

# Containers Management System

Harshwardhan Mehrotra  
Computer Science Engineering  
MIT ADT University  
Pune, India  
harshwardhanmehrotra@gmail.com

Shreyansh Dubey  
Computer Science Engineering  
MIT ADT University  
Pune, India  
shyam8dubey@gmail.com

Viren Ghuikhedkar  
Computer Science Engineering  
MIT ADT University  
Pune, India  
virenghuikhedkar7890@gmail.com

Soham Dhande  
Computer Science Engineering  
MIT ADT University  
Pune, India  
sohamdhande11@gmail.com

Prof. Nilesh Marathe  
Computer Science Engineering  
MIT ADT University  
Pune, India  
nilesh.marathe@mituniversity.edu.in

**Abstract**—“Container Management” is a Web Application that create a system for creation, updation and deletion of containers. This will improve work for IT team to smoothen up the container handling and will increase portability, more consistent operation greater efficiency. This will reduce their hassle of keeping track of containers. This helps overall in preparing the IT team in managing containers.. Being a web application built with django, the software has the advantage of being portable and usable anywhere. Users can login anywhere at any time and monitor their network control. Our dashboard provides 3 essential domain options to work with, firstly to create a container, next is updation of the existing container and deletion of a container

**Index Terms**—Container Management System, Containers, Docker, Docker API, Docker Images

## I. INTRODUCTION

Teams within an organization will often start experimenting with containers by pulling down container images and running them on their local machines. They may then move on to sharing containers they’ve developed with other team members through a container registry. As proficiency grows, they may even wire several containers together and deploy them as a single unit. As organizations start leveraging containers beyond the development/test phases and start adopting them in production, it becomes quickly apparent that additional tools are required to manage the containers effectively. This is where container management tools come into play

## II. PROBLEM STATEMENT

A vast number of containers can become too complex for an IT team to handle .So Container management will use software to automatically create, deploy and scale containers so as to make the work easier to handle. Container management is necessary when enterprises rely on containers to quickly deploy and update applications

## III. LITERATURE SURVEY

Sr No.	Referred Paper/Journal	Year of Publication	Summary of paper referred
1	A Comparative Study of Containers and Virtual Machines in Big Data Environment	2018	This paper helped us to understand the need of containers and why they are better than using virtual machines in cloud.
2	A Performance Study of Containers in Cloud Environment	2016	This journal helped us to understand how that system containers are more suitable to sustain I/O-bound workload
3	A survey on Docker and its significance in cloud	2016	This journal gave us a a brief about Docker. Docker which is a light weight virtualization tool has enabled developers to build and run applications in distributed environment effectively and efficiently

4	A Survey on Docker Container and its Use Cases	2020	This journal helped us understanding the uses of docker that It has a wider scope of benefits for both Developers and System Administrators by allowing Developers to write code without having to worry about the system, that it will ultimately be running on. Also, it potentially reduces the number of systems and offers flexibility for the operations staff.
5	Virtualization Using Docker Containers: For Reproducible Environments and Containerized Applications	2018	It contains contents regarding virtualization. The evolution of microservices has changed the requirements of underlying infrastructure, technologies, and tools which were once used to manage the applications. These services improved the agility of delivering software which are portable across all the platforms and infrastructures

#### IV. OBJECTIVE OF PROJECT

The Objective of our project is to:

- A DevOps approach to manage the continuous development and deployment would benefit – the cross-stagenature is obvious. While most papers look at the two sides separately, the need to link these in a continuous process and to feedback monitored operational data into development becomes obvious.
- Containers represent a progression from virtual machines towards lightweight application management. Lately, there is an observable continuing trend towards serverless architectures and other mechanisms to manage orchestration and deployment complexity such as unikernel technology towards more lightweightness.
- Similar to Docker kick-starting research on container technology as an active open-source environment, a similar impact may be expected with the emergence of similar

technologies in the cluster space such as Kubernetes and Mesos.

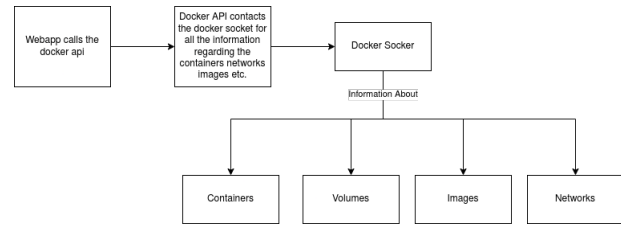


Fig. 1. Basic working Block Diagram

#### V. CONCLUSION

Organizations are packaging applications in containers and need to orchestrate multiple containers across cloud servers:

- The benefits of container based orchestration include adjustable cluster sizes for the deployment of containers, easier cluster maintenance and also quicker deployment. This is confirmed by the motivations cited, which prioritise the easy, automated container deployment and management in larger settings allowing for flexible migration and reconfiguration.
- Containerisation positively impacts on both development and deployment aspects such as testing and monitoring of container based applications, which we found both covered by the studies in terms of a mix of architecture design and continuous, quality driven management.
- The quality distribution shows a strong interest in optimised resource utilisation (effectively a cost factor) and performance, complemented by portability/interoperability and security. Interesting is also the emergence of two equally important perspectives: quality aspects directly observable by the consumers and the quality aspects relevant to the management of the applications within the platform itself. A theme that emerges is the support of continuous development through containers, joining both construction as well as operations and management. In the cloud, cloud-native platform services for development and deployment do exist, but require advanced PaaS orchestration support.

We can conclude that the field is moving towards container middleware (and even container PaaS) with isolation, construction, quality management, orchestration and distribution management as core concerns of a container PaaS middle-ware, but key features such as failure management are still missing.

## ACKNOWLEDGMENT

We would like to express our sincere gratitude to our Mini Project guide Prof Nilesh Marathe for their able guidance and support in completing our project. We would also like to extend our gratitude to the Principle of MIT-School of Engineering Dr. Kishore Ravande and our Dean and Computer Science Department Head Prof Dr. Rajneeshkaur Sachdeo for providing us with all the facilities that were required

### *A. Authors*

- 1) Harshwardhan Mehrotra
- 2) Shreyansh Dubey
- 3) Viren Ghuikhedkar
- 4) Soham Dhande
- 5) Prof. Nilesh Marathe