International Journal of Scientific Research in Science, Engineering and Technology

Print ISSN - 2395-1990
Online ISSN : 2394-4099

OPEN ACCESS

Available Online at :www.ijsrset.com
doi : https://doi.org/10.32628/IJSRSET

Containerization In Application Engineering: Docker and Kubernetes Adoption

Murali Kadiyala
Independent Researcher, USA

ARTICLEINFO

Article History:

Accepted: 15 Oct 2022
Published: 30 Oct 2022

Publication Issue :
Volume 9, Issue 5
September-October-2022

Page Number :
369-374

ABSTRACT

Containerization has become a major breakthrough in the field of application in modern software engineering. Through using containers to package applications and their dependencies, degrees of freedom related to the required configurations are given to developers while at the same time making the systems somewhat more robust through the use of technologies such as Docker and Kubernetes. Docker is an innovative and high-performance tool that is used for running containers while Kubernetes is an appropriate platform for deployment, scaling and even management of containerized applications. When used together, Docker and Kubernetes identify and enable easier and faster delivery of software, scalability and decrease operational costs and are central to the DevOps and Cloud-native architecture paradigms. Docker and Kubernetes are described further, and the opportunities and prospects of their application, as well as the common problems of modern application development are examined in this paper.

Keywords : Containerization, Docker, Kubernetes, Application Engineering, DevOps

Introduction

Containerization has become the new norm of developing, deploying and managing applications in the today's market. That way, due to the encapsulation of applications and their dependencies into portable containers, developers can make the flexibility more fine-grained and make their applications more robust, as they do not depend on certain configurations and provisions of production environments. Docker and Kubernetes, two of the most popular technologies empowers have enabled have been key enablers of the containerization of applications. Docker can be described as a powerful and fast technology that offers the means for creating efficient containers that host the application in a platform-agnostic manner. Kubernetes, however, is an open-source automation platform, which helps to orchestrate the deployment of containers; it is a tool of great value for organizations that use containers at scale. Docker and Kubernetes together provide a solution to fasten the time to deliver new software versions, save time and effort in managing distributed applications. Docker and Kubernetes are high-tech platforms that are relevant to the growing façade of organization’s software engineering infrastructures as it endeavours to propel its DevOps initiatives.

Literature review

Containers for Virtualization: An Overview

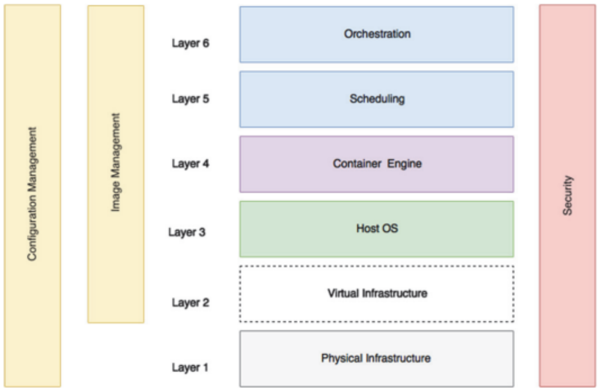

Figure 1: Architecture of cloud containers

(Source: da Silva *et al* 2018)

According to (da Silva *et al* 2018) discussing in detail the advantages of utilizing containers as part of cloud computing and virtualization. Containers, which do not need OS in order to run, are described and compared with hypervisors that need OS for VMs. As will be discussed below, the primary benefits of the use of containers in cloud virtualization include efficiency and speed. Such as, Docker, where issues associated with applications in the microservice architectures are highlighted as being flexible (da Silva et al., 2018). The study also explores the security risks as well as possible adverse consequences of containerization. The paper also entails a critical literature review of the available container technologies and their uses with an end view of projecting and comparing their growth. This paper focuses on explaining how containers can optimize the use of hardware resources and outlines the brunt of Docker and the other paradigm technologies such as LXC as well as Kubernetes. The paper ends with the assertion that, as container technologies provide for scalability and tremendous performance gains, there is also a lesser-talked-about side of container security that needs more attention as the technologies make their way into the cloud.

Containerize Your Apps with Docker and Kubernetes

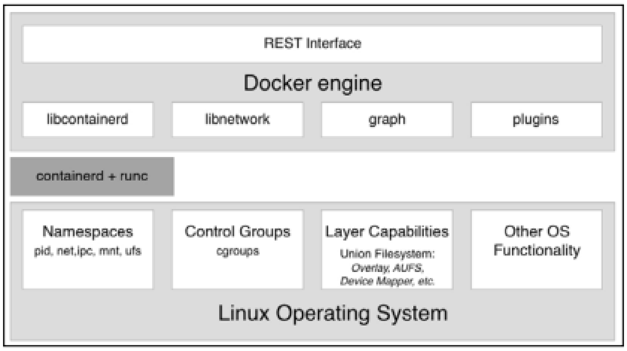

Figure 2: High-level architecture diagram of the Docker engine

(Source: Schenker 2018)

According to (Schenker 2018) the material of the report can be considered as a guide on how to work with application within the Docker and Kubernetes environment. This is complemented by an introductory section on containers to provide an understanding of their principle purpose to minimize interaction between links in the software delivery chain. This article covers Docker products as well as the ideas of containerization and containerization architecture and why these technologies are so important in today’s software deployment especially in the DevOps environment. The analysis also outlines how to setup a practical environment, how to pull down images and create vpn’s and volumes within containers (Schenker, 2018). It also involves in the deployment and management of containerized applications using Kubernetes in a way that showcases Kubernetes as a way of scaling, managing and securing applications in cloud environments. Other chapters focus on different aspects of distributed application architecture, another one is dedicated to Docker Compose for working with multi-container application and has a separate topic about container orchestration with Kubernetes before going to the details of deployment that guarantees the zero-downtime mode. This is specially designed for developers, system administrators, and dev ops engineers who are in the process of developing cloud-native applications using containers.

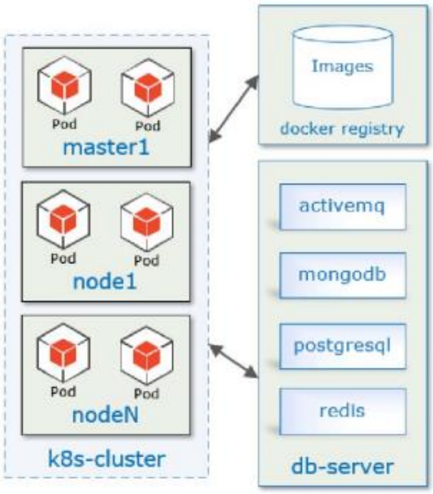

Figure 3 : Kubernetes Cluster for ATHENA: Architecture

(Source: Altaf *et al* 2018)

According to (Altaf *et al* 2018) mentioning ATHENA, a specific system designed for the ADF to facilitate the management of the continuum of helicopter pilot training. ATHENA is a strategic management/optimisation tool for such human resource planning and simulation requirements of the ADF. Concerning the cloud computing, this paper will discuss containerized applications which enable auto- scaling of ATHENA through Docker and Kubernetes. Since the ADF's usage continues to grow, the article describes how the platform leverages containerization to scale demanding simulations effectively. Specifically, the challenges that are presented deals with scaling simulation, especially those which are large scale and real-time processing. ATHENA uses Kubernetes for such deployment so that the application can scale up or down depending on the amount of work presented to the system (Altaf et al., 2018). It also reveals how the newly developed ATHENA can easily be integrated with cloud platforms such as OpenStack, scalability. It is described in general, as a framework for the cloud-native autoscaling, and it claims that container technologies are critical to modern defense and simulation applications.

Methods

Containerization of Applications with Docker

The discussion which follows concentrates on the use of Docker in containerization of applications. Docker is essentially a platform for packaging of an entire application in a single container along with all the specified dependencies and other necessities such as libraries and configurations and runtimes. The process of containerization starts with building up images and these images are modular to different hosts. This is demonstrated in the research as the way the approach eradicates the 'works on my machine' syndrome across the development, test and production platforms (Pérez et al., 2018). To that end, the Docker environment, particularly Docker Hub for images and Docker Compose for handling multiple containers, is used to facilitate this process. Also, the use of the Docker application by creating an environment within an application container is shown in microservice architectures in which separation of the applications into microservices is demonstrated where the large application is divided into smaller parts that can be easily managed, scaled, and updated.

Kubernetes for Orchestration and Auto-Scaling

Kubernetes requires that applications are containerized with Docker; its role is to act as the tool for deployment and scaling of these containers. It is a method that utilizes Kubernetes to orchestrate the running of applications that run on Dockers. Kubernetes has some of the key advantages that include pods, services, and deployments to scale applications with high availability (Kocher, 2018). The paper also describes how Kubernetes work in terms of scheduling of individual containers or containers failure handling and subsequent auto-rescheduling. It also covers the way Kubernetes works with cloud platforms for elastic and resilient process management.

Integration with Cloud Platforms and Security Considerations

Another key element of the methodology is the relations between Docker and Kubernetes with cloud platforms. Part of the research explores how services like Amazon Web Services, Microsoft Azure, and OpenStack are applied for deployment and orchestration of the containerized application. Its capabilities to integrate with cloud environments make it even easier to deploy applications on Kubernetes since it will help to manage the cloud computational resources. Security aspects will also be discussed, regarding to protecting the containerized solutions by segmentation, two types of access control, and container scanning (Sayfan, 2018). It also points out that, as with any other technology characterized by scalability and efficiency, organizations need to keep an eye on and protect their applications in containers.

Results

Performance Gains from Containerization

Due to Docker and Kubernetes, application outcomes derived optimal performance in the mobile environment. Another advantage of containerization was better and shorter time-to-deployment than earlier packaging, where the applications were packed in a standardized container for deployment in one or another environment (Pahl et al., 2017). Kubernetes helped to scale up and scale down the applications, which has lowered the time required to manage the change in workloads. In the context of cloud systems, it was possible to scale enterprise resources, depending on the demand at a particular period in order to enhance the quality of the service as well as the usage of resources. They were most apparent when it comes to the applications that utilized microservices because modularity and faster scaling were essential.

Efficiency in Resource Utilization

Due to containerization together with orchestration, the general resource utilization improved. Containers using Docker's design consume lesser memory and central processing unit (CPU) than conventional virtual machines (Truyen et al., 2016). One of them is a resource optimization focused on allocating the running containers in such a way that they occupy specific servers according to certain parameters. In this way, this method minimized operational costs by taking advantage of available resources and also ensured that every instance was used to the optimum level before being shut down without incurring the cost of running a full virtual machine (Rosa et al., 2018). This led to improved operational effectiveness particularly for the organizations that deal with numerous application within the multiple cloud system.

Security and Compliance Improvements

Security was one of the areas where the implementation of the container technology was valuable. Comparing the solutions Docker and Kubernetes, it can be stated that the points of security were more easily to be implemented with the two technologies such as isolation, access control and network segmentation. Containers provided enhanced fine-grained control to the different components of the applications and envisaged the security of the entire system. Moreover, the failure recovery and monitoring of Kubernetes made it possible for any substandard or infected container to be replaced with a better version (Pahl and Lee, 2015). Some of the ways that helped in enhancing the security of resources in the process include free security scanning tools, which check and detect threats, and other issues that could compromise the security of resources within the CI/CD pipeline before being deployed into the system.

Discussion

Docker and Kubernetes deployment has been seen to improve how these applications work in the cloud especially with regards to their performance, scalability and efficiency. The case studies and related literature reveal that containerization has much to offer for organizations which are interested in the modernization of app delivery. Docker ensures portability between different environments, Kubernetes offers an environment for scaling the application and easy management of the distributed systems (Zhang et al., 2018). Nevertheless, their efficacy goes hand in hand with the problems associated with them, namely, the issues of security and the scalability of the containers. Organizations still cannot afford to take their guards down for any form of security threats to their containerized applications. Despite these facts, the further usage of both Docker and Kubernetes still remains a breakthrough in the sphere of application engineering.

Future Direction

Containerization's future can be seen in improving security and making the Docker and Kubernetes ecosystems mesh well with the new technologies such as edge computing and AI subsequently auto-automation. In a year of hybrid and multiple clouds, there will be more requests for the group of technologies to improve the orchestration of applications in ships with the help of different clouds providers (Turnbull, 2014). Also, increasing the built-in support of serverless architectures in Kubernetes can extend the capabilities of the containerized applications. Research is expected to lie on efficiency in the deployment of resources and enhancing the monitoring of real-time security in the containers.

Conclusion

Docker and Kubernetes have become crucial in contemporary software development since they offer organizations and developers numerous features that enable them to containerize and access their applications seamlessly. They enhance operations productivity, system deployment, and flexibility for the task in response to certain triggers. As evident above, several drawbacks come with containerization with the most significant drawbacks being security and manageability of complexity. Given that organizations adopt cloud-native architectures and Socket Address DevOps practices further, Docker and Kubernetes are going to remain a vital collection for the development of the future application and further application deployment.

REFERENCES

[1]. da Silva, V.G., Kirikova, M. and Alksnis, G., 2018. Containers for virtualization: An overview. Applied Computer Systems, 23(1), pp.21-27.

[2]. Schenker, G.N., 2018. Containerize your Apps with Docker and Kubernetes. Packt Publishing.

[3]. Altaf, U., Jayaputera, G., Li, J., Marques, D., Meggyesy, D., Sarwar, S., Sharma, S., Voorsluys, W., Sinnott, R., Novak, A. and Nguyen, V., 2018, December. Auto-scaling a defence application across the cloud using docker and kubernetes. In 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion) (pp. 327-334). IEEE.

[4]. Pérez, A., Moltó, G., Caballer, M. and Calatrava, A., 2018. Serverless computing for container-based architectures. Future Generation Computer Systems, 83, pp.50-59.

[5]. Kocher, P.S., 2018. Microservices and containers. Addison-Wesley Professional.

[6]. Sayfan, G., 2018. Mastering Kubernetes: Master the art of container management by using the power of Kubernetes. Packt Publishing Ltd.

[7]. Turnbull, J., 2014. The Docker Book: Containerization is the new virtualization. James Turnbull.

[8]. Zhang, Y., Vasilescu, B., Wang, H. and Filkov, V., 2018, October. One size does not fit all: an empirical study of containerized continuous deployment workflows. In Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 295-306).

[9]. Pahl, C. and Lee, B., 2015, August. Containers and clusters for edge cloud architectures--a

technology review. In 2015 3rd international conference on future internet of things and cloud (pp. 379-386). IEEE.

[10]. Rosa, G., Scalabrino, S., Bavota, G. and Oliveto, R., 2018. What Quality Aspects Influence the Adoption of Docker Images?. ACM Transactions on Software Engineering and Methodology.

[11]. Pahl, C., Brogi, A., Soldani, J. and Jamshidi, P., 2017. Cloud container technologies: a state-of-the-art review. IEEE Transactions on Cloud Computing, 7(3), pp.677-692.

[12]. Truyen, E., Van Landuyt, D., Reniers, V., Rafique, A., Lagaisse, B. and Joosen, W., 2016, December. Towards a container-based architecture for multi-tenant SaaS applications. In Proceedings of the 15th international workshop on adaptive and reflective middleware (pp. 1-6).