

▮ Panduan Instalasi & Konfigurasi Nginx, Golang, dan Pengujian API

▮ 1. Download & Install Nginx

Unduh Nginx dari tautan resmi berikut:

▮ [Download Nginx](#)

Setelah diunduh, ekstrak file ZIP dan jalankan Nginx:

```
cd C:\

unzip nginx-1.27.4.zip

cd nginx-1.27.4

start nginx
```

▮ 2. Atur Konfigurasi Nginx

Edit file konfigurasi Nginx di:

```
C:\nginx-<version>\conf\nginx.conf
```

Tambahkan konfigurasi berikut untuk **load balancing** ke beberapa instance Golang:

```
http {

    upstream golang_backend {

        server 127.0.0.1:8081;

        server 127.0.0.1:8082;

        server 127.0.0.1:8083;

    }

    server {

        listen 8080;

        server_name localhost;
```

```
location / {
    proxy_pass http://golang_backend;

    proxy_set_header Host $host;

    proxy_set_header X-Real-IP $remote_addr;

    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}

}

}
```

Penjelasan Konfigurasi:

- `upstream golang_backend` → Mengelompokkan **beberapa server Golang** untuk load balancing.
- `server 127.0.0.1:8081;` , `server 127.0.0.1:8082;` , `server 127.0.0.1:8083;` → Tiga **instance Golang** berjalan di port berbeda.
- `proxy_pass http://golang_backend;` → Nginx meneruskan semua permintaan ke salah satu instance Golang.
- `proxy_set_header` → Meneruskan informasi HTTP dari klien ke backend.

▮ Setelah mengedit `nginx.conf` , restart Nginx untuk menerapkan perubahan:

```
cd C:\nginx-1.27.4

nginx -s reload
```

▮ 3. Jalankan Aplikasi Golang

Untuk menjalankan **beberapa instance** dari aplikasi Golang, gunakan perintah berikut:

```
$env:PORT = "8081"

go run main.go


$env:PORT = "8082"

go run main.go
```

```
$env:PORT = "8083"
```

```
go run main.go
```

❏ 4. Install Golang di Linux

Untuk menginstal **Golang versi terbaru** di Linux, gunakan perintah berikut:

```
wget https://go.dev/dl/go1.24.0.linux-arm64.tar.gz
```

```
rm -rf /usr/local/go && tar -C /usr/local -xzf go1.24.0.linux-arm64.tar.gz
```

```
export PATH=$PATH:/usr/local/go/bin
```

Pastikan **Golang sudah terinstal** dengan menjalankan:

```
go version
```

❏ 5. Menggunakan PowerShell untuk Pengujian API

❏ Mengirim 10 Permintaan GET ke API

Jalankan perintah berikut di **PowerShell** untuk mengirim **10 permintaan GET** ke API:

```
for ($i=1; $i -le 10; $i++) {  
  
    Invoke-WebRequest -Uri "http://localhost:8080/user" -Method Get  
  
}
```

❏ Mengirim 10 Permintaan POST ke API

Gunakan PowerShell untuk mengirim **10 permintaan POST** dengan data JSON:

```
for ($i=1; $i -le 10; $i++) {  
  
    $jsonData = @{  
  
        id = $i  
  
        name = "Alice$i"  
  
        email = "alice$i@example.com"  
  
        age = 25  
  
    }
```

```
} | ConvertTo-Json -Compress
```

```
Invoke-WebRequest -Uri "http://localhost:8080/user/create" `
-Method Post `
-Headers @{ "Content-Type"="application/json"} `
-Body $jsonData
}
```

📄 Kesimpulan

Panduan ini membantu Anda dalam:

- 📄 Menginstal & Menjalankan Nginx
- 📄 Mengonfigurasi Nginx untuk load balancing ke aplikasi Golang
- 📄 Menjalankan beberapa instance aplikasi Golang di port berbeda
- 📄 Menginstal Golang di Linux
- 📄 Menggunakan PowerShell untuk menguji API (GET & POST)
- 📄 Selamat mencoba! Jika ada pertanyaan, silakan tanyakan! 📄

📄 Kelebihan Markdown Ini:

- 📄 Struktur rapi dengan heading, kode terformat, dan emoji untuk mempermudah pemahaman.
- 📄 Dapat langsung digunakan di GitHub, GitLab, atau dokumentasi internal.
- 📄 Memisahkan setiap langkah dengan jelas sehingga mudah diikuti.

Coba jalankan di **Markdown Viewer** atau **GitHub README**, dan pastikan semua langkah berjalan dengan baik! 📄