

DEVOPS ASSIGNMENT 2

---SAMIKSHA(4NI19IS082)

SECTION A

1.Difference between Hypervisor and Docker

Hypervisor-

Hypervisor is building block for Hosting Applications and Virtual Machines. The role of hypervisor to take resources from Physical computers. Further these resources are emulated and abstracted as software and provided to various VMs in form of Virtual Compute, Virtual Storage & Virtual Network. One VM can't consume resources allocated to another VM. Hypervisor isolates and protects these VMs from each other. In case one of VM or Virtual Machines exhausts all its resources, Its not allowed to simply eat up resources for another VM

The Hypervisor runs on Host Machine using Host Operating systems and physical resources. It further hosts multiple VMs also called as Guest Machines. This guest machine or VM contains:-

1. Guest Operating system such as MAC, Windows or Linux
2. The application and supporting system binaries and libraries

Docker-

Docker is an essential tool designed to manage Container. Docker is directly responsible for create, deploy, and run applications based on containers. It was first released in 2013 and is developed by Docker, Inc.

Docker is used to run software packages called “containers”. Containers are created from “images” that specify their precise contents. Images are often created by combining and modifying standard images downloaded from public repositories. Docker manages them all. There are 3 mains components to entire Docker working -BUILD , SHIP and RUN which we will discuss in upcoming Cloud Videos

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	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

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.	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.Difference between Container and Virtual Machine

Container-

It creates an isolated environment in a physical host (server) by sharing the host operating system and executing the packaged applications. Container sits on top of the physical server and host OS, and each container can share the host OS Kernel as well as binaries and libraries, which is more lightweight (only a few MB in size) and takes a few seconds to boot.

Container 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 Machines-

Virtual machine is a technology used to virtualize computing environments which emulates of the underlying hardware such as CPUs, disks, network devices etc. to virtualize a physical computer (server), so you can perform multiple operating systems based on a single computer. The

complete OS makes the virtual machine heavy weight (often up to several gigabytes in size), which takes several minutes to boot it.

Generally, virtual machine needs to interact with the physical computer through the hypervisor which separates the VM and physical host and assign the hardware resource to virtual machines.

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.