Travis CI

Continuous Integration

Oskar Hollmann

User Technologies

26.5.2016

Have you met Mr. Travis?

- Travis is a Continuous Integration platform,
- free for open-source projects,
- developed in Berlin, Germany since 2011,
- used by prominent projects such as Node.js or Ruby on Rails.

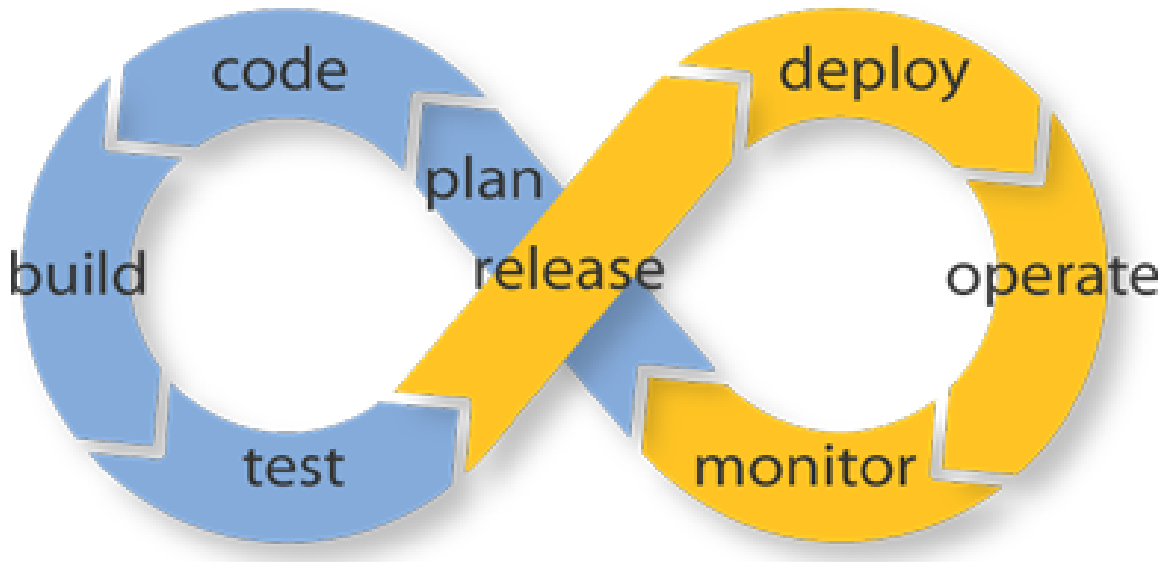


Figure: Continuous Integration diagram [360logica.com].

What can Travis do for you?

1. Automatically build Git repositories or pull requests.
2. Run all tests.
3. Integrate code coverage tools and code quality checks.
4. Deploy the code on AWS, Heroku, PyPi, RubyGems, ...

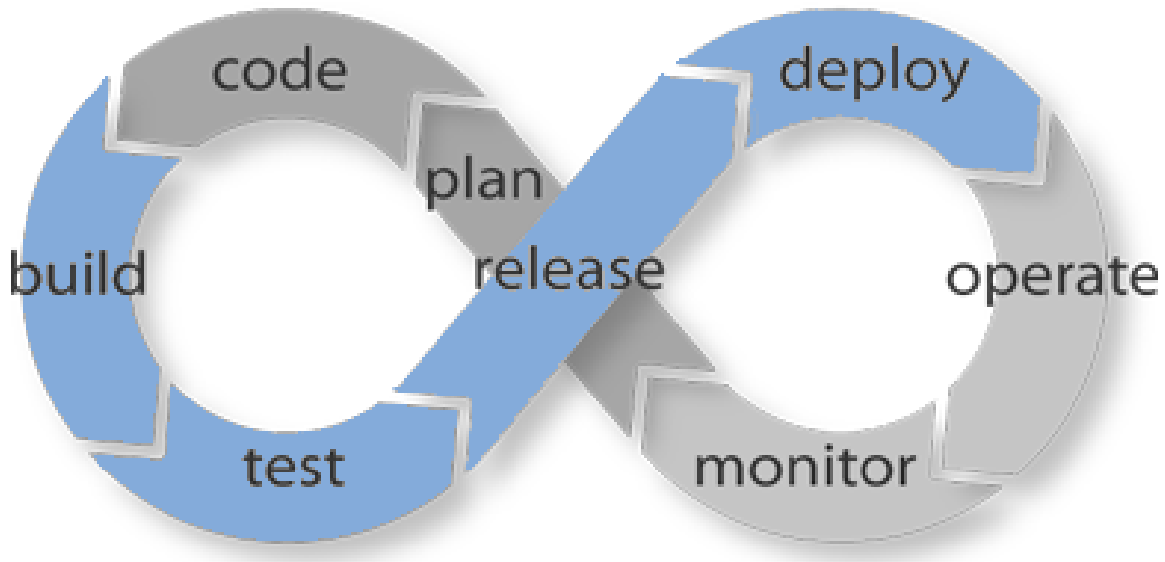


Figure: Continuous Integration diagram

1. Detection of pull request

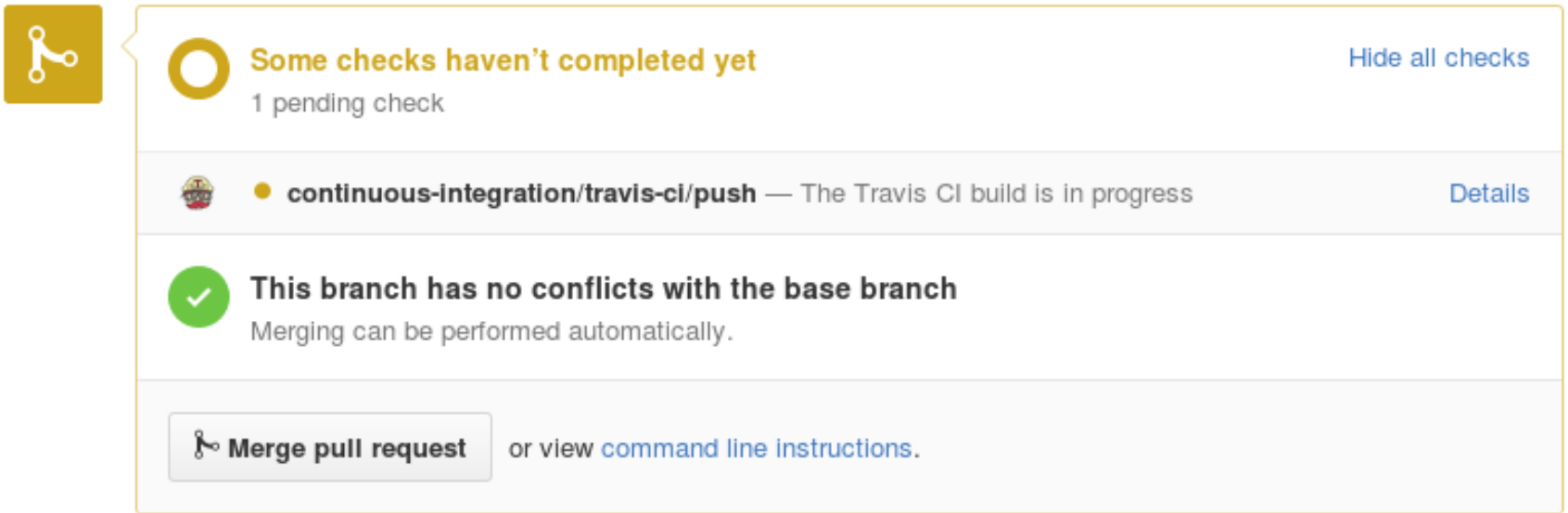


Figure: Travis CI integrates nicely with GitHub.

2. Test runs

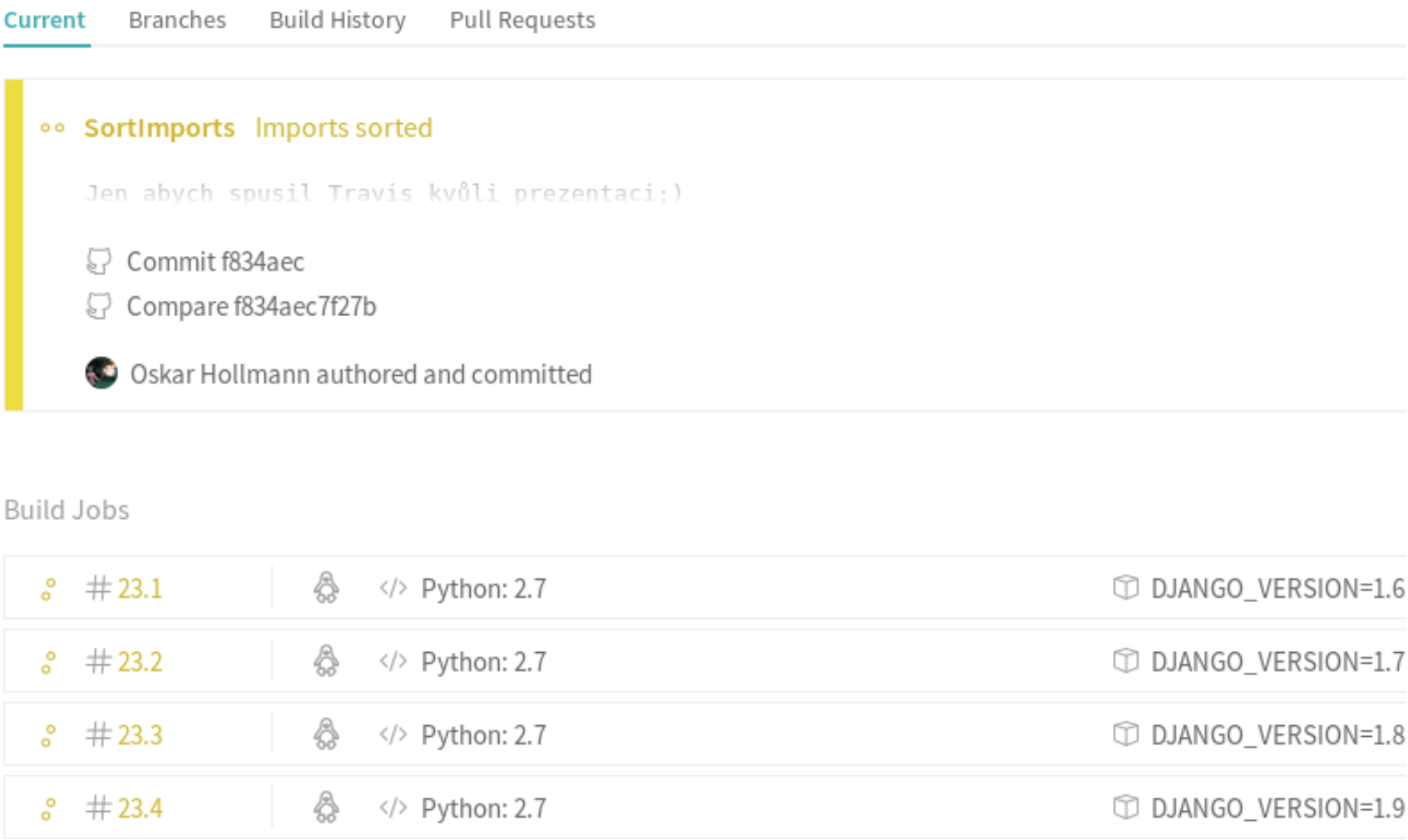
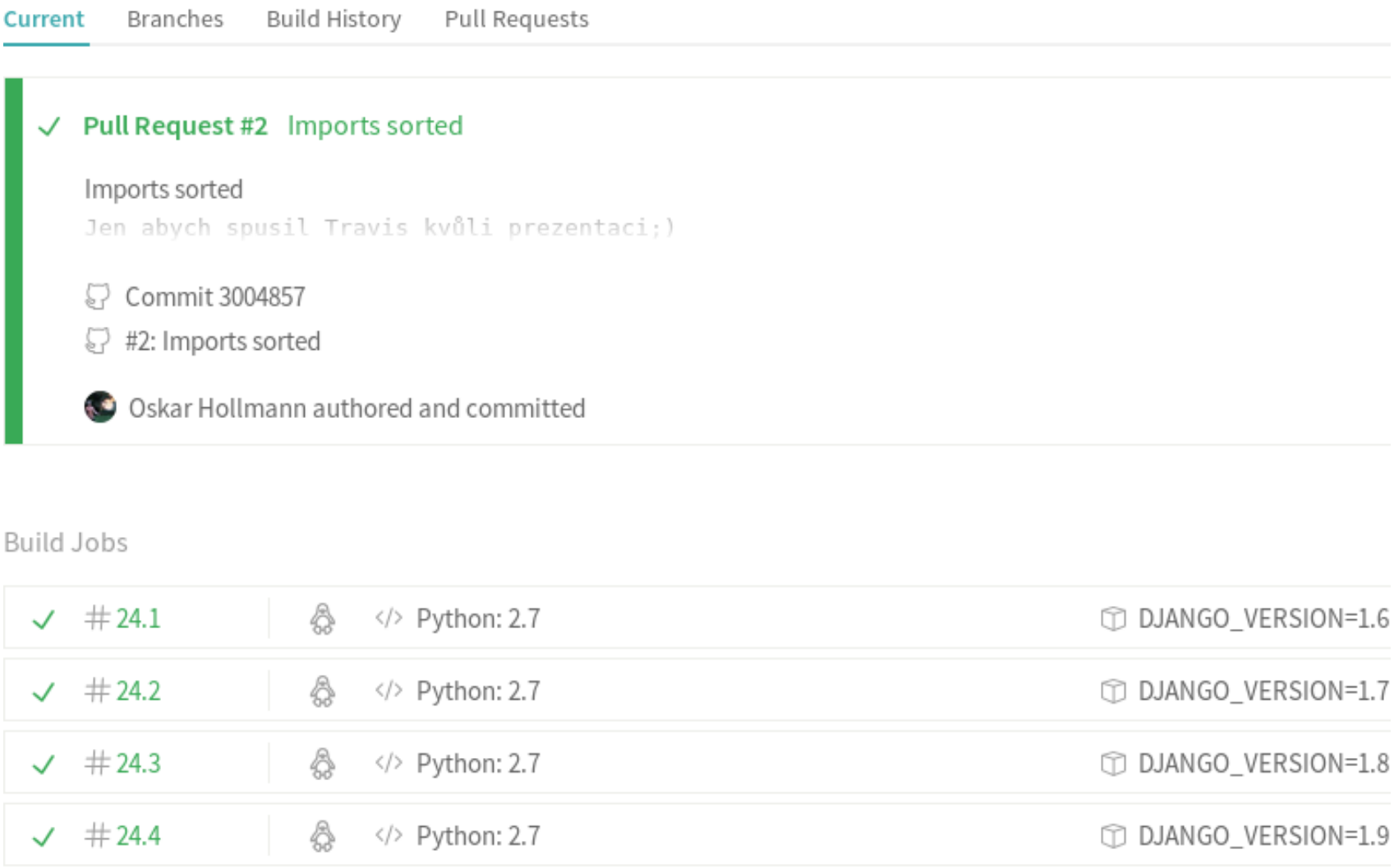


Figure: Tests are run using chosen versions of languages and libraries.

2. Test runs


Current
Branches
Build History
Pull Requests
Pull Request #2 Imports sorted
Imports sorted
Jen abych spusil Travis kvůli prezentaci;)
Commit 3004857
#2: Imports sorted
Oskar Hollmann authored and committed
Build Jobs
#24.1 Python: 2.7 DJANGO_VERSION=1.6
#24.2 Python: 2.7 DJANGO_VERSION=1.7
#24.3 Python: 2.7 DJANGO_VERSION=1.8
#24.4 Python: 2.7 DJANGO_VERSION=1.9

3. Report results

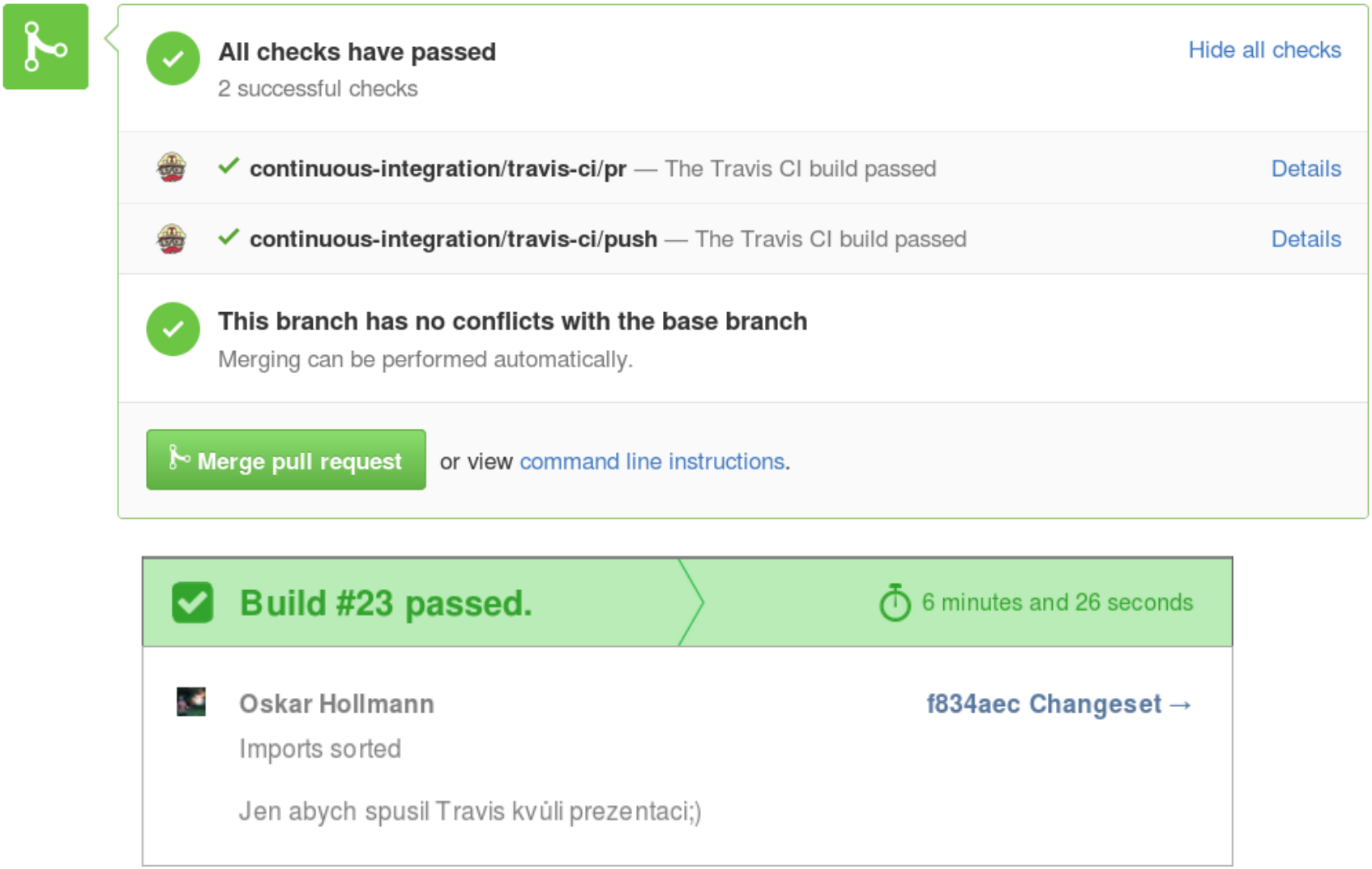


Figure: Results are pushed to Github and e-mail.

Configuration

- Tests run in a sandboxed Ubuntu Docker image.
- Configuration contained in a single **.travis.yml** file.
- Many stages of the test ran can be customised:
  - `addons`
  - `before_install`
  - `install`
  - `before_script`
  - `script`
  - `after_success/after_failure`
  - `before_deploy/deploy/after_deploy`
  - `after_script`

```
addons:
  apt:
    packages:
    - cmake
```

Example Configuration File

```
language: python

# Which version of the language to support?
python:
  - "2.7"
  - "3.5"

# Enviroment variables can be used to setup builds
# and test multiple versions of libraries
env:
  - DJANGO_VERSION=1.7
  - DJANGO_VERSION=1.8
  - DJANGO_VERSION=1.9
```

Example Configuration File

```
# All combinations of env. vars are tested
matrix:
  exclude: # But some may not be supported
    - python: "3.3"
      env: DJANGO_VERSION=1.9

# Install required dependencies
install:
  - cd example
  - pip install -r requirements/base.txt
  - pip uninstall -y -q django
  - pip install -q Django==$DJANGO_VERSION

# We can override the default build step
script:
   coverage run manage.py test app.tests
```

Thank you for your attention.

Any questions?