

IAM Assignment

OpenLDAP

Done by

Aravind Kumar (MSc. Computer Security)

OpenLDAP

It is a free, open-source implementation of the Lightweight Directory Access Protocol (LDAP) developed by the OpenLDAP Project. It is released under its own BSD-Style license called OpenLDAP Public License.

LDAP is a platform-independent protocol. Several common Linux distributions include OpenLDAP software for LDAP support.

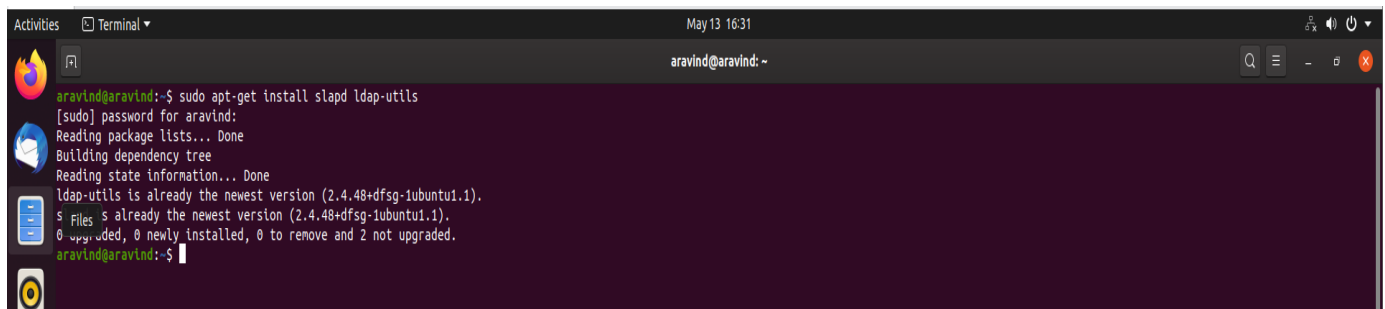
The following are the steps which we are going to follow for installing the OpenLDAP in Ubuntu OS and configure it.

Steps for installation

Step 1

Install LADP using the following command

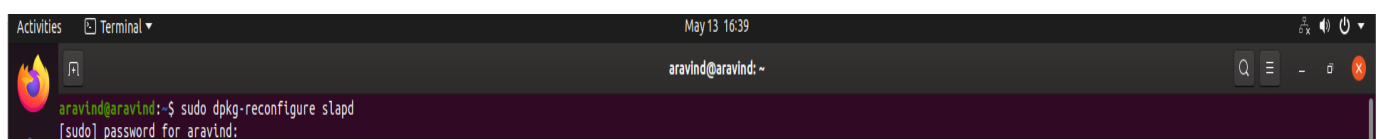
Command - `sudo apt-get install slapd ldap-utils`

A terminal window titled 'Terminal' with a timestamp of 'May 13 16:31'. The user 'aravind@aravind' is at the prompt. The command 'sudo apt-get install slapd ldap-utils' is entered. The terminal shows the following output: '[sudo] password for aravind:', 'Reading package lists... Done', 'Building dependency tree', 'Reading state information... Done', 'ldap-utils is already the newest version (2.4.48+dfsg-1ubuntu1.1).', 's files s already the newest version (2.4.48+dfsg-1ubuntu1.1).', '0 files to be removed, 0 newly installed, 0 to remove and 2 not upgraded.', and the prompt 'aravind@aravind:~\$'.

Step 2

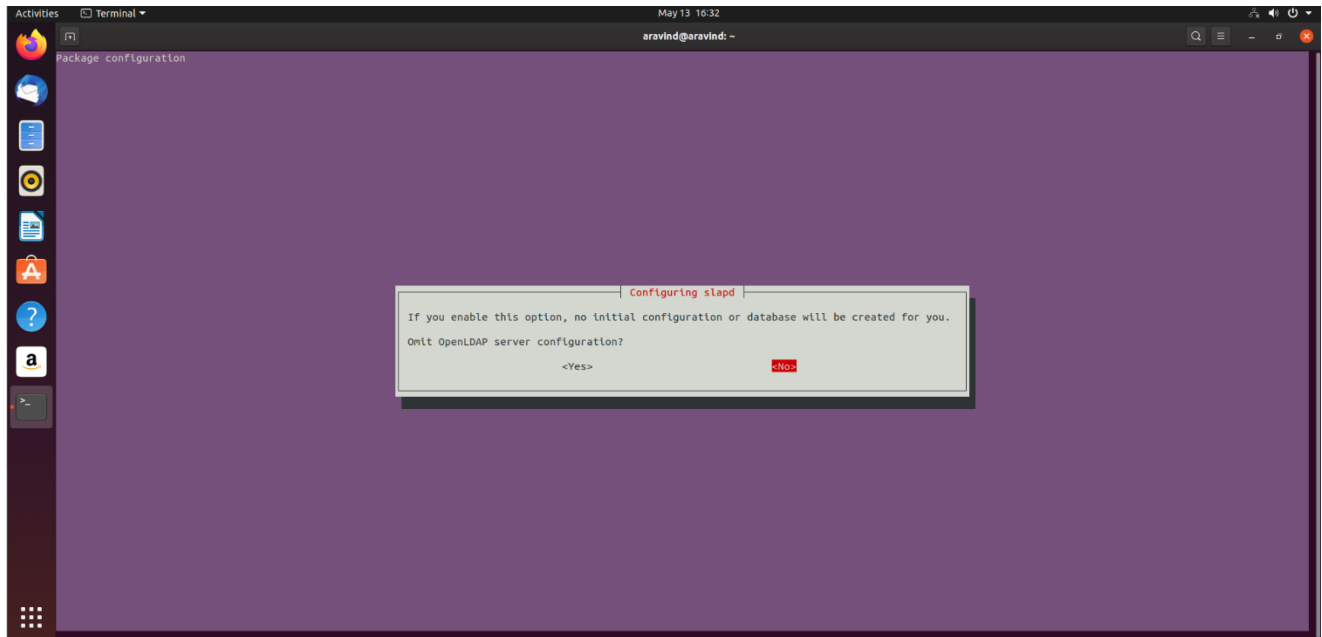
- Now we are going to reconfigure the package even though we just installed it.
- The slapd package has the ability to ask a lot of important configuration questions, but by default they are skipped over in the installation process.

Command - `sudo dpkg-reconfigure slapd`

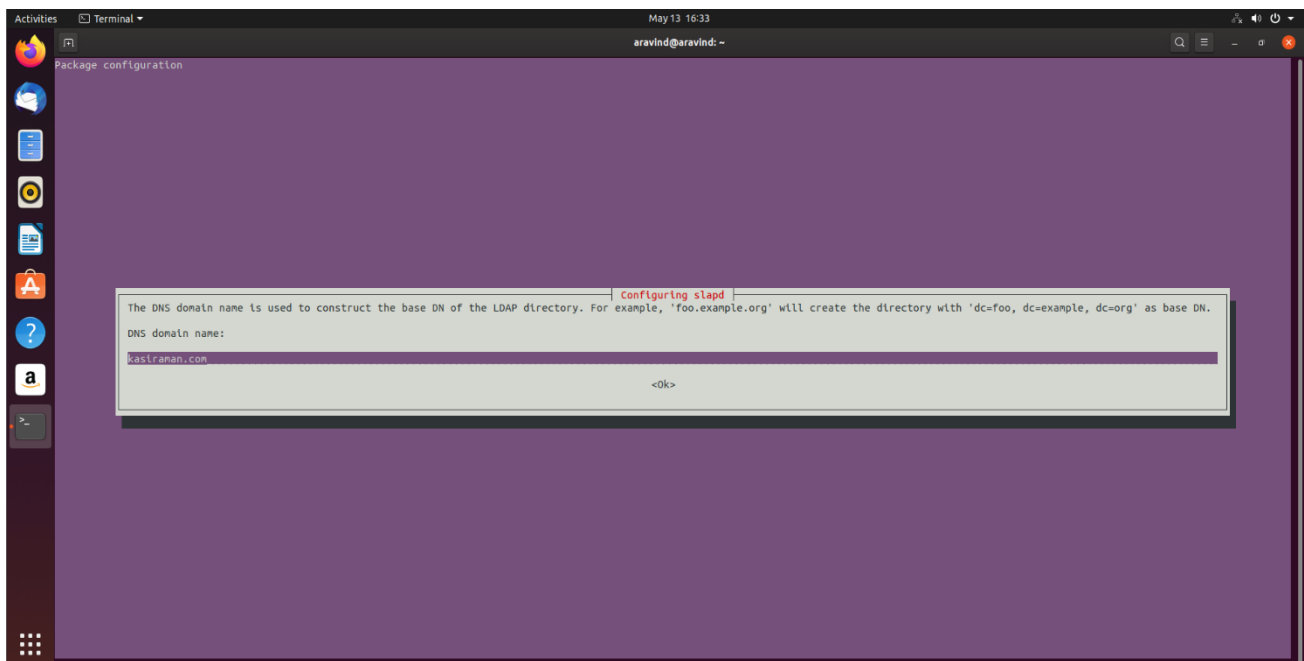
A terminal window titled 'Terminal' with a timestamp of 'May 13 16:39'. The user 'aravind@aravind' is at the prompt. The command 'sudo dpkg-reconfigure slapd' is entered. The terminal shows the following output: '[sudo] password for aravind:'.

There will be new screen with questions we need to answer for this process.

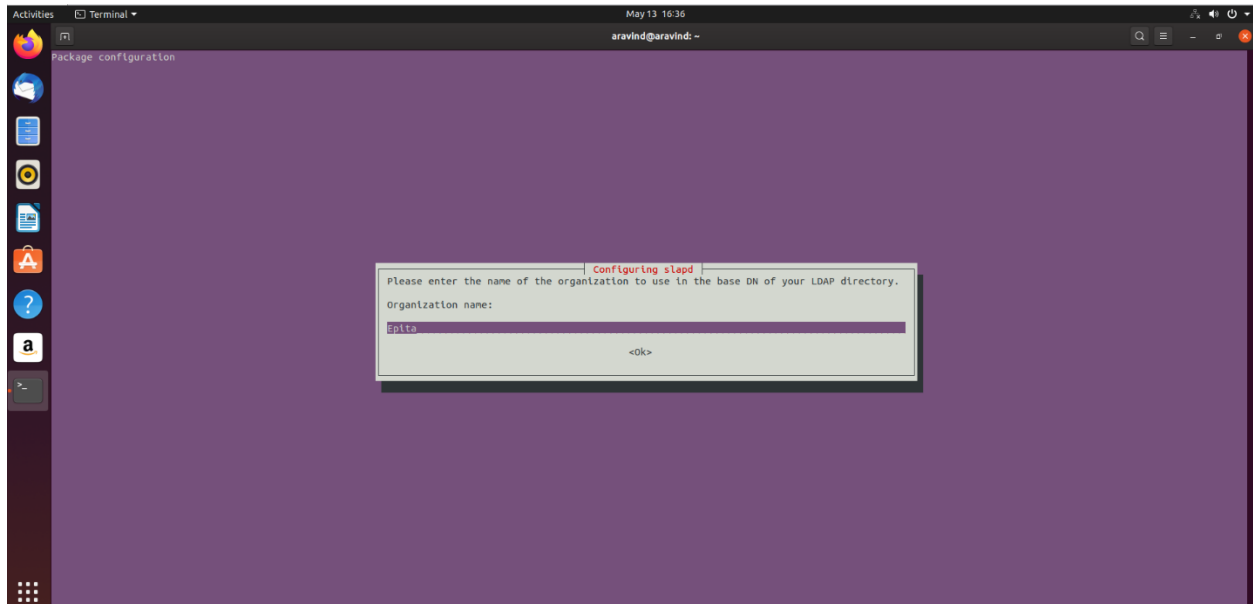
➤ **Omit OpenLDAP server configuration? – No**



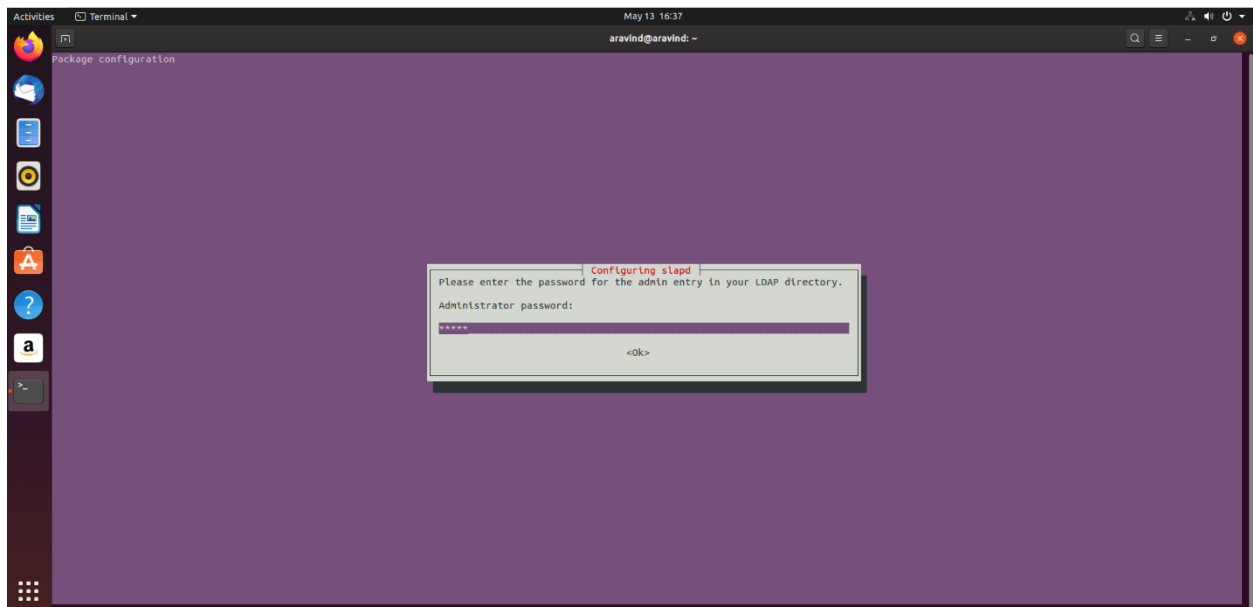
➤ **DNS domain name – give your last_name (as per my project) or any name of your choice**

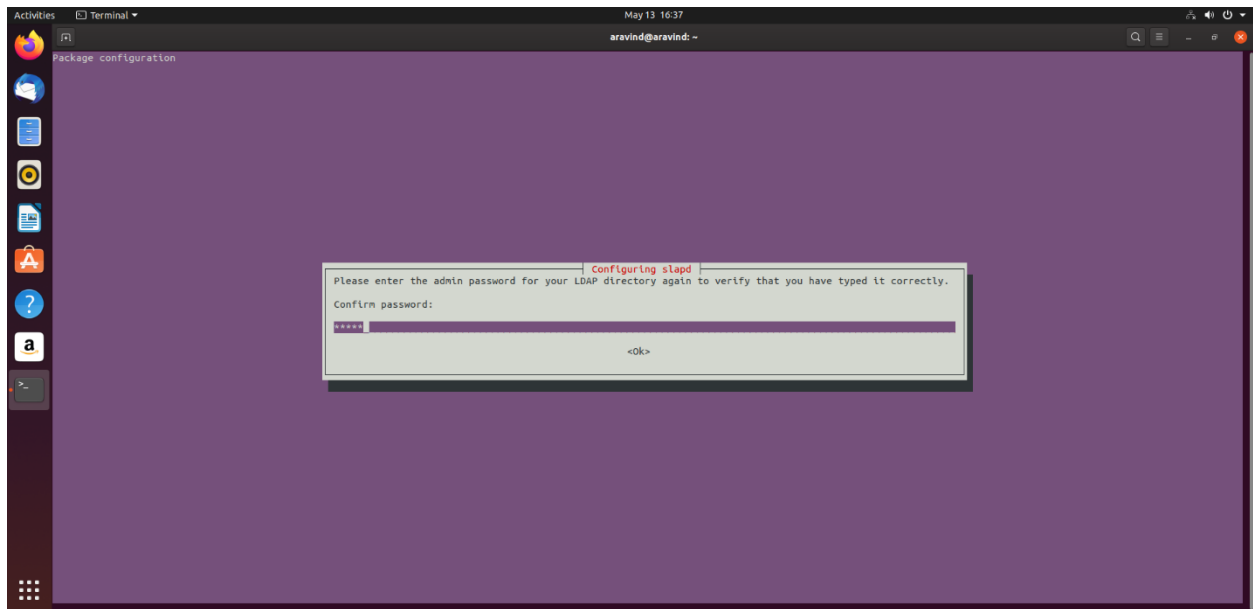


- **Organization name** – Epita (as per my project) you can give any name

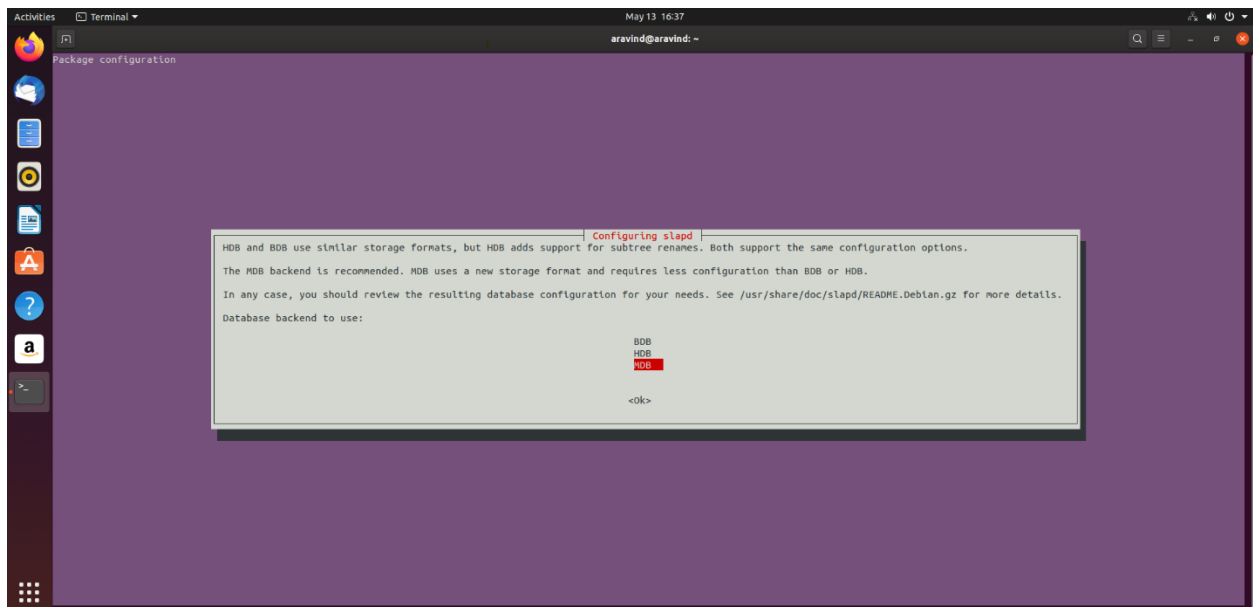


- **Administrator password** – give a password and confirm it again

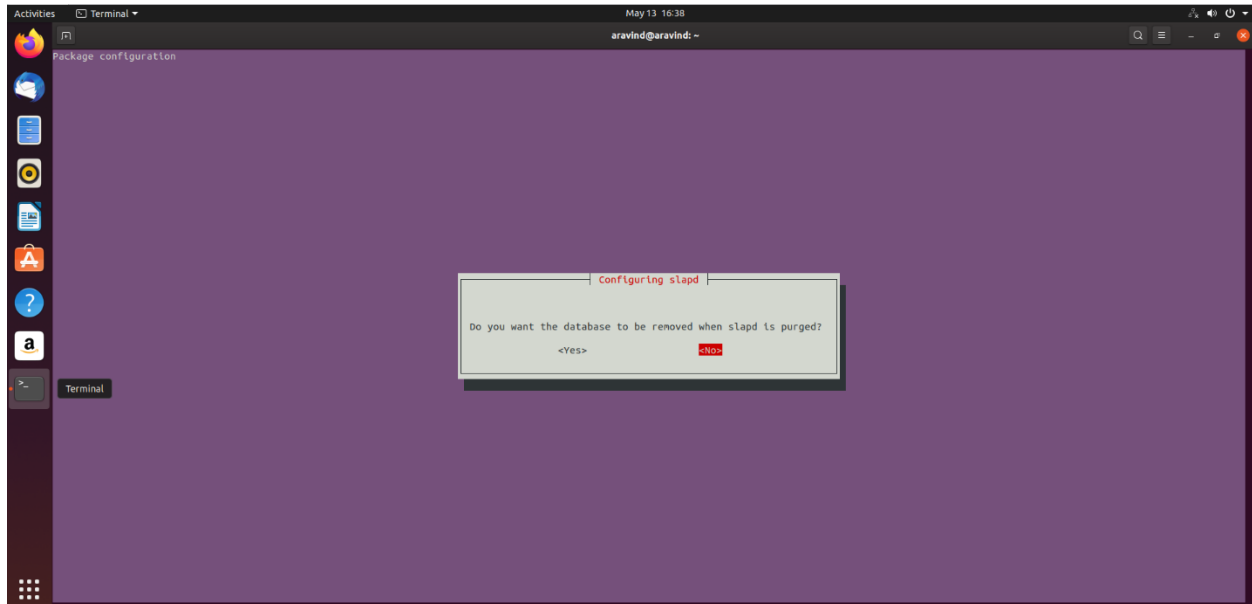




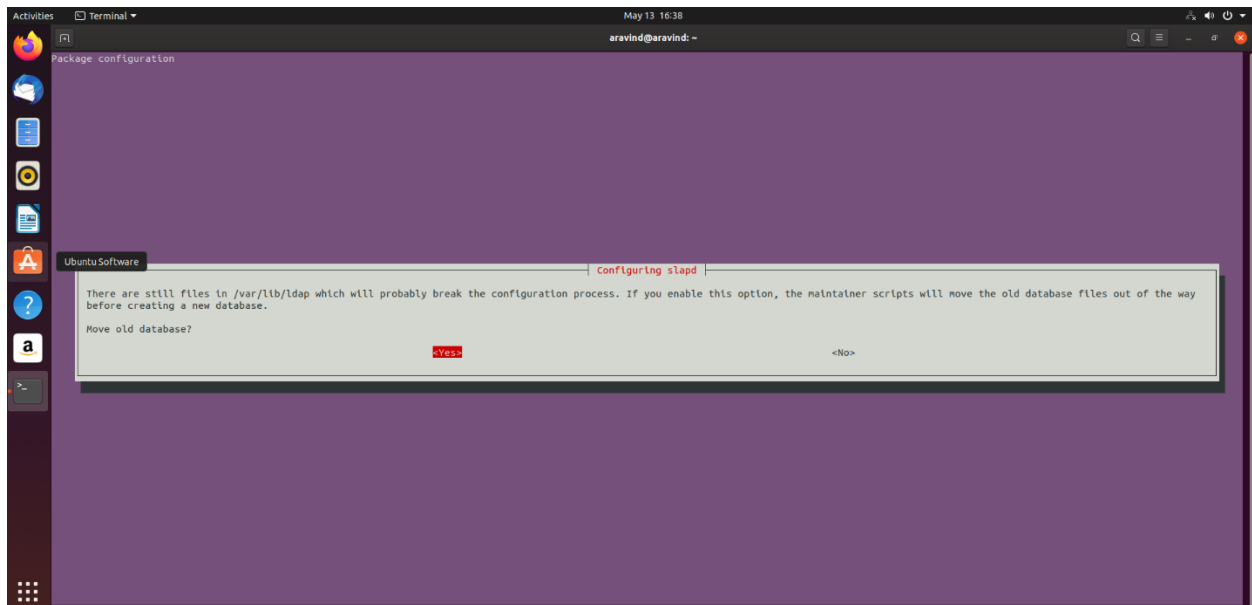
➤ Database backend- MDB



➤ **Remove the database when slapd is purged? – No**



➤ **Move old database? – Yes**



Step 3

Once all the steps are done, we now are going to perform the configuration verification checks

Command - `sudo slapcat`

```
aravind@aravind:~$ sudo slapcat
dn: dc=kasiraman,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: Epita
d
s LibreOffice Writer      class: organization
entryUUID: 36c7e162-2973-103a-8051-d1801aff8744
creatorsName: cn=admin,dc=kasiraman,dc=com
createTimestamp: 20200513143854Z
entryCSN: 20200513143854.126416Z#000000#0000#000000
modifiersName: cn=admin,dc=kasiraman,dc=com
modifyTimestamp: 20200513143854Z

dn: cn=admin,dc=kasiraman,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: eINTSEF9NStKYjBYQWZKaDRQbDNFL01LR1V5ZDRvb0RrSyt2cHg=
structuralObjectClass: organizationalRole
entryUUID: 36c8396e-2973-103a-8052-d1801aff8744
creatorsName: cn=admin,dc=kasiraman,dc=com
createTimestamp: 20200513143854Z
entryCSN: 20200513143854.128739Z#000000#0000#000000
modifiersName: cn=admin,dc=kasiraman,dc=com
modifyTimestamp: 20200513143854Z
```

The above command will print all the directory entries.

Step 4

Now we are going to check the config DIT

Command - `sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config dn`

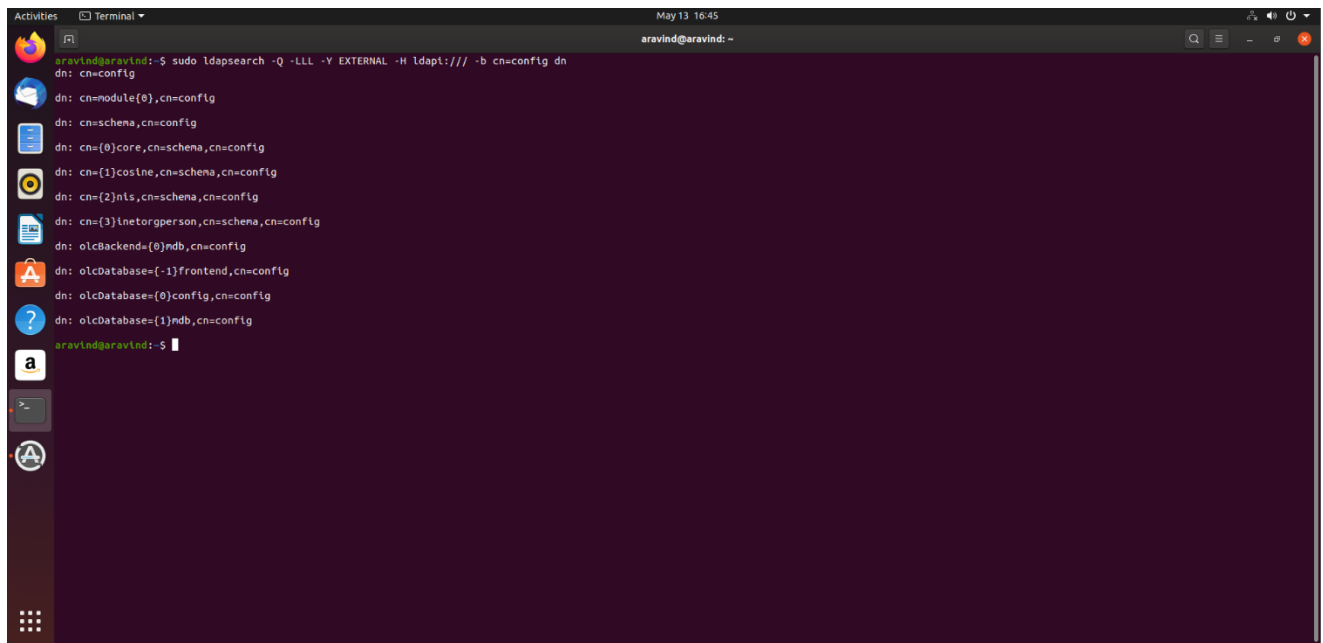
-Q: enable silent mode for SASL authentication

-Y: pick the SASL mode chosen for authentication. Normally, EXTERNAL implies an authentication by client certificate but in this case, it means that the authentication will be done by the UID and the GID of the system account. This is why you have to launch the command with “sudo”.

-L: choose to display the result in LDIF format. We could have said -LLL to have the same thing without all the lines commented.

-H: indicates the URI we want to use to connect. Here ldapi:/// says to connect to the Unix socket locally (communication goes through a local file rather than over the network).

-b: indicates the node from which you want to search. Here "dc= kasiraman,dc=com" is the root so you search throughout the DIT.

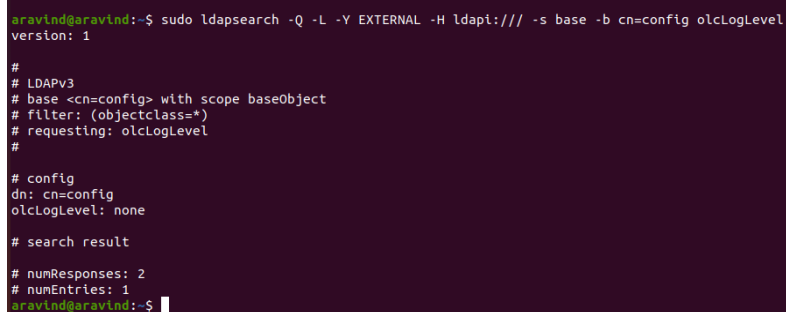
A terminal window with a dark purple background. The title bar shows 'May 13 16:45' and 'aravind@aravind: ~'. The command 'sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config dn' has been executed. The output lists several LDAP entries under the 'cn=config' branch, including 'cn=module{0},cn=config', 'cn=schema,cn=config', 'cn={0}core,cn=schema,cn=config', 'cn={1}cosine,cn=schema,cn=config', 'cn={2}nis,cn=schema,cn=config', 'cn={3}inetorgperson,cn=schema,cn=config', 'olcBackend={0}mdb,cn=config', 'olcDatabase={-1}frontend,cn=config', 'olcDatabase={0}config,cn=config', and 'olcDatabase={1}mdb,cn=config'. The prompt 'aravind@aravind: ~\$' is visible at the bottom.

```
aravind@aravind:~$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config dn
dn: cn=config
dn: cn=module{0},cn=config
dn: cn=schema,cn=config
dn: cn={0}core,cn=schema,cn=config
dn: cn={1}cosine,cn=schema,cn=config
dn: cn={2}nis,cn=schema,cn=config
dn: cn={3}inetorgperson,cn=schema,cn=config
dn: olcBackend={0}mdb,cn=config
dn: olcDatabase={-1}frontend,cn=config
dn: olcDatabase={0}config,cn=config
dn: olcDatabase={1}mdb,cn=config
aravind@aravind:~$
```

Step 5

In this step we are going to start the logging of slapd. But first we need to check the existing log level of slapd.

Command - `sudo ldapsearch -Q -L -Y EXTERNAL -H ldapi:/// -s base -b cn=config olcLogLevel`

A terminal window showing the output of the command 'sudo ldapsearch -Q -L -Y EXTERNAL -H ldapi:/// -s base -b cn=config olcLogLevel'. The output shows LDAPv3 search details for 'cn=config' with scope 'baseObject' and filter '(objectclass=*)'. It indicates that the requested attribute is 'olcLogLevel' and the current value is 'none'. The search result shows 2 responses and 1 entry.

```
aravind@aravind:~$ sudo ldapsearch -Q -L -Y EXTERNAL -H ldapi:/// -s base -b cn=config olcLogLevel
version: 1

#
# LDAPv3
# base <cn=config> with scope baseObject
# filter: (objectclass=*)
# requesting: olcLogLevel
#
# config
dn: cn=config
olcLogLevel: none
# search result
# numResponses: 2
# numEntries: 1
aravind@aravind:~$
```

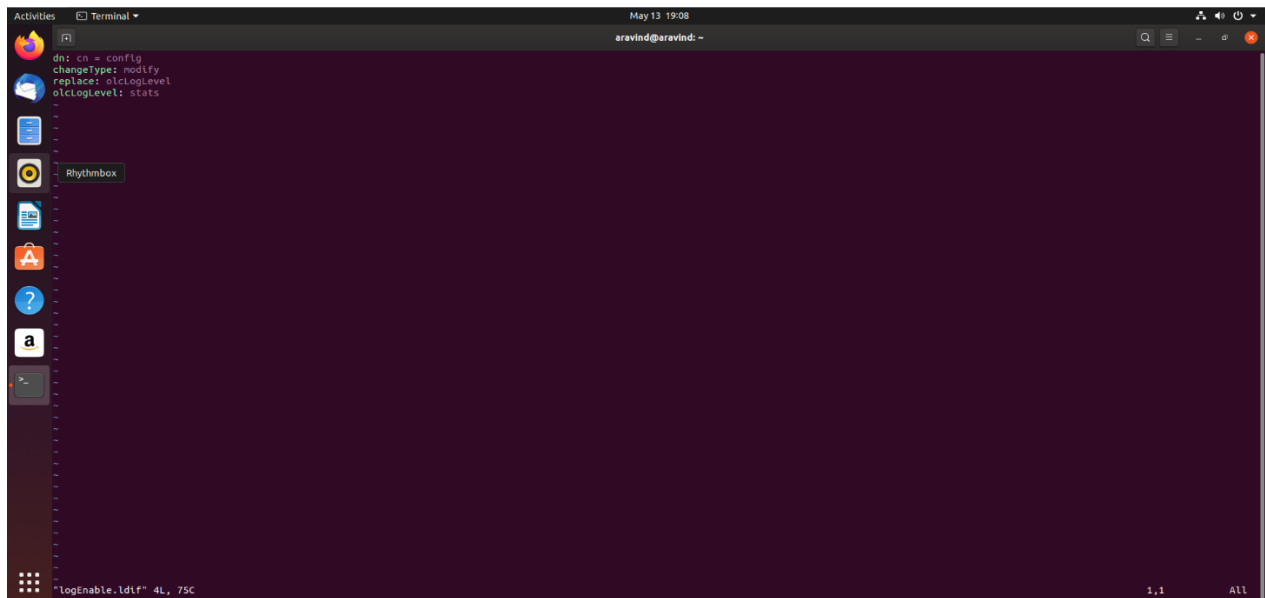
Once we check the existing level of the slapd we are going to start the logging of slapd and enable them

Command – `nano/vim logEnable.ldif`

Once we created the file, we need to manually enter the following commands lines listed below

Command -

```
dn: cn = config
changeType: modify
replace: olcLogLevel
olcLogLevel: stats
```



Step 6

Now we are going to apply the changes in the logEnable.ldif file

Command - `sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f logEnable.ldif`

Once we have enabled, the server will start generating logs, but it will be generated to the path **"/var/log/syslog"**.

In order to create a separate log file, we need to create a rsyslog daemon

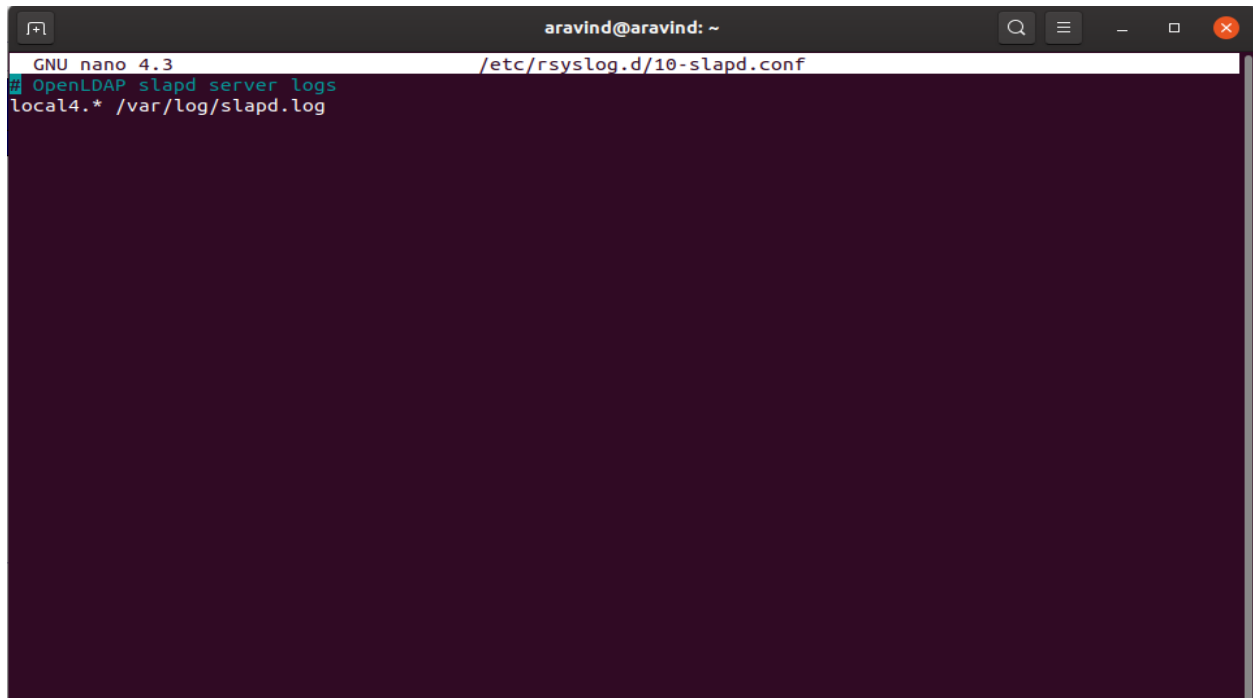
Command – `sudo nano /etc/rsyslog.d/10-slapd.conf`

Once created and entered into the file we need to enter the following lines as listed below

Command -

```
# OpenLDAP slapd server logs
```

local4.* /var/log/slapd.log



```
aravind@aravind: ~  
GNU nano 4.3 /etc/rsyslog.d/10-slapd.conf  
# OpenLDAP slapd server logs  
local4.* /var/log/slapd.log
```

Once done save the file. The above file will now create a separate log for the slapd server logs instead of saving in “/var/log/syslog”.

Now we need to restart the rsyslog

Command - `sudo systemctl restart rsyslog`

Step 7

Once done with the logging we need to create new entries for the ldap. For that we need to create a file called as treekasiraman.ldif (as per my project)

Command – `nano treekasiraman.ldif`

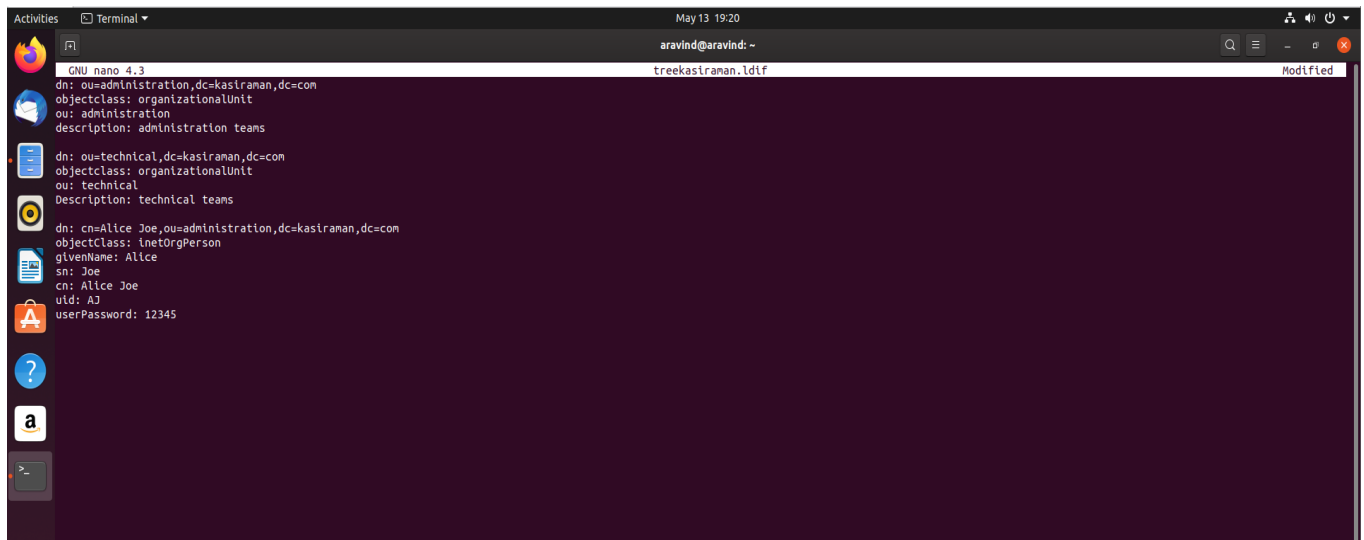
Once done, copy the following command from below and paste it inside the file

Command –

```
dn: ou=administration,dc=kasiraman,dc=com  
objectclass: organizationalUnit  
ou: administration  
description: administration teams
```

dn: ou=technical,dc=kasiraman,dc=com
objectclass: organizationalUnit
ou: technical
Description: technical teams

dn: cn=Alice Joe,ou=administration,dc= kasiraman,dc=com
objectClass: inetOrgPerson
givenName: Alice
sn: Joe
cn: Alice Joe
uid: AJ
userPassword: 12345



```
GNU nano 4.3 treekasiraman.ldif
dn: ou=administration,dc=kasiraman,dc=com
objectclass: organizationalUnit
ou: administration
description: administration teams

dn: ou=technical,dc=kasiraman,dc=com
objectclass: organizationalUnit
ou: technical
Description: technical teams

dn: cn=Alice Joe,ou=administration,dc=kasiraman,dc=com
objectclass: inetOrgPerson
givenName: Alice
sn: Joe
cn: Alice Joe
uid: AJ
userPassword: 12345
```

Save the file once done. Now we need to apply the changes to the treekasiraman.ldif file

Command - `sudo ldapadd -x -W -D "cn=admin,dc= kasiraman,dc=com" -H ldap://localhost -f treekasiraman.ldif`

- x: pick 'simple authentication' by password
- W: displays an interactive prompt for admin account credentials submission
- D: to indicate the Distinguished Name binding of the respective account to connect
- H: choose the connection method: ldap://localhost initiates a connection using the network on TCP port 389

Note: In case if you plan to re-run this command then make sure you include the -c (switch) to continue execution if any of the entry (in passed ldif file) already exists.

Delete an entry (optional): `sudo ldap -v -D "cn=admin,dc= kasiraman,dc=com" -W "cn=Alice Joe,ou=administration,dc= kasiraman,dc=com"`

Step 8

In this step we are going to setup a web-based front-end using the PHPLDAPADMIN client.

We are going to install PHPLDAPADMIN

Command - `sudo apt-get install phpldapadmin`

You can also use other frontal clients.

```
aravind@aravind:~$ sudo apt-get install phpldapadmin
Reading package lists... Done
Building dependency tree
Reading state information... Done
phpldapadmin is already the newest version (1.2.2-6.1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
aravind@aravind:~$
```

Once installed phpldapadmin we are going to configure it by modifying the config.php file in the phpldapadmin folder (Path – **`/etc/phpldapadmin/config.php`**)

Command – `sudo nano /etc/phpldapadmin/config.php`

Modify the following lines in the config.php file

- `servers->SetValue('server','host','localhost') //line ~293`
- `servers->SetValue('server','base',array('dc=epita,dc=com')) //line ~300`
- `servers->SetValue('server','bind_id',array('cn=admin,dc=epita,dc=com')) //line ~326`
- `config->custom->appearance['hide_template_warning'] = true; //line ~161:`
uncomment first

```
286 $servers->setValue('server','name','My LDAP Server');
287
288 /* Examples:
289 'ldap.example.com',
290 'ldaps://ldap.example.com/',
291 'ldapi://127.0.0.1:389'
292 (Unix socket at /usr/local/var/run/ldap) */
293 $servers->setValue('server','host','localhost');
294
295 /* The port your LDAP server listens on (no quotes). 389 is standard. */
296 // $servers->setValue('server','port',389);
297
298 /* Array of base DNS of your LDAP server. Leave this blank to have phpLDAPadmin
299 auto-detect it for you. */
300 $servers->setValue('server','base',array('dc=kastraman,dc=com'));
301
302 /* Five options for auth_type:
303 1. 'cookie': you will login via a web form, and a client-side cookie will
304 store your login dn and password.
305 2. 'session': same as cookie but your login dn and password are stored on the
306 web server in a persistent session variable.
307 3. 'http': same as session but your login dn and password are retrieved via
308 HTTP authentication.
309 4. 'config': specify your login dn and password here in this config file. No
310 login will be required to use phpLDAPadmin for this server.
311 5. 'sasl': login will be taken from the webserver's kerberos authentication.
312 Currently only GSSAPI has been tested (using mod_auth_kerb).
313
314 Choose wisely to protect your authentication information appropriately for
315 your situation. If you choose 'cookie', your cookie contents will be
316 encrypted using blowfish and the secret your specify above as
317 session['blowfish']. */
318 $servers->setValue('login','auth_type','session');
319
320 /* The DN of the user for phpLDAPadmin to bind with. For anonymous binds or
321 'cookie','session' or 'sasl' auth types, LEAVE THE LOGIN_DN AND LOGIN_PASS
322 BLANK. If you specify a login_attr in conjunction with a cookie or session
323 auth type, then you can also specify the bind_id/bind_pass here for searching
324 the directory for users (ie, if your LDAP server does not allow anonymous
325 binds. */
326 $servers->setValue('server','bind_id','cn=admin,dc=kastraman,dc=com');
327 # $servers->setValue('login','bind_id','cn=Manager,dc=example,dc=com');
328
329 /* Your LDAP password. If you specified an empty bind_id above, this MUST also
330 be blank. */
331 // $servers->setValue('login','bind_pass','');
332 -- INSERT --
```

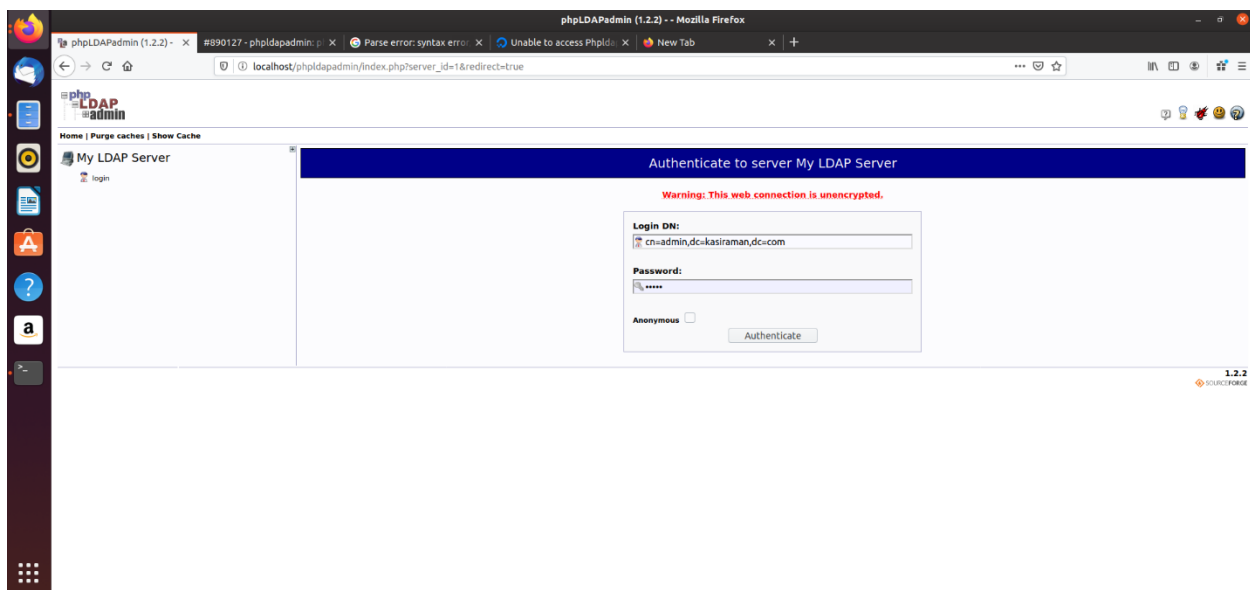
Once we modified the following, we need to restart the apache service.

Command - `systemctl restart apache2.service`

Step 9

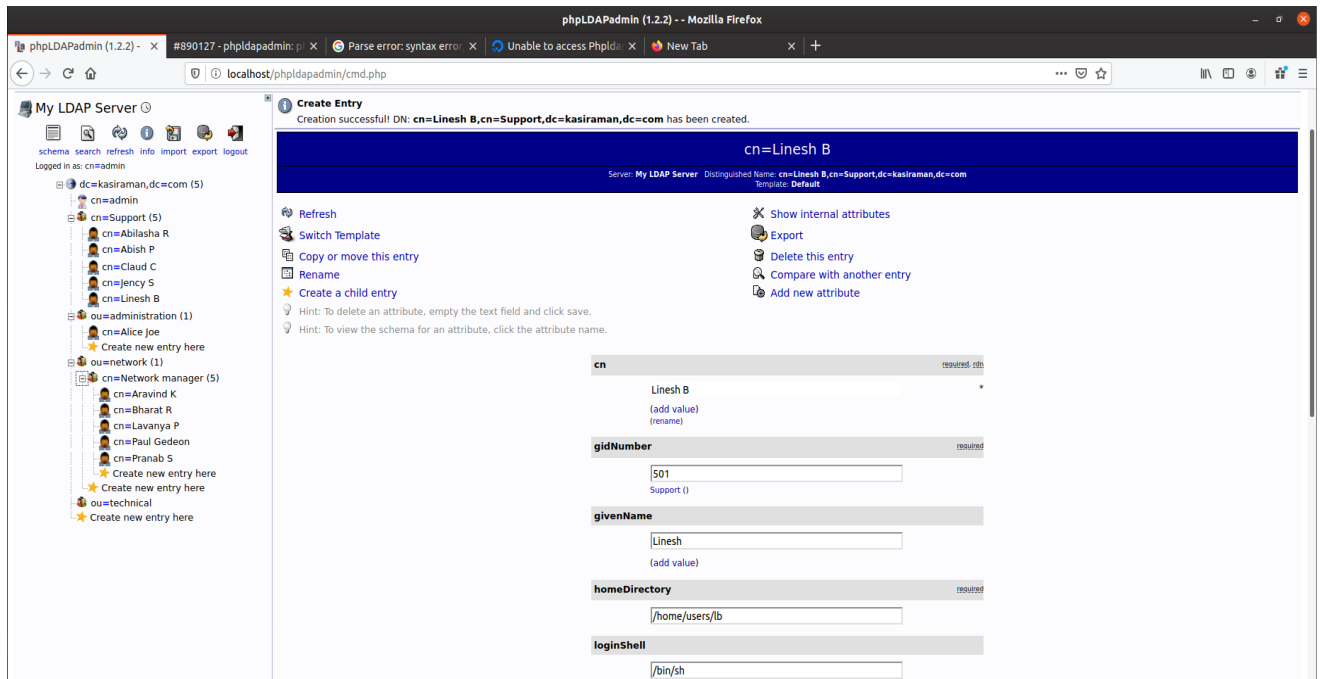
Open the browser and type the following

Command – `localhost/phpldapadmin/`



Once opened login with the LDAP password which we created in step 2.

After that we need to create a group and add the users in each group



NOTE

If faced any error which accessing the site localhost/phpldapadmin/, we need to do the following in order to access the site

Open nano usr/share/phpldapadmin/lib/functions.php

change line 54 to "function my_autoload(\$className) {"

Add this code "spl_autoload_register("my_autoload");" on line 777

On line 1083

change line 1083 to "\$CACHE[\$sortby] = __create_function('\$a, \$b',\$code);"

add the code below from the

http://php.net/manual/pt_BR/function.create-function.php page on line 1091

```
function __create_function($arg, $body) {
```

```
    static $cache = array();
```

```
    static $maxCacheSize = 64;
```

```
    static $sorter;
```

```
    if ($sorter === NULL) {
```

```

$sorter = function($a, $b) {
    if ($a->hits == $b->hits) {
        return 0;
    }

    return ($a->hits < $b->hits) ? 1 : -1;
};

$crc = crc32($arg . "\\x00" . $body);

if (isset($cache[$crc])) {
    ++$cache[$crc][1];
    return $cache[$crc][0];
}

if (sizeof($cache) >= $maxCacheSize) {
    uasort($cache, $sorter);
    array_pop($cache);
}

$cache[$crc] = array($cb = eval('return
function('.$arg.'){'.$body.'}'), 0);
return $cb;
}

```

By doing the changes we will be able to access the site.