

É fornecida uma aplicação *web* mínima, desenvolvida sobre Node.js com dados armazenados em Elasticsearch. A aplicação permite que sejam lançadas múltiplas instâncias em simultâneo, em portos distintos, prevendo-se a utilização de um *proxy* reverso com balanceamento de carga para distribuir pedidos pelas várias instâncias.

Pretende-se criar um serviço *systemd* que receba instruções de controlo num *socket* de domínio Unix, para:

- Lançar e adicionar ao *proxy* reverso uma nova instância da aplicação Node.js em porto a indicar.
- Terminar uma das instâncias existentes da aplicação Node.js e eliminá-la do *proxy* reverso.
- Desativar (sem eliminar) a aplicação no *proxy* reverso, parando todas as instâncias.
- (Re)ativar a aplicação no *proxy* reverso, (re)lançando todas as instâncias.

## Preparação

- Instalar npm, nodejs e nginx:  

```
sudo apt update  
sudo apt install npm nodejs nginx
```
- Instalar elasticsearch:  

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.5.0-amd64.deb  
sudo dpkg -i elasticsearch-8.5.0-amd64.deb  
sudo nano /etc/elasticsearch/elasticsearch.yml  
    xpack.security.enabled: false
```
- Descompactar o anexo tvsapp.tgz (atenção ao C maiúsculo, espaço e barra /):  

```
sudo tar xzvf tvsapp.tgz -C /  
cd /opt/isel/tvs/tvsapp/app  
npm install
```

## Exercícios

1. Confirme o funcionamento da aplicação *web* fornecida:
  - a. Lance apenas a aplicação *web*, ainda sem a base de dados:  

```
NODE_PORT=1234 node /opt/isel/tvs/tvsapp/app/tvsapp.js
```
  - b. Utilize o *browser* para aceder a <http://localhost:1234>  
(deverá ver a indicação de PORT: 1234 e de *database unavailable*)
  - c. Inicie o elasticsearch:  

```
sudo systemctl start elasticsearch
```
  - d. Utilize de novo o *browser* para aceder a <http://localhost:1234>  
(deverá ver a indicação de PORT: 1234 e um contador que avança com cada novo pedido efetuado)
  - e. Termine a aplicação *web* (ctrl-c), mas deixe o elasticsearch a correr
2. Instale a aplicação *web* como serviço e inicie 4 instâncias nos portos 4001, 4002, 4003 e 4004.  
NOTA: ver /opt/isel/tvs/tvsapp/service/ e [Service Templates](#).
3. Adicione em /etc/nginx/sites-available uma configuração de um site (tvsapp), a expor no porto 8086, para operar como balanceador de carga para as 4 instâncias locais nos portos 4001 a 4004. Ative a configuração em /etc/nginx/sites-enabled em conjunto com a que já lá estará (default). Utilize o *browser* para aceder a <http://localhost:8086> e verificar o funcionamento do balanceador.  
NOTA: ver /opt/isel/tvs/nginx/sites-available/ e [nginx Configuration Control](#)

4. Descubra como se diz `nginx`:
  - ☐ `ene-ginks`
  - ☐ `engine-eks`
  - ☐ `ene-ghinsh`
  - ☐ `endjaine-shis`
5. Usando a linguagem C, construa um serviço *systemd* para receber instruções num *socket* de domínio Unix, a colocar em `/run/isel/tvsctl/request`, que permita:
  - a. **ADD** : Iniciar uma nova instância da aplicação Node.js (como no ponto 2 anterior) num porto indicado e adicionar essa instância à configuração do *site tvsapp* no `nginx`.
  - b. **DEL** : Terminar uma das instâncias da aplicação Node.js, removendo essa instância da configuração do *site tvsapp* no `nginx`.
  - c. **STOP** : Terminar todas as instâncias da aplicação Node.js e desativar o *site tvsapp* no `nginx`.
  - d. **START** : Reativar o *site tvsapp* no `nginx` e reiniciar as instâncias da aplicação *web*.
6. Construa o programa cliente para poder enviar as instruções indicadas no ponto 5 ao serviço `tvsctl`.

## Entrega

Entregue o exercício 5 e 6 usando a *tag* **SE3** no repositório GitHub.

ISEL, 15 de novembro de 2022

Data limite de entrega: 5 de dezembro de 2022