Simon Gomez - Microservices

Mise en place d’un reverse Proxy

instalation du HA PROXY

```
sudo apt-get install haproxy
```

utilisation de HA PROXY

setup

1. add a new API in your motus app on the path **/port**
   - This API will return “MOTUS APP working on XXXX port YYYY” with XXXX being the os and YYYY the listening port

exemple :

```
MOTUS APP working on Simon-Ubuntu port 3000
```

1. use your motus app and make it listen on port 3000
2. spawn another instance on port 4000

TIPS :

- to get the os name https://nodejs.org/api/os.html#oshostname
- to easily specify a port on launch :

```
 const port = process.env.PORT || 3000
```

```
 PORT=5000 node index.js
```

Simple Use Case

- configure a haproxy to listen to port 3001 and foward to port 3000

Another Server

- configure a second server in the backend on your haproxy
- do a few request on the URL /port ==> check that you can reach both server
- kill one server ==> check that your application is still working on port 3001
- relaunch the server

Unfair balancing

- configure your haproxy to ensure 75% of the request goes to the 3000 app
- check the configuration with a few request

healthcheck

- configure a healthcheck URL
- use the haproxy in debug mode
- kill the app running on port 4000

- check the log
- check the application is still working on port 3001
- relaunch the server

stickiness

- configure a cookie to ensure stickiness
- do a few request using your browser
- do a few request using your curl
- explain the diference

Bonus

configure a haproxy to balance based on the path

1. add another path /anotherport in your motus App
2. configure the haproxy in order to have request /port that will go to your first server and request /anotherport that will go to the second

TIPS :

- you have to use acl in Haproxy

utilisation d'un proxy nginx

```
sudo apt-get install nginx
```

setup a proxy using the documentation

Published with GitHub Pages