**Virtual Machines to Containers**

**From Virtual Machines to Containers: A Comprehensive Guide to Docker**

Introduction:

Before understanding Docker, it is essential to first grasp the concept of containers.

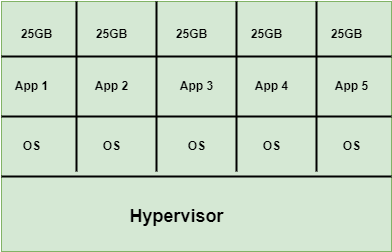
**What are Containers?**

* In an organisation first we have physical server, from physical server they upgraded to virtual machines and finally they upgraded to containers.
* In an organization we use so many operating systems (OS), so in-spite of physical server we use virtual machine.
* Virtual machine is logical thing which we are not going to see.
* By using **Hypervisor,** we can create ’n’ number of VM’s.
* Virtual machines have a full (OS) means they have complete isolation containers do not have a full (OS).
* Virtual machines have solved some problems of physical servers, containers have solved some problems of virtual machines.
* So, to address the virtual machines problems containers concept came into picture.

**What is Virtual Machine?**

* Virtual machine technology involves logically dividing a single physical server into multiple VMs. To run multiple VMs on a physical server, we need a software called a ***Hypervisor***.
* If we have different operating systems (e.g., Ubuntu, CentOS, Red Hat), we can create VMs for each OS, allowing us to test our applications across multiple environments.
* This approach eliminates the need to create new physical servers, which would incur additional costs. Instead, we can create multiple VMs on a single server, making it easier and more cost-effective to test our applications.

**Drawbacks of VM:**

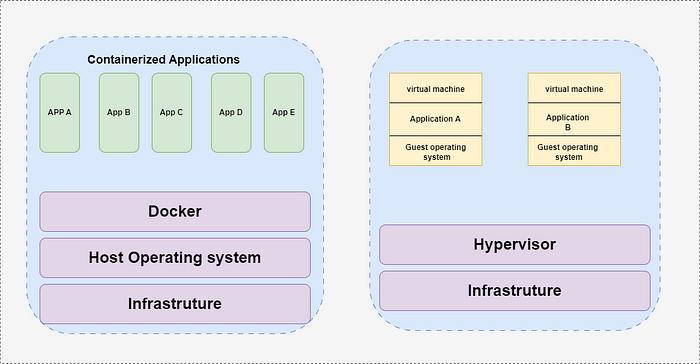


1. Let’s say we have a physical server with 100 GB of RAM and 100 GB of CPU. This server is divided among four applications, each allocated 25 GB of RAM and 25 GB of CPU.
2. On a particular day, Application 1 uses only 10 GB of RAM and 6 GB of CPU out of its allocated 25 GB. This results in 15 GB of RAM and 19 GB of CPU being unused and wasted.
3. To address this inefficiency, containers were introduced. Containers allow for more efficient resource utilization by sharing the host OS kernel and dynamically allocating resources, thereby reducing wastage and improving overall efficiency.

**Advantages of Containers:**

* Inside a virtual machine, you can run multiple containers to improve resource utilization. Containers are ***lightweight*** and do not require a full operating system.
* A container is essentially a package or bundle that includes the application, its libraries, dependencies, and system dependencies. Containers are used by host operating systems such as those running on VMs or EC2 servers. They are lightweight because they share the resources of the base OS rather than requiring a complete OS for each container.
* The container consists of a base image (OS), system dependencies, and application libraries. To run containers, a platform like ***Docker*** is needed, which provides the necessary environment and tools for container management.

**Difference between Virtual machine and Container**



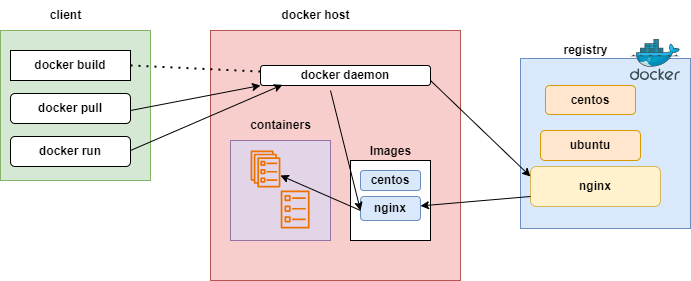
**What is Docker?**

* Docker is a containerization platform that facilitates the creation, management, and deployment of containers. It helps the community by providing tools to write and manage Docker images, which are the blueprints for containers.
* While containerization is the underlying concept or technology, Docker is the platform that implements and operationalizes this concept, making it easier to work with containers.

**Advantage of Docker:**

1. Instead of running a single virtual machine, you can run 10 to 20 containers on the same VM, depending on the resource usage of the containers.
2. The advantage of containers is their ability to efficiently utilize resources. If a container is not actively running, it allows other containers to use the available resources from the kernel or host operating system, optimizing overall resource utilization.

**Docker Architecture:**



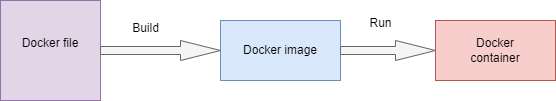
* The heart of Docker is the Docker daemon. When you run Docker CLI commands, they are sent to the Docker daemon, which executes them to create Docker images and containers. Additionally, the Docker daemon handles pushing Docker images to a Docker registry.
* If the Docker daemon goes down, Docker will stop functioning, and containers may cease to work because the daemon is responsible for managing and listening to the containers.
* One of the key benefits of Docker is that it simplifies complex workflows. Instead of performing numerous manual actions, you can write Dockerfiles that define an application or testing environment. Executing a Docker image created from these Dockerfiles sets up everything automatically, greatly enhancing efficiency and reducing manual effort.

**Docker Registry:**

* It is also called as D**ocker Hub**
* It is a place where we store our docker images
* Docker hub is one of the popular docker registry
* Docker hub is a place where we share Docker images with external world,we can create our own registries as well where we share images with other people.

**Life Cycle of Docker:**

***Docker File:***It is a file where we provide the steps to builds the docker image  
***Docker Build***: It is used to create image  
***Docker Run***: It is used to create container  
***Docker Pull***: It is used to enter into container from docker hub



**Conclusion:**

Containers improve resource efficiency and reduce overhead compared to virtual machines. Docker simplifies the management and deployment of these containers, offering a modern, scalable, and cost-effective approach to IT infrastructure.