



# Задание: Настройка NGINX как обратного прокси для 4 Hello World приложений

## Цель:

Развернуть 4 минимальных Hello World-приложения на разных языках (Python, Go, Java, Node.js) и настроить NGINX как обратный прокси, который будет направлять запросы на соответствующие приложения в зависимости от пути ( /python , /go , /java , /node ).

## Шаг 1: Установка зависимостей

```
sudo apt update
sudo apt install nginx openssl python3 python3-pip golang default-jdk nodejs
npm curl nano -y
```

## Шаг 2: Развёртывание приложений

### Python (Flask) — порт 8001

```
pip3 install flask
nano python_app.py
```

Вставь код:

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello():
    return "Hello from Python Flask!"
app.run(host='0.0.0.0', port=8001)
```

Запусти:

```
python3 python_app.py
```

### Node.js (Express) — порт 8002

```
mkdir nodeapp && cd nodeapp
npm init -y
npm install express
nano index.js
```

Вставь код:

```
const express = require('express');
const app = express();
app.get('/', (req, res) => res.send('Hello from Node.js Express!'));
app.listen(8002);
```

Запусти:

```
node index.js
```

## Go — порт 8003

```
nano go_app.go
```

Вставь код:

```
package main
import (
    "fmt"
    "net/http"
)
func handler(w http.ResponseWriter, r *http.Request) {
    fmt.Fprintf(w, "Hello from Go!")
}
func main() {
    http.HandleFunc("/", handler)
    http.ListenAndServe(":8003", nil)
}
```

Запусти:

```
go run go_app.go
```

## Java (Spring Boot) — порт 8004

```
curl -s "https://get.sdkman.io" | bash
source "$HOME/.sdkman/bin/sdkman-init.sh"
sdk install springboot
```

```
spring init --dependencies=web demo-java
cd demo-java
nano src/main/java/com/example/demo/DemoApplication.java
```

Вставь код:

```
package com.example.demo;
import org.springframework.boot.*;
import org.springframework.boot.autoconfigure.*;
import org.springframework.web.bind.annotation.*;
@RestController
@SpringBootApplication
public class DemoApplication {
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
    @GetMapping("/")
    public String home() {
        return "Hello from Java Spring Boot!";
    }
}
```

Запусти:

```
./mvnw spring-boot:run
```

---

## Шаг 3: Настройка NGINX

### Создай самоподписанный SSL-сертификат

```
sudo mkdir -p /etc/nginx/ssl
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -keyout /etc/nginx/ssl/selfsigned.key \
    -out /etc/nginx/ssl/selfsigned.crt \
    -subj "/CN=alinadobs"
```

Файл конфигурации `/etc/nginx/sites-available/helloworld.alinadobs`

```
server {
    listen 443 ssl;
    server_name helloworld.alinadobs;

    ssl_certificate      /etc/nginx/ssl/selfsigned.crt;
    ssl_certificate_key  /etc/nginx/ssl/selfsigned.key;
```

```

gzip on;
gzip_types text/plain application/json application/javascript text/css;

access_log /var/log/nginx/helloworld_access.log;
error_log /var/log/nginx/helloworld_error.log;

location /python/ {
    proxy_pass http://127.0.0.1:8001/;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
}

location /node/ {
    proxy_pass http://127.0.0.1:8002/;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
}

location /go/ {
    proxy_pass http://127.0.0.1:8003/;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
}

location /java/ {
    proxy_pass http://127.0.0.1:8004/;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
}
}

```

### Активируй конфигурацию:

```

sudo ln -s /etc/nginx/sites-available/helloworld.alinadobs /etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl reload nginx

```

## Шаг 4: Пропиши DNS (hosts)

### Linux / macOS:

```

sudo nano /etc/hosts

```

Добавь строку:

```
127.0.0.1 helloworld.alinadobs
```

## Windows:

1. Открой блокнот от имени администратора
2. Файл: `C:\Windows\System32\drivers\etc\hosts`
3. Добавь строку:

```
127.0.0.1 helloworld.alinadobs
```

## Чек-лист:

Этап	Проверка
	Установлены все зависимости (nginx, Python, Node.js, Go, Java)
	Все 4 приложения работают на своих портах
	NGINX проксирует запросы с <code>/python</code> , <code>/node</code> , <code>/go</code> , <code>/java</code>
	HTTPS работает через <code>https://helloworld.alinadobs/...</code>

Теперь ты умеешь: - Настраивать обратный прокси для нескольких сервисов - Работать с SSL и gzip в nginx - Писать и запускать минимальные приложения на 4 популярных языках