FACULDADE DE COMPUTAÇÃO E INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Computação Distribuída – Atividade Prática

Atividade de Laboratório

- Crie uma infraestrutura no Amazon EC2 ou Azure para rodar um serviço web (webservice) com balanceamento de carga, em nuvem.
    - Você deve implementar um mecanismo de balanceamento de cargas utilizando o servidor web Ngnix; (1 servidor front-end balanceador de carga)
    - Os servidores Web backend rodarão diretamente o webservice escutando na porta 5000 (2 servidores)

- Disponibilize, no seu servidor, um serviço de conversão de real para dólar e euro como um webservice RESTFul: o usuário informa o valor em real, seu serviço obtém (de alguma API externa) a taxa do dólar e do euro para venda e devolve o valor em dólares e euros correspondentes ao valor em real fornecido, com saída JSON.

- A função deve se chamar: **convertemoeda**

- A URL deverá ser <http://nome_da_maquina.dominio/convertemoeda/<VALOR>

- E o resultado deverá ser em JSON:

```
{
   “conversao”: [
    “real”: <VALOR>,
    “dolar”: <VALOR_EM_DOLAR>,
    “euro”: <VALOR_EM_EURO>,
   ]
}
```

- Você deve implementar seu webservice em Python com Flask (esta atividade deve ser feita desta forma; outras atividades poderão, eventualmente, ser em linguagem livre).
    - Exemplos/tutoriais:
        - https://dzone.com/articles/restful-web-services-with-python-flask (bem simples, fácil de usar)
        - https://opensource.com/article/17/3/writing-web-service-using-python-flask (mais complexo, mostra uma aplicação bem estruturada)
        - https://programminghistorian.org/en/lessons/creating-apis-with-python-and-flask (bem completa, documentação detalhada)

- https://www.datascienceblog.net/post/programming/flask-api-development/ (mostra como documentar a API com Swagger, muito bom)
- https://realpython.com/flask-by-example-part-1-project-setup/ (quase um curso completo para uma aplicação bem completa e complexa)

- **O que entregar no GitHub:** Todo o código, um relatório em PDF com todos os prints da execução, captura de telas e detalhamento da implementação.
- No dia determinado pelo professor os grupos farão a apresentação/demonstração da execução do sistema em funcionamento.

Balanceamento de Cargas

A descrição abaixo é para uma arquitetura de páginas estáticas. Você deve adaptar esta arquitetura no que for necessário para webservices.

Objetivo da tarefa é verificar o funcionamento do balanceador de cargas para um serviço de página web.

1. Crie três máquinas virtuais na AWS com Ubuntu server. Utilize máquinas do T2.micro.
   1.1. Uma máquina para o balanceador de cargas (nginx)
   1.2. Duas máquinas para serem servidor web (apache)

Conexões entre as máquinas:
Navegador (computador) ---------> balanceador -------- webserver1
|---------- webserver2

2. As máquinas deverão se chamar: <tipo-aluno-tia>
   2.1. balanceador-Mario-1234567
   2.2. webserver1-Mario-1234567
   2.3. webserver2-Mario-1234567

3. Para cada máquina, testar se o nome do host está correto utilizando o comando hostname. Se o nome não estiver correto, modifique utilizando o comando `hostnamectl set-hostname`.

```
ubuntu@webserver $ hostname
webserver
ubuntu@webserver$ sudo hostnamectl set-hostname
webserver1-Mario-1234567
ubuntu@ webserver$ hostname
```

```
webserver1-Mario-1234567
```

Afim de evitar problema de resolução de nome na máquina, atualize o nome da máquina em /etc/hosts

```
ubuntu@webserver1-Mario-1234567:~$ cat/etc/hosts
127.0.0.1       localhost
127.0.1.1       webserver1-Mario-1234567

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Para o nome da máquina aparecer corretamente, faça um novo login com o comando su:

```
ubuntu@ webserver:~$ su - ubuntu
Senha:
ubuntu@webserver1-Mario-1234567:~$
```

4. Nas máquinas webserver1 e webserver2, instalar o apache. Cada máquina, deve ter uma página html de boas vindas com o nome do aluno, tia e nome da máquina

Webserver1:

ubuntu@ webserver1-Mario-1234567:~$ cat /var/www/html/index.html

```
<html>
    <h1> Seja bem vindo ao site de Mario - TIA 1234567</h1>
    <h1> webserver1-Mario-1234567</h1>
<HTML>
```

Webserver2:

ubuntu@ webserver2-Mario-1234567:~$ cat /var/www/html/index.html

```
<html>
    <h1> Seja bem vindo ao site de Mario  - TIA 1234567</h1>
    <h1> webserver2-Mario-1234567</h1>
<HTML>
```

5. Instale o balanceador de cargas nginx na máquina balanceador. Utilize o algoritmo de balanceamento round-robin (é o algoritmo padrão). Está máquina vai ser responsável por fazer o balanceamento das cargas para para webserver1 e webserver2.

5.1. Instalação (pode ser que o nginx já esteja nos repositórios oficiais; verifique antes).

```
$ nginx=stable
$ sudo add-apt-repository ppa:nginx/$nginx
$ sudo apt-get update
$ sudo apt-get install nginx

$ sudo service nginx restart
```

5.2. Do seu computador, teste a conexão com a máquina balanceador (utilize o comando ping). Do seu computador, acesse o navegador e coloque o IP do balanceador. Aparecerá uma página web “Welcome to nginx! ...”. Está mensagem informa que o nginx está funcionando, mas ainda não está redirecionando para as outras máquinas webserver.

5.3. Crie um arquivo de configuração chamado load-balance.conf, conforme apresentado abaixo.

Primeiramente é criado um módulo upstream que será referenciado como backend, com os IPs dos servidores web. Troque os IPs abaixo pelo IPs das suas respectivas máquinas webserver. Em seguida, utilize o módulo server, onde é referenciado o servidor principal (em server_name utilize IP Público IPv4 do balanceador), e é criado um proxy para o módulo upstream backend.

```
ubuntu@balanceador-Mario-1234567:~$ cat
/etc/nginx/conf.d/load-balancer.conf
upstream backend {
   server XXXX.XXXX.XXXX.XXXX; # webserver1-Mario-1234567
   server XXXX.XXXX.XXXX.XXXX; # webserver2-Mario-1234567

}

server {
   listen 80;
   server_name XXXX.XXXX.XXXX.XXXX; # balanceador-Mario-1234567
   location / {
      proxy_pass http://backend;
   }
}

ubuntu@balanceador-Mario-1234567:~$ sudo service nginx restart
```

6. Do seu computador acesse o navegador e coloque o IP Público (IPv4) do balanceador. Atualize a página e verifique que a cada atualização está sendo apresentado uma página de cada servidor web. Atualize várias vezes, caso esteja sempre aparecendo a mesma página web, limpe a cache do navegador.

7. (Opcional) Faça dois clones da máquina webserver, modifique a página web e faça a máquina balanceador redirecionar também para estas duas máquinas. Execute no navegador do seu computador e observe o funcionamento.

8. (Opcional) Pesquise e modifique o algoritmo de balanceamento e veja o funcionamento.

9. Entrega:

- **arquivo .zip com duas imagens**, de acordo com as especificações e imagens abaixo;
- Para **cada** webserver, entregar uma imagem da tela do computador com:
    - [computador] o navegador executando a página
    - [webserver1] conteúdo do arquivo /var/www/html/index.html a partir do terminal da máquina webserver1
    - [webserver1] resultado do ifconfig

- [balanceador] o arquivo de configuração do nginx da máquina balanceador mostrando os IPs das máquinas (localizados em /etc/nginx/conf.d/load-balancer.conf)