DevOps Assignment – 2

Name: Aman Narayan Singh

USN: 4NI19IS119

Branch: ISE

Section: ‘B’

Q.1.) Comparison between hypervisor and docker?

- **Functioning Mechanism**

  The most significant difference between hypervisors and Dockers is the way they boot up and consume resources.

  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.

- **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.

- **Architecture Structure**

  If we consider both hypervisor and Docker's architecture, we can notice that the Docker engine sits right on top of the host OS. It only creates instances of the application and libraries.

  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.

  If we were to run a portable program and want to run multiple instances of it, then containers are the best way to go. Hence you can benefit significantly with a Docker.

  Dockers help with the agile way of working. Within each container, different sections of the program can be developed and tested. In the end, all containers can be combined into a single program. Hypervisors do not provide such capability.

- **Security**

  Hypervisors are much more secure since the additional layer helps keep data safe.

  One of the major differences between the two is the capability to run operating systems or rather run on operating systems.

- **OS Support**

  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.

Tabular comparison of hypervisor and docker :

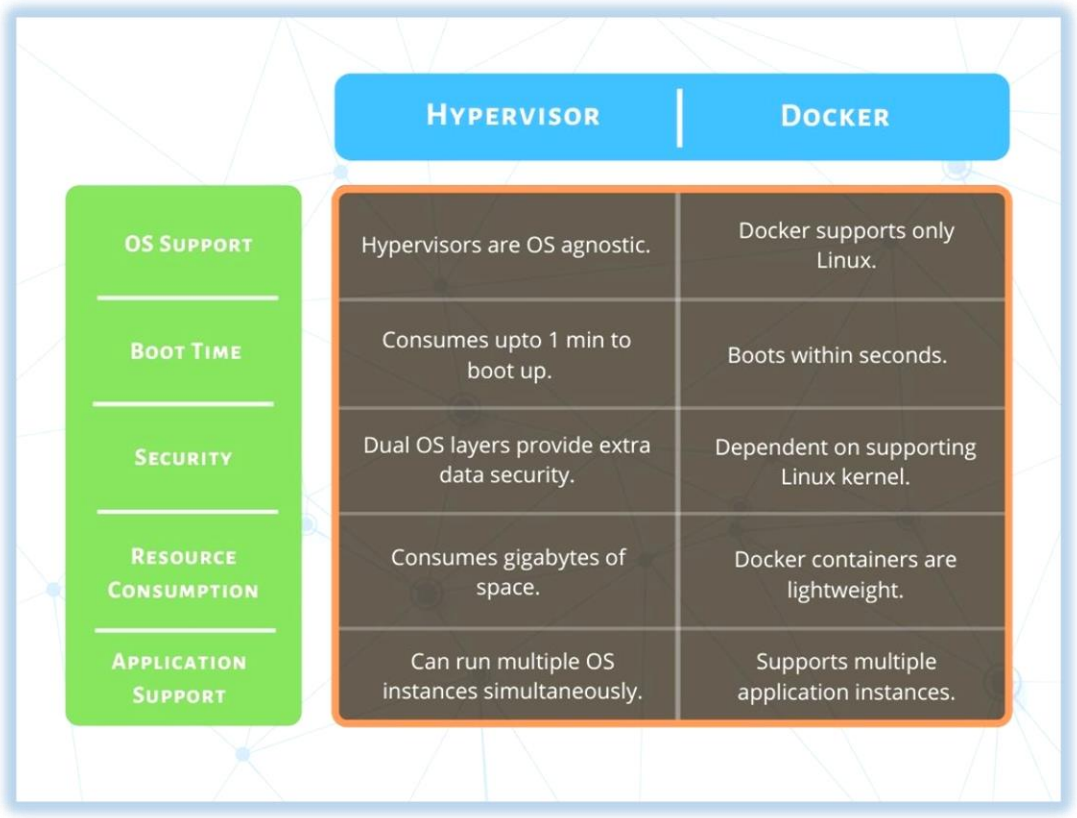

Figure 1: Tabular Comparison of hypervisor and docker

Q.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.

Containers

Containers are lightweight software packages that contain all the dependencies required to execute the contained software application. These dependencies include things like system libraries, external third-party code packages, and other operating system level

applications. The dependencies included in a container exist in stack levels that are higher than the operating system.

Virtual Machines

Virtual machines are heavy software packages that provide complete emulation of low level hardware devices like CPU, Disk and Networking devices. Virtual machines may also include a complementary software stack to run on the emulated hardware. These hardware and software packages combined produce a fully functional snapshot of a computational system.

Tabular comparison between virtual machine and containers

S. No.	Virtual Machine (VM)	Containers
1.	The hardware is virtualized to execute several Operating system instances with VMs.	Containers facilitate a way for virtualizing the operating system so that several workloads can execute on an individual operating system instance.
2.	VM is managed via hypervisor and uses VM hardware.	Containers give services of OS from an underlying host and also separate the applications utilizing virtual-memory hardware.
3.	VM facilitates the abstract machine which utilizes device drivers addressing an abstract machine.	Container facilitates the abstract operating system.
4.	VM technologies are well-known within various embedded communities.	The container has been grown on several clouds and servers with organizations like Google and Facebook. For example, all services of Google Docs get a container/instance.
5.	Higher overhead	Lower overhead
6.	VM permits us for installing other software so virtually we control it as disputed to install the software on a computer directly.	The containers are software that permits distinct application's functionalities independently.
7.	Applications executing on virtual machine system can execute distinct OS.	Applications executing within the container environment contribute to an individual OS.
8.	VM facilitates a way for virtualizing any computer system.	Container only virtualizes the OS.

9.	VMs have a large size.	Containers are very light (some megabytes).
10.	VM runs in minutes due to its large size.	Containers run in seconds.
11.	It utilizes a lot of memory of the system.	Containers utilize very less system memory.
12.	It is highly secured.	It is less secure.
13.	VM is helpful if we need each resource of OS to execute several applications.	A container is helpful if we needed to maximize various executing applications with minimal servers.
14.	VM examples: VMware, Xen, KVM	Container examples: Containers via Docker, PhotonOS, RancherOS.