Parcial 2

Estudiantes:
Jorge Hernandez - A00317220

Repositorio de Github : github.com/GeorgeArturo/sd-exam2

Objetivo

1.Implementar un balanceador de carga con contenedores

Procedimiento

Para esta actividad es necesario desplegar 4 devices, el primer device va a contener nginx quien sera el encargado de hacer de balanceador de carga, los otros 3 devices seran los servidores web, que para este caso tienen un archivo diciendo "hola soy la web #".

Es necesario tener las imagenes de nginx y httpd instaladas en el equipo.

Lo primero que hice para esto fue configurar las web, a continuación solo mostrare el ejemplo de una web, pues se realizo lo mismo para las 3 webs.

Dockerfile

```
FROM httpd
ADD index.html /usr/local/apache2/htdocs/index.html
```

el index que se añade esta en los recursos.
Como se dijo anteriormente este fue el procedimiento para todas las webs, lo unico que cambia es el index de cada una.

A continuación mostrare como fue la configuración del nginx
*nginx.conf

```
worker_processes 4;

events { worker_connections 1024; }

http {
    sendfile on;

    upstream app_servers {
        server web1:80;
        server web2:80;
        server web3:80;
    }
```

```
    server {
        listen 80;

        location / {
            proxy_pass         http://app_servers;
            proxy_redirect     off;
            proxy_set_header   Host $host;
            proxy_set_header   X-Real-IP $remote_addr;
            proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header   X-Forwarded-Host $server_name;
        }
    }
}
```

Aqui se hace la conexion entre el nginx y las web

Dockerfile

Se usa el contenedor con nginx pre instalado
FROM nginx

Se elimina el archivo de configuración default y su carpeta

RUN rm /etc/nginx/conf.d/default.conf && rm -r /etc/nginx/conf.d

Se agrega el archivo de configuracion de nginx

ADD nginx.conf /etc/nginx/nginx.conf

Se agrega esta linea para que el contenedor no termine su ejecucion.

RUN echo "daemon off;" >> /etc/nginx/nginx.conf

CMD service nginx start
```
Finalmente teniendo configurando las webs y el balanceador de carga, se configura el docker compose que sera quien levante toda la infraestructura.

Docker compose
``` python
version: '2'

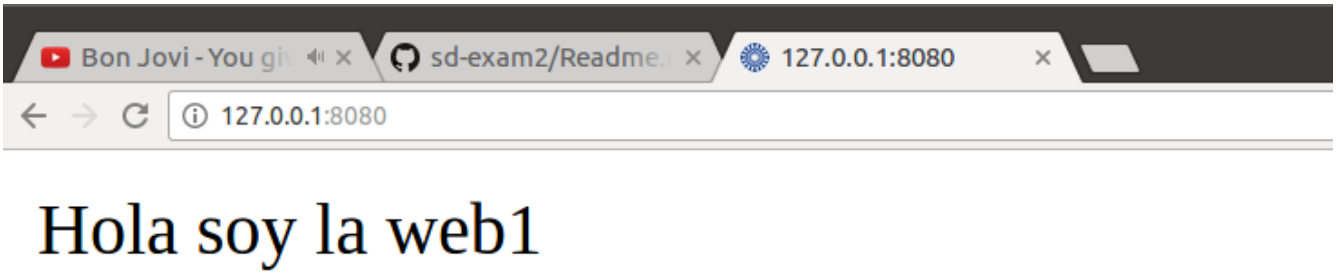
services:
  web1:
    build:
      context:  ./web1
      dockerfile: Dockerfile
    expose:
      - "5000"
```
```

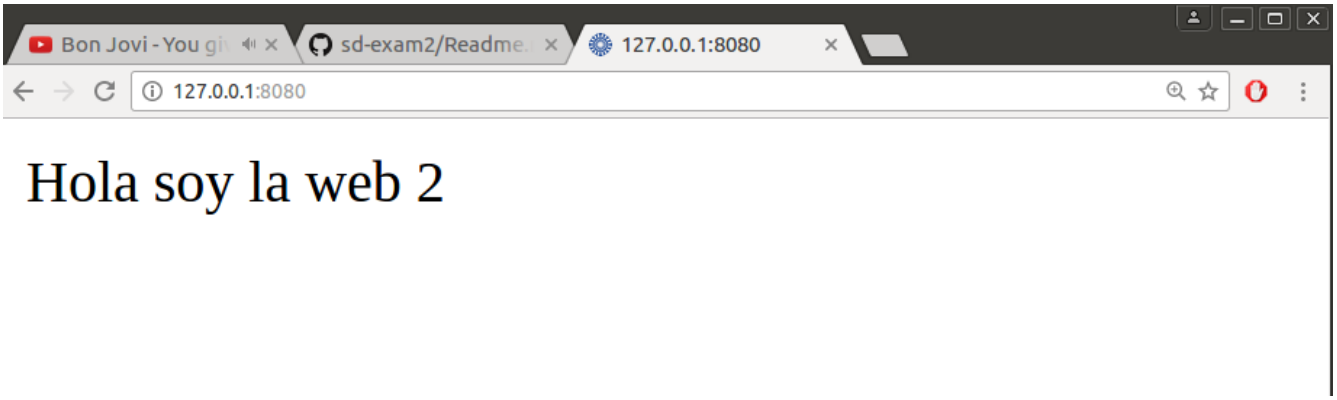
```
web2:
  build:
    context:  ./web2
    dockerfile: Dockerfile
  expose:
    - "5000"

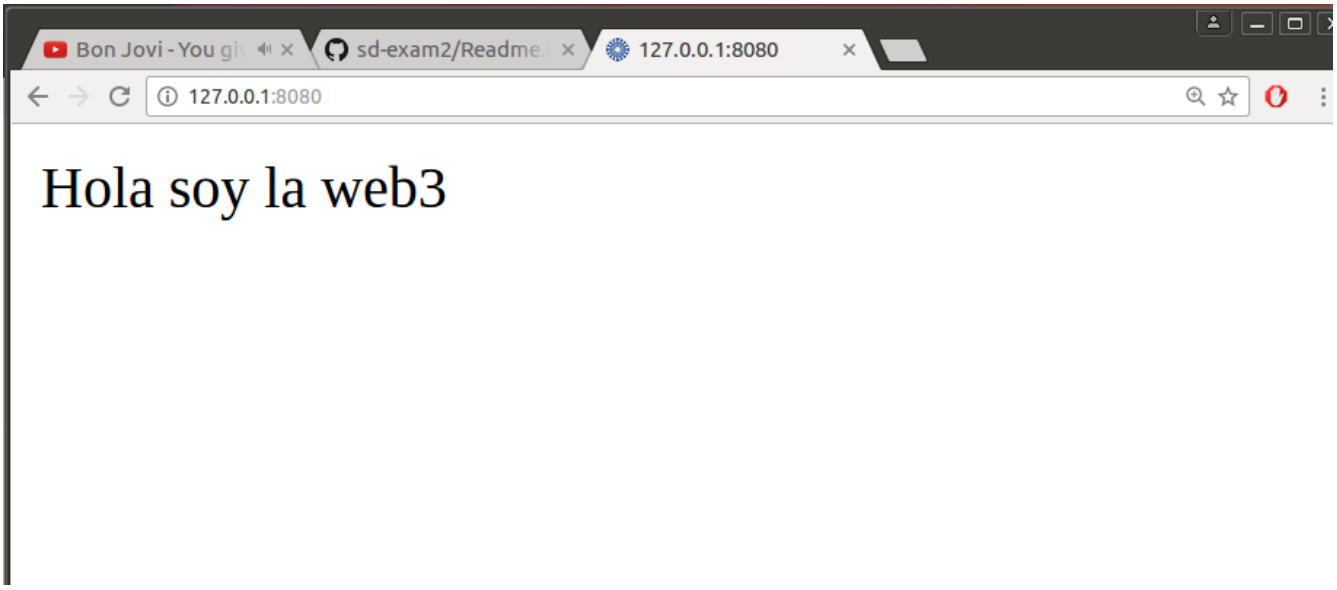
web3:
  build:
    context:  ./web3
    dockerfile: Dockerfile
  expose:
    - "5000"

proxy:
  build:
    context:  ./nginx
    dockerfile: Dockerfile
  ports:
    - "8080:80"
  links:
    - web1
    - web2
    - web3
```

A continuación una prueba de funcionamiento




Bon Jovi - You gi
sd-exam2/Readme.
127.0.0.1:8080
127.0.0.1:8080
Hola soy la web 2


Bon Jovi - You gi
sd-exam2/Readme.
127.0.0.1:8080
127.0.0.1:8080
Hola soy la web3