# Authentication with LDAP server

## Server set-up:

Start by launching the docker-compose containing OpenLDAP and PhpMyLDAP

docker-compose -p LDAP up

This docker-compose link the docker "osixia/openldap:1.3.0" with the docker "osixia/phpldapadmin:latest". Some environment variables that you should modify to suits your configuration are set up in this compose :

* LDAP\_ORGANISATION: "MyCompany"
* LDAP\_DOMAIN: "mycompany.lan"
* LDAP\_ADMIN\_PASSWORD: "AdminPassword"

All thoose variables are useed in the LDAP server itself. For the phpldapadmin part you dont need to modify the environment variables.

The ports used for the docker are :

* 389 for the connexion to the LDAP server
* 636 for the connexion to the LDAP server through TLS/SSL
* 80 for connexion to the PhpLDAPadmin interface (forwarded to the port 8080 of your computer)

Once the dockers are set-up and started you can access to the LDAP database interface with the URL :

[http://localhost:8080](http://localhost:8080/)

You can now access the control panel with the login DN :

* cn=administrator,dc=mycompany,dc=lan

And the password :

* AdminPassword

When successfully loged into the administration panel start by adding 2 "Generic: Organisational Unit"

* People
* Groups

Then add as many "Generic: Posix Group" (like emlpoyee, trainee, development, administration, ...) as you need as child item of ur organisational unit "Groups".

And finaly create your "Generic: User Account" as child items of your organisational unit "People" (linked to the rightfull group in the "GID Number" section)

## UBUNTU 19.04 Client set-up:

To configure your client to be able to authetify with your LDAP server you will need to install in your client:

* libnss-ldap
* libpam-ldap
* ldap-utils
* nscd

sudo apt update -y

sudo apt install libnss-ldap libpam-ldap ldap-utils nscd

At the end of the installation the LDAP connexion config screen appears and this is how you should configure it :

* LDAP server URI: [ldap://Ip-of-your-ldap-server:389](ldap://Ip-of-your-ldap-server)
* Distinguish Name: dc=mycompany,dc=lan
* LDAP version: 3
* Make local root Database: (usually) YES
* Does the LDAP database require login: (usually) NO
* LDAP account for root: cn=administrator, dc=mycompany,dc=lan
* LDAP root account password: AdminPassword

After the configuration of the connexion to your LDAP server you need to modify some config files to be able to log with your LDAP accounts.

/etc/ldap.conf

Here is the config file for the connexion to the LDAP server. You should see in this file the informations you wrote in the LDAP connexion config screen.

NOTE : If you made mistakes at the previous step do not modify this file, use dpkg-reconfigure to re set your configuration.

Uncomment line 43 and replace generic informations with your company's :

binddn cn=administrator,dc=mycompany,dc=lan

Close and save the file.

/etc/nsswitch.conf

Here you will specify to your machine that it should interogate the LDAP database when trying to login :

Add at line 7 and 8 "ldap" at the end of the lines :

passwd: files systemd ldap

group: files systemd ldap

Close and save the file.

/etc/pam.d/common-password

Just remove [use\_authtok] from lone 26

password [success=1 user\_unknowk=ignore default=die] pam\_ldap.so use\_authtok try\_first\_pass

Close and save the file

/etc/pam.d/common-session

Here you set-up your users directoryies when loging in their accounts.

Add this line at the end of the file :

session optional pam\_mkhomedir.so skel=/etc/skel umask=077

/etc/sudoers

Here you can add any LDAP group to the sudoers group.

Just add this line for each group you want as sudoers :

%group\_you\_want\_as\_sudoers ALL=(ALL:ALL)ALL

Close and save the file (use :wq! command to ignore read only attribute)

You can now log into your computer with your LDAP login and password.

If you want to change your password just use passwd once loged into your session.

## Connexion Jorani LDAP :

In order to link your LDAP account with your Jorani account you need first to make sure LDAP plugin for PHP is installed in your Jorani server.

Once it's done, modify your configuration file to accept LDAP

/var/www/html/application/config/config.php

Modify the following lines :

$config['ldap\_enabled']= FALSE;

$config['ldap\_host']= '127.0.0.1';

$config['ldap\_port']= 389;

$config['ldap\_basedn']= 'uid=%s,ou=people,dc=company,dc=com'

$config['ldap\_basedn\_db']= FALSE;

Into :

$config['ldap\_enabled']= TRUE;

$config['ldap\_host']= 'Ip-of-your-LDAP-server'

$config['ldap\_port']= 389;

$config['ldap\_basedn']= 'cn=%s,ou=People,dc=mycompany,dc=lan

$config['ldap\_basedn\_db']= TRUE;

Close and save the file.

Now that your Jorani server is correctly set up to accept LDAP query you have to link your Jorani accounts to your LDAP accounts.

Go to your Jorani database administration panel and fill the 'ldap\_path' category for each member of your database.

Exemple : Let say that you have in your Jorani database an user named "Toto Tartenpion". His DN in your LDAP database is "cn=toto tartenpion,ou=People,dc=korys,dc=io" (wich you can find in this user's page in your LDAP admin panel).

You should write in the "ldap-path" section "cn=toto tartenpion,ou=People,dc=korys,dc=io".

You can, if you want, change the login of the user in your Jorani database to match the LDAP database.

In this configuration your users can log into Jorani with their Jorani login and their LDAP password and also their Jorani password. You can deactivate the Jorani password by simply removing the entry password in your Jorani database. Your user will still be able to log with his LDAP password but only with it.