



Studi Kasus – Fase 3 (C03)

IT Network Plan : Implementasi Parsial Rancangan Solusi

Penulis : NR, RF, RR, FS, AAT, AVA
Versi : 1 (20241204-0800)



© 2024, Fakultas Ilmu Komputer Universitas Indonesia
This work is licensed under [Creative Commons Attribution-ShareAlike 4.0 International \(CC BY-SA 4.0\)](https://creativecommons.org/licenses/by-sa/4.0/) license.

Riwayat Versi

Versi	<i>Timestamp</i>	Halaman	Perubahan
1	20241204-0800	All	Rilis Pertama

Daftar Isi

Riwayat Versi	3
Daftar Isi	4
Informasi Umum	6
Ekspektasi Hasil Pembelajaran	6
Studi Kasus	6
Studi Kasus – Fase 3 (C03)	6
Tutorial Implementasi Parsial Jaringan di Layanan Awan	7
Network Topology	7
Virtual Private Cloud & Subnetting	8
VPC Network Peering	13
Firewall Configuration	16
Virtual Machine Configuration	21
Deploy Web Server to Virtual Machine with Docker	25
Langkah 1: Memastikan Software Sudah Terupdate	26
Langkah 2: Menginstall Docker Engine	26
Langkah 3: Memberikan Akses ke Docker Engine	26
Langkah 4: Login Ulang	27
Langkah 5: Memastikan Docker Engine Telah Terinstall	27
Langkah 6: Melakukan Pull pada Docker Image	27
Langkah 7: Deployment pada Web Server	28
Langkah 8: Daftar Docker Container	29
Langkah 9: Mematikan Docker Container	30
Langkah 10: Merestart Docker Container	30
Langkah 11: Menghapus Docker Container	30
Langkah 12: Menghapus Docker Image	30
Network Trials	31
SSH	31
Deployment Web Server	31
Fase 3 (C03) : Implementasi Parsial Rancangan Solusi	33

Kriteria Dasar Implementasi.....	33
Ketentuan Pembuatan Dokumen	34
Peraturan.....	36
Keterlambatan	36
Plagiarisme	36

Studi Kasus – Fase 3 (C03)

IT Network Plan : Implementasi Parsial Rancangan Solusi

Informasi Umum

- Tipe Tugas : Kelompok
- Batas Waktu Pengumpulan : Jumat, 27 Desember 2024 17.00 Waktu SCellE
- Berkas yang Dikumpulkan :
 - **[Laporan]** Berkas laporan sistematis dalam format *paper* (.pdf)
 - **[Presentasi]** Ringkasan laporan dalam format *salindia/slide* presentasi (.pdf)
 - Tautan video presentasi
- Format Penamaan Berkas :
 - C03_[Nama Kelompok]_[Kode Jenis Dokumen].[Tipe Berkas]
Contoh: C03_Life At Jarkom_Laporan.pdf, C03_Life At Jarkom_Presentasi.pdf

Ekspektasi Hasil Pembelajaran

Studi Kasus

Tugas ini bertujuan agar mahasiswa mampu menganalisis, menilai dan mengimplementasi (C4) perancangan dan pemilihan teknologi yang dilakukan oleh organisasi untuk membangun infrastruktur sistem dan jaringan komputer.

Studi Kasus – Fase 3 (C03)

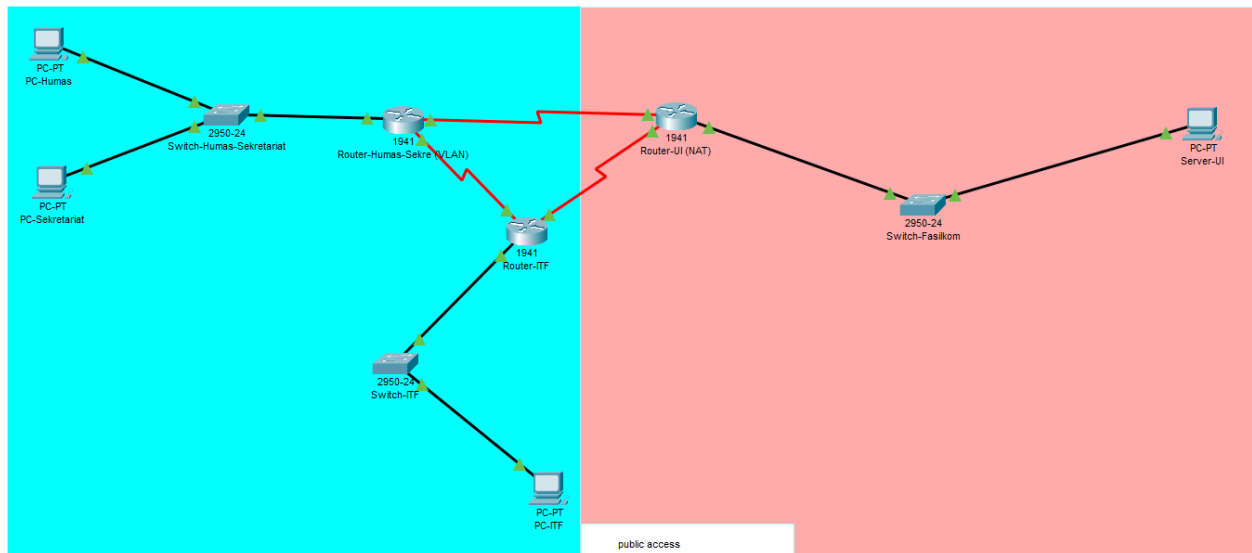
Mahasiswa diharapkan mampu melakukan **implementasi ulang (C4)** rancangan solusi sistem dan jaringan komputer melalui implementasi parsial pada layanan awan.

Tutorial Implementasi Parsial Jaringan di Layanan Awan

Sebelum memulai mengerjakan fase studi kasus ini, Anda dapat mencoba tutorial berikut ini untuk mendapatkan pemahaman mengenai cara mengimplementasikan jaringan di layanan awan. Dalam kasus ini, Anda akan menggunakan layanan Google Cloud Platform.

Network Topology

Untuk latihan kali ini, Anda akan mencoba berlatih untuk mengaplikasikan topologi jaringan pada Cisco Packet Tracer yang terlampir pada gambar 1.1 ke GCP.



Gambar 1.1

Diberikan alamat IP **10.x.0.0/25** untuk jaringan Fasilkom dan **10.y.0.0/21** untuk jaringan server-server UI yang berjalan yang di mana kedua jaringan tersebut berada di **satu jaringan utama yang sama**. Untuk distribusi alokasi alamat IP ke masing-masing subnet dapat dilihat pada tabel dibawah:

Subnet	Network Address	Default Gateway
Humas	10.x.0.0/28	10.x.0.1
Sekretariat	10.x.0.16/28	10.x.0.17
ITF	10.x.0.32/28	10.x.0.33
Server-UI	10.y.0.0/28	10.y.0.1

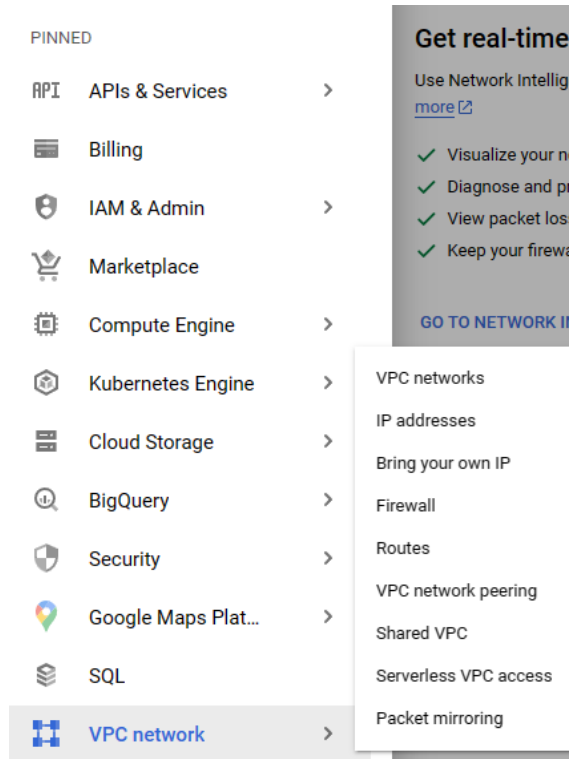
*catatan:

- nilai x diisi dengan hasil modulo 3 digit NPM Anda dengan nilai 256
- nilai y diisi dengan hasil floor dari pembagian nilai x dengan nilai 2

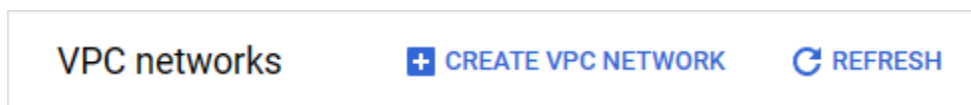
Virtual Private Cloud & Subnetting

Pada bagian ini, Anda akan mempelajari cara untuk membuat VPC Network dan subnet berdasarkan spesifikasi yang telah ditentukan sebelumnya. Adapun VPC network yang akan anda buat sebanyak **2 VPC Network**.

1. Silakan gunakan *project* yang telah anda buat pada **A01a**.
2. Pada menu navigasi silakan *hover* ke “VPC network” lalu pilih “VPC networks”.



3. Buat VPC network baru dengan menekan “Create VPC Network”.



4. Silakan isi setiap konfigurasi sesuai dengan spesifikasi di bawah.

Konfigurasi	Jawaban
Jaringan Fasilkom	
Name	<nama_singkat>-vpc-fasilkom-<NPM>
Description	*opsional

MTU		1460
VPC network ULA internal IPv6 range		Disabled
Subnet creation mode		Custom (klik “Add Subnet” untuk menambahkan subnet baru)
Subnet (Humas)	Name	subnet-humas
	Description	*opsional
	Region	us-west1
	IP stack type	IPv4 (single-stack)
	Ipv4 range	10.x.0.0/28
	Private Access Google	off
	Flow logs	off
Subnet (Sekretariat)	Name	subnet-sekretariat
	Description	*opsional
	Region	us-west1
	IP stack type	IPv4 (single-stack)
	Ipv4 range	10.x.0.16/28
	Private Access Google	off
	Flow logs	off
Subnet ITF	Name	subnet-itf
	Description	*opsional
	Region	us-west1
	IP stack type	IPv4 (single-stack)
	Ipv4 range	10.x.0.32/28
	Private Access Google	off
	Flow logs	off
Jaringan UI		
Name		<nama_singkat>-vpc-ui-<NPM>
Description		*opsional
MTU		1460
VPC network ULA internal IPv6 range		Disabled
Subnet creation mode		Custom (klik “Add Subnet” untuk menambahkan subnet baru)
Subnet-Server-UI	Name	subnet-server-ui
	Description	*opsional
	Region	us-west1
	IP stack type	Ipv4 (single-stack)
	Ipv4 range	10.y.0.0/28

	Private Access	Google	off
	Flow logs		off

*Untuk konfigurasi **Firewall rules** dan **Dynamic routing mode** silakan diisi nilai *default*.

Contoh pengisian:

Name *
 naufal-vpc-fasilkom-1906299010 ?

❗ Name is already in use

Description

Maximum transmission unit (MTU)
 1460 ▼ ?

VPC network ULA internal IPv6 range ?
 Enabling this feature will assign a /48 from Google defined ULA prefix fd20::/20.
☐ Enabled
☒ Disabled

Subnets
 Subnets let you create your own private cloud topology within Google Cloud. Click Automatic to create a subnet in each region, or click Custom to manually define the subnets. [Learn more](#)

Subnet creation mode ?
☒ Custom
☐ Automatic

Subnets

Subnets let you create your own private cloud topology within Google Cloud. Click Automatic to create a subnet in each region, or click Custom to manually define the subnets. [Learn more](#)

Subnet creation mode

☒ Custom

☐ Automatic

Edit subnet

Name *

subnet-humas

Lowercase letters, numbers, hyphens allowed

Description

Region *

us-west1

IP stack type

☒ IPv4 (single-stack)

☐ IPv4 and IPv6 (dual-stack)

IPv4 range *

10.10.0.0/28

E.g. 10.0.0.0/24

CREATE SECONDARY IPV4 RANGE

Private Google Access

☐ On

☒ Off

Flow logs

Turning on VPC flow logs doesn't affect performance, but some systems generate a large number of logs, which can increase costs in Logging. [Learn more](#)

☐ On

Firewall rules

Select any of the firewall rules below that you would like to apply to this VPC network. Once the VPC network is created, you can manage all firewall rules on the Firewall rules page.

IPv4 FIREWALL RULES

<input type="checkbox"/>	Name	Type	Targets	Filters	Protocols / ports	Action	Priority
<input type="checkbox"/>	naufal-vpc-1906299010-allow-custom	Ingress	Apply to all	IP ranges: 10.10.0.0/28 10.10.0.16/28 10.10.0.32/28 20.10.0.0/24	all	Allow	65,534
<input type="checkbox"/>	naufal-vpc-1906299010-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65,534
<input type="checkbox"/>	naufal-vpc-1906299010-allow-rdp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65,534
<input type="checkbox"/>	naufal-vpc-1906299010-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65,534
	naufal-vpc-1906299010-deny-all-ingress	Ingress	Apply to all	IP ranges: 0.0.0.0/0	all	Deny	65,535
	naufal-vpc-1906299010-allow-all-egress	Egress	Apply to all	IP ranges: 0.0.0.0/0	all	Allow	65,535


Dynamic routing mode

- ☒ **Regional**
Cloud Routers will learn routes only in the region in which they were created
- ☐ **Global**
Global routing lets you dynamically learn routes to and from all regions with a single VPN or interconnect and Cloud Router

Berikut penjelasan penting dari masing-masing konfigurasi:

Konfigurasi	Penjelasan
Maximum transmission unit (MTU)	Batas ukuran paket data (dalam byte) terbesar yang bisa didukung pada protokol <i>network layer</i> yang sudah termasuk <i>header</i> dan <i>data</i> . Saat VPC sudah dibuat, Anda tidak dapat mengubah konfigurasi ini karena jika diubah akan berpotensi <i>packet loss</i> dan tidak didukung oleh <i>console</i> .
Subnet creation mode	Menentukan jenis pembuatan subnet, Jika memilih automatic maka subnet dibuat secara otomatis berdasarkan <i>network address</i> dari masing-masing <i>region</i> yang tersedia oleh GCP.
IP stack type	Memilih jenis ip yang digunakan oleh subnet, Anda dapat memilih IPv4 dan IPv6 namun untuk rentang IPv6 sudah ditetapkan oleh GCP dengan <i>prefix /64</i> .
Private Google Access	Menentukan apakah subnet dapat mengakses layanan google/internet tanpa mengalokasikan alamat IP eksternal.
Dynamic routing mode	Saat Anda memilih global , Hal ini akan menyebabkan semua subnet terlepas dari <i>region</i> pada Subnet yang dipilih dan akan diperkenalkan ke router dan <i>region</i> lokal Anda saat menggunakan <i>router cloud</i> . Hal ini memungkinkan Anda hanya perlu satu VPN dengan <i>router cloud</i> untuk mengenali <i>route</i> secara dinamis dari semua <i>Google Cloud region</i> pada jaringan.

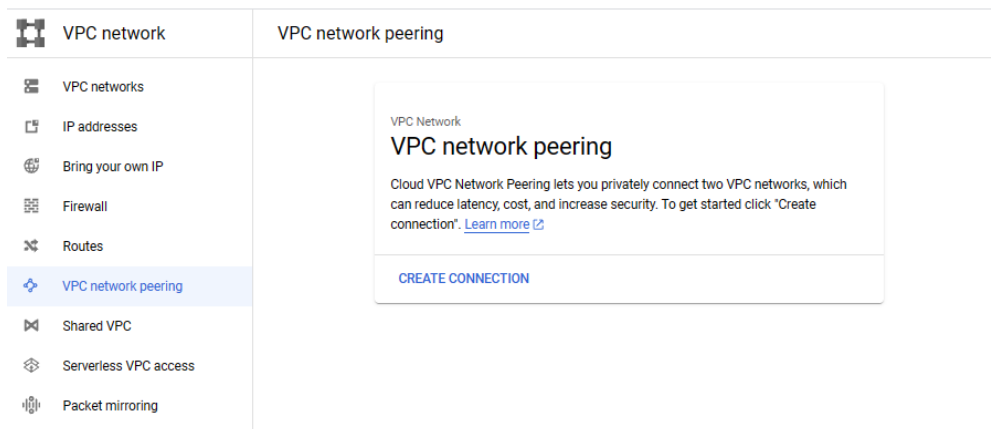
5. Jika sudah diisi silakan klik “*Create*” untuk membuat VPC network.
6. Saat VPC network berhasil dibuat maka akan ditampilkan pada halaman *dashboard* VPC networks.

VPC networks							
<div>  Filter Enter property name or value </div>							
Name ↑	Subnets	MTU ?	Mode	Internal IP ranges	Gateways	Firewall rules	Global dynamic routing
default	41	1460	Auto			7	Off
naufal-vpc-fasilkom-1906299010	3	1460	Custom			0	Off
naufal-vpc-ui-1906299010	1	1460	Custom			0	Off

VPC Network Peering

Untuk melakukan *routing* antar VPC, Anda dapat memanfaatkan *VPC Network Peering* yang memungkinkan konektivitas antar VPC melalui Internal alamat IP. Aliran *traffic* antar VPC ini tidak melalui internet, melainkan melalui jaringan internal Google. Anda dapat bertukar *route* secara otomatis dalam bentuk *static* maupun *dynamic* yang bergantung terhadap konfigurasi *export/import custom routes* saat *peering*. Keunggulan *VPC Network Peering* adalah *peering traffic* (*traffic* mengalir antar VPC yang tersambung) memiliki *latency*, *throughput*, dan *availability* yang sama layaknya sebuah *traffic* internal pada jaringan yang sama. Namun, fitur ini hanya bisa membuat konektivitas melalui IPv4 dan CIDR *range* dari alamat IP kedua VPC yang tidak boleh tumpah tindih (*overlap*). Berikut langkah-lankah dalam pembuatan *VPC Peering*:

1. Pada halaman *dashboard* “VPC network” pilih “VPC Network peering”.



2. Klik “Create Connection” untuk membuat *network peering* (Jika ada konfirmasi silakan klik “Continue”). Disini Anda akan membuat dua *network peering* agar bisa saling bertukar antar VPC satu sama lain.

← Create peering connection

1 Your VPC network will be fully connected to the peered VPC network (full mesh topology). Routes to subnets in the peered VPC network will be automatically created.

Name * ?
Lowercase letters, numbers, hyphens allowed

Your VPC network * ?

Peered VPC network

☒ In project jarkom-jarkomdat
☐ In another project

VPC network name * ?

☒ IPv4 (single-stack) ?
☐ IPv4 and IPv6 (dual-stack) ?

Exchange IPv4 custom routes ?
 You can choose to import or export static and dynamic routes over the VPC peering connection

☐ Import custom routes ?
☐ Export custom routes ?

Exchange subnet routes with privately used public IPv4 addresses ?
 You can choose to import or export subnet routes with public IP over the VPC peering connection

☐ Import subnet routes with privately used public IPv4 addresses ?
☒ Export subnet routes with privately used public IPv4 addresses ?

CREATE **CANCEL**

3. Silakan isi setiap konfigurasi sesuai dengan spesifikasi berikut:

Konfigurasi	Jawaban
Network Peering pertama	
Name	peer-vpc-fasilkom-and-vpc-ui
Your VPC Network	vpc fasilkom
Peered VPC Network	*pilih project pada A01a
VPC network name	vpc ui
IP address	IPv4 (single-stack)
Exchange custom routes	*tidak di ceklis keduanya
Exchange subnet routes with Public IP	Export subnet routes with public IP
Network Peering kedua	
Name	peer-vpc-ui-and-vpc-fasilkom
Your VPC Network	vpc ui
Peered VPC Network	*pilih project pada A01a
VPC network name	vpc fasilkom
IP address	IPv4 (single-stack)
Exchange custom routes	*tidak di ceklis keduanya
Exchange subnet routes with Public IP	Export subnet routes with public IP

*Catatan: untuk konfigurasi lainnya silakan diisi *default*.

Contoh:

← Create peering connection

1 Your VPC network will be fully connected to the peered VPC network (full mesh topology). Routes to subnets in the peered VPC network will be automatically created.

Name *
peer-vpc-ui-and-vpc-fasilikom
Lowercase letters, numbers, hyphens allowed

Your VPC network *
naufal-vpc-ui-1906299010

Peered VPC network
☒ In project jarkom-jarkomdat
☐ In another project

VPC network name *
naufal-vpc-fasilikom-1906299010

☒ IPv4 (single-stack)
☐ IPv4 and IPv6 (dual-stack)

Exchange IPv4 custom routes
 You can choose to import or export static and dynamic routes over the VPC peering connection
☐ Import custom routes
☐ Export custom routes

Exchange subnet routes with privately used public IPv4 addresses
 You can choose to import or export subnet routes with public IP over the VPC peering connection
☐ Import subnet routes with privately used public IPv4 addresses
☒ Export subnet routes with privately used public IPv4 addresses

Berikut penjelasan penting dari masing-masing konfigurasi:

Konfigurasi	Penjelasan
Exchange IPv4 custom routes	Secara <i>default</i> , <i>peering connection</i> hanya menukar rute subnet namun Anda dapat melakukan kostuminasi rute subnet dengan melakukan import/export rute IPv4. Rute yang menggunakan <i>instance tags</i> atau rute <i>internet gateway</i> tidak akan di import/export.
Exchange subnet routes with privately used public IPv4 addresses	VPC <i>peering</i> akan selalu bertukar rute subnet yang tidak menggunakan alamat IP publik yang digunakan secara pribadi. Anda dapat melakukan import jika ingin melakukan rute subnet dengan menggunakan alamat IP publik yang digunakan secara menerima rute dari VPC tujuan yang akan mengeksponnya.

4. Klik “Create” untuk menyimpan konfigurasi *network peering*, jika berhasil maka akan ditampilkan di halaman *VPC network peering*.

VPC network peering								
CREATE PEERING CONNECTION REFRESH DELETE								
Filter Enter property name or value								
<input type="checkbox"/> Name ↑	Your VPC network	Peered VPC network	Peered project ID	Status	IP stack type	Custom routes	Subnet routes with public IPv4	
<input type="checkbox"/> peer-vpc-fasilkom-and-vpc-ui	naufal-vpc-fasilkom-1906299010	naufal-vpc-ui-1906299010	jarkom-jarkomdat	Active	IPv4	None	Export subnet routes with public IP	⋮
<input type="checkbox"/> peer-vpc-ui-and-vpc-fasilkom	naufal-vpc-ui-1906299010	naufal-vpc-fasilkom-1906299010	jarkom-jarkomdat	Active	IPv4	None	Export subnet routes with public IP	⋮

*Catatan: Jika anda baru membuat satu *network peering* saja maka status *network peering* tersebut adalah **inactive**, hal tersebut disebabkan karena belum ada pertukaran *routes* antar VPC sehingga diperlukan untuk membuat *network peering* dengan *source* dan *destination* VPC yang ditukar agar bisa melakukan pertukaran *routes* dan *peer* antar VPC serta merubah status kedua VPC menjadi **active**. VPC yang telah di-*peer* akan selalu otomatis bertukar *routes* pada alamat IP internal dari subnet-subnetnya.

Firewall Configuration

VPC yang telah Anda buat sebelumnya secara *default* tidak memiliki konfigurasi *firewall* apapun. Tujuan penggunaan *firewall* adalah untuk pembatasan akses port *end device* yang dilakukan oleh *end device* lainnya. Adapun konfigurasi *firewall* yang akan Anda implementasikan pada kedua VPC sebagai berikut:

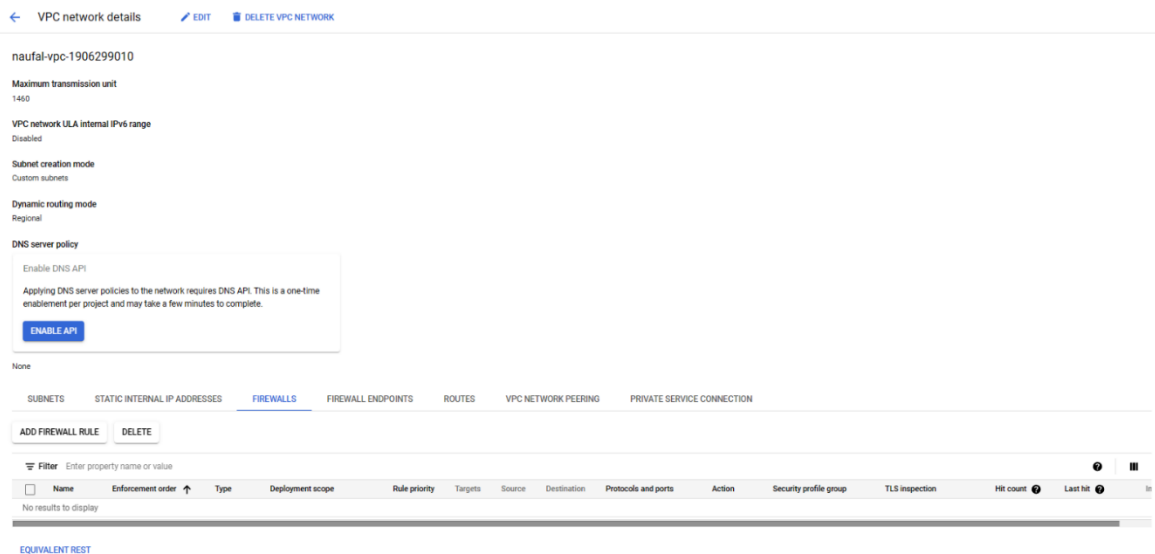
Firewall	Konfigurasi		Penjelasan
	Field	Jawaban	
vpc ui			
allow-ssh-from-internet	Name	allow-ssh-from-internet-2	
	Description	*optional	
	Logs	off	
	Network	vpc ui	
	Priority	1000	
	Direction of traffic	ingress	
	Action on match	allow	
	Targets	specified target tags	
	Target tags	ssh-public-2	
	Source filter	IPv4 ranges	
	Source IPv4 ranges	0.0.0.0/0	
	Second source filter	none	
	Destination filter	none	
	Protocols and ports	TCP: 22 (ssh only)	
Protocols and ports	other: icmp		

allow-http-from-internet	Name	allow-http-from-internet-2	
	Description	*optional	
	Logs	Off	
	Network	vpc ui	
	Priority	1000	
	Direction of traffic	ingress	
	Action on match	allow	
	Targets	specified target tags	
	Target tags	http-public-2	
	Source filter	IPv4 ranges	
	Source IPv4 ranges	0.0.0.0/0	
	Second source filter	none	
	Destination filter	none	
	Protocols and ports	TCP: 80 (http only)	
vpc Fasilkom			
allow-ssh-from-subnet-itf	Name	allow-ssh-from-itf	
	Description	*optional	
	Logs	Off	
	Network	vpc fasilkom	
	Priority	1000	
	Direction of traffic	Ingress	
	Action on match	Allow	
	Targets	specified target tags	
	Target tags	ssh-itf	
	Source filter	IPv4 ranges	
	Source IPv4 ranges	10.x.0.32/28	
	Second source filter	None	
	Destination filter	None	
	Protocols and ports	Tcp: 22 (ssh only)	
allow-ssh-from-internet	Name	allow-ssh-from-internet	
	Description	*optional	
	Logs	Off	
	Network	vpc fasilkom	
	Priority	1000	
	Direction of traffic	Ingress	
	Action on match	Allow	
	Targets	specified target tags	
	Target tags	ssh-public	

	Source filter	IPv4 ranges	
	Source IPv4 ranges	0.0.0.0/0	
	Second source filter	None	
	Destination filter	None	
	Protocols and ports	TCP: 22 (ssh only)	

Berikut langkah-langkah untuk pembuatan *firewall*:

1. Masuk ke halaman dashboard *VPC network* yang telah Anda buat, lalu pilih tabel “*firewalls*”



2. Klik “*Add Firewall*” untuk menambahkan *firewall*, kemudian masukkan setiap konfigurasi firewall yang telah anda tentukan.

Contoh:

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. [Learn more](#)

Name *
allow-ssh-from-internet ?
Lowercase letters, numbers, hyphens allowed

Description

Logs

Turning on firewall logs can generate a large number of logs which can increase costs in Logging. [Learn more](#)

☐ On

☒ Off

Network *
naufal-vpc-1906299010 ?

Priority *
1000 [CHECK PRIORITY OF OTHER FIREWALL RULES](#) ?
Priority can be 0 - 65535

Direction of traffic ?

☒ Ingress

☐ Egress

Action on match ?

☒ Allow

☐ Deny

Targets
Specified target tags

Target tags *
ssh-public

Source filter
IPv4 ranges

Source IPv4 ranges *
0.0.0.0/0

Second source filter
None

Destination filter
None

Protocols and ports ?

☐ Allow all

☒ Specified protocols and ports

☒ TCP

Ports
22
E.g. 20, 50-60

☐ UDP

Ports
E.g. all

☐ Other

Protocols
Separate multiple protocols by commas, e.g. ah, sctp

▼ DISABLE RULE

CREATE CANCEL

Berikut penjelasan penting dari masing-masing konfigurasi:

Konfigurasi	Penjelasan
Priority	Menentukan skala prioritas dari penerapan <i>firewall</i> pada jaringan yang diterapkan oleh <i>instance</i> . Skala prioritas dengan nilai rendah maka akan diprioritaskan terlebih dahulu.
Direction of traffic	Menentukan <i>traffic</i> jaringan yang diterapkan pada <i>instance</i> di mana untuk ingress untuk <i>traffic</i> manetwsuk dan egress untuk <i>traffic</i> keluar.
Action on match	Mengatur aturan <i>traffic</i> jaringan pada <i>instance</i> dimana allow memungkinkan <i>instance</i> untuk menerima <i>traffic</i> jaringan dan deny untuk

	menolak <i>traffic</i> jaringan berdasarkan protokol atau port yang telah ditentukan.
--	---

3. Jika setiap konfigurasi sudah diisi silakan klik “*create*” untuk membuat membuat *firewall*.
4. *Firewall* berhasil dibuat dan akan ditampilkan di tabel *firewalls*.

SUBNETS STATIC INTERNAL IP ADDRESSES FIREWALLS FIREWALL ENDPOINTS ROUTES VPC NETWORK PEERING PRIVATE SERVICE CONNECTION												
ADD FIREWALL RULE DELETE												
Filter Enter property name or value												
<input type="checkbox"/>	Name	Enforcement order ↑	Type	Deployment scope	Rule priority	Targets	Source	Destination	Protocols and ports	Action	Security profile group	TLS inspection
	▼ vpc-firewall-rules		VPC firewall rules	Global								
<input type="checkbox"/>	allow-ssh-from-internet	1	Ingress firewall rule	Global	1000	Tags...	IPv4 range	—	tcp:22	Allow	—	—

*Catatan: pastikan setiap penamaan vpc dan target tags adalah unik.

Virtual Machine Configuration

Pada pembuatan *virtual machine* atau *instance* langkah-langkahnya sama seperti yang telah anda lakukan pada pengerjaan **A01a** dengan penyesuaian beberapa *field* yang sesuai dengan spesifikasi yang diperlukan. Adapun spesifikasi *virtual machine* yang Anda buat pada topologi jaringan ini sebagai berikut:

Konfigurasi	PC-Humas	PC-Sekre	PC-ITF	Webserver-UI
Name	pc-humas	pc-sekre	pc-itf	webserver-ui
Region	us-west1 (Oregon)			
Zone	us-west1-b			
Machine series	E2			
Machine type	E2-micro			
Operating System	Ubuntu 24.04 LTS (x86/64, amd64)			
Boot disk	Balanced persistent disk; size 10 GB			
Network tags	ssh-itf	ssh-itf	ssh-public	ssh-public-2, http-public-2
Network Interface	vpc fasilkom			vpc ui
Subnetwork	10.x.0.0/28	10.x.0.16/28	10.x.0.32/28	10.y.0.0/28
Network Service Tier	Premium			

*Untuk konfigurasi lain silakan diisi dengan nilai *default*.

Contoh:

Name *
server-pacil

MANAGE TAGS AND LABELS

Region *
us-west1 (Oregon)
Region is permanent

Zone *
us-west1-b
Zone is permanent

Machine configuration

General purpose

Compute optimized

Memory optimized

GPUs

Machine types for common workloads, optimized for cost and flexibility

Series	Description	vCPUs	Memory	Platform
<input type="radio"/> C3	Consistently high performance	4 - 176	8 - 1,408 GB	Intel Sapphire Rapids
<input type="radio"/> C3D	Consistently high performance	4 - 360	8 - 2,880 GB	AMD Genoa
<input checked="" type="radio"/> E2	Low cost, day-to-day computing	0.25 - 32	1 - 128 GB	Based on availability
<input type="radio"/> N2	Balanced price & performance	2 - 128	2 - 864 GB	Intel Cascade and Ice Lake
<input type="radio"/> N2D	Balanced price & performance	2 - 224	2 - 896 GB	AMD EPYC
<input type="radio"/> T2A	Scale-out workloads	1 - 48	4 - 192 GB	Ampere Altra Arm
<input type="radio"/> T2D	Scale-out workloads	1 - 60	4 - 240 GB	AMD EPYC Milan
<input type="radio"/> N1	Balanced price & performance	0.25 - 96	0.6 - 624 GB	Intel Skylake

Machine type

Choose a machine type with preset amounts of vCPUs and memory that suit most workloads.
Or, you can create a custom machine for your workload's particular needs. [Learn more](#)

PRESET

CUSTOM

e2-micro (2 vCPU, 1 core, 1 GB memory)

vCPU
0.25-2 vCPU (1 shared core)

Memory
1 GB

ADVANCED CONFIGURATIONS

Availability policies

VM provisioning model

Standard

Choose "Spot" to get a discounted, preemptible VM. Otherwise, stick to "Standard". [Learn more](#)

VM PROVISIONING MODEL ADVANCED SETTINGS

Display device

Enable to use screen capturing and recording tools.

☐ Enable display device

Confidential VM service

Confidential Computing is disabled on this VM instance

ENABLE



22 of 36

Container

Deploy a container image to this VM instance

DEPLOY CONTAINER

Boot disk

Name	server-pacil
Type	New balanced persistent disk
Size	10 GB
License type 	Free
Image	 Ubuntu 22.04 LTS

CHANGE

Identity and API access

Service accounts

Service account

Compute Engine default service account

Requires the Service Account User role (roles/iam.serviceAccountUser) to be set for users who want to access VMs with this service account. [Learn more](#)

Access scopes

- ☒ Allow default access
- ☐ Allow full access to all Cloud APIs
- ☐ Set access for each API

Firewall

Add tags and firewall rules to allow specific network traffic from the Internet

- ☐ Allow HTTP traffic
- ☐ Allow HTTPS traffic
- ☐ Allow Load Balancer Health Checks

Observability - Ops Agent

Monitor your system through collection of logs and key metrics.

☐ Install Ops Agent for Monitoring and Logging

Advanced options

Networking

Hostname and network interfaces

Network tags

ssh-public 

icmp-public 

http-public 



Hostname



Set a custom hostname for this instance or leave it default. Choice is permanent

IP forwarding

☐ Enable

Network performance configuration

Network interface card

—



Network bandwidth

☐ Enable per VM Tier_1 networking performance

Maximum outbound network bandwidth: 1Gbps

VM to Public IP: 1Gbps

Network interfaces ?

Network interface is permanent

Edit network interface

Network *

naufal-vpc-1906299010

Subnetwork *

server-fasilkom IPv4 (20.10.0.0/24)

1

To use IPv6, you need an IPv6 subnet range. [LEARN MORE](#)

IP stack type

☒ IPv4 (single-stack)

☐ IPv4 and IPv6 (dual-stack)

Primary internal IPv4 address

Ephemeral (Automatic)

Alias IP ranges

+ ADD IP RANGE

External IPv4 address

Ephemeral

Network Service Tier

☒ Premium ?

☐ Standard (us-west1) ?

Public DNS PTR Record ?

☐ Enable for IPv4

PTR domain name

Deploy Web Server to Virtual Machine with Docker

Agar pengguna bisa mengunjungi *web server* yang telah Anda buat, Anda perlu melakukan *deployment website* ke perangkat yang Anda tuju. Terdapat berbagai metode *deployment* yang dapat Anda lakukan salah satunya adalah *deployment* ke *platform* seperti Heroku, Vercel, Netlify, dan sebagainya. Pada Latihan kali ini anda akan mencoba untuk melakukan *deployment* ke *virtual machine* Anda menggunakan Docker dengan control penuh atas mesin *deployment*-nya.



Docker merupakan *platform open source* untuk mengembangkan, mengirim, dan menjalankan aplikasi. Docker memungkinkan Anda untuk untuk memisahkan aplikasi Anda dari infrastrukturnya. Pada dasarnya, Docker adalah *platform* virtualisasi yang memungkinkan *software (web server)* ditempatkan

dalam *container* yang akan dikemas dan dieksekusi dalam lingkungan yang terisolasi. *Platform* ini memungkinkan Anda membuat *docker image* siap pakai yang akan menjadi *template* bagi *docker container* untuk melakukan *deployment*. Pada latihan ini, Anda akan menjalankan langkah-langkah berikut di *instance GCP webserver-ui*.

Langkah 1: Memastikan Software Sudah Terupdate

Anda disarankan untuk memastikan bahwa *software* yang telah terinstal pada *instance GCP* telah *terupdate*. Langkah ini bersifat tidak wajib namun merupakan *best practice* untuk memastikan *software* yang ada telah *terupdate* ke versi terbaru.

```
sudo apt update && sudo apt upgrade -y
```

Langkah 2: Menginstall Docker Engine

Docker engine menjalankan *container* yang akan ditempatkan dalam *instance GCP*. Ada beberapa cara untuk menginstall *docker engine* namun Docker telah menyediakan *script* yang berisi seluruh *installation command*. Pada latihan ini, Anda akan menggunakan *script* tersebut untuk menginstall Docker. Anda dapat mencoba menginstall Docker menggunakan cara lain [di sini](#).

Script instalasi Docker di-hosting pada <https://get.docker.com> dan dapat diakses melalui *web browser* jika Anda ingin melihat cara kerjanya. Untuk menginstall docker, Anda dapat mengunduh *script* dan menjalankannya langsung di *instance GCP* Anda sendiri. Anda dapat menjalankan *script* dengan *option* '--dry-run' saat menjalankan *script* untuk mempelajari langkah-langkah *script* yang akan dijalankan saat dipanggil:

```
curl -fsSL https://get.docker.com -o get-docker.sh  
sudo sh ./get-docker.sh
```

```
mochammad_naufal91@webserver-ui:~$ sudo sh ./get-docker.sh  
# Executing docker install script, commit: e5543d473431b782227f8908005543bb4389b8de  
+ sh -c apt-get update -qq >/dev/null  
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null  
+ sh -c install -m 0755 -d /etc/apt/keyrings  
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | gpg --dearmor --yes -o /etc/apt/keyrings/docker.gpg  
+ sh -c chmod a+r /etc/apt/keyrings/docker.gpg  
+ sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu jammy stable" > /etc/apt/sources.list.d/docker.list  
+ sh -c apt-get update -qq >/dev/null  
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq docker-ce docker-ce-cli containerd.io docker-compose-plugin docker-ce-rootless-extras docker-buildx-plugin >/dev/null
```

Script dibuat untuk berjalan tanpa pengawasan sehingga Anda hanya perlu menunggu sampai *script* selesai dijalankan. Perintah yang sedang dijalankan akan ditampilkan di terminal.

Langkah 3: Memberikan Akses ke Docker Engine

Untuk tujuan keamanan, tidak semua user diizinkan untuk mengakses *docker engine* secara langsung. Secara *default* hanya *root user* dan *escalated user* (melalui *sudo*) yang dapat mengakses *docker engine*. Docker menyediakan *user group* yang dapat mengakses *docker engine*.

Untuk mendaftarkan *user* saat ini ke *user group* tersebut, jalankan *command* berikut (perhatikan bahwa `$USER` adalah variabel shell yang menyimpan nama *user* saat ini):

```
sudo usermod -aG docker $USER
```

```
mohammad_naufal91@webserver-ui:~$ sudo usermod -aG docker $USER
```

Langkah 4: Login Ulang

Pendaftaran *user* ke *user group* docker tidak langsung ter-*apply*. Jika Anda menggunakan SSH, tutup koneksi SSH saat ini dan lakukan login ulang. Anda dapat menggunakan perintah 'exit' untuk memutuskan koneksi SSH saat ini.

Langkah 5: Memastikan Docker Engine Telah Terinstall

Setelah *user* saat ini terdaftar dengan benar di *user group* docker, *user* tersebut sekarang dapat dengan bebas mengakses *docker engine* tanpa memerlukan *privilege escalation* atau menggunakan *root user*. Untuk memverifikasi penginstalan *docker engine* dan *user* saat ini memiliki akses terhadapnya, jalankan perintah berikut:

```
docker -v
```

```
mohammad_naufal91@webserver-ui:~$ docker -v  
Docker version 24.0.7, build afdd53b
```

Langkah 6: Melakukan Pull pada Docker Image

Docker image adalah *template* yang akan Anda gunakan untuk menyebarkan aplikasi Anda. Anda dapat membuat *docker image* Anda sendiri dari kode aplikasi dan mendeploynya dengan Docker. Pada latihan ini, Tim Asdos Jarkom x Jarkomdat telah menyediakan *docker image* webserver-lab berbasis http1 sehingga Anda akan melakukan *pull docker image* tersebut.

Secara umum, docker melakukan pull dari Docker Hub yang dapat diakses pada link [berikut](#). Ada beberapa *provider docker image* lainnya, seperti AWS Elastic Container Registry dan GCP Container Registry. *Docker image* yang akan digunakan dipublikasikan melalui Docker Hub dan diidentifikasi dengan nama **compnetcsui/webserver-lab-http1:latest**. *Command* ``docker pull`` akan mengunduh versi terbaru (atau versi yang telah ditentukan) dari *image* tersebut dan menyimpannya di mesin. *Command* berikut digunakan untuk melakukan pull pada *docker image* dari Docker Hub:

```
docker pull <image name>:<tag>
```

```
# Dalam kasus ini: docker pull compnetcsui/webserver-lab-http1:latest
```

```
mohammad_naufal91@webserver-ui:~$ docker pull compnetcsui/webserver-lab-http1:latest
latest: Pulling from compnetcsui/webserver-lab-http1
7264a8db6415: Pull complete
33fbadb7f8c6: Pull complete
5558ef53d261: Pull complete
3ae8a7119b2e: Pull complete
bceb9a572618: Pull complete
91ba017c9d90: Pull complete
0c170b3406cb: Pull complete
Digest: sha256:8d170e18d54cd70cbdebc5fbba4f28136ec4290653614aa80639418d42e7fab9
Status: Downloaded newer image for compnetcsui/webserver-lab-http1:latest
docker.io/compnetcsui/webserver-lab-http1:latest
```

Selanjutnya Anda dapat mengecek daftar *docker image* yang telah berhasil di-pull dengan menjalankan perintah berikut:

docker images

```
mohammad_naufal91@webserver-ui:~$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
compnetcsui/webserver-lab-http1  latest      660f0182b176     2 months ago    192MB
```

Langkah 7: Deployment pada Web Server

Pada sebelumnya, Docker telah mendapatkan *image* untuk *web server* yang disediakan oleh tim Asdos Jarkom x Jarkomdat. Pada langkah ini, Anda akan menjalankan beberapa *command* untuk membuat container dan menjalankan web server di dalamnya.

```
docker run --name webserver-lab-http1 --restart <restart policy> -e ID=<your npm> -p 80:8080 -d <image name>
```

```
mohammad_naufal91@webserver-ui:~$ docker run --name webserver-lab-http1 --restart unless-stopped -e UID=190629
9010 -p 80:8080 -d compnetcsui/webserver-lab-http1
0a7617791236fccf57b06152627d22ec3bf7335d08d08e545c90642ffa3aa4ec
```

Ada beberapa *option* yang digunakan dalam penyebaran *web server* ini:

1. **--name <identifier>**

Option ini digunakan untuk memberikan *unique identifier* ke *container* yang Anda jalankan. Hal ini demi menghindari masalah umum dalam menjalankan banyak *container* dari *image* yang sama dalam satu mesin.

2. **--restart <restart policy>**

Docker menyediakan *option restart* untuk menentukan apakah *container* Anda dimulai secara otomatis saat *container* tersebut keluar, atau saat Docker di-restart. Berikut adalah nilai yang mungkin dari *option* '--restart':

- **no:** Jangan merestart container secara otomatis (default).
- **on-failure[:max-retries]:** Restart container jika keluar karena error (exit code selain 0). Anda juga dapat membatasi berapa kali daemon Docker mencoba untuk merestart container menggunakan opsi ':max-retries' secara opsional.
- **always:** Selalu restart container jika berhenti. Jika container dihentikan secara manual, container direstart hanya ketika Docker daemon direstart atau container itu sendiri direstart secara manual.
- **unless-stopped:** Mirip dengan 'always', kecuali ketika container dihentikan (baik secara manual ataupun tidak), container tidak direstart bahkan setelah Docker daemon direstart. Ini adalah option yang paling direkomendasikan untuk image ini.

3. **-e <KEY=VALUE>:** Environment Variables

Option ini digunakan untuk memasukkan *environment variable* dengan nama variabel KEY dan nilai VALUE ke dalam *container*. Dalam hal ini, *environment variable* UID akan diberikan nilai NPM Anda sendiri

4. **-p <HostPort>:<ContainerPort>:** Port Forwarding

Secara default, saat Anda menjalankan container menggunakan 'docker run', container tidak membuka port apa pun ke luar. Untuk membuat port dapat diakses dari luar, atau dari container Docker lainnya yang berjalan pada network berbeda, gunakan flag '--publish' atau '-p'. Flag ini membuat aturan firewall yang memetakan port pada *host* Docker (*instance GCP*) ke port container. Perintah -p 80:8080 akan meneruskan koneksi apa pun yang dibuat ke port 80 (HTTP) milik *host* (*instance GCP*) menuju port 8080 *web server* dalam *container*.

5. **-d: Detached Mode**

Dalam *detached mode*, *container* dijalankan di *background* dan id milik *container* akan di-print. *Container* yang dimulai dalam mode ini keluar ketika *root process* yang digunakan untuk menjalankan *container* keluar. Namun, jika *option* ini tidak digunakan, eksekusi container akan mengendalikan *shell* saat ini, sehingga apabila shell ditutup, maka container juga akan keluar. Jika *container* memiliki *interactive shell*, maka *option* ini seharusnya dihilangkan. Namun, web server yang kita gunakan dimaksudkan untuk dijalankan secara otomatis.

Langkah 8: Daftar Docker Container

Dalam skenario asli, Anda perlu memastikan bahwa *container* Anda aktif dan berjalan dengan baik. Docker menyediakan beberapa *command* yang dapat digunakan untuk memeriksa *container* yang sedang aktif. Salah satunya adalah 'docker ps', yang digunakan untuk mencetak daftar *container*.

docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0a7617791236	compnetsul/webserver-lab-http1	"docker-entrypoint.s..."	22 seconds ago	Up 21 seconds	0.0.0.0:80->8080/tcp, :::80->8080/tcp	webserver-lab-http1

Anda dapat melihat informasi tentang container, termasuk CONTAINER ID dan nama *container*.

Langkah 9: Mematikan Docker Container

Best practice dalam menggunakan Docker adalah menghentikan *container* yang tidak digunakan. *Docker container* menggunakan resource bahkan saat *idle* karena Anda menjalankan *platform* sendiri di dalam mesin Anda. Meskipun Anda bisa saja mematikan seluruh mesin (dalam kasus ini *instance GCP*) karena mesinnya hanya digunakan untuk meng-*host web server*, Anda tidak dapat melakukan hal itu apabila mesinnya ingin digunakan untuk keperluan lain. Anda harus bisa menghentikan *container* tanpa mematikan seluruh mesin. *Command* berikut menghentikan *docker container*:

```
docker stop <container name | container id>
```

```
mohammad_naufal91@server-pacil:~$ docker stop 66f38ca84bbf  
66f38ca84bbf
```

Langkah 10: Merestart Docker Container

Untuk me-*restart* kembali *docker container* yang sudah Anda hentikan dapat menggunakan *command* berikut:

```
docker restart <container name | container id>
```

```
mohammad_naufal91@webserver-ui:~$ docker restart 11dc74c8cc9d  
11dc74c8cc9d  
mohammad_naufal91@webserver-ui:~$
```

Langkah 11: Menghapus Docker Container

Meskipun Anda sudah menghentikan *docker container* sehingga akses ke *container* pun terputus, file *container* tersebut masih disimpan oleh Docker. Untuk memastikan *container* benar-benar dihapus, jalankan *command* berikut:

```
docker rm <container name | container id>
```

```
mohammad_naufal91@webserver-ui:~$ docker rm 0a7617791236  
0a7617791236
```

Langkah 12: Menghapus Docker Image

Untuk menghapus *docker images* yang tersimpan pada Docker, anda perlu memastikan apakah semua *container* yang menggunakan *image* yang akan dihapus sudah benar-benar terhapus pada Docker. Jika Anda sudah menghapus semua *container* yang berkaitan maka anda dapat menghapus *image* dengan menjalankan *command* berikut:

```
docker rmi <image name | image id>
```

```

mohammad_naufal91@webserver-ui:~$ docker rmi 660f0182b176
Untagged: compnetcsui/webserver-lab-http1:latest
Untagged: compnetcsui/webserver-lab-http1@sha256:8d170e18d54cd70cbdebc5fbb4f28136ec4290653614aa80639418d42e7fab9
Deleted: sha256:660f0182b1764a7528cfdc95ed940de827e669f8537bdc47fa87f6f91aafb0b2
Deleted: sha256:90aa51d3fcc7ffeb915adf0c0de394945f88d1d2d48b62c9dcc433bd7f17c79c
Deleted: sha256:ffb62af9c9e81a45852caf94c6cfe89bc431c75e138e8987d7fda63fb7c99e90
Deleted: sha256:53f1eef652567692e0a86a08a8b9b2c68fc00cc8cbafaea85944ed514356563d
Deleted: sha256:fcf51e472fa4537485d1350628efed4e6ac9acbecfe0701d344377bd4a3abaaa
Deleted: sha256:353fd11344b9dc1c437385596102b626fdcf34473286bd15151751a7b8d655c2
Deleted: sha256:1fe3a38b32fe61c6875b4355468319296356fcc352f6b793bba7ce78af9b0738
Deleted: sha256:4693057ce2364720d39e57e85a5b8e0bd9ac3573716237736d6470ec5b7b7230

```

Network Trials

Untuk menguji konektivitas antar perangkat anda dapat membaca kembali metode pengujian yang telah Anda lakukan pada **A01 dan A02** seperti mengakses SSH, mengakses Web Server, melakukan ping, dan sebagainya. Pada bagian ini anda akan melihat contoh *output* dari hasil pengujian dari metode-metode yang telah disebutkan sebelumnya.

SSH

- SSH ke webserver-ui (melalui *local device*)

```

PS C:\Users\mocha\ssh> ssh -i latihan mohammad_naufal91@34.168.178.27
The authenticity of host '34.168.178.27 (34.168.178.27)' can't be established.
ED25519 key fingerprint is SHA256:/j2I0rhNhFON9VqPf+Y7GfeOv38MnGGtuT1tsrN6W8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.168.178.27' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1018-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Fri Nov 10 01:03:40 UTC 2023

System load:  0.00390625   Processes:            117
Usage of /:   28.5% of 9.51GB Users logged in:       1
Memory usage: 37%          IPv4 address for docker0: 172.17.0.1
Swap usage:   0%           IPv4 address for ens4:   10.5.0.2

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

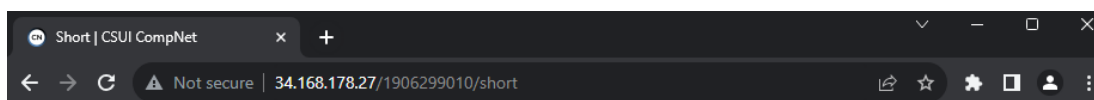
Last login: Fri Nov 10 00:53:55 2023 from 35.235.240.146
mohammad_naufal91@webserver-ui:~$

```

Deployment Web Server

Silakan akses web server ui pada *link* <http://<alamat-ip>:<port>/<npm-anda>/short>

- Akses Web Server http-serverlab dari *local device* (melalui *web browser*)



Short - Hello, 1906299010

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Nulla facilisi nullam vehicula ipsum a arcu cursus vitae congue. Neque egestas congue quisque egestas diam in arcu cursus. Vivamus at augue eget arcu. Nibh praesent tristique magna sit amet. Diam vel quam elementum pulvinar etiam non quam. Mauris sit amet massa vitae. Volutpat diam ut venenatis tellus in metus vulputate. Tellus at urna condimentum mattis pellentesque id nibh. Netus et malesuada fames ac turpis egestas integer eget. Viverra suspendisse potenti nullam ac tortor vitae purus faucibus. Augue eget arcu dictum varius duis at consectetur lorem. Sit amet facilisis magna etiam. Ullamcorper a lacus vestibulum sed. Enim facilisis gravida neque convallis.

- Akses Web Server http-serverlab dari pc-itf (melalui curl cli dengan alamat ip internal)

```
mochammad_naufal91@pc-itf:~/.ssh$ curl http://10.5.0.2/1906299010/short
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Short | CSUI CompNet</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
  </head>

  <body>
    <h1>Short - Hello, 1906299010</h1>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
      tempor incididunt ut labore et dolore magna aliqua. Nulla facilisi nullam
      vehicula ipsum a arcu cursus vitae congue. Neque egestas congue quisque
      egestas diam in arcu cursus. Vivamus at augue eget arcu. Nibh praesent
      tristique magna sit amet. Diam vel quam elementum pulvinar etiam non quam.
      Mauris sit amet massa vitae. Volutpat diam ut venenatis tellus in metus
      vulputate. Tellus at urna condimentum mattis pellentesque id nibh. Netus
      et malesuada fames ac turpis egestas integer eget. Viverra suspendisse
      potenti nullam ac tortor vitae purus faucibus. Augue eget arcu dictum
      varius duis at consectetur lorem. Sit amet facilisis magna etiam.
      Ullamcorper a lacus vestibulum sed. Enim facilisis gravida neque
      convallis.
    </p>
  </body>
</html>
```




Fase 3 (C03) : Implementasi Parsial Rancangan Solusi

Pada fase ini mahasiswa diharapkan dapat mengimplementasikan rancangan topologi yang dihasilkan pada fase 2 dengan menggunakan teknologi Virtual Private Cloud dan Google Computer Engine pada Google Cloud Platform. Implementasi ini cukup merepresentasikan infrastruktur organisasi yang sudah banyak mengadopsi teknologi *cloud*. Pada tahapan ini, mahasiswa diharapkan dapat memanfaatkan layanan atau aplikasi TI yang juga digunakan oleh klien, seperti aplikasi web, basis data, dan lain-lain, guna mendemonstrasikan kecocokan rancangan solusi dengan kebutuhan klien yang teridentifikasi.

Kriteria Dasar Implementasi

Dalam penerapan implementasi parsial rancangan solusi, terdapat beberapa kriteria yang perlu dipenuhi sebagai berikut :

- Mengimplementasikan semua sub-jaringan yang ada dengan perwakilan satu perangkat pada tiap sub-jaringan. Perangkat dalam hal ini diwakili oleh satu mesin Google Compute Engine, terlepas dari jenisnya pada topologi.
- Mekanisme *routing* tidak perlu diimplementasikan pada Google Cloud Platform.
- Mengimplementasikan semua solusi pembatasan akses (*firewall*, ACL, dan lain-lain) yang diidentifikasi pada C01 dan C02 melalui solusi jaringan Virtual Private Cloud.
- Jika terdapat solusi pembatasan akses yang membatasi akses SSH, implementasikan solusinya dan laporkan hasil pengujian pada laporan terlebih dahulu. Kemudian, solusi tersebut perlu dicabut agar mahasiswa dapat melanjutkan proses pengujian jaringan dengan kasus uji lainnya.
- Mengimplementasikan aplikasi yang teridentifikasi di C01 dan C02. Berikut merupakan beberapa aplikasi yang mungkin dapat diimplementasikan sesuai dengan kebutuhan klien masing-masing:
 - **Web Server:** Nginx, Django, Spring Boot, dll.
 - **Database:** PostgreSQL, MySQL, MongoDB, Redis, dll.
 - **ERP:** Odoo
 - **E-Learning:** Moodle
 - **File Server:** Nextcloud
- Layanan klien yang bisa diakses secara publik harus bisa di akses dari komputer lokal masing-masing peserta.
- Satu jaringan privat harus diimplementasikan dalam satu proyek GCP bersama.
- Pengujian harus dilakukan sesuai dengan aplikasi yang digunakan oleh klien dan tidak dianjurkan menggunakan protokol ICMP (Ping).
- Disarankan untuk menggunakan mesin E2-Micro untuk perangkat end-device dan E2-Medium untuk perangkat server agar dapat mengurangi penggunaan kredit GCP.

- Direkomendasikan hanya menggunakan layanan Google Compute Engine dan VPC, tidak perlu menggunakan layanan lainnya.

Ketentuan Pembuatan Dokumen

Dokumen yang perlu dibuat dan dikumpulkan:

- Laporan sistematis yang mencakup hasil implementasi yang telah disusun dalam bentuk dokumen bertipe PDF dan dikumpulkan pada slot pengumpulan C03.

Kode Jenis Dokumen: Laporan

- Laporan sistematis yang mencakup hasil implementasi yang telah disusun dalam bentuk *salindia/slides* bertipe PDF dan dikumpulkan pada slot pengumpulan C03.

Kode Jenis Dokumen: Presentasi

- *Feedback* untuk presentasi kelompok lain dalam bentuk kiriman/*post* pada utas/*thread* kelompok yang sesuai di forum “Presentasi dan Feedback C03” maksimal satu pekan setelah *deadline* C03.
- Tautan video presentasi (durasi maksimal 15 menit) dalam bentuk kiriman/*post* pada utas/*thread* di forum “Presentasi dan Feedback C03”. Video dapat diunggah di YouTube, OneDrive, atau layanan sejenis dengan syarat video tersebut dapat diakses setidaknya oleh akun UI.

Aspek yang dinilai:

- **(20%)** Ketepatan pemilihan teknologi

Penilaian aspek ini meliputi pemilihan teknologi dengan benar dan sesuai dengan kebutuhan organisasi serta kriteria dasar implementasi yang telah ditetapkan pada Studi Kasus Fase 3.

- **(20%)** Kesesuaian antara rancangan topologi dan implementasi

Penilaian aspek ini meliputi pengimplementasian teknologi dengan benar dan sesuai dengan topologi perencanaan solusi perbaikan jaringan serta sesuai dengan kebutuhan organisasi.

- **(20%)** Hasil uji konektivitas jaringan

Penilaian aspek ini meliputi penyusunan skenario pengujian jaringan yang tepat dan melakukan pengujian jaringan dengan benar sesuai kriteria dasar yang ditetapkan pada Studi Kasus Fase 3 dan batasan yang diidentifikasi dari organisasi yang bersangkutan.

- **(20%)** Analisis hasil uji konektivitas jaringan

Penilaian aspek ini meliputi menyusun analisis berdasarkan hasil uji konektivitas jaringan dengan lengkap serta sistematis.

- **(20%)** Umpan balik atas video presentasi kelompok lain

Penilaian aspek ini meliputi pemberian umpan balik mengenai solusi perbaikan jaringan secara positif dan konstruktif. Umpan balik diberikan dalam bentuk poin sesuai dengan kriteria dasar implementasi solusi perbaikan jaringan serta hasil uji konektivitas simulasi jaringan yang dijabarkan pada utas / thread yang relevan.

Peraturan

Keterlambatan

Anda diharapkan dapat mengumpulkan hasil pekerjaan yang dilakukan sebelum batas waktu pengumpulan. Jika terdapat kondisi di mana Anda terpaksa terlambat mengumpulkan hasil pekerjaan, terdapat jangka waktu tambahan di mana Anda masih diperbolehkan mengumpulkan hasil pekerjaan dengan konsekuensi tertentu. Jika X adalah durasi setelah batas waktu pengumpulan yang ditetapkan sampai waktu Anda mengumpulkan hasil pekerjaan, Anda akan menerima penalti nilai pekerjaan sebagaimana diatur pada peraturan berikut ini:

- $X < 10$ menit : Tidak ada penalti
- $10 \text{ menit} \leq X < 2 \text{ jam}$: 25% penalti
- $2 \text{ jam} \leq X < 4 \text{ jam}$: 50% penalti
- $4 \text{ jam} \leq X < 6 \text{ jam}$: 75% penalti
- $X \geq 6 \text{ jam}$: Cut-off (Pekerjaan anda tidak akan diterima)

Plagiarisme

Anda diperbolehkan berdiskusi tentang pekerjaan Anda dengan peserta kuliah lain atau pihak lainnya, namun Anda harus memastikan bahwa **semua pekerjaan yang dikumpulkan adalah murni hasil pekerjaan Anda sendiri**. Anda dilarang keras melakukan tindak plagiarisme atau kecurangan akademik lainnya. Menurut kamus daring Merriam-Webster, plagiarisme berarti:

- Mencuri dan mengklaim (ide atau kata orang lain) sebagai milik sendiri
- Menggunakan hasil (karya/pekerjaan orang lain) sebagai milik sendiri
- Melakukan pencurian literatur/sastra
- Merepresentasikan ulang sebuah ide/produk yang sudah ada sebagai sesuatu yang bersifat baru dan orisinil.

Tim pengajar memiliki hak untuk meminta klarifikasi terkait dugaan ketidakjujuran akademik, terutama plagiarisme, dan memberikan konsekuensi berupa **pengurangan nilai hasil pekerjaan atau pencabutan nilai (nilai diubah menjadi nol) untuk hasil pekerjaan yang terkonfirmasi dikerjakan secara tidak jujur**.