SA06AAQ6.2 Containerització d'aplicació amb servici LDAP

Index

1. Introducció

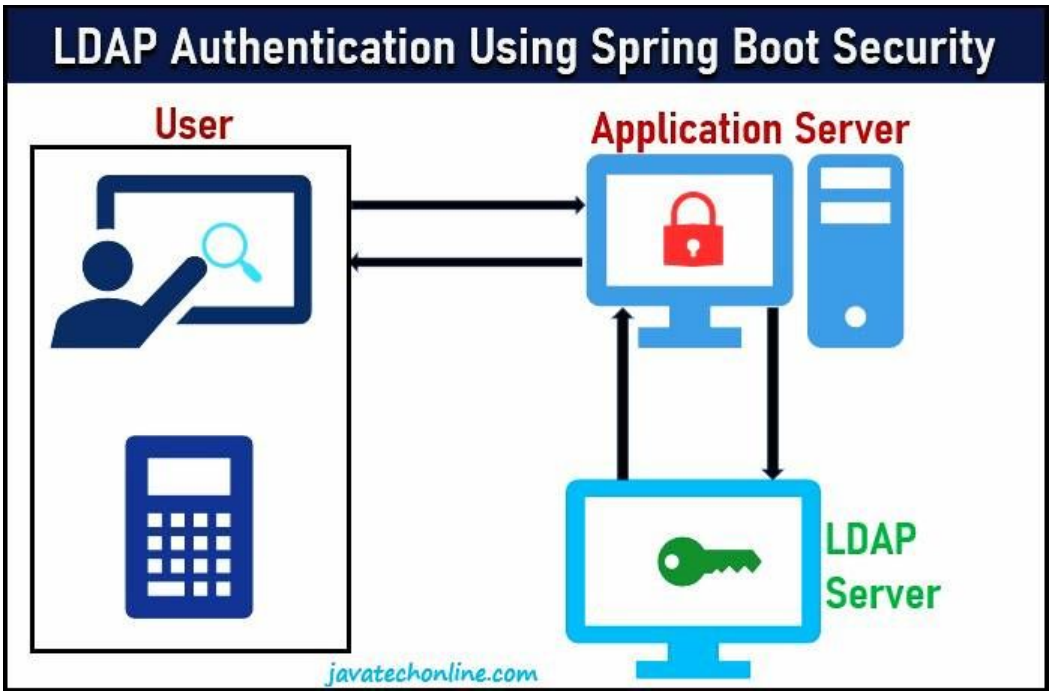
Imagina una **empresa que té diversos serveis web** (com aplicacions internes, portals de clients, etc.). Un opció és gestionar de forma individual els usuaris de cada aplicació, però això generaria un munt de problemes amb la gestió d'usuaris i els permisos de cadascun d'ells. A més a més, aquests usuaris haurien d'establir les contrasenyes de forma independent per cada aplicació, en lloc d'això, les empreses poden utilitzar LDAP com a punt central d'autenticació d'usuaris.

Per fer-ho, l'administrador configura un servidor LDAP amb les dades de tots els empleats. Cada aplicació, com el portal web de l'empresa, la intranet,... consulta aquest servidor per autenticar els usuaris. D'aquesta forma, quan un usuari inicia sessió, el servidor web pregunta al servidor LDAP: Qui és aquest usuari? És un administrador o un usuari normal? I el servidor web decideix què pot fer l'usuari segons el seu rol.

2. LDAP (Lightweight Directory Access Protocol)

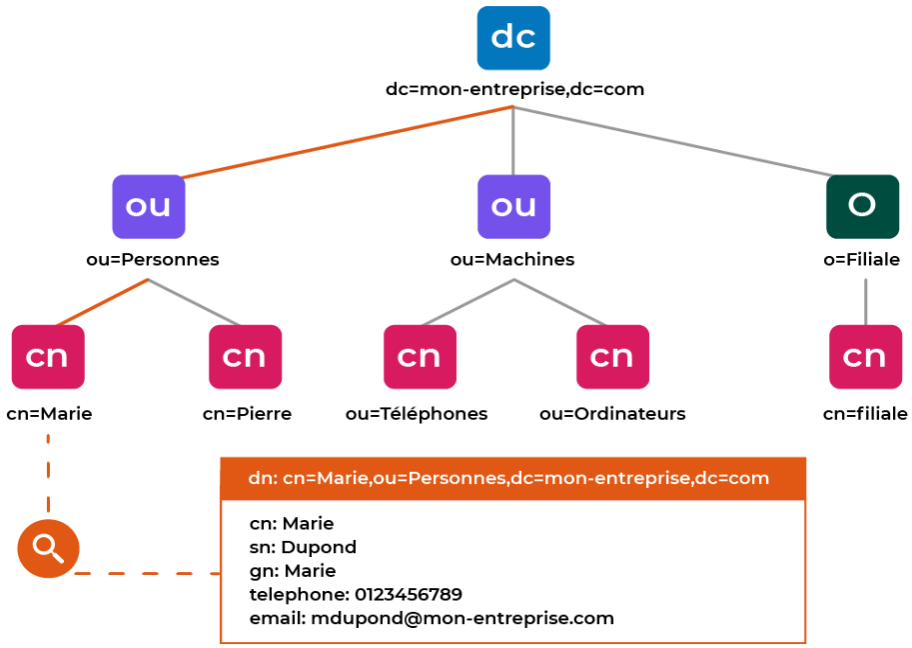
LDAP (Lightweight Directory Access Protocol) és un protocol utilitzat per accedir i gestionar serveis de directori distribuïts. **Un directori és una base de dades** optimitzada per a la lectura, que s'utilitza sovint p**er emmagatzemar informació d'usuaris**, grups, permisos, i altres recursos d'una xarxa.

En el context d'una xarxa informàtica, LDAP actua com un mecanisme centralitzat d'autenticació. Això significa que qualsevol aplicació o servei que necessite verificar la identitat dels seus usuaris pot consultar un servidor LDAP per validar credencials i obtenir informació relacionada amb permisos o grups.



2.1 Components principals de LDAP

La informació en LDAP s'**estructura** de forma **jeràrquica**, com el servici DNS, a aquesta estructura l'anomenem **arbre LDAP.**



1. **Base DN (Distinguished Name)**

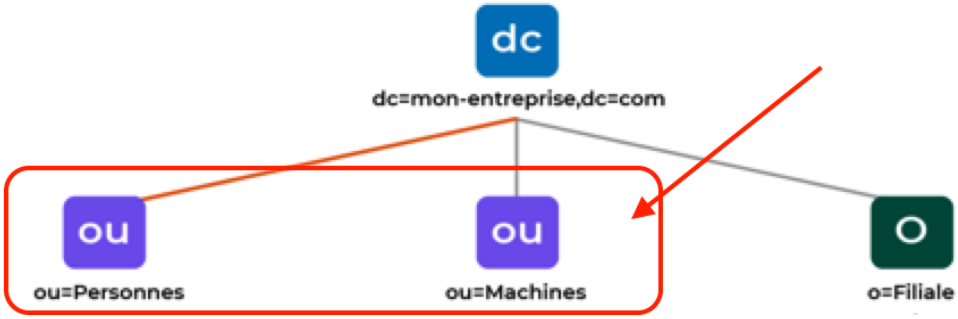
   La **Base DN** defineix el punt inicial de l'arbre. És com l'arrel d'un directori de fitxers en un

sistema operatiu. Aquest pot tenir diferents components de domini **dc**: Al exemple anterior tenim **2 components de domini** `mon-enterprise` i `com`, de manera que estem identificant a la empresa com a `mon-enterprise.com`

2. Unitats Organitzatives (OU)

Les **Unitats Organitzatives (Organizational Units)** són contenidors que agrupen objectes relacionats. Són similars a carpetes dins d'un sistema de fitxers. Cada **OU** pot contindre altres **OU** o entrades individuals.

- Exemple:
  - `ou=Personnes`: Conté tota la informació dels usuaris.
  - `ou=Machines`: Conté tota la informació de les Màquines.



3. Entrades Individuals (Entries)

Les entrades representen els objectes concrets, com usuaris, grups, dispositius, etc. Cada entrada té:

- Un **DN (Distinguished Name):** Nom únic que el identifica dins de l'arbre, està **compost** per totes les **entrades** que van **desde l'element fins** a la **base DN**.
- Un **conjunt d'atributs** que descriuen l'objecte.

```
dn: uid=user1,ou=Personnes,dc=mon-enterprise,dc=com
objectClass: inetOrgPerson
uid: user1
cn: Marie
sn: Dupond
userPassword: user1password
mail: user1@mon-enterprise.com
```

4. Cada element té atributs

Cada element de l'arbre LDAP té **atributs** que descriuen informació específica sobre ell. Els atributs són com propietats d'un fitxer (nom, data de creació, etc.) i no poden ser qualsevol cosa, **estan definits en esquemes (schemas)** que determinen quins atributs es poden utilitzar, quins tipus de dades poden contindre, i amb quins objectes estan associats. Però també és possible definir atributs personalitzats si necessites alguna cosa específica.

Cada esquema conté:

- **Atributs:** Descriuen dades específiques (per exemple, nom, correu electrònic, contrasenya).
- **ObjectClass:** Defineixen quin tipus d'objecte és una entrada, es a dir, quins atributs són obligatoris i quins són opcionals. Per exemple, per definir un usuari, s'utilitza el **objectClass** `inetOrgPerson` que conté els següents atributs.
  - **Obligatoris:** `sn` (cognom), `cn` (nom complet).
  - **Opcionals:** `uid`, `mail`, `telephoneNumber`, `description password`.

> Podem consultar tots els atributs que definieixes una persona d'una organització en la RFC corresponent

2.2 Funcionament Bàsic

1. Un usuari introdueix les seues credencials (nom d'usuari i contrasenya).
2. L'aplicació web fa una connexió amb el servidor LDAP i intenta autenticar l'usuari amb les credencials proporcionades.
3. Si l'autenticació és correcta, el servidor LDAP pot proporcionar informació addicional, com els grups o rols als quals pertany l'usuari.
4. Amb aquesta informació, l'aplicació decideix quins recursos o funcionalitats estan disponibles per a l'usuari.

3. Exemple contenerització aplicació amb Servei LDAP

Per a exemplificar **la posada en marxa i configuració d'un servei LDAP i la seu utilització través d'una aplicació Web farem servir els següents serveis:**

Aquesta proposta utilitza:

- **OpenLDAP (osixia/openldap):** Per al servidor LDAP.
- **php:apache:** Per executar l'aplicació web amb suport LDAP.

- **phpLDAPadmin:** Per administrar LDAP gràficament.

1. Creació del directori de Treball

En primer lloc creem el directori de treball `ldap-example` i crearem les següents carpetes i fitxers:

```
ldap-example
     |------docker-compose.yml
     |------Dockerfile
     |------web/
         |--------index.php
```

2. Creació del fitxer `docker-compose.yml`

docker-compose.yml

```
services:
  ldap:
    image: osixia/openldap:1.5.0
    container_name: ldap_server
    environment:
      LDAP_ORGANISATION: "MyOrganization"
      LDAP_DOMAIN: "example.com"
      LDAP_ADMIN_PASSWORD: "admin_password"
    ports:
      - "389:389"
      - "636:636"
  phpldapadmin:
    image: osixia/phpldapadmin:latest
    container_name: phpldapadmin
    environment:
      - PHPLDAPADMIN_LDAP_HOSTS=ldap
      - PHPLDAPADMIN_HTTPS=false
    ports:
      - "8080:80"
    depends_on:
      - ldap

  web:
    build:
      context: .
      dockerfile: Dockerfile
    container_name: php_web
    volumes:
      - ./web:/var/www/html
    ports:
      - "80:80"
    depends_on:
      - ldap
```

Activitat 1

- Indica els diferents serveis declarats en el fitxer `docker-compose.yml` i la funció que realitzarà dins de la infraestructura configurada.

3. Creació del fitxer Dockerfile per a la containerització de l'aplicació web

```
FROM php:8.2-apache

# Instal·la l'extensió LDAP
RUN apt-get update && apt-get install -y libldap2-dev && docker-php-ext-configure ldap --with-libdir=lib/x86_64-linux-gnu && docker-php-ext-install ldap

# Habilita els mòduls necessaris d'Apache
RUN a2enmod rewrite

# Copia el codi PHP al contenidor
COPY ./web /var/www/html

# Estableix permisos per al directori web
RUN chown -R www-data:www-data /var/www/html
```

Activitat 2

- Explica la funcionalitat que du a terme cada una de les lines del fitxer Dockerfile anterior.

4. Implementa el fitxer de prova `index.php` amb el següent codi que ens permet autenticar-nos fent ús del servidor LDAP

index.php

```
<?php
$ldap_host = "ldap";
$ldap_port = 389;
$ldap_user = "cn=admin,dc=example,dc=com";
$ldap_pass = "admin_password";
$ldap_base_dn = "dc=example,dc=com";
```

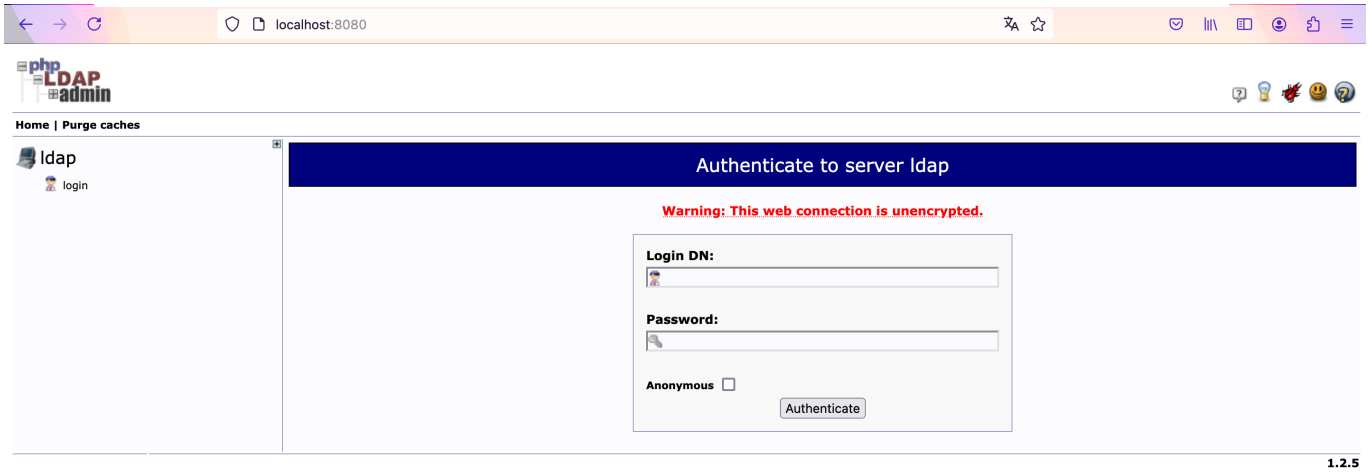
```
$ldap_conn = ldap_connect($ldap_host, $ldap_port);
ldap_set_option($ldap_conn, LDAP_OPT_PROTOCOL_VERSION, 3);

if (!$ldap_conn) {
    die("Error: No es pot connectar al servidor LDAP.");
}

if (@ldap_bind($ldap_conn, $ldap_user, $ldap_pass)) {
    echo "<h2>Connexió establerta!</h2>";

    // Cerca simples
    $result = ldap_search($ldap_conn, $ldap_base_dn, "(objectClass=*)");
    if ($result) {
        $entries = ldap_get_entries($ldap_conn, $result);
        echo "<pre>";
        print_r($entries);
        echo "</pre>";
    } else {
        echo "No s'han trobat resultats a LDAP.";
    }
} else {
    die("Error: No es pot autenticar amb el servidor LDAP.");
}
ldap_unbind($ldap_conn);
```

5. Compila la imatge e Inicia els serveis

```
$ docker compose build
$ docker compose up
```

6. **Accés a la interfície per a configurar el servidor de LDAP**

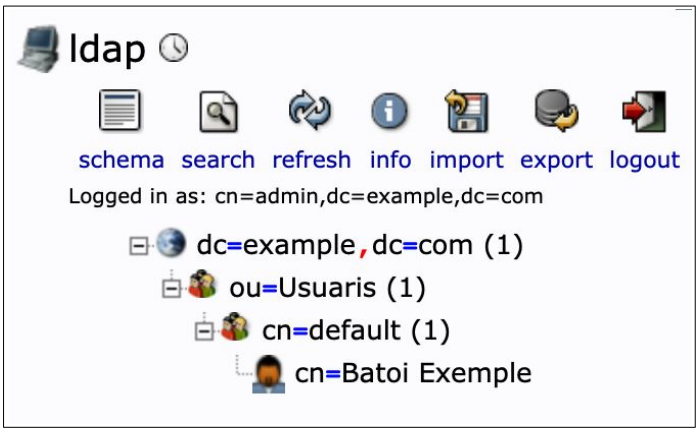
Una vegada tenim els serveis en marxa, podem accedir aa la interfície gràfica per a configurar o veure la informació del nostre servidor de ldap a través de la direcció http://127.0.0.1:8080.



Utilitzarem com **usuari** el definit en el servici ldap: "`cn=admin,dc=example,dc=org`" i **contrasenya** "`admin_password`"

Activitat 3

- Crea una unitat organitzativa anomenada "usuaris" i crea un usuari per cada membre del grup que esteu fent la pràctica. Per això hauràs de:
  - Crear la OU usuaris
  - Crear un Grup Posix anomenat 'default' que pertanya a la organització (això hem de fer-ho perqué l'esquema per defecte d'usuaris té el atribut gid definit com a requerit
  - Crear els usuaris associats al grup POSIX i amb el shell "No login"



C/ Societat Unió Musical, 8 - 03802 Alcoi
Tel. 966 52 76 60 - Fax 966 52 76 61
03012165.secretaria@edu.gva.es
www.cipfpbatoi.es

Activitat 4

- **4.1.** Modifica els arxius de configuració de l'arquitectura per a que el servidor LDAP faça ús d'una contrasenya segura per a l'administrador per a la connexió del administrador i la organització base siga "`ddaw.com`"
- **4.2.** Modifica l'aplicació web per a que mostre la informació del usuaris en format tabla.
- **4.3.** Explica el procés que seguiries per a autenticar els usuaris fent ús del servici LDAP que acabes de configurar.
- **4.5.** ¿Com podem donar d'alta els rol que té cada usuari? Busca informació al respecte

4. Bibliografia / Webgrafía

- PHP ldap. https://www.php.net/manual/es/book.ldap.php
- Docker phpldapadmin image. https://github.com/osixia/docker-phpLDAPadmin
- Docker ldap server image. https://github.com/osixia/docker-openldap