Python automated testing 101

by Roman Podoliaka (roman.podoliaka@gmail.com (mailto:roman.podoliaka@gmail.com))

Kharkiv, May the 12th of 2016

@rpodoliaka (http://twitter.com/rpodoliaka)

http://podoliaka.org (http://podoliaka.org)

Why automate your test cases?

- (open source) projects do not necessarily have dedicated QA engineers
- projects are (usually) complex and time of engineers is precious
- regression testing
- test driven development

Automated testing

Test cases are usually written by software engineers:

- **Unit** (individual functions or classes)
- **Integration** (modules or logical subsystems)

Test cases are usually written by QA engineers:

- **Functional** (checking a program against design specifications)
- **System** (checking a program against system requirements)

Case study: binary search

```
In [4]: def binary_search(arr, key):
            return _binary_search(arr, key, 0, len(arr))


        def _binary_search(arr, key, left, right):
            if left >= right:
                return

            middle = left + int((right - left) / 2)

            if arr[middle] == key:
                return middle
            elif arr[middle] < key:
                return _binary_search(arr, key, left, middle)
            else:
                return _binary_search(arr, key, middle, right)
```

A test case example

```
In [5]: import unittest


        class TestBinarySearch(unittest.TestCase):
            def setUp(self):
                super().setUp()

                self.empty = []
                self.arr = list(range(10))

            def test_empty_arr(self):
                self.assertIsNone(binary_search(self.empty, 42))

            def test_key_not_found(self):
                self.assertIsNone(binary_search(self.arr, 42))
```

```
In [6]: suite = unittest.TestSuite()
        suite.addTests([TestBinarySearch('test_empty_arr'),
                        TestBinarySearch('test_key_not_found')])

        unittest.TextTestRunner().run(suite)
```

```
..
----------------------------------------------------------------------
Ran 2 tests in 0.003s

OK
```

```
Out[6]: <unittest.runner.TextTestResult run=2 errors=0 failures=0>
```

pytest: test runner

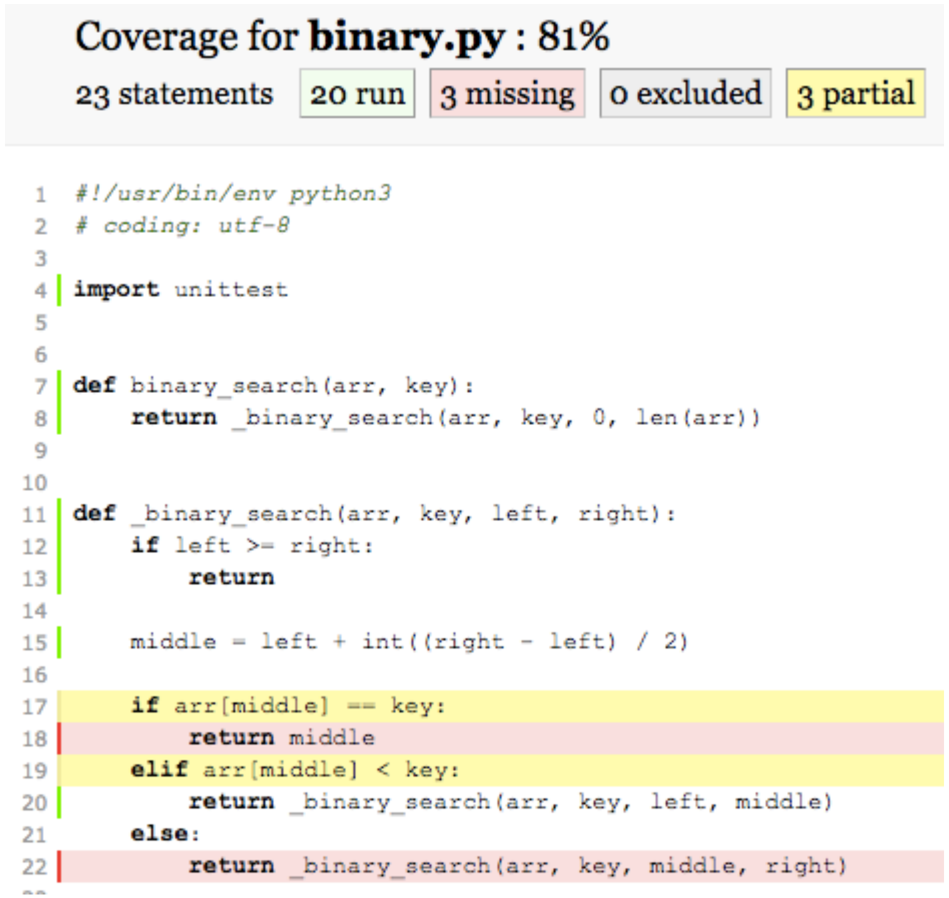
- pytest (http://pytest.org/) is the most popular test runner for Python
- discovers and runs test cases
- supports advanced features (e.g. test fixtures, running tests in parallel, running only failed test cases, etc)

```
(.venv3)Romans-Air:code malor$ py.test -v binary.py
== test session starts

platform darwin -- Python 3.4.3, pytest-2.9.1, py-1.4.30, pluggy-0.3.1 -- /Users/m
alor/.venv3/bin/python3.4
cachedir: .cache
rootdir: /Users/malor/Dropbox/talks/autotesting/code, inifile:
collected 2 items

binary.py::TestBinarySearch::test_empty_arr PASSED
binary.py::TestBinarySearch::test_key_not_found PASSED

== 2 passed in 0.05 seconds
```

coverage: measuring of test coverage

- coverage (https://coverage.readthedocs.io) provides reporting on what lines of code have actually been executed
- allows you to understand which parts of the code are covered by tests and which are not
- test coverage is a measurable and comparable metric: a high test coverage value is required (but not enough) to make sure code works as expected

```
(.venv3)Romans-Air:code malor$ coverage report -m
Name          Stmts   Miss Branch BrPart  Cover   Missing
-------------------------------------------------------
binary.py        23      3      8      3    81%   18, 22, 40, 17->18, 19->22, 39->40
```


Coverage for **binary.py** : 81%

23 statements | 20 run | 3 missing | 0 excluded | 3 partial

```python
#!/usr/bin/env python3
# coding: utf-8

import unittest


def binary_search(arr, key):
    return _binary_search(arr, key, 0, len(arr))


def _binary_search(arr, key, left, right):
    if left >= right:
        return

    middle = left + int((right - left) / 2)

    if arr[middle] == key:
        return middle
    elif arr[middle] < key:
        return _binary_search(arr, key, left, middle)
    else:
        return _binary_search(arr, key, middle, right)
```

tox: manager of virtual environments

- tox (http://tox.readthedocs.io/) standartizes the way tests are run for Python projects - it's as simple as:

```
$ tox
```

- allows to define and manage multiple virtual environments (e.g. different Python versions or a separate virtual environment for building docs):

```
$ tox -epep8,py27,py35
```

```
[tox]
envlist = pep8,py27,py35

[testenv]
deps = pytest
       pytest-cov
commands = py.test binary.py --cov --cov-append

[testenv:pep8]
deps = flake8
commands = flake8 binary.py
```

```
(.venv3)Romans-Air:code malor$ tox
___________________________________________________________________ summary
  pep8: commands succeeded
  py27: commands succeeded
  py35: commands succeeded
  congratulations :)
```

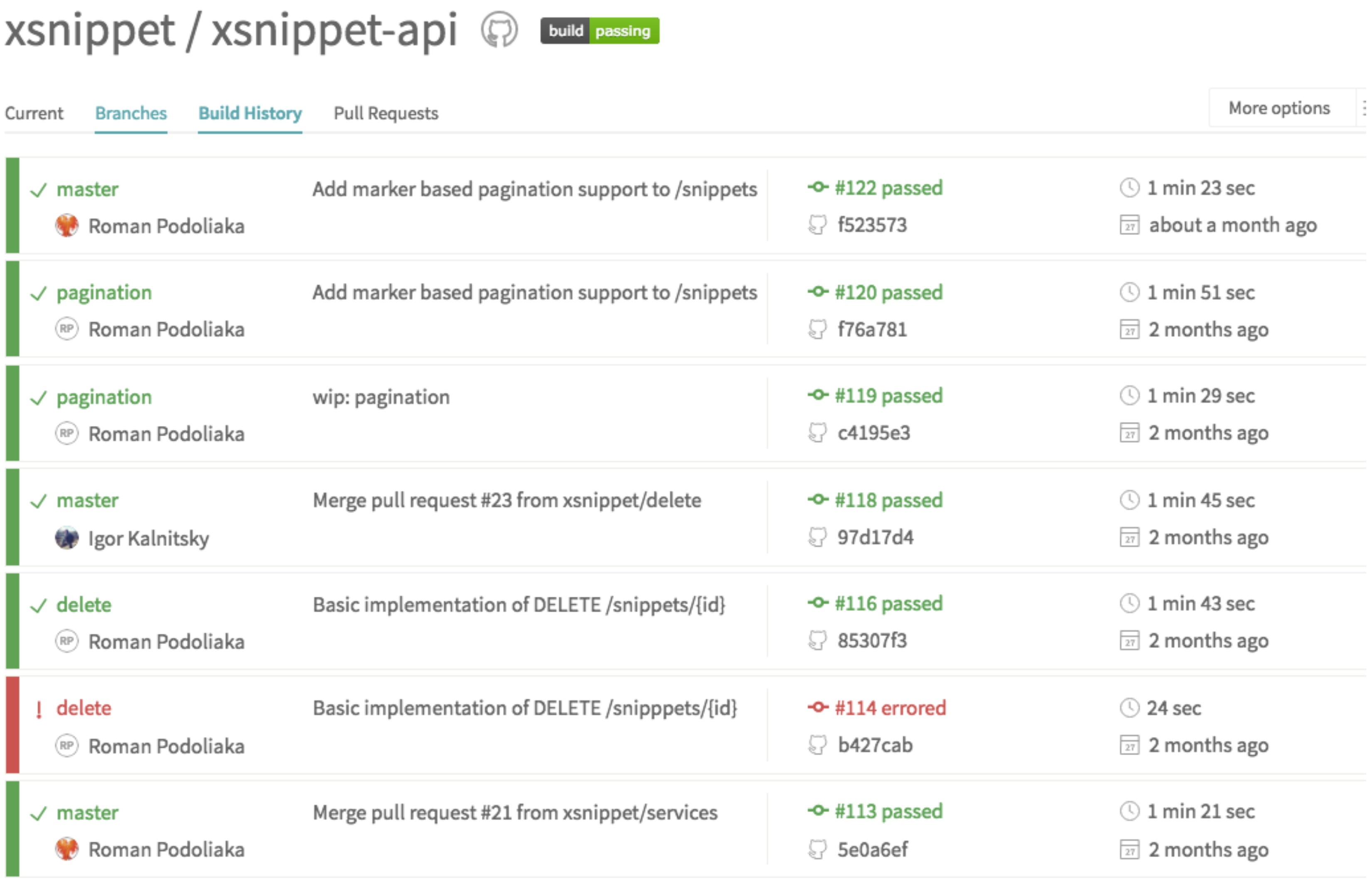
CI (continuous integration)

- testing is only efficient, if it's enforced (e.g. so that one can't "forget" to run the tests)
- CI is intended to automate all the required build steps, including testing
- CI can (and should) be run in "gating" mode: a commit can not be merged, if tests fail

Travis CI

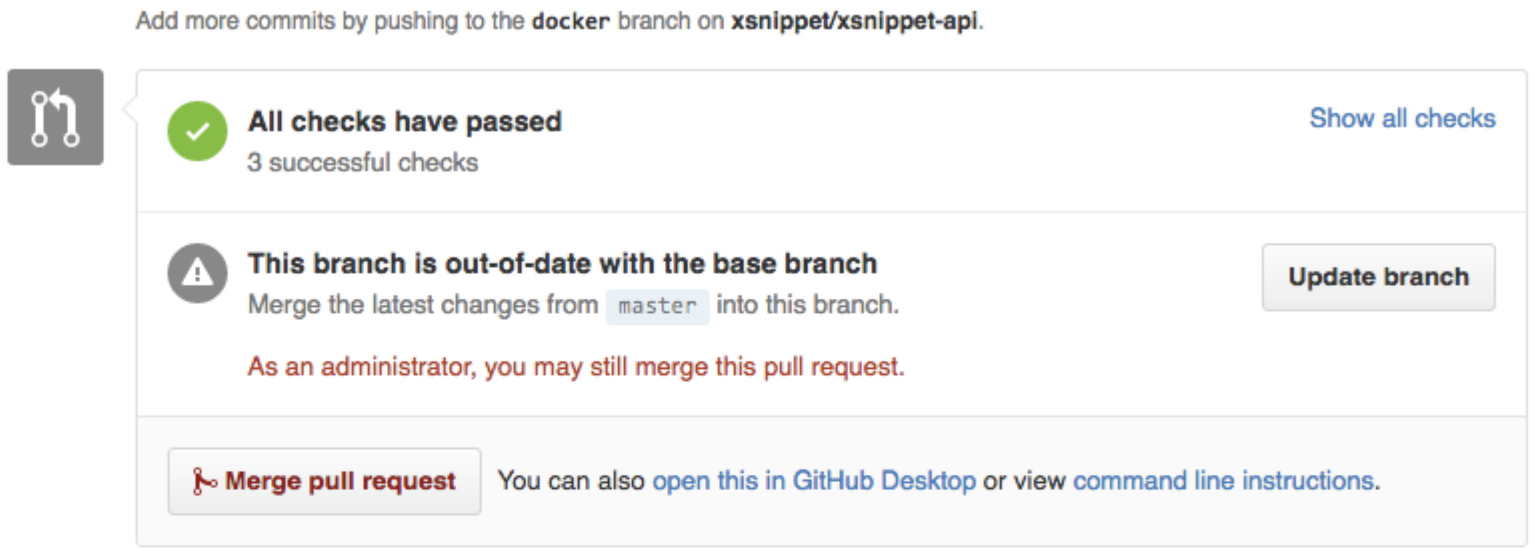
- https://travis-ci.org/ (https://travis-ci.org/)
- a third-party CI system, that is easily (https://docs.travis-ci.com/user/for-beginners) integrated with GitHub
- free of charge for open source projects
- declarative (https://docs.travis-ci.com/user/getting-started/) configuration stored in a repo

```yaml
sudo: false
language: python
python: 3.5

services:
  - mongodb

install:
  # greenlet is needed to submit concurrency=greenlet based coverage
  # to coveralls.io service
  - pip install tox coveralls greenlet

script:
  - tox

after_success:
  - coveralls

notifications:
  email: false
```


xsnippet / xsnippet-api build passing

Current Branches Build History Pull Requests More options

✓ master Roman Podoliaka	Add marker based pagination support to /snippets	#122 passed f523573	1 min 23 sec about a month ago
✓ pagination Roman Podoliaka	Add marker based pagination support to /snippets	#120 passed f76a781	1 min 51 sec 2 months ago
✓ pagination Roman Podoliaka	wip: pagination	#119 passed c4195e3	1 min 29 sec 2 months ago
✓ master Igor Kalnitsky	Merge pull request #23 from xsnippet/delete	#118 passed 97d17d4	1 min 45 sec 2 months ago
✓ delete Roman Podoliaka	Basic implementation of DELETE /snippets/{id}	#116 passed 85307f3	1 min 43 sec 2 months ago
! delete Roman Podoliaka	Basic implementation of DELETE /snipppets/{id}	#114 errored b427cab	24 sec 2 months ago
✓ master Roman Podoliaka	Merge pull request #21 from xsnippet/services	#113 passed 5e0a6ef	1 min 21 sec 2 months ago



Add more commits by pushing to the **docker** branch on **xsnippet/xsnippet-api**.

All checks have passed
3 successful checks

This branch is out-of-date with the base branch
Merge the latest changes from `master` into this branch.

As an administrator, you may still merge this pull request.

You can also open this in GitHub Desktop or view command line instructions.

XSnippet API

build passing coverage 93%

XSnippet is a simple web-service for sharing code snippets on the Internet. Years ago it was started as educational project, and nothing changed since then. That's why the fourth reincarnation is written in Python using `asyncio`.

XSnippet API, however, implements only RESTful API, nothing more. Web UI is not a part of that project anymore.

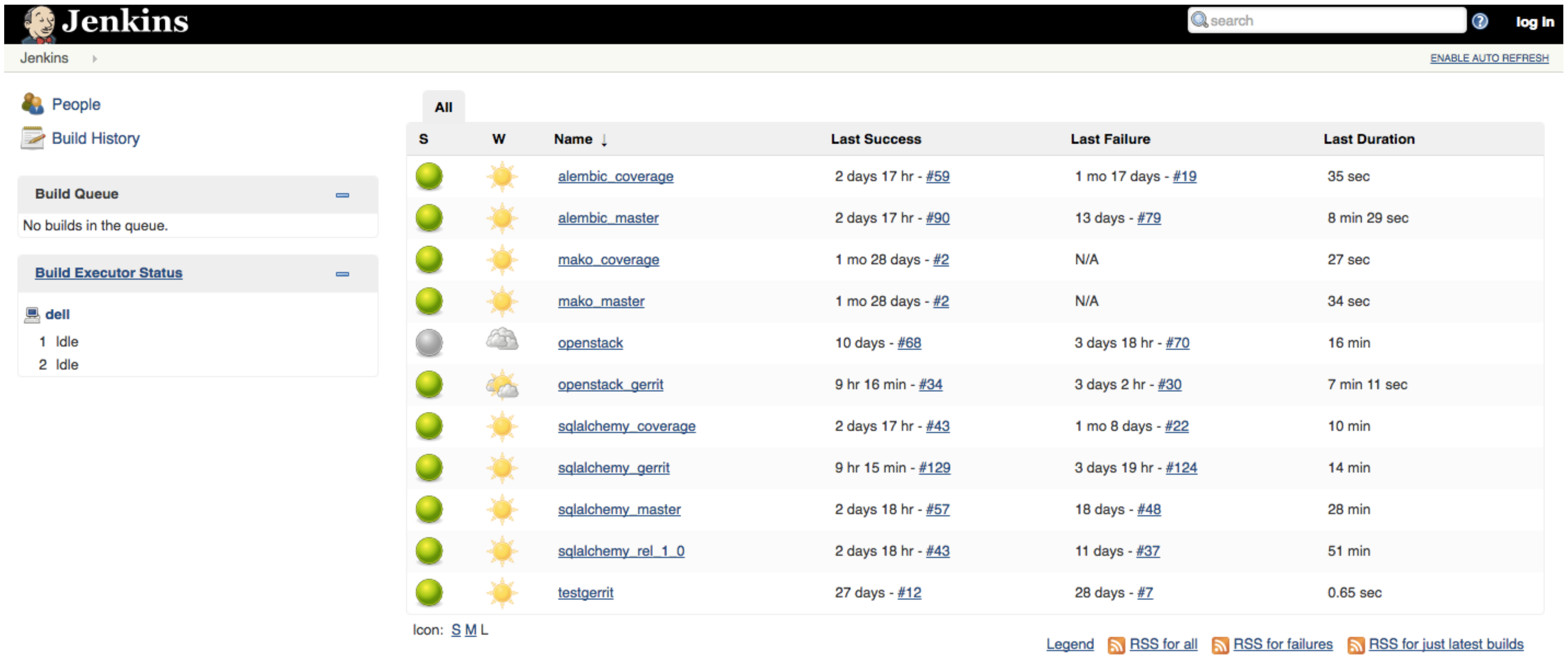
The source code is distributed under MIT license. Feel free to contribute by sending us pull requests or patches. Your feedback is welcome as well!

Links

- Source: https://github.com/xsnippet/xsnippet-api
- Bugs: https://github.com/xsnippet/xsnippet-api/issues

Jenkins

- https://jenkins.io/ (https://jenkins.io/)
- a comprehensive solution for CI/CD
- a number of plugins for automation of builds steps
- number #1 choice when you need a custom CI
- can be configured declaratively by the means of Jenkins Job Builder (http://docs.openstack.org/infra/jenkins-job-builder/)



Jenkins

search

log in

Jenkins

ENABLE AUTO REFRESH

People

Build History

Build Queue

No builds in the queue.

Build Executor Status

dell

1 Idle

2 Idle

All

S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	alembic_coverage	2 days 17 hr - #59	1 mo 17 days - #19	35 sec
●	☀	alembic_master	2 days 17 hr - #90	13 days - #79	8 min 29 sec
●	☀	mako_coverage	1 mo 28 days - #2	N/A	27 sec
●	☀	mako_master	1 mo 28 days - #2	N/A	34 sec
○	☁	openstack	10 days - #68	3 days 18 hr - #70	16 min
●	⛅	openstack_gerrit	9 hr 16 min - #34	3 days 2 hr - #30	7 min 11 sec
●	☀	sqlalchemy_coverage	2 days 17 hr - #43	1 mo 8 days - #22	10 min
●	☀	sqlalchemy_gerrit	9 hr 15 min - #129	3 days 19 hr - #124	14 min
●	☀	sqlalchemy_master	2 days 18 hr - #57	18 days - #48	28 min
●	☀	sqlalchemy_rel_1_0	2 days 18 hr - #43	11 days - #37	51 min
●	☀	testgerrit	27 days - #12	28 days - #7	0.65 sec

Icon: S M L

Legend RSS for all RSS for failures RSS for just latest builds

```
<?xml version='1.0' encoding='UTF-8'?>
<project>
  <actions/>
  <description></description>
  <logRotator>
    <daysToKeep>7</daysToKeep>
    <numToKeep>-1</numToKeep>
    <artifactDaysToKeep>-1</artifactDaysToKeep>
    <artifactNumToKeep>-1</artifactNumToKeep>
  </logRotator>
  <keepDependencies>false</keepDependencies>
  <properties/>
  <scm class="hudson.scm.NullSCM"/>
  <canRoam>true</canRoam>
  <disabled>false</disabled>
  <blockBuildWhenDownstreamBuilding>false</blockBuildWhenDownstreamBuilding>
  <blockBuildWhenUpstreamBuilding>false</blockBuildWhenUpstreamBuilding>
  <triggers class="vector"/>
  <concurrentBuild>false</concurrentBuild>
  <builders>
    <hudson.tasks.Shell>
      <command># Move into the jenkins directory
cd /var/lib/jenkins

#Add all top level xml files.
git add *.xml

# Add all job config.xml files.
git add jobs/*/config.xml

# Add all user config.xml files.
git add users/*/config.xml

# Add all user content files.
git add userContent/*
```

```
# Remove files from the remote repo that have been removed locally.
COUNT=`git log --pretty=format: --name-only --diff-filter=B | wc -l`
if [ $COUNT -ne 0 ]
  then git log --pretty=format: --name-only --diff-filter=B | xargs git rm
fi

# Commit the differences
git commit -a -m &quot;Automated commit of jenkins chaos&quot;

# Push the commit up to the remote repository.
git push origin master

</command>
    </hudson.tasks.Shell>
  </builders>
  <publishers/>
  <buildWrappers/>
```

```yaml
- common-sync-params: &common-sync-params
    # Branches to sync (see also short names below)
    upstream-branch: 'stable/mitaka'
    downstream-branch: '9.0/mitaka'
    fallback-branch: 'master'

    # Branch short names for jobs naming
    src-branch: mitaka
    dst-branch: 9.0

    # Syncronization schedule
    sync-schedule: 'H 3 * * *'  # every day at 3:XX am, spread evenly

    # Gerrit parameters
    gerrit-host: 'review.fuel-infra.org'
    gerrit-port: '29418'
    gerrit-user: 'openstack-ci-mirrorer-jenkins'
    gerrit-creds: 'a4be8c41-43e0-4269-94d8-53133cfe3ae5'
    gerrit-topic: 'sync/stable/mitaka'

    name: 'sync-{name}-{src-branch}-{dst-branch}'
    jobs:
        - 'sync-{name}-{src-branch}-{dst-branch}'
```

Useful links

- Travis CI (https://travis-ci.org/) - CI as a service, that is easily integrated with GitHub
- Test Driven Development (http://www.amazon.com/Test-Driven-Development-Kent-Beck/dp/0321146530) - a book on TDD by Kent Beck
- Understanding the OpenStack CI System (http://www.joinfu.com/2014/01/understanding-the-openstack-ci-system/) - a blog post on OpenStack CI internals by Jay Pipes

Summary

- “If it’s not tested, it’s broken.” (Monty Taylor)
- test coverage is an important metric, but it's not the only one - use common sense!
- writing tests is not always fun, but it makes your life easier
- Travis CI is a good start, but sometimes a more complex or just custom configuration is needed - this is where Jenkins shines

Thank you!

Slides: http://podoliaka.org/talks/ (http://podoliaka.org/talks/)

Your feedback is welcome: @rpodoliaka (https://twitter.com/rpodoliaka) or roman.podoliaka@gmail.com (roman.podoliaka@gmail.com)