Lessons learned building a tool for developers.
Join now for free

GET STARTED >

OR

Australia

[Select State]

BROWSE MEMBERS >

Over 450,000 members and growing... join today
WE SHAPE YOUR IDEAS AND BRING THEM TO LIFE
WE BUILD AND DESIGN BEAUTIFUL MOBILE AND WEB APPLICATIONS
Mario Visic
Señor Developer
Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.

- Martin Fowler (the smartest guy alive)
rsanheim opened this pull request 6 days ago

Replicate user

No one is assigned

WIP ... /cc @rtomayko

- Good to merge — Build #84854 succeeded in 848s (Details)

1 participant
Note: Travis is amazing and I love them. You should use them.
Repositories
Public and private

This application will be able to **read and write all public and private repo data**. This includes the following:

- Code
- Issues
- Pull requests
- Wikis
- Settings
- Webhooks and services
- Deploy keys

[Learn more](#)

[Authorize application](#)
Repositories
Public and private

This application will be able to read and write all public and private repo data. This includes the following:

- Code
- Issues
- Pull requests
- Wikis
- Settings
- Webhooks and services
- Deploy keys

Learn more

Authorize application
* Your Application Name

Keith Pitt

* Heroku API Key

a4fa34fa34fa34fa34fa34fa34notrealobviously

You can get your Heroku API Key here

more options
Create an Amazing Workflow.

Semaphore is a fully managed, high performance testing and deployment solution for your company.

Start Your FREE Trial
You'll be up and running in a minute!

Enter your email address
Password

Start 30 Day FREE Trial

By signing up you agree to our terms of service and privacy policy.
Repositories

Public and private

This application will be able to **read and write all public and private repo data**. This includes the following:

- Code
- Issues
- Pull requests
- Wikis
- Settings
- Webhooks and services
- Deploy keys

Learn more

Authorize application
- Ugly
- Java? Lolwut?
- Too hard to setup
- GitHub Pull Requests Status
Ugly

Java? Lolwut?

Too hard to setup

GitHub Pull Requests Status
1 sudo apt-get install ruby
2 git clone git@github.com:keithpitt/app.git
3 cd app
4 rake
A semi-hosted continuous integration service
Updated Copyright information in README

User: healthpitt

Completed 14 days ago

Output:

```
Resolving dependencies...
Using rake (18.1.0)
Using timers (1.1.0)
Using celluloid (0.15.2)
Using ffl (2.5.3)
Using childprocess (0.4.0)
Using mime-types (2.0)
Using multipart-post (2.0.0)
Using at (2.5.4)
Using buildbox (0.6) from source at .
Using safe_yaml (0.9.7)
Using crack (9.4.1)
Using diff-tcs (1.2.5)
Using docile (1.1.0)
Using lockfile (2.1.0)
Using multi_json (1.8.7)
Using rspec-core (2.14.7)
Using rspec-expectations (2.14.4)
Using rspec-mocks (2.14.4)
Using rspec (2.14.1)
Using simplecov-html (0.8.8)
Using simplecov (0.8.1)
Using webmock (1.15.0)
```
✓ Not Ugly
✓ No Java
✓ Easy to setup
✓ GitHub Pull Request Status
✓ Can do what ever I want on the server
✓ Safe and secure production deploys
Review permissions

- **Personal user data**
  - Email addresses (read-only)

- **Commit statuses**
  - Read and write access

Authorize application
Alright listen up.
Challenges

Rendering terminal output
The agent is a memory hog
Having to do all the things
Do as little work as possible

All the gems (73 in Buildbox)
CSS framework (Twitter Bootstrap)
Deploy onto Heroku/Ninefold
Keep your vision strong

Don’t forget the goals
Don’t go off on tangents
Why are you doing this again?
Build what you want to exist
Keep the tech stack simple

Don’t do SOA
Don’t **scale** to your 1 customer
Keep the code **ghetto**
Use what you know
Avoid rewrites

Ignore all the new shiny things
Just slows you down
Founder code is encouraged
Don’t write invoicing stuff

Invoices are a must for B2B
Generally receipts don’t cut it
It’s such a buzz kill
chargebee.com + Pin Payments
Launch on the first day

After your 10th commit, be in production
Tease out any issues
Get people looking at it
Users can come later
Continuous Delivery
Regrets

Didn’t blog enough
Not enough drip marketing
Didn’t make a video on public launch
Took stuff personally
Future

Hosted Agents (but still secure)

Pipelines

Rewrite agent in Golang

Better IA
That’s it. I have stickers.