// Tutorial //

How To Set Up a Node.js Application for Production on Ubuntu 22.04

 # Introduction

Node.js adalah lingkungan runtime JavaScript sumber terbuka untuk membangun aplikasi sisi server dan jaringan. Platform berjalan di Linux, macOS, FreeBSD, dan Windows. Meskipun Anda dapat menjalankan aplikasi Node.js di baris perintah, tutorial ini akan berfokus untuk menjalankannya sebagai layanan. Ini berarti mereka akan memulai ulang saat reboot atau kegagalan dan aman untuk digunakan di lingkungan produksi.

 # Prerequisites

- Koneksi internet
- Server linux ubuntu sudah siap digunakan
- Nginx sudah terinstall

 # Step 1 — Installing Node.js with Apt from the Default Repositories

Ubuntu 22.04 berisi versi Node.js di repository defaultnya yang dapat digunakan untuk memberikan pengalaman yang konsisten di berbagai sistem. Pada saat penulisan, versi di repositori adalah 12.22.9. Ini bukan versi terbaru, tetapi harus stabil dan cukup untuk eksperimen cepat dengan bahasa tersebut.

ketikkan perintah berikut sebagai langkah awal:

apt update

Lalu instal Node.js:

apt install nodejs

Periksa bahwa instalasi telah berhasil dengan bertanya kepada node terkait nomor versinya:

nodejs -v

output:

v12.22.9

Jika paket di dalam repositori sesuai dengan kebutuhan Anda, itu adalah semua yang Anda perlukan untuk menyisipkan Node.js. Pada sebagian besar kasus, Anda juga perlu menginstal npm, manajer paket Node.js. Anda dapat melakukan ini dengan menginstal paket npm dengan apt:

apt install npm

Ini akan memungkinkan Anda untuk menginstal modul dan paket untuk digunakan dengan Node.js. Pada titik ini, Anda telah berhasil menginstal Node.js dan npm dengan menggunakan apt dan repositori perangkat lunak Ubuntu asali.

 # Step 2 — Creating a Node.js Application

Mari kita menulis aplikasi First App with Node JS yang mengembalikan "First App with Node JS" ke permintaan HTTP apa pun. Contoh aplikasi ini akan membantu Anda bangun dan berjalan dengan Node.js. Anda dapat menggantinya dengan aplikasi Anda sendiri — pastikan Anda memodifikasi aplikasi Anda untuk mendengarkan alamat IP dan port yang sesuai.

Pertama mari kita buat direktori project node.js nya, dengan perintah:

mkdir /var/www/nodejs

jika sudah kita bisa masuk saja direktori yang kita buat tadi, dengan perintah:

cd /var/www/nodejs

Pertama, menggunakan nano atau editor teks favorit Anda, buatlah contoh aplikasi bernama firstApp.js:

nano firstApp.js

Masukkan kode berikut ke dalam file:

```
const http = require('http');

const hostname = ip_public_or_your_domain;
const port = 3000;

const server = http.createServer((req, res) => {
```

```
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end(First app with Node.js!\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Simpan file dan keluar dari editor. Jika Anda menggunakan nano, tekan Ctrl+X, lalu saat diminta, Y alu Enter.

Aplikasi Node.js ini mendengarkan alamat (localhost) dan port (3000) yang ditentukan, dan mengembalikan "First app with Node.js!" dengan 200 kode sukses HTTP. Karena kami sedang mendengarkan localhost, klien jarak jauh tidak akan dapat terhubung ke aplikasi kami.

Untuk menguji aplikasi Anda, ketik:

node firstApp.js

Anda akan menerima output berikut:

output:

Server running at http://ip_public_or_your_domain:3000/

Untuk menguji aplikasi, buka sesi terminal lain di server Anda, dan sambungkan localhost ke curl:

curl http://ip_public_or_your_domain:3000/

Jika Anda mendapatkan output berikut, aplikasi bekerja dengan baik dan mendengarkan alamat dan port yang benar:

output:

First app with node.js!

Jika Anda tidak mendapatkan hasil yang diharapkan, pastikan aplikasi Node.js Anda berjalan dan dikonfigurasi untuk mendengarkan pada alamat dan port yang tepat.

 # Step 3 — Installing PM2

Selanjutnya mari kita install PM2, process manager untuk aplikasi Node.js. PM2 memungkinkan untuk melakukan daemonisasi aplikasi sehingga mereka akan berjalan di latar belakang sebagai layanan.

Gunakan npm untuk menginstal PM2 versi terbaru di server Anda:

npm install pm2@latest -g

Opsi -g memberitahu npm untuk menginstal modul secara global , sehingga tersedia di seluruh sistem. Mari pertama-tama gunakan pm2 start perintah untuk menjalankan aplikasi Anda, firstApp.js, di latar belakang:

pm2 start firstApp.js

Ini juga menambahkan aplikasi Anda ke daftar proses PM2, yang dikeluarkan setiap kali Anda memulai aplikasi:

output:

```
[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /var/www/nodejs/firstApp.js in fork_mode (1 instance)
[PM2] Done.
```

id	name	namespace	version	mode	pid	uptime	↺	status	cpu	mem	user	watching
0	firstApp	default	N/A	fork	6707	0s	0	online	0%	11.3mb	root	disabled

```
root@server-js:/var/www/nodejs#
```

Selain yang telah kami bahas, PM2 menyediakan banyak sub perintah yang memungkinkan Anda mengelola atau mencari informasi tentang aplikasi Anda. Berikut ini adalah perintah-perintah utama pm2

- Hentikan aplikasi dengan perintah ini (sebutkan PM2 App name atau id):

  pm2 stop app_name_or_id
- Mulai ulang aplikasi:

  pm2 restart app_name_or_id
- Daftar aplikasi yang saat ini dikelola oleh PM2:

  pm2 list
- Dapatkan informasi tentang aplikasi tertentu menggunakan App name:

  pm2 info app_name
- Monitor proses PM2 dapat ditarik dengan sub perintah monit. Ini menampilkan status aplikasi, CPU, dan penggunaan memori:

  pm2 monit

 # Step 4 — Setting Up Nginx as a Reverse Proxy Server

Aplikasi Anda sedang berjalan dan mendengarkan pada localhost, tetapi Anda perlu menyiapkan cara bagi pengguna untuk mengaksesnya. Kami akan menyiapkan server web Nginx sebagai proxy terbalik untuk tujuan ini.

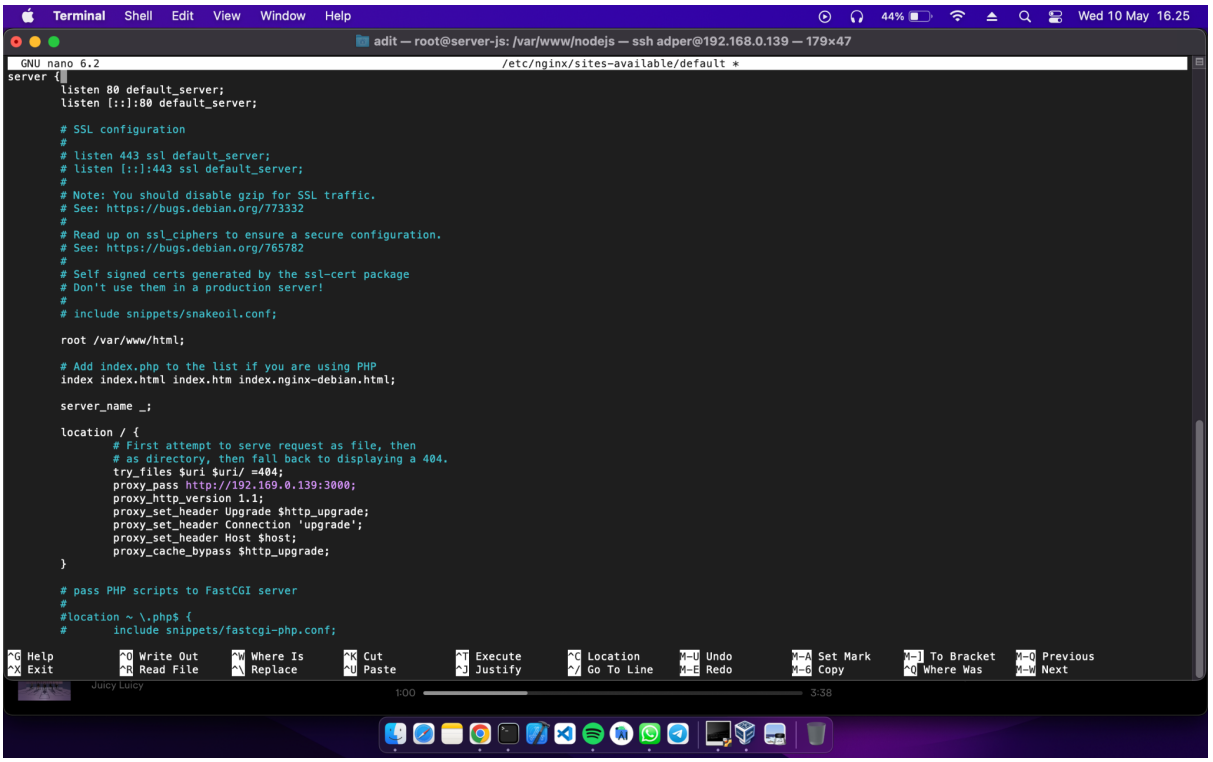
Dalam tutorial prasyarat, Anda mengatur konfigurasi Nginx Anda di file. Buka file ini untuk di edit: /etc/nginx/sites-available/default

nano /etc/nginx/sites-available/default

Di dalam server blok, Anda harus memiliki location /blok yang sudah ada. Ganti isi blok itu dengan konfigurasi berikut. Jika aplikasi Anda diatur untuk mendengarkan pada port yang berbeda, perbarui bagian yang disorot ke nomor port yang benar:

```
server {
...
        location / {
                proxy_pass http://ip_public_or_your_domain:3000;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection 'upgrade';
                proxy_set_header Host $host;
                proxy_cache_bypass $http_upgrade;
        }
...
}
```

contoh sebagai berikut:



Dengan asumsi bahwa aplikasi Node.js Anda sedang berjalan, dan konfigurasi aplikasi dan Nginx Anda sudah benar, Anda seharusnya sekarang dapat mengakses aplikasi Anda melalui

proxy balik Nginx. Cobalah dengan mengakses URL server Anda (alamat IP publik atau nama domainnya). Untuk pengujian kalian bisa membuka browser anda, kemudian bisa akses halaman **http://ip_public_or_your_domain:3000/**

 # Conclusion

Selamat! Anda sekarang memiliki aplikasi Node.js yang berjalan di belakang proxy terbalik Nginx di server Ubuntu 22.04. Penyiapan proxy balik ini cukup fleksibel untuk memberi pengguna Anda akses ke aplikasi lain atau konten web statis yang ingin Anda bagikan.