

# ***Docker for Multi-containers Web Application***

Vivek Sharma<sup>1</sup>, Harsh Kumar Saxena<sup>2</sup>, Akhilesh Kumar Singh<sup>3</sup>

<sup>1</sup> Assistant Professor, GLAU Mathura(UP) India

<sup>2</sup> Student, GLAU Mathura(UP) India

<sup>3</sup> Assistant Professor, GLAU Mathura(UP) India

{ viveksharma.cea<sup>1</sup>, harsh.saxena\_ccv17<sup>2</sup>, akhileshkr.singh<sup>3</sup>@gla.ac.in }

**Abstract.** Open Source Software are gaining popularity over licensed software and Docker is one of them. Docker has many advantages when compared with virtual machines environment. This attracts developers for building micro services that will run over containers. The demand in the market for docker containers is rapidly increasing. Light weight and speed are the factors due to which docker outperforms virtual machine environment. The time required for starting and running a docker container is very less when compared to virtual machine. Different docker containers shares the resources provided by the operating system via kernel. On the top of the operating system kernel, the entire container is executed in a confined way. This paper considers the advantages of docker and proposed a docker deployed applications.

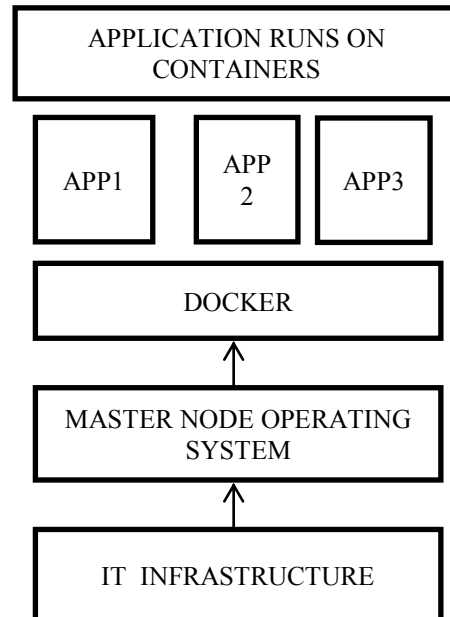
**Keywords:** Container, docker, swarm, docker file, virtual machine.

## I. INTRODUCTION

The container [1] technology is in the market for many years but docker [2] is evolving as one of best innovation in the container technology as it has all the necessary capabilities that provide runtime environment to applications. All the docker application can run anywhere without any change. Even in any cloud environment that is capable of running cloud based application.

Docker is popular due to its high performance; the application can be migrated from one container to another and one host to another host without damaging the behavior and performance of application. The application can be easily deployed and manage in every server. The linux containers [3] are very famous in docker. But docker can do it very easily by providing and maintain the concept of Scalability. This scaling in the docker is done by the user based upon the needs and requirements of users and the organization. Containers are the one cloud platform for developers and others. Same as

cloud computing environment, in docker we do not require to worry about the server management and Programming and how and where the servers are managed, we only require to work inside the container and maintain and hosting the application which lie in our containers. Working on docker as a developer is effective and simple because the code and the application is build inside the container can be easily transfer from any development and testing environment and also can be transfer to any production environment. These containers can work in any of the environment because they have all the dependencies which are by default build and embedded in this.



**Figure 1. Container Service Software Stack**

In docker, one cannot need any hypervisor, with this docker can use each and every resources in very efficient manner. We can execute, build and run any amount of containers in a single host or a machine this is the main comparison of docker with the virtual machine environment. While using

docker, there is no wastage of resources and all, so provide very high density and all. The linux kernel, a monolithic architecture, is used by docker and docker is completely depends on it, and local host used to provides linux kernel facility to docker. For the windows and Mac machines, docker provides a facility called complete virtualization. Even for the some machines where docker is not able to provide the facility of complete virtualization, one can use and adopt the boot2docker technology for taking up the steps of complete virtualization. The checking of security issues must be done always and time to time when using the docker technology tools.

Related work section highlights the journey of docker and then deploys a new innovative way by which we can use docker technology to work with web application. This paper aims to deploy web application on multiple containers where each container contains separate dependencies to hold the application environment.

## II. RELATED WORK

Docker provides the various ways to automate the application and the resources. An extra layer is added by docker when a container application is virtualized and executed. In the year 2015, Russell confirms that it is possible to test your code and deploy it to the production working environment, Turnbull comes with the statement that anyone can begin with a docker with very easy configurations and docker binary with the Linux kernel. There are four main component provided by docker and these are docker client, docker registries, docker images and docker containers.

Docker server [4] is combined with the docker client. The docker server requested by the docker client and it responds to docker client. Both the docker server [5] also called as docker daemon and the docker client are stay on the same machine and sometime docker client is connected with the daemon which is presented in the other machine.

**Docker Images:** Image is a base for doing anything in docker. It is considered as snapshot of work which comprised in a file and contains all the system dependencies and parameter.

**Docker Containers:** In simple words, containers are the application that is ready and we can start working on that. It is created using the docker image. The application is run in a separate isolated way. It contains everything that is needed to deploy an application.

**Docker registries:** All the docker images are stored and distributed in the docker registry. If an image containing multiple versions then tags are helpful for identifying the image and also its version. A docker repository is a way to organize the docker registries and also we can download the images in our local system using docker repositories. There is another previous concept that had been used by peoples is Virtualization [6]. It is not very much different from docker. Virtualization is considered as the base for cloud computing infrastructure. Docker comes with the platform in which we do need to install very large ISO image or any file to install any operating system or starting any program, in docker we have benefit that all the application, dependencies ,operating system, containers and images are very small in size . Suppose one want to install the Ubuntu operating system in virtual box infrastructure, it needs approximately 2 GB of ISO file depends upon the version but, in case of docker, docker image of Ubuntu is pulled from docker hub i.e. approximately 0f 150MB of size.

In the year 2014 Seo et al, uses the cloud computing platform and deploy two servers with both of them having the same configurations. Out of both the two server, first server uses docker and other uses openstack. But the concept that VMs are not dependent on others machine and works independently, so in openstack we can easily configured, deploy and manage networks but when comes to the concept of docker, docker do not require any guest machine and directly runs on the host operating system. So docker is very quickly separate, transfer and gathers images and from here, the main motive of using docker cloud on comparing with virtual machine cloud comes into the existence. This fact also clarifies the performance [7] of docker, which also includes in the advantages of docker.

Previously the virtualization technology was very famous, because cloud uses virtualization as infrastructure services. The hypervisors is used by the virtualization in which an extra layer has been added and used between any host operating system and the new operating system that we want to install. It's very popular example is using the virtual box. But docker uses the same concept but in a very different manner, instead of adding extra layer between host operating system and the operating system that we want to install called the guest operating system, we will introduce a new layer called the Docker Engine [8]. It is used between host operating system and the application that we want to runs and virtualized. And this only becomes possible because the guest operating

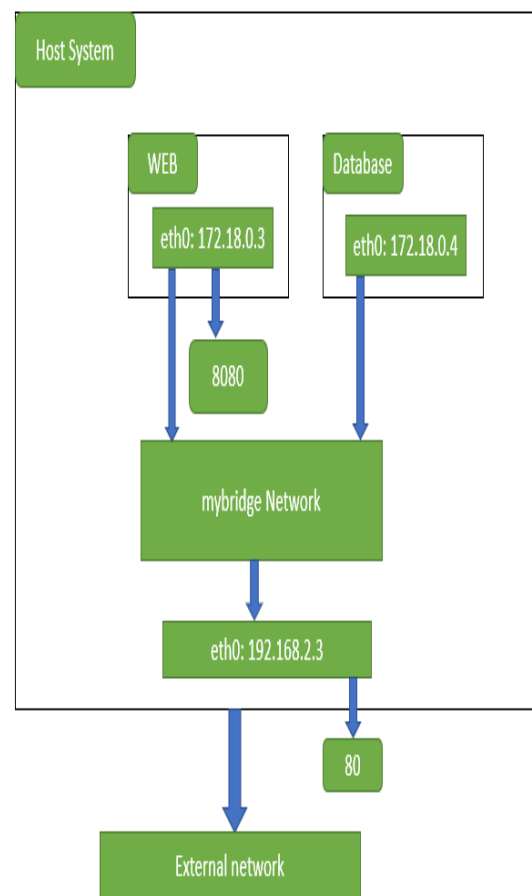
system is not used by the docker but it is used by the virtualization, which helps in enhances the performance and speed of docker technology. In case of performance, in the year 2014, Seo also mentioned that the KVM is less effective than docker and docker provides quite much high performance in comparison of using the KVM and also mentioned that in terms of boot time and calculating speed , docker is much more efficient than using the KVM.

### III. PROPOSED APPROACH

This paper sketches the work of a web application i.e. deployed on multiple container environment and then the work done by all the container is merged. All the web frontend related work is executed on one container and all the database related work is done in other container. Also, the feature of docker networking and compose [9] is used to provide the network between containers and orchestrate them. It can also be done by making this architecture highly available using the docker swarm cluster [10]. In docker swarm, load balancer is used to balance the load between different containers. Among them one container is considered as master container while other are considered as slave container. This web application can be accessed using browser or can be accessed through some other container.

Web application deployment on docker can be done in many ways. It depends on the user requirements. If the requirement is to deploy a static website, one can directly use a linux image or any other container image and deploy the static website directly by using port binding. If the scenario comes that deploy application when all the database and the application frontend are on the same container, the one can install an docker image and make container of it and doing the port binding for accessing it with the host operating system web browser and then deploy any server according to the user requirements and needs and deploy the application on it. It can also be done using Docker file [11] which will make it easier to deploy the application. In the Docker file, all the dependencies are written that includes the base image and here user can directly write the command to install the desired server to run the application and also apply the port binding in the same container.

In the docker swarm methodology [12], the main advantage is that, docker swarm is basically used for providing high availability and scalability and load balancing of application, services and resources when it is required. Basically on using the docker swarm, a user needs at least two operating system running as both guest on a single host or any other ways. This configuration is called master slave configuration [13] or multi node cluster [14]. As many services can be built in docker swarm cluster and all are connected the services using a single network i. e. also known as default network or bridge network, overlay network.



**Figure 2 Deployment of Multi- container applications using Docker**

Fig 2 defines the method and behavior of the containers and the application where a host operating system like windows, linux, centos, RHEL, etc. then two separate containers were created which can also be considered as services in terms of docker swarm clusters. In the first container database is deployed while in second container web application is deployed. After this

two containers are created and separate IP will be generated for both the containers. Services or containers are exposed to any port. Web container is exposed to port 8080. Both of these containers are connected using a bridge network which is having the port 80. And using this port 80 and the IP address, the application can be access by external networks like host system, guest system, slave node, master node, etc.

There can be another methodology we can use to deploy these type of multi container application by which one can access it publicly throughout any network. By creating an instance on any of the public cloud platform like Amazon Web Services, Google Cloud Platform, and Microsoft Azure, etc.[15,16] then install docker in it and using docker compose or the docker swarm facility and follow the same steps of deploy multi container application using docker swarm or docker compose mentioned above. Other way is that, most of the cloud providers provide the mechanisms and services of orchestrations for management of deployment of applications.

#### IV. CONCLUSION

Docker provides the capabilities to automate the containerized application. It provides simple and effective workflow and an additional layer has been added in the host operating system. Deploying web application on docker containers are quite easy, and provides inter operability and portability as well. It is very cost effective and time saving solution in the IT world. Using docker networking, docker swarm and docker compose; it is very simple to deploy a multi-tier web application on docker. There are less number of resources consumed by docker as compared to the virtual machines, so they have better speed and performance. Deploying a multi container web application on docker is very effective and easy instead of using any other cloud computing platform or infrastructure. Migration of containerized service is quite easy as compared to migration of virtual machine. In case of docker, entire code of the web application can be put into a single image and pushed to the docker hub for easily download that can be configured very easily, A container gives a more prominent effectiveness, both in dealing with the managing application creating application instances, and in asset utilization. Basic adjustment and easy deployment of application is one of the most important principal of container services like Docker and which is completely in the DevOps soul.

#### References

1. Russell, B. (2015). Passive Benchmarking with docker LXC, with KVM and Openstack
2. Boettiger, C. (2015). An introduction to Docker for reproducible research
3. Scheepers, M. J. (2014). Virtualization and containerization of application infrastructure.
4. Joy, A. M. (2015). Performance comparison between Linux containers and virtual machines.
5. Boettiger, C. (2015). An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1), 71-79.
6. Bui, T. (2015). Analysis of docker security. *arXiv preprint arXiv:1501.02967*.
7. Felter, W., Ferreira, A., Rajamony, R., & Rubio, J. (2014). An updated performance comparison of virtual machines and linux containers. *technology*, 28, 32.
8. Turnbull, J. (2014). *The Docker Book: Containerization is the new virtualization*
9. RightScale (2008) Define Cloud Computing. RightScale Blog, 26 May 2008. <http://blog.rightscale.com/2008/05/26/define-cloud-computing/>. Accessed: 9 Jun 2009.
10. Howe, B. 2012. Virtual appliances, cloud computing, and reproducible research. *Computing in Science & Engineering*. 14, 4 (Jul. 2012), 36–41.
11. Mohammad Ahmadi, "An Introduction to Docker and Analysis of its Performance", *IJCSNS International Journal of Computer Science and Network Security*, VOL.17 No.3, March 2017.
12. David Bernstein, "Containers and Cloud: From LXC to Docker to Kubernetes", *IEEE Cloud Computing*, Year: 2014, Volume: 1, Issue: 3
13. Sachchidanand Singh, Nirmala Singh, Containers & Docker: Emerging Roles & Future of Cloud Technology, 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATecT), Year:2016, 2016 IEEE.
14. Dmitry Medvinsky, Luke Hutschal, Adrien Magnus", *Managing Containers with Docker Shipyard*", Russia,2015.
15. E. Bacis et al., "Docker Policy Modules: Mandatory Access Control for Docker Container", *Proceeding of 2015 IEEE Conference Communications and Network Security (CNS)*, 2015.
16. A.K.Singh, P.Singh, An Approach for Web based GIS Route Finder System, *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*.