

What Is Containers Architecture?

 medium.com/fintechexplained/what-is-a-container-architecture-design-54826e93fc18

How Are Containers Different Than Virtual Machines? Most Important Technical Concept Of The Decade

Recently I received a number of messages from the readers asking whether I can write about containers. As I have a very limited containers knowledge, I was hooked onto learn about them.

This article is therefore dedicated to “Containers”.

Conclusion: Get to know containers—Soon everyone in Technology will be talking about them

Please read the [Disclaimer](#).

Additionally, have a look at this article where I explicitly invite anyone to message me with their proposed subject or question and I can then research and write for them. Not only it helps me with my quest to learn new concepts better, it also helps us collectively spread knowledge across the globe:

We are in the cloud era. Containers are one of the core components in this era.

Article Structure

This article is composed of following subsections:

1. Understanding of how OS and Kernel work. This knowledge is required to form the basics.
2. How virtual machines work
3. What containers are
4. Benefits and drawbacks of containers
5. Software container applications

Containers, light-weight software applications, offer a number of benefits including reduction in time to deliver an application, scalability, ease of application management, support and better user-experience.

Before we look into containers, let's understand how Operating System works. In particular, let's get ourselves familiar with the role of kernel which sits in the heart of an operating system.



Photo by [Michał Parzuchowski](#) on [Unsplash](#)

First let's build the basics and understand how an Operating System Works?

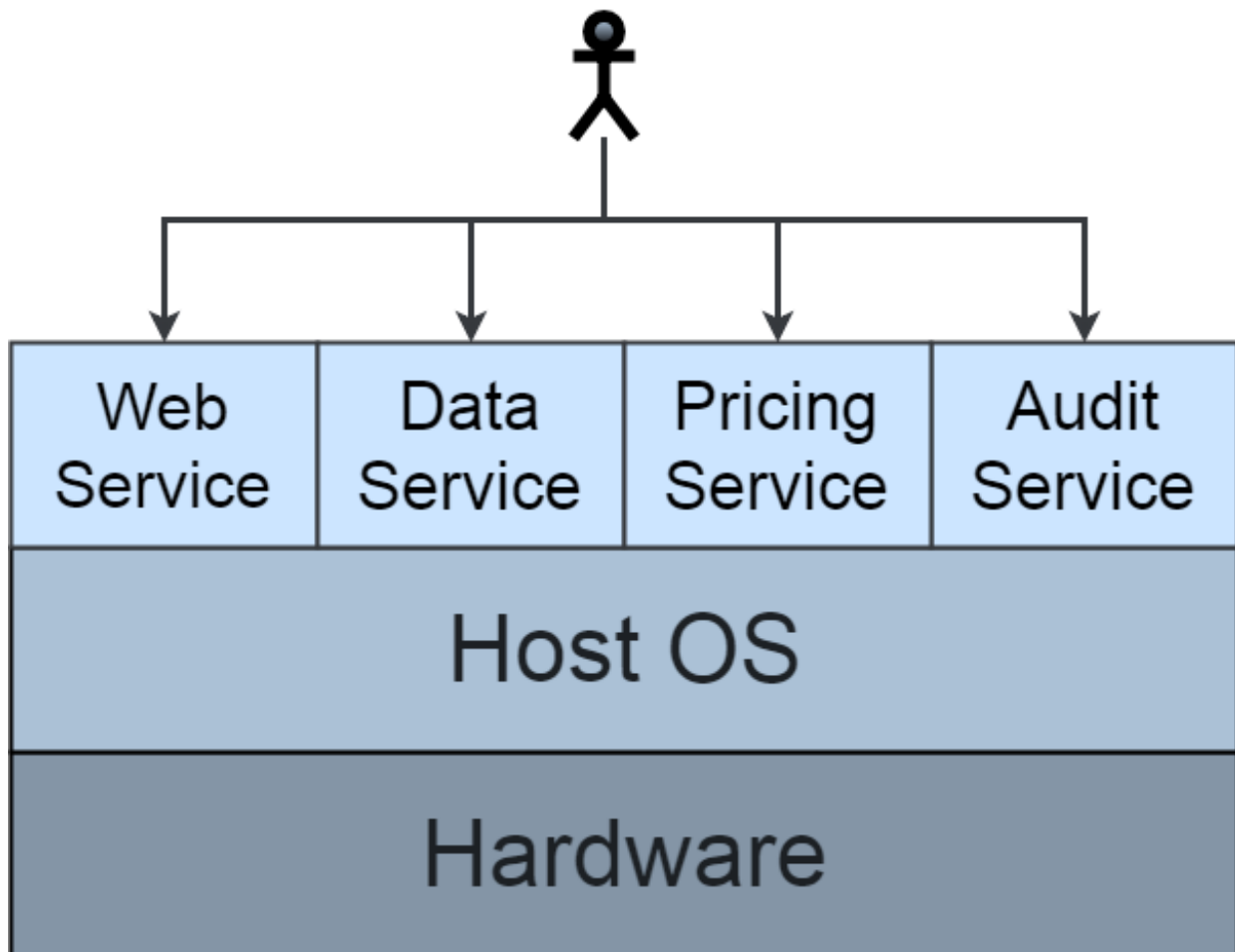
An operating system (OS) is the most important software of a computer system. It manages the hardware of the computer. Additionally, the operating system is responsible for managing software applications which end-users and other applications interact with.

| Operating System is A Bundle Of Software Applications

The operating system has a number of responsibilities including scheduling tasks, processes, allocating memory so that the applications can run in isolation. Additionally, it provides an abstraction layer on how applications communicate with the hardware resources.

There are a number of operating systems available in market such as Microsoft Windows, Mac OS, Google Android and Linux Ubuntu amongst others.

OS sits between software applications and hardware as demonstrated in the image below



An operating system consists of a number of internal programs whereby programs themselves are merely instructions. There is one program that forms the core of an OS and it is known as the **kernel**.

This brings us to the second section of this article. Kernel plays the role of a heart in an operating system.

Let's briefly understand the role of a kernel

What Is A Kernel?

A kernel is the most important program of an OS. The primary operating system of a computer system is known as the Host OS. We can launch a number of additional OS on a host OS as I will explain later in this article.

These additional operating systems are known as guest OS.

| Kernel is the core of an operating system.

Think of a kernel as a bridge that connects software applications on a computer system with the processing of data on the hardware.

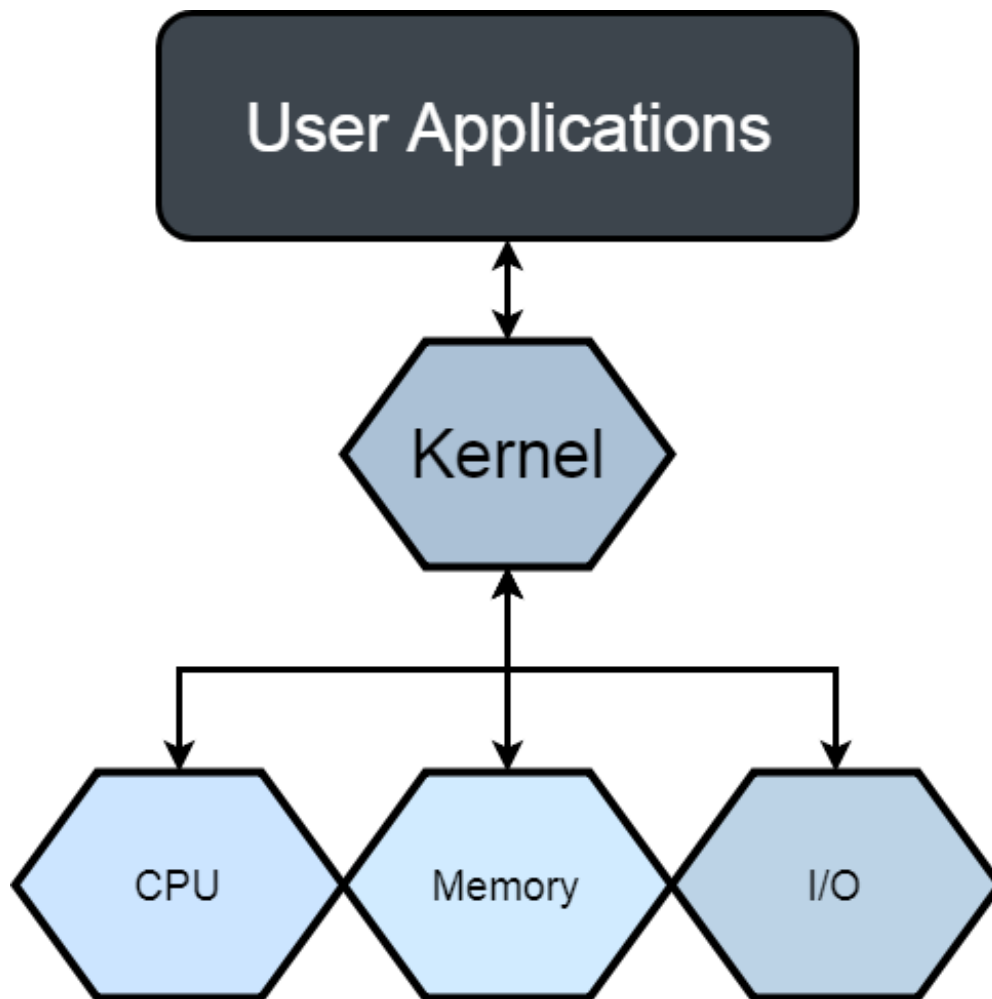
The importance of a kernel is demonstrated by the fact that it is located in its own area of memory. Therefore a kernel gets a dedicated area in the computer RAM memory. The area is isolated from other applications. The nature of the design prevents user applications to corrupt or interrupt with kernel of an operating system.

| User applications run within User Space and Kernel is executed within Kernel Space.

When a computer boots up, kernel is one of the first computer programs to be loaded in the computer system. Additionally, kernel remains in memory until the computer system is switched off.

| Kernel sits in the center of the computer system; in-between central processing unit (CPU), memory, hardware and user applications.

Kernel handles communication with the hardware devices such as mouse, keyboard, monitor, disks, and printers. Kernel also converts instructions for the CPU. It effectively performs the low-level tasks.



Note: CPU executes instructions by decomposing them into sub-instructions.
One could imagine CPU as the brain of the computer system.

The key point to take from the above diagram is that the kernel is responsible for memory management of the applications—it sets up and loads applications into memory. Consequently, it is responsible for maintaining processes.

Quick Introduction To Hypervisor

Kernels have unrestricted access whereas user applications have to access hardware through other processes. One of the key processes is known as **Hypervisor**. Hypervisor sits in between computer system applications and physical hardware. As a result, if an application wants to interact with a CPU, it will have to pass through the operating system and then the hypervisor.

The design introduces a number of problems which require our attention:

The multi-layered design can impact the user experience. We can't run two host OS simultaneously on a physical computer system. Additionally an application can corrupt/interrupt other applications. Setting up infrastructure of hardware systems is time consuming and expensive.

Therefore containers were then introduced to save cost, improve maintainability, scalability, further enhance the user experience and to reduce the time-to-deliver of an application.

Up till now, we have been building the basic blocks to understand how containers work and the benefits they offer.

Before we take a deep down into containers, let's understand another piece in the puzzle first—virtual machines.

This brings us to the third section of the article: Virtual Machines

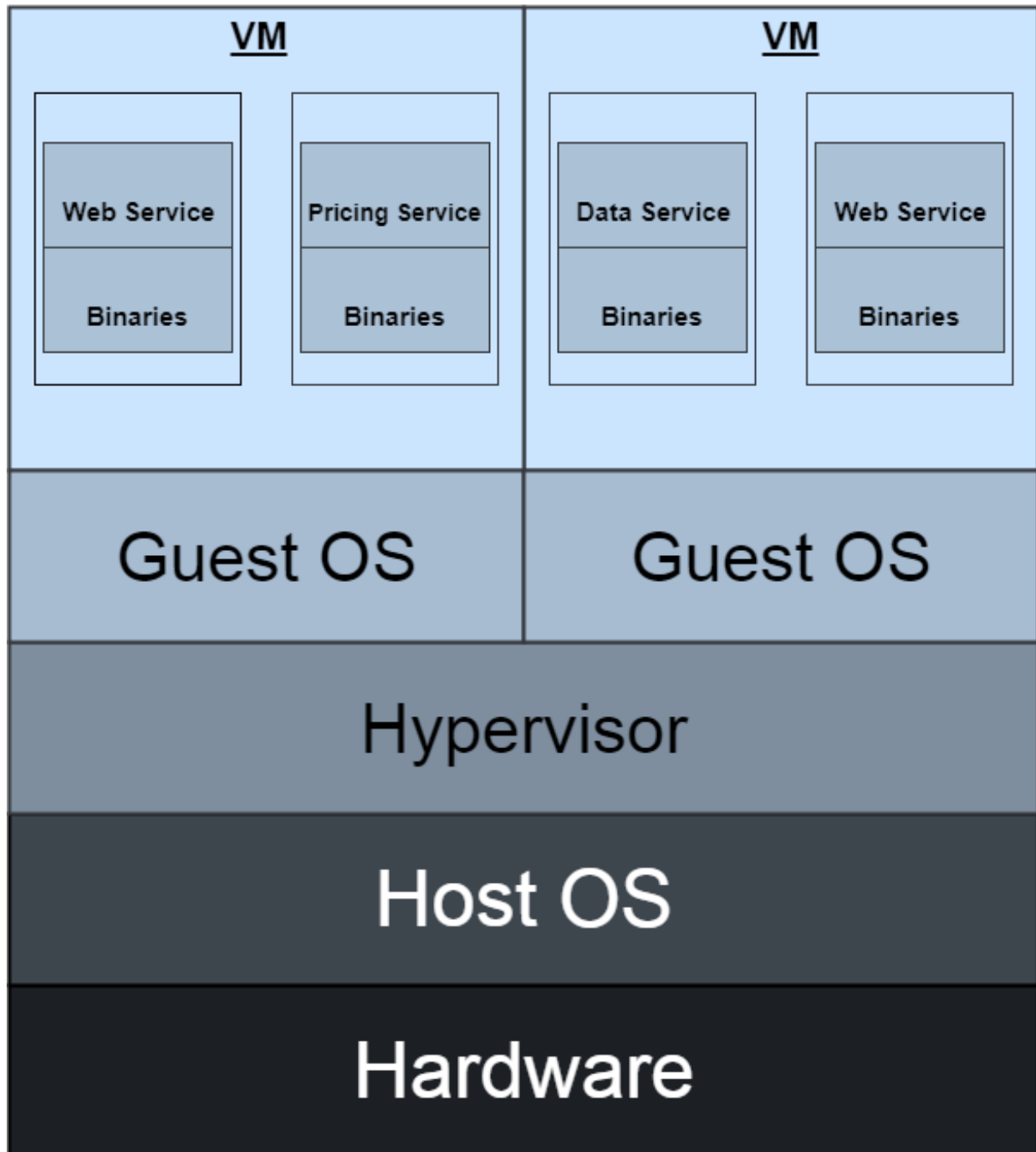
Virtual Machines (VM)

Virtual machines separate the software applications from the hardware of a computer system. We can **clone** a computer system by creating its virtual machine. VMs provide an abstraction layer and hide the complexities of underlying hardware components. VMs can be used to operate the physical hardware systems. They have been heavily used in the industry for over a decade now. A hypervisor process launches virtual machines.

| A virtual machine creates a virtual hardware system.

We can create a virtual machine on a Microsoft Windows operating system and install a Linux operating system on it. This offers greater flexibility to IT users.

| Think of a virtual machine as a version of your computer system.



The guest OS are all sharing the host OS. Each guest OS is launched within a separate virtual machine.

Installing, deploying and replicating a VM is cheaper than buying a full set of infrastructure. We can deploy and install a custom application on a virtual machine. It requires a separate OS (known as guest OS) to run the application on a VM.

As a consequence, the VMs have their own guest OS which include their own kernel, drivers and the binaries of the applications. Subsequently, the VM operates on a hypervisor which runs on the guest operating

system. As a result, if an application wants to interact with a CPU, it will have to pass through the operating system and then the hypervisor.

Passing through hypervisor and host OS can impact the user experience.

VMs introduce problems because not only they are a memory-hungry programs, they add duplication of application binaries between different virtual machines. Additionally it takes a longer time to boot up a VM.

Scenario:

Let's assume you want to launch two VMs within a Windows OS. In addition to it, you desire to host your server code (services) on one VM and the client code (UI) of your application on the other VM. You might decide to prepare a VM with Windows OS and then clone it. The fact that you have now cloned a VM introduces duplication of OS and libraries. It also means that the drivers will need to live in both of the OS. Therefore both of the VMs in turn will consume larger memory space.

Subsequently, you can only launch a handful of VMs on the physical server.

Virtual machines are heavy computer programs. They require server memory. Additionally, only a number of VMs can simultaneously run on a computer system.

The degradation of VM boot-up time, installation, performance, maintenance and replication has introduced the birth of a new technique which is known as containerisation.

How does a container work?

Virtual machines clone an operating system for themselves whereas containers can share an operating system.

Firstly, containers do not require a guest OS. They are software programs. All applications within a container run in the User Space of the OS which is line with the design of a host OS whereby only kernel is given

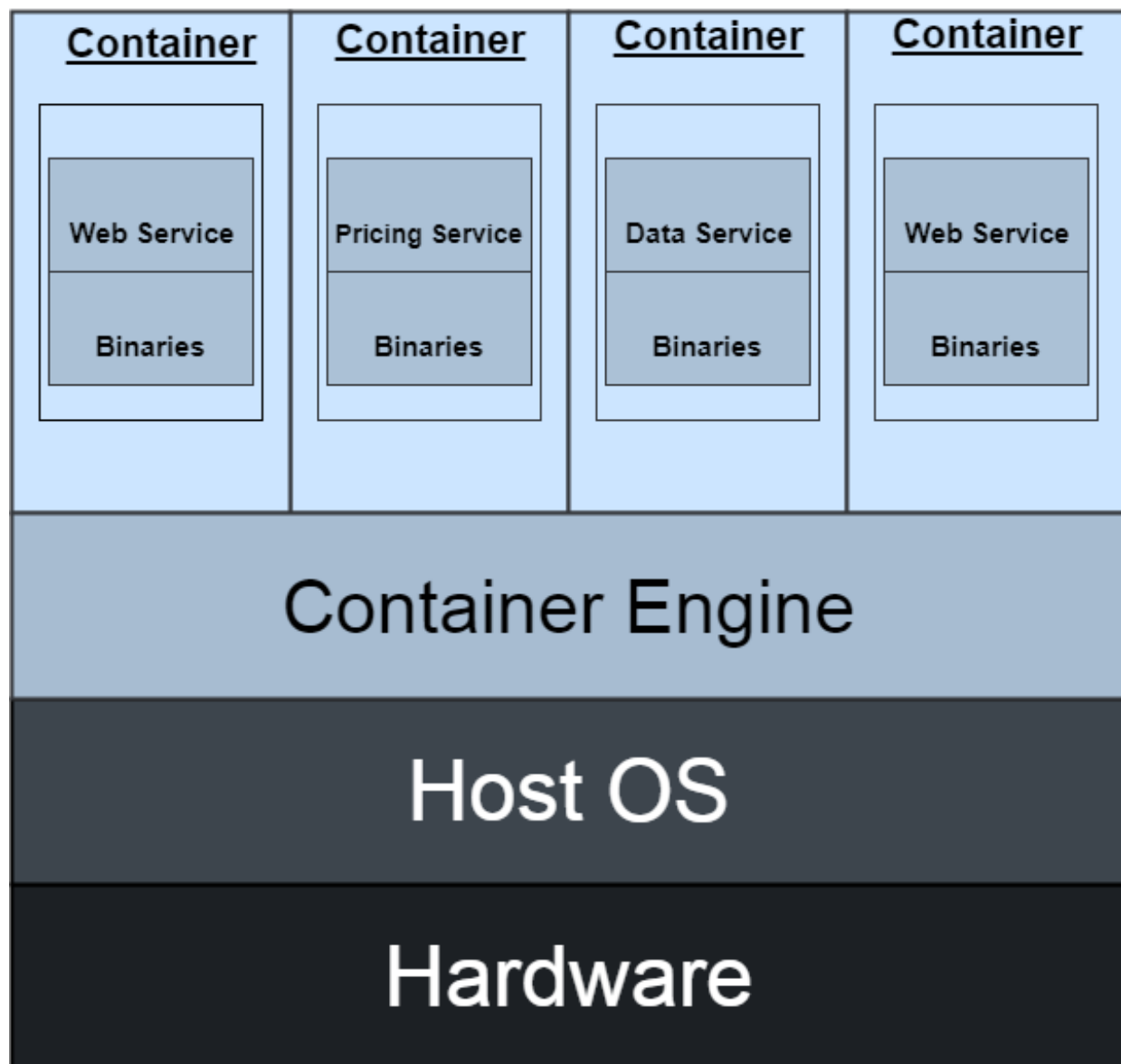
its dedicated space. This allows applications to communicate with the CPU without passing through the guest operating system and then the hypervisor. As a result, containers offer better performance.

The issues introduced by VMs is somewhat resolved because a single host OS and its drivers including application binaries are shared. It therefore implies that only the relevant binaries and resources are hosted on the container.

Containers replicate the file system and enable us to run applications in a secure environment. All of the resources and the files run within the container file-system. The environment variables along with the libraries are stored within the containers. This promotes faster execution of instructions in containers when compared to the hypervisor based instances.

Containers offer sandbox environment. They bring abstraction to the OS.

Containers are launched in a container engine. Subsequently a container engine can launch a number of containers. The containers have a file based configuration system. These files can be versioned, backed-up and monitored. Therefore it is much easier to compare two containers.



Containers contain images which hold the entire information that the containers need to execute. Containers encourage splitting monolithic applications into micro-applications and then setting up communication amongst them. The fundamentals of microservices allow IT teams to only enhance, implement and deploy the required parts of the applications.

Microservices go hand in hand with containers as distributed microservices can be hosted and scaled using containers. If you want to read more on microservices then have a look at this article:

Instead of a hardware system virtualisation, containers virtualise the OS.

All containers can share the same OS, therefore they can all share the same kernel. As a result, the boot-up time is faster. Note—we do not need to launch container on a guest OS. This design increases efficiency

in containers based architecture.

| Containers are light-weight computer programs.

Containers Vs Virtual Machines — Horizontal Comparison

In a VM, a host OS is installed that communicates with the hardware. Then the binaries are installed on the guest VM. Additional applications such as web applications, database servers etc are then installed on the VM. On the other hand, in a container application, multiple guest OS can be installed on a host OS and each guest OS can host a separate application. As an instance, web application can be deployed on a different guest OS than the database server. All of the guest OS communicate to the same underlying hardware.

| As a result, the containers are easier to migrate and clone. They require less memory. A number of containers can be hosted on a physical server.

Containers create a wrapper around applications that can all access the same hardware resources. As a result, it further improves maintainability and availability of the applications.

Containers can be isolated from each other. This further protects the applications and encourages security between the applications. Furthermore, new containers can be created and launched for an application that can all access the same OS kernel. Containers can be hosted on VMs that themselves can be hosted on cloud. Each container can run on a different OS too.

Scenario:

Let's assume a scenario that you want to deploy an application to a container. You can then make your application level updates and directly deploy the artifacts on the container image. The container image can then be deployed to the host OS. Furthermore, you can build an image and deploy it from development to test to production environments. The consistency in images encourages stability in the application across

systems. Each image can therefore be versioned and its progress over time can be tracked. Furthermore the minimal size of container images reduces the time it is required to deliver an enterprise level application.

| The concept of containers is all about sharing resources where appropriate.



Photo by [Lucrezia Carnelos](#) on [Unsplash](#)

What Are The Benefits Of Containers?

Containers offer following benefits:

1. Application level isolation
2. Faster to set up than VM
3. Takes less memory than VM
4. Easier to migrate, back up and transport due to their smaller sizes when compared to VMs
5. Faster communication to hardware therefore they can be efficient for performance

6. Improves application deployment and maintenance due to self-contained container images.
7. Reduces time to deliver the application
8. Encourages micro architecture and design

What Are The Common Drawbacks Of Containers?

Further improvements can be made on the containers.

1. Containers are not as mature as VMs and their performance is still to be tested on larger scales.
2. Not many IT admin consultants have experience with container technology. It therefore makes it difficult to support the applications in long-term.
3. Additionally, containers have become the buzz-word due to the cloud era but will it fly in the future is still debatable.
4. It adds “Yet Another Tool To Manage” in IT infrastructure.
5. As the applications are not fully isolated, they are not as secure as they are in VMs and the security can be further improved.

Which Software Providers Offer Containers?

- Dockers have gained enormous popularity recently. Docker has a number of layers including base image that has OS. Applications can be installed on the appropriate layers only.
- CoreOS—another famous container technology that can execute Docker containers.
- Rocket—gaining popularity

Summary

This article provided an overview of containers. Furthermore, it outlined how Kernel and Applications isolation in OS work. It also covered an overview of a virtual machine and benefits and drawbacks of containers.

Lastly it listed a number of software container applications.

Your questions and comments are welcome. Please let me know if you have any feedback and if you want me to write on any other article that interests you.

Hope it helps.