

## Load Balancer

Load balancer adalah middleware yang dapat mengatur pembagian tugas terhadap server-server yang terhubung dengan load balancer. Load balancer sangat berguna untuk menangani transaksi dalam jumlah yang besar. Salah satu penyedia software load balancer adalah nginx. Kita bisa mendeploy nginx pada pc, virtual machine maupun docker. Load balancer ada beberapa macam, yaitu:

### 1. Round Robin

Buka file `/etc/nginx/sites-available/default`, lalu edit file menjadi seperti di bawah:

```
vagrant@vagrant-ubuntu-trusty-64: ~
GNU nano 2.2.6 File: /etc/nginx/sites-available/default Modified

# http://wiki.nginx.org/QuickStart
# http://wiki.nginx.org/Configuration
#
# Generally, you will want to move this file somewhere, and start with a clean
# file but keep this around for reference. Or just disable in sites-enabled.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

upstream worker{
    server 192.168.2.3:80;
    server 192.168.2.4:80;
}

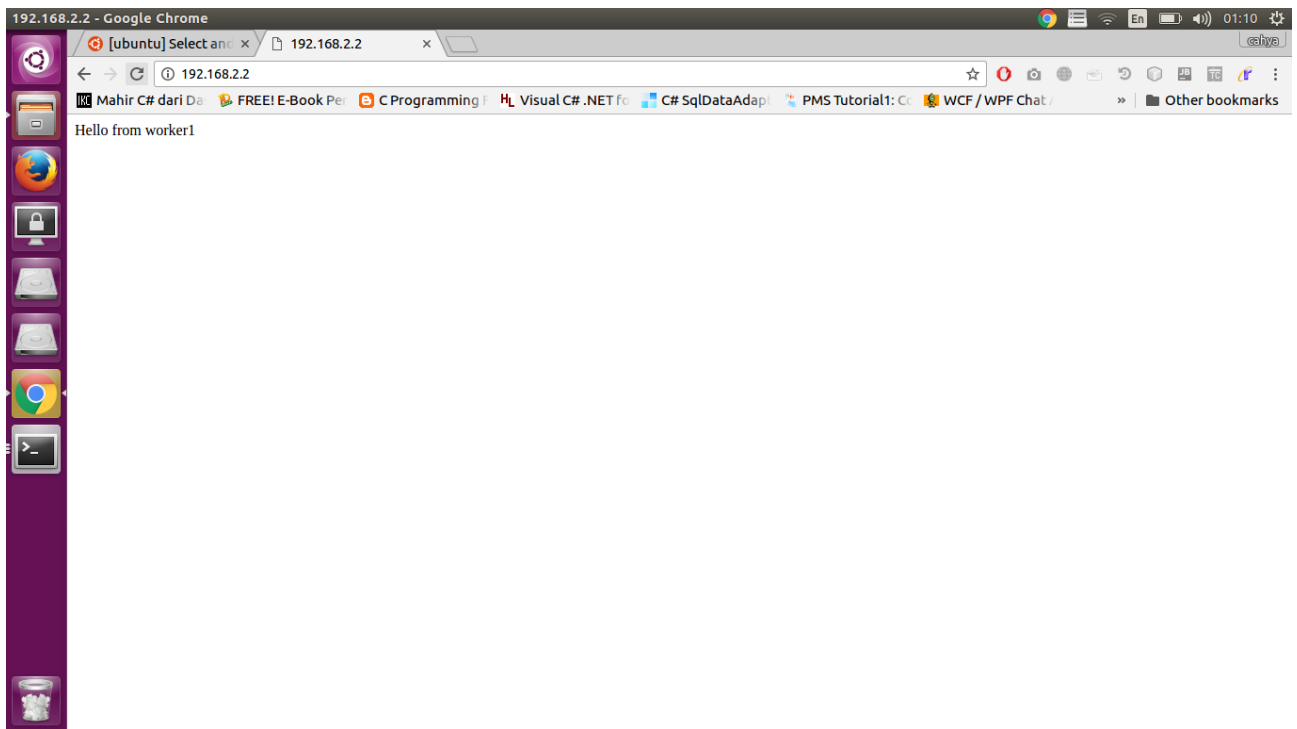
server {
    listen 80 default_server;
    listen [::]:80 default_server ipv6only=on;

    root /usr/share/nginx/html;
}

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

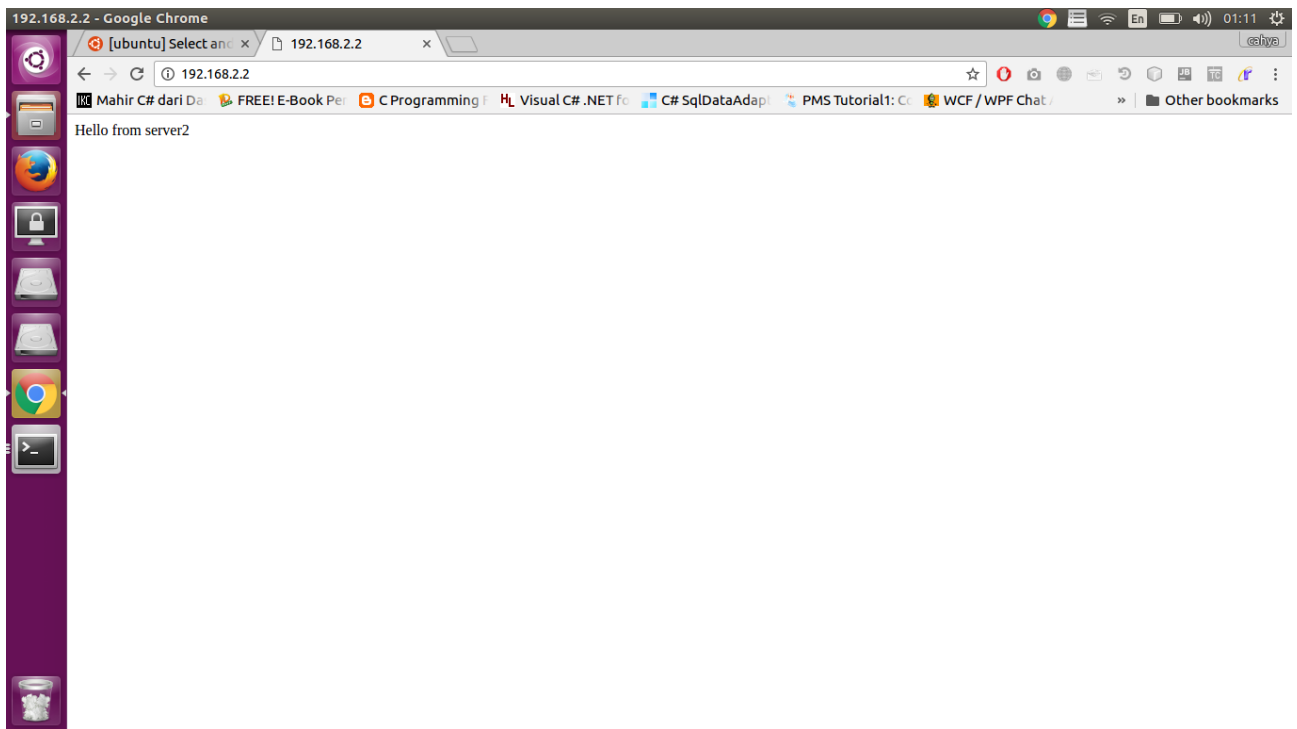
Pengujian:

Pengujian 1:

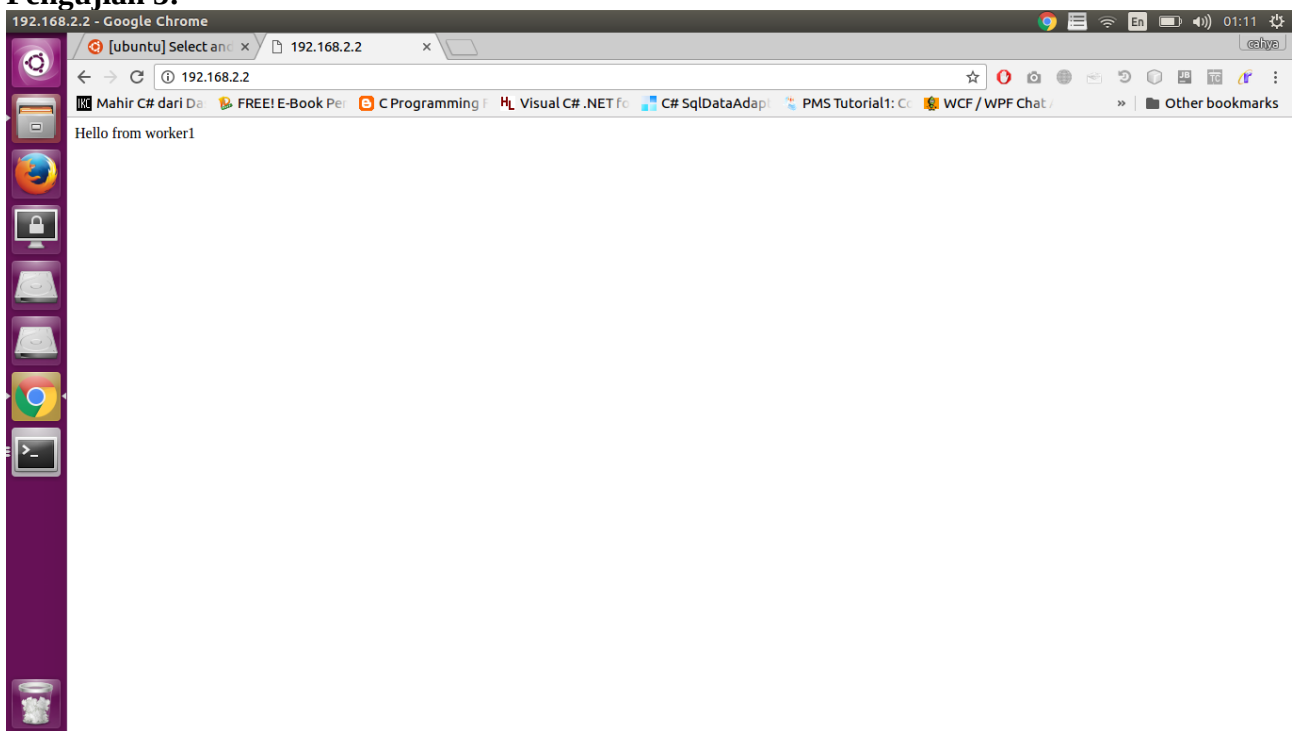


## Pengujian 2:

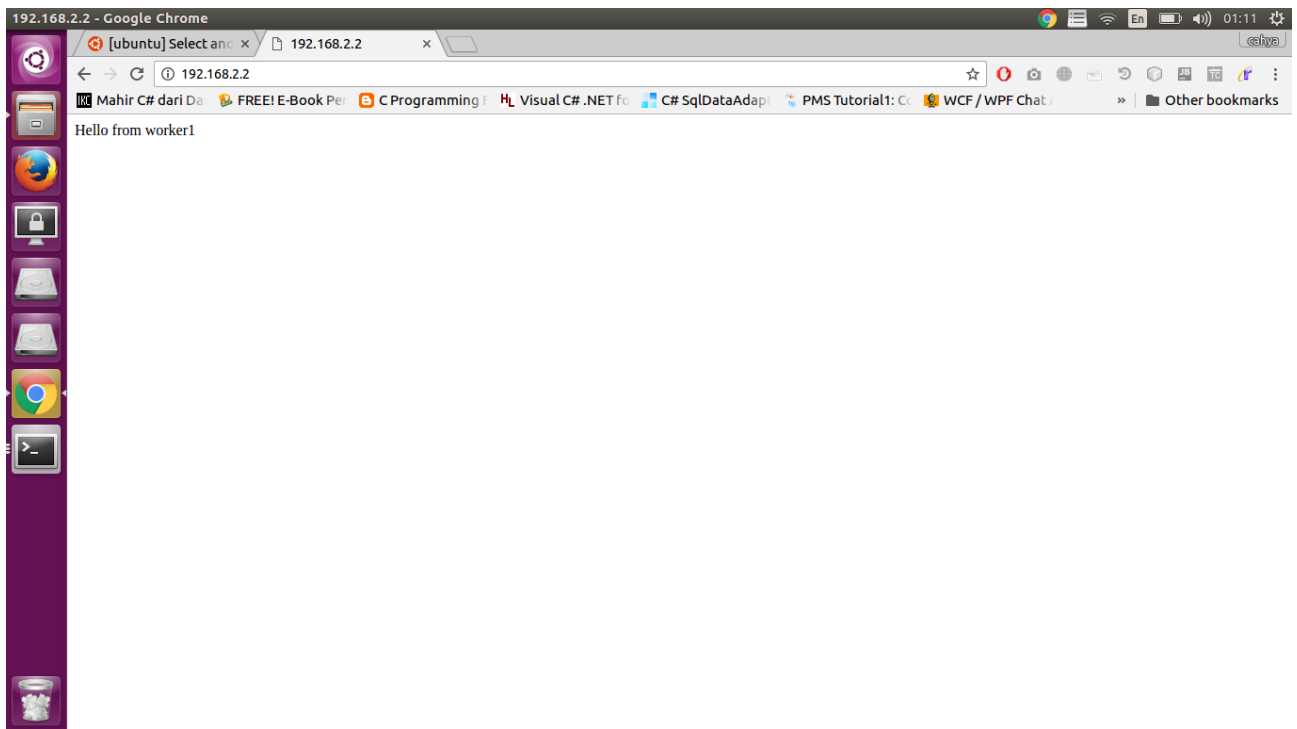




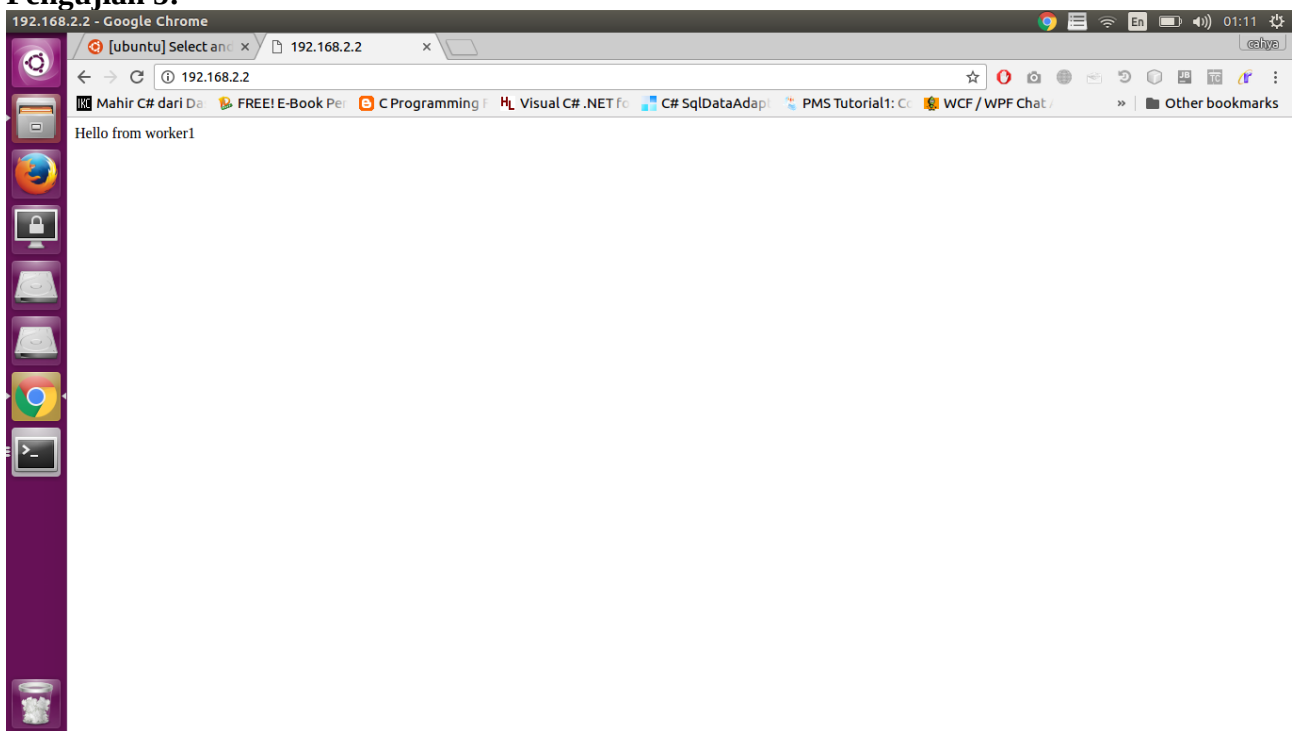
### Pengujian 3:



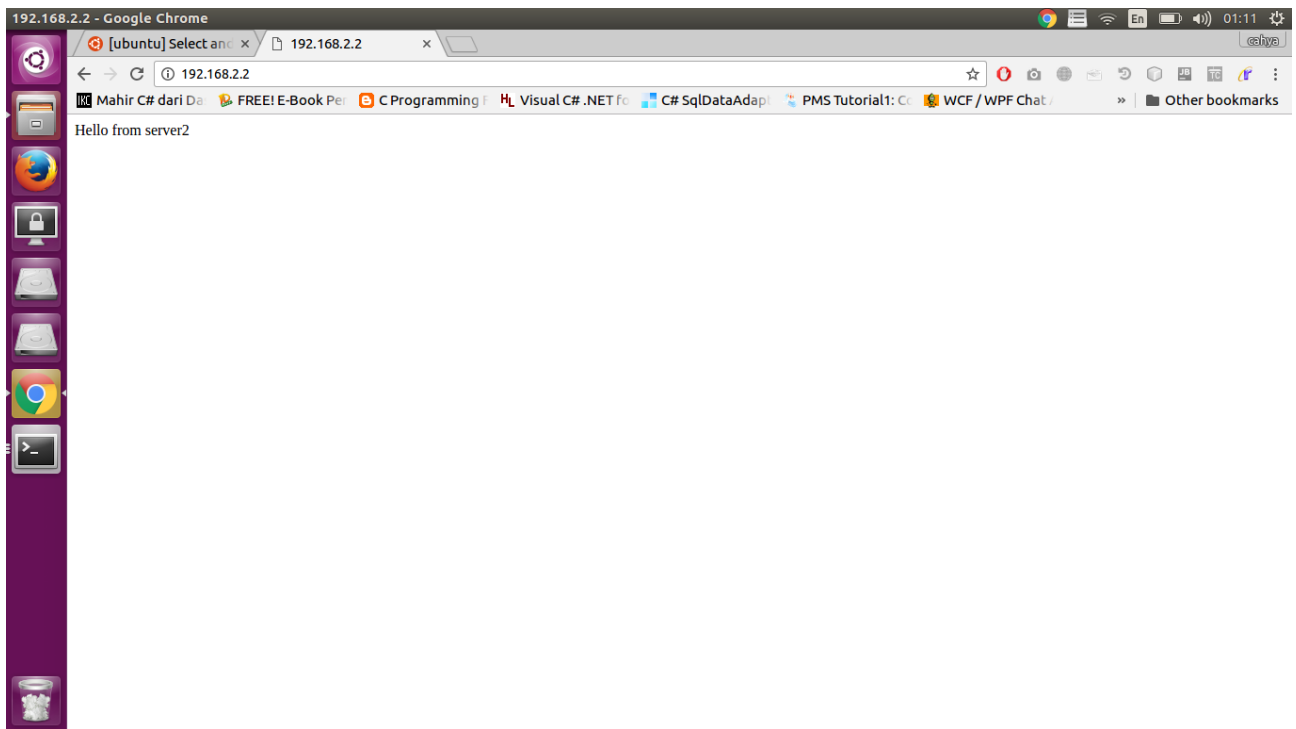
### Pengujian 4:



### Pengujian 5:



## Pengujian 6:



### **Kesimpulan:**

Round-robin menangani request sesuai urutan worker pada setting nginx. Dapat kita lihat hasil percobaan bahwa worker1 dengan worker2 bergantian menangani request.

### **2. Least-Connected**

Buka file `/etc/nginx/sites-available/default`, lalu edit file menjadi seperti di bawah:



```
vagrant@vagrant-ubuntu-trusty-64: /etc/nginx/sites-available
GNU nano 2.2.6 File: default Modified
upstream worker{
    least_conn;
    server 192.168.2.3:80;
    server 192.168.2.4:80;
}

server {
    listen 80 default_server;
    listen [::]:80 default_server ipv6only=on;

    root /usr/share/nginx/html;
    index index.html index.htm;

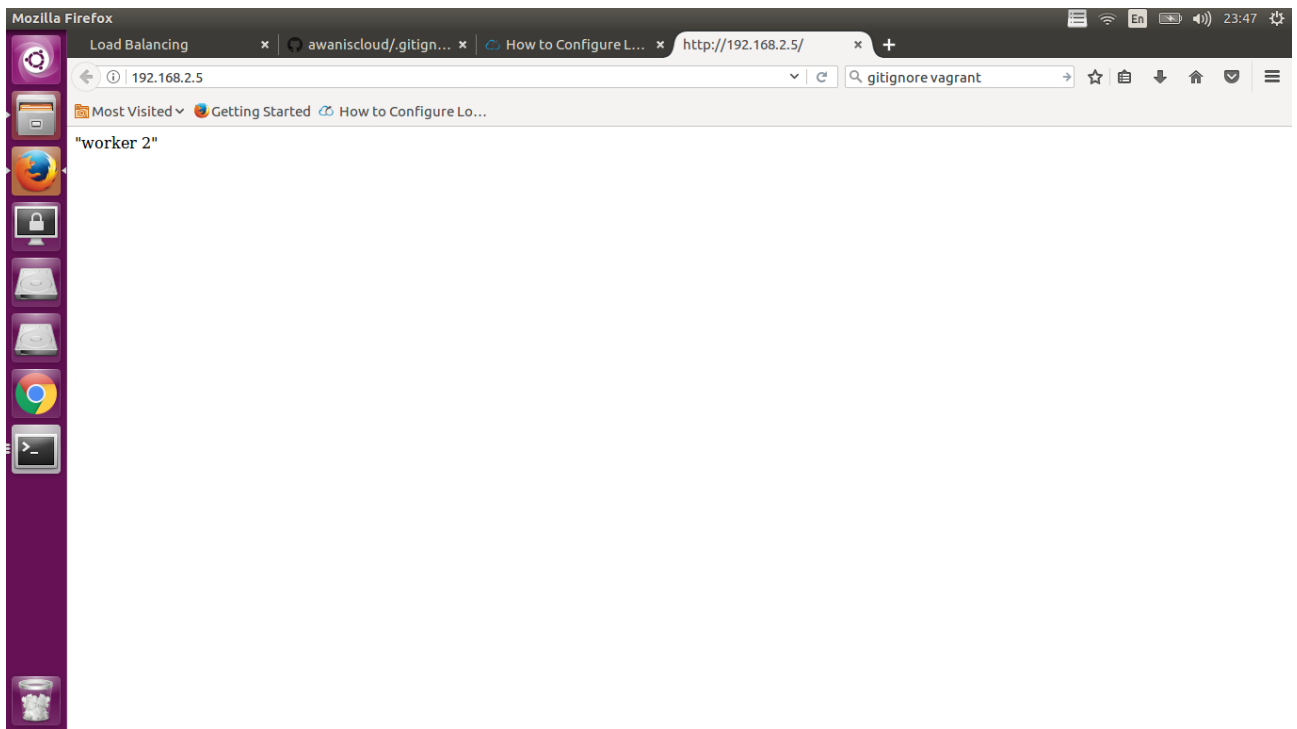
    # Make site accessible from http://localhost/
    server_name localhost;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.

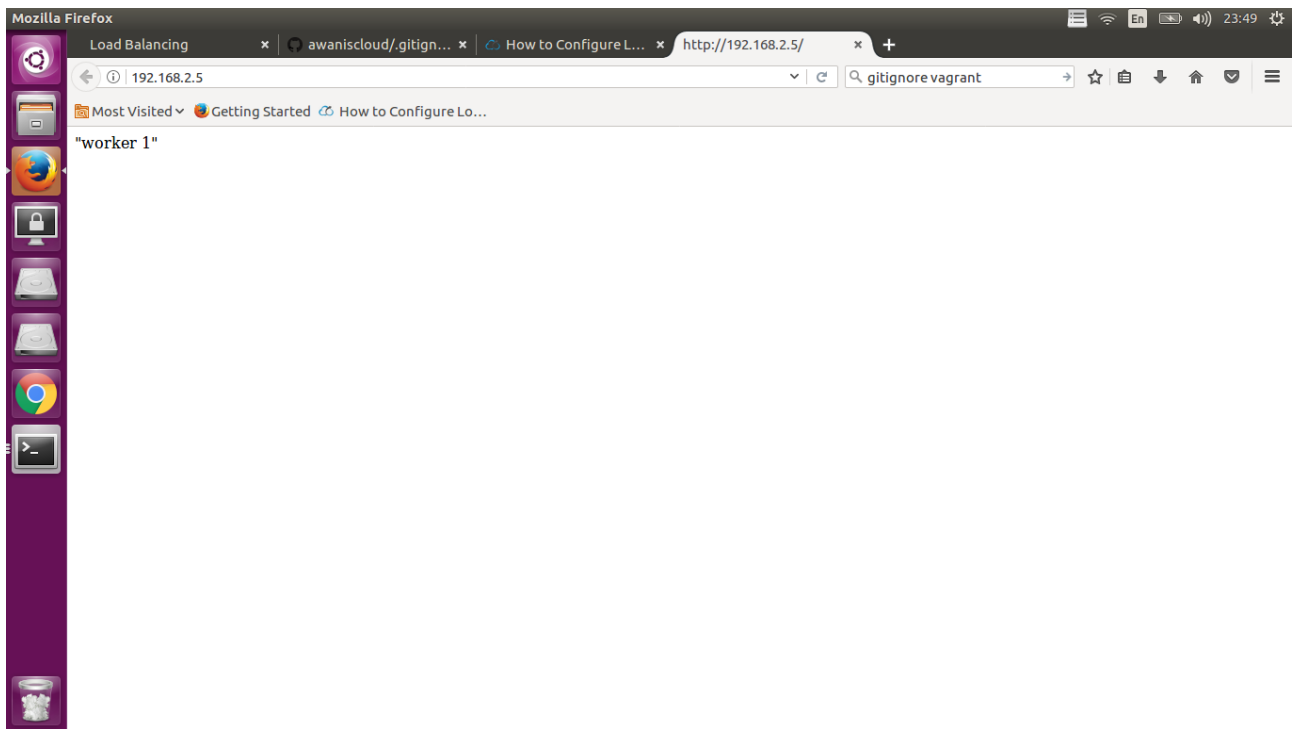
```

Pengujian 1:

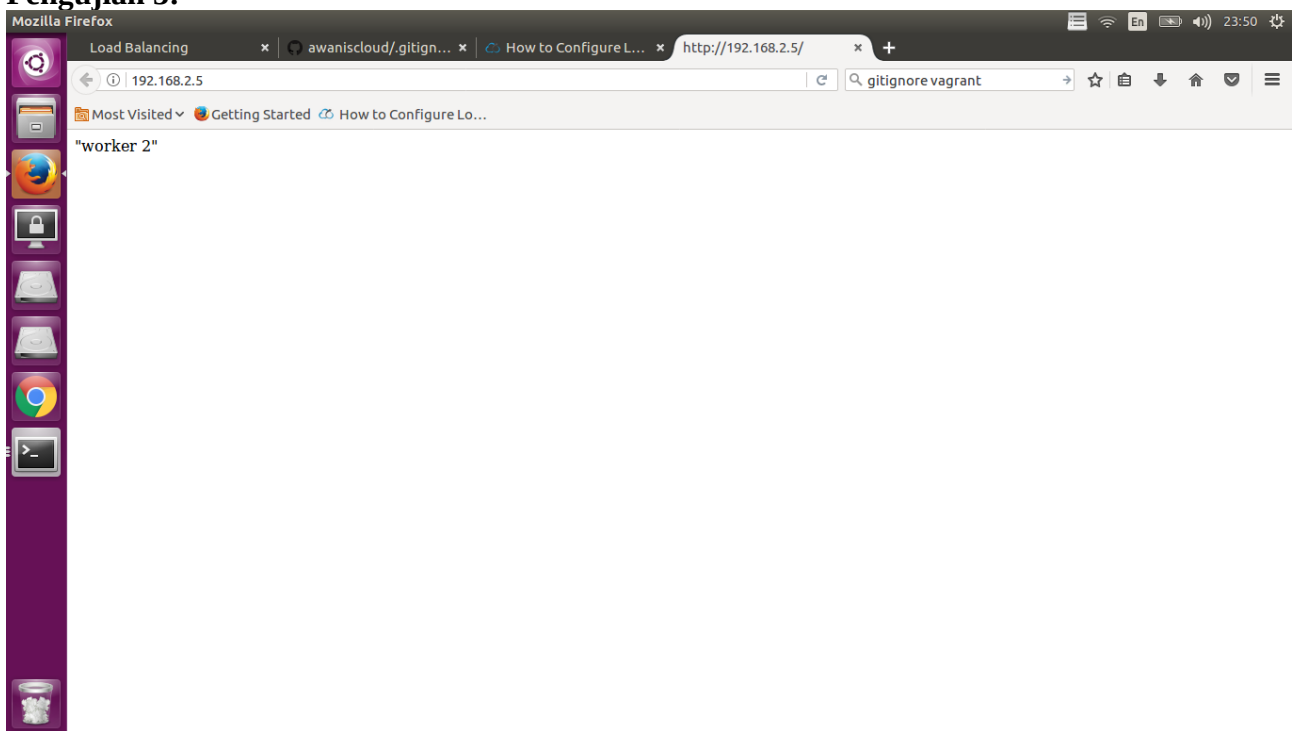




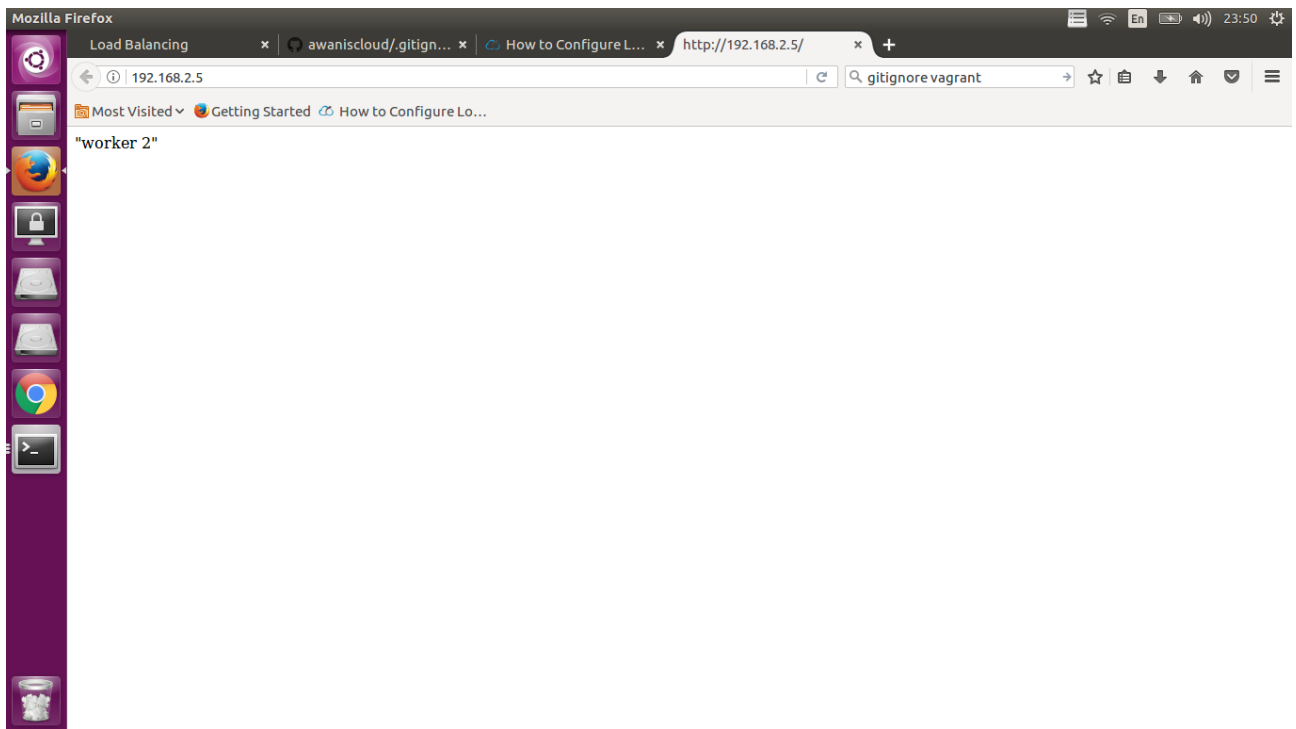
## Pengujian 2:



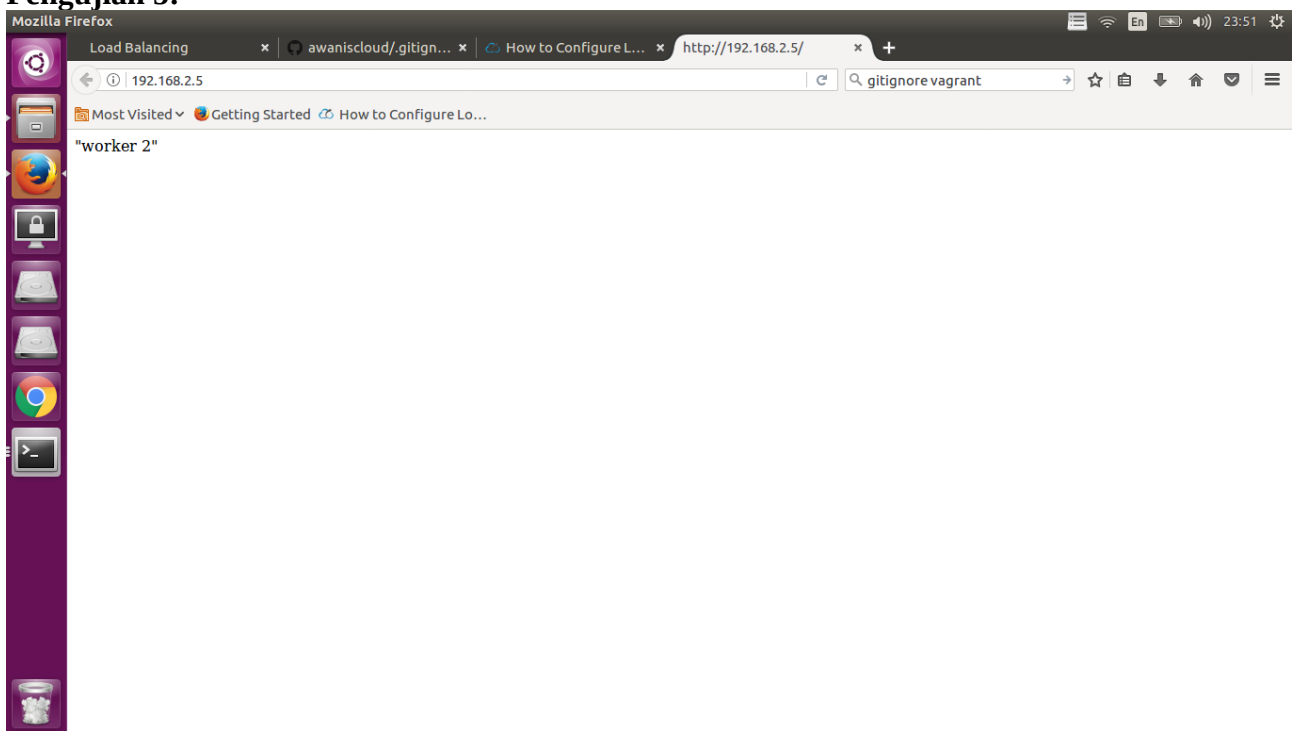
### Pengujian 3:



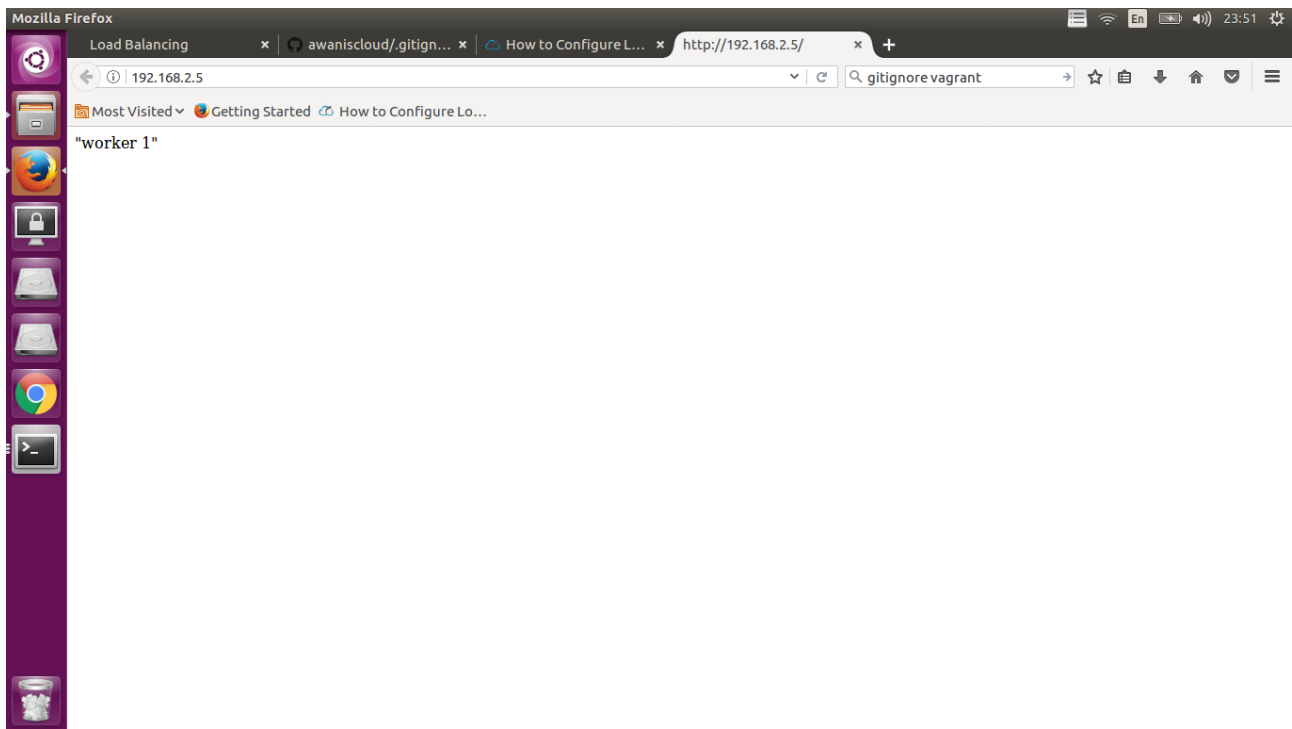
### Pengujian 4:



## Pengujian 5:



## Pengujian 6:



### Kesimpulan:

Least-connected memilih server berdasarkan tingkat kepadatan transaksi yang ditangani oleh sebuah server. Least-connected selalu memilih server yang tidak sibuk. Dari percobaan di atas, dapat kita lihat bahwa worker2 dominan dalam menangani server. Karena kami selalu menutup koneksi setelah mengirimkan request sehingga secara otomatis server selalu kosong. Dengan demikian server seharusnya selalu mengakses worker yang sama. Munculnya worker 1 jarang karena terjadi apabila worker2 belum benar-benar terputus dengan request sebelumnya.

### 3. IP Hash

Buka file `/etc/nginx/sites-available/default`, lalu edit file menjadi seperti di bawah:

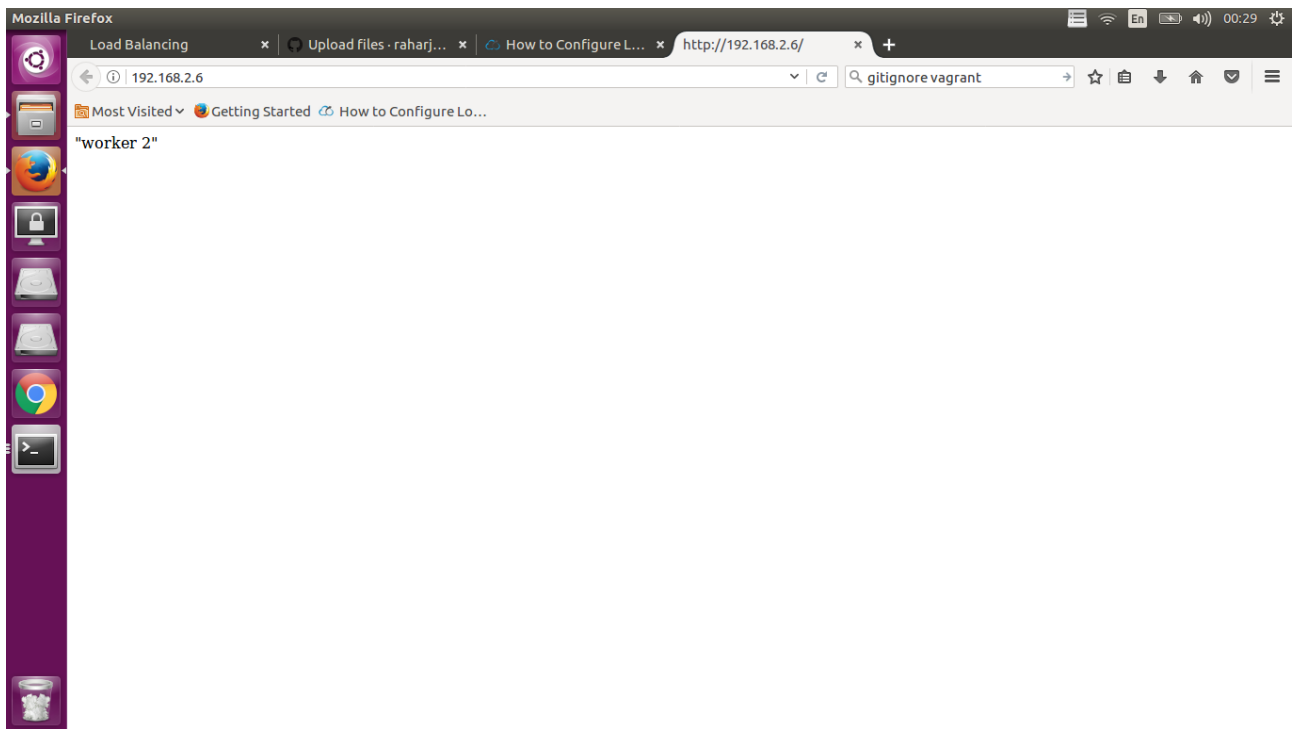
```
vagrant@vagrant-ubuntu-trusty-64: /etc/nginx/sites-available
GNU nano 2.2.6      File: default      Modified

# http://wiki.nginx.org/QuickStart
# http://wiki.nginx.org/Configuration
#
# Generally, you will want to move this file somewhere, and start with a clean
# file but keep this around for reference. Or just disable in sites-enabled.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

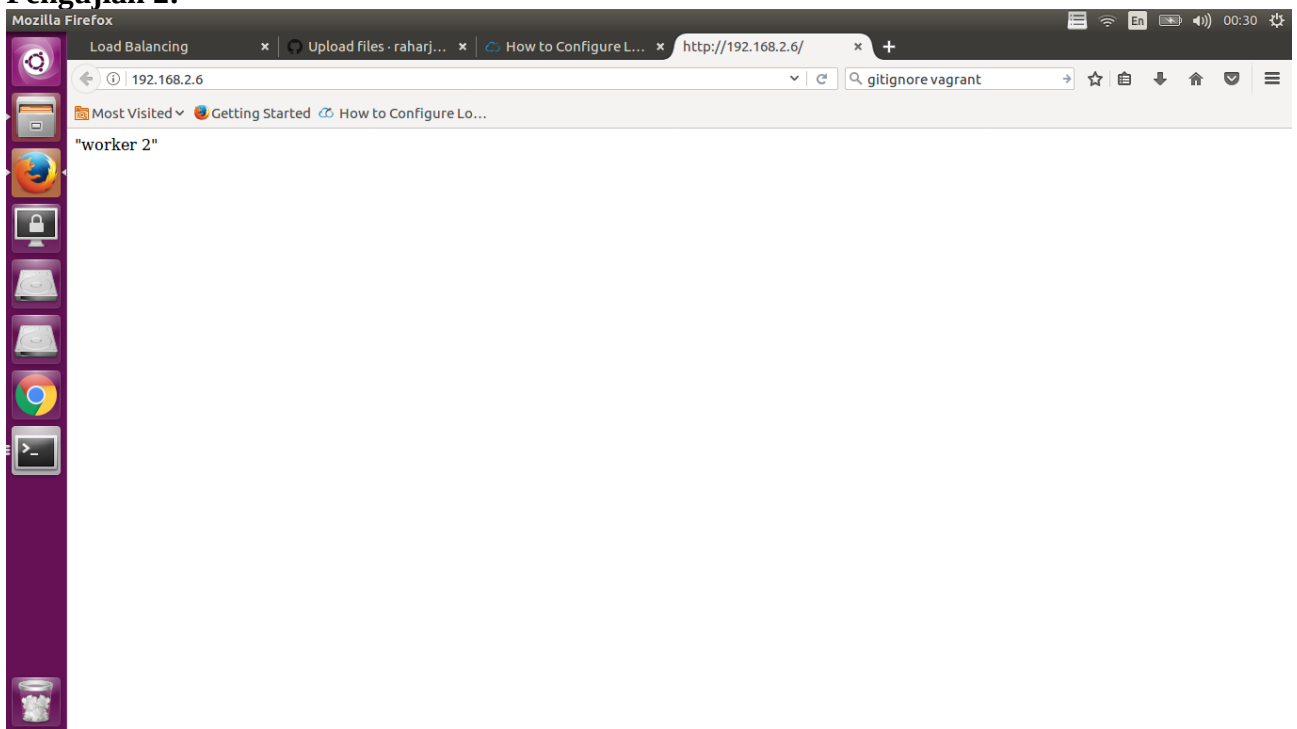
upstream worker{
    ip_hash;
    server 192.168.2.3:80;
    server 192.168.2.4:80;
}

server {
    listen 80 default_server;
    listen [::]:80 default_server ipv6only=on;
```

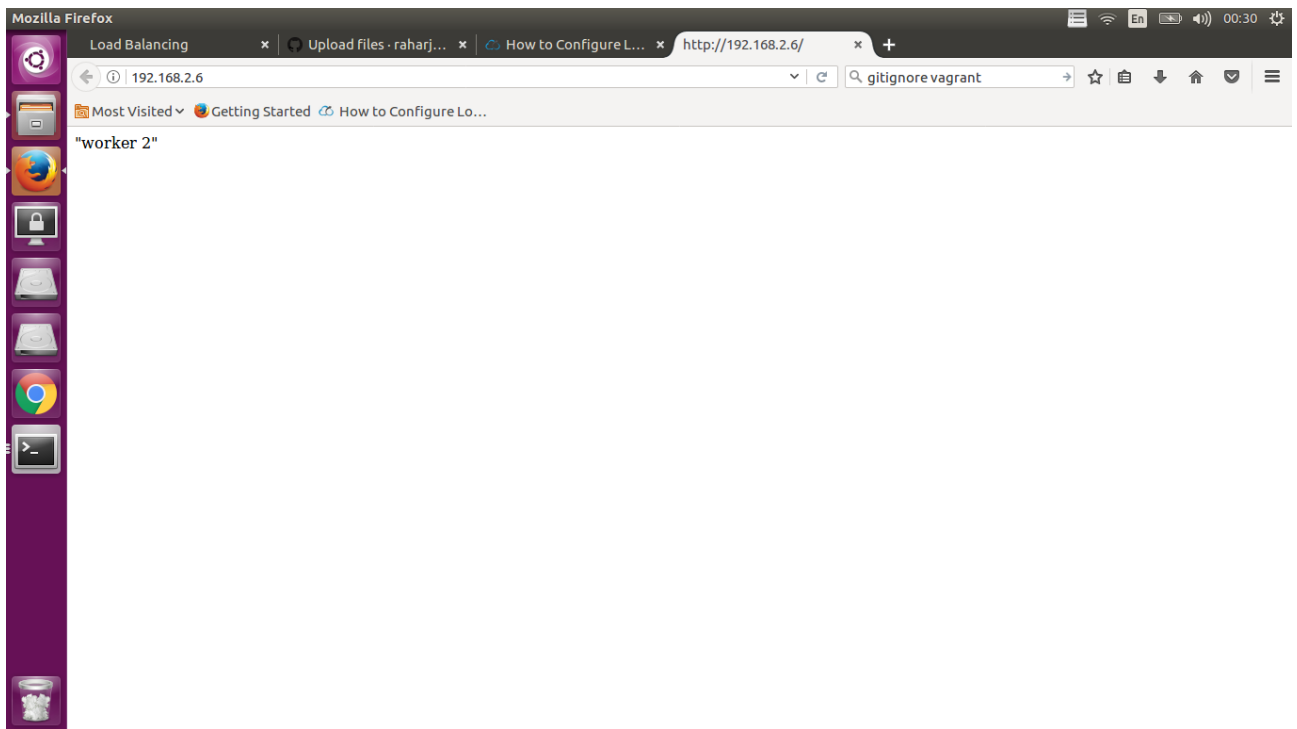
### Pengujian 1:



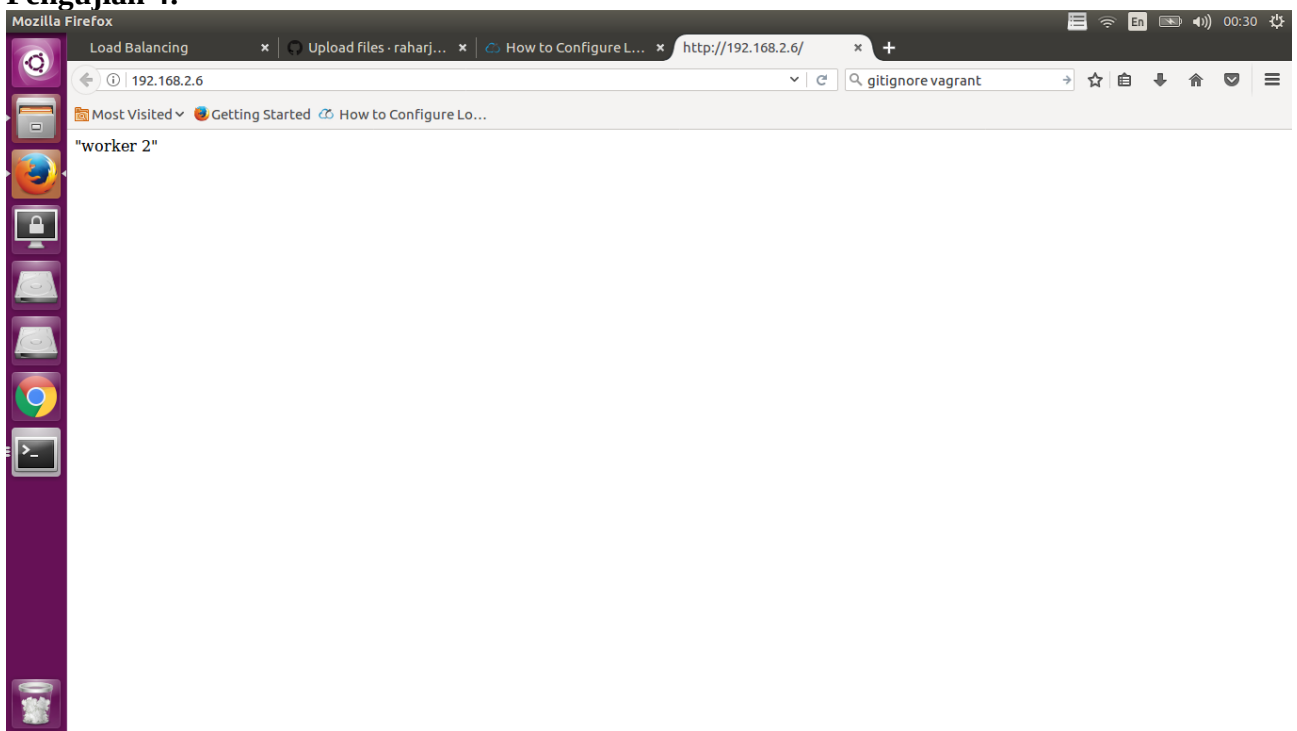
## Pengujian 2:



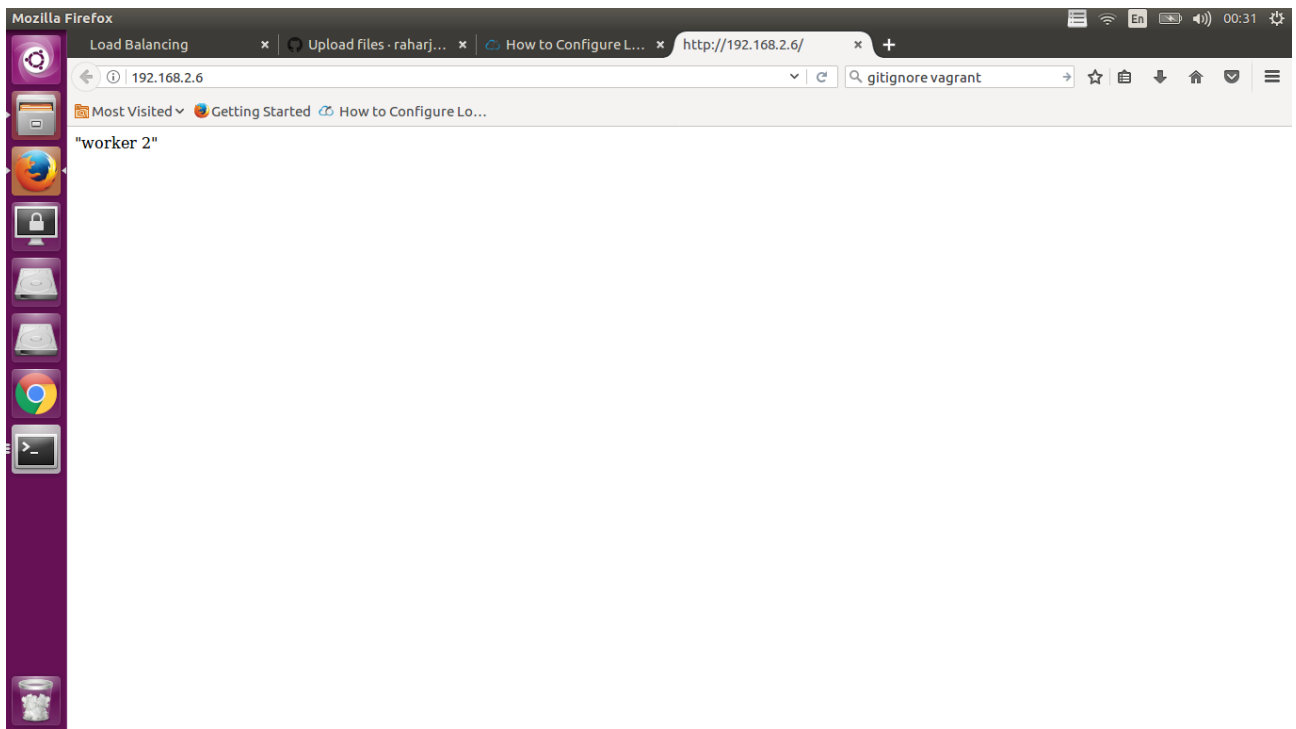
## Pengujian 3:



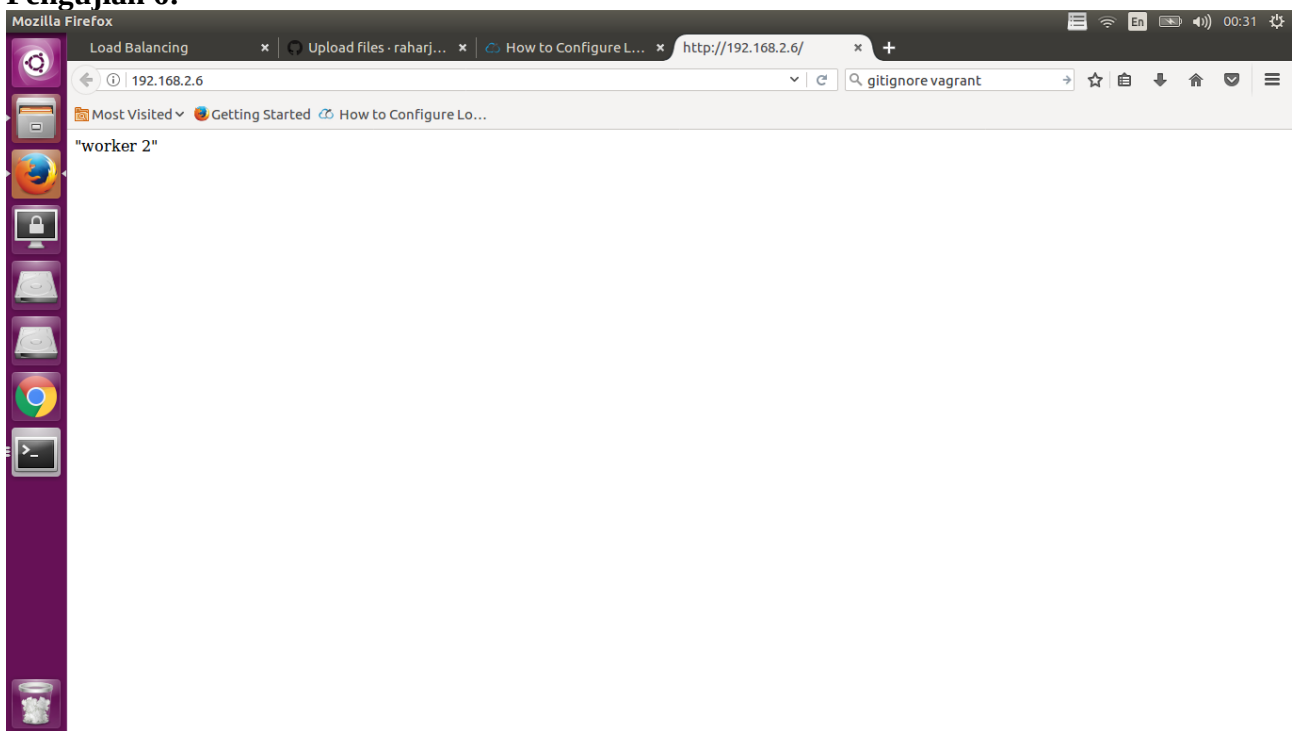
#### Pengujian 4:



#### Pengujian 5:



### Pengujian 6:



### Kesimpulan:

IP hash menangani request berdasarkan IP. Satu IP akan selalu dilayani oleh server yang sama. Dapat kita lihat contoh diatas.