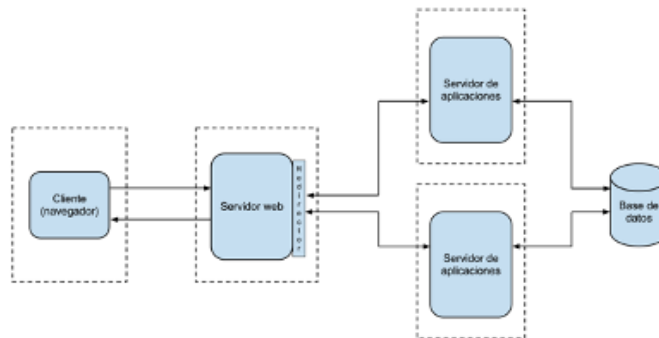


# Balanceo de carga en Apache

---



Jesús Joel Meneses Meneses  
2º DAW A  
DPL---Despliegue de Aplicaciones Web

---

## Índice

1. Activación de los módulos necesarios en Apache

2. Configuración de Apache para trabajar como balanceador de carga para el tráfico HTTP

---

## 1. Activación de los módulos necesarios en Apache

Activamos los nodos necesarios en apache, que son los de la siguiente imagen

Para desactivar un modulo es con el comando siguiente

```
a2dismod proxy_ajp
```

Una vez realizado el paso anterior reiniciaremos el servicio Apache

```
systemctl restart apache2
```

```
systemctl restart apache2
joel@joel-VirtualBox:/etc/apache2/sites-available$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
systemctl restart apache2
joel@joel-VirtualBox:/etc/apache2/sites-available$ sudo a2enmod deflate
Considering dependency filter for deflate:
Module filter already enabled
Module deflate already enabled
joel@joel-VirtualBox:/etc/apache2/sites-available$ sudo a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
systemctl restart apache2
joel@joel-VirtualBox:/etc/apache2/sites-available$ sudo a2enmod proxy_balancer
Considering dependency proxy for proxy_balancer:
Module proxy already enabled
Considering dependency alias for proxy_balancer:
Module alias already enabled
Considering dependency slotmem_shm for proxy_balancer:
Enabling module slotmem_shm.
Enabling module proxy_balancer.
To activate the new configuration, you need to run:
systemctl restart apache2
joel@joel-VirtualBox:/etc/apache2/sites-available$ sudo a2enmod proxy_connect
Considering dependency proxy for proxy_connect:
Module proxy already enabled
Enabling module proxy_connect.
To activate the new configuration, you need to run:
systemctl restart apache2
joel@joel-VirtualBox:/etc/apache2/sites-available$ sudo a2enmod proxy_html
Considering dependency proxy for proxy_html:
Module proxy already enabled
Considering dependency xml2enc for proxy_html:
Enabling module xml2enc.
Enabling module proxy_html.
To activate the new configuration, you need to run:
systemctl restart apache2
joel@joel-VirtualBox:/etc/apache2/sites-available$ sudo a2enmod lbmethod_byrequests
Considering dependency proxy_balancer for lbmethod_byrequests:
Considering dependency proxy for proxy_balancer:
Module proxy already enabled
Considering dependency alias for proxy_balancer:
Module alias already enabled
Considering dependency slotmem_shm for proxy_balancer:
Module slotmem_shm already enabled
Module proxy_balancer already enabled
Enabling module lbmethod_byrequests.
To activate the new configuration, you need to run:
systemctl restart apache2
joel@joel-VirtualBox:/etc/apache2/sites-available$ sudo systemctl restart apache2
```

## 2. Configuración de Apache para trabajar como balanceador de carga para el tráfico HTTP

Editamos el archivo 000-default.conf que está en el directorio /etc/apache2/sites-available:

Comprobamos el puerto apache

```
GNU nano 4.8                                ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8080

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

[ 15 líneas leídas ]
```

Añadimos las directivas Proxy y ProxyPass dentro de VirtualHost.

Tendremos que reemplazar IP-HTTP-SERVER-1, IP-HTTP-SERVER-2, IP-HTTP-SERVER-3, y IP-HTTP-SERVER-4, en nuestro caso deberá de ser localhost para las máquinas que estamos utilizando como Front-End

```
GNU nano 4.8                                /etc/apache2/sites-available/000-default.conf
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

# Dejamos la configuración del VirtualHost como estaba
# sólo hay que añadir las siguiente directivas: Proxy y ProxyPass

<Proxy balancer://mycluster>
    # Server 1
    BalancerMember http://localhost:9001
    # Server 2
    BalancerMember http://localhost:9002
    # Server 3
    BalancerMember http://localhost:9003
    # Server 4
    BalancerMember http://localhost:9004
</Proxy>

ProxyPass / balancer://mycluster/

</VirtualHost>
```

Utilizamos la app web de tareas anteriores, para crear una instancias de 4 nodos

El unico fichero que hay que modificar es docker-compose.yml

```
app-web-demo > docker-compose.yml
```

```
1  version: '3.5'
2  services:
3    wildfly_primero:
4      build:
5        context: .
6        args:
7          WILDFLY_NAME: wildfly_primero
8          CLUSTER_PW: secret_password
9      image: wildfly_primero
10     ports:
11       - 9001:8080
12     networks:
13       - default
14   wildfly_segundo:
15     build:
16       context: .
17       args:
18         WILDFLY_NAME: wildfly_segundo
19         CLUSTER_PW: secret_password
20     image: wildfly_segundo
21     ports:
22       - 9002:8080
23     networks:
24       - default
25   wildfly_tercero:
26     build:
27       context: .
28       args:
29         WILDFLY_NAME: wildfly_tercero
30         CLUSTER_PW: secret_password
31     image: wildfly_tercero
32     ports:
33       - 9003:8080
```

Luego haremos un mvn clean install para crearnos el war

Seguidamente eliminamos cualquier contenedor con el siguiente comando

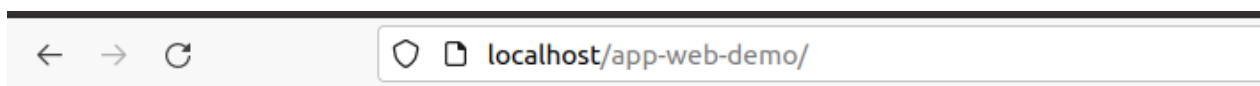
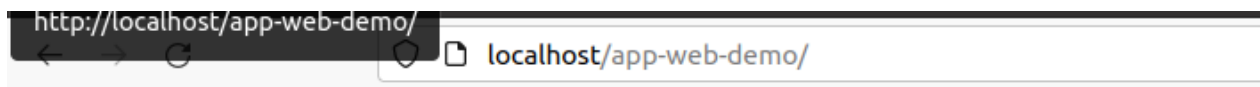
```
docker-compose down
```

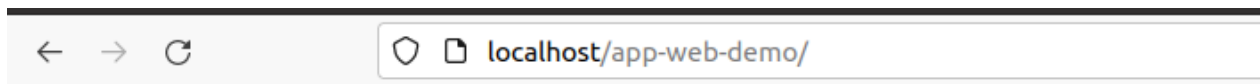
Luego construimos las imagenes

```
docker-compose up --build
```

```
g.cluster.password=${CLUSTER_PW}
--> Using cache
--> 8f5ca2824ed2
Successfully built 8f5ca2824ed2
Successfully tagged wildfly_cuarto:latest
Starting app-web-demo_wildfly_tercero_1 ... done
Starting app-web-demo_wildfly_segundo_1 ... done
Starting app-web-demo_wildfly_cuarto_1 ... done
Starting app-web-demo_wildfly_primero_1 ... done
joel@joel-VirtualBox:~/Proyectos/app-web-demo$ sudo docker-compose up --build
```

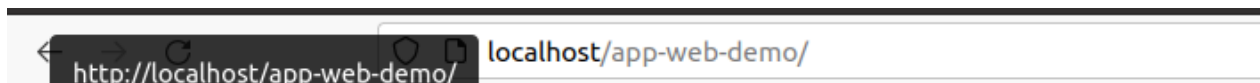
Abrimos el navegador y lo probamos





# Hola Joel Meneses!

Este es el puerto: 9001



# Hola Joel Meneses!

Este es el puerto: 9003

Enlace github: [enlace](#)