IMPLEMENTASI OPENVPN SERVER MENGGUNAKAN RED HAT ENTERPRISE LINUX PADA AMAZON WEB SERVICES

Ayman Human Sukma, Mhd. Ilyasyah Drilanang, Sybil Auzi, Dedy Kiswanto
Fakultas Matematika Ilmu Pengetahuan Alam / Matematika, Universitas Negeri Medan, Medan
mdrilanang.4233550027@mhs.unimed.ac.id

ABSTRAK

Virtual Private Network (VPN) adalah teknologi yang memungkinkan koneksi jaringan yang aman melalui internet. Pada penelitian ini, server VPN diimplementasikan menggunakan OpenVPN pada platform Amazon Web Services (AWS) dengan Red Hat Enterprise Linux (RHEL) sebagai sistem operasi server. Manajemen dan pengujian koneksi VPN dilakukan dengan menggunakan mesin Windows 11 sebagai klien. Penelitian bertujuan untuk membangun infrastruktur VPN yang aman dan dapat diakses dari berbagai lokasi pada layanan service cloud. Pengujian dilakukan untuk mengamati performansi kinerja dan hasil akan membantu untuk memungkinkan pengembangan kinerja dan keamanan jaringan setelah implementasi VPN. Hasil penelitian menunjukkan bahwa server VPN berjalan dengan stabil dengan kecepatan latensi rata-rata 48ms yang menunjukkan bahwa ini adalah koneksi yang aman dengan kinerja yang dapat diterima.

Kata kunci: *OpenVPN, AWS, Red Hat Enterprise Linux, VPN*

1. PENDAHULUAN

Manusia bermasyarakat dan saling terhubung, di mana pengguna serta perusahaan terlibat dengan teknologi digital untuk menjalankan aktivitas dan proses bisnis sehari-hari. Beberapa tahun terakhir telah ditandai oleh evolusi teknologi komunikasi dan komputasi yang berkelanjutan dan cepat menuju infrastruktur publik, beralih dari arsitektur berbasis layanan ke cloud, dan baru-baru ini ke layanan mikro dan Internet of Things (IoT). Pada saat yang sama, pentingnya infrastruktur pribadi telah kembali didorong oleh meningkatnya kebutuhan akan perlindungan data, yang menghasilkan peraturan baru.[1] Kini kita menyebut komputasi awan sebagai penyediaan daya komputer, basis data, ruang penyimpanan, perangkat lunak aplikasi, dan sumber daya TI lainnya berdasarkan permintaan, berbasis internet, dan bayar sesuai pemakaian. Platform layanan awan memberi Anda akses instan ke sumber daya TI yang fleksibel dan hemat biaya, yang dapat digunakan untuk segala hal mulai dari menjalankan aplikasi yang berbagi foto dengan jutaan pengguna seluler hingga menangani operasi penting organisasi. Dengan komputasi awan, kita dapat terhindar dari pembelian perangkat keras yang mahal dan menghabiskan banyak waktu untuk tugas-tugas administratif. Sebagai alternatif, kita dapat menyediakan jumlah dan pengaturan sumber daya komputer yang dibutuhkan untuk menjalankan ide cerdas Anda berikutnya atau memelihara infrastruktur TI Anda. Semua yang kita butuhkan ada di ujung jari Anda dalam sekejap, dan hanya akan dikenakan biaya untuk apa yang benar-benar kita gunakan. Komputasi awan menyederhanakan proses mendapatkan akses jaringan ke berbagai server, penyimpanan, basis data, dan layanan aplikasi.[2]

Berdasarkan latar belakang penelitian yang telah diuraikan sebelumnya, pengertian dan bagaimana pengimplementasian setup server VPN menggunakan AWS yang dilakukan merupakan bagian dari rumusan masalah yang akan diuraikan dalam penelitian ini, bertujuan untuk mendeskripsikan pengertian VPN dan AWS di dalam cloud computing, menjelaskan bagaimana menerapkan dan meganalisa proses setup server VPN dengan menggunakan layanan Amazon Web Services (AWS).

2. TINJAUAN PUSTAKA

2.1. Penggunaan AWS

Metode pemantauan kesehatan tradisional sering kali padat karya, tidak efisien, dan kurang responsif terhadap kebutuhan pertanian modern. Seiring dengan meningkatnya jumlah perangkat IoT dalam pertanian, masalah skalabilitas dan beban komputasi menjadi menonjol, yang membutuhkan solusi yang efisien dan dapat diskalakan. Penelitian ini memperkenalkan arsitektur berbasis cloud yang ditujukan untuk meningkatkan pemantauan kesehatan ternak. Sistem ini dirancang untuk melacak indikator kesehatan penting seperti pola gerakan, suhu tubuh, dan detak jantung, memanfaatkan AWS untuk penanganan data yang kuat dan Python untuk pemrosesan data dan analisis waktu nyata. Sistem yang diusulkan menggabungkan teknologi IoT Pita Sempit (Nb IoT), yang dioptimalkan untuk komunikasi jarak jauh dengan bandwidth rendah, sehingga cocok untuk lokasi pertanian pedesaan dan terpencil. Skalabilitas arsitektur memungkinkan pengelolaan yang efektif dari berbagai jumlah perangkat IoT, yang penting untuk beradaptasi dengan perubahan ukuran kawanan dan skala pertanian. Eksperimen awal yang dilakukan untuk menilai kinerja sistem telah menunjukkan daya tahan dan efektivitasnya, yang menunjukkan keberhasilan integrasi layanan AWS IoT Cloud dengan perangkat IoT yang digunakan. [3] Baru-baru ini dalam lanskap keamanan siber, berbagai tokoh telah menyebar dengan kekhasan yang berbeda. Misalnya ada peretas Black Hat, yang bertujuan untuk

melakukan kerusakan pada sistem atau untuk secara diam-diam mencuri informasi sensitif tetapi ada juga peretas Etis atau White Hat, yang bertujuan untuk menyelidiki kerentanan sistem yang sedang dianalisis hanya dengan persetujuan pemiliknya. Dalam konteks ini, peretas Red Hat, yang didefinisikan sebagai pejuang dunia peretas, sedang muncul. Tujuan utama mereka adalah untuk secara independen menemukan dan memecahkan kerentanan, dengan mencegah serangan siber. Dalam makalah ini kami mengusulkan sebuah metode yang bertujuan untuk mengotomatiskan proses penemuan dan mitigasi kerentanan yang biasanya dilakukan oleh peretas Red Hat. Kami memanfaatkan rangkaian alat dari beberapa alat yang terkenal dan kami mengevaluasi metode yang diusulkan dengan memanfaatkan distro Linux Metaesploitable, yang menunjukkan bahwa metode yang diusulkan mampu secara otomatis mengurangi kerentanan yang menimpa enam layanan yang tersebar luas.[4]

2.2. Penggunaan VPN

Jaringan pribadi virtual (VPN) merupakan infrastruktur perangkat keras/perangkat lunak yang menerapkan saluran komunikasi pribadi dan rahasia yang biasanya berjalan melalui Internet. VPN saat ini merupakan salah satu teknologi yang paling dapat diandalkan untuk mencapai tujuan ini, juga karena sebagai teknologi yang terkonsolidasi, memungkinkan untuk menerapkan patch yang tepat untuk memperbaiki celah keamanan apa pun. Secara khusus, sistem operasi menyediakan nilai metrik kinerja yang berbeda untuk berbagai kombinasi variabel konfigurasi. Sasaran pertama yang dikejar adalah menemukan algoritme untuk menjamin rasio transmisi/enkripsi data seefisien mungkin. Sasaran kedua adalah meneliti algoritme yang mampu menjamin spektrum kompatibilitas terluas dengan infrastruktur terkini yang mendukung teknologi VPN, untuk memperoleh sistem koneksi yang aman untuk jaringan IoT yang tersebar secara geografis di area yang sulit dikelola seperti lingkungan pinggiran kota atau pedesaan. Banyak jenis protokol VPN telah diusulkan yang menawarkan berbagai tingkat keamanan dan fitur. Jadi, protokol-protokol tersebut juga menyajikan kinerja yang berbeda dalam hal parameter latensi dan keluaran throughput. Pemilihan solusi VPN yang tepat sangat penting bagi kinerja sistem.[5] Dalam pekerjaan saat ini, kami berfokus pada analisis penggunaan layanan cloud serta visualisasi penggunaan. Biasanya, sebuah perusahaan memiliki banyak aplikasi yang akan didukung melalui layanan cloud. Saat aplikasi baru dikembangkan, aplikasi lama sering kali jarang digunakan atau ditinggalkan, sehingga alokasi layanan harus dianalisis dan disesuaikan berdasarkan penggunaan. Kami membatasi pekerjaan kami pada analisis layanan AWS, karena AWS adalah salah satu penyedia layanan cloud terbesar. Biasanya, perusahaan memiliki beberapa akun AWS dan masing-masing perlu menganalisis penggunaan di seluruh akun ini. Saat ini tidak ada alat ringan yang tersedia untuk mengelola beberapa akun di satu tempat, dan tujuan kami adalah untuk menyusun sistem yang akan menyelesaikan masalah ini. [6]

Virtual Private Network (VPN) merupakan cara populer untuk menghubungkan jaringan lokal ke cloud. VPN menghubungkan jaringan lokal ke jaringan penyedia cloud dengan mengenkripsi terowongan melalui internet publik. Hal ini menjaga keamanan transfer data. Google Cloud Platform (GCP), Amazon Web Services (AWS), dan Microsoft Azure hanyalah beberapa penyedia cloud besar yang menawarkan layanan VPN. Anda dapat membuat konektivitas ke AWS melalui AWS Site-to-Site VPN atau AWS client VPN. AWS Site-to-Site VPN memungkinkan bisnis menghubungkan jaringan lokal atau pusat data mereka ke Amazon Virtual Private Cloud (VPC) dengan cara yang aman. Azure VPN Gateway dari Microsoft Azure memungkinkan Anda menghubungkan jaringan lokal ke jaringan virtual Azure dengan aman dan lintas lokasi. Cloud VPN dari GCP merupakan layanan terkontrol yang menghubungkan jaringan lokal ke jaringan di GCP Virtual Private Cloud (VPC). [7] Dalam bidang teknologi informasi yang berkembang pesat, pencarian solusi yang aman dan hemat biaya telah menjadi hal yang sangat penting bagi perusahaan yang ingin memperkuat infrastruktur digital mereka. Makalah penelitian ini membahas aspek penting dari pencarian ini, dengan fokus pada eksplorasi layanan VPN terkelola tingkat perusahaan yang memanfaatkan potensi perangkat sumber terbuka dan terjangkau dalam kerangka kerja Amazon Web Services (AWS) yang tangguh. Untuk mendukung pendekatan transformatif ini maka mengusulkan integrasi strategis OpenVPN sebagai client dan pfSense sebagai firewall/router, keduanya dijalankan dalam infrastruktur AWS yang dapat diskalakan. .[8]

Dengan menjadikan internet sebagai media untuk berkomunikasi dapat mempercepat aktifitas baik dalam bekerja ataupun bertukar informasi. Dengan adanya jaringan internet ini memudahkan manusia untuk mencari maupun menerima informasi dengan cepat, namun tingkat kecepatan dalam bertukar informasi yang bervariasi bergantung pada besaran bandwith yang dimiliki oleh setiap provider yang terkadang lambat atau cepat sehingga besaran biaya yang harus dibayarkan pun bervariasi bahkan cenderung mahal. Demikian juga akan kekhawatiran penyalahgunaan kebocoran data pribadi pengguna jika digunakan oleh terlalu banyak pengguna pada saat yang bersamaan dan jaminan keamanan data ataupun gangguan lain yang terkadang membuat pengguna merasa tidak aman menggunakan platform yang disediakan oleh provider. Salah satu upaya yang dapat dilakukan untuk mengatasi permasalahan tersebut adalah dengan membangun sebuah Virtual Private Network (VPN). Pada penelitian ini VPN dibangun dengan mengkombinasikan Elastic Compute Cloud

(EC2) dan OpenVPN menggunakan terminal sebagai remote access penggunaan protokol jaringan untuk kebutuhan remote server yang diakses dari kendali jarak jauh.[9] Latensi jaringan merupakan faktor penting untuk kualitas pengalaman yang dirasakan untuk banyak aplikasi. Dengan meningkatnya fokus pada aplikasi interaktif dan waktu nyata, yang memerlukan latensi yang andal dan rendah, kemampuan untuk memantau latensi secara terus-menerus dan efisien menjadi lebih penting dari sebelumnya. Ada banyak alat untuk mengukur latensi jaringan secara aktif dengan mengirimkan probe jaringan, seperti ping, IRTT, dan RIPE Atlas. Sementara pemantauan aktif berguna untuk mengukur konektivitas dan latensi jaringan idle dengan cara yang terkendali, ia tidak dapat secara langsung menyimpulkan pengalaman lalu lintas aplikasi latensi. Probe jaringan dapat diperlakukan secara berbeda dari lalu lintas aplikasi oleh jaringan, karena misalnya manajemen antrean aktif dan penyeimbangan beban, dan oleh karena itu latensinya juga dapat berbeda. Lebih jauh lagi, banyak alat pemantauan aktif mengharuskan agen untuk disebarkan langsung pada target yang dipantau, yang tidak layak bagi ISP yang ingin memantau latensi pelanggannya.[10]

3. METODE PENELITIAN

3.1. Persiapan Infrastruktur AWS

3.1.1. Buat Instance EC2 di AWS:

a. Login ke AWS Management Console.
b. Pilih EC2 dan klik Launch Instance.
c. Pilih Red Hat Enterprise Linux (RHEL) sebagai AMI (Amazon Machine Image).
d. Pilih tipe instance t2.micro untuk beban kerja dasar.
e. Konfigurasikan Security Group:
   - Buka port 22 (TCP) untuk SSH.
   - Buka port 1194 (UDP) untuk OpenVPN.

3.1.2. Akses Instance RHEL melalui SSH:

Setelah instance diluncurkan, akses server melalui SSH dengan perintah berikut:

ssh -i "keypair.pem" ec2-user@<Public_IP_Address>

Ganti <Public_IP_Address> dengan IP publik instance EC2.

3.2. Instalasi dan Konfigurasi OpenVPN di RHEL

Instal OpenVPN dan Easy-RSA untuk proses sertifikat di server:

sudo dnf install openvpn easy-rsa -y

```
[ec2-user@ip-172-31-31-112 ~]$ sudo dnf install -y openvpn
Updating Subscription Management repositories.
Extra Packages for Enterprise Linux 9 openh264  904  B/s | 2.5 kB     00:02
Red Hat Enterprise Linux 9 for x86_64 - AppStre  63 MB/s |  42 MB     00:00
Red Hat Enterprise Linux 9 for x86_64 - BaseOS   66 MB/s |  33 MB     00:00
Red Hat CodeReady Linux Builder for RHEL 9 x86_  40 MB/s | 9.3 MB     00:00
Red Hat Enterprise Linux 9 Client Configuration  35 kB/s | 3.0 kB     00:00
Dependencies resolved.
================================================================================
 Package              Architecture   Version                Repository    Size
================================================================================
Installing:
 openvpn              x86_64         2.5.11-1.el9           epel         655 k
Installing dependencies:
 pkcs11-helper        x86_64         1.27.0-6.el9           epel          62 k

Transaction Summary
Installed:
  openvpn-2.5.11-1.el9.x86_64         pkcs11-helper-1.27.0-6.el9.x86_64

Complete!
```

Gambar 1. OpenVPN terinstall

```
[ec2-user@ip-172-31-31-112 ~]$ sudo dnf install -y openvpn easy-rsa
Updating Subscription Management repositories.
Last metadata expiration check: 0:03:49 ago on Mon 21 Oct 2024 05:53:04 PM UTC.
Package openvpn-2.5.11-1.el9.x86_64 is already installed.
Dependencies resolved.
================================================================================
 Package            Architecture      Version               Repository      Size
================================================================================
Installing:
 easy-rsa           noarch            3.1.6-1.el9           epel            67 k

Transaction Summary
================================================================================
Install  1 Package
Installed:
  easy-rsa-3.1.6-1.el9.noarch

Complete!
```

Gambar 2. Easy-RSA terinstall

3.2.1. Konfigurasi CA (Certificate Authority).

Buat direktori CA dan file Public Key Infrastructure (PKI) sebagai tempat pengolahan dan pembuatan key dan sertifikat untuk server dan client yang terlibat:

```
mkdir ~/openvpn-ca
cd ~/openvpn-ca
./easyrsa init-pki
./easyrsa build-ca nopass
```

```
[ec2-user@ip-172-31-31-112 easy-rsa]$ ./easyrsa init-pki

Notice
------
'init-pki' complete; you may now create a CA or requests.

Your newly created PKI dir is:
* /home/ec2-user/easy-rsa/pki

Using Easy-RSA configuration:
* /home/ec2-user/easy-rsa/pki/vars

IMPORTANT:
  Easy-RSA 'vars' template file has been created in your new PKI.
  Edit this 'vars' file to customise the settings for your PKI.
  To use a global vars file, use global option --vars=<FILE>

[ec2-user@ip-172-31-31-112 easy-rsa]$
[ec2-user@ip-172-31-31-112 easy-rsa]$ ./easyrsa build-ca
Using Easy-RSA 'vars' configuration:
* /home/ec2-user/easy-rsa/pki/vars

Using SSL:
* openssl OpenSSL 3.0.7 1 Nov 2022 (Library: OpenSSL 3.0.7 1 Nov 2022)

Enter New CA Key Passphrase:
```

Gambar 3. Setup Easy-RSA untuk Certificate Authority

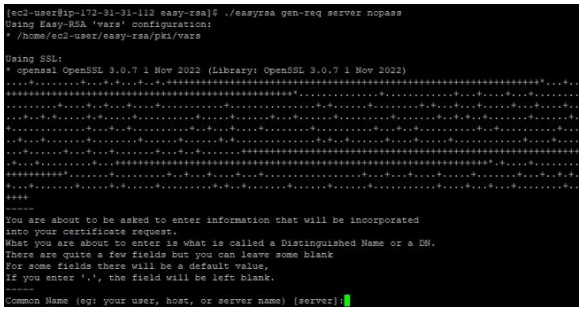
Gambar 4. Setup Easy-RSA setelah dimasukkan kata sandi Passphrase

Gambar 5. Common Name telah dibuat

3.2.2. Pembuatan sertifikat Server dan Client.

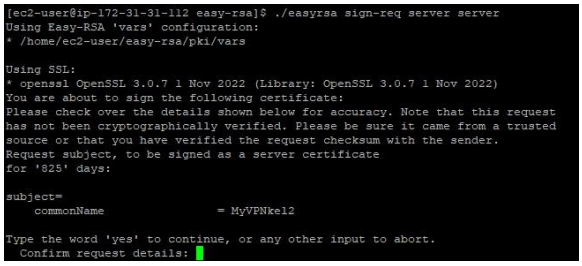
Buat Sertifikat Server:

./easyrsa gen-req server nopass
./easyrsa sign-req server server
./easyrsa gen-dh



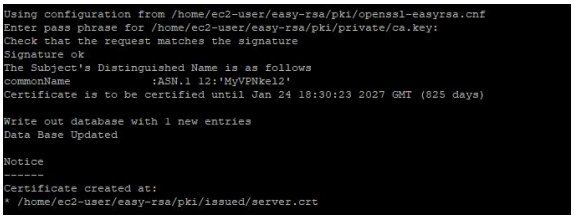
Gambar 6. Membuat Request No Pass

Gambar 7. Sertifikat Server



Gambar 8. Membuat Request Sertifikat Server

Gambar 9. Meminta Password setelah di request nya dikonfirmasi dengan “yes”



Gambar 10. Password berhasil dimasukkan dan Sertifikat Server berhasil dibuat

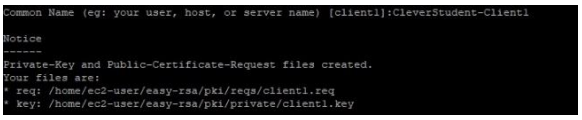
Gambar 11. Membuat parameter Diffie-Hellman

Buat sertifikat client:

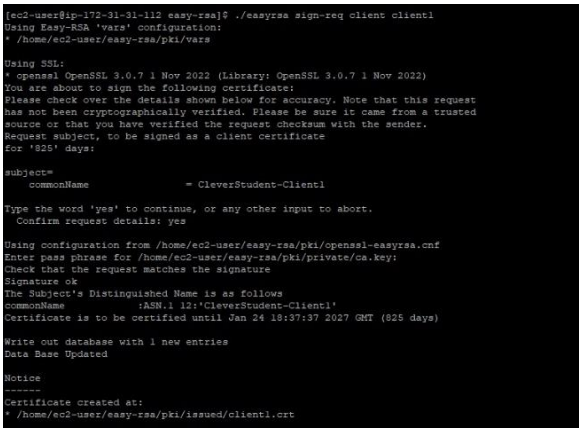
./easyrsa gen-req client1 nopass
./easyrsa sign-req client client1

Parameter No Pass dimasukkan karena autentikasi hanya bentuk sertifikat dan kunci terkhusus, tidak melalui password dan username

Gambar 12. Sertifikat Client



Gambar 13. Data client berhasil dibuat



Gambar 14. Sign Request Client

3.2.3. Konfigurasi OpenVPN:

Copy file sampel konfigurasi OpenVPN ke directory openvpn dan edit untuk mengolah server:

sudo cp /usr/share/doc/openvpn/sample/sample config-files/server.conf /etc/openvpn/
sudo nano /etc/openvpn/server.conf

Sesuaikan baris setting di konfigurasi:

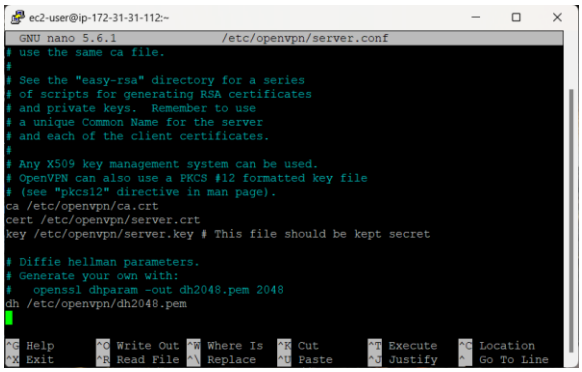
port 1194
proto udp
dev tun

ca /etc/openvpn/ca.crt
cert /etc/openvpn/server.crt
key /etc/openvpn/server.key
dh /etc/openvpn/dh.pem

keepalive 10 120
cipher AES-256-CBC
persist-key
persist-tun

status /var/log/openvpn-status.log
log-append /var/log/openvpn.log
verb 3

Penambahkan cipher untuk keamanan tambahan serta penyesuaian file sertifikat dan key server:



Gambar 15. Key dan sertifikat server

3.2.4. Aktifkan IP Forwarding

Gambar 16. Setting IP Forwarding

Edit file sysctl untuk mengaktifkan IP forwarding:

sudo nano /etc/sysctl.conf
net.ipv4.ip_forward=1

Simpan perubahan:
sudo sysctl –p

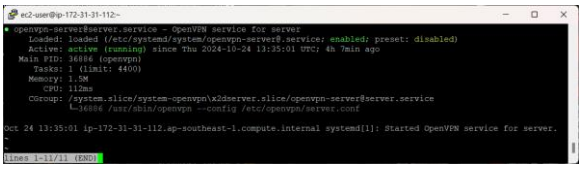
3.2.5. Konfigurasi Firewall

Atur firewall di RHEL untuk mengizinkan lalu lintas VPN:
sudo firewall-cmd --zone=public --add-port=1194/udp --permanent
sudo firewall-cmd –reload

Gambar 17. List Firewall

Setelah konfigurasi selesai, mulai layanan OpenVPN:
sudo systemctl start openvpn-server@server.servic
sudo systemctl enable openvpn@server

Lihat status untuk pemastian aktif server:
sudo systemctl status openvpn@server



Gambar 18. Status Server OpenVPN

3.3. Pengelolaan dan Pengujian Client

3.3.1. Buat file konfigurasi client (client.ovpn) dengan setting berikut

client
dev tun
proto udp
remote <Server_Public_IP> 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert client1.crt
key client1.key
remote-cert-tls server
cipher AES-256-CBC

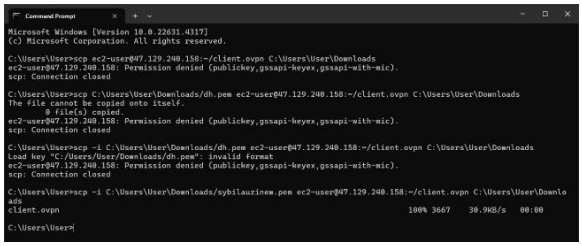
verb 3

Sesuaikan alamat remote IP adalah alamat pada instansi EC2 RHEL serta file sertifikat, key dan CA dengan file client yang sudah terbuat melalui Easy-RSA pada tahap 3.2, nomor 2.

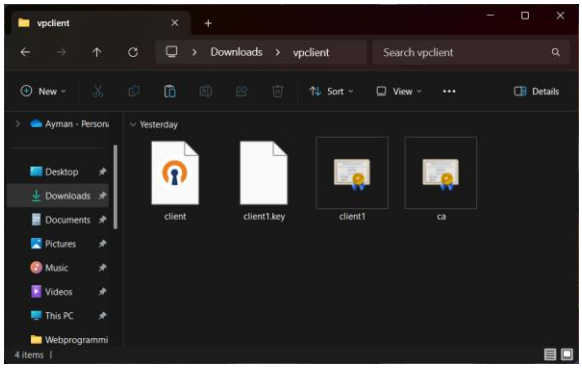
3.3.2. Pemindahan Data Client

File client config serta seritifikat, key-nya dicopy serta dipindahin melalui perintah seperti Secure Copy Protocol (SCP) dari file data client di instansi

ec2 RHEL ke dalam directory yang diinginkan dalam device client.



Gambar 19. Pemindahan file data client



Gambar 20. Hasil pemindahan file data client ke device

3.3.3. Hubungkan Client ke VPN

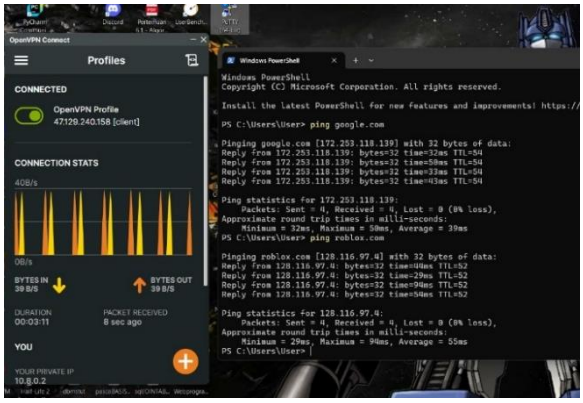
Jalankan OpenVPN client melalui membuka file client.ovpn sehingga muncul beranda profil untuk pendaftaran, pastikan di perangkat/device sudah terinstall OpenVPN client untuk koneksi:

Gambar 21. Status koneksi Client

3.3.4. Pengujian Konektivitas

Uji konektivitas dengan melakukan ping ke server melalui situs eksternal:

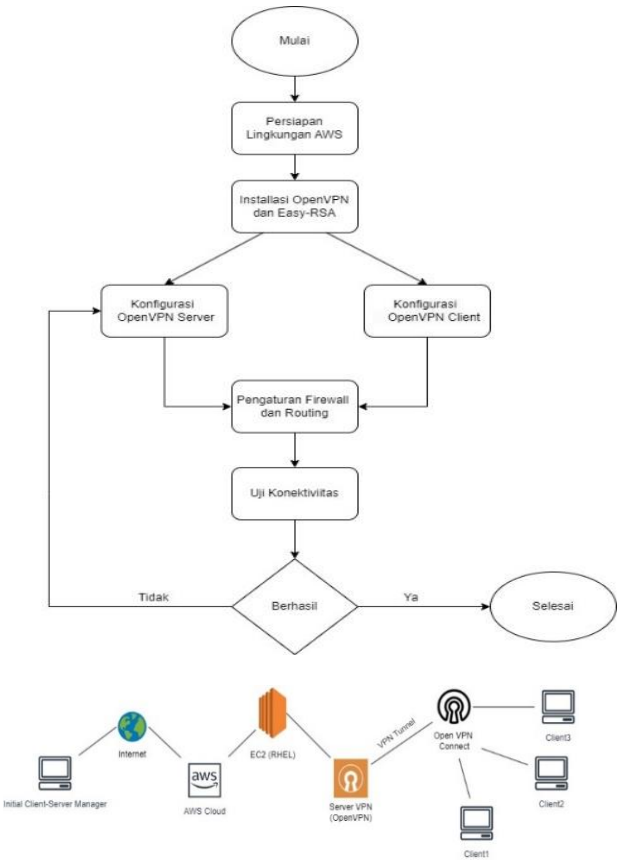
ping google.com (atau website lainnya)



Gambar 22. Hasil pengujian koneksi

4. HASIL DAN PEMBAHASAN

Praktek pengerjaan dicantum dalam diagram berikut, yang berupa hasil client sebagai penghubung ke server hanya terlibat salah satu contoh yaitu dalam suatu perangkat Windows 11:



Gambar 23. Alur Pengerjaan

Setelah implementasi server VPN di AWS menggunakan Red Hat Enterprise Linux dan pengelolaan client, pengujian konektivitas menunjukkan bahwa server VPN berjalan stabil. Koneksi antara client dan server melalui OpenVPN berhasil dilakukan dengan baik, dan data yang ditransmisikan melalui VPN dienkripsi menggunakan AES-256, yang memberikan keamanan tambahan. Pengujian menunjukkan bahwa; koneksi VPN berhasil diinisiasi, kecepatan performansi transfer data pada

website Google dan Roblox tertampil 39ms dan 55ms berturut-turut sehingga kecepatan rata-rata adalah 48ms sehingga menunjukkan latency dan kecepatan jaringan tetap stabil meskipun melalui VPN. Firewall dan IP Forwarding pada RHEL berfungsi dengan baik dalam mempersilakan lalu lintas traffic dari client VPN diteruskan melalui servernya.

5. KESIMPULAN & SARAN

Hasil penelitian dapat disimpulkan bahwa praktek dalam mengimplementasikan server VPN menggunakan OpenVPN pada Red Hat Enterprise Linux (RHEL) di-host oleh EC2 dalam AWS meliputi; penginstalan OpenVPN dan EasyRSA untuk mengolah server dan pembuatan identifikasi server dan client yang akan koneknya, mengkonfigurasi server untuk dicocokkan dengan data server yang terbuat, pemindahan data client kepada perangkat dikehendaki sebagai client melalui SCP, mengkonek server melalui OpenVPN Client dalam perangkat tersebut. Dalam penelitian ini pengamatan koneksi melalui ping ke server menunjukkan kecepatan transfer data rata-rata 48 milidetik, hasil tersebut menunjukkan bahwa performansi VPN berjalan stabil dan aman. Solusi ini efektif untuk membangun koneksi jarak jauh ke jaringan melalui cloud. Penelitian ini belum sempurna dan perlu ditingkatkan penambahan pengujian ping kecepatan transfer data pada situs lainnya untuk keefektivitasan dan keberadaan pemanfaatan nilai guna praktek VPN pada service cloud AWS, yaitu dengan melakukan pengujian ping ke server situs lainnya dalam terminal client.

DAFTAR PUSTAKA

[1] Anisetti, M., Ardagna, C.A., Bena, N., Damiani, E., (2020), Stay Thrifty, Stay Secure: A VPN-Based Assurance Framework for Hybrid Systems, 1-14.

[2] Bari, A., Khan, I., Samrin, R., Khare, A., (2024), VPC & Public Cloud Optimal Performance in Cloud Environment, Educational Administration: Theory and Practice, 30(6), 1789-1798.

[3] Bhaskaran, H.S., Gordon, M., Neethirajan, S., (2024), Development of a cloud-based IoT system for livestock health monitoring using AWS and python, Smart Agricultural Technology 9, 1-16.

[4] Filiol, E., Mercaldo, F., Santone, A., (2021), A Method for Automatic Penetration Testing and Mitigation: A Red Hat Approach, Procedia Computer Science, 192, 2039-2046.

[5] Gentile, A.F., Macri, D., Rango, F.D., Tropea, M., Greco, E., (2022), A VPN Performances Analysis of Constrained Hardware Open-Source Infrastructure Deploy in IoT Environment, Future Internet, 14(264), 1-27.

[6] George, L.C., Guo, Y., Stepanov, D., Peri, V.K.R., Elvitigala, L. Spichkova, M., (2020), Usage visualisation for the AWS services, Procedia Computer Science, 176, 3710-3717.

[7] Kaur, T., (2024), Connecting and Securing On-Prem Network to Cloud Infrastructure, International Journal of Research in Computer Applications and Information Technology, 7(1), 54-63.

[8] Makani, S.T., Panchakarla, B.P., Pulyala, S.R., (2022), Enterprise-Grade Hosted VPN Services with AWS Infrastructure, Journal of Engineering and Applied Sciences Technology, 4(4), 1-8.

[9] Srivastava, A., Rizvi, S.W.A., Priya, R., (2023), Implementing Cloud Security through AWS for Blood Bank Application, Journal of Informatics Electrical and Electronics Engineering, 4(81), 1-10.

[10] Sundberg, S., Brunstrom, A., Ferlin-Reiter, S., Hoiland-Jorgensen, T., Brouer, J.D., (2023), Efficient Continuous Latency Monitoring with eBPF, 191-207.