# 4.1.1.5 Virtualized Nodes

Return to Platforms

## About

Return to Top

A Virtual Node can reside on any machine that supports either Virtual Machines (VMs) or Application Containers (or refered to as Containers). These can run many different Hardware Platforms and or Operating Systems (OS) Platforms. VMs are limited more by the hardware platforms they run on because of the footprint size of hypervisor, whereas containers are smaller and can consequently run on more machines.

Superficially, the two can appear almost identical. However, VMs have a full Host OS as well as a Guest OS while the Application Containers are lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. They are a bit less secure than VMs since Containers manage memory versus giving each instance its own “machine” complete with memory.

Containers and VMs have similar goals: to isolate an application and its dependencies into a self-contained unit that can run anywhere (i.e., Virtual Nodes)

Moreover, containers and VMs remove the need for physical hardware, allowing for more efficient use of computing resources through sharing, both in terms of energy consumption and cost effectiveness.

The main difference between containers and VMs is in their architectural approach.

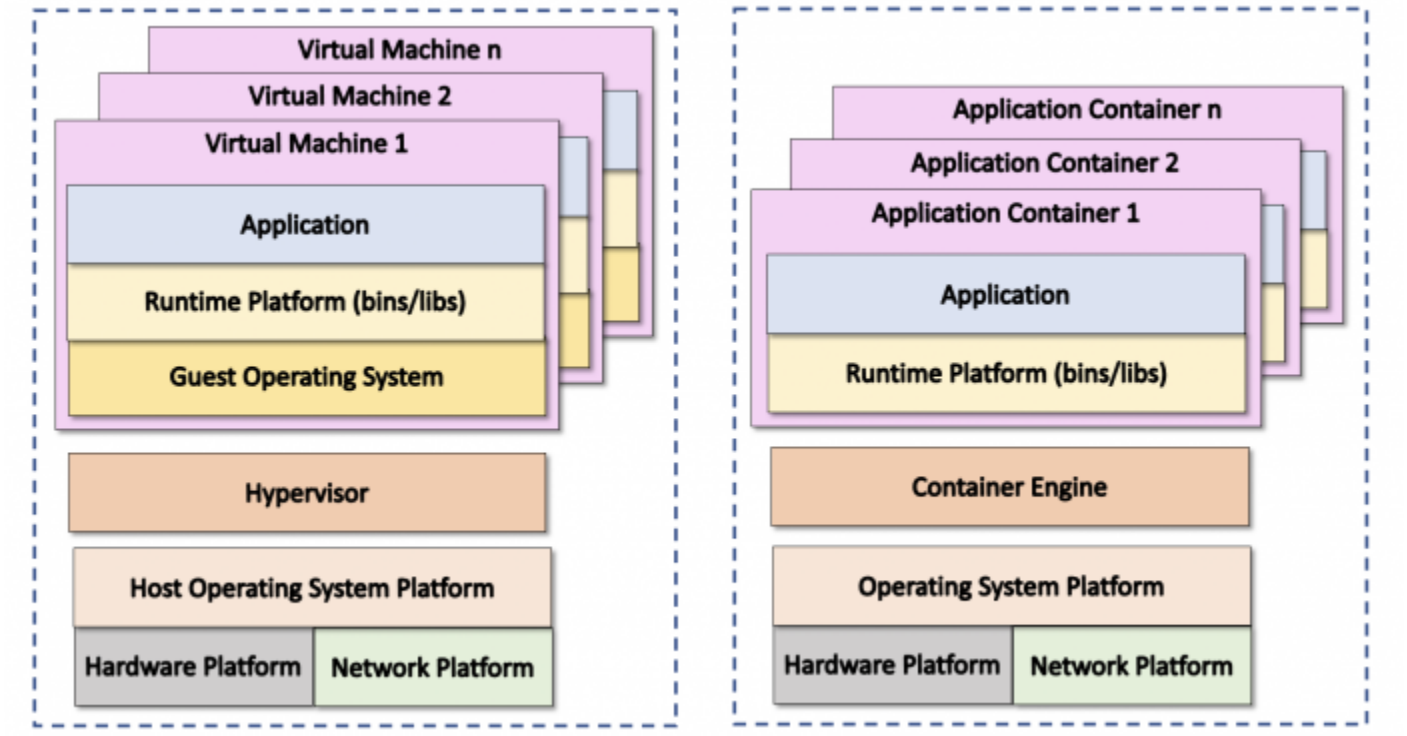



Figure 1: TVirtual Machines and Application Containers.

# Virtual Machines

Return to Top

A Virtual Machine (VM) is a software program that behaves as if it were a complete computer. Often the VMs are software versions of an existing Hardware Platform. For example, the Hardware Server running an Reduced Instruction Set Computer (RISC) processor, might host a VM that behave as if it were an x86, 68000, CISC, RISC, ARM, etc. The VM will have a virtual processor that executes the appropriate instruction set such as Complex Instruction Set Computer (CISC). Generally, the Hypervisor also known as a Virtual Machine Monitor (VMM) creates, runs and monitors the execution of the VMs. Each VM is isolated from each other and from the host machine which is a good for security reasons. For example, the memory (also virtual) is allocated for only for one VM. If another VM needs to use the same physical memory, the memory is “zeroed” before it can be reused.

As a general rule, Hypervisor's are heavy when compared to Container Engines and Virtual Machines are heavier than Containers. The “heaviness” is in reference to the resources required. For example, a Hypervisor takes more memory and Central Processing Unit (CPU) than a Container Engine because of the amount of work that is required to create and manage the VMs. This means that the smaller the host machine, the more likely that Containers are a better solution. Another reason that VMs are heavier is that each VM has its own copy of a a Guest Operating system while Containers are designed to share key parts of the host operating system.

# Application Containers

Return to Top

An Application Container (also know as a Container) is a stand-alone, all-in-one package for a software application that run on an Container Engine. Containers include the the binaries and libraries used to run the application as well the support for the hardware needed to execute the application. Another way to think about a Container is that it is that everything need to execute the application is bundled into a single package (general one file).

Core Operating System (Core OS) is a system for container-based virtualization. Core OS deploys applications in virtual containers as a way to provide effective hardware virtualization for businesses.

- Container Engine

- Disk Image
- Virtual Disk Image (VDI)
- Virtual Machine Images

From:
https://omgwiki.org/dido/ - **DIDO Wiki**

Permanent link:
**https://omgwiki.org/dido/doku.php?id=dido:public:ra:1.4_req:1_func:platform:virtnodes&rev=1609871949**

Last update: **2021/01/05 13:39**