Architecture

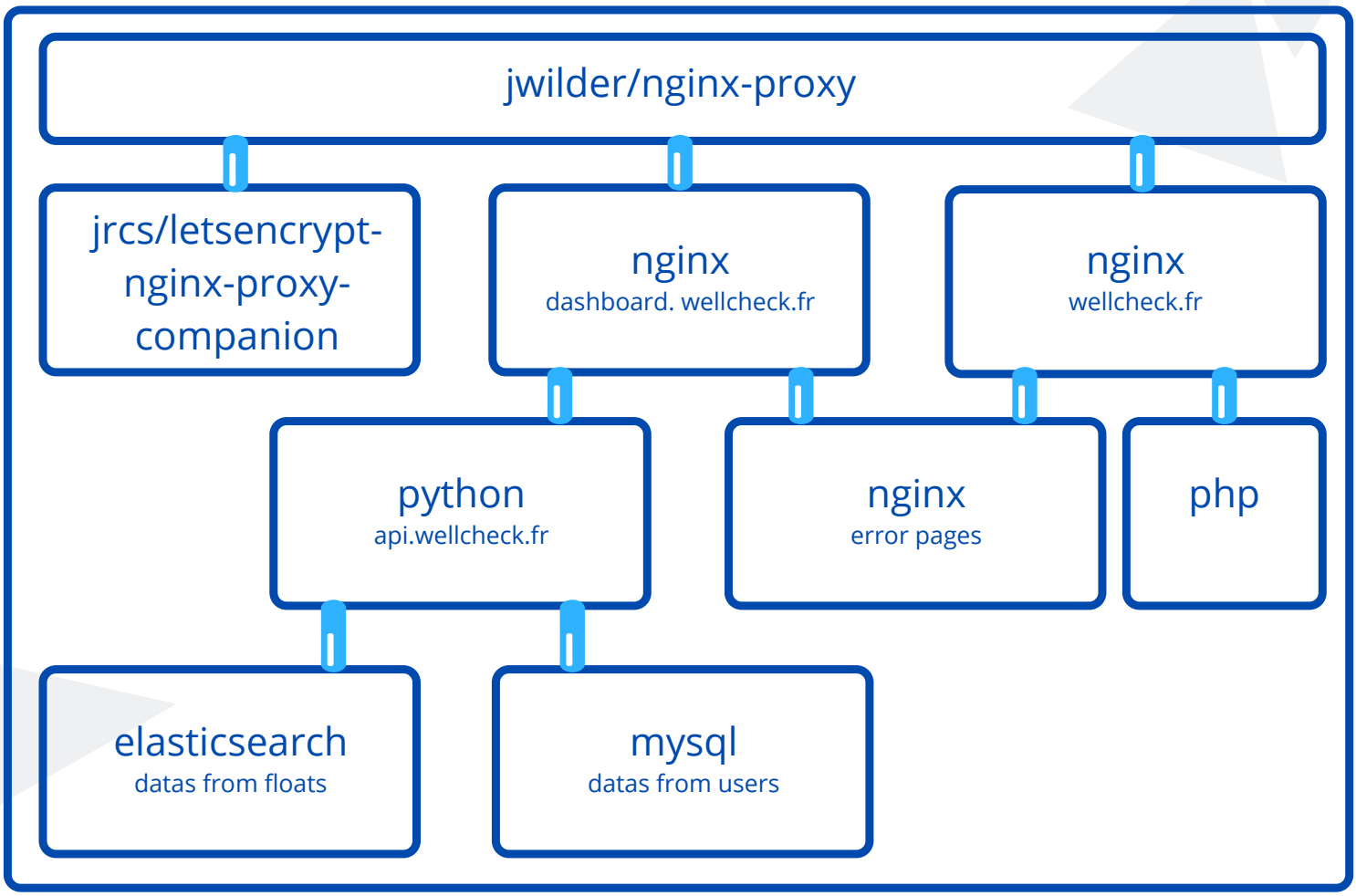
Summary :

Eliot Courtel
February 2020

wellcheck.fr/documentation

All the architecture is powered using docker and docker-compose on a single server this allow us to expand in the future simply by scaling, deploying or developping much faster than in an usual local environnement

server#1



1/ Containers overview

Each docker got a special function:

- jwilder/nginx-proxy :
global reverse proxy linking to the WWW

- jrcs/letsencrypt-nginx-proxy-companion :
handles the automated creation, renewal and use of Let's Encrypt certificates

- nginx (wellcheck.fr) :
serve the differents pages from wellcheck.fr

- php :
allow the execution of php code from wellcheck.fr

- **nginx (dashboard.wellcheck.fr):**
serve the client webapp allowing the user to monitor his account & float(s)

- nginx (error pages) :
serve custom error pages

- python (api.wellcheck.fr) :
allow the execution of the wellcheck's API for webapp & phoneapp

- elasticsearch :
store datas from the float allowing fast query & flexibility

- mysql:
store datas from users

2/ Containers interactions

Dockers interactions:

- jwilder/nginx-proxy :
nginx (wellcheck.fr)
nginx (dashboard.wellcheck.fr)
python (api.wellcheck.fr)

- jrcs/letsencrypt-nginx-proxy-companion :
jwilder/nginx-proxy

- nginx (wellcheck.fr) :
php
nginx (error pages)

**- php : **

- nginx (dashboard.wellcheck.fr):
python (api.wellcheck.fr)

**- nginx (error pages) : **

- python (api.wellcheck.fr) :
elasticsearch
mysql

**- elasticsearch : **

**- mysql: **

3/ Virtuals networks

There are two main networks allowing us handle acces right,
The front network (*proxy*) (green) is accessible directly from the jwilder/nginx-proxy container (main acces to the server)
The back-end network (*back-net*) (red) only contain databases container must be accessed by the python (api) container

server#1