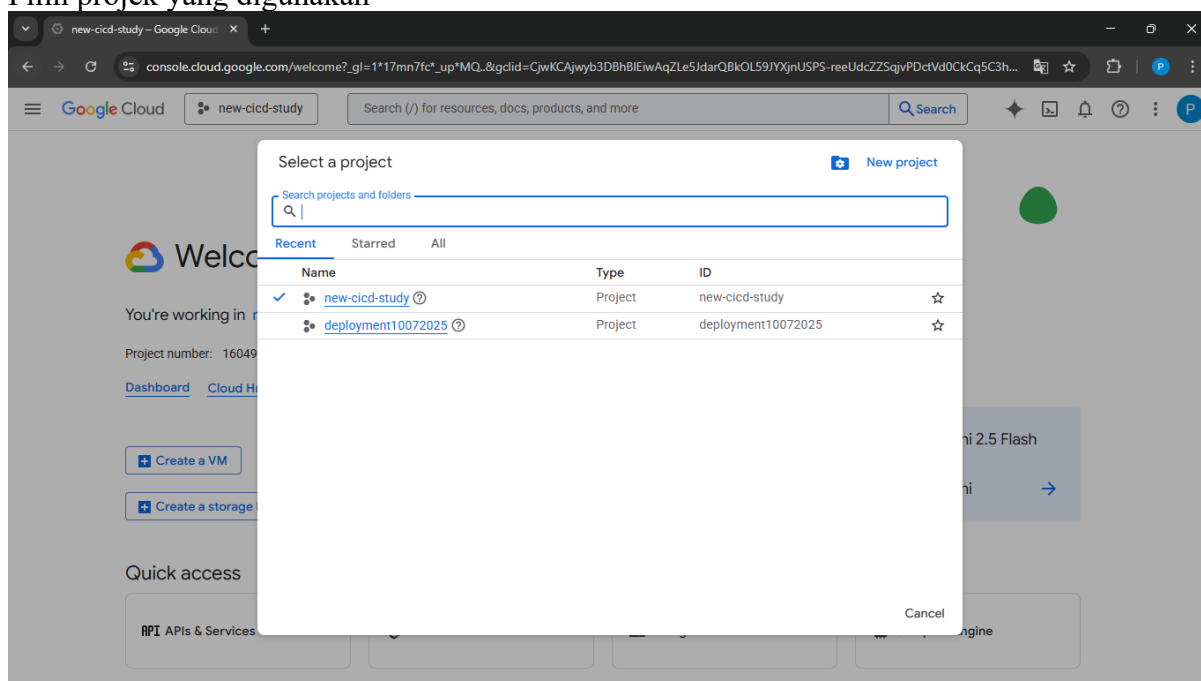


Daftar ISI

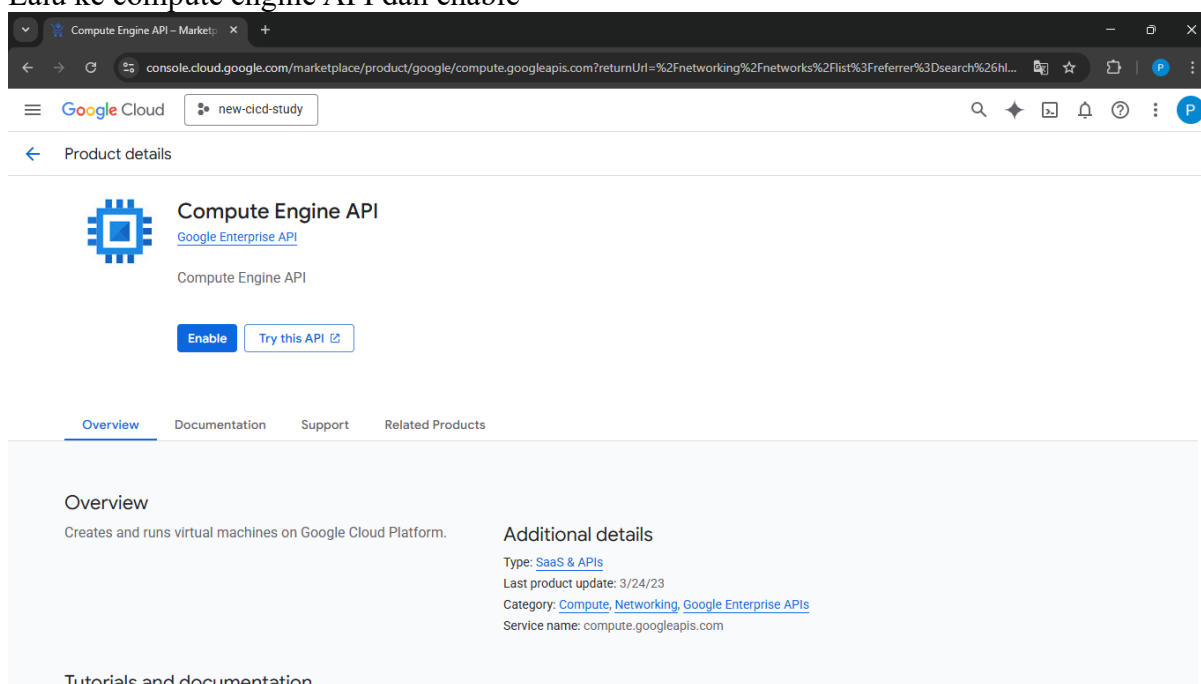
Daftar ISI	1
Membuat dan Menjalankan Compute Engine Instance	2
Mengonfigurasi Kebutuhan pada Compute Engine Instance.....	4
Memasang Node.js dan Menjalankan Web Server di Compute Engine Instance	5
Process Manager	10
Statik IP.....	12
Konfigurasi Nginx	13

Membuat dan Menjalankan Compute Engine Instance

Pilih proyek yang digunakan



Lalu ke compute engine API dan enable



Lalu Silakan masuk ke **Navigation menu > VPC network > Firewall**.

Properti	Nilai
name	app-server-firewall
description	Allow Custom TCP Port 9000
targets	Specified target tags

Source filter	Ipv4 ranges
Source IPv4 ranges	0.0.0.0
Protocol and ports	<i>Specified protocols and ports > centang tcp > isi 9000</i>

Jika sudah, lanjut klik tombol **Create**. Sampai di sini, Anda sudah berhasil membuat firewall rule. Sekarang mari kita lanjut ke tahap pembuatan instance.

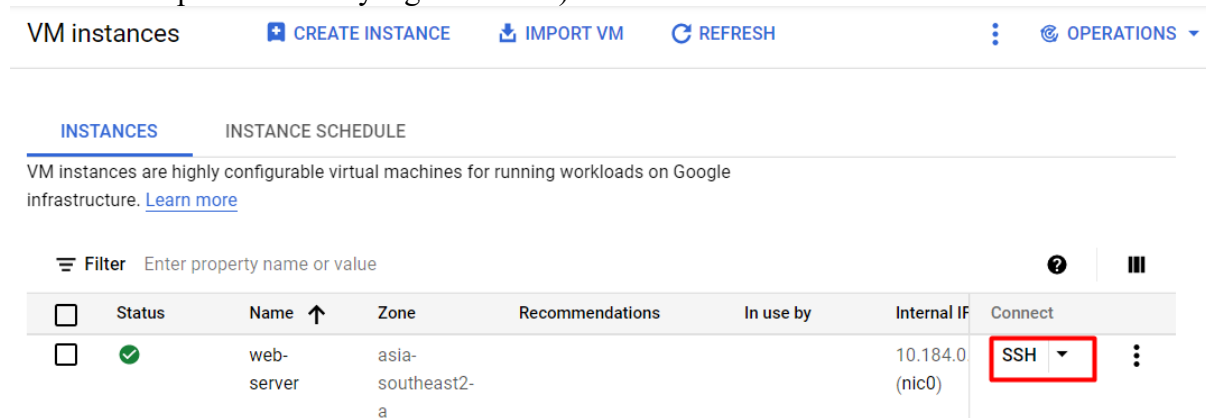
1. Untuk membuat Compute Engine instance, silakan akses **Navigation menu > Compute Engine**.
2. Setelah masuk di halaman Compute Engine, klik tombol **Create Instance** untuk membuat instance.
3. Di halaman pembuatan instance, sesuaikan konfigurasinya dengan tabel berikut

Properti	Nilai
Name	web-server
Region	asia-southeast2 (Jakarta)
Zone	asia-southeast2-a
Machine type	e2-micro (2 vCPU, 1 GB memory)
Boot disk	Type: New balanced persistent disk Size: 10 GB Image: Ubuntu 20.04 LTS

4. Kemudian, buka menu **Networking, Disks, Security, Management, Sole-Tenancy**.
5. Di bagian *Networking*, pada kolom *Network tags* isikan **web-server**. Jika Anda ingat, ketika membuat firewall rule, kita menentukan *target tags*-nya adalah *web-server*. Itu artinya, firewall rule tersebut akan diterapkan ke VM instance ini.
6. Jika semua sudah sesuai, lanjut klik tombol **Create** yang berada di paling bawah halaman. Nantinya Anda akan dibawa kembali ke halaman *VM instances*, tunggu hingga instance Anda memiliki centang hijau pada kolom *status*-nya yang berarti sudah berjalan dengan baik.

Mengonfigurasi Kebutuhan pada Compute Engine Instance

Untuk memastikan hal tersebut, silakan akses kembali instance menggunakan SSH (klik tombol **SSH** pada instance yang Anda buat).



VM instances

[CREATE INSTANCE](#) [IMPORT VM](#) [REFRESH](#) [OPERATIONS](#)

INSTANCES INSTANCE SCHEDULE

VM instances are highly configurable virtual machines for running workloads on Google infrastructure. [Learn more](#)

Filter Enter property name or value

<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	Connect
<input type="checkbox"/>	✓	web-server	asia-southeast2-a			10.184.0 (nic0)	SSH

Kemudian, tuliskan perintah git --version.

Pastikan versi git muncul ya. Jika gagal, berarti git belum terpasang. Anda dapat dengan mudah memasang sistem git pada Ubuntu dengan perintah:

```
1. sudo apt-get install git
```

Selanjutnya, unduh proyek web server kita pada instance. Proses unduh remote repository ke local repository dinamakan dengan **cloning**.

Silakan kunjungi halaman remote repository pada browser, kemudian klik tombol **Code**.

```
git clone https://github.com/periartaa/notes-app-back-end.git
```

Untuk memastikan proyek tersimpan pada instance, Anda bisa tuliskan perintah ls, maka akan terdapat folder dengan nama notes-app-back-end.

Masuk ke folder proyek tersebut dengan menggunakan perintah berikut:

```
cd notes-app-back-end
```

Anda bisa tuliskan kembali perintah ls untuk melihat isi di dalam folder proyek.

Memasang Node.js dan Menjalankan Web Server di Compute Engine Instance

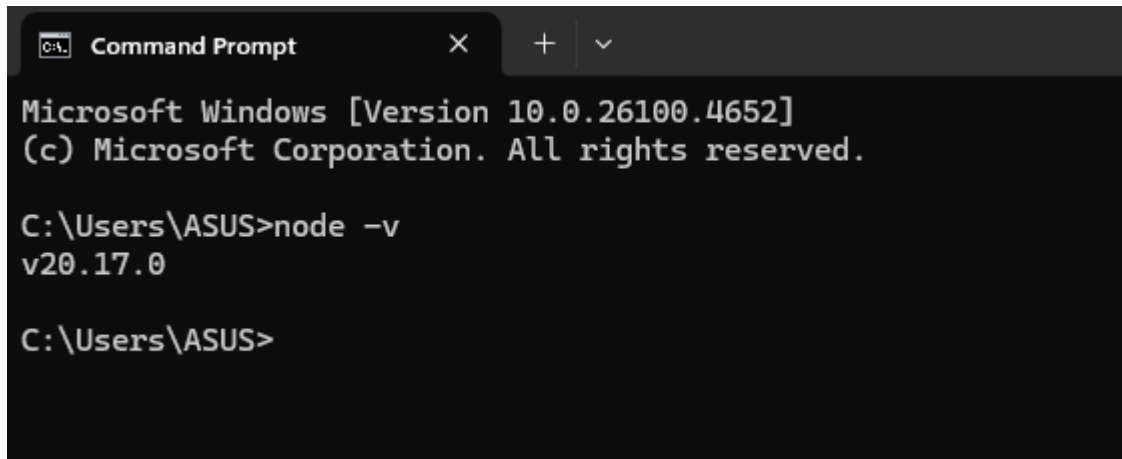
Untuk memasang nvm pada Ubuntu, silakan eksekusi perintah berikut:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.37.2/install.sh | bash
```

Penting: Agar nvm dapat digunakan, silakan keluar dulu dari SSH dengan perintah exit. Kemudian, akses kembali instance Anda.

Selanjutnya, pasang Node.js versi yang sesuai dengan komputer Anda dengan perintah:

```
nvm install <versi nodejs>
```

A screenshot of a Windows Command Prompt window. The title bar says 'Command Prompt'. The text inside shows the Windows version '10.0.26100.4652' and the copyright notice '(c) Microsoft Corporation. All rights reserved.' The user has entered the command 'node -v' and the output is 'v20.17.0'. The prompt is currently at 'C:\Users\ASUS>'.

```
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>node -v
v20.17.0

C:\Users\ASUS>
```

Well done! Node.js berhasil terpasang. Kini Anda bisa menjalankan web server. Pastikan Anda berada di dalam folder **notes-app-back-end** ya. Jika belum, silakan masuk ke foldernya dengan perintah:

```
cd notes-app-back-end
```

Lalu, pasang seluruh module/package dependencies yang digunakan pada proyek kita dengan mengeksekusi perintah npm install.

Lanjut, jalankan proyek dengan perintah npm run start.

```
ssh.cloud.google.com/v2/ssh/projects/new-cicd-study/zones/asia-southeast2-a/instances/web-server?authuser=0&hl=en_US&projectNumber=160497865945&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/new-cicd-study/zones/asia-southeast2-a/instances/web-server?authuser=0&hl=en_US&projectNumber=160497865945&useAdminProxy=true

SSH-in-browser

periarta2004@web-server:~$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.37.2/install.sh | bash
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 13527 100 13527 0 0 32039 0 --:--:-- --:--:-- --:--:-- 32054
=> nvm is already installed in /home/periarta2004/.nvm, trying to update using git
=> Compressing and cleaning up git repository

=> nvm source string already in /home/periarta2004/.bashrc
=> bash completion source string already in /home/periarta2004/.bashrc
npm notice
npm notice New major version of npm available! 10.8.2 -> 11.4.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.4.2
npm notice To update run: npm install -g npm@11.4.2
npm notice
=> You currently have modules installed globally with 'npm'. These will no
=> longer be linked to the active version of Node when you install a new node
=> with 'nvm'; and they may (depending on how you construct your '$PATH')
=> override the binaries of modules installed with 'nvm':

/home/periarta2004/.nvm/versions/node/v20.17.0/lib
└── corepack@0.29.3
=> If you wish to uninstall them at a later point (or re-install them under your
=> 'nvm' Nodes), you can remove them from the system Node as follows:

$ nvm use system
$ npm uninstall -g a_module

=> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
periarta2004@web-server:~$ nvm install v20.17.0
v20.17.0 is already installed.
Now using node v20.17.0 (npm v10.8.2)
periarta2004@web-server:~$ node -v
v20.17.0
periarta2004@web-server:~$ ls
notes-app-back-end
periarta2004@web-server:~$ cd notes-app-back-end/
periarta2004@web-server:~/notes-app-back-end$ npm install
added 144 packages, and audited 145 packages in 3s
```

```
ssh.cloud.google.com/v2/ssh/projects/new-cicd-study/zones/asia-southeast2-a/instances/web-server?authuser=0&hl=en_US&projectNumber=160497865945&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/new-cicd-study/zones/asia-southeast2-a/instances/web-server?authuser=0&hl=en_US&projectNumber=160497865945&useAdminProxy=true

SSH-in-browser

=> override the binaries of modules installed with 'nvm':

/home/periarta2004/.nvm/versions/node/v20.17.0/lib
└── corepack@0.29.3
=> If you wish to uninstall them at a later point (or re-install them under your
=> 'nvm' Nodes), you can remove them from the system Node as follows:

$ nvm use system
$ npm uninstall -g a_module

=> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
periarta2004@web-server:~$ nvm install v20.17.0
v20.17.0 is already installed.
Now using node v20.17.0 (npm v10.8.2)
periarta2004@web-server:~$ node -v
v20.17.0
periarta2004@web-server:~$ ls
notes-app-back-end
periarta2004@web-server:~$ cd notes-app-back-end/
periarta2004@web-server:~/notes-app-back-end$ npm install
added 144 packages, and audited 145 packages in 3s

28 packages are looking for funding
run 'npm fund' for details

4 vulnerabilities (2 low, 1 moderate, 1 high)

To address all issues, run:
  npm audit fix

Run 'npm audit' for details.
periarta2004@web-server:~/notes-app-back-end$ npm run start
> submission@1.0.0 start
> node ./src/server.js

Server berjalan pada http://localhost:9000
```

Wah tampaknya server sudah berjalan nih. Coba kita cek melalui browser. Silakan kembali ke halaman *VM instances* di Cloud Console, lalu salin **External IP** instance Anda.

Enter property name or value

Name ↑	Zone	Recommendations	In use by	Internal IP	External IP
web-server	asia-southeast2-a			10.184.0.3 (nic0)	34.50.75.67 (nic0)

Buka tab browser baru, lalu kunjungi alamat `http://external-ip:9000`. Misalnya, <http://34.59.75.67:9000>. Apakah ada respons dari server?



This site can't be reached

The connection was reset.

Try:

- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

ERR_CONNECTION_RESET

Reload

Details

Memperbaiki Masalah

Kembali ke VSCode dan buka berkas **server.js**.

Ubah nilai properti host menjadi seperti ini:

```
src > JS server.js
1  const Hapi = require("@hapi/hapi");
2  const routes = require("./routes");
3
4  const init = async () => {
5    const server = Hapi.server({
6      port: 9000,
7      // host: "localhost",
8      host: process.env.NODE_ENV !== "production" ? "localhost" : "0.0.0.0",
9      routes: {
10       cors: {
11         origin: ["*"],
12       },
13     },
14   });
15
16   server.route(routes);
17 }
```

Dengan begitu, properti host akan bernilai sesuai dengan environment yang ditetapkan. Selanjutnya, kita tetapkan environment pada npm runner script. Buka berkas **package.json**, lalu atur npm runner script menjadi seperti ini:

```
"scripts": {
  "start-prod": "NODE_ENV=production node ./src/server.js",
  "start-dev": "nodemon ./src/server.js",
  "lint": "eslint ./src"
},
```

Kita tidak perlu menetapkan NODE_ENV pada start-dev karena nodemon secara default menggunakan nilai *development* pada NODE_ENV. Karena itu juga, pada proses production kita tidak menggunakan nodemon lagi, cukup gunakan node saja.

Simpan seluruh perubahan, kemudian commit seluruh perubahan yang ada dengan perintah:

```
git add .
git commit -m "fix bugs host value"
```

Kemudian, push perubahannya ke remote repository dengan perintah:

```
git push origin master
```

Selanjutnya, akses kembali Compute Engine instance melalui **SSH** dan masuk ke folder **notes-app-back-end**.

Update proyek web server dengan perubahan yang sudah dilakukan sebelumnya dengan menggunakan perintah:

```
git pull origin master
```

Coba jalankan kembali server-nya. Namun, kali ini menggunakan proses *production*. Eksekusi perintah `npm run start-prod`.

The screenshot shows a terminal window titled "SSH-in-browser" with a URL bar containing an SSH connection to a Google Cloud instance. The terminal output includes the Linux version (6.1.0-37-cloud-amd64), Debian version (6.1.140-1), and system architecture (x86_64). It displays the Debian GNU/Linux license notice and the last login time (Thu Jul 10 12:21:16 2025 from 35.235.248.145). The user 'periarta2004' navigates to the directory '~/.notes-app-back-end' and runs 'npm run start-prod'. The output shows the command 'submission@1.0.0 start-prod' and 'NODE_ENV=production node ./src/server.js'. Finally, it states 'Server berjalan pada http://0.0.0.0:9000'.

```
Linux web-server 6.1.0-37-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.140-1 (2025-05-22) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jul 10 12:21:16 2025 from 35.235.248.145
periarta2004@web-server:~$ cd notes-app-back-end/
periarta2004@web-server:~/notes-app-back-end$ npm run start-prod

> submission@1.0.0 start-prod
> NODE_ENV=production node ./src/server.js

Server berjalan pada http://0.0.0.0:9000
```

Silakan akses server melalui browser dengan menggunakan **External IP** yang diikuti dengan **port 9000**.

The screenshot shows a web browser window with the address bar displaying '34.101.179.25:9000'. The page content shows a JSON error response: {'statusCode':404,'error':'Not Found','message':'Not Found'}. The browser's developer tools are open, showing the error details.

```
{ "statusCode": 404, "error": "Not Found", "message": "Not Found" }
```

Process Manager

Memasang Node.js Process Manager pada Compute Engine instance

[pm2](#) merupakan salah satu Node.js Process Manager yang populer digunakan. Kita akan menggunakan pm2 ini untuk memantau web server yang ada di Compute Engine instance.

Silakan akses kembali instance Anda melalui **SSH**. Kemudian, pasang pm2 dengan perintah `npm install -g pm2`.

Setelah proses instalasi selesai, masuk ke direktori **notes-app-back-end** dengan perintah `cd notes-app-back-end`. Jalankan node process menggunakan `pm2` dengan perintah:

```
pm2 start npm --name "notes-api" -- run "start-prod"
```

[illegible]

```
SSH-in-browser
```

PM2 is a Production Process Manager for Node.js applications
with a built-in Load Balancer.

Start and Daemonize any application:
\$ pm2 start app.js

Load Balance 4 instances of api.js:
\$ pm2 start api.js -i 4

Monitor in production:
\$ pm2 monitor

Make pm2 auto-boot at server restart:
\$ pm2 startup

To go further checkout:
<http://pm2.io/>

[PM2] Spawning PM2 daemon with pm2_home=/home/periartha2004/.pm2

[PM2] PM2 Successfully daemonized

[PM2] Starting /home/periartha2004/.nvm/versions/node/v20.17.0/bin/npm in fork_mode (1 instance)

[PM2] Done.

id	name	mode	Q	status	cpu	memory
0	notes-api	Fork	0	online	0%	30.0mb

periartha2004@web-server:~\$ pm2 restart notes-api

Use --update-env to update environment variables

[PM2] Applying action restartProcessId on app [notes-api] (ids: [0])

[PM2] [notes-api](0) ✓

id	name	mode	Q	status	cpu	memory
0	notes-api	Fork	15	online	0%	18.9mb

periartha2004@web-server:~\$ pm2 stop notes-api

[PM2] Applying action stopProcessId on app [notes-api] (ids: [0])

[PM2] [notes-api](0) ✓

pm2 berhasil menjalankan web server dan ia akan memantau prosesnya. Bila proses itu terhenti entah karena terjadi crash atau apa pun, ia akan secara otomatis menjalankan ulang proses. Dengan begitu, Anda tidak perlu khawatir server akan mengalami *downtime* lagi.

Di pm2, kita dapat me-restart proses secara manual dengan cara:

```
pm2 restart notes-api
```

Kita bisa juga menghentikan prosesnya dengan cara:

```
pm2 stop notes-api
```

Untuk menjalankan kembali proses, gunakan perintah:

```
pm2 start notes-api
```

Statik IP

Anda perlu memesan alamat IP terlebih dahulu. Ini akan menjadikan IP tersebut "milik" proyek Anda dan dapat dialokasikan ke instance VM.

Melalui Google Cloud Console:

1. Buka [Google Cloud Console](#).
2. Navigasi ke **VPC network > IP addresses**.
3. Di halaman "External IP addresses" atau "Internal IP addresses", klik **RESERVE EXTERNAL STATIC ADDRESS** atau **RESERVE INTERNAL STATIC ADDRESS**.
4. Isi detail berikut:
 - **Name:** Nama untuk alamat IP statis Anda (misalnya, my-static-ip).
 - **Region:** Pilih region yang sama dengan instance VM Anda (untuk IP eksternal regional atau IP internal). Untuk IP eksternal global, pilih "Global".
 - **Network Service Tier:** Pilih Premium atau Standard (biasanya Premium untuk performa terbaik).
 - **IP version:** IPv4 atau IPv6.
 - **Attach to:** (Opsional) Anda bisa langsung melampirkannya ke instance VM yang sudah ada di sini. Jika tidak, lewati langkah ini dan lakukan di langkah selanjutnya.
5. Klik **RESERVE**.

VPC Network / IP addresses / Reserve a static address

Reserve external static IP address

Name: my-static-ip ⓘ
Lowercase letters, numbers, hyphens allowed

Description: ip external : 34.101.179.25 ⓘ

Network Service Tier ⓘ
☒ Premium ⓘ
Current project-level tier, [change](#) ⓘ
☐ Standard ⓘ

IP version
☒ IPv4
☐ IPv6

Type
☒ Regional
☐ Global
To be used with Global forwarding rules. [Learn more](#) ⓘ

Region: asia-southeast2 (Jakarta) ⓘ

Attached to: web-server ⓘ

Konfigurasi Nginx

1. Koneksi ke server melalui SSH

Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
web-server	asia-southeast2-a			10.184.0.3 (nic0)	34.50.75.67 ↗ (nic0)	SSH ▾ ⋮

2. Instalasi Nginx (Jika Belum Terinstal):

```
sudo apt update
sudo apt install nginx -y
```

3. Buat File Konfigurasi Nginx Baru untuk Aplikasi Anda:

```
sudo nano /etc/nginx/sites-available/notes-app
```

Masukkan konten berikut ke dalam file notes-app:

```
server {
    listen 80; # Nginx akan mendengarkan di port 80 (HTTP standar)
    server_name 34.50.75.67; # Ganti dengan IP publik server Anda
    (34.50.75.67) atau domain Anda jika ada

    location / {
        proxy_pass http://localhost:9000; # Meneruskan permintaan ke aplikasi
        yang berjalan di port 9000
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    # Opsional: Jika Anda ingin mengkonfigurasi HTTPS di kemudian hari,
    Anda bisa tambahkan blok 'listen 443 ssl' di sini
}
```

4. Aktifkan Konfigurasi (Buat Symlink):

```
sudo ln -s /etc/nginx/sites-available/notes-app /etc/nginx/sites-enabled/
```

5. Hapus Konfigurasi Default Nginx (Opsional tapi Disarankan):

```
sudo rm /etc/nginx/sites-enabled/default
```

6. Uji Sintaks Konfigurasi Nginx:

```
sudo nginx -t
```

7. **Restart Nginx:**

```
sudo systemctl restart nginx
```

8. **Pastikan Nginx Berjalan (Opsional):**

```
sudo systemctl status nginx
```

```
ssh.cloud.google.com/v2/ssh/projects/new-cicd-study/zones/asia-southeast2-a/instances/web-server?authuser=0&hl=en_US&projectNumber=160497865945&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/new-cicd-study/zones/asia-southeast2-a/instances/web-server?authuser=0&hl=en_US&projectNumber=160497865945&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
perla2004@web-server:~$ sudo apt update
sudo apt install nginx -y
Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [30 B]
Get:2 file:/etc/apt/mirrors/debian-security.list Mirrorlist [39 B]
Hit:7 https://packages.cloud.google.com/apt google-compute-engine-bookworm-stable InRelease
Hit:3 https://deb.debian.org/debian bookworm InRelease
Hit:4 https://deb.debian.org/debian bookworm-updates InRelease
Hit:5 https://deb.debian.org/debian bookworm-backports InRelease
Hit:6 https://deb.debian.org/debian-security bookworm-security InRelease
Hit:8 https://packages.cloud.google.com/apt cloud-sdk-bookworm InRelease
Hit:9 https://packages.cloud.google.com/apt google-cloud-ops-agent-bookworm-2 InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.22.1-9+deb12u2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
perla2004@web-server:~$ sudo nano /etc/nginx/sites-available/notes-app
perla2004@web-server:~$ sudo ln -s /etc/nginx/sites-available/notes-app /etc/nginx/sites-enabled/
perla2004@web-server:~$ sudo rm /etc/nginx/sites-enabled/default
perla2004@web-server:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
perla2004@web-server:~$ sudo systemctl restart nginx
perla2004@web-server:~$ sudo systemctl status nginx
* nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-07-11 09:58:54 UTC; 5s ago
     Docs: man:nginx(8)
   Process: 6542 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 6543 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 6544 (nginx)
     Tasks: 3 (limit: 4681)
    Memory: 2.2M
       CPU: 11ms
   CGroup: /system.slice/nginx.service
           └─6544 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─6546 "nginx: worker process"
```

9. Cara mengetahui

Buka web browser Anda dan masukkan alamat IP publik server Anda:

`http://34.50.75.67`

Jika konfigurasi berhasil, Nginx akan menerima permintaan di port 80 dan meneruskannya ke aplikasi Anda yang berjalan di port 9000. Anda seharusnya melihat respons dari aplikasi notes-app-back-end Anda.

```
34.50.75.67:9000
Not secure 34.50.75.67:9000
Pretty-print
{"statusCode":404,"error":"Not Found","message":"Not Found"}
```