



Unidad 4 / Escenario 7

Lectura fundamental

Travis-CI, branch build flow y pull request build flow

Contenido

- 1 ¿Qué es?
- 2 Travis-ci.org Vs travis-ci.com
- 3 Ciclo de vida de un Build
- 4 ¿Qué podemos controlar?

Palabras clave: build, deploy, branch, pull request

1. ¿Qué es?

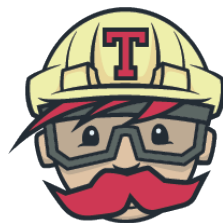
Travis CI es un proyecto de software libre enfocado en el uso de Integración Continua en proyectos Github, bajo la premisa “Build Apps with confidence (Crea aplicaciones con confianza)”. Esta herramienta compila sus aplicaciones y gestiones de repositorios de manera eficiente, permitiendo reunir herramientas en lenguajes como; Phyton, Ruby, Java, Php y muchos más.

La facilidad de integración en proyectos y repositorios github ha generado gran atracción en compañías tecnológicas, tales como Heroku, Zendesk y BitTorrent que utilizan esta tecnología.

¿Sabía qué...?



La división del proyecto entre varias ramas permite la recuperación de versiones en caso de ser necesario. Siempre es mejor, previo a migrar los cambios a la rama siguiente, tener seguridad del correcto funcionamiento.



Travis CI

Figura 1. Logo Travis CI

Fuente: Travis-ci (2017)

1.1. Características:

Así como Travis CI ayuda a la mejora de despliegue de pruebas y a una correcta gestión de las principales operaciones de los repositorios, cumple con algunas características:

1.1.1. Configurado en segundos:

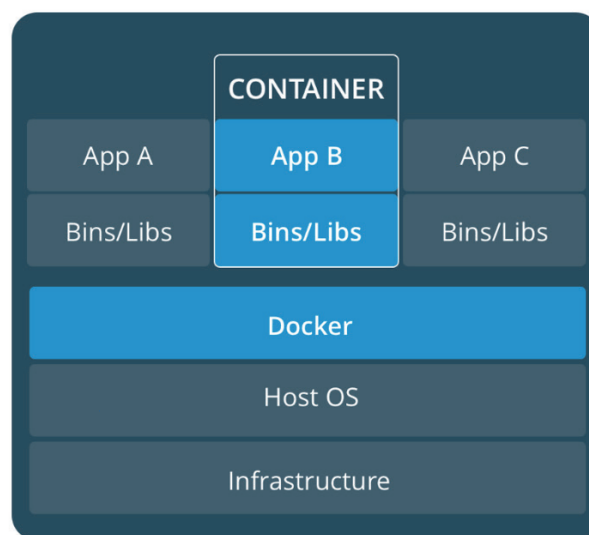
Con tres simples pasos es posible integrar esta herramienta a sus repositorios Github, tan sencillo como hacer el registro con su cuenta Github (Figura 2), seleccionar el proyecto que se quiere monitorear (Figura 3) y crear el archivo `.travis.yml` en la raíz del repositorio. De esta forma se monitorea cada proceso ejecutado en el repositorio seleccionado.

Author	Commit	Message	Date	Buils
EC2 Default U...	8f53845	last changes	2016-11-18	
EC2 Default U...	3a258eb	sept	2016-09-06	
Eduardo Oliver...	3298276	Merged develop into master	2016-05-11	
EC2 Default U...	dbcd67e	admin config	2016-05-11	
EC2 Default U...	cde8fa8	commit modify post	2016-05-10	
EC2 Default U...	2f2384e	5 de mayo	2016-05-05	
EC2 Default U...	885e1c8	fix facebook	2016-05-02	
EC2 Default U...	2ac7fa2	final commit before store	2016-05-01	
EC2 Default U...	3298276	changes in httpd config and chat	2016-04-29	
EC2 Default U...	31364d7	Initial commit	2016-04-27	

El archivo `.travis.yml` es un fichero YAML (Figura 4), que especificará la configuración de la integración con la herramienta, en este archivo se establece información como lenguaje utilizado, versión, ramas en las que se ejecutará, etc.

Figura 2. Pantallazo Registro Travis CI

Fuente: elaboración propia



máquina especial, es decir, en caso de que trabajes bajo el lenguaje de Objective C o Swift este usará una máquina basada en OS X.

Figura 3. Pantallazo Selección Proyecto Travis

Fuente: elaboración propia

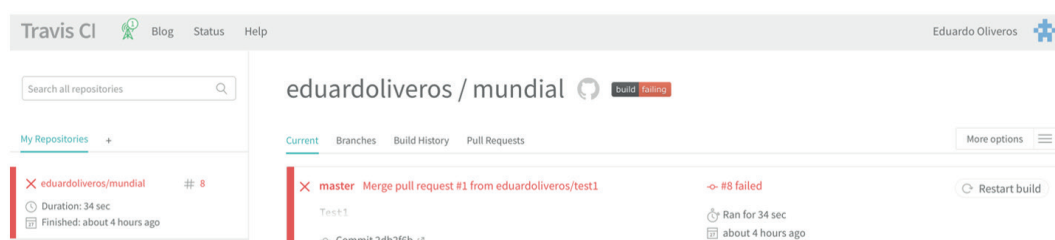


Figura 4. Pantallazo Documento `.travis.yml` para aplicación iOS

Fuente: elaboración propia

1.1.2. Soporte a múltiples plataformas:

Diferentes bases de datos y servicios son preinstalados en cada integración de Travis. Estos pueden ser activados en la configuración de la compilación, dando un gran apoyo a todos los lenguajes de programación.

1.1.3. Testeo de Pull Request:

Travis CI permite dar seguridad en cada aprobación de Pull Request, de esta forma cada equipo se asegura, previamente, de que la implementación de cambios y mejoras funcionen correctamente antes de la migración de producción.



1.1.4. Deploy en cualquier lugar:

Una de las características más destacadas de esta herramienta es la facilidad con la que puede pasar de su entorno de pruebas o desarrollo, a un entorno de producción, desde cualquier lugar.

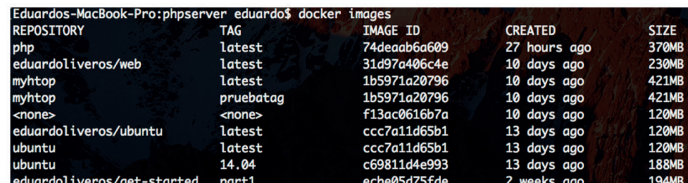
2. Travis-ci.org Vs travis-ci.com

Aunque originalmente Travis CI fue desarrollado como soporte a proyectos de código abierto, con el tiempo se amplió a proyectos privados, o de código cerrado, de esta forma Travis-ci.org se encarga de los proyectos libres y travis-ci.com se encarga de los proyectos privados, conocido como Travis Pro.

3. Ciclo de vida de un Build

En Travis CI cada build es realizado bajo dos escenarios:

- **Install:** destinado para la instalación de dependencias necesarias, como vemos el caso de la figura 5, en la que se reserva ese espacio para instalar hugo, framework para construir servidores web. (travis-ci, s.f.)

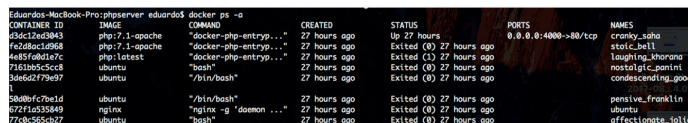


REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
php	latest	74deaab6a609	27 hours ago	370MB
eduardoliveros/web	latest	31d97a406c4e	10 days ago	230MB
myhtop	latest	1b5971a20796	10 days ago	421MB
myhtop	pruebatag	1b5971a20796	10 days ago	421MB
	<none>	f13ac0616b7a	10 days ago	120MB
eduardoliveros/ubuntu	latest	ccc7a11d65b1	13 days ago	120MB
ubuntu	latest	ccc7a11d65b1	13 days ago	120MB
ubuntu	14.04	c69811d4e993	13 days ago	188MB
eduardoliveros/get-started	part1	ecbe05d75fde	2 weeks ago	194MB

Figura 5. Pantallazo Estructura YAML para instalación de dependencias

Fuente: elaboración propia

- **Script:** Corre todos los scripts necesarios para la compilación del proyecto. La figura 6 muestra un ejemplo para el lenguaje Ruby. En este lenguaje se usa la palabra “rake” como comando de construcción por defecto. (travis-ci, s.f.)



CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
9a6c12a0d943	php:7.1-apache	"docker-php-entryp..."	27 hours ago	Up 27 hours	0.0.0.0:4000->80/tcp	cranky_saha
fe2d8c1d968	php:7.1-apache	"docker-php-entryp..."	27 hours ago	Exited (0) 27 hours ago		stolc_bell
4e85f0d1a7c	php:latest	"docker-php-entryp..."	27 hours ago	Exited (13) 27 hours ago		laughing_thorona
7133ba5c5c8	ubuntu	"bash"	27 hours ago	Exited (0) 27 hours ago		nostalgic_paniel
3a6e627f9e97	ubuntu	"/bin/bash"	27 hours ago	Exited (0) 27 hours ago		condescending_goadell
5b88fc7ba1d	ubuntu	"/bin/bash"	27 hours ago	Exited (0) 27 hours ago		pensive_franklin
672f1a535849	nginx	"nginx -g 'daemon ...'"	27 hours ago	Exited (0) 27 hours ago		ubuntu
77c0e355c27	ubuntu	"bash"	27 hours ago	Exited (0) 27 hours ago		affectionate_joliot

Figura 6. Pantallazo YAML para ejecución de scripts

Fuente: elaboración propia

En total, el ciclo de vida del “Build” contiene 11 pasos, en los que se pueden ejecutar operaciones de administración de dependencias y configuraciones de proyecto.

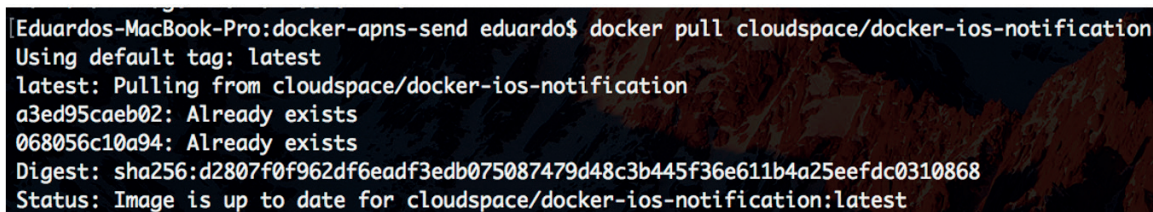
- apt addons (Opcional)
- cahce components
- before_install
- install
- before_script
- script
- before_cache (usado para limpiar el cache previo)
- after_success o after_failure
- before_deploy
- deploy
- after_deploy

4. ¿Qué podemos controlar?

Travis CI junto con la integración de Github, permite la administración de todos los repositorios que se quieran, principalmente se encuentran 4 secciones:

4.1. Monitorear actividad reciente

Dentro de la sección Current se ve la actividad más reciente en el repositorio. En este se muestra el último suceso, de todo el repositorio, ya sean nuevos cambios en una rama o *pull request* que requiere revisión. De igual forma, permite ver la información detallada de este proceso, como se observa en la figura 7.



```
Eduardos-MacBook-Pro:docker-apns-send eduardo$ docker pull cloudspace/docker-ios-notification
Using default tag: latest
latest: Pulling from cloudspace/docker-ios-notification
a3ed95caeb02: Already exists
068056c10a94: Already exists
Digest: sha256:d2807f0f962df6eadf3edb075087479d48c3b445f36e611b4a25eefdc0310868
Status: Image is up to date for cloudspace/docker-ios-notification:latest
```

Figura 7. Pantallazo Sección Current

Fuente: elaboración propia

4.2. Administración de ramas

Para la administración de ramas, Travis CI presenta dos secciones dentro de su panel de trabajo, la primera sección “Branches”, muestra un historial de cambios por cada rama presente dentro del repositorio, esta posee información como, nombre, fecha, encargado, número de cambios, entre otros.

En la sección de cambios, el cambio más reciente se encontrará al lado izquierdo, y el cambio más antiguo al lado derecho, teniendo en cuenta que solo se mostrarán 5 cambios por rama, como podemos observar en la figura 8.

```
Eduardos-MacBook-Pro:~ eduardo$ docker commit d3dc12ed3043 ubuntuDep
sha256:697c9a3ac08bd2e2a3b8aa371f2d7c7b7fe6356db59d7ae122c290f988169489
Eduardos-MacBook-Pro:~ eduardo$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntuDep	latest	697c9a3ac08b	13 seconds ago	465MB
php	latest	74deaab6a609	28 hours ago	370MB
eduardoliveros/web	latest	31d97a406c4e	10 days ago	230MB
myhtop	latest	1b5971a20796	10 days ago	421MB
myhtop	pruebatag	1b5971a20796	10 days ago	421MB
<none>	<none>	f13ac0616b7a	10 days ago	120MB
eduardoliveros/ubuntu	latest	ccc7a11d65b1	13 days ago	120MB
ubuntu	latest	ccc7a11d65b1	13 days ago	120MB
ubuntu	14.04	c69811d4e993	13 days ago	188MB
eduardoliveros/get-started	part1	ecbe05d75fde	2 weeks ago	194MB
friendlyhello	latest	ecbe05d75fde	2 weeks ago	194MB

Figura 8. Pantallazo Historial de actividad de las ramas

Fuente: elaboración propia

La segunda función para la administración y supervisión de ramas, es “Build History”, esta sección muestra un historial detallado de los cambios que se han realizado en el repositorio, teniendo como base la fecha de realización del cambio, como se muestra en la figura 9.

```
Eduardos-MacBook-Pro:phpserver eduardo$ docker rmi 1815c82652c0
Untagged: hello-world:latest
Untagged: hello-world@sha256:f3b3b28a45160805bb16542c9531888519430e9e6d6fffc09d72261b0d26ff74f
Deleted: sha256:1815c82652c03bfd8644afda26fb184f2ed891d921b20a0703b46768f9755c57
Deleted: sha256:45761469c965421a92a69cc50e92c01e0cfa94fe026cdd1233445ea00e96289a
```

Figura 9. Pantallazo Historial compilación de las ramas

Fuente: elaboración propia

4.3. Administración de Pull Request

Para la verificación de cada *Pull request*, Travis CI maneja una sección llamada “*Pull request*”, esta sección muestra el estado de cada operación generada, evitando así, la aprobación de estos, en caso de ser erróneos o no estar compilados.

```
Eduardos-MacBook-Pro:docker-apns-send eduardo$ docker run -it ubuntu:dep
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. See
t the 'ServerName' directive globally to suppress this message
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. See
t the 'ServerName' directive globally to suppress this message
[Thu Aug 24 19:30:38.341550 2017] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.10 (Debian) PHP/7.1.8 conf
igured -- resuming normal operations
[Thu Aug 24 19:30:38.341620 2017] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
^C[Thu Aug 24 19:30:59.136934 2017] [mpm_prefork:notice] [pid 1] AH00169: caught SIGTERM, shutting down
Eduardos-MacBook-Pro:docker-apns-send eduardo$ docker run -it ubuntu:dep bash
root@fa9e9429ab1d:/var/www/html#
```

Figura 10. Pantallazo Sección Pull Request

Fuente: elaboración propia

4.4. Construcción de Tiggers

Esta operación es una funcionalidad Beta por el momento, permite la modificación en tiempo real de nuestros archivos de configuración de la conexión con Travis CI, de esta forma, es posible ejecutar operaciones, dentro de ramas existentes para pruebas, sin necesidad de realizarlo directamente en el código.

```
Eduardos-MacBook-Pro:phpserver eduardo$ docker rm 77c0c565cb27
77c0c565cb27
```

Figura 11. Pantallazo Creacion Tigger Build

Fuente: elaboración propia

En síntesis...

En caso de que se manejen proyectos en Github, Travis CI es una excelente herramienta de integración continua que permite el monitoreo y planificación de procesos de actualización de repositorios y, de forma simple, la ejecución de pruebas de compatibilidad y funcionalidades, instalación de dependencias, entre otros.



Referencias bibliográficas

Laguiar. (2016, 5 8). wiki.genexus.com. Retrieved 7 6, 2017, from GeneXus Community Wiki: [https://wiki.genexus.com/commwiki/servlet/wiki?29683,Manejo+de+ambientes+\(Desarrollo%2C+Testeo%2C+PreProduccion%2C+Producci%C3%B3n\),](https://wiki.genexus.com/commwiki/servlet/wiki?29683,Manejo+de+ambientes+(Desarrollo%2C+Testeo%2C+PreProduccion%2C+Producci%C3%B3n),)

martinfowler. (2006, 05 01). martinfowler.com. Retrieved 07 02, 2017, from martinfowler.com: <https://martinfowler.com/articles/continuousIntegration.html>

Mateo, F. G. (2007, 06 21). Estructura manejo de repositorios trunk branch. Retrieved 08 23, 2017, from um.es: <https://www.um.es/atika/documentos/NORPuestaEnLineaAplicacionesWeb.pdf>

s.a. (n.d.). Documentacion Git. Retrieved 08 23, 17, from Git: <https://git-scm.com/doc>

travis-ci. (s.f.). <https://travis-ci.com>. Retrieved 08 28, 2017, from travis-ci: <https://docs.travis-ci.com>

Referencias de imágenes

Figura 1. Travis CI, GmbH (s.f.). Logo Travis. Recuperado de <https://travis-ci.com/logo>.

Texto aclaratorio

*En este material se han tomado logos de las marcas Travis con fines netamente educativos.

INFORMACIÓN TÉCNICA



FACULTAD DE
**INGENIERÍA, DISEÑO
E INNOVACIÓN**

Módulo: Énfasis Profesional I (Integración continua)

Unidad 4: Herramientas de despliegue automatizado

Escenario 7: Travis-CI, branch build flow y pull request build flow

Autor: Eduardo Enrique Oliveros Acosta

Asesor Pedagógico: Amparo Sastoque Romero

Diseñador Gráfico: Paola Andrea Melo

Asistente: Eveling Peñaranda

Este material pertenece al Politécnico Grancolombiano. Por ende, es de uso exclusivo de las Instituciones adscritas a la Red Ilumino. Prohibida su reproducción total o parcial.