

PRÁCTICA 14: CONFIGURACIÓN MULTIUSUARIO CON NGINX PROXY MANAGER EN UN ENTORNO COMPARTIDO (CAMPUS)

OBJETIVO

Simular un entorno real de hosting donde varios alumnos, con distintas máquinas dentro de la misma red (con IP pública compartida), usan **Nginx Proxy Manager (NPM)** como gateway para acceder a sus servicios web mediante dominios personalizados y certificados SSL.

ESCENARIO DEL CAMPUS

- Todos los alumnos están conectados a la misma red local (LAN).
- Hay un único router con **una sola IP pública dinámica**.
- Se usará **DuckDNS o Freename** para asignar un dominio a cada alumno.
- Una **máquina central** tendrá instalado **Nginx Proxy Manager (NPM)** para hacer de **proxy inverso compartido**.

RESUMEN DE ARQUITECTURA

INTERNET

|



[IP pública dinámica del campus]

|

[Router con puertos 80 y 443 redirigidos]

|



[Servidor con Nginx Proxy Manager]

|

Entradas por dominio:
alumno1.duckdns.org
alumno2.duckdns.org
alumno3.forever

→ 192.168.1.101:80

→ 192.168.1.102:3000

→ 192.168.1.103:8080

Material necesario

- Un **servidor central** (Ubuntu Server o máquina con Docker y NPM).
- Cada alumno debe tener su propia **máquina Ubuntu Server** con:
 - Servicio web en algún puerto (Apache, Node.js, etc.).
 - IP interna estática (ej: 192.168.1.101).
- Acceso a **DuckDNS o Freename** para registrar subdominios.

- Un router con redirección de puertos 80 y 443 al servidor NPM.

PASOS

1. Configurar Nginx Proxy Manager en el servidor central

(Solo una vez para todo el grupo)

```
mkdir ~/npm && cd ~/npm  
nano docker-compose.yml
```

Contenido:

```
version: "3"  
services:  
  npm:  
    image: jc21/nginx-proxy-manager  
    container_name: nginx-proxy-manager  
    restart: always  
    ports:  
      - "80:80"  
      - "443:443"  
      - "81:81"  
    volumes:  
      - ./data:/data  
      - ./letsencrypt:/etc/letsencrypt
```

Iniciar:

```
docker-compose up -d
```

2. Redirigir puertos en el router

En el router del campus:

- Redirigir **puerto 80 y 443** → **IP local del servidor NPM**

3. Cada alumno registra su subdominio

Ejemplos:

- alumno1: alumno1.duckdns.org

- alumno2: alumno2.forever
 - alumno3: proyecto123.ddns.net
-

4. Cada alumno configura su máquina

1. Asignar IP estática o reservar en DHCP del router.
 - Ej: 192.168.1.101 para alumno1
 2. Ejecutar su servicio web en su máquina:
 - Apache: en puerto 80
 - Node.js: en puerto 3000
 - Python Flask: 5000
-

5. Configurar el proxy en NPM (admin)

Para cada alumno:

Proxy Host → Add Proxy Host

- **Domain Name:** alumno1.duckdns.org
- **Forward hostname/IP:** 192.168.1.101
- **Forward port:** 80 (o el que use su app)
- **SSL tab:**
 - Request new SSL cert
 - Activar **Force SSL** y **HTTP/2 support**

Repetir para cada alumno.

RESULTADO ESPERADO

- Cada alumno accede a su web con **su dominio propio**, con **SSL válido**:

```
https://alumno1.duckdns.org → redirigido a su PC por NPM
https://alumno2.forever → redirigido a otro PC
```

Beneficios educativos

- Simula un entorno real multiusuario de hosting.

- Practican DDNS, certificados, proxys, dominios y arquitectura real.
 - Escalable para laboratorios grandes o aulas con una sola IP.
-

Opcional: limitar acceso por IP interna

- Si quieres que solo ciertos dispositivos puedan acceder, puedes usar reglas de redireccionamiento IP o seguridad interna de NPM.
-

Requisitos técnicos clave

- IPs locales fijas o reservadas (ej. 192.168.1.101+).
- Un solo router, pero con **forwarding activo**.
- Dominio actualizado con IP pública dinámica (DuckDNS o similar).
- Docker y NPM en un solo servidor accesible desde fuera.