

Documentación Técnica: Script de Sincronización (Sync QA)

Fecha: 4 de Diciembre, 2025 Autor: Daniela Romero Archivo: ./sync_qa.sh

1. Origen y Necesidad

El Problema

Mantener un entorno de Quality Assurance (QA) o Desarrollo actualizado requiere que los datos sean lo más parecidos posible a los de Producción. Anteriormente, el proceso de actualización implicaba ejecutar manualmente una secuencia compleja de comandos en la terminal:

1. Crear un backup de la base de datos de producción (`pg_dump`).
2. Loguearse como superusuario para borrar la base de datos de QA antigua.
3. Crear una base de datos nueva y asignar permisos.
4. Restaurar el backup en la nueva base (`psql < ...`).
5. Copiar manualmente las imágenes nuevas de una carpeta a otra (`cp` o `rsync`).

Este proceso manual es propenso a errores humanos (borrar la base equivocada, olvidar un paso) y consume tiempo valioso.

La Solución

Se creó el script de Bash `sync_qa.sh` para automatizar todo este flujo en un solo comando. Esto garantiza que el entorno de QA pueda actualizarse en segundos, reduciendo el riesgo operativo.

2. Análisis del Código

El script combina comandos de sistema (Linux) con comandos de base de datos (PostgreSQL). A continuación, se detalla la lógica de cada sección crítica:

A. Definición de Variables

```
FECHA=$(date +%F) BACKUP_PATH="/home/user/backups/db"  
ARCHIVO_BACKUP="$BACKUP_PATH/backup_prod_$FECHA.sql"
```

- Origen: Se define una estructura dinámica para los nombres de archivo usando la fecha actual. Esto evita sobrescribir backups del mismo día y mantiene un historial ordenado.

B. Extracción de Datos (El Dump)

```
pg_dump -h localhost -U user -d db > $ARCHIVO_BACKUP
```

- Por qué `-h localhost`: Durante la configuración manual, descubrimos que PostgreSQL rechazaba la conexión local por "Peer Authentication" (el usuario de Linux no coincidía

con el usuario de DB).

- Solución: Al forzar la conexión vía host (`-h`), PostgreSQL cambia al método de autenticación por contraseña (MD5/Password), permitiendo el acceso correcto.

C. Reinicio de la Base de Datos (El Drop)

```
sudo -u postgres psql <<EOF
SELECT pg_terminate_backend(pid) ...;
DROP DATABASE "db_QA";
CREATE DATABASE "db_QA" OWNER "user";
```

- Uso de `sudo -u postgres` : Un usuario normal no puede borrar bases de datos que no le pertenecen o matar procesos de otros. Usamos el superusuario `postgres` para tener permisos absolutos.
- `pg_terminate_backend` : Antes de borrar (`DROP`), es obligatorio desconectar a cualquier usuario o app (como PM2) que esté usando la base de datos; de lo contrario, el comando fallaría porque la DB está "ocupada".
- `OWNER`: Es vital asignar `OWNER "user"` al crearla, para que la aplicación QA pueda leerla sin errores de permisos.

D. Sincronización de Archivos (Assets)

```
rsync -av --progress ~/app/backend/uploads/ ~/app-qa/backend/uploads/
```

- Por qué `rsync` y no `cp` : Las bases de datos solo guardan referencias (texto). Las imágenes físicas viven en el disco.
- `rsync` es incremental: Solo copia los archivos nuevos o modificados. Si Producción tiene 10GB de fotos y solo agregó 1MB hoy, `rsync` solo transfiere ese 1MB. `cp` copiaría los 10GB de nuevo innecesariamente.

3. Instrucciones de Uso: Paso a Paso

Sigue esta guía cada vez que quieras actualizar el entorno de QA.

Paso 1: Conexión

Conéctate a tu servidor vía SSH: `ssh user@tu_ip`

Paso 2: Ubicación

Asegúrate de estar en la carpeta raíz de tu usuario (donde guardaste el script): `cd ~`

Paso 3: Ejecución

Ejecuta el script llamándolo por su nombre:

```
./sync_qa.sh
```

Paso 4: Autenticación

Durante la ejecución, el script puede detenerse y pedirte contraseñas:

1. Contraseña de [sudo] para el usuario: Es la contraseña de tu usuario de Linux. La pide para borrar la base de datos vieja.
2. Password for user: Es la contraseña de la Base de Datos (la que está en tu archivo .env). La pide para hacer el backup y restaurarlo.

Paso 5: Verificación

Una vez que veas el mensaje verde === ;LISTO! QA ESTÁ SINCRONIZADO === :

1. Ve a tu navegador.
2. Abre tu sitio de QA (ej: qa.misitio.com.ar).
3. Presiona Ctrl + F5 para recargar ignorando caché.
4. Verifica que las nuevas propiedades e imágenes aparezcan correctamente.

4. Diagrama de Flujo del Script

```
[Inicio]
  |
  v
[Crear carpeta de Backups]
  |
  v
[pg_dump: Extraer datos de PROD] --> (Genera archivo .sql)
  |
  v
[sudo postgres: Matar conexiones activas en QA]
  |
  v
[DROP DATABASE QA] --> [CREATE DATABASE QA]
  |
  v
[psql: Restaurar archivo .sql en QA]
  |
  v
[rsync: Copiar imágenes nuevas de PROD a QA]
  |
  v
[Fin: Entorno Sincronizado]
```

5. Anexo: Código Fuente del Script

Copia y pega este contenido en un archivo llamado sync_qa.sh si necesitas recrearlo.

```
#!/bin/bash

# --- CONFIGURACIÓN ---
```

```

FECHA=$(date +%F) BACKUP_PATH="/home/user/backups/db"
ARCHIVO_BACKUP="$BACKUP_PATH/backup_prod_$FECHA.sql"

# Colores para mensajes
VERDE='\[033[0;32m'
NC='\[033[0m' # No Color

echo -e "${VERDE}== INICIANDO SINCRONIZACIÓN PROD -> QA ==${NC}"

# 1. Crear carpeta de backups si no existe
mkdir -p $BACKUP_PATH

# 2. Hacer Backup de Producción
echo -e "${VERDE}[1/5] Creando Backup de Producción...${NC}"
# Usamos -h localhost para evitar el error de autenticación Peer
pg_dump -h localhost -U user -d db > $ARCHIVO_BACKUP

if [ $? -eq 0 ]; then
    echo "Backup creado exitosamente: $ARCHIVO_BACKUP"
else
    echo "Error al crear el backup. Abortando."
    exit 1
fi

# 3. Borrar y Recrear DB de QA
echo -e "${VERDE}[2/5] Reiniciando Base de Datos de QA...${NC}"
# Usamos sudo -u postgres para tener permisos de DROP/CREATE sin contraseña
sudo -u postgres psql <<EOF
SELECT pg_terminate_backend(pid) FROM pg_stat_activity WHERE datname = 'db_QA'; DROP
DATABASE "db_QA";
CREATE DATABASE "db_QA" OWNER "user"; EOF

# 4. Restaurar Datos en QA
echo -e "${VERDE}[3/5] Restaurando datos en QA...${NC}"
psql -h localhost -U user -d db_QA < $ARCHIVO_BACKUP

# 5. Sincronizar Imágenes (Rsync)
echo -e "${VERDE}[4/5] Sincronizando imágenes...${NC}"
rsync -av --progress ~/app/backend/uploads/ ~/app-qa/backend/uploads/

# 6. Reiniciar Apps (Opcional, descomentar si es necesario)
# echo -e "${VERDE}[5/5] Reiniciando servidor QA...${NC}"
# pm2 restart all

echo -e "${VERDE}== ¡LISTO! QA ESTÁ SINCRONIZADO ==${NC}"

```