

FEDERICO TOLEDO

[BLOG](#)[CHARLAS Y TALLERES](#)[MI LIBRO](#)[ABOUT ME](#)

Travis-CI para integración continua

Shares

Posted on [June 21, 2018](#) by [Federico](#)

En este post voy a compartir parte del [trabajo de los alumnos del curso de testing de la UCU en 2017](#), que expliqué el objetivo en [este post](#). Esta vez comparto la sección dedicada a la herramienta para orquestar la integración continua, llamada Travis-CI, y los pasos para comenzar a configurar un pipeline.

[Travis-CI](#) es un sistema de [Integración Continua](#), gratuita para proyectos Open Source y de pago para proyectos privados. Se integra sin problemas con GitHub y automáticamente ejecuta el pipeline definido en cada push o pull requests. Testea y builda aplicaciones escritas en Ruby, Node, Objective-C, Go, Java, C# y F#, entre otras (que corran en Linux).

Posee su propia "Deploy engine" la cual permite a los desarrolladores testear por completo sus aplicaciones para luego realizar exitosamente el deploy sin sorpresas. Travis-CI tiene un proveedor para Azure Web Apps, que permiten combinar GitHub y Travis-CI con Azure para realizar los deploys automáticos.

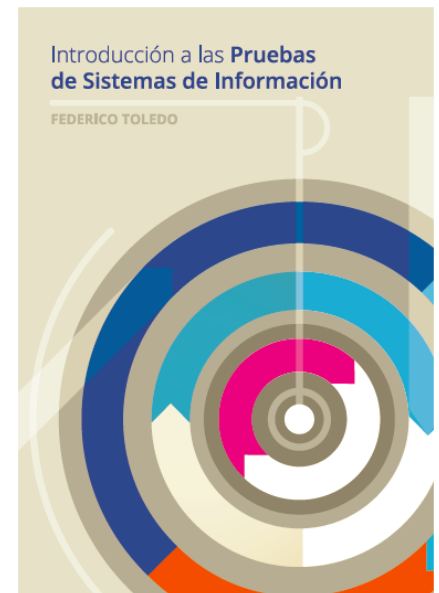
Si bien es similar a Jenkins, tiene sus diferencias. No utiliza tareas programadas, no tiene costos de mantenimiento (está hosteado como servicio), el setup es "simple", entre otras.

Cuenta con aproximadamente 300.000 proyectos y 235.00 usuarios probando los mismos. Estos números hablan sobre la importancia de

Search

[Mi libro de testing](#)

En el siguiente link encontrarás para descargar gratis mi libro "[Introducción a las pruebas de sistemas de información](#)"



Posts relacionados:

la herramienta y lo robusta que es. Agregado a esto, algunas de las compañías más importantes que utilizan Travis son Facebook, Heroku, Mozilla, Rails, Twitter, Zendesk, Slack.

La documentación es bastante completa. En la página oficial se puede encontrar documentación paso a paso con variados lenguajes. Además, Travis-CI cuenta con una comunidad activa muy grande, por ende encontrar soluciones a problemáticas es bastante común mediante la búsqueda en Google.

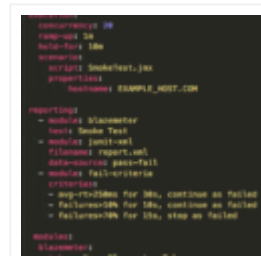
Shares

Requerimientos

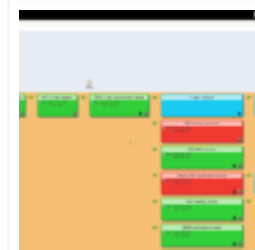
Para la tarea fue necesario contar con un repositorio GitHub que contenga el código de OpenCart (el sistema bajo pruebas) y con una cuenta en Travis-CI y en Azure (se utilizaron las opciones gratuitas).

Pasos a seguir

- El primer paso es crear un repositorio GitHub que contenga el código de la aplicación a la cual queremos incorporar en Travis-CI.
- Una vez el repositorio es creado, debemos dar permisos a Travis para conectarse con el mismo. Para esto debemos ir a las configuraciones del repositorio, hacer click sobre "Servicios" y seleccionar Travis-CI. Una vez allí seleccionar "Add to GitHub".



Cómo usar
Taurus y Jenkins



Pruebas de
Performance en



Pruebas de
Accesibilidad en

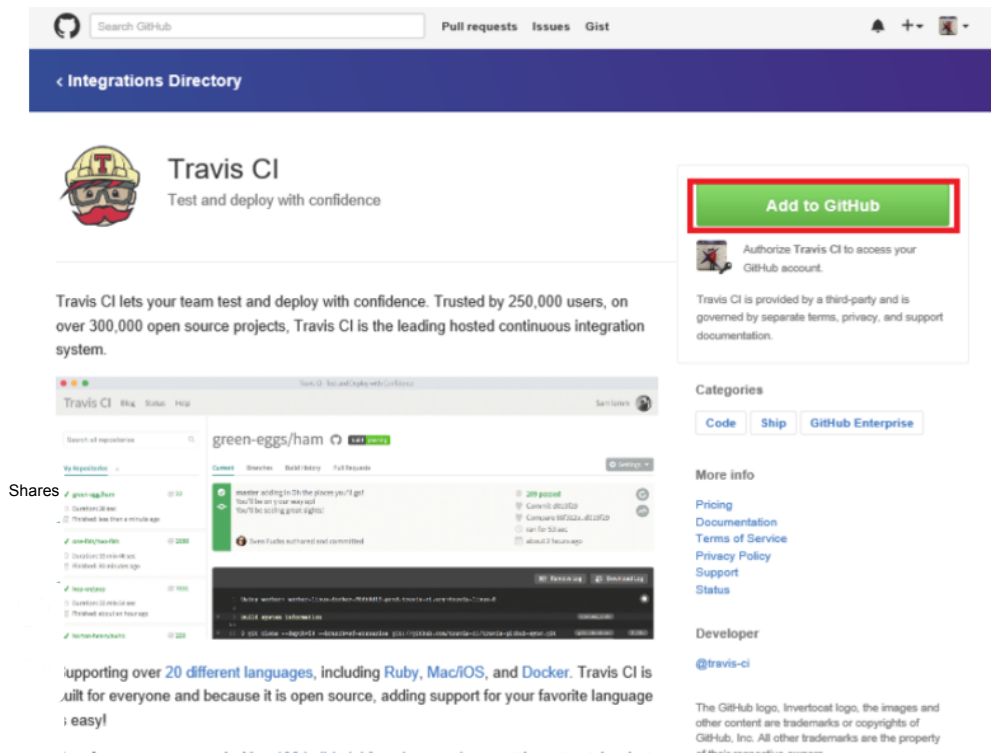


Alternativas para
ejecutar

Tweets recientes



RT @morvader:
The only test coverage percentage you should be worry about is 0% , which



Travis CI lets your team test and deploy with confidence. Trusted by 250,000 users, on over 300,000 open source projects, Travis CI is the leading hosted continuous integration system.

Supporting over 20 different languages, including Ruby, MacOS, and Docker. Travis CI is built for everyone and because it is open source, adding support for your favorite language is easy!

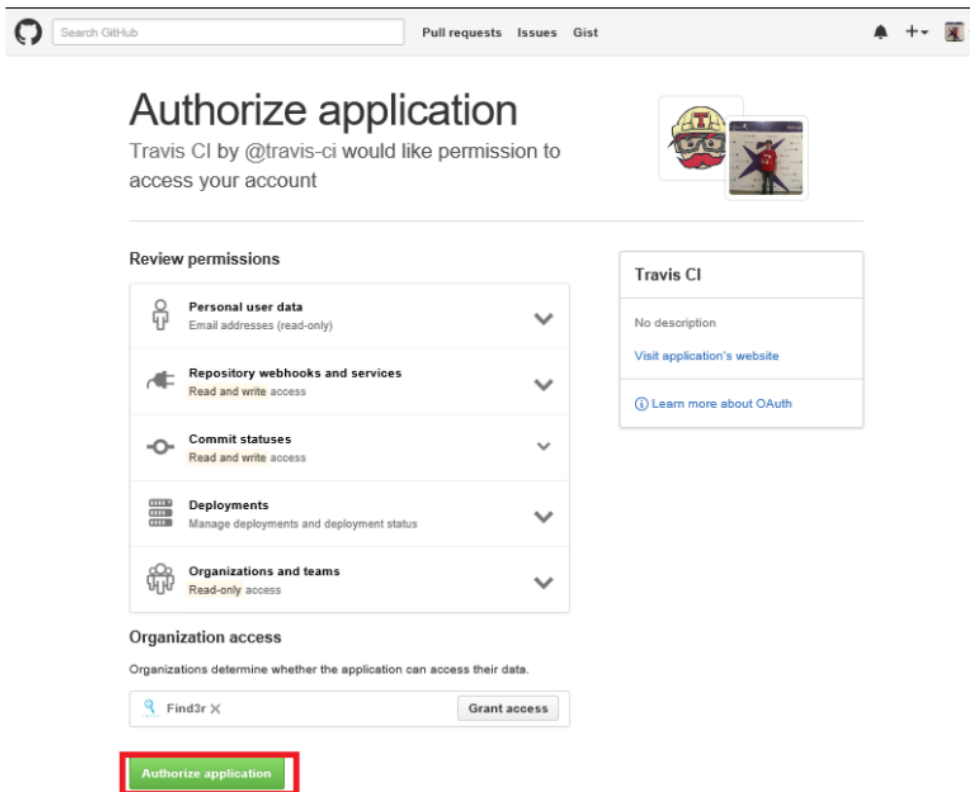
Categories: Code, Ship, GitHub Enterprise

More info: Pricing, Documentation, Terms of Service, Privacy Policy, Support, Status

Developer: @travis-ci

The GitHub logo, Inverto logo, the images and other content are trademarks or copyrights of GitHub, Inc. All other trademarks are the property of their respective owners.

Luego de esto, en la próxima página debemos otorgar todos los permisos requeridos a Travis-CI.



Authorize application

Travis CI by @travis-ci would like permission to access your account

Review permissions

- Personal user data**
Email addresses (read-only)
- Repository webhooks and services**
Read and write access
- Commit statuses**
Read and write access
- Deployments**
Manage deployments and deployment status
- Organizations and teams**
Read-only access

Organization access

Organizations determine whether the application can access their data.

Find3r X Grant access

Authorize application

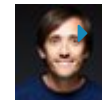
- Una vez en Travis-CI, seleccionar el "+", buscar el repositorio sobre el cual queremos trabajar y hacer click sobre el switch.

means you don't have any tests.

Any numbe...

<https://t.co/TnjBloXdqx>

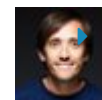
2 days ago



RT @SrPerf: Y sale mi primer post de JMeter, espro les guste y ayude a los migrantes de LR a JM....

<https://t.co/ihmJ7NyUoe>

4 days ago



@DiegoGavilanes6 Referilos también a @NahualUY

donde siempre decimos, ayudamos a los que tengan ganas de salir adel...

<https://t.co/1xvi6kzrzs>

5 days ago



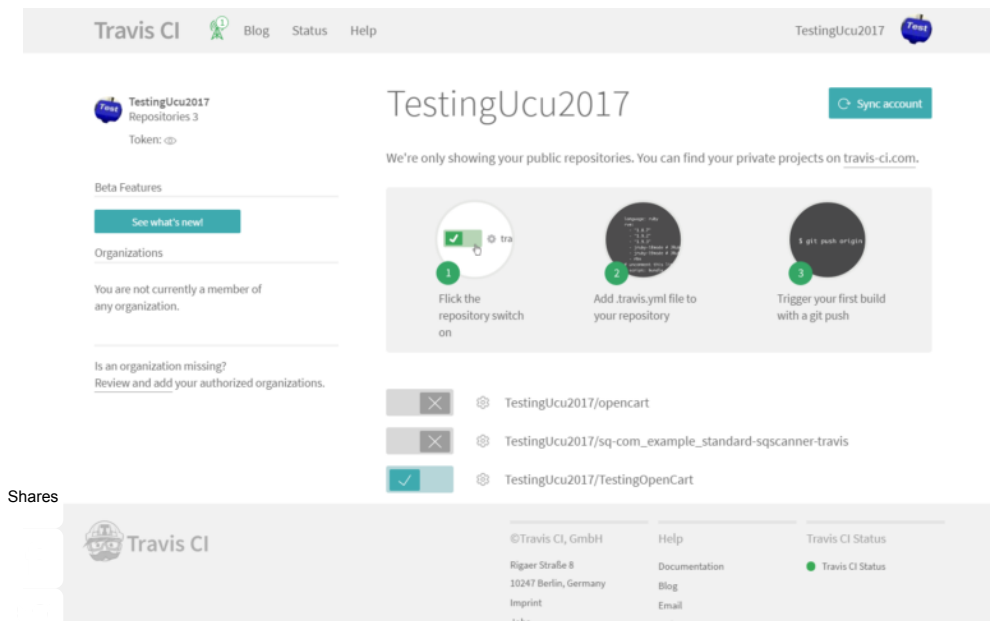
Nice to see you again

@TristanLombard2!

5 days ago

Seguir a @ftoledo { 2.722 seguidores }

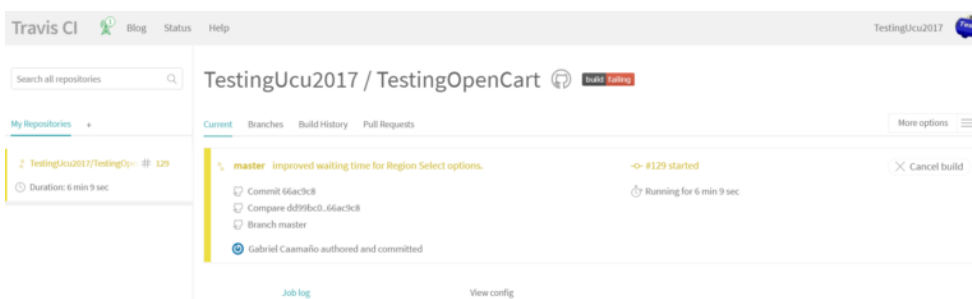
Categorías



Volver al repositorio GitHub y crear el archivo **.travis.yml**. Este archivo debe estar alojado en la raíz del proyecto. Le indicará a Travis-CI lo que debe realizar cada vez que ejecuta un Build. En el archivo incluir al menos las siguientes líneas:

```
Language: java
Jdk: oraclejdk8
```

- Estas líneas indican el lenguaje de programación y la versión de la JDK que se debe utilizar para hacer el build (en este caso la aplicación lo necesita).
- Luego de este paso hacemos el commit correspondiente. En el momento que sea realiza el commit, Travis-CI va a ejecutar el Build, por lo que si abrimos la pestaña veremos que hay una tarea corriendo.



- ▶ Agile (66)
- ▶ Automation (41)
- ▶ Blog (299)
- ▶ Blogs recomendados (3)
- ▶ Bug log (4)
- ▶ Charlas (63)
- ▶ Cultura (14)
- ▶ Desarrollo (17)
- ▶ DevOps y CI/CD (62)
- ▶ Dinámicas (12)
- ▶ eLearning (13)
- ▶ Escritor invitado (47)
- ▶ Eventos (1)
- ▶ Gatling (10)
- ▶ GeneXus (16)
- ▶ Github (2)
- ▶ Introducción al testing (32)
- ▶ JMeter (13)
- ▶ Liderazgo (12)
- ▶ Mindmaps (6)
- ▶ Mis quotes favoritas (4)
- ▶ Mobile (3)
- ▶ Monitoring (14)
- ▶ Nahual y RSE (17)
- ▶ Off-topic (31)

- En caso de que el Build sea satisfactorio, Travis-CI presentará los resultados en verde, de lo contrario los mismos se verán rojos. En cualquiera de los dos casos, hay una pequeña descripción.

TestingKsu017/TestingCp #129
Duration: 5 min 15 sec

Shares

| | | | | |
|----------|--|----------------|---------------|---------|
| ✓ master | Improved waiting time for Region Select options. | → #129 started | 5 min 15 sec | 66ac9c8 |
| ✗ master | Improved page loading wait. | → #128 failed | 10 min 14 sec | d899bc0 |
| ✗ master | Increased user-simulation interval a little. | → #127 failed | 10 min 43 sec | 35f15d2 |
| ✗ master | changed login test case to register test case. | → #126 failed | 10 min 54 sec | 40480ce |
| ✗ master | Update .travis.yml | → #125 failed | 6 min 6 sec | 7b61a3f |
| ! master | added assertions to test cases. | → #124 errored | 2 min 58 sec | 50d0364 |
| ! master | Update .travis.yml | → #123 errored | 3 min 1 sec | fc7535b |
| ✓ master | Update .travis.yml | → #122 passed | 2 min 59 sec | 2fda5bb |
| ✓ master | Update .travis.yml | → #121 passed | 3 min 4 sec | bd1e3ff |

Estos han sido los pasos mínimos para configurar Travis-CI. Se pueden agregar muchas tareas más como notificaciones por email, por Slack, correr otros tipos de pruebas, hacer build con varios SDK distintos, entre otras. Todo esto es configurable mediante el archivo .travis.yml y en la documentación oficial explica todo lo que podemos hacer (<https://docs.travis-ci.com/>).

- Performance (70)
- Scrum (18)
- Seguridad (1)
- Selenium (4)
- Softskills (16)
- Software testing guru (4)
- Testing (80)
- Testing Exploratorio (21)
- TestingUY (20)
- TesTips (2)
- Trabajo (12)
- Usabilidad y Accesibilidad (9)

Mes a mes

- December 2019 (1)
- November 2019 (4)
- October 2019 (4)
- September 2019 (3)
- August 2019 (3)
- July 2019 (6)
- June 2019 (4)
- May 2019 (7)
- April 2019 (4)
- March 2019 (2)
- February 2019 (4)

Posts Relacionados:



Cómo usar Taurus y Jenkins



Pruebas de Performance en



Pruebas de Accesibilidad en



Alternativas para ejecutar

Leave a Reply

Comment

Shares

Name *

Email *

Website

1

I'm not a robot

reCAPTCHA
Privacy - Terms

POST COMMENT

- ▶ December 2018 (9)
- ▶ November 2018 (11)
- ▶ October 2018 (9)
- ▶ September 2018 (10)
- ▶ August 2018 (7)
- ▶ July 2018 (15)
- ▶ June 2018 (9)
- ▶ May 2018 (7)
- ▶ April 2018 (15)
- ▶ March 2018 (10)
- ▶ February 2018 (7)
- ▶ January 2018 (10)
- ▶ December 2017 (6)
- ▶ November 2017 (10)
- ▶ October 2017 (10)
- ▶ September 2017 (15)
- ▶ August 2017 (10)
- ▶ July 2017 (11)
- ▶ June 2017 (10)
- ▶ May 2017 (8)
- ▶ April 2017 (10)
- ▶ March 2017 (8)
- ▶ February 2017 (2)
- ▶ January 2017 (4)
- ▶ December 2016 (11)

▶ [November 2016 \(12\)](#)

▶ [October 2016 \(10\)](#)

▶ [September 2016 \(9\)](#)

[Blog](#) [Charlas, talleres y webinars](#) [Mi libro](#) [About me](#) [Aviso Legal](#)

Shares

Copyright © 2019 Federico Toledo . Powered by WordPress | Theme: AccessPress Ray

