CoMBInES
CONFERENCE ON MANAGEMENT, BUSINESS, INNOVATION, EDUCATION AND SOCIAL SCIENCES

Volume 1 No 1 (2021)

Diterima : February 01, 2021
Disetujui : February 05, 2021
Diterbitkan: February 24, 2021

Conference on Management, Business, Innovation, Education and Social Science
https://journal.uib.ac.id/index.php/combines

Analisis Load Balancing Menggunakan Docker Swarm

Stefanus Eko Prasetyo[1], Agung Wijaya[2]
Email stefanus@uib.ac.id, 1731065.agung@uib.edu

[1]Ilmu Komputer, Universitas Internasional Batam, Batam, Indonesia
[2]Ilmu Komputer, Universitas Internasional Batam, Batam, Indonesia

Abstrak

Load Balancing merupakan sebuah metode yang digunakan dalam memaksimalkan troughput, traffic dapat dijalankan dengan optimal, diperkecilkan latency dan menghindari overload pada overload pada salah satu jalur koneksi. Dalam uji coba ini menggunakan container Docker Swarm yang menjadi wadah virtualisasi load balancing. Variabel yang digunakan dalam acuan uji coba load balancing ini adalah response time. Hasil setelah pengujian menunjukkan bahwa load balancing yang terdapat pada virtualisasi container docker swarm dapat berjalan dengan baik tanpa gangguan.

Kata Kunci:
Load Balancing, Virtualisasi, Docker Swarm

Abstract
Load Balancing is a method used to maximize throughput, traffic can be run optimally, reduce latency and avoid overloading on one of the connection paths. This trial, using the Docker Swarm container which is a load balancing virtualization container. The variable used in this load balancing trial reference is the response time. The results after testing show that the load balancing contained in the docker swarm container virtualization can run well without interruption.

Keywords:
Load Balancing, Virtualisasi, Docker Swarm.

Pendahuluan

Jaringan dapat dikatakan sebagai sebuah hubungan maupun komunikasi yang dilakukan oleh beberapa perangkat yang terhubung pada sebuah titik tertentu. Beberapa jaringan juga dapat dihubungkan maupun disatukan dengan menggunakan sebuah teknik yang disebut virtualisasi.

Virtualisasi merupakan sebuah teknik yang dimana dapat menciptakan sebuah wadah virtual dari sebuah objek yang memiliki wujud fisik seperti wadah penyimpanan data, system

operasi dan juga server. Salah satu teknik virtualisasi yaitu virtualisasi berbasis container (Hakim, Riyanto, and Fauzan 2020).

Virtualisasi berbasis container merupakan sebuah teknik dimana virtualiasi tersebut tidak memerlukan hypervisor untuk berfungsi, container ini dapat dijalankan langsung pada sistem operasi Pada VM Ware mengharuskan pengguna untuk mengatur resource ketika instalasi, sedangkan pada container tidak diharuskan karena container memiliki resource tersendiri yang disediakan oleh host. Jika kekurangan resource maka container akan mengambil resource yang terdapat pada hardware sesuai yang dibutuhkan. Salah satu virtualisasi berbasis container yaitu docker.

Docker merupakan sebuah tempat atau wadah yang menggunakan sistem berbasis container, yang digunakan dalam mengembangkan web server guna mempermudah fase deploy pada software maupun aplikasi web(Rexa, Data, and Yahya 2019). Jika hanya menggunakan satu hostingan server maka akan terjadi single point of failure, dimana server gagal merespon permintaan dari pengguna mengakibatkan sistem tidak dapat berfungsi dengan normal dikarenakan overload permintaan dari pengguna.

Jika hal tersebut dapat diatasi dengan menggunakan teknik web cluster, web cluster memiliki nama lain yaitu server farm, web cluster merupakan sebuah gabungan dari beberapa computer server yang digunakan oleh sebuah lembaga maupun sebuah organisasi untuk mencapai kebutuhan yang diperlukan oleh server untuk melampaui kemampuan sebuah mesin(Bella, Data, and Yahya 2018).

Jika ada salah satu host yang mati maka host yang lain yang tidak mengalami salah host akan menangani request dari pengguna sehingga mendapat fungsi yang tersedia. Bagaimanapun untuk mengatur banyak host sangatlah kompleks, tetapi Docker memiliki modul untuk menangani masalah tersebut yang bernama Docker Swarm.

Docker swarm juga memiliki dua node yang bekerja disebut node Manager dan node Worker. Node manager merupakan node yang mengatur dan memberikan sebuah tugas pada node Worker. Sedangkan node Worker adalah node yang bekerja sesuai dengan tugas yang diberikana dari node manager.

Docker swarm merupakan sebuah docker yang berbasis container yang dimana cara kerja menggunakan routing mesh yang memberikan kemungkinan untuk setiap node dapat mengakses semua port yang disebarkan dari layanan yang sedang beroperasi dalam swarm dan ketika tidak terdapat task yang bekerja dinode routing mesh akan meneruskan semua permintaan dari pengguna ke container yang sedang beroperasi ke dalam node(Cérin et al. 2018).

Dalam suatu web cluster harus memiliki satu atau lebih entitas interface load balancing. Load balancing berfungsi memaksimalkan troughput, traffic dapat dijalankan dengan optimal, diperkecilkan latency dan menghindari overload pada overload pada salah satu jalur koneksi. Load balancer dibagi menjadi 2 yaitu load balancer software dan hardware. Terdapat banyak software load balancing dianataranya yaitu HAproxy, Nginx, dan Zevenet(Munadi and Sanjoyo 2017).

Berdasarkan uraian yang terdapat di atas mengenai load balancing menggunakan docker swarm guna menganalisis pengunaan pada dua jaringan yang berbeda, dengan ini penulis menggunakan docker swarm untuk mencoba melakukan load balancing yang berjudul "Analisis load balancing menggunakan docker swarm".

Tinjauan Pustaka

Adapun penelitaian ini berdasarkan hasil tinjauan dari beberapa penelitian yang telah dipelajari sebelumnya.

Penelitian yang berjudul "Implementasi High Availability Cluster Web Server Menggunakan Virtualisasi Container Docker"(Putra, Fitri, and Iskandar 2020). Penelitian ini dilakukan guna untuk menangani permasalahn request yang berlebihan pada web server (overload).

Penelitian yang berjudul "Implementasi Virtualisasi Server Berbasis Docker Container" (Dwiyatno, Rakhmat, and Gustiawan 2020). Penelitian ini dilakukan guna untuk Mengimplementasikan sistem virtualisasi server berbasis docker container.

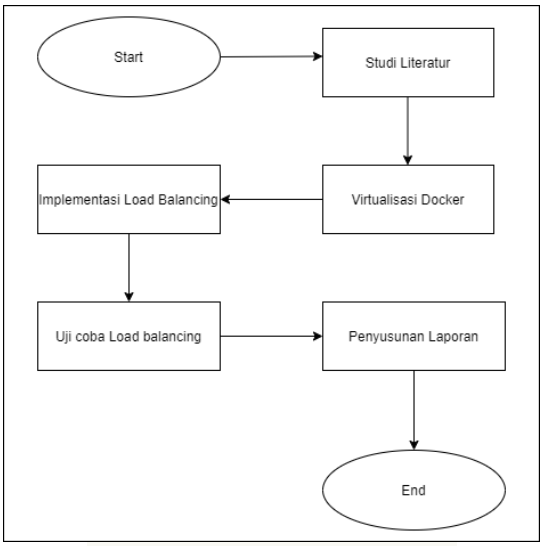
Penelitian yang berjudul "Load Balancing Server Web Berdasarkan Jumlah Koneksi Klien Pada Docker Swarm" (Afis, Data, and Yahya 2019). Penelitian ini dilakukan guna untuk meningkatkan kinerja server web hanya dengan menggunakan single backend server web.

Penelitian yang berjudul "Web Server Load Balancing Pada Arsitektur Docker Swarm"(Farid, Idhom, and Wahanani 2020). Penelitian ini dilakukan guna untuk meningkatkan kinerja server web hanya dengan menggunakan single backend server web.

Penelitian yang berjudul "Design And Implementation Of Lightweight Virtualization Using Docker Container In Distributing Web Application With Experimental Methods" (Prasetyo et al. 2021). Penelitian ini dilakukan guna untuk mengukur pengaruh suatu perlakuan tertentu terhadap sebuah variable dengan perlakuan yang berbeda.

Metodologi Penelitian

Penelitian yang dilakukan oleh penulis terdiri dari berbagai langkah yang akan dijadikan sebagai sebuah kerangka sistematis berupa sebuah alur penelitian. Alur penelitian ini akan digunakan untuk menjelaskan tahap-tahap yang dilakukan oleh penulis, penelitian mencari perumusan masalah yang terdapat dari awal sampai akhir penelitian yaitu penyusunan laporan penelitian. Alur penelitian yang dirancang oleh penulis dapat dilihat pada Gambar 3.1 berikut ini.



Gambar 1 Alur Penelitian

Alur penelitian pada Gambar 1 diatas dapat dipaparkan sebagai berikut :

Perumusan Masalah

Pada tahap ini penulis mempelajari permasalahan yang belum pernah alami dalam bagian tertentu. Permasalahan yang ditemukan akan langsung mencari tau apa aja solusi yang dapat diselesaikan pada masalah tersebut dan memperoleh berbagai manfaat yang terlibat dalam menyelesaikan dari penelitian tersebut.

Studi Literatur

Pada tahap ini penulis mengembangkan ilmu lebih lanjut mengenai permasalahan yang ada dan mencari cara untuk menselesaikan masalah tersebut dengan mempelajari tutorial dan mencari jurnal penelitian yang sebelumnya telah dilakukan oleh peneliti lainnya mengenai topik masalah yang sedang dibahas oleh peneliti, sehingga bisa membantu penulis untuk menyelesaikan penelitian ini.

Virtualisasi Docker

Pada tahap ini penulis melakukan virtualisasi berbasis container pada sistem yang telah disiapkan sebelumnya

Implementasi Load Balancing

Pada tahap ini melakukan konfigurasi load balancing pada virtualisasi pada container swarm.

Uji Coba Load balancing

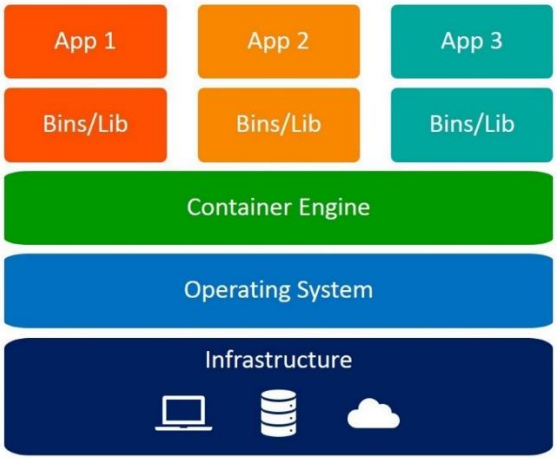
Pada tahap ini penulis melakukan uji coba pada docker yang sudah di konfigurasi sebelumnya dan melihat hasil akhirnya.

Penyusupan Laporan

Pada tahap ini penulis mendokumentasikan akhiran yang didapat kan dari penelitian yang telah dilakukan oleh penulis ke dalam sebuah laporan yang akan dijadikan sebagai bagian referesnsi bagi penulis lainnya kedepannya.

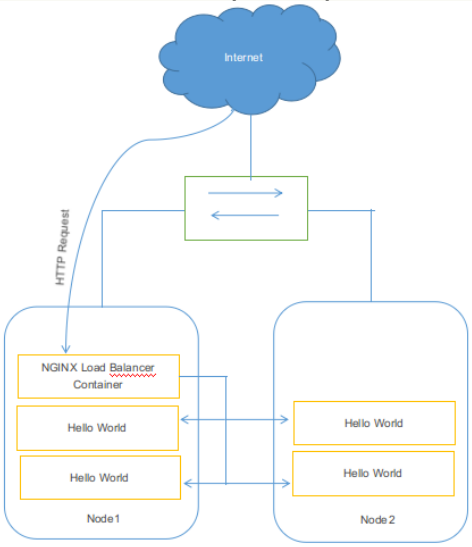
Analisa Permasalahan

pada sekarang ini banyak yang menggunakan virtual machine untuk melakukan testing dijadikan sebagai servertertapi dikarenakan virtual machine menggunakan hypervisor maka menggunakan banyak recource sehingga membuat penggunaan sumber daya menjadi kurang efesien. setelah hadirnya docker, virtualisasi yang berbasis container yang dimana tersebut menjadikan fitur tersebut dapat beroperasi dengan senidri dan tidak memakan resource terlalu banyak. Gambaran arsitektur virtualisasi berbasis container dapat dilihat pada gambar 3.2 berikut.



Gambar 2 Arsitektur Virtualisasi Berbasis Container

Pada umumnya perusahaan menggunakan jaringan dimana memiliki kejadiaan jaringan tidak stabil dikarenakan koneksi ISP (Internet Service Provider) sedang mengalami gangguan atau layanan internet sedang bermasalah sehingga membuat pekerjaan perusahaan menjadi tidak efesien maka dari itu digunakkannya load balancing agar dapat menjaga kestabilan jaringan pada perusahaan, ketika sebuah jaringan down / mengalami gangguan dapat di alihkan langsung oleh provider lainnya sehingga tidak menggangu keberlangsungan pekerjaan di perusahaan tersebut , dibawah ini akan ditampilkan pada Gambar 3.3 berikut:



Gambar 3 Visualisasi Docker

Berikut ini penulis akan menjelaskan kebutuhan hardware & software yang penulis butuhkan dalam perancangan sistem penelitian ini. Kebutuhan hardware dalam penelitian yang berjudul "Analisis load balancing menggunakan docker swarm" adalah sebagai berikut :

1. Sebuah cloud server untuk tempat mengimplementasikan docker dan virtual box.
2. Sebuah laptop untuk melakukan analisa kinerja pada docker & container.

Kebutuhan Software dalam penelitian yang berjudul "Analisis load balancing menggunakan docker swarm" adalah sebagai berikut:

1. Docker sebagai virtualisasi berbasis container.
2. Virtuali Box sebagai virtualisasi berbasis hypervisor.

Hasil dan Pembahasan

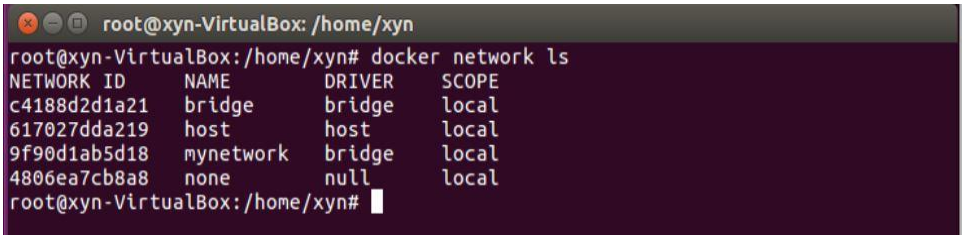
Dalam melakukan implementasi ini, ada beberapa software yang perlu di install agar implementasi ini bejalan dengan lancar. Download software yang diperlukan, lalu install software tersebut. Software yang perlu di download dan di install yaitu:

1. VirtualBox v6.1.16

VirtualBox merupakan sebuah software untuk melakukan virtualisasi sistem operasi dimana biasanya digunakan untuk mencoba mengisntall server / windows tanpa fisik dan dapat menghemat daya.

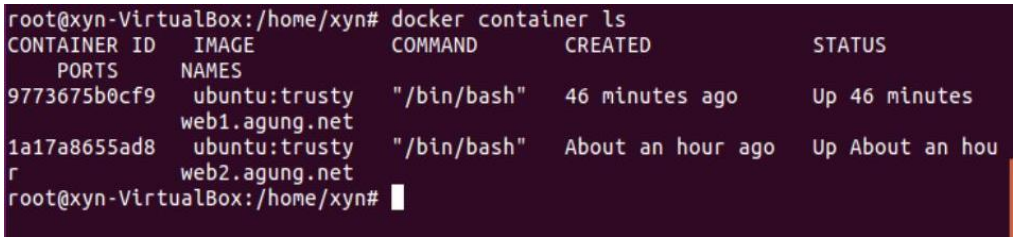
2. Linux ubuntu desktop v16.04

Linux merupakan sebuah operasi sistem unix yang open source baik GUI / CLI dapat melakukan update secara langsung tanpa harus menggunakan third party.



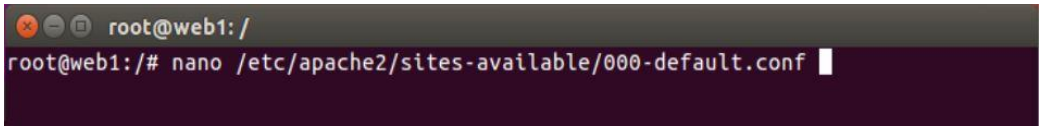
Gambar 4

Pertama buat network pada docker dengan menggunakan command docker network create --subnet=172.20.0.0/16 mynetwork dan untuk melihat network yang kita buat pada docker dapat menggunakan command docker network ls.



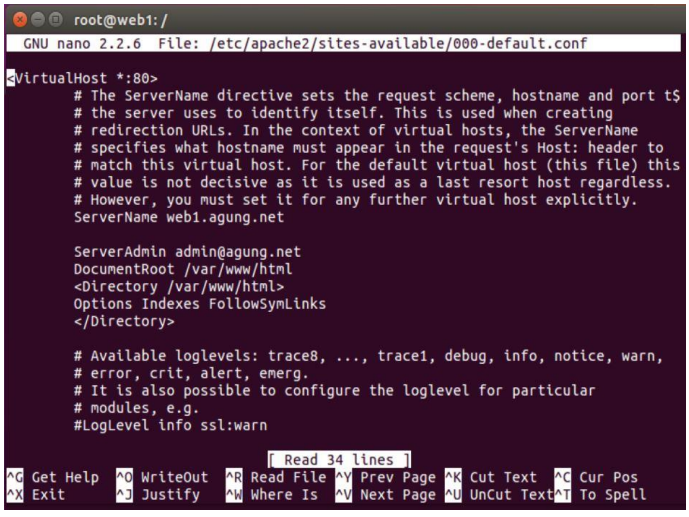
Gambar 5

Pada tahap ke dua untuk membuat container 1 didalam network yang sudah dibuat sebelumnya dapat menggunakan command docker run --net mynetwork --ip 172.20.0.9 --hostname web1.agung.net --name web1.agung.net -i -t ubuntu:trusty untuk melihat container yang sudah kita buat pada docker dapat menggunakan command docker container ls,



Gambar 6

Kemudian dengan menggunakan command nano untuk mengedit file pada /etc/apache2/sites-available/000-default.conf, Setelah kita membuat container pada network maka otomatis akan masuk ke dalam container tersebut.



Gambar 7

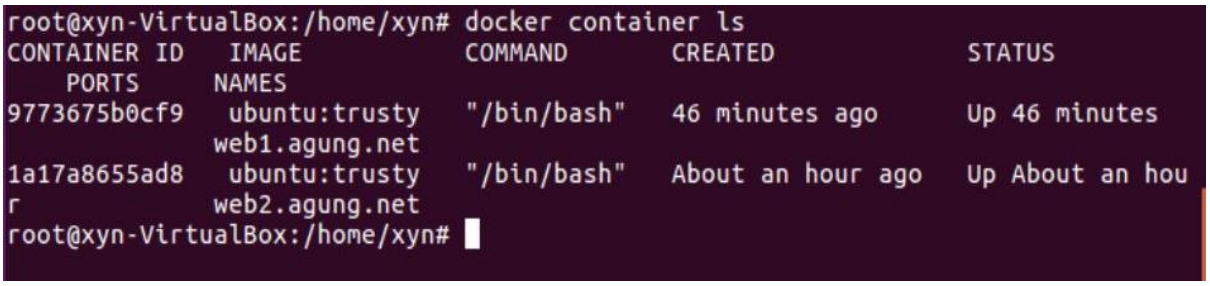
Lalu ubah server name menjadi value web1.agung.net, kemudian mengubah serverAdmin admin@agung.net setelah tambahkan masukan command <Directory /var/www/html> lalu enter setelah itu tambahkan lagi command Options Indexes FollowSymlinks lalu enter dan tambahkan command </Directory> setelah itu exit dengan command Ctrl + X

Gambar 8

Selanjutnya untuk melakukan restart web server dan cleaning apache2 dengan menggunakan command service apache2 restart.

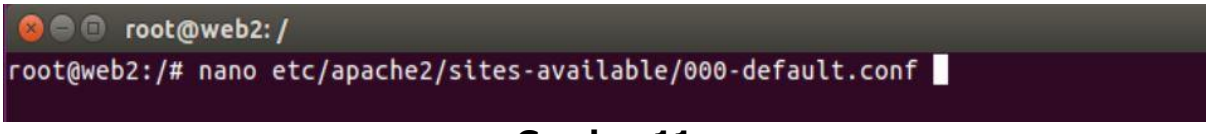
Gambar 9

Ketika berhasil maka memunculkan hasil seperti ini pada ip 172.20.0.9.



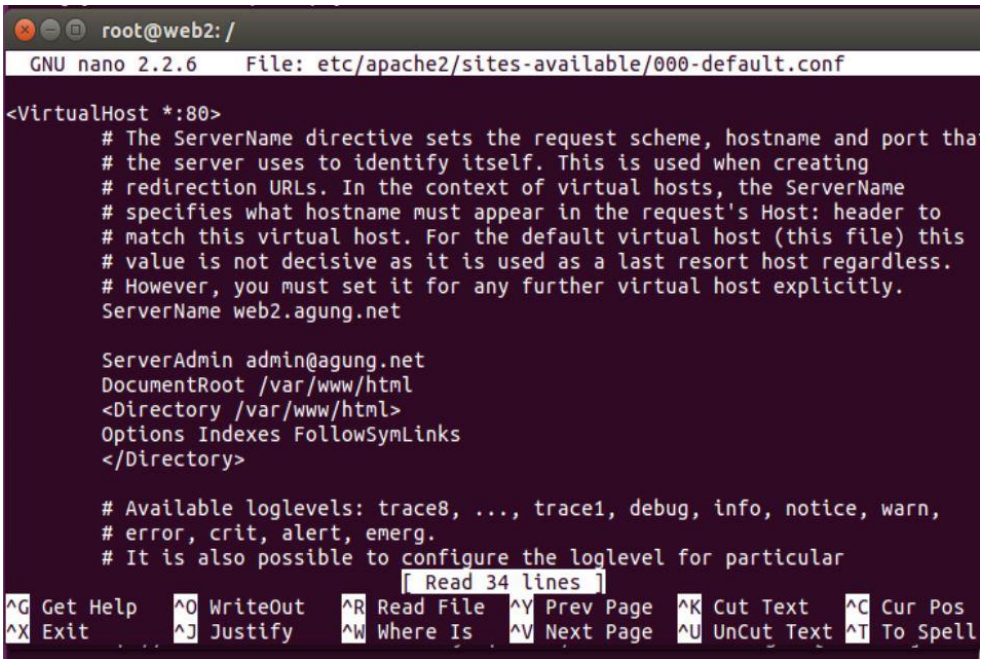
Gambar 10

Selanjutnya untuk membuat container 1 didalam network yang sudah dibuat sebelumnya dapat menggunakan command docker run --net mynetwork --ip 172.20.0.10 --hostname web2.agung.net --name web2.agung.net -i -t ubuntu:trusty untuk melihat container yang sudah kita buat pada docker dapat menggunakan command docker container ls.



Gambar 11

Kemudian dengan menggunakan command nano untuk kedua kalinya untuk mengedit file pada /etc/apache2/sites-available/000-default.conf, Setelah kita membuat container pada network maka otomatis akan masuk ke dalam container tersebut.



Gambar 12

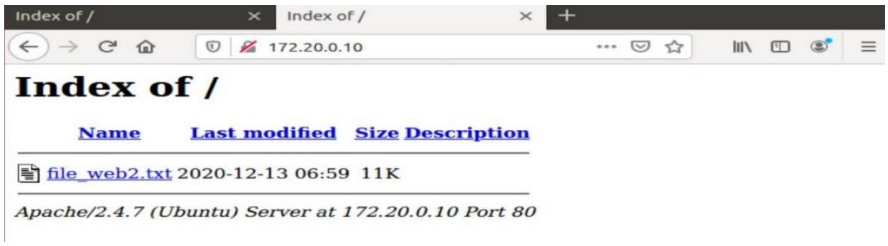
Lalu ubah server name menjadi value web2.agung.net, kemudian mengubah serverAdmin admin@agung.net setelah tambahkan masukan command <Directory /var/www/html> lalu enter setelah itu tambahkan lagi command Options Indexes FollowSymlinks lalu enter dan tambahkan command </Directory> setelah itu exit dengan command Ctrl + X.

Gambar 13

Untuk memindahkan file dari index.php menjadi file_web2.txt dengan mengunakan command mv /var/www/html/index.php /var/www/html/file_web2.txt.

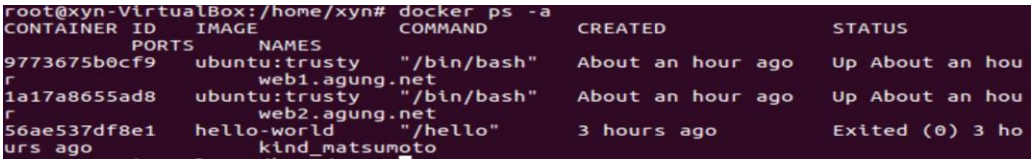
Gambar 14

Selanjutnya untuk kedua kalinya melakukan restart web server dan cleaning apache2 dengan menggunakan command service apache2 restart.



Gambar 15

Ketika berhasil maka memunculkan hasil seperti ini pada ip 172.20.0.10.

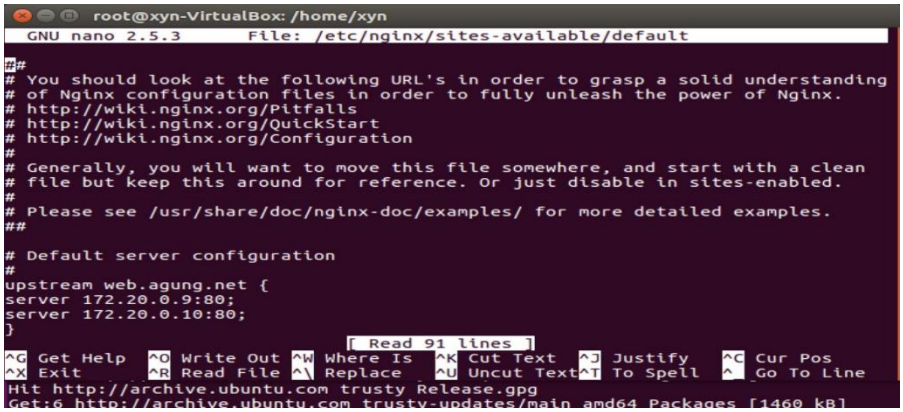


Gambar 16

Selanjutnya untuk melihat container yang terdapat pada docker menggunakan command docker ps -a.

Gambar 17

Setelah itu dillanjutkan dengan mengedit file pada nginx/sites-available/default dengan menggunakan command nano /etc/nginx/sites-available/default.



Gambar 18

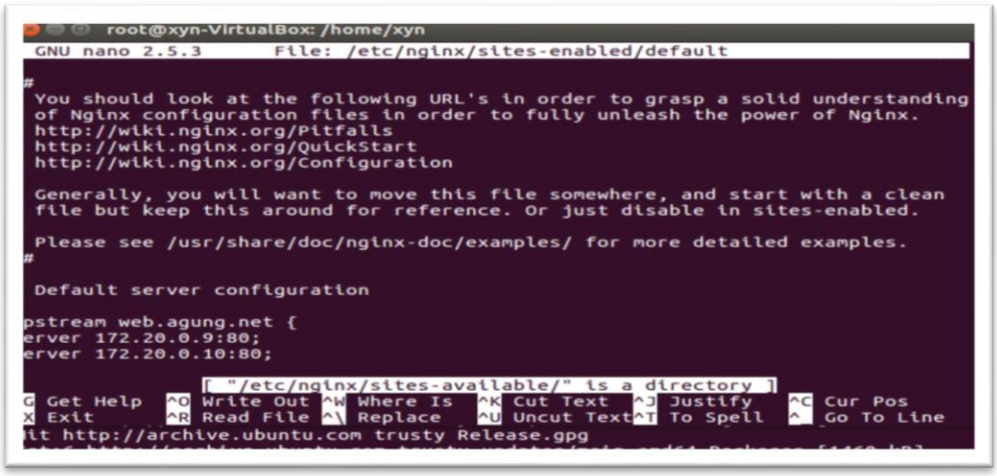
Selanjutnya untuk menggunakan kedua ip yang telah dibuat untuk dapat diakses dari dalam dengan menambahkan script kedua ip agar dapat saling mengbackup ketika antara satu jaringan dapat berfungsi atau tidak dapat diakses.

Gambar 19

Selanjutnya ini untuk mengambil file dengan command apt-get nginx -y

Gambar 20

Selanjutnya ini digunakan untuk mengakses dari dalam dengan menggunakan command nano /etc/nginx/sites-available/ /etc/nginx/sites-enabled/default.



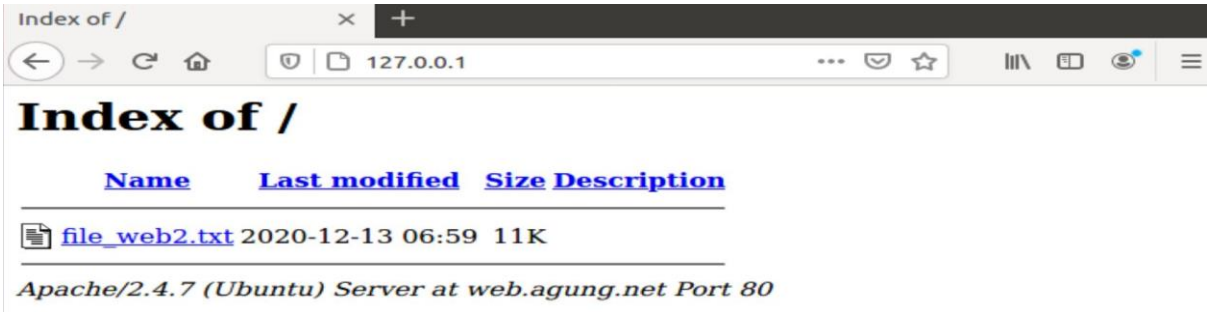
Gambar 21

Selanjutnya untuk menggunakan kedua ip yang telah dibuat untuk dapat diakses dari dalam dengan menambahkan script kedua ip agar dapat saling mengbackup ketika antara satu jaringan dapat berfungsi atau dapat diakses.

Gambar 22

Gambar 23

Selanjutnya ini merupakan contoh tampilan yang suda berhasil ketika website direstart akan memunculkan websited yang sama namun dengan file yang berbeda dikarenakan load balancing yang telah diterapkan.



Gambar 24

Selanjutnya ini merupakan contoh tampilan yang suda berhasil ketika website diresfresh maka akan memunculkan website yang sama namun dengan file yang berbeda dikarenakan load balancing yang telah diterapkan.

.

Kesimpulan

Kesimpulan yang didapatkan dari simulasi ini, dapat diketahui bahwa docker swarm dapat dijalankan pada ubuntu 16.0 dan dapat menggunakan docker untuk melakukan service load balancing, ketika salah satu server down maka akan dialihkan server lain yang terhubung sehingga menjaga website tetap aktif dan berjalan dengan normal.

Daftar Pustaka

Afis, Dimas Setiawan, Mahendra Data, and Widhi Yahya. 2019. "Load Balancing Server Web Berdasarkan Jumlah Koneksi Klien Pada Docker Swarm." *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya* 3(1):925–30.

Bella, Mochamad Rexa Mei, Mahendra Data, and Widhi Yahya. 2018. "Web Server Load Balancing Based On Memory Utilization Using Docker Swarm." *3rd International Conference on Sustainable Information Engineering and Technology, SIET 2018 - Proceedings* 220–23.

Cérin, Christophe, Tarek Menouer, Walid Saad, and Wiem Ben Abdallah. 2018. "A New Docker Swarm Scheduling Strategy." *Proceedings - 2017 IEEE 7th International Symposium on Cloud and Service Computing, SC2 2017* 2018-Janua:112–17.

Dwiyatno, Saleh, Edy Rakhmat, and Oki Gustiawan. 2020. "Implementasi Virtualisasi Server Berbasis Docker Container." *Prosisko* 7(2):165–75.

Farid, Irfan, Mohammad Idhom, and Henni Endah Wahanani. 2020. "Web Server Load Balancing Pada Arsitektur." 1(3):775–82.

Hakim, Dimara Kusuma, Johan Kun Riyanto, and Achmad Fauzan. 2020. "Pengujian Algoritma Load Balancing Pada Virtualisasi Server." *Sainteks* 16(1):33–41.

Munadi, Rendy, and Danu Dwi Sanjoyo. 2017. "Implementasi Dan Analisis Computer Clustering System Dengan Menggunakan Virtualisasi Docker." 4(3):3548–56.

Prasetyo, Stefanus Eko, Prodi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, and Average Latency. 2021. "JITE (Journal of Informatics and Telecommunication Engineering) Design and Implementation of Lightweight Virtualization Using Docker Container in Distributing Web Application with Experimental." 4(January):270–76.

Putra, Muhammad Aldi Aditia, Iskandar Fitri, and Agus Iskandar. 2020. "Implementasi High Availability Cluster Web Server Menggunakan Virtualisasi Container Docker." *Jurnal Media Informatika Budidarma* 4(1):9.

Rexa, Mohamad, Mahendra Data, and Widhi Yahya. 2019. "Implementasi Load Balancing Server Web Berbasis Docker Swarm Berdasarkan Penggunaan Sumber Daya Memory Host." *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya* 3(4):3478–87.