Assignment 2: Distributed Operating System Principles (COP5615)

Name: Vikas Chaubey**, UFID:** 3511 5826**, Email:** vikas.chaubey@ufl.edu

During the class Professor discussed about **"Docker"** while discussing processes, threads and Virtualization etc. Since Docker is very useful in practical software applications**, I decided to read few research papers as well as few online resources to increase my knowledge base for this topic.** Here is the summary of what I learned after reading those resources.

Resource references:

1) **Research Paper Reference:**

"An Introduction to Docker and Analysis of its Performance" By:
Babak Bashari Rad, Harrison John Bhatti, Mohammad Ahmadi
Asia Pacific University of Technology and Innovation
Technology Park Malaysia, Kuala Lumpur, Malaysia

2) **Other Online Blogs Posts:**

- https://www.tutorialspoint.com/docker/index.htm
- https://docker-curriculum.com/
- https://docs.docker.com/get-started/

1) what is docker?

It is an open runtime and packaging tool which can be used for building, distributing, and running applications in a portable, lightweight manner. Docker allows the software developers to pack the software application along all of its dependencies into a single unit called containers. Docker makes use of containers in order to run the application on host system. Software developers can create the images of the package and ship it for deployment. On the deployment servers these docker images could be used to create instances of containers which contain the application and all its dependencies. Docker containers are very light weight and can eliminate the use of virtual machine for the software deployment on the host machines which are very taxing on server resources.

2) What is Virtualization?

Virtualization: Virtualization is an old concept, which has been in used in cloud computing. Virtualization is the process of creating a software-based, or virtual, representation of something, such as virtual applications, servers, storage and networks. It is the an effective way to reduce IT

expenses while boosting efficiency and agility for all size businesses. Virtualization is used for deployment of software applications.
For Example: Virtual machines

Virtual Machines: when a virtual machine is set up on a host system, there exists a middle layer which lies between the host operating system and guest operating system(Virtual machine OS), this layer is called the "Hypervisor”. It works between the operating system and CPU. The virtualization divides it into two segments: the first one is Para-Virtualization and the second one is Full Virtualization.

Architecture of virtual Machines:
As we can see in the below diagram, in order to setup a virtual machine on a host server which has its own operating system, A hypervisor is required on top of which guest operating system is installed. This virtual machine or environment could be used to deploy software applications by installing it on the virtual machine and all its related dependencies.

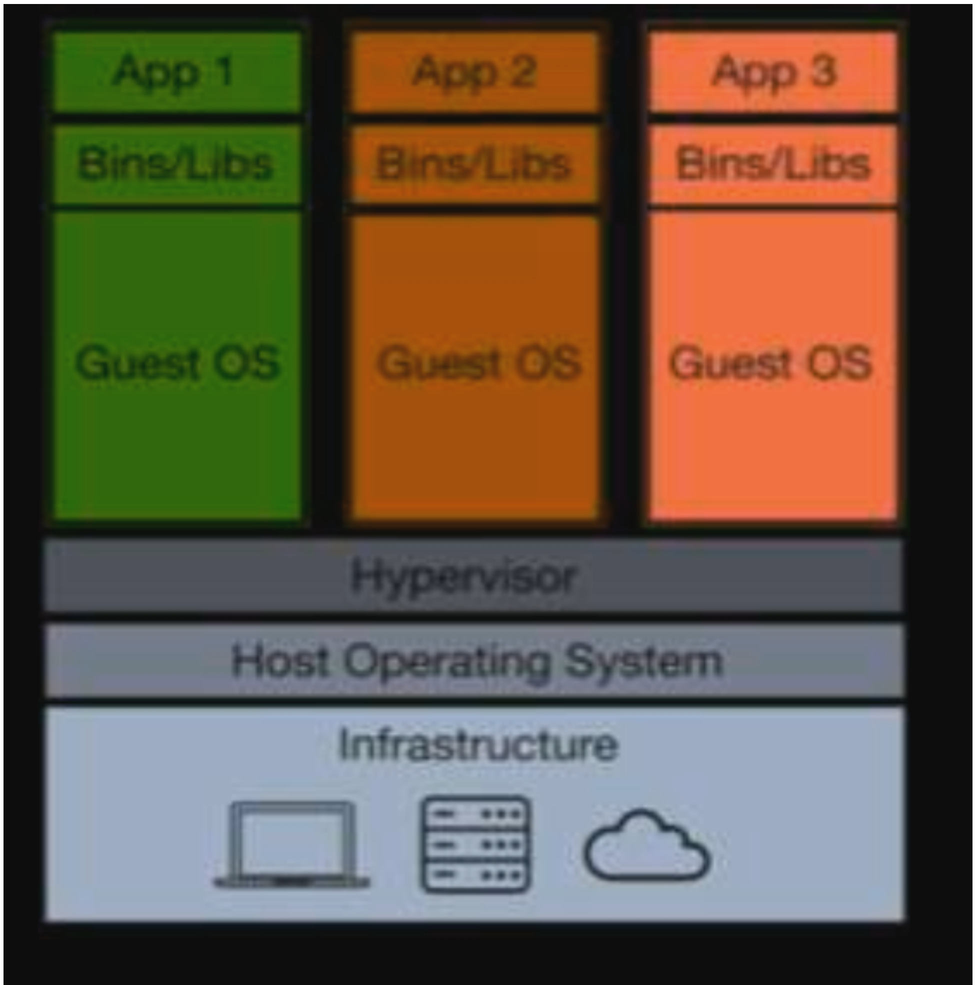

Fig. 2 Virtual Machine architecture [11].

Drawbacks of Virtual Machines:

Even though the virtual machines provide good isolation for software deployment still they are very resource heavy, since virtual machines make use of guest operating systems, these guest operating systems use the host server's resources extensively. Hence it is not possible to setup many virtual machines on a single hots system because of limited resources available. Hence on a single host machine it's not possible to run many software applications deployed using virtual machines because of resource limitations.

3) How Docker Works:

Docker allows the software developers to pack the software application along all of its dependencies into a single unit called containers. Docker makes use of containers in order to run the application on host system. Containers are very light and consume a lot of less system resources compared to a virtual machine. As opposed to virtual machine containers do not require a hypervisor on host system to communicate with host machine operating system. There are four main internal components of docker, including Docker Client and Server, Docker Images, Docker Registries, and Docker Containers.

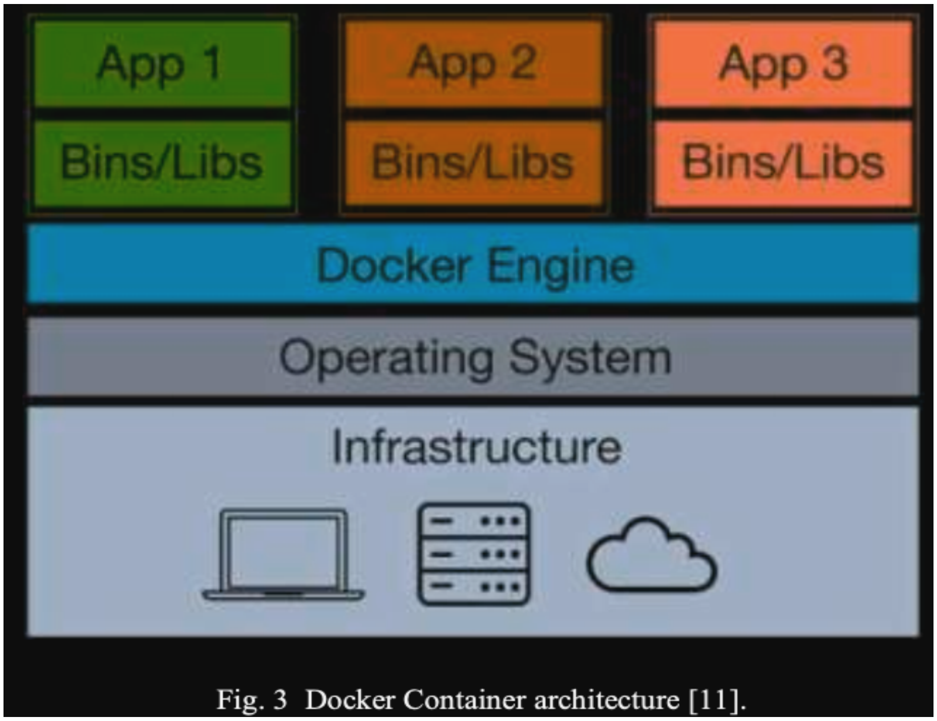

Fig. 3 Docker Container architecture [11].

Docker Client and server: Docker follows a client server architecture which constitutes a docker daemon running on the host system, this docker daemon is the main process which is responsible for creation of docker images of docker containers. This docker container can be accessed by a docker rest API, A docker client can access the docker daemon using this Rest Api.

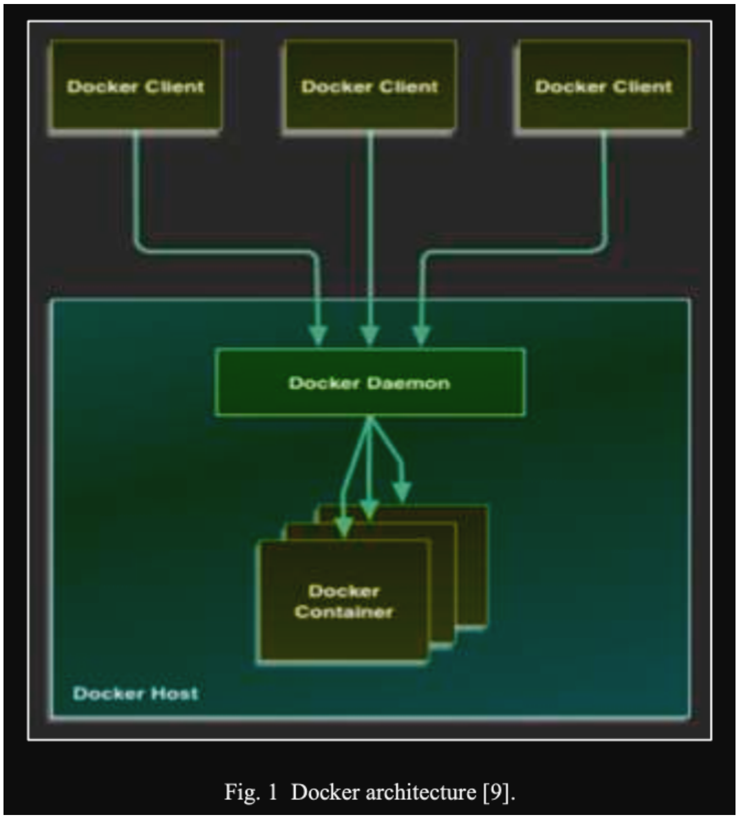

Fig. 1 Docker architecture [9].

Docker Images:

A Docker image is a read-only template that contains a set of instructions for creating a container that can run on the Docker platform. It provides a convenient way to package up applications and preconfigured server environments. When a developer builds a software application the application and its dependencies could be shipped to testing or deployment environments, where this image could be used to create instances of containers which actually run application on the host system. In order to build a docker image a base image is required; the base image is basically a operating system image such as ubuntu and fedora etc. These base images can create a container with capability of running Operating system. The second method is to create a docker file. The docker file contains a list of instructions when “Docker build” command is run from the bash terminal it follows all the instructions given in the docker file and builds an image. This is an automated way of building an image.

Docker Containers:

A docker container is an instance of docker image. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. to run a docker container a hypervisor is not required, these docker containers are standard that means they could be portable anywhere also docker containers are very lightweight which means containers share the machine's OS system kernel and therefore do not require an OS per application, driving higher server efficiencies and reducing server and licensing costs. Docker containers provide good isolation among different applications running on the same host machine.

Docker Registries:

Docker registry is a stateless and highly scalable server-side application where Docker images are stored. It is similar as source code repositories the only difference is that docker registries save docker images. It could be used to push and pull docker images from a single source point. There are mainly two types of registries public and private. Docker hub is a public registry where everyone can pull available images and push their own images without creating an image from the scratch.

Advantages of Docker Containers:

1)The time required to build a container is very fast because they are really small. Development, testing, and deployment can be done faster as containers are small.
2)The applications built using docker containers are extremely portable that means they could be moved as a single element and their performance remains unchanged on different host systems.
3) Docker provides very good scalability, Docker has the ability that it can be deployed in several physical servers, data servers, and cloud platforms.
4) docker applications could be rapidly delivered among development, testing and deployment environments.
5) Since docker containers do not use hypervisor hence it can make use of available system resources more efficiently, the performance of a Docker Containers is higher because of higher density and no overhead wastage of resources.

Disadvantages of Docker:

1) Docker does not provide complete virtualization as it makes use of Linux kernels which are managed by local host.
2) Docker does not support the old machines; it works only with 64-bit systems.
3) Even though docker containers provide good isolation, the isolation provided by docker is not as good as Virtual machine isolation.