Server project: documentation

by Mao Heng, Sebastian Rieger, Michal Stachnio, Twinkel Van Impe and Danilo Zocco

# Aim of the project

* set up a server
* find a database
* write python script to read from database to file
* profit?

# Create the droplet

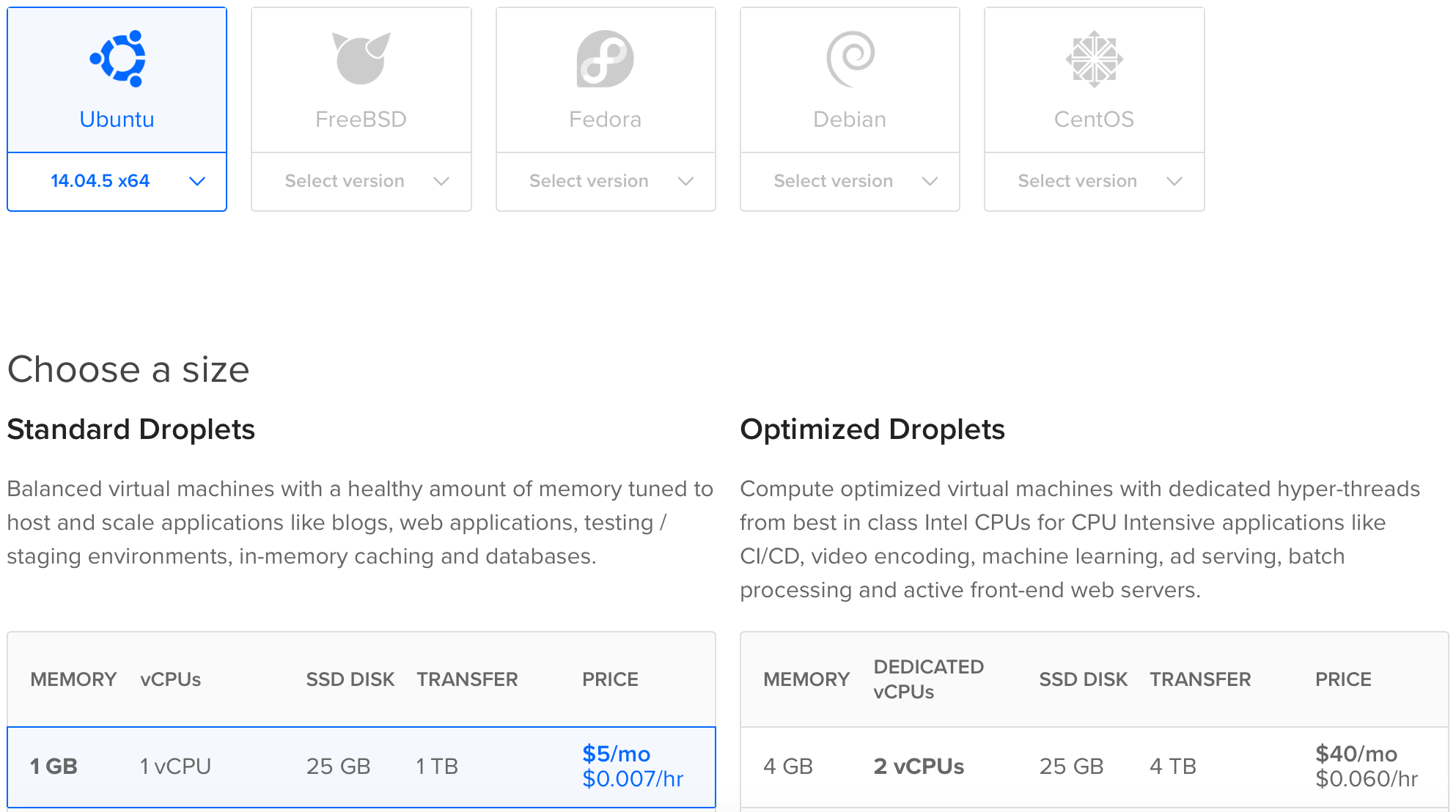
Prior

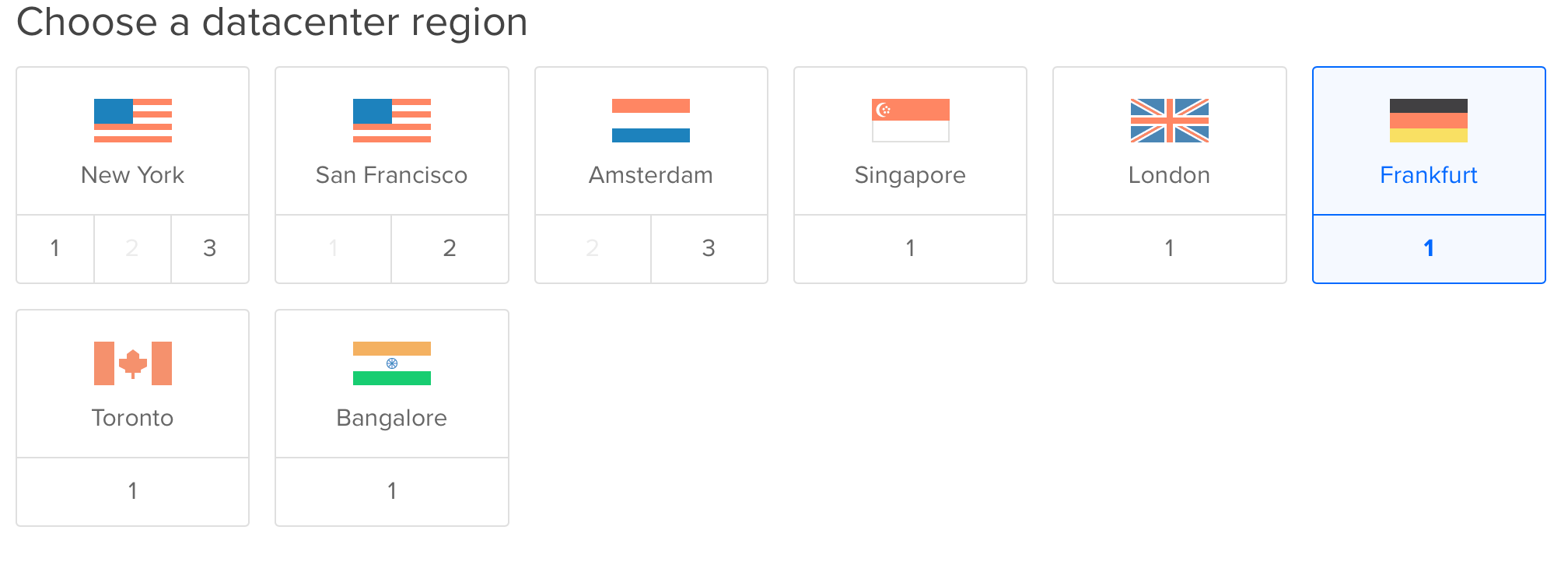
* Make Github account, apply for education pack and look up code for DigitalOcean
* Make DigitalOcean account
* Insert the Github code in your DigitalOcean profile (‘Settings’ > ‘Billing’) to receive $50 credit.

Tutorials used

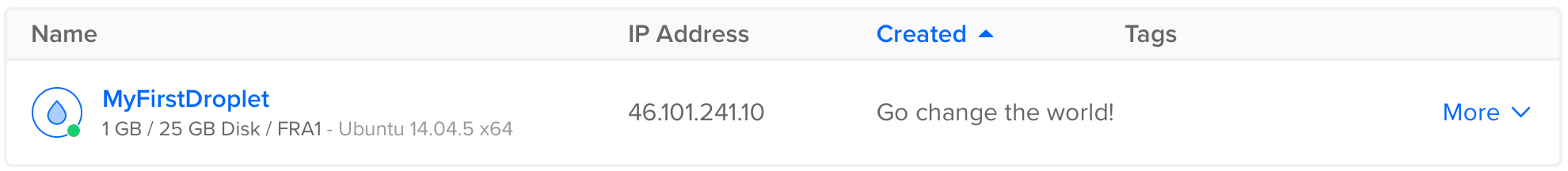
* <https://www.digitalocean.com/community/tutorials/how-to-create-your-first-digitalocean-droplet>
* <https://www.digitalocean.com/community/tutorials/how-to-use-ssh-keys-with-digitalocean-droplets>
* <https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-16-04>

Screenshots of settings





The result

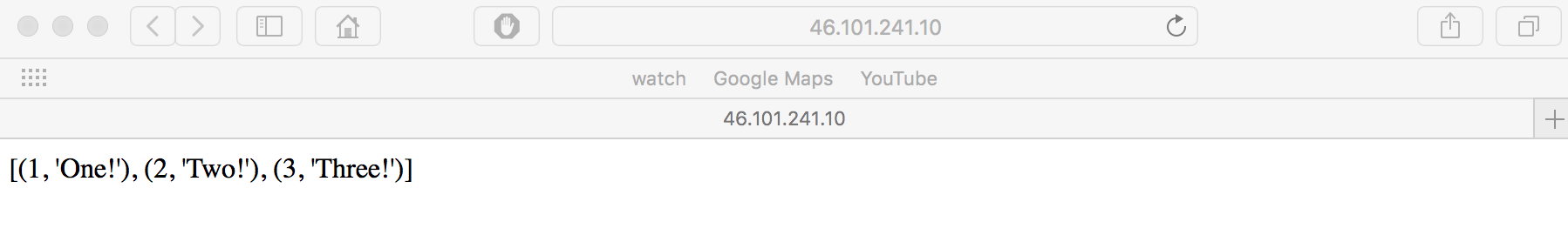


# apache, mysql, python

Tutorials used

* <https://www.digitalocean.com/community/tutorials/how-to-install-python-3-and-set-up-a-local-programming-environment-on-ubuntu-16-04>
* <https://www.digitalocean.com/community/tutorials/how-to-set-up-an-apache-mysql-and-python-lamp-server-without-frameworks-on-ubuntu-14-04>
  + Skip step 1
  + In step 5, *#!/usr/bin/python* should be replaced by *#!/usr/bin/python3* (first line of the python script)

The result



# Database set-up

Tutorials used:

* <https://www.digitalocean.com/community/tutorials/how-to-create-and-manage-databases-in-mysql-and-mariadb-on-a-cloud-server>
* <https://www.digitalocean.com/community/tutorials/how-to-create-a-table-in-mysql-and-mariadb-on-an-ubuntu-cloud-server>
* slides lecture 1

# The code

tutorials used

* **MySQL and python:** <https://technik.blogbasis.net/mit-mysql-in-python-arbeiten-07-03-2013>
* **Cron:** <http://www.unixgeeks.org/security/newbie/unix/cron-1.html>

Our code

* **Our files:** <https://github.com/MStachnio/NumMethodsDataAnalysis>
* **Python script:** Option\_scraper.txt
* **The commands used (including Cron set-up):** Commands server.docx

Database

* **Source of data:** <https://finance.yahoo.com>
* **Example file of data:** see option\_chain.csv

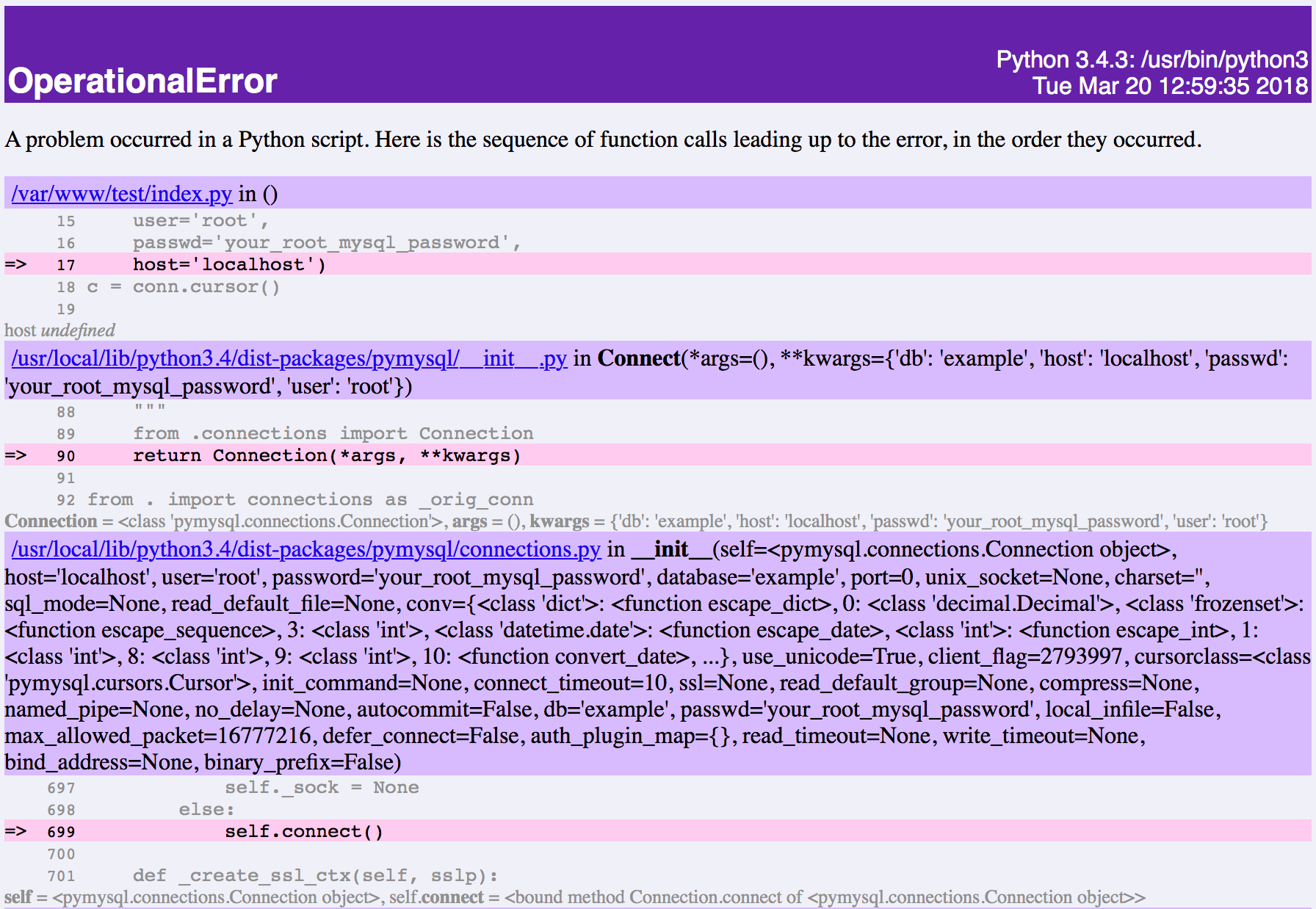
Result

see 46.101.111.222

# final Result

[insert screenshot of working server here]

# Winner of “most beautiful error” competition



# possible Future projects based on this one

* Add a beautiful interface to the page
* Compare our retrieved price data with predicted prices by a model