

Load Balancing dengan Nginx di Ubuntu 18.04

A.A. Gede Yudhi Paramartha

Pendahuluan

Mungkin anda pernah berpikir, berapa banyak orang akan mengunjungi situs atau aplikasi web anda? Apakah server anda cukup untuk menangani seluruh pengguna tersebut? Berapa banyak pengguna yang akan mengakses situs dalam waktu yang sama? Itulah beberapa pertanyaan yang harus dijawab pada saat anda mengembangkan sebuah aplikasi web. Kita harus berasumsi bahwa aplikasi web yang kita bangun akan dikunjungi oleh berkali-kali lipat pengguna di masa depan. Sebagai contoh, pada tahun pertama Facebook diluncurkan, tidak lebih dari 1 juta pengguna terdaftar pada situs tersebut. Namun, dalam waktu kurang lebih 10 tahun, jumlah pengguna Facebook mencapai lebih dari 1 Miliar orang. Berarti jumlah pengguna bertambah seribu kali lipat dalam waktu 10 tahun.

Dengan asumsi itu, apa yang harus kita lakukan? Mengupgrade Server dengan CPU yang lebih cepat? Menambah RAM hingga mencapai batas maksimal yang dapat diinstal di server kita? Hal tersebut tidaklah cukup, karena jumlah CPU dan RAM yang dapat diinstall dalam satu server tentu saja terbatas, namun asumsi jumlah pengguna mungkin saja tidak terbatas. Maka dari itu, salah satu solusi yang mungkin dapat dilakukan adalah menambah server yang tersedia. Satu buah server, mungkin dapat menangani 10.000 pengguna. Dengan asumsi menggunakan server yang sama, kita dapat menghitung secara kasar bahwa dengan menambah satu server, maka jumlah pengguna yang dapat dilayani adalah sebanyak 20.000 orang, menambah dua buah server maka dapat melayani 30.000 orang, dan begitu seterusnya.

Lalu bagaimana caranya? Apakah cukup hanya menambah server saja? Tentu saja tidak! Kita harus membagi beban pengguna ke setiap server yang tersedia. Itulah yang dapat ditangani oleh Load Balancer. Load Balancer adalah sebuah teknologi yang mendistribusikan beban (jaringan, aplikasi, ataupun pengguna) ke beberapa titik pemrosesan, dalam hal ini adalah server. Dengan menggunakan load balancer, kita dapat membagi beban pengguna ke beberapa server yang tersedia. Keuntungan menggunakan teknologi ini selain membagi beban pengguna adalah adanya redundansi dan *fault tolerance* pada aplikasi web yang kita bangun. Jika satu server mati, maka server lain masih dapat menangani pengguna, sehingga ketersediaan layanan (service availability) dapat dijaga.

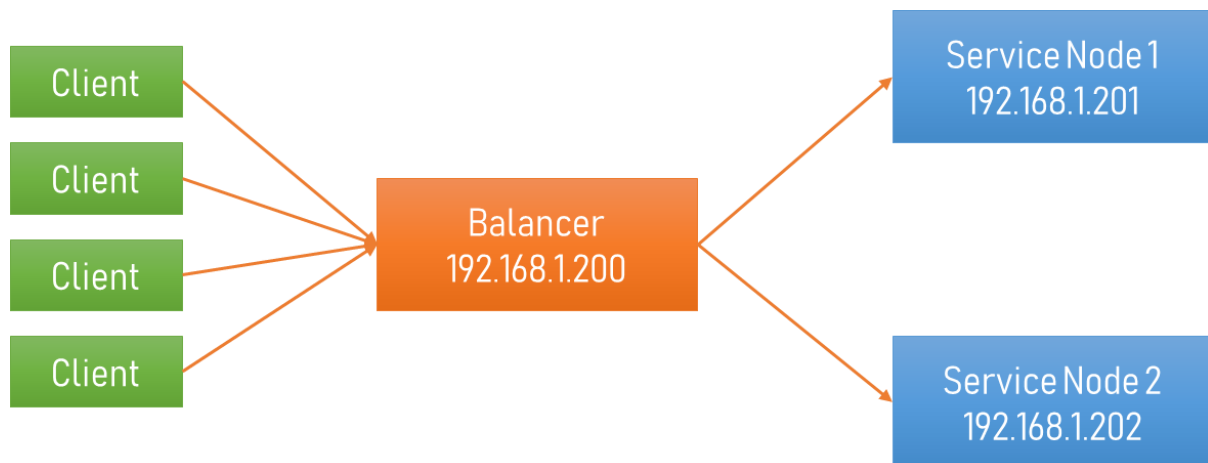
Pada tutorial ini, kita akan membangun sebuah load balancer sederhana dengan Nginx pada sistem operasi Ubuntu Server 18.04 LTS.

Persiapan

Kita membutuhkan minimal 3 buah server (komputer) yang terhubung ke jaringan yang sama. Satu server berperan sebagai *load balancer*, sedangkan dua server lainnya berperan sebagai *service node* atau tempat aplikasi web anda tersedia. Anda dapat menggunakan virtual machine (seperti VirtualBox) untuk melakukan simulasi load balancing pada tutorial ini. Setiap server diinstall dengan sistem operasi Ubuntu Server 18.04 LTS. Berikut ini adalah konfigurasi IP address pada setiap server.

- Load Balancer : 192.168.1.200/24
- Service Node 1 : 192.168.1.201/24
- Service Node 2 : 192.168.1.202/24
- Komputer client : 192.168.1.100/24 (atau IP address bebas asalkan masih ada pada jaringan yang sama dengan Load balancer)

Berikut ini adalah diagram jaringan yang dibutuhkan:



Instalasi Nginx

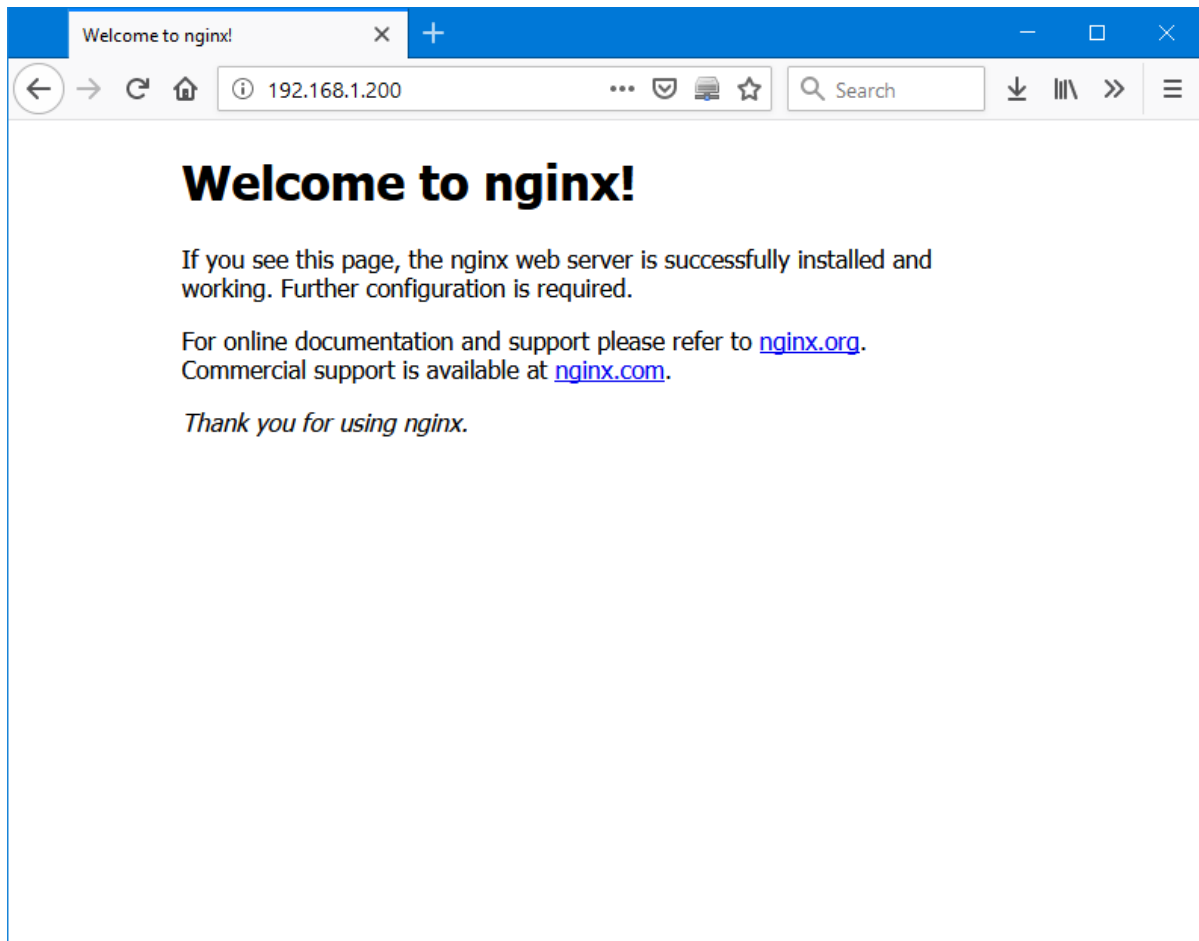
Install Nginx di setiap server yang tersedia dengan menggunakan perintah berikut ini:

- `sudo apt update`
- `sudo apt install nginx`

Jalankan Nginx dengan perintah berikut ini:

- `sudo service nginx start`

Cek bahwa Nginx telah berjalan dengan benar dengan mengakses IP-address server di client melalui web browser.



Catatan: Ulangi tahapan ini di setiap server.

Buat Aplikasi Web di Service Node

Selanjutnya anda membuat aplikasi web pada kedua Service Node. Anda dapat membuat aplikasi web apa saja yang anda inginkan dan diinstall di setiap service node. Jika anda ingin menginstall aplikasi berbasis PHP, anda harus menginstall PHP terlebih dahulu. Namun pada tutorial ini, kita hanya akan membuat sebuah file HTML yang akan melayani pengguna. Berikut ini adalah langkah-langkahnya:

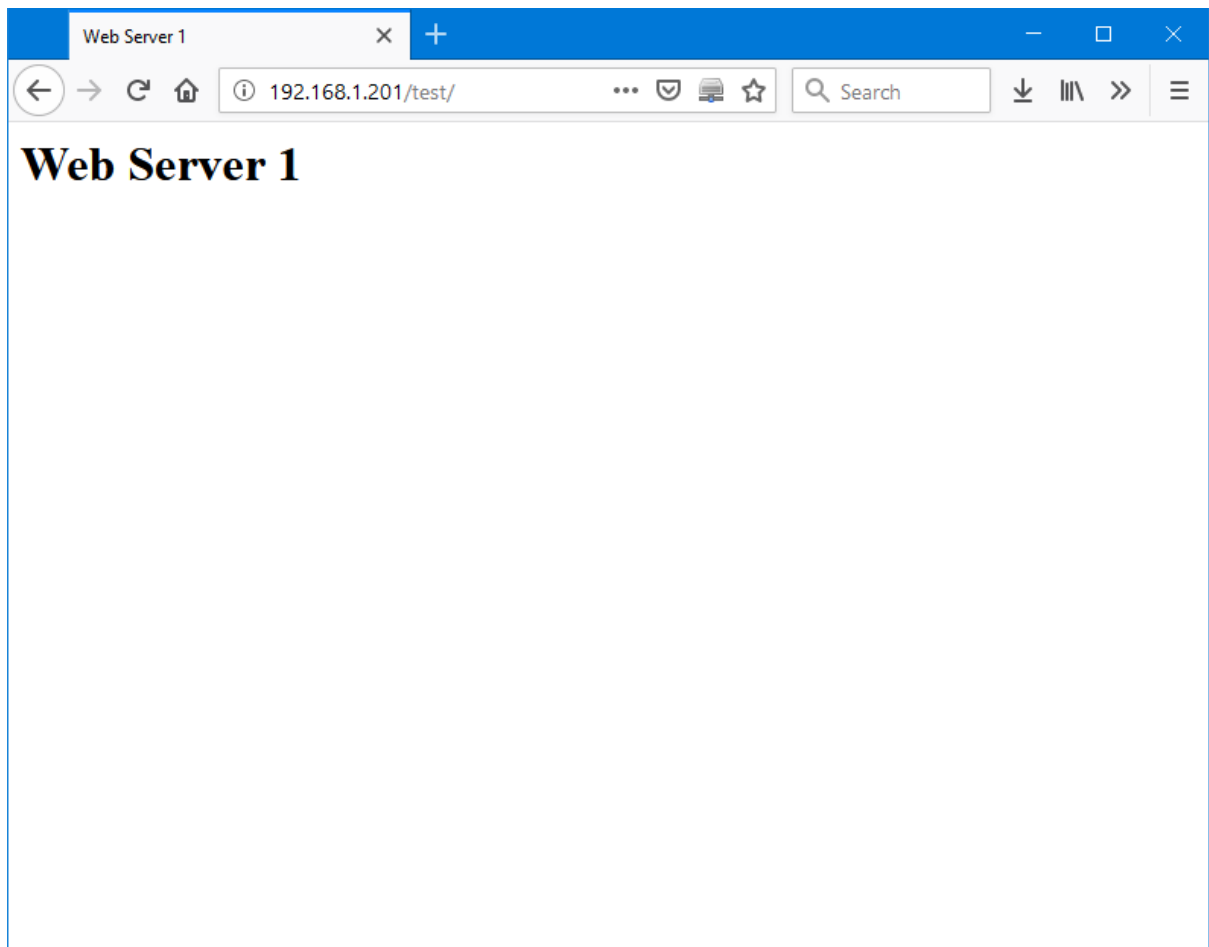
Buat file HTML di Service Node 1

- `sudo nano /var/www/html/test/index.html`

Tulis kode HTML berikut ini:

```
<html>
  <head>
    <title>
      Web Server 1
    </title>
  </head>
  <body>
    Web Server 1
  </body>
</html>
```

Jalankan aplikasi melalui web browser di client:



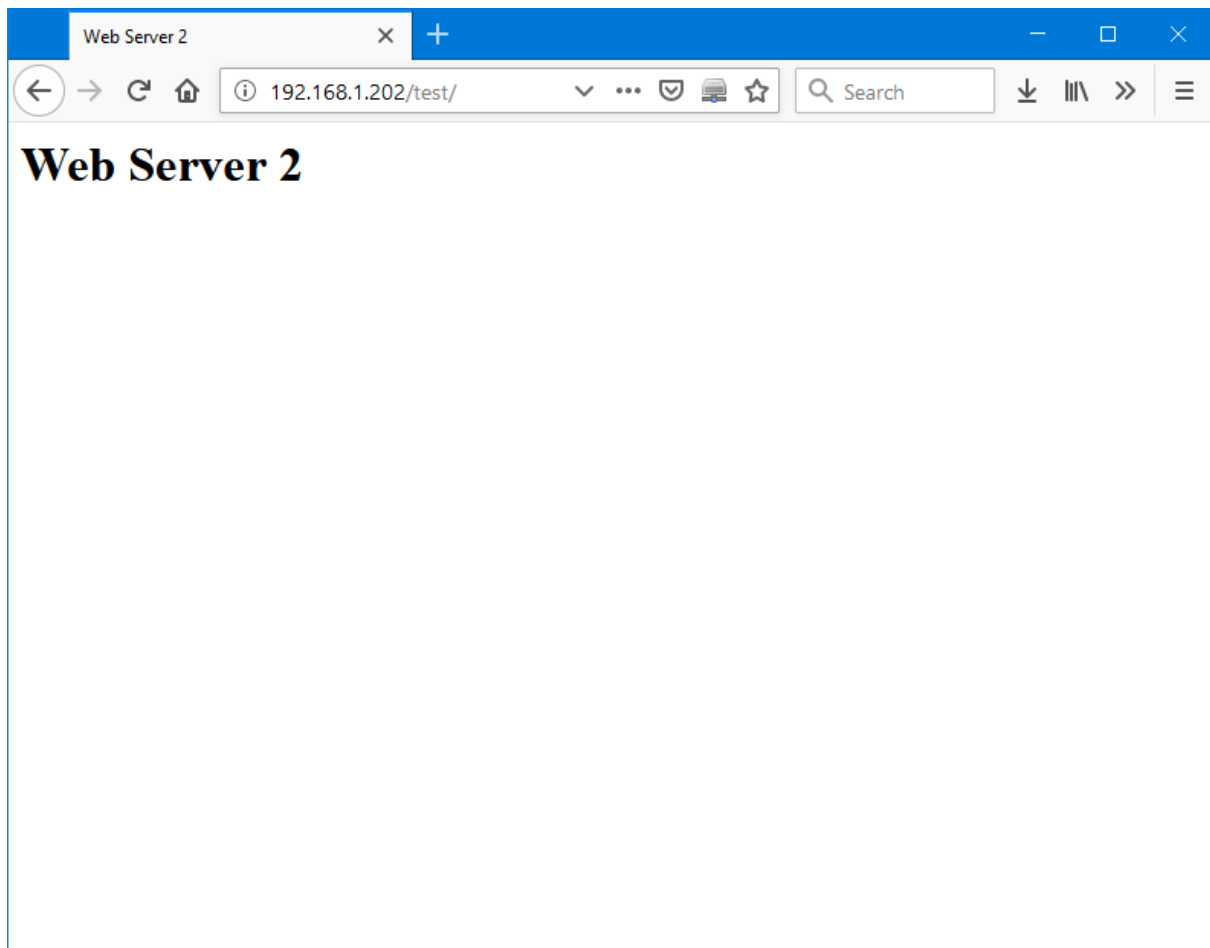
Lakukan hal yang sama pada Service Node 2. Buat file HTML di Service Node 2:

- `sudo nano /var/www/html/test/index.html`

Tulis kode HTML berikut ini:

```
<html>
  <head>
    <title>
      Web Server 2
    </title>
  </head>
  <body>
    Web Server 2
  </body>
</html>
```

Jalankan aplikasi melalui web browser di client:



Perhatikan bahwa kedua Service Node memiliki keluaran yang sedikit berbeda. Service Node 1 memunculkan kalimat “Web Server 1” di browser, sedangkan Service Node 2 memunculkan kalimat “Web Server 2”. Dalam implementasinya, seharusnya kedua Service Node melayani aplikasi yang identik. Namun pada tutorial ini, hal ini dilakukan untuk melihat perbedaan antara 1 server dengan server yang lain.

Konfigurasi Load Balancer

Jika aplikasi web pada Service Node 1 dan Service Node 2 sudah berjalan dengan benar, maka selanjutnya kita akan mengkonfigurasi load balancer. Pada server Load Balancer, buat sebuah file konfigurasi bernama “test” di /etc/nginx/sites-available:

- `sudo nano /etc/nginx/sites-available/test`

isi file tersebut dengan kode dibawah ini:

```
upstream test {
    server 192.168.1.201;
    server 192.168.1.202;
}

server {
    listen 80;

    server_name 192.168.1.200;

    location /test {
        proxy_pass http://test;
    }
}
```

Jika anda lihat pada kode diatas, terdapat beberapa perintah (*directives*) yang ditulis. Berikut penjelasannya:

- Directive **upstream** dengan nama “test” adalah sebuah perintah dimana request akan dialihkan ke suatu server. Pada contoh di atas, setiap request akan dialihkan ke server dengan IP address 192.168.1.201 atau 192.168.1.202.
- Directive **server** berarti Nginx akan menunggu request dari port 80, pada server 192.168.1.200 (IP address server Load Balancer). Jika anda telah menggunakan DNS server atau telah menyeting hostname, anda dapat menyesuakannya.
- Directive **location** adalah lokasi aplikasi web tersedia. Pada tutorial ini, aplikasi web anda tersedia di folder “/etc/www/html/test”, maka dari itu directive location **/test** yang digunakan.
- Directive **proxy_pass** ditulis di dalam directive **location**, yang menandakan ke mana request dialihkan. `http://test` mengarah pada directive **upstream test** yang sudah dijelaskan sebelumnya.

Dengan directive di atas, secara umum dapat dijelaskan sebagai berikut:

Setiap request yang mengarah pada URI `http://192.168.1.200/test` dengan port 80 akan dialihkan ke `http://192.168.1.201/test` atau ke `http://192.168.1.202/test`.

Aktifkan Konfigurasi di Nginx

Untuk mengaktifkan konfigurasi yang telah dibuat, buat symlink ke sites-enabled dengan perintah berikut ini:

- `sudo ln -s /etc/nginx/sites-available/test /etc/nginx/sites-enabled/`

Selanjutnya restart nginx di Load Balancer untuk melihat perubahannya:

- `sudo service nginx restart`

Uji Coba Balancer

Untuk menguji apakah balancer berjalan dengan baik, buka <http://192.168.1.200/test> melalui web browser di client. Jika berhasil, maka setiap kali anda me-refresh web browser anda akan diarahkan ke salah satu Service Node yang tersedia seperti animasi di bawah ini. Anda dapat menguji balancer melalui beberapa client pada jaringan yang sama. Setiap client akan diarahkan ke salah satu Server Node dengan metode Round Robin (bergantian).

Catatan: Lihat animasi uji coba di <http://static.paramartha.co/assets/balancer.gif>

