

4-11-2018

Tarea 1

Not My Job

Arquitectura de Software

Profesor: Francisco Marshall

Ayudantes: Raúl García - Sergio Troncoso

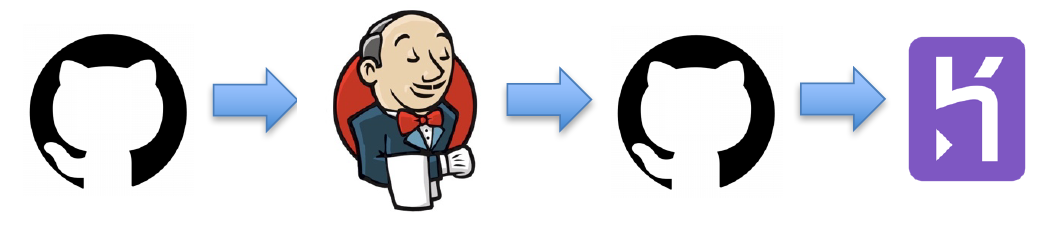
Gustavo Galvez – Boris Contreras – Francisco Alvares

# Introducción

Se debe crear un entorno de integración con las siguientes herramientas:

* GitHub: Sistema de control de versiones donde se alojará la aplicación
* Jenkins: Servidor de integración
* Heroku: Plataforma de servicios (PaaS) para el despliegue

El flujo de integración es el siguiente:



Cada vez que se realice un “push” en el repositorio github y se suba código, Jenkins detecte este cambio, compile la aplicación y realice las tareas correspondientes, si todo está correcto vuelva a hacer un “push” a github donde heroku desplegará.

# Procesos

1. Se realizo una clonación del repositorio https://github.com/gnarula/django-ribbit indicado en el enunciado, utilizando el comando git clone. El clon de este repositorio genero el siguiente repositorio: https://github.com/Gustavogalvez/Tarea-1

2. Se procedió a realizar un archivo dockerfile para crear una imagen en Docker, su contenido es el siguiente:

from python:2.7

workdir /ribbit

copy . /ribbit

run pip install -r requeriments.txt

run python manage.py syncdb --noinput

run python manage.py migrate ribbit\_app

<<<<<<< HEAD

run python manage.py runserver

=======

cmd python manage.py runserver 0.0.0.0:$PORT

>>>>>>> f54df45c56f27bf371a1895b67d3ce9916b36596

3. Se realizo el build del dockerfile, mientras paralelamente se creaba una cuenta en Heroku.

4. Se procedió a montar un servidor de Heroku, para poder comprobar su correcto funcionamiento.

5. Una vez comprobado su funcionamiento, se procedió a automatizar mediante el uso de Jenkins, generando un archivo Jenkinsfile.

6. El primer problema encontrado fue otorgar las credenciales correspondientes para git y heroku, lo que se solucionó entregando usuario y contraseña para el caso de git y el usuario y la contraseña entregadas por heroku en el archivo netrc para el caso de heroku.

7. Luego de esto se indicó que se debía actualizar siempre que se realizaran push dentro del git base, es decir realizar cambios dentro de la aplicación, especificando como destino el git de heroku para reflejar los cambios dentro de la aplicación web o eso debería realizar, pero no se pudo lograr.

# Conclusión

Integrar un entorno de integración no es tan fácil como parece, hacer que las distintas herramientas (versionado como desarrollo) es un tanto complicado, como por ejemplo al momento de comunicar el repositorio con Jenkins surgían problemas de autentificación, es decir que lo más complicado que se encontró es hacer que los entornos se puedan comunicar sin trabas. Docker soluciona los problemas de entorno.

Al final de todo es entendible que se generen integraciones continuas en las empresas que producen software, debido a que no se pueden detener por pequeños problemas y deben mantener estos al margen, estos procesos de integración ayudan a encontrar los problemas lo antes posible.

## Anexos

La aplicación de Heroku se encuentra en: https://tarea-1.herokuapp.com/

El repositorio Github se encuentra en: https://github.com/Gustavogalvez/Tarea-1

Dentro del repositorio se puede encontrar el archivo Dockerfile.