Devops - Assignment - 03

RollNo: 21111036
Class: MSc. C.S. B-21

1. Explain VM Vs Container resource consumption in your own words with help of diagram?

   Ans.
   A Virtual Machine is a software that allows us to run an entire guest operating system inside the host operating system.
   The VM will have a fixed pre-allocated resources to the guest OS and the host will not be able use these resources once they are allocated to the guest, even if the guest is not using them.
   The VM is acts as a layer between the guest and the host. Each resource required by the guest is requested to the host by the VM and the host will provide the resources. To achieve this some sort of hypervisor is required.
   This is an overhead.

   ```
   +------+ +-------+ +--------+
   | App1 | | App2 | |  App3 |
   +------+ +-------+ +--------+
   |  VM1 | | VM2  | |  VM3  |
   +------+ +-------+ +--------+
   +---------------------------+
   |  Virtual Machine Layer    |
   +---------------------------+
   |     Operating System      |
   +---------------------------+
   |         Hardware          |
   +---------------------------+
   ```

   A container is an isolated environment that directly runs over the host operating system.
   Hence, there is no need to explicitly mention the system resources like CPU, RAM etc. that will be required for its execution.
   The container will request and release resources as and when required, this allows the resources not being used by the container to be utilized by the Host.
   Since container runs directly on top of the host, there is no overhead.

```
+------+ +------+ +-------+
| App1 | | App2 | | App3  |
+------+ +------+ +-------+
+-------------------------+
|     Container Engine    |
+-------------------------+
|     Operating System    |
+-------------------------+
|         Hardware        |
+-------------------------+
```

---

2. Explain containerisation in your own words?
   Ans.
   Containerisation helps us to create images of isolated environments for our applications.
   This image will contain all the necessary software dependencies to run our applications.
   Using these images we can deploy our applications on different machines.
   Images will be executed on top of a Container Engine which manages all the resource handling required for the image.
   Hence there is only a need to install a compatible container engine for the hardware and the rest will be handled by the engine itself.
   This makes sure that all containers will run consistently across different systems.
   Containerisation also allows us to scale our systems and manage them with ease.

---

3. Explain container orchestration in your own words?
   Ans.
   Container orchestration is the automation of much of the operational effort required to run containerized workloads and services. This includes a wide range of things software teams need to manage a container’s life cycle, including provisioning, deployment, scaling (up and down), networking, load balancing and more.
   It involves using a container orchestration system, which provides tools for automating and managing the life cycle of containers, as well as for coordinating their deployment and operation across multiple hosts.
   Some examples of container orchestration systems include Kubernetes, Docker Swarm, and Apache Mesos. These systems provide a powerful set of tools and features for managing containerised applications at scale, and have become an essential part of modern application deployment and management.