

Manual de Instalación en AWS con PostgreSQL

Escuelas Deportivas Chile - Guía de Despliegue

Índice

1. [Requisitos Previos](#)
 2. [Configuración de la Base de Datos \(RDS PostgreSQL\)](#)
 3. [Configuración del Servidor \(EC2\)](#)
 4. [Configuración de Almacenamiento \(S3\)](#)
 5. [Instalación de la Aplicación](#)
 6. [Configuración de Variables de Entorno](#)
 7. [Despliegue con PM2](#)
 8. [Configuración de Nginx \(Proxy Inverso\)](#)
 9. [Configuración SSL con Certbot](#)
 10. [Solución de Problemas](#)
 11. [Mantenimiento y Actualizaciones](#)
-

1. Requisitos Previos

Cuenta AWS

- Cuenta activa de AWS con permisos para crear:
- Instancias EC2
- Bases de datos RDS
- Buckets S3
- Security Groups

Software Local (para preparar archivos)

- Node.js 18+
- Git
- Cliente SSH (Terminal, PuTTY, etc.)

Servicios AWS Requeridos

Servicio	Propósito	Especificación Mínima
EC2	Servidor de aplicación	t3.small (2 vCPU, 2GB RAM)
RDS	Base de datos PostgreSQL	db.t3.micro (1 vCPU, 1GB RAM)
S3	Almacenamiento de archivos	Bucket estándar

2. Configuración de la Base de Datos (RDS PostgreSQL)

2.1 Crear Instancia RDS

1. Acceder a la consola AWS → RDS → “Create database”

2. Configuración:

```

Engine: PostgreSQL

Version: 15.x (o la más reciente estable)

Template: Free tier (para desarrollo) o Production

DB Instance Identifier: escuelas-deportivas-db

Master Username: postgres

Master Password: [Contraseña segura - guardar en lugar seguro]

```

1. Configuración de instancia:

Instance Class: db.t3.micro (desarrollo) o db.t3.small (producción)

Storage: 20 GB SSD (gp2)

Enable storage autoscaling: Sí

2. Conectividad:

VPC: Default VPC (o tu VPC personalizada)

Public access: Sí (solo para configuración inicial, luego cambiar a No)

VPC Security Group: Crear nuevo → escuelas-db-sg

Database port: 5432

3. Configuración adicional:

Initial database name: escuelas_deportivas

Backup retention: 7 días

Enable encryption: Sí

2.2 Configurar Security Group de RDS

1. EC2 → Security Groups → escuelas-db-sg

2. Inbound Rules:

Type: PostgreSQL

Port: 5432

Source: [Security Group de tu instancia EC2]

2.3 Obtener Endpoint de Conexión

Una vez creada la instancia, copiar el **Endpoint**:

Ejemplo: escuelas-deportivas-db.abc123xyz.us-east-1.rds.amazonaws.com

3. Configuración del Servidor (EC2)

3.1 Crear Instancia EC2

1. EC2 → “Launch Instance”

2. Configuración:

Name: escuelas-deportivas-server

AMI: Ubuntu Server 22.04 LTS

Instance type: t3.small

Key pair: Crear nuevo o usar existente

3. Network settings:

VPC: Default VPC

Subnet: Cualquier zona disponible

Auto-assign Public IP: Enable

Security group: Crear nuevo → escuelas-app-sg

4. Security Group Rules (escuelas-app-sg):

SSH (22) - Tu IP

HTTP (80) - 0.0.0.0/0

HTTPS (443) - 0.0.0.0/0

Custom TCP (3000) - Tu IP (para pruebas)

5. Storage:

Root volume: 20 GB gp3

3.2 Conectar a la Instancia

```
# Dar permisos al archivo de clave
chmod 400 tu-clave.pem
```

```
# Conectar vía SSH
ssh -i tu-clave.pem ubuntu@[IP-PUBLICA-EC2]
```

3.3 Instalar Dependencias del Sistema

```
# Actualizar 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 install -y nodejs

# Verificar instalación
node --version # Debe mostrar v18.x.x
npm --version

# Instalar Yarn
sudo npm install -g yarn

# Instalar PM2 (gestor de procesos)
sudo npm install -g pm2

# Instalar Nginx
sudo apt install -y nginx

# Instalar Git
sudo apt install -y git

# Instalar cliente PostgreSQL (para pruebas)
sudo apt install -y postgresql-client
```

3.4 Verificar Conexión a RDS

```
psql -h [ENDPOINT-RDS] -U postgres -d escuelas_deportivas
# Ingresar contraseña cuando se solicite

# Si conecta correctamente, salir con:
\q
```

4. Configuración de Almacenamiento (S3)

4.1 Crear Bucket S3

1. S3 → "Create bucket"
2. Configuración:
 - Bucket name: escuelas-deportivas-uploads
 - Region: Misma región que EC2
 - Object Ownership: ACLs disabled
 - Block Public Access: Desmarcar "Block all public access" (para archivos públicos)

4.2 Configurar Política del Bucket

Bucket → Permissions → Bucket Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadForPublicFolder",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::escuelas-deportivas-uploads/public/*"
    }
  ]
}
```

4.3 Configurar CORS

Bucket → Permissions → CORS:

```
[
  {
    "AllowedHeaders": ["*"],
    "AllowedMethods": ["GET", "PUT", "POST", "DELETE", "HEAD"],
    "AllowedOrigins": ["https://tudominio.com", "http://localhost:3000"],
    "ExposeHeaders": ["ETag"]
  }
]
```

4.4 Crear Usuario IAM para la Aplicación

1. IAM → Users → “Create user”

2. Configuración:

User name: escuelas-deportivas-app

Access type: Programmatic access

3. Crear política personalizada:

```
json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::escuelas-deportivas-uploads",
        "arn:aws:s3:::escuelas-deportivas-uploads/*"
      ]
    }
  ]
}
```

4. Guardar las credenciales:

- Access Key ID
- Secret Access Key

5. Instalación de la Aplicación

5.1 Preparar Archivos Localmente

En tu máquina local:

```
# Clonar o copiar el proyecto
cd /ruta/a/escuelas_deportivas_chile/nextjs_space

# Instalar dependencias
yarn install

# Generar cliente Prisma
yarn prisma generate

# Construir la aplicación
yarn build
```

5.2 Transferir Archivos al Servidor

Opción A - Usando SCP:

```
# Comprimir proyecto (excluyendo node_modules)
tar -czvf app.tar.gz --exclude='node_modules' --exclude='.next' --exclude='.build' .

# Transferir al servidor
scp -i tu-clave.pem app.tar.gz ubuntu@[IP-EC2]:~/
```

Opción B - Usando Git:

```
# En el servidor EC2
cd ~
git clone [URL-DE-TU-REPOSITORIO] escuelas-deportivas
```

5.3 Configurar en el Servidor

```
# Conectar al servidor
ssh -i tu-clave.pem ubuntu@[IP-EC2]

# Descomprimir (si usaste SCP)
mkdir -p ~/escuelas-deportivas
tar -xzf app.tar.gz -C ~/escuelas-deportivas

# Ir al directorio
cd ~/escuelas-deportivas

# Instalar dependencias
yarn install --production=false

# Generar cliente Prisma
yarn prisma generate
```

6. Configuración de Variables de Entorno

6.1 Crear archivo .env

```
cd ~/escuelas-deportivas
nano .env
```

6.2 Contenido del archivo .env

```
# Base de datos PostgreSQL (RDS)
DATABASE_URL="postgresql://postgres:[TU-PASSWORD]@[ENDPOINT-RDS]:5432/escuelas_deportivas?schema=public"

# NextAuth
NEXTAUTH_URL="https://tudominio.com"
NEXTAUTH_SECRET="[Generar 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"
AWS_BUCKET_NAME="escuelas-deportivas-uploads"
AWS_FOLDER_PREFIX="prod/"

# Configuración de producción
NODE_ENV="production"
```

6.3 Aplicar Migraciones de Base de Datos

```
# Crear tablas en la base de datos
yarn prisma db push

# Ejecutar seed (datos iniciales)
yarn prisma db seed

# Verificar que las tablas se crearon
yarn prisma studio
# Presionar Ctrl+C para salir
```

7. Despliegue con PM2

7.1 Construir la Aplicación

```
cd ~/escuelas-deportivas
yarn build
```

7.2 Crear Configuración PM2

```
nano ecosystem.config.js
```

Contenido:

```
module.exports = {
  apps: [{
    name: 'escuelas-deportivas',
    script: 'node_modules/next/dist/bin/next',
    args: 'start',
    cwd: '/home/ubuntu/escuelas-deportivas',
    instances: 'max',
    exec_mode: 'cluster',
    autorestart: true,
    watch: false,
    max_memory_restart: '1G',
    env: {
      NODE_ENV: 'production',
      PORT: 3000
    }
  }]
};
```


7.3 Iniciar Aplicación

```
# Iniciar con PM2
pm2 start ecosystem.config.js

# Ver estado
pm2 status

# Ver logs
pm2 logs escuelas-deportivas

# Configurar inicio automático
pm2 startup
pm2 save
```

7.4 Comandos Útiles de PM2

```
pm2 restart escuelas-deportivas # Reiniciar
pm2 stop escuelas-deportivas    # Detener
pm2 delete escuelas-deportivas  # Eliminar
pm2 monit                       # Monitor en tiempo real
```

8. Configuración de Nginx (Proxy Inverso)

8.1 Crear Configuración de Nginx

```
sudo nano /etc/nginx/sites-available/escuelas-deportivas
```

Contenido:

```

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

    # Redirigir HTTP a HTTPS (descomentar después de configurar SSL)
    # return 301 https://$server_name$request_uri;

    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_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;
        proxy_read_timeout 86400;
    }

    # Archivos estáticos
    location /_next/static {
        proxy_pass http://localhost:3000/_next/static;
        proxy_cache_valid 60m;
        add_header Cache-Control "public, immutable";
    }

    # Límites de tamaño para uploads
    client_max_body_size 10M;
}

```

8.2 Activar Configuración

```

# Crear enlace simbólico
sudo ln -s /etc/nginx/sites-available/escuelas-deportivas /etc/nginx/sites-enabled/

# Eliminar configuración default
sudo rm /etc/nginx/sites-enabled/default

# Verificar configuración
sudo nginx -t

# Reiniciar Nginx
sudo systemctl restart nginx
sudo systemctl enable nginx

```

9. Configuración SSL con Certbot

9.1 Instalar Certbot

```

sudo apt install -y certbot python3-certbot-nginx

```

9.2 Obtener Certificado SSL

```
sudo certbot --nginx -d tudominio.com -d www.tudominio.com
```

Seguir las instrucciones y proporcionar:

- Email para notificaciones
- Aceptar términos de servicio
- Elegir redirección automática a HTTPS

9.3 Renovación Automática

```
# Probar renovación
sudo certbot renew --dry-run

# Certbot configura automáticamente un cron job
```

10. Solución de Problemas

Error: “Connection refused” a la base de datos

1. Verificar Security Group de RDS permite conexiones desde EC2
2. Verificar que el endpoint RDS es correcto
3. Probar conexión manual:

```
bash
```

```
psql -h [ENDPOINT] -U postgres -d escuelas_deportivas
```

Error: “EACCES permission denied”

```
# Dar permisos al directorio
sudo chown -R ubuntu:ubuntu ~/escuelas-deportivas
```

Error: Aplicación no inicia

```
# Ver logs detallados
pm2 logs escuelas-deportivas --lines 100

# Verificar variables de entorno
cat ~/escuelas-deportivas/.env

# Reconstruir aplicación
cd ~/escuelas-deportivas
yarn build
pm2 restart escuelas-deportivas
```

Error: 502 Bad Gateway en Nginx

```
# Verificar que la app está corriendo
pm2 status

# Verificar logs de Nginx
sudo tail -f /var/log/nginx/error.log

# Verificar puerto
curl localhost:3000
```

Error: Uploads no funcionan

1. Verificar credenciales AWS en .env
2. Verificar permisos del bucket S3
3. Verificar política CORS del bucket

11. Mantenimiento y Actualizaciones

11.1 Actualizar la Aplicación

```
cd ~/escuelas-deportivas

# Detener aplicación
pm2 stop escuelas-deportivas

# Actualizar código (según tu método)
git pull origin main
# O subir nuevos archivos con SCP

# Instalar dependencias nuevas
yarn install

# Aplicar cambios de base de datos
yarn prisma db push

# Reconstruir
yarn build

# Reiniciar
pm2 restart escuelas-deportivas
```

11.2 Backups de Base de Datos

RDS configura backups automáticos. Para backup manual:

```
# Crear snapshot en AWS Console
# RDS → Databases → [tu-db] → Actions → Take snapshot
```

11.3 Monitoreo

```
# Monitorear con PM2
pm2 monit

# Ver uso de recursos
htop

# Ver espacio en disco
df -h
```

11.4 Logs Importantes

```
# Logs de la aplicación
pm2 logs escuelas-deportivas

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

# Logs del sistema
sudo journalctl -u nginx -f
```

Resumen de URLs y Endpoints

Recurso	URL/Endpoint
Aplicación	https://tudominio.com
Base de datos	[endpoint].rds.amazonaws.com:5432
Bucket S3	escuelas-deportivas-uploads.s3.amazonaws.com
Panel Admin	https://tudominio.com/admin/login

Checklist Final

- ☐ Instancia EC2 funcionando
- ☐ RDS PostgreSQL accesible
- ☐ Bucket S3 configurado
- ☐ Variables de entorno configuradas
- ☐ Migraciones de base de datos aplicadas
- ☐ Seed ejecutado (usuario admin creado)
- ☐ Aplicación corriendo con PM2
- ☐ Nginx configurado como proxy
- ☐ SSL/HTTPS habilitado


- [] Dominio apuntando a IP de EC2
- [] Backups automáticos habilitados en RDS

Contacto y Soporte

Para problemas específicos de AWS, consultar la [documentación oficial de AWS](https://docs.aws.amazon.com/) (<https://docs.aws.amazon.com/>).

Credenciales por defecto del administrador:

- Email: admin@escuelasdeportivas.cl
- Password: admin123

 **IMPORTANTE:** Cambiar la contraseña del administrador inmediatamente después de la instalación.