Note of Virtualisation

Normally when having an application running you have some Hardware (HW), where upon you have the operating system (OS) running then the application (SW) is running in the environment of this operating system. E.g., a Google Chrome browser (SW) is running on a windows 10 (OS) at an ASUS Laptop.

This is how the situation have been for many a years, though lately the concept of virtualisation have taken over starting in mobile phones and now a days moving in more traditional platforms as computers and laptops. Actually at the server site (server clusters) these technologies have been used for more than a decade. However, that is another story at 'client' computers the virtualisation is rather new. The mayor benefit for doing this is to separate the different application whereby you can lower the risk for a security hole in or break down off one application infect other applications.

When talking off Virtualisation this can appear at different forms, which will be discussed in this paper. One form is to have a Virtual Machine (e.g. VMware®, VirtualBox or similar) another form is to have a Virtual Operating System (e.g. Docker or similar). Below are these two forms discussed in greater detail.

The Virtual Machine

The more traditional forms off virtualisations, the Virtual Machines works like; you have a computer (HW) with some Operating System (OS), this HW and OS together is the host system. On this host system a Virtual Machine is simulated i.e. you have an application (VMware®, VirtualBox or similar) running to simulate a Machine (HW).

When having this Virtual Machine you can install any Operating System (OS), e.g. on a Windows host machine, you can install another Windows OS, Linux OS or whatever you find appropriated. You can now run this new OS on the Host like having a complete ordinary computer with the chosen OS and the application / programs you would like to install.

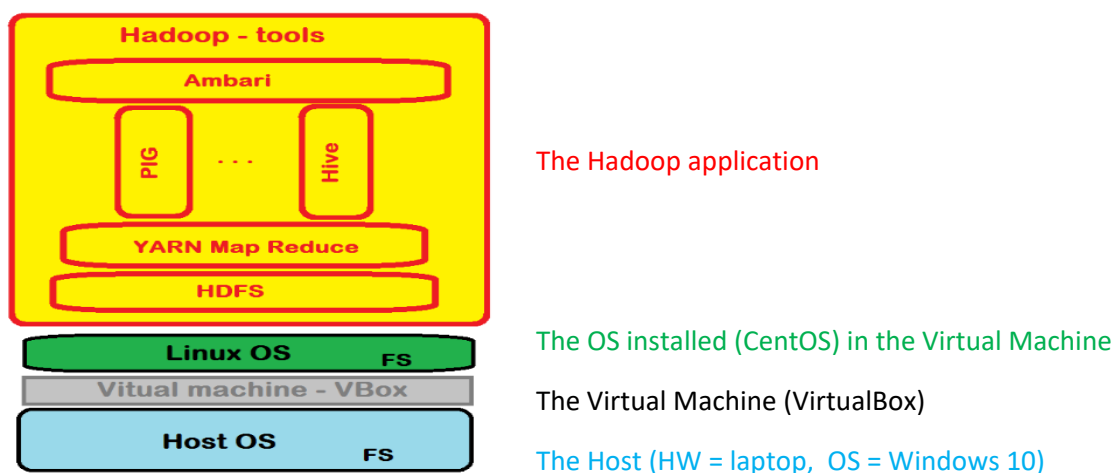An example is to have Big Data environment Hadoop running like:

Figure 1: Hadoop installation using a Virtual Machine

In this way, you can have your normal Windows based computer and install a product (Hadoop) even this product only is running in Linux. Many Student with an Apple MacBook have a similar approach where they have a Virtual Machine with Windows to be able to run Visual Studio.

On one host system, you can have several Virtual Machines running. Each Virtual Machine are running separately from each other i.e. no overlapping CPU, Registers, Memory, Disk nor Network. This way you have the possibility to emulate a network setup with one Server (e.g. Windows Server) and two 'Client' computers (e.g. Windows 7) one on each LAN see figure below:
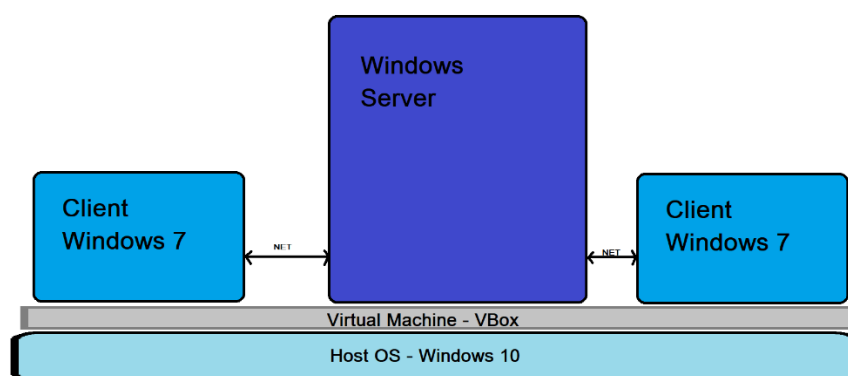

Figure 2: Three Virtual Machines on the same Host Computer

Summery

All this is very fine and not that difficult to setup where you can use many different OS's e.g. Windows XP, Ubuntu, Solaris etc. along with applications, but working with Virtual Machines require many resources off the Host system both regarding the CPU along with Disk usage:

The CPU is need to run the Virtual Machine Application and inside the Virtual Machine the OS and the application.

For the Disk usage each Virtual Machine must have a one-2-one byte disk space on the Host computer, this disk area are the disk where the OS is installed. E.g. the Hadoop solution from figure 1 is using 25 GB disk space.

The Virtual Operating System

To get rid of most of the drawbacks for the Virtual Machines you can now make use of Virtual Operating System e.g. Docker. Each Virtual Operating System use the underlying Host OS more directly by mapping all OS-call to the Host OS, so you have only one OS to power, at the same time each application have its own memory, registers and so forth furthermore can the application share the disk of the Host Computer. This way each application is running in a sandbox so they do not disturb each other unless you let several applications share the same disk space.

Before you can run applications on a Virtual OS, you have to wrap them inside a container. An example for the Hadoop (similar the one in figure 1):
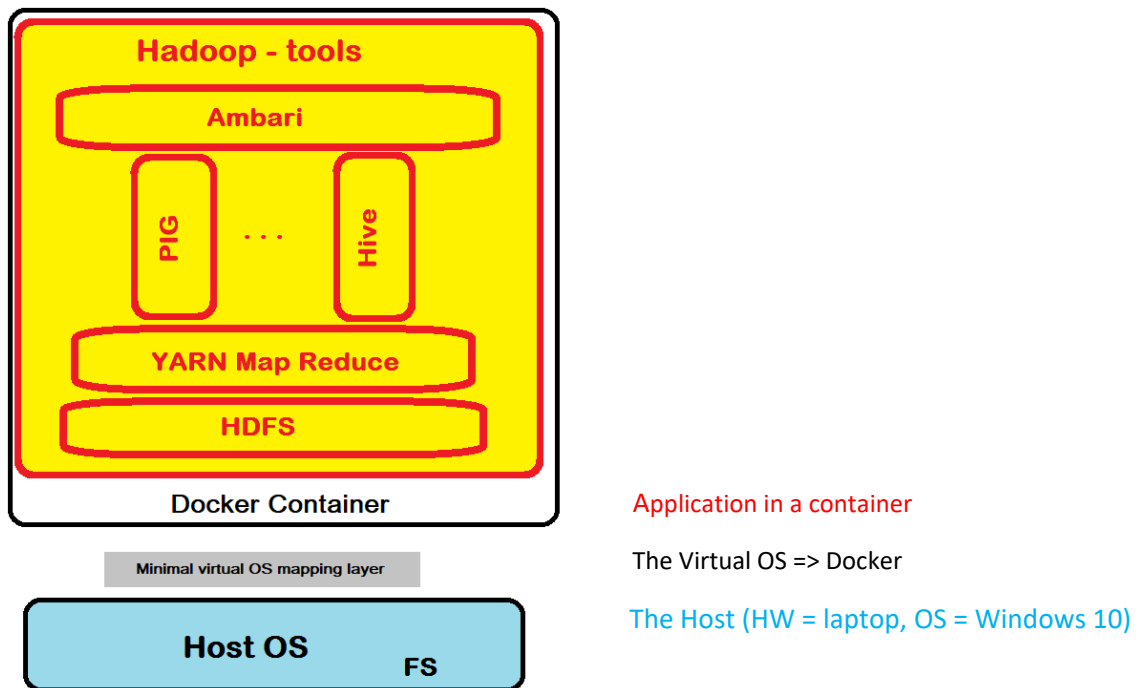
Figure 3: Hadoop installation using a Virtual Operating System

The idea with Virtual Operating Systems like Docker is to have every application running in its own container a kind of micro service idea. Hereby can each application as already mentioned run separately, so a security hole in one application cannot propagate to another application. Because of these kind of firewalls between application you have to link two application together if one should use another e.g. a webserver that need access to a database-server.

A smart feature is that you can have several instances of the same service (application) running independently in parallel, you just start up two or more containers on the Virtual OS. You also have the possibility to have a container running with a test version and another with the production version.

In general, this kind of Virtual OS (Docker) solution is much appropriated together with continuous integration, for the reason an application can easily be updated as only the changes from one version to the next are transferred / downloaded from the application source (in Docker terminology the hub).

Containers in Docker

As described earlier, all application needed to be wrapped in a container to run on Docker (the Virtual OS). It is worth mention until now in Docker you have two types of containers first is a Unix-container (Linux/OSX) and second a Windows-container (only for Windows 10 Anniversary and Windows Server 2016 or newer). All kind of containers can run on either Windows, Linux or OSX.

By default Docker will use Unix-Containers, so if you would like to use a Windows-Container (mostly .Net applications and MSSQL-server), you need to configure Docker to switch to run Windows-Container, though on Windows-computers this will require to let your computer run Hyper-V. When

you switch to Windows-Container, it will modify your Windows to use Hyper-V, if you have enabled virtualisation in your BIOS. Unfortunately, by enabling Hyper-V you will disable the possibility to run VirtualBox – So you cannot run VirtualBox and Docker at the same time! You should be able run them in parallel at Linux or OSX computers.

For Docker you create a Container (an image), which you can download and start in as many instances you like. Except for the fact that your computer have limited resources so, you cannot start a whole server farm on your labtop☺.

You will have access to a long row of already existing containers i.e. images. You can find them at https://hub.docker.com/, wherefrom you can download images (containers) in two ways.

1. Manually by typing commands in a 'command terminal',
    e.g. **`"prompt$> docker pull <<image-name>>".`**

2. Using a tool called **`'Kitematic'`**, this tool is will be installed together with Docker on a Windows computer.

At this site you have quite a lot images and new images (containers) are continuously uploaded even Microsoft has many images for their NanoServer (mini Windows Server), MSSQL-server, IIS-server et al. You can also find many images for Unix e.g. Apache-Server with or without PHP support, MySQL-server, several Linux systems, Mail-servers etc.

Literature used for studying Docker

Soft Overview Wikipedia

- **https://en.wikipedia.org/wiki/Docker_(software)**

Dockers own homepage

- **https://www.docker.com/**
- **Documentation: https://docs.docker.com/**

Nice description of Docker

- **https://opensource.com/resources/what-docker**
  (having a 47 min long video – a little bit long)

Alternative intro (YouTube)

- **https://www.youtube.com/watch?v=aLipr7tTuA4**

Nice series of videos to understand Docker in a wider perspective.

- **https://www.youtube.com/playlist?list=PLtWgqZUXAsNhMnuVgkP-nIuuKrRHuVF4Q**

Series of videos for Docker with Windows containers
If you are to design and implement you own windows containers:

- **https://www.youtube.com/watch?v=7ra_tB2q0O0&list=PL6tu16kXT9PrTeP07thlsrF8Sf9zH Xmh5**

Fine hands-on tutorial

- **https://www.katacoda.com**