

**Jurnal Politeknik Caltex Riau**Terbit Online pada laman <https://jurnal.pcr.ac.id/index.php/jkt/>

| e- ISSN : 2460-5255 (Online) | p- ISSN : 2443-4159 (Print) |

PERBANDINGAN KINERJA INGRESS CONTROLLER PADA KUBERNETES MENGGUNAKAN TRAEFIK DAN NGINX

Wisnu Ramadhani¹, Muhammad Arif Fadhly Ridha^{2*}¹Politeknik Caltex Riau, Teknik Informatika, email: wisnu18ti@mahasiswa.pcr.ac.id²Politeknik Caltex Riau, Teknik Informatika, email: fadhly@pcr.ac.id

*Corresponding Author: fadhly@pcr.ac.id

[1] Abstrak

Pada perkembangan zaman saat ini banyak cara untuk mengekspos sebuah service pada server *kubernetes* agar dapat digunakan oleh client salah satunya NodePort dan LoadBalancer, tetapi mengekspos menggunakan NodePort dan Loadbalancer client harus tau semua IP Node dan semua IP LoadBalancer yang terespos ke public. Oleh karena itu, *ingress* hadir untuk memudahkan client mengakses service dengan hanya menggunakan domain *ingress*. Proyek akhir ini membandingkan *latency*, *throughput*, tingkat kecepatan server menyelesaikan request, pemakaian CPU dan pemakaian memori pada saat membuka halaman web, antara kube *ingress* *traefik*, kube *ingress* *nginx* dan kube tanpa *ingress*. Diperoleh hasil pengujian *latency* tertinggi yaitu 356103.5ms oleh Server Kubernetes cluster tanpa ingress (Loadbalancer) dan *latency* terendah yaitu 5954.1ms oleh Server Kubernetes cluster ingress *traefik*. *Throughput* tertinggi yaitu 3268.8 second oleh Server Kubernetes cluster ingress *nginx* dan *Throughput* terendah yaitu 11.1 second oleh Server Kubernetes cluster ingress *traefik*. Dalam pengujian menggunakan Apache Benchmark Server Kubernetes cluster ingress *traefik* dapat menyelesaikan 100, 300 dan 500 request client lebih dahulu dibandingkan kedua server lainnya. Dalam keadaan *busy*, pemakaian CPU tertinggi adalah server Kubernetes cluster *ingress* *nginx* yaitu 99%, pemakaian memory tertinggi adalah server Kubernetes cluster ingress *traefik* yaitu 46%.

Kata kunci: *Kubernetes, LoadBalancer, Ingress, Traefik, Nginx, NodePort.*

[2] Abstract

In today's development, there are many ways to expose a service on a kubernetes server so that it can be used by clients, one of which is NodePort and LoadBalancer, but exposing using NodePort and Loadbalancer client must know all Node IPs and all LoadBalancer IPs that are exposed to the public. Therefore, ingress is here to make it easier for clients to access services

by only using the ingress domain. This final project compares latency, throughput, server speed rate of completing requests, CPU usage and memory usage at the time of opening a web page, between kube ingress traefik, kube ingress nginx and kube without ingress. The highest latency test results were obtained, namely 356103.5ms by a Kubernetes Server cluster without ingress (Loadbalancer) and the lowest latency of 5954.1ms by a kubernetes server cluster ingress traefik. The highest throughput is 3268.8 seconds by the Kubernetes Server cluster ingress nginx

and the lowest throughput is 11.1 second by the Kubernetes Server cluster ingress traefik. In testing using Apache Benchmark Server Kubernetes cluster ingress traefik can complete 100, 300 and 500 client requests ahead of the other two servers. In a busy state, the highest CPU usage is the Kubernetes cluster ingress nginx server which is 99%, the highest memory usage is the Kubernetes server cluster ingress traefik which is 46%.

Keywords: *Kubernetes, LoadBalancer, Ingress, Traefik, Nginx, NodePort.*

1. Pendahuluan

Kubernetes merupakan sebuah platform open-source yang mampu melakukan management clustering container dalam jumlah besar atau bisa disebut sebagai cluster orchestration yang memiliki tugas melakukan penjadwalan, scaling, recovery dan monitoring container. Untuk membangun layanan microservice. Kubernetes termasuk sebuah software yang dapat melakukan clustering management container sehingga membuat sebuah aplikasi selalu tersedia dan dapat menerima banyak request. Pada Kubernetes cluster diperlukan service sebagai jembatan untuk dapat mengakses satu atau lebih Pod, service juga perlu di ekspos ke luar jaringan dengan tujuan agar aplikasi dari luar kubernetes cluster dapat mengakses Pod yang berada di belakang service tersebut [1].

Mengekspos service dapat dilakukan menggunakan 3 cara yaitu dengan menggunakan NodePort, LoadBalancer dan Ingress. NodePort bekerja dengan membuka port yang akan meneruskan request ke Service yang dituju, LoadBalancer dapat membuat service bisa diakses via LoadBalancer, dan LoadBalancer akan meneruskan request ke NodePort dan dilanjutkan ke Service, sedangkan Ingress adalah resource yang memang ditujukan untuk mengekspos Service. Namun Ingress hanya beroperasi di level HTTP. Jika menggunakan NodePort atau LoadBalancer, client harus dapat mengetahui semua IP address dari Node dan LoadBalancer. Berbeda dengan NodePort atau LoadBalancer, jika menggunakan Ingress, client hanya butuh tahu satu lokasi ip address Ingress [2]. Ingress juga memiliki beberapa keunggulan diantaranya dapat diakses menggunakan nama atau domain, memiliki Load Balancer dan juga Ingress memiliki Sticky Session. Agar Ingress dapat berjalan pada Kubernetes cluster, diperlukan sebuah ingress controller. Ingress controller yang dapat dijalankan pada Kubernetes diantaranya adalah Traefik dan Nginx.

Berdasarkan uraian diatas, perlu dilakukan perbandingan kinerja antara Traefik ingress dan Nginx ingress untuk dapat melihat perbedaan performa pada saat cluster menerima banyak request dari client, sehingga hasil dari penelitian ini dapat menilai *ingress controller* mana yang lebih baik performa dan lebih responsive penggunaannya. Maka dilakukanlah penelitian mengenai “Perbandingan Kinerja Ingress Controller Pada Kubernetes Menggunakan Traefik dan Nginx”[4].

2. Metode Penelitian

Pada penelitian ini menggunakan 3 buah server berbasis sistem operasi linux yang dipasang teknologi virtualisasi yang berbeda yaitu KVM, setiap server akan memiliki 2 buah *nodes* KVM. Server pertama dipasang ingress traefik, server kedua dipasang ingress nginx dan server ketiga tanpa menggunakan ingress.

Adapun metode pengujian yang dilakukan pada penelitian ini yaitu :

- Pengujian *Performance*

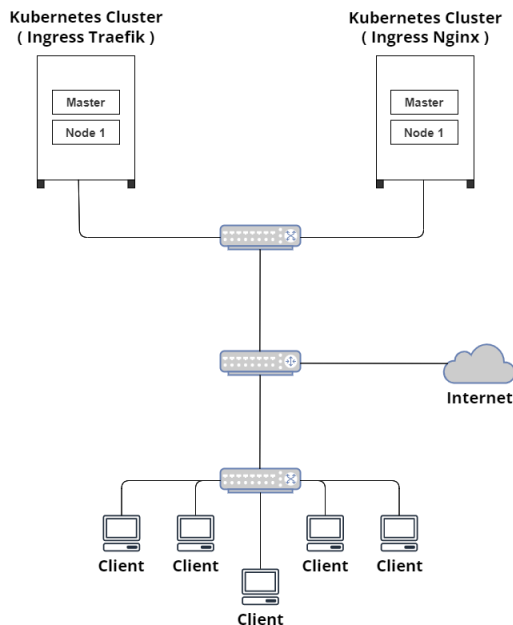
Pengujian *Performance* yang akan diuji dengan beberapa cara, yaitu :

a) Pengujian *Stress Testing*

Pengujian ini dilakukan dengan menggunakan aplikasi *stress tools web server* yaitu Jmeter dan Apache Benchmark. pembebanan dilakukan untuk mengetahui dan mengukur kekuatan sebuah ingress dalam menangani request client dalam jumlah besar secara bersamaan dan dalam satu waktu pada server.

b) Pengujian performa virtualisasi web server dalam keadaan *standby* dan *busy*

Pengujian ini dilakukan dengan menggunakan aplikasi stress tools web server yaitu Jmeter dan Apache Benchmark. pembebanan dilakukan untuk mengetahui dan mengukur kekuatan sebuah ingress dalam menangani request client dalam jumlah besar secara bersamaan dan dalam satu waktu pada server.



Gambar 2.1 Topologi Jaringan

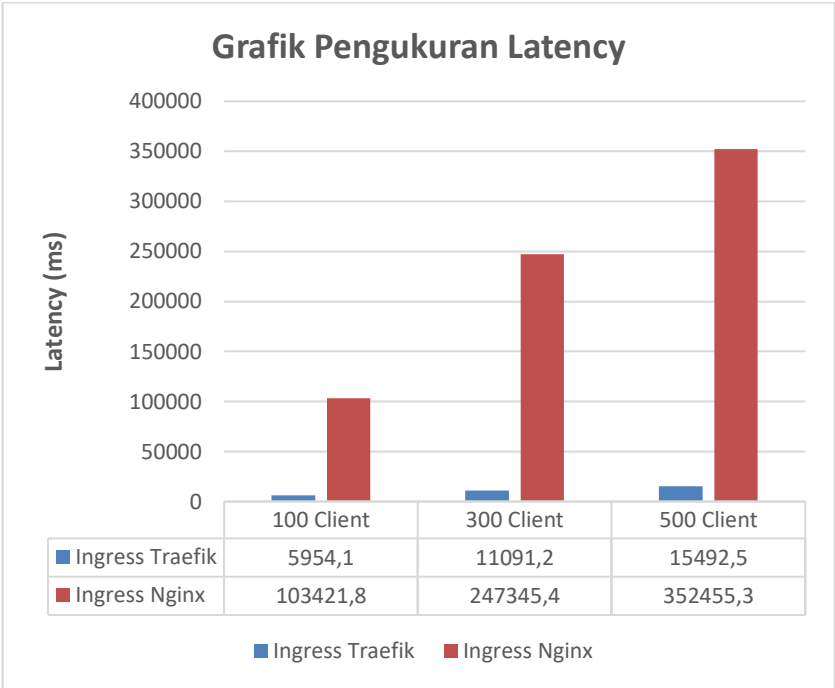
3. Hasil dan Pembahasan

1. Hasil dan Analisis pengujian *Stress Testing*

Pengujian ini menggunakan 2 tools pengujian yaitu :

a) Pengujian Jmeter

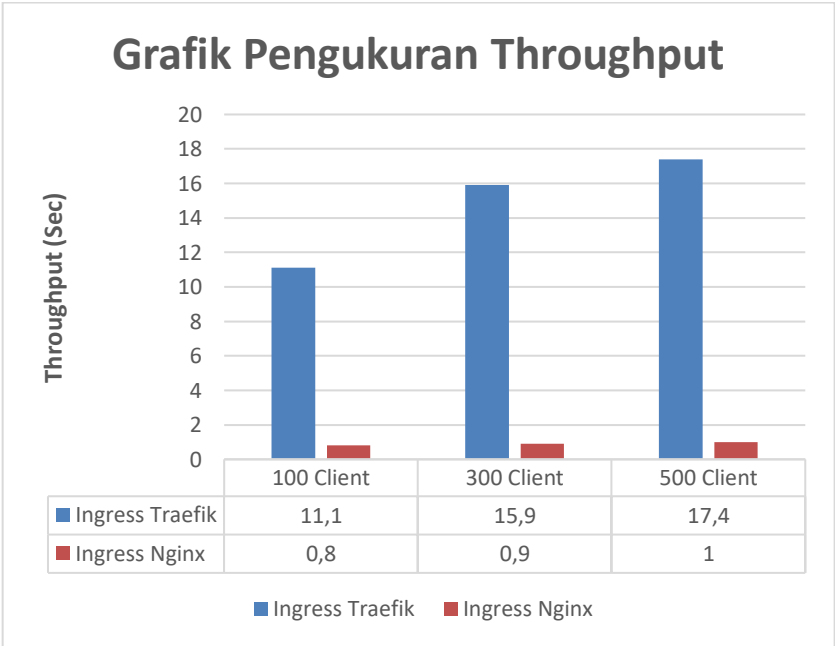
Berdasarkan pengujian *stress testing* terhadap *web server* pada masing-masing *server* menggunakan Jmeter, pengujian bertujuan untuk mengukur tingkat *latency* dan *throughput* yang dilakukan dengan 3 kali tahapan pengujian yaitu 100 *client*, 300 *client* dan 500 *client* dengan masing-masing pengujian dilakukan 5 kali percobaan dalam mengambil data.



Gambar 3.1 Grafik pengukuran latency

Berdasarkan pengujian yang telah dilakukan, didapatkanlah grafik seperti pada Gambar 3.1 yang merupakan hasil rata-rata jumlah latency dari semua pengujian yang dilakukan pada Server Kubernetes cluster ingress traefik dan Kubernetes cluster ingress nginx.

Ingress nginx memiliki jumlah latency paling tinggi yaitu 352455.3 ms, sedangkan jumlah latency paling rendah adalah Server Kubernetes cluster ingress traefik yaitu 5954.1 ms. Kinerja Server Kubernetes cluster ingress traefik lebih baik dibandingkan Server Kubernetes cluster ingress tnginx, karena Server Kubernetes cluster ingress traefik memaksimalkan penggunaan *memory* untuk mendorong kinerja server agar lebih cepat penggunaannya. *latency* yang bagus adalah *latency* dengan jumlah rendah karena dapat mengakses dan membuka halaman web lebih cepat.



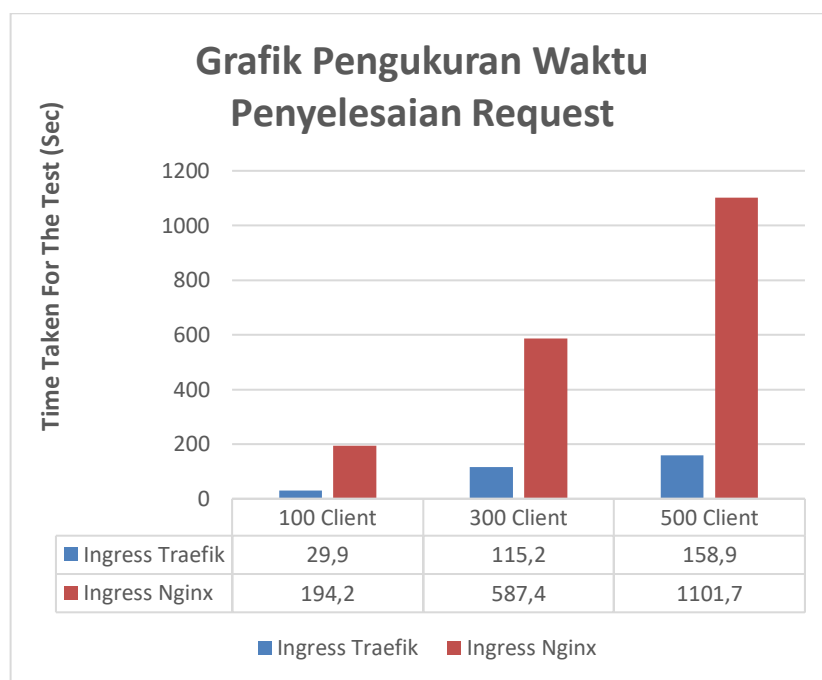
Gambar 3.2 Grafik pengukuran throughput

Pada grafik throughput dapat terlihat Server Kubernetes cluster ingress traefik memiliki jumlah throughput paling tinggi yaitu 17.4/second, sedangkan jumlah throughput paling rendah adalah Server Kubernetes cluster ingress nginx yaitu 0.7/second.

Throughput pada aplikasi pengujian jmeter ini bukan melihat besar data yg keluar masuk pada server, tetapi jumlah request per detik nya. Server *Kubernetes cluster ingress traefik* dapat menerima jumlah request lebih banyak dibandingkan server ingress nginx dikarenakan Server *Kubernetes cluster ingress traefik* menggunakan sumber daya *memory* yang cukup besar untuk memaksimalkan kinerja pada server.

b) Pengujian Apache Benchmark

Berdasarkan pengujian stress testing terhadap web server pada masing-masing server menggunakan Apache Benchmark, pengujian bertujuan untuk mengukur tingkat kecepatan server menyelesaikan request dengan 3 tingkatan pengujian yaitu 100 client, 300 client dan 500 client dengan masing-masing pengujian dilakukan 5 kali percobaan dalam mengambil data.



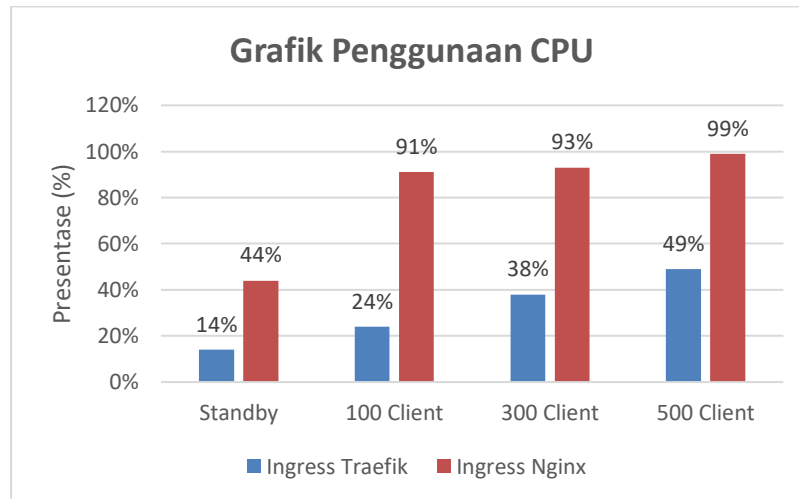
Gambar 3.3 Grafik Pengukuran Waktu Penyelesaian Request

Berdasarkan pengujian yang telah dilakukan, didapatkanlah grafik seperti pada Gambar 3.3 yang merupakan hasil kecepatan server menyelesaikan request dari beberapa tingkat pengujian yang dilakukan pada Server Kubernetes cluster ingress traefik dan Kubernetes cluster ingress nginx.

Pada grafik pengukuran waktu penyelesaian request diatas dapat terlihat diantara kedua Server yang di lakukan pengujian bahwa Server *Kubernetes cluster ingress traefik* adalah server paling cepat menyelesaikan 100, 300 dan 500 request client dibandingkan server Kubernetes cluster ingress nginx, karena Server *Kubernetes cluster ingress traefik* memakan cukup besar memory, dimana hal tersebut membuat kinerja dari Server *Kubernetes cluster ingress traefik* menjadi lebih bagus dan cepat.

2. Hasil dan Analisis pengujian performa virtualisasi web server

Berdasarkan pengujian yang dilakukan pada saat keadaan *server standby* dan *busy* diperoleh data penggunaan CPU.

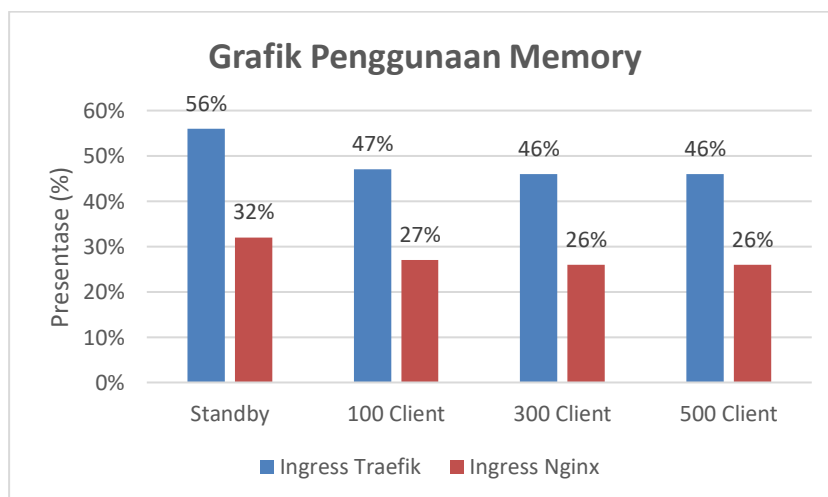


Gambar 3.4 Grafik penggunaan CPU saat standby dan busy

Pada saat dalam keadaan standby, penggunaan CPU terendah adalah server Kubernetes cluster ingress traefik yaitu 14% , sedangkan penggunaan CPU tertinggi adalah server Kubernetes cluster ingress nginx yaitu 49%.

Pada saat keadaan busy, server Kubernetes cluster ingress traefik menunjukkan penggunaan CPU terendah yaitu 49%, sedangkan penggunaan CPU tertinggi adalah server Kubernetes cluster ingress nginx yaitu 99%. Server Kubernetes cluster ingress traefik memiliki kelebihan yang dapat menyeimbangkan beban yang membuat penyebaran microservices menjadi lebih mudah, sehingga penggunaan CPU menjadi lebih kecil dibandingkan Server Kubernetes cluster ingress nginx.

Pengujian dengan membedakan jumlah client ternyata memberikan hasil penggunaan CPU yang berbeda. Semakin banyak client yang mengakses maka akan naik penggunaan CPU. Selanjutnya adalah analisa penggunaan memory pada masing-masing server pada saat standby dan diakses oleh 100 client, 300 client dan 500 client.



Gambar 3.5 Grafik penggunaan Memory saat standby dan busy

Pada saat dalam keadaan standby, server Kubernetes cluster ingress nginx menunjukkan penggunaan memory terendah yaitu 32%, sedangkan penggunaan memory tertinggi adalah server Kubernetes cluster ingress traefik yaitu 56%.

Pada saat dalam keadaan busy, server Kubernetes cluster ingress nginx menunjukkan penggunaan memory terendah yaitu 26%, sedangkan penggunaan memory tertinggi adalah server Kubernetes cluster ingress traefik yaitu 47%.

Penggunaan *memory* Server *Kubernetes cluster ingress traefik* lebih besar dibandingkan server Kubernetes cluster ingress nginx, dikarenakan server *Kubernetes cluster ingress traefik* memaksimalkan penggunaan *memory* untuk dapat membuat kinerja dari server menjadi lebih bagus dan cepat.

4. Kesimpulan

Dari hasil Pengujian dapat diambil kesimpulan sebagai berikut:

- i. Kubernetes cluster dapat diimplementasikan di dalam virtualisasi KVM.
- ii. Dalam pengujian menggunakan Apache Benchmark Server Kubernetes cluster ingress traefik dapat menyelesaikan 100, 300 dan 500 request client lebih dahulu dibandingkan kedua server lainnya.
- iii. Server Kubernetes cluster ingress traefik lebih unggul dibandingkan Server Kubernetes cluster ingress nginx, tetapi penggunaan memory pada Server Kubernetes cluster ingress traefik lebih besar dibandingkan server Kubernetes cluster ingress nginx.
- iv. Server Kubernetes cluster ingress nginx sangat tepat digunakan apabila memiliki spesifikasi perangkat server yang menengah kebawah, karna penggunaan memory pada server tersebut tergolong kecil dibandingkan penggunaan memory pada Server Kubernetes cluster ingress traefik.

Daftar Pustaka

- [1] Apridayanti, S. I. Desain Dan Implementasi Virtualisasi Berbasis Docker Untuk Deployment Aplikasi Web.
- [2] Aziz, A. &. Analisis Web Server untuk Pengembangan Hosting Server Institusi: Perbandingan Kinerja Web Server Apache dengan Nginx.
- [3] Github. Traefik. Diambil kembali dari Github: <https://github.com/traefik/traefik>.
- [4] Dwiyatno, S. R. Implementasi Virtualisasi Server Berbasis Docker Container. Jurnal PROSISKO.
- [5] AHat, R. What is KVM? Virtualization. Diambil kembali dari <https://www.redhat.com/en/topics/virtualization/what-is-KVM>.
- [6] Chandra, A. Y. Analisis Performansi Antara Apache & Nginx Web Server. JURNAL SISTEM DAN INFORMATIKA (JSI), 48-56.
- [7] Rijayana, I. TEKNOLOGI LOAD BALANCING UNTUK MENGATASI BEBAN SERVER.