Packaging and continual integration

…

Kim Brugger

I don’t care if it works on your machine!

We are not shipping your machine!”

-Vidiu Platon

Packaging & CI: Outline

Introduction

Virtualenv

The pip package manager

Hands on

CI

CI hands on

Reflection

Intro: Compute environment requirements

What are your dependencies?

Any special version of 3rd party software

Default installation location

Any generic solutions available?

Intro: Containerising your product

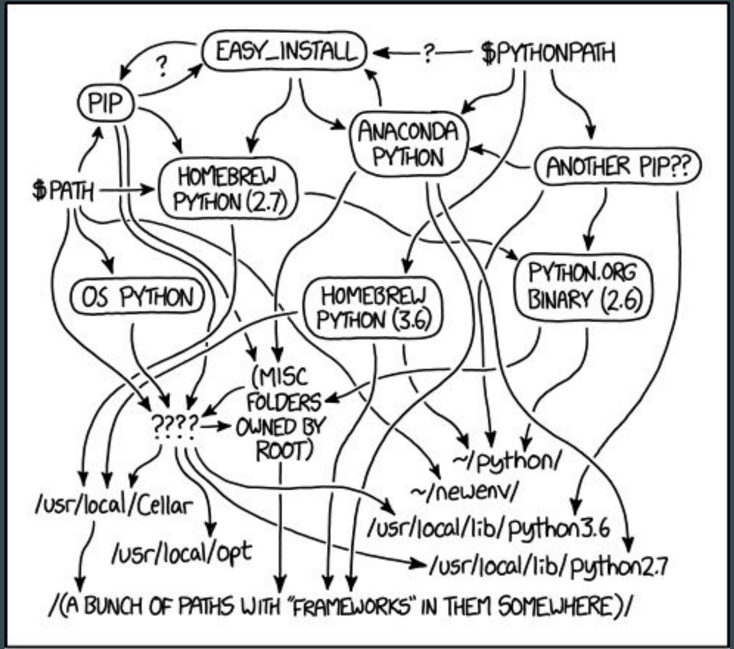
Pip packages: Install via pip

```
# Installation options

# if registered with pypi
pip install fizzbuzz

# install from github
pip install https://github.com/bruggerk/fizzbuzz/

# install branch from github
pip install https://github.com/bruggerk/fizzbuzz@1.0.0

# install branch from directory
pip install ~/fizzbuzz

# install branch from zipfile
pip install fizzbuzz_1.0.0.tgz

# install to a non-standard path
pip install --prefix=/tmp/test fizzbuzz

# install requirements from text file:
pip install -r < requirements.txt
```

```
#setup.py
from setuptools import setup

setup(name=buzzword_generator',
    version='1.0.0',
    description='a buzzword generator',
    url='https://github.com/bruggerk/buzzword_generator/',
    author='Kim Brugger',
    author_email='kim.brugger@uib.no',
    license='MIT',
    packages=['buzzer'],
    install_requires=['munch', ],
    classifiers=['Development Status :: 1.0.0',
                 'License :: MIT License',
                 'Programming Language :: Python :: 3.4'  ],
    scripts=['bin/buzzer.py',
        ],
    # install our config files into an share.
    data_files=[('share/buzzer/', ['share/dict.txt' ])],
)
```

Virtual environments

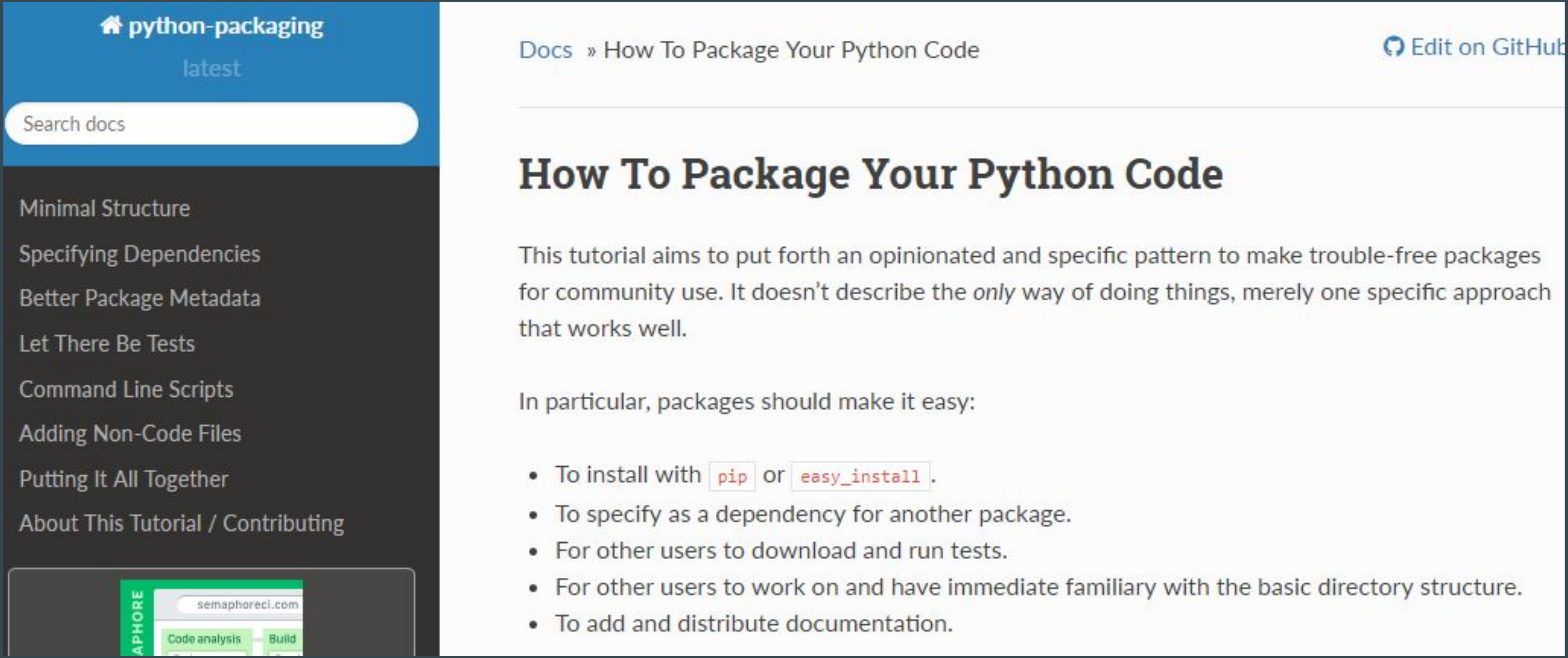
- Isolation - packages live in their own ecosystem
- Permission - No need to beg system admins for packages
- Localisation - Don't mess with other projects
- Organisation - defined package dependencies
- virtualenv - virtual environments for python
- venv - an alternative?



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Virtualenv: demonstration/Hands on

Building a python package



python-packaging
latest

Search docs

Docs » How To Package Your Python Code

Edit on GitHub

How To Package Your Python Code

This tutorial aims to put forth an opinionated and specific pattern to make trouble-free packages for community use. It doesn't describe the *only* way of doing things, merely one specific approach that works well.

In particular, packages should make it easy:

- To install with `pip` or `easy_install`.
- To specify as a dependency for another package.
- For other users to download and run tests.
- For other users to work on and have immediate familiary with the basic directory structure.
- To add and distribute documentation.

Building a fizzbuzz package

The "Fizz-Buzz test" is an interview question designed to help filter out the 99.5% of programming job candidates who can't seem to program their way out of a wet paper bag. The text of the programming assignment is as follows:

"Write a program that prints the numbers from 1 to 100. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”."

Setup hands on

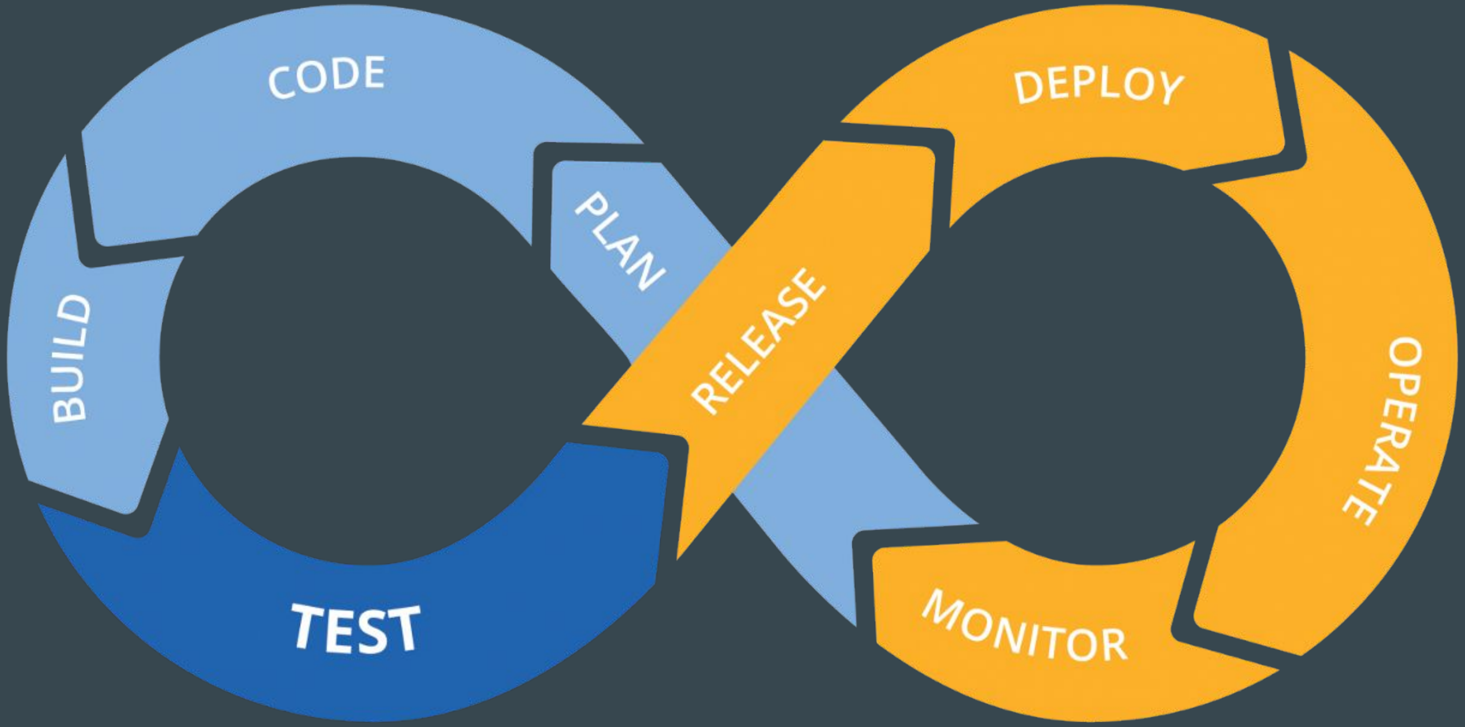
Clone norbis_package repository

Alter setup.py template

Try and install it in a virtualenv

```
#setup.py
from setuptools import setup

setup(name='buzzword_generator',
     version='1.0.0',
     description='a buzzword generator',
     author='C Monkey',
     author_email='c.monkey@banalab.uk',
     license='MIT',
     packages=['buzzer'],
     classifiers=['Development Status :: 1.0.0',
                  'License :: MIT License',
                  'Programming Language :: Python :: 3.4'  ],
     scripts=['bin/buzzer.py',
         ],
    )
```

The infinite DevOps cycle

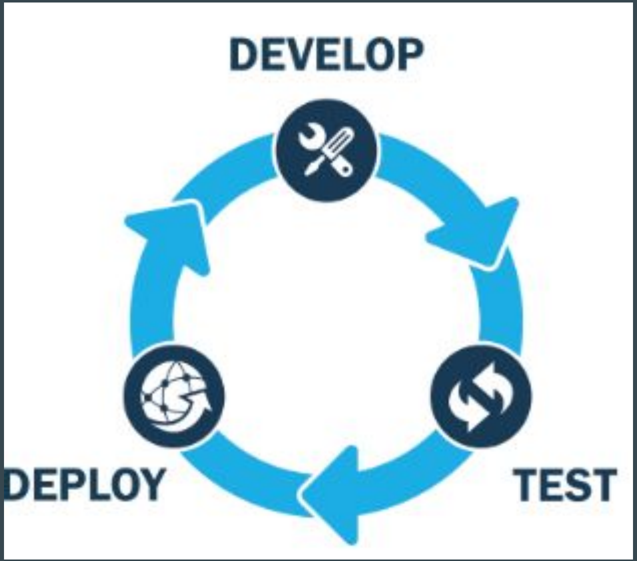


How to instruct CI/CD system to run the pipeline?

A CI system runs pipeline based on a CI configuration file.

A CI/CD configuration file would normally contain the following information:

1. Build
   a. build dependency
   b. build artifacts
2. Test
   a. unit test
   b. functional test
   c. end-to-end test
3. Deploy
   a. ssh to production
   b. upload application
   c. restart service



Continuous integration testing

Travis

- Hosted CI service bound to GitHub
- Tests are run on a dedicated virtual machine
- Supports a wide variety of languages, including R, Python, Perl and many many more
- Supports also several analyzers
- Free to use for Open Source projects. For serious use you might need a paid version.

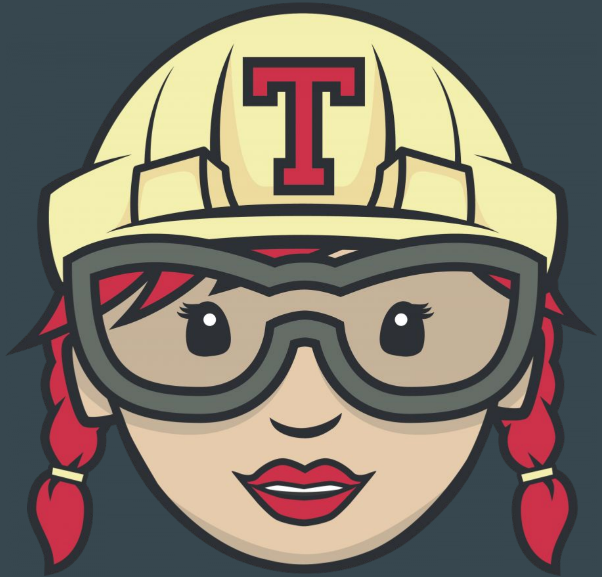
Jenkins

- Jenkins is an open source automation server
- There are tons of plugins available, to do almost anythin
- The plugin system allows you to do complex setups. For example by combining Ansible and Cloud environments you could automatically test and deploy this to myriad different environments, for example on different operatin systems.

Travis test file

.travis.yml

```
python:
  - "2.7"
  - "3.5"
# command to install dependencies
install:
  - pip install -r requirements.txt
# command to run tests
script:
  - pytest
```



Travis hands on

Fork git repository: https://github.com/bruggerk/norbis_travis

Sign in to travis: https://travis-ci.org/

Select repository to auto test

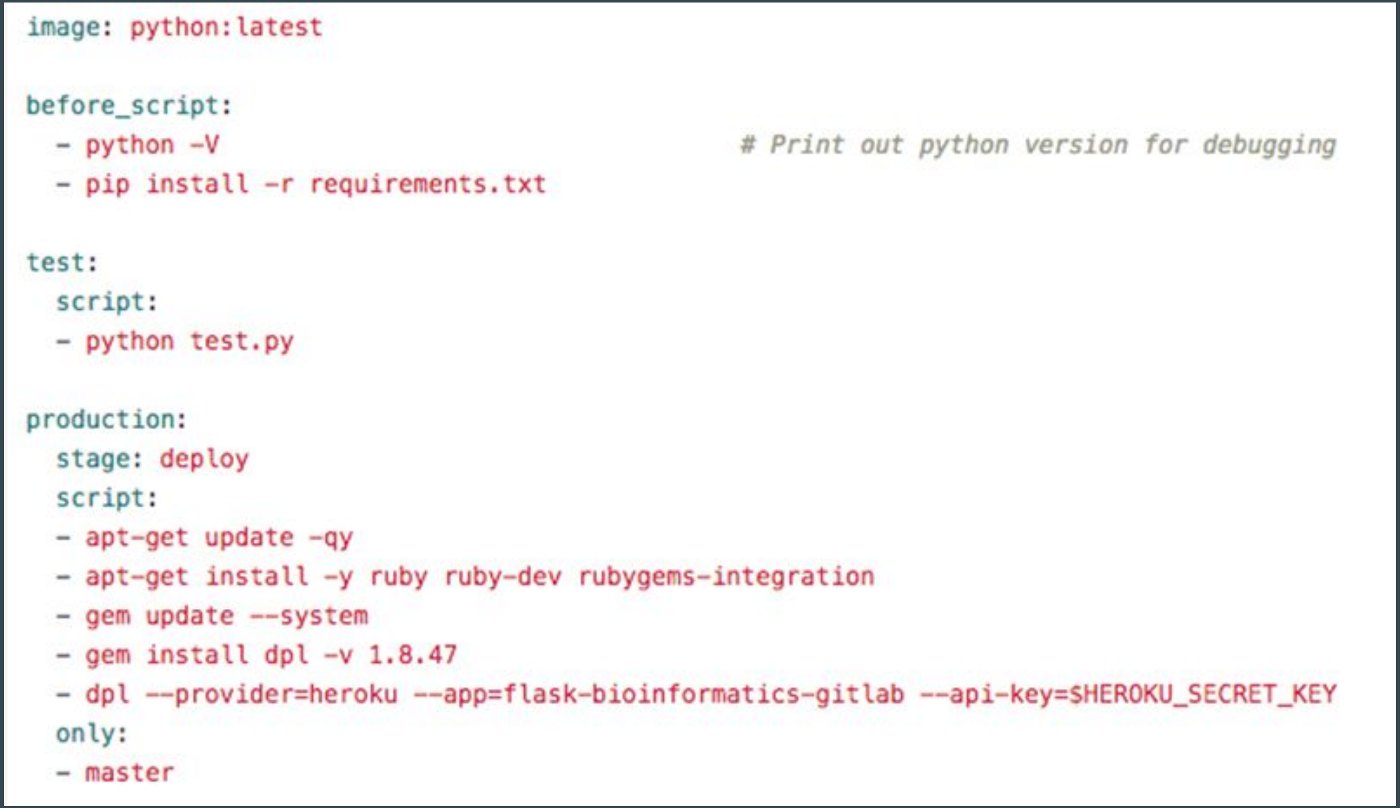
Wait for a short while

See error

Fix error and push changes

Rerun new test.

CD: .gitlab-ci.yml example



```yaml
image: python:latest

before_script:
  - python -V                                      # Print out python version for debugging
  - pip install -r requirements.txt

test:
  script:
  - python test.py

production:
  stage: deploy
  script:
  - apt-get update -qy
  - apt-get install -y ruby ruby-dev rubygems-integration
  - gem update --system
  - gem install dpl -v 1.8.47
  - dpl --provider=heroku --app=flask-bioinformatics-gitlab --api-key=$HEROKU_SECRET_KEY
  only:
  - master
```

Packaging: reflection & thoughts

How do you distribute your code

How do you ensure things are working on other systems?

How do you alert users about a new versions?

How applicable is CI for you?