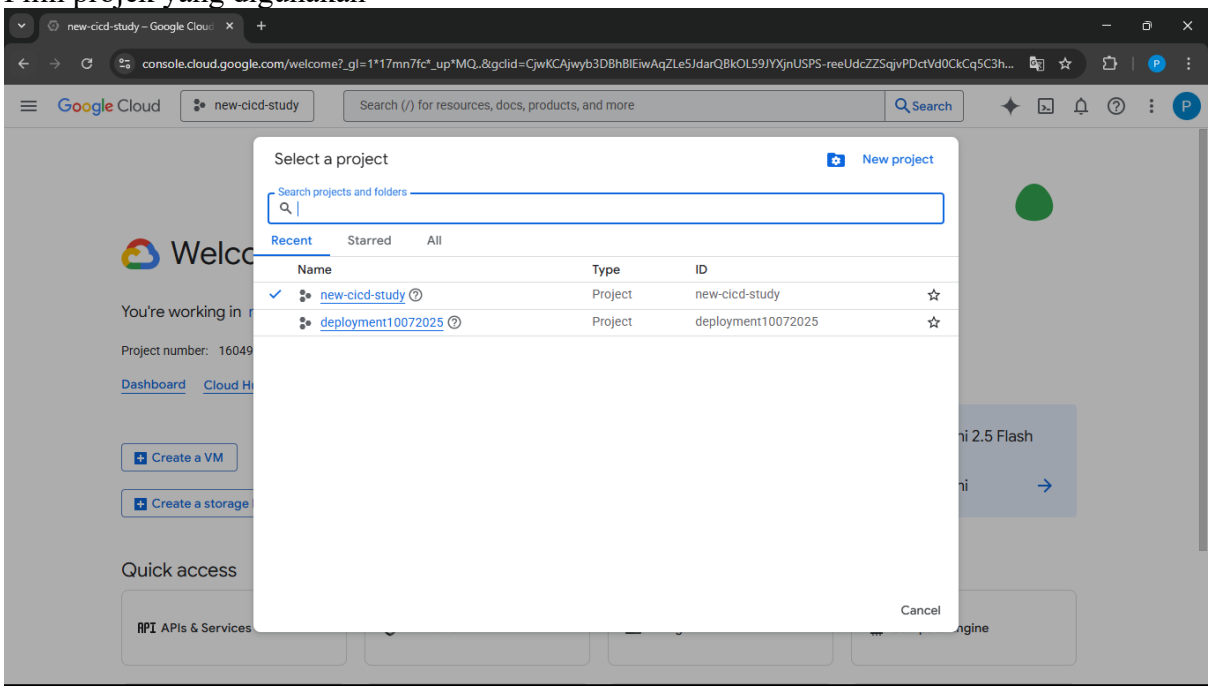
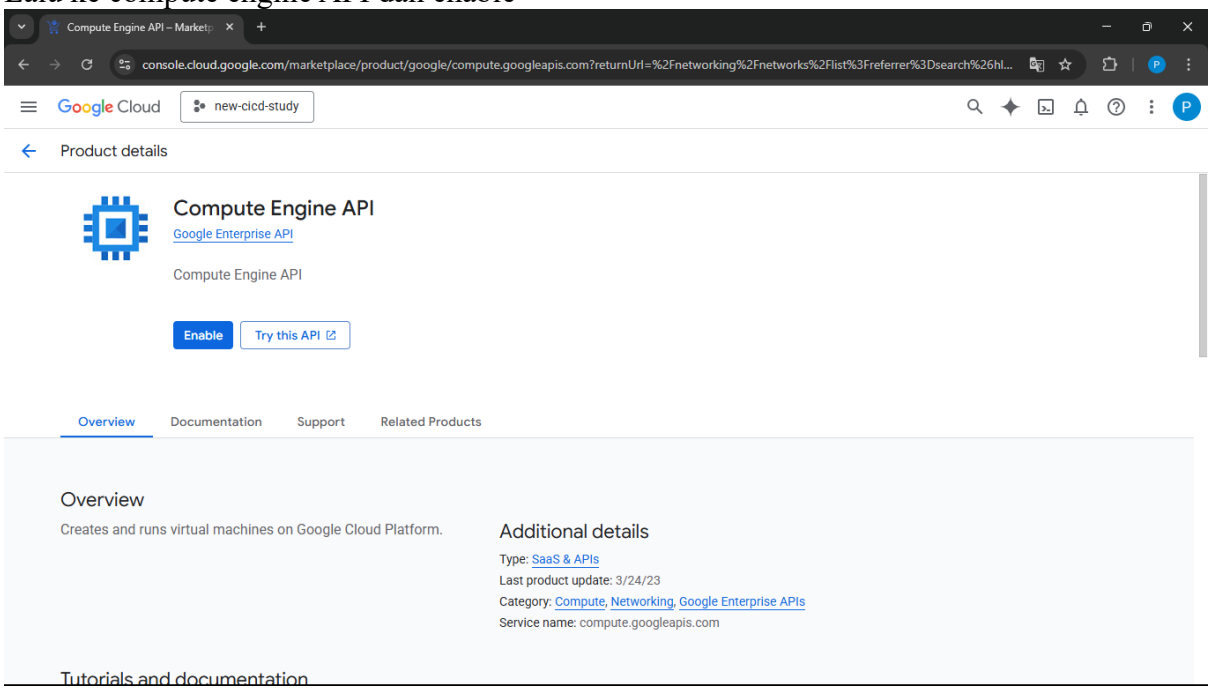
Daftar ISI

Membuat dan Menjalankan Compute Engine Instance

Pilih projek 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	*Specified protocols and ports* > centang *tcp* > isi 9000

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

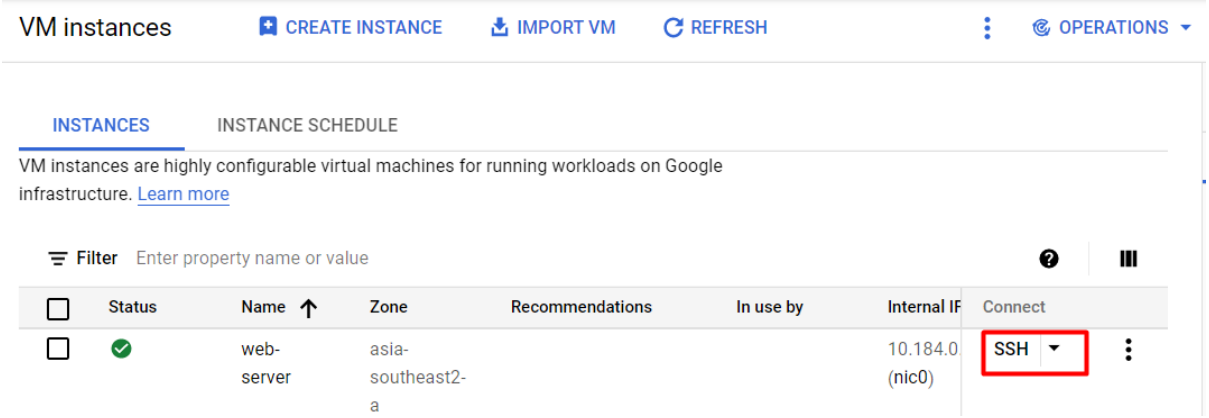
1. Untuk membuat Compute Engine instance, silakan akses **Navigation menu > Compute Engine**.
2. Setelah masuk di halaman Compute Engine, klik tombol **Create Instance** untuk membuat instance.
3. Di halaman pembuatan instance, sesuaikan konfigurasinya dengan tabel berikut

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).



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

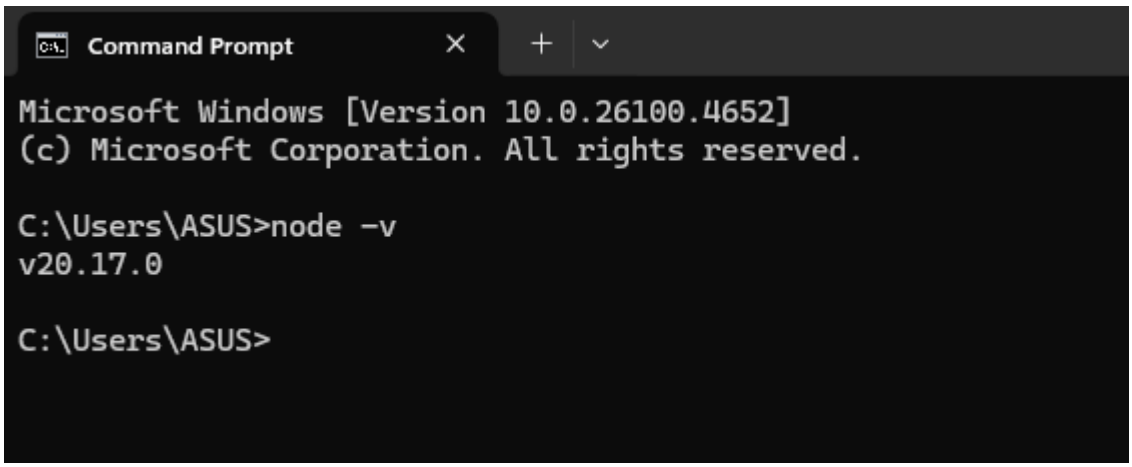
Untuk memasang nvm pada Ubuntu, silakan eksekusi perintah berikut:

```
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:

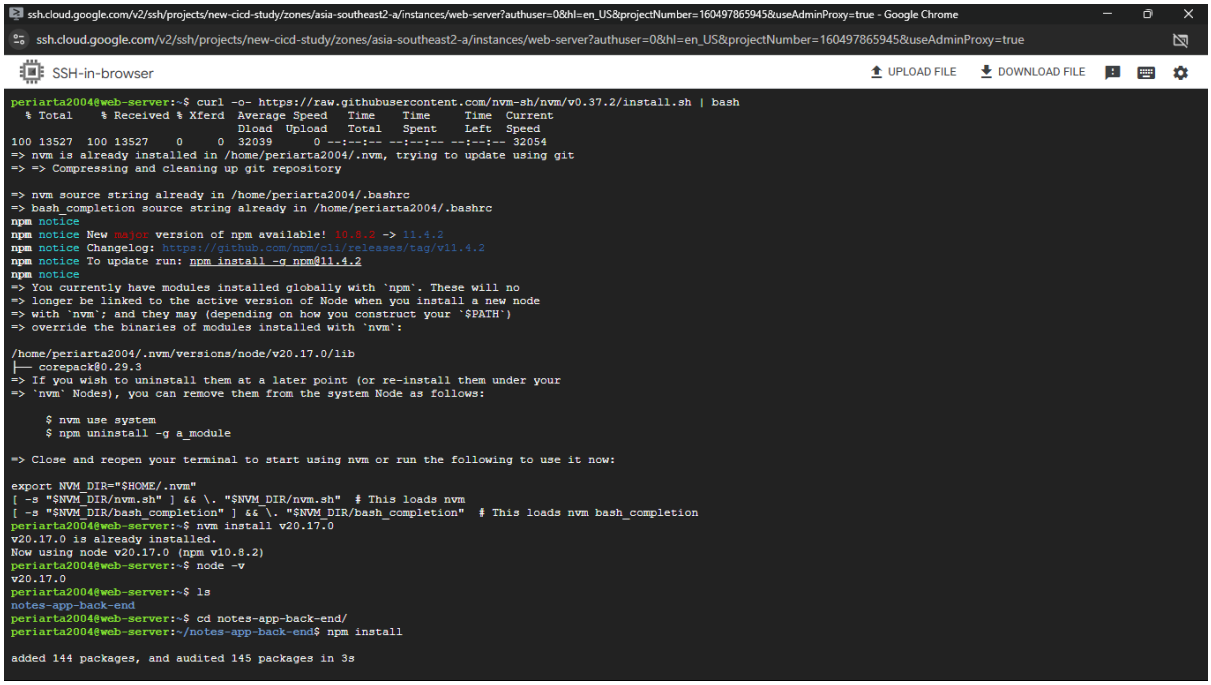
```
nvm install <versi nodejs>
```

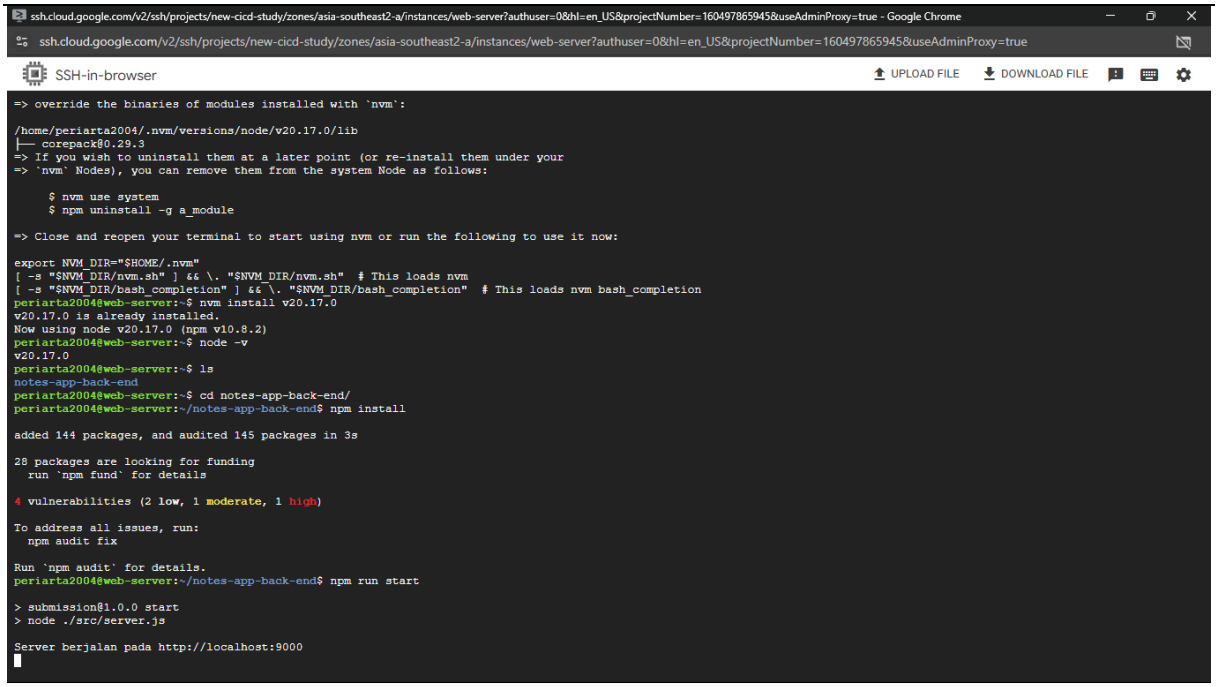


Well done! Node.js berhasil terpasang. Kini Anda bisa menjalankan web server. Pastikan Anda berada di dalam folder **notes-app-back-end** ya. Jika belum, silakan masuk ke foldernya dengan perintah:

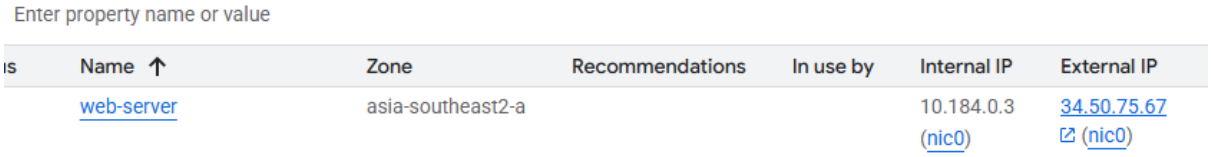
```
cd notes-app-back-end
```

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

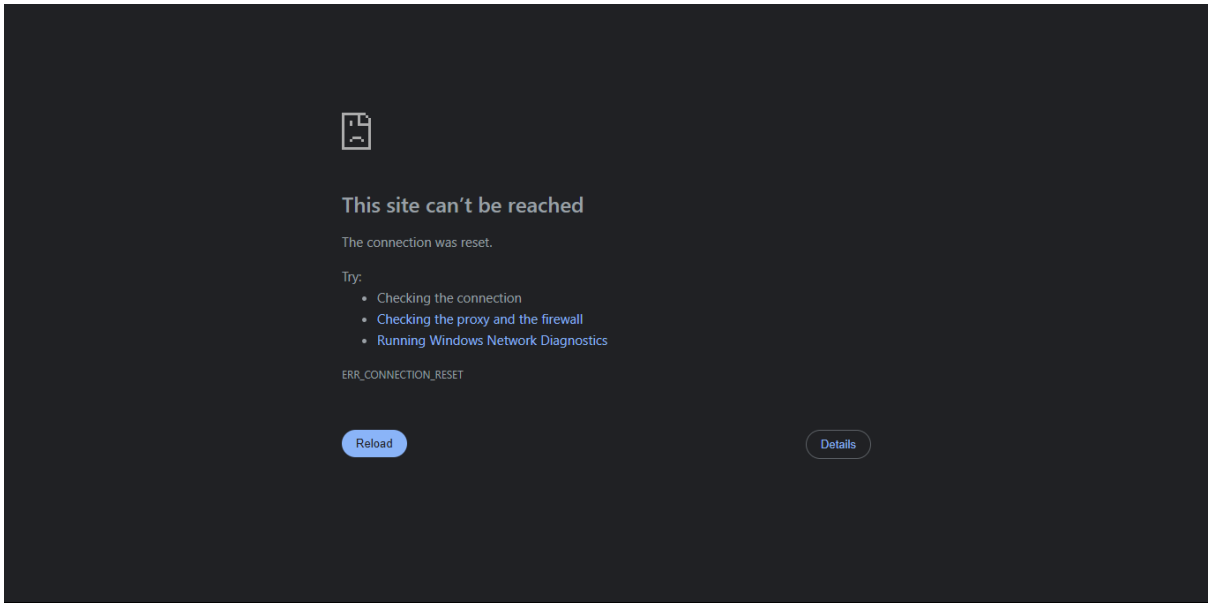




Wah tampaknya server sudah berjalan nih. Coba kita cek melalui browser. Silakan kembali ke halaman *VM instances* di Cloud Console, lalu salin **External IP** instance Anda.



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);
```

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"
```

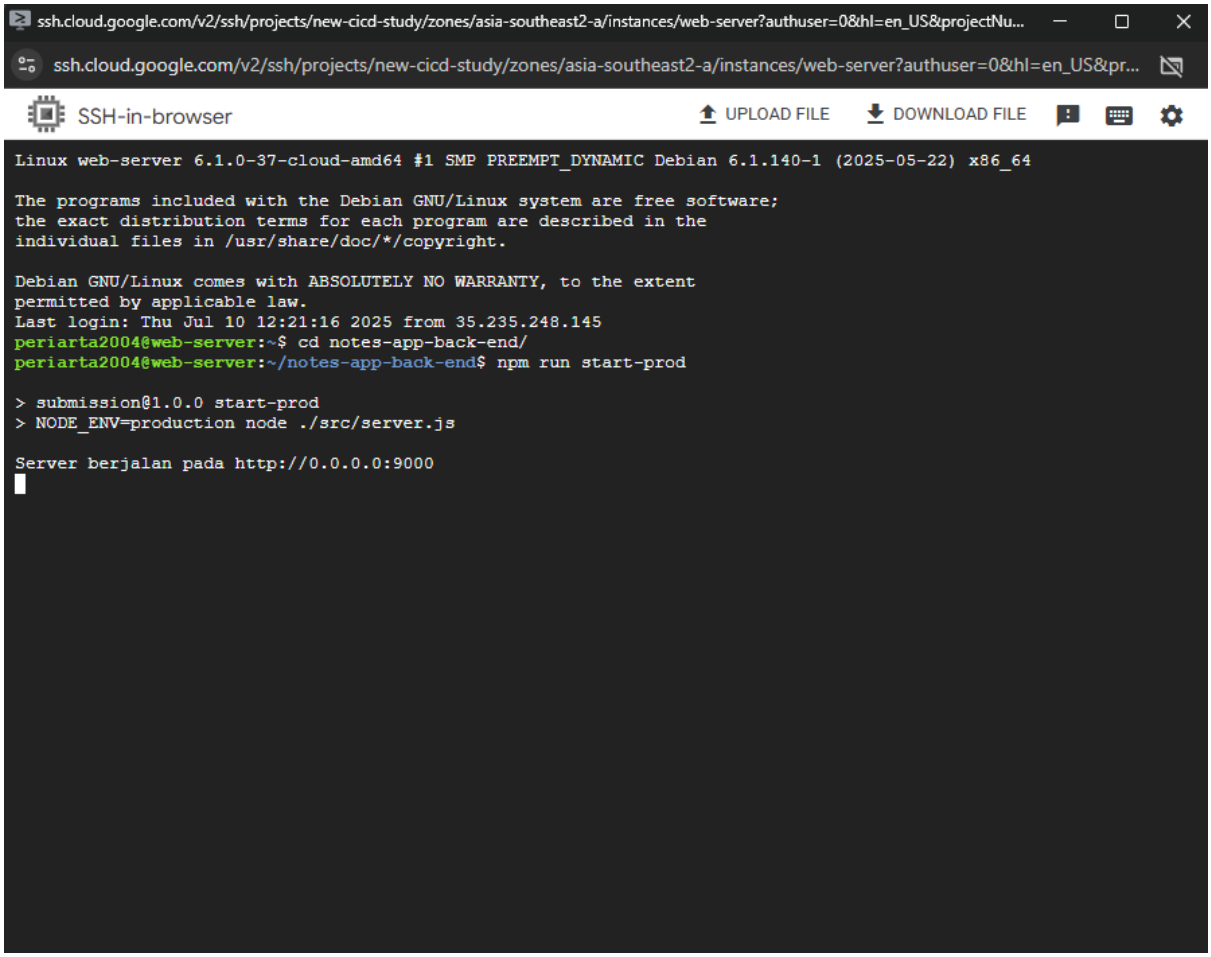
Kemudian, push perubahannya ke remote repository dengan perintah:

```
git push origin master
```

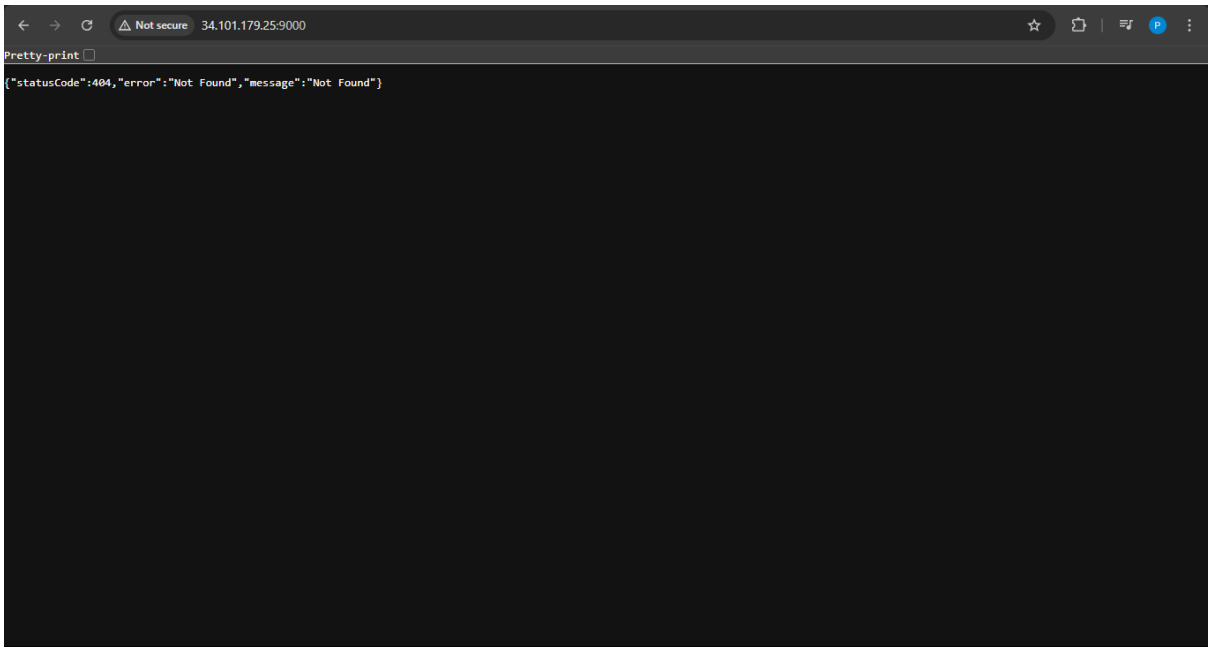
Selanjutnya, akses kembali Compute Engine instance melalui **SSH** dan masuk ke folder **notes-app-back-end**.
Update proyek web server dengan perubahan yang sudah dilakukan sebelumnya dengan menggunakan perintah:

```
git pull origin master
```

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



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



Process Manager

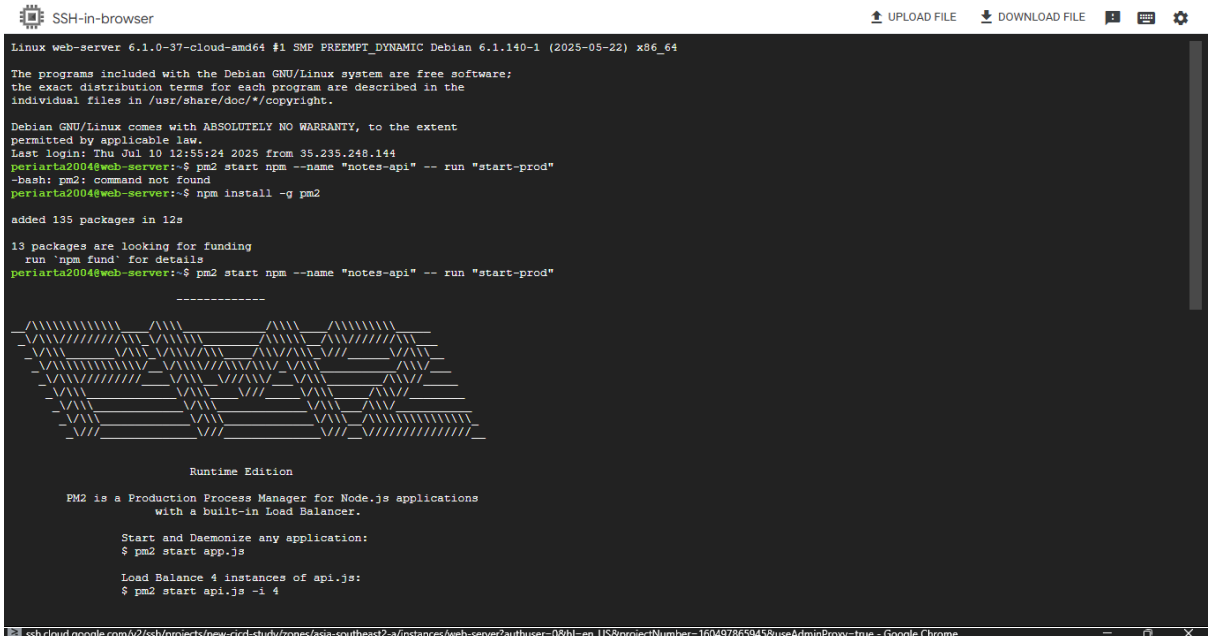
Memasang Node.js Process Manager pada Compute Engine instance

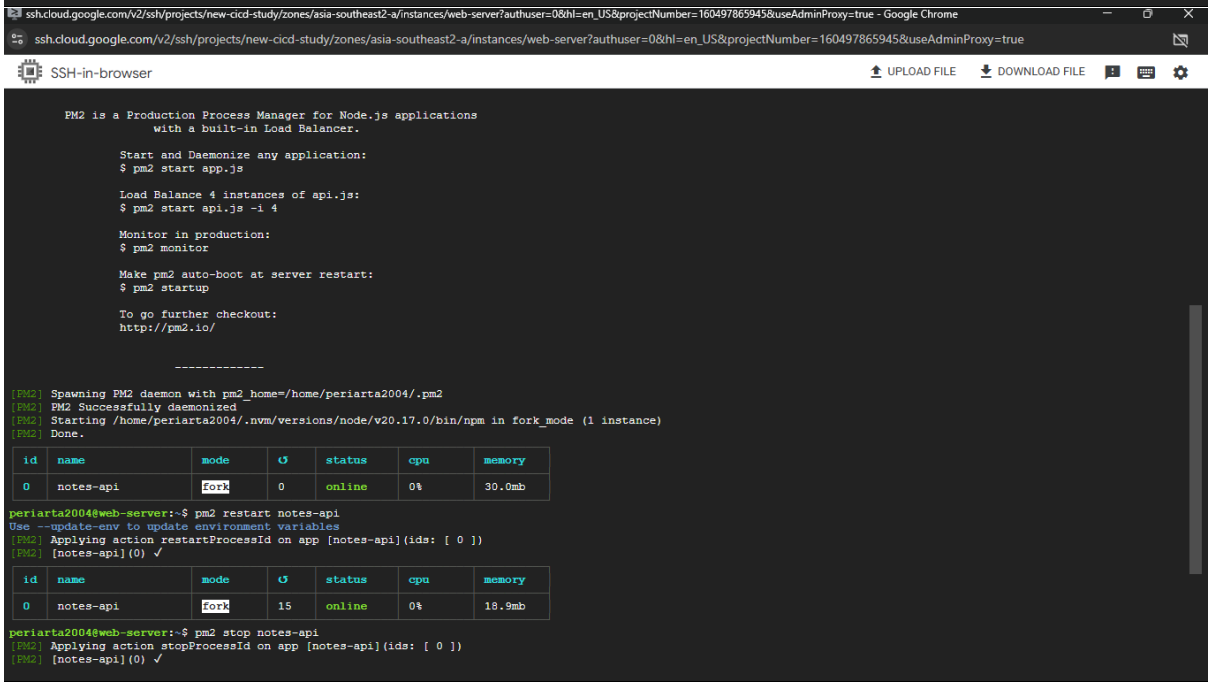
pm2 merupakan salah satu Node.js Process Manager yang populer digunakan. Kita akan menggunakan pm2 ini untuk memantau web server yang ada di Compute Engine instance.

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"
```





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:

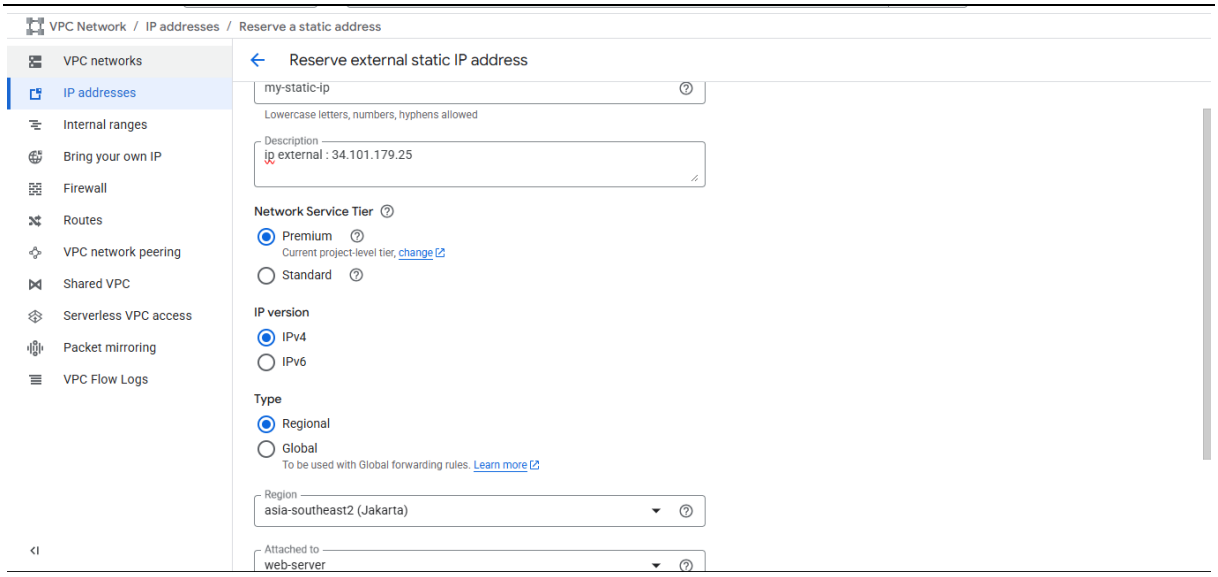
```
pm2 start notes-api
```

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**.



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:

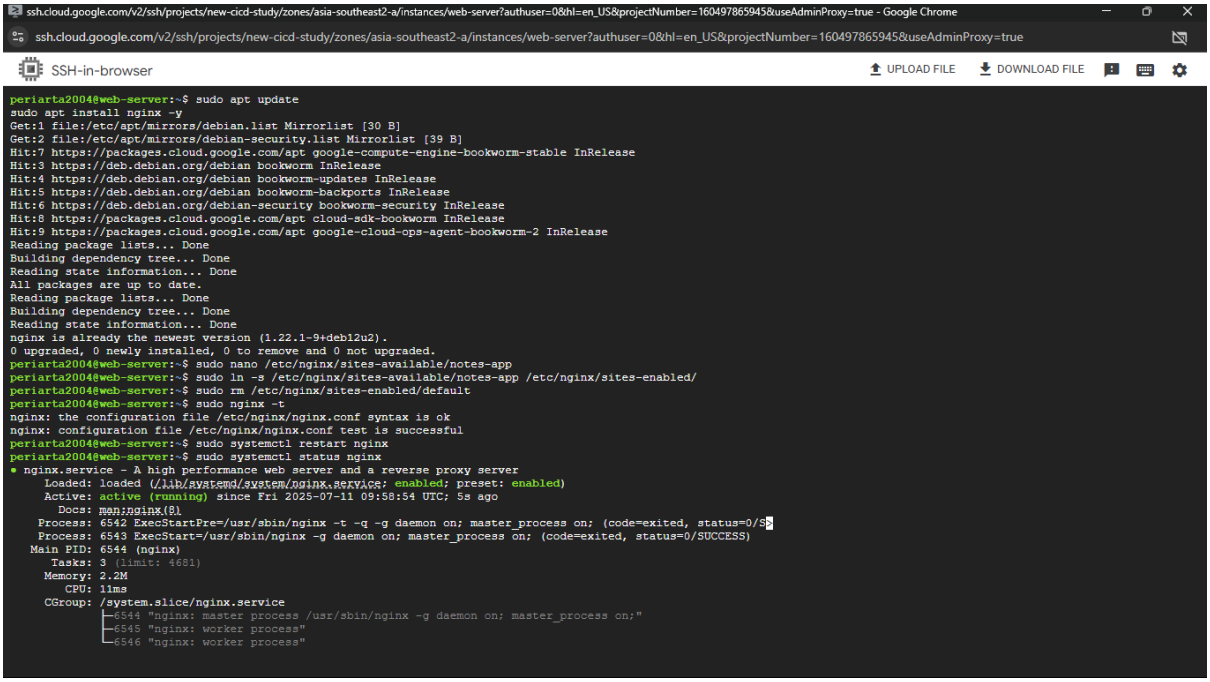
```
sudo nginx -t
```

7. **Restart Nginx:**

```
sudo systemctl restart nginx
```

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

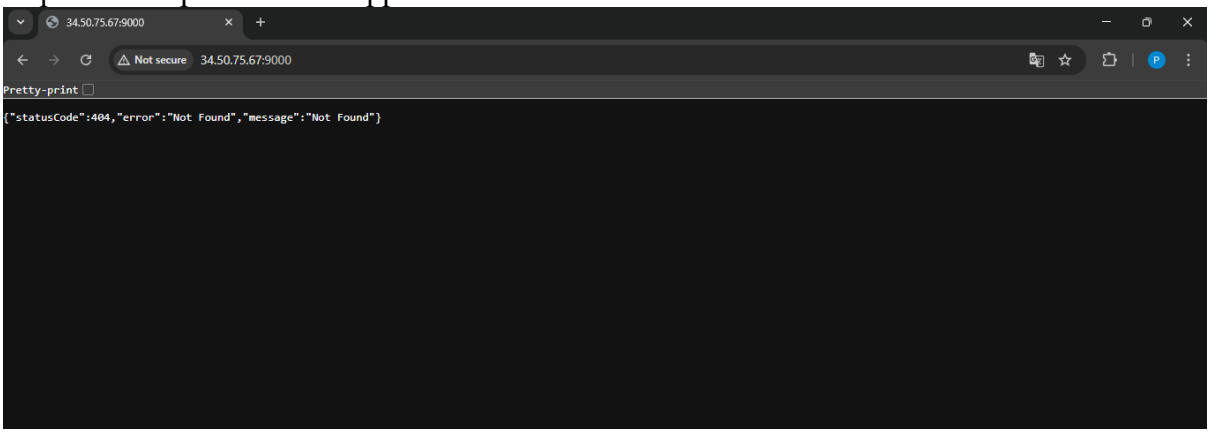
```
sudo systemctl status nginx
```



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.



Pengujian

Masuk ke SSH

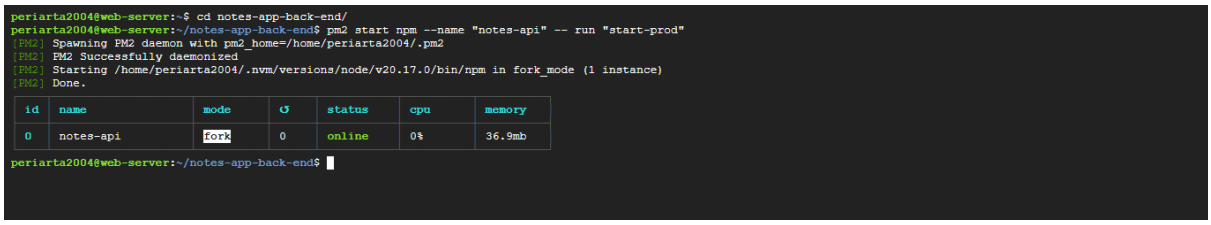
Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
web-server	asia-southeast2-a			10.184.0.3 (nic0)	34.50.75.67 (nic0)	SSH

Buka folder direktory

```
periarta2004@web-server:~$ cd notes-app-back-end/
```

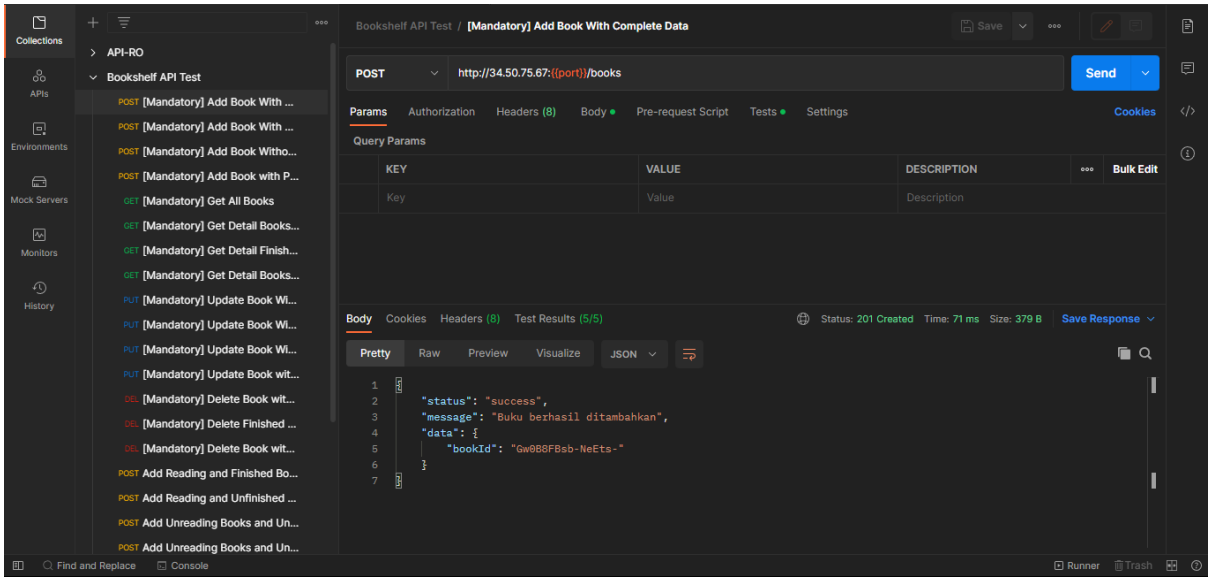
Lalu jalankan untuk menghidukan server

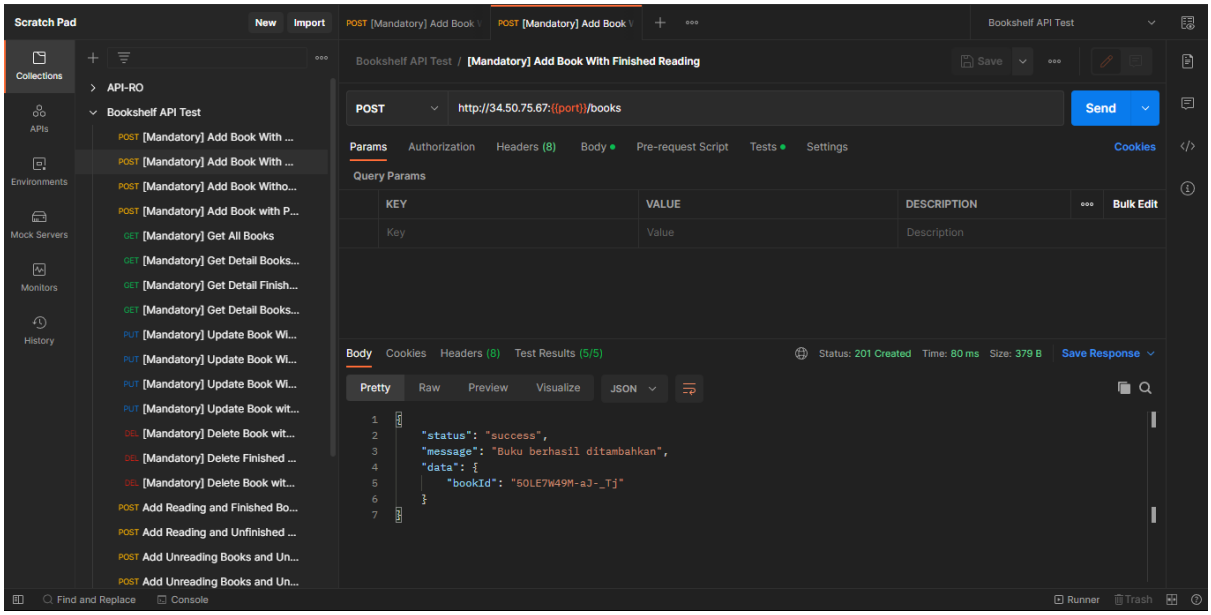
```
periarta2004@web-server:~/notes-app-back-end$ pm2 start npm --name "notes-api" -- run "start-prod"
```



Masuk ke postman
Sesuai ipddress
Contoh

```
http://34.50.75.67:{{port}}/books
```




Scratch Pad
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
API-RO
Bookshelf API Test
POST [Mandatory] Add Book With ...
POST [Mandatory] Add Book With ...
POST [Mandatory] Add Book Witho...
POST [Mandatory] Add Book with P...
GET [Mandatory] Get All Books
GET [Mandatory] Get Detail Books...
GET [Mandatory] Get Detail Finish...
GET [Mandatory] Get Detail Books...
PUT [Mandatory] Update Book Wi...
PUT [Mandatory] Update Book Wi...
PUT [Mandatory] Update Book Wi...
PUT [Mandatory] Update Book wit...
DEL [Mandatory] Delete Book wit...
DEL [Mandatory] Delete Finished ...
DEL [Mandatory] Delete Book wit...
POST Add Reading and Finished Bo...
POST Add Reading and Unfinished ...
POST Add Unreading Books and Un...
POST Add Unreading Books and Un...
POST [Mandatory] Add Book
POST [Mandatory] Add Book
Bookshelf API Test
Bookshelf API Test / [Mandatory] Add Book With Finished Reading
Save
POST
http://34.50.75.67:{{port}}/books
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
Key
Value
Description
Body
Cookies
Headers (8)
Test Results (5/5)
Status: 201 Created
Time: 80 ms
Size: 379 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"status": "success",
"message": "Buku berhasil ditambahkan",
"data": {
"bookId": "5OLE7W49M-aJ-_Tj"
}
}
Find and Replace
Console
Runner
Trash