

Practica3

Indice

Despliegue de servidores web con usuarios autenticados mediante LDAP usando Docker y docker Compose.....	2
Lanzar con Docker de NGINX + dominio de autenticación LDAP + OpenLDAP.....	2
Desplegament amb Docker de PHP + Apache amb autenticació LDAP.....	3

Despliegue de servidores web con usuarios autenticados mediante LDAP usando Docker y docker Compose

Lanzar con Docker de NGINX + dominio de autenticación LDAP + OpenLDAP

Vamos a crear un Directorio donde estará nuestro index que se mostrara al lanzar el docker de NGINX.

```
david@david-VirtualBox:~/Escritorio$ mkdir app
david@david-VirtualBox:~/Escritorio$ cd app
david@david-VirtualBox:~/Escritorio/app$ cd ..
david@david-VirtualBox:~/Escritorio$ mkdir practica6-3
david@david-VirtualBox:~/Escritorio$ cd practica6-3/
david@david-VirtualBox:~/Escritorio/practica6-3$ sudo nano index.html
[sudo] contraseña para david:
david@david-VirtualBox:~/Escritorio/practica6-3$ mv index.html app/
david@david-VirtualBox:~/Escritorio/practica6-3$ cd app/
david@david-VirtualBox:~/Escritorio/practica6-3/app$ cat index.html
<html>
<body>
<h1>Hola Mundo!</h1>
</body>
</html>
david@david-VirtualBox:~/Escritorio/practica6-3/app$
```

Creamos otro directorio llamado Conf que es donde estará guardada la config de Nginx.

```
david@david-VirtualBox:~/Escritorio/practica6-3$ mkdir conf
david@david-VirtualBox:~/Escritorio/practica6-3$ cd conf/
david@david-VirtualBox:~/Escritorio/practica6-3/conf$ sudo nano ldap_nginx.conf
david@david-VirtualBox:~/Escritorio/practica6-3/conf$ cat ldap_nginx.conf
server {
    listen 8080;

    location = / {
        auth_request /auth-proxy;
    }

    location = /auth-proxy {
        internal;

        proxy_pass http://nginx-ldap:8888;
        # URL y puerto para conectarse al servidor LDAP
        proxy_set_header X-Ldap-URL "ldap://openldap:1389";

        # Base DN
        proxy_set_header X-Ldap-BaseDN "dc=example,dc=org";

        # Bind DN
        proxy_set_header X-Ldap-BindDN "cn=admin,dc=example,dc=org";

        # Bind password
        proxy_set_header X-Ldap-BindPass "adminpassword";
    }
}
```

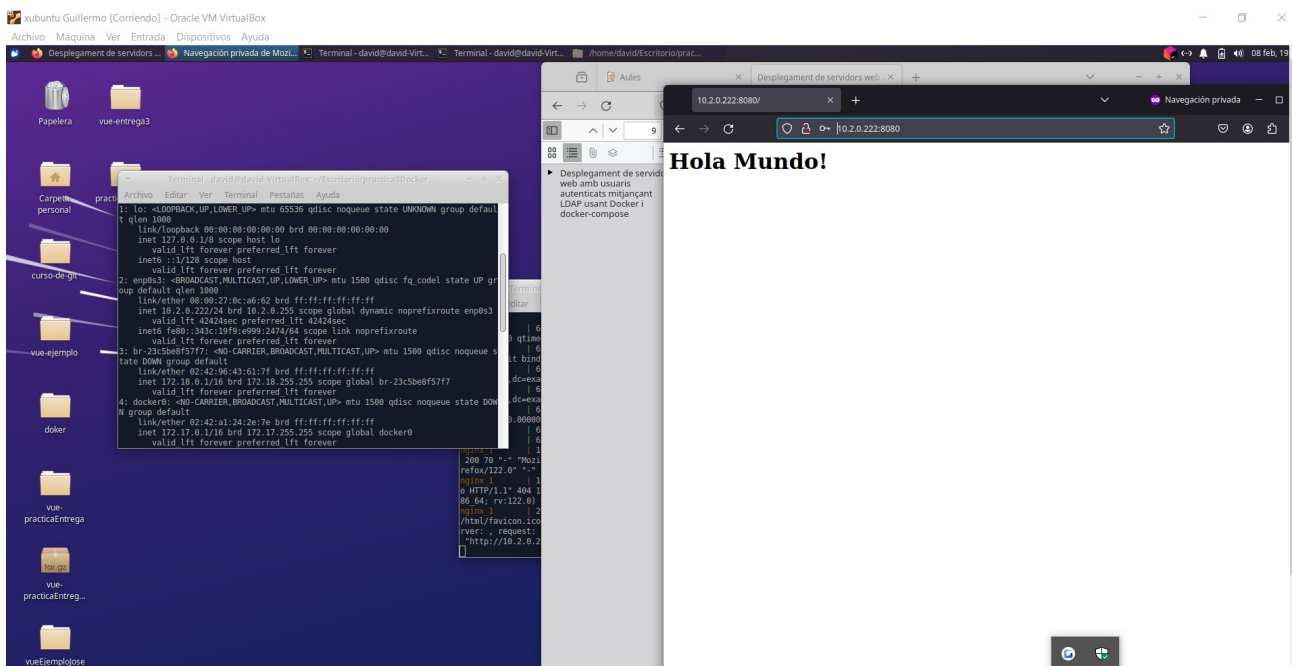
Esta configuración sirve para decirle a Nginx lo siguiente:

- 1- que escuche el puerto 8080.
 - 2- Cuando entre al sitio web , se le solicite la autorización del directorio llamado /auth-proxy.
 - 3- Nos crea un nuevo location para ese directorio.
 - 4- Indica la URL de nuestro openldap.
 - 5- DN(nombre distinguido) sobre el en el que realizan las búsquedas de openldap.
 - 6- Que nos pida el usuario y contraseña para realizar las consultas.
- Ahora en la raíz crearemos nuestro docker-compose.yml.

```
david@david-VirtualBox:~/Escritorio/practica3Docker/conf$ cat docker-compose.yml
version: '2'

services:
  nginx-ldap:
    image: iamtopcat/nginx-ldap-auth-daemon
    ports:
      - 8888:8888
  nginx:
    image: bitnami/nginx
    ports:
      - 8080:8080
    volumes:
      - ./app:/app
      - ./conf/ldap_nginx.conf:/opt/bitnami/nginx/conf/server_blocks/ldap_nginx.conf
  openldap:
    image: bitnami/openldap
    ports:
      - '1389:1389'
    environment:
      - LDAP_ADMIN_USERNAME=admin
      - LDAP_ADMIN_PASSWORD=adminpassword
      - LDAP_USERS=david
      - LDAP_PASSWORDS=pruebal
```

Después hacemos docker-compose up para que active los contenedores y si todo esta correcto al hacer http://nuestra_ip:8080 deberíamos ver Hola mundo de nuestro index.php.



Despliegue amb Docker de PHP + Apache amb autenticació LDAP

Creamos un nuevo directorio llamado practica6.3 y creamos un index.php

```
david@david-VirtualBox:~/Escritorio/practica6.3$ cat index.php
<?php
echo 'Well, hello LDAP authenticated user!';
```

Ahora creamos otro directorio llamado Docker con un archivo llamado Dockerfile con esta estructura.

```

david@david-VirtualBox:~/Escritorio/practica6.3/Docker$ cat Dockerfile
# ./Docker/Dockerfile --> directorio donde está este archivo

# Imagen base sobre la que vamos a trabajar
FROM php:7-apache

# Activamos el módulo LDAP de Apache ejecutando el comando
RUN a2enmod authnz_ldap

# Añadimos las reglas/configuración de LDAP al directorio conf-enabled de Apache
# (crearemos este archivo en el siguiente paso)
COPY Docker/ldap-demo.conf /etc/apache2/conf-enabled/

# Añadimos ayuda de depuración (debugging) en la configuración de apache
# En caso de necesitarlo, lo descomentamos para ejecutar el siguiente comando
# ____ echo "LogLevel debug" >> apache2.conf

# Establecemos el directorio de trabajo adecuado
WORKDIR /var/www/html/demo

# Configuramos Apache para usar la configuración ldap definida arriba, la copiamos de nuestro ordenador al contenedor
COPY Docker/.htaccess ../.htaccess

# Copiamos los archivos del proyecto que necesitamos, al contenedor
COPY index.php ../

```

Dentro del mismo directorio creamos un archivo llamado `ldap-demo.conf` que es donde estableceremos todos los criterios cuando hagamos la conexión (usuario, password, openldap y url).

```

david@david-VirtualBox:~/Escritorio/practica6.3/Docker$ cat ldap-demo.conf
# ./Docker/ldap-demo.conf
PassEnv LDAP_BIND_ON
PassEnv LDAP_PASSWORD
PassEnv LDAP_URL
<AuthnProviderAlias ldap demo>
AuthLDAPBindDN ${LDAP_BIND_ON}
AuthLDAPBindPassword ${LDAP_PASSWORD}
AuthLDAPURL ${LDAP_URL}
</AuthnProviderAlias>

```

Ahora crearemos el archivo `.htaccess` que es un archivo que nos dará acceso cuando nos pida la contraseña y el usuario.

```

david@david-VirtualBox:~/Escritorio/practica6.3/Docker$ cat .htaccess
# ../.htaccess
AuthBasicProvider demo
AuthType Basic
AuthName "Protected Area"
Require valid-user
david@david-VirtualBox:~/Escritorio/practica6.3/Docker$

```

Volvemos a nuestro directorio de trabajo principal y ejecutamos el siguiente comando para construir la imagen

```
david@david-VirtualBox:~/Escritorio/practica6.3$ docker build \
-t docker-ldap \
-f ./Docker/Dockerfile \

[+] Building 27.3s (12/12) FINISHED          docker:default
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/php:7-apache
=> [internal] load metadata for docker.io/library/php:poll.token for registry-1.docker.io
=> [internal] load dockerignore
=> [internal] transfer context: 28
=> [1/6] FROM docker.io/library/php:7-apache:sha256:c9d7688f7383267347 23.9s
=> resolve docker.io/library/php:7-apache:sha256:c9d7688f7383267347 0.4s
=> sha256:a081c3e38412721050a0d1b0baf290e05278 31.49MB / 31.41MB 5.4s
=> sha256:c42871a04238052524a5957c5199c36c8b403305ee 2768 / 2260 0.4s
=> sha256:c9d7688f738326734777066aac0180011ec751d 1.86MB / 1.86MB 0.0s
=> sha256:1083497ee729949406c220bc345261120a0b07 3.04MB / 3.04MB 0.4s
=> sha256:185740867f6ac32f9770e4d5794ae5c41da37 91.43MB / 91.43MB 14.5s
=> sha256:2ba3732f422b7c386cf99e097f093abc135efca8 12.51MB / 12.51MB 0.8s
=> sha256:f654c8a82f60730b7427e47bd4776ee2a260ee4217 270B / 270B 0.7s
=> sha256:251850490f5b7fccc053132191e48085bdaa 19.25MB / 19.25MB 10.4s
=> extracting sha256:a081c3e38412721050a0d1b0baf290e05278 4.8s
=> sha256:9c236420ac70bc464508c2c13029762acf1742408c76 475B / 475B 5.2s
=> sha256:fe42347c4ecf90333acd9cad13012387ee39d13827a25 514B / 514B 5.4s
=> sha256:d14eb2ed1e17ae09f5fcb44b0562e2867c401c2 18.70MB / 18.70MB 10.4s
=> extracting sha256:c42871a04238052524a5957c5199c36c8b403305ee 0.8s
=> sha256:6080f73ac62a06c4c22f9e0d7c7e3a305f9c16171 491B / 491B 10.6s
=> sha256:d2c43c5fbc861f83ee5565c7102ca60d0f35e1 18.20MB / 18.20MB 14.6s
=> sha256:ab5904de476386d7b07c34debe77c72103c466 2.46MB / 2.46MB 10.4s
=> sha256:89d3902b7f336112c5690a297f60e726f6b1f0a 240B / 240B 11.8s
=> sha256:854655aa99c358d4acc4ade8f2084309009f31fe 892B / 892B 11.2s
=> extracting sha256:156740b07ef8a5270770e4d5794ae5c41da37 0.4s
=> extracting sha256:f654c8a82f60730b7427e47bd4776ee2a260ee42175 0.0s
=> sha256:251850490f5b7fccc053132191e48085bdaa 1.1s
=> extracting sha256:9c236420ac70bc464508c2c13029762acf1742408c765 0.0s
=> extracting sha256:fe42347c4ecf90333acd9cad13012387ee39d13827a25c 0.0s
=> extracting sha256:d14eb2ed1e17ae09f5fcb44b0562e2867c401c2372829e 0.1s
=> extracting sha256:89d3902b7f336112c5690a297f60e726f6b1f0a 0.0s
=> extracting sha256:d2c43c5fbc861f83ee5565c7102ca60d0f35e1 0.0s
=> extracting sha256:ab5904de476386d7b07c34debe77c72103c466ade90 0.0s
=> extracting sha256:89d3902b7f336112c5690a297f60e726f6b1f0a 0.0s
=> extracting sha256:854655aa99c358d4acc4ade8f2084309009f31fead2 0.0s
=> [internal] load build context
=> transferring context: 555B
=> [2/6] RUN apt-get update; apt-get install -y apache2
=> [3/6] COPY Docker/ldap-demo.conf /etc/apache2/conf-enabled/
=> [4/6] WORKDIR /var/www/html
=> [5/6] COPY Docker/.htaccess /
=> [6/6] COPY index.php ./
=> exporting to image
=> exporting layers
=> writing image sha256:a50f279c721fe0974de0d533bc81ea21c4cad0477420 0.0s
=> saving to docker.io/library/docker-ldap 0.0s
```

Vamos a comprobar que podemos acceder a la pagina web utilizando el siguiente comando(antes de ello comprobar que ningún contenedor que tenemos activo con el mismo puerto, puesto que nos puede dar problemas.)

```
david@david-VirtualBox:~/Escritorio/practica6.3$ docker run \
-p 3000:80 \
--name ldap_demo \
-e LDAP_BIND_ON='cn=read-only-admin,dc=example,dc=com' \
-e LDAP_PASSWORD='password' \
-e LDAP_URL='LDAP://ldap.forumsys.com/dc=example,dc=com' \
docker-ldap
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[Thu Feb 08 19:23:52.175179 2024] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.54 (Debian) PHP/7.4.33 configured -- resuming normal operations
[Thu Feb 08 19:23:52.175386 2024] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
```

Ahora para acceder debemos entrar en http://ip_nuestra:3000 y nos pedira nuestras credenciales que son usuarios posibles (read-only-admin,example,com) y password = password.

Desplegament de servidors web

Desplegament de servidors web amb usuaris autenticats mitjançant LDAP usant Docker i docker-compose

12 (13 de 13)

Tamaño automático

6. Dins del nostre directori de continguts, creem un fitxer de configuració de Dockerfile amb el següent contingut:

```
1 docker build -t ldap-test-server .
2 docker run --name ldap-test-server -p 38080:38080 -e LDAP_URL=ldap://ldap.forumsys.com/dc=example,dc=com -e LDAP_BIND_DN=cn=admin,dc=example,dc=com -e LDAP_BIND_PASSWORD=secret ldap-test-server
```

7. Correm el contenidor i tornem a l'aplicació web amb la flag --eureka.enabled=true per integrar el nostre servidor públic en Internet.

```
1 docker run --name ldap-test-server -p 38080:38080 -e LDAP_URL=ldap://ldap.forumsys.com/dc=example,dc=com -e LDAP_BIND_DN=cn=admin,dc=example,dc=com -e LDAP_BIND_PASSWORD=secret ldap-test-server
```

8. No ens queda més que anar a la nostra aplicació web i introduir les nostres credencials.

Tasca 3

Documenta tot el que has fet i els resultats pertinents.

1.5 Referències

- Para qué sirve el protocolo LDAP
- OpenLDAP conceptos básicos
- Using Nginx and NGINX
- Simple Docker/Apache
- bitnami/nginx-ldap-auth
- nginxinc/nginx-ldap-auth
- Dockerfile cheatsheet

500 Internal Server Error

10.2.0.222:3000/demol

Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator at webmaster@localhost to inform them of the time this error occurred, and the actions you performed just before this error.

More information about this error may be available in the server error log.

Apache/2.4.54 (Debian) Server at 10.2.0.222 Port 3000