Comprehensive Review of Virtualization Tools

Akarshan Agarwal[1], Shreevaths K. Satish Rao[2], B.M. Mahendra[3]

[1]Student, Department of Electronics and Communication Engineering, R.V. College of Engineering, Bengaluru, Karnataka, India
[2]Student, Department of Electronics and Communication Engineering, R.V. College of Engineering, Bengaluru, Karnataka, India
[3]Assistant Professor, Department of Electronics and Communication Engineering, R.V. College of Engineering, Bengaluru, Karnataka, India

---***---

Abstract - *Virtualization is process of creation of hardware or operating systems and processes virtually on the underlying hardware which can be a local machine or cloud. It is a growing technology and is widely used by IT companies for development of software as it allows full and optimal use of resources. In this paper we are going to explain virtualization and its different aspects and perform a comparative study on the most widely used virtualization tools in the market. Docker is platform based service which implements OS level virtualization and delivers software as packages in containers. Kubernetes is a container orchestration tool which is used for automated software deployment, scaling and management. Vagrant is a platform for construction and management of virtual environments. VirtualBox is an open source virtualization software for creating virtual environments. This paper details the use of these tools and states a comparison between the similar softwares while drawing conclusions on the combination of their implementation.*

***Key Words*: Virtualization, Docker, Kubernetes, Vagrant, VirtualBox, Virtual Machine**

1. INTRODUCTION

In the current industrial scenario data management and storage is a huge problem. IT Companies are spending a fortune on these aspects. As the storage demand rises, companies are looking for an efficient and secure way to store their applications and data within their budget. There are only two possible solutions, that is, Dedicated and Virtualized Servers. Dedicated Servers also known as bare metal servers are physical storage arrays that store the data of a single client. Virtualized server hosts the data of multiple clients which allows multiple clients to run independent from each other with the same underlying hardware.

Virtualization is an emerging technology which is used my number of IT companies to manage data storage issues and make their applications more flexible and scalable. In computing terms, virtualization basically is the virtual creation of a resource of device for example a server or an operating system or a network. Main advantages of virtualization is high availability for critical application and streamlined application deployment and migration.

G. Sunitha Rekha *et. al*. [1] talks about Virtualization, its demand, benefits and vulnerabilities. She has explained the different types of virtualization and their benefits. She has the listed out their vulnerabilities and concluded stating that virtualization is not inherently insecure but all the vulnerabilities need to managed and in doing so makes virtualization a must use asset.

This paper presents a comparative review of the most commonly used virtualization softwares in the software industry today. The comparison and analysis has been done on the features each software provides and their relevant benefits. The structure of the paper is, the following section gives a detailed insight on the different types of virtualization, the various benefits and the importance of using virtualization. Section 2 explains the most commonly used virtualization tools in the industry in brief and their some working concepts followed by their comparative review in Section 3. Section 4 deals with the conclusions and drawn from this study.

1.1 ASPECTS OF VIRTUALIZATION

Virtualization is commonly believed to be the creation of an abstraction layer over physical hardware known as Virtual Machines (VM). Simply put, a virtual machine is a virtual representation of a physical computer with its own operating system and cannot directly interact with the underlying hardware. It needs a layer known as a hypervisor as seen in fig. 1. Hypervisor is responsible for allocation of resources like storage and computing power to the virtual machines. But this is hardware virtualization.

Another form of virtualization is process virtualization which are mainly used to virtualize and automate processes using containers as shown in fig. 1. A container is a virtual runtime environment that emulates an operating system rather than using the underlying hardware's operating system. The containers are managed by a container engine which deploys containerized applications, allocates memory, and manages security and implements spatial isolation. So simply put virtualization can a Virtual Hardware or a Virtual Operating System. A hybrid container Architecture is the combination of both and is commonly used for application development.

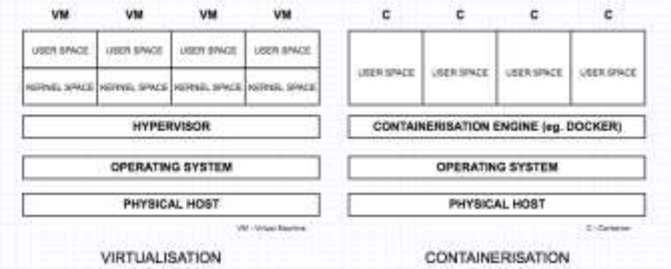

Fig -1: Aspects of Virtualization

1.2 IMPORTANCE OF VIRTUALIZATION

Virtualization Technology has become an important part of development and hosting of applications due one major benefit which reduction of costs. Using a non-virtualized environment is inefficient as a lot of computing power of the physical hardware can go to waste. Virtualization reduces downtime of the applications due to any physical damage to the hardware and increases their resiliency by easily replicable environments for hosting. Ideally suited for DevOps as development and testing can be carried out on the same machine without deployment on cloud. One of the major benefits is the eco-friendly aspect as it has considerably reduced the carbon footprint from the data centers. In *et. al*. [2] Kumar Rakesh and Charu Shilpi have discussed the benefits and importance of virtualization in cloud computing with a detailed pros and cons view. Through their study they have drawn a conclusion of the importance of virtualization and their increasing use by the IT industry.

2. VIRTUALIZATION TOOLS

Multiple tools are available today that offer functionalities and feature that we have discussed above. Often a tool becomes more popular due to its ease of use and multiple features that it offers, we are going to discuss some such tools in brief which are available for open source usage.

a. Docker

The containerization is the method of implementing the virtualization at the operating level of the physical hardware. Virtual machine brings the abstraction at the hardware level but the containerization will implement the abstraction at the operating system level. Containerizing the applications can bring more efficiency because there is no concept of guest OS which utilizes the resources of the host Soothed binary files and library related to the application are run on the host kernel which provides faster way of execution.

Docker by Docker, Inc. is an open platform for developing an application, deploying it and running on the remote distributed platform along with the packaged dependencies. Docker permits to run the application in the isolated environment which comprises of all the resources. Docker consist of images, Docker engine and containers to run the software remotely.

Docker image is basically a static view of the service or the dependencies. Images are called as instances of the container. Images contain the code, runtime binaries and configure files. Images are stored in the registry. Containers are built from the images which includes all the packages to run the application. These can be run in active mode or detached mode. And the registry stores all the images.

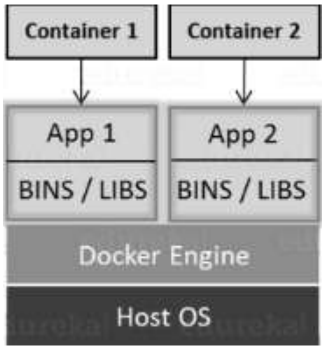

Fig -2: Docker Structure

As we can see in fig. 2 single application will run on single container with its own independent libraries and packages. This states that there is an isolation between each process. One single host OS is been used by the Docker engine to provide the services to multiple containers.

Some Features of Docker:

- Containers provide the isolated environment with all the binary files are wrapped inside it.
- Docker containers are light weighted which provides the advantage of scaling.
- Docker provides the tool to cluster the images stored in the registry called as swarm. Swarm works on the basis of the Docker API.
- It provides the routing mesh option to guide the incoming request to the published ports.

Docker Swarm is a network of interconnected Docker containers. It is an orchestration tool which helps in deployment and management of multiple containers across multiple machine. It is governed by the Dockers API.

b. Kubernetes

Kubernetes is an open source container orchestration which is used for automation of scaling, deployment and management of applications. It was originally designed by Google and is no managed by Cloud Native Computing Foundation. This tool focuses on optimization of resources like memory through containers which are used for efficient distribution of application across a cluster of nodes. It is mainly used to schedule containers inside infrastructure instead of direct utilization of containers or nodes.

Some Features of Kubernetes are:

- Grouping of containers using Pods
- The self-healing and auto-scalable capability

- Management of DNS
- Has Load Balancer capabilities
- Ha features to roll update or to rollback an update
- Feature to monitor and log resources

c. Vagrant

Vagrant by HashiCorp, Inc. is a tool for managing virtual machines from command line which developed by Mitchell Hashimoto. It wraps around the virtual machine to provide consistent interface. It is tool for cross-platform solution which gives permission to the end clients to create and configure the environment for Docker containers, VM box and virtual machine, AWS. Vagrant is written in ruby which supports the development in different language. Vagrant has its own version of the virtual machines called as virtual boxes. The repository is maintained for pre-built vagrant boxes. The repository is called as a HashiCorp repository. Vagrant files are main components in deciding which type of machines will be built. These files contain OS configuration files. These files are written in the ruby syntax. Vagrant will isolate all dependencies and their configuration inside single consistent environment. Working with vagrant stats with downloading a virtual box from a repository. Then the vagrant file must be placed inside a project folder before starting a virtual machine. Then developer has to connect with the running virtual machine with the command line.

Some feature of Vagrant:

- It enables a developer to create multiple cloned and isolated environment which can be remotely distributed.
- Automation feature of vagrant enable the creation of virtual machine to the virtual box.
- The environments created by vagrant are portable which means they can be reproduced on any system.
- It allows you to implement multiple simultaneous virtual environment with single Config file.
- Synched folders will users share resources between virtual machine and host system.

d. VirtualBox

VirtualBox by Oracle Corporation is an advanced, free and open source hosted hypervisor which is used for x86 virtualization. It is a hardware assisted virtualization tool which enables users to load multiple OSes under a singular hosting operating system. Each hosted VM is configurable and can also be put into networks which provides connection between the hosted OSes and the hosting OS.

Some features of Oracle's VirtualBox are:

- Highly portable and capable of running on both 32 and 64 bit environments
- Does not require hardware virtualization and can run on older hardware also
- Has A VM groups feature allowing users to control the VM as a group or individually
- Guest Additions and Shared folder feature
- Remote machine display and multi-generated branch snapshots

3. COMPARATIVE STUDY OF VIRTUALIZATION TOOLS

We have seen above that virtualization can be broadly classified in two distinct categories. So for comparison we will be comparing the above mentioned softwares with each other in each category.

3.1 PROCESS BASED VIRTUALIZATION

A number of software options are available in the market for container orchestration, but the major players are Docker and Kubernetes. Both are open source platforms providing much of the same functionality, there are some fundamental difference in their way of operation.

Installation and Configuration of cluster: Docker installation is easy and fast. With Swarm deployment of nodes and joining it to a cluster is all there is to do. Also, the Swarm is flexible in allowing a node to join an existing cluster or a new cluster as a manager or worker and seamlessly switching the roles. Kubernetes comparatively requires more time to setup as many manual configurations are need to tie its components together. Also, a number of information is needed for the configuration of clusters. These include roles of the nodes, the number of nodes present and the IP addresses of the nodes.

Setting up of containers: The functionality is provided in Docker Swarm API but since it uses the same functionality from Docker and does not encompass all of its commands, there is no easy way of carrying out an operation whose functionality is missing in Docker API. Kubernetes has its own client, API and YAML definitions unique to it and differ from standard equivalent Docker CLI. Thus, Docker compose cannot be used to define containers and platform specific YAML definitions and commands are required.

Scalability and Availability: Docker swarm has a comparatively higher deployment speed for containers, even for huge clusters and have fast reaction to scaling demand. Replicas of these clusters can be deployed with a single command. Docker Swarm has high availability with the same container used to deploy multiple nodes and copy of containers to ensure service is running if a certain container goes down. Kubernetes comparatively is slower in deployment speed which is a trade-off for a complex and unified set of API and guaranteed cluster states. Kubernetes similar to Docker Swarm has high availability which it maintains through replication.

Load Balancing and Service Discovery: Docker Swarm has internal load balancing. The containers and all the nodes in it are connected through a network which allows connection from any node of a container to another. This network allows communication through virtual private IP addresses and service names making service highly discoverable throughout the swarm. Kubernetes also allows load balancing by setting up load balancing pods which are containers defined as services. Policies can be defined to setup a load balancer. These container pods when defined are stored in a key value store which is used for service discovery. The other option is manually making the services discoverable.

Data Volumes and Networking: Docker directories are data volumes which can be shared by one or more containers. The directories exist even when the containers using them are deleted. The volumes are local but can be made global using the volume plugins. The network configured on Docker is automatically secured through TLS with certification. Kubernetes volumes are similar to Docker and shared within container pods with one major difference being their lifetime which ends when the pods are removed. The network on Kubernetes if through etcd and require manual configuration and installation of certificates for TLS security.

Through the comparison, we have noticed how the Docker Swarm and Kubernetes are similar but fundamentally differ. Swarm is more inclined towards a user's ease of use and its integration with Docker while Kubernetes aims to remain open and modular. Similar difference is present in both the software's installation and configuration. To conclude it is safe to state that Docker aims at providing fast and simple solution and is preferred by developers looking for simplicity mainly while development of applications. Kubernetes on the other hand provides support for complex use cases and supports higher demands and is generally employed for production environments.

3.2 HARDWARE BASED VIRTUALIZATION

The heavily used and older type of virtualization, many softwares are available which perform the necessary setup required. The two widely used softwares are VirtualBox and Vagrant. While they both essentially provide the same functionality they have a considerably different way of working and many developers recommend using a combination of both for the best virtualization experience.

Vagrant can briefly be defined as a tool for construction and distribution of development environments. It provides framework and configurations for creation and management of completely portable environments. These environments can live in the cloud or on the local system. VirtualBox on the other hand is powerful virtualization product rich in features and performance. It mainly runs an operating system on a single machine and allows free switching between the OS instances running. To be precise Vagrant is a Virtual Machine Management tool while VirtualBox is a Virtualization Platform.

Vagrant's major benefits are boxes, fast deployment of environments and SSH and synced folders while VirtualBox is highly portable and does not require any specific configuration for hardware virtualization. It has shared folders, 3D virtualization and seamless windows. Both have an open source version available. In the current scenario Vagrant is commonly used for development of applications and VirtualBox is used with Vagrant for better management and provisioning provided by Vagrant. But Vagrant quickly eats up the space as the environments number increases and has a huge impact on battery life in local machines. For these reasons, isolated developers prefer VirtualBox setup which requires more effort in setting up but is fairly easy to use and manage while Vagrant is used in IT world for development highly scalable and flexible applications.

4. CONCLUSION

In this paper we have carried put a general comparative review of the types of virtualization and the most commonly used softwares respectively. We can see that each of the software does the job perfectly and possess unique functionalities which make it useful in targeted scenario. Summing up all our points we can conclude that both Virtual machines and Containers are completely different aspects of virtualization and are very important in software development. Often a combination tools from process and hardware virtualization is used for development as well as production environments. Both VirtualBox and Vagrant are necessary tools and the combination of both is widely preferred. Lastly Docker and Kubernetes which are a different brood of virtualization and with the features and functionality they offer, clear choices can be made between the two based on the required use case as mentioned above.

REFERENCES

[1] G. Sunitha Rekha, "A Study on Virtualization and Virtual Machines," International Journal of Engineering Science Invention, vol. 7 issue 5 version III, May. 2018, pp. 51-55

[2] KUMAR, Rakesh; CHARU, Shilpi. An Importance of Using Virtualization Technology in Cloud Computing. Global Journal of Computers & Technology, [S.l.], v. 1, n. 2, p. 56-60, mar. 2015. ISSN 2394-501X.

[3] Docker website and official documentation.URL https://docs.docker.com/

[4] Kubernetes website and official documentation. URL https://kubernetes.io/docs/home/

[5] Vagrant website and official documentation. URL https://www.vagrantup.com/docs

[6] VirtualBox website and official documentation. URL https://www.virtualbox.org/wiki/Documentation

[7] D. Merkel, Docker: lightweight Linux containers for consistent development and deployment. 2014