Journal of Information Technology and Computer Science (INTECOMS)
Volume 5 Nomor 2, Desember 2022
e-ISSN : 2614-1574
p-ISSN : 2621-3249

RANCANG BANGUN SISTEM LOAD BALANCER DENGAN LAYANAN CLOUD AMAZONE WEB SERVICES

DESIGN OF APPLICATION LOAD BALANCER WITH CLOUD COMPUTING SERVICES - AMAZON WEB SERVICES

Rio Pratama[1], Akhyar Lubis[2], Sri Wahyuni[3]
Universitas Pembangunan Panca Budi
riopratam10@gmail.com

ABSTRACT

The server is a vital component in information service providers. This is important in increasing work productivity, efficiency, and services, which are the company's business triggers. The server becomes so crucial because it provides access requests to clients. There are many requests/requests to the server, so traffic increases due to high traffic impacting low server performance, which can even cause down servers to be inaccessible. The biggest concern of cloud computing is overloading the system for any individual, group, or organization. One form of maintaining service availability performance from a server is with a web load balancer system by balancing the load on one server and sharing it with other servers on the same cluster node. The purpose of this study is to implement a load balancer application with auto-scaling groups on the Amazon web service cloud computing service. The method used is a prototype with five stages of testing. From the experiments conducted on 3 EC2 servers with the Nginx webserver installed, requests are still available even though some servers are down. At the same time, group auto-scaling will add a new instance to meet the desired capacity. This maintains instance count even if the instance state is not healthy.

Keywoards*: Application Load Balancer, aws, Target group*

ABSTRAK

Server merupakan komponen vital dalam penyedia layanan informasi. Hal ini menjadi penting dalam peningkatan produktifitas kerja, efisiensi dan juga layanan yang menjadi bisnis triger perusahaan. Server menjadi begitu penting karena menyediakan akses permintaan ke client. Banyak nya permintaan/request ke server sehingga traffik meningkat akibat traffik yang tinggi berdampak kepada performa server rendah bahakan menyebabkan server down tidak dapat diakses. Kekhawatiran terbesar komputasi awan adalah kelebihan beban sistem untuk setiap individu, kelompok ataupun organisasi. Salah satu bentuk menjaga performa ketersediaan layanan dari sebuah server adalah dengan sistem load balancer web dengan menyeimbangkan beban pada satu server dan dibagi ke server lainnya pada node cluster yang sama. Tujuan dalam penelitian ini adalah melakukan impelemantasi aplikasi load balancer dengan auto scalling group pada layanan cloud computing amazone web service. Metode yang digunakan adalah prototype dengan lima tahapan pengujian. Dari hasil percobaan yang dilakukan pada 3 buah server EC2 yang terinstall webserver nginx, request permintaan masih tetap tersedia walau beberapa server mengalami down sementara group auto scaling akan menambahkan sebuah instance baru untuk memenuhi kapasitas yang diinginkan. Hal ini mempertahankan jumlah instance walau keadaan instance tidak sehat.

Kata Kunci*: aplikasi load balancer, aws, Target group*

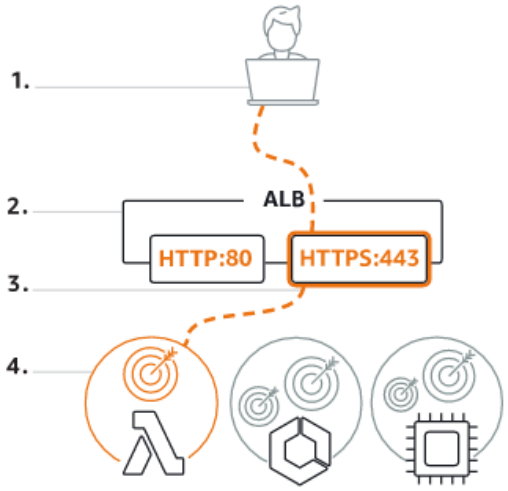
PENDAHULUAN

Server merupakan komponen vital dalam penyedia layanan informasi. Hal ini menjadi penting dalam peningkatan produktifitas kerja, efisiensi dan juga layanan yang menjadi bisnis triger perusahaan (Khare Shivangi and Chourasia, 2022). Server sebagai penyedia layanan diperlukan kehandalan dalam memberikan layanan tanpa mengalami masalah. Server menjadi begitu penting karena menyediakan akses permintaan ke client. Banyak nya permintaan/request ke server sehingga traffik meningkat akibat

kebanjiran pengunjung. Hal ini dapat menyebabkan performa server menjadi rendah dan dapat menyebabkan server down tidak dapat diakses. Peletakan server dapat bersifat on-premise dengan on cloud. Banyak organisasi membangun aplikasi menggunakan layanan cloud. Kekhawatiran terbesar komputasi awan adalah kelebihan beban sistem untuk setiap individu, kelompok ataupun organisasi. Salah satu bentuk menjaga performa ketersediaan layanan dari sebuah server adalah dengan sistem load balancer web dengan menyeimbangkan beban pada satu server dan dibagi ke server lainnya pada node cluster yang sama. Load balancer ini menjadi solusi untuk mengatasi server yang padam (Rafli, 2022). Untuk menjaga server selalu tersedia ketika diakses dan realiable adalah konsep high availabiliy. Kemampuan server diharapkan mampu membagi beban dari request yang diterima dari pengguna dapat disebar dan tidak berfokus pada satu server saja.

Load balancing merupakan pendistribusian beban kerja pada dua atau lebih suatu koneksi jaringan secara seimbang agar pekerjaan dapat berjalan optimal dan tidak overload pada jalur koneksi (Mikola & Nurcahyo, 2022). Load balancer melakukan pendistribusian lalulintas masuk pada aplikasi dibeberapa target didalam satu atau lebih *Availability Zones*. Tujuan dari load balancer adalah mendistribusikan beban kerja ke beberapa sumber daya komputasi untuk meningkatkan ketersediaan dan toleransi kesalahan aplikasi

Terdapat empat jenis load balancer yaitu *application load balancer*, *network load balancer*, *gateway load balancer* dan *classic load balancer*. Application load balancer bekerja pada lapisan aplikasi pada model OSI layer yaitu dengan dengan lalu lintas http dan https. Application load balancer memberikan routing dan tingkat visibiltas pada arsitektur aplikasi. Target group digunakan untuk permintaan route untuk meregister target. *Network load balancer* beroperasi pada layer koneksi (layar 4), koneksi routing ke target. Network load balancer dalam mendukung performa, sentralisasi, sertifikasi pengembang, dan mendukung UDP dan pengalamatan IP statik. Classic load balancer mendukung load balancing application menggunakan http, https, tcp dan ssl.



Gambar 1. Cara kerja Aplikasi load balancer

1. Klient melakukan permintaan request ke server aplikasi
2. Listerner pada aplikasi load balancer menerima permintaan disesuaikan dengan protokol dan port yang telah dikonfigurasi.
3. Linterner menerima mengevaluasi permintaan masuk berdasarkan aturan yang telah didefenisikan. Jika berlaku akan merutekan permintaan ke group yang sesuai.
4. Target health dalam satu atau beberapa group menerima trafik berdasarkan algoritma load balancing.

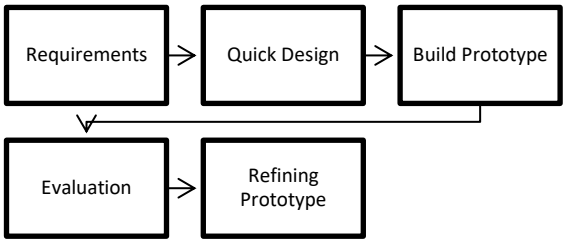
(Ramsari & Ginanjar, 2022) melakukan penelitian terkait infrastruktur berbasis cloud menggunakan GCP dengan tujuan tingkat reliabilitas dan availability tinggi dengan menerapkan *autohealing, auto scaling, multiple zones, load balancing dan fileover* sehingga dapat menjaga realibiltas dan ketersediaan dari infrastrukur yang dibangun. Penelitian (Wijaya & Franklyn, 2021) menerapkan sistem stock pada haultime thrift shop

menggunakan arsitektur GCP. Dengan adanya sistem infrastruktur yang dibangun dapat meminimalisir terjadinya downtime dengan menerapkan backup dan juga high availbility. (Sholihah et al., 2022) melakukan penelitian pada media pembelajaran memanfaatkan load balancing dengan menggunakan algoritma round robin dan least connected. Hasil dari load balancing ini dapat digunakan untuk modul pembelajaran berbasis android. (Ramadhan & Wulandari, 2022) melakukan penelitian untuk menerapkan *high availability* dengan menggunakan *least connected* dan *load balancer* pada infrastruktur server LMS. Pengujian yang dilakukan ketika salah satu server terjadi kesalahan maka server web yang lain mengambil request sehingga dari sisi pengunjung sistem masih dapat berjalan.

Pada penelitian ini, penulis melakukan eksperiment aplikasi load balancer menggunakan amazon web services dengan auto scalling group. Penggunaan tiga buah server yang diletakkan dalam regional yang sama dengan menerapkan auto scalling group. Dengan adanya aplikasi load balancer ini diharapkan memberikan manfaat terkait performa server sehingga ketika mengalami down atau trafik yang berlebih dapat berpindah kepada layanan server lainnya. Dalam penelitian ini, penulis menggunakan layanan AWS dimana inti dari operasi cloud adalaha virtualisasi (A. Mishra, 2017). AWS memiliki lebih dari 200 layanan unggulan yang ditawarkan secara lengkap dari pusat data secara global (Posey, 2021). Beberapa layanan yang digunakan diantaranya Virtual Private Cloud (VPC) dan Amazon EC2. Virtual Private Cloud ini sangat mirip dengan jaringan tradisional yang nantinya dioperasikan pada pusat data sendiri (S. Mishra et al., 2022).

METODE

Metode penelitian yang penulis gunakan menggunakan metode prototype yang terdiri dari 5 tahapan diantaranya requirements, quick design, build prototype, evalution dan refining protoype.



Gambar 2. Metode Penelitian

Tahap Requirements

Kebutuhan dalam membangun load balancer dikumpukan, jenis layanan yang digunakan dan seberapa banyak penggunaan layanan server virtual yang digunakan. Pemilihan jenis load balancer disesuaikan dengan kebutuhan.

Quick Design

Desain Topologi infrastruktur jaringan cloud dilakukan berdasarkan analisa kebutuhan yang sudah didefenisikan sebelumnya. Hasil rancangan ini menjadi model dalam penerapan selanjutnya.

Build Prototype

Tahapan ini adalah mengimplementasikan layanan layanan yang digunakan termasuk juga pengaturan konfigurasi pengalamatan IP berdasarkan design infrastruktur diawal.

Evaluation

Evaluasi dilakukan dengan melakukan pengujian. Pengujian untuk menguji apakah aplikasi load balancer bekerja dengan baik dan optimal ketika ada beberapa server yang mengalami gangguan. Jika prototype yang dibangun ini tidak sesuai maka dilakukan perbaikan prototype.

Refining Protoype

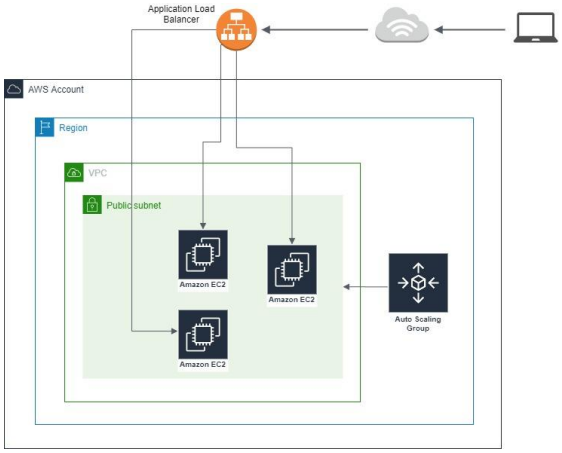
Pada tahap ini melakukan perbaikan dari hasil evaluasi yang telah dilakukan.

HASIL DAN PEMBAHASAN

Dalam membangun aplikasi load balancer dengan layanan amazone web service penulis melakukan dua tahapan yaitu analisa perancangan dan implementasi.

Perancangan Topologi Infrastruktur

Setelah kebutuhan telah ditentukan, tahapan selanjutnya adalah merancang infrastruktur yang akan digunakan. Setiap layanan disusun dalam bentuk topologi yang digambarkan pada gambar satu dibawah ini.



Gambar 3. Topologi Infrastruktur Jaringan

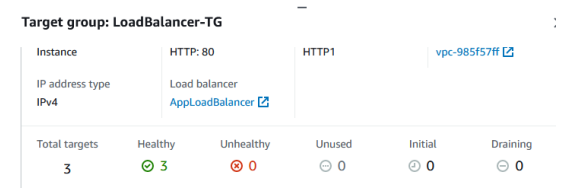
Skenario aplikasi load balancer ini dimulai dari permintaan akses ke server dari klient. Application load balancer akan menerima permintaan dari client dengan protokol yang sudah dikonfigurasi. Setiap instance dibangun webserver dengan nginx dalam satu regional dan satu publik subnet.

Implementasi Aplikasi Load Balancer

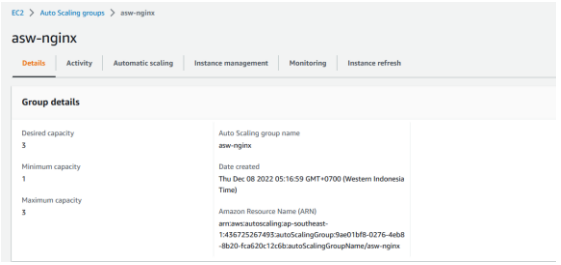
Aplikasi load balancer dibangun memanfaatkan layanan cloud aws. Penggunaan Ipversi 4 yang digunakan untuk internal load balancer. Pada pengaturan load balancer menggunakan VPC dalam satu regional. Setiap server EC2 dengan spesifikasi t2. micro, Amazon Linux 2 Kernel 5.10 AMI, 8 GiB dengan sistem Nginx.

Tabel 1. Pemilihan Zone

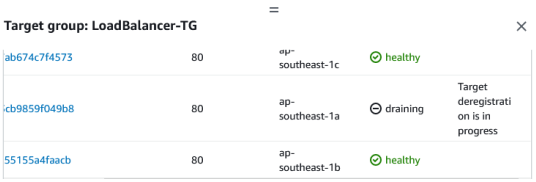
No	Availability Zone	Subnet
1	ap-southeast-1a (apse1-az2)	172.31.32.0/20
2	ap-southeast-1b (apse1-az1)	172.31.16.0/20
3	ap-southeast-1c (apse1-az3)	172.31.0.0/20



Gambar 4. Target group loadbalancer

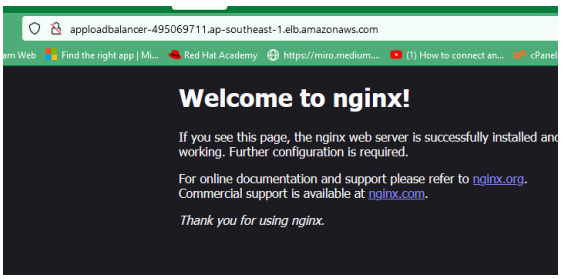


Gambar 5. Autoscaling Group Nginx

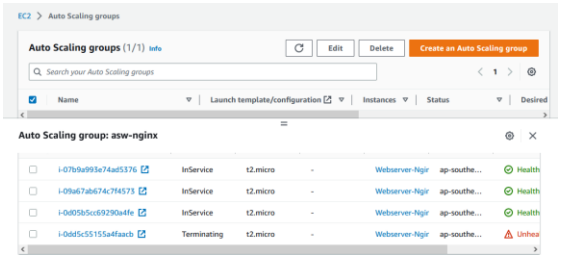


Gambar 6. Targe Group draining

Pengujian dengan memadamkan server yang berada di availability zone ap-southeast-1a, server nginx tetap dapat berjalan.



Gambar 7. Testing Webserver



Gambar 8. Auto Scaling Group

Kondisi dengan memadamkan server pada ec2, group auto scaling akan menambahkan sebuah instance baru untuk memenuhi kapasitas yang diinginkan. Hal ini mempertahankan jumlah instance walau keadaan instance tidak sehat.

SIMPULAN

Dari hasil pengujian yang dilakukan pada tiga buah server yang terhubung ke aplikasi load balancing dengan mendaftarkan sistem target group berdasarkan instance berhasil dilakukan. Load balancer mendistribusikan ke seluruh target dalam group target. Aplikasi pada nginx masih dapat berjalan walau beberapa server mengalami down.

DAFTAR PUSTAKA

Khare Shivangiand Chourasia, U. and D. A. J. (2022). Load Balancing in Cloud Computing. In G. and M. S. and H. T. Kumar Amitand Ghinea (Ed.), *Proceedings of the International Conference on Cognitive and Intelligent Computing* (pp. 601–608). Springer Nature Singapore.

Mikola, A., & Nurcahyo, A. C. (2022). Analisis Load Balancing Berbasis Mikrotik Dalam Meningkatkan Kemampuan Server di Institut Shanti Bhuana. *Journal of Information Technology*, *2*(2), 17–20. https://doi.org/10.46229/jifotech.v2i2.481

Mishra, A. (2017). *Amazon web services for mobile developers : building Apps with AWS*.

Mishra, S., Kumar, M., Singh, N., & Dwivedi, S. (2022). A Survey on AWS Cloud Computing Security Challenges & Solutions. *Proceedings - 2022 6th International Conference on Intelligent Computing and Control Systems, ICICCS 2022*, 614–617. https://doi.org/10.1109/ICICCS53718.2022.9788254

Posey, B. (2021). *What is a Server?* https://www.techtarget.com/whatis/definition/server

Rafli, M. (2022). Jurnal Pengujian Kinerja Load Balancing Web Server menggunakan Nginx Riverse Proxy Berbasis OS Centos 7. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *9*(3), 1824–1840. https://doi.org/10.35957/jatisi.v9i3.2185

Ramadhan, I., & Wulandari, L. (2022). Infrastruktur High-Available Learning Management System Universitas Menggunakan Least-Connected Load Balancer. *Jurnal Masyarakat Informatika*, *13*(2), 99–100. https://doi.org/10.14710/jmasif.13.2.49176

Ramsari, N., & Ginanjar, A. (2022). Implementasi Infrastruktur Server Berbasis Cloud Computing Untuk Web Service Berbasis Teknologi Google Cloud Platform. *Conference SENATIK STT Adisutjipto Yogyakarta*, *7*. https://doi.org/10.28989/senatik.v7i0.472

Sholihah, W., Abdul Malik, A., Novianty, I., & Aziezah, N. (2022). Load Balancing Simulation Android Application as an Online Learning Media. *E3S Web of Conferences*, *348*, 00032. https://doi.org/10.1051/e3sconf/202234800032

Wijaya, G., & Franklyn, F. V. (2021). Perancangan Dan Implementasi Desain Arsitektur Cloud Yang Aman Untuk Sistem Stock Pada Halutime Thrift Shop. *Conference on Business, Social Sciences …*, *1*(1), 39–46. https://journal.uib.ac.id/index.php/conescintech/article/view/5828