

Ejercicio 1. Despliegue de una aplicación desde GitHub

Vamos a seguir el [tutorial de Heroku](#) para PHP.

Hay que tener instalado en local composer (además de git y PHP). Si utilizas tu máquina Debian puedes seguir el siguiente tutorial para instalarlo de manera global (hasta el paso 2, inclusive).

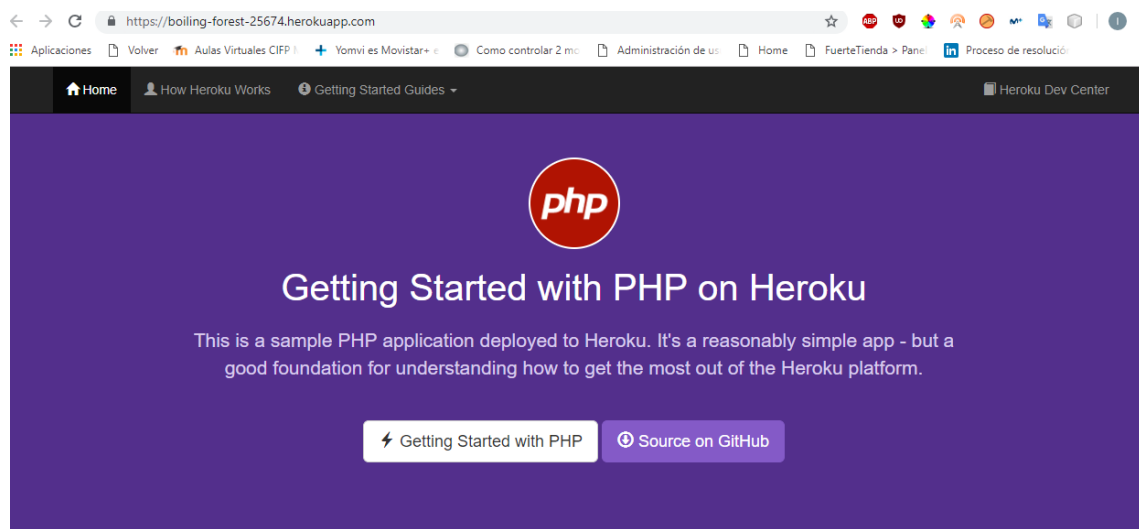
Instalamos Composer

```
root@debian-ivan:~# composer self-update -f
All settings correct for using Composer
Downloading...

Composer (version 1.8.3) successfully installed to: /usr/local/bin/composer
Use it: php /usr/local/bin/composer
```

Instanciamos la aplicación de prueba

```
remote: Verifying deploy... done.
To https://git.heroku.com/boiling-forest-25674.git
 * [new branch]      master -> master
root@debian-ivan:/home/ivan/heroku_php/php-getting-started# heroku ps:scale web=1
Scaling dynos... done, now running web at 1:Free
root@debian-ivan:/home/ivan/heroku_php/php-getting-started#
```

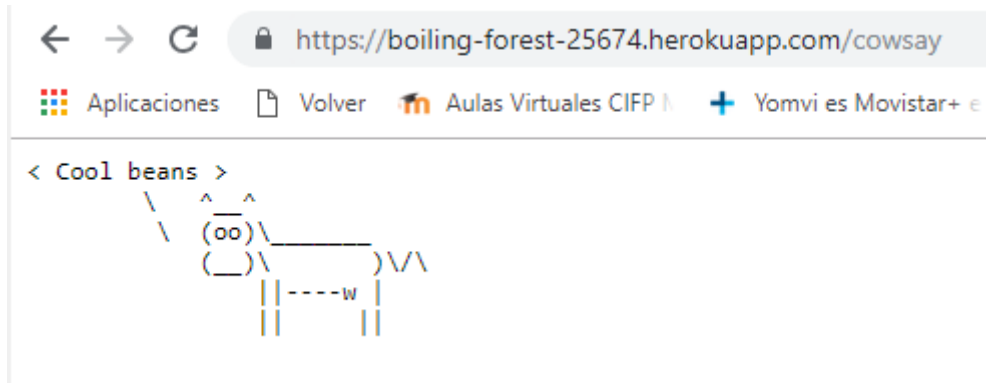


To deploy your own copy, and learn the fundamentals of the Heroku platform, head over to the [Getting Started with PHP on Heroku](#) tutorial.

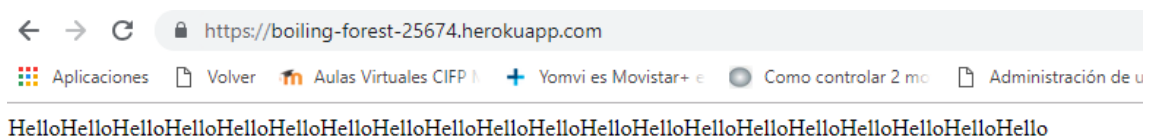
Vemos el profile de esa aplicación

```
root@debian-ivan:/home/ivan/heroku_php/php-getting-started# cat Procfile
web: vendor/bin/heroku-php-apache2 web/
```

Seguimos el tutorial y agregamos una nueva dependencia, agregamos la ruta “cowsay”, entramos en la web y vemos la vaca

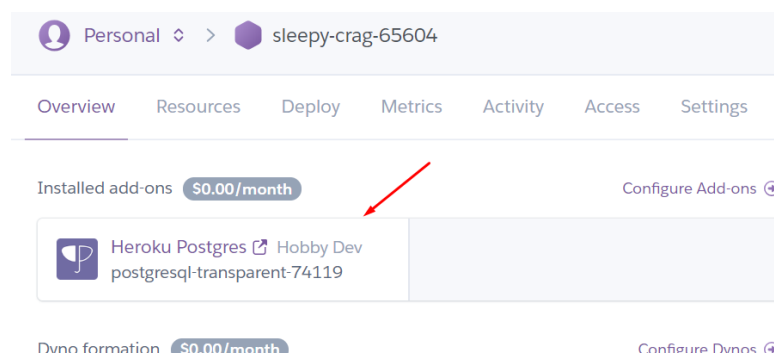


Realizamos el siguiente apartado del tutorial



Apartado de la BBDD en PostgreSQL

Agregamos el add-on de “heroku-postgresql” (heroku addons:create heroku-postgresql:hobby-dev)



Con “heroku config” vemos la DATABASE_URL, que utilizará mas adelante para conectarse

```
ivan@debian-ivan:~/heroku_php/php-getting-started$ heroku config
=== boiling-forest-25674 Config Vars
DATABASE_URL: postgres://rdwkoamviggutl:1e8a3d2e4b386e51fed413d22e5ddc1c52219b010352a66e47e4ae4893df8cd3@ec2-54-243-228-140.compute-1.amazonaws.com:5432/d56ef99rd0fgit
HEROKU_POSTGRES_URL: postgres://fveqkqulsukzit:7bf10fb017b12f05f9f94f8ba2d1bf5cdfe677da8dd018fba805a738dc90df06@ec2-107-20-185-27.compute-1.amazonaws.com:5432/d1s06qtl5hip0
NAME: 20
ivan@debian-ivan:~/heroku_php/php-getting-started$
```

Agregamos la dependencia de un proveedor de PDO

```
composer require csanquer/pdo-service-provider=~1.1dev
```

Modificamos Index.php para agregar la conexión con PDO

```
<?php

require('../vendor/autoload.php');

$dbopts = parse_url(getenv('DATABASE_URL'));
$app->register(new Csanquer\Silex\PdoServiceProvider\Provider\PdoServiceProvider('pdo'),
    array(
        'pdo.server' => array(
            'driver' => 'pgsql',
            'user' => $dbopts["user"],
            'password' => $dbopts["pass"],
            'host' => $dbopts["host"],
            'port' => $dbopts["port"],
            'dbname' => ltrim($dbopts["path"], '/')
        )
    )
);

$app = new Sillex\Application();
$app['debug'] = true;

// Register the monolog logging service
$app->register(new Sillex\Provider\MonologServiceProvider(), array(
    'monolog.logfile' => 'php://stderr',
```

Observamos como carga en una variable “\$dbopts” los datos para la conexión de la BBDD, y así poder pasársela en el array de la conexión.

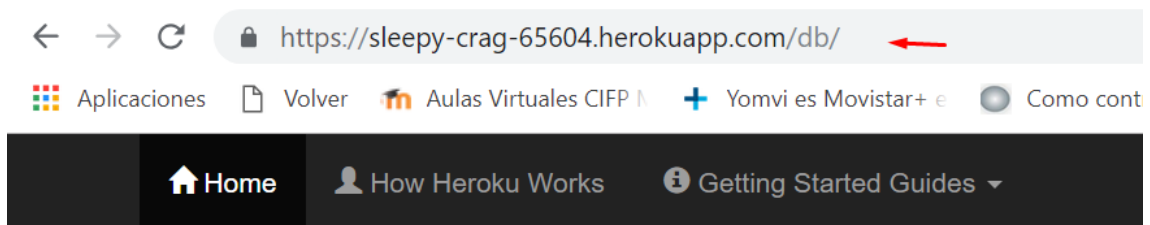
Agregamos un nuevo controlador que apunte a “/db”

```
$app->get('/db/', function() use($app) {
    $st = $app['pdo']->prepare('SELECT name FROM test_table');
    $st->execute();

    $names = array();
    while ($row = $st->fetch(PDO::FETCH_ASSOC)) {
        $app['monolog']->addDebug('Row ' . $row['name']);
        $names[] = $row;
    }

    return $app['twig']->render('database.twig', array(
        'names' => $names
    ));
});
```

Procedemos acceder a la aplicación en “/db”



Got these rows from the database:

- Nameless!

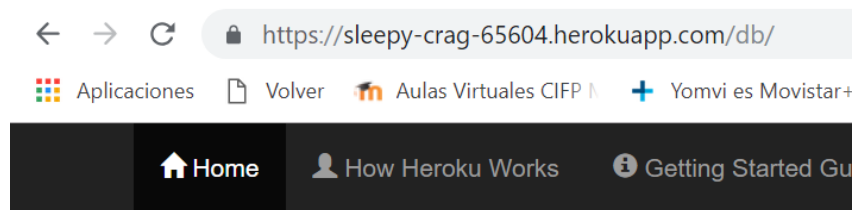
Nos conectamos a postgres con “heroku pg:psql”

```
root@debian-ivan:/home/ivan/heroku_php/php-getting-started/web/views# heroku ps:psql
> Warning: ps:psql is not a heroku command.
Did you mean pg:psql? [y/n]: y
--> Connecting to postgresql-shaped-18739
psql (9.6.10, servidor 10.6 (Ubuntu 10.6-1.pgdgl6.04+1))
ADVERTENCIA: psql versión mayor 9.6, servidor versión mayor 10.
Algunas características de psql podrían no funcionar.
conexión SSL (protocolo: TLSv1.2, cifrado: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compresión: desactivado)
Digite «help» para obtener ayuda.
boiling-forest-25674::DATABASE=>
```

Creamos la tabla e insertamos datos

```
boiling-forest-25674::DATABASE=> create table test_table (id integer,name text);
CREATE TABLE
boiling-forest-25674::DATABASE=> insert into test_table values(1,'hello database');
INSERT 0 1
boiling-forest-25674::DATABASE=>
```

Entramos en la web, y vemos los cambios realizados



Got these rows from the database:

- hello database
- Hola Jesús !!

Nos vamos al panel de administración y vemos como existe una tabla con 2 registros.

Datastores > postgresql-transparent-74119

SERVICE heroku-postgresql PLAN hobby-dev BILLING APP sleepy-crag-65604

Overview Durability Settings

HEALTH

✓ Available

PRIMARY Yes VERSION 10.6 CREATED 9 minutes ago MAINTENANCE Unsupported ⓘ ROLLBACK Unsupported ⓘ

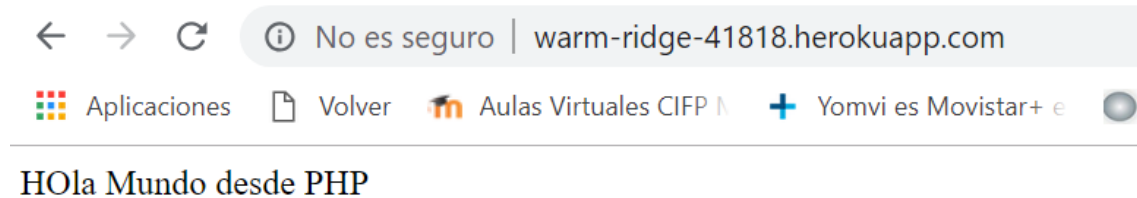
UTILIZATION

1 of 20	2 of 10,000	7.7 MB	1
CONNECTION	ROWS ✓ IN COMPLIANCE	DATA SIZE	TABLE











Ejercicio 2. Despliegue de una aplicación propia

De manera similar a lo que has hecho en el ejercicio 2 haz el deploy de una aplicación sencilla sin Homestead. [Aquí](#) tienes un artículo que lo explica.

Después de haber seguido el tutorial intente subir un proyecto con Laravel, pero me daba errores por todos lados, debido a ello decidí subir una aplicación sencilla que dijera “Hola Mundo desde PHP”



Vemos en el panel de administración que es una aplicación PHP

 mighty-cove-33513	 Node.js • heroku-18 • United States ★
 serene-retreat-88969	 Node.js • heroku-18 • United States ★
 sleepy-crag-65604	 PHP • heroku-18 • United States ★
 still-reef-56200	 Node.js • heroku-18 • United States ★
 warm-ridge-41818	 PHP • heroku-18 • United States ★