Medicon Engineering Themes
Volume 5 Issue 1 July 2023
Article Type: Research Article
ISSN: 2834-7218

Unlocking the Potential of Docker Containers: A Journey into Performance Enhancements and Benefits

Dheeraj Gupta*, Shubham Sharma, Abhishek Chaurasiya, Vicky kumar and Indresh Kumar Yadav

Department of Electrical Engineering, GNIOT, India

***Corresponding Author:** Dheeraj Gupta, Department of Electrical Engineering, GNIOT, India.

Received: June 12, 2023; **Published:** June 28, 2023

DOI: 10.55162/MCET.05.148

Abstract

In this paper, we delve into the realm of cloud computing, which has experienced exponential growth due to its ability to rapidly deliver IT services at a minimal cost. Virtualization has played a crucial role in enabling the cloud model, but it is worth noting that virtual machines tend to be slightly heavier compared to containers. This is primarily due to the use of a hypervisor on the underlying operating system, while containers utilize a lightweight container manager. With the abundance of alternatives available for Docker containers today, our objective is to elucidate the concept of Docker containers and conduct a comprehensive performance comparison with its alternatives. Our aim is to provide individuals and organizations with valuable insights, enabling them to select the optimal virtualization platform that aligns with their specific requirements.

Keywords: Docker; Docker Container; Virtualization; Virtual Machine; Cloud Computing

Introduction

Cloud computing change the way we are accessing computer today as an individual or as an organization. Virtualization can be achieved through hypervisor as well as through Container engine, but container engines are light in weight as compare to container engine. Docker is an open source platform used for developing, shipping, testing and running application. Docker separate the application from the underlying infrastructure which make it quick to develop, test, manage and share others. Docker is used by developers and administrator. Docker containers keep running on top of the kernel's operating system like Virtual Machines.

Virtual Machine

Virtual machine is capable of emulating a virtual environment. It has a server called virtual server named as hypervisor. A machine virtually mimic on the physical hardware of system to mirror the exactly same environment on which our application was installed as shown in fig 1.

Docker

There is a need of providing solution for software development to Docker Container which is an open source platform that provides means to various alternatives towards development of software. Container is stand-alone, lightweight and self executable bundle of software that has all included required to run itself. Containers have property which make them isolated from each other and they are considered to be not dependent on any platform. Components in Docker made it possible to execute it inside a virtual machine if required by any application. The main purpose of Docker is that it allows you to run microservice application in a distributed architecture.

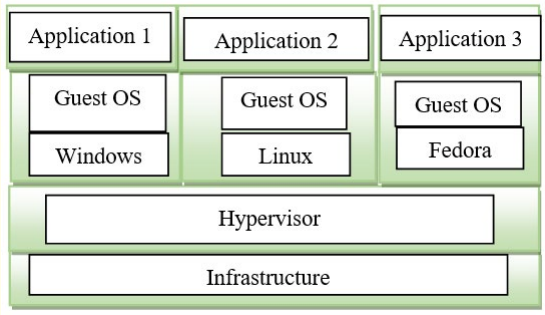

Figure 1: Virual Machine using Hypervisor.

Docker is known to provide additional facility of extra levelling on engine called deployment engine above the container, which generally are only virtualized and eventually executed. Dockers not only provide a light weight and speedy platform but also provide the functionality of extracting the code before it is handed over to the production environment.

Docker Architecture

Docker's Architecture is based on client-server model. It is slightly more complicated than VM because it has more features involved then VM. It has four main parts as shown in fig 2:

- ***Docker Client***: It is the user interface for Docker. Docker Client help user to interact with Docker container.
- ***Docker Objects***: Docker Objects are main components of Docker: containers and images.
  - ✓ Containers are placeholder for software and can be read and written to.
  - ✓ Images are read-only used for creating new containers.
- ***Docker Daemon***: It's a background process responsible for receiving commands and passing them to containers via command line.
- ***Docker Registry***: Its commonly known as Docker hub, it's a registry where all the images are stored and retrieved.

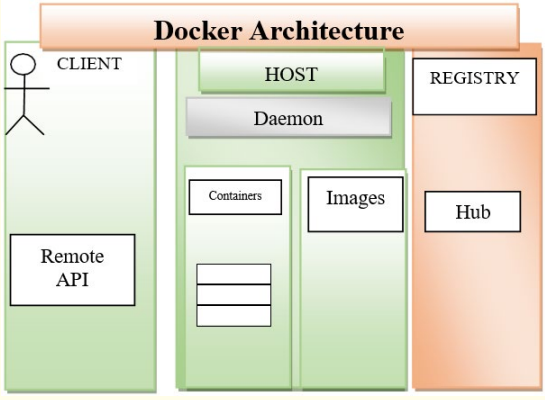

Figure 2: Docker Architecture.

Virtual Machines vs Docker Containers

A hypervisor is software or hardware that build and runs Virtual Machines. It's what act as a interface between the hardware and the virtual machine and therefore it is necessary to virtualize the server.

Within each virtual machine unique guest operating system can be run in isolated way. VMs with different operating systems are able to run on the same physical hardware as shown in fig 3. Each VM has its own binaries, libraries, and applications that it services, and the VM may take memory in many gigabytes in size.

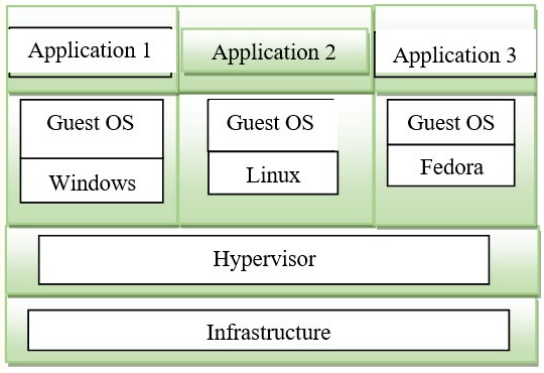

Figure 3: Virtual Machine.

Containers are the topmost layer of the infrastructure and is situated above the operating system like Linux or Windows. Due to sharing the underlying operating systems, these containers share kernels, binaries and libraries. These shared components are read only. Therefore Containers are very light and they take memory in megabytes which is very much less as compared to VM.

Containers reduce overhead because when containers are used only a single operating system is taken care of which make it light in weight as shown in fig 4. Less Bug fixes because single OS is used. Instead of Virtualizing the whole system Infrastructure Containers share the common OS which make it more portable than Virtual Machine.

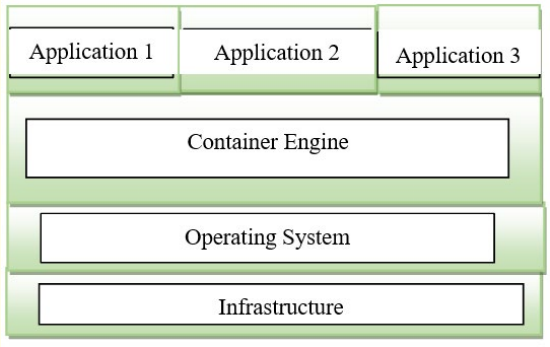

Figure 4: Container.

The main difference in VM and Container is that VM sit on physical Infrastructure and virtualize the physical infrastructure which help to run multiple instances of OS on single machine whereas Container sit on the top of host OS and common OS is shared by different containers are light in weight, more in speed, agile, and portable therefore it is best suited in streamline software development.

Advantages of Docker Container

Docker has become very popular within no-time, because of the advantages provided by docker container over VMs. The reason why docker is preferred over vm is that it provides rapid application deployment, lightweight footprint and minimal overhead, portability, simplified maintenance and version control.

Rapid Application Deployment

Containers provide rapid application deployment strategy because containers include minimum runtime requirement and reduce their size which increase the speed of application deployment. The containers are easier to build is it is smaller in size and require lesser time.

Sharing

Docker provide remote repository which help in sharing docker application with others.

Docker is very popular containerization tool which provides software application which contains everything that's need to run.

Simplified Maintenance

Docker reduce efforts and risks associated with application dependencies. It prevent unauthorized removal.

Establishes a process for maintenance personal authorization and maintains a list of authorized maintenance organizations or personnel.

Portability

Portability define that an application and all its dependencies can be bundled into single unit that is independent of host linux kernel. This application can transferred from one machine to another which run docker.

Disadvantages of Docker Container

Docker Container become very popular swiftly but it has some drawbacks also. In this section we discuss drawback of docker container which are listed below.

- Docker Container are not best suited for Graphical applications. Docker is designed as a solution to deploy on server which doesn't require graphical application but some time there is need to run such applications (like X11 video forwarding).
- Docker Containers are not able to provide high performance as bare metal. It is true that containers consume resources more efficiently than virtual machine but these are still a concern of performance.
- Currently Docker doesn't support older machines, It support 64 bit machines.
- Persistent data storage is complicated in docker container, whenever the docker container is shut down all its data is lost therefore before shutting down the docker container we have to store the data somewhere else from docker container. There are ways to store data persistently like Docker data volume but it still a challenge which has to be address.

Docker vs Other Containers

Docker containers has lots of alternatives. Below are some of alternatives mentioned with its comparisons.

Docker vs rkt

Rkt is titled as a rocket. When compared to docker rkt it's the prime focus remains. Therefore it is necessary for it to support multiple container format including docker images and its own format. public cloud portability, stateful app migration and rapid deployment are use cases for rkt.

Docker vs KVM

There are conflicting differences when it comes to working of Docker vs KVM. The issue can be explained better by considering the example of two servers having same configuration in a cloud environment. The two servers that were being used for an Open Stack platform for KVM by means of a virtualization tool and the other one was for Docker.

In order to monitor the performance of both the system(server), 20 images were generated on each server and the boot time was observed. Figure 5 shows that the boot time of KVM is quite more than docker.

The difference between KVM and docker is that KVM use guest OS whereas docker use host OS.

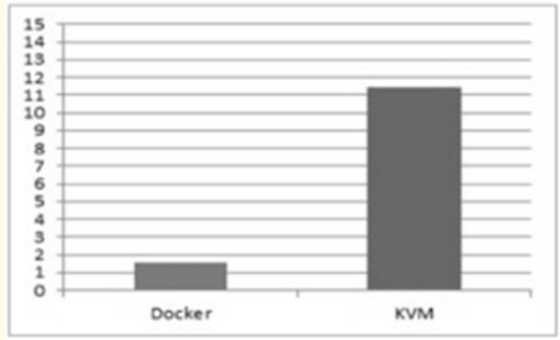

Figure 5: Docker vs KVM Average boot time.

To compute the working speed, python language was used. Figure 6 shows that working speed of 100,000 is averagely around 4.5s. To measure the operation speed, they obtain the average process time and standard deviation, by repeating the same process 100 times on docker and VM.

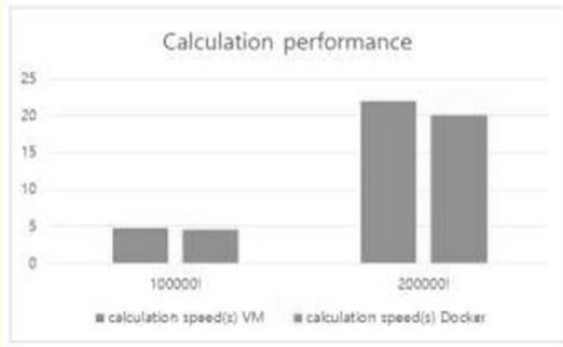

Figure 6: CPU Calculation Performance [8].

Figure 6 depicts that docker is faster as compared to virtual machine [7]. Seo et al. (2014) lays out the idea that virtual machine are independent in their working scenarios [8]. Due to this independence virtual machines are highly manageable with the policies related to network, security, user, and system. However docker do not support any guest operating system and hence time taken by docker is very less in distributing and gathering images. It has also been draw to attention that the booting time od docker is low. These are some of the main advantages of using docker cloud as compared with Virtual machine Cloud.

Docker vs LXD

As compared to Docker LXD it emulates the experience of operating a virtual machines in terms of container and perform this without the expense of emulating hardware resources.

Docker vs openVZ

Compared to Docker openVZ is a extension of the Linux kernel, which provides tools for virtualization to the user. It uses Virtual Environments to host Guest systems, which means it uses containers for entire operating systems, not individual applications and processes.

Docker vs LinuxVServer

Compared to Docker LinuxVServer use a patched kernel to provide operating system-level virtualization features. Each of the Virtual Private Servers is run as an isolated process on the same host system and there is a high efficiency as no emulation is required. However, it is archaic in terms of releases, as there have been none since 2007.

Conclusion

Docker revolutionizes the application deployment landscape by introducing a lightweight containerization approach. While virtual machines rely on hypervisors and guest operating systems to virtualize the entire hardware stack, Docker leverages container engines that add a minimal layer of isolation directly on the host operating system. This fundamental difference results in significant performance benefits for Docker containers. By eliminating the need for a guest operating system and reducing resource overhead, Docker achieves faster application performance compared to virtual machines. The streamlined nature of Docker enables efficient utilization of system resources, making it an attractive choice for organizations seeking optimal performance and resource efficiency in their virtualization strategies. With its ability to provide rapid application delivery and minimal performance impact, Docker continues to empower the development and deployment of modern applications in diverse computing environments.

References

1. G Eason, B Noble and IN Sneddon. “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions”. Phil. Trans. Roy. Soc. London A247 (1955): 529-551.
2. J Clerk Maxwell. “A Treatise on Electricity and Magnetism”. 3rd ed., vol. 2. Oxford: Clarendon (1892): 68-73.
3. IS Jacobs and CP Bean. “Fine particles, thin films and exchange anisotropy”. in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic (1963): 271-350.
4. K Elissa. “Title of paper if known”. unpublished.
5. R Nicole. “Title of paper with only first word capitalized”. J. Name Stand. Abbrev., in press.
6. Y Yorozu., et al. “Electron spectroscopy studies on magneto-optical media and plastic substrate interface”. IEEE Transl. J. Magn. Japan 2 (1987): 740-741.
7. M Young. The Technical Writer’s Handbook. Mill Valley, CA: University Science (1989).

Volume 5 Issue 1 July 2023