I have installed LDAP and phpLDAPadmin on my Ubuntu 14.04 (Trusty Tahr). It is working, and I'm able to access phpLDAPadmin GUI and am also able to create OU and groups using the GUI, but I'm not able to create users.

Open template file `/usr/share/phpldapadmin/lib/TemplateRender.php` in your favorite editor like:

```
 sudo vi  /usr/share/phpldapadmin/lib/TemplateRender.php
```

Search line

```
$default = $this->getServer()->getValue('appearance','password_hash');
```

and change it to

```
$default = $this->getServer()->getValue('appearance','password_hash_custom');
```

and save that file and reload browser it will not show this error.

Add Organizational Units, Groups, and Users

LDAP is very flexible. You can create hierarchies and relationships in many different ways, depending on what kind of information you need accessible and what kind of use case you have.

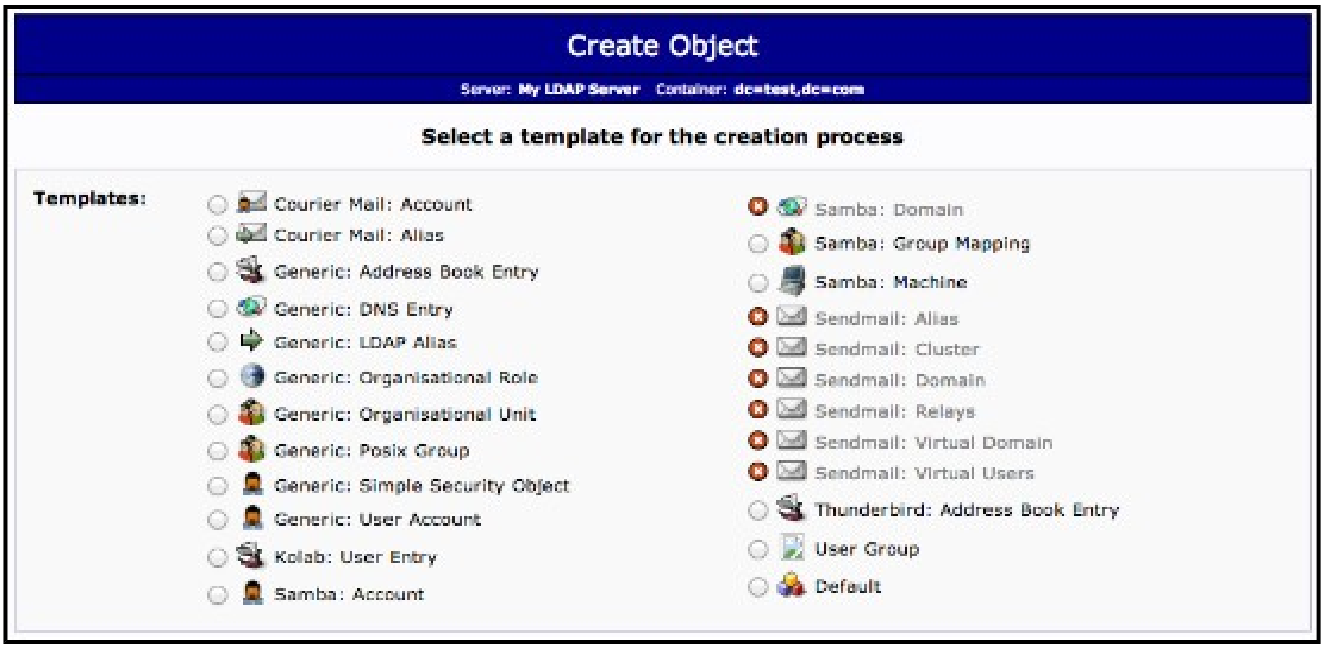
We will create some basic structure to our information and then populate it with information.

Create Organizational Units

First, we will create some categories of information where we will place the later information. Because this is a basic setup, we will only need two categories: groups and users.

Click on the "Create new entry here" link on the left-hand side.

Here, we can see the different kinds of entries we can create.



Create Object

Server: My LDAP Server Container: dc=test,dc=com

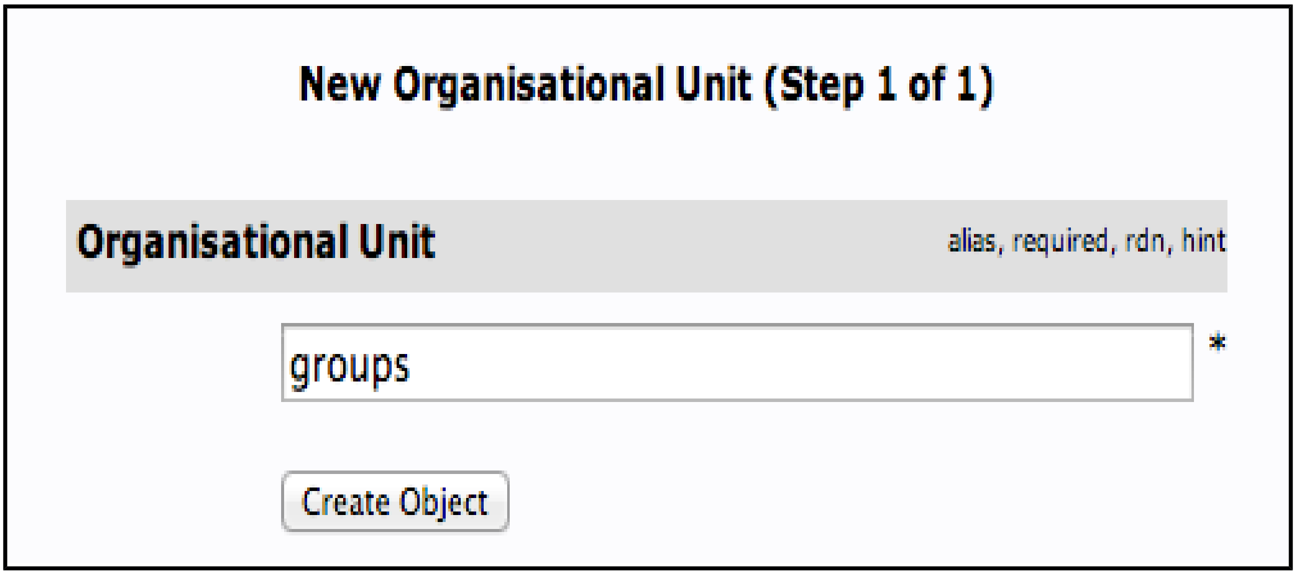
Select a template for the creation process

Templates:

○ Courier Mail: Account
○ Courier Mail: Alias
○ Generic: Address Book Entry
○ Generic: DNS Entry
○ Generic: LDAP Alias
○ Generic: Organisational Role
○ Generic: Organisational Unit
○ Generic: Posix Group
○ Generic: Simple Security Object
○ Generic: User Account
○ Kolab: User Entry
○ Samba: Account
⊗ Samba: Domain
○ Samba: Group Mapping
○ Samba: Machine
⊗ Sendmail: Alias
⊗ Sendmail: Cluster
⊗ Sendmail: Domain
⊗ Sendmail: Relays
⊗ Sendmail: Virtual Domain
⊗ Sendmail: Virtual Users
○ Thunderbird: Address Book Entry
○ User Group
○ Default

Because we are only using this as an organizational structure, rather than an information-heavy entry, we will use the "Generic: Organizational Unit" template.

We will be asked to create a name for our organizational unit. Type "groups":

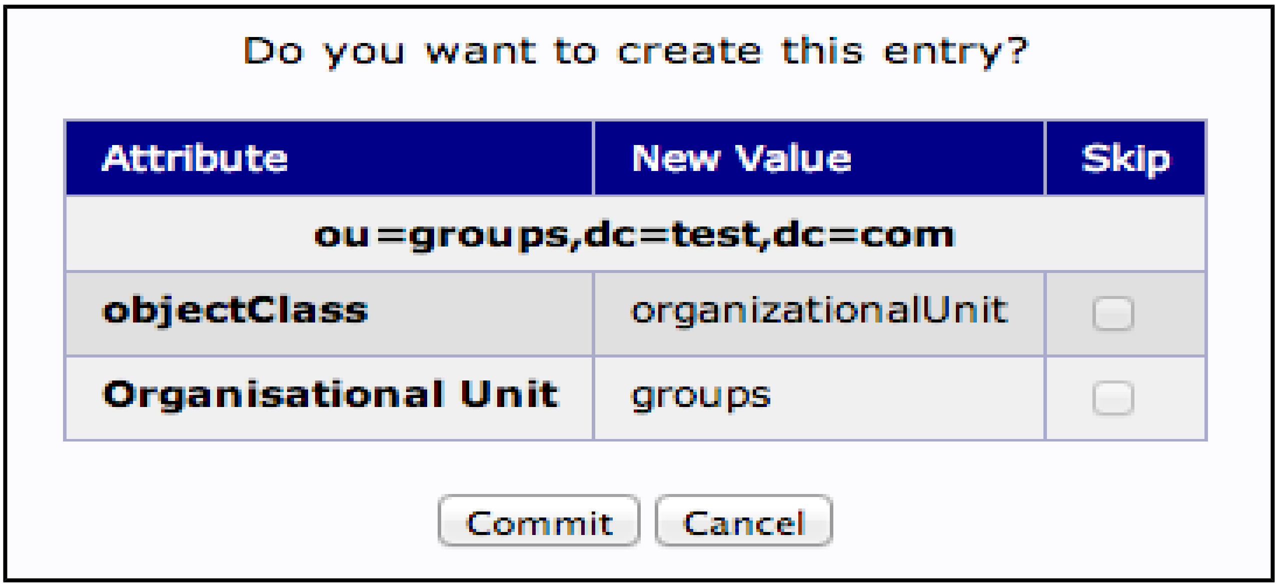


New Organisational Unit (Step 1 of 1)

Organisational Unit alias, required, rdn, hint

groups *

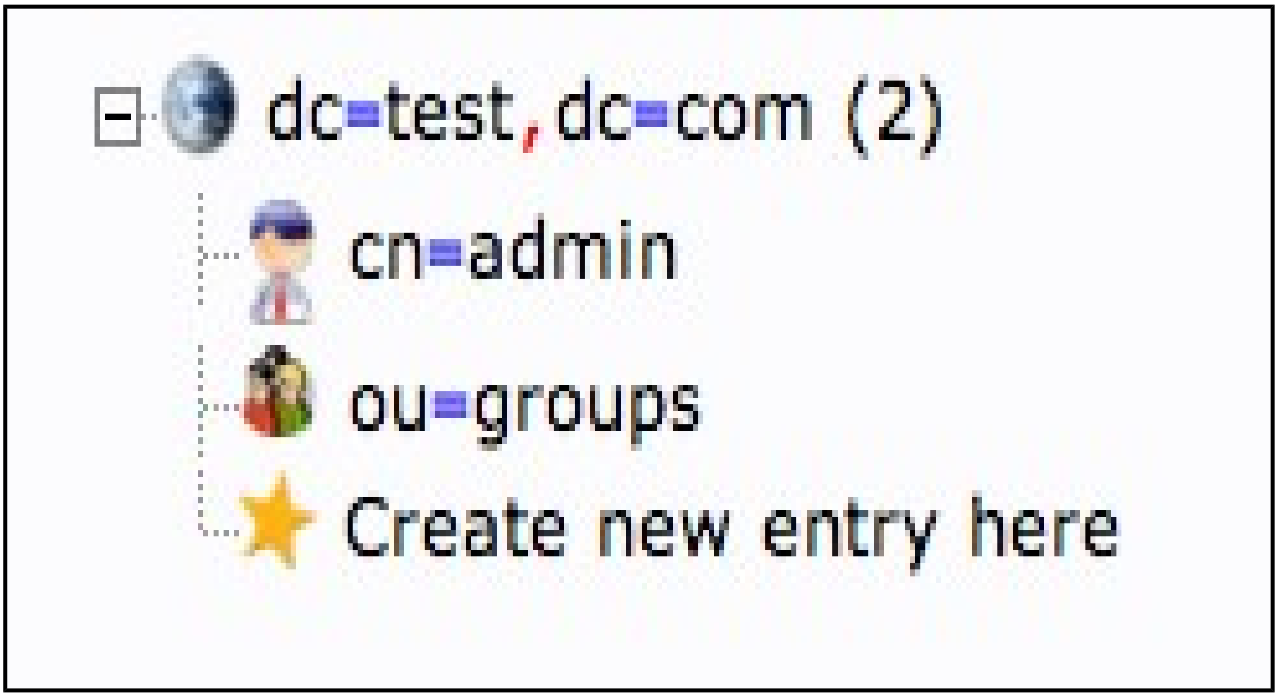
We will then need to commit the changes.



Do you want to create this entry?

Attribute	New Value	Skip
ou=groups,dc=test,dc=com		
objectClass	organizationalUnit	□
Organisational Unit	groups	□

When this is complete, we can see a new entry on the left-hand side.



dc=test,dc=com (2)

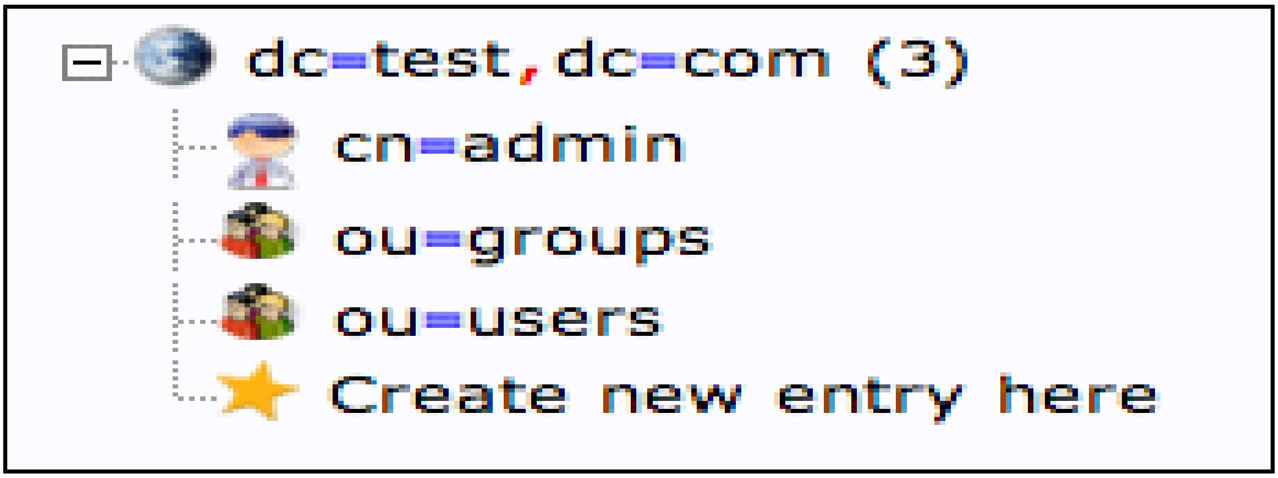
cn=admin

ou=groups

Create new entry here

We will create one more organizational structure to get ourselves going. Repeat the procedure, but this time, use the name "users".

When you are done, you should have something that looks like this:

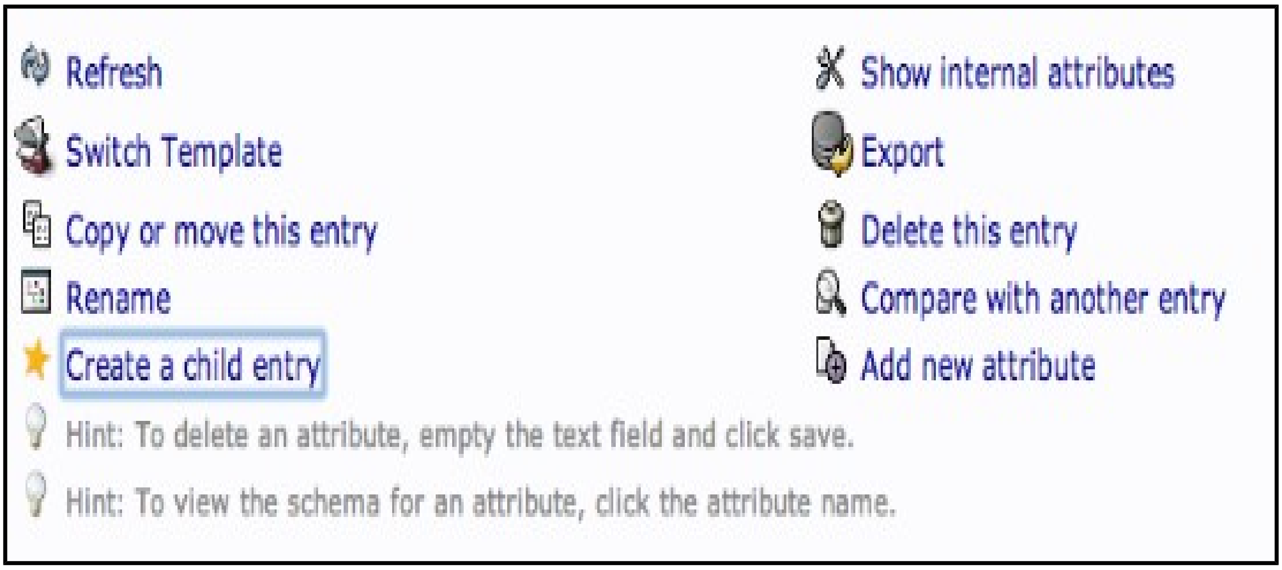


Create Groups

We will be creating three different groups that could be used to organize users into different "access" groups based on the privileges they require.

We will create an "admin" group, an "irc" group, and a "user" group. We could then allow members of different groups to authenticate if we set up client LDAP authentication.

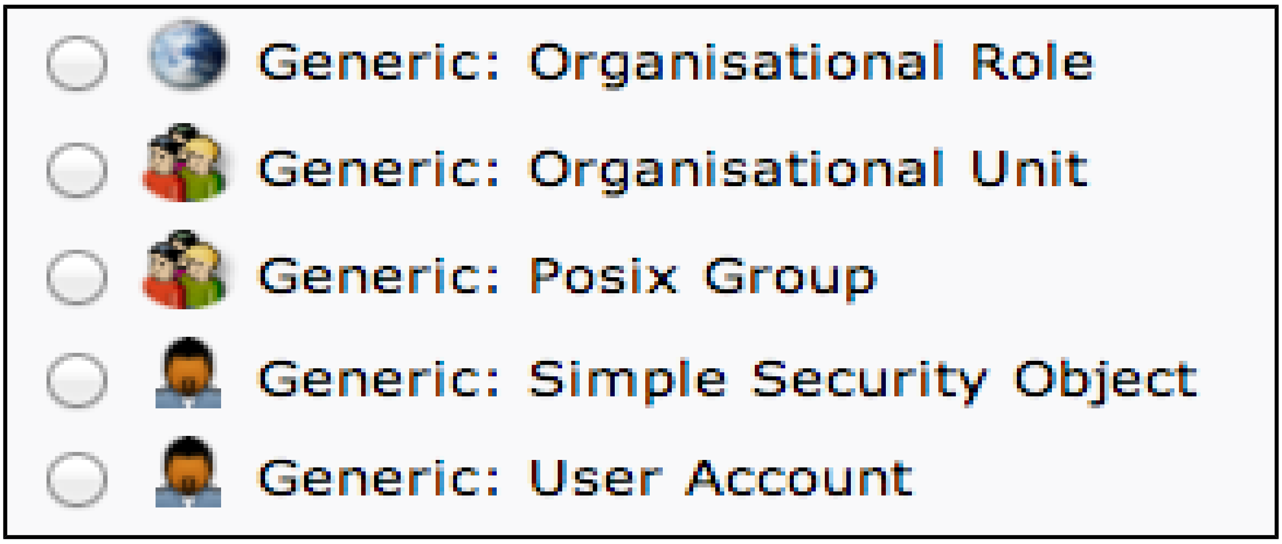
We want to create the groups within the "groups" organizational unit. Click on the "groups" category we created. In the main pane, click on the "Create a child entry" within the groups category.



Hint: To delete an attribute, empty the text field and click save.

Hint: To view the schema for an attribute, click the attribute name.

This time, we will choose the "Generic: Posix Group" category.



○ Generic: Organisational Role

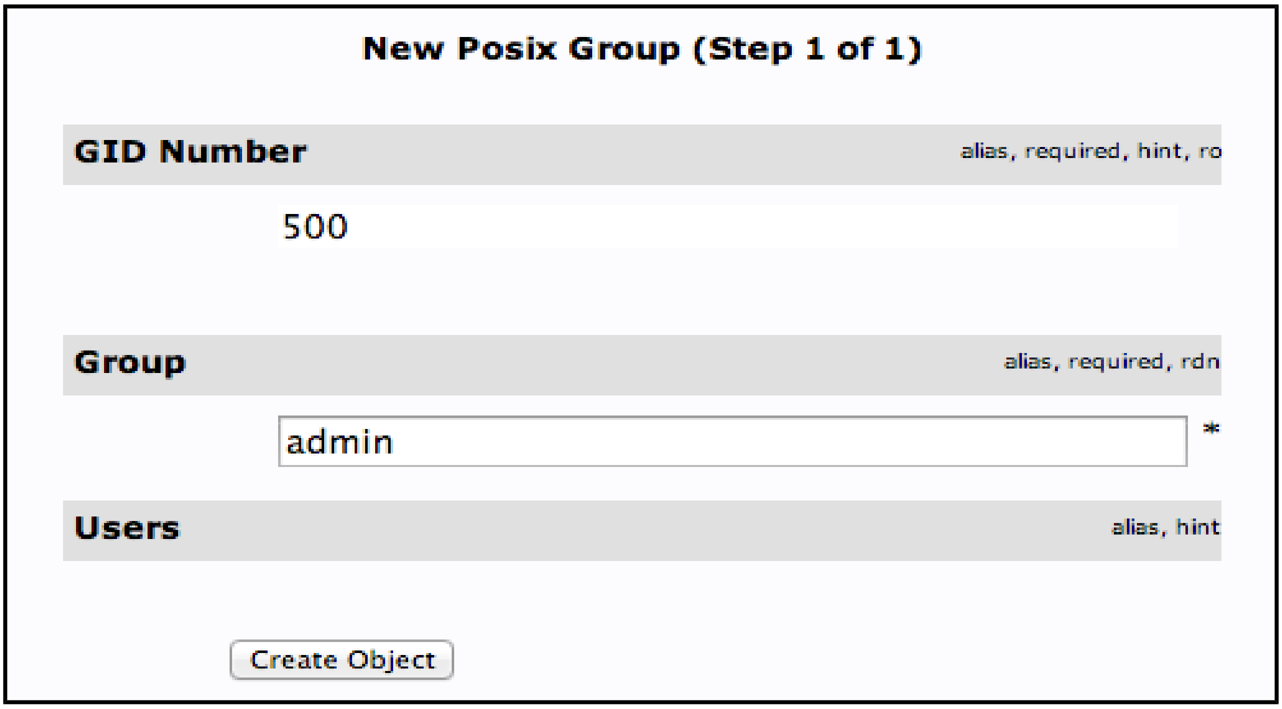
○ Generic: Organisational Unit

○ Generic: Posix Group

○ Generic: Simple Security Object

○ Generic: User Account

Fill in "admin" as the group name. Click "Create Object" and then confirm on the next page.



New Posix Group (Step 1 of 1)

GID Number alias, required, hint, ro

500

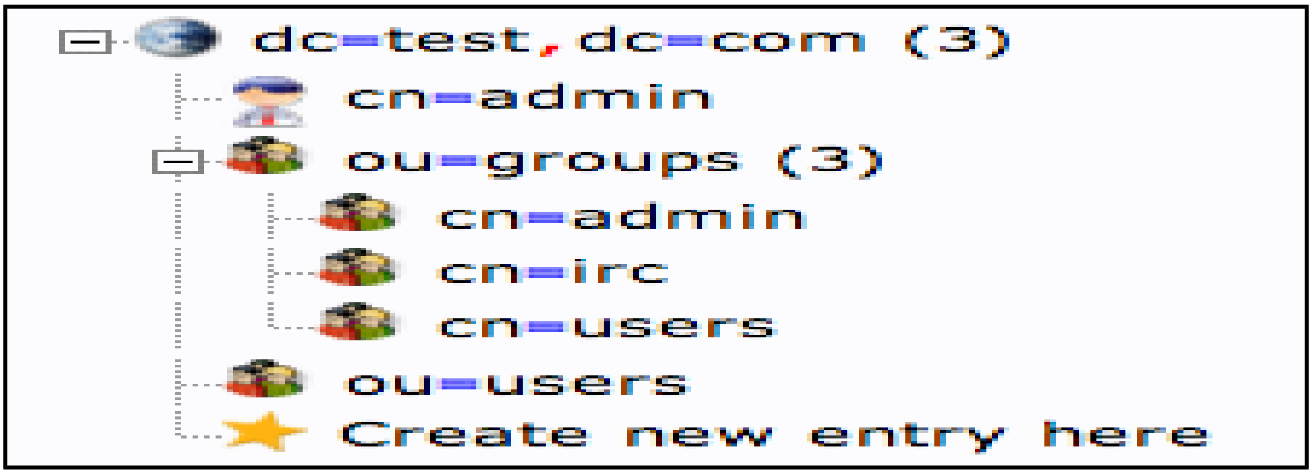
Group alias, required, rdn

admin *

Users alias, hint

Repeat the process, but simply replace the "admin" name with "irc" and "user". Be sure to re-click the "ou=groups" entry before creating child entries, or else you may create entries under the wrong category.

You should now have three groups in the left-hand panel:



dc=test,dc=com (3)

cn=admin

ou=groups (3)

cn=admin

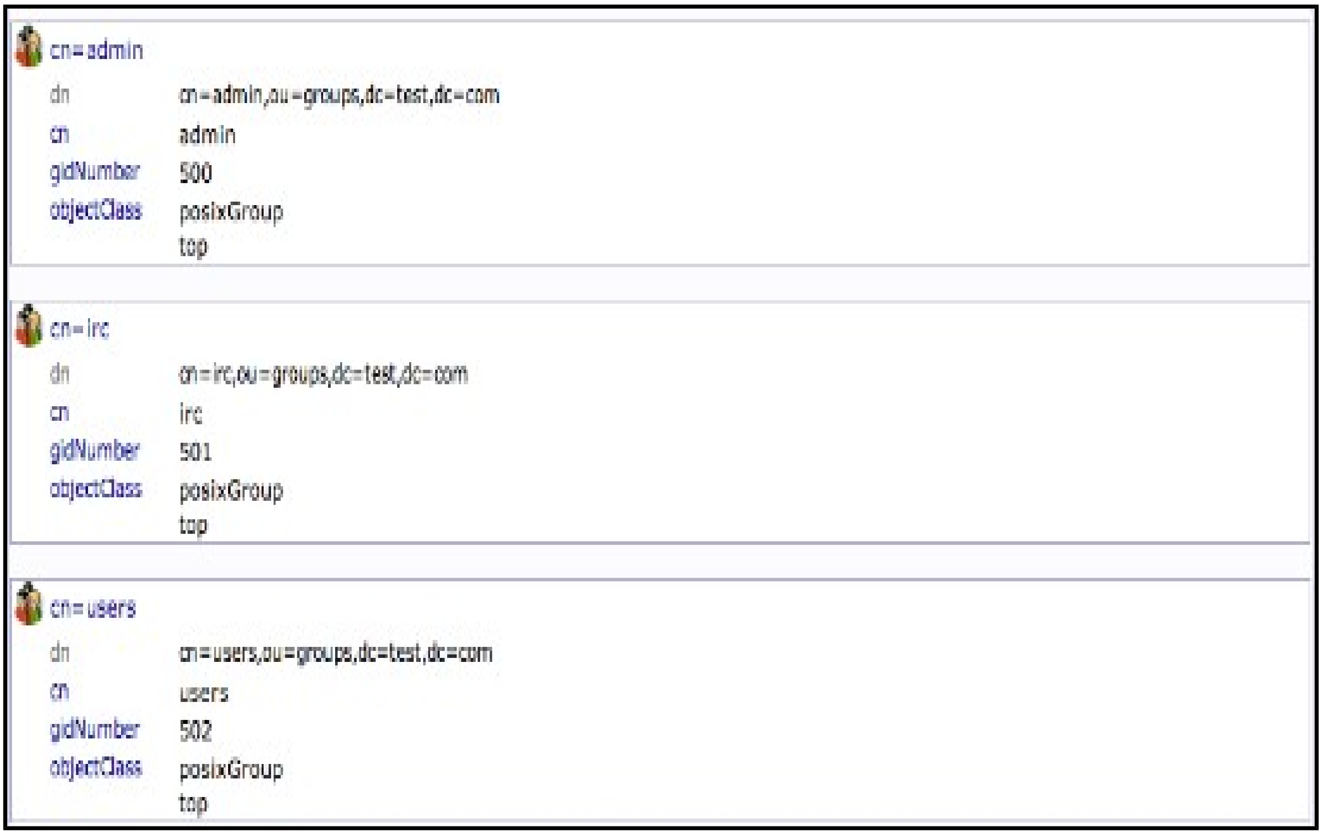
cn=irc

cn=users

ou=users

Create new entry here

You can see an overview of the entries in the "ou=groups" category by clicking on that entry, and then clicking on "View 3 children":



cn=admin

dn	cn=admin,ou=groups,dc=test,dc=com
cn	admin
gidNumber	500
objectClass	posixGroup top

cn=irc

dn	cn=irc,ou=groups,dc=test,dc=com
cn	irc
gidNumber	501
objectClass	posixGroup top

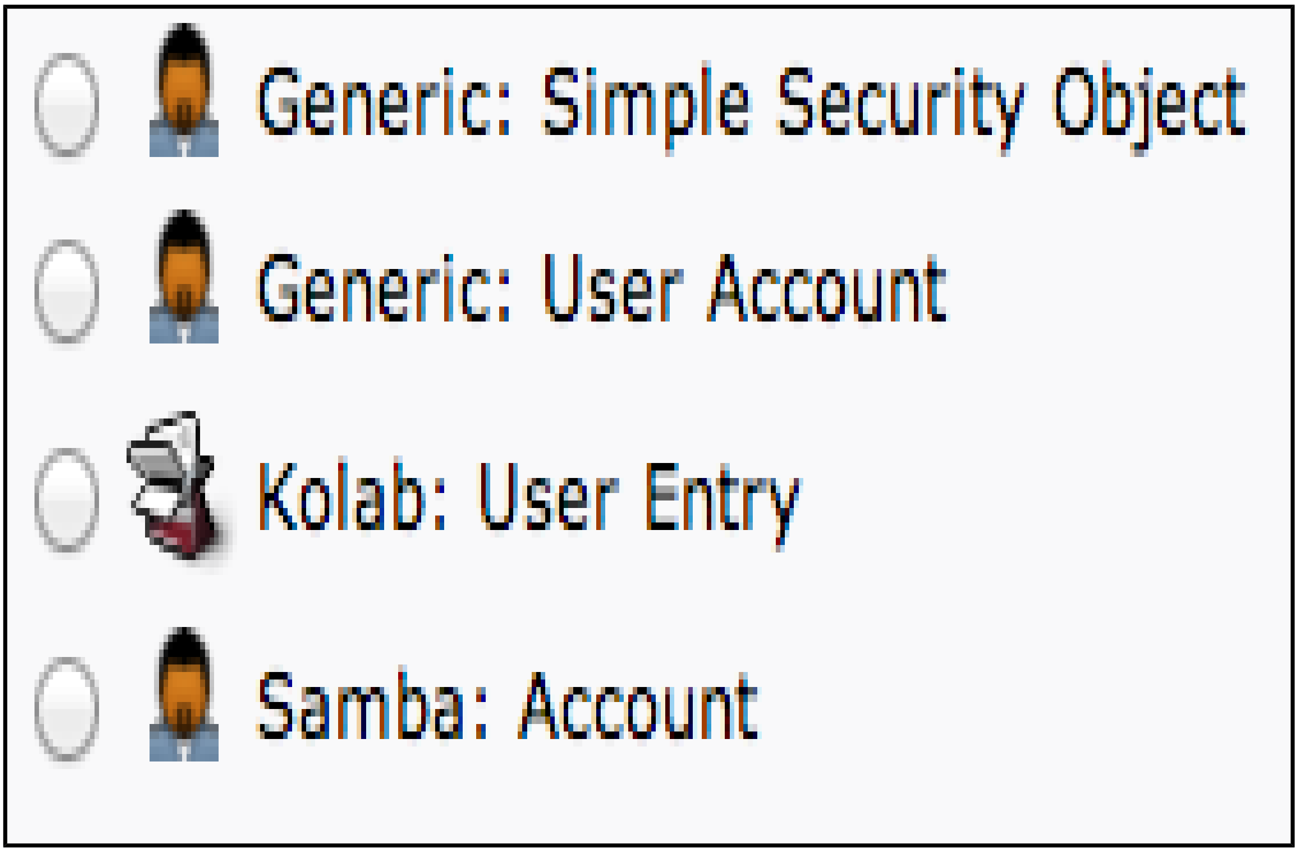
cn=users

dn	cn=users,ou=groups,dc=test,dc=com
cn	users
gidNumber	502
objectClass	posixGroup top

Create Users

Next, we will create users to put in these groups. Start by clicking the "ou=users" category. Click on "Create a child entry".

We will choose "Generic: User Account" for these entries.



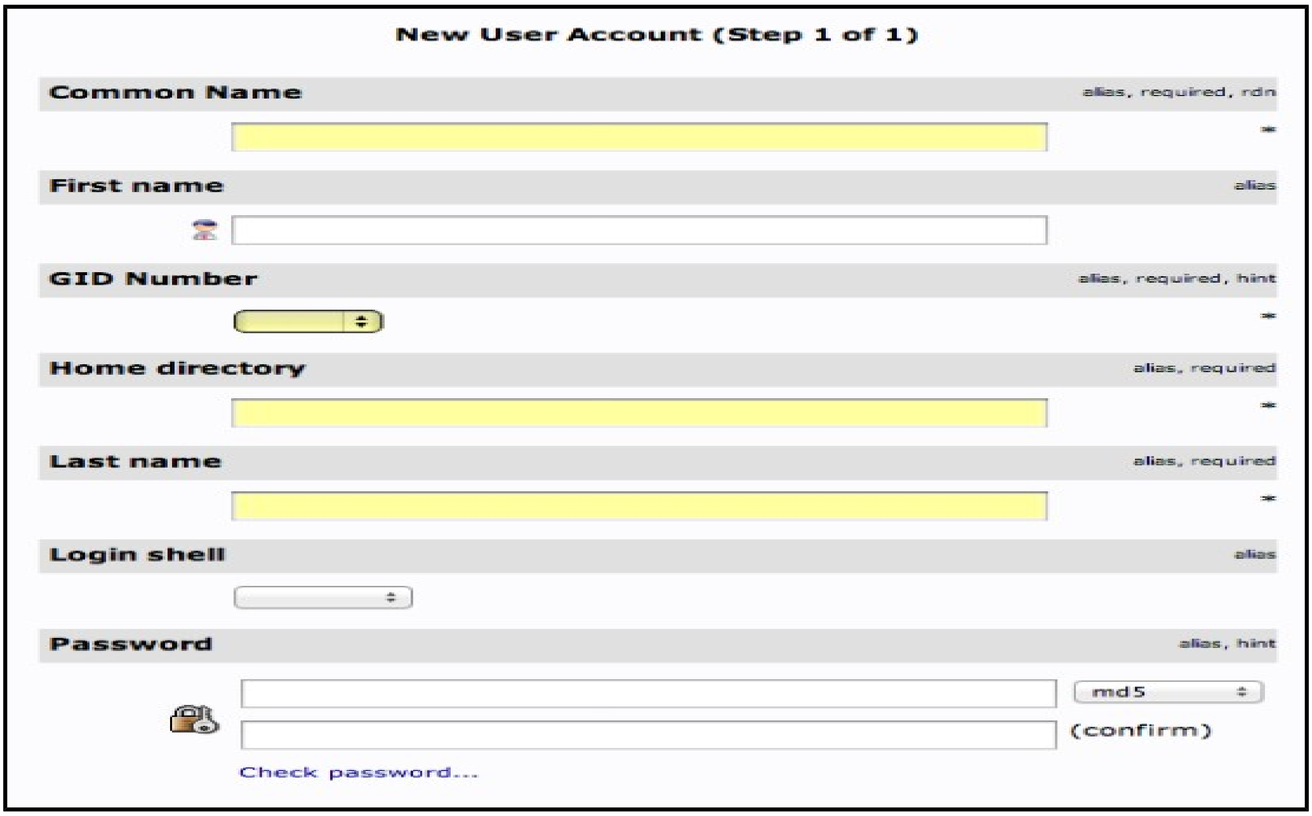
○ Generic: Simple Security Object

○ Generic: User Account

○ Kolab: User Entry

○ Samba: Account

We will be given a lot of fields to fill out:



New User Account (Step 1 of 1)

Common Name alias, required, rdn
*

First name alias

GID Number alias, required, hint
*

Home directory alias, required
*

Last name alias, required
*

Login shell alias

Password alias, hint
md5
(confirm)

Check password...

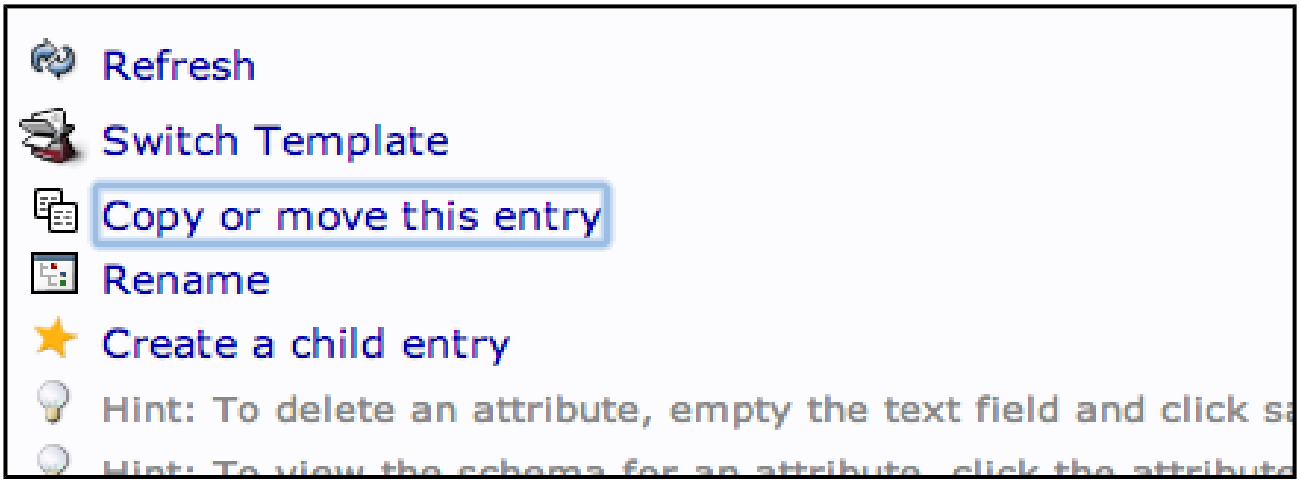
Fill in all of the entries with information that makes sense for your user.

Something to keep in mind is that the "Common Name" needs to be unique for each entry in a category. So you may want to use a username format instead of the default "FirstName LastName" that is auto-populated.

Click "Create Object" at the bottom and confirm on the following page.

To create additional users, we will take advantage of the ability to copy entries.

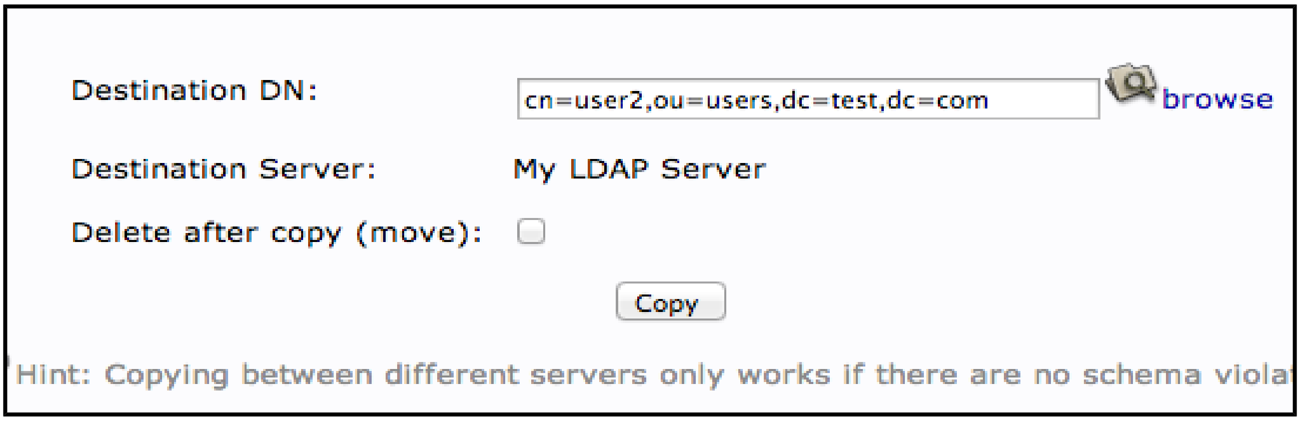
Click on the user you just created in the left-hand panel. In the main pane, click "Copy or move this entry":



Hint: To delete an attribute, empty the text field and click sa

Hint: To view the schema for an attribute, click the attribute

Adjust the "cn=user" portion of the entry to point it to the common name you'd like to use for the new entry. Click "Copy" at the bottom:


Destination DN: cn=user2,ou=users,dc=test,dc=com

Destination Server: My LDAP Server

Delete after copy (move): □

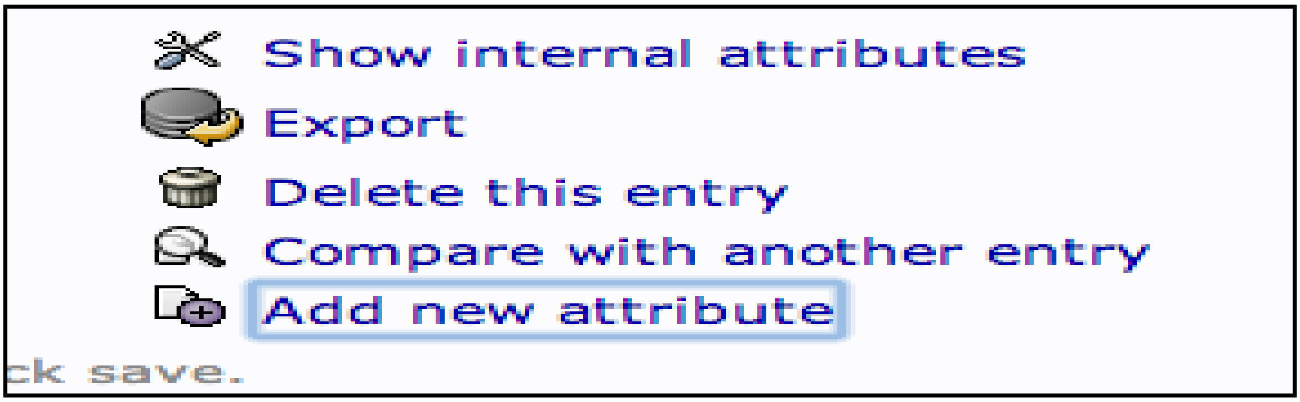
Hint: Copying between different servers only works if there are no schema viola

You will be given the next page populated with your first users data. You will need to adjust it to match the new users information.

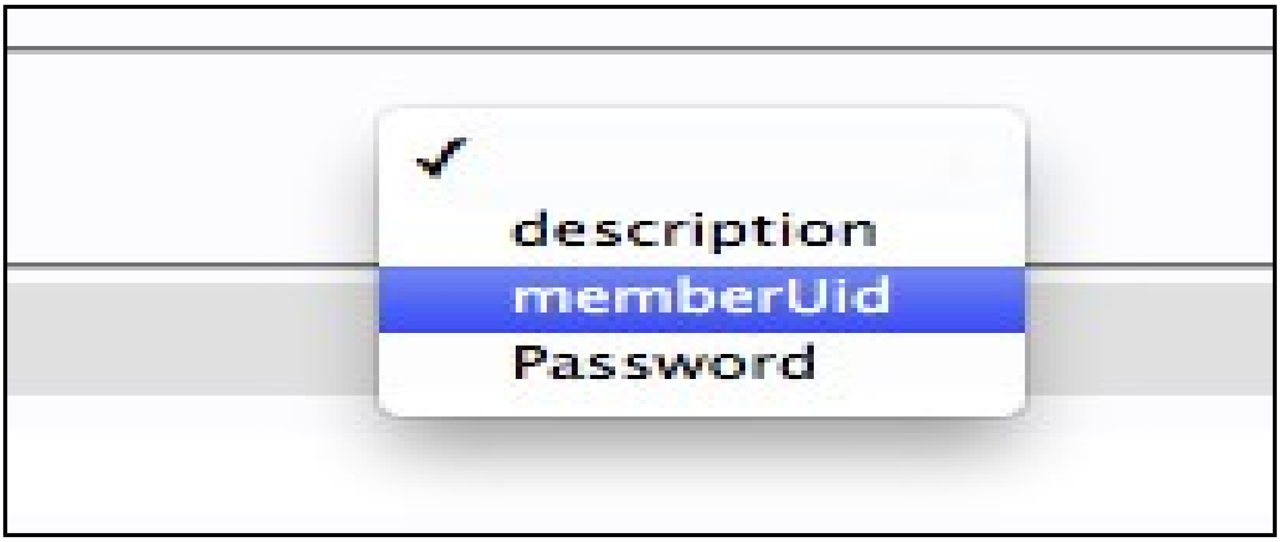
Be sure to adjust the uidNumber. Click the "Create Object" button at the bottom.

Add Users to Groups

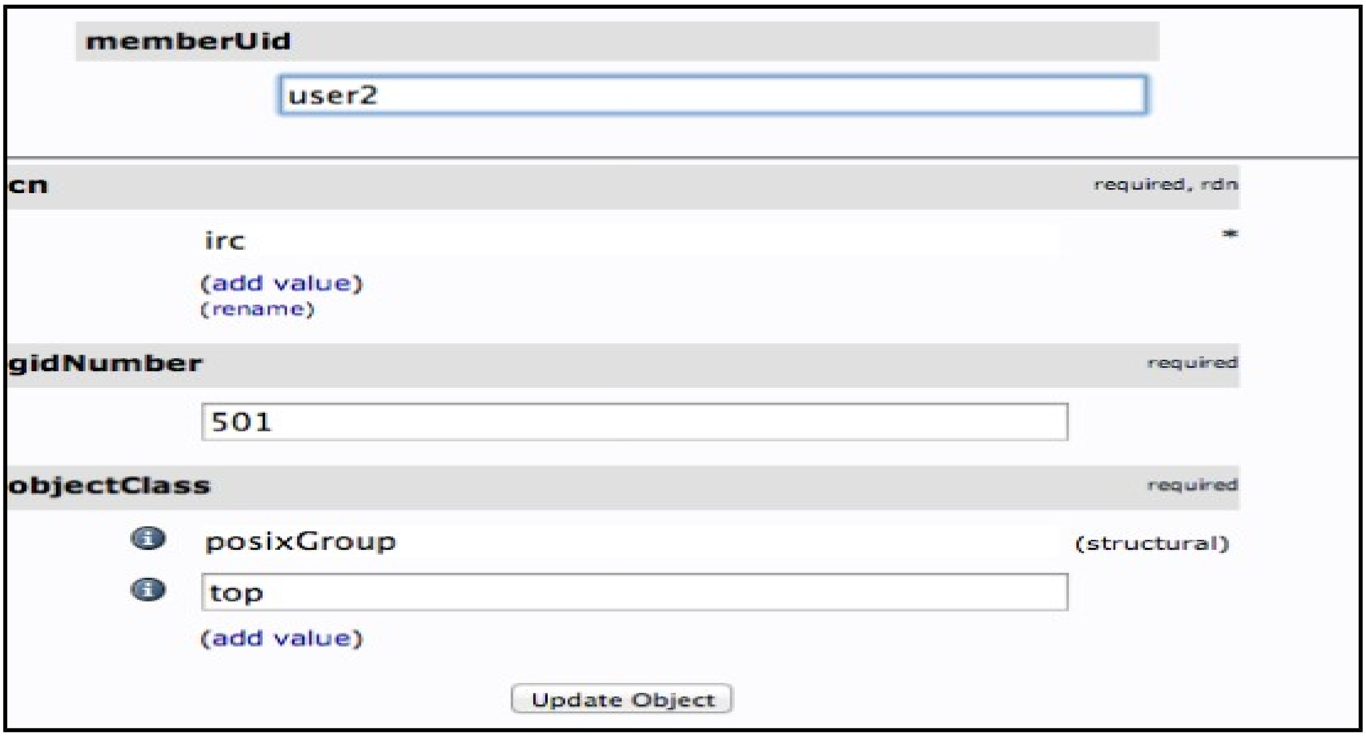
We can add users to various groups by clicking on the group in question. In the main pane, select "Add new attribute":



Select "memberUid" from the drop down menu:



In the text field that populates, enter the first user you'd like to add. Click "Update Object" at the bottom:


memberUid

user2

cn required, rdn

irc *

gidNumber required

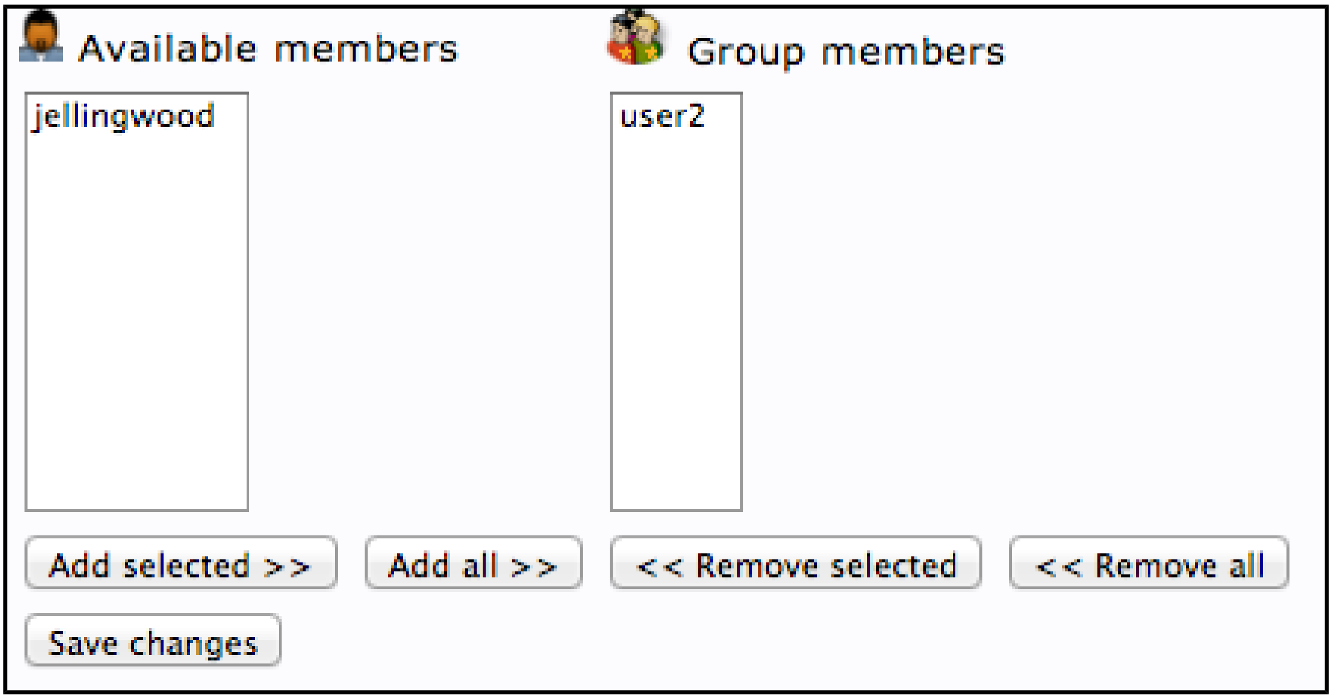
501

objectClass required

posixGroup (structural)

top

You can then add more members by clicking "modify group members" and selecting them from the available choices:



Install Client Packages

On the client machine, you will needs to install a few packages to make authentication function correctly with an LDAP server.

You can install them from the default Ubuntu repositories with the following commands:

```
sudo apt-get update
sudo apt-get install libpam-ldap nscd
```

You will be asked a variety of questions similar to the those asked when you were installing the server components.

- LDAP server Uniform Resource Identifier: ldap://**LDAP-server-IP-Address**
    - Change the initial string from "ldapi:///" to "ldap://" before inputing your server's information
- Distinguished name of the search base:
    - This should match the value you put in your LDAP server's `/etc/phpldapadmin/config.php` file.
    - Search for: " 'server','base',array " within the file.
    - Our example was "**dc=test,dc=com**"
- LDAP version to use: **3**
- Make local root Database admin: **Yes**

- Does the LDAP database require login? **No**
- LDAP account for root:
  - This should also match the value in your `/etc/phpldapadmin/config.php`.
  - Search for: " 'login','bind_id' " within the file
  - Our example was "**cn=admin,dc=test,dc=com**"
- LDAP root account password: **Your-LDAP-root-password**

If you make a mistake and need to change a value, you can go through the menu again by issuing this command:

```
sudo dpkg-reconfigure ldap-auth-config
```

Configure Client Software

We have to adjust a few files to tell our authentication files that they can look to our LDAP server for authentication information.

First, edit the `/etc/nsswitch.conf` file. This will allow us to specify that the LDAP credentials should be modified when users issue authentication change commands.

```
sudo nano /etc/nsswitch.conf
```

The three lines we are interested in are the "passwd", "group", and "shadow" definitions. Modify them to look like this:

```
passwd:         ldap compat
group:          ldap compat
shadow:         ldap compat
```

Next, we will add a value to our PAM configuration.

PAM, or Pluggable Authentication Modules, is a system that connects applications that can provide authentication to applications that require authentication.

PAM is already implemented on most computers, and works behind the scenes without needing user interaction. When we installed and configured our LDAP PAM module, most of the needed information was added to the configuration files.

Edit the `/etc/pam.d/common-session` file:

```
sudo nano /etc/pam.d/common-session
```

Add a line to the bottom of the configuration that reads:

```
session required    pam_mkhomedir.so skel=/etc/skel umask=0022
```

This will create a home directory on the client machine when an LDAP user logs in who does not have a home directory.

We have to restart a service for these changes to be implemented:

```
sudo /etc/init.d/nscd restart
```

Permissions

During the LDAP server configuration, we created a group called "admin". This was not chosen at random. It coincides with the "admin" group that is created by default on Ubuntu machines.

The LDAP users that you added to the "admin" group will have access to the `sudo` command.

This is because we have a line that gives members of the "admin" group sudo access within the `/etc/sudoers` file. Edit the file by issuing this command:

```
sudo visudo
```

There is a line that reads:

```
%admin ALL=(ALL) ALL
```

Entries that begin with a percentage sign (%) specify a group instead of a user. If you wish to disable this functionality, or only grant specific users this functionality, comment out this line:

```
#%admin ALL=(ALL) ALL
```

Log In as an LDAP User

We have now configured our client machine enough to be able to log in as one of our LDAP users. This user does not have to exist on the client machine.

In a new terminal window (it is best to keep your original terminal window logged in, in case of a configuration mistake), ssh into the client machine using an LDAP user's credentials:

```
ssh LDAP_user@LDAP_client_IP_Address
```

You should be able to log in as if your user had been created locally. Issue the print working directory command:

```
pwd
```

You should see that the home directory you selected for your user on the LDAP server is being used on this machine. It has been created on-demand to serve the LDAP user.

If you log out and log in with a different LDAP user, you can see that there will be two home directory entries:

```
ls /home
```

```
user1  user2
```

If your user is part of the "admin" group and you didn't disable the ability in the previous section, you will have normal sudo access, otherwise, you will not.

If you issue the `passwd` command to change your password, you can see that it will be modifying your LDAP credentials:

```
passwd
```

```
Enter login(LDAP) password:
```

Restricting Access by Group

If you only want members of certain groups to be able to log into this specific machine, you can configure that restriction within the PAM files.

Edit the following file with root privileges:

```
sudo nano /etc/pam.d/common-auth
```

At the bottom, we will specify that PAM should look at the security access file to see how to restrict user logins. Add this to the bottom:

```
auth    required    pam_access.so
```

Save and close the file.

The file that PAM references for security information when that setting is configured is at `/etc/security/access.conf`. Open this file now, with root privileges:

```
sudo nano /etc/security/access.conf
```

We need to add a rule to the end of the file.

The dash (-) at the beginning of the line means this is a restriction. From the first colon (:) to the next colon, we specify who this rule applies to.

We specify that this applies to all users except root and the group "admin". Groups are given within parentheses.

From the second colon to the end of the line, we will specify under which circumstances the rule should apply. In our case, the restriction will apply in all circumstances but local logins.

```
-:ALL EXCEPT root (admin):ALL EXCEPT LOCAL
```

This will allow us to restrict logins to the "admin" group. We can add other groups or change the group.

This will also allow us to log in through the "console access" button on the DigitalOcean console if we somehow lock ourselves out of SSH.

Keep in mind that this will apply to all users, not just LDAP users. So any users you create on the client machine will need to be a member of one of the specified groups.