



Copia de seguridad de una base de datos Postgres - Linux ubuntu



Cristhian Santa Cruz

hace 2 años

Veremos la forma más simple de crear una **copia de seguridad de nuestra base de datos Postgres**. Si tomaste los curso de **Django** que tenemos en **Udemy** (<https://www.udemy.com/course/curso-profesional-de-programacion-web-con-python-y-django/?referralCode=5199FC8137B709B2F4C7>) y en esta web, seguro en algún momento tuviste la necesidad de que crearas una copia de seguridad de tu base de datos y que también re-establecieras una base de datos en base a una copia de seguridad. Para ello veamos cómo es que lo hacemos:

Pasos a seguir:

Primero definitivamente necesitas tener acceso a tu usuario Postgres.

El nombre de tu BD o tu BD creada.

El usuario sudo o autenticado de tu terminal ubuntu.

En caso de windows ejecuta PSQshell como administrador.

Dirígete desde la terminal a una carpeta que tenga permisos de usuario, si no crea una carpeta y dale permisos con el comando:

```
chmod 777 "nombre_carpeta"
```

Accede a tu usuario Postgres desde la terminal de comandos:

```
su postgres
```

Con el usuario postgres activo y dentro de la carpeta con permisos ejecuta:

```
pg_dump "nombre_bd" > "nombre_archivo_backup"
```

Con ello si te fijas la carpeta donde ejecutaste el comando, se habrá creado el archivo con el nombre especificado. Ese archivo ya contiene tu **backup** de base de datos.

Cursos

Destacados:



Vue JS

3612 estudiantes

(/cursos/detail/vue)



3580 estudiantes

(/cursos/detail/django-profesional)



2496 estudiantes

(/cursos/detail/vue-basico)



1422 estudiantes

(/cursos/detail/pinia-vue)

Re-Establecer una copia de base de datos a una base de datos existente.

Ahora si ya tiene una **BD creada para tu proyecto en Django** y necesitas utilizar tu copia de seguridad para no estar ingresando datos de nuevo, puedes re-establecer la copia de seguridad. Este caso también sirve si surge algún problema en tu proyecto y necesitas re-establecer una copia de seguridad:

vuelve a ingresar con el usuario Postgres:

```
su postgres
```

Ahora ve donde tienes el archivo de backup desde la terminal, sin salir del usuario

Postgres, y ejecuta:

```
psql "nombre_db" < "nombre_archivo_copia"
```

Esto debe cargar varias líneas de comando ejecutándose, de lo contrario algo anda mal, y posiblemente sean los permisos de usuario. Así que preocúpate de ello.

Ahora si ejecutas tu proyecto **Django** seguro que te mandara un error de migraciones en la terminal. Por tanto para quitar ese error, necesitas hacer una **falsa migración**.

Para ello primero elimina todas las migraciones de tus aplicaciones, en la carpeta migrations, luego de ello ve a la terminal y con tu entorno activado y a la altura del manage.py ejecuta:

```
python manage.py makemigrations
```

Luego de ello una falsa migración

```
python manage.py migrate --fake
```

Con ello todo estará bien y estarás cargando los datos del **backup**. Recuerda que tenemos un video de esto en nuestro canal de youtube como **Neunapp**.

¿La respuesta fue útil?