**Deployment Documentation**

Deploying NodeJS apps to Linux web server using pm2, nginx, cloud flare

**Buat projek backend API**

1. Buat project backend API, missal pakai nodejs
2. Build projek dan pastikan ada executable server.js
3. Push ke github

**Buat VM di cloud provider**

1. Create compute engine instance at GCP
2. SSH to the VM

ssh root@128.199.242.170 🡪 ini external ip address dari VM

1. Update linux apps:

apt update 🡪 tidak perlu sudo karena sudah di root user

1. Install dependency:

apt install nodejs npm -y

**Run applikasi nodejs dengan pm2 di VM**

1. Clone github project:

git clone urlProjekGithub

cd namaProjek

1. Install dependencsi projek:

npm install

1. PM2 🡪 daemon tool utk run npm projek di production, make sure apps running 24/7, will restart the apps when the server crash or restart:

npm install pm2 -g

1. Jalankan pm2

pm2 startup 🡪 agar pm2 jalan saat server crash atau kerestart

pm2 start server.js –name –namaLabel 🡪 menjalanan entry point (server.js)

1. Buka apps yang sudah running lewat ip address public dari VM

128.199.242.170:30001 🡪 external ip address VM dan port yg diassign di dalam server.js

1. Sampai sini protocol yang digunakan masih bersifat http belum https. Bedanya pada https ada SSL, lebih aman

**Enable firewall**

Agar tidak semua orang bisa akses langsung server kita

1. Enable firewall di VM:

ufw enable

1. Allow port ssh agar developer bisa masuk:

ufw allow 22

**Pointing domain name ke server dan pembuatan SSL/TLS dengan Cloud Flare**

Agar nama web kita bisa diubah semau kita dan protocol aman.

1. Buka DNS management, misal cloud flare.
2. Buka DNS tab.

Daftarkan domain web kita.

Buat juga subdomain jika mau.

Contoh: youtube.2pai-dev.com 🡪 youtube = subdomain, 2pai-dev = domain

1. Buka SSL/TLS tab 🡪 origin server 🡪 create certificate
2. Simpan ‘origin certificate’ (public key) dan ‘private key’.
3. Cloud Flare juga melindungi server kita misalnya dari DDOS (Distributed Denial of Service), missal jika tiba2 ada 10000 request dalam 1 detik, maka Cloud Flare akan otomatis membloknya.

**Install dan setting webserver**

Bisa menggunakan nginx 🡪 sebagai proxy request yg masuk lewat https ke apps kita.

Nginx juga bisa sebagai Load Balancer jika kita ingin menambah VM server

1. Install nginx di VM server:

apt install nginx

1. Allow protocol yg digunakan nginx 🡪 port 80 (http) dan port 443 (https)

ufw allow ‘Nginx Full’

1. Buat config dari proxy apps kita di direktori VM: vi /etc/nginx/conf.d/nginx.conf

server {

listen 80;

listen [::]:80;

server\_name api.2pai-dev.com; # the hostname

return 302 https://$server\_name$request\_uri; ## all traffic through port 80 will be forwarded to 443

}

server {

listen 443 ssl http2;

listen [::]:443 ssl http2;

ssl on;

ssl\_certificate /etc/ssl/certs/cert.pem; #path to your public key

ssl\_certificate\_key /etc/ssl/private/cert.key; #path to your private key

server\_name api.2pai-dev.com; # the hostname

location / {

proxy\_pass http://127.0.0.1:3001; # URL Rest API

}

}

Penjelasan:

traffic masuk dari port 80 ke server name youtube.2pai-dev.com, akan diforward ke port 433 (https).

Port 433 mengenable SSl dengan certificate yg dienable di Clouf Flare (public key dan private key)

Lalu semua request yg masuk ke route path akan diproxy ke url apps kita di localhost:3001

1. Delete onfigurasi nginx yg lama:

Misal dengan: mv /etc/nginx/sites-enabled/default default.bak

1. Install certificate dari Cloud Flare ke dalam server kita.

Copy origin certificate/public key, paste ke /etc/ssl/certs/cert.pem

Copy private key, paste ke /etc/ssl/private/cert.key

1. Test apakah ada error atau tidak:

nginx -t

1. Reload service nginx nya:

service nginx reload

1. Buka url, sekarang sudah https:

youtube.2pai-dev.com