



Implementasi AWS Cloud Computing dan Devops pada Infrastruktur Aplikasi Monitoring Linux Server

Eriek Orlando¹, Yudi Irawan Chandra²

^{1,2}sistem Informasi, STMIK Jakarta STI&K

¹eriek.orlando@gmail.co.id. ²yirawanc@gmail.com

Abstract

This study aims to optimize the application monitoring infrastructure of Linux servers in cloud computing using AWS and Terraform. Traditional infrastructures such as bare-metal servers incur high costs, while server virtualization with VMware is expensive. IT companies need to reduce costs, improve efficiency, and maximize server utilization. AWS offers server rental services based on the pay- as-you-go principle. Hashicorp has developed Terraform as an automation tool to build infrastructure and applications in cloud computing. Terraform has gained popularity in the DevOps community as it automates the application deployment process on AWS, enabling teams to efficiently manage infrastructure and applications. By leveraging AWS Cloud Computing and utilizing automation with Terraform, the best solution can be achieved.

Keywords: Amazon Web Service (AWS), Cloud Computing, DevOps; Infrastructure, Application, Monitoring, Linux Server

Abstrak

Penelitian ini bertujuan mengoptimalkan infrastruktur aplikasi monitoring Linux server di cloud computing menggunakan AWS dan Terraform. Infrastruktur tradisional seperti server baremetal memiliki biaya tinggi, sedangkan virtualisasi server dengan VMware mahal. Perusahaan-perusahaan IT perlu mengurangi biaya, meningkatkan efisiensi, dan memaksimalkan penggunaan server. AWS menawarkan layanan sewa server dengan prinsip pay as you go. Hashicorp menciptakan Terraform sebagai alat otomatisasi untuk membangun infrastruktur dan aplikasi di cloud computing. Terraform menjadi alat populer dalam dunia DevOps untuk mengotomatisasi proses deployment aplikasi di AWS, membantu tim mengelola infrastruktur dan aplikasi dengan efisien. Dengan memanfaatkan AWS Cloud Computing dan otomatisasi dengan Terraform, solusi terbaik dapat tercapai.

Kata kunci: Amazon Web Service (AWS); Cloud Computing; DevOps; Infrastruktur; Aplikasi; Monitoring; Linux Server

© 2024 Jurnal Pustaka Robot Sister

1. Pendahuluan

Pemanfaatan teknologi informasi tidak terlepas dari penggunaan jaringan internet, untuk dapat memiliki dan mengakses sebuah informasi, maka dibutuhkan sumber daya untuk dapat menjalankan website tersebut di jaringan internet [1]. Dengan menggunakan layanan internet, cloud computing dapat menyediakan perangkat lunak dan perangkat

keras melalui layanan internet dan memungkinkan pengelolaan sumber daya secara kolektif melalui internet. Cloud computing merupakan evolusi dari berbagai teknologi yang sudah ada sebelumnya, dan dapat membagi sumber daya komputasi seperti hardware, process, network, storage, application, services kepada pengguna dalam infrastruktur jaringan untuk kemudahan penggunaan dan

pelayanan[2]. Ilmu pengetahuan dalam bidang teknologi internet kini telah berkembang, salah satu perkembangan yang terjadi adalah teknologi cloud computing, yang terdapat model pengembangan public cloud dan private cloud. Public cloud memberikan kemudahan pada pengguna untuk mengeluarkan biaya sesuai dengan sumber daya yang digunakan sedangkan private cloud merupakan salah satu solusi dalam menjaga keamanan data sehingga private cloud memungkinkan para pengguna dapat melacak permasalahan dalam menjalankan sistem [3].

Solusi untuk mengoptimalkan biaya telah dicoba, seperti menggunakan virtualisasi server dengan VMware, namun masih ada kekurangan dalam hal efisiensi karena biaya lisensi yang mahal. Dalam beberapa tahun terakhir, permasalahan penggunaan server yang efisien dan pengurangan biaya telah menjadi fokus utama dalam dunia infrastruktur teknologi informasi [4]. Untuk mengatasi hal ini, AWS (Amazon Web Service) menciptakan inovasi baru dengan memperkenalkan cloud computing melalui layanan mereka yang dikenal sebagai Amazon Elastic Compute Cloud (EC2). Dengan menggunakan EC2, perusahaan dapat menyewa server sesuai kebutuhan mereka, mengikuti prinsip pay-as-you-go, dan hanya membayar untuk apa yang digunakan [5].

Selain menyewa server, AWS juga menyediakan layanan lain seperti Amazon Relational Database Service (RDS) yang memungkinkan pengguna untuk mengelola database dengan mudah tanpa perlu khawatir tentang upgrade dan pemeliharaan versi database. Namun, terdapat permasalahan lain, yaitu pengguna di AWS tetap perlu melakukan deployment service secara manual, dan jika jumlahnya banyak dan cukup kompleks, akan membutuhkan waktu yang lama [6].

Untuk mengatasi kelemahan tersebut, terdapat tools automasi seperti Terraform yang dikembangkan oleh perusahaan lain seperti Hashicorp. Terraform menjadi salah satu tools yang populer dalam dunia DevOps, karena dapat mengotomatisasi proses deployment aplikasi di AWS dengan efisien dan efektif. Dengan menggunakan Terraform, pengguna dapat membangun dan mengelola infrastruktur cloud secara konsisten dan aman, mengurangi kompleksitas konfigurasi yang harus dilakukan secara manual [7].

2. Metode Penelitian

Software Development Life Cycle (SDLC) adalah kerangka kerja yang mendefinisikan tahapan-tahapan dalam pengembangan perangkat lunak dari awal hingga akhir. Dalam konteks penelitian "Implementasi AWS Cloud Computing dan DevOps pada Infrastruktur Aplikasi Monitoring Linux Server" [8]. Software Development Life Cycle (SDLC) dapat diintegrasikan sebagai berikut:



Gambar 1. Tahapan penelitian

Perencanaan (Planning) : menentukan tujuan dan ruang lingkup proyek, serta mengidentifikasi kebutuhan sistem. **Aktivitasnya** melakukan analisis kebutuhan terhadap sistem monitoring server Linux, menyusun rencana proyek, termasuk timeline, anggaran, dan sumber daya yang dibutuhkan, menetapkan kriteria kesuksesan dan metrik evaluasi. **Analisis Kebutuhan (Requirements Analysis)** : mengumpulkan dan mendokumentasikan kebutuhan bisnis dan teknis dari aplikasi monitoring. **Aktivitasnya** wawancara dengan stakeholder untuk mengidentifikasi kebutuhan monitoring, dokumentasi kebutuhan sistem, seperti jenis data yang akan dimonitor, frekuensi monitoring, dan kebutuhan integrasi dengan layanan AWS, analisis persyaratan keamanan dan kepatuhan. **Desain Sistem (System Design)** : merancang arsitektur sistem dan komponen yang akan dikembangkan. **Aktivitasnya** merancang arsitektur AWS Cloud yang mencakup layanan seperti EC2, S3, CloudWatch, dan lainnya, merancang pipeline DevOps untuk otomatisasi deployment dan monitoring, menggunakan alat seperti Jenkins, GitLab CI, atau AWS CodePipeline, membuat diagram arsitektur sistem dan dokumentasi desain teknis. **Implementasi (Implementation)** : mengembangkan dan mengkonfigurasi sistem sesuai dengan desain yang telah dibuat. **Aktivitasnya** Deploy layanan AWS Cloud sesuai dengan desain arsitektur, konfigurasi server Linux dan instalasi monitoring tools, pengaturan pipeline CI/CD untuk otomatisasi deployment dan monitoring. **Pengujian (Testing)** : memastikan bahwa sistem berfungsi sesuai dengan spesifikasi dan bebas dari kesalahan. **Aktivitasnya** pengujian fungsionalitas aplikasi monitoring untuk memastikan semua fitur bekerja dengan baik, pengujian kinerja sistem untuk memastikan bahwa aplikasi monitoring dapat menangani beban kerja yang diharapkan, pengujian keamanan untuk memastikan bahwa sistem aman dari ancaman. **Deployment (Deployment)** : melakukan implementasi sistem ke lingkungan

produksi dan memastikan sistem berjalan dengan baik. **Aktivitasnya** migrasi sistem ke lingkungan produksi di AWS Cloud, konfigurasi akhir dan verifikasi sistem, pelatihan pengguna dan dokumentasi penggunaan. **Pemeliharaan (Maintenance)** : memastikan bahwa sistem berjalan dengan baik setelah implementasi dan melakukan perbaikan serta peningkatan yang diperlukan. **Aktivitasnya** monitoring berkelanjutan terhadap performa server Linux, penyelesaian bug dan isu yang ditemukan selama operasional, peningkatan sistem berdasarkan feedback pengguna dan analisis hasil monitoring [9].

Dengan menggunakan Software Development Life Cycle (SDLC), penelitian ini dapat mengikuti alur pengembangan yang sistematis dan terstruktur, memastikan setiap aspek implementasi AWS Cloud Computing dan DevOps pada aplikasi monitoring Linux server dapat dijalankan dengan baik dan sesuai dengan tujuan penelitian [10].

3. Hasil dan Pembahasan

3.1. Analisa Masalah

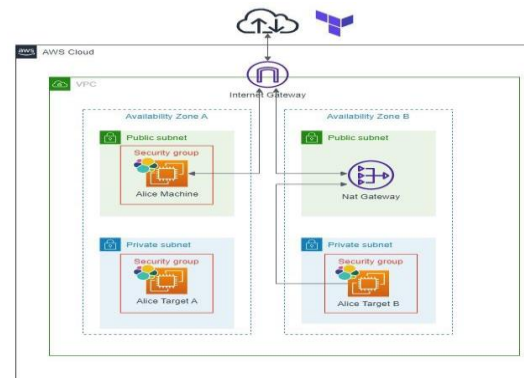
Dalam proses pembangunan infrastruktur, tim pekerja awalnya menghadapi dua masalah utama, pertama masalah biaya yang tinggi ketika menggunakan server baremetal. Sebagai contoh, model Huawei tipe 2288H V5 dengan spesifikasi CPU 16 Core, Memory 128 GB, dan Storage 50 GB memiliki harga sekitar \$5000 atau sekitar 80 juta rupiah. Selain itu, waktu pemesanan yang cukup lama, sekitar 1 bulan, juga menjadi tantangan dalam menggunakan server baremetal. Kedua, setelah beralih ke Cloud Computing, tim pekerja menghadapi masalah baru terkait dengan keamanan, skalabilitas, dan efisiensi infrastruktur yang dibangun. Proses deployment aplikasi yang manual dan tidak otomatisasi dan berpotensi mengakibatkan kesalahan dan penurunan produktivitas.

Untuk mengatasi kendala-kendala tersebut, diperlukan pemanfaatan perkembangan teknologi informasi dan komunikasi, khususnya AWS Cloud dan DevOps. Tim pekerja melakukan analisis kebutuhan yang mendalam dan memanfaatkan AWS Cloud untuk membangun infrastruktur yang aman dan efisien. Tim pekerja juga menerapkan prinsip DevOps dengan menggunakan alat-alat seperti Terraform dan Ansible untuk otomatisasi proses deployment. Selain itu, tim pekerja melakukan pemantauan kinerja dan pemeliharaan rutin untuk memastikan infrastruktur tetap optimal.

3.2. Perancangan Topologi Infrastruktur AWS

Perancangan topologi infrastruktur AWS merupakan tahapan penting dalam pembangunan infrastruktur teknologi informasi yang terintegrasi dan terhubung. Dalam perancangan topologi ini, harus dipertimbangkan secara matang komponen apa saja yang akan terlibat dan terhubung dalam infrastruktur

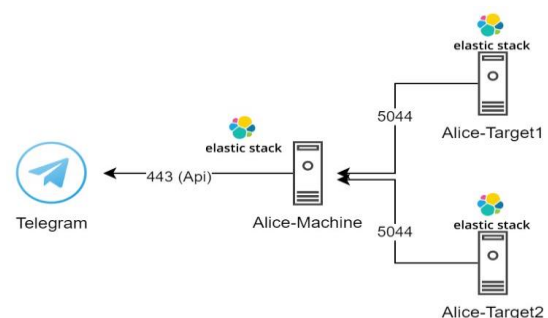
aplikasi monitoring server. Topologi harus mampu menggambarkan alur kerja keseluruhan aplikasi dengan jelas, sehingga dapat meminimalkan terjadinya kesalahan dalam proses pengembangan dan deployment. Dengan perancangan topologi yang tepat terhadap infrastruktur teknologi informasi dan proses deployment yang dapat menjadi lebih efisien dan terstandarisasi, sehingga dapat meningkatkan produktivitas dan kualitas kerja dalam dunia teknologi informasi. Topologi ini dapat dilihat pada gambar 2.



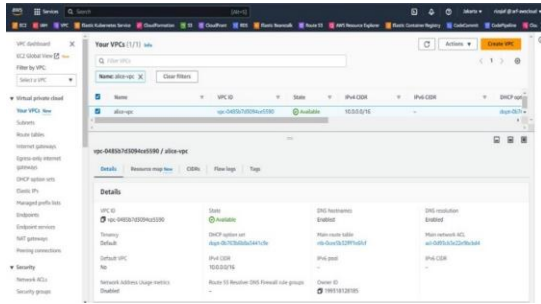
Gambar 2. Topologi Infrastruktur AWS

3.3. Perancangan Arsitektur Aplikasi

Perancangan arsitektur aplikasi merupakan langkah penting dalam pembangunan aplikasi yang handal dan dapat diandalkan. Pada tahap ini, perlu dipertimbangkan faktor-faktor seperti kebutuhan hardware, software, dan konfigurasi jaringan yang sesuai dengan kebutuhan aplikasi. Tujuannya adalah untuk memastikan bahwa semua komponen infrastruktur dapat bekerja secara optimal dan terintegrasi dengan baik, sehingga aplikasi dapat berjalan dengan baik dan memberikan kinerja yang stabil dan responsif. Selain itu, perancangan arsitektur juga harus memperhatikan aspek keamanan dan skalabilitas, sehingga aplikasi dapat berkembang dan diakses dengan aman dan mudah di masa depan. Dengan demikian, perancangan infrastruktur aplikasi yang baik dapat menjadi kunci keberhasilan dalam pengembangan aplikasi yang handal dan berkinerja tinggi seperti terlihat pada gambar 3.



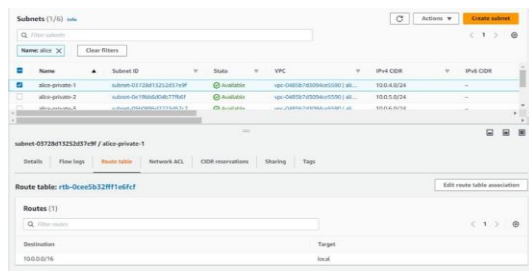
Gambar 3. Arsitektur Aplikasi



Gambar 9. Verifikasi VPC

2. Subnet Private

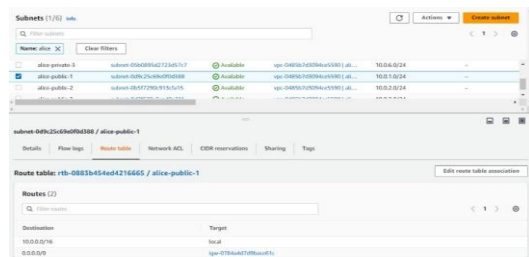
Subnet Private (alice-private-1, alice-private-2 dan alice-private-3) secara otomatis dideploy oleh Terraform sesuai kode yang ditentukan, seperti yang terlihat pada gambar 10.



Gambar 10. Verifikasi Subnet Private

3. Subnet Public

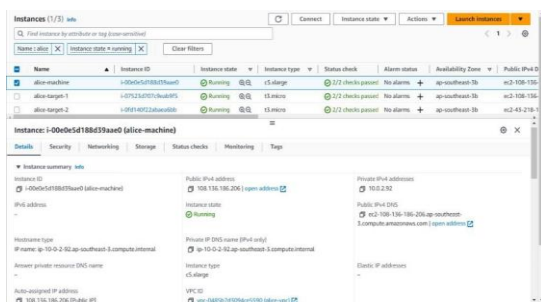
Subnet Public (alice-public-1, alice-public-2 dan alice-public-3) secara otomatis dideploy oleh Terraform sesuai kode yang ditentukan, seperti yang terlihat pada gambar 11.



Gambar 11. Verifikasi Subnet Public

4. EC2

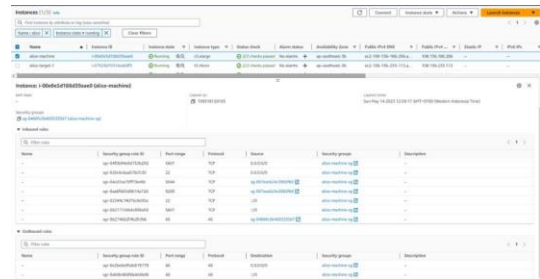
EC2 (alice-machine, alice-target-1, alice-target-2) secara otomatis dideploy oleh Terraform sesuai kode yang ditentukan, seperti yang terlihat pada gambar 12.



Gambar 12. Verifikasi EC2 Instance

5. Security Group Alice-Machine

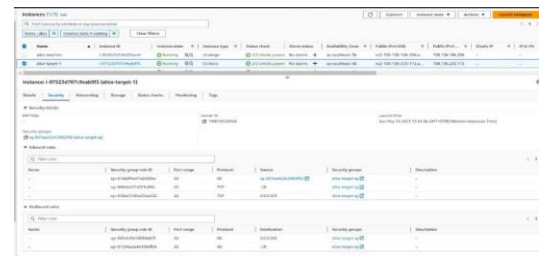
Security group alice-machine secara otomatis dideploy oleh Terraform sesuai kode yang ditentukan, seperti yang terlihat pada gambar 13.



Gambar 13. Security Group Alice-Machine

6. Security Group Alice-Target

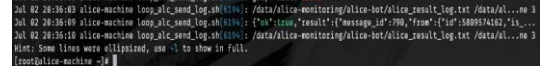
Security group alice-target secara otomatis dideploy oleh Terraform sesuai kode yang ditentukan, seperti yang terlihat pada gambar 14.



Gambar 14. Security Group Alice-Target

7. Aplikasi yang berjalan pada server Alice-Machine

secara otomatis berjalan dengan ditandai oleh munculnya alert pada Telegram yang telah diintegrasikan dengan aplikasi untuk monitoring, seperti yang terlihat pada gambar 15.



Gambar 15. Aplikasi Alice Running

Tampilan aplikasi monitoring pada telegram desktop, seperti yang terlihat pada gambar 16.



Gambar 16. Tampilan Alice Monitoring Telegram Via Desktop

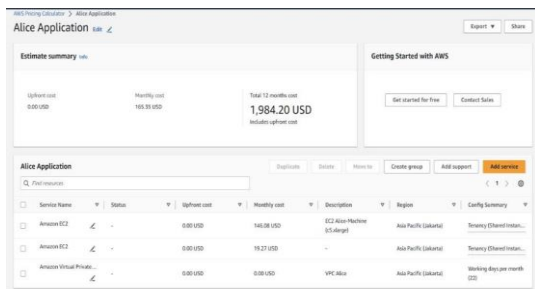
8. Tampilan aplikasi monitoring pada telegram mobile, seperti yang terlihat pada gambar 17.



Gambar 17. Tampilan Alice Monitoring Telegram Via Mobile

3.7. Hasil Uji Coba

Pada tahap ini akan dipaparkan hasil uji coba penggunaan Terraform untuk otomatisasi pembangunan server di AWS, serta estimasi biaya yang diperlukan untuk menyewa server di AWS. Melalui halaman resmi AWS (<https://calculator.aws/>), estimasi harga dengan spesifikasi yang telah ditentukan dapat ditampilkan. Berikut adalah estimasi biaya untuk uji coba seperti yang terlihat pada gambar 18.



Gambar 18: Estimasi Harga AWS

Tabel 1. Durasi Pembuatan Infrastruktur AWS dengan Terraform

Tahap Uji Coba	Waktu Mulai	Waktu Selesai	Durasi
Proses Deployment	19.58	20.04	6 menit

Tabel 2. Estimasi Biaya Sewa Server AWS

Service Name	Jumlah	Layanan	Biaya/bulan (IDR)	Biaya/tahun (IDR)
Alice-Machine (c5.xlarge)	1	Amazon EC2	2,191,200	26,294,400
Alice-Target (t3.micro)	2	Amazon EC2	289,080	3,468,900
VPC, Subnet and Security Group	1	Network	0	0
Total			2,480,280	29,763,300

4. Kesimpulan

Penggunaan AWS Cloud Computing dan DevOps dengan Terraform sebagai alat otomatisasi deployment aplikasi memberikan solusi yang efisien

dalam hal pengelolaan infrastruktur dan peningkatan skalabilitas. Implementasi ini memberikan solusi modern dan efisien dalam mengelola infrastruktur aplikasi. Perusahaan dapat dengan mudah mengatur dan mengelola sumber daya infrastruktur mereka melalui layanan yang disediakan oleh AWS, seperti provisioning server dengan cepat melalui layanan Amazon EC2 menggunakan tools Terraform. Dengan mengoptimalkan biaya dan meningkatkan efisiensi operasional, perusahaan dapat fokus pada pengembangan bisnis mereka dan meningkatkan daya saing di pasar.

Berdasarkan hasil penelitian dan simpulan yang diperoleh, terdapat beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut dan optimalisasi implementasi seperti pertimbangan migrasi aplikasi yang pada saat ini masih berjalan di infrastruktur tradisional ke AWS Cloud Computing untuk meningkatkan skalabilitas, keamanan, dan efisiensi, serta mengurangi biaya operasional dan pemeliharaan infrastruktur. Memberikan pelatihan berkelanjutan kepada tim kerja mengenai teknologi AWS dan praktik DevOps terbaru, ini akan memastikan bahwa tim selalu siap untuk mengelola dan memelihara infrastruktur dengan efektif.

Daftar Rujukan

- [1]. Lubis A, Wahyuni S. Implementasi Cloud Computing Menggunakan Platform AWS Pada Website Rumah Kue Havivya Medan. *Tekmapro*. 2024;19(1):134-147.
- [2]. Syahrul Mubarak M, Izman Herdiansyah M. Implementasi Cloud Computing Amazon Web Services (AWS) Pada Web Reservasi Kamar Hotel. *Kaji Ilm Inform dan Komput*. 2023;4(2):698-708. doi:10.30865/klik.v4i2.1212
- [3]. Hidayat Y, Arifwidodo B. Implementasi Web Server Menggunakan Infrastruktur As Code Terraform Berbasis Cloud Computing. *Format J Ilm Tek Inform*. 2021;10(2):192. doi:10.22441/format.2021.v10.i2.010
- [4]. Alfianti T, Diansyah TM, Liza R. Analisis Perbandingan Manajemen Konfigurasi Menggunakan Ansible Dan Shell Script Pada Cloud Server Deployment Aws. *JiTEKH*. 2020;8(2):78-84. doi:10.35447/jitekh.v8i2.308
- [5]. Fandy, Rosmasari, Putra GM. Pengujian Kinerja Web Server Atas Penyedia Layanan Elastic Cloud Compute (EC2) Pada Amazon Web Services (AWS). *Adopsi Teknol dan Sist Inf*. 2022;1(1):21-35. doi:10.30872/atasi.v1i1.45
- [6]. Roshandy A. Evaluasi Kinerja Moodle pada Cloud menggunakan Amazon Web Service (AWS) Relational Database Service. *J Pengemb Teknol Inf dan Ilmu Komput - JPTIK*. 2022;6(5):2047-2053. <http://j-ptiik.ub.ac.id>
- [7]. Gustian D, Fitrisia Y, Novayani W, Purwantoro E.S.G.S S. Implementasi Automation Deployment pada Google Cloud Compute VM menggunakan Terraform. *INOVTEK Polbeng - Seri Inform*. 2023;8(1):50. doi:10.35314/isi.v8i1.3095
- [8]. Harimurti Y, Udariansyah D. Implementasi Service EC2 & S3 Amazon Web Service Pada Niche Blog Menggunakan Metode SDLC. *Media Online*. 2023;4(2):675-685. doi:10.30865/klik.v4i2.1192
- [9]. Ridwan M, Fitri I, Benrahman B. Rancang Bangun Marketplace Berbasis Website menggunakan Metodologi Systems Development Life Cycle (SDLC) dengan Model

Waterfall. *J JTik (Jurnal Teknol Inf dan Komunikasi)*. 2021;5(2):173. doi:10.35870/jtik.v5i2.209

Informasi Pengadaan Barang Menggunakan Teknologi Cloud Computing. *J Ris Komputer*. 2023;10(1):2407-389. doi:10.30865/jurikom.v10i1.5580

[10]. Hardiansyah D, Priyatna A. Rancang Bangun Sistem