the hardware and the virtual computer. The hypervisor is essential for making virtualization possible. It controls how physical resources are distributed among virtual computers. Each virtual machine has a unique guest operating system running on it. They are less mobile than containers and less agile.

<b>Containers</b>	<b>Virtual Machines (VM)</b>
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.