



# Guía de Despliegue - SpeaklyPlan

Esta guía te ayudará a desplegar la aplicación SpeaklyPlan en tu propio servidor.



## Requisitos Previos

- **Node.js** 18.x o superior
- **PostgreSQL** 14.x o superior
- **Yarn** o npm
- **Git**
- **Cuenta de AWS S3** (para almacenamiento de imágenes)
- **Servidor** con Ubuntu/Debian (recomendado) o cualquier SO compatible con Node.js



## Configuración del Servidor

### 1. Instalar Dependencias del Sistema

```
# Actualizar el sistema
sudo apt update && sudo apt upgrade -y

# Instalar Node.js 18
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt-get install -y nodejs

# Instalar Yarn
sudo npm install -g yarn

# Instalar PM2 (para gestionar procesos)
sudo npm install -g pm2

# Instalar PostgreSQL
sudo apt install -y postgresql postgresql-contrib
```

### 2. Configurar PostgreSQL

```
# Acceder a PostgreSQL
sudo -u postgres psql

# Crear base de datos y usuario
CREATE DATABASE speaklyplan;
CREATE USER speaklyplan_user WITH ENCRYPTED PASSWORD 'tu_contraseña_segura';
GRANT ALL PRIVILEGES ON DATABASE speaklyplan TO speaklyplan_user;

# Salir
\q
```

### 3. Configurar AWS S3

1. Crear un bucket en AWS S3
2. Configurar IAM con permisos:
  - s3:PutObject

- s3:GetObject
- s3:DeleteObject

3. Obtener las credenciales: `AWS_ACCESS_KEY_ID` , `AWS_SECRET_ACCESS_KEY` , `AWS_REGION`

## Instalación de la Aplicación

### 1. Clonar el Repositorio

```
# En tu servidor
cd /var/www # o el directorio que prefieras
git clone <URL_DE_TU_REPOSITORIO> speaklyplan
cd speaklyplan
```

### 2. Instalar Dependencias

```
yarn install
```

### 3. Configurar Variables de Entorno

Crea el archivo `.env` en la raíz del proyecto:

```
# Base de Datos
DATABASE_URL="postgresql://speaklyplan_user:tu_contraseña_segura@localhost:5432/speaklyplan?connect_timeout=15"

# NextAuth
NEXTAUTH_URL="https://tudominio.com" # o http://localhost:3000 para desarrollo
NEXTAUTH_SECRET="genera_un_secret_aleatorio_aqui" # Genera con: openssl rand -base64 32

# AWS S3
AWS_ACCESS_KEY_ID="tu_access_key_id"
AWS_SECRET_ACCESS_KEY="tu_secret_access_key"
AWS_REGION="us-east-1" # o tu región
AWS_BUCKET_NAME="tu-bucket-name"
AWS_FOLDER_PREFIX="speaklyplan/" # opcional

# Abacus AI (para el tutor AI)
ABACUSAI_API_KEY="tu_api_key_de_abacus"
```

### 4. Configurar la Base de Datos

```
# Generar el cliente de Prisma
yarn prisma generate

# Ejecutar las migraciones
yarn prisma migrate deploy

# Poblar con datos iniciales (opcional)
yarn tsx scripts/seed.ts
```

### 5. Importar Datos Exportados (si aplica)

Si tienes una exportación de datos:

```
# Copia el archivo database_export.json a tu servidor
# Luego ejecuta:
node scripts/import-db.js
```

## 6. Construir la Aplicación

```
yarn build
```

## Despliegue en Producción

### Opción 1: Usando PM2 (Recomendado)

```
# Iniciar la aplicación
pm2 start yarn --name "speaklyplan" -- start

# Guardar la configuración de PM2
pm2 save

# Configurar PM2 para iniciar al arranque del sistema
pm2 startup
# Ejecuta el comando que PM2 te muestre

# Ver logs
pm2 logs speaklyplan

# Ver estado
pm2 status

# Reiniciar
pm2 restart speaklyplan

# Detener
pm2 stop speaklyplan
```

### Opción 2: Usando systemd

Crea el archivo `/etc/systemd/system/speaklyplan.service` :

```
[Unit]
Description=SpeaklyPlan Next.js Application
After=network.target

[Service]
Type=simple
User=www-data
WorkingDirectory=/var/www/speaklyplan
Environment=NODE_ENV=production
ExecStart=/usr/bin/yarn start
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Luego:

```
sudo systemctl enable speaklyplan
sudo systemctl start speaklyplan
sudo systemctl status speaklyplan
```

## Configuración de Nginx (Proxy Inverso)

```
server {
    listen 80;
    server_name tudominio.com www.tudominio.com;

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        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;
    }
}
```

```
# Activar la configuración
sudo ln -s /etc/nginx/sites-available/speaklyplan /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx
```



## Configurar SSL con Let's Encrypt

```
# Instalar Certbot
sudo apt install -y certbot python3-certbot-nginx

# Obtener certificado SSL
sudo certbot --nginx -d tudominio.com -d www.tudominio.com

# Renovación automática (ya configurada por defecto)
sudo certbot renew --dry-run
```



## Comandos Útiles

### Actualizar la Aplicación

```
cd /var/www/speaklyplan
git pull
yarn install
yarn build
pm2 restart speaklyplan
```

## Ver Logs

```
# Logs de PM2
pm2 logs speaklyplan

# Logs de Nginx
sudo tail -f /var/log/nginx/access.log
sudo tail -f /var/log/nginx/error.log
```

## Backup de Base de Datos

```
# Exportar
pg_dump -U speaklyplan_user -d speaklyplan > backup_$(date +%Y%m%d_%H%M%S).sql

# O usar el script personalizado
node scripts/export-simple.js
```

## Restaurar Base de Datos

```
# Desde SQL dump
psql -U speaklyplan_user -d speaklyplan < backup.sql

# Desde JSON export
node scripts/import-db.js
```

## Seguridad

### 1. Firewall

```
# Permitir SSH, HTTP y HTTPS
sudo ufw allow 22/tcp
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw enable
```

### 2. PostgreSQL

- Cambiar el puerto por defecto (5432)
- Configurar `pg_hba.conf` para conexiones seguras
- Usar contraseñas fuertes

### 3. Variables de Entorno

- Nunca commitear el archivo `.env` a Git
- Usar secretos seguros para `NEXTAUTH_SECRET`
- Rotar las credenciales de AWS periódicamente

## Actualizaciones

### Git

```
# Configurar Git
git init
git add .
git commit -m "Initial commit"

# Agregar repositorio remoto
git remote add origin <URL_DE_TU_REPOSITORIO>
git push -u origin main
```

### Flujo de Actualización

1. Hacer cambios en desarrollo
2. Commitear y pushear a Git
3. En el servidor:

```
bash
git pull
yarn install
yarn build
pm2 restart speaklyplan
```



### Resolución de Problemas

#### Error: “Cannot connect to database”

- Verificar que PostgreSQL esté corriendo: `sudo systemctl status postgresql`
- Verificar la `DATABASE_URL` en `.env`
- Verificar permisos del usuario en PostgreSQL

#### Error: “NextAuth configuration invalid”

- Verificar que `NEXTAUTH_URL` sea correcto
- Verificar que `NEXTAUTH_SECRET` esté configurado

#### Error: “AWS S3 upload failed”

- Verificar credenciales de AWS
- Verificar permisos del bucket
- Verificar que el bucket existe

### La aplicación no responde

```
# Verificar estado de PM2
pm2 status

# Ver logs
pm2 logs speaklyplan

# Reiniciar
pm2 restart speaklyplan
```

## Soporte

---

Para problemas o preguntas, consulta la documentación en el repositorio o contacta al equipo de desarrollo.

---

**¡Tu aplicación SpeaklyPlan está lista para producción! 🎉**