



Guía de Persistencia y Backups en EasyPanel

Fecha: 9 de Octubre, 2025

Aplicación: Mueblería La Económica

Estado: ☒ Configurado



Volúmenes Persistentes Configurados

Según tu configuración en EasyPanel:

1. Base de Datos PostgreSQL

Nombre: laeconomica-postgres-data
Montaje: `/var/lib/postgresql/data`

Qué protege:

- ☒ Todos los datos de PostgreSQL
- ☒ Usuarios, productos, ventas, inventario
- ☒ Configuración de la base de datos

Persistencia: ☒ Los datos NO se pierden en deploy

2. Directorio de Backups

Nombre: laeconomica-data
Montaje: `/backup`

Qué protege:

- ☒ Backups manuales de la base de datos
- ☒ Archivos de respaldo históricos

Persistencia: ☒ Los backups NO se pierden en deploy



Usuario Administrador

Credenciales por Defecto

Email: admin@laeconomica.com
Contraseña: Admin123!



IMPORTANTE: Cambia esta contraseña después del primer login.

Cómo Crear el Usuario Admin

Opción 1: Automático en Deploy

El script `seed-admin.sh` se ejecuta automáticamente si está en el Dockerfile. Verifica los logs del contenedor:

```
docker logs <container-name>
```

Busca:

```
✓ Usuario admin creado exitosamente!
```

Opción 2: Manual en Contenedor

```
# Entrar al contenedor
docker exec -it <container-name> sh

# Ejecutar seed
/app/seed-admin.sh
```

Opción 3: Desde EasyPanel Terminal

1. Ir a tu aplicación en EasyPanel
2. Pestaña "Terminal" o "Console"
3. Ejecutar:

```
sh /app/seed-admin.sh
```

Sistema de Backups

Backup Manual

Desde el contenedor:

```
docker exec -it <container-name> sh /app/backup-manual.sh
```

Con nombre personalizado:

```
docker exec -it <container-name> sh /app/backup-manual.sh "antes-actualizacion"
```

Resultado:

```
✓ Backup creado: /backup/manual-20251009_143022.sql
```

Ver Backups Disponibles

```
docker exec -it <container-name> ls -lh /backup/
```

Restaurar un Backup

```
docker exec -it <container-name> sh /app/restore-backup.sh /backup/manual-20251009_143022.sql
```

⚠ **CUIDADO:** Esto reemplazará todos los datos actuales.

Comportamiento en Deploy

Deploy Normal

1. EasyPanel detecta cambios en GitHub
2. Construye nueva imagen Docker
3. Detiene **contenedor** viejo
4. Inicia **contenedor** nuevo
5. ☒ Volúmenes persisten
6. ☒ Base de datos **intacta**
7. ☒ Backups **intactos**

Deploy Forzoso (Rebuild)

1. Rebuild completo de la imagen
2. Detiene **contenedor** viejo
3. Inicia **contenedor** nuevo
4. ☒ Volúmenes persisten
5. ☒ Base de datos **intacta**
6. ☒ Backups **intactos**

¿Cuándo SE PIERDEN los datos?

SOLO si:

- ☒ Eliminas el volumen `laeconomica-postgres-data` manualmente
- ☒ Eliminas el proyecto completo de EasyPanel
- ☒ Eliminas el servicio de PostgreSQL

NUNCA por:

- ☒ Deploy normal
- ☒ Deploy forzoso (rebuild)
- ☒ Actualización de código
- ☒ Reinicio de contenedor

Checklist de Protección de Datos

Antes de Cada Deploy

- ☐ Crear backup manual:

```
bash
docker exec -it <container> sh /app/backup-manual.sh "pre-deploy-$(date +%Y%m%d)"
```
- ☐ Verificar que los volúmenes están montados:

```
bash
docker inspect <container> | grep -A 10 "Mounts"
```
- ☐ Verificar backup guardado:

```
bash
docker exec -it <container> ls -lh /backup/
```

Después de Cada Deploy

- ☐ Verificar que la aplicación arrancó:

```
bash
docker logs <container> | tail -20
```
- ☐ Verificar conexión a base de datos:

```
bash
docker exec -it <container> sh -c 'echo "SELECT 1;" | psql $DATABASE_URL'
```
- ☐ Verificar volúmenes intactos:

```
bash
docker exec -it <container> ls -l /var/lib/postgresql/data
```

Estrategia de Backup Recomendada

Frecuencia

Evento	Backup
Antes de deploy	<input checked="" type="checkbox"/> Siempre
Después de cambios importantes	<input checked="" type="checkbox"/> Siempre
Semanal	<input checked="" type="checkbox"/> Recomendado
Mensual	<input checked="" type="checkbox"/> Recomendado

Retención

- Backups diarios: 7 días
- Backups semanales: 4 semanas
- Backups mensuales: 12 meses

Automatización Futura

Puedes crear un cron job en el contenedor:

```
# Backup diario a las 2 AM
0 2 * * * /app/backup-manual.sh "auto-$(date +%Y%m%d)"

# Limpiar backups antiguos (más de 30 días)
0 3 * * 0 find /backup -name "*.sql" -mtime +30 -delete
```



Verificación de Persistencia

Test de Persistencia

1. Crear datos de prueba:

```
sql
INSERT INTO "User" (email, name, role)
VALUES ('test@test.com', 'Test User', 'VENDEDOR');
```

2. Hacer deploy o reiniciar contenedor

3. Verificar que los datos persisten:

```
sql
SELECT * FROM "User" WHERE email = 'test@test.com';
```

4. Resultado esperado: ☒ Los datos siguen ahí



Recuperación de Desastres

Escenario 1: Datos Corruptos

```
# 1. Ver backups disponibles
docker exec -it <container> ls -lh /backup/

# 2. Restaurar último backup bueno
docker exec -it <container> sh /app/restore-backup.sh /backup/manual-20251009.sql
```

Escenario 2: Deploy Falló

```
# 1. Rollback en EasyPanel (UI)
# 2. O restaurar backup manualmente
docker exec -it <container> sh /app/restore-backup.sh /backup/pre-deploy.sql
```

Escenario 3: Volumen Eliminado Accidentalmente



Si tienes backups:

```
# 1. Recrear el volumen en EasyPanel
# 2. Restaurar desde backup más reciente
docker exec -it <container> sh /app/restore-backup.sh /backup/ultimo-backup.sql
```

Si NO tienes backups:

-  Los datos se perdieron
- **Lección:** Siempre hacer backups antes de operaciones críticas

Configuración Actual (Resumen)

Componente	Estado	Protegido
Base de datos PostgreSQL	 Configurado	 Sí
Volumen postgres-data	 Montado	 Sí
Volumen backup	 Montado	 Sí
Scripts de backup	 Incluidos	N/A
Usuario admin	 Seed listo	N/A
Deploy seguro	 Configurado	 Sí

Acciones Recomendadas Ahora

1. Hacer el rebuild con el nuevo Dockerfile
2. Ejecutar seed-admin.sh para crear usuario
3. Crear primer backup manual
4. Probar login con usuario admin
5. Verificar que todo funciona

Comandos Rápidos de Referencia

```
# Ver logs del contenedor
docker logs <container-name>

# Entrar al contenedor
docker exec -it <container-name> sh

# Crear usuario admin
docker exec -it <container-name> sh /app/seed-admin.sh






# Crear backup
docker exec -it <container-name> sh /app/backup-manual.sh

# Ver backups
docker exec -it <container-name> ls -lh /backup/

# Restaurar backup
docker exec -it <container-name> sh /app/restore-backup.sh /backup/archivo.sql

# Ver volúmenes montados
docker inspect <container-name> | grep -A 10 Mounts
```

Con esta configuración, tus datos están protegidos contra:

-  Deploys normales
-  Deploys forzosos
-  Reinicios de contenedor
-  Actualizaciones de código
-  Errores durante el deploy

Solo necesitas backups manuales para proteger contra:

- Eliminación accidental de volúmenes
- Corrupción de datos
- Cambios de esquema problemáticos



¡Tu aplicación tiene persistencia completa!