Understanding Application Containerization with Docker and Docker Swarm

Venkata Soma

New York Mets

Abstract: ***The study of Docker and Docker Swarm and the application of containerization explores the components, processes, and benefits of the system in the sports industry. It identifies the challenges within this virtualization and optimization process regarding security and scalability. The research suggests recommendations for the improvement of security framework and performance algorithms to apply the technology in real-world cases.***

Keywords: Docker, Docker Swarm, cloud computing, containerization, microservices, management, sports

1. Introduction

a) Project Specification

Docker Swarm is a container orchestration instrument for assembling and scheduling docker containers or nodes as a solitary virtual system. This tool assists the IT administrators or developers in connecting multiple hosts, handling each node's resources, and improving application availability throughout a single system. Docker Swarm uses different scheduling strategies to distribute the load including spread, binpack, and random [1]. Containerization is a lightweight virtualization technology, that introduces a revolution in the management of cloud applications. This project's goal is to understand the application process with deep learning methods to create a reliable and effective system within the docker swarm for the management of the sports sector. The criteria during the scheduling of the containerization will be delved into for future recommendations.

b) Aim and Objectives

Aims:
This research aims to understand the functionalities of containerization along with the benefits of Docker Swarm. It focuses on the development of the necessary skills within individuals for utilizing Docker in real-world situations.

Objectives:

- To explore the usefulness and advantages of Docker Swarm in application development and deployment.
- To investigate the process of Docker Swarm to orchestrate and manage the containerization application in cluster circumstances.
- To identify loopholes in the security measures and suggest strategies for troubleshooting and monitoring Docker containers.

c) Research Question

- What are the basic principles of application containerization to implement with Docker?
- How do Docker containers improve the efficiency of application development from the traditional virtual method?

d) Research Rationale

Issue 1: The rapidly evolving development of modern applications requires speed with accuracy and consistency in using the resources. In this context, Docker can work as a leading containerization platform by encapsulating the applications into its portable containers.

Issue 2: The findings of this project will guide the developers to achieve more streamlined, scalable, and reliable application deployment processes through Docker and Docker Swarm.

2. Literature review

a) Research background

The continuous increase in the amount of data generated by ‘Internet of Things (IoT) devices’ reflects challenges to support the ‘cloud infrastructure’ for processing and storing the data. To manage the ‘Big Data domain’ within a ‘low-power and low-cluster’ of single board computers such as Raspberry Pi, Docker is used to containerize and deploy an ‘Apache Hadoop’ and ‘Apache Spark cluster’ [2]. Docker container r is a runnable instance of a ‘Docker image’, a layered template with instructions that hold all the essentials for running applications [3]. Docker Swarm mainly schedules and optimizes the workloads by ensuring the availability of resources to run the system efficiently. The effective management and processing of edge devices employed the Dockers.

b) Critical Assessment

Containerization is the process of virtualizing applications to run them in complex data management circumstances in the sports industry. The containers of Docker are lightweight rather than traditional virtual machines and work at a high speed with fewer resources [4]. It can facilitate the storage of large amounts of data such as footage of the game or player performance statistics. However, the containers share the host OS kernel, which may lead to potential security vulnerabilities regarding information such as the health data of players [5]. The application of Docker Swarm is more simple than other tools such as Kubernetes and can be easily integrated with existing Docker.

DOI: https://dx.doi.org/10.21275/SR24810090806

c) Linkage to Aim

The research of [6] demonstrates the advantages of using Docker to improve application deployments by reducing conflicts and simplifying the update process. Several studies provide an overview of the application and significance of Docker by using the principles regarding technologies in scalable systems.

d) Encapsulation of applications

The encapsulation process of Docker for an application works to configure and maintain its libraries in a single container through different components. For example, Docker file explains the environments and requirements and provides an image as a snapshot of the application [7]. The software containers work for distribution without hazards of dependencies or incompatibilities in various processes such as live data from games or player sensors.

e) Utilizing Docker Swarm for clustering

The application of Docker Swarm confirms the operational efficacy of sports analysis and fan engagement even during high-traffic situations. The existing literature on Docker Swarms delves into its capabilities of containerization within the cluster of nodes with balancing of workload and fault tolerance [8]. The application of Docker Swarm facilitates the management of a costly and centralized cloud nature in the case of large venues such as sports stadiums. The architecture of the clustering within a container consists of single or more ‘master nodes’ along with one or several ‘worker nodes.’ In this case, a load balancer works to pinpoint the available clusters prepared to deliver the outputs of the designed operations.

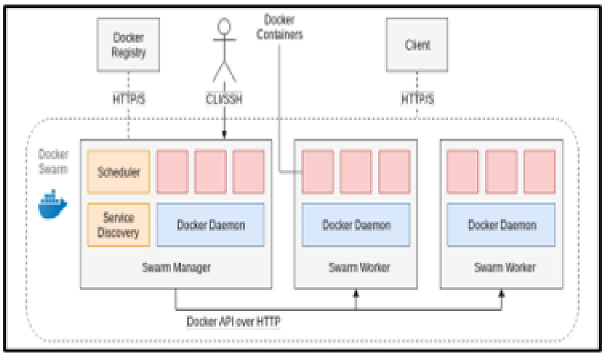

Figure 2.1: Docker Swarm Configuration [1]

f) Theoretical Framework

Microservices architecture involves the brakes down of larger applications into precise, smaller, and loosely coupled services to develop, deploy, and maintenance of scalability. ‘Container technology’ represents the upcoming evolutionary phase in distributed computing that aims to assist in the optimization and the utilization of distributed resources for quality throughput, specifically in the context of microservices [10]. The theory of container orchestration sheds light on the analysis of real-time platforms of sports, especially during the peak hours such as playoff seasons. Optimization of the resources is crucial to maintain the demand and reduce the overheads related to match ticketing, interaction with fans, and game analysis.

g) Literature Gap

The backlogs in the existing literature show the other findings about the issues such as differentiation between containers and the host systems to maintain the security. The journals are not sufficient to encompass the dynamic resource allocation with deallocation such as CPU, Memory, and storage in the case of real-time containerization. The scalability and orchestra of Docker Swarm require improved algorithms depending on a variety of loads and the complexity of situations. The study needs to focus on the identification of performance bottlenecks in containerized environments, including network and storage.

3. Methodology

a) Research Philosophy

This project is based on the overview of the utilization of the container system by the NY Mets within the Docker and Docker Swarm in the various activities within the sports industry. The research will use the philosophy of interpretivism to emphasize on the perspectives of researchers on this topic. It will explore the opinions of the developers, users, and administrators about the significance of Docker Swarm clusters and the performance of containers. The interpretivism encompasses social theories and perspectives that embrace a view of reality as socially constructed.

b) Research Approach

The research regarding the complete framework of Docker and Docker Swarm for the development of software tools and techniques for simultaneous management in sports sectors. This research includes the deductive approach to investigate the efficacy of the integration of Docker with continuous deployment for automated testing of the sports instruments. By using the deductive method this project will provide the opinion of previously working individuals through data collection and analysis.

c) Research design

To collect and analyze the data about the performance of Docker within the actions of the sports industry, the secondary qualitative method is used. It will help to provide an overview of the Docker development and deployment process including the formation of ‘Docker images’ to container orchestration.

d) Data collection method

The data collection will be performed through peer review of previously published scholarly articles, and journals accessed through Google Scholar and PubMed. The obtained information will be accumulated and analyzed based on thematic analysis.

e) Ethical considerations

In this study, the maintenance of the ethical perspectives is one of the most significant sections. Firstly, the privacy and permission laws must be followed when using the confidential information about sports events. The performance of the Docker and Docker Swarm needs to be managed by eliminating biases. Containerization requires appropriate management during the encapsulation in a single package.

Paper ID: SR24810090806 DOI: https://dx.doi.org/10.21275/SR24810090806

4. Results

a) Critical Analysis

The process of orchestration navigates the problem of coordination between the ‘software and hardware’ in a cluster and replicates actions to provide high availability in outcomes. Once the desired condition is recognized, the ‘orchestrator’ works to action to mitigate the gap between the current state and the expected one. The novel of [11] shares an example of real-time applications on Edge Cloud. It adds that the system uses ‘QoS parameters’ to determine the best edge node to which a user can rely on and connect. The authors designed a framework, that operates through ‘the VMM layer’, capable of monitoring without the need for monitoring agents in the ‘guest OS’.

b) Finding and Discussion

Theme 1: The requirements of Cloud networking in the sports industry

The findings of the research suggest that the utilization of containerization is important for the sports industry to organize and manage the application sufficiently. The docker can support the dynamic scaling with the accommodation of huge workloads [12]. The main nodes and worker nodes work to prevent any single points of failure in the management. The utilization of Docker files helps to work uniformly within multi-faceted environments. For example, in the case of management of a sports team can focus on resources, limitations, and performances through containerization in diverse environments.

Theme 2: Components of Docker and Docker swarm

The organization, NY Mets can design the Docker and Docker-related orchestration tool such as Docker Swarm in the case of live broadcasting of major sports events. The article of [9] says that a Docker includes several components for processing the technique of the application of Docker Containers will handle the enhanced and congested viewer traffic during a live streaming service of any sports channel. Docker Engine is one of the Docker applications that is built focusing on the client-server architecture. The Docker daemon is referred to as Dockerd which is the key component of Docker Container technology. It acts as the server that manages all Docker objects such as images, containers, volumes, and networks. For example, a client can provide a ‘command line interface (CLI)’ to communicate with the Docker daemon by using a ‘REST (representational state transfer)’ ‘API (application programming interface)’ [13]. ‘Docker Daemon’ is an integral part of the ‘Docker ecosystem’ that enables the scalability, portability, and agility of containerization within the operations of the sports industry. The Docker client acts as the command-line interface (CLI) that facilitates the actions of the containers. Another

Theme 3: Significance of the Docker and Docker Swarm in the process

The findings suggest that ‘Docker containers can run on any system that supports Docker. It requires assurance of a continuous runtime environment regardless of the underlying infrastructure. The application of Docker is beneficial for many reasons such as Docker containers sharing the host OS kernel which results with lower overhead compared to full virtualization solutions [14]. The monitoring of Docker involves factors for the assessment to keep a record of events, resource usage, and other metrics. These parameters are used to evaluate the performance of the systems and to provide knowledge about Docker-related tasks. Docker has a wide and active community with features of extensive documentation and a rich ecosystem of tools. These platforms help to build a strong performance analysis tool in the case of different sports such as baseball. The process becomes streamlined and easier to find support and resources for Docker depending on actions especially in during high-traffic events like playoffs or championships.

c) Evaluation

Docker swarm delivers an easier-to-use experience to the sports authorities who were earlier acquainted with Docker [15]. Containerization works through a ‘streamlined’ form of ‘virtualization’ by designing specialized containers from specific images as bundles of self-contained applications. These containing processes require a few resources and time. Moreover, this process of containerization through Docker offers a high level of compatibility in the case of packaging the applications. This is crucial for creating portable and interoperable software solutions in cloud environments specifically in the sports industry.

5. Conclusion

The research summarizes the importance of Docker and Docker Swarm on the management and deployment of containerization and orchestration capabilities. It sheds light on the advantages of the application of Docker to manage the high-demand scenarios in the sports sector such as live streaming of important matches. This lightweight visualization process efficiently works for resource utilization accurately in a simpler way. The application of Docker Swar maintains load balancing and fault tolerance in container computing within the system.

6. Research Recommendations

The potential security vulnerabilities associated with Docker containers need to include specific security measures through regular scanning of images and communications. The challenges recognized in scalability require refining and optimization of container orchestration during large-scale sports events. The docker can integrate with advanced technologies such as artificial intelligence or edge computing to smoothen the decision-making process with reports from real-time analytics.

7. Future Work

By addressing the loopholes in the findings, it can be suggested that the development of new and modified algorithms can enhance the performance of Docker Swarm for working in complex situations. The feedback from the user’s needs to be received and analyzed about their satisfaction regarding the containerization process in the Docker application. The future research needs to focus on the integration process of Docker with IoT devices for increasing real-time data processing and analytics in sports applications.

References

[1] J.N. Acharya and A.C. Suthar, 2021, October. "Docker container orchestration management: A review." In *International Conference on Intelligent Vision and Computing* (pp. 140-153). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-97196-0_12

[2] R. Scolati, I. Fronza, N. El Ioini, A. Samir, and C. Pahl, 2019, May. "A containerized big data streaming architecture for edge cloud computing on clustered single-board devices." In *Closer* (pp. 68-80). DOI:10.5220/0007695000680080

[3] D. Nüst, D. Eddelbuettel, D. Bennett, R. Cannoodt, D. Clark, G. Daróczi, M. Edmondson, C. Fay, E. Hughes, L. Kjeldgaard, and S. Lopp, 2020. "The rockerverse: Packages and applications for containerization with R." *arXiv preprint arXiv:2001.10641*. https://arxiv.org/abs/2001.10641

[4] P. Saha, A. Beltre, and M. Govindaraju, 2019. "Scylla: A Mesos framework for container-based MPI jobs." *arXiv preprint arXiv:1905.08386*. https://www.researchgate.net/publication/333259856_Scylla_A_Mesos_Framework_for_Container_Based_MPI_Jobs

[5] S. Kadri, A. Sboner, A. Sigaras, and S. Roy, 2022. "Containers in bioinformatics: Applications, practical considerations, and best practices in molecular pathology." *The Journal of Molecular Diagnostics*, 24(5), pp. 442-454. https://doi.org/10.1016/j.jmoldx.2022.01.006

[6] M. Ileana, M.I. Oproiu, and C.V. Marian, 2024, May. "Using Docker Swarm to improve performance in distributed web systems." In *2024 International Conference on Development and Application Systems (DAS)* (pp. 1-6). IEEE. https://dasconference.ro/usb2024/data/papers/D08.pdf

[7] C. Lin, S. Nadi, and H. Khazaei, 2020, September. "A large-scale data set and an empirical study of Docker images hosted on Docker Hub." In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 371-381). IEEE. https://www.researchgate.net/profile/Hamzeh-Khazaei/publication/344198434_A_Large-scale_Data_Set_and_an_Empirical_Study_of_Docker_Images_Hosted_on_Docker_Hub/links/5f5aec4da6fdcc116409389c

[8] P. Muzumdar, A. Bhosale, G.P. Basyal, and G. Kurian, 2024. "Navigating the Docker ecosystem: A comprehensive taxonomy and survey." *arXiv preprint arXiv:2403.17940*. DOI:10.9734/AJRCOS/2024/v17i1411

[9] I. Shabani, E. Mëziu, B. Berisha, and T. Biba, 2021. "Design of modern distributed systems based on microservices architecture." *International Journal of Advanced Computer Science and Applications*, 12(2). https://www.academia.edu/download/79374916/Paper_20-Design_of_Modern_Distributed_Systems.pdf

[10] A. Saboor, M.F. Hassan, R. Akbar, S.N.M. Shah, F. Hassan, S.A. Magsi, and M.A. Siddiqui, 2022. "Containerized microservices orchestration and provisioning in cloud computing: A conceptual framework and future perspectives." *Applied Sciences*, 12(12), p.5793. https://doi.org/10.3390/app12125793

[11] S. Dilek, K. Irgan, M. Guzel, S. Ozdemir, S. Baydere, and C. Charnsripinyo, 2022. "QoS-aware IoT networks and protocols: A comprehensive survey." *International Journal of Communication Systems*, 35(10), p.e5156. https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.5156

[12] S. Luo, H. Xu, K. Ye, G. Xu, L. Zhang, G. Yang, and C. Xu, 2022, November. "The power of prediction: microservice auto scaling via workload learning." In *Proceedings of the 13th Symposium on Cloud Computing* (pp. 355-369). https://www.mdpi.com/2076-3417/12/12/5793

[13] J.M.O. Candel, 2020. *DevOps and Containers Security: Security and Monitoring in Docker Containers*. BPB Publications. https://books.google.com/books?hl=en&lr=&id=J13VDwAAQBAJ&oi=fnd&pg=PP18&dq=a+client+can+provide+a+command+line+interface+(CLI)+to+communicate+with+the+Docker+daemon+by+using+a+REST+(representational+state+transfer)+API+(application+programming+interface)

[14] R. Bullington-McGuire, A.K. Dennis, and M. Schwartz, 2020. *Docker for Developers: Develop and run your application with Docker containers using DevOps tools for continuous delivery*. Packt Publishing Ltd. https://books.google.com/books?hl=en&lr=&id=Hif4DwAAQBAJ&oi=fnd&pg=PP1&dq=The+application+of+Docker+is+beneficial+for+many+reasons+such+as+Docker+containers+sharing+the+host+OS+kernel+which+results+in+significantly+lower+overhead+compared+to+full+virtualization+solutions

[15] F.N. Nwebonyi, 2020. "Establishing trust and confidence among entities in distributed networks." https://repositorio-aberto.up.pt/bitstream/10216/127994/2/410091.pdf