Smart Speed Cameras

[CSC8110] Cloud Computing

Pedro Silva

16-12-2016

System Design

As suggested in the coursework document, I took on this challenge not only to discover how we can achieve cloud computing using Azure, but also to explore new software tools and programming languages. Although I've previously worked with *python*, I was still unfamiliar with how packaging, testing and development is accomplished in a continuous, collaborative and systematic way. Therefore, I chose to carry out this project using *python* to understand how project directories should be organized, the convention when it comes to naming directories, source and test files, find out the equivalent build tool of Maven for *python*, how to publish a library, etc.

To that extent we introduce the following technologies:

- **GitHub with git-flow**: A solution for hosting projects and building software collaboratively. Git-flow is a git extension that allows you to operate on a repository following Vincent Driessen's branching model.
- **Travis CI with CodeClimate**: Provides continuous integration on top of Github projects, i.e. on successful pushes to the remote repository (Github) a build is triggered on a fresh containerized environment (within Travis' public cloud) which installs your project and runs the entire set of test suites. Code coverage reports generated by pytest-cov are submitted to CodeClimate for assessment and developer feedback.
- **Virtualenv with virtualenvwrapper**: Used to provide separation between python projects being developed on the same machine, which may have different or even conflicting requirements.
- **Setuptools**: Used to package your python project/library so that it can be easily installed easily across heterogeneous systems.
- **Py.test with code coverage**: Tools for automated testing which generate test coverage reports.
- **Sphinx with autodoc**: Generates documentation for your project in similar fashion to javadoc.

I started by setting up the project layout and skeleton, as well as configuring most of these tools so that I can then concentrate primarily on the programming and logical components of the project. Most of these development and integration tools were suggested by Jeff Knupp on a blog post entitled "Open Sourcing a Python Project the Right Way".

At the end of this process and running the azure examples

Figure 1. Basic project layout and directory structure

Pipeline