Spring Ldap

https://spring.io/projects/spring-ldap

https://spring.io/guides/gs/authenticating-ldap

https://www.baeldung.com/spring-ldap

https://www.baeldung.com/spring-ldap

LightWeightDirectoryAccessProtocol, its main purpose is for authentication.

Application protocol for accessing and maintaining distributed directory information services over an Internet Protocol Network.

Mostly used to store organisational information because there is hierarchy of information. It also helps in maintaining user information which is used for authentication and authorisation.

Directory for users to validate their credentials.

If the internet application is used by countable people so instead of using role based authentication using database we should we use ldap security.

We can use for the common data being shared.

Ldap is faster than rdbms but cannot handle huge data.

So we need to connect to ldap server to get the information and authenticate against ldap.
This is made easy by spring security with out of the box classes which we need to configure.

We connect to already existing ldap server and create our own.

Ldap is a file with extension .ldif where we can configure username, password, domain, organisation, pattern instead of in database.

Add dependency of spring-ldap-core (spring integration library to work with ldap), spring-security-ldap(needed for security with ldap), unboundid-ldapsdk (open source implementation of ldap to setup local instance of ldap)

Go to application.properties and add

Spring.ldap.embedded.port=8389 (local instance to run)

// mention the file which contains seeded data for the ldap

Spring.ldap.embedded.ldif=classpath:ldap-data.ldif (ldif stands for ldap data interchange format -> syntax for working with ldap)
// it is equivalent to sql, it helps in representing data in a text format what is there in tree structure (hierarchy of data)

Now put .ldif file under resources folder.

Copy the data to populate (mention in this file) in the instance from https://spring.io/guides/gs/authenticating-ldap

This sets up the user data. It contains ldif syntax.

e.g. of object class organisational person

dn: uid=joe,ou=otherpeople,dc=springframework,dc=org
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Joe Smeth
sn: Smeth
uid: joe
userPassword: joespassword

This belongs to organisational unit -> otherpeople and has unique id = Joe and person name is Joe Smith

And has password

We have to build the security app to look at the ldap instance so that we can authenticate people

Now we have to mention the property
This tells the ldap what the root node is

spring.ldap.embedded.base-dn=dc=springframework,dc=org

So first node is org and then springframework and everything below this.

This belongs to the root node specified in ldif file

In the SecurityConfigure class which extends WebSecurityConfigurerAdapter

```
@Override
Public void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.ldapAuthentication().userDnPatterns("uid={0},ou=people").groupSearchBase("ou=groups") // we are saying organisational  unit to search is groups
        .contextSource().url("ldap://localhost:8389/dc=springframework,dc")
        .and().passwordCompare().passwordEncoder(new LdapShaPasswordEncoder())
        .passwordAttribute("userPassword");
}
```

Dn stands for distinguish name -> the way user information is stored in ldif format

Dn format looks like uid=joe,ou=otherpeople,dc=springframework,dc=org

So we want to fill the uid with any person who want to authenticate, so then we can find in ldap and authenticate