

LAPORAN AKHIR PENELITIAN

RINGKASAN

Baik perusahaan berskala besar seperti Amazon sampai dengan UKM telah menggunakan arsitektur *microservice* untuk meningkatkan skalabilitasnya. Mereka juga mempertimbangkan pembiayaan untuk mengembangkan aplikasinya. Penelitian ini akan membahas proses migrasi aplikasi E-rapor dari arsitektur monolitik ke arsitektur *microservice* untuk meningkatkan kinerja melayani user lebih banyak dengan waktu lebih cepat dan mengurangi biaya sumber daya. Akan dihitung pula besarnya waktu proses migrasinya. Selain proses migrasi, penyesuaian kode dan peningkatan kinerja database juga dipertimbangkan.

Pada penelitian ini ditingkatkan pelayanan kepada pengguna dengan mengoptimalkan dari segi arsitektur *software*. Aplikasi diubah arsitekturnya dari monolitik ke *microservice* karena ketika menggunakan arsitektur monolitik, skalabilitas hanya mengandalkan upgrade pada server yang besar dan membutuhkan biaya yang banyak dibandingkan *microservice* yang mengandalkan pengoptimalan dari sisi *software* dengan *service worker* mengacu pada pengelolaan server yang lebih baik.

Hasil pengujian *stress test* dari arsitektur *microservice* mampu menangani 90 pengguna dalam satu waktu dibanding menggunakan arsitektur monolitik yang hanya 40 pengguna dalam satu waktu. Penelitian ini berada dalam tingkat ketersiapan teknologi level 7, demonstrasi prototipe sistem telah di uji coba di SD Muhammadiyah Ambarketawang 3.

Kata kunci: *database; microservice; monolitik; upgrade server; worker*

HASIL PELAKSANAAN PENELITIAN

Dalam penelitian ini didapatkan hasil dari beberapa skenario percobaan untuk membandingkan arsitektur monolitik dan *microservice* yang akan dijelaskan di bawah ini.

Skenario Pengujian 1 - Arsitektur monolitik

Menggunakan konfigurasi semua servis (reverse proxy, php interpreter, database server) berjalan di satu server.

- Single Virtual server dengan spesifikasi:
- vCPU 1 unit
- RAM 2048MB
- SSD 55GB
- Sistem Operasi Ubuntu Server 20.04

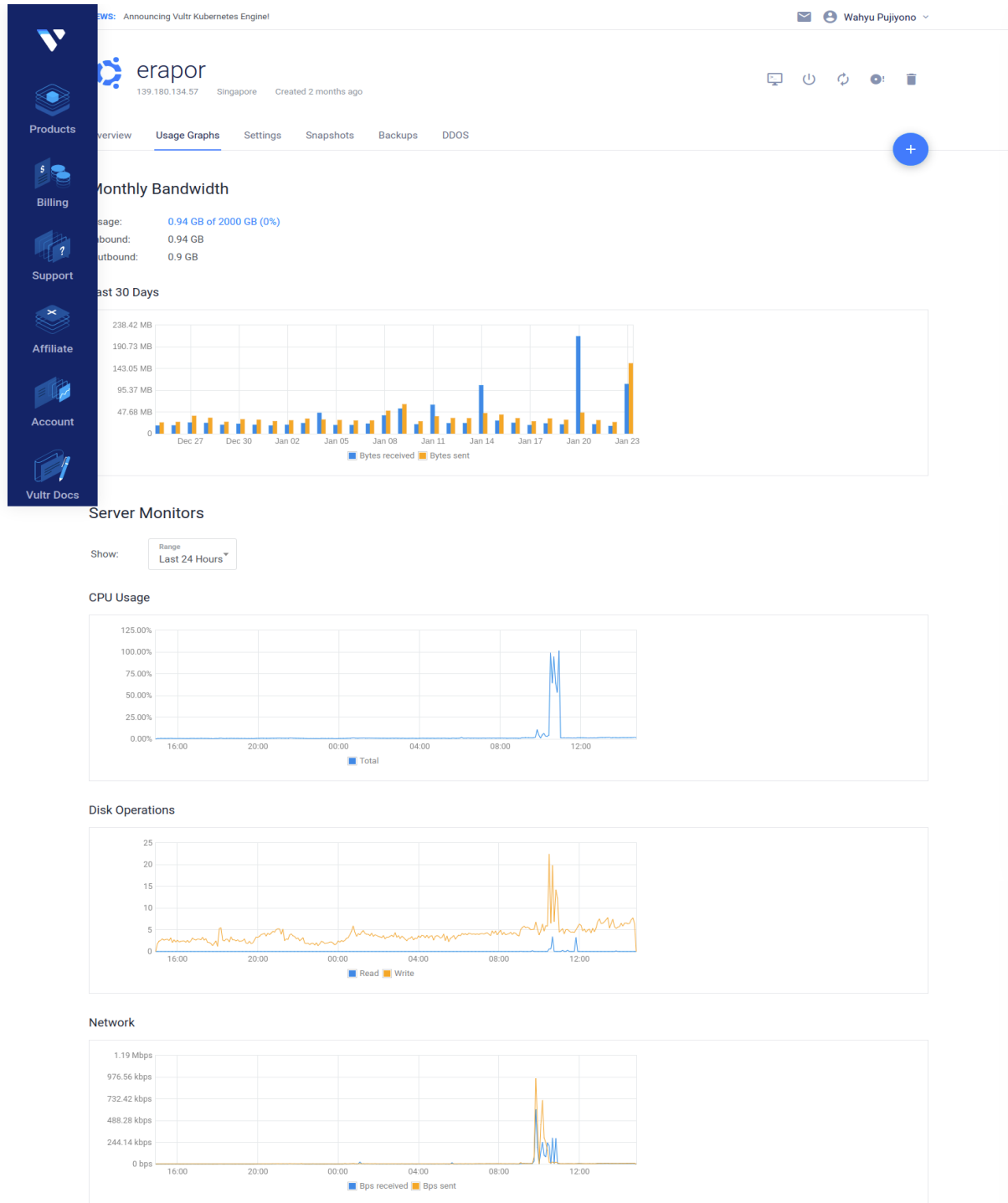
Stack yang digunakan:

- Reverse proxy NGINX
- PHP interpreter php7.2-fpm
- Database server MariaDB versi 10.3.25-MariaDB-0ubuntu1

Konfigurasi PHP:

- `memory_limit = 1024M`
- `post_max_size = 8M`
- `upload_max_filesize = 1024M`
- `max_execution_time = 1500`
- `max_input_time = 60`

Pengujian menggunakan skenario disajikan pada gambar 1 berikut:



Gambar 1. Pengujian skenario 1 dari sisi server

Tabel 1. Hasil pengujian skenario 1 dari sisi user

Jumlah CPU	Jumlah Thread User	Berhasil Upload	Gagal Upload
1 CPU	15 thread user	100%	0%
1 CPU	20 thread user	75%	25%

Penjelasan:

- Dari 15 user 100% berhasil melakukan request ke server
- Dari 20 user 75% berhasil melakukan request ke server dan 25% gagal

Skenario Pengujian 2 - Arsitektur monolitik dengan multi server

Menggunakan konfigurasi multi server

- Server untuk Load balancer

Spesifikasi:

- vCPU 1 unit
- RAM 512MB
- SSD 10GB
- Sistem Operasi Ubuntu Server 20.04

Stack yang digunakan:

- Load Balancer (round robin) NGINX

- PHP interpreter

Spesifikasi:

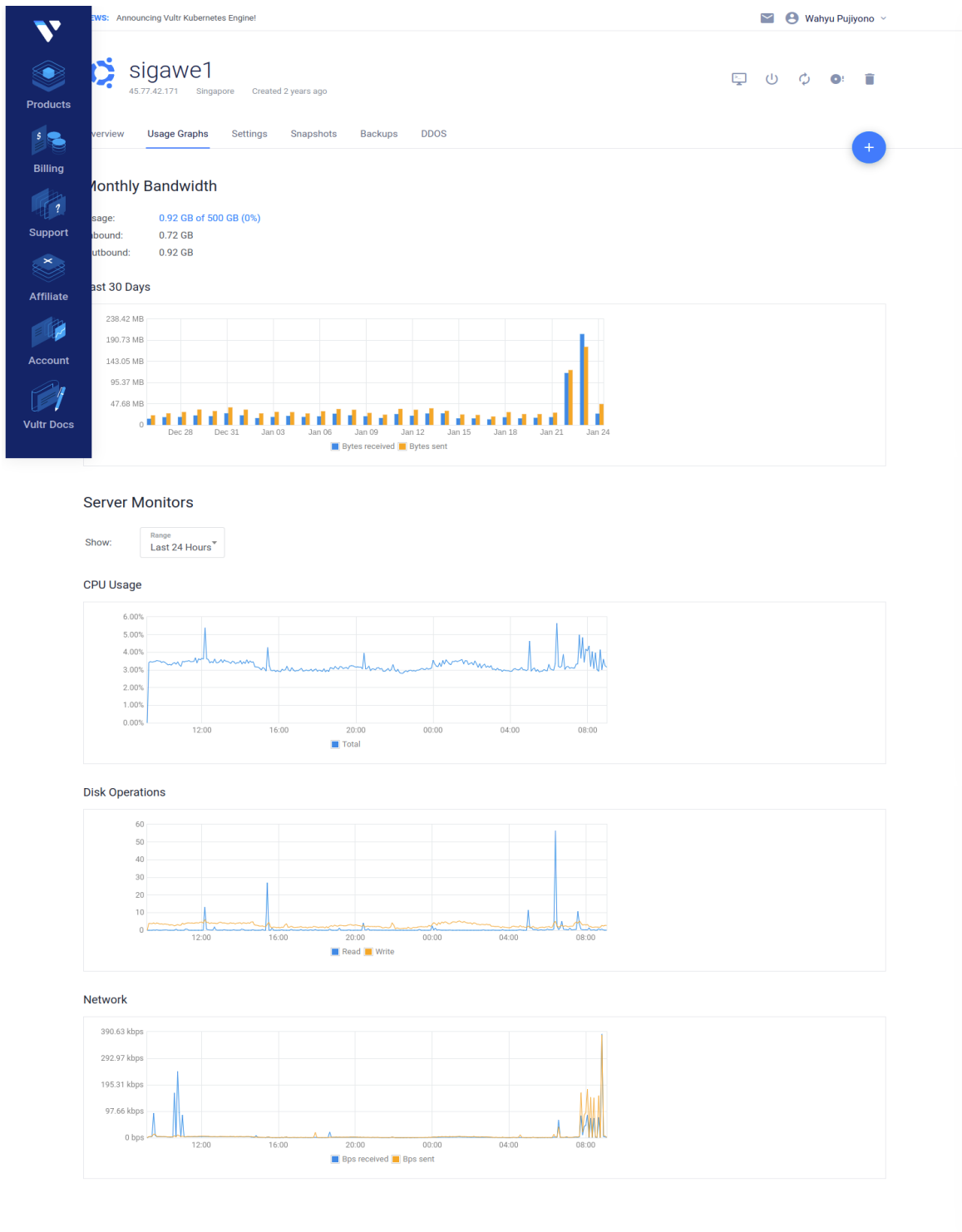
- vCPU 1 unit
- RAM 2048MB
- SSD 55GB
- Sistem Operasi Ubuntu Server 20.04

Konfigurasi PHP:

- memory_limit = 1500M
- post_max_size = 1500M

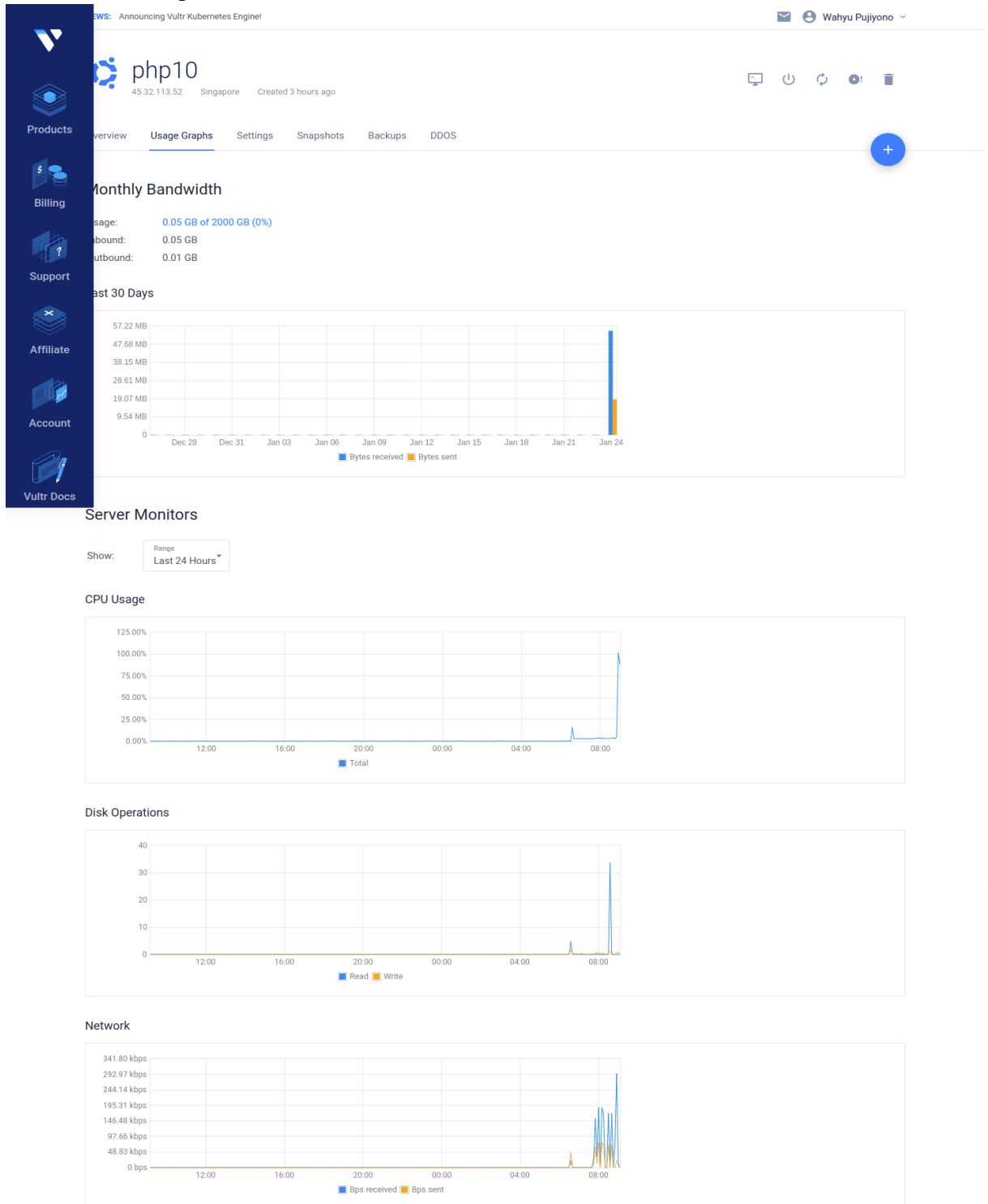
- upload_max_filesize = 1024M
 - max_execution_time = 1500
 - max_input_time = 1500
-
- Database server
- Spesifikasi:
- vCPU 1 unit
 - RAM 1024MB
 - SSD 25GB
 - Sistem Operasi Ubuntu Server 20.04

- **Server Load Balancer**



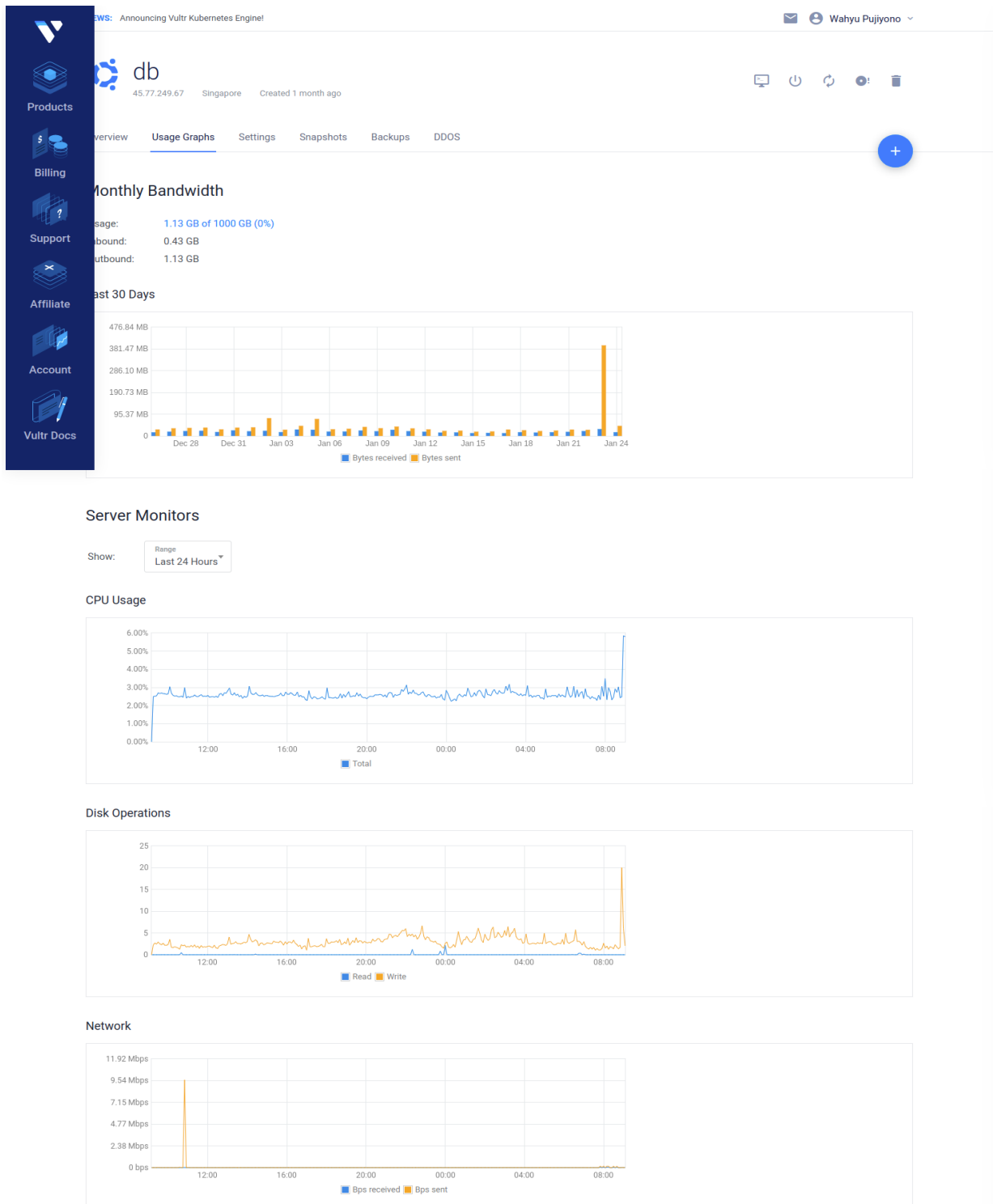
Gambar 2. Pengujian skenario 2 dari sisi Server Load Balancer

- Server PHP Interpreter



Gambar 3. Pengujian skenario 2 dari sisi Server PHP Interpreter

- Server Database



Gambar 4. Pengujian skenario 2 dari sisi Server Database

Tabel 2. Hasil pengujian skenario 2 dari sisi user

Jumlah CPU	Jumlah Thread User	Berhasil Upload	Gagal Upload
1 CPU	15 thread user	100%	0%
1 CPU	20 thread user	75%	25%

Penjelasan:

- Dari 15 user 100% berhasil melakukan request ke server
- Dari 20 user 75% berhasil melakukan request ke server dan 25% gagal

Skenario Pengujian 3 - Arsitektur monolitik dengan multi server yang ditingkatkan

Menggunakan konfigurasi multi server

- Server untuk Load balancer

Spesifikasi:

- vCPU 1 unit
- RAM 512MB
- SSD 10GB
- Sistem Operasi Ubuntu Server 20.04

Stack yang digunakan:

- Load Balancer (round robin) NGINX

- PHP interpreter

Spesifikasi:

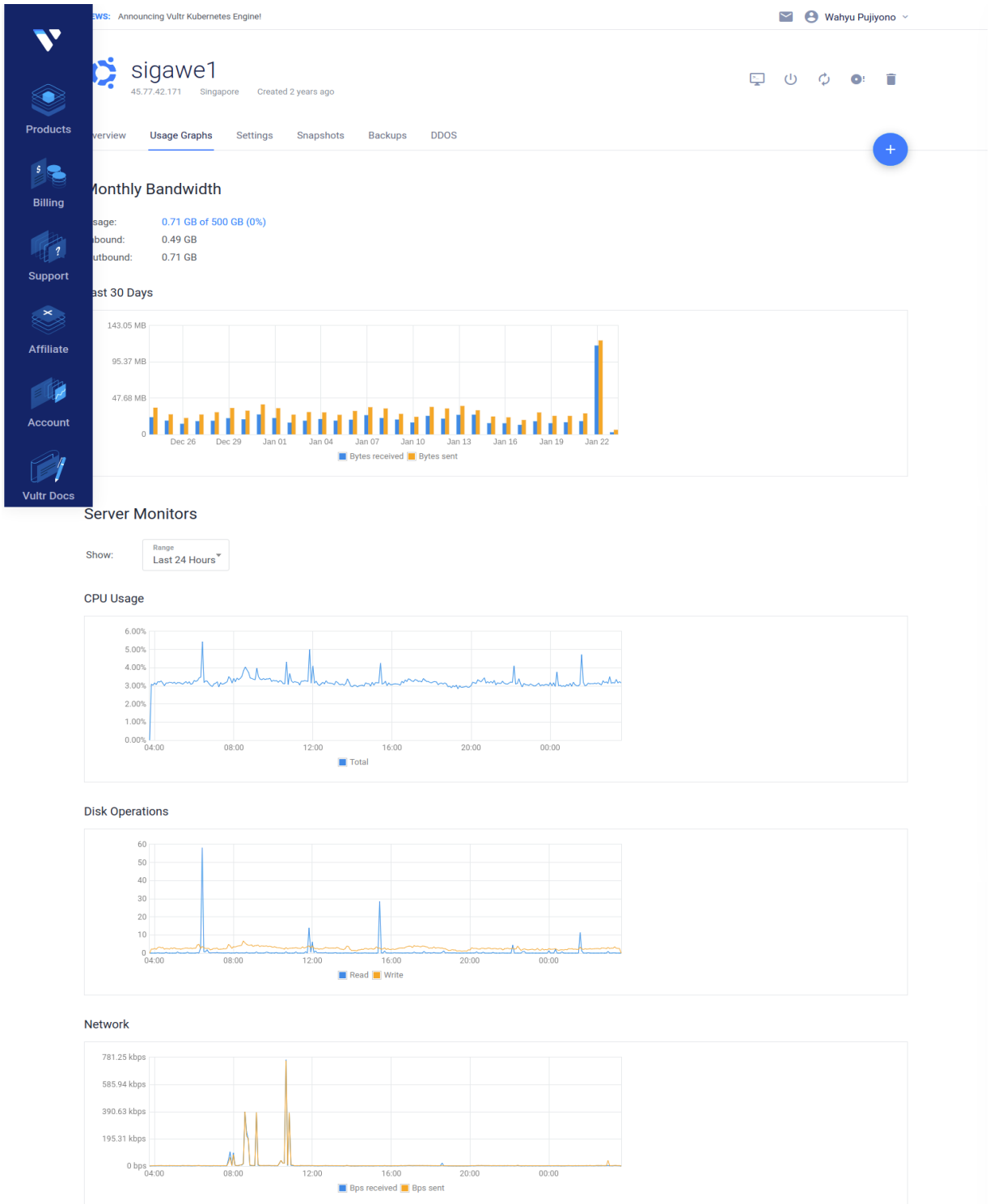
- vCPU 2 unit
- RAM 4096MB
- SSD 80GB
- Sistem Operasi Ubuntu Server 20.04

Konfigurasi PHP:

- memory_limit = 3072M
- post_max_size = 3072M

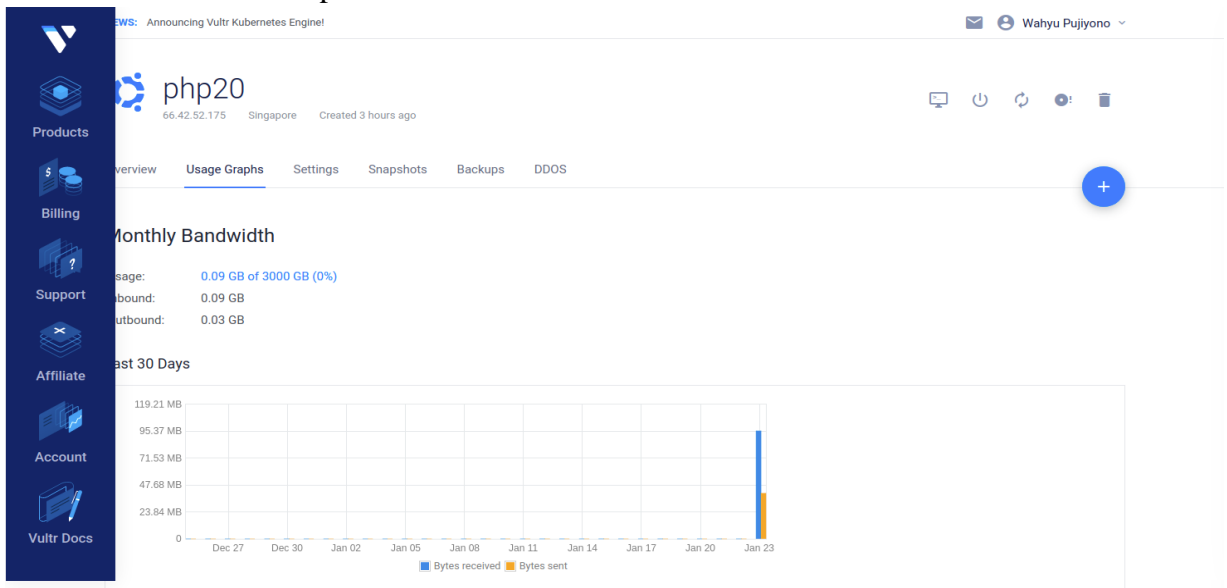
- upload_max_filesize = 2048M
 - max_execution_time = 1500
 - max_input_time = 1500
-
- Database server
- Spesifikasi:
- vCPU 1 unit
 - RAM 1024MB
 - SSD 25GB
 - Sistem Operasi Ubuntu Server 20.04

- Server Load Balancer



Gambar 5. Pengujian skenario 3 dari sisi Server Load Balancer

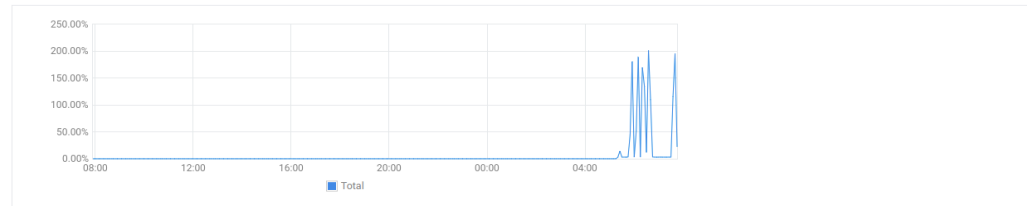
● Server PHP Interpreter



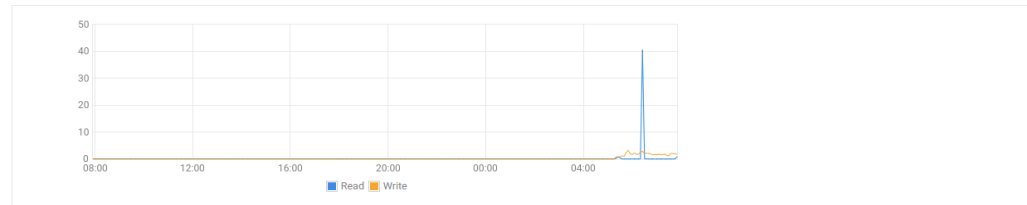
Server Monitors

Show: Last 24 Hours

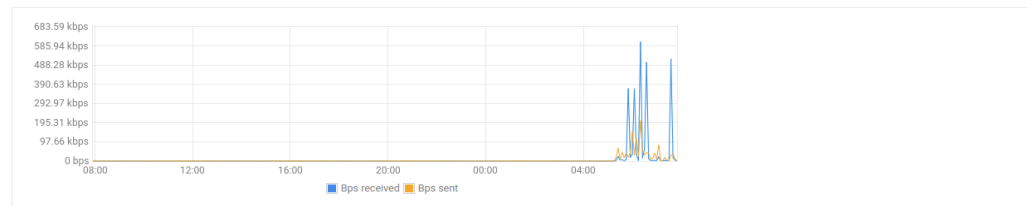
CPU Usage



Disk Operations

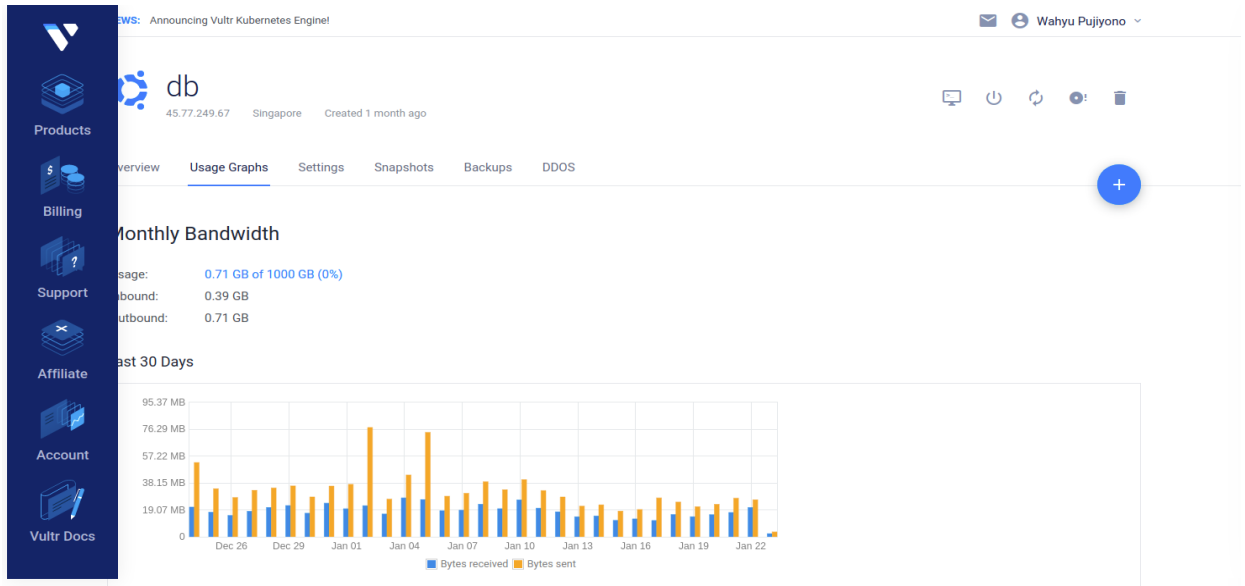


Network



Gambar 6. Pengujian skenario 3 dari sisi Server PHP Interpreter

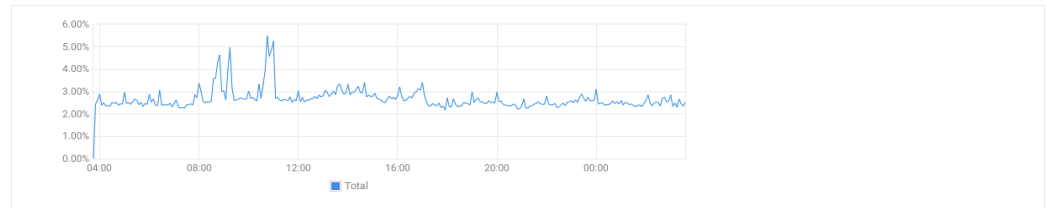
Server Database



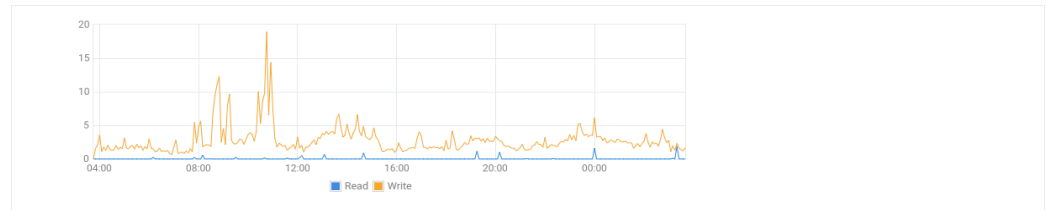
Server Monitors

Show:

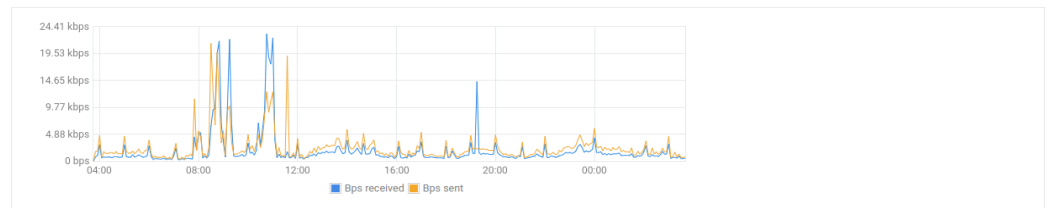
CPU Usage



Disk Operations



Network



Gambar 7. Pengujian skenario 3 dari sisi Server Database

Tabel 3. Hasil pengujian skenario 3 dari sisi user

Jumlah CPU	Jumlah Thread User	Berhasil Upload	Gagal Upload
2 CPU	20 thread user	100%	0%
2 CPU	30 thread user	83.33%	16.67%

Penjelasan:

- Dari 20 user 100% berhasil melakukan request ke server
- Dari 30 user 83.33% berhasil melakukan request ke server dan 16.67% gagal

Skenario Pengujian 4 - Arsitektur monolitik dengan multi server yang ditingkatkan

Menggunakan konfigurasi multi server

- Server untuk Load balancer

Spesifikasi:

- vCPU 1 unit
- RAM 512MB
- SSD 10GB
- Sistem Operasi Ubuntu Server 20.04

Stack yang digunakan:

- Load Balancer (round robin) NGINX

- PHP interpreter

Spesifikasi:

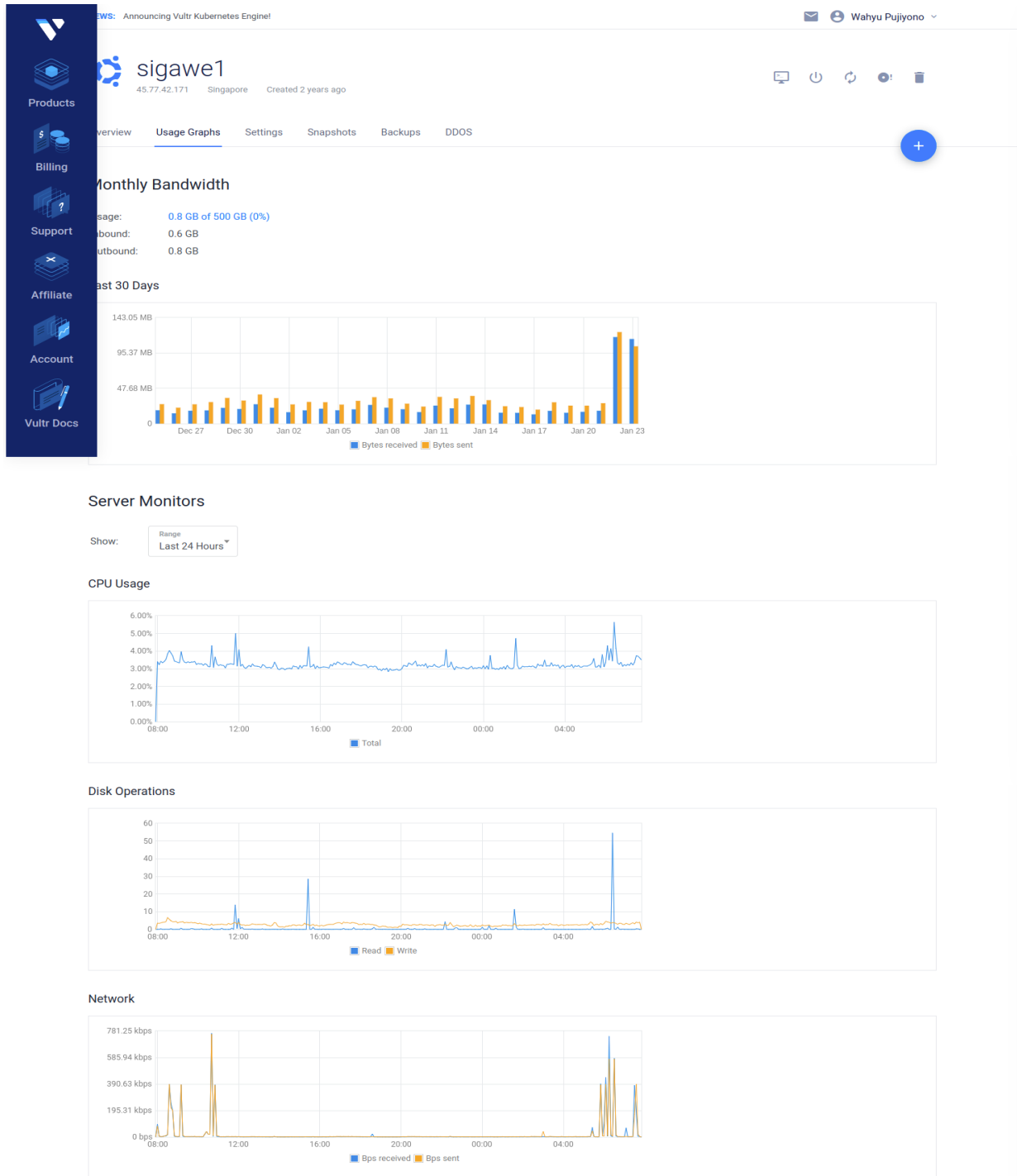
- vCPU 4 unit
- RAM 8192MB
- SSD 160GB
- Sistem Operasi Ubuntu Server 20.04

Konfigurasi PHP:

- memory_limit = 6144M
- post_max_size = 6144M

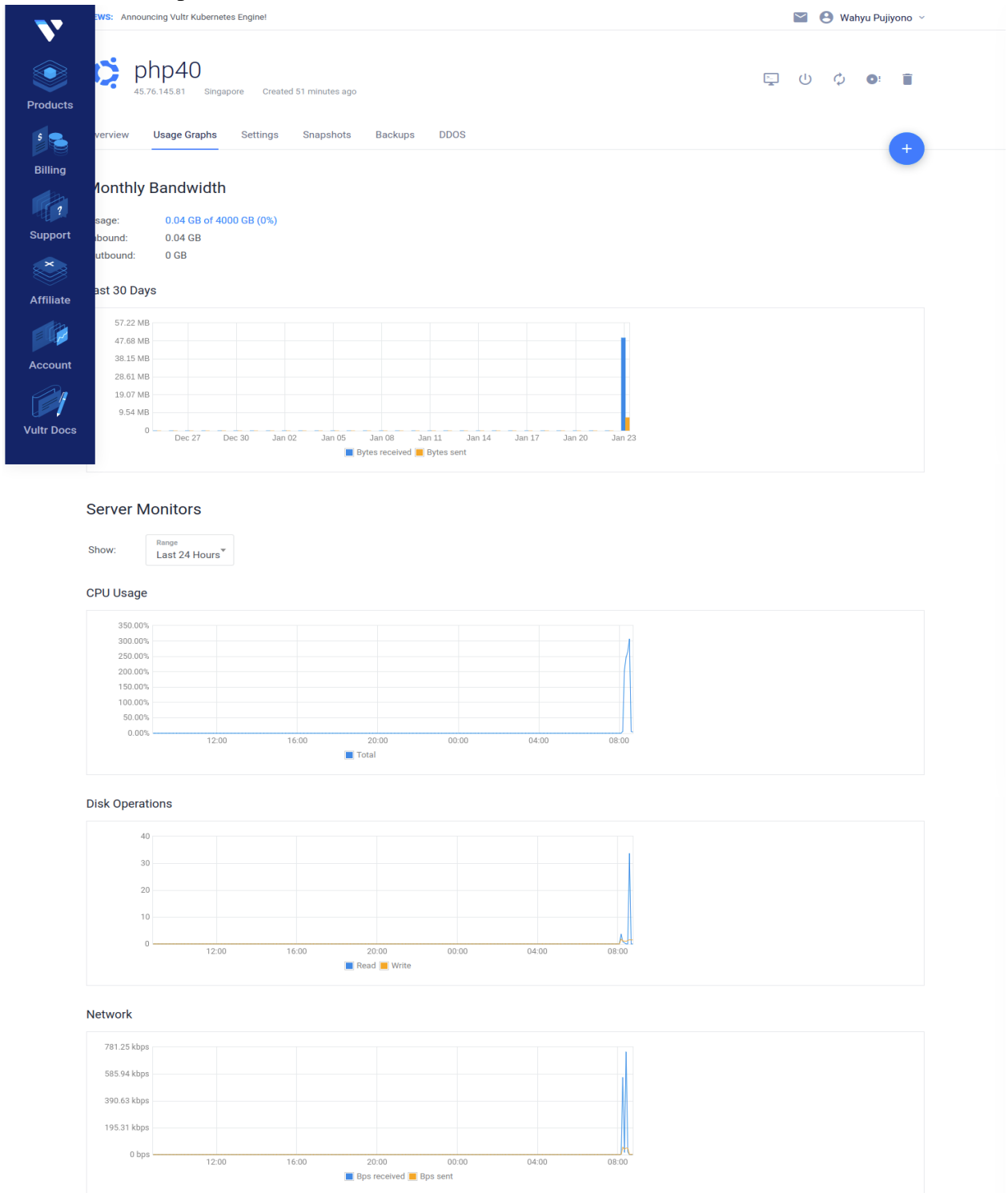
- upload_max_filesize = 6144M
 - max_execution_time = 1500
 - max_input_time = 1500
-
- Database server
- Spesifikasi:
- vCPU 1 unit
 - RAM 1024MB
 - SSD 25GB
 - Sistem Operasi Ubuntu Server 20.04

- Server Load Balancer



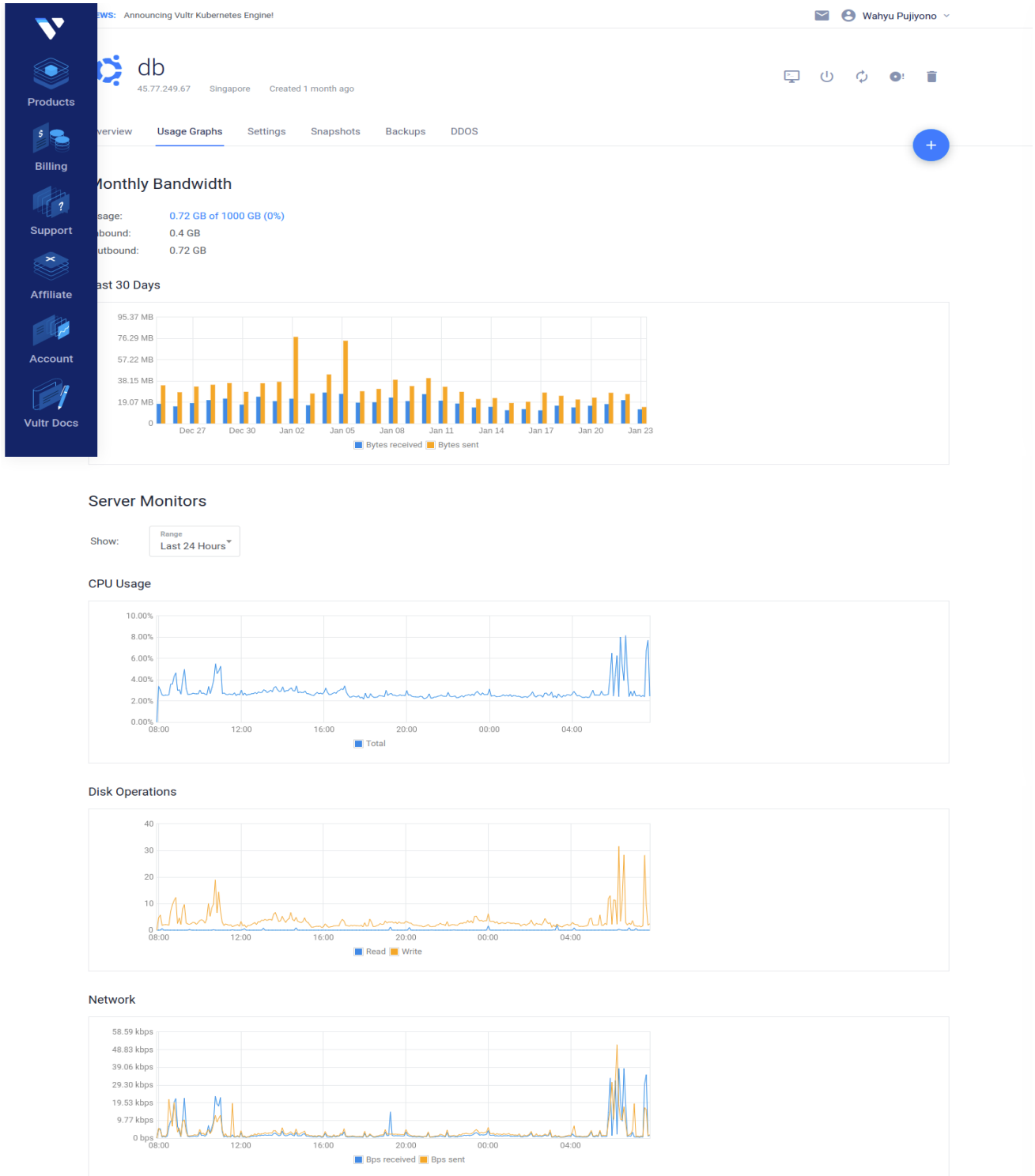
Gambar 8. Pengujian skenario 4 dari sisi Server Load Balancer

- Server PHP Interpreter



Gambar 9. Pengujian skenario 4 dari sisi Server PHP Interpreter

● Server Database



Gambar 10. Pengujian skenario 4 dari sisi Server Database

Tabel 4. Hasil pengujian skenario 4 dari sisi user

Jumlah CPU	Jumlah Thread User	Berhasil Upload	Gagal Upload
4 CPU	30 thread user	100%	0%
4 CPU	40 thread user	92.5%	7.5%

Penjelasan:

- Dari 30 user 100% berhasil melakukan request ke server
- Dari 40 user 92.5% berhasil melakukan request ke server dan 7.5% gagal

Skenario Pengujian 5 - Arsitektur microservice

Menggunakan arsitektur microservice

- Server untuk Load balancer

Spesifikasi:

- vCPU 1 unit
- RAM 2048MB
- SSD 55GB
- Sistem Operasi Ubuntu Server 20.04

Stack yang digunakan:

- Load Balancer (round robin) NGINX

- PHP interpreter (4 server)

Spesifikasi:

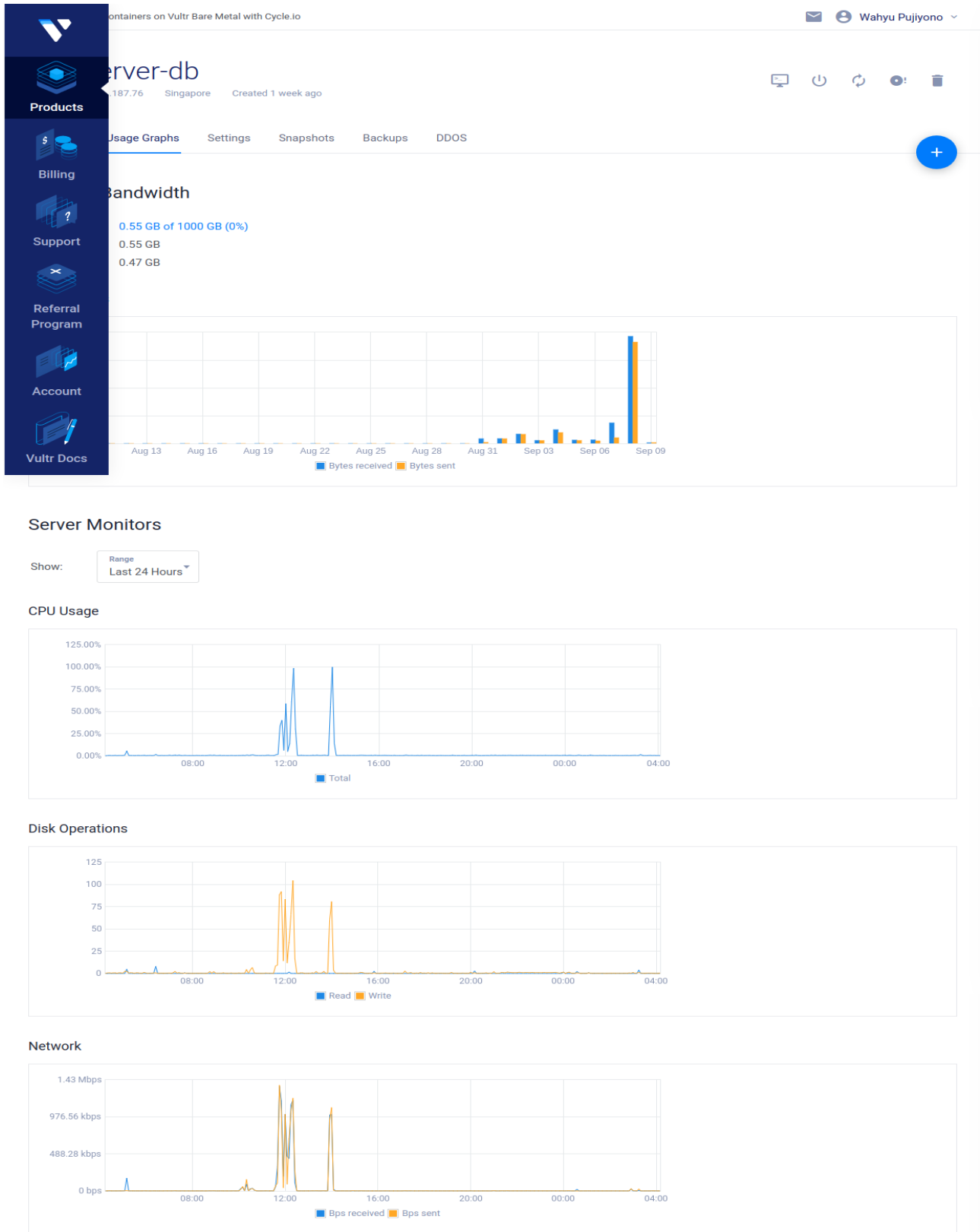
- vCPU 1 unit
- RAM 1024MB
- SSD 25GB
- Sistem Operasi Ubuntu Server 20.04

Konfigurasi PHP:

- memory_limit = 6144M
- post_max_size = 6144M
- upload_max_filesize = 6144M

- `max_execution_time = 1500`
 - `max_input_time = 1500`
-
- Database server
- Spesifikasi:
- vCPU 1 unit
 - RAM 2048MB
 - SSD 55GB
 - Sistem Operasi Ubuntu Server 20.04

• Server Database



Gambar 11. Pengujian skenario 5 dari sisi Server Database

Server PHP Interpreter

Containers on Vultr Bare Metal with Cycle.io

Wahyu Pujiono

server-php1

48.78.25 Singapore Created 1 week ago

Usage Graphs Settings Snapshots Backups DDOS

Bandwidth

0.48 GB of 1000 GB (0%)

0.48 GB

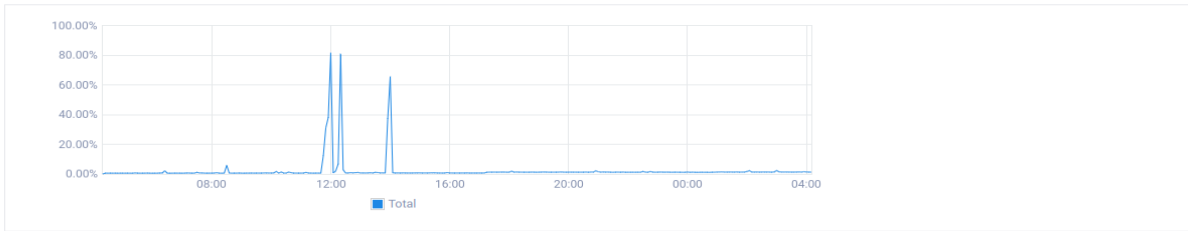
0.16 GB

Date	Bytes received	Bytes sent
Aug 31	~0.48 GB	~0.16 GB
Sep 03	~0.05 GB	~0.02 GB
Sep 06	~0.02 GB	~0.01 GB
Sep 09	~0.05 GB	~0.02 GB

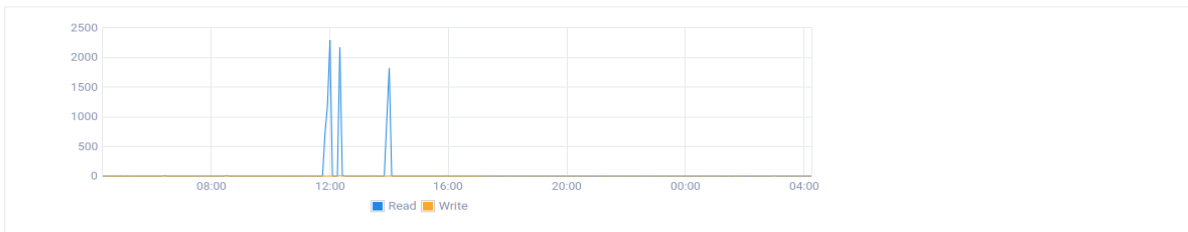
Server Monitors

Show: Last 24 Hours

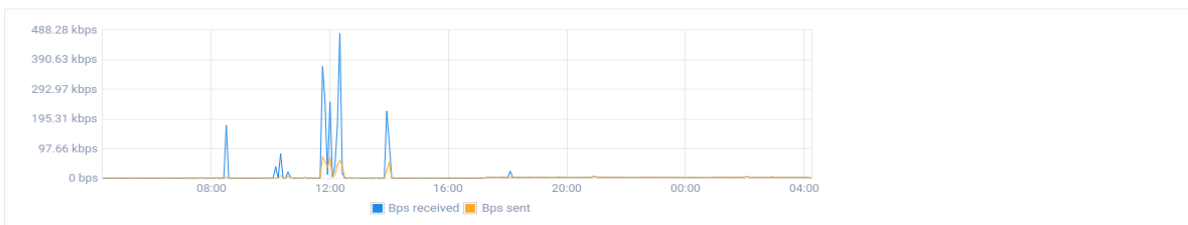
CPU Usage



Disk Operations



Network



- Products
- Billing
- Support
- Referral Program
- Account
- Vultr Docs

server-php2

192.209.107 Singapore Created 1 week ago



Usage Graphs Settings Snapshots Backups DDOS

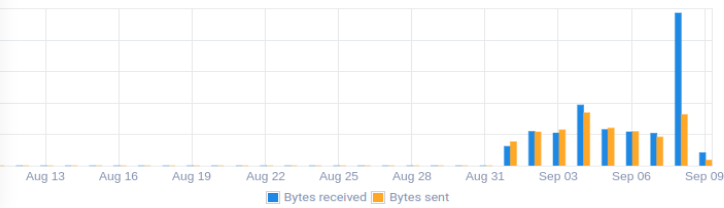


Bandwidth

0.3 GB of 1000 GB (0%)

0.3 GB

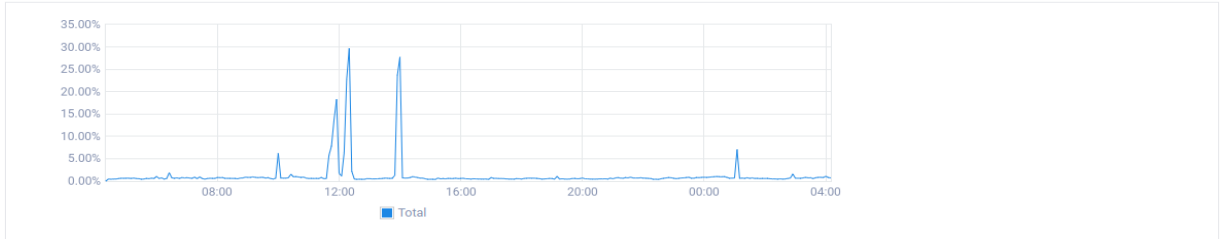
0.22 GB



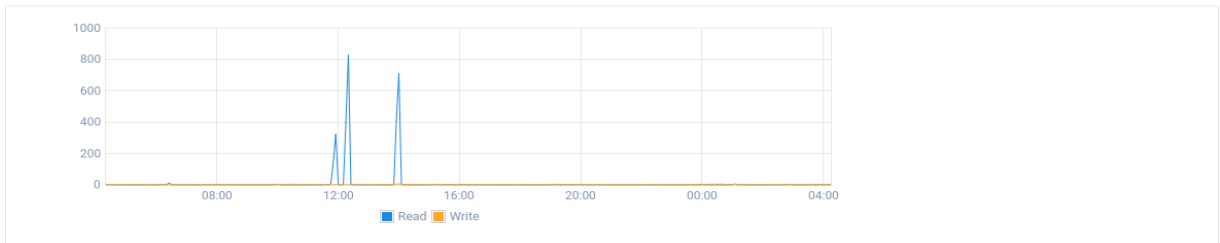
Server Monitors

Show: Range Last 24 Hours

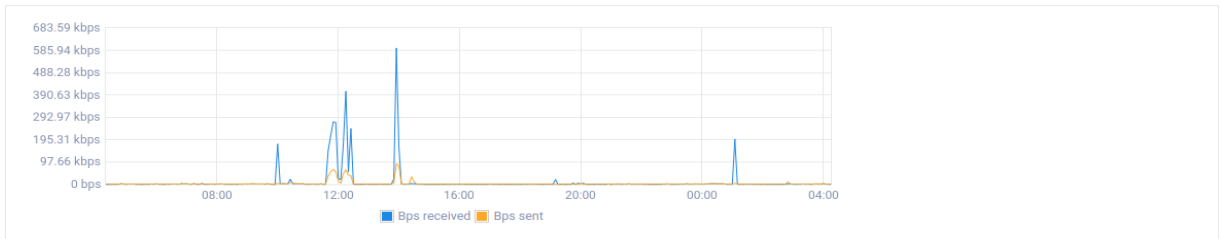
CPU Usage



Disk Operations



Network

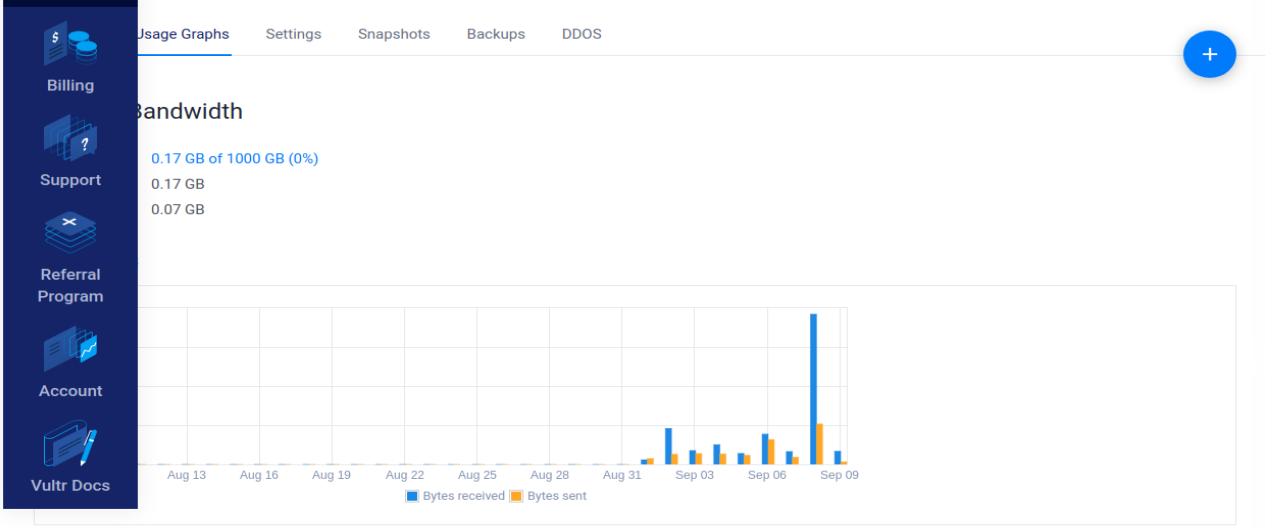


- Products
- Billing
- Support
- Referral Program
- Account
- Vultr Docs

server-php3

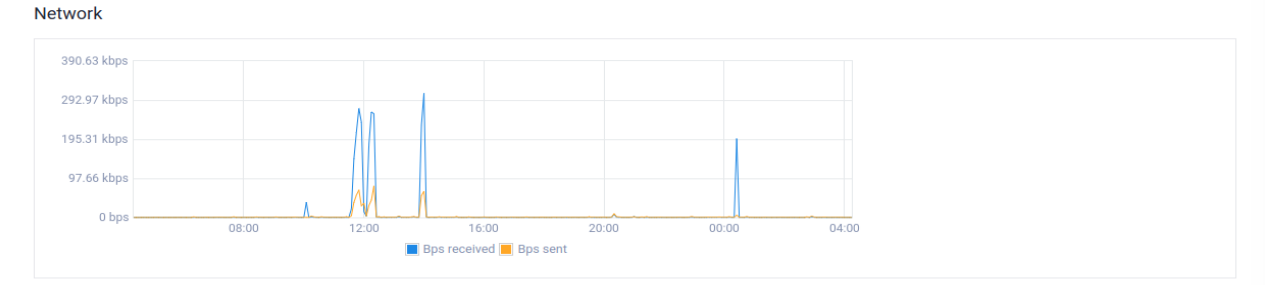
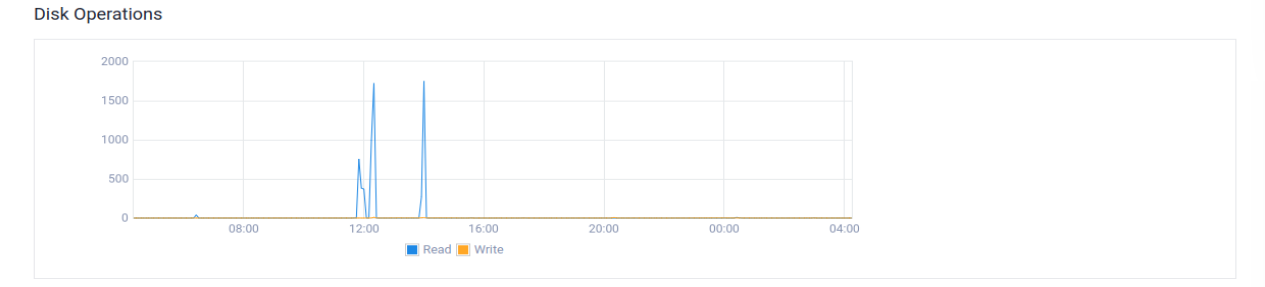
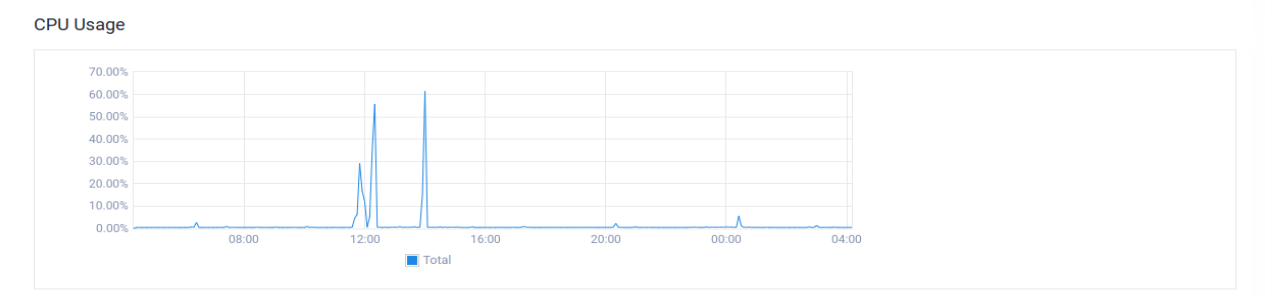
174.146 Singapore Created 1 week ago

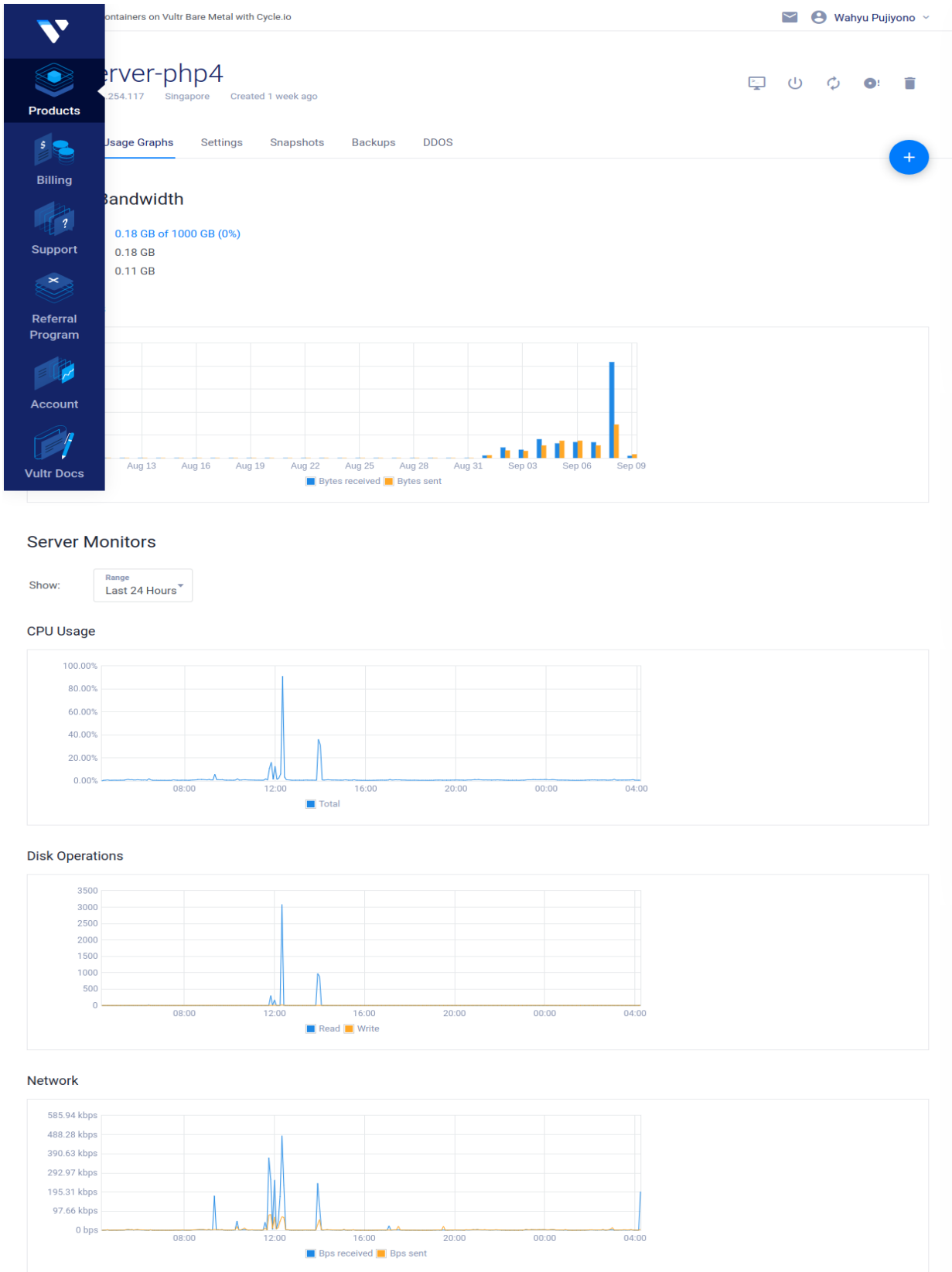
Usage Graphs Settings Snapshots Backups DDOS



Server Monitors

Show:





Gambar 12. Pengujian skenario 5 dari sisi Server PHP Interpreter

Tabel 5. Hasil pengujian skenario 5 dari sisi user

Jumlah CPU	Jumlah Thread User	Berhasil Upload	Gagal Upload
1 CPU	30 thread user	100%	0%
1 CPU	40 thread user	100%	0%
1 CPU	50 thread user	100%	0%
1 CPU	60 thread user	100%	0%
1 CPU	70 thread user	100%	0%
1 CPU	80 thread user	100%	0%
1 CPU	90 thread user	100%	0%
1 CPU	100 thread user	68%	32%

Penjelasan:

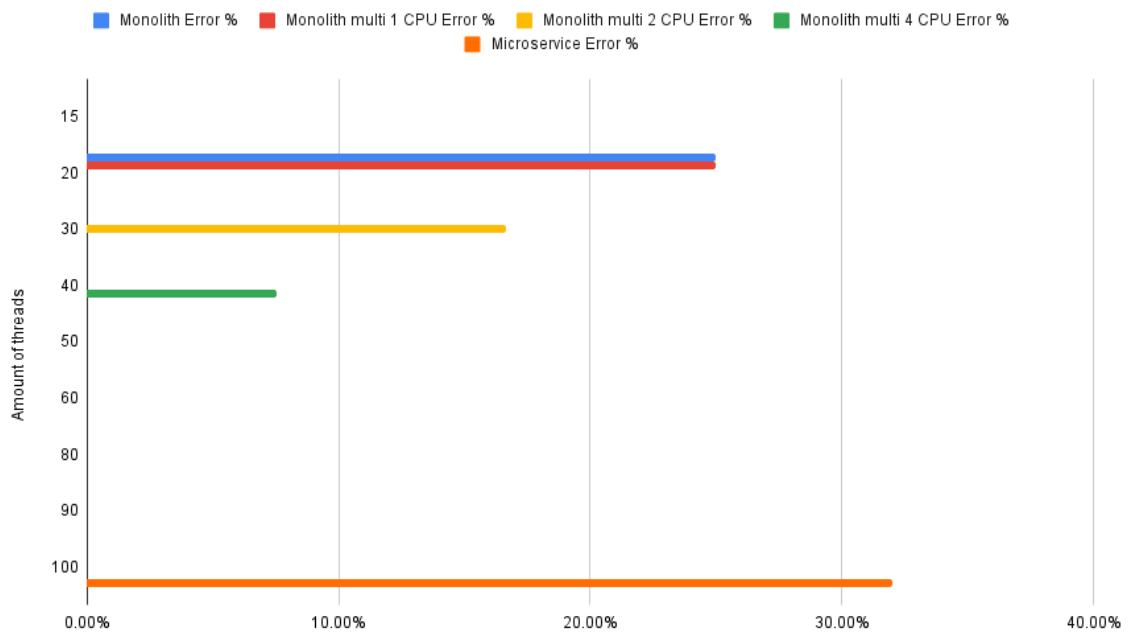
- Dari 30 user 100% berhasil melakukan request ke server
- Dari 40 user 100% berhasil melakukan request ke server
- Dari 50 user 100% berhasil melakukan request ke server
- Dari 60 user 100% berhasil melakukan request ke server
- Dari 70 user 100% berhasil melakukan request ke server
- Dari 80 user 100% berhasil melakukan request ke server
- Dari 90 user 100% berhasil melakukan request ke server
- Dari 100 user 68% berhasil melakukan request ke server dan 32% gagal

Dari hasil penelitian didapatkan perbandingan hasil antara arsitektur monolitik dan microservice sebagai berikut:

Tabel 6. Rangkuman pengujian Monolitik dan Microservice

Jumlah User	Monolith Error %	Monolith multi 1 CPU Error %	Monolith 2 CPU multi Error %	Monolith 4 CPU multi Error %	Microservice Error %
15	0.00%	0.00%			
20	25.00%	25.00%	0.00%		
30			16.67%	0.00%	0.00%
40				7.50%	0.00%
50					0.00%
60					0.00%
80					0.00%
90					0.00%
100					32.00%

Perbandingan Monolitik dan Microservice



Gambar 13. Perbandingan Monolitik dan Microservice

Tabel 7. Capaian hasil pelayanan server beserta dengan tingkat erromya

Amount of threads	Monolith Error %	Monolith Throughput (/min)	Monolith multi 1 CPU Error %	Monolith multi 1 CPU Throughput (/min)	Monolith multi 2 CPU Error %	Monolith multi 2 CPU Throughput (/min)	Monolith multi 4 CPU Error %	Monolith multi 4 CPU Throughput (/min)	Microservice Error %	Microservice Throughput (/min)
15	0.00%	1.8	0.00%	1.90	-	-	-	-	-	-
20	25.00%	2.3	25.00%	2.10	0.00%	3.40	-	-	-	-
30	-	-	-	-	16.67%	3.90	0.00%	5.10	0.00%	53.90
40	-	-	-	-	-	-	7.50%	5.30	0.00%	42.66
50	-	-	-	-	-	-	-	-	0.00%	51.46
60	-	-	-	-	-	-	-	-	0.00%	50.60
80	-	-	-	-	-	-	-	-	0.00%	43.79
90	-	-	-	-	-	-	-	-	0.00%	48.94
100	-	-	-	-	-	-	-	-	32.00%	42.03

Legend:

- testing tidak dilakukan karena sudah menunjukkan hasil yang signifikan

error > 0% performa tidak baik

Dilihat dari data diatas arsitektur microservice lebih unggul dari arsitektur monolitik dan monolitik yang telah ditingkatkan spesifikasi servernya. Arsitektur monolitik dengan spesifikasi tertinggi pada 4 CPU di skenario 4 hanya mencapai maksimal pada 40 user dengan kegagalan sebesar 7.50%. Untuk arsitektur microservice mencapai maksimal 100 user dengan kegagalan 32%. Jadi bisa dikatakan arsitektur microservice 2x lebih unggul daripada arsitektur monolitik.

HASIL PENGUJIAN ALPHA

Pengujian ini diikuti oleh 14 peserta dari guru SD Muhammadiyah Ambarketawang 3 dengan hasil rangkuman sebagai berikut:

Tabel 8. Hasil data kuisioner

No	Pertanyaan	Penilaian			
		SS	S	KS	TS
1	Apakah aplikasi mudah digunakan?	1	13		
2	Apakah tampilan aplikasi tersebut menarik?		14		
3	Apakah tampilan Sistem Rapor Sekolah mudah dipahami?		13	1	
4	Apakah fasilitas penginputan data - data yang terdapat pada Sistem Rapor Sekolah sudah sesuai?		12	1	1
5	Apakah data rapor sekolah bisa dicetak melalui aplikasi?	5	9		
6	Apakah data laporan rapor siswa sudah sesuai dengan hasil input rapor?		13	1	
7	Apakah Sistem Rapor Sekolah sangat membantu dalam menginputkan nilai akhir rapor?	2	12		
8	Apakah proses dalam Sistem Rapor Sekolah sesuai dengan kebutuhan guru?		12	1	1
	Jumlah	8	98	4	2

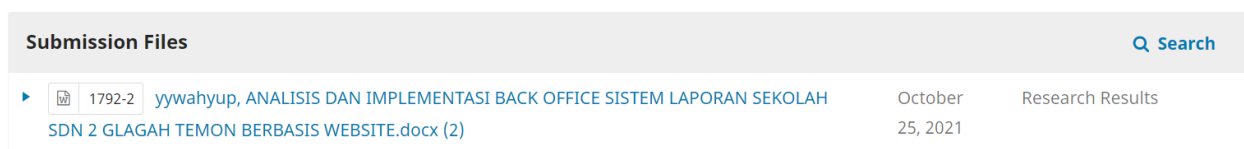
Data kuisioner dengan menggunakan cara pembagian kertas yang kemudian diisi, hasil menunjukkan guru – guru di SD Muhammadiyah Ambarketawang 3 setuju dengan aplikasi E-Rapor yang dijalankan mudah digunakan. Segi tampilan aplikasi E-Rapor tersebut menarik dan mudah dipahami bagi para guru yang akan menginputkan hasil nilai dari siswa ke dalam aplikasi. Para guru juga setuju bahwa fasilitas penginputan data - data yang terdapat pada Sistem Rapor Sekolah sudah sesuai dengan apa yang dibutuhkan dalam membuat hasil rapor nantinya yang akan dibagikan kepada wali siswa. Fasilitas cetak data rapor melalui aplikasi yang memudahkan nantinya dalam membagikan hasil data rapor dalam bentuk *soft file* yang kemudian dapat dicetak kapan saja oleh wali siswa.

Para guru setuju bahwa Sistem Rapor Sekolah yang disimulasikan dan dijalankan sangat membantu dalam proses penginputan nilai akhir pada rapor siswa dan dengan adanya Sistem Rapor Sekolah sudah sesuai dengan kebutuhan guru dalam memproses nilai akhir siswa dalam bentuk rapor nantinya.

STATUS LUARAN

1. Hasil submit

<https://ejournal.ikado.ac.id/index.php/teknika/authorDashboard/submission/426>



The screenshot shows a table titled 'Submission Files' with a search icon and the text 'Search'. The table contains one row with the following data:

Submission Files		Search	
▶	1792-2 yywahyup, ANALISIS DAN IMPLEMENTASI BACK OFFICE SISTEM LAPORAN SEKOLAH SDN 2 GLAGAH TEMON BERBASIS WEBSITE.docx (2)	October 25, 2021	Research Results

Gambar 14. Bukti hasil submit

Ini hasil paper yang melibatkan mahasiswa bernama Bayu Maulana yang diambil dalam skripsi S-1 di Jurusan Teknik Informatika Universitas Ahmad Dahlan. Paper tersebut di submit pada jurnal teknika pada tanggal 25 Oktober 2021.

2. Hak Kekayaan Intelektual

Aplikasi E rapor menggunakan arsitektur microservice telah didaftarkan pada hak kekayaan intelektual sebagai berikut:


REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	: EC00202201236, 6 Januari 2022
Pencipta	
Nama	: Drs. Wahyu Pujiyono, M.Kom., Bayu Maulana Yogi Kisworo dkk
Alamat	: Jl. Celeban Gang Panda 494, RT22 RW05 Tahunan, Umbulharjo, Yogyakarta, Yogyakarta, DI YOGYAKARTA, 55167
Kewarganegaraan	: Indonesia
Pemegang Hak Cipta	
Nama	: Drs. Wahyu Pujiyono, M.Kom., Bayu Maulana Yogi Kisworo dkk
Alamat	: Jl. Celeban Gang Panda 494, RT22 RW05 Tahunan, Umbulharjo, Yogyakarta, Yogyakarta, DI YOGYAKARTA, 55167
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: Aplikasi E-Rapor Dengan Metode Microservice
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 6 Januari 2022, di Yogyakarta
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan	: 000316386

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
 Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n Menteri Hukum dan Hak Asasi Manusia
 Direktur Jenderal Kekayaan Intelektual
 u.b
 Direktur Hak Cipta dan Desain Industri



Dr. Syarifuddin, S.T., M.H.
 NIP.197112182002121001

Disclaimer:
 Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

Gambar 15. Surat pernyataan ciptaan

PERAN MITRA

Mitra bertugas dalam melakukan pengujian aplikasi. Peran mitra berupa:

1. Tempat Pengujian
2. Koneksi Internet
3. Perangkat Komputer atau Laptop
4. Penguji (Staff Guru)

Terlampir surat pernyataan dari mitra terkait realisasi dukungan mitra sebagai berikut:



Gambar 16. Surat pernyataan realisasi dukungan mitra

KENDALA PELAKSANAAN PENELITIAN

Kendala yang dihadapi pada saat penelitian tahap ini adalah adanya masalah load database yang mencapai 100% dikarenakan data yang digunakan untuk pengujian menjadi berlipat ganda pada database, sehingga menyebabkan database menjadi berat dan pengujian selanjutnya menjadi gagal terus menerus. Solusi yang kita gunakan untuk menanggulangi masalah tersebut adalah Dengan melakukan pengulangan pengujian serta adanya perubahan prosedur pengujian dengan cara reset terhadap database terlebih dahulu supaya database tidak mencapai batas maksimal dalam pengolahan data dalam beberapa sesi pengujian.

RENCANA TINDAK LANJUT PENELITIAN

Pada kondisi sekarang untuk fitur upload data siswa input data masih menggunakan file berformat .xls atau excel file. Untuk rencana selanjutnya untuk fitur upload data siswa akan diubah menjadi input data berbasis web, dimana user akan meng-inputkan data pada web erapor. Rencana selanjutnya akan ada pengembangan fitur lainnya yaitu, fitur untuk penerimaan siswa baru, fitur kenaikan kelas, dan fitur kelulusan bagi siswa.

DAFTAR PUSTAKA

- [1] A. Bucchiarone, N. Dragoni, S. Dustdar, S. T. Larsen, and M. Mazzara, "From monolithic to microservices: An experience report from the banking domain," *Ieee Softw.*, vol. 35, no. 3, pp. 50–55, 2018.
- [2] S. Sharma, N. Uniyal, B. Tola, and Y. Jiang, "On monolithic and microservice deployment of network functions," in *2019 IEEE Conference on Network Softwarization (NetSoft)*, 2019, pp. 387–395.
- [3] W. Hasselbring and G. Steinacker, "Microservice architectures for scalability, agility and reliability in e-commerce," in *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*, 2017, pp. 243–246.
- [4] N. A. Sani, W. A. Fillah, A. Tjahyanto, and H. Suryotrisongko, "Development of Microservice Based Application E-Inkubator: Incubation and Investment Service Provider for SMEs," *Procedia Comput. Sci.*, vol. 161, pp. 1064–1071, 2019.
- [5] K. Gos and W. Zabierowski, "The Comparison of Microservice and Monolithic Architecture," in *2020 IEEE XVIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)*, 2020, pp. 150–153.
- [6] V. Singh and S. K. Peddoju, "Container-based microservice architecture for cloud applications," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 2017, pp. 847–852.
- [7] W. Pujiyono, F. Noviyanto, and Muslihudin, "Sistem Pengolahan Data Nilai Kenaikan Kelas Berbasis Web Dan Pengujian Beta," Yogyakarta, 2020.
- [8] S. Li *et al.*, "A dataflow-driven approach to identifying microservices from monolithic applications," *J. Syst. Softw.*, vol. 157, p. 110380, 2019.
- [9] C.-Y. Fan and S.-P. Ma, "Migrating monolithic mobile application to microservice architecture: An experiment report," in *2017 IEEE International Conference on AI & Mobile Services (AIMS)*, 2017, pp. 109–112.
- [10] J.-P. Gouigoux and D. Tamzalit, "From monolith to microservices: Lessons learned on an industrial migration to a web oriented architecture," in *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*, 2017, pp. 62–65.
- [11] R. Chen, S. Li, and Z. Li, "From monolith to microservices: A dataflow-driven approach," in *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*, 2017, pp. 466–475.
- [12] G. Mazlami, J. Cito, and P. Leitner, "Extraction of microservices from monolithic software architectures," in *2017 IEEE International Conference on Web Services (ICWS)*, 2017, pp. 524–531.
- [13] O. Al-Debagy and P. Martinek, "A comparative review of microservices and monolithic architectures," in *2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)*, 2018, pp. 149–154.
- [14] Z. Ren *et al.*, "Migrating web applications from monolithic structure to microservices architecture," in *Proceedings of the Tenth Asia-Pacific Symposium on Internetware*, 2018, pp. 1–10.

- [15] L. De Lauretis, “From Monolithic Architecture to Microservices Architecture,” in *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2019, pp. 93–96.
- [16] S. Eski and F. Buzluca, “An automatic extraction approach: Transition to microservices architecture from monolithic application,” in *Proceedings of the 19th International Conference on Agile Software Development: Companion*, 2018, pp. 1–6.
- [17] C. A. J. Acevedo, J. P. G. y Jorge, and I. R. Patiño, “Methodology to transform a monolithic software into a microservice architecture,” in *2017 6th International Conference on Software Process Improvement (CIMPS)*, 2017, pp. 1–6.
- [18] A. Saransig and F. Tapia, “Performance analysis of monolithic and micro service architectures—containers technology,” in *International Conference on Software Process Improvement*, 2018, pp. 270–279.
- [19] D. I. Permatasari, “Pengujian Aplikasi Menggunakan Metode Load Testing dengan Apache Jmeter pada Sistem Informasi Pertanian,” *JUSTIN (Jurnal Sist. dan Teknol. Informasi)*, vol. 8, no. 1, pp. 135–139, 2020.
- [20] A. De Camargo, I. Salvadori, R. dos S. Mello, and F. Siqueira, “An architecture to automate performance tests on microservices,” in *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services*, 2016, pp. 422–429.