iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35152

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

Containerization of Web Application using Docker

Neeraj Sharma[1], Prof. Amita Goel[2], Nidhi Sengar[3], Vasudha Bahl[4]
[1]*Student,* [2]*Professor,* [3,4]*Assistant Professor, Department of Information and Technology, Maharaja Agarsen Institute of technology, Delhi*

Abstract: Containerization in modern times became an efficient way of increasing productivity in IT development and operation processes. As the corporate world keeps growing, people want optimize the use of docker and want the proper usage of this valuable tool. Most of the companies and developers make use of Docker containers because it is a drastic improvement for cloud development, and DevOps more specifically. In this research, I present a detailed explanation of Docker technology which will comprise all the major aspects of docker. Further the paper will give an overview about deployment of web application using docker.

I. INTRODUCTION

A. What is Docker?

Docker is an open-source platform for developing, shipping, and running applications. Docker allows us to distinguish our applications from our infrastructure so we can deliver software short span of time. With help of Docker, we can manage our infrastructure in the similar ways we manage our applications. When taken advantage of Docker's methodologies for deployment, testing, and deploying code quickly, we can imperatively reduce the time lag between code writing and running it in production.

Docker renders the ability to package and run a software or application in an isolated environment called a container. The isolation and security give you the ability to run multiple containers simultaneously on a given host. Containers are easy to use and contain everything needed to run an application, so there is least dependency on what is currently installed on the host.

Docker gives a platform to manage the efficiency and lifecycle of the container:

1) Develop the application and all its component using container.
2) The container then becomes the unit for testing and distributing your application.
3) When ready, deploy the application into the production environment, as an orchestrated service or a container. This works in the same way whether your production environment is a local data center, cloud provider, or a hybrid of the two.

B. What is a Web Application?

A web application in the simplest terms is a computer program that optimizes web browsers and web technology to efficaciously do tasks over the Internet. Web applications are devised to run on Web servers (such as internet) and use Web browsers such as Mozilla Firefox or Chrome as the client interface. Web applications are typically client/server applications. For example, the shopping or E-commerce websites, which help client order products on basis of its request the server responds.

As cited above, Web application is a client-server application program, therefore in the client-server premises, multiple networks (computers) can share information like saving the information into a database. The "client" can be used to fill the information, and the 'server' can be used for storage of the information. The application can be as straightforward as a chat board or a student form on a website, or it can be compounded as a hospital management system or multi-player gaming app that you run on your phone.

There can be two types of web applications dynamic and non-dynamic websites. When going on a non-dynamic website, your web browser will make HTTP requests to download the Frontend technologies (HTML, CSS, and other files) needed to devise the initial webpage that user want to browse. But, once the web page is loaded, the web browser or the website or the web application would not send any other HTTP requests to the web server. On the contrary, dynamic web apps know how to respond to different client requests. More precisely, dynamic web apps generally use JavaScript, Ajax to make HTTP requests to the web server to ask for new content (either on demand of user or depending on how the web app is created).

C. Contribution

Contribution of the paper can be summarized as follows**:**

1) I propose the notion of a dynamic web application which will serve different functionalities.
2) I will then containerize the dynamic website using docker.
3) I will give a holistic view on development, deployment and management of the dynamic web application.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

II. LITERATURE REVIEW

Asha Mandava et al., [1] studied the various technologies which can be used in designing and developing the web applications also studied and compared different web application developing frameworks. Shushant Kumar Bhandi, [2] progressed in the same field and gave an insight on how to tackle we application related problems also how to make a web application more effective and efficient.

Mohammad Ahmadi et al., [3] studied the different functionalities of docker in detail, also gave an analysis of the performance of docker. The study also takes us through advantages and disadvantages of docker. Raghavi Suresh et al., [4] gave comparative analysis between docker and VMware in cloud computing. The study also briefs about virtualization and types of virtualization. Srinath Reddy Meadusani [5] studies how virtualization can be obtained using docker containers. The research starts with brief view of nature and significance of docker container. The research gives a detailed information about methodology and implementation of docker containers. Bernd Harzog, [6] research paper gives detailed expression on management of application in docker containers. The paper gives a detailed study about management of containers during their lifecycle specifically in the production phase.

III. BACKGROUND

The goal providing the background is to give a brief overview of the background of main technologies behind our research, namely web application engineering, cloud computing.

A. Web Application Engineering

Who devised the present web app? The answer of the question brings lies somewhere around the 1990-s, when the Internet was inundated with the word documents in the shape of static HTML pages. After few years it became possible to add video, images and audio files to the web pages. But the substratum didn't change as they remained static. The willingness to make HTML pages responsive and dynamic was efficacious enough for the development of the language Java Script in 1995. It permitted web pages with various interactive and responsive elements, including vector animation. There was a drastic change from static to dynamic web pages in year 2005 with the inception of Ajax, a new way to responsive web design and web app development, asynchronous web apps. The term web app has entrenched its importance in the web history. But it was not the peak, modern problem required a new level and complexities in web apps, as a result in 2015 Alex Russell and Frances Berrman introduced the world to Progressive Web Application (PWA), that are "websites that took all the right contents in perfect blend". In simpler words, web app had the functionality similar to the native app and the client could pin point any difference between the app and PWA.

B. Advent of Cloud Services

Before emerging of the cloud computing, there was Client and Server networking and computing which basically means a centralized storage under which the software apps, all the required data and all the powers are handed on the server side. If a single client wants to hold on to specific data or run a program, then he or she needs to connect to the server and then have the access, and then he or she can do his or her business.

Then came, distributed computing came into frame, where computers are connected together and share their resources, files, and data when needed. On the basis of above need of computing, there was birth of cloud computing services that later changed face of technology world. In 1999, Salesforce.com started rendering of applications to client using a simple static or dynamic website. The applications were rendered to enterprises over the Internet or cloud, and this way the notion of computing sold as utility were true. In the year 2002, Amazon started Amazon Web Services, providing tools like storage, computation and even artificial intelligence. In 2009, Google also came into the race to provide cloud computing applications.

C. Docker

Docker was founded by Kamel Founadi, Solomon Hykes, and Sebastien Pahl during 2010 and launched in 2011. Solomon Hykes lied the substratum of Docker project in France as an inside project within dotCloud, a platform-as-a-service company.

Docker came to the public in Santa Clara at PyCon in the year 2013. It was then released as a open-source in the same year in March.

During that time, it used LXC as its default execution environment. After One year, with the advent of version 0.9, Docker substituted LXC with its own component, libcontainer, which had its code written in the Go programming language.

In the year 2017, Docker created the Moby project for the purpose of open research and development.

IV. THE WEB FRAMEWORK

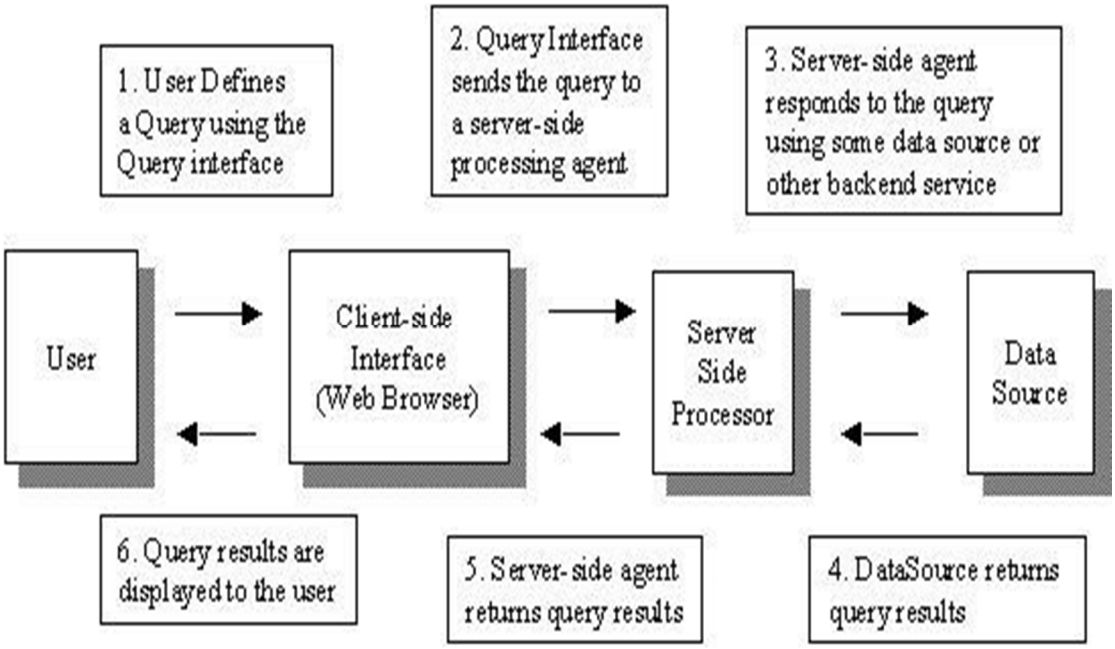


Figure 1: Web framework

The purpose of this section is devoted to render a detailed explanation of the framework constituents. This explanation gives a precise description of the goals, environment and the application.

A. Goal

A goal is an objective the system should achieve through communication of agents (user and software) in the software-to-be and in the environment. The goal of the dynamic website is to develop a social network for the users through which the users across the globe can interact. The major goal which the web page serves is to help people socialize.

B. Environment

By "environment" we mean whatever in the machine provides a surrounding in which the application is supposed to operate. Environment is crucial because it strongly influences the behavior of the application. Further in the terms of AWS the environment helps in identifying which tools are apt for our application. In this application the environment used is visual studio code, which will provide our backend code, frontend code and database a proper domain. We use Node.js for the backend development. Node.js is a runtime environment used for the development of web applications and website development. Its open-source environment allows code to be re-used and re-distributed. Node.js is highly established as a preferred platform to create web-based APIs since it is based on the concept of server-side scripting. The development environment has the capability for creating some of the best real-time web-based applications. Also, I use Socket.IO which enables real-time, directional and easy communication. One of the major advantages of Socket.IO library is it works on every platform, browser or device, while not compromising on reliability and speed.

C. Application

The application is the core of the machine. It implements one or more services or functionalities. Also, the relationship between application and functionality is a many-to-many one, as one application could provide several functionalities. Your application should have clearly defined functionalities which are easy to learn and run. The more your application is user friendly the better it is. The environment could be different according to your comfort but end motive is same that is to create an application with least glitches and a near perfect UI. For the purpose of this research paper, I use a web chat application which uses Socket.IO library for bidirectional communication. Every time a new browser is opened it establishes a new session.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

V. DOCKER FRAMEWORK

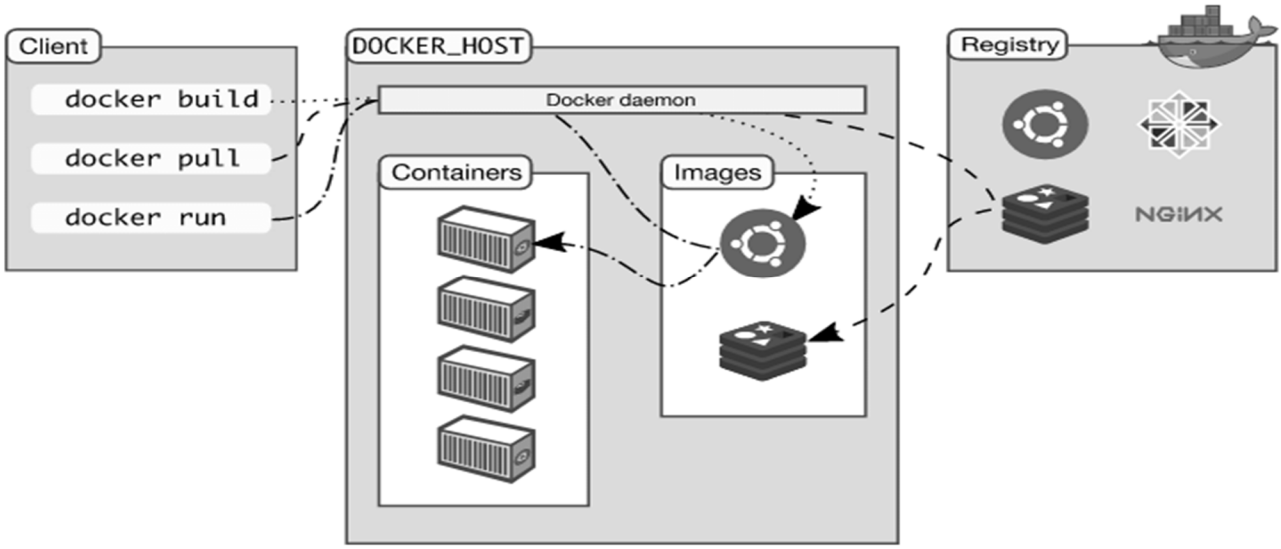


Figure 2: Docker Architecture

Docker has a client-server architecture. The Docker client connects to the Docker daemon, which then does the heavy work of creating, running, and distributing the Docker containers. The Docker client and daemon don't need different systems, you can connect a Docker client to a remote Docker daemon. The Docker client and daemon connect using a REST API, network interface or over UNIX sockets. One more Docker client is Docker Compose, that lets you connect with applications having multiple containers.

A. The Docker Daemon

The Docker daemon listens for Docker API requests and is responsible for managing Docker objects such as images, containers, networks, and volumes. A daemon can communicate with other daemons to manage several Docker services.

B. The Docker Client

The Docker client is the way by which Docker users interact with Docker. Commands such as docker run, client sends the commands to docker daemon, which runs them out. The docker command uses the Docker API. The Docker client can connect with multiple daemons.

C. Docker Registry

A Docker registry is held responsible for keeping docker images. Docker Hub is a public registry that is open to all for use, and Docker is set to look for docker images on Docker Hub by default. You can have your own private registry.

When the docker pull or docker run commands are used, the required images are pulled from your set registry. When the docker push command is used, image made by you is pushed to your configured registry.

D. Images

An image is a read-only template or blueprint having instructions for creating a Docker container. Generally, an image is based on another image, with additional customization. For example, you may build an image which is based on the ubuntu image, but installs the nginx web server and your application, as well as the configuration details needed to make your application run.

You can create your own customized image or use an already present image in the docker registry. To build your own image, you firstly need to create a Dockerfile with the steps needed to create the image and run it. Each instruction in a Dockerfile is a layer in the image. When you modify the Dockerfile and rebuild the docker image, only particular layers which have modified are rebuilt. This is the main reason which makes images so lightweight, small, and fast, compared to several other virtualization technologies.

E. Containers

A docker container is a runnable instance of a docker image. Docker container can be created, started, stopped, moved, or deleted using the Docker API or CLI. You can connect a container to several networks, add storage to it, or create a new image based on its current condition.

A container is well isolated from host machine and other containers. The level of isolation of containers network, storage and other objects from other containers and host machine can be controlled.

A container is defined by configuration options you provide to it when you create or start it and its image When a container is removed, any modifications to its state that are not stored in persistent storage disappear.

VI. CONCLUSION

Dynamic web applications deployed and containerized through Docker is an economical, client friendly, efficient, reliable, scalable, low latency option with dynamic functionality and blazing fast performance.

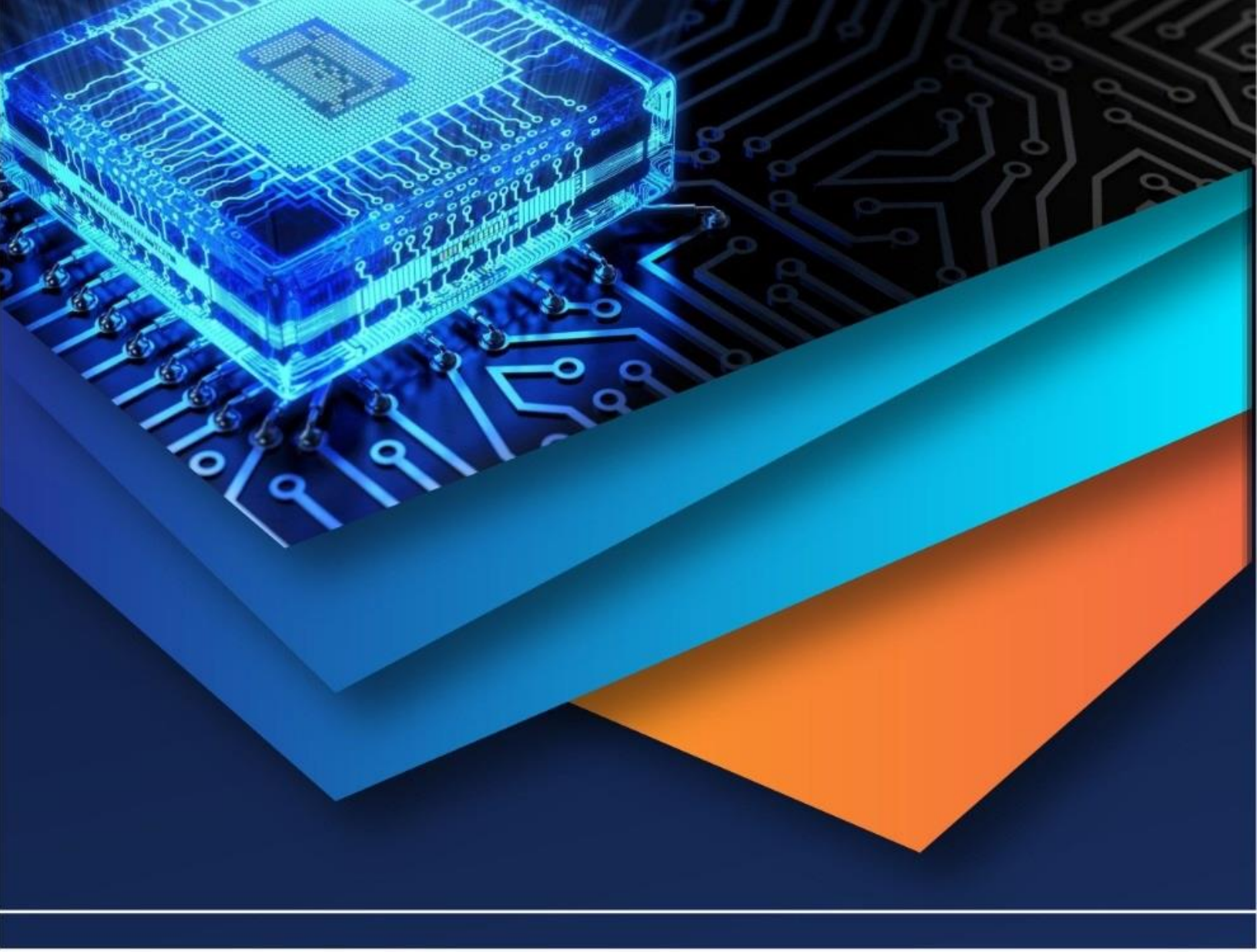
This paper has proposed a framework for understanding for developing dynamic web applications. Further the paper proposed the way to deploy and containerized your web application using Docker. The application is ready to interact with the world. Our approach at every phase or level of the development, deployment and management was to build a user-friendly application. The produced research was understandable, easy to read and had highly detailed diagrams. The conclusion is that the use Docker for Web application deployment and containerization resulted in a good notion from the perspective of the user.

REFERENCES

[1] Asha Mandava and Solomon Antony (March 2012), A review and analysis of technologies for developing web applications, Murray state University Murray, Kentucky, Conference Paper.
[2] Sushant Kumar Bhandi (May-June 2018), Web Application Development and Tackling the Problems, Vivekanand Education Society's Institute of Technology (VESIT), Mumbai, India, published at: IOSR Journal of Computer Engineering (IOSR-JCE).
[3] Babak Bashari Rad, Harrison John Bhatti, Mohammad Ahmadi (March 2017), An Introduction to Docker and Analysis of its Performance, Asia Pacific University of Technology and Innovation Technology Park Malaysia, Kuala Lumpur, Malaysia, published at: IJCSNS International Journal of Computer Science and Network Security.
[4] Prakash P and Raghavi Suresh (March 2017), Comparative Analysis on Docker and Virtual Machine in cloud computing, Dept. of Computer Science and Engineering Amrita University Coimbatore India, published at: International Journal of Pure and Applied Mathematics.
[5] Srinath Reddy Meadusani (May 2018), Virtualization Using Docker Containers: For Reproducible Environments and Containerized Applications, St. Cloud State University, published at: conference paper.
[6] Bernd Harzog (October 2014), Managing Applications in Docker Containers, Analyst – Virtualization and Cloud Performance Management, published at: conference paper.

FIGURES

Figure 1. Web framework: extropia.com/tutorials/devenv/intro_to_app_dev.html
Figure 2. Docker Architecture: https://docs.docker.com/get-started/overview/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY