DOI: http://dx.doi.org/10.26483/ijarcs.v9i3.6102

ISSN No. 0976-5697

Volume 9, No. 3, May-June 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

# COMPARISON OF THE TWO GIANTS IN CONTAINER PLATFORMS: KUBERNETES WITH OPENSTACK VS. DOCKER

Vishrant Ojha
Department Of Computer Science and Engineering
Sardar Patel University of Police, Security and Criminal
Justice Jodhpur, India

Arjun Choudhary
Department Of Computer Science and Engineering
Sardar Patel University of Police, Security and Criminal
Justice Jodhpur, India

Krishna Parihar
Department Of Computer Science and Engineering
Sardar Patel University of Police, Security and Criminal
Justice Jodhpur, India

***Abstract***—OpenStack is the fascinating cloud operating system that manages and supervision large association of computer system, storage, and networking assistance through a data handling center [2]. Entirely handle through a dashboard that grants administrators supervision with a web intersection. OpenStack persons unwrap virtual machines and another illustration that carries off distinct works for organizing the cloud arena on the air. It represents the structural orientation measuring convenient, which way works that take advantage direct running coincidentally, can lightly operate more than or fewer persons on the air by simply communicative improving to greater extent instances. For example, a mobile application that necessitates transfusing with a remote server might be able to split the task of communicating with every user across several multiple instances, all communicating with each other but scaling instantly and easily as the application benefits more users. OpenStack is an open source software, which medium that someone who prefers to term approach the source code can make all changes or modifications they required, and independently share these changes back out to the community at the giant. It also means that OpenStack has the advantage of thousands of developers all over the universe working side by side to cultivate the strongest, most robust, and most secure product that they can.

***Keywords***—OpenStack; Docker swarm; Cloud computing; Open source; Kubernetes; Docker.

## I. Introduction

A place where you can access services and apps like files, data from any computer system with internet access is called as a cloud. The process of this mechanism is known as cloud computing. Five essential cloud characteristics as per NIST definition [3] are: a) Broad network access, b) Measured service (pay-as-you-go), c) Rapid elasticity, d) Resource pooling, e) On-demand self-service. Use anywhere, anytime, with any device, accessing any services. Docker is used for container image manufacturing and running the containers whereas kubernetes (K8s) is container orchestration tools. Docker usage itself to run as the container. Kubernetes desired container runs time atmosphere, it helps to organize the container easily. Kubernetes is portable in that it builds by the public, private, and hybrid clouds and extendible via a plug-gable infrastructure also self- healing in that it will automatically restart and space containers through safe and sound nodes supposing a node rarely goes away.

## II. COMPONENTS OF OPEN STACK [4][5]

Open Stack is in open nature that's why is made up of several different moving parts. Anybody may connect moreover components to open stack to help it to converge their needs. Open Stack global organization has cooperatively known 9 fundamental sections that are role "core" of the open stack.

### *A. Nova*

It's also called as primary computing of the open stack. It is used in respect of deploying and handling huge numbers pool of VM's and illustrations to maintain computing actions.

### *B. Swift*

It's a storehouse method for files and objects. This makes scaling simple like developer doesn't worry regarding the efficiency of a single channel at the back of the software.

### *C. Cinder*

It's also called as primary computing of the open stack. It is used in respect of deploying and handling huge numbers pool of VM's and illustrations to maintain computing actions.

TABLE I. POPULAR PROJECTS OF THE OPEN STACK [1]

| **Favourite Project Set Up** | |
|---|---|
| **Project** | **Service** |
| Cinder | Block storage |
| Glance | Image service |
| Neutron service | Networking |
| Nova | Compute service |
| Keystone | Identity service |
| Swift | Object storage |

### *D. Neutron*

It provides the networking capability for open stack. It ensures that each of the components of an open stack deployment can communicate with each other efficiently and quickly in manure.

### *E. Horizon*

It is the dashboard behind the open stack. It's a graphical interface to open stack. After installation of open stack, this may be the first component users actually "see". Developers approach all of the organs of the open stack via an API (application programming interface). The control-board supplies, computer system managers with a view into what's ongoing incoming to a cloud, managed accordingly as per necessity.

### *F. Keystone*

It gives identification works tended to open stack. It is virtually centring indexes of entirely open stack cloud end users, mapped adverse all the cloud services provided, which they have permission to utilize.

### *G. Glance*

It provides image service to open stack. The image refers to virtual copies ("images") of a hard disk. When deploying new VM instances, glance allows the use of these images as templates.

### *H. Ceilometer*

It provides telemetry services and allow to cloud billing services to individual cloud users. It maintains an empirical consideration of each consumer computer system information and what versatile elements, they were using at that time on an open stack cloud.

### *I. Heat*

It is an orchestration component of open stacks. In this developer to keep the need of the cloud action inwards a dossier this determines what instruments are necessary for respect of that application. It assists to handle the system requirement tended to run the cloud employments.

## III. OPEN STACKS AUTOMATES [1]

The OpenStack cloud can automate various techniques.

1. Compute
2. Storage, backup, and recovery
3. Networking and content delivery
4. Data and analytics
5. Security, Identity, and compliance
6. Management tools
7. Deployment tools
8. Application services
9. Monitoring and metering

## IV. OPENSTACK ARCHITECTURE

Open Stack is the cloud software system that supervision giant shares of computes, warehousing and networking instruments end-to-end the data handling center, entirely handled via a control board that allows executives supervision while charging their consumers to provide assistance care of the system convergence [1]. Ubuntu is the platform of preferred for the most management OpenStack clouds, with Canonical [6].

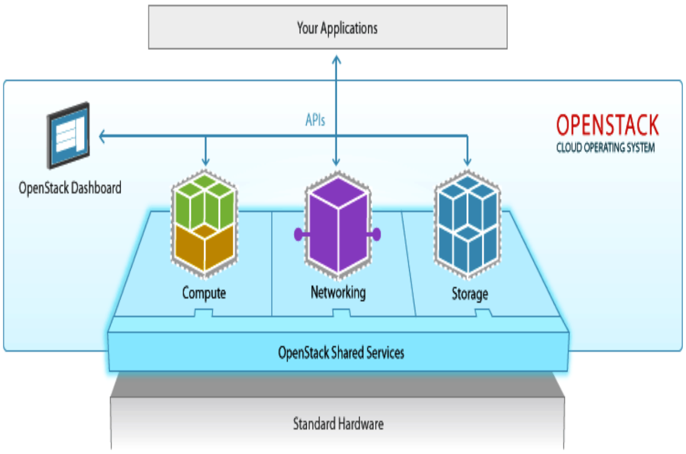

Fig. 1. OpenStack Architecture [8]

OpenStack can't straightaway be established along with computer hardware. It demands software systems which promote virtualization incoming backside. Now Oracle Linux(Xen), Red hat enterprise Linux(KVM), Ubuntu(KVM), VMware ESXi, Oracle, Solaris(zones), Microsoft Hyper-v promotes OpenStack cloud programs. That's the reason OpenStack strategically preference of several varieties of systems, direct service providers impressive to propose cloud computing works along with standardized computer hardware [1].

## V. KUBERNETES

K8s is an open-source bandstand in order to containers deployments manage, scalable, automatism afterward manipulation crosswise groups of recipients [9]. A podium draws up along Google's comprehensive experiment by seasons of functioning with LXC.

Its major objective to supply sections also instruments by following up traffic by functioning experiments inwards private and public clouds to assemblage instruments during logistic units. Its potency lays inwards easy scaling, environment unbeliever portability, and flexible growth. Juju is used to makes easy to deploy container management [7].

## VI. DOCKER SWARM

Swarm represents domestic cluster in respect of Docker. Fundamentally Docker Swarm didn't allow more inwards imagination by container automation, just with an up to date Docker Engine 1.12 [18], container platform represent nowadays settled during it origin by primarily company endorsement.

Docker Swarm represents structured about 4 core group doctrines: simplex already strong up-to-date "just works" consumer taste, a flexible zeroes single point of failure computer architecture, protected by default up to date automatically intuitive credentials and backward concordance by present sections. The liability of rearwards characteristic is specifically essential by the current consumers.

TABLE II. COMPARISONS BETWEEN K8S VS. DOCKER [11]

| Factors | K8s | Docker |
|---|---|---|
| **Installation and cluster configuration** | Takes some work to get up and running | Easy and fast to install and configure |
| **Container setup** | Client, API, and YAML definitions are unique to Kubernetes | Functionality is provided and limited by the Docker API |
| **Scalability** | Provides strong guarantees by cluster state via cost to velocity | Quick container deployment and scaling even in very large clusters |
| **High availability** | Provided through container replication and service redundancy | Same as Kubernetes |
| **Load balancing** | Enabling load balancing requires manual service configuration | Automated internal load balancing through any node in the cluster |
| **Container updates and rollbacks** | Progressive updates and service health monitoring through the update | Process scheduling to maintain services while updating |
| **Networking** | TLS authentication requires manual configuration for security | Automatically configured TLS authentication and container networking |
| **Service Discovery** | Containers defined services are easily discovered | Discoverable throughout the cluster network |

## VII. KUBERNETES ARCHITECTURE

Kubernetes was much attractive in respect of application developers, as it diminished their dependency on infrastructure and operations teams. Vendors also liked Kubernetes; therefore, it provided a suitable way to embrace the container agitation also provide an industrial solution to the operational challenges of running your personal K8s deployment. Kubernetes is open source under the CNCF (Cloud Native Computing Foundation), in contrast to Docker Swarm which, though open source, is reliably controlled by Docker, Inc. [16].

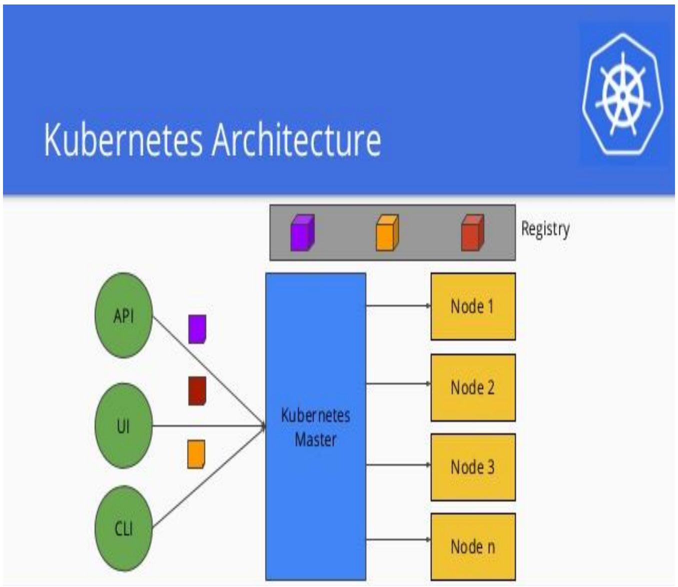

Fig. 2. Kubernetes Architecture [10]

Kubernetes core strength is providing application developers powerful tools for orchestrating stateless Docker containers. As long as there are various initiatives to extend the field of the plan to extra workloads (like analytics and stateful data services), these initiatives are still in very soon phases and it remains to be seen how successful they may be [16]. Kubernetes architecture is in two stages one is a master and another are slaves, connected through the registry like application programming interface (API), User interface (UI), Command language interface (CLI) all are connected to kubernetes master to nodes.

## VIII. DOCKER ARCHITECTURE

The Docker structure presently leverages Linux Containers (LXC) who has characteristics as cgroups and namespaces to inventory supervision, powerful procedure and separation and so on. Inward extension Docker structure can be lifting Kernel supported Virtual Machine (KVM) by making the identical belongings [13].

Linux Containers (LXC) are OS grade virtualization of work various separated Linux infrastructure along a singular Linux standard host it’s working like consumers location intersection in order to Linux kernel containment characteristics. Linux Containers (LXC) are supported along with court which carries binaries, libraries and configuration files consequently called chroot jail [14] and this access gives for separated surroundings along uppermost of the kernel [14].

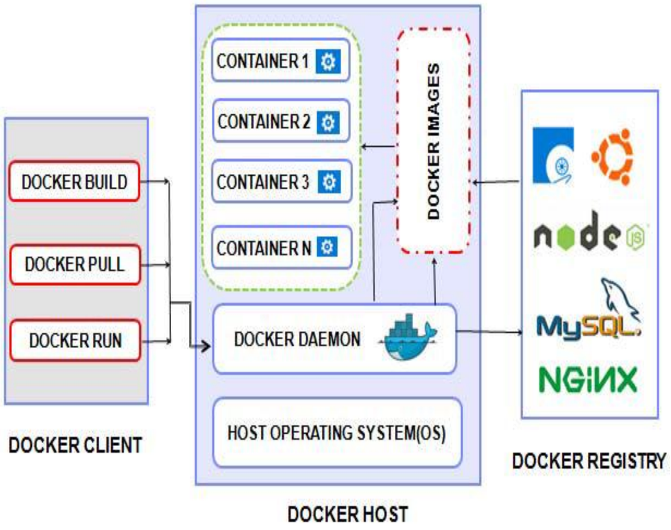

Fig. 3. Docker Architecture [12]

Container platforms by Cloud Foundry, Kubernetes, and CoreOS offering practicable virtualization substitutes simply Docker containers have acquired very much pace and progress to publicize position. Multinational Companies be like IBM, Microsoft and VMware are operative on to build their container strategies for the cloud [15].

Docker customer and daemon keep work on identical infrastructure or a Docker subscriber keep communicate directly to electric sockets or RESTful APIs to a distant Docker daemon. Docker images are only read template and Docker registries carry these images. Docker containers area unit designed from Docker image and it hold on belongings required to work a an program, our-self keep improvise behaviour by the move, run, start, stop or delete along Docker containers and every container is separated and protected program structure [17].

## IX. CONCLUSION

In this paper, ubiquitously comparing, it's believable to observing the situation of Kubernetes and Docker Swarm originally different. Swarm focusing along facility to utilization by unification up-to-date Docker core sections while Kubernetes stays open and standard. The common distinctness keep detected while installation and set up to each one of the platforms. Docker supplies an ease success i.e. rapid by collect originated up to date as long as Kubernetes target by endorse superior conditions up to date superior complication. Many of the common motives, Docker golden ager favourite in creators which choose easiness and rapid deployments. The common term, Kubernetes be applied inwards creative activity arena of huge utmost profile cyberspace companies functioning common services.

## REFERENCES

[1] Open Stack. [Online]. Available: https://www.openstack.org/software/ Last Accessed: December 21, 2017.

[2] Open Source [Online]. Available: https://www.openstack.org/ Last Accessed: December 21, 2017.

[3] Final Version of NIST Cloud Computing Definition. [Online]. Available: https://www.nist.gov/news-events/news/2011/10/final-version-nist-cloud-computing-definition-published Last Accessed: December 25, 2017.

[4] About OpenStack. [Online]. Available: https://www.openstack.org/software/project-navigator Last Accessed: December 26, 2017.

[5] Rakesh Kumar and Bhanu Bhushan Parashar. (November 2014) "Dynamic resource allocation and management using open stack", National Conference on Emerging Technologies in Computer Engineering (NCETCE) – 2014, Supported by Computer Society Chapter, IEEE Delhi Section.

[6] Ubuntu cloud. [Online]. Available: https://www.ubuntu.com/cloud Last Accessed: December 26, 2017.

[7] About Kubernetes [Online]. Available: https://www.ubuntu.com/kubernetes Last Accessed: December 27, 2017.

[8] Victoria Martinez de la Cruz in a nutshell. [Online]. Available: http://vmartinezdelacruz.com/in-a-nutshell-how-openstack-works/ Last Accessed: December 27, 2017.

[9] Cloud Computing. [Online]. Available: http://www.ibm.com/ibm/cloud/Last Accessed: December 27, 2017.

[10] Kubernetes: an overview. by Janakiram MSV. [Online]. Available: https://thenewstack.io/kubernetes-an-overvie/ Last Accessed December 28, 2017.

[11] Upcloud Ltd [FI]. Published by Janne Ruostemaa November 15, 2016. [Online]. Available: https://www.upcloud.com/blog/docker-swarm-vs-kubernetes/ Last Accessed: December 28, 2017.

[12] Understand the architecture. [Online]. Available: https://docs.docker.com/glossary/?term=Docker Last Accessed: December 29, 2017.

[13] Dev stacks. [Online]. Available: https://docs.openstack.org/devstack/latest/ Last Accessed: December 29, 2017.

[14] Matteo Riondato. Chapter14, Jails. [Online]. Available: https://www.freebsd.org/doc/en/books/handbook/jails.html Last Accessed: December 30, 2017.

[15] Virtualization without the Hypervisor. [Online]. Available: http:// docs.media.bitpipe.com/io_12x/io_128710/item_1261181/Virtualization%20Without%20The%20Hypervisor_hb_final.pdf.

[16] Amr Abdelrazik. [Online]. Available: https://mesosphere.com/blog/docker-vs-kubernetes-vs-apache-mesos/Last Accessed: December 31, 2017.

[17] Sachchidanand Singh and Nirmala Singh. "Containers & docker emerging roles & future of cloud technology". Page no. (804-807).

[18] Docker Swarm. [Online]. Available: https://docs.docker.com/engine/swarm/ Last Accessed: December 31, 2017.