Install Survethi-Monitoring-Tool (Ubuntu 20.04)

Nanni, 27 October 2020 (created 16 June 2020)

tags:
Docker
Survethi
Ubuntu20.04
ufw
docker-compose
git
clone
docker
container

Either Ubuntu-desktop or Ubuntu-server the commands will be the same, we'll use Docker.

- installed the OS completely with default settings, only choose:

- username: <username> (e.g. "survethi")
- password: <a strong password> (e.g. "Strong.Password!123!")

- once finished the installation, login with username and password

- install openssh-server

```
$ sudo apt-get install openssh-server
```

- open the SSH port in the firewall and port 5000 (flask) for the tool

```
$ sudo ufw allow ssh
$ sudo ufw allow 5000
$ sudo ufw logging on
$ sudo ufw enable
```

- note down the IP of the server:
(e.g. 192.168.1.242)

```
$ ip address show
```

- now it is possible to connect via SSH (e.g. using Putty - latest version)

- install Docker:

```
$ sudo wget https://get.docker.com/ -O get_docker.sh
$ sudo chmod +x get_docker.sh
```

```
$ sudo ./get_docker.sh
```

- let's install docker-compose also:

```
$ sudo apt-get install docker-compose
```

- once finished:

```
$ docker --version
$ docker-compose --version
$ sudo groupadd docker
$ sudo usermod -aG docker <myuser>
```

- logout and login (to use the new groups setting):

```
$ logout
```

- Now let's pull the code (the first time is clone command):

```
$ git clone git://github.com/informatici/survethi-monitoring-tool
```

- edit the configurations:
(since is a Python file (.py) never use tabs for spaces, only spaces, and indentation must be perfect)

```
$ cd survethi-monitoring-tool
$ nano config.py (set production settings)
```

- run docker-compose (the first time it builds all, after only what needed)

```
$ docker-compose up --build
```

- test the application

```
from another machine browse to http://<IP noted down above>:5000
http://<IP noted down above>:5000/diseases -> list the diseases in the OH DB
check if any error on the server
```

- send process to background (**detaching**)

```
CTRL-p CTRL-q
```

- list running containers

```
$ docker container ls
```

- see stdout (attach standard output of the container)

```
$ docker attach <container ID>
```

- stopping the container

```
$ docker stop tool
```

- start and restart the container

```
$ docker start tool
$ docker restart tool
```

- rebuild the container (after changes to the code)

```
$ docker-compose build
```

- rebuild and start (attached) the container (after changes to the code)

```
$ docker-compose up --build
```

- rebuild and start (detached) the container (after changes to the code)

```
$ docker-compose up --build -d
```

- once we are happy with the created container, let's make it starting at boot:

```
$ docker update --restart=always <container ID (taken from docker container ls)
```

==== PRODUCTION ====
- you need settings in config.py for the first section 'production' (i.e. before production indeed)
- change to 'Production' environment:

```
$ cd survethi-monitoring-tool
$ nano Dockerfile (change FLASK_ENV to production)
```

- rebuild and start (attached) the container with:

```
$ docker-compose up --build
```

==== PRODUCTION ====

==== UPDATES ====
- pull the new code with:

```
$ cd survethi-monitoring-tool
$ git pull git://github.com/informatici/survethi-monitoring-tool
```

- rebuild and start (attached) the container with:

```
$ docker-compose up --build
```

==== UPDATES ====

==== ADDENDUM ====
In order to work, the tool has to point to an *OpenHospital* DB modified with the script `db_changes/` folder (some may have been already applied)
For testing we can replicate 'wolisso' -> 'wolisso_training' and apply the script to the latter
We have to specify this DB in the config.py file in the 'testing' section
'development' section could be used as well to point another DB (maybe on local computer) for development purposes
'production' section should always point to the main DB once the tool is in production (means used by the real users) and to enable it change Dockerfile:

```
$ nano Dockerfile
ENV FLASK_ENV production
```

==== ADDENDUM ====