

Infrastruktur Jaringan Berbasis Cloudflare Tunnel untuk Mendukung Deployment Aplikasi

Ahmad Riziq Maulana Hamzah[#], Tri Nur Arifin[#]

[#] Fakultas Teknik Informatika, Universitas Dian Nusantara, Jakarta, Jakarta Barat, 11470, Indonesia
E-mail: 411221022[at]mahasiswa.undira.ac.id, tri.nur.arifin[at]undira.ac.id

ABSTRACTS

On-premise K3s services need public access for demonstrations, webhooks, and testing, but port forwarding can increase the inbound attack surface. This study evaluates Cloudflare Tunnel as an outbound service publication path and compares it with FRP (Fast Reverse Proxy) as an open-source reverse-proxy baseline. The method is a comparative experiment on a three-node K3s testbed using Ubuntu virtual machines on Proxmox and a three-replica podinfo workload. Grafana k6 constant-arrival-rate tests were executed on the /healthz endpoint through a 100 requests/second A/B scenario and staged targets from 100 to 1,000 requests/second. The observed metrics included success rate, p95 latency, dropped iterations, uptime, DNS resolution, and Nmap results. At 100 requests/second, both paths achieved more than 98.9% success, but FRP produced lower p95 latency, 48.51-136.16 ms compared with Cloudflare Tunnel at 126.44-1,732.67 ms. At 1,000 requests/second, Cloudflare Tunnel was more stable, maintaining 98.555-99.928% success on executed requests with 5,429-8,674 dropped iterations, whereas FRP decreased to 68.497-70.727% success with 40,566-59,558 dropped iterations. Both endpoints recorded 100% uptime over 24 hours and 99.94% over 30 days. DNS and Nmap checks show that Cloudflare hides the origin behind edge proxy addresses, while FRP exposes the public server IP and ports 22, 80, and 443. These results provide a practical basis for solution selection: Cloudflare Tunnel is more suitable for origin hiding and high-load stability, while FRP is better for low-load latency when additional hardening is applied.

Manuscript received Jul 15, 2026;
revised Aug 8 2026; accepted Aug
10, 2026 Date of publication Sep
30, 2026. International Journal,
JITSI : Jurnal Ilmiah Teknologi
Sistem Informasi licensed under a
Creative Commons Attribution-
Share Alike 4.0 International
License



ABSTRAK

Layanan K3s on-premise membutuhkan akses publik untuk demonstrasi, webhook, dan pengujian, tetapi port forwarding dapat menambah permukaan inbound. Penelitian ini bertujuan mengevaluasi Cloudflare Tunnel sebagai jalur publikasi layanan berbasis koneksi outbound dan membandingkannya dengan FRP (Fast Reverse Proxy) sebagai baseline reverse proxy open-source. Metode penelitian menggunakan eksperimen komparatif pada testbed K3s tiga node VM Ubuntu di Proxmox dengan aplikasi podinfo tiga replica. Beban diuji memakai Grafana k6 constant-arrival-rate pada endpoint /healthz melalui skenario A/B 100 request/detik dan uji bertahap 100-1.000 request/detik. Metrik yang diamati meliputi success rate, latency p95, dropped iterations, uptime, resolusi DNS, dan hasil Nmap. Pada 100 request/detik, kedua jalur memiliki success rate di atas 98,9%, tetapi FRP menghasilkan p95 lebih rendah, yaitu 48,51-136,16 ms dibanding Cloudflare Tunnel 126,44-1.732,67 ms. Pada target 1.000 request/detik, Cloudflare Tunnel lebih stabil dengan success rate 98,555-99,928% pada request yang dieksekusi dan 5.429-8.674 dropped iterations, sedangkan FRP turun menjadi 68,497-70,727% dengan 40.566-59.558 dropped iterations. Kedua endpoint mencatat uptime 24 jam 100% dan uptime 30 hari 99,94%. Pemeriksaan DNS dan Nmap menunjukkan Cloudflare menyembunyikan origin melalui edge proxy, sedangkan FRP menampilkan IP

server publik serta port 22, 80, dan 443. Hasil ini bermanfaat sebagai dasar pemilihan solusi: Cloudflare Tunnel lebih sesuai untuk penyembunyian origin dan stabilitas beban tinggi, sementara FRP unggul pada latency beban rendah dengan hardening tambahan.

Keywords / Kata Kunci — *Cloudflare Tunnel, K3s, Network Infrastructure, Zero Trust, Deployment, On-Premise.*

CORRESPONDING AUTHOR

Ahmad Riziq Maulana Hamzah
 Fakultas Teknik Informatika, Universitas Dian Nusantara, Jakarta, Jakarta Barat, 11470, Indonesia
 Email: 411221022[at]mahasiswa.undira.ac.id

1. PENDAHULUAN

Adopsi arsitektur cloud-native mendorong organisasi menggunakan orkestrasi kontainer untuk menyederhanakan deployment, penskalaan, dan pemulihan aplikasi. K3s merupakan distribusi Kubernetes ringan yang sesuai untuk lingkungan edge, laboratorium, dan on-premise dengan sumber daya terbatas [1], [2]. Pada PT Cloudfren Node Asia, K3s digunakan sebagai platform pengembangan, tetapi layanan yang berjalan pada jaringan internal tetap membutuhkan mekanisme akses publik untuk demonstrasi, webhook, integrasi eksternal, dan pengujian.

Metode sementara seperti kubectrl port-forward bergantung pada sesi pengguna dan tidak ditujukan sebagai mekanisme publikasi persisten [3]. Pembukaan inbound port dan port forwarding pada perangkat jaringan dapat menyediakan koneksi langsung, tetapi menambah permukaan serangan serta beban konfigurasi firewall, alamat IP publik, sertifikat, dan pemetaan port. Kebutuhan ini mendorong penggunaan tunnel yang dibentuk dari jaringan internal menuju penyedia atau server perantara.

NIST mendefinisikan Zero Trust sebagai pendekatan yang tidak memberikan kepercayaan implisit hanya berdasarkan lokasi jaringan dan menempatkan perlindungan pada pengguna, aset, serta sumber daya [4]. Cloudflare Tunnel membentuk koneksi outbound dari konektor cloudflared menuju jaringan Cloudflare sehingga layanan internal dapat dipublikasikan tanpa membuka koneksi inbound langsung ke node. Cloudflare Access dapat menambahkan autentikasi dan kebijakan identitas sebelum akses ke aplikasi [5], [6]. Sebagai pembanding, FRP merupakan reverse proxy open-source yang dapat mengekspos server di belakang NAT atau firewall melalui komponen frpc dan frps [7].

Naskah awal berfokus pada keberhasilan implementasi Cloudflare Tunnel, tetapi belum menyediakan bukti kuantitatif yang memadai. Artikel arsitektur jaringan memerlukan testbed yang dapat direplikasi, skenario beban yang jelas, metrik terukur, dan pembanding. Oleh karena itu, penelitian ini difokuskan menjadi evaluasi performansi Cloudflare Tunnel dan FRP pada kluster K3s terbatas. FRP diposisikan sebagai baseline akses publik, bukan sebagai implementasi ZTNA yang ekuivalen. Kontribusi penelitian ini adalah:

- merancang topologi testbed yang mempertahankan backend dan beban kerja yang sama untuk jalur Cloudflare Tunnel dan FRP;
- membandingkan throughput, latency p95, HTTP success rate, timeout/non-200, failed-request rate, serta jumlah virtual user pada beban terkontrol;
- mengidentifikasi titik saturasi ketika target 1.000 request/detik belum dapat dicapai secara stabil; dan
- menyediakan protokol lanjutan untuk mengukur overhead CPU/RAM serta memvalidasi keamanan melalui port scanning dan trafik serangan ringan yang berizin.

2. METODOLOGI PENELITIAN

2.1. Desain Penelitian

Penelitian menggunakan pendekatan eksperimen komparatif pada satu testbed. Variabel bebas adalah jalur tunnel, yaitu Cloudflare Tunnel dan FRP. Variabel terikat meliputi request rate yang tercapai, HTTP 200 rate, non-200/timeout rate, latency p95, failed-request rate, jumlah virtual user aktif, serta overhead CPU dan RAM pada node K3s dan proses tunnel. Variabel kontrol mencakup endpoint, payload, backend, durasi, lokasi load generator, protokol HTTPS, dan interval pengamatan.

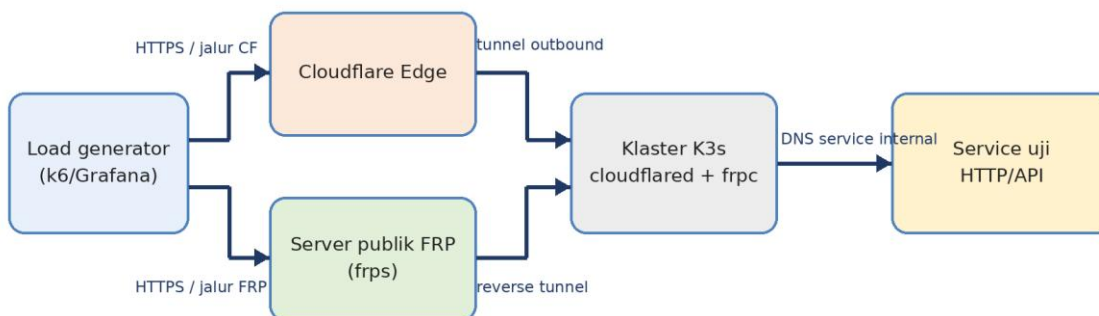
Eksperimen dibagi menjadi dua kelompok. Pertama, pengujian A/B berurutan pada target sekitar 100 request/detik untuk mengurangi interferensi antarjalur. Kedua, uji stres dengan jumlah virtual user yang dinaikkan hingga sekitar 3.000 untuk mengamati titik saturasi. Grafik dashboard yang tersedia digunakan sebagai bukti observasi; angka yang tidak terekam pada dashboard tidak direkonstruksi.

2.2. Topologi Testbed

Load generator mengirim request HTTPS ke dua hostname publik yang mengarah ke service uji yang sama. Jalur pertama melewati edge Cloudflare dan konektor cloudflared di kluster. Jalur kedua melewati server publik

FRP (frps) dan konektor FRP (frpc) menuju service internal. Dengan backend yang sama, perbedaan metrik jaringan terutama dapat dikaitkan dengan jalur tunnel, selama tidak ada perubahan kondisi jaringan dan sumber daya antar-run.

Topologi testbed komparatif Cloudflare Tunnel dan FRP



Metrik: request rate, HTTP 200, status non-200/timeout, p95, failed rate, VU, CPU, dan RAM.

GAMBAR 1. Topologi testbed komparatif. Nilai spesifikasi perangkat harus dilengkapi sesuai konfigurasi aktual

2.3. Spesifikasi Perangkat

TABEL 1. Spesifikasi Perangkat yang digunakan

Komponen	Spesifikasi yang digunakan
Node K3s	3 mini PC identik sebagai host Proxmox; masing-masing 4 core CPU, RAM 12 GB, disk 128 GB. Setiap host menjalankan VM Ubuntu untuk node K3s dengan 4 vCPU dan RAM 10 GB;
Klaster	3 node VM Ubuntu; dashboard agregat menampilkan 12 CPU, RAM 34,72 GiB, dan storage 405,86 GiB; [LENGKAPI versi K3s, CNI, ingress controller, dan konfigurasi storage]
Cloudflare Tunnel	Argo CD Healthy dan Synced; Deployment, ServiceAccount, 2 pod aktif (1/1); [LENGKAPI versi image cloudflared, resource request/limit, protokol QUIC/HTTP2]
FRP	frpc: Argo CD Healthy dan Synced; Deployment, ServiceAccount, Secret konfigurasi, 1 pod aktif (1/1). frps: Tencent Lighthouse region Singapore/Silicon Valley, IP 43.134.177.115, 4 vCPU, RAM 8 GB, SSD 180 GB, peak bandwidth 30 Mbps, kuota trafik 5.120 GB/bulan;
Aplikasi uji	podinfo, 3 replica pada ubuntu-server1/2/3; snapshot CPU total $\pm 0,05$ core, RAM total $\pm 62,1$ MiB, restart 0;
Load generator	Grafana k6, constant-arrival-rate, GET /healthz, timeout 10 s, maksimum 3.000 VU;
Observabilitas	Grafana dashboard dan uptime monitoring tipe Ping dengan heartbeat 60 detik, retries 0;

2.4. Arsitektur dan Implementasi Jaringan

Arsitektur yang diusulkan menempatkan konektor cloudflared di dalam klaster K3s. Konektor tersebut membangun koneksi outbound terenkripsi menuju jaringan edge Cloudflare. Ketika pengguna mengakses domain publik, permintaan diterima oleh Cloudflare dan diteruskan melalui tunnel menuju konektor cloudflared. Selanjutnya, konektor meneruskan trafik ke layanan K3s berdasarkan DNS internal yang telah dikonfigurasi.

Implementasi dilakukan melalui beberapa tahapan. Pertama, tunnel dibuat pada dashboard Cloudflare Zero Trust dan dihubungkan dengan domain yang dikelola perusahaan. Kedua, konektor cloudflared dijalankan di dalam klaster menggunakan token tunnel. Ketiga, public hostname dikonfigurasi dan dipetakan menuju service internal K3s. Keempat, status koneksi diperiksa melalui dashboard Cloudflare dan log konektor. Terakhir, aplikasi diuji melalui jaringan di luar lingkungan perusahaan.

Dalam rancangan ini, alamat IP pod dan node tidak diekspos secara langsung. Trafik dari internet hanya diteruskan melalui koneksi tunnel yang dibentuk oleh konektor dari dalam jaringan. Cloudflare Access juga dapat diterapkan untuk membatasi akses berdasarkan identitas pengguna dan kebijakan yang telah ditentukan. Begitupun juga FRP tidak terekspos hanya menghubungkan FPRC yang ada di local device dan FRPS yang ada di server online.

2.5. Skeanrio Test

Pengujian menggunakan script k6 constant-arrival-rate. Script menerima satu atau beberapa BASE_URL, memilih target berdasarkan iterationInTest modulo jumlah URL, lalu mengirim HTTP GET ke /healthz. Request timeout ditetapkan 10 detik. Metrik khusus yang direkam meliputi response time, failed checks, success rate, timeout rate, dan HTTP 200 rate. Summary memuat avg, min, median, maksimum, p90, p95, dan p99.

Konfigurasi manual menetapkan BASE_URLS ke endpoint FRP dan Cloudflare.

- DOMAIN_LABELS=frp

- cloudflare_tunnel
- RATE=2000 iterasi/detik
- DURATION=1m
- PRE_ALLOCATED_VUS=1200
- MAX_VUS=3000

Karena dua URL dipilih secara bergantian, target nominalnya sekitar 1.000 request/detik per endpoint. Threshold menetapkan failed rate <5%, p95 <1.000 ms, p99 <2.000 ms, success rate >95%, dan timeout rate <1%. Pada run matriks terpisah, target diuji pada 100, 250, 500, 750, dan 1.000 request/detik. Arrival-rate model memulai iterasi secara independen dari response time; dropped iterations menunjukkan generator tidak memperoleh VU bebas atau sistem mengalami degradasi [8]-[11].

TABEL 2. Parameter yang digunakan dalam penelitian

Parameter	Nilai
Endpoint	https://pod.test-frp.cloudfren.id https://pod-test-cf.cloudfren.id
Path/metode	GET /healthz
Executor	constant-arrival-rate; timeUnit 1 s
Konfigurasi manual	RATE 2.000 total; DURATION 1 menit; pre-allocated 1.200 VU; maksimum 3.000 VU
Timeout	10 detik
Threshold	failed <5%; p95 <1.000 ms; p99 <2.000 ms; success >95%; timeout <1%

TABEL 3. Skenario yang digunakan dalam penelitian

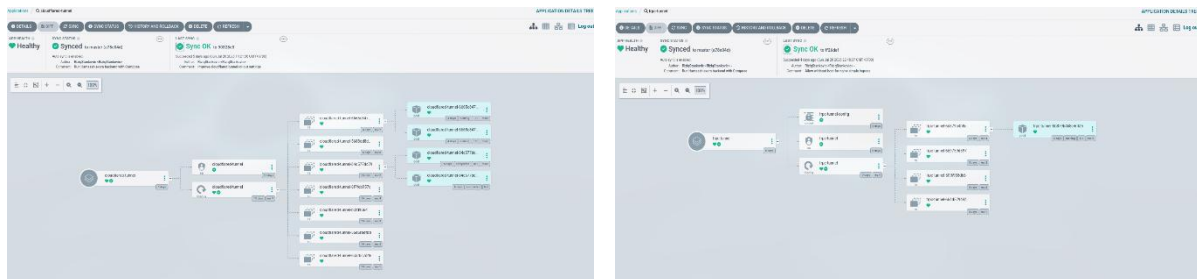
Skenario	Beban	Durasi	Tujuan
A/B baseline	±100 req/s per jalur, diuji bergantian	±3 menit/jalur	Membandingkan p95, error, dan VU pada beban yang sama
Stress/saturation	Naik hingga ±3.000 VU	±1-2 menit peak	Menentukan throughput maksimum teramati dan pola kegagalan
Matriks bertahap	100, 250, 500, 750, 1.000 req/s	1-3 run valid/target	Menguji degradasi sampai 1.000 req/s
Resource overhead	Idle dan tiap tahap beban	Sama dengan load test	Mengukur delta CPU/RAM cloudflared dan frpc
Security validation	Nmap TCP connect + service detection, port 1-1000	Satu scan/hostname	Membandingkan layanan publik FRP dan edge Cloudflare

3. HASIL DAN PEMBAHASAN

3.1. Implementasi Cloudflare Tunnel dan FRP

Pada jalur Cloudflare Tunnel, konektor cloudflared berjalan di lingkungan K3s dan membentuk koneksi outbound menuju edge Cloudflare. Hostname publik diarahkan ke service internal melalui DNS Kubernetes. Node dan pod tidak memerlukan inbound port publik langsung. Pada jalur FRP, frpc membentuk reverse tunnel menuju frps pada server publik, kemudian trafik diteruskan ke service internal yang sama. Kedua jalur berhasil menyediakan endpoint publik dan dapat diamati melalui dashboard.

Perbandingan ini tidak menyatakan bahwa FRP memiliki kontrol identitas yang setara dengan Cloudflare Access. FRP digunakan sebagai baseline reverse proxy open-source. Evaluasi ZTNA pada Cloudflare juga perlu memverifikasi bahwa kebijakan Access benar-benar aktif; jika hostname hanya menggunakan Tunnel tanpa Access, hasil yang diukur terutama merepresentasikan tunneling dan proxy, bukan keseluruhan fungsi ZTNA.



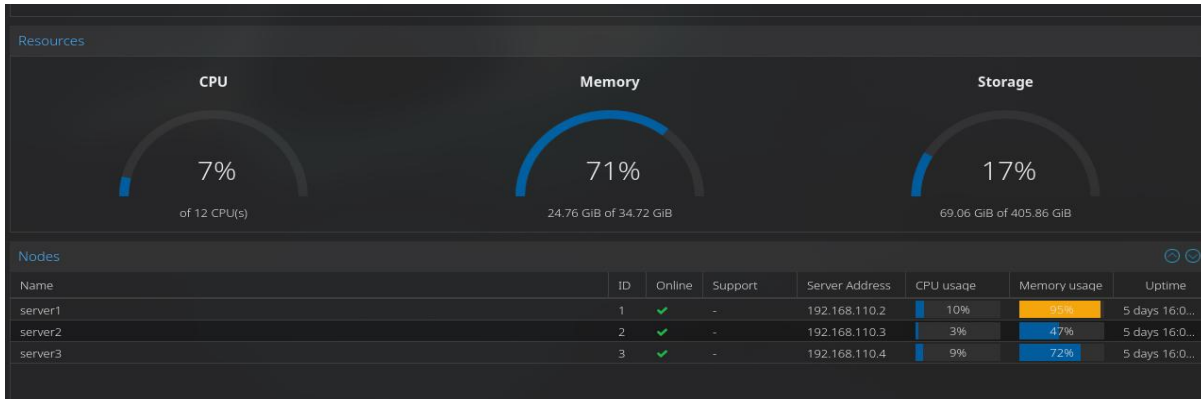
GAMBAR 2. Status aplikasi Argo CD: cloudflared-tunnel (kiri) dan frpc-tunnel (kanan).

Argo CD menunjukkan kedua aplikasi berstatus Healthy dan Synced. cloudflared-tunnel dikelola sebagai Deployment dengan ServiceAccount dan dua pod aktif 1/1 pada ReplicaSet saat ini. frpc-tunnel menggunakan Deployment, ServiceAccount, Secret konfigurasi, dan satu pod aktif 1/1. Bukti ini memverifikasi model deployment dan jumlah replica saat pengamatan, tetapi tidak menampilkan tag image, versi binary, environment lengkap, atau resource request/limit.



GAMBAR 3. Spesifikasi paket Tencent Lighthouse FRPS (kiri) dan snapshot resource server (kanan)

Server publik frps menggunakan Tencent Lighthouse di region Singapore/Silicon Valley dengan 4 vCPU, RAM paket 8 GB, SSD 180 GB, peak bandwidth 30 Mbps, dan kuota trafik bulanan 5.120 GB. Pada snapshot sistem, empat core CPU masing-masing terpakai sekitar 9,1%, 4,7%, 6,0%, dan 4,7%; load average 0,43, 0,40, dan 0,27; memori terpakai 3,69 dari 7,44 GiB; swap terpakai 1,15 dari 1,94 GiB; serta uptime 80 hari 21 jam 12 menit. Karena waktu snapshot tidak dikorelasikan dengan run k6, angka tersebut hanya mendeskripsikan kondisi server saat pengambilan gambar dan bukan overhead frps pada setiap tingkat beban.



GAMBAR 4. Ringkasan resource agregat dan kondisi tiga node Proxmox.

Klaster on-premise menggunakan tiga mini PC dengan spesifikasi identik. Setiap mini PC memiliki 4 core CPU, RAM fisik 12 GB, dan disk 128 GB serta menjalankan satu VM Ubuntu untuk node K3s dengan alokasi 4 vCPU dan RAM 10 GB. Dashboard menampilkan tiga node online, total 12 CPU, penggunaan CPU agregat 7%, penggunaan memori 24,76 dari 34,72 GiB (71%), dan storage 69,06 dari 405,86 GiB (17%). Pada tingkat node, CPU server1, server2, dan server3 masing-masing sekitar 10%, 3%, dan 9%, sedangkan penggunaan memorinya sekitar 95%, 47%, dan 72%. Nilai storage agregat pada dashboard lebih besar daripada hasil perkalian disk fisik yang disebutkan, sehingga jenis pool, overhead, atau sumber storage tambahan perlu dijelaskan pada konfigurasi testbed.

3.2. Hasil Pengujian A/B pada 100 Request/Detik



GAMBAR 5. Dashboard pengujian A/B sekitar 100 request/detik. FRP ditampilkan kuning dan Cloudflare Tunnel hijau

Pada periode sekitar 00:35-00:38, jalur FRP mempertahankan total request rate mendekati 100 request/detik. Setelah pergantian sekitar 00:38, Cloudflare Tunnel juga mempertahankan nilai yang sama hingga sekitar 00:41. HTTP 200 rate pada kedua jalur mengikuti total request rate, sehingga mayoritas request berhasil.

CSV memuat tiga run valid per jalur pada target 100 request/detik. FRP mencatat success rate 98,925-100%, p95 48,51-136,16 ms, dan nol dropped iteration. Cloudflare mencatat success rate 99,667-99,789%, p95 126,44-1.732,67 ms, dan nol dropped iteration. Dengan demikian, kedua jalur mampu mempertahankan arrival rate pada beban ini, tetapi FRP secara konsisten memberikan p95 lebih rendah. Variasi Cloudflare yang mencapai p95 1,73 detik menunjukkan satu run anomali yang perlu dijelaskan menggunakan log pada timestamp yang sama.

Grafik VU pada salah satu run memperlihatkan FRP umumnya membutuhkan sekitar 14-16 VU setelah fase awal, sedangkan Cloudflare sekitar 15-20 VU dengan kenaikan transien hingga sekitar 50 VU. Pada arrival rate yang sama, kebutuhan VU lebih tinggi konsisten dengan iterasi yang membutuhkan waktu lebih lama. Hasil ini mendukung kesimpulan bahwa FRP lebih cepat pada beban rendah, tanpa menunjukkan perbedaan throughput yang berarti pada 100 request/detik.

TABEL 4. Hasil pengukuran berdasarkan indikator

Indikator	FRP	Cloudflare Tunnel	Interpretasi terbatas
Run valid	3	3	Tiga observasi per jalur
Success rate	98,925-100%	99,667-99,789%	Keduanya sangat tinggi
Latency p95	48,51-136,16 ms	126,44-1.732,67 ms	FRP lebih rendah; satu run CF anomali
Dropped iterations	0	0	Target arrival rate dapat dijadwalkan
VU dashboard	±14-16	±15-20; transien ±50	CF membutuhkan concurrency lebih tinggi pada run visual

Catatan: angka success rate, p95, dan dropped iterations berasal dari matrix-summary.csv; estimasi VU berasal dari pembacaan dashboard. Dataset belum memuat actual_rps dan confidence interval.

3.3. Hasil Uji Tes dan Monitoring Grafik



GAMBAR 6. Uji stres dengan kenaikan hingga sekitar 3.000 VU

CSV menyediakan hasil bertahap yang lebih kuat daripada pembacaan dashboard. Pada Cloudflare, run 250 dan 500 RPS tidak mencatat dropped iteration dengan success rate 99,927-100%. Pada 750 RPS, satu run tetap mencapai 100% tanpa dropped iteration, sedangkan run lain mencatat success 98,870% dan 3.022 dropped iterations. Pada 1.000 RPS, success request yang dieksekusi tetap 98,555-99,928%, tetapi seluruh run memakai 3.000 VU, p95 meningkat menjadi 794-3.982 ms, dan 5.429-8.674 iterations terbuang. Artinya Cloudflare mempertahankan kualitas respons yang tinggi, tetapi belum selalu dapat menjadwalkan seluruh target 1.000 RPS.

FRP menunjukkan degradasi lebih awal. Pada 250 RPS, dua run valid mencapai 100% success dan p95 50,00-59,72 ms tanpa dropped iteration. Pada 500 RPS, success turun menjadi 93,259-95,597%, p95 menjadi 2.307-2.505 ms, dan dropped iterations mencapai 1.155-1.708. Pada 750 RPS, success hanya 73,663-75,093% dengan p95 7.989-8.173 ms dan 14.820-21.862 dropped iterations. Pada 1.000 RPS, success turun menjadi 68,497-70,727%, p95 8.754-8.877 ms, serta 40.566-59.558 dropped iterations. Satu run FRP 250 RPS dengan nol request dikeluarkan dari analisis karena tidak menghasilkan sampel.

Hasil kuantitatif menunjukkan titik silang: FRP unggul pada latency 100-250 RPS, tetapi Cloudflare jauh lebih stabil mulai 500 RPS. Dropped iterations tetap harus dibaca bersama actual_rps; kolom actual_rps dalam CSV masih kosong sehingga tingkat pencapaian request rate aktual perlu dihitung dari timestamp atau output summary JSON.

TABEL 5. Hasil pengukuran 100 – 1000 RPS

Target	Cloudflare Tunnel	FRP
100 RPS	Success 99,667-99,789%; p95 126-1.733 ms; drop 0	Success 98,925-100%; p95 49-136 ms; drop 0
250 RPS	Success 99,927-100%; p95 87-135 ms; drop 0	Success 100%; p95 50-60 ms; drop 0
500 RPS	Success 99,998-100%; p95 69-88 ms; drop 0	Success 93,259-95,597%; p95 2.307-2.505 ms; drop 1.155-1.708
750 RPS	Success 98,870-100%; p95 73-233 ms; drop 0-3.022	Success 73,663-75,093%; p95 7.989-8.173 ms; drop 14.820-21.862
1.000 RPS	Success 98,555-99,928%; p95 794-3.982 ms; drop 5.429-8.674	Success 68,497-70,727%; p95 8.754-8.877 ms; drop 40.566-59.558

3.4. Resource Aplikasi

Name	Namespace	CPU	Memory	Container	Restarts	Controlled By	Node	Status	Age
podinfo-54575d849b-tl5qv	prod	0.020	20.6MiB	Running	0	ReplicaSet	ubuntu-server1	Running	2m4s
podinfo-image-prepuller-lpp68	prod	0.000	256.0KiB	Running	0	DaemonSet	ubuntu-server1	Running	2m5s
podinfo-54575d849b-mbj8q	prod	0.020	22.5MiB	Running	0	ReplicaSet	ubuntu-server2	Running	106s
podinfo-image-prepuller-7glqv	prod	0.000	264.0KiB	Running	0	DaemonSet	ubuntu-server2	Running	2m5s
podinfo-54575d849b-8k5bz	prod	0.010	19.0MiB	Running	0	ReplicaSet	ubuntu-server3	Running	2m5s
podinfo-image-prepuller-5pb4n	prod	0.000	256.0KiB	Running	0	DaemonSet	ubuntu-server3	Running	2m5s

GAMBAR 7. Snapshot penggunaan CPU, memori, status, dan penempatan pod aplikasi pada tiga node K3s

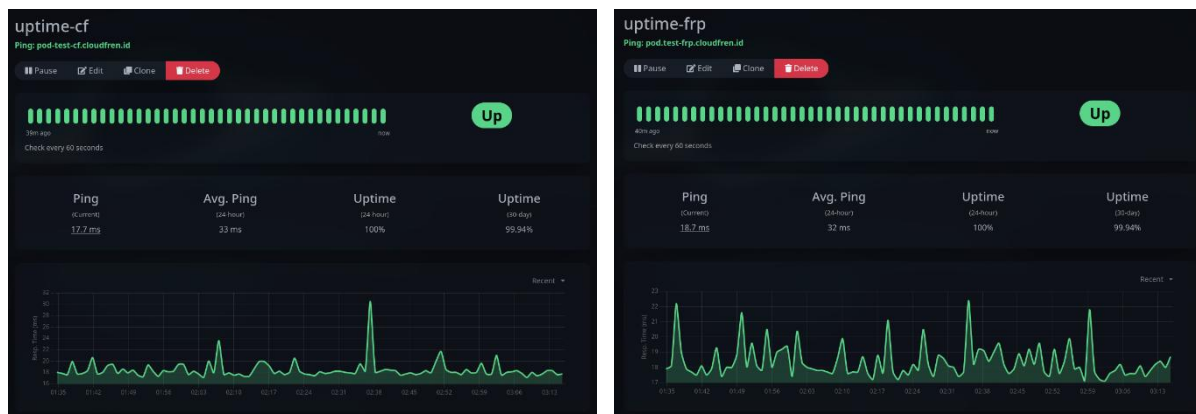
Snapshot menunjukkan tiga replica aplikasi podinfo berstatus Running dan tersebar pada ubuntu-server1, ubuntu-server2, serta ubuntu-server3. Penggunaan CPU masing-masing replica adalah sekitar 0,020; 0,020; dan 0,010 core, sehingga total teramati sekitar 0,050 core. Penggunaan memori masing-masing adalah 20,6 MiB; 22,5 MiB; dan 19,0 MiB, atau total sekitar 62,1 MiB. Ketiga replica tidak mencatat restart pada saat pengambilan gambar. Tiga pod image-prepuller menggunakan CPU 0,000 core dan memori sekitar 256-264 KiB per pod.

Data ini membuktikan bahwa workload aplikasi berada dalam kondisi ringan dan tersebar pada tiga node pada saat snapshot. Namun, usia pod baru sekitar 106 detik sampai 2 menit, dan tampilan difilter dengan kata podin sehingga pod cloudflared, frpc, ingress, serta metrik total node tidak terlihat. Karena itu, angka tersebut tidak dapat digunakan sebagai bukti overhead tunnel atau konsumsi resource ketika beban 100-1.000 request/detik dijalankan.

TABEL 5. Hasil pengukuran Replica

Replica	Node	CPU	Memori	Restart
podinfo-...-tl5qv	ubuntu-server1	0,020 core	20,6 MiB	0
podinfo-...-mbj8q	ubuntu-server2	0,020 core	22,5 MiB	0
podinfo-...-8k5bz	ubuntu-server3	0,010 core	19,0 MiB	0
Total podinfo	3 node	±0,050 core	±62,1 MiB	0

3.5. Hasil ping dan UPTIME aplikasi



GAMBAR 8. Monitoring Cloudflare Tunnel (kiri) dan FRP (kanan) dengan interval pemeriksaan 60 detik.

Kedua endpoint berstatus Up pada saat pengamatan dan diperiksa setiap 60 detik. Endpoint Cloudflare mencatat ping saat ini 17,7 ms, rata-rata 24 jam 33 ms, uptime 24 jam 100%, dan uptime 30 hari 99,94%. Endpoint FRP mencatat ping saat ini 18,7 ms, rata-rata 24 jam 32 ms, uptime 24 jam 100%, dan uptime 30 hari 99,94%. Selisih rata-rata ping 24 jam hanya sekitar 1 ms, sehingga secara praktis kedua jalur menunjukkan availability dan latency monitoring yang hampir setara pada periode tersebut.

Grafik recent memperlihatkan spike sesaat pada kedua jalur. Cloudflare sempat meningkat hingga sekitar 30 ms, sedangkan FRP umumnya berada sekitar 17-22 ms. Data monitoring ini berguna untuk bukti ketersediaan, tetapi tidak dapat menggantikan latency HTTP p95 pada load test karena metode ping, lokasi probe, protokol, serta workload berbeda.

TABEL 6. Hasil pengukuran Replica

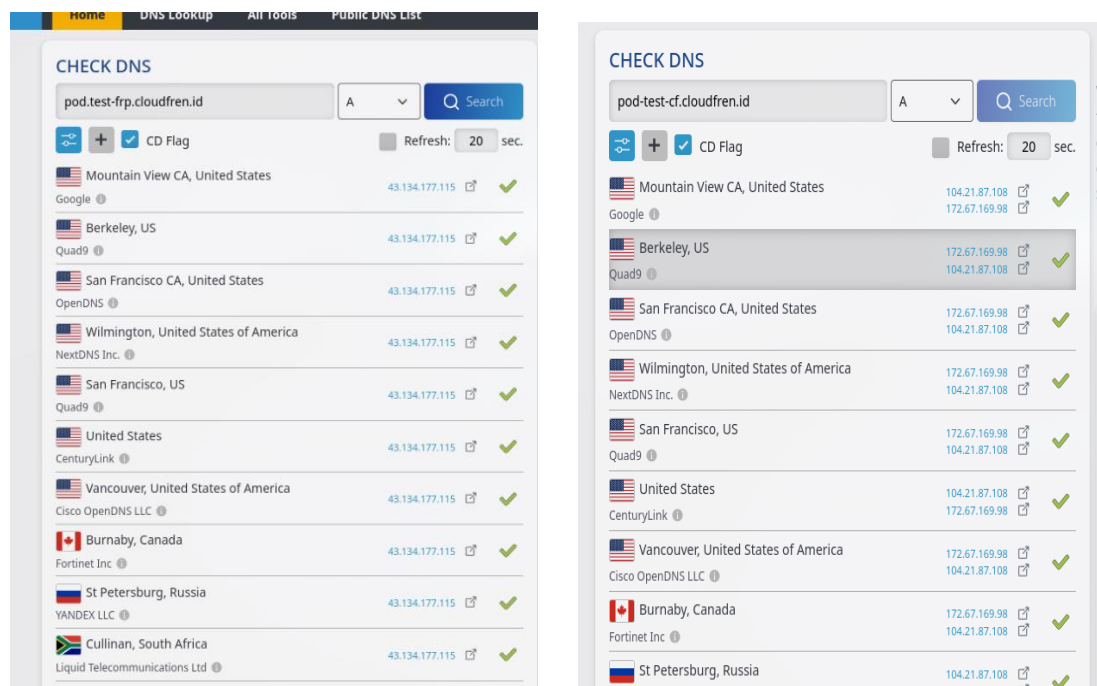
Metrik	Cloudflare Tunnel	FRP	Interpretasi
Ping saat snapshot	17,7 ms	18,7 ms	CF lebih rendah 1 ms pada satu titik
Rata-rata ping 24 jam	33 ms	32 ms	Selisih kecil; FRP lebih rendah 1 ms
Uptime 24 jam	100%	100%	Setara pada periode 24 jam
Uptime 30 hari	99,94%	99,94%	Setara pada periode 30 hari



GAMBAR 9. Konfigurasi monitor ping Cloudflare pada Uptime

Konfigurasi monitor menggunakan tipe Ping dengan heartbeat interval 60 detik, retries 0, heartbeat retry interval 60 detik, dan resend notification 0. Dengan retries 0, satu pemeriksaan gagal dapat langsung menandai endpoint down. Konfigurasi yang sama perlu dipastikan pada monitor FRP agar perbandingan uptime benar-benar setara.

3.6. Hasil Pemeriksaan Resolusi DNS



GAMBAR 10. Pemeriksaan DNS record A FRP (kiri) dan Cloudflare Tunnel (kanan) dari beberapa resolver publik

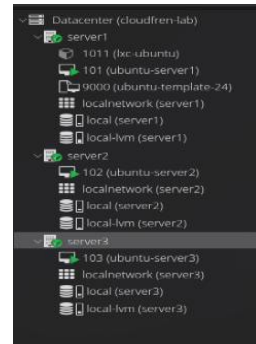
Hostname `pod.test-frp.cloudfren.id` secara konsisten mengembalikan satu alamat IPv4 publik, yaitu 43.134.177.115, pada resolver yang ditampilkan. Pola tersebut menunjukkan bahwa pengguna memperoleh alamat server publik FRP secara langsung. Jika IP tersebut adalah origin frps, alamat origin menjadi terlihat dan perlu dilindungi dengan firewall, pembatasan port, TLS, autentikasi, serta hardening sistem.

Hostname `pod.test-cf.cloudfren.id` mengembalikan dua alamat IPv4, yaitu 104.21.87.108 dan 172.67.169.98, dengan urutan yang dapat berbeda antar resolver. Alamat tersebut merupakan jalur proxy Cloudflare yang terlihat oleh klien, bukan bukti bahwa alamat origin K3s dipublikasikan. Temuan ini mendukung klaim bahwa Cloudflare menyembunyikan origin dari resolusi DNS publik dan menyediakan redundansi pada lapisan proxy.

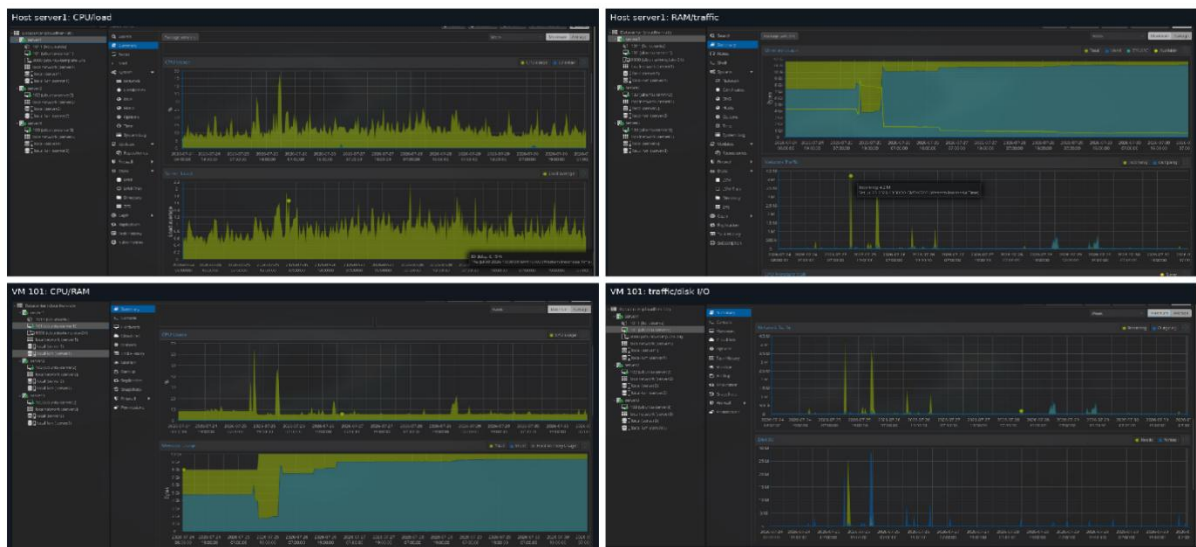
Pemeriksaan DNS tidak membuktikan bahwa seluruh port origin tertutup atau bahwa serangan telah dimitigasi. Verifikasi tersebut tetap memerlukan port scan berizin terhadap alamat yang relevan, pemeriksaan firewall, dan korelasi log Cloudflare, ingress, serta node K3s.

3.7. Overhead CPU dan RAM

Metrik Proxmox tersedia untuk periode mingguan 24-31 Juli 2026 dan telah dipetakan secara benar: host server1 dengan VM 101 (ubuntu-server1), host server2 dengan VM 102 (ubuntu-server2), serta host server3 dengan VM 103 (ubuntu-server3). Grafik mencakup CPU, server load, penggunaan RAM, network traffic, dan disk I/O. Rentang ini mencakup tanggal pelaksanaan pengujian, sehingga dapat digunakan untuk menunjukkan kondisi infrastruktur pada periode eksperimen.

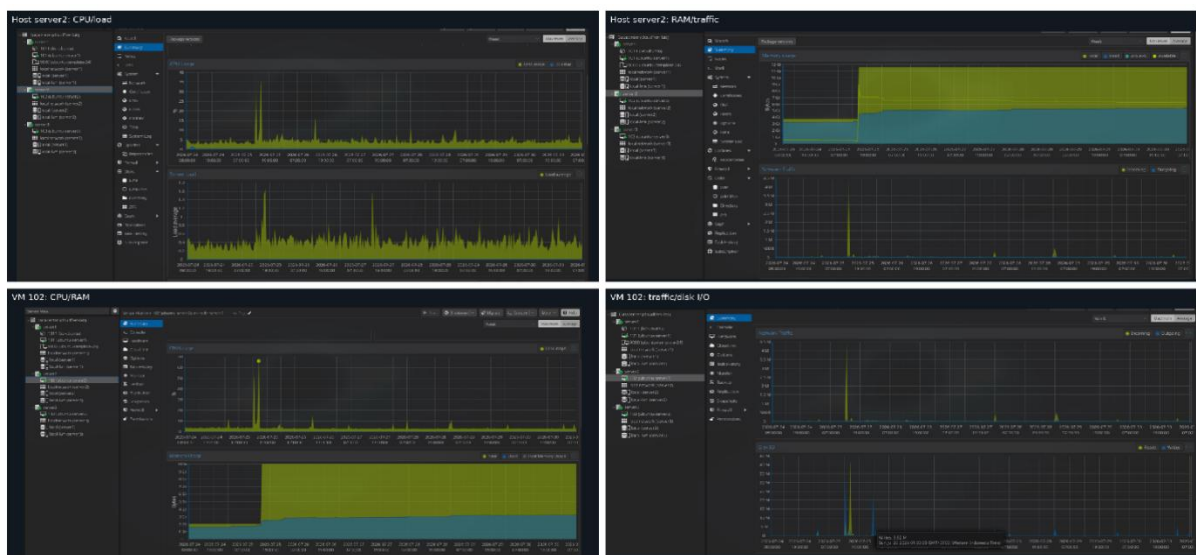


GAMBAR 11. Pemetaan tiga host Proxmox terhadap VM Ubuntu node K3s



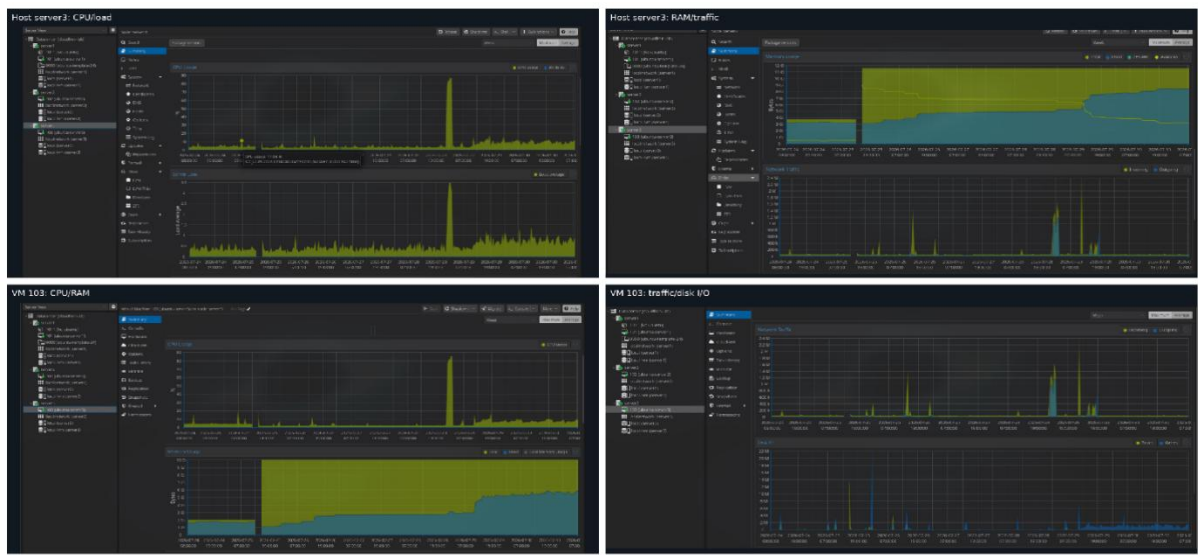
GAMBAR 11. Metrik host server1 dan VM 101 (ubuntu-server1), periode 24-31 Juli 2026

Server1 umumnya menggunakan CPU sekitar 8-15% dengan beberapa spike hingga sekitar 47%; load average sebagian besar di bawah 1,2 dengan spike sekitar 2,2. Memori host meningkat hingga sekitar 11 GiB dari total sekitar 12 GiB. VM 101 umumnya memakai CPU sekitar 6-10% dengan spike sekitar 63%, sementara penggunaan memorinya meningkat bertahap hingga sekitar 9,3 GiB dari alokasi 10 GiB. Network traffic menunjukkan spike incoming sekitar 4,2 MB pada 25 Juli, dan disk I/O VM menunjukkan burst write mendekati 30 MB.



GAMBAR 12. Metrik host server2 dan VM 102 (ubuntu-server2), periode 24-31 Juli 2026.

Server2 memiliki CPU dasar sekitar 3-6% dengan spike sekitar 35%; load average umumnya di bawah 0,7 dan sesekali mencapai sekitar 1,6. Penggunaan memori host berada sekitar 5-5,5 GiB setelah VM aktif. VM 102 menggunakan CPU sekitar 2-4% pada kondisi normal dengan spike sekitar 66%, sedangkan memori meningkat dari sekitar 2,5 GiB menjadi 3,2 GiB dari alokasi 10 GiB. Network traffic VM memiliki spike incoming sekitar 4,3 MB dan disk I/O memperlihatkan burst read sekitar 41 MB serta write sekitar 33 MB.



GAMBAR 13. Metrik host server3 dan VM 103 (ubuntu-server3), periode 24-31 Juli 2026.

Server3 umumnya memakai CPU sekitar 3-10%, tetapi memiliki spike besar sekitar 85-88% pada 28 Juli dengan load average mendekati 3,5. Memori host meningkat dari sekitar 3 GiB menjadi sekitar 8,4 GiB. VM 103 menunjukkan pola CPU yang sama dengan spike sekitar 85%, dan penggunaan memori meningkat hingga sekitar 5,8 GiB dari alokasi 10 GiB. Network traffic mencapai sekitar 2,4 MB incoming, sedangkan disk I/O memiliki burst write sekitar 22 MB.

TABEL 7. Hasil pengukuran Server

Pasangan	CPU utama	RAM akhir teramati	Traffic/Disk	Interpretasi
server1 / VM101	Host spike $\pm 47\%$; VM $\pm 63\%$	Host $\pm 11/12$ GiB; VM $\pm 9,3/10$ GiB	Traffic $\pm 4,2$ MB; write ± 30 MB	Tekanan memori tertinggi
server2 / VM102	Host spike $\pm 35\%$; VM $\pm 66\%$	Host $\pm 5,5/12$ GiB; VM $\pm 3,2/10$ GiB	Traffic $\pm 4,3$ MB; read ± 41 MB	Resource paling longgar
server3 / VM103	Host/VM spike ± 85 - 88%	Host $\pm 8,4/12$ GiB; VM $\pm 5,8/10$ GiB	Traffic $\pm 2,4$ MB; write ± 22 MB	Spike CPU tertinggi

Bukti ini menguatkan bahwa resource node dipantau pada tanggal eksperimen, tetapi grafik menggunakan agregasi mingguan dan tampilan maksimum. Karena tidak tersedia ekspor numerik dengan timestamp setiap run k6, spike tertentu belum dapat diatribusikan secara pasti kepada Cloudflare Tunnel, FRP, atau tahapan RPS tertentu. Untuk menghitung overhead tunnel, penelitian tetap memerlukan ekspor time-series pada jendela setiap run serta pemisahan metrik proses/pod cloudflared, frpc, frps, ingress, dan aplikasi.

3.8. Validasi Keamanan



GAMBAR 14. Hasil Nmap TCP connect scan port 1-1000 pada endpoint FRP (kiri) dan Cloudflare (kanan).

Pemindaian berizin dilakukan pada 31 Juli 2026 pukul 06.30 WIB menggunakan Nmap 7.99 dengan perintah `nmap -Pn -sT -sV -p 1-1000`. Endpoint FRP mengarah ke 43.134.177.115 dan memperlihatkan port 22/SSH, 80/HTTP, serta 443/HTTPS terbuka; 997 port lain berstatus filtered. Nmap juga dapat mengidentifikasi OpenSSH 9.6p1 dan nginx 1.24.0. Temuan ini menunjukkan bahwa alamat server FRP dan layanan SSH-nya terlihat dari internet.

Endpoint Cloudflare dipindai pada alamat edge 104.21.87.108, dengan alamat edge lain juga terdeteksi. Port yang terlihat adalah 53/TCP, 80/HTTP, dan 443/HTTPS; 997 port lainnya filtered. Port 80 dan 443 dikenali sebagai Cloudflare HTTP proxy. Respons port 53 adalah NOTIMP, sehingga temuan tersebut tidak menunjukkan layanan DNS origin. Karena pemindaian mengenai edge Cloudflare, hasilnya mendukung penyembunyian origin dan tidak membuktikan keadaan firewall origin secara langsung.

Secara arsitektural, Cloudflare Tunnel mengurangi eksposur karena koneksi dibentuk outbound dan tidak mengharuskan port inbound langsung ke node. Namun, hasil Nmap ini belum membuktikan mitigasi DDoS atau bahwa seluruh trafik berbahaya dihentikan sebelum mencapai aplikasi. Klaim tersebut masih memerlukan korelasi Cloudflare Security Events/WAF, firewall, ingress, dan log aplikasi.

TABEL 7. FRP, Cloudflare Tunnel dan Status

Bukti	FRP	Cloudflare Tunnel	Status
Port scan eksternal	22, 80, 443 terbuka; IP server terlihat	53, 80, 443 pada edge; origin tidak terlihat	Tersedia
Log firewall/perimeter	[lampirkan hit]	[lampirkan hit/drop]	Belum tersedia
Cloudflare Security Events/WAF	Tidak berlaku	[lampirkan event/action]	Belum tersedia
Log ingress/aplikasi	[lampirkan request]	[buktikan request serangan tidak tiba]	Belum tersedia

4. KESIMPULAN

Berdasarkan hasil analisis, implementasi, dan pengujian komparatif, Cloudflare Tunnel dan FRP sama-sama dapat menyediakan akses publik menuju layanan pada klaster K3s on-premise, tetapi memiliki karakteristik yang berbeda. FRP menghasilkan latency p95 lebih rendah pada target 100–250 RPS. Pada target 500 RPS, kinerja FRP mulai menurun, sedangkan Cloudflare Tunnel menunjukkan tingkat keberhasilan yang lebih tinggi. Pada target 1.000 RPS, FRP mencatat success rate 68,497–70,727%, sedangkan Cloudflare Tunnel mempertahankan 98,555–99,928% pada request yang sempat dieksekusi, meskipun latency p95 meningkat menjadi 794–3.982 ms dan terdapat 5.429–8.674 dropped iterations. Karena actual RPS belum tercatat, hasil tersebut membuktikan ketahanan relatif Cloudflare Tunnel pada target beban tinggi, tetapi belum membuktikan bahwa seluruh target 1.000 request/detik benar-benar berhasil dikirim.

Dari aspek keterjangkauan, kedua endpoint mencatat uptime ping 100% dalam 24 jam dan 99,94% dalam 30 hari, dengan rata-rata ping 33 ms pada Cloudflare dan 32 ms pada FRP; perbedaan ini tidak menunjukkan keunggulan availability yang berarti. Dari aspek arsitektur dan eksposur, Cloudflare Tunnel membangun koneksi outbound-only, menampilkan alamat edge pada DNS publik, dan menyembunyikan origin, sedangkan FRP menampilkan alamat server publik dengan port 22, 80, dan 443 terbuka. Temuan Nmap mendukung perbedaan eksposur tersebut, tetapi tidak membuktikan mitigasi DDoS tanpa korelasi log keamanan. Dengan demikian, Cloudflare Tunnel lebih sesuai ketika ketahanan pada target beban tinggi, penyembunyian origin, dan pengelolaan akses terpusat menjadi prioritas. FRP tetap relevan ketika latency pada beban rendah dan kontrol penuh atas jalur data lebih diutamakan, dengan syarat TLS, autentikasi, pembatasan port, serta firewall diterapkan. Penelitian selanjutnya perlu mencatat actual RPS, melakukan minimal tiga pengulangan berdurasi lebih panjang, menyinkronkan metrik resource dengan setiap run, dan menambahkan baseline langsung tanpa tunnel agar kontribusi setiap komponen dapat diukur secara lebih kuat

REFERENSI

- [1] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, “Borg, Omega, and Kubernetes: Lessons learned from three container-management systems over a decade,” *Queue*, vol. 14, no. 1, pp. 70–93, 2016, doi: 10.1145/2898442.2898444.
- [2] K3s Project, “K3s—Lightweight Kubernetes,” K3s Documentation, 2026, accessed Jul. 13, 2026.
- [3] Kubernetes Authors, “kubectl port-forward,” Kubernetes Documentation, 2026, accessed Jul. 13, 2026.
- [4] ngrok, “Free Plan Limits,” ngrok Documentation, 2026, accessed Jul. 13, 2026.
- [5] A. Alanda, H. A. Mooduto, H. Amnur and M. Fadhel, "Scaling Kubernetes Architectures for High Availability in Cloud," 2025 International Conference on Computer Sciences, Engineering, and Technology Innovation (ICoCSETI), Jakarta, Indonesia, 2025, pp. 818-823, doi: 10.1109/ICoCSETI63724.2025.11020084.
- [6] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, Zero Trust Architecture, NIST Special Publication 800-207. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2020, doi: 10.6028/NIST.SP.800-207.
- [7] Cloudflare, “Cloudflare Tunnel,” Cloudflare One Documentation, 2026, accessed Jul. 13, 2026.
- [8] Kubernetes Authors, “DNS for Services and Pods,” Kubernetes Documentation, 2026, accessed Jul. 13, 2026.

- [9] Cloudflare, “Kubernetes Deployment Guide for Cloudflare Tunnel,” Cloudflare One Documentation, 2026, accessed Jul. 13, 2026.
- [10] Cloudflare, “Access Policies,” Cloudflare One Documentation, 2026, accessed Jul. 13, 2026.
- [11] J. E. Goldman and P. T. Rawles, *Applied Data Communications: A Business-Oriented Approach*, 4th ed. Hoboken, NJ, USA: John Wiley & Sons, 2004.
- [12] I. Čilić, P. Krivić, I. Podnar Žarko, and M. Kušek, “Performance evaluation of container orchestration tools in edge computing environments,” *Sensors*, vol. 23, no. 8, Art. no. 4008, 2023, doi: 10.3390/s23084008.
- [13] V. Kjorveziroski and S. Filiposka, “Kubernetes distributions for the edge: Serverless performance evaluation,” *The Journal of Supercomputing*, vol. 78, no. 11, pp. 13728–13755, 2022, doi: 10.1007/s11227-022-04430-6.
- [14] D. Rahman, H. Amnur, and I. Rahmayuni, “Monitoring Server dengan Prometheus dan Grafana serta Notifikasi Telegram”, *jitsi*, vol. 1, no. 4, pp. 133 - 138, Dec. 2020.
- [15] D. Yakubov and D. Hästbacka, “Comparative analysis of lightweight Kubernetes distributions for edge computing: Performance and resource efficiency,” *arXiv preprint arXiv:2504.03656*, 2025.
- [16] A. Valantis, N. Makris, and T. Korakis, “Orchestration software for resource constrained datacenters: An experimental evaluation,” in *Proc. 2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*, Milan, Italy, 2022, pp. 121–126, doi: 10.1109/NetSoft54395.2022.9844043