

# Implementation Of Docker Container On Local Network By Applying Reverse Proxy

1<sup>st</sup> Henni Endah Wahanani  
Department of Informatics  
Faculty of Computer Science  
University of Pembangunan Nasional  
"Veteran" Jawa Timur  
Surabaya, Indonesia  
[henniendah.if@upnjatim.ac.id](mailto:henniendah.if@upnjatim.ac.id)

2<sup>nd</sup> Mohammad Idhom  
Department of Sains Data  
Faculty of Computer Science  
University of Pembangunan Nasional  
"Veteran" Jawa Timur  
Surabaya, Indonesia  
[Indonesiaidhom@upnjatim.ac.id](mailto:Indonesiaidhom@upnjatim.ac.id)

3<sup>rd</sup> Kiki Yuniar Kristiawan  
Department of Informatics  
Faculty of Computer Science  
University of Pembangunan Nasional  
"Veteran" Jawa Timur  
Surabaya, Indonesia  
[kiki.nagh.arwolves@gmail.com](mailto:kiki.nagh.arwolves@gmail.com)

**Abstract**— Virtualization is an implementation of a software. Virtualization technology has changed the direction of the computer industrial revolution by reducing capital costs and operating costs. The availability of a virtualization will also increase the availability of higher services and data protection mechanisms. Docker is configured to create multiple containers on a network, each container containing one image. The three containers will be created in one compose where each container is connected to each other for wordpress configuration and two composes will be created. Furthermore, from each compose a reverse proxy configuration is carried out which aims to set a different domain address. Lighten computer performance and can reduce the required storage so as to make hosting more effective and efficient. Containers also provide a security advantage over complete control over management running on separate, isolated containers.

**Keywords**— *Container, Docker, Reverse Proxy, hosting*

## I. INTRODUCTION

Virtualization technology has changed the direction of the computer industrial revolution by reducing capital costs and operating costs. The availability of a virtualization will also increase the availability of higher services and data protection mechanisms. In this virtual application, it will make it easier to run and implement a hosting implementation mechanism using containers in an application virtualization.

Virtualization is an implementation of a software or other that will be implemented in one candela with the same work and in the same function but still one of the software is a virtualization.

Hosting is a website that will display in cyberspace by allocating space on a web server and storing files that will be used on the web so that they can be viewed online. Proper hosting means having access to the right allocation of resources to keep your website loading fast and reliably for visitors.

A web service is an interface that describes a set of operations that can be accessed on a network via standardized XML messages. Web service is a software system designed to support machine-to-machine interoperation on an internet network. Web services wrap operations into an interface written in XML notation. This interface hides the service

implementation details. Information exchange that occurs in a web service also uses messages in XML format.

CMS is a software used to manage content. What is meant by content is all forms of digital information, in the form of image files, audio, video, text and other computer files. CMS is in the form of a web-based application that is implemented using a programming language that has web-based programming capabilities.

In this study, the author intends to conduct an in-depth analysis of opinions on the use of hosting on containers. By using a docker hub that has been provided and can be classified and visualized. Later, the results of that will be applied to the Linux Ubuntu 18.04 Operating System.

## II. METHODOLOGY



Fig. 1. Topology

In Figure 1, it is expected to be able to apply a reverse proxy to the container process on the wordpress website. And can access on every operating system that has entered a network and has added DNS domain and IP addresses that have been set in the hosts for each operating system used.

In a case study we will implement a reverse proxy to set a domain to be addressed to an IP address with a different port. From this problem, we use the main server, which is using apache2 to manage the course of a domain with the reverse proxy method. And to use wordpress you will need a database

and a webserver, the database to be used is MySQL and the webserver used is Nginx.

#### A. Literature Study

This stage is carried out to collect some literature that will be applied to the method process to be implemented. To provide an convenience in configuring the design of a network, several applications will be carried out. So, using an implementation of an application virtualization called docker container and implementing a reverse proxy for the share in the port to be used in the domain address sharing. And will also provide a speed in the process of requesting to open a page. In its application, it has made a number of references which were developed through several books for an additional study of the literature.

#### B. Analysis

In the smooth running of a research it will take some tools and materials to build a system that will do a software / software and hardware / hardware. There are several system requirements needed:

##### 1) Server

In its application the web server and main server run on the Linux Ubuntu 18.04 LTS operating system. In a Local Linux OS with several reasons for development on a Local Server, namely: Full Administration As Owner (root), Free to install and uninstall software, Stable server (reduced downtime), Have a Local IP Network, Access enough server resources, Easy web management. The specifications for the Virtual Private Server used are as follows:

TABLE I. SPECIFICATION SERVER

CPU	RAM	Storage	OS
4Core	16GB	60GB SSD	Ubuntu 18.04 LTS

Table 1 shows the local server specifications used in a study using a 4core CPU. RAM (Random Access Memory) with a capacity of 16384MB / 16GB and Unlimited Bandwidth. By Using the Ubuntu 18.04.5 LTS Linux Operating System.

##### 2) Docker

Docker is used for an implementation of a case study that will run through a virtual application. The Docker is an application of the container taken from the provided images so that in processing it will be easier to handle in the configuration of an application in the case study. The containers used are:

TABLE II. LIST CONTAINER

Compose					
Container	Port	Wp1	Port	Wp2	Port
Portainer	860	nginx	90	nginx_1	91
		PhpAdm	8000	PhpAdm1	8001
Apache-Proxy	80	Wp	9000	Wp1	9001
		Wp_db	3306	Wp_db1	3307

#### 3) Docker Compose

Docker compose is used in a configuration to perform a one-time processing up to run multiple images that have been connected in a network for a case study. and in a configuration, the user will be able to set it according to his wishes to apply in a research

#### C. System Build

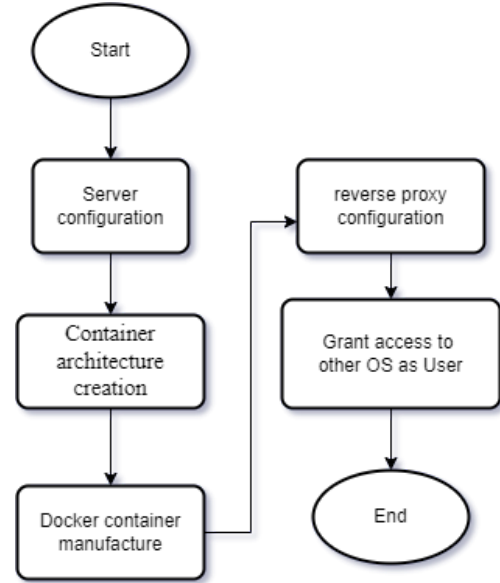


Fig. 2. System creation flow

Figure 2 shows the flow for a process of making the system to be built. The flow shows several processes, namely:

##### 1) Server Configuration

The process is carried out to prepare the server so that it can be used according to the specifications that will be needed. On the server it will be used using the Ubuntu 18.04 LTS operating system

##### 2) Container architecture creation

In this process, the design process will be carried out to build a project that runs in a container along with the resources that will be needed (image, network, volume, port, and environment) that will be needed in the research process.

##### 3) Docker container manufacture

This process is in a process that is entered after setting up from the Container architecture. For the process of making Docker Compose, the first thing is that you need a file with the file name docker\_composer.yml. the format used is yml or yaml, all configurations of images will be made into a container and will provide connecting services from several containers that will be created in a docker compose to be able to run the docker compose.

##### 4) Reverse Proxy configuration

In this process it is done to configure a link and port that has been run. The reverse proxy will direct from each url to the domain that has been provided to be accessed using a domain that has been configured using the reverse proxy.

#### 5) Grant access to other OS as User

Granting access rights to the file host server and other operating systems that will access a page in the container as a user.

#### D. Testing

Testing this system is an application of a test to find out from a system whether there is an error or bug in a system. but in the system or development that I have built this is an application in the application of docker containers to run from several applications that have been connected in several applications that are safe to use. and will carry out a success check in the development of the docker container implementation.

For the parameter in the trial is the speed of the web server for accessing in large quantities. And testing for an ease in building configurations in a container with the application of a reverse proxy and its management.

So in the Steps for testing it with a test scenario. Testing with access to 10 times the action of loading the first and second web on a wifi network with the same speed capacity by using the Google Developers Lighthouse feature provided by Google will issue a performance value in each web from Docker.

From the test scenario above, the test flow is obtained as follows:

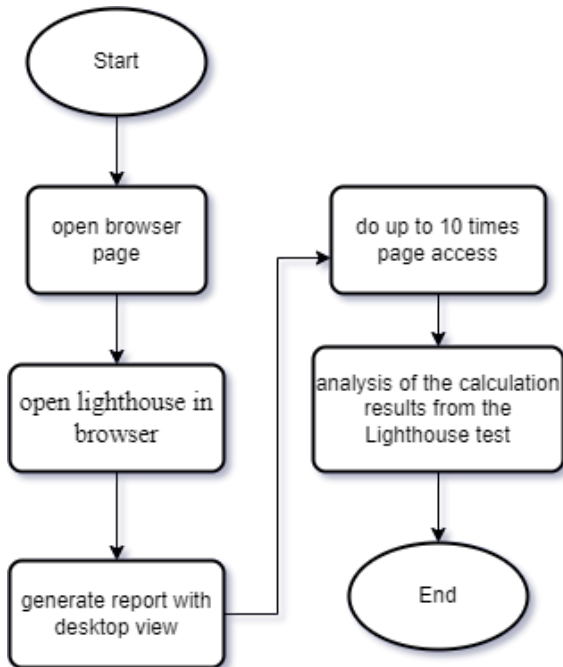


Fig. 3. System test flow

The flow of the test in Figure 3 can be explained that Lighthouse will test the speed of each website that is being opened in the browser on the intended website. In Lighthouse there will be several values that will be displayed, one of which is the performance of the web that is being opened

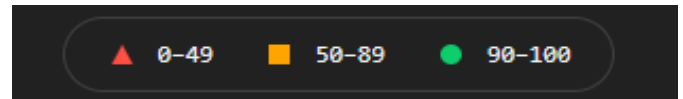


Fig. 4. Lighthouse Performance color coded score

From Figure 4 it can be explained that the score value is from the color code score. And on the results of the website that implements the hosting method using a Docker container for the hosting system development method and also adds a reverse proxy as an application to the designated domain address and port. The results shown can be seen from the test value of the wordpress home page, the performance value of the user load 10 times in a row is very good.

And it can also be explained that the score values of the color coded metric scores and performance scores are colored according to the following ranges:

- 0 – 49 (Red) : Bad
- 50 – 89 (Orange) : Need repair
- 90 – 100 (Green) : Good

To provide a good user experience, a site should strive to have a good score (90-100). A "perfect" score of 100 is both challenging to achieve and not expected. For example, taking a score from 99 to 100 requires the same amount of metric improvement as 90 to 94.

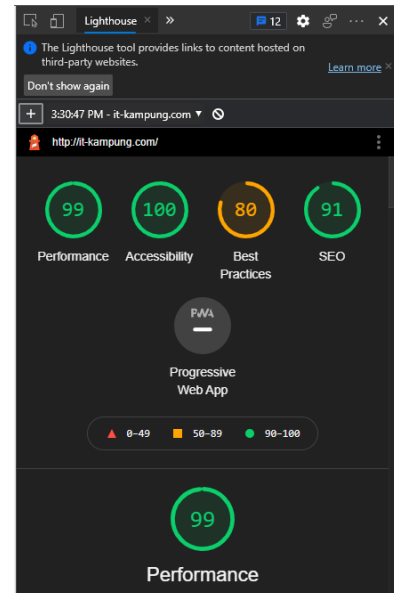


Fig. 5. Google Developers Lighthouse test view

Much of the variability in your Performance score and overall metric scores is not due to Lighthouse. When your Performance score fluctuates, it's usually due to a change in the underlying condition. Common problems include :

- A/B testing or changes to ads served
- Internet traffic routing changes

- From this test, it will be possible to validate that by using the test and also with the same specifications but different results. Here are the test parameters:

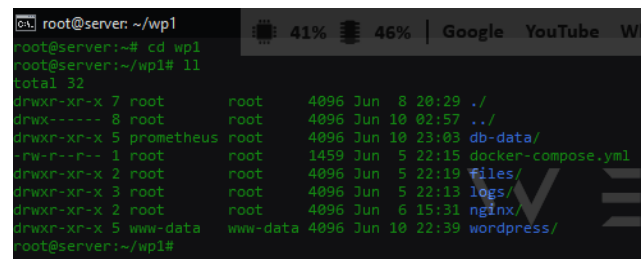
Testing	Parameters	Total Tread	Internet Speed
Lighthouse	HTTP	10	20MBps

After testing the system, then analyzing the results of the Lighthouse testing that has been obtained. So that you get the final result of the way the hosting system uses a docker container and also a reverse proxy.

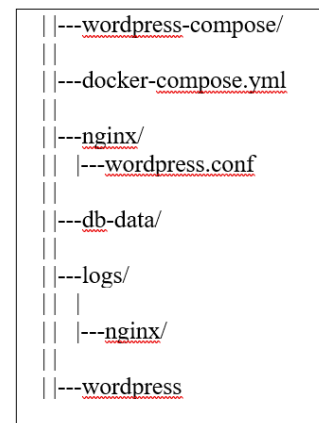
The results of the application of docker in building a hosting implementation using containers in an application virtualization. It can be seen in the following reviews.

The screenshot shows the Portainer Docker dashboard interface. At the top, the 'Portainer.io' logo is on the left, and 'Portainer support' and 'server #1234567890' are on the right. The main header is 'Dashboard' with a sub-header 'Endpoint summary'. The left sidebar contains navigation links: Home, LOCAL, Dashboard, Add Templates, Stacks, Containers, Images, Networks, Volumes, Events, Host, GETTING STARTED, Extensions, Users, Endpoints, Integrations, and Settings. The main content area is titled 'Endpoint info' and shows details for a local endpoint: 'local' with a status icon, '1.2 GB' of memory, and 'Standalone 20/10/7'. Below this, there are fields for 'URL' (value: '/var/run/docker.sock') and 'Tags'. A summary section displays four metrics: '2 Stacks' (with a stack icon), '8 Images' (with an image icon and '1.8 GB' of size), '11 Containers' (with a container icon and status indicators: 3 healthy, 2 running, 0 unhealthy, 0 stopped), and '6 Networks' (with a network icon). At the bottom, a status bar indicates 'A newer version is available'.

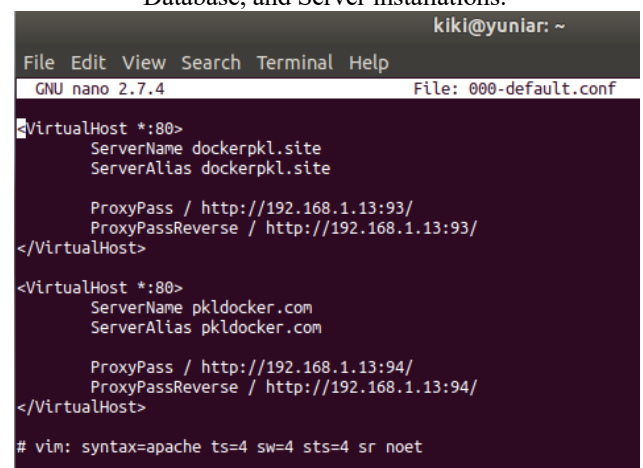
Portaine is a management that is intended for containers that are running or not. In the installation process, it aims to provide a User Interface for Container users and provide convenience in carrying out a configuration in the container.



In the picture above, you can see what is needed and to run docker compose, you must enter the folder in which there is a docker-compose.yml file. the file that will execute from the webserver implementation construction configuration.



In Figure 8 is an arrangement for the folders that must be prepared and the compose files to apply the WordPress, Database, and Server installations.



In Figure 9 it can be seen that the configuration of each port will make one virtualhost in it and provide the configuration of the port address and domain that will be used for calling ports that have been entered in the virtualhost.

After configuring the 000-default.conf file for reverse proxy on every website and required container. And then reload the apache2 system with the @service apache2 reload command, it will reload the apache2 system and execute the configuration that has been set in the 000-default.conf file.

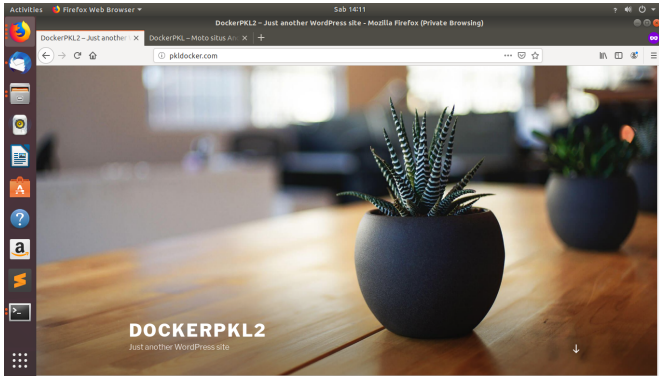


Fig. 10. Docker has successfully appeared using the domain name

And after the reverse proxy configuration process is complete, the next is the trial process when accessed using a domain that has been set in the apache2 configuration. As in Figure 8, it has been successfully accessed using the domain that has been set.

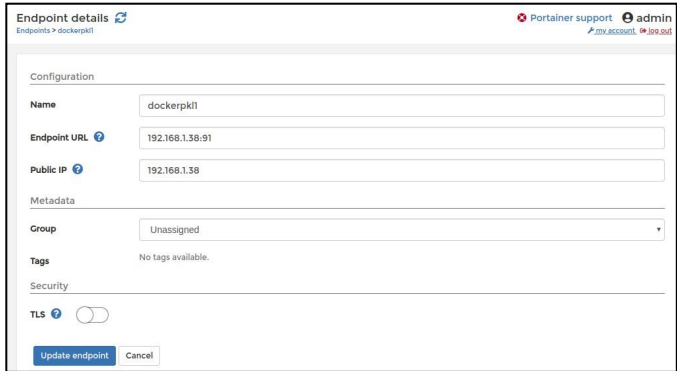


Fig. 11. The process of adding a new endpoint

Next, go to the portainer dashboard to remotely web pages into a network by selecting Endpoints. So that other operating systems that are connected to the same network can access the web page that has been running on the docker. After entering the endpoints, the next step is to add access to the wordpress remote process according to the IP and port links that have been used.

Then fill in the remote name to give the name of the remote process at the endpoints. Endpoint URL [Linux IP Address]:[nginx port in docker-compose], Linux IP Address.

- An example of a URL in Figure 11 : 192.168.100.38:90
- Public example which is in Figure 11 : 192.168.100.38

After adding the process to add new endpoints, it will then be displayed for a list of endpoints that have been created or active endpoints as shown in figure 12.

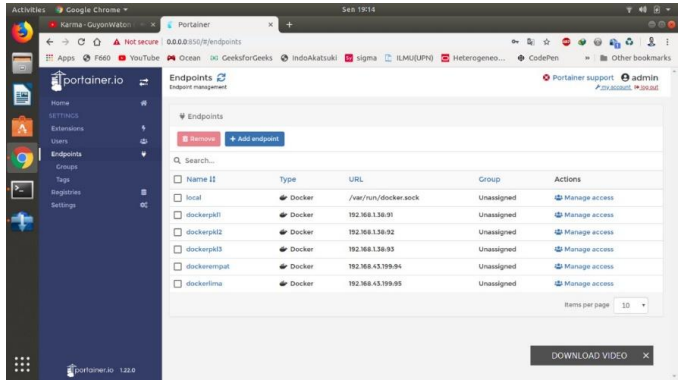


Fig. 12. View list of available edpoints

Next it will give access to other OS devices that want to be able to access the page that has been run on the virtual docker container with the domain page and IP address that is running. It can be seen in Figure 13 which is the place for IP access to be accessed in a browser.

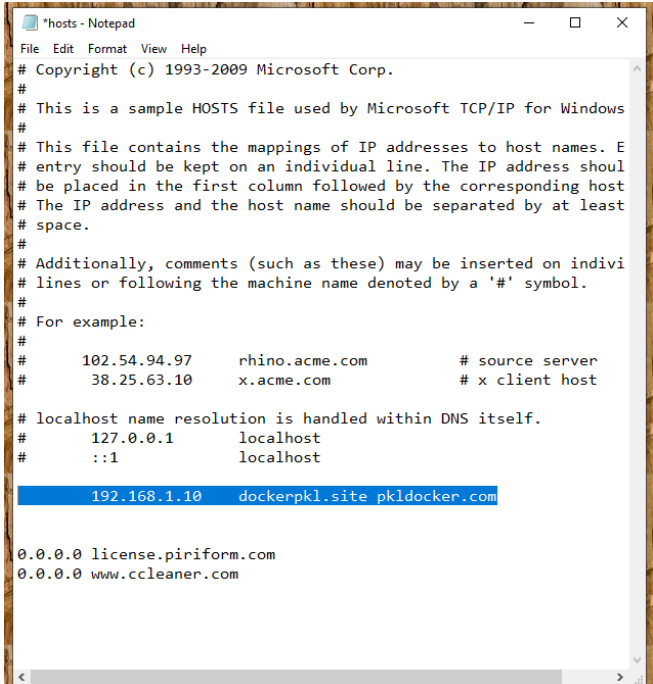


Fig. 13. Add linux IP and domain

#### IV. TESTING

In this section, we will test the throughput speed process to access websites using docker and reverse proxy configurations. The test will be carried out using the Lighthouse Google Developers method. The results will be known how the speed of the website that has been run. And for this test using the same specifications.



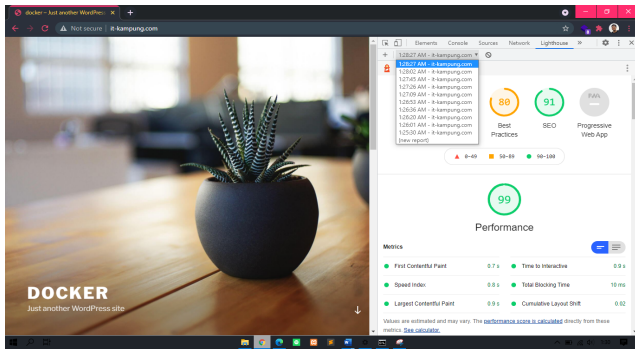


Fig. 14. Lighthouse Testing

From the picture above, the value shown is very good, until it reaches a score of 99 Performance out of 100 perfect scores. With internet speed and the same website model, the value of a website that implements a docker container and with a reverse proxy makes the loading process lighter and faster

TABLE IV. GOOGLE LIGHTHOUSE PERFORMANCE TEST RESULTS

Websi te Test	Load User										rang e
	1	2	3	4	5	6	7	8	9	10	
Websti te	93	93	92	94	92	94	88	95	73	91	90,5

From the results in table 4 it can be seen using the Google Lighthouse Performance test which was tested on a website that uses the Hosting method system which can be seen from the test value of the wordpress home page page getting an average value of 90.5 for the performance of the user load 10 times in a row - come along.

## V. CONCLUSIONS

Based on the author's experimental results, the conclusions that can be drawn from research related to Docker Container implementation on Local networks by implementing a Reverse Proxy are that Reverse Proxy can manage IP and Port addressing to the domain more easily. By applying this method, it will be easier and produce good performance, which is 90.5% for average performance.

Suggestions for further research from the author is to use an additional SSL certificate system to get https access on a website. added an additional package for a security monitoring on the website in order to show it to be more secure.

## REFERENCES

- [1] D. S. D. M. & Y. W. Afis, "Load Balancing Server Web Berdasarkan Jumlah Koneksi Klien Pada Docker Swarm," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2019.
- [2] Anselmi, " Combining Size-Based Load Balancing with Round-Robin for Scalable Low Latency," *IEEE Transactions on Parallel and Distributed Systems*, 2020.
- [3] D. R. S. D. A. H. D. S. Daniel Fraunholz, "Cloxy: A Context-aware Deception-as-a-Service Reverse Proxy for Web Service," *Novel MTD Frameworks and Techniques*, pp. 40-47, 2018.
- [4] dev, "Retrieved from Lighthouse performance scoring," 21 oktober 2021. [Online]. Available: [https://web.dev/performance-scoring/?utm\\_source=lighthouse&utm\\_medium=devtools](https://web.dev/performance-scoring/?utm_source=lighthouse&utm_medium=devtools).
- [5] docker, "Retrieved from Docs Docker Documentation," Docker Documentaion Team, 2018. [Online]. Available: <https://docs.docker.com/>. [Accessed 23 juni 2021].
- [6] Kinsta, "knowledgebase," kinsta, 2 September 2021. [Online]. Available: <https://kinsta.com/knowledgebase/content-management-system/>. [Accessed 9 Desember 2021].
- [7] N. Naik, "Migrating from Virtualization to Dockerization in the Cloud: Simulation and Evaluation of Distributed Systems," *IEEE 10th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Environments*, pp. 1-8, 2016.
- [8] namecheap, "what is web hosting definition," namecheap, [Online]. Available: <https://www.namecheap.com/hosting/what-is-web-hosting-definition/>. [Accessed 9 Desember 2021].
- [9] S. S. Raj Agarwal, "An Efficient Technique for finding SQL Injection using Reverse Proxy Server," *International Research Journal of Engineering and Technology (IRJET)*, vol. VI, no. 09, pp. 1564-1569, 2019.
- [10] O. C. B. M. A. H. H. R. Rizki Agung Muzaki, "Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall," *International Workshop on Big Data and Information Security (IWBIS)*, pp. 85-90, 2020.
- [11] A. H. J. S. D. S. Mohammad Hosseini, "Session-persistent Load Balancing for Clustered Web Servers without Acting as a Reverse-proxy," *Institute of Electrical and Electronics Engineers*, pp. 360-364, 2021.
- [12] k. sharma, "Efficient load balancing using improved central load balancing technique," *International Conference on Inventive Systems and Control (ICISC)*, 2018.
- [13] X. H. Zhenzhong Yang, "Dynamic Configuration of Reverse Proxy Cache based on Multi-Dimensional Time Series Prediction of Visit Traffic," *Institute of Electrical and Electronics Engineers*, pp. 237-240, 2016.