iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 10 **Issue:** VI **Month of publication:** June 2022

**DOI:** https://doi.org/10.22214/ijraset.2022.44437

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: 7.538*
*Volume 10 Issue VI June 2022- Available at www.ijraset.com*

# Managing and Creation of Images in Graphical Mode

Kukutapu Nithish Kumar[1], Teegala Bhargav Reddy[2], Manelli Mahesh Chary[3], Kondamudi Dayakar[4]
[1, 2, 3]*Student, Department of Electronics and Computer Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India*
[4]*Professor, Department of Electronics and Computer Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India*

***Abstract:*** ***As the number of applications grows, it becomes increasingly difficult to ship and test the software applications. The growth in the creation and testing of software applications caused an issue of dependencies. Virtual machines could eliminate some of the problems, but the number of dependencies continues to increase due to the wide range of requirements for the environments. Virtual machine partitions allow many environments to operate on a single operating system. Containers solved the problem of dependencies. Containers are lightweight, may be utilized to generate a variety of environments, and are used to ship applications from one team to another. The deployment procedure is time-consuming for redesign, especially when adding new features and versions. In this paper, we introduce a web application with features to manage images and containers and introduce the architecture of the web application with Podman. This paper highlights the advantages of using web applications to manage containers and images.***

***Keywords:*** **Virtual machine, Containers, Podman.**

## I. INTRODUCTION

DevOps is a trending word for the past five years in this technological world. The top companies across the world are trying to speed up their development process with the help of the DevOps methodology. DevOps can be defined in many different ways based on requirements and usage, Simply DevOps is a combination of Software development and Operations where development comprises the Build, Code, Test, and Plan phases of the software development lifecycle (SDLC), and Operations comprises of the Monitor, Deploy, Operate and Release phases of SDLC. DevOps is a continuous, agile, and infinite loop method beginning with Development followed by Integration, Testing, Monitoring, Feedback, Deployment, and Operations Phases, all seven phases together are called the DevOps lifecycle. Container orchestration tools are required to manage large scale of containers and multi-tiered applications [13]. We use many different tools like TestNG, JUnit, Selenium, Docker, Podman Jenkins, Chef, Puppet, Ansible, Vargant, and SaltStack in one or many phases of the DevOps life cycle. One of the tools is Podman which is a daemonless container engine for developing, managing, and running Open Container Initiative (OCI) Containers on your Linux System. These containers can either be run by root or by the non-privileged user. Podman can be used in every Operating System (OS) but, in windows, before running the commands we have to initialize a Podman machine, a type of virtual machine that will give you a Linux environment on Windows. All these installation, Initialization, and execution of Podman commands have to be done on Command Line Interface (CLI) which is a hectic task for both professionals and beginners. So, in this paper, we had proposed a web application with the help of this application users can execute podman commands graphically which makes the task easier and quicker.

## II. LITERATURE SURVEY

The existing methodologies could be very complex to understand for the non-DevOps users. The features of the existing system like the ability to start and deploy the containers based on the commands, and the ability to work in only some of the operating systems like Linux distributions [1]. These are some of the disadvantages of the existing model which could potentially lead to many problems and reduce the scope of usage of the containers. Container orchestration tools are required to manage large-scale containers and multi-tiered applications [13]. These tools help build the applications faster and more efficiently. A container could also help in testing and monitoring the application [14]. The need for containers starts to increase gradually due to the increase in applications. The need for automation also increases due to the increase in production of applications so, faster deployment of the applications is necessary to increase the production of applications.

©IJRASET: All Rights are Reserved 

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 10 Issue VI June 2022- Available at www.ijraset.com*

## III. IMPLEMENTATION

The installation of essential software, such as MongoDB, Node.js, and a code editor like Visual Studio, is the first step. Create a pacakge.json file in Visual Studio after installing the required software. This file stores key metadata about a project that is required before it can be published to NPM, as well as functional properties that npm uses to install dependencies, run scripts, and identify the package's entry point. Now, Install the necessary external modules, such as express, Mongoose, and MongoDB. Create an entry file called app.js and finish creating the project's code using the MVC architectural pattern.

### *A. Starting A Web Application*

This application can be hosted anywhere like in AWS cloud, Azure, Google cloud, and local host. Here we hosted this application on localhost by executing the command "node app.js" in the terminal. We can see the result by typing the address as "localhost:port number" in the browser. The port number is nothing but the number given in the app.js file.

### *B. Pull Image From The Repository*

When we enter into the Main page and select Images Option in the website, we will enter into the images page. We should select the pull image button to pull the image from the repository. The web application prompts for the name of the image to pull the image. The web application will send the data input to the server at the backend and the server combines the necessary command with the input and sends the command to the podman engine running at the backend. podman tries to pull the image from the docker hub to the local repository as shown in fig 1.

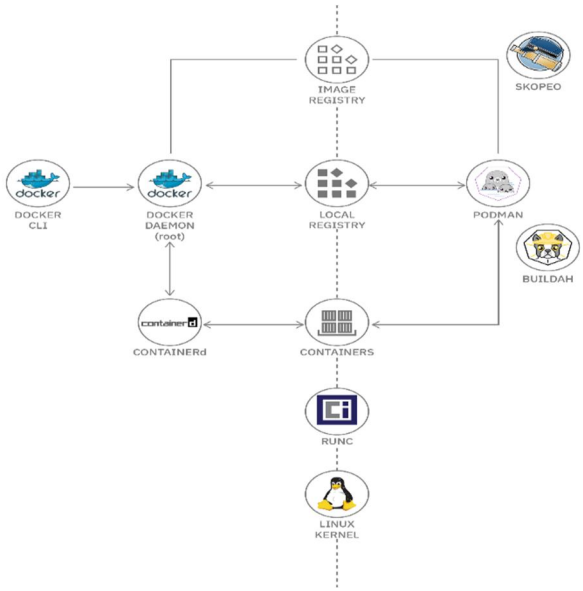



Figure 1: Pulling image from remote repository

### *C. Starting The Container*

When we enter into the main page and select the Containers Option on the website, we will enter into the containers page. We should click the launch container button to start the container. The web application prompts for the name of the image and container. Give desired values to parameters and the web application starts the container. The web application will send the data to the server and the server intern connects to the podman engines and executes the command to run the container. We can quickly develop applications and deploy them to any environment, allowing us to handle a wide range of DevOps problems [1].

## IV. ARCHITECTURE

In this paper, to develop the application, we utilized Node.js, which is an open-source server environment that runs on JavaScript. We have Express on top of node.js, which is a web application framework that aids in the development of single-page, multi-page, and hybrid web applications as well as the management of servers and routes. The architecture of this application can be seen as two parts frontend and backend.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

### *A. Frontend*

The user sees and interacts with the express files with the ".ejs" extension which contains the frontend code in HTML, CSS, BOOTSTRAP, and JAVASCRIPT languages. The user can feel comfortable while working in graphical mode. All the ejs files are in the views folder and if we have any external CSS files, we will place them in the public folder and provide a path to include them in the app.js file. Front end of the application is as shown in the fig 2.

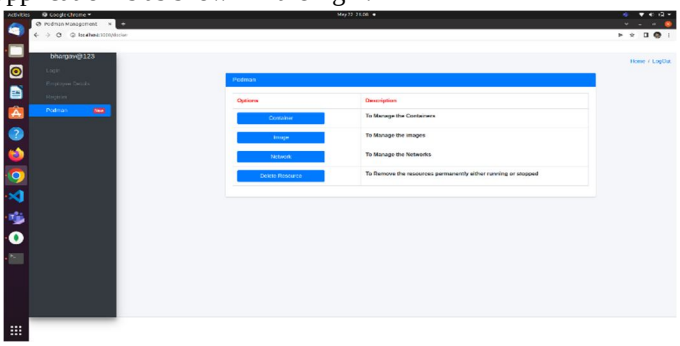


Figure 2: Frontend of Web Application

### *B. Backend*

The backend part of the application contains mainly three things MongoDB, Web Server, and Docker Engine, and these three things communicate with each other where web server acting as a median. With an express module in Node.js, We can create our server, and also, we have an HTTP module in Node.js which is used to transfer data over the Hyper Text Transfer Protocol (HTTP), Here the data may be the request from the user to the web server with the help of a web browser or the response by the web server to the user requests.Node.js and MongoDB both combined leads to a JSON based web-service [15]. The web server interacts with the MongoDB database (To store and retrieve user data and the commands executed by that particular user as logs) and the Podman Engine (To create containers and networks, pull images, use created networks, list stopped and running resources, to delete unuseful resources). The connectivity is as shown in fig 3.




Figure 3: Backend of Web Application

### *C. Database*

In Node.js we have MongoDB and mongoose modules, with the help of them we will connect and communicate with MongoDB from the web server to get required data and to post data into the database. The data is in MongoDB (Which is a NoSQL database) are data is stored as collections and documents. It is used to store user data and user commands as logs. Architecture of the application is as shown in Fig 4.

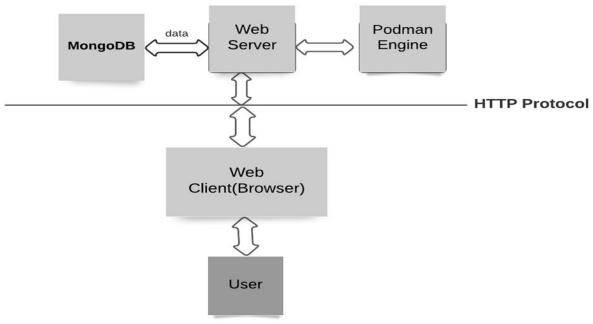



Figure 4: Architecture of the Web Application

## V. RESULTS

The results explain how the Containerization strategy is accurate compared to that of the previous one and ensures that it takes less time for the Containerization of the application. There are multiple strategies followed but using the Podman tool time of Containerization is less and almost equivalent to constant time.

### *A. Feature Analysis Of Proposed Model*

| Features | Docker | Podman | Website |
|---|---|---|---|
| Setting up | Setting up is very simple. | Setting up is a little complex compared to Docker. | Does not need setting up. |
| Daemon-less | Docker cannot run as Daemon-less | Podman can run as Daemon-less | Podman can run as Daemon-less |
| Root | Container has to run by root user | Container can be run by non-root user | Container can be run by non-root user |
| Load Balancing | Simple Load Balancing | Simple Load Balancing | Simple Load Balancing |
| Operating System | Can run on Linux Distributions, Windows with application. | Can run on Linux Distributions, Windows with application. | Can run on any Operating System which has a browser. |
| Monitoring | Logs are not stored. | Logs are not stored | Uses MongoDB for storing logs. |

Table 1: feature analysis of proposed model

The comparison is performed based on features and the requirements of current scenarios. The web application proposed in this paper can eliminate the resource requirements needed to start the existing tools. The proposed web application could help DevOps learners to learn the concepts related to containers quickly and easily. It takes a large amount of time to set up and run containers and document the results. With the help of the web application, we will be able to store logs and document the results.

### *B. Results*

*1)* The first task is to start the Web application from the Browser as shown in fig 5.

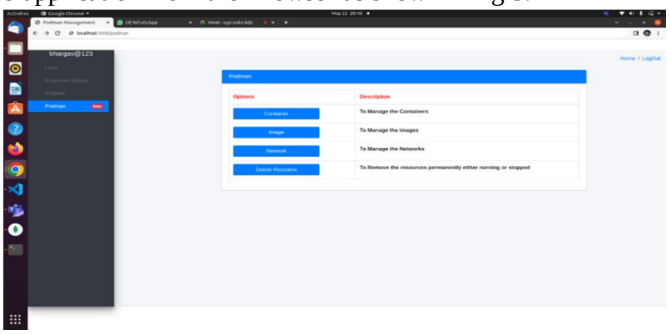


Figure 5: Main Page

*2)* Once the application is launched, we have to select one of the options. First, we can select containers for managing containers as shown in Fig 6.

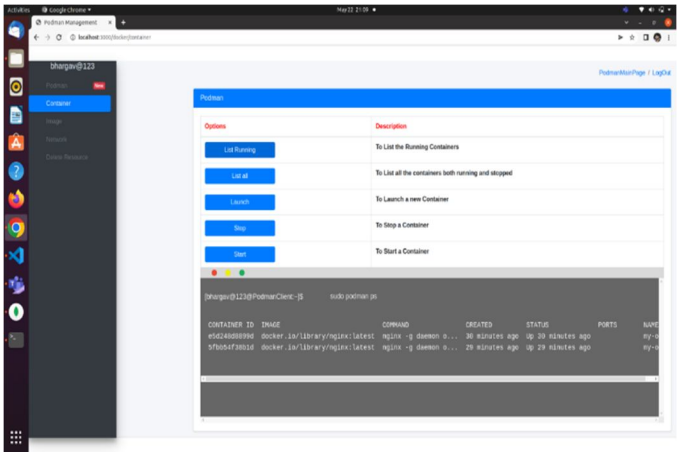


Figure 6: container Page

*3)* We can stop the container by giving the name of the desired container as input as shown in 7.

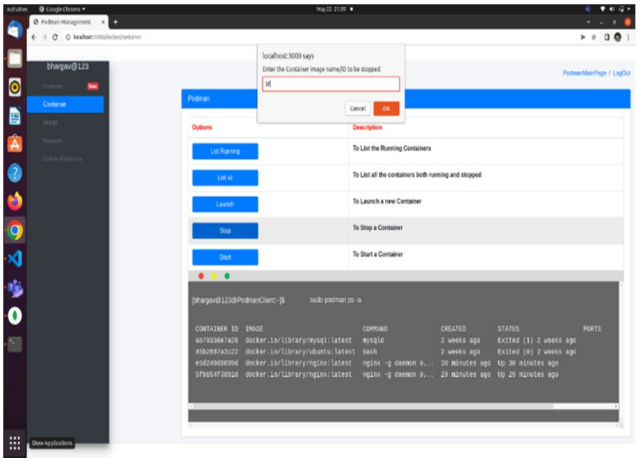


Figure 7: stopping the container

*4)* Select image option from the main page to manage the images as shown in 8.

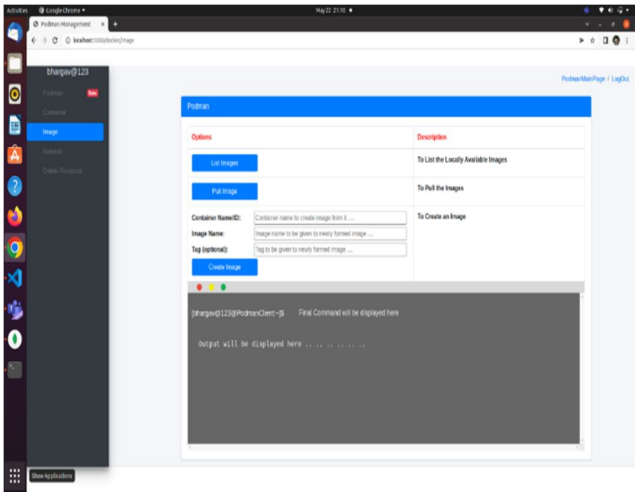


Figure 8: Image Page

*5)* Select Network Option to Manage the network in the containers as shown in 9.

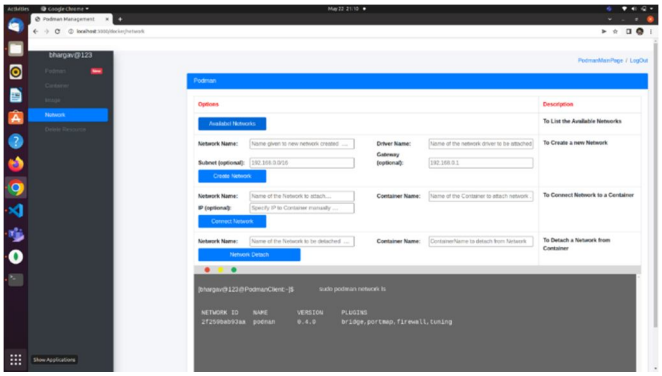


Figure 9: Network Page

*6)* Select the Delete Resource option from the main page for deleting the resources (Networks, Containers, Images) as shown in Fig 10.

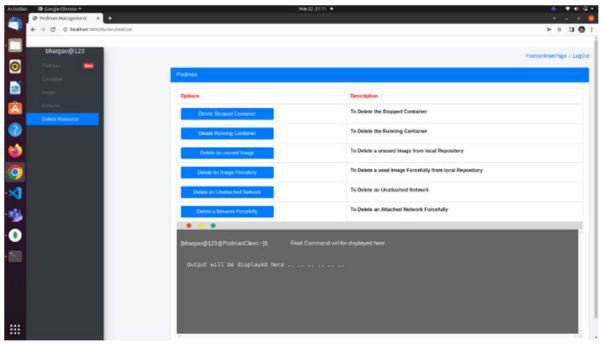


Figure 10: Resource Page

## VI. CONCLUSION AND FUTURE SCOPE

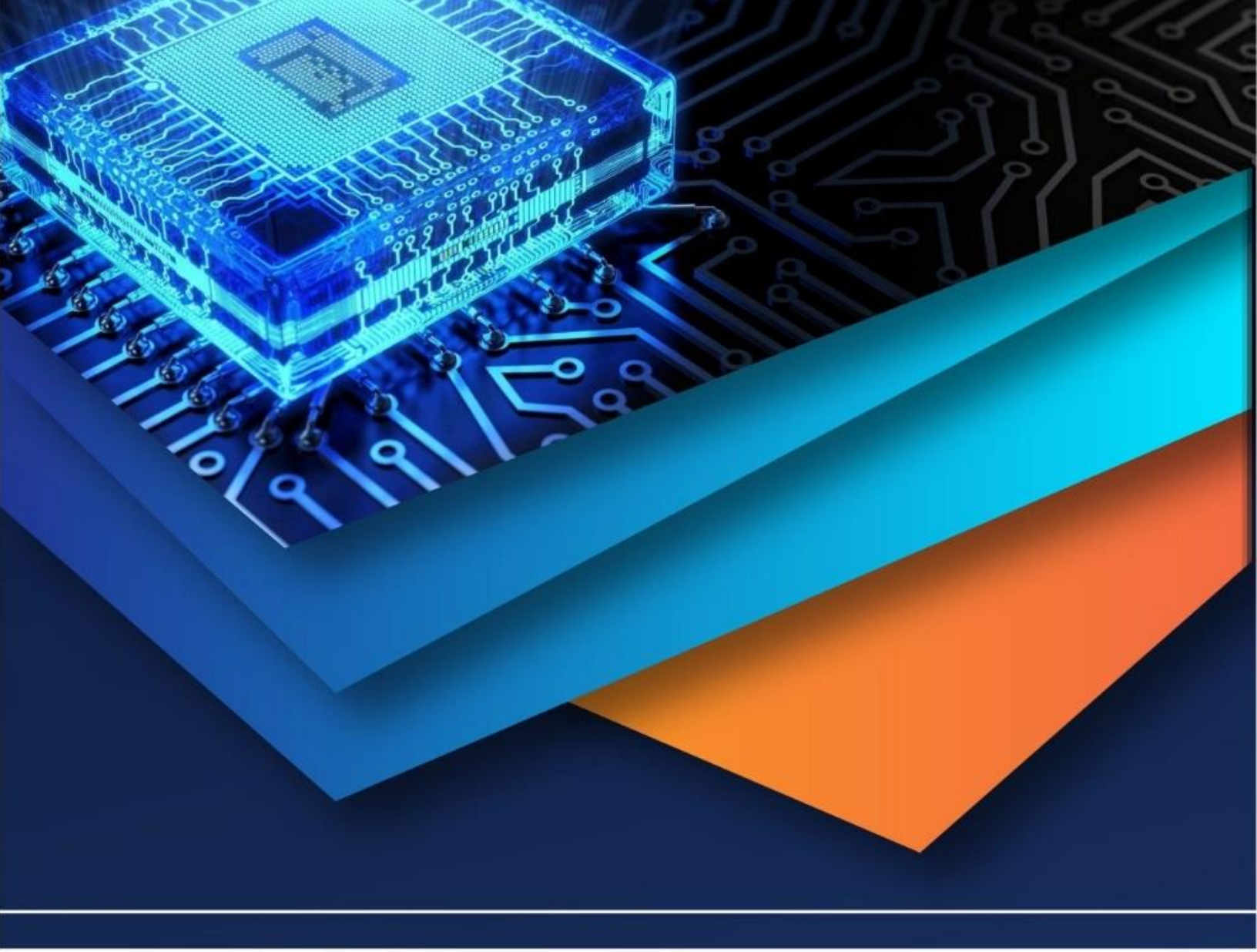
This paper proposes a web application to make DevOps accessible to all learners, enhance DevOps efficiency, and speed up DevOps research. This application makes user interaction as easy and efficient as possible. Image management and Storage of logs are two notable aspects of the whole application. This application builds on the tool called Podman. Podman is used to leverage the concept of pods. Podman Commands are used in the graphical user interface to create and manage the images in different operating systems. In addition to the current trend, a similar concept of automation can be applied in the process of Containerization, to automate the development of the image. We can increase the features to operate the application in the voice mode to faster access the application. The application can be extended to monitor the applications. The application Can be extended to be used in the field of BigData.

## REFERENCES

[1] S. Singh and N. Singh, "Containers & Docker: Emerging roles & future of Cloud technology," 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2016, pp. 804-807, doi: 10.1109/ICATCCT.2016.7912109.
[2] W. Xia et al., "Design and Implementation of Docker-based Image Repository Management System of Dispatching and Control Cloud," 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2021, pp. 512-516, doi: 10.1109/IAEAC50856.2021.9390890.
[3] N. -T. Chau, J. Yoon, T. -P. Doan and S. Jung, "AppPACK: A Packaging Model for Single-Purpose Lightweight Virtualization Environment," in IEEE Access, vol. 9, pp. 30071-30079, 2021, doi: 10.1109/ACCESS.2021.3055856.
[4] B. Đorđević, V. Timčenko, M. Lazić and N. Davidović, "Performance comparison of Docker and Podman container-based virtualization," 2022 21st International Symposium INFOTEH-JAHORINA (INFOTEH), 2022, pp. 1-6, doi: 10.1109/INFOTEH53737.2022.9751277.
[5] F. Paraiso, S. Challita, Y. Al-Dhuraibi and P. Merle, "Model-Driven Management of Docker Containers," 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), 2016, pp. 718-725, doi: 10.1109/CLOUD.2016.0100.
[6] C. Pahl, A. Brogi, J. Soldani and P. Jamshidi, "Cloud Container Technologies: A State-of-the-Art Review," in IEEE Transactions on Cloud Computing, vol. 7, no. 3, pp. 677-692, 1 July-Sept. 2019, doi: 10.1109/TCC.2017.2702586.
[7] V. Sharma, H. K. Saxena and A. K. Singh, "Docker for Multi-containers Web Application," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020, pp. 589-592, doi: 10.1109/ICIMIA48430.2020.9074925.
[8] J. Ma, B. An, D. Cao and X. Chen, "Comparing Container-Based Microservices and Workspace as a Service: Which One to Choose?," 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE), 2018, pp. 240-245, doi: 10.1109/SOSE.2018.00040.

[9] G. Liu, B. Huang, Z. Liang, M. Qin, H. Zhou and Z. Li, "Microservices: architecture, container, and challenges," 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2020, pp. 629-635, doi: 10.1109/QRS-C51114.2020.00107.
[10] N. Naik, "Docker container-based big data processing system in multiple clouds for everyone," 2017 IEEE International Systems Engineering Symposium (ISSE), 2017, pp. 1-7, doi: 10.1109/SysEng.2017.8088294.
[11] P. Dziurzanski and L. S. Indrusiak, "Value-Based Allocation of Docker Containers," 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), 2018, pp. 358-362, doi: 10.1109/PDP2018.2018.00064.
[12] H. Kang, M. Le and S. Tao, "Container and Microservice Driven Design for Cloud Infrastructure DevOps," 2016 IEEE International Conference on Cloud Engineering (IC2E), 2016, pp. 202-211, doi: 10.1109/IC2E.2016.26.
[13] R. Muddinagiri, S. Ambavane and S. Bayas, "Self-Hosted Kubernetes: Deploying Docker Containers Locally With Minikube," 2019 International Conference on Innovative Trends and Advances in Engineering and Technology (ICITAET), 2019, pp. 239-243, doi: 10.1109/ICITAET47105.2019.9170208.
[14] R. Madhumathi, "The Relevance of Container Monitoring Towards Container Intelligence," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2018, pp. 1-5, doi: 10.1109/ICCCNT.2018.8493766.
[15] A. J. Poulter, S. J. Johnston and S. J. Cox, "Using the MEAN stack to implement a RESTful service for an Internet of Things application," 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), 2015, pp. 280-285, doi: 10.1109/WF-IoT.2015.7389066.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)