MODIFICACIONES PROYECTO REACT-SYMFONY + BD + SSL

García García, Ana María

GitHub: https://github.com/anaGG07/Despliegue.git

1. Modificaciones en Local

1.1 Estructura de Carpetas

```
proyecto/
├── frontend/
├── backend/
├── web/
│   ├── certs/
│   │   └── .gitignore
│   ├── default.conf
│   ├── DockerfileWebAMG
│   └── entrypoint.sh
├── database/
│   └── init.sql
├── .env
├── .env.template
├── .gitignore
└── docker-compose.yml
```

1.2 Crear Carpeta y Archivos para SSL

```
# Crear estructura en web/
mkdir -p web/certs
touch web/entrypoint.sh
chmod +x web/entrypoint.sh
```

1.3 Configurar entrypoint.sh

```
#!/bin/sh
set -e

# Verifica si el certificado ya existe
if [ ! -f /etc/nginx/certs/selfsigned.crt ]; then
    echo "Generando certificado autofirmado..."
    openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
        -subj "/C=ES/ST=Granada/L=Granada/O=IESHLANZ/OU=DAWT/CN=Ana Maria Garcia
Garcia/emailAddress=tu.email@ejemplo.com" \
        -keyout /etc/nginx/certs/selfsigned.key \
        -out /etc/nginx/certs/selfsigned.crt
fi

# Ejecuta Nginx en primer plano
exec nginx -g "daemon off;"
```

1.4 Configurar default.conf para SSL y PHPMyAdmin

```
# Redirección HTTP a HTTPS
server {
    listen 80;
```

```
    server_name localhost;
    return 301 https://$host$request_uri;
}

# Configuración HTTPS
server {
    listen 443 ssl;
    server_name localhost;

    # Certificados SSL
    ssl_certificate /etc/nginx/certs/selfsigned.crt;
    ssl_certificate_key /etc/nginx/certs/selfsigned.key;

    # Configuración SSL
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers on;
    ssl_ciphers HIGH:!aNULL:!MD5;

    # Frontend
    location / {
        proxy_pass http://frontend:5173;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;
    }

    # Backend
    location /api {
        proxy_pass http://backend:8000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;
    }

    # PHPMyAdmin
    location /phpmyadmin/ {
        proxy_pass http://phpmyadmin:80/;
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Forwarded-Port $server_port;
```

```
    }
}
```

1.5 Configurar .env en la Raíz

```
# Database Configuration
MYSQL_ROOT_PASSWORD=root_password
MYSQL_DATABASE=AMG_BD
MYSQL_USER=alumnoDAW
MYSQL_PASSWORD=passAMG
```

1.6 Configurar .gitignore en la Raíz

```
# Environment files
.env
.env.*
!.env.template

# SSL certificates
web/certs/*
!web/certs/.gitignore

# Node modules
node_modules/
dist/

# Symfony
/backend/var/
/backend/vendor/
```

1.7 Configurar .gitignore en web/certs

```
*
!.gitignore
```

2. Despliegue en AWS

2.1 Preparación en AWS

```

# Clonar repositorio / actualizar
cd ~/Despliegue
git clone https://github.com/anaGG07/Despliegue.git
git pull origin main
cd Despliegue

# Crear .env
touch .env
nano .env  # Añadir contenido del .env del frontend
```

2.2 Modificar default.conf en AWS

```
cd web
nano default.conf
```

Cambiar:

```
server_name localhost;
```

Por:

```
server_name ip-aws;
```

2.3 Modificar frontend/.env

cd ../frontend

nano .env

Cambiar:

```
VITE_API_URL=https://localhost
```

Por:

```
VITE_API_URL=https://ip-aws
```

2.3 Levantar Servicios

```
# Asegurarse de estar en la carpeta correcta
cd ~/Despliegue

# Levantar servicios
docker-compose up -d --build

# Verificar servicios
docker-compose ps
```

2.4 Verificación

Acceder desde el navegador a:

- Frontend: https://[ip-aws]
- Backend: https://[ip-aws]/api/amg
- PHPMyAdmin: https://[ip-aws]/phpmyadmin/

3 Permisos y Seguridad

- Asegurar puertos 80 y 443 abiertos en AWS
- Verificar permisos de los certificados SSL