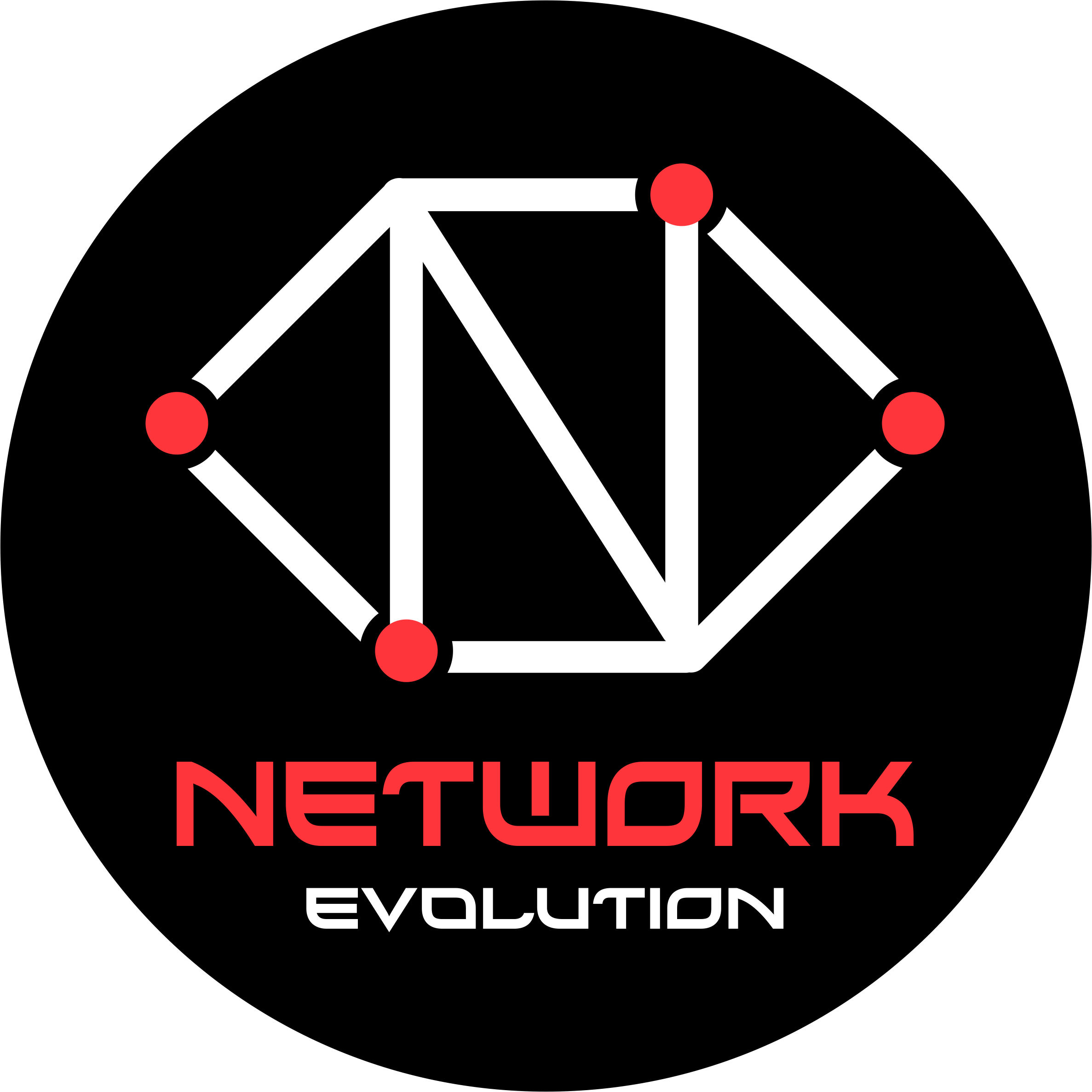
**LOAD BALANCING WEB SERVER NGINX WITH HAPROXY ON CENTOS 7**

****

**Vassa Metayasha**

**S1 TT 05D – 17101161 – Divisi NFV**

**PROGRAM STUDI S1 TEKNIK TELEKOMUNIKASI**

**FAKULTAS TEKNIK TELEKOMUNIKASI DAN ELEKTRO**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2019**

**BAB I**

**DASAR TEORI**

**1. Load Balancing**

Load Balancing adalah sebuah teknik untuk melakukan distribusi beban trafik pada dua atau lebih jalur koneksi secara seimbang sehingga setiap jalur koneksi akan menerima beban trafik yang sama. Load balancing ini sangat banyak digunakan oleh perusahaan - perusahaan skala besar. Load balancing biasanya sering digunakan untuk kebutuhan trafik web yang sangat tinggi seperti web e-commerce. Load balancing biasanya digunakan pada saat sebuah server jika user yang melakukan akses server tersebut telah melebihi jumlah maksimal. Ketika user yang mengakses melebihi kapasitas maka dengan menggunakan teknik load balancing, beban trafik tersebut akan dialihkan ke server yang lain.

Berikut beberapa teknik algoritma yang bisa digunakan oleh haproxy pada load balancing :

1. Round-Robin. Algoritma round-robin mendistribusikan beban kepada semua server anggota cluster sehingga masing - masing server mendapat beban yang sama dalam waktu yang sama. Round-robin cocok saat server anggota cluster memiliki kemampuan proccessing yang sama, jika tidak beberapa server bisa jadi menerima request lebih dari kemampuan proccessing server itu sendiri sedang yang lainnya hanya mendapat beban lebih sedikit dari resource yang dimiliki.
2. Weighted round-robin. Algoritma weighted round-robin melakukan perhitungan perbedaan kemampuan proccessing dari masing masing server anggota cluster. Administrator memasukan secara manual parameter beban yang akan ditangani oleh masing masing server anggotacluster, kemudian scheduling sequence secara otomatis dilakukan berdasarkan beban server.Request kemudian diarahkan ke server yang berbeda sesuai dengan round-robin scheduling sequence.
3. Least-connection. Algoritma Least-connection melakukan pengiriman request pada server anggota cluster, berdasarkan pada server mana yang memiliki fewest connections (koneksi paling sedikit).
4. Load-based. Algoritma Load-based mengirimkan paket request ke server anggota clusterberdasarkan server mana yang memiliki beban terkecil.

Pada load balancing kali ini akan menggunakan teknik algoritma round robin agar beban pada setiap server sama. [1]

**2. Web Server**

Server atau Web server adalah sebuah software yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS pada klien yang dikenal dan biasanya kita kenal dengan nama web browser (Mozilla Firefox, Google Chrome) dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman web dan pada umumnya akan berbentuk dokumen HTML. Fungsi utama Server atau Web server adalah untuk melakukan atau akan mentransfer berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa. halaman web yang diminta terdiri dari berkas teks, video, gambar, file dan banyak lagi. pemanfaatan web server berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web termasuk yang di dalam berupa teks, video, gambar dan banyak lagi.

Salah satu web server yang digunakan kali ini adalah nginx. NGINX adalah software web server yang open source. Ketika pertama kali dirilis, NGINX hanya berfungsi sebagai HTTP web serving saja. Namun sekarang, software tersebut juga berperan sebagai reverse proxy, HTTP load balancer, dan email proxy untuk IMAP, POP3, dan SMTP. [2]

**3. Virtualisasi Server (Cloud Computing)**

Cloud Computing adalah sistem komputerisasi berbasis jaringan/internet, dimana suatu sumber daya, software, informasi dan aplikasi disediakan untuk digunakan oleh komputer lain yang membutuhkan. Mengapa konsep ini bernama komputasi awan atau cloud computing? Ini karena internet sendiri bisa dianggap sebagai sebuah awan besar (biasanya dalam skema network, internet dilambangkan sebagai awan) yang berisi sekumpulan besar komputer yang saling terhubung, jadi cloud computing bisa diartikan sebagai komputerisasi berbasis sekumpulan komputer yang saling terhubung.

Cloud computing bisa dianggap sebagai perluasan dari virtualisasi. Perusahaan bisa menempatkan aplikasi atau sistem yang digunakan di internet, tidak mengelolanya secara internal. Contoh cloud computing untuk versi public adalah layanan-layanan milik Google seperti Google Docs dan Google Spreadsheet. Adanya kedua layanan tersebut meniadakan kebutuhan suatu aplikasi office untuk pengolah kata dan aplikasi spreadsheet di internal perusahaan. Contoh cloud computing untuk keperluan non public adalah Amazon EC2 ( Amazon Elastic Compute Cloud ). Amazon menyediakan komputer induk, kita bisa mengirim dan menggunakan sistem virtual dan menggunakannya dalam jangka waktu dan biaya sewa tertentu. [3]

**BAB II**

**MANFAAT DAN TUJUAN**

**A. MANFAAT**

1. Dapat membangun server secara virtual dengan menerapkan teknik load balancing
2. Dapat menerapkan sistem load balancing untuk mengarahkan trafik ke beberapa server
3. Dapat membedakan fungsi antara nginx (web server) dengan Haproxy (load balancer)

**B. TUJUAN**

1. Mengetahui Kinerja Load Balancing menggunakan metode algoritma round robin.
2. Melakukan pengukuran throughput, request loss, delay dan jitter pada nginx dengan haproxy.
3. Mengetahui perbedaan penggunaan antara haproxy dengan nginx.

**BAB III**

**KONFIGURASI PERANGKAT**

1. Siapkan 3 VM yang sudah terinstall OS Centos 7, dan berikan masing – masing VM RAM sebesar 512 MB dan harddisk 10gb:

link install Centos 7 : <https://www.server-world.info/en/note?os=CentOS_7&p=install>

link download Centos 7 : <https://www.server-world.info/en/note?os=CentOS_7&p=download>

**Matikan Windows Firewall terlebih dahulu pada Laptop/PC anda dengan cara :**

* Buka Control panel
* Pilih system and security
* Pilih windows firewall
* Kemudian pilih Turn Windwos Firewall on or off
* Kemudian pilih turn off seluruhnya.

1. VM 1 = Loadbalancer (NAT dan Adapter hanya host)

Login dengan Root Privileges

Install terlebih dahulu nano dengan cara :

yum –y install nano

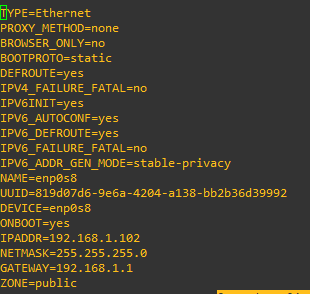
cek port yang tersedia :

nmcli d

maka akan muncul enp0s3 (DHCP) dan enp0s8 (static)

kemudian setting IP adapter hanya host (static) :

nano /etc/sysconfig/network-scripts/ifcfg-enp0s8



Jika sudah save dengan cara tekan CTRL + X, kemudian Y.

Lalu setting IP NAT (DHCP):

nano /etc/sysconfig/network-scripts/ifcfg-enp0s3

ubah pada bagian paling bawah menjadi ONBOOT=”yes”.

Selanjutnya restart network dengan cara :

systemctl restart network

1. VM 2 = nginx1

Login dengan Root Privileges

Install terlebih dahulu nano dengan cara :

yum –y install nano

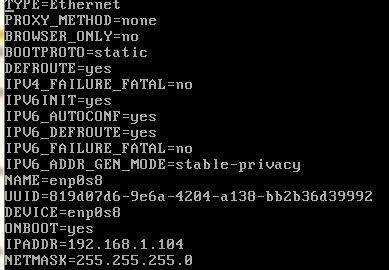
cek port yang tersedia :

nmcli d

maka akan muncul enp0s3 (DHCP) dan enp0s8 (static)

kemudian setting IP adapter hanya host (static) :

nano /etc/sysconfig/network-scripts/ifcfg-enp0s8



Jika sudah save dengan cara tekan CTRL + X, kemudian Y.

Lalu setting IP NAT (DHCP):

nano /etc/sysconfig/network-scripts/ifcfg-enp0s3

ubah pada bagian paling bawah menjadi ONBOOT=”yes”.

Selanjutnya restart network dengan cara :

systemctl restart network

1. VM 3 = nginx2

Login dengan Root Privileges

Install terlebih dahulu nano dengan cara :

yum –y install nano

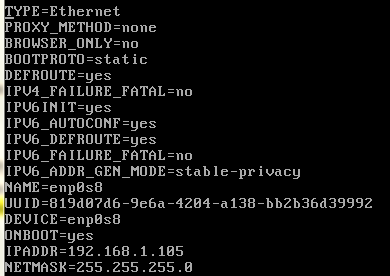
Cek port yang tersedia :

nmcli d

Maka akan muncul enp0s3 (DHCP) dan enp0s8 (static)

Kemudian setting IP adapter hanya host (static) :

nano /etc/sysconfig/network-scripts/ifcfg-enp0s8



Jika sudah save dengan cara tekan CTRL + X, kemudian Y.

Lalu setting IP NAT (DHCP):

nano /etc/sysconfig/network-scripts/ifcfg-enp0s3

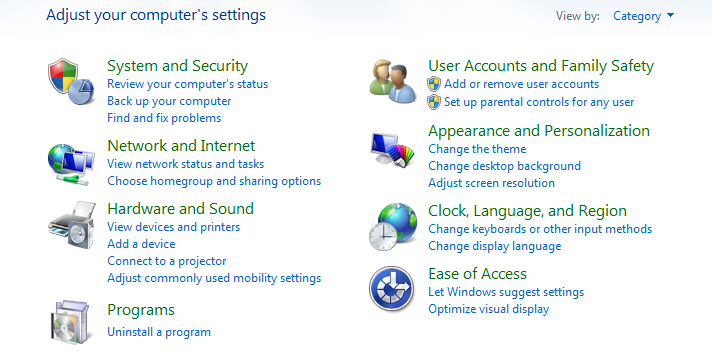
ubah pada bagian paling bawah menjadi ONBOOT=”yes”.

Selanjutnya restart network dengan cara :

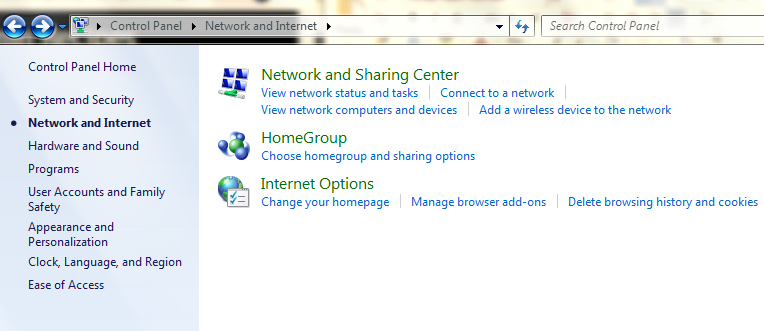
systemctl restart network

**KEMUDIAN SETTING PADA VIRTUAL BOX ADAPTER PADA LAPTOP/PC ANDA:**

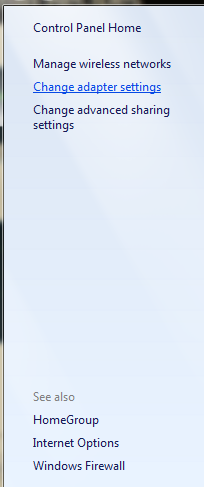
1. Buka control panel
2. Pilih Network dan Internet



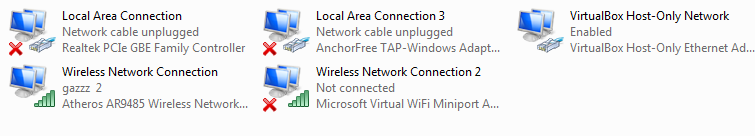
1. Pilih network and sharing



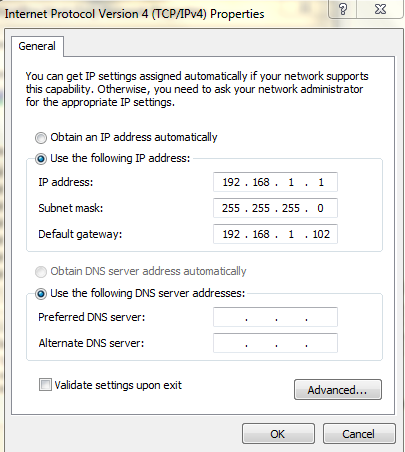
1. Pilih change adapter settings



1. Pilih yang virtual box klik 2x



1. Pilih propertis
2. Kemudian cari “Internet Protocol Version 4 (TCP/Ipv4), klik 2x
3. Setting seperti dibawah :



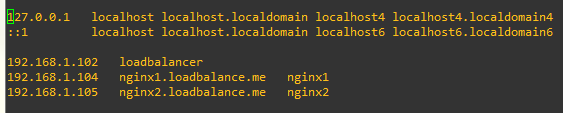
1. Kemudian OK

**MULAI SETTING LOAD BALANCER**

STEP 1 :

* Pada semua VM lakukan ini (loadbalancer, nginx1, dan nginx2)

1. Nano /etc/hosts
2. Tambahkan seperti dibawah :



1. CTRL + X, kemudian Y ( untuk keluar dan save )

STEP 2 :

* Pada @192.168.1.102 (loadbalancer)

1. yum –y update
2. kemudian install haproxy :

yum –y install haproxy

1. lakukan konfigurasi pada haproxy
2. pertama backup file haproxy.cfg :

mv /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.orig

1. kemudian buka file haproxy.cfg :

nano /etc/haproxy/haproxy.cfg

1. Paste Konfigurasi dibawah, cara pastenya pastikan kalian konfigurasinya tidak di virtual box, melainkan menggunakan aplikasi Putty :

#---------------------------------------------------------------------

# Global settings

#---------------------------------------------------------------------

global

log 127.0.0.1 local2 #Log configuration

chroot /var/lib/haproxy

pidfile /var/run/haproxy.pid

maxconn 4000

user haproxy #Haproxy running under user and group "haproxy"

group haproxy

daemon

# turn on stats unix socket

stats socket /var/lib/haproxy/stats

#---------------------------------------------------------------------

# common defaults that all the 'listen' and 'backend' sections will

# use if not designated in their block

#---------------------------------------------------------------------

defaults

mode http

log global

option httplog

option dontlognull

option http-server-close

option forwardfor except 127.0.0.0/8

option redispatch

retries 3

timeout http-request 10s

timeout queue 1m

timeout connect 10s

timeout client 1m

timeout server 1m

timeout http-keep-alive 10s

timeout check 10s

maxconn 3000

#---------------------------------------------------------------------

#HAProxy Monitoring Config

#---------------------------------------------------------------------

listen haproxy3-monitoring \*:8080 #Haproxy Monitoring run on port 8080

mode http

option forwardfor

option httpclose

stats enable

stats show-legends

stats refresh 5s

stats uri /stats #URL for HAProxy monitoring

stats realm Haproxy\ Statistics

stats auth howtoforge:howtoforge #User and Password for login to the monitoring dashboard

stats admin if TRUE

default\_backend app-main #This is optionally for monitoring backend

#---------------------------------------------------------------------

# FrontEnd Configuration

#---------------------------------------------------------------------

frontend main

bind \*:80

option http-server-close

option forwardfor

default\_backend app-main

#---------------------------------------------------------------------

# BackEnd roundrobin as balance algorithm

#---------------------------------------------------------------------

backend app-main

balance roundrobin #Balance algorithm

option httpchk HEAD / HTTP/1.1\r\nHost:\ localhost #Check the server application is up and healty - 200 status code

server nginx1 192.168.1.104:80 check #Nginx1

server nginx2 192.168.1.105:80 check #Nginx2

1. kemudian keluar dari konfigurasi dan save (CTRL + X, kemudian Y)
2. Selanjutnya lakukan konfigurasi pada rsyslog untuk haproxy

Nano /etc/rsyslog.conf

1. Kemudian hilangkan tanda pagar # (komentar) pada bagian dibawah :

$ModLoad imudp

$UDPServerRun 514

1. Dan tambahkan dibawah UDPServerRun 514, jika ingin menggunakan ip yang lebih spesifik :

$UDPServerAddress 127.0.0.1

1. Save dan keluar dari konfigurasi (CTRL + X, kemudian Y)
2. Kemudian buat file konfigurasi haproxy baru :

Nano /etc/rsyslog.d/haproxy.conf

1. Copy paste konfigurasi dibawah :

local2.=info /var/log/haproxy-access.log #For Access Log

local2.notice /var/log/haproxy-info.log #For Service Info - Backend, loadbalancer

1. Save dan keluar dari konfigurasi (CTRL + X, kemudian Y)
2. Sekarang restart rsyslog dan start haproxy :

Systemctl restart rsyslog

Systemctl start haproxy

Systemctl enable haproxy

STEP 3 :

Pada step ini proses install nginx dari epel repository. **Lakukan ini pada VM nginx1 dan VM nginx2** :

1. Install epel repository :

yum –y install epel-release

1. Kemudian install nginx :

yum –y install nginx

1. Setelah selesai proses instalasi, kemudian ubah isi yang ada pada File index.html :

Cd /usr/share/nginx/html/

1. Kemudian ketik pada server nginx1 :

echo “<h1>nginx1.loadbalance.me</h1>” > index.html

1. Pada server nginx2 :

echo “<h1>nginx2.loadbalance.me</h1>” > index.html

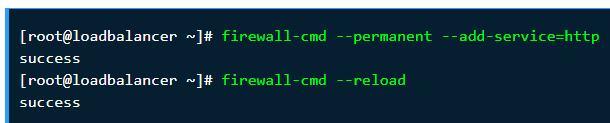
1. Selanjutnya start nginx dengan cara :

systemctl enable nginx

systemctl start nginx

1. Kemudian setting firewall agar zone menjadi public (**LAKUKAN INI PADA SEMUA VM**):

Lakukan pada VM loadbalancer, VM nginx1, VM nginx2.



STEP 4 (boleh dilakukan, boleh tidak dilakukan) rekomendasi dilakukan :

Pada step kali ini, matikan jaringan NAT yang terhubung pada Server nginx1 dan server nginx2 saja dan biarkan NAT pada loadbalancer menyala.

LAKUKAN INI PADA VM nginx1 dan VM nginx2 :

1. untuk keluar dari seluruh folder

cd (ketik saja cd kemudian enter)

1. nmcli d
2. kali ini enp0s3 akan dimatikan (diputus) koneksinya
3. buka :

nano /etc/sysconfig/network-scripts/ifcfg-enp0s3

1. kemudian ubah ONBOOT=”yes” menjadi ONBOOT=”no”
2. save dan keluar dari konfigurasi (CTRL + X, kemudian Y)
3. kemudian restart network

systemctl restart network

STEP 5 : melakukan test

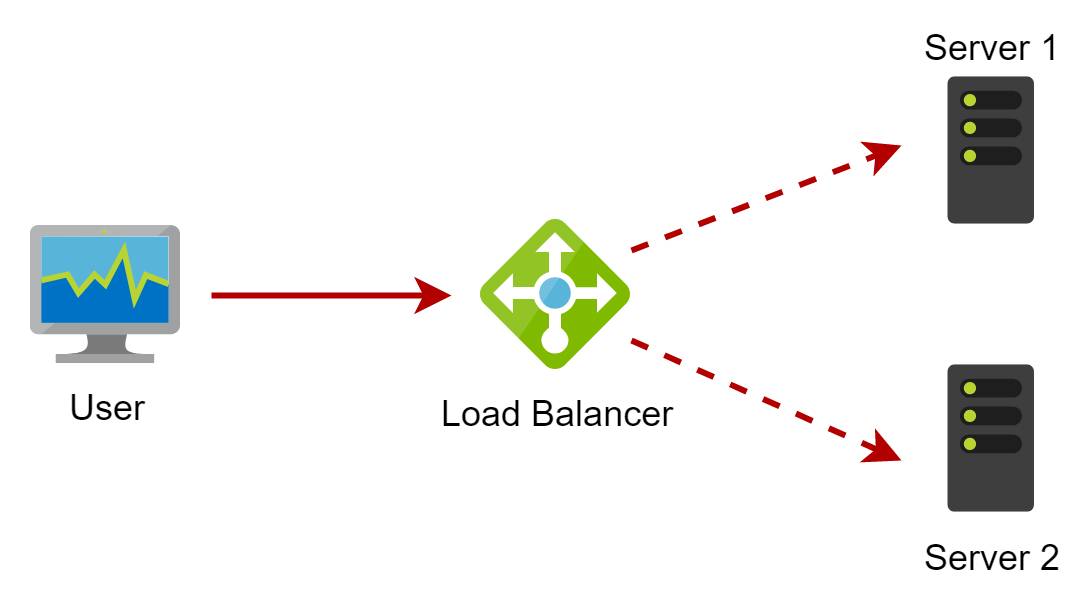
1. Buka Google chrome (Rekomendasi)
2. Ketik 192.168.1.102 (sebagai IP Loadbalancer)
3. Kemudian enter
4. Maka akan muncul Tulisan yang ada pada file index.html tadi
5. Kemudian refresh menggunakan F5

Maka akan berubah tampilan dari server nginx1 menjadi tampilan server nginx2.

**BAB IV**

**HASIL DATA DAN ANALISIS**

**1. Gambar diagram alur proses dan penjelasan.**



Gambar 1.1 Topologi Load balancing [4]

Load balancing kali ini menggunakan 1 client, 1 load balancer (untuk mengatur pembagian beban server), 2 web server (menggunakan Nginx). Dimana client disini adalah laptop pribadi dan loadbalancer serta web server disini adalah berbentuk virtual (diinstall dengan Virtual Box). Dimana pada load balancer diinstall packet service yaitu Haproxy yang dimana haproxy ini diberikan algoritma round robin untuk mengatur pemberian beban serta membagi beban kepada server. Dimana client akan mengakses server dengan menggunakan alamat ip yang nantinya load balancer disini akan bertugas untuk mengatur jalur akses dari client menuju server yang diingikan.

**2. Service yang dianalisis dan penjelasan.**

Pada analisa Quality of Service kali ini menggunakan aplikasi wireshark dimana aplikasi ini bisa menghitung jumlah througput, packet loss, delay, dan jitter. Dimana kali ini akan membandingkan QoS antara Load Balancing server menggunakan haproxy dengan algorritma round robin dengan Web Server Nginx (Single VM).

1. Throughput

Throughput atau bisa dikatakan sebagai bandwith yang sebenarnya adalah kemampuan sebenarnya suatu jaringan mengirimkan data, yang dinamis sesuai keadaan dari trafik jaringan.

1. Packet Loss

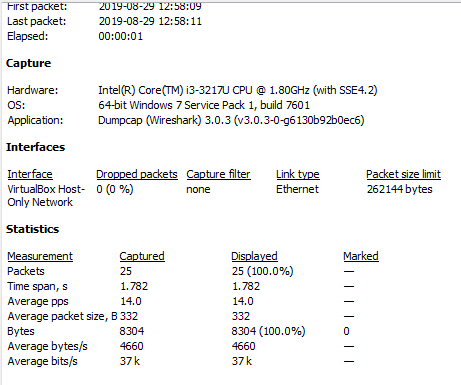
Adalah perbandingan seluruh paket data yang hilang dengan total paket data dikirimkan antara sumber dan tujuan pengiriman.

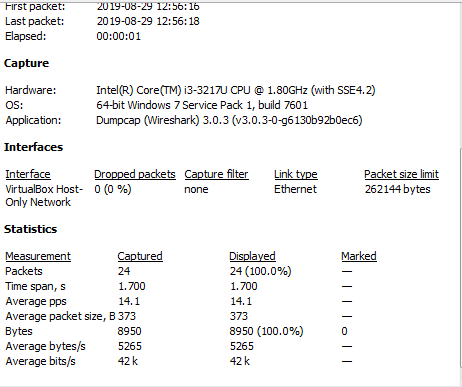
1. Delay

Adalah waktu tunda sebuah paket data yang disebabkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuan.

1. Jitter

Adalah kumpulan dari semua delay terjadi dalam jaringan pada saat waktu tertentu.

**3. Hasil Data dan penjelasan.**

Gambar 3.1 QoS nginx (single VM)

Gambar 3.2 QoS Load Balancing Haproxy

Kali ini saya akan membandingkan throughput, packet loss, delay, dan jitter dengan menjalankan load balancing server dan web server (single VM) dalam waktu 1,7 detik melakukan 10x request terhadap web server (nginx) single VM dan pada haproxy. Bisa dilihat pada gambar diatas, keduanya mendapatkan packet captured hampir sama dalam wakti 1,7 detik tetapi mendapatkan jumlah througput, delay dan jitter yang berbeda.

1. Throughput

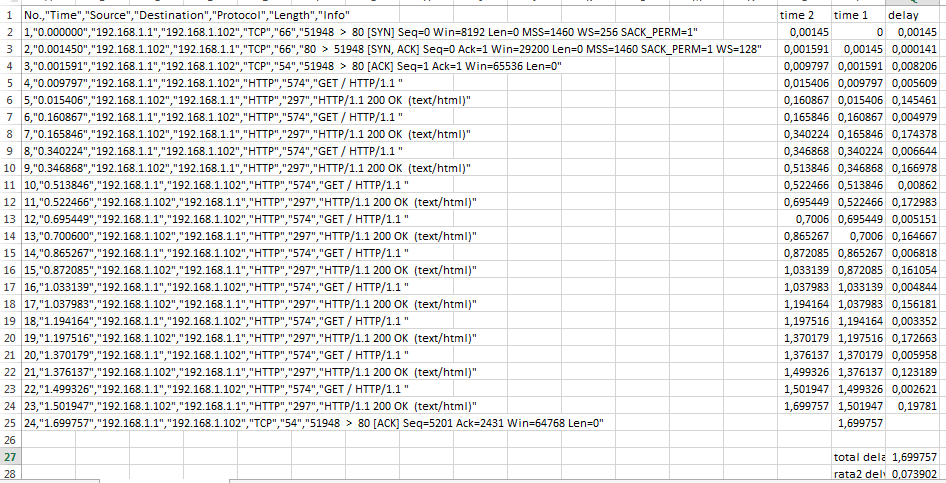
Throughput pada gambar diatas terletak pada bagian Average bits/s. Pada gambar pertama pada bagian load balancing, throughput yang didapatkan sebanyak 42k bit/s berbeda dengan throughput yang didapatkan pada web server (single VM), pada bagian single VM mendapatkan lebih banyak jumlah througput sebanyak 37k bit/s.

1. Packet Loss

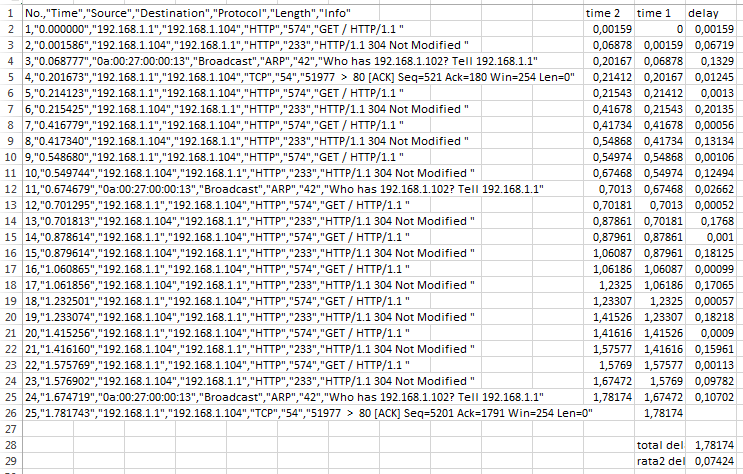
Pada pengukuran keduanya tidak terdapat packet loss dikarenakan waktu yang dilakukan terlalu singkat serta jumlah file yang diberikan sangat kecil sehingga tidak terjadi packet yang hilang.

1. Delay

Pengukuran delay secara manual bisa dilakukan dengan menyimpan data TCP kemudian waktu yang didapatkan dihitung dengan cara “time 2 – time 1” maka akan mendapatkan hasil delay pada setiap captured. Kemudian total delay bisa dilihat dalam gambar diatas yaitu pada load balancing dilakukan selama 1,700 detik serta 1,782 detik pada nginx (single VM). Selanjutnya jika ingin mengukur rata – rata delay dengan cara “total delay / (jumlah captured – 1)” diukur pada Ms.Excel agar lebih mudah. Seperti gambar dibawah :



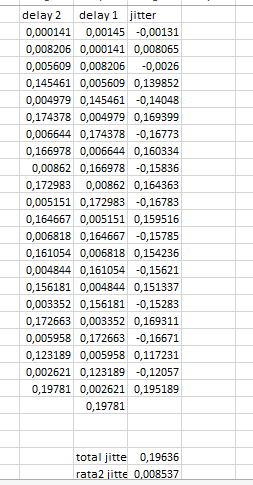
Gambar 3.3 Pengukuran Delay pada Load Balancing



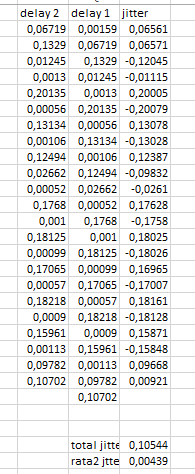
Gambar 3.4 Pengukuran Delay pada Nginx (Single VM)

1. Jitter

Adalah kumpulan dari semua delay terjadi dalam jaringan pada saat waktu tertentu. Pengukuran jitter tidak jauh beda dengan pengukuran delay. Jika delay diukur dengan time 1 dan time 2 yang didapatkan dari hasil tiap captured. Jitter diukur dengan cara “delay 2 – delay 1” dengan hasil delay yang didapatkan setiap captured. Begitupun rata – rata jitter diukur dengan cara “jumlah jitter / (jumlah captured – 1) Seperti gambar dibawah :



Gambar 3.5 Pengukuran Jitter Load Balancing



Gambar 3.6 Pengukuran Jitter Nginx

**BAB IV**

**KESIMPULAN**

Berdasarkan data yang telah dikumpulkan dapat dilihat bahwa dalam waktu 1,7 detik throughput yang didapatkan oleh load balancing haproxy lebih banyak dibandingkan dengan nginx (single VM). Dapat dikatakan bahwa sistem load balancing menggunakan 3 buah Mesin VM lebih baik dibandingkan dengan menggunakan single VM, karena dengan menggunakan sistem load balancing haproxy server lebih mudah menerima banyak data atau paket yang diterima. Berbeda dengan nginx yang hanya menggunakan single VM, walaupun dengan jumlah waktu dan request yang sama tetapi web server single VM ini tidak begitu banyak menerima jumlah paket (throughput).

**DAFTAR PUSTAKA**

|  |  |
| --- | --- |
| [1] | R. Mufrizal, "Belajar Load Balancing Dengan Haproxy, Docker, dan Spring Boot," [Online]. Available: https://rizkimufrizal.github.io/belajar-load-balancing-dengan-haproxy-docker-dan-spring-boot/. [Accessed 28 Agustus 2019]. |
| [2] | IDCloudHost, "Pengertian Web Server dan Fungsinya," [Online]. Available: https://idcloudhost.com/pengertian-web-server-dan-fungsinya/. [Accessed 28 Agustus 2019]. |
| [3] | M. V. Sugianto, "Keuntungan Teknologi Virtualisasi Cloud Computing," [Online]. Available: https://www.excellent.co.id/product-services/vmware/keuntungan-teknologi-virtualisasi-cloud-computing/. [Accessed 28 Agustus 2019]. |
| [4] | R. Ariyolo, "Konfigurasi Dasar Server Load Balancer di Alibaba Cloud," [Online]. Available: https://www.ariyolo.id/konfigurasi-dasar-server-load-balancer-di-alibaba-cloud. [Accessed 29 Agustus 2019]. |
| [5] | M. Arul, "How to setup HAProxy as Load Balancer for Nginx on CentOS 7," [Online]. Available: https://www.howtoforge.com/tutorial/how-to-setup-haproxy-as-load-balancer-for-nginx-on-centos-7/#comments. [Accessed 27 Agustus 2019]. |