

6N11 - Computação Distribuída
41829611 - Gabriel Coelho da Cunha

Webservice com balanceamento de carga

Implementação

Na máquina local:

1. Criei e configurei as máquinas na azure
 - a. Segui os passos de criação apresentados em aula
 - b. Configurei o DNS de cada máquina

DNS name : balanceador-gabriel-41829611.brazilsouth.cloudapp.azure.com

DNS name : webserver1-gabriel-41829611.brazilsouth.cloudapp.azure.com

DNS name : webserver2-gabriel-41829611.brazilsouth.cloudapp.azure.com
2. Instalei python3-pip e python3-venv
 - a. python3-pip para instalação de pacotes python3
 - b. python3-venv para utilização de ambientes python3 virtuais
3. Criei e configurei um ambiente virtual do python3
 - a. Criei o ambiente virtual
 - i. `python3 -m venv .venv`
 - b. Entrei no ambiente virtual
 - i. `source .venv/bin/activate`
 - c. Instalei os pacotes flask e requests no ambiente virtual
 - i. `pip install flask requests`
 - d. Gerei o arquivo requisitos.txt para configuração em outras máquinas
 - i. `pip freeze > requisitos.txt`
4. Criei os arquivos da aplicação conversor_de_moedas
 - a. conversor_de_moedas.py contendo a classe ConversorDeMoedas que realiza a requisição na api de conversão, converte o valor em real para dólar e euro, formata os valores para ter no máximo duas casas decimais e retorna o json no formato abaixo:

```
{
  "maquina": "hostname do servidor que processou",
  "conversao": {
    "real": valor_em_real,
    "dolar": valor_em_dolar,
    "euro": valor_em_euro
  }
}
```

- b. app.py que inicializa a aplicação flask e define a rota de chamada que recebe valor_em_real como parâmetro:

```
http://nome_da_maquina.dominio/convertemoeda/<valor_em_real>
```

- c. main.py que inicializa o servidor web no endereço 0.0.0.0 para possibilitar acesso externo

5. Criei o arquivo de configuração do nginx load-balancer.conf
6. Copiei a pasta conversor_de_moedas que contém a aplicação em flask para ambos os webserver

```
→ webserver1 (main) x scp -r ./conversor_de_moedas azureuser@webserver1-gabriel-41829611.brazilsouth.c
loudapp.azure.com:~/
app.py                                100% 348    11.1KB/s  00:00
conversor_de_moedas.py                100% 1414   36.4KB/s  00:00
main.py                              100% 129     4.3KB/s  00:00
requisitos.txt                        100% 173     6.1KB/s  00:00
→ webserver2 (main) x scp -r ./conversor_de_moedas azureuser@webserver2-gabriel-41829611.brazilsouth.c
loudapp.azure.com:~/
app.py                                100% 348    10.4KB/s  00:00
conversor_de_moedas.py                100% 1414   53.7KB/s  00:00
main.py                              100% 129     3.9KB/s  00:00
requisitos.txt                        100% 173     5.6KB/s  00:00
```

7. Copiei o arquivo load-balancer.conf para o balanceador

```
→ balanceador (main) x scp ./load-balancer.conf azureuser@balanceador-gabriel-41829611.brazilsouth.clo
udapp.azure.com:~/
load-balancer.conf                    100% 295    10.7KB/s  00:00
```

Na máquina balanceador:

1. Atualizei os repositórios de pacotes
2. Instalei o nginx
3. Deletei o arquivo padrão do nginx

```
azureuser@balanceador-gabriel-41829611:~$ sudo rm /etc/nginx/sites-enabled/default
```

4. Movi o arquivo load-balancer.conf para a pasta em que o nginx lê configurações

```
azureuser@balanceador-gabriel-41829611:~$ sudo mv ~/load-balancer.conf /etc/nginx/conf.d/
```

5. Reiniciei o serviço do nginx

```
azureuser@balanceador-gabriel-41829611:~$ sudo service nginx restart
```

Nas máquinas webserver1 e webserver2:

1. Atualizei os repositórios de pacotes
2. Instalei python3-pip e python3-venv
3. Criei o ambiente virtual python3

```
azureuser@webserver1-gabriel-41829611:~$ python3 -m venv ~/conversor_de_moedas/.venv
```

4. Configurei o ambiente virtual python3

```
azureuser@webserver1-gabriel-41829611:~$ source ~/conversor_de_moedas/.venv/bin/activate  
(.venv) azureuser@webserver1-gabriel-41829611:~$ pip install -r ~/conversor_de_moedas/requisitos.txt
```

5. Iniciei a aplicação:

```
(.venv) azureuser@webserver1-gabriel-41829611:~$ python3 ~/conversor_de_moedas/main.py  
* Serving Flask app "app" (lazy loading)  
* Environment: production  
  WARNING: This is a development server. Do not use it in a production deployment.  
  Use a production WSGI server instead.  
* Debug mode: off  
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

Execução

1. No navegador, acessei o balanceador e redirecionou para o webserver1



The screenshot shows a web browser with the address bar displaying `balanceador-gabriel-41829611.brazilsouth.cloudapp.azure.com/convertemoeda/6`. The browser's navigation bar includes tabs for Entertainment, Work, Study, Investment, News, Developer, TCC, To do courses, Access, and Tools. The main content area displays a REST client response for a GET request. The response status is 200, and the body is a JSON object. The JSON object has a `"maquina"` property with the value `"webserver1-gabriel-41829611"` and a `"conversao"` property which is an object containing `"real": 6.0`, `"dolar": 1.08`, and `"euro": 0.9`.

```
1 // 20210421091612
2 // http://balanceador-gabriel-41829611.brazilsouth.cloudapp.azure.com/convertemoeda/6
3
4 {
5   "maquina": "webserver1-gabriel-41829611",
6   "conversao": {
7     "real": 6.0,
8     "dolar": 1.08,
9     "euro": 0.9
10  }
11 }
```

2. Recarreguei a página e redirecionou para o webserver2



The screenshot shows the same web browser as the previous one, but with the page reloaded. The address bar still displays `balanceador-gabriel-41829611.brazilsouth.cloudapp.azure.com/convertemoeda/6`. The REST client response now shows a different `"maquina"` value: `"webserver2-gabriel-41829611"`. The `"conversao"` object remains the same, with `"real": 6.0`, `"dolar": 1.08`, and `"euro": 0.9`.

```
1 // 20210421091752
2 // http://balanceador-gabriel-41829611.brazilsouth.cloudapp.azure.com/convertemoeda/6
3
4 {
5   "maquina": "webserver2-gabriel-41829611",
6   "conversao": {
7     "real": 6.0,
8     "dolar": 1.08,
9     "euro": 0.9
10  }
11 }
```