
Práctica 6.3

Índice

Despliegue con Docker de NGINX + demonio de autenticación LDAP + OpenLDAP.....	3
Despliegue con Docker de PHP + Apache con autenticación LDAP.....	5

Despliegue con Docker de NGINX + demonio de autenticación LDAP + OpenLDAP

Primero creamos entorno de trabajo con un primer directorio llamado app que contendrá indexa.html

```
ec2-user@ip-172-31-29-56:~/app
[ec2-user@ip-172-31-29-56 app]$ cat index.html
<html>
<body>
<h1>¡Hola Mundo!</h1>
</body>
</html>
[ec2-user@ip-172-31-29-56 app]$
```

Después crearemos otro directorio con el contenido de configuración de Nginx

```
ec2-user@ip-172-31-29-56 app]$ cd conf/
ec2-user@ip-172-31-29-56 conf]$ nano ldap_nginx.conf
ec2-user@ip-172-31-29-56 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";

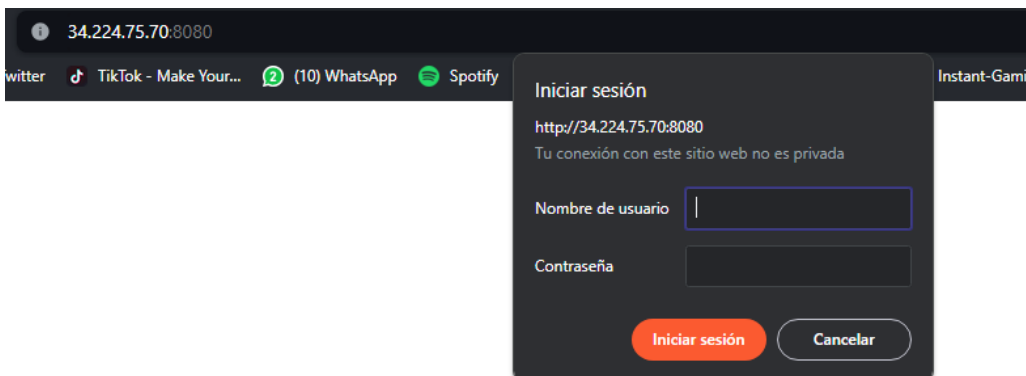
        # Bind password
        proxy_set_header X-Ldap-BindPass "adminpassword";
    }
}
ec2-user@ip-172-31-29-56 conf]$
```

Ahora hacemos el .yaml de la siguiente forma.

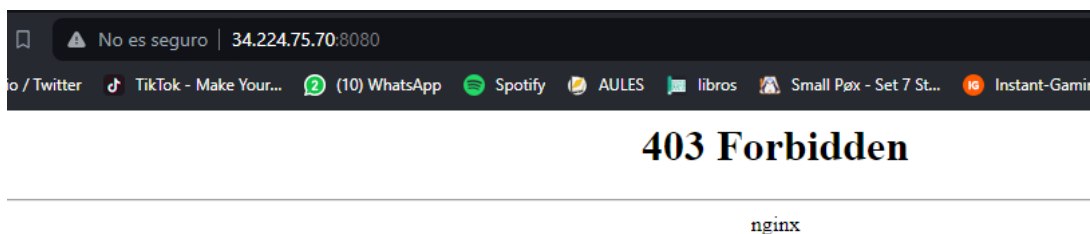
```
[ec2-user@ip-172-31-29-56 app]$ nano docker-compose.yml
[ec2-user@ip-172-31-29-56 app]$ cat docker-compose.yml
version: '2'

services:
  nginx-ldap: #
    image: bitnami/nginx-ldap-auth-daemon-archived #
    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=customuser
      - LDAP_PASSWORDS=custompassword
[ec2-user@ip-172-31-29-56 app]$
```

Por último desplegamos el contenedor con docker-compsoe up y podremos acceder.



Si accedemos con el usuario no tendremos privilegios y nos saldrá el siguiente mensaje



Despliegue con Docker de PHP + Apache con autenticación LDAP

Primero creamos entorno de trabajo con un primer directorio llamado Practica6.3 que contendrá index.php

```
[ec2-user@ip-172-31-29-56 app]$ mkdir Practica6.3
[ec2-user@ip-172-31-29-56 app]$ cd Practica6.3/
[ec2-user@ip-172-31-29-56 Practica6.3]$ nano index.php
[ec2-user@ip-172-31-29-56 Practica6.3]$ cat index.php
<?php
echo 'Ey, hola usuario autenticado por LDAP!';
[ec2-user@ip-172-31-29-56 Practica6.3]$
```

Ahora crearemos otro directorio llamado Docker y dentro el Dockerfile con el siguiente contenido.

```
[ec2-user@ip-172-31-29-56 Practica6.3]$ mkdir Docker
[ec2-user@ip-172-31-29-56 Practica6.3]$ cd Docker/
[ec2-user@ip-172-31-29-56 Docker]$ nano Dockerfile
[ec2-user@ip-172-31-29-56 Docker]$ cat Dockerfile
# ./Docker/Dockerfile --> directorio donde se encuentra este archivo

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

# Activamos el módulo LDAP de Apache ejecutand el siguiente 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
# RUN 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 ./
[ec2-user@ip-172-31-29-56 Docker]$
```

A continuación creamos el archivo ldap-demo.conf que establecen los criterios de conexión.

```
ec2-user@ip-172-31-29-56:~/app/Practica6.3/Docker
[ec2-user@ip-172-31-29-56 Docker]$ nano ldap-demo.conf
[ec2-user@ip-172-31-29-56 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 ${LDAP_URL}
</AuthnProviderAlias>
[ec2-user@ip-172-31-29-56 Docker]$
```

Después creamos el archivo .htaccess

```
[ec2-user@ip-172-31-29-56 Docker]$ nano .htaccess
[ec2-user@ip-172-31-29-56 Docker]$ cat .htaccess
# .Docker/.htaccess
AuthBasicProvider demo
AuthType Basic
AuthName "Protected Area"
Require valid-user
[ec2-user@ip-172-31-29-56 Docker]$
```

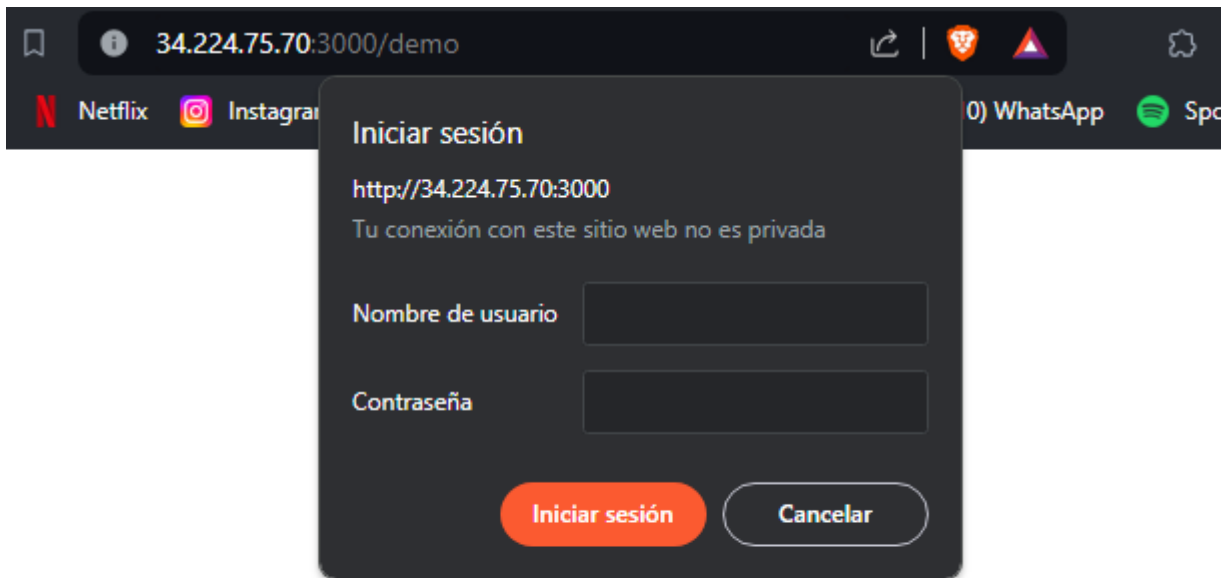
Para acabar, dentro de nuestro directorio de trabajo, construimos la imagen.

```
[ec2-user@ip-172-31-29-56 Practica6.3]$ sudo docker build . -t docker-ldap -f ./Docker/Dockerfile
[+] Building 13.6s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load .dockerignore
=> [internal] load metadata for docker.io/library/php:7-apache
=> [1/6] FROM docker.io/library/php:7-apache@sha256:c9d7e688f73832673479770d66aacc8108011ec751d1905ff63fae3fe2e0ca6d
=> resolve docker.io/library/php:7-apache@sha256:c9d7e688f73832673479770d66aacc8108011ec751d1905ff63fae3fe2e0ca6d
=> sha256:a693fa5e3b4127f210583aaa6189abf6286ee5a73deeaab460f8f33ebc6b64e2 31.41MB / 31.41MB
=> sha256:18b3497ee7f7099a90b66c23a0bc3d5261b17bab367263e1b40e9b004c39e882 3.04kB / 3.04kB
=> sha256:156740b07ef8a632f9f7bda4e57e4ee5541ade376ad9169351a1265382e39de 91.63MB / 91.63MB
=> sha256:c9d7e688f73832673479770d66aacc8108011ec751d1905ff63fae3fe2e0ca6d 1.86kB / 1.86kB
=> sha256:20a3732f422b7b28dcf99e8597f093a8c135efca62f0dc02a2d92916369413 12.51kB / 12.51kB
=> sha256:c428f1a49423085254a2a5957cc5199c36c8b403305e0e877d580bd0ec9e763 226B / 226B
=> sha256:fb5a4c8af82f00730b7427e47bda7f76cea2e2b9aea421750bc9025aface98d8 270B / 270B
=> sha256:25f85b498fdb5fc6cce951513219fe480850daba71e6e997741e984d18483971 19.25MB / 19.25MB
=> sha256:9b233e420ac7b0ca645bb82c213029762acf1742400c076360dc303213c309d5 475B / 475B
=> extracting sha256:a603fa5e3b4127f210583aaa6189abf6286ee5a73deeaab460f8f33ebc6b64e2
=> sha256:fe42347c4ecfc90333acd9cad13912387eea39d13827a25cfa78727fa5d200e9 514B / 514B
=> sha256:d14eb2ed1e17ae00f5fcb44b0d562e2867c401c20372829e2cf443fc409342fa 10.76MB / 10.76MB
=> sha256:66d98f73ac62e86c0c226f9eedcbc7eda305df0c1e171ca5caf81cb8b1c40cb 491B / 491B
=> sha256:d2c43c5efbc861f83ee6565c7102ca660d6f35e158324fbb042de5017e43afe8 10.20MB / 10.20MB
=> sha256:ab590b48ea476386dd7b07c34de9eff7cf2103c4668ade985fe31e59f15deee8 2.46kB / 2.46kB
=> sha256:80692ae2d067c8358112c56490a2a97f69ef395f08f7662a31490644c9a813ef 246B / 246B
=> sha256:05e465aaa99a358add4acecdade8f39843089069f31fea0201533d3a00a98c9a 892B / 892B
=> extracting sha256:c428f1a49423085254a2a5957cc5199c36c8b403305e0e877d580bd0ec9e763
=> extracting sha256:156740b07ef8a632f9f7bda4e57e4ee5541ade376ad9169351a1265382e39de
=> extracting sha256:fb5a4c8af82f00730b7427e47bda7f76cea2e2b9aea421750bc9025aface98d8
=> extracting sha256:25f85b498fdb5fc6cce951513219fe480850daba71e6e997741e984d18483971
=> extracting sha256:9b233e420ac7b0ca645bb82c213029762acf1742400c076360dc303213c309d5
=> extracting sha256:fe42347c4ecfc90333acd9cad13912387eea39d13827a25cfa78727fa5d200e9
=> extracting sha256:d14eb2ed1e17ae00f5fcb44b0d562e2867c401c20372829e2cf443fc409342fa
=> extracting sha256:66d98f73ac62e86c0c226f9eedcbc7eda305df0c1e171ca5caf81cb8b1c40cb
=> extracting sha256:d2c43c5efbc861f83ee6565c7102ca660d6f35e158324fbb042de5017e43afe8
=> extracting sha256:ab590b48ea476386dd7b07c34de9eff7cf2103c4668ade985fe31e59f15deee8
=> extracting sha256:80692ae2d067c8358112c56490a2a97f69ef395f08f7662a31490644c9a813ef
=> extracting sha256:05e465aaa99a358add4acecdade8f39843089069f31fea0201533d3a00a98c9a
=> [internal] load build context
=> transferring context: 800B
=> [2/6] RUN a2enmod authnz_ldap
=> [3/6] COPY Docker/ldap-demo.conf /etc/apache2/conf-enabled/
=> [4/6] WORKDIR /var/www/html/demo
=> [5/6] COPY Docker/.htaccess ./htaccess
=> [6/6] COPY index.php ./
=> exporting to image
=> exporting layers
=> writing image sha256:87871af4b8c8e612dc322389f516bd2a98b87cddb137ef64f66028f4bbb162c5
=> naming to docker.io/library/docker-ldap
=> pushing to docker.io/library/docker-ldap
[ec2-user@ip-172-31-29-56 Practica6.3]$
```

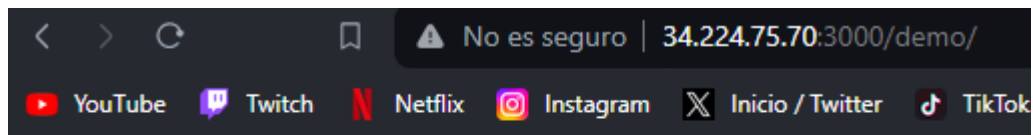
Y por último corremos el contenedor indicando las credenciales de nuestra cuenta LDAP.

```
[ec2-user@ip-172-31-29-56 Practica6.3]$ sudo docker run \
> -p 3000:80 \
> --name ldap_demo \
> -e LDAP_BIND_ON='uid=admin,cn=users,cn=accounts,dc=demo1,dc=freeipa,dc=org' \
> -e LDAP_PASSWORD='Secret123' \
> -e LDAP_URL='LDAP://ipa.demo1.freeipa.org' \
> docker-ldap
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' direct
ive 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' direct
ive globally to suppress this message
[Sun Feb 11 19:41:44.303581 2024] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.54 (Debian) PHP/7.4.33 configured -- resuming norma
l operations
[Sun Feb 11 19:41:44.303982 2024] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
45.86.187.182 - - [11/Feb/2024:19:42:41 +0000] "GET /demo HTTP/1.1" 401 729 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit
/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36"
```

Para comprobar que funciona nos conectamos a la ip pública de la máquina en el puerto 3000 y nos logueamos.



Como podemos observar, todo funciona correctamente.



Ey, hola usuario autenticado por LDAP!