

Automation Load Balancing Web Server

Proyek Akhir MSIB

Fahrijar Yusuf

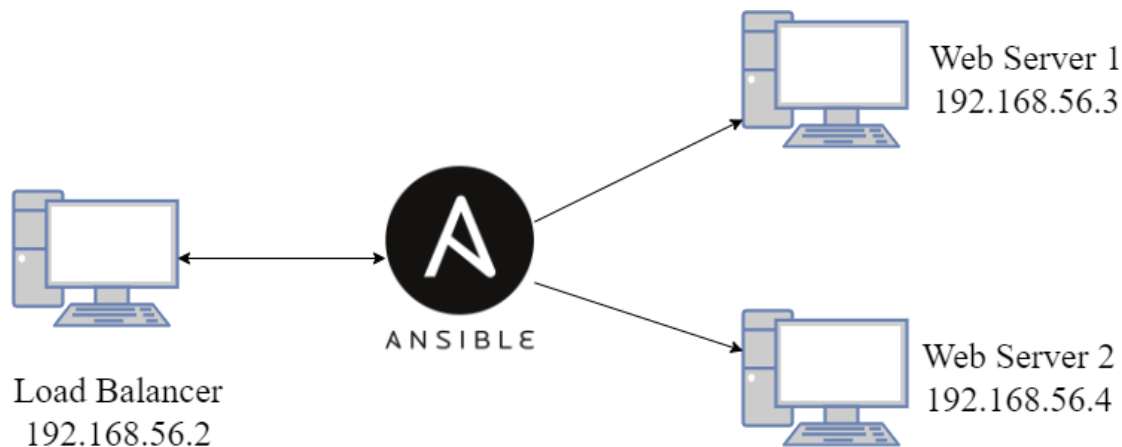
Juli 2022

1. Implementasi Sistem

Persiapan proyek dengan menggunakan 3 *virtual machine* dengan sistem operasi ubuntu server yang berbasis *command-line interface* (CLI). Setiap komputer akan berada dalam network yang sama yaitu:

1. loadb: 192.168.56.2
2. websev1: 192.168.56.3
3. websev2: 192.168.56.4

Proyek ini menggunakan Apache sebagai *web server*, Haproxy sebagai *load balancer*, dan Ansible untuk melakukan automasi. Topologi dapat dilihat pada Gambar dibawah ini.



Gambar 1. Topologi Proyek

Pada *web server*, setiap komputer belum dilakukan instalasi Apache. Karena untuk pemasangan aplikasi akan dilakukan oleh Ansible secara automasi untuk mempermudah pemasangan aplikasi pada 2 perangkat atau lebih. Dapat diperhatikan pada Gambar 2 yang menjelaskan pada jam tersebut Apache belum terinstall.

```
root@websev1:/home/web# systemctl status apache2
Unit apache2.service could not be found.
root@websev1:/home/web# date
Sat Jul  2 03:41:45 PM UTC 2022
root@websev1:/home/web#

root@websev2:/home/web# systemctl status apache2
Unit apache2.service could not be found.
root@websev2:/home/web# date
Sat Jul  2 03:40:58 PM UTC 2022
root@websev2:/home/web# _
```

Gambar 2. Pengecekan *web server*

Gambar 2 menjelaskan jika Apache belum terpasang pada kedua komputer tersebut.

1.1 Pemasangan Key SSH

Ansible belum dapat terhubung dengan setiap hosts, maka dilakukan konfigurasi ssh agar ansible dapat dengan mudah menjalankan semua hosts dalam satu waktu. Dengan konfigurasi seperti

`ssh-keygen`

Kemudian akan diminta direktori mana yang akan menjadi pusat penyimpanan dari kunci ssh tersebut. Dapat dilihat pada Gambar 3.

```
root@loadb:/home/loadb# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /root/.ssh/ansible
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/ansible
Your public key has been saved in /root/.ssh/ansible.pub
The key fingerprint is:
SHA256:UsU0G1WpAnvyALcbZhIFDuanHuI08NiVDC1GdhUCrbY root@loadb
The key's randomart image is:
+---[RSA 3072]-----+
| .B++++..+=... |
| +0*=00 +... |
| ...0*+ +0 . |
| .+0+. 0.0 . |
| 0+=. +.BS. |
| 00E. ... |
| .. |
| |
+-----[SHA256]-----+
```

Gambar 3. Pemasangan file konfigurasi ssh-keygen

Langkah selanjutnya adalah membuat key ssh yang akan digunakan, proyek ini akan menggunakan ssh untuk kedua hosts yang menjadi web server agar dapat diautomasi ansible. Dimulai dengan pemasangan dengan syntax:

```
ssh-copy-id -i ~/.ssh/ansible web@192.168.56.3
```

```
ssh-copy-id -i ~/.ssh/ansible web@192.168.56.4
```

```

root@loadb:/home/loadb# ssh-copy-id -i ~/.ssh/ansible web@192.168.56.3
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/ansible.pub"
The authenticity of host '192.168.56.3 (192.168.56.3)' can't be established.
ED25519 key fingerprint is SHA256:GVjqqy1alxFis0Ui+ZpM43JLh1WWB96RKSek030y+wds.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
all the new keys
web@192.168.56.3's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'web@192.168.56.3'"
and check to make sure that only the key(s) you wanted were added.

root@loadb:/home/loadb# _

```

Gambar 4. Pemasangan ssh key untuk setiap hosts

Pada Gambar 4 memperlihatkan pemasangan key untuk websev1 telah berhasil ditambahkan. Kemudian jalankan ssh untuk mengecek apakah ansible sudah dapat dijalankan. Perhatikan pada Gambar 5.

```

root@loadb:/home/loadb# ssh -i ~/.ssh/ansible web@192.168.56.3
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-25-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sat Jul  2 03:16:54 PM UTC 2022

System load:  0.044921875      Processes:            107
Usage of /:   49.6% of 8.02GB   Users logged in:     1
Memory usage: 21%              IPv4 address for enp0s3: 192.168.56.3
Swap usage:   0%

0 updates can be applied immediately.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet
or proxy settings

Last login: Sat Jul  2 15:12:22 2022
web@websev1:~$ _

```

Gambar 5. Uji coba ssh berhasil

1.2 Ansible

Penginstalan Ansible membutuhkan repository untuk dapat disimpan semua konfigurasi. Maka sebelum instalasi Ansible, lakukan “*add-apt-repository --yes --update --ppa:ansible/ansible*” untuk menambahkan repository. Selanjutnya lakukan instalasi Ansible dengan perintah “*apt-get install ansible -y*”

```
root@loadb:/home/loadb# apt install ansible -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ansible is already the newest version (2.10.7+merged+base+2.10.8+dfsg-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Gambar 6. Instalasi Ansible

Pada Gambar 6 memperlihatkan pemasangan Ansible telah berhasil dilakukan, selanjutnya buatlah direktori untuk penyimpanan konfigurasi. Kemudian buat file untuk mengeset user dan key yang akan digunakan ansible dengan membuat nama grup seperti pada Gambar 7.

```
[all:vars]
ansible_user=web
ansible_ssh_private_key_file=/root/.ssh/ansible

[webserver]
192.168.56.3
192.168.56.4
```

Gambar 7. Settingan digunakan agar dapat diakses ansible

Setelah konfigurasi diatas telah selesai, kemudian cek status apakah sudah terhubung dengan para host atau tidak dengan perintah “ansible-inventory -i penyimpanan –list” dapat perhatikan pada Gambar 8.

```
root@loadb:/home/loadb/ansible# ansible-inventory -i inventory --list
{
  "_meta": {
    "hostvars": {
      "192.168.56.3": {
        "ansible_ssh_private_key_file": "/root/.ssh/ansible",
        "ansible_user": "web"
      },
      "192.168.56.4": {
        "ansible_ssh_private_key_file": "/root/.ssh/ansible",
        "ansible_user": "web"
      }
    }
  },
  "all": {
    "children": [
      "ungrouped",
      "webserver"
    ]
  },
  "webserver": {
    "hosts": [
      "192.168.56.3",
      "192.168.56.4"
    ]
  }
}
```

Gambar 8. Pemasangan inventory untuk hosts

Selanjutnya lakukan pengujian koneksi ansible ke 2 hosts tersebut apakah dapat berjalan atau error dengan perintah “*ansible -i penyimpanan all -m ping*” perhatikan pada Gambar 9.

```
root@loadb:/home/loadb/ansible# ansible -i inventory all -m ping
192.168.56.3 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.56.4 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
root@loadb:/home/loadb/ansible#
```

Gambar 9. Ping hosts menggunakan Ansible

Selanjutnya buatlah file untuk dilakukan *playbook*, disini saya membuat file untuk tugas penginstallan Apache untuk digunakan 2 hosts web server dengan nama file “*apache.yml*”.

```
---
- hosts: webserver
  tasks:
    - name: instal apache
      apt:
        name: apache2
        state: latest
        become: yes
```

Gambar 10. Konfigurasi task instalasi Apache

Gambar 10 adalah konfigurasi untuk *playbook* “*apache.yml*”. Selanjutnya jalankan *playbook* tersebut dengan perintah “*ansible-playbook -i inventory apache.yml*” dapat diperhatikan pada Gambar 11.

```

root@loadb:/home/loadb/ansible# ansible-playbook -i inventory apache.yml
Command 'ansible-playbook' not found, did you mean:
  command 'ansible-playbook' from deb ansible (2.10.7+merged+base+2.10.8+dfsg-1)
  command 'ansible-playbook' from deb ansible-core (2.12.0-1)
Try: apt install <deb name>
root@loadb:/home/loadb/ansible# ls
apache.yml  inventory
root@loadb:/home/loadb/ansible# ansible-playbook -i inventory apache.yml

PLAY [webserver] *********************************************************************

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.4]
ok: [192.168.56.3]

TASK [instal apache] *************************************************************
changed: [192.168.56.4]
changed: [192.168.56.3]

PLAY RECAP *********************************************************************
192.168.56.3      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
0 ignored=0
192.168.56.4      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
0 ignored=0

root@loadb:/home/loadb/ansible# _

```

Gambar 11. *Playbook* berhasil dijalankan

Gambar 11 menjelaskan jika *playbook* telah berhasil berjalan dengan pesan “*ok*” dan “*changed*”.

1.3 Apache

Proyek ini menggunakan 2 *web server*, tentu untuk membuat program akan memakan waktu yang lama. Maka untuk mempercepat waktu konfigurasi, dilakukan *ansible* agar konfigurasi dapat dilakukan pada 2 *web server* pada waktu yang sama. Setelah *playbook* instalasi Apache telah berhasil selanjutnya lakukan pengecekan apakah sudah dapat berjalan atau belum dengan perintah “*systemctl status apache2*”.


```

root@websev1:/# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2022-07-02 17:27:37 UTC; 3min 30s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2385 (apache2)
     Tasks: 55 (limit: 1034)
    Memory: 5.2M
       CPU: 95ms
    CGroup: /system.slice/apache2.service
            └─2385 /usr/sbin/apache2 -k start
              └─2387 /usr/sbin/apache2 -k start
                └─2388 /usr/sbin/apache2 -k start

Jul 02 17:27:37 websev1 systemd[1]: Starting The Apache HTTP Server...
Jul 02 17:27:37 websev1 apachectl[2384]: AH00558: apache2: Could not reliably determine the server's
Jul 02 17:27:37 websev1 systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)

root@websev1:/# date
Sat Jul  2 05:31:23 PM UTC 2022
root@websev1:/# _

```

Gambar 12. Status Apache

Gambar 12 diperlihatkan pada waktu tersebut Apache sudah berhasil dipasang. Selanjutnya buatlah file *playbook* baru untuk konfigurasi *web server*. Saya membuat file yang diberi nama “*tampilan.yml*” dapat diperhatikan Gambar 13 untuk konfigurasi tasknya.

```

---
- hosts: webserver
  remote_user: web
  vars:
    variable1: 'Pengerjaan Proyek Akhir'
  tasks:
    - name: Copy teks ini ke Server
      template:
        src: index.j2
        dest: /var/www/html/index.html
        become: yes

```

Gambar 13. Konfigurasi *web server*

Gambar 13 diperlihatkan jika saya membuat file “*index.j2*” untuk memasukkan konfigurasi ke file yang berada di */var/www/html/index.html*. Gambar 14 akan menjelaskan konfigurasi “*index.j2*”.

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Ansible Testing Apache Web Server - {{ variable1 }} </title>
</head>

<body>
Proyek Akhir Kampus Merdeka - {{ inventory_hostname }}
</body>
</html>

```

Gambar 14. Konfigurasi *web server*

Selanjutnya jalankan playbook kembali untuk ansible mengatur untuk menjalankannya pada kedua hosts tersebut. Gambar 15 memperlihatkan status dari task yang dijalankan.

```
root@loadb:/home/loadb/ansible# ansible-playbook -i inventory tampilan.yml

PLAY [webserver] *********************************************************************

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.3]
ok: [192.168.56.4]

TASK [Copy teks ini ke Server] ********************************************************
changed: [192.168.56.3]
changed: [192.168.56.4]

PLAY RECAP *********************************************************************
192.168.56.3      : ok=2    changed=1    unreachable=0    failed=0    skipped=0
0 ignored=0
192.168.56.4      : ok=2    changed=1    unreachable=0    failed=0    skipped=0
0 ignored=0
```

Gambar 15. Status konfigurasi apache

1.4 Haproxy

Proyek ini menggunakan metode komputasi *load balancing*. Maka dibutuhkan software yang menjadi *load balancer* untuk mengatur beban pada setiap *web server*. Install Haproxy sebagai *software* dari *load balancer* dengan perintah “*apt install haproxy -y*” perhatikan pada Gambar 16.

```
root@loadb:/home/loadb# apt install haproxy -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
haproxy is already the newest version (2.4.14-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@loadb:/home/loadb#
```

Gambar 16. Instalasi Haproxy

Selanjutnya konfigurasi Haproxy agar dapat terhubung ke setiap *web server* dengan konfigurasi file “*haproxy.cfg*” dapat perhatikan pada Gambar 17.

```

        errorfile 502 /etc/haproxy/errors/502.http
        errorfile 503 /etc/haproxy/errors/503.http
        errorfile 504 /etc/haproxy/errors/504.http

frontend http_front
    bind *:80
    mode http
    default_backend http_back

backend http_back
    mode http
    balance roundrobin
    option forwardfor
    http-request set-header X-Forwarded-Port %[dst_port]
    http-request add-header X-Forwarded-Proto https if { ssl_fc }
    option httpchk HEAD / HTTP/1.1rnHost:localhost
    server websev1 192.168.56.3:80
    server websev2 192.168.56.4:80

listen stats
    bind *:1818
    stats enable
    stats hide-version
    stats refresh 30s
    stats show-node
    stats uri /stats
    stats auth user:pass

```

Gambar 17. Konfigurasi Haproxy

Langkah selanjutnya lakukan restart dan kemudian cek status apakah konfigurasi yang dilakukan berhasil dijalankan atau tidak, dapat perhatikan Gambar 18.

```

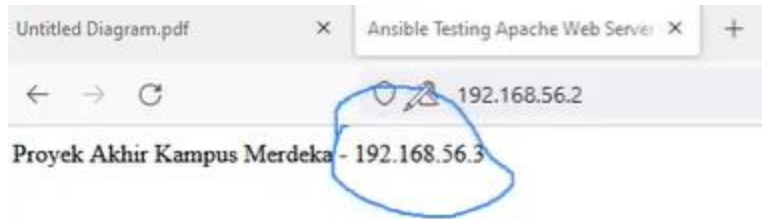
root@loadb:/# systemctl restart haproxy
root@loadb:/# systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-07-03 08:36:52 UTC; 40s ago
     Docs: man:haproxy(1)
           file:///usr/share/doc/haproxy/configuration.txt.gz
   Process: 54642 ExecStartPre=/usr/sbin/haproxy -Ws -f $CONFIG -c -q $EXTRA_OPTS (code=exited)
   Main PID: 54645 (haproxy)
    Tasks: 2 (limit: 1034)
   Memory: 71.9M
      CPU: 218ms
   CGroup: /system.slice/haproxy.service
           └─54645 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S
             54647 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S

Jul 03 08:36:52 loadb systemd[1]: Starting HAProxy Load Balancer...
Jul 03 08:36:52 loadb haproxy[54645]: [NOTICE] (54645) : New worker #1 (54647) forked
Jul 03 08:36:52 loadb systemd[1]: Started HAProxy Load Balancer.
lines 1-17/17 (END)

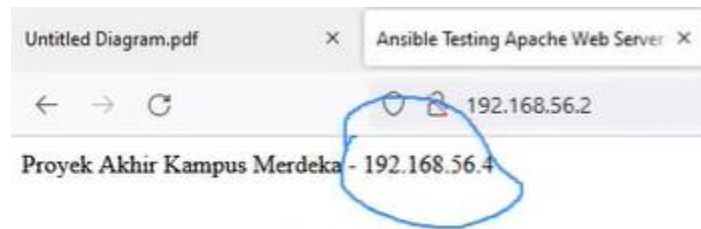
```

Gambar 18. Status Haproxy

Setelah konfigurasi berhasil dijalankan, lakukan pengujian *load balancing* pada browser dengan memanggil alamat *load balancer*. Proyek ini menggunakan alamat 192.168.56.2 sebagai *load balancer*, perhatikan Gambar 19 dan Gambar 20.



Gambar 19. Pengujian *load balancing*



Gambar 20. Pengujian *load balancing*

2. Hasil

Proyek ini mendapatkan hasil jika pemasangan *web server* Apache berhasil dilakukan dengan automasi dari Ansible dan pengujian *load balancing* untuk kedua *web server* dapat berjalan dengan sesuai. Demikian laporan proyek yang telah saya kerjakan.