International Journal Of Information System & Technology
Vol. x, No. x, (20xx), pp. xx-xx

Cloud Computing Implementation with Docker Engine Swarm Mode for Data Availability Infrastructure of Rice Plants

Oktalia Juwita[1], Diksy Media Firmansyah [2]
[1] *Information System Department, University of Jember*
[2] *Computer Science Department, University of Jember*
[1]*oktalia.juwita@gmail.com*, [2]*diksy@unej.ac.id*

Abstract

Data of rice plant is important in Indonesia because rice is the staple food for Indonesian people. Rice plant data can be formatted into web service, so anyone can access the information from anywhere by using internet. But, high numbers of request are the problem for web server apps. One of the solution is by distributing request into some server. In this paper, we will compare failed request and time per request in conventional server and clustered server with docker swarm. Server apps in clustered server shows lower value of failed request than conventional server in our experiment. With two containers, number of failed request obtain 0.78% lower than conventional server in 25.000 requests, and 0.69% lower than conventional server in 50.000 requests.

Keywords: *Cluster Server, Data Availability, Docker Swarm*

1. Introduction

Rice plant data is one important data in Indonesia because rice is primary food for most Indonesian people. Even more, data are presented into digital format, like a website [1]. With digital data availability of rice plant [2], Indonesian people can know how rice plant productivity in Indonesia anywhere and anytime. This data also can used for decision support of crops in Indonesia, like export import possibility, agricultural land that must be increase or decrease, etc.

Cloud computing is a one solution approach of digital data availability [3]. This approach appeared because important digital data have much users but computer resource is limitless usually. With cloud computing, many computer resources can be combined which produce better performance. Cloud computing also can split big resource into many smaller resources [4]. Docker [5] [6] [7] and Proxmox [8] is the example of many cloud computing system.

In this paper, we developed infrastructure model for cloud computing with docker swarm mode. Docker itself is one of cloud computing vendor based on container approach, that called docker engine. Docker engine is widely used because easy use to connecting and scaling containers that deploy with swarm mode. Then, we apply rice plant digital data to MySQL database system [9] and test performance with ApacheBench [10].

The rest of this paper is structured as follows. Section 2 describes the research relates to the proposed method, that is Cloud Computing, Virtual Machine and Docker. Section 3 presents the proposed method whose result is provided in section 4. Finally, conclusion is drawn in section 5.

ISSN: XXXX-XXXX IJXX

2. Related Works

2.1. *Cloud Computing*

Cloud computing is a model that allows a system to access computer resources with specific configuration [3]. Cloud computing must be providing and releasing the resource quickly and simply. There are five characteristics of cloud computing, that is:

- On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
- Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).
- Resource pooling. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.
- Rapid elasticity. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
- Measured service. Cloud systems automatically control and optimize resource use by leveraging a metering capability1 at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

2.2. *Virtual Machine*

One of cloud computing approach is virtual machine or can be called as a hypervisor [4]. Virtual machine is fully component of operating system (called Guest OS) that installed inside another operating system (called Host OS). The use of virtual machine allows some people to have their own operating system in a machine without disturbing the operating system of other users.

One of operating system that support virtual machine is Proxmox Virtual Environment / VE [8]. Virtual machines are deployed inside Proxmox VE can be arranged freely. Moreover, Proxmox VE management interface is website based. So, provider and customer can be customizing the virtual machine easily.

2.3. *Docker*

Other approach for cloud computing is container [4]. Like virtual machine, container approach has ability to encapsulate service into standalone operating system. But, operating system inside container do not have kernel. So, container is lighter than virtual machine. Kernel that used in container is Host OS kernel.

Docker is one of cloud computing vendor for container based [5]. Application that docker used to deploy container called docker engine. The advantages of docker engine is swarm mode that can be used for scaling the services easily [6]. Swarm mode inside docker engine also have internal load balancer. So, user no need to install load balancer infrastructure independently [7].

3. Method

In this paper, we proposed four models to measure performance of apache2 web server [1]. First and second models are using single virtual machine. Third and fourth models are using double virtual machine with half resource if compared with first and second models.

3.1. *Model 1*

In model 1, we used single virtual machine and install apache2 web server directly. The resources are 2 core CPU and 2 GB memory. This model simulated web server that not used docker swarm. Illustration of this model can be viewed in Figure 1.

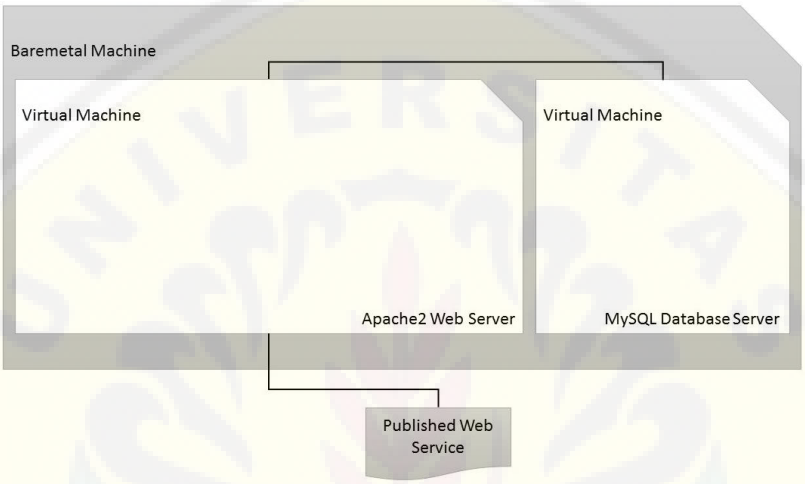

Figure 1. Single VM and install Apache2 directly

3.2. *Model 2*

Like model 1, we used single virtual machine in the model 2. But, docker swarm installed in the virtual machine and apache2 web server running inside docker container. The resources also 2 core CPU and 2 GB memory. This model simulated single web server that used docker swarm. Illustration of this model can be viewed in Figure 2.

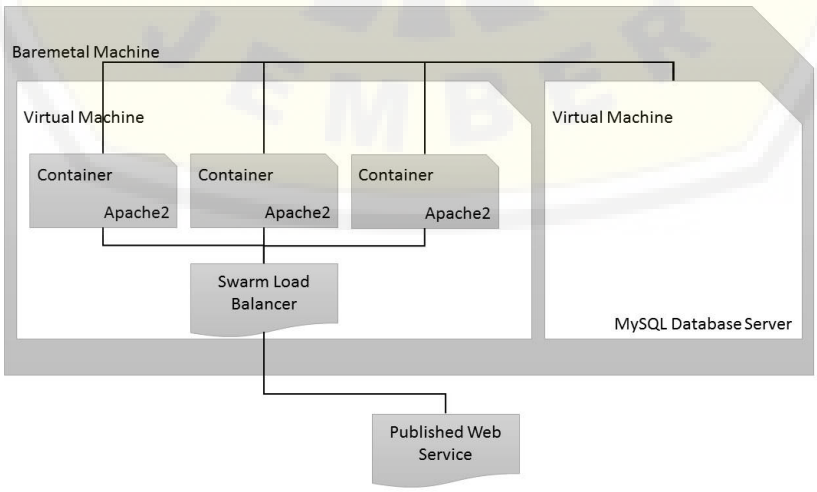

Figure 2. Single VM and use docker swarm.

3.3. *Model 3*

In model 3, we used double virtual machine and install docker swarm inside each virtual machine. The resources are 1 core CPU and 1 GB memory. This model simulated single baremetal server that parse into some virtual machine and used docker swarm for cluster system. Illustration of this model can be viewed in Figure 3.

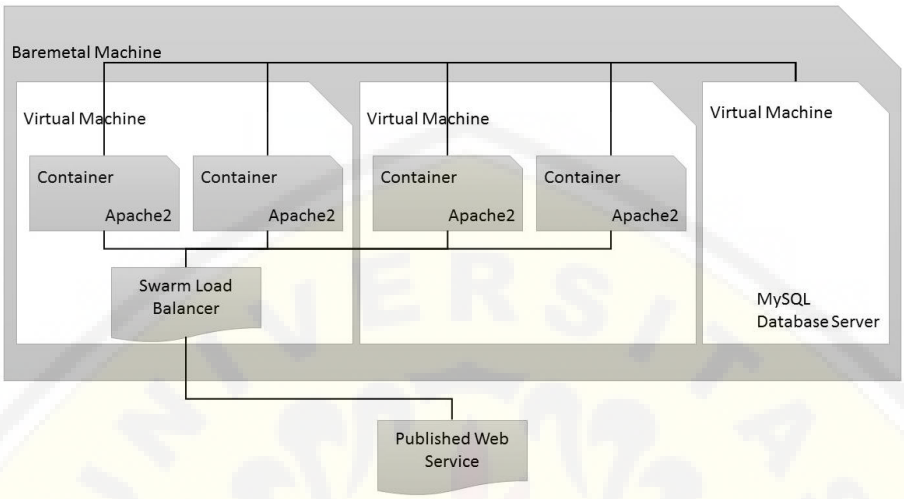

Figure 3. Use 2 VM inside single baremetal machine

3.4. *Model 4*

Like model 3, we used double virtual machine in the model 4. But, first and second virtual machine are divided by baremetal machine. The resources also 1 core CPU and 1 GB memory. This model simulated cluster of baremetal server that used docker swarm for cluster system. Illustration of this model can be viewed in Figure 4.

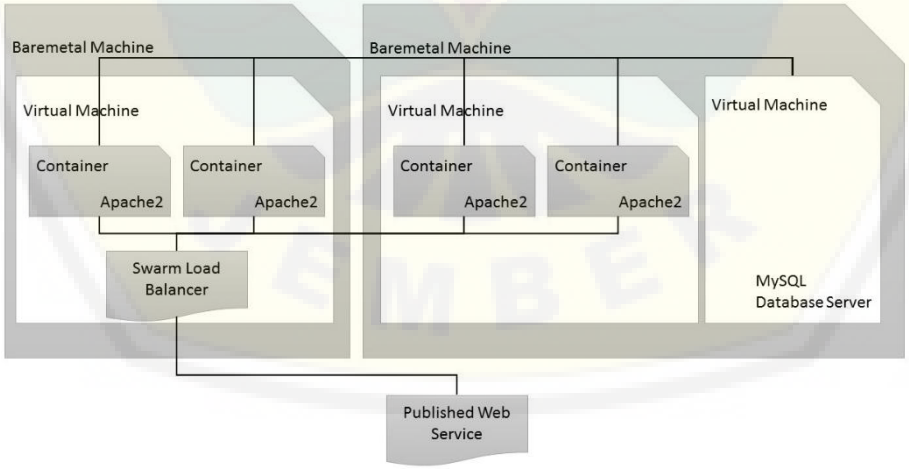

Figure 4. Use 2 VM and split into 2 baremetal machine

4. Result and Discussion

Rice plant data with 693 row data are used in this experiment to benchmarking performance of proposed models [2]. Original format of rice plant data that used is xls (Microsoft Excel). So, we transform rice plant data to MySQL [9] format for easy use in programming language. It also simulated general condition that website / web service use SQL format for database system.

ApacheBench is tool that used in this paper to benchmarking performance [10]. ApacheBench can access the web service with many requests simultaneously and display the quantity of failed request. Failed request is number of request that cannot fully response by targeted server. Lower of failed request value represent that web server is better. That is because web server can handle more request in the same time. Experimental result for 25.000 and 50.000 request simulation can be viewed in Table 1.

Table 1. Number of Failed Request.

Number of Container ➔	**1**	**2**	**4**	**8**
25,000 requests				
Architecture 1	894			
Architecture 2	1108	**195**	1887	4981
Architecture 3	1389	658	9316	16121
Architecture 4	2303	609	7019	8981
50,000 requests				
Architecture 1	1923			
Architecture 2	804	**347**	14147	22478
Architecture 3	1821	879	3426	40220
Architecture 4	1855	609	9593	33992

From Table 1, we found that single virtual machine with docker swarm (architecture 2) have lowest failed request. But, this condition is only for 2 number of container. For 25,000 simultaneously request, architecture 2 only obtain 195 failed request (0.78% from all request). For 50,000 simultaneously request, architecture 2 only obtain 347 failed request (0.69% from all request). If number of container are increased, the result is worse.

5. Conclusion

In this paper, we proposed four infrastructure models to compare the web server performance. The first model represent server without docker swarm and the other model represent server with docker swarm. Then, we test performance of each server model with 25,000 and 50,000 simultaneously request using ApacheBench.

Models with docker swarm are proven develop lower failed request than models without docker swarm (install web server directly). The best result seen on architecture 2 with 2 docker container. But, lower failed request than architecture 1 seen on all docker swarm architecture with 2 docker container. So, web server apps with docker swarm is better than install web server directly, but only in some number of container.

6. Acknowledgement

In the future, we will expand the proposed model for benchmarking. By developing several models again, we hope can be found relation of optimum container quantity and

hardware specification. We also want to try another container vendor and compare it with docker swarm.

References

[1] T. A. Software Foundation (2017). Apache2 HTTP Server Project [Online]. Available: https://httpd.apache.org/.

[2] U. K. P. B. P. P. P. UKP-PPP (2013), Tanaman Padi per Provinsi [Online]. Available: https://data.go.id/dataset/tanaman-padi-per-provinsi. [Accessed 15 6 2017].

[3] P. Mell and T. Grance (2011). The NIST Definition of Cloud Computing. National Institute of Standards and Technology (NIST). Gaithersburg,.

[4] D. Bernstein (2014). Containers and Cloud: From LXC to Docker to Kubernetes. *IEEE Cloud Computing*, pp. 81-84.

[5] D. Inc. (2017). Docker Reference Architecture: Designing Scalable, Portable Docker Container Networks [Online]. Available: https://success.docker.com/Architecture/Docker_Reference_Architecture%3A_Designing_Scalable%2C_Portable_Docker_Container_Networks.

[6] D. Inc (2017). Docker Reference Architecture: Universal Control Plane 2.0 Service Discovery and Load Balancing [Online]. Available: https://success.docker.com/Architecture/Docker_Reference_Architecture%3A_Universal_Control_Plane_2.0_Service_Discovery_and_Load_Balancing.

[7] D. Inc (2017). Use swarm mode routing mesh [Online]. Available: https://docs.docker.com/engine/swarm/ingress/.

[8] P. S. S. GmbH (2017). Proxmox Downloads [Online]. Available: https://www.proxmox.com/en/downloads.

[9] O. Corporation (2017). MySQL Downloads [Online]. Available: https://www.mysql.com/downloads/.

[10] T. A. Software Foundation (2017). ab - Apache HTTP server benchmarking tool [Online]. Available: https://httpd.apache.org/docs/2.4/programs/ab.html.

Authors

Oktalia Juwita
Information System Department, University of Jember
oktalia.juwita@gmail.com

Diksy Media Firmansyah
Computer Science Department, University of Jember
diksy@unej.ac.id