Dev-ops assignment-2

Name: CHANDRASHEKAR B

USN: 4NI19IS025

Section: A

Topic – Hypervisor and Docker Container and virtual Machines

1. Different Between Hypervisor and Docker

Key Differences of Hypervisor vs Docker

Running on an operating system and having the ability to control the run of an operating system is different. This simplifies the understanding of docker and hypervisor. The hardware can also be controlled with a hypervisor and it works with layers of OS in the same system. This makes the system more secure and if any virus attack happens on software, the entire system is notified via hypervisor to protect it from malicious attacks. An additional layer of security is not present in docker and it runs on the OS. It cannot control the OS and thus the system must be protected from virus attacks, if any, that happens in the system.

Hypervisors support Windows, Linux, and Mac systems and work with the same efficiency in any of these systems. Also, users can work on the same physical system and on different operating systems. Hypervisors provide this flexibility to users. But dockers are not this flexible and work only on Linux operating systems. This makes the users be biased about the docker usage in the system as most users will not be flexible with Linux. But programmers using Linux OS always prefer Dockers to Hypervisors and their community is slowly growing.

Hypervisors play a major role in virtualization and it is used in all the systems where virtual machines are used. Almost all cloud computing services use hypervisors to manage their virtual machines. Dockers are used in places where microservices are used. Also, DevOps methodology is in place, it is better to use Dockers as this is faster and has container setup for different environments and to store different levels of code.

When we are developing an application that must be highly secure, it is good to go with Hypervisors. It does not share instances with any other operating systems and none other than the user can access the same through the same physical machine. Tests on different conditions can be performed securely and this application can be isolated from other applications. This is not possible with containers as the instances are shared among containers and hence the security nature of the application will be lost.

Hypervisor	Docker
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.	Dockers work only on the software of the operating system and not on the hardware side. It takes the host kernel and works on the principle of virtualization.
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.	Docker does not allow users to create multiple instances of operating systems in the same computer but it makes virtualization by making containers in the same system. Containers help users to work separately on different or the same applications. The same operations are carried out by containers in the system.
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.	Resource requirement is low as containers are working on the same operating system and this makes the system share resources within the containers.
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.	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.
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.	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 an agile model.
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 does not have an OS for itself and thus it creates instances and parameters by sitting on top of OS. This helps in modifying the instances if needed. It works solely on the host OS and does not control the hardware of the system.

2. Different between Container and virtual Machines

Virtual machines (VMs)

VM simulates a physical machine with an application, supporting binaries and an operating system (OS image) encapsulated into it and is an abstraction of the hardware layer. VM technology can use one physical server to run the equivalent of multiple servers (each of which is called a VM). So, while multiple VMs run on one physical machine, each VM has its own copy of an OS, applications and their related files, libraries and dependencies.

VMs were introduced before containers and transformed deployment of multiple applications efficiently on existing machines. Before the introduction of VM, it was normal for each application to run on a separate physical machine.

VM is achieved through VM hardware and are managed by a hypervisor. A hypervisor, can be a combination of hardware, firmware or software that deploys, monitors, and manages VMs. A hypervisor supports multiple VMs each with their own OS.

Containers

A Container simulates a logical packaging of an application with required binaries encapsulated into it and is an abstraction of the software layer. The encapsulated container image is isolated from the environment in which it runs. This allows container-based applications to be deployed with ease, regardless of the target environment which could be a private data center, the public cloud, or even a personal computer. A single container can be used to run a microservice or even a software process to a larger application. Containers deploy by using virtual-memory hardware and operate though orchestrators usually provided by the containerization platform, the orchestrator manages resources used by the container and facilitate OS level communications.

SNo.	Virtual Machines(VM)	Containers
1	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.
2.	Applications running on VM system can run different OS.	While applications running in a container environment share a single OS.
3.	VM virtualizes the computer system.	While containers virtualize the operating system only.
4.	VM size is very large.	While the size of container is very light; i.e. a few megabytes.
5.	VM takes minutes to run, due to large size.	While containers take a few seconds to run.
6.	VM uses a lot of system memory.	While containers require very less memory.
7.	VM is more secure.	While containers are less secure.
8.	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.
9.	Examples of VM are: KVM, Xen, VMware.	While examples of containers are:RancherOS, PhotonOS, Containers by Docker.