

# Docker and Its Impact on Modern Software Development

**\*Mohit Sharma**

**\*\*Dr. M.M.S Rauthan**

The advent of containerization has revolutionized software development and deployment by enabling lightweight, portable, and consistent environments across diverse systems. Docker, a leading platform in this domain, has simplified the process of building, sharing, and running containerized applications. It also provides Docker Hub, which is a cloud service for sharing applications. Costs can be reduced by replacing traditional virtual machines with docker containers. It excellently reduces the cost of re-building the cloud development platform.

**Keywords :** Docker, Docker Container, Virtual Machine, Virtualization, Cloud Computing.

## 1. Introduction

Docker is an open source platform that runs applications and makes the process easier to develop, distribute. The applications that are built in the docker are packaged with all the supporting dependencies into a standard form called a container. These containers keep running in an isolated way on top of the operating system's kernel.

Even though the technologies of the container have been around for over 10 years, docker, a generally new hopeful, is right now a standout amongst the best innovations, since it accompanies new capacities that prior technologies did not have. Initially, it gives the facility to create and control containers. Besides that, applications can easily be packed into lightweight docker containers by the developer. These virtualized applications can easily be worked anywhere without any alteration. Moreover, docker can convey more virtual situations than different innovations, on the same equipment. To wrap things up, docker can easily coordinate with third-party instruments, which help to easily deploy and manage docker containers. Docker containers can easily be deployed into the cloud- based environment.

This paper is a review on the technology of docker. Next section will introduce the technology of docker. In Section 3, a more detailed description of docker and its components will be presented. Section 4 briefly compares the technology of Virtual Machine and Docker. Sections 5 will discuss the advantages and disadvantages of docker containers. Finally, in section 6 discuss features in virtual machines and containers will be briefly summarised, following with a short summary of the paper.

## 2. Docker

Docker provides a facility to automate the applications when they are deployed into Containers. In a Container environment where the applications are virtualized and executed, docker adds an extra layer of deployment engine on top of it. The way that docker is designed is to give a quick and a

---

## Docker and Its Impact on Modern Software Development

*Mohit Sharma & Dr. M.M.S. Rauthan*

lightweight environment where code can be run efficiently and moreover it provides an extra facility of the proficient work process to take the code from the computer for testing before production. as quick as it is possible, docker allows you to test your code and deploy it into the production environment. docker is amazingly simple. Certainly, you can begin with a docker with a simple configuration system, a docker binary with Linux kernel.

### **3. Docker Inside**

There are four main internal components of docker, including Docker Client and Server, Docker Images, Docker Registries, and Docker Containers. These components will be explained in detail in the following sections.

#### **3.1 Docker Client and Server**

Docker can be explained as a client and server based application. The docker server gets the request from the docker client and then processes it accordingly. The complete RESTful (Representational state transfer) API and a command line client binary are shipped by docker. Docker daemon/server and docker client can be run on the same machine or a local docker client can be connected with a remote server or daemon, which is running on another machine.

#### **3.2 Docker Images**

There are two methods to build an image. The first one is to build an image by using a read-only template. The foundation of every image is a base image. Operating

system images are basically the base images, such as Ubuntu 14.04 LTS, or Fedora 20. The images of the operating system create a container with an ability to completely run the OS. Base images can also be created from scratch. Required applications can be added to the base image by modifying it, but it is necessary to build a new image. The process of building a new image is called “committing a change”. The second method is to create a docker file. The docker file contains a list of instructions when “Docker build” command is run from the bash terminal it follows all the instructions given in the docker file and builds an image. This is an automated way of building an image.

#### **3.3 Docker Registries**

Docker images are placed in docker registries. It works correspondingly to source code repositories where images can be pushed or pulled from a single source. There are two types of registries, public and private. Docker Hub is called a public registry where everyone can pull available images and push their own images without creating an image from scratch. Images can be distributed to a particular area (public or private) by using docker hub feature.

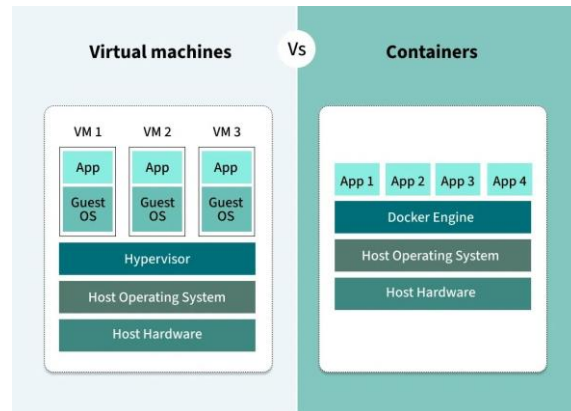
#### **3.4 Docker Containers**

Docker image creates a docker container. Containers hold the whole kit required for an application, so the application can be run in an isolated way. For example, suppose there is an image of Ubuntu OS with SQL SERVER, when this image is run with docker run command, then a container will be created and SQL SERVER will be running on Ubuntu OS.

---

## **Docker and Its Impact on Modern Software Development**

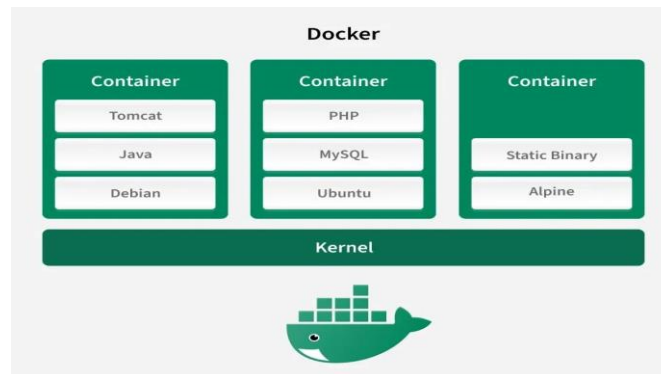
*Mohit Sharma & Dr. M.M.S. Rauthan*



**Fig. 1 Virtual Machine vs. container**

#### 4. Virtual Machine vs. Docker

Virtualization is an old concept, which has been in use in cloud computing, after IaaS has been accepted as a crucial technique for system constitution, resource provisioning, and multi-tenancy. Virtualized resources play the main role in solving the problems using the core technique of cloud computing. Hypervisor is lying between host and guest operating systems. It is a virtual platform and it handles more than one operating system in the server. It works between the operating system and CPU. Virtualization divides it into two segments: the first one is Paravirtualization and the second one is Full Virtualization. Linux containers are managed by the docker tool and it is used as a method of operating system level virtualization. In a single control host there are many Linux containers, which are isolated. Resources such as Network, Memory, CPU, and Block I/O are allocated by Linux kernel and it also deals with cgroups without starting a virtualization machine.



**Fig. 2 Docker container**

---

## Docker and Its Impact on Modern Software Development

*Mohit Sharma & Dr. M.M.S. Rauthan*

#### 4.1 Docker Containers

Docker image creates a docker container. Containers hold the whole kit required for an application, so the application can be run in an isolated way. For example, suppose there is an image of Ubuntu OS with SQL SERVER, when this image is run with docker run command, then a container will be created and SQL SERVER will be running on Ubuntu OS. The demand and the advancement of Linux containers can be seen in the last few years.

#### 5. Advantages of Docker Container

Docker has become popular very quickly, because of the benefits provided by docker containers. The main advantages of docker are speed, portability, scalability, rapid delivery, and density.

##### 4.2 Speed

Speed is one of the most exceedingly touted advantages of Containers. When the benefits of using docker are highlighted, it would be incredible not to mention the speed of docker in the conversation (Chavis & Architect, 2015). The time required to build a container is very fast because they are really small. Development, testing, and deployment can be done faster as containers are small. Containers can be pushed for testing once they have been built and then from there, on to the production environment.

##### 4.3 Portability

Those applications that are built inside docker containers are extremely portable. These portable applications can easily be moved as a single element and the performance remains the same.

##### 4.4 Scalability

Docker has the ability to be deployed in several physical servers, data servers, and cloud platforms. It can also be run on every Linux machine. Containers can easily be moved from a cloud environment to a local host and from there back to cloud again at a fast pace. Adjustments can easily be done; the scale can simply be adjusted by the user according to the need.

##### 4.5 Rapid Delivery

The format of Docker Containers is standardized so programmers do not have to stress over one another's tasks. The responsibility of the administrator is to deploy and maintain the server with containers, whereas the responsibility of the programmer is to look after the applications inside the docker container. Containers can work in every environment as they have all the required dependencies embedded within the applications and they are all tested. Docker provides a reliable, consistent, and improved service.

#### 6. Dockerizing Django Application

Docker has revolutionized the way we build, ship, and run applications. It provides a consistent and reliable environment for application, making it easier to develop and deploy. you have these tools

---

### Docker and Its Impact on Modern Software Development

*Mohit Sharma & Dr. M.M.S. Rauthan*

installed on your system. First is download and install Docker from the official Docker website. second is Ensure you have Python installed on your system, as Django is a Python web framework third is Django Project: You should have a Django project. then you can build and run your Docker containers using the commands.

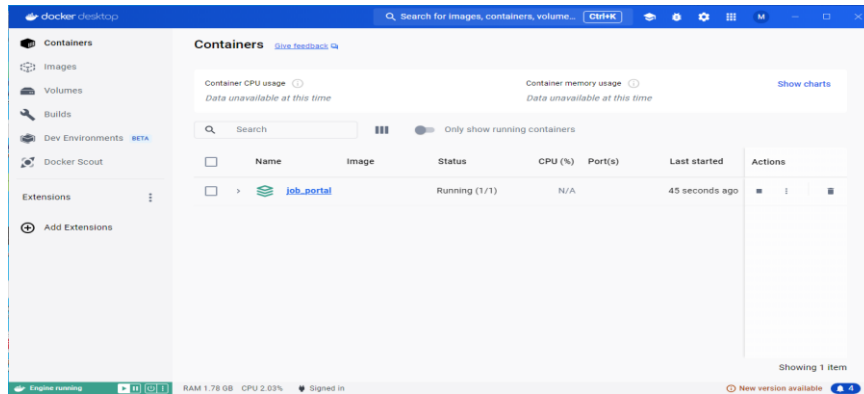


Fig. 3 Dockerizing django application

7. Containerization

Market share and customer base information about the top Containerization software. Check out our list of Containerization Software, analyze them by market share, region, company size along with information about competitors, prospects, tech stacks, funding information, contract renewal dates and more.

7.1 Top 5 Containerization technologies in 2025

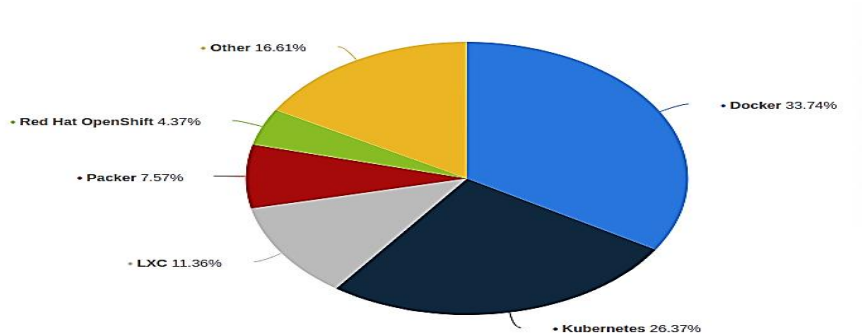


Fig. 4 Containerization technologies

Docker and Its Impact on Modern Software Development

Mohit Sharma & Dr. M.M.S. Rauthan

### 7.1 Containerization Customers by Geography

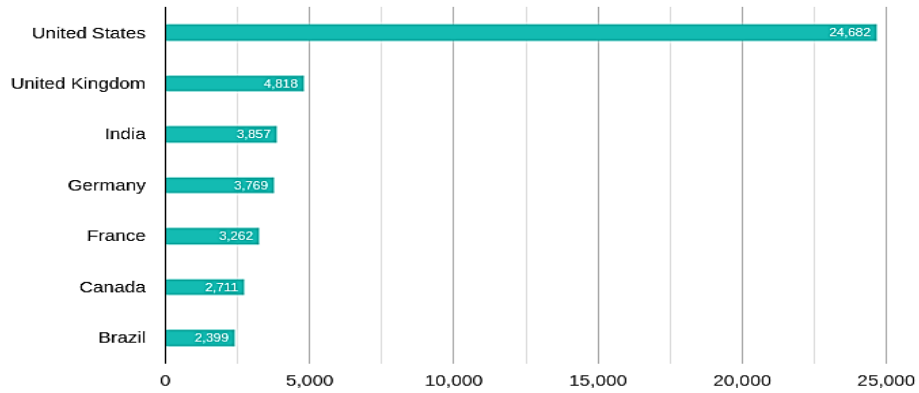


Fig. 5 Containerization Customers by Geography

### 7.3 Containerization Customers by Industry

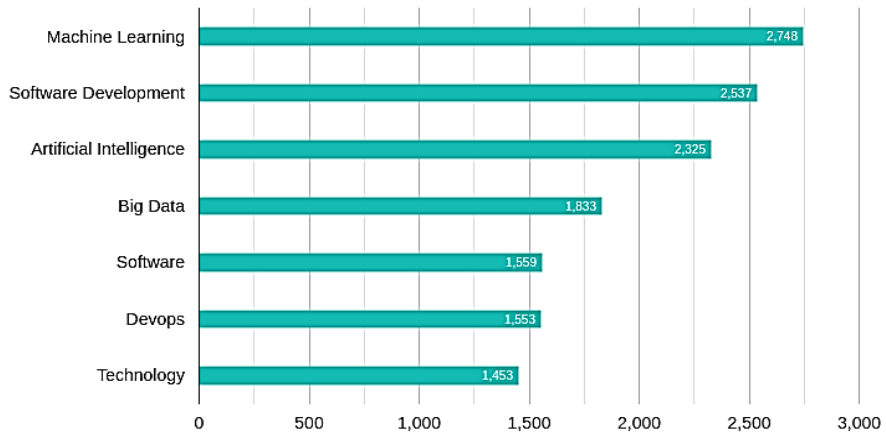


Fig. 6 Containerization Customers by Industry

---

## Docker and Its Impact on Modern Software Development

Mohit Sharma & Dr. M.M.S. Rauthan

## 8. Summary

Docker containers have become popular in modern software development due to their efficiency, portability, and scalability. While challenges such as security and orchestration exist, the ecosystem around Docker is rapidly evolving to address these issues. The continued adoption of containers is expected to grow, driven by trends like edge computing, microservices, and cloud-native development. Docker has transformed the way software is developed, deployed, and managed by providing a lightweight, portable, and efficient solution for containerization. Its ability to package applications with all their dependencies into isolated containers ensures consistency across environments, from development to production.

**\*Masters of Technology**

**\*\*Professor**

**Department of Computer Sciences & Engineering  
Hemvati Nandan Bahuguna Garhwal University  
Srinagar, Garhwal Uttarakhand, India**

## 9. References

- [1] Merkel, D. (2014). Docker: Lightweight Linux Containers for Consistent Development and Deployment. Linux Journal. Retrieved from: <https://www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment>.
- [2] Boettiger, C. (2015). An Introduction to Docker for Reproducible Research. ACM Computing Research Repository(CoRR). Retrieved from: <https://arxiv.org/abs/1504.01450>.
- [3] Docker, Inc. (2023). Docker Documentation.  
<https://docs.docker.com/>
- [4] Statista, 6sense (2025) Top 5 Containerization technologies in 2025.  
<https://www.statista.com/statistics/1256245/containerization-technologies-software-market-share/>
- [5] Bui, T. (2015). Analysis of docker security. arXiv preprint arXiv:1501.02967.
- [6] Felter, W., Ferreira, A., Rajamony, R., & Rubio, J. (2014). An updated performance comparison of virtual machines and linux containers. *technology*, 28, 32.
- [7] Harji, A. S., Buhr, P. A., & Brecht, T. (2013). Our troubles with Linux Kernel upgrades and why you should care. *ACM SIGOPS Operating Systems Review*, 47(2), 66-72.
- [8] Joy, A. M. (2015). Performance comparison between Linux containers and virtual machines. Paper presented at the Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in.

---

## Docker and Its Impact on Modern Software Development

*Mohit Sharma & Dr. M.M.S. Rauthan*

- [9] Russell, B. (2015). Passive Benchmarking with docker LXC, KVM & OpenStack.
- [10] Scheepers, M. J. (2014). Virtualization and containerization of application infrastructure: A comparison.
- [11] Seo, K.-T., Hwang, H.-S., Moon, I.-Y., Kwon, O.-Y., & Kim, B.-J. (2014). Performance Comparison Analysis of Linux Container and Virtual Machine for Building Cloud.
- [12] Turnbull, J. (2014). The Docker Book: Containerization is the new virtualization.
- [13] Van der Aalst, W., Weijters, T., & Maruster, L. (2004). Workflow mining: Discovering process models from event logs. Knowledge and Data Engineering, IEEE Transactions on, 16(9), 1128-1142.
- [14] Varghese, B., Subba, L. T., Thai, L., & Barker, A. (2016). Container-Based Cloud Virtual Machine Benchmarking. arXiv preprint arXiv:1601.03872.

---

## **Docker and Its Impact on Modern Software Development**

*Mohit Sharma & Dr. M.M.S. Rauthan*