

Entendido. Veo que ya tienes los discos configurados en tu host Proxmox:

- **sda** (894GB): Sistema Proxmox (ya particionado con LVM)
- **sdb** (14.6TB): Ya particionado en dos partes:
 - sdb1 (10.9TB) → /mnt/DATA
 - sdb2 (3.6TB) → /mnt/NAS
- **sdc** (953GB): Ya particionado → /mnt/DATABASES

Como quieras usar el HDD entero sin particionar, voy a rehacer la guía asumiendo que **reformatearás el HDD de 14.6TB como una sola partición**. También ajustaré toda la guía para que use la estructura de discos correctamente.

Guía Completa: Servidor Proxmox + Nginx Proxy Manager + Cloudflare Tunnel

Versión sin Coolify - Entorno de Producción

Información del Proyecto

Componente	Especificación
CPU	Xeon 2683 v4
RAM	64GB
Disco Sistema Proxmox (sda)	894GB NVMe/SSD - Sistema Proxmox
SSD Bases de Datos (sdc)	953GB - Docker + Bases de Datos
HDD Datos (sdb)	14.6TB - Archivos pesados (SIN PARTICIONAR)
IP Servidor Proxmox	192.168.1.100
IP VM Ubuntu	192.168.1.101
Usuario	epicmaker

Dominios y Proyectos

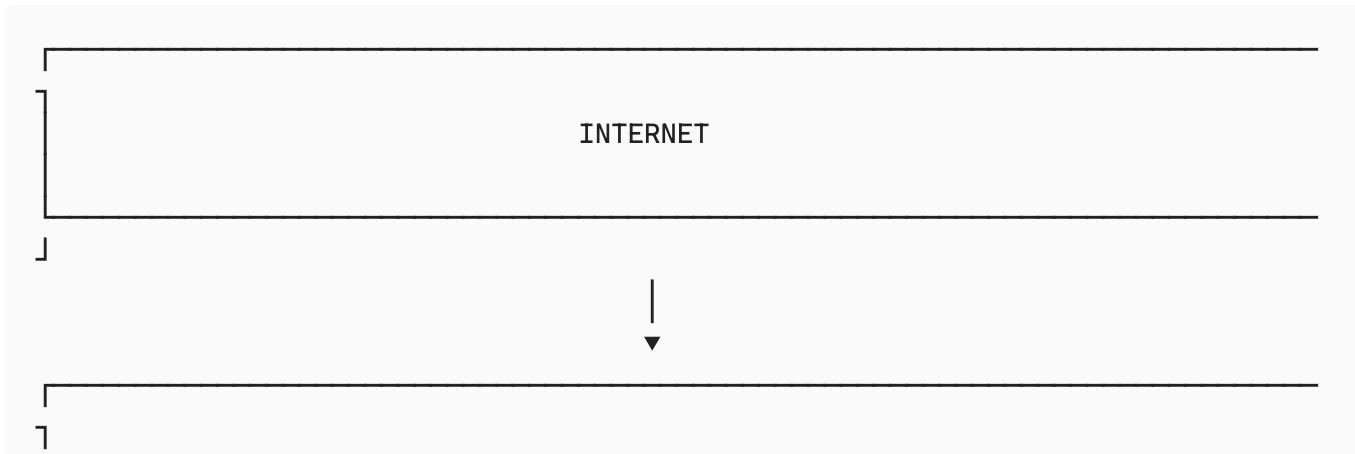
Dominio	Tipo	Descripción
lauraydaniel.es	Flutter Web + Appwrite	App de evento/boda
evolap.es	WordPress	Blog/Web corporativa
autodyno.es	HTML/CSS/JS	Web estática
epicmaker.dev	Panel NPM	Administración
mail-admin.es	Amazon SES	Envío de emails

Credenciales (GUÁRDALAS EN LUGAR SEGURO)

Servicio	Usuario	Contraseña
Proxmox	root	[Tu contraseña actual]
Ubuntu VM	epicmaker	Santander2020@.
Nginx Proxy Manager	admin@epicmaker.dev	NPM_Adm1n_2024!
MariaDB evolap	evolap_user	Ev0l4p_DB_2024!
MariaDB lauraydaniel	lauraydaniel_user	L4ur4D4n13l_DB!
WordPress evolap	admin	WP_Ev0l4p_2024!
Appwrite	admin@epicmaker.dev	Appwr1t3_2024!

IMPORTANTE: Cambia estas contraseñas por unas propias y guárdalas en un gestor como Bitwarden.

PARTE 1: Arquitectura del Sistema



CLOUDFLARE (CDN + WAF + SSL)

lauraydaniel.es | evolap.es | autodyno.es | epicmaker.dev



CLOUDFLARE TUNNEL (cloudflared)

Conexión segura SIN ABRIR PUERTOS en router



SERVIDOR PROXMOX (192.168.1.100)

Puerto Web: 8006



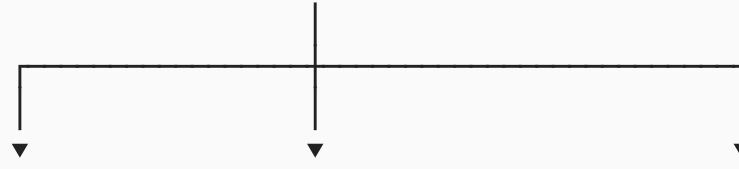
VM UBUNTU 24.04 (192.168.1.101)

12 vCPUs | 56GB RAM



NGINX PROXY MANAGER ← Gestiona dominios y certificados

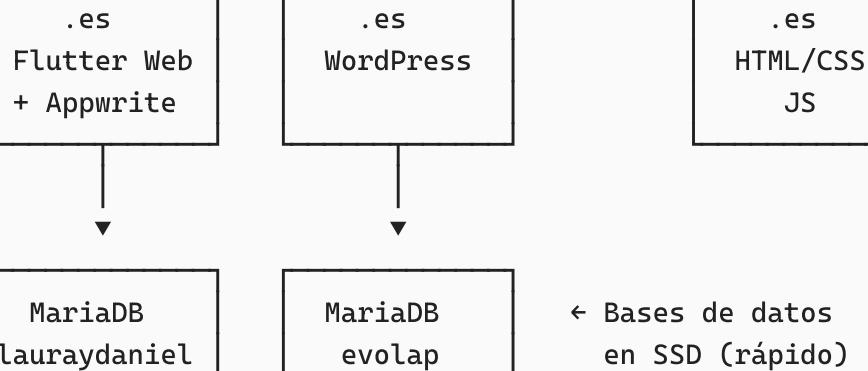
Puerto 80/443 (web) | 81 (panel admin)



lauraydaniel

evolap

autodyno



ALMACENAMIENTO EN HOST PROXMOX:

SSD 894GB (sda)
Sistema Proxmox
(no tocar)

SSD 953GB (sdc)
/mnt/DATABASES
Docker + BD
(passthrough VM)

HDD 14.6TB (sdb)
/mnt/HDD
Archivos + Media
(passthrough VM)

Distribución de almacenamiento

Disco	Tamaño	Uso	Montaje en VM
sda	894GB	Sistema Proxmox	No se toca
sdc	953GB	Docker + Bases de datos	/mnt/DATABASES
sdb	14.6TB	Archivos, media, uploads	/mnt/HDD

PARTE 2: Preparar Almacenamiento en Proxmox

2.1 Acceso a Proxmox

```
ssh root@192.168.1.100
```

2.2 Reformatear el HDD de 14.6TB (sdb) como una sola unidad

⚠️ ADVERTENCIA: Esto borrará TODOS los datos en sdb. Asegúrate de hacer backup si tienes algo importante.

```
# Ver estado actual
lsblk

# Desmontar las particiones actuales si están montadas
umount /mnt/DATA 2>/dev/null
umount /mnt/NAS 2>/dev/null

# Eliminar las particiones existentes y crear una nueva
wipefs -a /dev/sdb
parted /dev/sdb mklabel gpt
parted /dev/sdb mkpart primary ext4 0% 100%

# Formatear como ext4
mkfs.ext4 -L HDD-DATA /dev/sdb1

# Crear punto de montaje
mkdir -p /mnt/HDD

# Obtener UUID
blkid /dev/sdb1
```

Anota el UUID que aparece (ej: `UUID="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"`).

2.3 Configurar montaje permanente

Edita fstab:

```
nano /etc/fstab
```

Elimina las líneas antiguas de /mnt/DATA y /mnt/NAS si existen.

Añade esta línea (reemplaza con tu UUID real):

```
UUID=TU-UUID-SDB1 /mnt/HDD ext4 defaults 0 2
```

Montar y verificar:

```
mount -a  
df -h | grep HDD
```

Deberías ver: /dev/sdb1 ... /mnt/HDD con ~14.6TB disponibles.

2.4 Verificar que el SSD (sdc) está montado

```
df -h | grep DATABASES
```

Deberías ver: /dev/sdc1 ... /mnt/DATABASES

Si no está en fstab, añádelo:

```
blkid /dev/sdc1  
nano /etc/fstab
```

Añade:

```
UUID=TU-UUID-SDC1 /mnt/DATABASES ext4 defaults 0 2
```

2.5 Añadir almacenamiento en Proxmox Web UI

Accede a <https://192.168.1.100:8006>

Para el SSD de bases de datos:

1. Datacenter → Storage → Add → Directory
2. Configura:
 - **ID:** ssd-databases
 - **Directory:** /mnt/DATABASES
 - **Content:** Disk image, Container
3. Click Add

Para el HDD de datos:

1. Datacenter → Storage → Add → Directory
2. Configura:
 - **ID:** hdd-data
 - **Directory:** /mnt/HDD
 - **Content:** Disk image, Container, VZDump backup file, ISO image
3. Click Add

2.6 Descargar ISO de Ubuntu

```
cd /var/lib/vz/template/iso/  
wget https://releases.ubuntu.com/24.04/ubuntu-24.04.1-live-server-amd64.iso
```

PARTE 3: Crear Máquina Virtual Ubuntu

3.1 Crear VM desde interfaz web

Click en **Create VM**

Pestaña General

Campo	Valor
VM ID	100
Name	vm-servidor

Pestaña OS

Campo	Valor
ISO image	ubuntu-24.04.1-live-server-amd64.iso
Type	Linux
Version	6.x - 2.6 Kernel

Pestaña System

Campo	Valor
Machine	q35
BIOS	OVMF (UEFI)
Add EFI Disk	✓
EFI Storage	local-lvm
SCSI Controller	VirtIO SCSI single
Qemu Agent	✓

Pestaña Disks

Campo	Valor
Bus/Device	SCSI
Storage	local-lvm
Disk size	200 GB
Cache	Write back
Discard	✓
SSD emulation	✓

Pestaña CPU

Campo	Valor
Sockets	1
Cores	12
Type	host

Pestaña Memory

Campo	Valor
Memory	57344 MB (56GB)
Ballooning Device	✓

Pestaña Network

Campo	Valor
Bridge	vmbr0
Model	VirtIO (paravirtualized)
Firewall	✓

Click **Finish**.

3.2 Añadir disco SSD para Docker/BD

VM 100 → Hardware → Add → Hard Disk

Campo	Valor
Bus/Device	SCSI
Storage	ssd-databases
Disk size	900 GB
Cache	Write back
Discard	✓
SSD emulation	✓

3.3 Añadir disco HDD para datos

VM 100 → Hardware → Add → Hard Disk

Campo	Valor
Bus/Device	SCSI
Storage	hdd-data
Disk size	14000 GB
Cache	None

3.4 Instalar Ubuntu

Inicia la VM y abre la consola.

Durante la instalación:

- **Idioma:** Español
- **Teclado:** Español
- **Tipo:** Ubuntu Server (minimized)

Red (IMPORTANTE):

Campo	Valor
Subnet	192.168.1.0/24
Address	192.168.1.101
Gateway	192.168.1.1
DNS	1.1.1.1,8.8.8.8

Almacenamiento: Usa SOLO el disco de 200GB para el sistema. Los otros se configurarán después.

Usuario:

Campo	Valor
Hostname	servidor
Username	epicmaker
Password	Santander2020@.

SSH: Instalar OpenSSH server ✓

Reinicia cuando termine.

PARTE 4: Configuración Inicial de Ubuntu

4.1 Conectar por SSH

```
ssh epicmaker@192.168.1.101
```

4.2 Actualizar e instalar herramientas

```
sudo apt update && sudo apt upgrade -y
```

```
sudo apt install -y \
    qemu-guest-agent \
    curl \
    wget \
    git \
    htop \
    nano \
    net-tools \
    zip \
    unzip \
    ufw \
    fail2ban \
    ca-certificates \
    gnupg \
    lsb-release
```

```
sudo systemctl enable --now qemu-guest-agent
```

4.3 Identificar los discos

```
lsblk
```

Verás algo como:

- **sda**: ~200GB (sistema, ya montado en /)
- **sdb**: ~900GB (SSD para Docker/BD)
- **sdc**: ~14TB (HDD para datos)

4.4 Formatear y montar el SSD (sdb)

```
# Crear partición
sudo parted /dev/sdb mklabel gpt
sudo parted /dev/sdb mkpart primary ext4 0% 100%

# Formatear
sudo mkfs.ext4 -L SSD-DOCKER /dev/sdb1

# Crear punto de montaje
sudo mkdir -p /mnt/DATABASES

# Montar temporalmente
sudo mount /dev/sdb1 /mnt/DATABASES

# Obtener UUID
sudo blkid /dev/sdb1
```

Añadir a fstab:

```
sudo nano /etc/fstab
```

Añade (reemplaza con tu UUID real):

```
UUID=TU-UUID-SDB1 /mnt/DATABASES ext4 defaults 0 2
```

Verificar:

```
sudo mount -a  
df -h | grep DATABASES
```

4.5 Formatear y montar el HDD (sdc)

```
# Crear partición (SIN subdividir - disco completo)  
sudo parted /dev/sdc mklabel gpt  
sudo parted /dev/sdc mkpart primary ext4 0% 100%  
  
# Formatear (puede tardar varios minutos por el tamaño)  
sudo mkfs.ext4 -L HDD-DATA /dev/sdc1  
  
# Crear punto de montaje  
sudo mkdir -p /mnt/HDD  
  
# Montar temporalmente  
sudo mount /dev/sdc1 /mnt/HDD  
  
# Obtener UUID  
sudo blkid /dev/sdc1
```

Añadir a fstab:

```
sudo nano /etc/fstab
```

Añade:

```
UUID=TU-UUID-SDC1 /mnt/HDD ext4 defaults 0 2
```

Verificar:

```
sudo mount -a  
df -h | grep HDD
```

Deberías ver ~14TB disponibles.

4.6 Crear estructura de directorios

```
# En el SSD (rápido) - Para Docker y bases de datos  
sudo mkdir -p /mnt/DATABASES/docker-data  
sudo mkdir -p /mnt/DATABASES/apps/evolap-db
```

```
sudo mkdir -p /mnt/DATABASES/apps/lauraydaniel-db
sudo mkdir -p /mnt/DATABASES/apps/appwrite-db
sudo chown -R epicmaker:epicmaker /mnt/DATABASES

# En el HDD (grande) – Para archivos pesados
sudo mkdir -p /mnt/HDD/archivos
sudo mkdir -p /mnt/HDD/media
sudo mkdir -p /mnt/HDD/uploads
sudo mkdir -p /mnt/HDD/wordpress-uploads
sudo mkdir -p /mnt/HDD/appwrite-storage
sudo chown -R epicmaker:epicmaker /mnt/HDD
```

4.7 Configurar Firewall

```
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow ssh
sudo ufw allow 80/tcp
sudo ufw allow 81/tcp
sudo ufw allow 443/tcp
sudo ufw enable
```

4.8 Configurar Fail2ban

```
sudo systemctl enable fail2ban
sudo systemctl start fail2ban
sudo nano /etc/fail2ban/jail.local
```

Contenido:

```
[DEFAULT]
bantime = 3600
findtime = 600
maxretry = 5
ignoreip = 127.0.0.1/8 192.168.1.0/24

[sshd]
enabled = true
port = ssh
maxretry = 3
bantime = 86400
```

```
sudo systemctl restart fail2ban
```

PARTE 5: Instalar Docker

5.1 Instalar Docker

```
# Limpiar instalaciones anteriores
sudo apt remove docker docker-engine docker.io containerd runc 2>/dev/null

# Dependencias
sudo apt update
sudo apt install -y ca-certificates curl gnupg lsb-release

# Clave GPG
sudo mkdir -m 0755 -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -
o /etc/apt/keyrings/docker.gpg

# Repositorio
echo \
  "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >
/dev/null

# Instalar
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin

# Añadir usuario al grupo docker
sudo usermod -aG docker epicmaker
```

IMPORTANTE - Cierra sesión y reconecta:

```
exit
ssh epicmaker@192.168.1.101
```

Verificar:

```
docker --version  
docker compose version
```

5.2 Configurar Docker para usar el SSD

```
sudo systemctl stop docker  
sudo nano /etc/docker/daemon.json
```

Contenido:

```
{  
  "data-root": "/mnt/DATABASES/docker-data",  
  "storage-driver": "overlay2",  
  "log-driver": "json-file",  
  "log-opt": {  
    "max-size": "10m",  
    "max-file": "3"  
  }  
}
```

```
sudo systemctl start docker  
sudo systemctl enable docker  
docker info | grep "Docker Root Dir"
```

Debe mostrar: Docker Root Dir: /mnt/DATABASES/docker-data

5.3 Crear red compartida

```
docker network create proxy
```

5.4 Crear estructura de carpetas para proyectos

```
sudo mkdir -p /opt/docker  
sudo chown -R epicmaker:epicmaker /opt/docker
```

Estructura final:

```
/opt/docker/  
  └── npm/          # Nginx Proxy Manager  
  └── evolap/        # WordPress
```

```
└─ lauraydaniel/          # Flutter + Appwrite
   └─ autodyno/           # Web estática
```

PARTE 6: Instalar Nginx Proxy Manager

6.1 Crear directorio y configuración

```
mkdir -p /opt/docker/npm
cd /opt/docker/npm
nano docker-compose.yml
```

Contenido:

```
version: "3.8"

services:
  npm:
    image: jc21/nginx-proxy-manager:latest
    container_name: npm
    restart: always
    ports:
      - "80:80"
      - "81:81"
      - "443:443"
    environment:
      TZ: Europe/Madrid
      DB_SQLITE_FILE: "/data/database.sqlite"
    volumes:
      - ./data:/data
      - ./letsencrypt:/etc/letsencrypt
    networks:
      - proxy

networks:
  proxy:
    external: true
```

6.2 Levantar el contenedor

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

6.3 Acceder al panel

Abre en tu navegador:

```
http://192.168.1.101:81
```

Credenciales por defecto:

```
Email: admin@example.com  
Password: changeme
```

Cámbialo a:

```
Email: admin@epicmaker.dev  
Password: NPM_Adm1n_2024!
```

PARTE 7: Configurar Cloudflare

7.1 Crear cuenta y añadir dominios

1. Ve a cloudflare.com
2. Crea una cuenta
3. Añade cada dominio:
 - lauraydaniel.es
 - evolap.es
 - autodyno.es
 - epicmaker.dev
 - mail-admin.es

7.2 Cambiar Nameservers en GoDaddy

Para cada dominio en GoDaddy:

1. Ve a **Mis dominios** → selecciona el dominio
2. **DNS** → **Nameservers** → **Cambiar**
3. Selecciona **Personalizado**
4. Introduce los nameservers de Cloudflare (ej: aria.ns.cloudflare.com , brad.ns.cloudflare.com)

5. Guarda

7.3 Configurar SSL en cada dominio

En Cloudflare, para cada dominio:

1. **SSL/TLS → Overview** → Selecciona **Full (strict)**

2. **SSL/TLS → Edge Certificates**:

- Always Use HTTPS: **ON**
- Automatic HTTPS Rewrites: **ON**
- Minimum TLS Version: **TLS 1.2**

7.4 Crear Cloudflare Tunnel

1. Ve a **Zero Trust** (menú lateral)

2. **Networks → Tunnels**

3. **Create a tunnel**

4. Tipo: **Cloudflared**

5. Nombre: `servidor-epicmaker`

6. **Save tunnel**

Copia el token (empieza con `eyJ...`).

7.5 Instalar cloudflared en la VM

```
cd ~
curl -L --output cloudflared.deb
https://github.com.cloudflare/cloudflared/releases/latest/download/cloudflared-
linux-amd64.deb
sudo dpkg -i cloudflared.deb

# Instalar servicio con TU token
sudo cloudflared service install eyJhIjoiTU_TOKEN_COMPLETO...
```

Verificar:

```
sudo systemctl status cloudflared
```

7.6 Configurar rutas del túnel

En Cloudflare → Zero Trust → Tunnels → tu túnel → **Configure** → **Public Hostname**

Añade estas entradas:

Subdomain	Domain	Service Type	URL
(vacío)	lauraydaniel.es	HTTP	localhost:80
www	lauraydaniel.es	HTTP	localhost:80
api	lauraydaniel.es	HTTP	localhost:80
(vacío)	evolap.es	HTTP	localhost:80
www	evolap.es	HTTP	localhost:80
(vacío)	autodyno.es	HTTP	localhost:80
www	autodyno.es	HTTP	localhost:80
panel	epicmaker.dev	HTTP	localhost:81

PARTE 8: Configurar Amazon SES (mail-admin.es)

8.1 Acceder a SES

1. Ve a aws.amazon.com y accede
2. Busca **SES** en la consola
3. Selecciona región **eu-west-1** (Irlanda)

8.2 Verificar dominio mail-admin.es

1. **Verified identities** → **Create identity**
2. Tipo: **Domain**
3. Domain: `mail-admin.es`
4. **Create identity**

8.3 Añadir registros DNS en Cloudflare

En Cloudflare, dominio **mail-admin.es**, añade:

Registros DKIM (SES te da 3):

Tipo	Nombre	Contenido	TTL
CNAME	xxx._domainkey	xxx.dkim.amazonses.com	Auto
CNAME	yyy._domainkey	yyy.dkim.amazonses.com	Auto

Tipo	Nombre	Contenido	TTL
CNAME	zzz._domainkey	zzz.dkim.amazonses.com	Auto

Registro SPF:

Tipo	Nombre	Contenido	TTL
TXT	@	"v=spf1 include:amazonses.com ~all"	Auto

Registro DMARC:

Tipo	Nombre	Contenido	TTL
TXT	_dmarc	"v=DMARC1; p=quarantine; rua=mailto: admin@mail-admin.es "	Auto

8.4 Crear credenciales SMTP

1. **SMTP settings → Create SMTP credentials**
2. Nombre: ses-smtp-servidor
3. **Create user**
4. **DESCARGA las credenciales** (solo se muestran una vez)

Guarda estos datos:

```
SMTP_HOST=email-smtp.eu-west-1.amazonaws.com
SMTP_PORT=587
SMTP_USER=AKIAxxxxxxxxxxxx
SMTP_PASSWORD=Bxxxxxxxxxxxxxx
```

8.5 Solicitar producción

1. **Account dashboard → Request production access**
2. Completa el formulario

PARTE 9: Desplegar evolap.es (WordPress)

9.1 Crear directorio y configuración

```
mkdir -p /opt/docker/evolap
cd /opt/docker/evolap
nano docker-compose.yml
```

Contenido:

```
version: "3.8"

services:
  wordpress:
    image: wordpress:latest
    container_name: evolap-wordpress
    restart: always
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_NAME: evolap_db
      WORDPRESS_DB_USER: evolap_user
      WORDPRESS_DB_PASSWORD: Ev0l4p_DB_2024!
      WORDPRESS_TABLE_PREFIX: ev_
    volumes:
      - ./html:/var/www/html
      - /mnt/HDD/wordpress-uploads/evolap:/var/www/html/wp-content/uploads
  networks:
    - proxy
    - evolap-internal
  depends_on:
```

```
- db
```

```
db:
```

```
  image: mariadb:10.11
```

```
  container_name: evolap-db
```

```
  restart: always
```

```
  environment:
```

```
    MYSQL_ROOT_PASSWORD: Ev0l4p_R00t_2024!
```

```
    MYSQL_DATABASE: evolap_db
```

```
    MYSQL_USER: evolap_user
```

```
    MYSQL_PASSWORD: Ev0l4p_DB_2024!
```

```
volumes:
```

```
  - /mnt/DATABASES/apps/evolap-db:/var/lib/mysql
```

```
networks:
```

```
  - evolap-internal
```

```
networks:
```

```
proxy:
```

```
  external: true
```

```
evolap-internal:
```

```
  driver: bridge
```

9.2 Crear directorios para datos

```
# Base de datos en SSD (rápido)
sudo mkdir -p /mnt/DATABASES/apps/evolap-db
sudo chown -R 999:999 /mnt/DATABASES/apps/evolap-db

# Uploads en HDD (grande)
sudo mkdir -p /mnt/HDD/wordpress-uploads/evolap
sudo chown -R www-data:www-data /mnt/HDD/wordpress-uploads/evolap
```

9.3 Levantar los contenedores

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

9.4 Configurar Proxy Host en NPM

1. Accede a NPM: <http://192.168.1.101:81>
2. **Proxy Hosts → Add Proxy Host**

Pestaña Details:

- **Domain Names:** evolap.es , www.evolap.es
- **Scheme:** http
- **Forward Hostname / IP:** evolap-wordpress
- **Forward Port:** 80
- **Block Common Exploits:** ✓
- **Websockets Support:** ✓

Pestaña SSL:

- **SSL Certificate:** Request a new SSL Certificate
- **Force SSL:** ✓
- **HTTP/2 Support:** ✓
- **Email:** admin@epicmaker.dev

Click **Save**.

9.5 Completar instalación WordPress

1. Abre <https://evolap.es>
2. Selecciona idioma **Español**
3. Configura:
 - Título del sitio: **Evolap**

- Usuario: **admin**
- Contraseña: **WP_Ev0l4p_2024!**
- Email: admin@epicmaker.dev

4. Click **Instalar WordPress**

9.6 Configurar plugins esenciales

Instala estos plugins desde **Plugins → Añadir nuevo**:

1. **WP Mail SMTP** - Para enviar correos
2. **Wordfence Security** - Seguridad
3. **Yoast SEO** - Posicionamiento
4. **WP Super Cache** - Rendimiento

9.7 Configurar WP Mail SMTP

1. Ve a **WP Mail SMTP → Settings**
2. Configura:
 - **From Email:** noreply@mail-admin.es
 - **From Name:** Evolap
 - **Mailer:** Other SMTP
 - **SMTP Host:** email-smtp.eu-west-1.amazonaws.com
 - **Encryption:** TLS
 - **SMTP Port:** 587
 - **Authentication:** ON
 - **SMTP Username:** [Tu AKIAXXXXXXX]
 - **SMTP Password:** [Tu contraseña SES]
3. **Save Settings** y **Send Test Email**

PARTE 10: Desplegar autodyno.es (Web Estática)

10.1 Crear estructura del proyecto

```
mkdir -p /opt/docker/autodyno/www  
cd /opt/docker/autodyno
```

10.2 Crear los archivos de la web

index.html:

```
nano www/index.html
```

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Autodyno - Servicios de automoción profesional">
    <meta name="keywords" content="autodyno, coches, automoción, taller, reparación">
    <title>Autodyno - Servicios de Automoción</title>
    <link rel="stylesheet" href="css/styles.css">
</head>
<body>
    <header>
        <nav class="navbar">
            <div class="container">
                <a href="/" class="logo">
                    <span class="logo-auto">Auto</span><span class="logo-dyno">dyno</span>
                </a>
                <ul class="nav-menu">
                    <li><a href="#inicio">Inicio</a></li>
                    <li><a href="#servicios">Servicios</a></li>
                    <li><a href="#nosotros">Nosotros</a></li>
                    <li><a href="#contacto">Contacto</a></li>
                </ul>
                <button class="nav-toggle" aria-label="Abrir menú">
                    <span></span>
                    <span></span>
                    <span></span>
                </button>
            </div>
        </nav>
    </header>

    <main>
        <section id="inicio" class="hero">
            <div class="hero-content">
                <h1>Bienvenido a Autodyno</h1>
                <p>Tu taller de confianza para el cuidado de tu vehículo</p>
                <a href="#contacto" class="btn btn-primary">Solicitar Cita</a>
            </div>
        </section>
```

```
<section id="servicios" class="services">
  <div class="container">
    <h2>Nuestros Servicios</h2>
    <div class="services-grid">
      <div class="service-card">
        <div class="service-icon">🔧</div>
        <h3>Mecánica General</h3>
        <p>Reparaciones y mantenimiento completo de tu
vehículo.</p>
      </div>
      <div class="service-card">
        <div class="service-icon">▢</div>
        <h3>Neumáticos</h3>
        <p>Cambio, equilibrado y alineación de neumáticos.</p>
      </div>
      <div class="service-card">
        <div class="service-icon">🔋</div>
        <h3>Electricidad</h3>
        <p>Diagnóstico y reparación del sistema eléctrico.</p>
      </div>
      <div class="service-card">
        <div class="service-icon">❄️</div>
        <h3>Aire Acondicionado</h3>
        <p>Recarga y reparación del sistema de climatización.</p>
      </div>
    </div>
  </div>
</section>

<section id="nosotros" class="about">
  <div class="container">
    <h2>Sobre Nosotros</h2>
    <div class="about-content">
      <div class="about-text">
        <p>En <strong>Autodyno</strong> llevamos más de 15
años ofreciendo servicios de automoción de calidad.</p>
        <p>Contamos con las últimas tecnologías en diagnóstico
y reparación.</p>
      </div>
      <div class="about-stats">
        <div class="stat">
          <span class="stat-number">15+</span>
          <span class="stat-label">Años de
experiencia</span>
        </div>
      </div>
    </div>
  </div>
</section>
```

```

        </div>
        <div class="stat">
            <span class="stat-number">5000+</span>
            <span class="stat-label">Clientes
satisfechos</span>
        </div>
    </div>
</div>
</div>
</div>
</section>

<section id="contacto" class="contact">
    <div class="container">
        <h2>Contacto</h2>
        <div class="contact-content">
            <div class="contact-info">
                <div class="contact-item">
                    <span>📍</span>
                    <p>Calle Principal 123, Madrid</p>
                </div>
                <div class="contact-item">
                    <span>📞</span>
                    <p>+34 912 345 678</p>
                </div>
                <div class="contact-item">
                    <span>✉</span>
                    <p>info@autodyno.es</p>
                </div>
            </div>
            <form class="contact-form" id="contactForm">
                <input type="text" name="nombre" placeholder="Tu
nombre" required>
                <input type="email" name="email" placeholder="Tu
email" required>
                <textarea name="mensaje" placeholder="Tu mensaje"
rows="4" required></textarea>
                <button type="submit" class="btn btn-primary">Enviar
Mensaje</button>
            </form>
        </div>
    </div>
</section>
</main>

<footer>
    <div class="container">

```

```
        <p>&copy; 2024 Autodyn. Todos los derechos reservados.</p>
    </div>
</footer>

<script src="js/main.js"></script>
</body>
</html>
```

css/styles.css:

```
mkdir -p www/css
nano www/css/styles.css
```

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

:root {
    --primary-color: #e63946;
    --secondary-color: #1d3557;
    --light-gray: #f8f9fa;
}

html {
    scroll-behavior: smooth;
}

body {
    font-family: 'Segoe UI', system-ui, sans-serif;
    line-height: 1.6;
    color: #333;
}

.container {
    max-width: 1200px;
    margin: 0 auto;
    padding: 0 20px;
}

/* Navbar */
.navbar {
    background: var(--secondary-color);
    padding: 1rem 0;
```

```
    position: fixed;
    width: 100%;
    top: 0;
    z-index: 1000;
}

.navbar .container {
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.logo {
    font-size: 1.8rem;
    font-weight: bold;
    text-decoration: none;
}

.logo-auto { color: #fff; }
.logo-dyno { color: var(--primary-color); }

.nav-menu {
    display: flex;
    list-style: none;
    gap: 2rem;
}

.nav-menu a {
    color: #fff;
    text-decoration: none;
    transition: color 0.3s;
}

.nav-menu a:hover {
    color: var(--primary-color);
}

.nav-toggle {
    display: none;
    flex-direction: column;
    gap: 5px;
    background: none;
    border: none;
    cursor: pointer;
}
```

```
.nav-toggle span {
  width: 25px;
  height: 3px;
  background: #fff;
}

/* Hero */
.hero {
  background: linear-gradient(135deg, var(--secondary-color), #457b9d);
  color: #fff;
  min-height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
  text-align: center;
  padding-top: 80px;
}

.hero h1 {
  font-size: 3rem;
  margin-bottom: 1rem;
}

.hero p {
  font-size: 1.3rem;
  margin-bottom: 2rem;
  opacity: 0.9;
}

/* Buttons */
.btn {
  display: inline-block;
  padding: 12px 30px;
  border-radius: 5px;
  text-decoration: none;
  font-weight: 600;
  transition: all 0.3s;
  border: none;
  cursor: pointer;
}

.btn-primary {
  background: var(--primary-color);
  color: #fff;
}
```

```
.btn-primary:hover {
  background: #c1121f;
}

/* Sections */
section {
  padding: 80px 0;
}

section h2 {
  text-align: center;
  font-size: 2.5rem;
  margin-bottom: 3rem;
  color: var(--secondary-color);
}

/* Services */
.services {
  background: var(--light-gray);
}

.services-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 2rem;
}

.service-card {
  background: #fff;
  padding: 2rem;
  border-radius: 10px;
  text-align: center;
  box-shadow: 0 5px 15px rgba(0,0,0,0.08);
  transition: transform 0.3s;
}

.service-card:hover {
  transform: translateY(-10px);
}

.service-icon {
  font-size: 3rem;
  margin-bottom: 1rem;
}

.service-card h3 {
```

```
    color: var(--secondary-color);
    margin-bottom: 0.5rem;
}

/* About */
.about-content {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 4rem;
    align-items: center;
}

.about-text p {
    margin-bottom: 1rem;
    font-size: 1.1rem;
}

.about-stats {
    display: flex;
    justify-content: space-around;
    flex-wrap: wrap;
    gap: 2rem;
}

.stat {
    text-align: center;
}

.stat-number {
    display: block;
    font-size: 3rem;
    font-weight: bold;
    color: var(--primary-color);
}

.stat-label {
    color: var(--secondary-color);
}

/* Contact */
.contact {
    background: var(--light-gray);
}

.contact-content {
    display: grid;
```

```
grid-template-columns: 1fr 1fr;
gap: 4rem;
}

.contact-item {
  display: flex;
  align-items: center;
  gap: 1rem;
  margin-bottom: 1.5rem;
}

.contact-form input,
.contact-form textarea {
  width: 100%;
  padding: 12px 15px;
  margin-bottom: 1rem;
  border: 1px solid #ddd;
  border-radius: 5px;
  font-size: 1rem;
}

.contact-form input:focus,
.contact-form textarea:focus {
  outline: none;
  border-color: var(--primary-color);
}

/* Footer */
footer {
  background: var(--secondary-color);
  color: #fff;
  text-align: center;
  padding: 2rem;
}

/* Responsive */
@media (max-width: 768px) {
  .nav-menu {
    display: none;
    position: absolute;
    top: 100%;
    left: 0;
    width: 100%;
    background: var(--secondary-color);
    flex-direction: column;
    padding: 1rem 0;
  }
}
```

```

        text-align: center;
    }

.nav-menu.active { display: flex; }
.nav-toggle { display: flex; }
.hero h1 { font-size: 2rem; }

.about-content,
.contact-content {
    grid-template-columns: 1fr;
    gap: 2rem;
}
}

```

js/main.js:

```

mkdir -p www/js
nano www/js/main.js

```

```

document.addEventListener('DOMContentLoaded', function() {
    // Toggle menú móvil
    const navToggle = document.querySelector('.nav-toggle');
    const navMenu = document.querySelector('.nav-menu');

    if (navToggle) {
        navToggle.addEventListener('click', function() {
            navMenu.classList.toggle('active');
        });
    }

    // Cerrar menú al click en enlace
    document.querySelectorAll('.nav-menu a').forEach(link => {
        link.addEventListener('click', () => {
            navMenu.classList.remove('active');
        });
    });
}

// Formulario de contacto
const contactForm = document.getElementById('contactForm');
if (contactForm) {
    contactForm.addEventListener('submit', function(e) {
        e.preventDefault();
        alert('¡Gracias por tu mensaje! Te contactaremos pronto.');
        this.reset();
    });
}

```

```
}

  console.log('Autodyno loaded');
};


```

robots.txt:

```
nano www/robots.txt
```

```
User-agent: *
Allow: /

Sitemap: https://www.autodyno.es/sitemap.xml
```

sitemap.xml:

```
nano www/sitemap.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>https://www.autodyno.es/</loc>
    <lastmod>2024-12-01</lastmod>
    <changefreq>monthly</changefreq>
    <priority>1.0</priority>
  </url>
</urlset>
```

10.3 Crear docker-compose.yml

```
nano docker-compose.yml
```

```
version: "3.8"

services:
  web:
    image: nginx:alpine
    container_name: autodyno-web
    restart: always
    volumes:
      - ./www:/usr/share/nginx/html:ro
```

```
networks:  
  - proxy  
  
networks:  
  proxy:  
    external: true
```

10.4 Levantar el contenedor

```
docker compose up -d
```

10.5 Configurar Proxy Host en NPM

1. En NPM → **Proxy Hosts** → **Add Proxy Host**

Pestaña Details:

- **Domain Names:** autodyno.es , www.autodyno.es
- **Scheme:** http
- **Forward Hostname / IP:** autodyno-web
- **Forward Port:** 80
- **Block Common Exploits:** ✓

Pestaña SSL:

- **SSL Certificate:** Request a new SSL Certificate
- **Force SSL:** ✓
- **HTTP/2 Support:** ✓

Click **Save**.

PARTE 11: Desplegar lauraydaniel.es (Flutter Web + Appwrite)

11.1 Crear estructura de directorios

```
mkdir -p /opt/docker/lauraydaniel  
cd /opt/docker/lauraydaniel
```

11.2 Crear docker-compose.yml

```
nano docker-compose.yml
```

```
version: "3.8"

x-logging: &x-logging

logging:
  driver: json-file
  options:
    max-file: "5"
    max-size: "10m"

services:
  # =====
  # APPWRITE CORE
  # =====

  appwrite:
    image: appwrite/appwrite:1.5.7
    container_name: lauraydaniel-appwrite
    <<: *x-logging
    restart: always

  networks:
    - proxy
    - lauraydaniel-internal

  volumes:
```

- /mnt/DATA/appwrite-storage/uploads:/storage/uploads:rw
- /mnt/DATA/appwrite-storage/cache:/storage/cache:rw
- ./appwrite-config:/storage/config:rw
- ./appwrite-certificates:/storage/certificates:rw
- ./appwrite-functions:/storage/functions:rw

depends_on:

- mariadb
- redis

environment:

- _APP_ENV=production
- _APP_WORKER_PER_CORE=6
- _APP_LOCALE=es
- _APP_CONSOLE_WHITELIST_ROOT=enabled
- _APP_CONSOLE_WHITELIST_EMAILS=admin@epicmaker.dev
- _APP_SYSTEM_EMAIL_NAME=Laura y Daniel
- _APP_SYSTEM_EMAIL_ADDRESS=noreply@mail-admin.es
- _APP_OPTIONS_ABUSE=disabled
- _APP_OPTIONS_FORCE_HTTPS=disabled

-

_APP_OPENSSL_KEY_V1=0dLxP9Qv7rXk4sM2eUj8aTfY3wHbN6cVpBzRqTgSdJhQmAkEnWsZpUrTyG
hKdLpF0nJ2mC3xV5bA7fR9qT2dW4eS6vU8yX0

- _APP_DOMAIN=api.lauraydaniel.es
- _APP_DOMAIN_TARGET=api.lauraydaniel.es
- _APP_DOMAIN_FUNCTIONS=functions.lauraydaniel.es

```
- _APP_REDIS_HOST=redis  
- _APP_REDIS_PORT=6379  
- _APP_DB_HOST=mariadb  
- _APP_DB_PORT=3306  
- _APP_DB_SCHEMA=appwrite  
- _APP_DB_USER=appwrite_user  
- _APP_DB_PASS=Q9tA7pR4yN2sk8wF3dV0!  
- _APP_DB_ROOT_PASS=R00tS3cur3P4ss!98zTq  
- _APP_SMTP_HOST=email-smtp.eu-west-1.amazonaws.com  
- _APP_SMTP_PORT=587  
- _APP_SMTP_SECURE=tls  
- _APP_SMTP_USERNAME=AKIATESTSMTPUSER12345  
- _APP_SMTP_PASSWORD=BiTestSES/Password890!xyz  
- _APP_USAGE_STATS=enabled  
- _APP_STORAGE_LIMIT=30000000000  
- _APP_INFLUXDB_HOST=influxdb  
- _APP_INFLUXDB_PORT=8086  
- _APP_STATSD_HOST=telegraf  
- _APP_STATSD_PORT=8125
```

```
# =====
```

```
# APPWRITE REALTIME
```

```
# =====
```

```
appwrite-realtime:

  image: appwrite/appwrite:1.5.7

  container_name: lauraydaniel-realtime

  <<: *x-logging

  entrypoint: realtime

  restart: always

  networks:

    - proxy

    - lauraydaniel-internal

  depends_on:

    - mariadb

    - redis

  environment:

    - _APP_ENV=production

    - _APP_WORKER_PER_CORE=6

    - _APP_OPTIONS_ABUSE=disabled

    - _APP_OPENSSL_KEY_V1=0dLxP9Qv7rXk4sM2eUj8aTfY3wHbN6cVpBzRqTgSdJhQmAkEnWsZpUrTyG
      hKdLpF0nJ2mC3xV5bA7fR9qT2dw4eS6vU8yX0

    - _APP_REDIS_HOST=redis

    - _APP_REDIS_PORT=6379

    - _APP_DB_HOST=mariadb

    - _APP_DB_PORT=3306

    - _APP_DB_SCHEMA=appwrite
```

```
- _APP_DB_USER=appwrite_user

- _APP_DB_PASS=Q9tA7pR4yN2sK8wF3dV0!

- _APP_USAGE_STATS=enabled

# =====

# WORKERS

# =====

appwrite-worker-audits:

image: appwrite/appwrite:1.5.7

container_name: lauraydaniel-worker-audits

<<: *x-logging

entrypoint: worker-audits

restart: always

networks:

- lauraydaniel-internal

depends_on:

- redis

- mariadb

environment:

- _APP_ENV=production

-

_APP_OPENSSL_KEY_V1=0dLxP9Qv7rXk4sM2eUj8aTfY3wHbN6cVpBzRqTgSdJhQmAkEnWsZpUrTyG
hKdLpF0nJ2mC3xV5bA7fR9qT2dW4eS6vU8yX0

- _APP_REDIS_HOST=redis
```

- _APP_REDIS_PORT=6379
- _APP_DB_HOST=mariadb
- _APP_DB_PORT=3306
- _APP_DB_SCHEMA=appwrite
- _APP_DB_USER=appwrite_user
- _APP_DB_PASS=Q9tA7pR4yN2sk8wF3dV0!

appwrite-worker-webhooks:

```

image: appwrite/appwrite:1.5.7

container_name: lauraydaniel-worker-webhooks

<<: *x-logging

entrypoint: worker-webhooks

restart: always

networks:
  - lauraydaniel-internal

depends_on:
  - redis

environment:
  - _APP_ENV=production
  -
_APP_OPENSSL_KEY_V1=0dLxP9Qv7rXk4sM2eUj8aTfY3wHbN6cVpBzRqTgSdJhQmAkEnWsZpUrTyG
hKdLpF0nJ2mC3xV5bA7fR9qT2dW4eS6vU8yX0
  - _APP_REDIS_HOST=redis
  - _APP_REDIS_PORT=6379

```

```
appwrite-worker-deletes:

  image: appwrite/appwrite:1.5.7

  container_name: lauraydaniel-worker-deletes

  <<: *x-logging

  entrypoint: worker-deletes

  restart: always

  networks:

    - lauraydaniel-internal

  volumes:

    - /mnt/DATA/appwrite-storage/uploads:/storage/uploads:rw

    - /mnt/DATA/appwrite-storage/cache:/storage/cache:rw

    - ./appwrite-functions:/storage/functions:rw

    - ./appwrite-builds:/storage/builds:rw

    - ./appwrite-certificates:/storage/certificates:rw

  depends_on:

    - redis

    - mariadb

  environment:

    - _APP_ENV=production

    -
_APP_OPENSSL_KEY_V1=0dLxP9Qv7rXk4sM2eUj8aTfY3wHbN6cVpBzRqTgSdJhQmAkEnWsZpUrTyG
hKdLpF0nJ2mC3xV5bA7fR9qT2dW4eS6vU8yX0

    - _APP_REDIS_HOST=redis
```

```
- _APP_REDIS_PORT=6379  
- _APP_DB_HOST=mariadb  
- _APP_DB_PORT=3306  
- _APP_DB_SCHEMA=appwrite  
- _APP_DB_USER=appwrite_user  
- _APP_DB_PASS=Q9tA7pR4yN2sK8wF3dV0!
```

```
# =====
```

```
# DATABASE
```

```
# =====
```

```
mariadb:
```

```
  image: mariadb:10.11
```

```
  container_name: lauraydaniel-mariadb
```

```
  <<: *x-logging
```

```
  restart: always
```

```
  networks:
```

```
    - lauraydaniel-internal
```

```
  volumes:
```

```
    - /mnt/DATABASES/apps/lauraydaniel-db:/var/lib/mysql:rw
```

```
  environment:
```

```
    - MYSQL_ROOT_PASSWORD=R00tS3cur3P4ss!98zTq
```

```
    - MYSQL_DATABASE=appwrite
```

```
    - MYSQL_USER=appwrite_user
```

```
- MYSQL_PASSWORD=Q9tA7pR4yN2sK8wF3dV0!
```

```
command: 'mysqld --innodb-flush-method=fsync'
```

```
redis:
```

```
image: redis:7.2-alpine
```

```
container_name: lauraydaniel-redis
```

```
<<: *x-logging
```

```
restart: always
```

```
command: >
```

```
    redis-server
```

```
    --maxmemory      512mb
```

```
    --maxmemory-policy allkeys-lru
```

```
    --maxmemory-samples 5
```

```
networks:
```

```
- lauraydaniel-internal
```

```
volumes:
```

```
- ./redis-data:/data:rw
```

```
# =====
```

```
# MONITORING
```

```
# =====
```

```
influxdb:
```

```
image: appwrite/influxdb:1.5.0
```

```
container_name: lauraydaniel-influxdb

<<: *x-logging

restart: always

networks:

  - lauraydaniel-internal

volumes:

  - ./influxdb-data:/var/lib/influxdb:rw


telegraf:

image: appwrite/telegraf:1.4.0

container_name: lauraydaniel-telegraf

<<: *x-logging

restart: always

networks:

  - lauraydaniel-internal

environment:

  - _APP_INFLUXDB_HOST=influxdb

  - _APP_INFLUXDB_PORT=8086


# =====

# FLUTTER WEB FRONTEND

# =====

flutter-web:


```

```

image: nginx:alpine

container_name: lauraydaniel-web

restart: always

volumes:
  - ./flutter-build:/usr/share/nginx/html:ro
  - ./nginx-flutter.conf:/etc/nginx/conf.d/default.conf:ro

networks:
  - proxy

networks:
proxy:
  external: true

lauraydaniel-internal:
  driver: bridge

```

11.3 Crear configuración Nginx para Flutter

```

nano nginx-flutter.conf

server {
  listen 80;
  server_name lauraydaniel.es www.lauraydaniel.es;
  root /usr/share/nginx/html;
  index index.html;

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

  location / {

```

```

        try_files $uri $uri/ /index.html;
    }

location ~* \.(js|css|png|jpg|jpeg|gif|ico|svg|woff|woff2)$ {
    expires 1y;
    add_header Cache-Control "public, immutable";
}
}

```

11.4 Crear directorios

```

# Directorios locales
mkdir -p appwrite-config appwrite-certificates appwrite-functions appwrite-builds
mkdir -p redis-data influxdb-data flutter-build

# Base de datos en SSD
sudo mkdir -p /mnt/DATABASES/apps/lauraydaniel-db
sudo chown -R 999:999 /mnt/DATABASES/apps/lauraydaniel-db

# Storage en HDD
sudo mkdir -p /mnt/HDD/appwrite-storage/uploads
sudo mkdir -p /mnt/HDD/appwrite-storage/cache
sudo chown -R epicmaker:epicmaker /mnt/HDD/appwrite-storage

```

11.5 Crear página temporal de Flutter

```
nano flutter-build/index.html
```

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Laura y Daniel - Próximamente</title>
    <style>
        * { margin: 0; padding: 0; box-sizing: border-box; }
        body {
            font-family: 'Georgia', serif;
            background: linear-gradient(135deg, #f5f7fa 0%, #e4e8eb 100%);
            min-height: 100vh;
            display: flex;
            align-items: center;

```

```

        justify-content: center;
        text-align: center;
    }
    .container { padding: 2rem; }
    h1 { font-size: 3rem; color: #2c3e50; margin-bottom: 1rem; }
    .hearts { font-size: 2rem; color: #e74c3c; margin: 1rem 0; }
    p { font-size: 1.2rem; color: #7f8c8d; }
    .date { font-size: 1.5rem; color: #2c3e50; margin-top: 2rem; font-weight: bold; }

```

</style>

</head>

<body>

```

<div class="container">
    <h1>Laura & Daniel</h1>
    <div class="hearts">❤</div>
    <p>Nuestra web está en construcción</p>
    <p class="date">Próximamente...</p>
</div>

```

</body>

</html>

11.6 Levantar los contenedores

```
docker compose up -d
```

Espera unos minutos. Verificar:

```
docker ps | grep lauraydaniel
```

11.7 Configurar Proxy Hosts en NPM

Para el frontend (lauraydaniel.es):

1. NPM → Proxy Hosts → Add Proxy Host

Pestaña Details:

- **Domain Names:** lauraydaniel.es , www.lauraydaniel.es
- **Scheme:** http
- **Forward Hostname / IP:** lauraydaniel-web
- **Forward Port:** 80
- **Websockets Support:** ✓

Pestaña SSL:

- Request new SSL Certificate, Force SSL ✓

Para el API de Appwrite (api.lauraydaniel.es):

1. NPM → Proxy Hosts → Add Proxy Host

Pestaña Details:

- **Domain Names:** api.lauraydaniel.es
- **Scheme:** http
- **Forward Hostname / IP:** lauraydaniel-appwrite
- **Forward Port:** 80
- **Websockets Support:** ✓

Pestaña SSL:

- Request new SSL Certificate, Force SSL ✓
-

PARTE 12: Publicar en Google

12.1 Google Search Console

Para cada dominio:

1. Ve a search.google.com/search-console
2. **Añadir propiedad → Dominio**
3. Introduce el dominio
4. Google te da un registro TXT

En Cloudflare, añade:

Tipo	Nombre	Contenido
TXT	@	google-site-verification=XXXXXX

12.2 Enviar sitemaps

- **evolap.es:** https://evolap.es/sitemap_index.xml (generado por Yoast)
- **autodyno.es:** <https://autodyno.es/sitemap.xml>

12.3 Google Analytics

1. Ve a analytics.google.com
 2. Crear propiedad para cada sitio
 3. Añadir código gtag.js al <head>
-

PARTE 13: Comandos Útiles

Docker

```
docker ps                      # Ver contenedores  
docker logs -f nombre          # Ver logs  
docker exec -it nombre sh      # Entrar  
docker system prune -af        # Limpiar  
docker stats                   # Recursos
```

Por proyecto

```
# Evolap  
cd /opt/docker/evolap && docker compose logs -f  
  
# Autodyno  
cd /opt/docker/autodyno && docker compose restart  
  
# Lauraydaniel  
cd /opt/docker/lauraydaniel && docker compose logs -f appwrite
```

Sistema

```
htop                         # Monitor  
df -h                          # Discos  
free -h                        # RAM
```

Cloudflared

```
sudo systemctl status cloudflared  
sudo journalctl -u cloudflared -f
```

PARTE 14: Resumen de Almacenamiento

Ubicación	Disco	Uso
/mnt/DATABASES/docker-data	SSD 953GB	Datos Docker
/mnt/DATABASES/apps/evolap-db	SSD 953GB	BD WordPress
/mnt/DATABASES/apps/lauraydaniel-db	SSD 953GB	BD Appwrite
/mnt/HDD/wordpress-uploads	HDD 14.6TB	Uploads WordPress
/mnt/HDD/appwrite-storage	HDD 14.6TB	Storage Appwrite
/mnt/HDD/archivos	HDD 14.6TB	Archivos generales
/mnt/HDD/media	HDD 14.6TB	Archivos multimedia

Checklist Final

Proxmox

- HDD reformateado como unidad completa (14.6TB)
- Storage ssd-databases añadido
- Storage hdd-data añadido
- VM creada con 12 cores y 56GB RAM

Ubuntu VM

- IP estática 192.168.1.101
- SSD montado en /mnt/DATABASES
- HDD montado en /mnt/HDD (completo, sin particiones)
- Docker usando SSD
- Red proxy creada

Nginx Proxy Manager

- Instalado en :81
- Proxy hosts para todos los dominios

Cloudflare

- Dominios añadidos
- Tunnel funcionando

- SSL Full (strict)

Apps

- evolap.es (WordPress) funcionando
- autodyno.es (HTML/CSS/JS) funcionando
- lauraydaniel.es (Flutter + Appwrite) funcionando

SEO

- Search Console configurado
 - Sitemaps enviados
 - Analytics instalado
-

Versión: 2.0 - HDD completo sin particionar

Dominios: lauraydaniel.es, evolap.es, autodyno.es

Última actualización: Diciembre 2024