Vamos a hacer el deploy de una aplicación PHP con Laravel. El proceso no es muy distinto al realizado para hacer el despliegue de una aplicación PHP "normal". La ventaja es que al estar trabajando con Homestead todas las herramientas (composer, git, PHP, etc.) que necesita Heroku ya están instaladas.

Ejercicio 1. Deploy de un proyecto Laravel nuevo

Crea un proyecto Laravel nuevo y consulta el siguiente artículo donde se explica cómo hacer el despliegue.

Es una buena práctica configurar también las variables de entorno (que se encuentran en .env en local) a los valores de producción, desde el menú config del dashboard de la app, tal y como se muestra en el siguiente pantallazo:

Creamos un nuevo proyecto en nuestro Homestead **"laravel new heroku"**

Hacemos un git init.

```
vagrant@homestead:~/code/heroku$ clear
vagrant@homestead:~/code/heroku$ git init
Initialized empty Git repository in /home/vagrant/code/heroku/.git/
vagrant@homestead:~/code/heroku$ git add .
vagrant@homestead:~/code/heroku$ git commit -m "laravel-limpio"
[master (root-commit) b643f54] laravel-limpio
 Committer: vagrant <vagrant@homestead>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 91 files changed, 13864 insertions(+)
 create mode 100644 .editorconfig
 create mode 100644 .env.example
 create mode 100644 .gitattributes
 create mode 100644 .gitignore
```

Creamos el Procfile

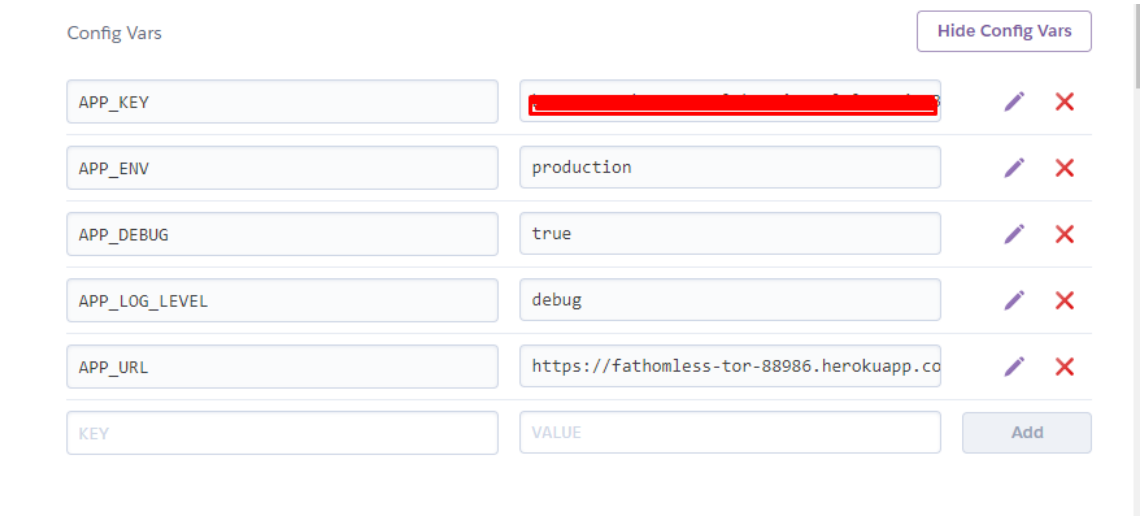
```
vagrant@homestead:~/code/heroku$ echo web: vendor/bin/heroku-php-apache2 public/ > Procfile
vagrant@homestead:~/code/heroku$
```

Configuramos la clave de aplicación

```
https://fathomless-tor-88986.herokuapp.com/ | https://git.heroku.com/fathomless-tor-88986.git
vagrant@homestead:~/code/heroku$ artisan key:generate --show
[redacted]
vagrant@homestead:~/code/heroku$ ^C
vagrant@homestead:~/code/heroku$ heroku config:set APP_KEY=[redacted]
```

Ya tenemos el deploy de nuestra aplicación realizado.

Configuramos las variables de entorno.

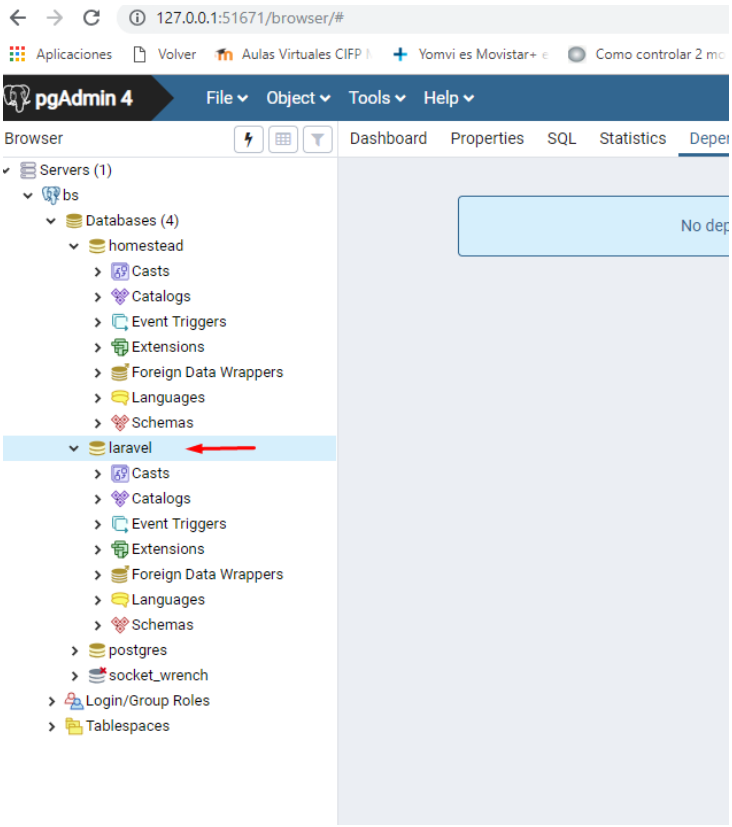


Ejercicio 2. Añadir y configurar la base de datos en local

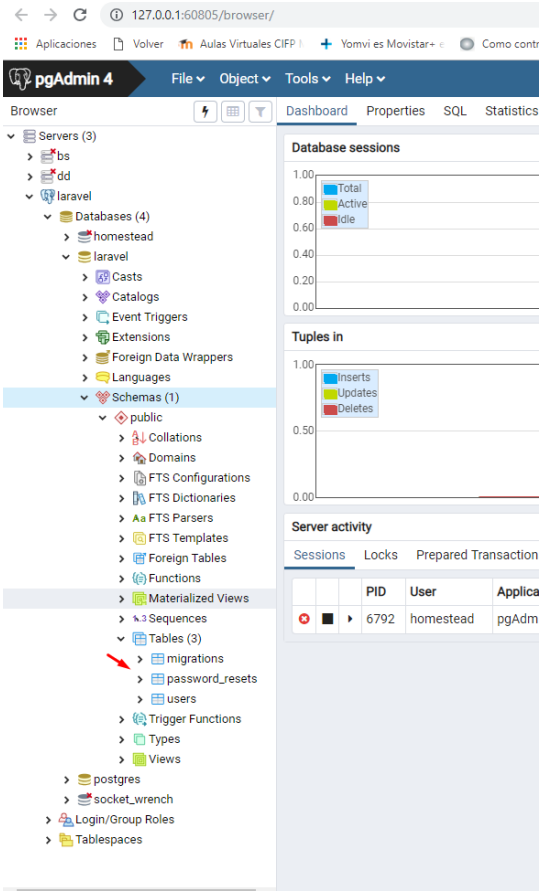
Homestead ya trae instalado Postgresql, que es lo que vamos a probar. ¿Por qué Postgresql y no Mysql o MariaDB? Pues porque ya lo hemos configurado en Heroku en otras prácticas y, además, es lo que proporciona por defecto Heroku sin necesidad de pasar por caja.

1. Para trabajar más cómodamente con la BBDD instala pgadmin (el phpmyadmin de Postgresql) en tu equipo (no en Homestead).
2. Crea una BBDD desde pgadmin y configura los valores en el fichero .env de la aplicación.
3. Ejecuta artisan migrate y comprueba que se han creado las migraciones para tu BBDD.

Creamos la BD laravel

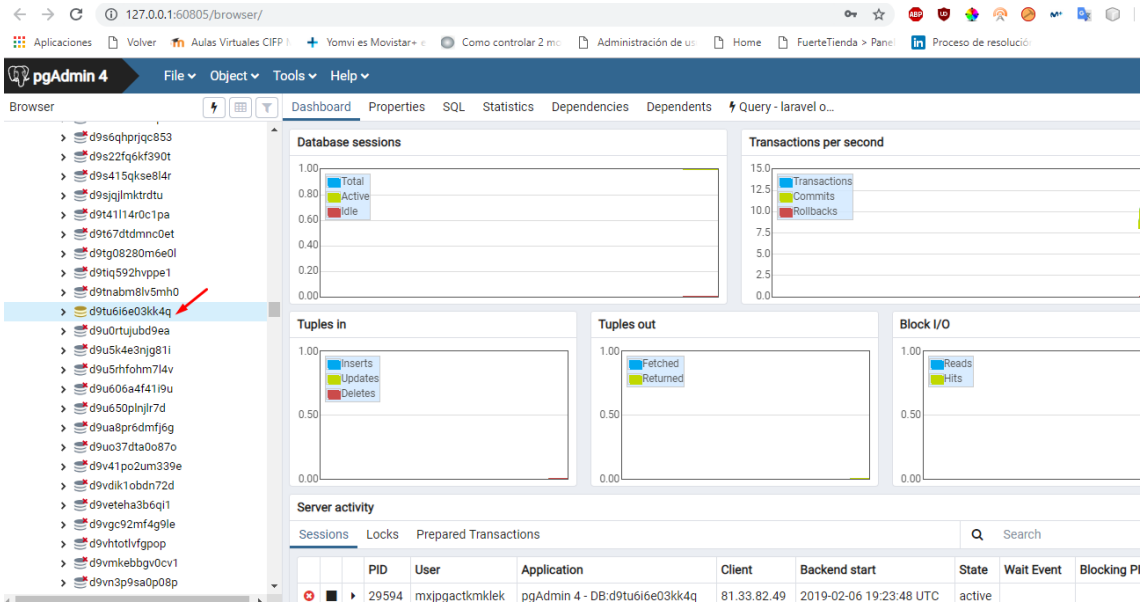


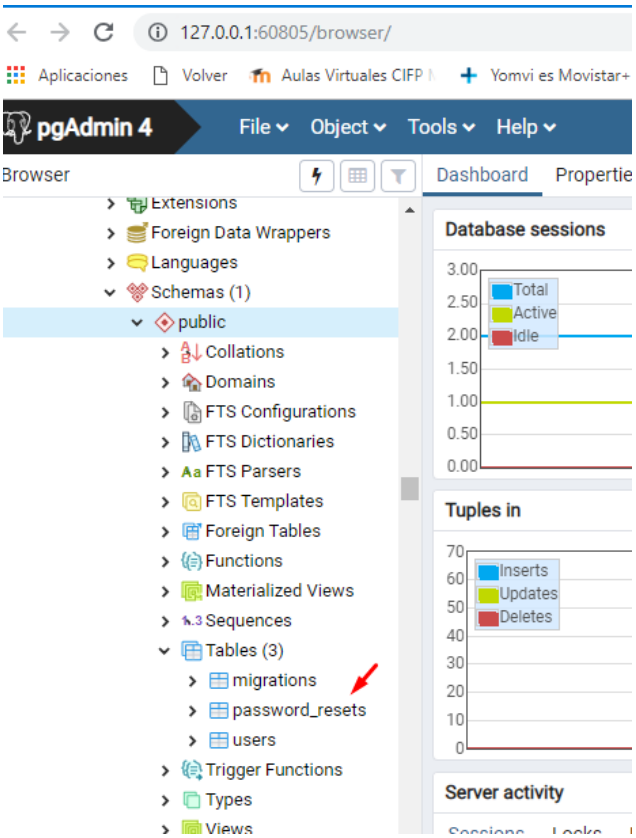
Vemos las tablas creadas.



Ejercicio 3. Añadir y configurar la BBDD en Heroku

En la práctica anterior creamos la BBDD para Heroku desde la línea de comando. En este ejercicio vamos a hacerlo desde el dashboard y configuraremos la BBDD en remoto (ten en cuenta que tendrás una BBDD en local -desarrollo- y otra en Heroku), para ello vamos a utilizar el siguiente curso gratuito de Udemy (inglés con transcripción), a partir del capítulo 9, inclusive.





Nos registramos en la aplicación e iniciamos sesión