**DEVOPS**

**SELF-LEARNING ASSIGNMENT-2**

**SHIVANI L H**

**4NI19IS089**

**A Section**

* **INTRODUCTION**

## Before we get into the topic that is given in the self-learning exercise. Let us start from where it all originated and that would be DevOps.

## DevOps, which stands for development and operations, is considered as a solution to the problem of how we can collaboratively develop, test, operate, deploy in addition to sharing tasks performed by a company's application development and IT operations teams. It can also be expressed as an infinite loop that is meant to improve the work of software development life cycle, consisting of stages being: build, test, release, operate and monitor. Once feedback is given, the loop resets.

## In a broader sense, DevOps endorses for better communication and joint effort between teams within an organization. However, in its most narrow perception, DevOps makes sure to illustrate the adoption of iterative software development, automation, and programmable infrastructure deployment and maintenance.

## **What is Hypervisor?**

Ahypervisor, 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. Hypervisors make it possible to use more of a system’s available resources and provide greater IT mobility since the guest VMs are independent of the host hardware. This means they can be easily moved between different servers.

* **What is Docker?**

Docker is an open-source containerization technology based on Linux that allows developers to write, operate, and bundle programs for container deployment. It is a containerization platform that is free and open source. It allows developers to package programs into containers, which are standardized executable components that combine application source code with the OS libraries and dependencies needed to run that code in any environment.

* **Hypervisor VS Docker**

|  |  |
| --- | --- |
| **Hypervisor** | **Docker** |
| The software that supports virtual machine creation in which a virtual platform is provided for the operating system to manage and run virtual machines is called Hypervisor. | Docker is a virtualization service used in operating systems, where software is distributed in a container with software, libraries, and configuration files. |
| There are two types of hypervisors: the bare metal runs directly on the hardware while the second type runs on the operating system. | Docker, on the other hand, runs on the host kernel itself. |
| A hypervisor allows users to build multiple versions of a complete operating system. | Dockers can run multiple applications or instances of the same application. It does this with containers. |
| The hypervisor is independent of the operating system. They can run on Windows, Mac, and Linux. | Dockers, on the other hand, is to Linux only. However, this shouldn’t be a deterrent to Dockers as Linux is a strong ecosystem. |
| The hypervisor allows users to run multiple instances of the entire operating system. This makes them resource intensive. However, they need dedicated resources for any of the shared hardware that the hypervisor allocates at startup time. | Dockworkers, however, have no such requirement. users can create as many containers as they need. Depending on application requirements and available processing capabilities, Docker provides it for containers. |
| The hypervisor consumes up to 1 minute to boot up and function regularly. | The docker boots up within seconds. Since Dockers do not require such resource allocation to create containers. |
| A well-known example of a hosted hypervisor is Oracle VM VirtualBox. Others include VMware Server and Workstation, Microsoft Virtual PC, KVM, QEMU and Parallels. | An example would be a website, API and database must be connected. This is what Docker Compose allows us to do. We can create a file that defines how containers relate to one another. |

* **What is Container?**

Containers are a form of operating system virtualization. A single container might be used to run anything from a small microservice or software process to a larger application. Inside a container are all the necessary executables, binary code, libraries, and configuration files. They are more lightweight and portable, with significantly less overhead.

* **What are Virtual Machines?**

A virtual machine is a computer file, typically called an image, that behaves like an actual computer. It can run in a window as a separate computing environment, often to run a different operating system or even to function as the user's entire computer experience, as is common on many people's works on computers.

* **Containers VS Virtual Machines**

|  |  |
| --- | --- |
| **Containers** | **Virtual Machines** |
| 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 and control it virtually. |
| It sits on the top of a physical server and its host operating system. | It runs on top of an emulating software called the hypervisor which sit between the hardware and the virtual machine. |
| 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. |
| The size of container is very light, i.e., a few megabytes because of which they take a few seconds to run. | VM size is very large. VM takes minutes to run, due to large size. |
| Containers are useful when we are required to maximize the running applications using minimal servers. | VM’s are useful when we require all of OS resources to run various applications. |
| Containers require very less memory however containers are less secure. | VM uses a lot of system memory however VM is more secure then containers. |
| Examples of containers are RancherOS, PhotonOS, Containers by Docker. | Examples of Virtual Machines are KVM, Xen, VMware. |

**REFERENCES**

1. <https://www.techtarget.com/searchitoperations/definition/DevOps>
2. <https://www.educba.com/hypervisor-vs-docker/>
3. <https://www.ibm.com/cloud/blog/containers-vs-vms>