Travis-CI
Introduction
Slavek Kabrda
Continuous Integration

- Merging all working copies in main repo continuously (several times per day)
- Originated as a practice in eXtreme Programming
- Running test suite periodically/after every commit was added later to CI concept
- Nowadays used even without adopting XP
Benefits

● (Integration) bugs detected early
● Easy to track commits introducing bugs
● Availability of per-commit builds
Costs

- Writing and maintaining a test suite
- Setting up CI system
- Automating the CI process
What is Travis CI?

- [https://travis-ci.org/](https://travis-ci.org/)
- Hosted continuous integration service
- Only usable for GitHub hosted projects
  - [https://travis-ci.com/](https://travis-ci.com/) for private projects
- Documentation at [http://docs.travis-ci.com/](http://docs.travis-ci.com/)
How to Get it Working

1. Sign in (with your GitHub account)
2. Activate a webhook
3. Add ".travis.yml"
4. Push
How it Works

1. Developer pushes new commit(s)
2. GitHub let’s Travis-CI know that a new commit was pushed (PR was made)
3. Travis-CI checks out the GitHub repo
4. Travis-CI reads ".travis.yml", spins up VMs and runs tests
5. Travis-CI reports success/failure
6. (Travis-CI deploys)
.travis.yml - Content

- Specifies:
  - Language of your project (just one)
  - Language runtimes (JDK versions/Python versions/…)
  - Additional dependencies to install
  - Additional configuration and services
  - How to run tests
  - How to deploy
  - Many language-specific options
.travis.yml - Example

- (lots of language specific examples at http://docs.travis-ci.com/user/getting-started/)

language: python
python:
- "2.6"
- "2.7"
- "3.3"
- "3.4"
- "pypy"
install:
- pip install -r requirements.txt
- if [[ $TRAVIS_PYTHON_VERSION == '2.6' ]]; then pip install argparse; fi
script: nosetests
.travis.yml - Run Order

- before_install
- install
- before_script
- script
- after_success/after_failure
- after_script
.travis.yml - Dependencies

- http://docs.travis-ci.com/user/installing-dependencies/
- Interpreters/runtimes - installed by Travis-CI itself
- Services - installed, not necessarily enabled
- Libraries/extension modules - should be installed by language-specific means (PyPI/Rubygems/NPM/…)
- Can “apt-get” install Ubuntu packages
.travis.yml - Services

- http://docs.travis-ci.com/user/database-setup/
- MySQL, PostgreSQL, MongoDB, Redis, RabbitMQ, Memcached, ...

services:
  - redis
  - memcached

  # mysql starts on VM boot, no need to add it to services

before_script:
  # username 'travis'/'root', blank password
  - mysql -e 'create database myapp_test;'
.travis.yml - Deployment

- http://docs.travis-ci.com/user/deployment/
- OpenShift, Heroku, ...
- PyPI, Rubygems, ...

deploy:
  provider: openshift
  user: "YOU USER NAME"
  password: "YOUR PASSWORD" # can be encrypted
  domain: "YOUR OPENSHIFT DOMAIN"
.travis.yml - Notifications

- [http://docs.travis-ci.com/user.notifications/](http://docs.travis-ci.com/user/notifications/)
- Mail, IRC, custom webhook, ...
- Default: mail on every failure and when state changes from failure to success

```
notifications:
  irc:
    channels:
      - "chat.freenode.net#my-channel"
      - "chat.freenode.net#some-other-channel"
  on_success: [always|never|change] # default: always
  on_failure: [always|never|change] # default: always
```
language: ruby
rvm:
  - 2.0.0
  - 2.1.0
env:
  - DB=mongodb
  - DB=mysql
gemfile:
  - Gemfile
  - gemfiles/rails4.gemfile
language: python
matrix:
   include:
       - python: "2.7"
         env: TEST_SUITE=suite_2_7
       - python: "3.3"
         env: TEST_SUITE=suite_3_3
       - python: "pypy"
         env: TEST_SUITE=suite_pypy
script: ./test.py $TEST_SUITE
Encrypting sensitive data
Caching dependencies (paid version only)
OS X testing (new projects not accepted, capacity full for now)
GUI and headless browser testing
Command Line Client

- [https://github.com/travis-ci/travis.rb#readme](https://github.com/travis-ci/travis.rb#readme)

sudo dnf install rubygems
gem install travis
export PATH=$PATH:~/bin/

travis lint myproject/.travis.yml
Nice Examples

  - Complex Ruby testing setup with matrices, services and notifications
  - GTK+ GUI testing
- [https://github.com/travis-ci-examples](https://github.com/travis-ci-examples)
  - Various per-language examples by Travis-CI upstream
● CI is great
● CI is great
● Travis is a nice CI implementation and it’s for free (as in free beer)
● Did I say that CI is great?
Thanks!

Q&A