Virtualization vs Containers

Virtualization and virtual machines

Virtualization is the process where a software called hypervisor (which sits on top of your native operating system) allows other operating systems to run on top of your native operating system.

The hypervisor creates virtual machines which is an emulation of physical computer, it creates virtual resources like CPU, RAM, disk, and networking giving the virtual machine the allusion that it is on its own independent computer. However, in reality it is actually managed by the hypervisor.

So with virtualization, it will recreate the underlying hardware for every virtual machines that you spin up. Which might be inefficient but it is good isolation as each of the operating system won't influence each other.

In addition, with virtual machines you are archiving isolation of machines. Each virtual machine are isolated from each other and you are able to run a full running operating system.

Underlying resources for each virtual machine are accessed by hypervisor.

Virtual machine you have flexibility on hardware, you can give say 5 virtual machines all 8 cores, despite only having 8 cores on your host machine.

Containers

Container is a lighter-weight more faster way of handling virtualization. They do not use a hypervisor but instead they use a container engine to run multiple container on the same kernel.

Container do not recreate the physical hardware and they package all the dependencies and software, and even the operating system itself that is required to execute the contained software application. This basically allows you to run your application anywhere, regardless of the physical hardware that's underneath.

Containers allows micro service architectures and they are kind of built for it.

With containers you are archiving isolation of processes. Each container will be run as a process and they are isolated from each other. Normal process that we run will able to inspect other processes! That is possible!!! However, with container when they run as a process, it will apparent to themselves that they don't see other processes being run,

all the software and code that's necessary are packaged into them already.

Container's resources are accessed by kernel features.

Containers have portability, it has all the code that's necessary to run the container, you can take the docker image and run it pretty much anyway as long as you built it for ARM or x86 their corresponding architecture.

Revision #1
Created 2023-02-16 17:25:18 UTC by Tamarine
Updated 2023-06-23 14:12:53 UTC by Tamarine