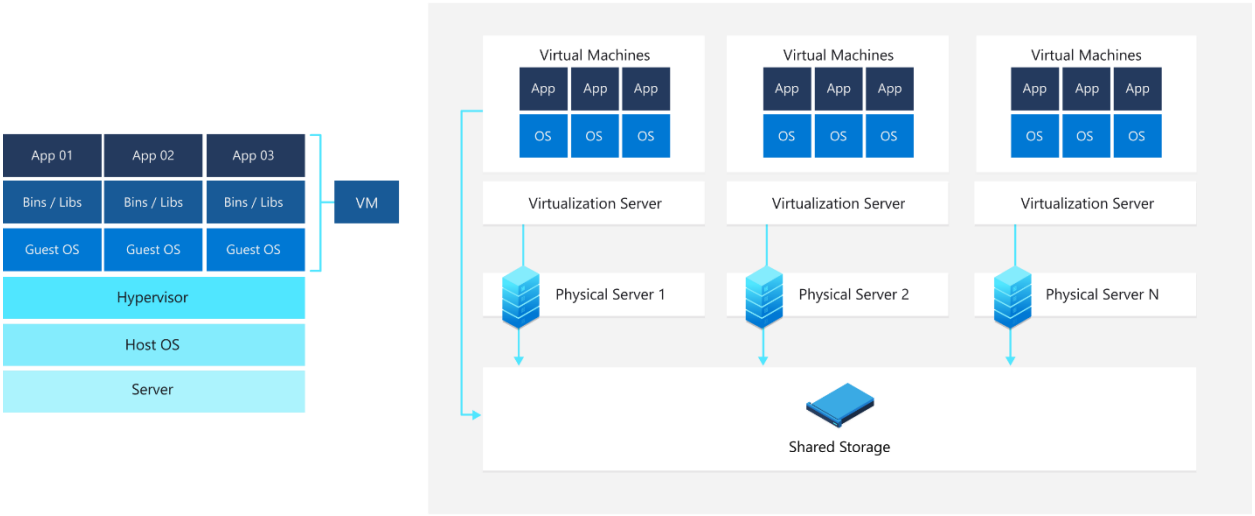
VM Container Assignment

1. Explain Working mechanism of Virtual Machine.

Virtual Machine (VM) is a software used virtualization of an application environment to work as a computer system. They are used to imitate as a computer inside another computer. VM creates an isolated environment to run its own OS and application independent on the underlining host system. While our computers are made up of physical parts, VM are thought of software defined virtual computers.



Computers that host the VMs uses special software called ***Hypervisors.*** Hypervisor emulates the computer's CPU, memory, hard disk, network and other hardware resources, creating a pool of resources that can be allocated to the individual VMs according to their specific requirements. It also schedules operations in VMs so they don't overrun each other when using resources

There are two types of hypervisors:

- **Type 1.** Type 1 or bare-metal hypervisor, this type of hypervisor runs directly on the physical host machine and has direct access to its hardware. Type 1 hypervisors typically run-on server computers and are considered more efficient and better-performing than Type 2 hypervisors, making them well suited to server, desktop and application virtualization. Examples of Type 1 hypervisors include Microsoft Hyper-V and VMware ESXi.

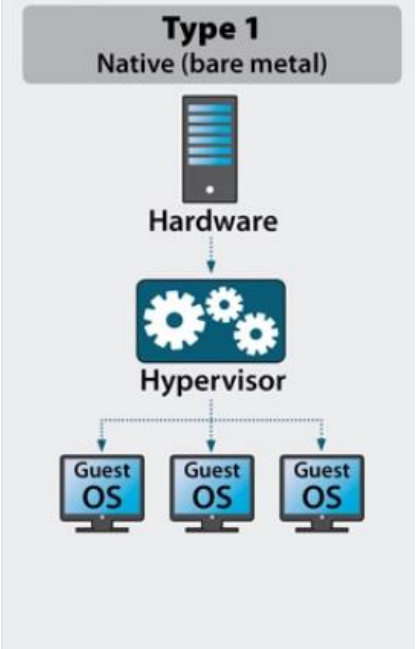


Fig: Type1 Hypervisor

- **Type 2.** Type2 or hosted hypervisor, is installed on top of the host machine's OS, which manages calls to the hardware resources. Type 2 hypervisors are generally deployed on end-user systems for specific use cases. For example, a developer might use a Type 2 hypervisor to create a specific environment for building an application, or a data analyst might use it to test an application in an isolated environment. Examples of Type 2 hypervisors include VMware Workstation and Oracle VirtualBox.

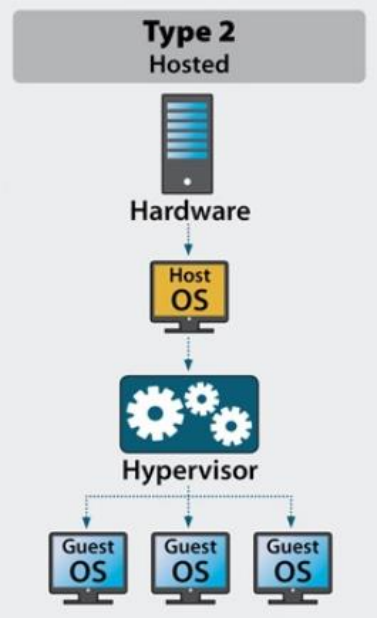


Fig: Type2 Hypervisor

2. Explain Working mechanism of Containers.

Containers directly comes from the concept of shipping containers, where each containers create its own environment for what's inside it. Containers allow applications to be deployed reliably and migrated quickly between various computing environments by packaging code, configuration settings, and dependencies into a single object.

The concept of containers is similar to that of virtual machines. The differentiating factor is that VMs virtualize at the hardware level and containers virtualize at the operating system level. The containerization approach creates a more lightweight and flexible environment by allowing applications to share an operating system while maintaining their own executables, code, libraries, tools, and configuration files.

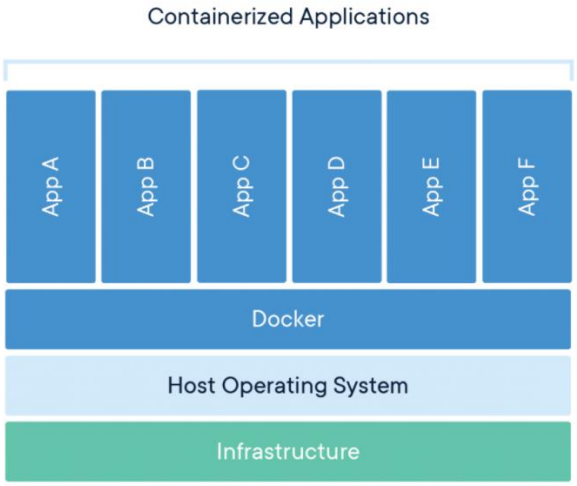
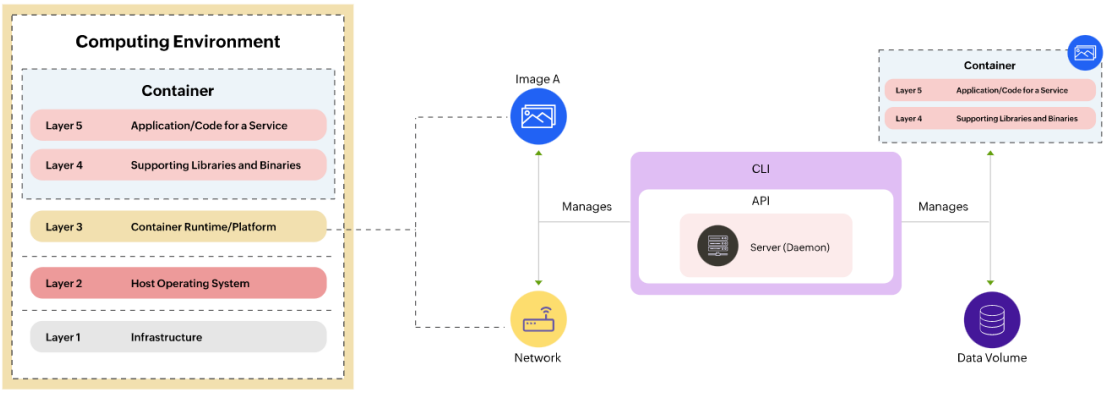


Fig: Containerization

A sample application, or a microservice, is packaged into a container image and deployed for use through the container platform. The container platform is a client-server software facilitating the execution of the container by providing three key operational components:

1. A daemon is a process that runs in the background. This daemon manages objects like images, containers, and other communication (network), and storage (data volume) objects needed by the microservice encapsulated within the container.
2. An application programming interface (API) allows programs to interact with and direct the daemon process.
3. A command line interface (CLI) client issues commands, like "pull" and "run", and is used to access container images from a configured registry. The command line uses the API to

control or interact with the daemon through direct commands, or scripts containing commands. The daemon, in turn, delivers the results through the Host OS System for further processing, or as a final output.



3. What problem does Virtualization solves and what is its drawback in context to modern application deployments?

Problem that Virtualization solves:

- VMs provide isolated environments, making it possible to run different types of OSes and applications on a single server. Organizations can deploy legacy and business applications in the environments they require, without having to deal with contention issues or needing to purchase multiple servers to support different environments.
- VMs make it easy to scale applications and accommodate fluctuating workloads, which is one reason virtualization plays such a key role in cloud computing.
- Organizations also turn to VMs because of the extra layer of security they provide against potential threats. If a VM is compromised, it can be deleted or rolled back to a recent backup or snapshot. Because it's isolated from the host and other VMs, the threat is limited to that VM.

Drawback of Virtualization:

- **Ineffective resource usage**

  Once a VM is assigned to a resource, it takes up the whole space, even when it needs less. This creates idle power that you can use elsewhere if your planning is inaccurate.

- **Not as portable**

  Virtual machines are gigabytes-sized chunks of software. The applications run on a virtual machine that is highly dependent on the OS and the emulated hardware it runs on.

4. What is the problems Container solving in regard to app deployment and how it solves?

Containers can solve some drawbacks from Virtualization but use case of each architecture depends on the requirement of the application.

In regards to app deployment containers can solve problems like:

- **Increased portability**

  Containerization helps to package applications and their dependencies into portable application images that are easy to distribute to artifact repositories and then onto container hosts that will run them.

- **Running applications in isolation**

  Container engine and command line tool make it simple to retrieve application images and start isolated instances of each application process.

- **Greater efficiency**

  Containers makes it easy to follow agile development methodology and allows applications to be more rapidly deployed, patched, or scaled.