

Continuous Deployment - Github Actions



UNIVERSIDAD DE
COSTA RICA

SR-CIE

Carrera de
Informática Empresarial
Sedes Regionales

Qué es deployar?

- Deploy quiere decir “desplegar”
- En la industria se le suele decir “Hacer un Deploy” a la acción de instalar una aplicación en un servidor.

Cómo se hacía en los 90?

- Usando un FTP

- conectarse al ftp

- subir la carpeta del proyecto

- entrar en la página

- if (!funciona)

- Arreglar configuraciones, permisos, etc.

- else

- Fin.

Cómo se hace hoy?

- Instalación automatizada
- Servidores especiales
 - Heroku
 - Google App Engine
 - Amazon S3
- GIT
- Manejo de dependencias

Servidores de alojamiento

El alojamiento es el lugar donde va a correr nuestra aplicación y estará disponible públicamente

Podemos elegir cualquiera para nuestro proyecto, buscaremos uno gratuito que se integre fácilmente con nuestra app Laravel.



Heroku

Cómo hacemos un deploy a la nube?

Heroku es una plataforma como servicio (PaaS) que permite el building, ejecución y mantenimiento de aplicaciones.

Cuenta con un plan gratuito



Requisitos para instalación

- Cuenta en Heroku
- Heroku CLI

Instrucciones:

Heroku usa:

- Buildpacks: soporte de stacks de tecnología (PHP, Node, Java, etc)
- Servicios (BBDD, etc que necesites en tu app)

Instrucciones para Laravel (básicas):

<https://devcenter.heroku.com/articles/getting-started-with-laravel>

Heroku cuenta con una herramienta CLI llamada Heroku Toolbelt

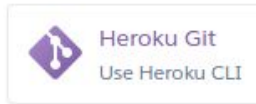
<https://devcenter.heroku.com/articles/heroku-cli>

```
$ heroku
```

```
$ heroku login
```

Formas de deploy en Heroku

Deployment method



GitHub
Connect to GitHub



Container Registry
Use Heroku CLI

¿Qué necesita un proyecto Heroku?

- b 1) Contar con una aplicación en Heroku
- ✓ 2) Archivo de configuración: Procfile
- b 3) Módulo PHP
- b 4) Addon de base de datos (Opcional)

Creación de una App en Heroku

Para crear una aplicación en Heroku:

> `heroku create myapp-staging`

Paréntesis: ¿Qué es Staging?

Es un ambiente intermedio entre testing y producción.
Se procura tener cambios **estables**



¿Qué otras cosas hace heroku create?

Crea:

- la aplicación en heroku.com
- un repositorio específico para el proyecto (alojado en Heroku).

Además, agrega una referencia git remota 'heroku' a nuestro proyecto.

Los cambios se hacen actualizando el branch master de este repositorio.

El Procfile determina el web server que correrá la aplicación

En este caso, usaremos apache + php

Procfile

```
web: vendor/bin/heroku-php-apache2 public/
```

Comandos para setup de la aplicación

```
echo web: vendor/bin/heroku-php-apache2 public/ >
```

```
Procfile
```

```
git add .
```

TMB CREA UN REMOTO EN GIT

```
git commit -m "Procfile for Heroku"
```

```
#heroku create
```

```
heroku buildpacks:set heroku/php
```

DESPUÉS VEMOS QUE
ES ESTO

```
heroku config:set APP_KEY="TOKEN DEL SERVER"
```

```
git push heroku main
```


Para instalar el addon de PostgreSQL en Heroku:

```
heroku addons:create heroku-postgresql:hobby-dev
```

Esto nos crea una variable de entorno **DATABASE_URL** que la debemos usar en `config/database.php` para determinar los datos de conexión.

SOPORTE PARA POSTGRES EN HEROKU:

<https://devcenter.heroku.com/articles/heroku-postgresql>

Setup Integración Continua Laravel + Heroku

Para configurar Laravel en Heroku, una solución simple es **prescindir del archivo de ambiente .env**, usando por el contrario **variables de entorno**.

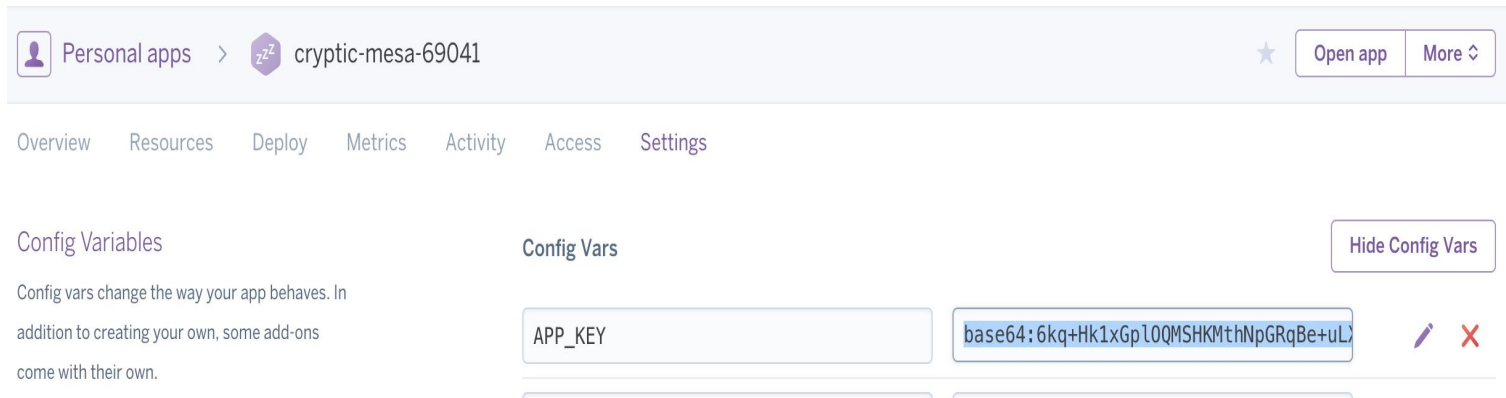
Las variables de entorno se alojan en Heroku.

Configuramos a través de variables de entorno las variables de Laravel:

- Conexión a Base de datos (DATABASE_URL) ***Auto-generada al agregar addon PostgreSQL**
- APP_KEY



Ingresamos al dashboard de Heroku y a la aplicación.
Agregar la variable **APP_KEY** en Settings → Config Variables



Es lo mismo que hicimos con:

```
heroku config:set APP_KEY="TOKEN DEL SERVER"
```


Configuración del ambiente: Variables de ambiente

Debemos agregar variables de ambiente a Heroku para poder conectar a la BBDD sin subir esa información al repositorio.

Config Vars

Config vars change the way your app behaves. In addition to creating your own, some addons come with their own.

Config Vars

APP_KEY	base64:tC/p/tK7
DATABASE_URL	postgres://satn
DB_CONNECTION	pgsql
KEY 	VALUE

Usando el repositorio de forma manual

Una vez creada la App, podemos evitar usar el CLI, usando el repositorio como medio de comunicación.

Para hacer un deploy, agregamos el repositorio y hacemos push.

```
> git remote -v
```

```
# git remote add heroku URL_REPO_HEROKU
```

```
> git push heroku master
```

```
$ heroku run php artisan migrate --seed
```

Este buildpack corre npm install en cada push

```
$ heroku buildpacks:add heroku/nodejs
```

```
{  
  "private": true,  
  "scripts": {  
    ...  
    "heroku-postbuild": "npm run prod"  
  },  
}
```

Llama a este comando después del install

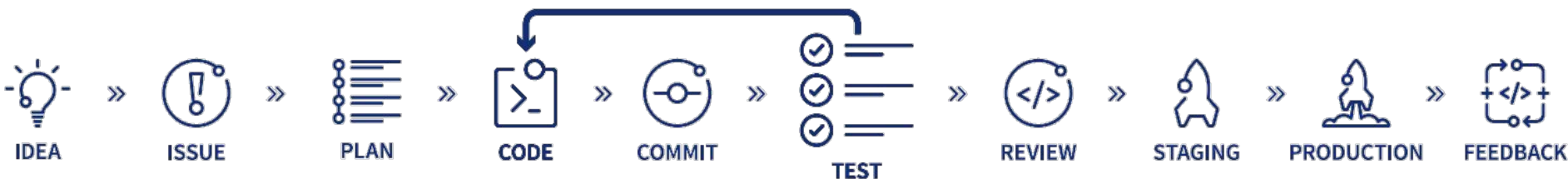
Continuous Deployment

Continuous Deployment (CD)

Continuous Delivery (CD)

Asegura que el nuevo código está en listo para ser desplegado (instalado)

- reduce el tiempo de ciclo
- crea feedback rápido entre el desarrollo y los clientes





El objetivo es agregar una etapa de **deploy**

2 técnicas para implementar el deploy en Heroku:

- **Push a Heroku desde Github**
- **Pull desde Heroku vía SSH**



Agregamos un nuevo workflow

```
name: Deploy

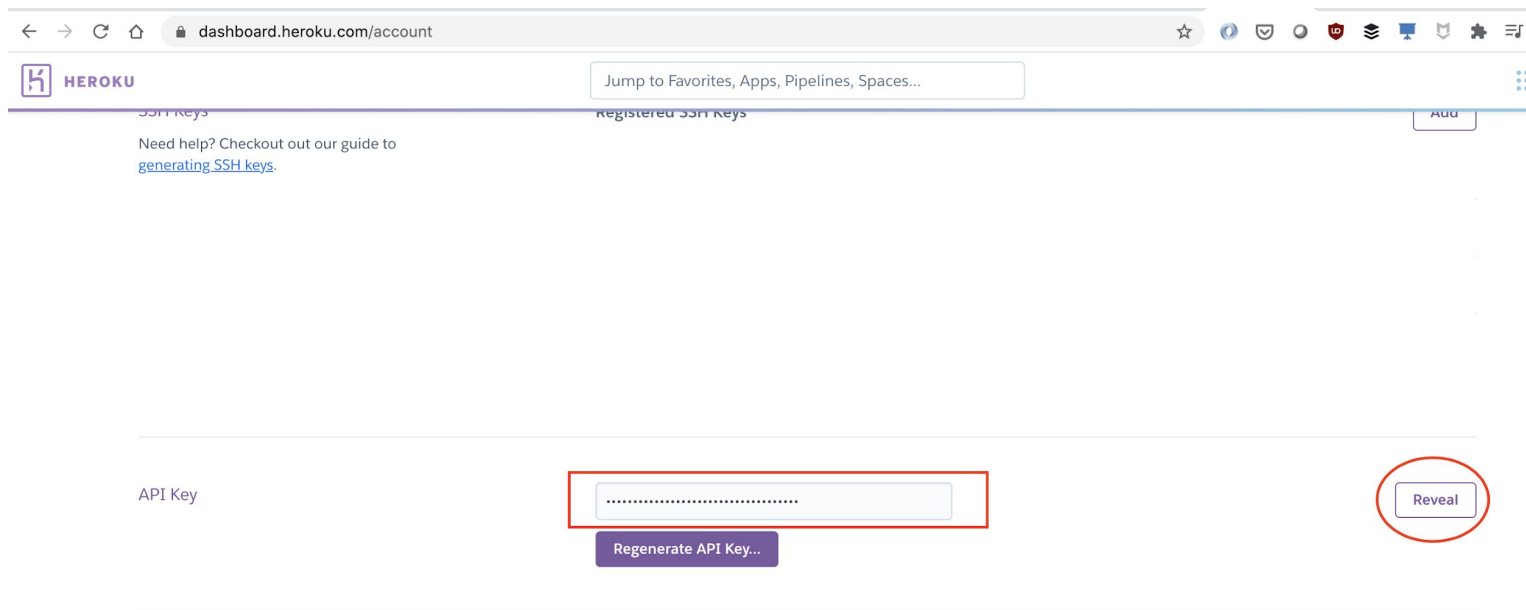
on:
  push:
    branches:
      - master

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: akhileshns/heroku-deploy@v3.5.7 # This is the action
        with:
          heroku_api_key: ${secrets.HEROKU_API_KEY}
          heroku_app_name: "YOUR APP's NAME" #Must be unique in Heroku
          heroku_email: "YOUR EMAIL"
```

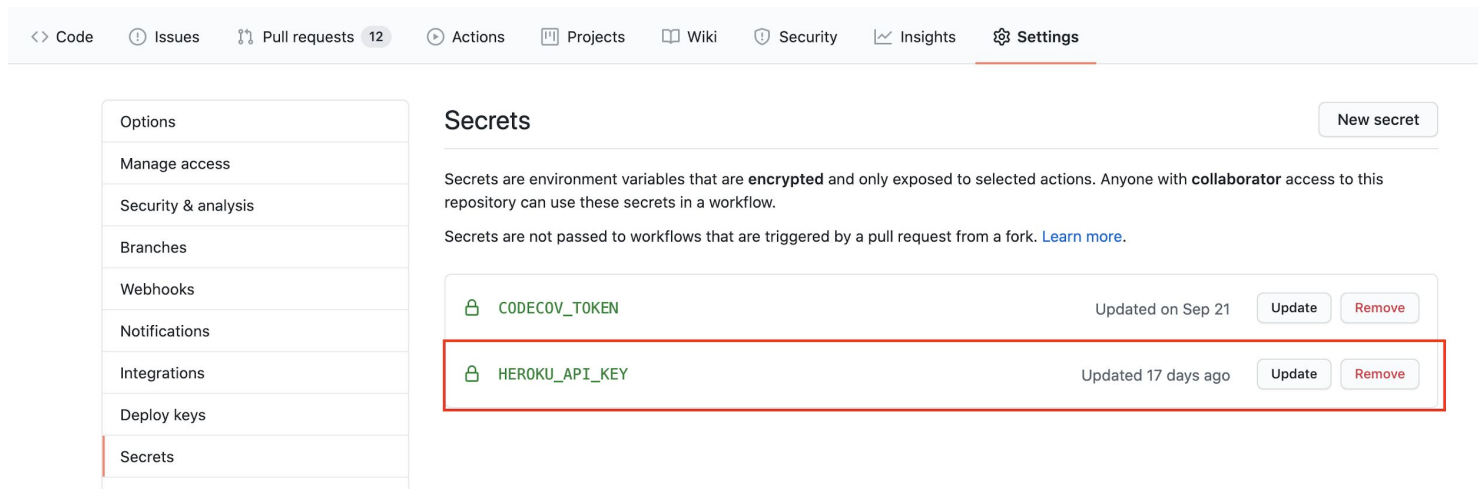
Agregar este secreto
en Github

Heroku API Key

En Account Settings, en **API KEY**, hacer click en reveal y copiar la API Key



Crear un secret **HEROKU_API_KEY** con la key que copiamos de Heroku.



The screenshot shows the GitHub interface for a repository's settings. The top navigation bar includes links for Code, Issues, Pull requests (12), Actions, Projects, Wiki, Security, Insights, and Settings (which is highlighted). On the left, a sidebar menu lists various settings categories: Options, Manage access, Security & analysis, Branches, Webhooks, Notifications, Integrations, Deploy keys, and Secrets (which is highlighted with a red border). The main content area is titled 'Secrets' and features a 'New secret' button. Below this, there is explanatory text about secrets being encrypted environment variables. A list of existing secrets is shown, with two entries: 'CODECOV_TOKEN' (updated on Sep 21) and 'HEROKU_API_KEY' (updated 17 days ago). The 'HEROKU_API_KEY' entry is highlighted with a red rectangular border. Each secret entry includes an 'Update' button and a 'Remove' button.

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Secrets

Secrets

New secret

Secrets are environment variables that are **encrypted** and only exposed to selected actions. Anyone with **collaborator** access to this repository can use these secrets in a workflow.

Secrets are not passed to workflows that are triggered by a pull request from a fork. [Learn more](#).

CODECOV_TOKEN	Updated on Sep 21	Update	Remove
HEROKU_API_KEY	Updated 17 days ago	Update	Remove

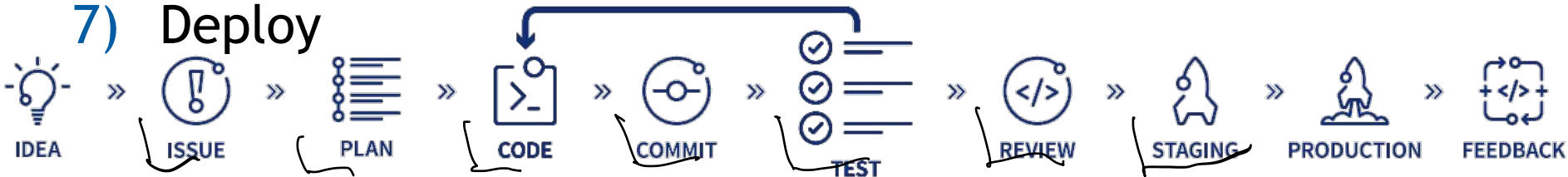
git push origin main

“build, test, and deploy”

225 results		Event ▾	Status ▾	Branch ▾	Actor ▾
✓	Configure GHA to deploy to Heroku (#46) [Closes #47] Static Code Analysis #60: Commit e6940de pushed by ignaciojonas	master		2 minutes ago 1m 27s	...
✓	Configure GHA to deploy to Heroku (#46) [Closes #47] Dusk Tests #42: Commit e6940de pushed by ignaciojonas	master		3 minutes ago 2m 29s	...
✓	Configure GHA to deploy to Heroku (#46) [Closes #47] Laravel #118: Commit e6940de pushed by Ignaciojonas	master		2 minutes ago 2m 12s	...
✓	Configure GHA to deploy to Heroku (#46) [Closes #47] Deploy #5: Commit e6940de pushed by ignaciojonas	master		2 minutes ago 1m 43s	...

Nuevo Requerimiento Cambiar el color de fondo de la aplicación

- 1) Crear issue
- 2) Planificar / Asignar
- 3) Implementar
- 4) Commit & Push
- 5) Correr tests
- 6) Merge Request
- 7) Deploy



Referencias

- <https://github.com/marketplace/actions/deploy-to-heroku>
- <https://hackernoon.com/has-devops-changed-the-role-of-a-tester-b140dddc7824>
- <https://woohuiaren.me/blog/how-to-laravel-testing-on-gitlab-ci-with-docker/>
- <https://mattstauffer.com/blog/laravel-on-heroku-using-a-postgresql-database/>
- <https://github.com/Unicen-Tupar/bomberos/wiki>
- <https://devcenter.heroku.com/articles/platform-api-reference#dyno>