Effectiveness of Cloud Docker Container In Supporting ERP System Development

Ahmad Wijayanto
Master of Information Systems, Diponegoro University, Indonesia
Email: ahyat2011@gmail.com
Eko Sediyono
Computer Science, Satya Wacana Christian University, Indonesia
Email: eko@uksw.edu
Aris Puji Widodo
Faculty of Science and Mathematics, Diponegoro University, Indonesia
Email: arispw@gmail.com

---**ABSTRACT**---

Technology in business is always evolving with high scalability and elasticity. Enterprise Resource Planning (ERP) with new architectures and techniques is constantly being developed and studied. The company always has problems in the process of development, revolution, operation and documentation on the company's products and systems themselves. With the study developed in this journal, it is hoped that it can help developers and companies to implement the effectiveness of the system with flexible development. This study explores the use of Docker, Content Delivery Network (CDN) and Gitlab for development and operations in implementing Enterprise Resource Planning (ERP) systems with odoo frameworks in cloud data centers. The result is structurally more flexible and efficient, and more organized in documentation. This study provides guidance for companies looking to implement cloud-based ERP, governments that have complex infrastructure system service governance and cloud ERP provider vendors.

Keywords – **Effectivity, Docker, Container, ERP, Odoo.**

Date of Submission: Nov 11, 2022 Date of Acceptance: Dec 01, 2022

I. INTRODUCTION

Many companies are trying to innovate to cope with the fast-paced and ever-changing business landscape. In some cases, legacy systems with monolithic architecture are unable to adapt to the needs of very high demands. The ability of virtual machines to run different operating systems adds to the large number of resource resources needed in operation. Then each operating system has large files and needs one virtual machine to run one operating system.

The problem occurs at a time when the scale of the amount of data is getting bigger, to scale up there is no other way by buying a new server running on one server and segmenting a part of the users into the new system. Thus, there is a need for data reconciliation in the new system. When the hardware is experiencing problems, repairs need to be made so that it must be moved on the server first to the shadow server which of course takes a lot of time and is inefficient.

Manual development in terms of setting up and configuring server infrastructure is certainly time-consuming and draining. Once finished, if one component fails to operate, allowing other components will be constrained. While the current system is known to have many services that must be presented. A slow process can cause one or two testing processes to pass which can be inconvenient for the entire company.

Docker is different when compared to virtual machines, docker runs on top of an operating system that can split services and even merge services without the need to create a new operating system. Many enterprises use a variety of architectures to accelerate procedures and optimize the cost of developing and deploying applications on top of the cloud. Docker can be an alternative in making it easier to develop applications that have many services in it. [17]

In cloud service providers, docker has become a trend of technological development with open-source advantages and can run any service in it. Then is this docker effective enough in implementing ERP in companies. In this study, the odoo framework was used as a case study in the application of ERP systems. In this study, it is expected to be able to answer the increasingly complex needs of companies with the results of the system applied.

Basically, ERP is a long-term strategic investment, which is why evaluation must be carried out to keep up with existing technology. The hope is that organizations can have various challenges with the realization of benefits after the implementation of ERP system development, such as communication challenges, solutions and system complexity that can continue to be developed starting from managing problems, methods, tools, and technology of the system.

II. LITERATURE REVIEW

Cloud providers are now moving from centralized, large-scale data centers towards a more distributed multi-cloud setup commanded from a larger and smaller network of virtualized infrastructure runtime nodes. Containerization is currently discussed as a lightweight virtualization solution [10]. Complex ERP systems require a more effective and lightweight solution than Virtual Machine Hypervisor based virtualization technology. A typical company will have a vast number of activities, policies, and processes that help manage and mitigate digital risks [6].

Hypervisor virtualization technology is used to manage services with a structure directly within the operating system. Unlike docker, which can set up smaller structure services and clustering. By using containers, the system can be run portable and lightly, containers can develop, test, and deploy applications to a large number of servers and can connect containers with one an-other. This allows the development of applications by means of microservices or microservices [19].

Each container is an instance of an image, which describes a software stack with dependencies required by the application, compiled and configuration files copied in an image. The images used are manufactured, optimized for performance and security. In a ready made docker image, it cannot be run before it is isolated on a container [14]. Containers can be run anywhere, be it on desktop, traditional IT, or the cloud by using docker's modern virtualization. Thus, this container system is a transformation in the development of software architecture from monolithic one server to microservice architecture which means it can be directly managed on smaller services or microservices [18]. By using docker, the application runs on each environment. Docker automates applications that will be deployed to containers. To run code efficiently, docker helps provide a fast and lightweight environment. Docker's four main sections: Docker Container, Docker Client-Server, Docker Images, and Docker Engine [2].

Enterprise Resource Planning (ERP) systems have significantly affected to organizational performance and business strategy. Realizations and plans in the post-implementation development of ERP systems can systematically increase job satisfaction and project success by preventing the expansion of the scope, schedule, and costs. Thus the failure of system practices must be prevented with the aim of not happening failures and chaos in ongoing projects [7]. ERP systems have a tremendous impact on Smart Factories. In the research that has been carried out [12], the use of the cloud can be adjusted according to needs starting from hardware, security needs, scalability and then storage can be adjusted according to the needs needed only for application functions to be implemented in the company. Cloud computing features include resource availability regardless of actual location, simple acquisition and maintenance, and a flexible pay as you go cost model. Then came Edge Computing which is a new paradigm that focuses on the process and storage of data that is close to the source of events rather than on remote servers thus introducing many positive implications [15].

From an end-user perspective, the biggest concern when using cloud storage is security because cloud infrastructure services provide storage virtually [5]. Content Service Providers (CSPs) can extend existing cloud-based Content Delivery Network (CDN) to edge environments to achieve cost effectiveness. DNS also plays an important role when a Content Delivery Network (CDN) is used to replicate content that is on the primary server to multiple servers located in geographically different areas. This affects the security side where the security element is moved closer to the source of the attack. Thus it can protect privacy with a flexible connection so that data transfer is more optimal [8].

DevOps is a combination of culture philosophy, set of practices, and tools that can improve an organization/company's ability to de-liver applications or software quickly. His philosophy is to remove any obstacles that occur in the application development process and apply appropriate sharing of responsibilities to each team. The practice here aims to achieve the speed and quality of the application development process, namely by streamlining procedures related to how the team works. By utilizing tools or software that are aligned with the application development process and automating repetitive tasks so that they are not done manually. Thus, it can make the release process more efficient and the application more reliable[16].

The following is previous research conducted regarding the discussion of docker containers, ERP implementation in the cloud and then DevOps implementation in companies. In conducting this research, it is necessary to conduct a literature review to be used as a reference material to strengthen research results by identifying methods that have been carried out, developing previous research that has a balanced correlation. Some literature reviews that have similar methods are obtained including the following.

2.1 Weilun, T., Li, W., & Guangtao, X. 2019

In a study conducted by Wei Lun et al in 2019 with the title " De-sign of Information System Architecture of Garment Enterprises Based on Microservices " The purpose of this study is to design an ERP system architecture based on microservices to solve ERP company resource planning system design problems that improve company management efficiency by implementing a system that supports scalability and good resilience.

2.2 Berggren, J., & Karlsson, J. 2022

In research conducted by Johannes Berggren and Jens Karlsson in 2022 on " Differences in Performance Between Containerization & Virtualization " With a focus on HTTP requests, it shows that hardware docker operation is lighter than kvm / hypervisor virtualization.

2.3 Perri, D., Simonetti, M., & Gervasi, O. 2022

The research that Perri et al in 2022 with the title " Deploying Efficiently Modern Applications on Cloud " The purpose of this study is to improve backend configuration with the use of docker containers and how to deploy web apps more efficiently. (Perri et al., 2022)

2.4 Birihanu, E., Mahmud, J., Kiss, P., Kamuzora, A., Skaf, W., Horvath, T., Jursonovics, T., Pogrzeba, P., & Lendak, I. 2022

Research conducted by Ermiyas Birihanu et al in 2022 under the title " Client Error Clustering Approaches in Content Delivery Networks (CDN) " Experiments were run on a set of proxy logs collected for 7 days from a single physical CDN server running several types of services (VoD, live TV, files). The dataset consists of 2.2 billion log lines. Analysis shows that a CDN is the backbone for high-quality Video on Demand (VoD) access and can improve overall service quality.

2.5 Putro, Z. P., & Supono, R. P. 2022

Research conducted by Zuki Pristiantoro and Riza Adrianti in 2022 with the title " Comparison Analysis of Apache and Nginx Web Server Load Balancing on Proxmox VE in Supporting Server Performance " It is known that how Apache and Nginx perform on hypervisor architecture.

2.6 Almeida, F., & Simoes, J. 2022

Then the research conducted by Fernando Almeida et al in 2022 with the title " Exploring the Benefits of Combining DevOps and Agile " DevOps became a solution to the agile method with the use of development, documentation, and testing tools automatically.

III. METHOD

The development method used in system development uses the Development & Operation (DevOps) approach. The step begins with planning, technically the step taken after the infrastructure is prepared is to prepare the code integrated with GitLab, then the build process is carried out and the code will automatically be corrected by GitLab tools at the testing stage, and automatically if there are no obstacles, the ERP will be immediately released and deployed, and can be directly operated and monitored and will take place continuously. This process so quickly is an advantage in the methods on this study. Broadly speaking as shown in Figure 3.1.

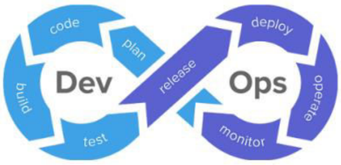
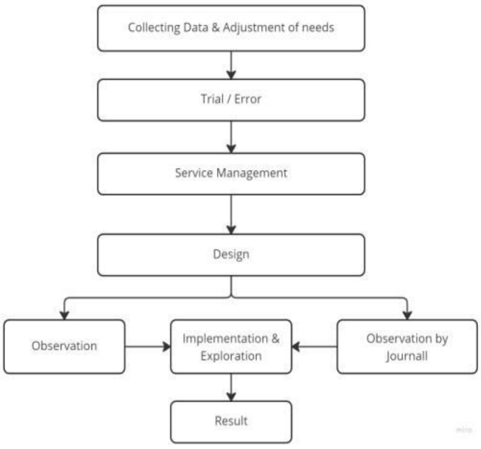


Figure 3.1 Research Methods

IV. RESULT OF RESEARCH AND DISCUSSION

4.1 Design Process

Research on the effectiveness in the application of cloud-based ERP requires a systematic problem solving step to get the right results. The troubleshooting steps are shown in Figure 4.1.



4.2 Design

The system that researchers are trying to implement is how to use docker containers to run Odoo ERP. It is known that ERP is such a complex system, we try to do the design of experiments as effectively as possible in terms of services. By adding a Content Delivery Network (CDN) as a reverse proxy domain, so there is no need to install additional services such as nginx, Apache to reverse domain on the Domain Name Server (DNS) on the server to increase the load on the server.

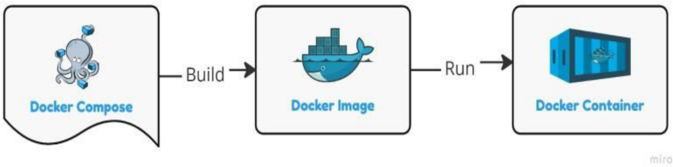


Figure 4.2 Application Build Process on Docker-Compose

Architecturally, researchers use Amazon Web Service (AWS) as a cloud edge computing provider. In this research method, we use a reference from Microsoft in system development using docker shown in Figure 4.3. In Figure 4.2 Docker Compose is a file that contains a list of Command Line Interface (CLI) instructions that Docker Engine will run to create an image or easier instructions for building an application into the form of a Docker Image. We're using an image on the docker hub registry that has been developed by Odoo, so we just need to call it via docker-compose and then run it together between the Odoo image and the PostgreSQL image. Here is a loop development that can be applied to the ERP system built in this study. The development loop shown in Figure 4.3 serves to facilitate the continuous development of the system in the future.

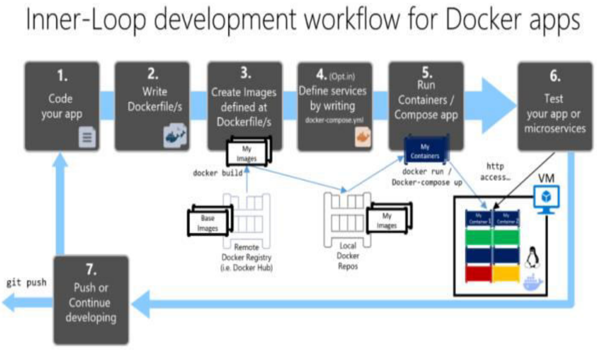

Inner-Loop development workflow for Docker apps
1. Code your app
2. Write Dockerfile/s
3. Create Images defined at Dockerfile/s
4. (Opt.in) Define services by writing docker-compose.yml
5. Run Containers / Compose app
6. Test your app or microservices
7. Push or Continue developing
git push
docker build
My Images
Base Images
Remote Docker Registry (i.e. Docker Hub)
Local Docker Repos
My Images
My Containers
docker run / Docker-compose up
http access…
VM
My Container 1
My Container 2

Figure 4.3 Docker Apps Loop Development Process

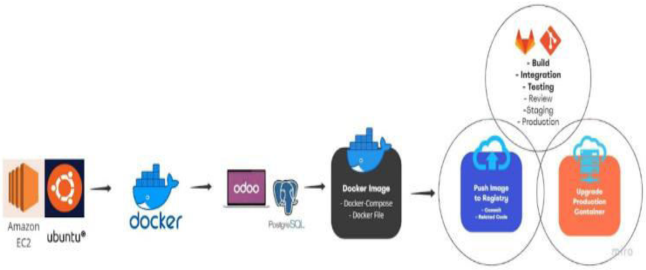


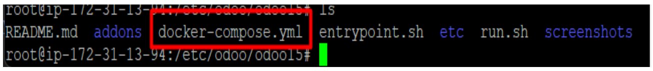
Figure 4.4 Infrastructure and System Design

In designing ERP system design, we use Odoo and PostgreSQL frameworks because it is one of the opensource ERP that has modules that can be developed according to company needs. The development method used in this study is to use Development & Operation (DevOps). Technically, integrating the code in computer development into the repository then the tools run tests automatically, quickly, and continuously. Figure 4.4 shows that the development process was tested and an error then resulted in the design in Figure 4.4 where the Continuous Integration process took place. The code is successfully integrated, so the application can be built and then released automatically. In this case we use GitLab as a tool for implementing this method. In Figure 4.4 the coding process will be documented and pushed to GitLab.

4.3 Implementation

In this study in measuring effectiveness empirically by using a usability approach from the practitioner's point of view in assessing the system applied at the time of use by users. The main thing on which it is based is how to run Odoo and PostgreSQL easily and in development more efficiently and identification easier in case of problems. Here are the steps to implement the system in this study.

In this case the docker-compose file containing the code to call the application image is prepared to run and then this docker-compose file is stored in the GitLab repository to find out the tracking of changes that occur during the application and repository development process can easily be run or called through any server. In conducting the experiment, the authors used Amazon Web Service (AWS) EC2 as a cloud service provider.

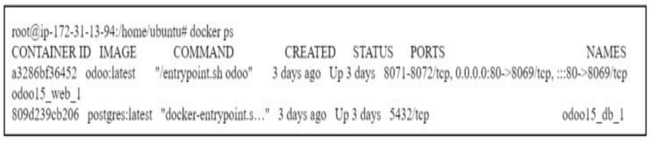


```
root@ip-172-31-13-94:/etc/odoo/odoo15# ls
README.md  addons  docker-compose.yml  entrypoint.sh  etc  run.sh  screenshots
root@ip-172-31-13-94:/etc/odoo/odoo15#
```

Figure 4.5 Docker-compose file

In the preparation process please be aware that the application needs to be set on the docker-compose file run on port 80 to replace the reverse proxy that usually uses Nginx, Apache or the like. In docker containers, it is commonly referred to as expose.

Using docker-compose in one execution can be done for multiple applications simultaneously, in which case the author tries to execute to run Odoo and PostgreSQL together through a different container as shown in Figure 4.5. In this run, the Odoo and PostgreSQL images are automatically downloaded and then run in a container. This will make it easier for developers to identify system issues after development with microservices services.

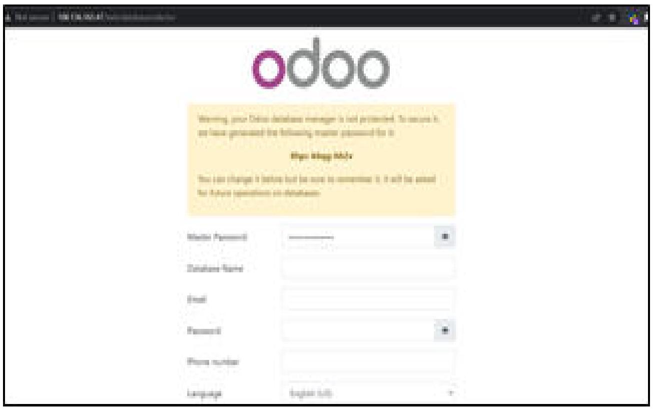


```
root@ip-172-31-13-94:/home/ubuntu# docker ps
CONTAINER ID  IMAGE          COMMAND                  CREATED     STATUS     PORTS                                                NAMES
a3286bf36452  odoo:latest    "/entrypoint.sh odoo"    3 days ago  Up 3 days  8071-8072/tcp, 0.0.0.0:80->8069/tcp, :::80->8069/tcp
odoo15_web_1
809d239cb206  postgres:latest  "docker-entrypoint.s…"  3 days ago  Up 3 days  5432/tcp                                         odoo15_db_1
```

Figure 4.6 Running Container

In Figure 4.6, the container is already running with the name odoo15_web_1 and odoo_db_1. In this study using Odoo version 15 or latest using the PostgreSQL database. When running multiple containers, it's easy to manage applications inside Docker Engine, the de facto industrial runtime.

In Figure 4.6 Odoo can be directly accessed via a public ip address without using port 8069. Because at the time of preparation docker-compose has been modified to the port so that it can immediately run on port 80.



odoo

[illegible]

Master Password ••••••••••

Database Name

Email

Password

Phone number

Language English (US)

Figure 4.7 Initial View of Odoo ERP In The Browser

4.4 Exploration

4.4.1 Utilization of Domain Name System (DNS) Management on Cloudflare CDN

By default, Odoo runs on port 8069 so this expose does need to be used so that there is no need to add an ip address:8069 when accessed through a browser. The expose keyword in the Docker file tells Docker that the container is notified to run the application on the specified port. This minimizes the use of services or services because the more services, the more problems will be caused. Thus the server adjustment again needs to be made. Of course, this is inefficient in implementing an Enterprise Resource Planning (ERP) system that has many services in it.

Once Odoo runs fine without using reverse proxy services like Nginx, Apache etc, the next experiment is to utilize Cloud-flare as a reverse proxy domain for public addresses on systems that have been built. Basically, a system needs a reverse proxy service for domain translation settings. Domain Name System (DNS) is used to access a website without having to remember the Internet Protocol Address (IP Address) of each site visited. The thing to note is that the server often goes down when overloaded this will affect the IP Address on the system which will change. With a domain, website visitors are easier to interact with the system because they only need to remember the addressing with a certain name. Another factor that allows for a change in the IP address of a website is the problem of location.

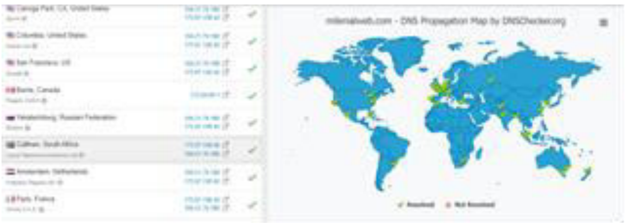
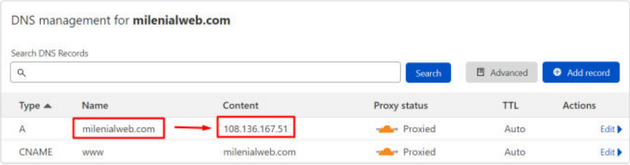


Figure 4.8 DNS Checker Propagation

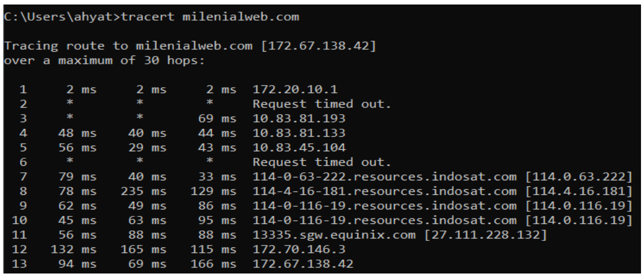
Visitors may get a different IP address depending on where the visitor is accessing the website as shown in Figure 4.8 regarding DNS propagation. However, with the Content Delivery Network (CDN), the data of all content on the server is moved to a CDN and then distributed to each service provider's edge server location. This allows visitors not to have to pass between distant countries to get to the real server, because the content is at the edge location of the server in the area closest to the visitor.

In Figure 4.8, that the A, AAAA, or CNAME record is proxied, the DNS query is forwarded to Cloudflare Anycast IP with the original DNS target. This process means that all requests destined for the proxy hostname will go to Cloudflare first and then be forwarded to the origin server. The request in question is the user in accessing the application address. Cloudflare can also be used to optimize, cache, and protect all requests that lead to the application, as well as protect the origin server from DDoS attacks. Because the request goes to the hostname that is proxied through Cloudflare before it reaches the origin server, it comes from a Cloudflare IP address (and is potentially blocked or restricted). This is evidenced by checking through the site to find out which servers were detected on the use of the system.


DNS management for **milenialweb.com**

Type ▲	Name	Content	Proxy status	TTL	Actions
A	milenialweb.com	108.136.167.51	Proxied	Auto	Edit ▶
CNAME	www	milenialweb.com	Proxied	Auto	Edit ▶

Figure 4.9 Cloudflare DNS Management


```
C:\Users\ahyat>tracert milenialweb.com

Tracing route to milenialweb.com [172.67.138.42]
over a maximum of 30 hops:

  1     2 ms     2 ms     2 ms  172.20.10.1
  2     *        *        *     Request timed out.
  3     *        *       69 ms  10.83.81.193
  4    48 ms    40 ms    44 ms  10.83.81.133
  5    56 ms    29 ms    43 ms  10.83.45.104
  6     *        *        *     Request timed out.
  7    79 ms    40 ms    33 ms  114-0-63-222.resources.indosat.com [114.0.63.222]
  8    78 ms   235 ms   129 ms  114-4-16-181.resources.indosat.com [114.4.16.181]
  9    62 ms    49 ms    86 ms  114-0-116-19.resources.indosat.com [114.0.116.19]
 10    45 ms    63 ms    95 ms  114-0-116-19.resources.indosat.com [114.0.116.19]
 11    56 ms    88 ms    88 ms  13335.sgw.equinix.com [27.111.228.132]
 12   132 ms   165 ms   115 ms  172.70.146.3
 13    94 ms    69 ms   166 ms  172.67.138.42
```

Figure 4.10 Process of IP Journey from Host to Edge Location Server

Then by using Cloudflare, the original public IP was hidden so that what was detected was IP 172.67.138.42. Cloud-flare's working principle uses Edge locations around the world to help speed up communication with users no matter where the user is located. Basically, Cloudflare stores cached copies on edge location servers with a location or close distance to customers so that content can be delivered faster.

4.4.2 Utilization of Gitlab as repository storage

In this study, all the code files used were stored in the GitLab repository. This makes it easier for developers to track all activities in code changes that have been developed. In addition, Odoo is an opensource framework that can be developed with the addition of modules according to company requests.

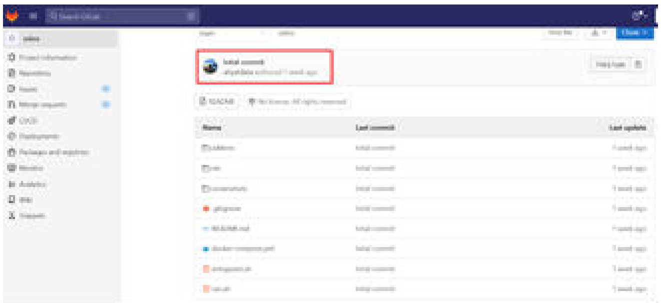
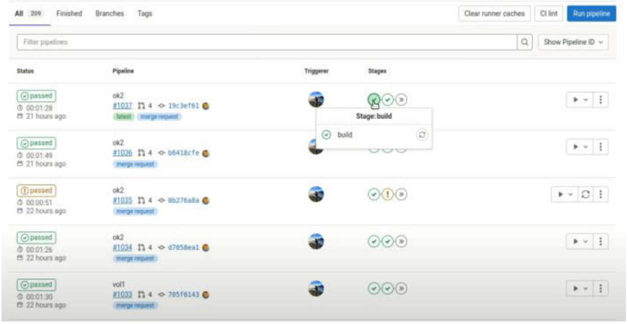


Figure 4.11 Gitlab Repository page

By using GitLab as shown in Figure 4.11, the development process will affect the development process and system operation by utilizing the CICD feature of the system that is developed to be deployed automatically in the event of addition and testing is carried out automatically.

4.4.3 Continuous Integration Continues Delivery (CICD) process on Gitlab

All processes in the activity in Continues Integration are documented as shown in Figure 4.12 from build to test.



Status	Pipeline	Triggerer	Stages
passed 00:01:28 21 hours ago	ok2 #1037 4 19c3ef61 latest merge request		
passed 00:01:49 21 hours ago	ok2 #1036 4 b6418cfe merge request		
passed 00:00:51 22 hours ago	ok2 #1035 4 0b276a8a merge request		
passed 00:01:26 22 hours ago	ok2 #1034 4 d7058ea1 merge request		
passed 00:01:30 22 hours ago	vol1 #1033 4 705f0143 merge request		

Figure 4.12 Gitlab deploy page

4.6 Performance Observation Results

After conducting various experiments and having been combined with other linear research results, here are the results obtained by being shown in Table 4.1, Table 4.2 and Table 4.3 below.

Table 4.1 Effectiveness of Docker Usage

Parameter	Hypervisor	Docker
Installation Time	Longer Should Be One By One	Faster because of a single execution
Integration	Per Hardware	Fellow Containers
Communication	Ethernet	Signals, Pipes, Sockets
Booting Time	Minutes	Seconds
Update Development	More complicated should provide shadow server backups	Simpler because the service can be moved to another container
Deploy Automatically	Not Supportive	Support
Data Process	Requires additional web server services	Only expose on ports
System Size	Bigger	Smaller

Table 4.2 Effectiveness of CDN Usage

Parameter	Without CDN	Using CDN
Installation Time	Longer	More Fast
DNS Propagation	Waiting 24 Hours	Minutes
User Connection Lenght	More Long	More Short
Latensi	Biger	Lower
Load Times	More Slow	More Fast
Security	Open IP Address	More Secure

Table 4.3 Effectiveness of Repository Usage

Parameter	Without Repository	Using Repository
Organization	Communication Betwen Management	Organized Communication using tools
History	Individual Docummentation	Well documented
Change	Changes are Hard to Rever	Can be undo
Merging	Must be replaced docummentt	Can be direct merge
Security	Can be lost	Neatly stored
Development	Must contact the developer	Can be directly in clonning
Deploy	Manual	Automatic deploy

This article emphasizes running odoo as efficiently as possible so it doesn't use nginx as a web server. Just imagine if you have to need additional time to set up nginx and domain. Indirectly it will reduce resources and speed on the server. Basically odoo is running on port 8069 so it requires nginx to redirect port 80. But with a special solution the author did without nginx, odoo can run properly. Implementation solution using Docker and CDN as the use of reverse proxies on Cloudflare are advantages and differences that stand out from other articles that still use traditional hosting services. From a security point of view, the author tells how Docker and CDN secure the services they host, for example, hiding the original IP address, the system does not directly come into contact with the network because it has to pass through a physical server first. Then the hacker will be distracted by the hidden IP address. Empirically the authors conducted technical testing to produce the best way of implementation. Then all experiences are summarized specifically in an information table that can be used as validation in implementing ERP using a docker container.

V. Conclusion

The use of docker is more efficient and lightweight with the hypervisor virtualization model, in the implementation of ERP systems. Docker can control odoo and postgresql with automatic or manual settings. The implementation of this docker helps in the further development of the ERP system and coupled with the implementation of GitLab all programs are stored and seen the history of development. CDN is very effective in helping the system publish to the internet and helping to translating domains faster. DevOps methods can improve development by means of automation that makes it different from other meth-ods.

Acknowledgements

Thank you to the lecturers and supervisors and family who always support and give the best for the author.

References

[1] Almeida, F., & Simoes, J. (2022). Exploring the Benefits of Combining DevOps and Agile. *Future Internet*, *14*. https://www.mdpi.com/1999-5903/14/2/63

[2] Amit, P. M., G, N. D., Kengond, S., & Mulla, M. M. (2020). Performance Evaluation of Docker Container and Virtual Machine. *Procedia Computer Science*, *171*, 1419-1428.

[3] Berggren, J., & Karlsson, J. (2022). Diferences in performance between containerization & virtualization. *Science in Software Engineering*. https://www.diva-portal.org

[4] Birihanu, E., Mahmud, J., Kiss, P., Kamuzora, A., Skaf, W., Horvath, T., Jursonovics, T., Pogrzeba, P., & Lendak, I. (2022). Client Error Clustering Approaches in Content Delivery Networks (CDN). *Networking and Internet Architecture*, *2210*(05314v1).

[5] Dave, J., & Vyas, D. (2018). Multi Level Data Security Model for Big Data on Public Cloud A New Model. *Int. J. Advanced Networking and Applications*, *09*(06), 3654-3657.

[6] Farrag, M. H., & Nasr, M. M. (2017). A Survey of Cloud Computing Approaches, Business Opportunities, Risk Analysis and Solving Approaches. *Int. J. Advanced Networking and Applications*, *09*(02), 3382-3386.

[7] Hietala, H., & Paivarinta, T. (2021). Benefits Realisation in Post-Implementation Development of ERP Systems: A Case Study. *Procedia Computer Science*, *181*, 419-426.

[8] Hours, H., Biersack, E., Loiseau, P., & Finamore, A. (2016). A study of the impact of DNS resolvers on CDN performance using a causal approach. *Computer Networks*, *109*, 200-210.

[9] Kirpitsas, I. K., & Pachidis, T. P. (2022). Evolution towards Hybrid Software Development Methods and Information Systems Audit Challenges. *MDPI Software 2022*, *1*, 316-363.

[10] Pahl, C., & Lee, B. (2015). Containers and Clusters for Edge Cloud Architectures a Technology Review. *2015 3rd International Conference on Future Internet of Things and Cloud*. 10.1109/FiCloud.2015.35

[11] Perri, D., Simonetti, M., & Gervasi, O. (2022). Deploying Efficiently Modern Applications on Cloud. *MDPI electronics 2022*, *11*, 450. https://doi.org/10.3390/electronics11030450

[12] Prakash, V., Savaglio, C., Garg, L., Bawa, S., & Spezzano, G. (2022). Cloud- and Edge-based ERP systems for Industrial Internet of Things and Smart Factory. *3rd International Conference on Industry 4.0 and Smart Manufacturing*, *200*, 537-545.

[13] Putro, Z. P., & Supono, R. P. (2022). Comparison Analysis of Apache and Nginx Webserver Load Balancing on Proxmox VE in Supporting Server Performance. *International Research Journal of Advanced Engineering and Science*, *7*(3), 144-151.

[14] Simonsson, J., Zhang, L., Morin, B., Baudry, B., & Monperrus, M. (2021). Observability and chaos engineering on system calls for containerized applications in Docker. *Future Generation Computer Systems*, *122*, 117-129.

[15] Sulieman, N. A., Celsi, L. R., Li, W., Zomaya, A., & Villari, M. (2021). Edge Oriented Computing : A Survey on Research and Use Cases. *MDPI Enegies 2022*, *15*, 452. https://doi.org/10.3390/en15020452

[16] Taryana, A., Fadli, A., Murdyantoro, E., & Nurshiami, S. R. (2020). DevOps Approach Embraces Forward and Reverse Engineering. *International Journal of Applied Information Technology*, *04*(02).

[17] Wan, X., Guan, X., Wan, T., Bai, G., & Choi, B. Y. (2018). Application deployment using Microservice and Docker containers: Framework and optimization. *Journal of Network and Computer Applications*, *119*, 97-109.

[18] Waseem, M., Liang, P., Shahin, M., Salle, A. D., & Marquez, G. (2021). Identifying architectural technical debt, principal, and interest in microservices A multiple-case study. *The Journal of Systems & Software*, *111061*(182).

[19] Weilun, T., Li, W., & Guangtao, X. (2019). Design of Information System Architecture of Garment Enterprises Based on Microservices. *Journal of Physics: Conference Series*, *1168*, Number 1 p032128-032128.

BIOGRAPHIES AND PHOTOGRAPHS

Ahmad Wijayanto S.Tr.T Currently working Developer at the Company, a network infrastructure practitioner and as a Student at the Master of Information Systems at Diponegoro University.

Prof. Dr. Ir. Eko Sediyono, M.Kom Currently he is a Professor at the Satya Wacana Christian University, several related studies including Algorithms, Digital Signature, Cryptography, Information Security, Semantic Web, & Ontology.

Dr. Aris Puji Widodo, S.Si., M.T Currently a Practitioner and Doctorate at the Faculty of Science and Mathematics at Diponegoro University, several related studies include Networking, Information Systems, IT Management & Health Care.