

LDAP GUI for Academic Institutes : Working Examples and Test Cases

Challenge of Code 2010
Association of Computing Machinery
Bangalore Chapter

Ashwin Tumma
College of Engineering, Pune
ashwin.tumma23@gmail.com

Swapnil Kulkarni
College of Engineering, Pune
swapnil514@gmail.com

1 Working Examples

This section describes the Working Examples of our tool. We have considered one machine in the network as our server, and installed the server side software on it. Client side software is installed on one of the client machine in the network.

1.1 Assumptions

- The user is able to run shell scripts at the terminal as a *sudo* user.
- The server has access to Internet Connection for downloading various packages.
- All the machines are connected in a network and can *access(ping)* the server.

1.2 ServerSide Tool

1.2.1 Installing OpenLDAP and Berkeley DB

The steps of installing OpenLDAP and Berkeley DB are explained Chapter 5 in the report. The Shell Script named `Install.sh` installs both the softwares.

Figure 2 shows the screen for getting basic parameters like *Name of Institute*, *Domain Name* and *Number of departments*. On clicking **Start Installation** button, the values you have entered are recorded in the tool and it directs you to screen as shown in figure 3. Here, you will see the instructions as to how to run the `Install.sh` script manually. We have deliberately not invoked this shell script from the webpage. If we have done so, the user wouldn't have got to see the actual compilation process of the software installation. On clicking **Start Configuration** button you will be redirected to screen as shown in the figure 4.

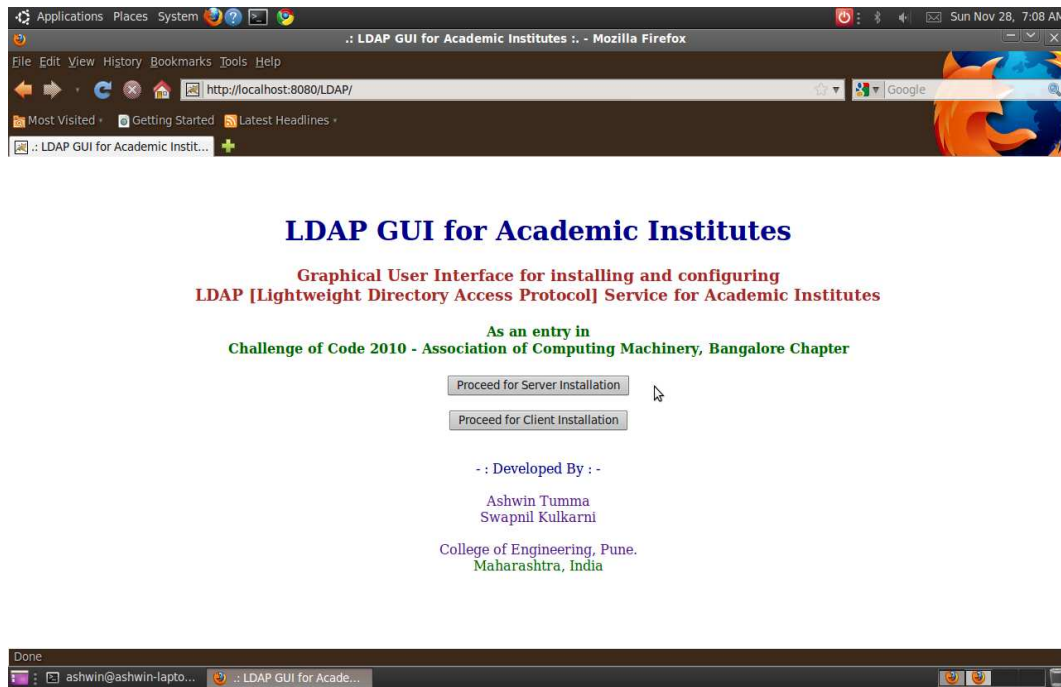


Figure 1: Screenshot of Homepage

1.2.2 Specify Server Configurations

As shown in figure 4, the `slapd.conf` file is created from the values the user has specified for the server and its status is displayed accordingly. This file is the heart of the server. All the parameters that govern the server running are specified in this file. The shell script named `Make_slapd_conf.sh` will be invoked and a `slapd.conf` file is created as per the parameters specified by the user and is automatically placed at this proper location, *i.e.* at `/usr/local/etc/openldap/slapd.conf`. An administrative account for the server management is created in this file. We have named it as *Manager*.

1.2.3 Create the Directory Information Tree

Next, the tool will then prompt for the names of the departments for which you want to use LDAP. The number of departments value is taken from the initial pages. On clicking the **Build Directory Information Tree** button, you will be redirected to the screen as shown in the figure 5. Here the status of your creation of Directory Information Tree is displayed. LDAP needs files of the form `.ldif` - **LDAP Directory Interchange Format**. A file named `hierarchy.ldif` is created which will showcase your tree and group files are created for the various uses and departments that you have specified. The shell script `CreateLDIF.sh` will create the `hierarchy.ldif` and other group files. You can refer to the shell script for more better explanation. Now the server has to be started and the `.ldif` files just created need to be added to the server.

1.2.4 Start Server and add files

As shown in figure 5 on clicking **Start Server** and **Add Files** button, you can start the server and the `.ldif` files are added to the server. The shell script named `ServerStart.sh` will start the server process by running the following command: `sudo /usr/local/libexec/slapd`. The process id on which the server is running can be obtained from the file

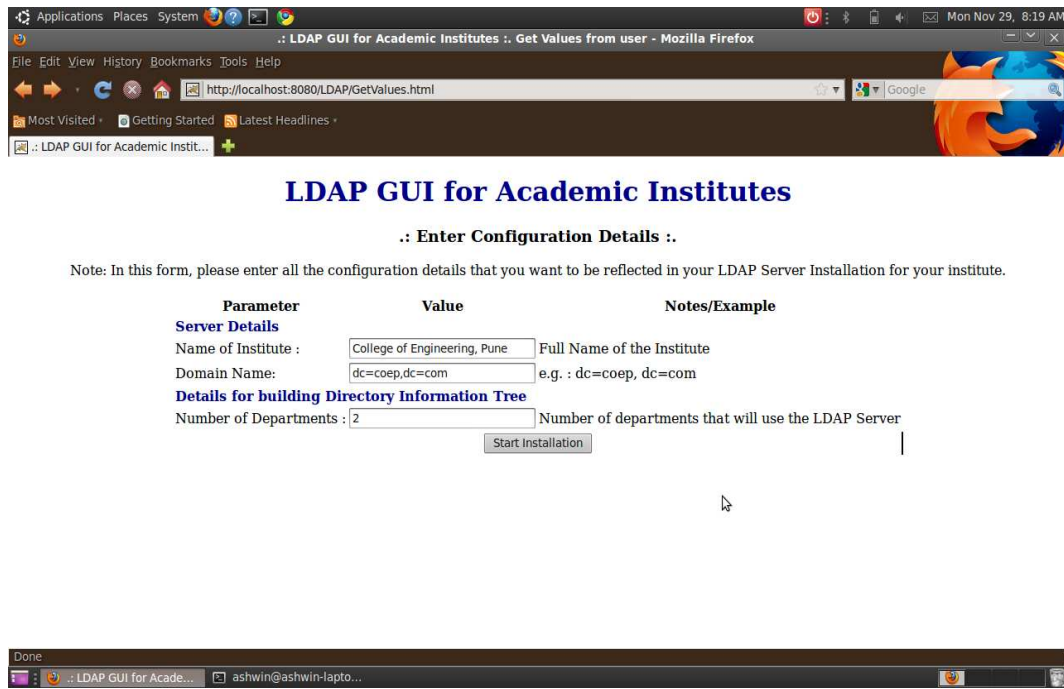


Figure 2: Get values of basic parameters for the Server

`/usr/local/var/run/slapd.pid`. Now, your server is functional and you are all set to create accounts for your users.

1.2.5 Create User Accounts

We now need to create accounts for various users. For this we have provided a sample `1.ldif` file for one user and have explained it in the tool. A shell script named `UserAdd.sh` can be used when user accounts have to be created in bulk. The user is advised to read the comments in the shell scripts to get a better understanding of the procedure.

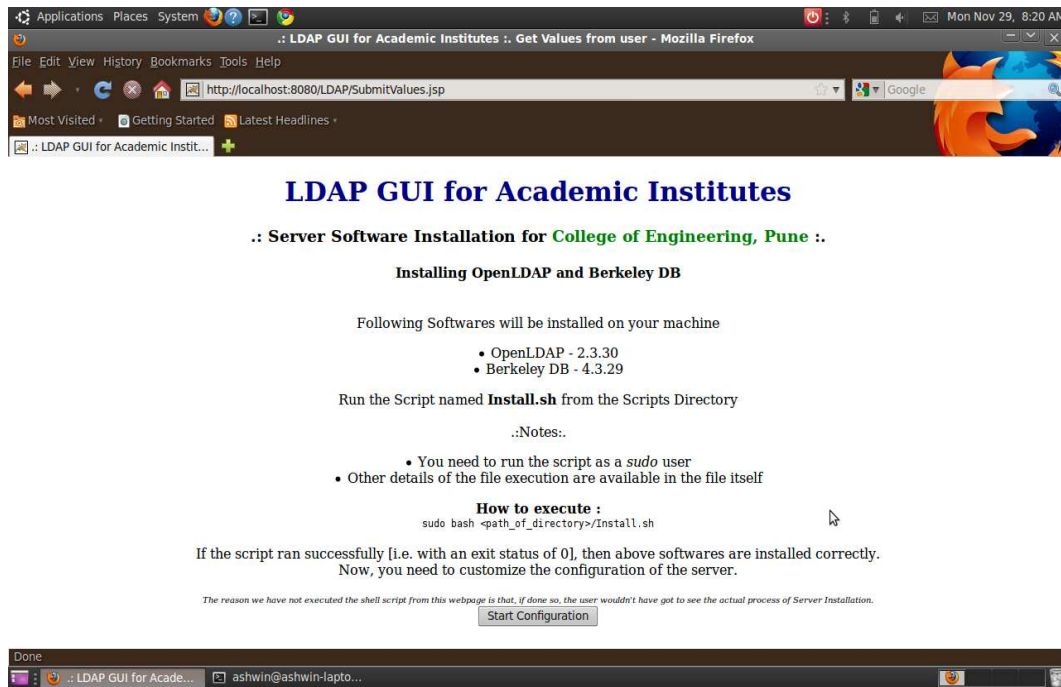


Figure 3: Installation of OpenLDAP and Berkeley DB

1.3 ClientSide Tool

The steps to configure a machine as an LDAP Client are as follows:

1. Download the package `libnss-ldap`
2. Edit the configuration files that Linux System uses to authenticate its users.

1.3.1 Package - `libnss-ldap`

Install the package by :

```
sudo apt-get install libnss-ldap
```

This automatically installs the following packages required for a machine to function as a LDAP Client: `libnss-ldap`, `auth-client-config`, `ldap-auth-client`, `ldap-auth-config`, `libpam-ldap`, `nscd`

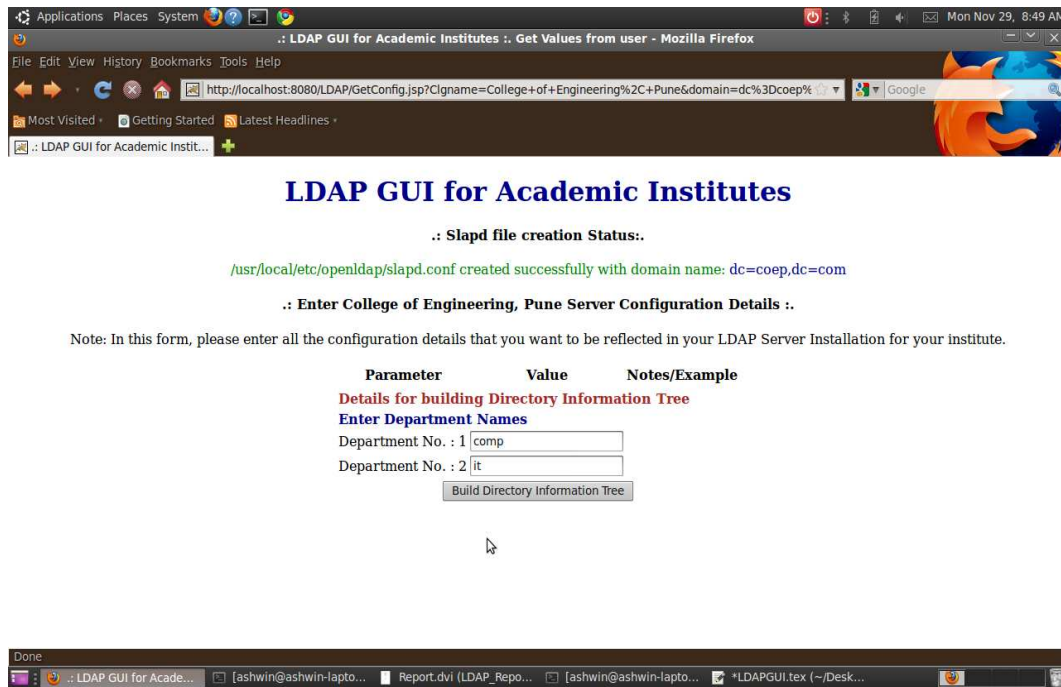


Figure 4: Slapd.conf creation status and input values of departments

Whilst installation, it will prompt for the IP address of the server, where you need to enter the IP address of your server.

- Enter the following when installation is going on:
ldap://your-ip-address
- Select Version 3
- Answer *No* to all other options.

1.3.2 Edit the configuration files

The shell script that we have provided needs to be executed on each client machine. Copy the file named **Client.sh** and directory named **ClientFiles** on each machine and then run the **Client.sh** file. The details of the shell script are as follows:

- Name : **Client.sh**
- Where to place : Can be placed in any directory on the client machine, only constraint is that the *current directory* of this file and the directory named **ClientFiles** should be the same.
- Supporting Files : **common-auth**, **common-account**, **common-session**, **common-password**, **nsswitch.conf**. These files are in the **ClientFiles** directory.
- How to execute : **sudo ./Client.sh**

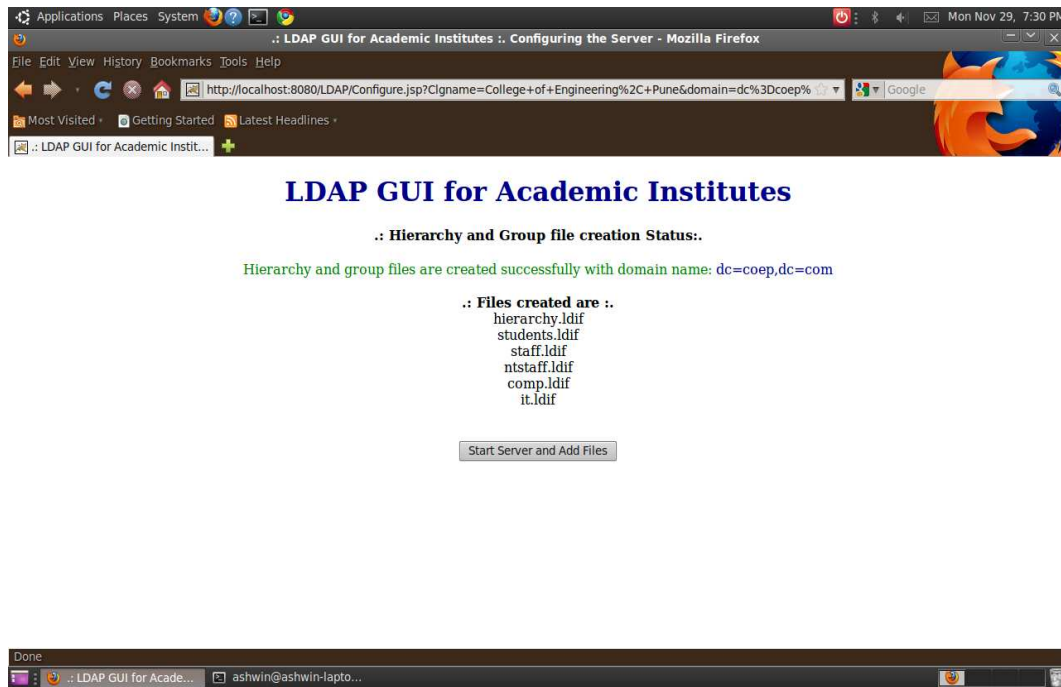


Figure 5: Build the Directory Information Tree

Details The `Client.sh` file is written to configure the machine as a LDAP Client. Basically what it does is that, it simply takes a backup of your original configuration files *viz.*

```
/etc/nsswitch.conf,
/etc/pam.d/common-auth,
/etc/pam.d/common-account,
/etc/pam.d/common-session,
/etc/pam.d/common-password
```

in the directory named `/etc/pam.d/backup_common_files/`. Then, copies our edited configuration file in their appropriate position.

Backup of the original files is taken so that, in case anything goes wrong, you can always restore the original files from the backup directory. Now, an obvious question is that why do we need to edit these files only? The `libnss-ldap` package has only provided the client machine the ability to act as a LDAP client, but to do so we also need to edit various configuration files also. Linux follows a standard procedure to authenticate its users by consulting the above mentioned files. It checks the settings in these files and moulds its procedure of authentication accordingly. We also specify in these files the library files to which the authentication process should link for using LDAP. These files are usually with the `.so` suffix.

Here's a brief explanation of it :

- `/etc/nsswitch.conf` : It is used to configure which services are to be used to determine information such as hostnames, password files, and group files. So by editing it we are instructing it to use LDAP Server Service for authenticating its users.
- `/etc/pam.d/common-auth`, `/etc/pam.d/common-account`,

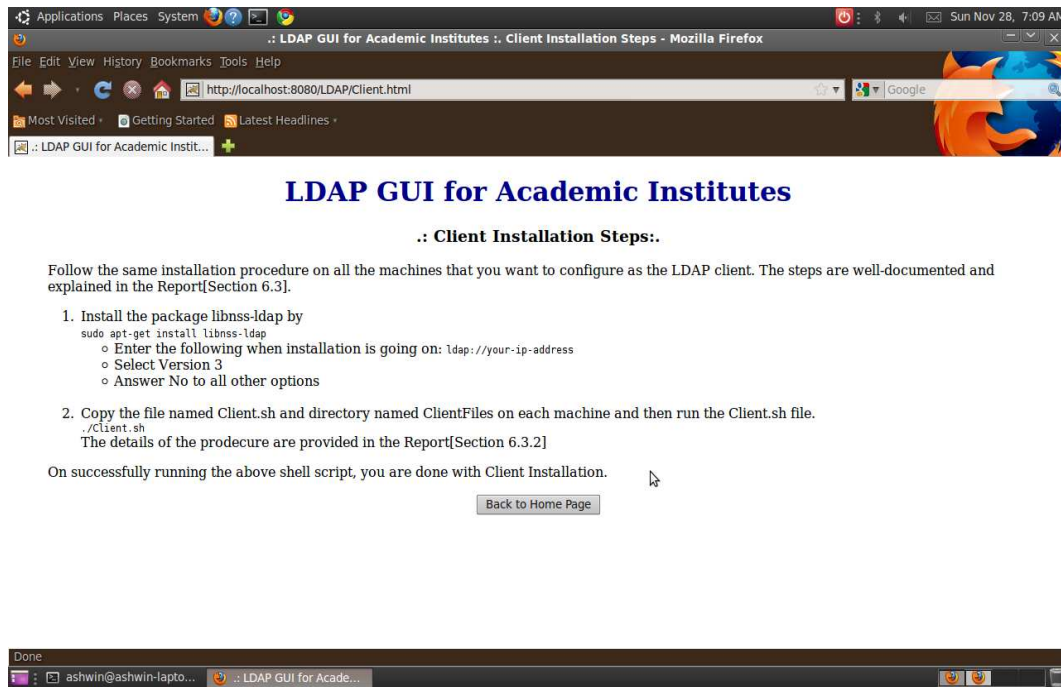


Figure 6: Screenshot of Client Installation procedure

`/etc/pam.d/common-session`, `/etc/pam.d/common-password`: These files merely represent different stages in the authentication process. By editing them, we are instructing the client machine that, along with its local authentication procedure, also consider the LDAP Server.

2 Test Cases

Here we consider one test case, wherein we first install the LDAP server on one machine and then install a client on one of the machines. Screenshots of server installation are already provided in the previous section. Here we focus on client installation. The figures below show the screenshots of installing the LDAP client on the machine which needs to act as a client.

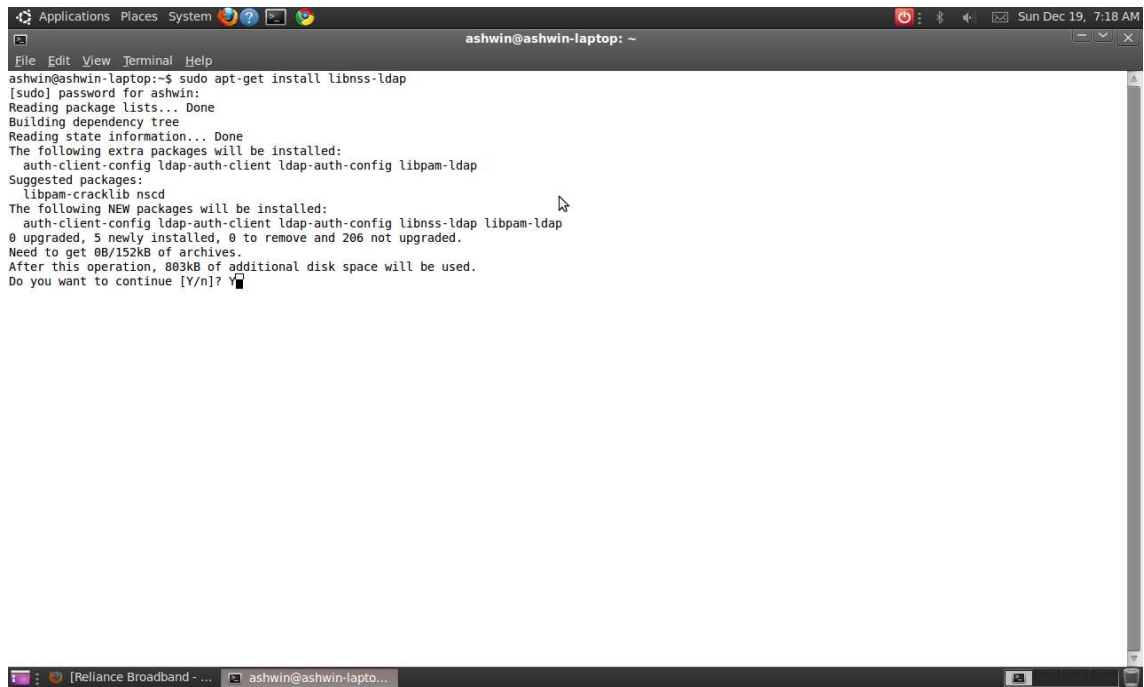
Figure 7 showcases the fetching of the `libnss-ldap` package from the repository.

Figure 8 shows the screenshot of the step wherein we need to specify the IP address of the LDAP Server. In our case, our server runs at `10.1.11.48` so we have set the value to be `ldap://10.1.11.48`

Figure 9 shows the step where we specify the domain name of our server. In our case, it is `dc=coep, dc=com`

Figure 10 shows that we select version 3 of LDAP.

Figure 11 shows some of the nominal configuration settings that are to be specified as per the user's requirement.

A screenshot of a Linux terminal window titled 'ashwin@ashwin-laptop: ~'. The window shows the command 'sudo apt-get install libnss-ldap' being executed. The output includes prompts for the password, package list reading, dependency tree building, and state information reading. It lists extra packages to be installed (auth-client-config, ldap-auth-client, ldap-auth-config, libpam-ldap) and suggested packages (libpam-cracklib, nscd). It also shows the new packages to be installed (auth-client-config, ldap-auth-client, ldap-auth-config, libnss-ldap, libpam-ldap), the number of packages to be upgraded, installed, removed, or not upgraded, and the disk space requirements. The prompt 'Do you want to continue [Y/n]?' is shown with 'Y' entered.

```
ashwin@ashwin-laptop:~$ sudo apt-get install libnss-ldap
[sudo] password for ashwin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  auth-client-config ldap-auth-client ldap-auth-config libpam-ldap
Suggested packages:
  libpam-cracklib nscd
The following NEW packages will be installed:
  auth-client-config ldap-auth-client ldap-auth-config libnss-ldap libpam-ldap
0 upgraded, 5 newly installed, 0 to remove and 206 not upgraded.
Need to get 0B/152kB of archives.
After this operation, 803kB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Figure 7: Getting the software `libnss-ldap` from the repositories

Later on simply running the `Client.sh` script our LDAP clients will be functional. To test the login, we can simply login at the terminal.

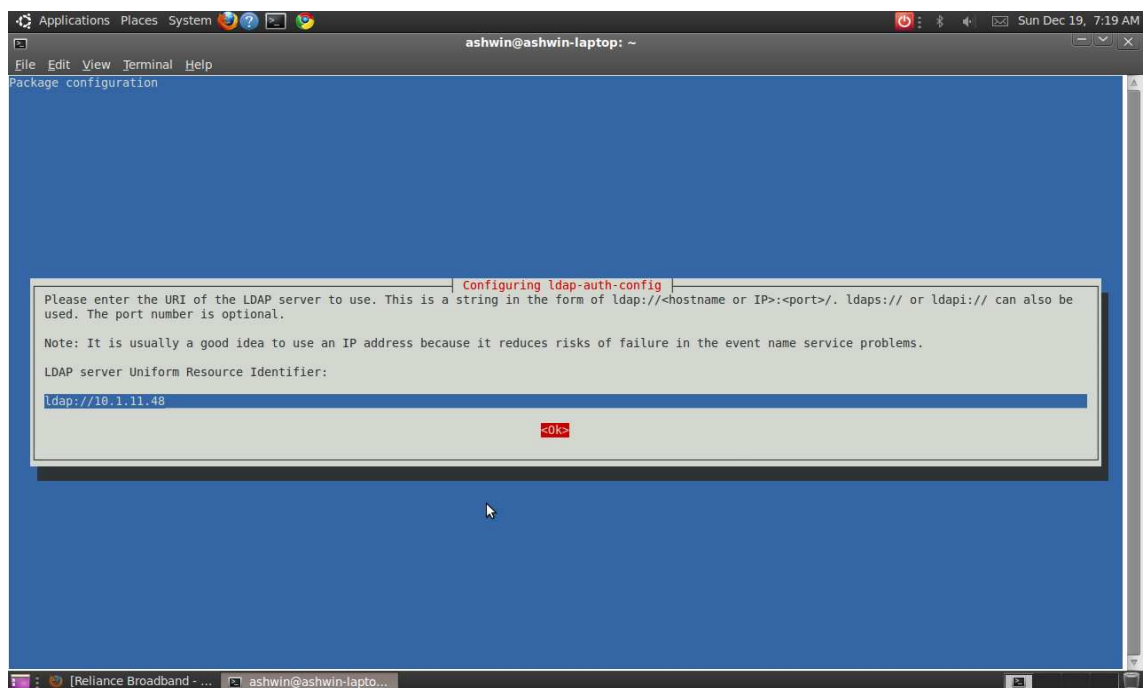


Figure 8: Inform the server's IP address

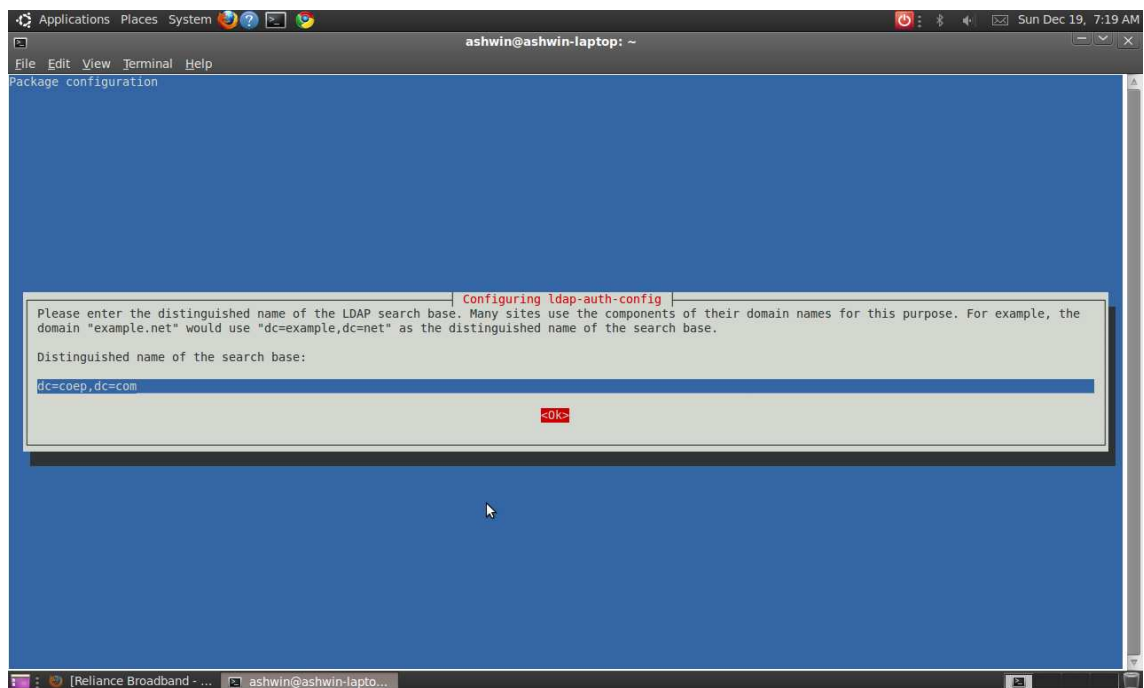


Figure 9: Specify the domain name

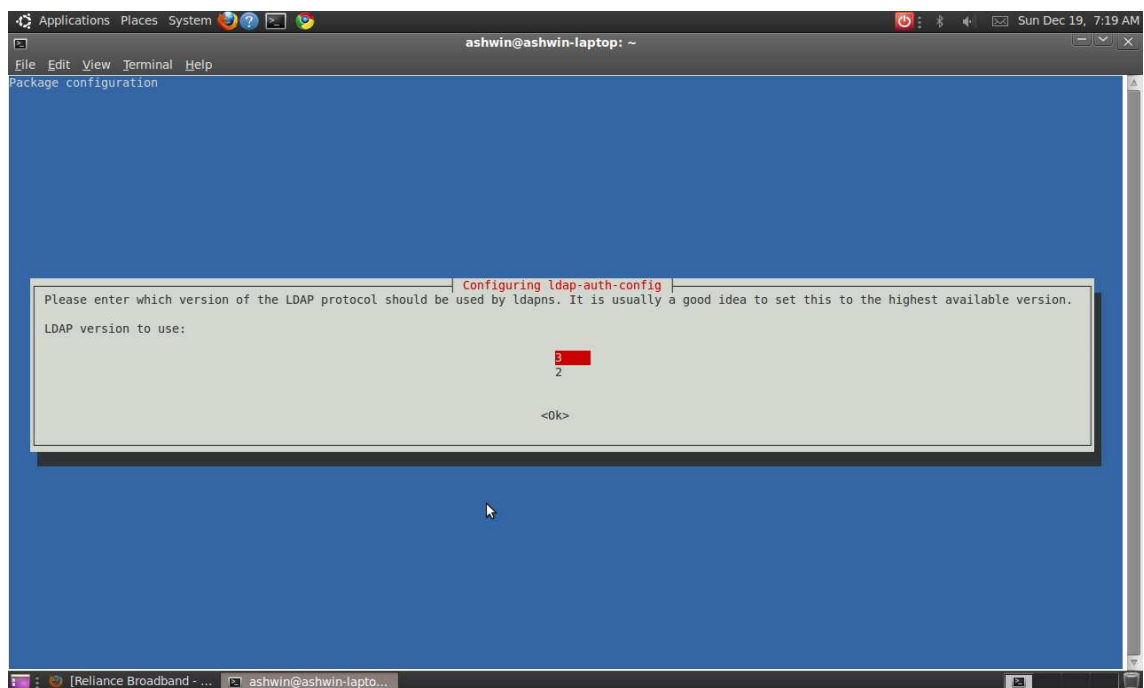


Figure 10: Selecting the LDAP Version

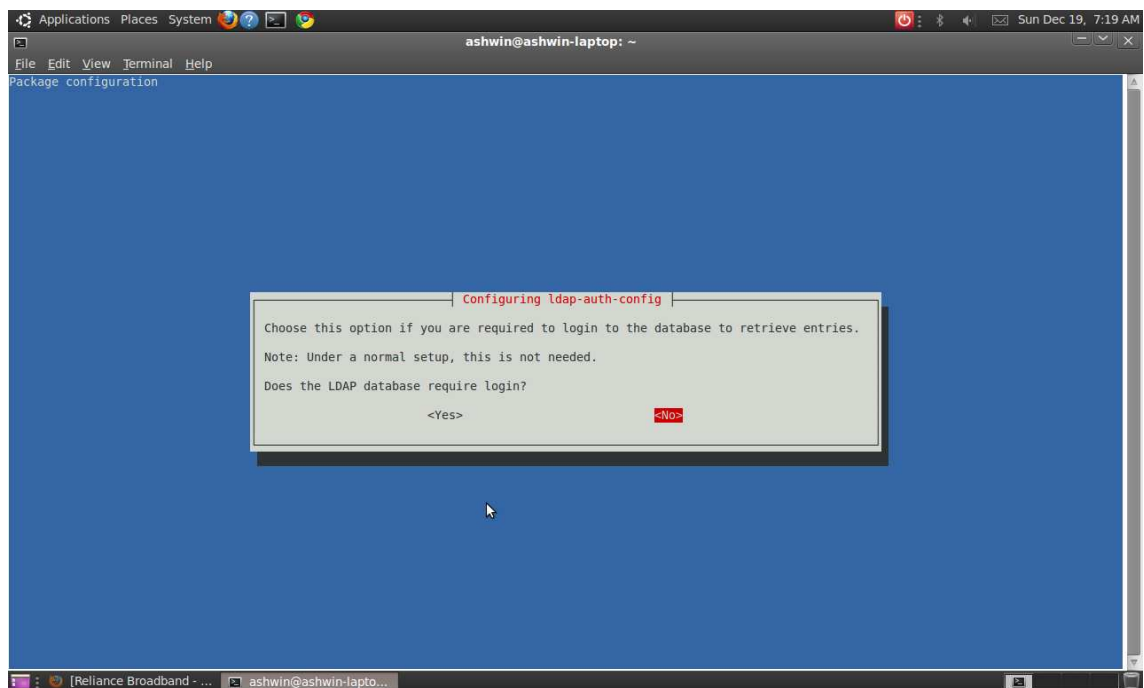


Figure 11: Further configurations