Якимчук Сергій

Linux-сервери. Побудова корпоративної доменної інфраструктури.

2012
http://yakim.org.ua

Зміст

Вступ

Даний курс, другий в циклі, розрахований на те, що б показати, як за допомогою використання вільного програмного забезпечення побудувати повноцінну корпоративну офісну мережу.

На відміну від невеликої мережі, при побудові серйозної інфраструктури вже не обійтися без якого-небудь централізованого сховища облікових записів користувачів. Традиційно для цього розгортається контролер домену на Windows-сервері. Все було б добре, якби не його вартість. Якщо говорити про мережі на кілька тисяч хостів, то, напевно, вартість покупки Windows Server буде виправдана. Але що робити в мережі, на кілька десятків (ну нехай до сотні) машин? Бігати до кожної машині вже важко, але пояснити навіщо потрібно витратити близько тисячі у.о. теж не всякому керівництву можна. Є, звичайно, варіант, встановити піратську версію. Але це може дуже дорого обійтися в разі перевірки програмного забезпечення.

Для того, що б системний адміністратор міг полегшити свою роботу і призначений цей курс. У ньому будуть розглянуті різні варіанти розгортання домену на Linux-сервері на основі Samba, а так само показані способи інтеграції різних сервісів з цим доменом.

Передбачається, що слухачі даного курсу вже мають навички роботи з серверами на Linux, а так само розуміють загальний принцип роботи доменної мережі. Для тих, хто тільки що почав знайомитися з цією операційною системою, вкрай рекомендується пройти курс початкової підготовки, наприклад, першу частину мого курсу.

Всі роботи проводяться на основі дистрибутиву Ubuntu Server 12.04 LTS.

Простий домен на Samba

Що таке Samba

Samba — пакет програм, які дозволяють звертатися до мережевих дисків і принтерів на різних операційних системах по протоколу SMB/CIFS. Має клієнтську та серверну частини. Є вільним програмним забезпеченням.

Починаючи з третьої версії Samba надає служби файлів і друку для різних клієнтів Microsoft Windows і може інтегруватися з операційною системою Windows Server, або як основний контролер домену (PDC), або як член домену. Вона також може бути частиною домену Active Directory.

Samba працює на більшості Unix-подібних систем, таких, як Linux, POSIX-сумісних Solaris і Mac OS X Server, на різних варіантах BSD.

Порівняння з Windows Server.

Головними відмінностями від серверних версій Windows є:

- відсутність підтримки для групових політик (непряма підтримка для версії 3.x в принципі можлива, версія Samba 4 буде включати підтримку групових політик)
- відсутність налаштувань профілів користувачів і комп'ютерів
- відсутність підтримки інфраструктури вузлів (sites) і реплікації каталогу у відповідності до налаштування міжвузлових зв'язків

Ще однією особливістю слід вважати те, що Samba працює тільки поверх TCP / IP, тоді як аналогічний сервіс в Windows може надаватися також поверх IPX і NetBEUI. Однак, сама Microsoft в останніх версіях Windows орієнтується на NBT, так що ця відмінність Samba неактуальна.

За твердженнями ITLabs, в умовах багатокористувацького доступу, швидкість роботи в якості ролей файлового і принт-сервера більш ніж в два рази вище в порівнянні з Windows Server 2003 з тими ж ролями.

З чого складається Samba

1. smbd — демон Samba, що забезпечує обслуговування користувачів, котрі хочуть доступитися до загальних документів сервера;
2. nmbd — демон сервера імен NetBIOS, що забезпечує доступ до служб імен NetBIOS через IP, одним словом, завдяки цьому системи під керуванням Windows бачать в своєму мережевому оточенні систему під керуванням Unix-подібних систем;
3. samba-client — пакунок, який дозволяє працювати як клієнту по протоколу SMB на системах Linux:
4. smbstatus — пакунок для моніторингу Samba;
5. smbpasswd — керування паролями Samba;

6. testparm — перевірка конфігураційного файлу Samba;
7. testprns — перевірка конфігурації принтерів.

Побудова простого домену на Samba

Якщо у нас невелика мережа і не планується інтегрувати в домен різні зовнішні сервіси, можна підняти домен виключно за допомогою Samba. У ролі сховища паролів в даному випадку буде виступати стандартний бекенд — smbpasswd.

Встановимо необхідні програми:
#apt-get install samba libpam-smbpass

Далі необхідно привести файл конфігурації */etc/samba/smb.conf* до такого вигляду:

```
[global]
   workgroup = STUDY
   server string = %h server (Samba, Ubuntu)
   dns proxy = no

   log file = /var/log/samba/log.%m
   max log size = 1000
   syslog = 0
   panic action = /usr/share/samba/panic-action %d

   security = user
   encrypt passwords = true
   passdb backend = tdbsam
   obey pam restrictions = yes
   unix password sync = yes
   passwd program = /usr/bin/passwd %u
   passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:* %n\n *password\supdated\ssuccessfully* .
   pam password change = yes
   map to guest = bad user

   domain logons = yes
;   logon path = \\%N\profiles\%U
   logon drive = H:
   logon script = logon.cmd
  add user script = /usr/sbin/adduser --quiet --disabled-password --gecos "" %u
  add machine script  = /usr/sbin/useradd -g machines -c "%u machine account" -d /var/lib/samba -s /bin/false %u
  add group script = /usr/sbin/addgroup --force-badname %g
```

```
   usershare allow guests = yes



[homes]
   comment = Home Directories
   browseable = no
   read only = yes
   create mask = 0700
   directory mask = 0700
   valid users = %S

[netlogon]
   comment = Network Logon Service
   path = /srv/samba/netlogon
   guest ok = yes
   read only = yes

;[profiles]
;   comment = Users profiles
;   path = /home/samba/profiles
;   guest ok = no
;   browseable = no
;   create mask = 0600
;   directory mask = 0700
```

Створюємо теку *netlogon* для роботи нашого контролера домену та порожній файл логон-скрипта:

```
$ sudo mkdir -p /srv/samba/netlogon
$ sudo touch /srv/samba/netlogon/logon.cmd
```

Далі необхідно додати групу *machines*, в якій будуть міститися облікові записи комп'ютерів домену.

```
$ sudo addgroup  machines
```

Перезапускаємо сервіси Samba:

```
$sudo restart smbd
$sudo restart nmbd
```

Створимо співставлення локальної групи на сервері *sudo* та групи доменних адміністраторів:

$sudo net groupmap add ntgroup="Domain Admins" unixgroup=sudo rid=512 type=d

І зробимо нашого користувача *user* адміністратором домену. Для цього перевіримо, що він входить до групи *sudo*:

$ id user

и додамо його до користувачів Samba:

$ sudo smbpasswd -a user

Ще необхідно дати групі *Domain Admins* дійсно права адміністраторів домену:

$ sudo net rpc rights grant -U sysadmin "STUDY\Domain Admins" SeMachineAccountPrivilege SePrintOperatorPrivilege SeAddUsersPrivilege SeDiskOperatorPrivilege SeRemoteShutdownPrivilege

Для перевірки коректності наших налаштувань введемо наш сервер у власний домен:

$ sudo net rpc join

Тепер можна вводити наш домен будь-який Windows-компьютер.

Однак такий домен має дуже і дуже обмежений функціонал. Це пов'язано з тим, що у нас в домені відсутній такий необхідний і корисний сервіс, як LDAP. У зв'язку з цим отримання доступу до облікових записів користувачів можливо буде виключно на рівні Windows-авторизації. Відповідно підключити до домену сторонні сервіси, такі як, наприклад, поштовий сервер, загальна адресна книга і т.д. не вийде.

У зв'язку з цим такий варіант побудови домену не отримав практично ніякого поширення.

Побудова домену на Samba та OpenLDAP

OpenLDAP

OpenLDAP Software — відкрита реалізація LDAP, розроблена проектом OpenLDAP Project. Розповсюджується під власною ліцензією, яка називається OpenLDAP Public License. LDAP — платформо-незалежний протокол. У числі інших є реалізації для різних модифікацій BSD, а також Linux, AIX, HP-UX, Mac OS X, Solaris, Microsoft Windows (NT і спадкоємці — 2000, XP, Vista, Windows 7) та z/OS.

LDAP (англ. Lightweight Directory Access Protocol — «полегшений протокол доступу до каталогів») — протокол прикладного рівня для доступу до служби каталогів X.500, розроблений IETF як полегшений варіант розробленого ITU-T протоколу DAP. LDAP — відносно простий протокол, що використовує TCP/IP та дозволяє проводити операції авторизації (bind), пошуку (search) та порівняння (compare), а також операції додавання, зміни або видалення записів. Зазвичай LDAP-сервер приймає вхідні з'єднання на порт 389 по протоколах TCP або UDP. Для LDAP-сеансів, інкапсульованих в SSL, зазвичай використовується порт 636.

Будь-який запис в каталозі LDAP складається з одного або декількох атрибутів і володіє унікальним ім'ям (DN — англ. Distinguished Name). Унікальне ім'я може виглядати, наприклад, таким чином: «cn=Іван Петров,ou=Співробітники,dc=example, dc=com». Унікальне ім'я складається з одного або декількох відносних унікальних імен (RDN — англ. Relative Distinguished Name), розділених комою. Відносне унікальне ім'я має вигляд ІмяАтрибута = значення. На одному рівні каталогу не може існувати двох записів з однаковими відносними унікальними іменами. У силу такої структури унікального імені запису в каталозі LDAP можна легко представити у вигляді дерева.

Запись может состоять только из тех атрибутов, которые определены в описании класса записи (object class), которые, в свою очередь, объединены в схемы (schema). В схеме определено, какие атрибуты являются для данного класса обязательными, а какие — необязательными.

Встановлення та налаштування OpenLDAP

У нас є сервер з іменем PDC та IP-адресою 192.168.56.10. Піднімати домен будемо з іменем study.local.

Перед початком роботи відкриємо файл */etc/hosts*

#nano /etc/hosts

та допишемо:

192.168.56.10 pdc

192.168.56.10 pdc.study.local

Далі встановимо OpenLDAP:
#apt-get install slapd ldap-utils

При налаштуванні slapd вводимо пароль адміністратора.
Для подальшої роботы до OpenLDAP мають бути підключені такі схеми:
core.ldif
cosine.ldif
nis.ldif
inetorgperson.ldif
openldap.ldif
misc.ldif

Дивимось, які схеми вже підключені:
#ls /etc/ldap/slapd.d/cn\=config/cn\=schema
Ми повинні побачити щось на зразок:
cn={0}core.ldif cn={1}cosine.ldif cn={2}nis.ldif cn={3}inetorgperson.ldif

Підключаемо схеми, яких нам бракує:
#ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/openldap.ldif
#ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/misc.ldif

Тепер встановимо Samba:
#aptitude install samba smbclient smbldap-tools

Тут слід перевірити версію *smb-ldaptools* і, можливо, виправити в ньому помилки. Інформація про це наприкінці розділу!

Розпаковуємо схему Samba:
#zcat /usr/share/doc/samba/examples/LDAP/samba.ldif.gz > /etc/ldap/samba.ldif

Та додаємо її до LDAP:
#ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/samba.ldif

Створюємо файл study.ldif:
#nano /etc/ldap/study.ldif

І заповнюємо його наступними даними:
 # Load modules for database type
dn: cn=module,cn=config
objectclass: olcModuleList
cn: module
olcModuleLoad: back_bdb.la

 # Create directory database
dn: olcDatabase=bdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcBdbConfig
olcDatabase: bdb
 # Domain name (e.g. home.local)
olcSuffix: dc=study,dc=local
 # Location on system where database is stored
olcDbDirectory: /var/lib/ldap
 # Manager of the database
olcRootDN: cn=admin,dc=study,dc=local
olcRootPW: admin
 # Indices in database to speed up searches
olcDbIndex: uid pres,eq
olcDbIndex: cn,sn,mail pres,eq,approx,sub
olcDbIndex: objectClass eq
 # Allow users to change their own password
 # Allow anonymous to authenciate against the password
 # Allow admin to change anyone's password
olcAccess: to attrs=userPassword
by self write
by anonymous auth
by dn.base="cn=admin,dc=study,dc=local" write
*by * none*
 # Allow users to change their own record
 # Allow anyone to read directory
*olcAccess: to **
by self write
by dn.base="cn=admin,dc=study,dc=local" write
*by * read*

У цьому файлі потрібно замінити ім'я домену на своє (наприклад: dc = firma, dc = net для домену firma.net) Замініть у всьому тексті. Будьте уважні.

Встановіть пароль адміністратора у відповідному рядку файлу.

У рядку olcRootPW замініть пароль адміністратора (в даному випадку обліковий запис admin) на свій. З точки зору безпеки погано використовувати для пароля слово «admin».

Теперь додаємо нашу схему в LDAP:
#ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/study.ldif

Та перезапускаємо сервіс LDAP:
#service slapd restart
Тепер додамо індекси для samba. Створимо файл *samba_indexes.ldif*

#nano /etc/ldap/samba_indexes.ldif

Та запишемо в нього:
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: uidNumber eq
olcDbIndex: gidNumber eq
olcDbIndex: loginShell eq
olcDbIndex: uid eq,pres,sub
olcDbIndex: memberUid eq,pres,sub
olcDbIndex: uniqueMember eq,pres
olcDbIndex: sambaSID eq
olcDbIndex: sambaPrimaryGroupSID eq
olcDbIndex: sambaGroupType eq
olcDbIndex: sambaSIDList eq
olcDbIndex: sambaDomainName eq
olcDbIndex: default sub

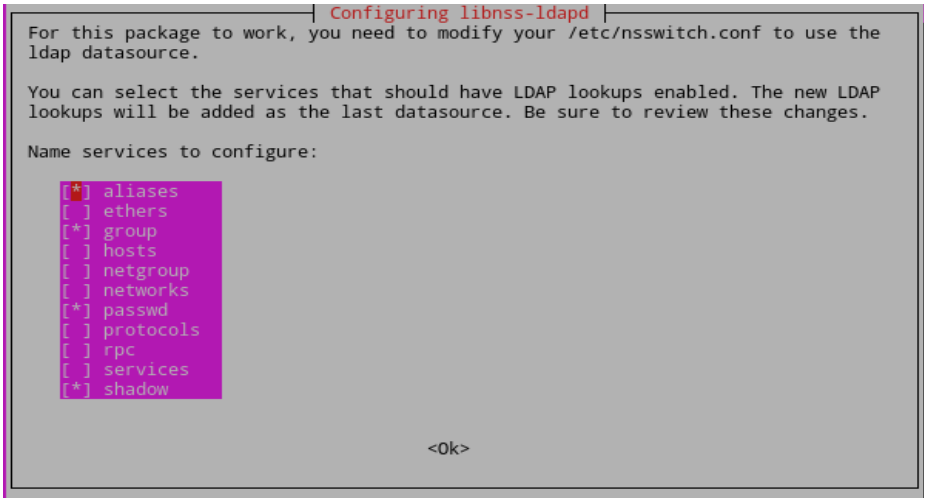
Додаємо в базу:
#ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/samba_indexes.ldif

Редагуємо */etc/ldap/ldap.conf.*
Запишемо в нього:
BASE dc=study,dc=local
URI ldap://192.168.56.10/

Далі перезапускаємо сервіс LDAP:
#service slapd restart

Конфигуруємо наш майбутній контролер домену на використання LDAP-аутентифікації:
#apt-get install libnss-ldapd

На питання, що будуть задані в процесі встановлення та налаштування відповідаємо, як показано на малюнку:



Тепер налаштовуємо pam аутентифікацію, для цього виконуємо команду:
#pam-auth-update
Та ставимо зірочки навпроти LDAP- та UNIX-аутентифікації .

Розпакуємо файли конфігурації smbldap-tool:
#zcat /usr/share/doc/smbldap-tools/examples/smbldap.conf.gz > /etc/smbldap-tools/smbldap.conf
#cp /usr/share/doc/smbldap-tools/examples/smbldap_bind.conf /etc/smbldap-tools/smbldap_bind.conf

Тепер налаштуємо *smbldap-tools*:

Редагуємо /etc/smbldap-tools/smbldap_bind.conf
Так як вторинного контролера домену ми не маємо, то приводимо файл до наступного вигляду:
#slaveDN="cn=Manager,dc=example,dc=com"
#slavePw="secret"
masterDN="cn=admin,dc=study,dc=local"
masterPw="admin"

Тепер нам необхідно перевірити SID нашого домену. Для цього виконуємо команду:
#net getlocalsid pdc
В результаті ми мусимо отримати щось подібне до:
SID for domain pdc is: S-1-5-21-3348427523-2420234745-2521914990

Отриманий SID запам'ятовуємо.

Тепер починаємо редгувати */etc/smbldap-tools/smbldap.conf*
#nano /etc/smbldap-tools/smbldap.conf

В розділі General Configuration:
SID="S-1-5-21-3348427523-2420234745-2521914990"
sambaDomain="STUDY"

В розділі LDAP Configuration:
Коментуємо те, що відноситься до Slave LDAP:
masterLDAP="192.168.56.10"
masterPort="389"
ldapTLS="0"
suffix="dc=study,dc=local"

В розділі SAMBA Configuration:
userSmbHome="\\PDC\%U"
userProfile="\\PDC\profiles\%U"
mailDomain="study.local"

Налаштування Samba

Настав час налаштування Samba.
Редагуємо файл */etc/samba/smb.conf* та приводимо його до такого вигляду:

```
[global]
server string =
workgroup = STUDY
netbios name = pdc

passdb backend = ldapsam:ldap://192.168.56.10
obey pam restrictions = no
security = user
encrypt passwords = yes
unix extensions = no

local master = yes
os level = 255
domain master = yes
preferred master = yes
time server = yes
admin users = admin

log level = 1
log file = /var/log/samba/workstations/%m.log
max log size = 50

getwd cache = yes
```

read raw = yes
write raw = yes
max xmit = 65536
wins support = yes
dns proxy = no
name resolve order = wins hosts bcast lmhosts
wide links = yes

ldap suffix = dc=study,dc=local
ldap user suffix = ou=Users
ldap group suffix = ou=Groups
ldap machine suffix = ou=Computers
ldap idmap suffix = ou=Idmap
ldap admin dn = cn=admin,dc=study,dc=local
ldap ssl = off
ldap passwd sync = yes
ldap delete dn = no
add machine script = /usr/sbin/smbldap-useradd -t 0 -w "%u"
passwd program = /usr/sbin/smbldap-passwd %u
*passwd chat = *New*password* %n\n *Retype*new*password* %n\n *all*authentication*tokens*updated**

domain logons = yes
load printers = no
logon script = startup.bat
logon path =

[netlogon]
path = /home/samba/netlogon
read only = yes
browseable = no

Зберігаємо файл та перевіряємо його коректність:
#testparm

Якщо все нормально — перезапускаємо сервіси Samba:
#service smbd restart
#service nmbd restart

Зверніть увагу, що в даній конфігурації ми не використовуємо переміщувані профілю для клієнтів домену.

Тепер samba необхідно вказати пароль адміна LDAP
#smbpasswd -w admin

Увага! В цій команді вказується саме пароль, а не логін!

Формуємо ldif файл sambadb.ldif
#smbldap-populate -a admin -e /etc/ldap/sambadb.ldif

Де admin — обліковий запис адміністратора.
Для того, що б не користуватися переміщуваним профілем облікового запису admin, відредагуємо файл sambadb.ldif. Видалимо з нього рядки:
sambaHomePath: \\PDC\admin
sambaHomeDrive: H:
sambaProfilePath: \\PDC\profiles\admin

Тепер заповнюємо нашу базу LDAP
#ldapadd -x -D cn=admin,dc=study,dc=local -W -f sambadb.ldif

Встановимо пароль адміну samba такий же, як встановили у LDAP командою smbldap-passwd:
#smbpasswd -a admin

Дамо групі Domain Admins права адміністраторів домену. Для цього виконаємо наступну команду:
#net rpc rights grant "Domain Admins" SeMachineAccountPrivilege SeTakeOwnershipPrivilege SeBackupPrivilege SeRestorePrivilege SeRemoteShutdownPrivilege SePrintOperatorPrivilege SeAddUsersPrivilege SeDiskOperatorPrivilege -Uadmin
Та введемо пароль нашого користувача admin.

Додамо наш сервер до домену:
#net rpc join -U admin

Далі потрібно буде ввести пароль
Enter admin's password:

Вводимо його і в результаті повинні отримати повідомлення:
Joined domain STUDY

Якщо все пройшло нормально (а так і повинно бути), то можна нас привітати — домен готовий до роботи. Тепер можна вводити в нього комп'ютери нашої мережі.

В якості необхідного доповнення розглянемо команди для адміністрування нашого домену за допомогою скриптів smbldap-tools:

- Додавання користувача: **smbldap-useradd -a -P username**
- Видалення користувача: **smbldap-userdel username**
- Додавання групи: **smbldap-groupadd -a groupname**

- Додавання користувача в групу: **smbldap-groupmod -m username groupname**
- Видалення користувача з групи: **smbldap-groupmod -x username groupname**
- Додавання комп'ютера до домену: **smbldap-useradd -t 0 -w username**
- Встановити основну групу користувача: **smbldap-usermod -g groupname username**

Помилки у пакунку smbldap-tools

Так само має сенс звернути увагу на один прикрий, але легко виліковний баг в smbldap-tools. В Ubuntu 12.04 зараз використовується smbldap-tools версії 0.9.7. При використанні будь-якого скрипта для керування доменом з'являється помилка:

Use of qw(...) as parentheses is deprecated at /usr/share/perl5/smbldap_tools.pm line 1423

Виправити цю помилку просто. Відкриваємо файл */usr/share/perl5/smbldap_tools.pm* Шукаємо в ньому рядок:

$sig_name qw(ALRM INT HUP QUIT TERM TSTP TTIN TTOU) {

И міняємо його на:

$sig_name (qw(ALRM INT HUP QUIT TERM TSTP TTIN TTOU)) {

Опис помилки: http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=670246

Ще одна помилка в пакеті smbldap-tools. При спробі додати користувача командою smbldap-useradd -a -P username видається помилка:

Failed to execute: /usr/sbin/smbldap-passwd.cmd: No such file or directory at /usr/sbin/smbldap-useradd line 668.

Лікується теж дуже просто:

 # ln -s /usr/sbin/smbldap-passwd /usr/sbin/smbldap-passwd.cmd

Windows 7 та домен на Samba

Ще одне важливе доповнення. При введенні до домену комп'ютера з Windows 7 швидше за все будуть проблеми. Що б таку машину ввести до домену потрібно буде поправити реєстр.

HKLM\System\CCS\Services\LanmanWorkstation\Parameters
DWORD DomainCompatibilityMode = 1
DWORD DNSNameResolutionRequired = 0

Звісно після цього комп'ютер необхідно перезавантажити.

Після введення комп'ютера до домену можлива поява повідомлення про помилку:

"Changing the Primary Domain DNS name of this computer to "" failed.
The name will remain "MYDOM". The error was:

The specified domain either does not exist or could not be contacted"

У такому випадку необхідно встановити хотфікс, який був спеціально випущений для вирішення цієї проблеми. Детальніше дивитися тут:
http://support.microsoft.com/kb/2171571

А взагалі докладний опис проблем з Windows 7 є тут:
http://wiki.samba.org/index.php/Windows7

Введення Linux-сервера до домену та налаштування на ньому тек спільного доступу з доменною авторизацією.

Встановлюємо необхідні пакети:
#apt-get install samba smbclient winbind

Тепер приводимо файл */etc/samba/smb.conf* до наступного вигляду:
[global]
workgroup = STUDY
realm = STUDY.LOCAL

#Виступаємо в ролі члена домену
security = ads

 # Запити про користувачів будемо передавати через winbind
auth methods = winbind
password server = 192.168.56.10

*idmap config * : range = 10000-20000*
*idmap config * : backend = tdb*

winbind enum users = Yes
winbind enum groups = Yes
winbind use default domain = Yes

#Паролі приймаємо у шифрованому вигляді
encrypt passwords = yes

#Вказуємо, що наш сервер не контролер домену
preferred master = No
domain master = No
domain logons = no
os level = 10
case sensitive = No
panic action = /usr/share/samba/panic-action %d
log file = /var/log/samba/log.%m
debug level = 0
syslog = 0

[share]
comment = share
path = /var/local/share
valid users = @"STUDY\Domain Users"
read list = @"STUDY\Domain Users"

write list = "STUDY\student"

Далі відкриваємо на редагування файл */etc/nsswitch.conf* та змінюємо в ньому:
passwd: compat winbind
group: compat winbind
shadow: compat winbind

Після цього можна вводити комп'ютер до домену:
#net rpc join -U admin

Якщо все було зроблено правильно, то повинні отримати рядок на зразок:
Joined domain STUDY.

Тепер необхідно перезапустити сервіс winbind:
 # service winbind restart

Зараз необхідно перевірити, що у нас все працює як треба.
По команді
#wbinfo -g
ми повинні побачити список доменних груп, а результатом виконання
#wbinfo -u
має стати список користувачів.

Крім цього, по команді
#id admin
повинна видатися інформація про доменного користувача admin.

Створимо теку загального доступу, що описана у файлі smb.conf
mkdir /var/local/share

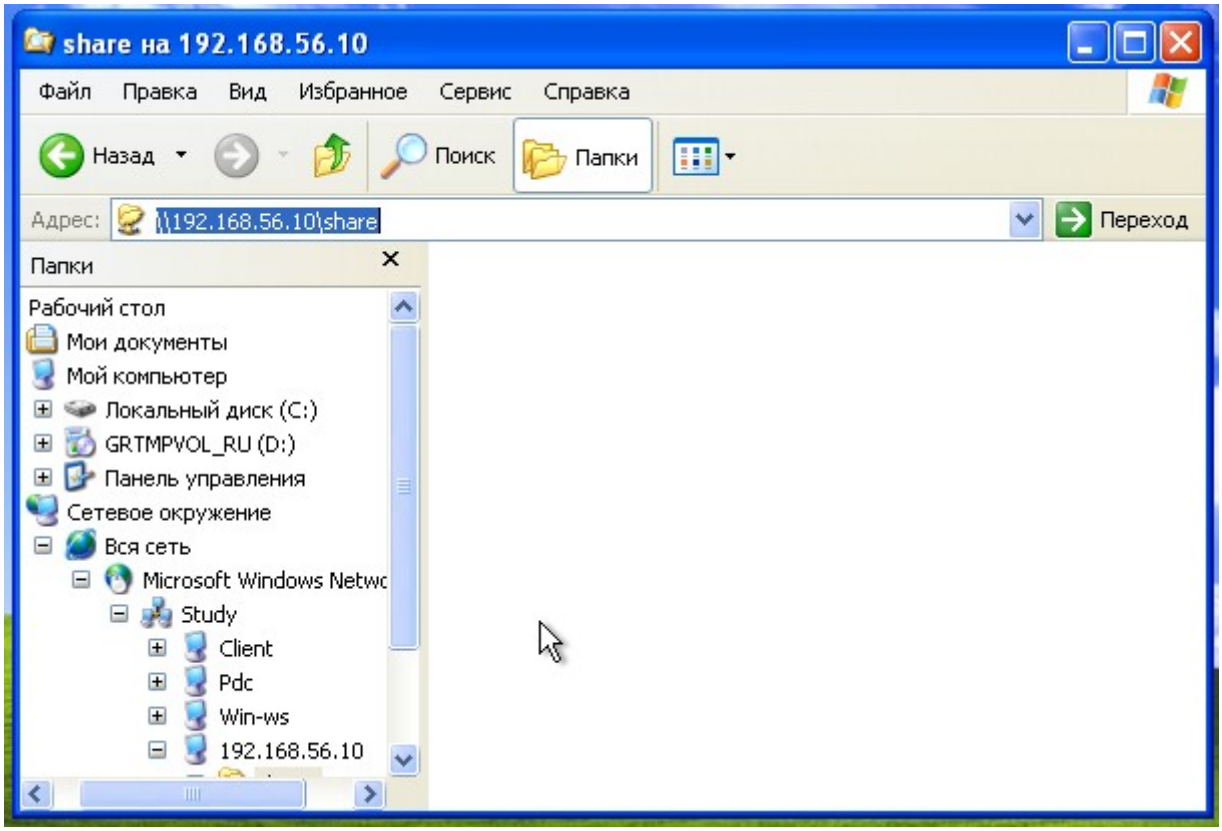
і виставимо на неї права повного доступу:
chmod 777 /var/local/share

Зверніть увагу, що права доступу по мережі регулюються через Samba і право на запис має тільки доменний користувач student.

Перевірка роботи прав доступу

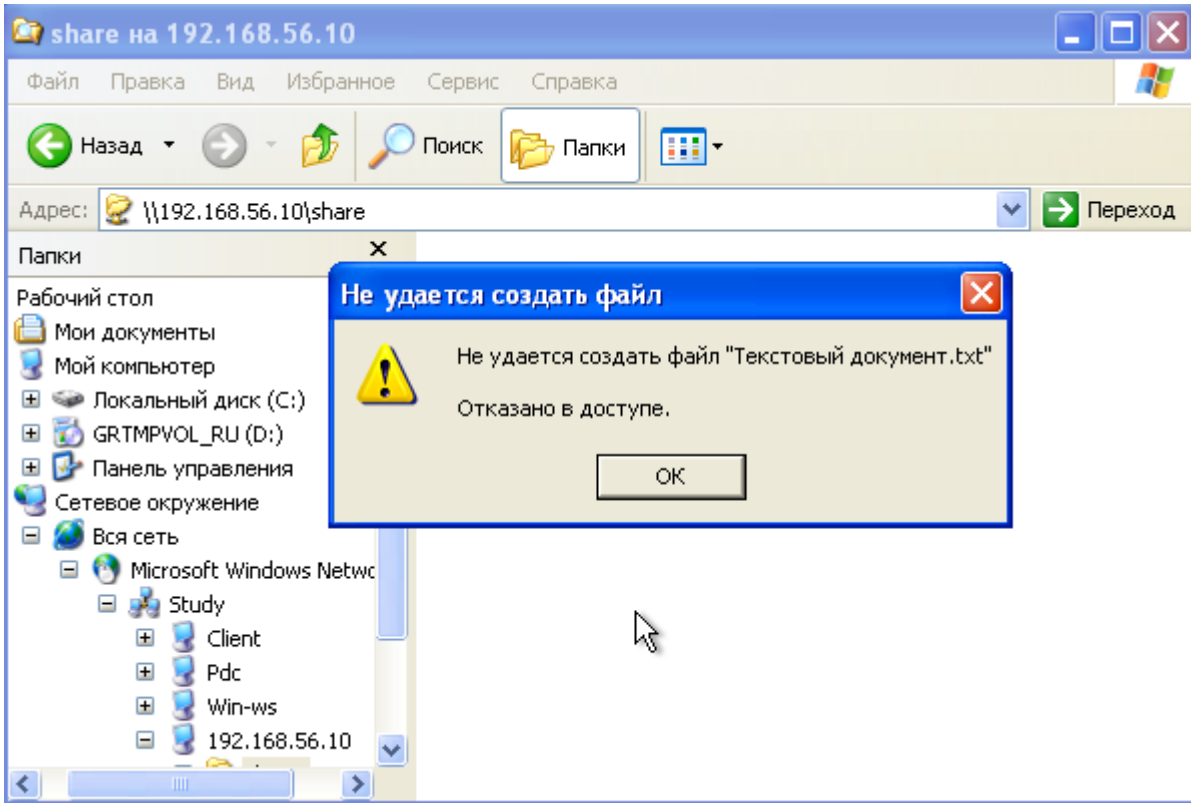
На комп'ютері з Windows XP, який вже введений до нашого домену, увійдіть з правами доменного адміністратора (логін admin і пароль admin).

Далі відкриємо теку загального доступу на Samba:



і спробуємо створити там якийсь файл.

У нас з'явиться повідомлення про помилку, так як право на запис до цієї теки ми дали тільки для користувача student.



Якщо ж ми залогінимось під користувачем student, то до цієї теки ми будемо мати повний доступ.

Phpldapadmin

Не завжди зручно керувати користувачами і групами домену з консолі. Тому розглянемо веб-інтерфейс для OpenLDAP — phpLdapAdmin.

PhpLdapAdmin — це веб-інтерфейс для керування сервером LDAP. Він написаний на php і поширюється під ліценцією GNU/GPL.

Проект з'явився восени 2002 року, коли Дейву Сміту, студенту з університету Брігама Янга (BYU) і ведучому розробнику, знадобився надійний веб-додаток, для керування своїми LDAP-серверами. Спочатку проект phpLDAPadmin називався DaveDAP, але в серпні 2003 року назва була змінена на phpLDAPadmin. З того часу ця програма завантажується в середньому близько 150 разів на день, і широко використовується у всьому світі.

Ще двоє інших розробників внесли свій внесок у кодову базу: Ксав'єр Ренард і Уве Ебель. Ксавьє займався LDIF імпортом / експортом та інтеграцією з сервером Samba. Заняттям Уве стала інтернаціоналізація додатку. Навесні 2005 року, Деон Джордж взяв на себе підтримку phpLDAPadmin.

Встановдення та налаштування phpLdapAdmin

Встановимо phpLdapAdmin.

Для подальшої роботи нам будуть потрібні веб-сервер Apache та php. Встановимо їх:

$ sudo aptitude install apache2 php5

Далі встановимо phpLdapAdmin:
$ sudo aptitude install phpldapadmin

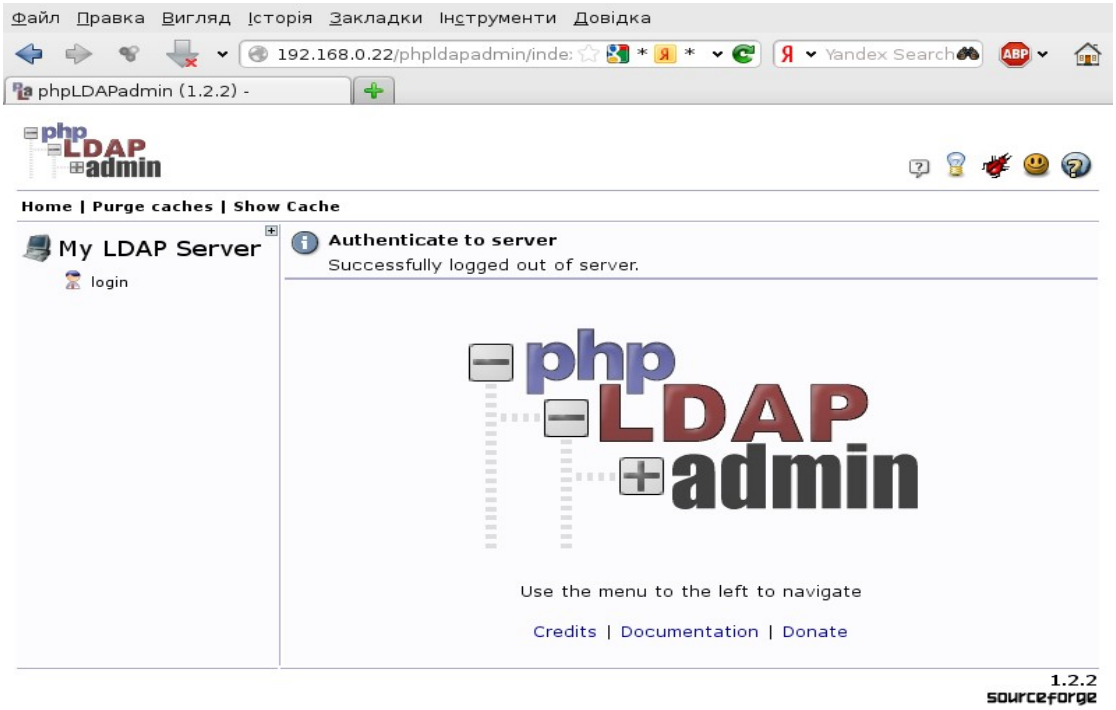
Тепер треба налаштувати цю програму.
Відкриємо файл конфігурації */etc/phpldapadmin/config.php*
та знайдемо там рядки:

$servers->setValue('server','base',array('dc=example,dc=com'));
$servers->setValue('login','bind_id','cn=admin,dc=example,dc=com');

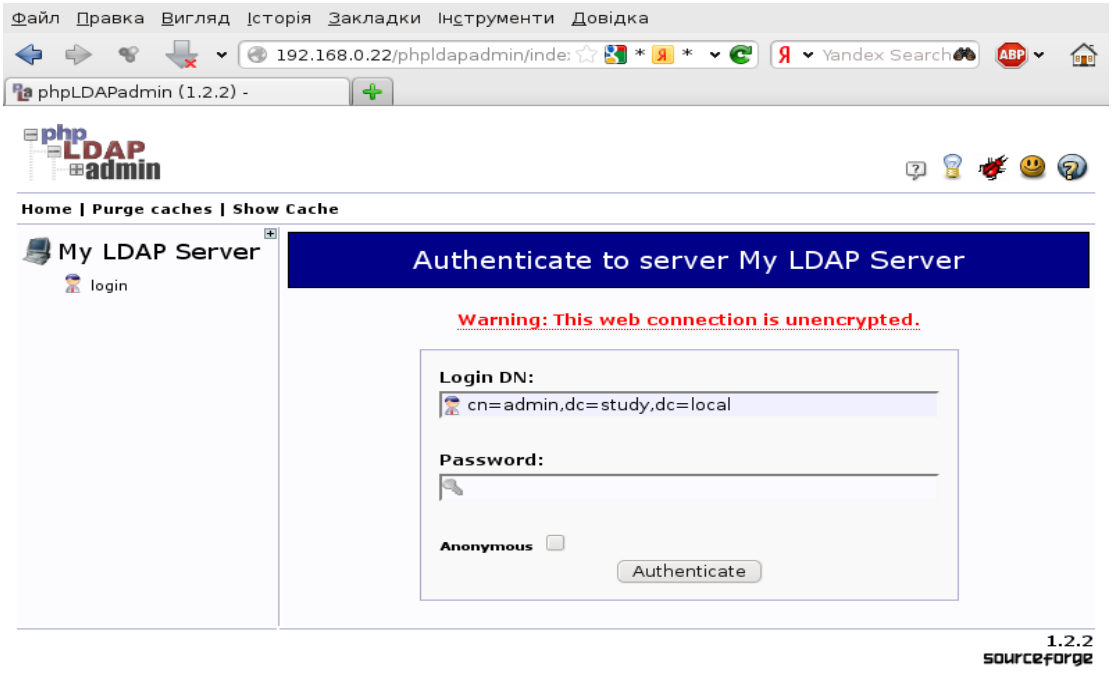
Змінимо їх, відповідно на:

$servers->setValue('server','base',array('dc=study,dc=local'));
$servers->setValue('login','bind_id','cn=admin,dc=study,dc=local');

На цьому налаштування phpLdapAdmin завершено.
Відкриємо його веб-інтерфейс за адресою *http://pdc/phpldapadmin*

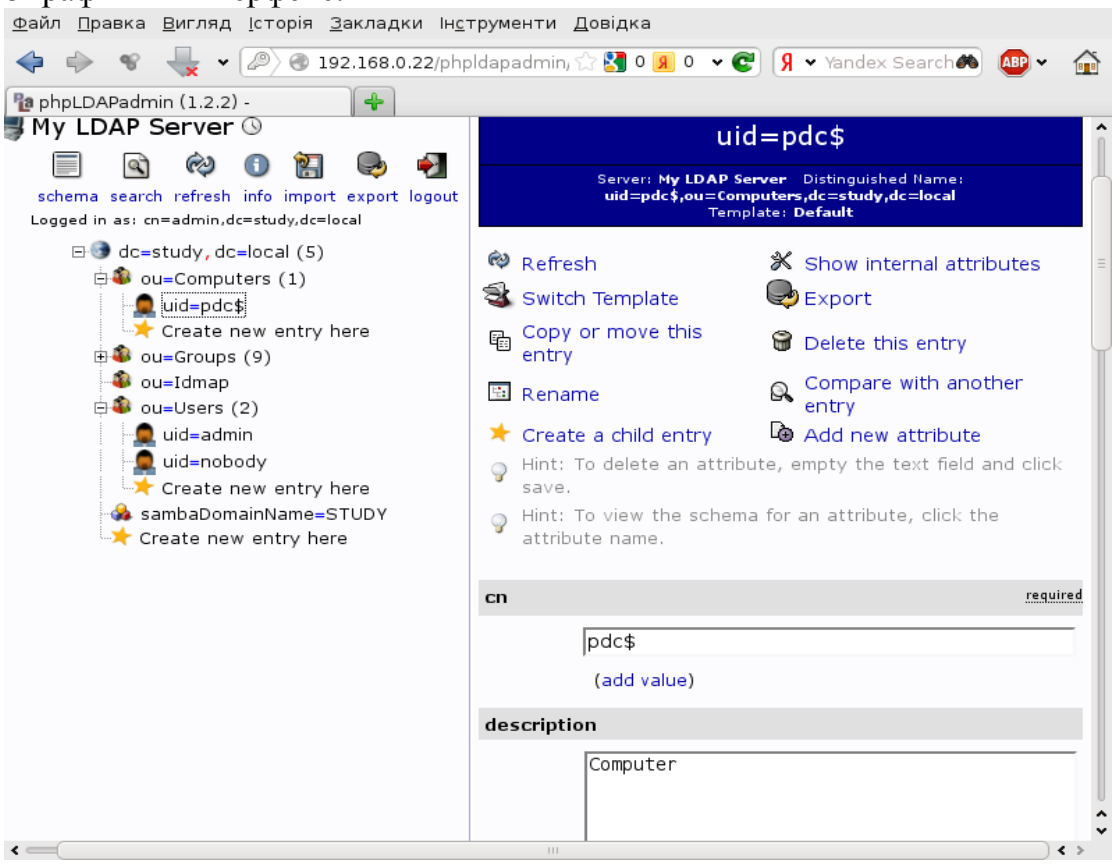


Натиснемо на “login” та введемо пароль до вказаного облікового запису:



Тепер нам доступні можливості керування користувачами не з командного рядка,

а через графічний інтерфейс.



Налаштування поштового серверу з інтеграцією в домені

Сьогодні кожна компанія використовує електронну пошту як один з основних засобів комунікацій у бізнесі. Необхідною умовою ефективного застосування пошти є наявність спеціальної програми — поштового сервера. Створювати корпоративні ящики з допомогою публічних сервісів в Інтернеті вже давно вважається свого роду поганим тоном. До того ж, такий підхід абсолютно небезпечний і супроводжується труднощами в керуванні.

Використання власного корпоративного поштового сервера надає ряд переваг:

- можливість створення необмеженої кількості поштових доменів (для різних компаній, юридичних осіб, напрямків, підрозділів) на одному сервері;
- можливість створення необмеженої кількості поштових скриньок для різних поштових доменів;
- можливість контролю вхідної та вихідної пошти;
- створення поштових скриньок необмежених розмірів (обмеження обумовлюється тільки технічними характеристиками сервера);
- підтримка протоколів передачі з шифруванням пошти ;
- можливість централізованого зберігання пошти на сервері;
- антивірусний захист вхідної та вихідної пошти;
- захист від спаму;
- web-інтерфейс доступу до електронної пошти з шифруванням даних, що передаються.

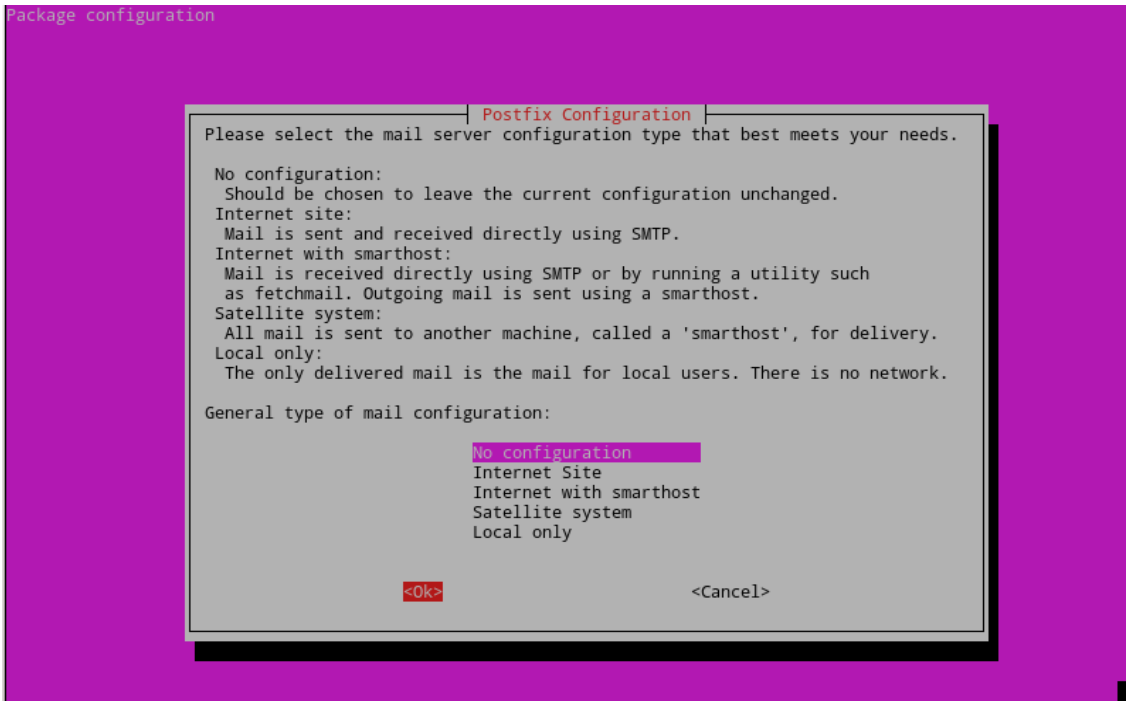
Розглянемо налаштування поштового сервера на основі Postfix і Dovecot із зберіганням облікових записів користувачів у LDAP.

Встановлення Postfix та Dovecot

Встановимо необхідні пакунки програм:

 # apt-get install postfix postfix-ldap dovecot-common dovecot-imapd dovecot-ldap

Після встановлення нам буде запропонований автоматичний варіант налаштування сервера Postfix. Виберемо варіант "No configuration":



Підготовка до налаштування поштового сервера:

Створимо місце для зберігання пошти для нашого домену study.local
#mkdir -p /var/spool/mail/study.local

Створимо групу *virtual* і користувача *virtual*:
#groupadd -g 5000 virtual
#useradd -g virtual -u 5000 virtual

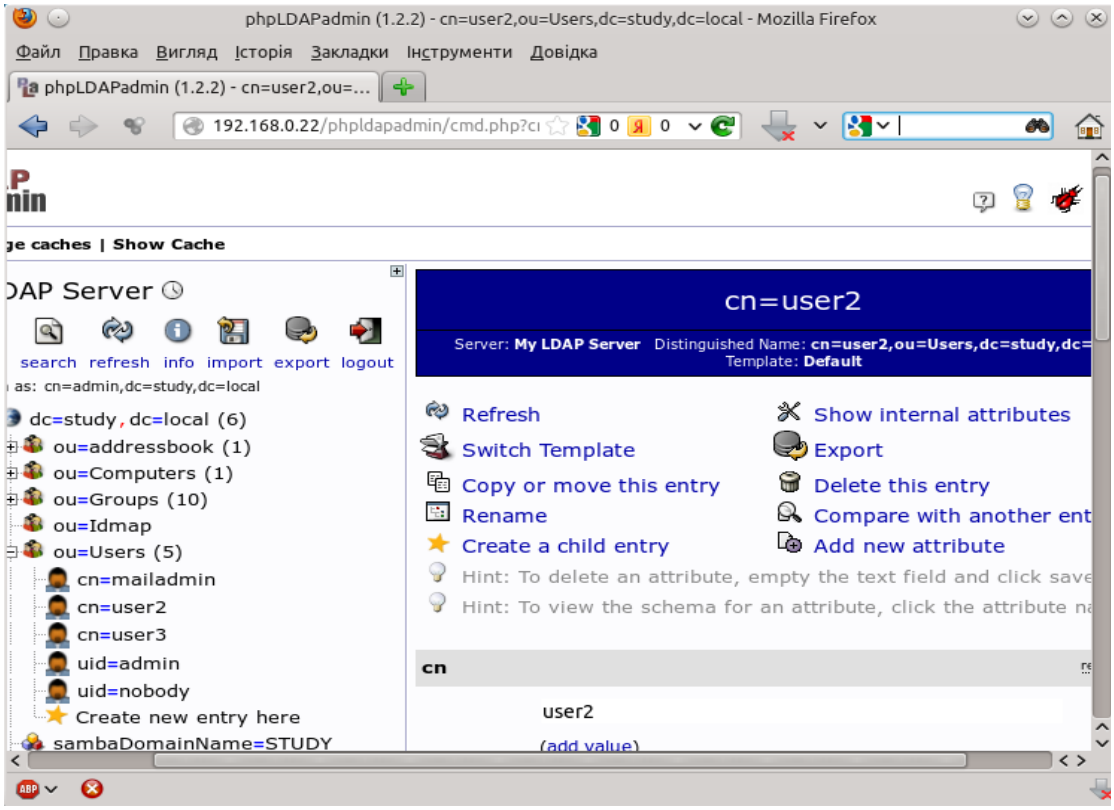
Для них ми призначили uid та gid 5000. Число було вибрано довільно, як досить велике.

І вкажемо власника і права на доступ до каталогу з поштою:
#chown virtual:virtual /var/spool/mail/study.local
#chmod 770 /var/spool/mail/study.local

Налаштування OpenLDAP

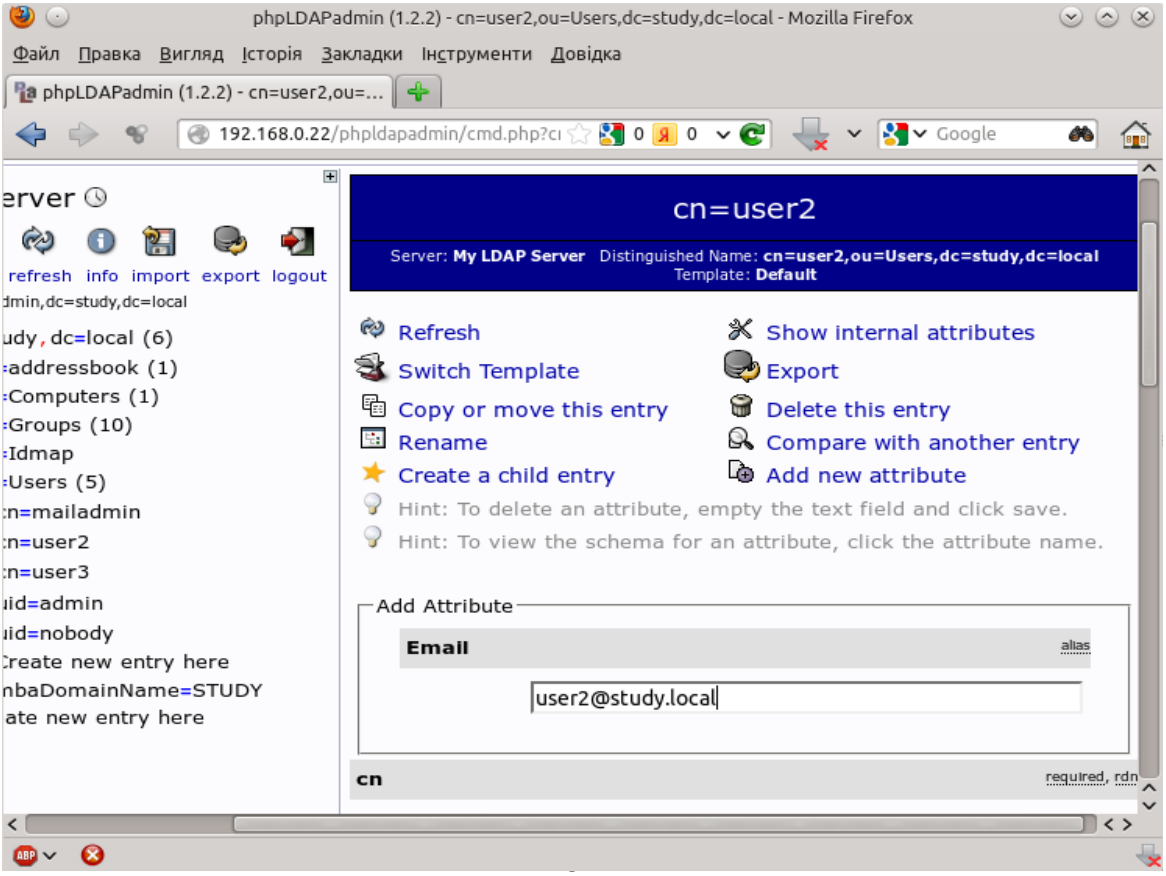
Відкриємо phpldapadmin і заведемо в LDAP нового користувача mailadmin з паролем mailadmin. Обліковий запис цього користувача буде використовуватися для доступу нашого поштового сервера до вмісту LDAP-каталогу.

Налаштуємо поштову адресу для раніше створених користувачів user2 і user3. Для цього відкриємо phpldapadmin і перейдемо на користувача.



Та оберемо “Add new attribute”.

У випадаючому списку виберемо "Email" і в полі, що з'явилося, пропишемо поштову адресу користувача — user2@study.local



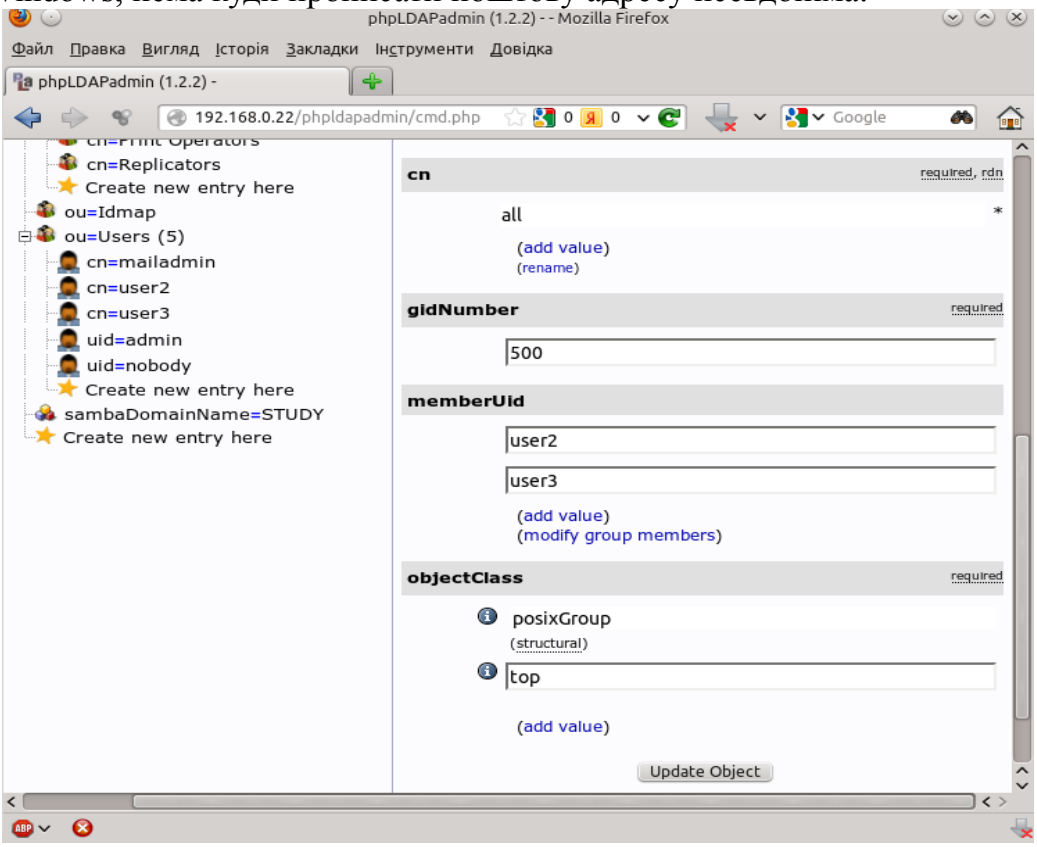
Після чого внизу сторінки натиснемо кнопку «Update object». Аналогічним чином пропишемо поштову адресу для користувача user3.

Створення поштових псевдонімів в OpenLDAP

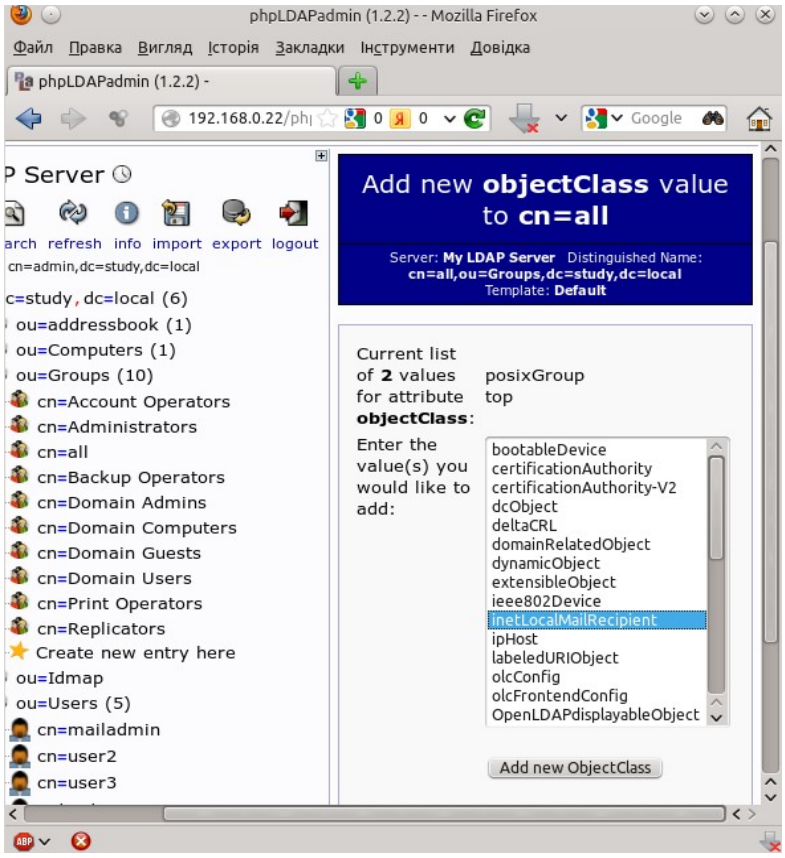
На жаль конфігурація за замовчуванням не підтримує таких речей, як групи розсилки або поштові псевдоніми. Звичайно в інтернеті можна знайти схеми, після додавання яких в OpenLDAP такий функціонал з'явиться. Але, насправді, в самому OpenLDAP для створення таких поштових адрес вже все є. Головне грамотно цим скористатися.

В консолі phpldapadmin перейдемо в гілку Groups і створимо там нову групу — all. Створювати будемо за шаблоном «Generic Posix Group».

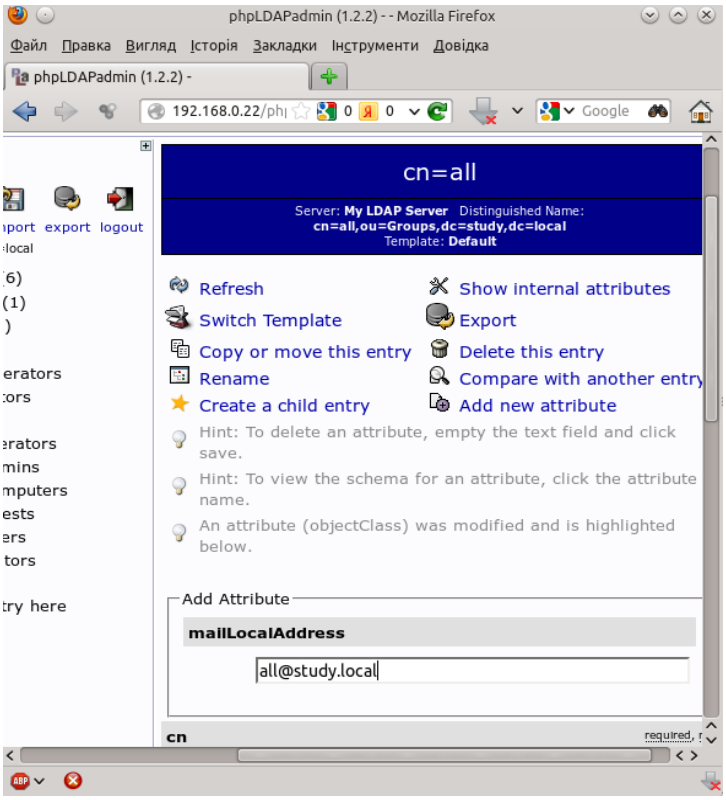
Відразу ж при створенні включимо туди користувачів user2 і user3. Згодом, звісно, туди можна буде включити і інших користувачів. Зверніть увагу, що після створення групи в ній, на відміну від аналогічної групи в домені Windows, нема куди прописати поштову адресу псевдоніма.



Для того, що б необхідне поле з'явилося, потрібно додати ще один objectClass — inetLocalMailRecipient



Після цього додаємо новий атрибут mailLocalAddress, куди і вводимо адресу псевдоніма — all@study.local.



Листи, що прийшли на цю адресу будуть перенаправлятися всім користувачам, що входять в групу all. В даном випадку це користувачі user2 і user3.

На цьому налаштування OpenLDAP для роботи з поштою закінчуємо.

Налаштування Postfix

Відкриваємо на редагування файл */etc/postfix/main.cf* і заповнимо його наступними даними:

 # Так наш сервер буде представлятися при відправці та отриманні пошти
smtpd_banner = $myhostname ESMTP (ubuntu)

biff = no

 # Забороняємо автоматично доповнювати неповне доменне ім'я в адресі листи
append_dot_mydomain = no

queue_directory = /var/spool/postfix # Вказуємо теку черги для Postfix

myhostname = pdc.study.local #Вказуємо ім'я нашого хоста

alias_maps =

myorigin = study.local

mydestination = localhost # Вказуємо, для яких доменів будемо приймати пошту

 # Вказуємо, для яких віртуальних доменів будемо приймати пошту
virtual_mailbox_domains = study.local

virtual_mailbox_base = /var/spool/mail/ # Початок шляху для зберігання пошти

virtual_alias_maps = ldap:/etc/postfix/ldapalias # Файл з описом поштових аліасів

virtual_mailbox_maps = ldap:/etc/postfix/ldap_virtual_mailbox_maps.cf #Файл с описанием почтовых ящиков

virtual_minimum_uid = 100

virtual_uid_maps = static:5000

virtual_gid_maps = static:5000

mynetworks = 127.0.0.0/8 # Вказуємо список довірених підмереж

recipient_delimiter = +

inet_interfaces = all # Приймаємо з'єднання на всіх інтерфейсах

 # Описуємо авторизацію через Dovecot
smtpd_sasl_auth_enable = yes

smtpd_sasl_type = dovecot

smtpd_sasl_path = private/auth

smtpd_sasl_security_options = noanonymous

broken_sasl_auth_clients = yes

smtpd_helo_required = yes # Обов'язково при з'єднанні вимагати helo

 # Далі налаштовуємо фільтри прийому та відправлення пошти
smtpd_recipient_restrictions = permit_mynetworks,
permit_sasl_authenticated,
check_helo_access hash:/etc/postfix/helo.list,
check_sender_access hash:/etc/postfix/ext_sender,
reject_unauth_destination,
reject_unknown_sender_domain,
reject_unknown_recipient_domain,
reject_non_fqdn_recipient,
reject_non_fqdn_sender,
reject_non_fqdn_hostname,
reject_invalid_hostname,
reject_unknown_hostname

Створимо файл */etc/postfix/helo.list*
#touch /etc/postfix/helo.list

Відкриємо його на редагування і внесемо в нього рядок:
study.local 550 Don't use my hostname

І виконаємо його хешування:
#postmap /etc/postfix/helo.list

Створимо файл */etc/postfix/ext_sender*

#touch /etc/postfix/ext_sender

Відкриємо його на редагування і внесемо в нього рядок:
study.local 550 Do not use my domain in your envelope sender

І виконаємо його хешування:
#postmap /etc/postfix/ext_sender

Тепер створимо файли конфігурації запитів до LDAP

ldap_virtual_mailbox_maps.cf

server_host = 192.168.56.10
bind = yes
bind_dn = cn=mailadmin,ou=Users,dc=study,dc=local
bind_pw = mailadmin
search_base = ou=Users,dc=study,dc=local
query_filter = (&(mail=%s))
result_attribute = mail
result_format = %d/%u/

ldapalias

server_host = 192.168.56.10
bind = yes
bind_dn = cn=mailadmin,ou=Users,dc=study,dc=local
bind_pw = mailadmin
search_base = ou=Groups,dc=study,dc=local
query_filter = (&(objectclass=posixGroup)(objectclass=inetLocalMailRecipient)(mailLocalAddress=%s))
leaf_result_attribute = mail
result_attribute = memberUid

Налаштування Dovecot

Розглянемо налаштування Dovecot версії 2.xx.

Тепер в теці */etc/dovecot* ми маємо багато файлів конфігурації. При чому навіть з підкаталогами.

Можна, звичайно, всю конфігурацію звести в один файл, але це буде суперечити тому, що задумали розробники.

Відкриємо основний файл конфігурації */etc/dovecot/dovecot.conf* і приведемо його до ось такого вигляду:

 # За яким протоколом працюємо
protocols = imap
 # Слухаємо з'єднання на всіх інтерфейсах по протоколу IPv4
*listen = **
 # Робоча тека
base_dir = /var/run/dovecot/
 # Ім'я інстансу (для відображення в лозі)
instance_name = dovecot
 # Рядок привітання

login_greeting = Dovecot ready.
 # Відключати клієнтські з'єднання при виключенні або перезавантаженні майстер-сервісу
shutdown_clients = yes
 # Сокет керуючого сервісу doveadm
doveadm_socket_path = doveadm-server
 # Підключаємо окремі файли конфігурації
!include conf.d/.conf*

Тепер переходимо до теки */etc/dovecot/conf.d*
Відкриємо в ній файл *10-auth.conf* і пропишемо в ньому два рядки:

disable_plaintext_auth = no
auth_mechanisms = plain login

Далі відредагуємо файл *10-mail.conf*
mail_location = maildir:/var/spool/mail/study.local/%n
mail_uid = 5000
mail_gid = 5000
mail_privileged_group = virtual
valid_chroot_dirs = /var/spool/mail/
!include auth-ldap.conf.ext

Далі нас буде цікавити файл *10-master.conf*

service imap-login {
inet_listener imap {
#port = 143
}
inet_listener imaps {
#port = 993
#ssl = yes
}
}
service auth {
 # Postfix smtp-auth
unix_listener /var/spool/postfix/private/auth {
mode = 0666
}
 # Auth process is run as this user.
user = postfix
group = postfix
}

В файлі *ssl.conf* треба прописати:
ssl = no

Подивившись у файл *auth-ldap.conf.ext* ми побачимо, що параметри з'єднання з сервером LDAP зберігаються у файлі */etc/dovecot/dovecot-ldap.conf.ext*

Відкриємо його:
 # nano /etc/dovecot/dovecot-ldap.conf.ext

І у кінець файлу додамо налаштування роботи з контролером домену

hosts = 192.168.56.10
auth_bind = yes
ldap_version = 2
base = ou=Users,dc=study,dc=local
dn = cn=mailadmin,ou=Users,dc=study,dc=local
dnpass = mailadmin
deref = never
scope = subtree

user_attrs = uidNumber=5000,gidNumber=5000
user_filter = (&(objectClass=posixAccount)(cn=%u))
pass_filter = (&(objectClass=posixAccount)(cn=%u))

Тепер перезапустимо обидва сервіси:
#service postfix restart
#service dovecot restart

На цьому налаштування поштового сервера завершено. Можна перевіряти його працездатність.

.

Налаштування роботи поштових сервісів в Microsoft AD Windows 2003

В якості поштових адрес будемо використовувати значення поля mail в обліковому записі користувача в AD.

В якості поштових псевдонімів будемо використовувати значення поля mail в групах AD. Листи будуть пересилатися на всіх користувачів, що входять в ці групи.

Нехай домен в Windows AD також називається study.local

Переналаштування нашого поштового сервера буде полягати виключно в зміні файлів Postfix і Dovecot, в яких описані параметри роботи з каталогом LDAP.

Приведемо їх до наступного вигляду:

/etc/postfix

/etc/postfix/ldap_virtual_mailbox_maps.cf

server_host = 192.168.0.10
bind = yes
bind_dn = cn=mailadmin,cn=Users,dc=study,dc=local
bind_pw = mailadmin
search_base = cn=Users,dc=study,dc=local
query_filter = (&(mail=%s))
result_attribute = mail
result_format = %d/%u/

/etc/postfix/ldapalias
server_host = 192.168.0.10
bind = yes
bind_dn = cn=mailadmin,cn=Users,dc=study,dc=local
bind_pw = mailadmin
search_base = cn=Users,dc=study,dc=local
query_filter = (&(objectClass=group)(mail=%s))
leaf_result_attribute = mail
special_result_attribute = member

/etc/dovecot/dovecot-ldap.conf.ext
hosts = 192.168.0.10
auth_bind = yes
ldap_version = 3
base = cn=Users,dc=study,dc=local
dn = cn=mailadmin,cn=Users,dc=study,dc=local
dnpass = mailadmin
deref = never
scope = subtree
user_filter = (&(ObjectClass=person)(sAMAccountName=%u))
pass_filter = (&(ObjectClass=person)(sAMAccountName=%u))

Після зміни файлів конфігурації звісно потрібно перезапустити обидва сервіси.

Спільна адресна книга в OpenLDAP

Чим більше організація, тим більше поштових адрес використовується в процесі роботи співробітників. І, звісно, виникає задача використання спільної адресної книги. Якщо, наприклад, в Exchange така функція вбудована, то при використанні поштового Linux-сервера цю задачу потрібно вирішувати окремо.

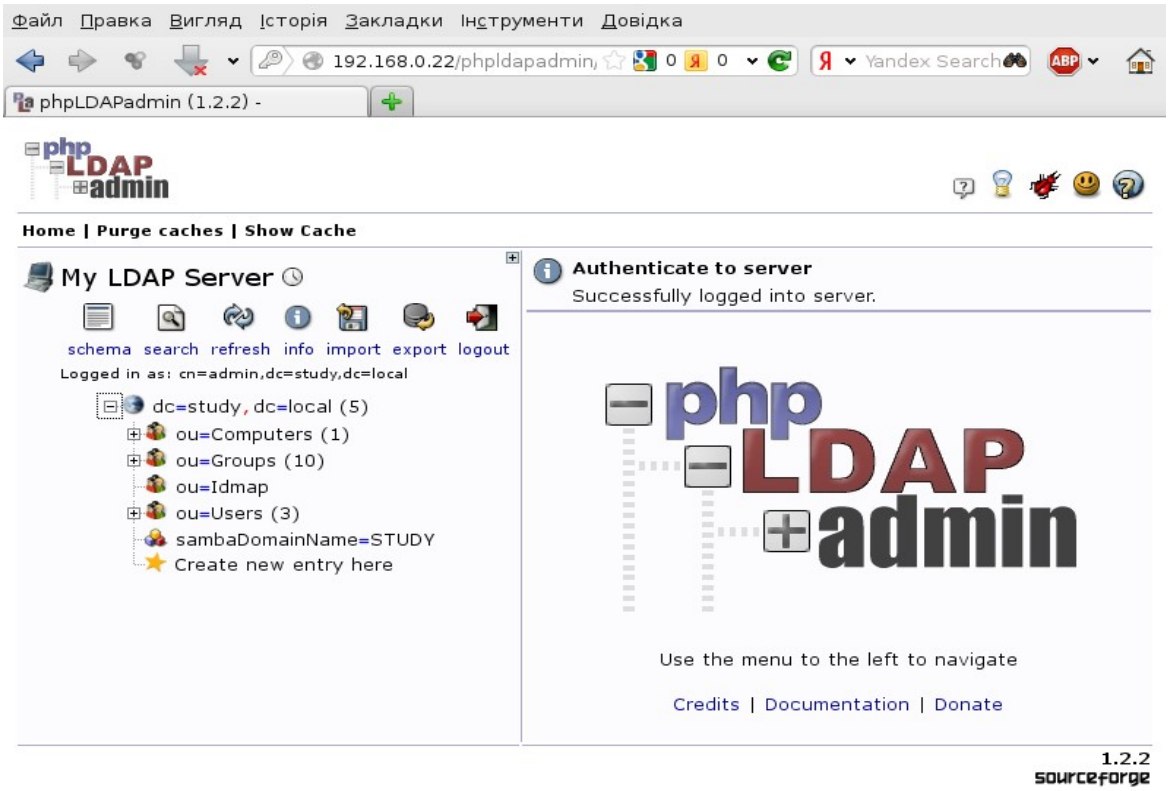
Звичайно, можна розглянути варіант використання локальних адресних книг і постійну їх синхронізацію, але це дуже непродуктивно.

Для того, що б і клієнтам користуватися було зручніше, і адміністратору було легше керувати системою, розглянемо створення централізованої книги контактів на базі OpenLDAP.

У зв'язку з тим, що у нас вже піднятий веб-інтерфейс phpLDAPAdmin, то налаштування будемо проводити з його допомогою.

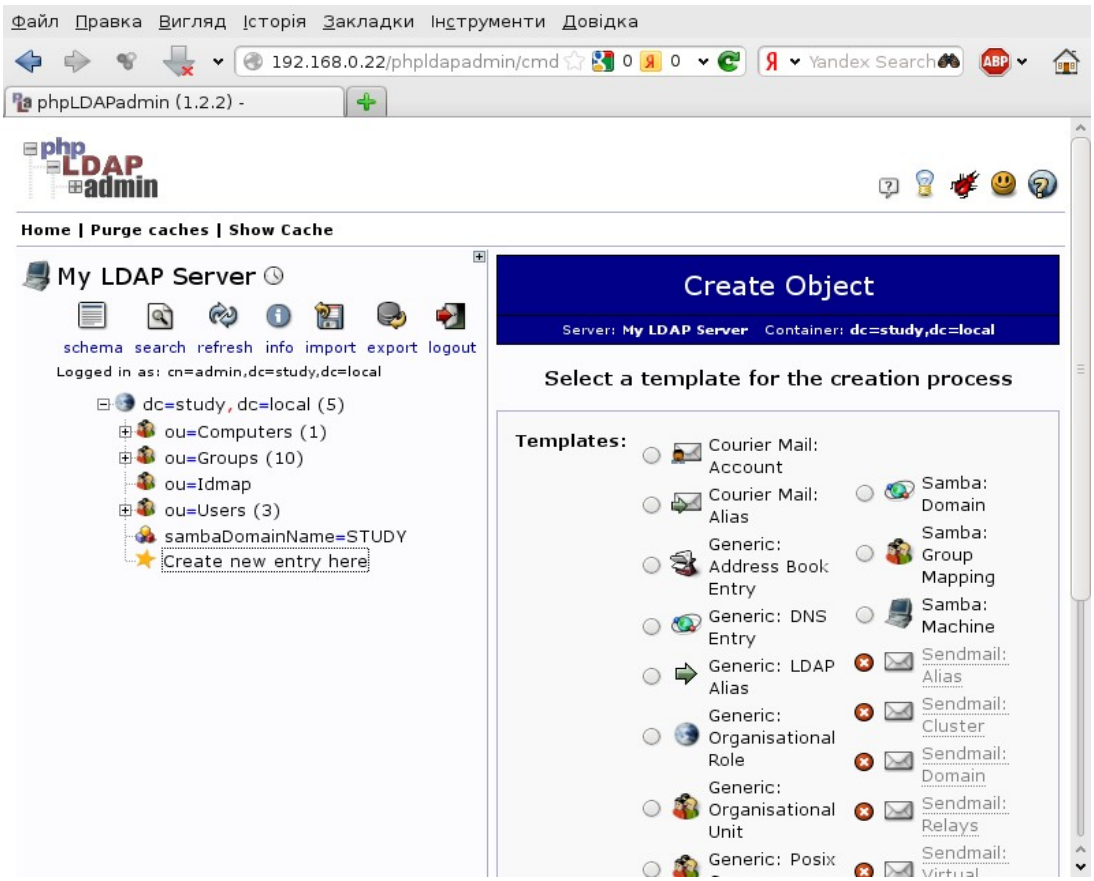
Створення адресної книги

Відкриємо інтерфейс за посиланням http://server-ip/phpldapadmin

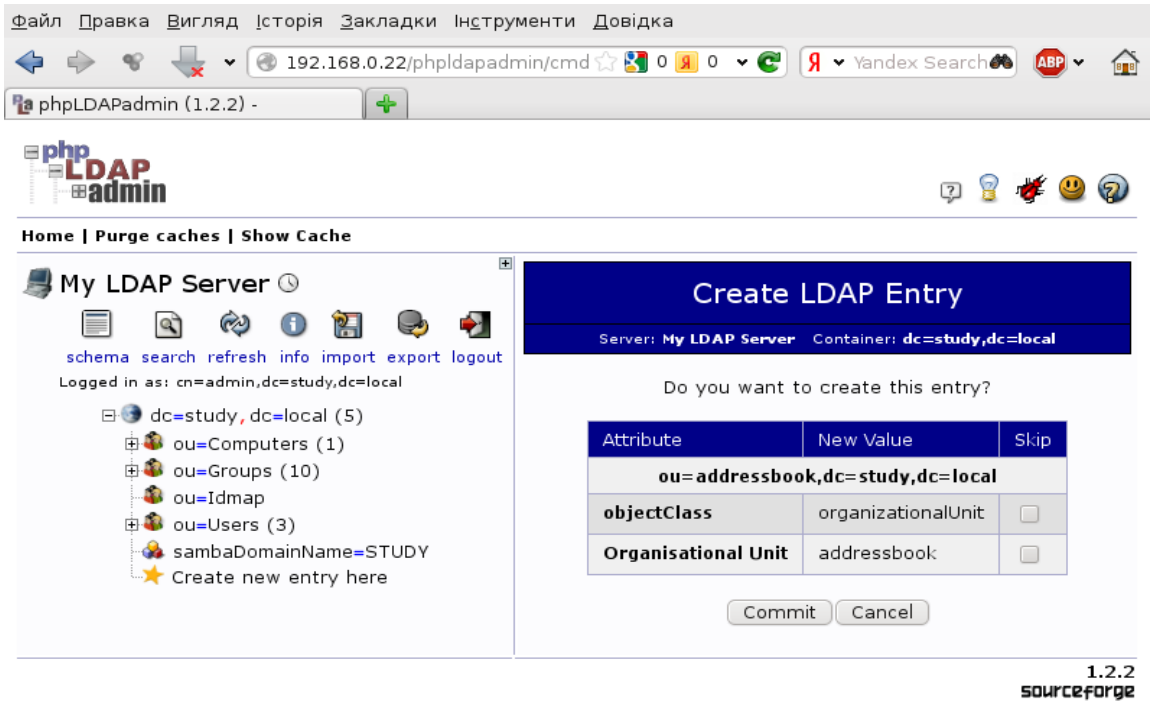


Тепер в дереві нашого LDAP-сервера створимо окрему гілку для зберігання записів адресної книги. Для цього виберемо пункт "Create new entry here" і в правій

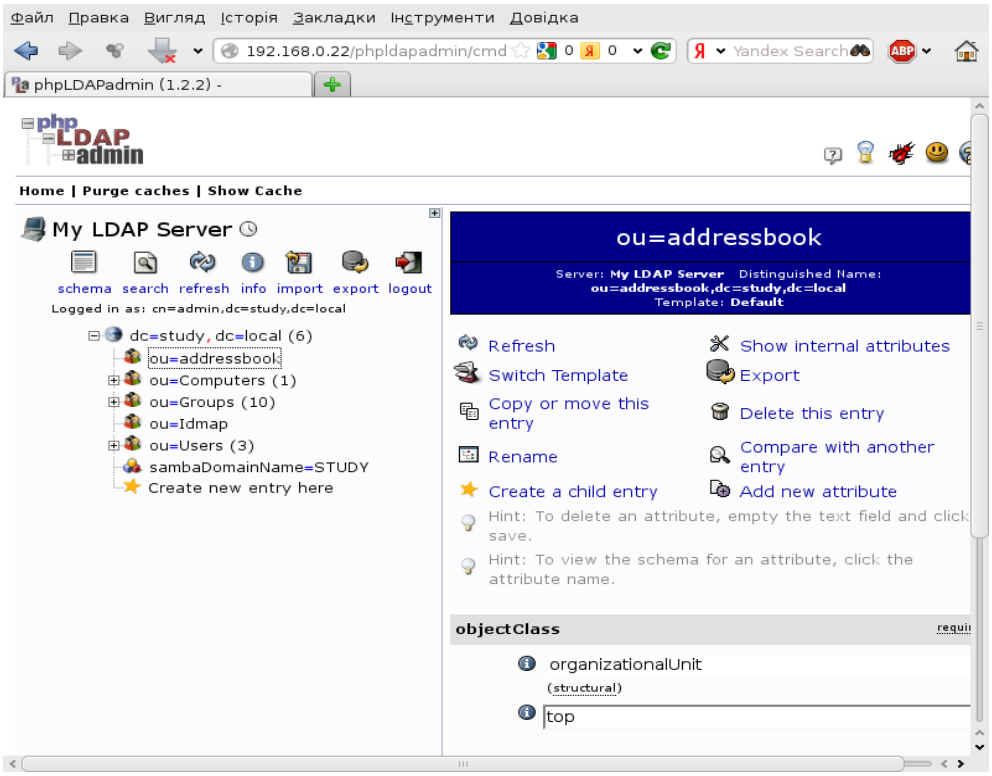
частині вікна "Generic: Organisation Unit".



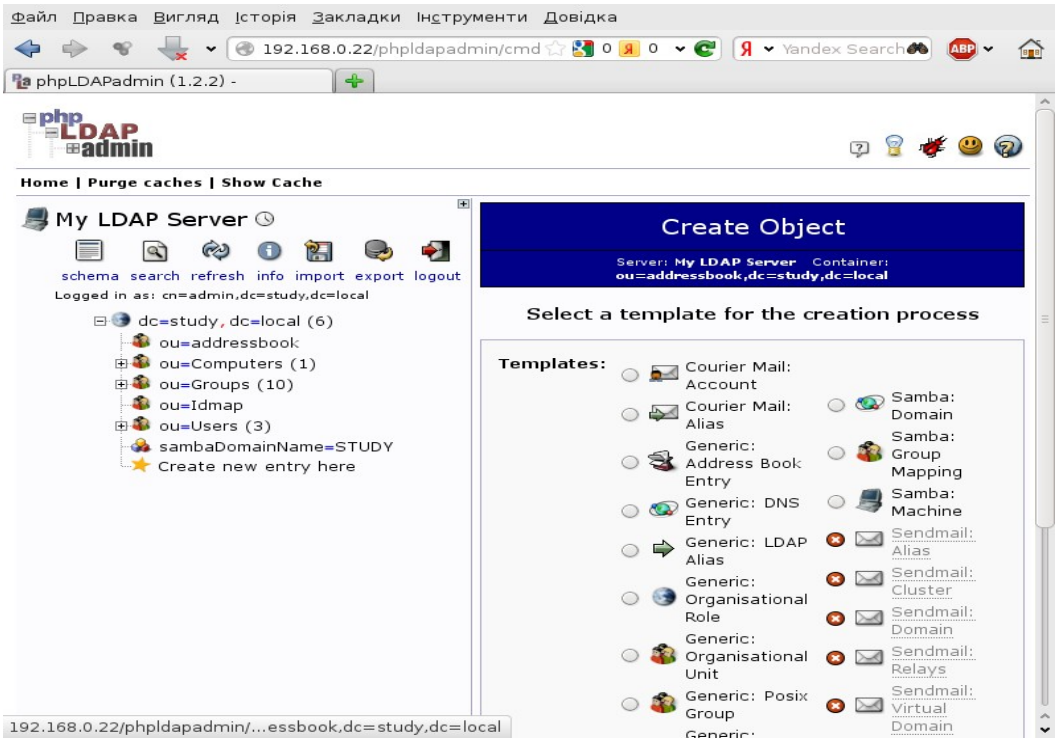
Дамо назву новій гілці — addressbook.



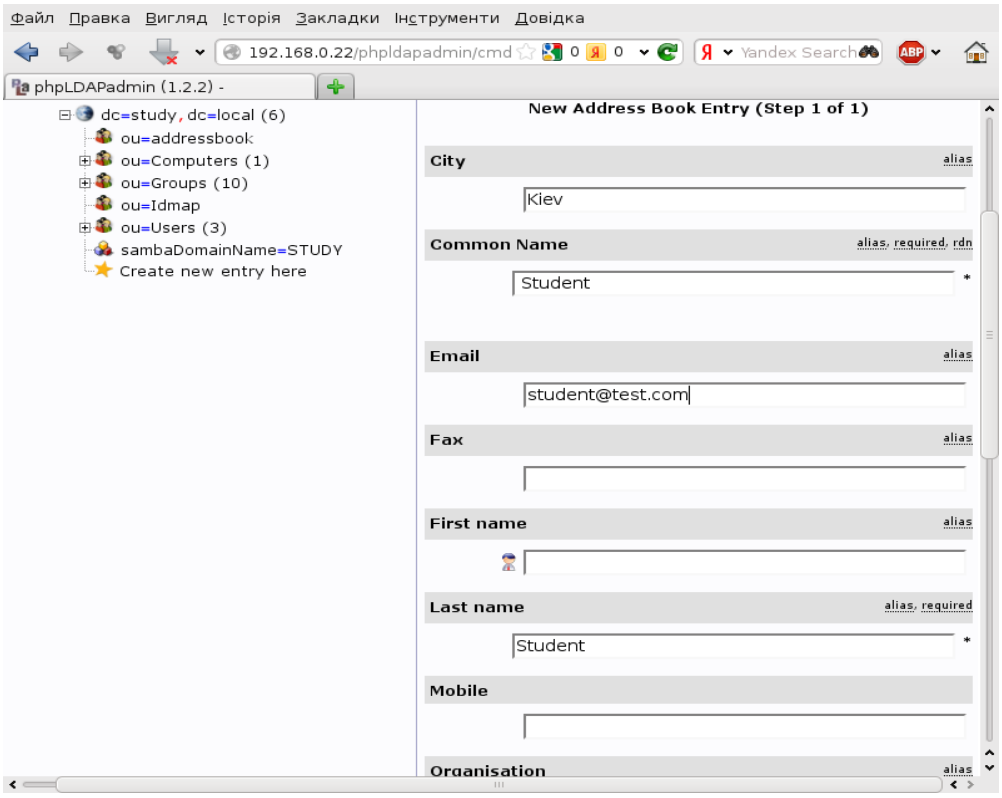
Тепер перейдемо до новоствореної гілки і створимо там наш перший запис в адресній книзі. Для цього виберемо "Create Child Entry".



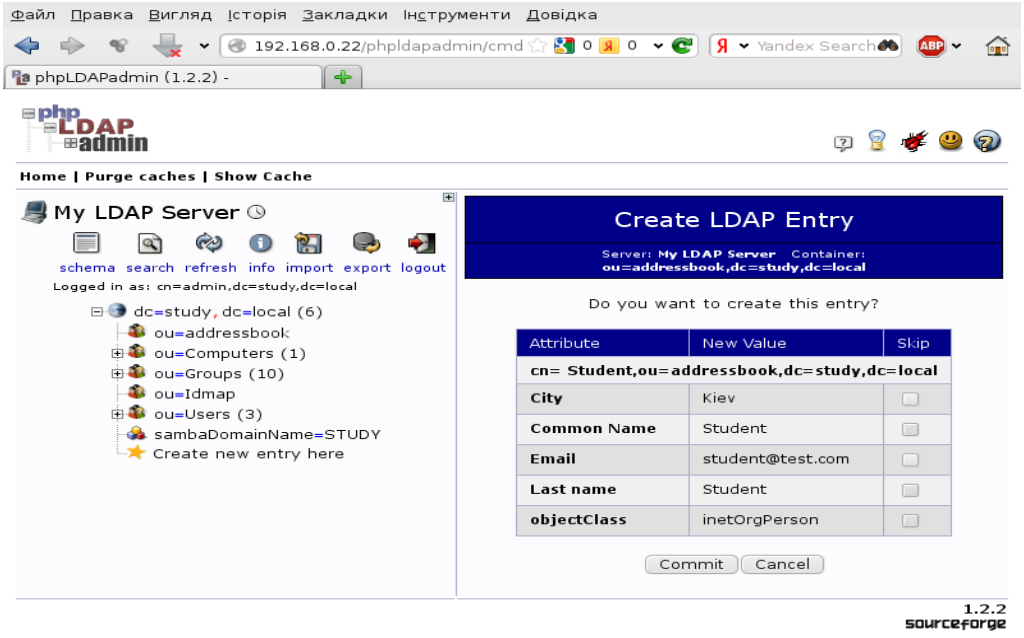
І створимо там об'єкт типу "Generic: Address Book Entry".



Заповнимо всі поля даних потрібною інформацією. Зверніть увагу, що всі обов'язкові поля виділені жовтим кольором.



І застосуємо зміни, натиснувши на кнопку "Commit"

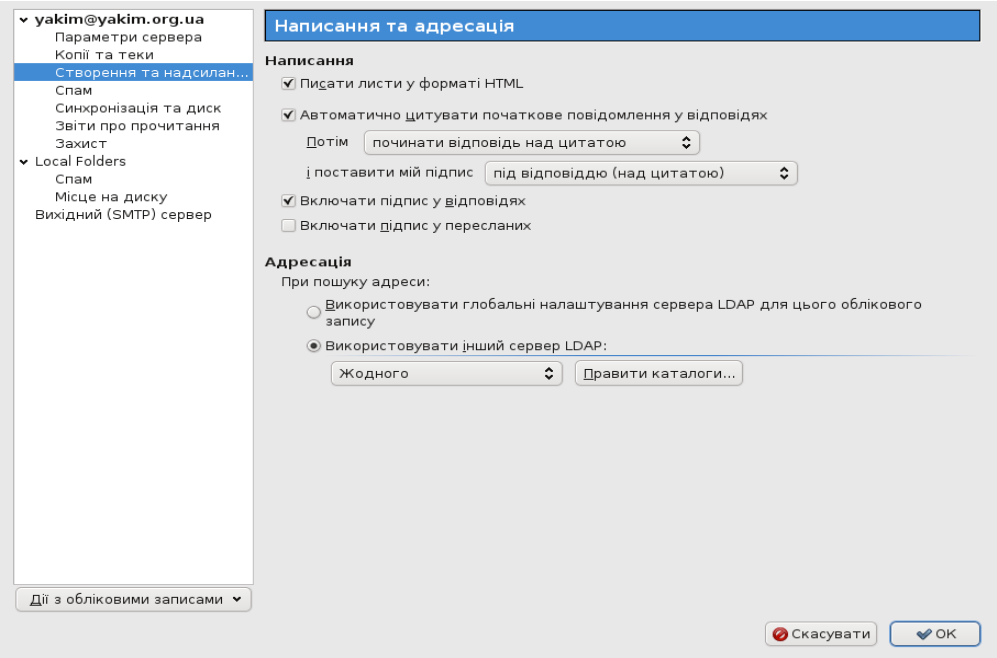


На цьому налаштування адресної книги завершено.

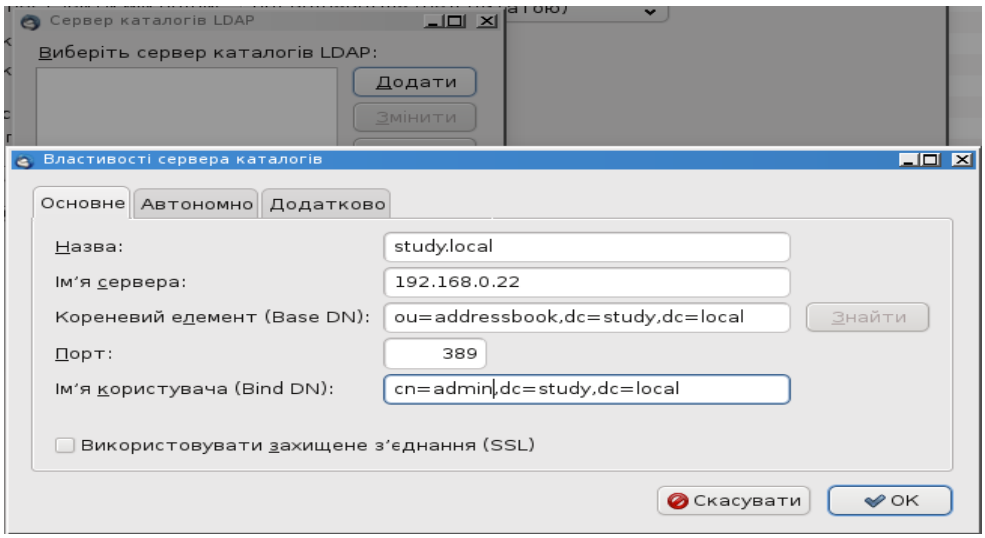
Підключення адресної книги до поштового клієнта.

Сама по собі адресна книга особливої цінності не представляє. Вона корисна тільки тоді, коли їй користуються наші співробітники. Тому розглянемо підключення нашої свіжостворених адресної книги до поштового клієнту на прикладі Thunderbird.

Для цього запустимо поштовий клієнт і у властивостях нашого облікового запису перейдемо на пункт «Створення та надсилання». У правій частині вікна виберемо «Використовувати інший сервер LDAP»

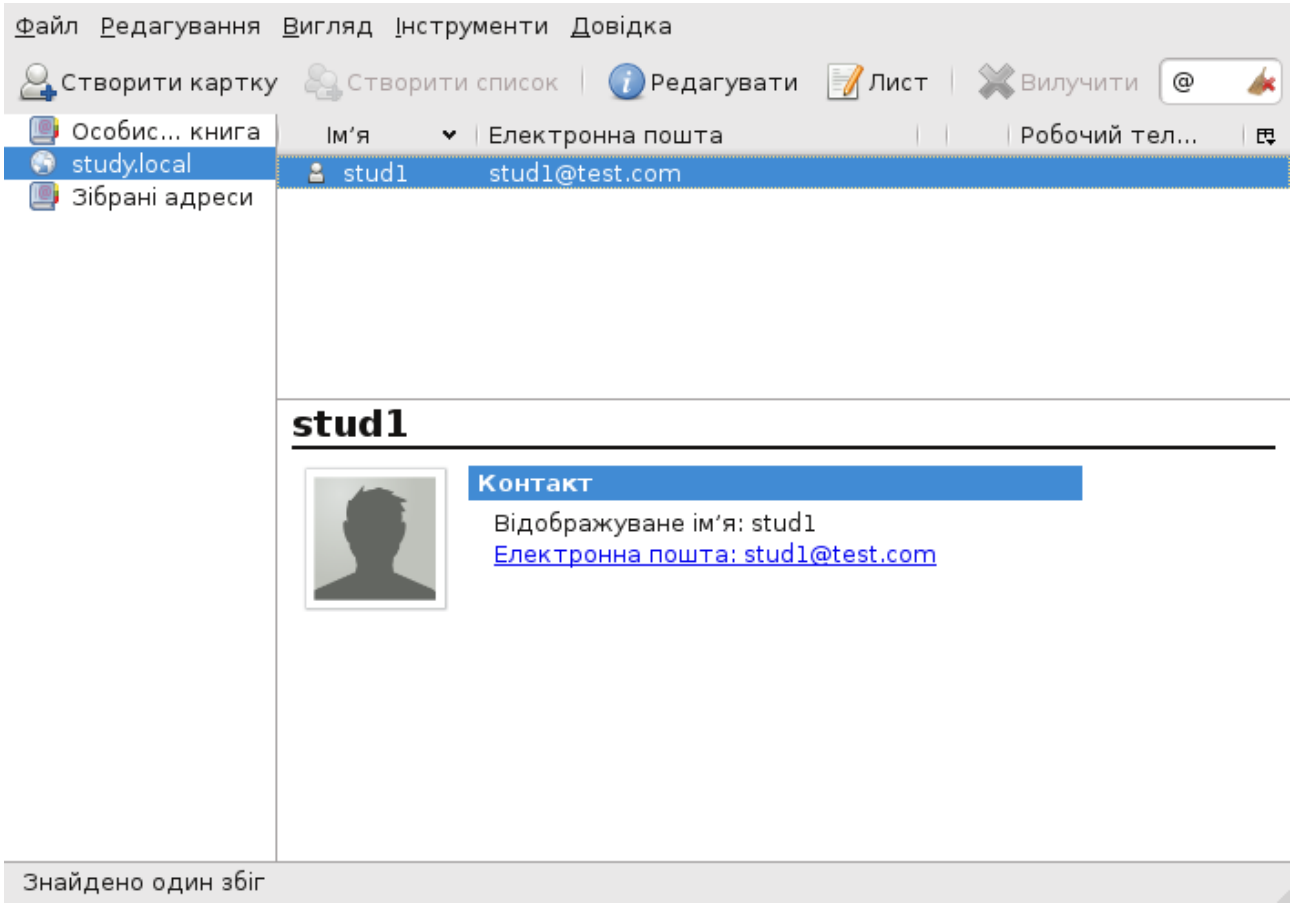


Далі тиснемо кнопку «Змінити сервер каталогів», «Додати», і вводимо параметри нашого OpenLDAP-сервера і налаштування доступу до адресної книги.



Після цього можна перевіряти працездатність спільної адресної книги.

Відкриємо в Thunderbird адресну книгу, виберемо ту, яка зберігається в LDAP (при наших налаштуваннях вона має називатися study.local) і в рядку пошуку введемо «@».



На жаль, поштові клієнти не підтримують загальний запит всього вмісту, тому потрібно що-небудь ввести в рядку пошуку. Так, як символ «@» міститься в будь-якій поштовій адресі, то це і буде запитом на показ всього вмісту адресної книги.

Поштовий веб-інтерфейс RoundCube

RoundCube Webmail — це клієнт для роботи з електронною поштою з веб-інтерфейсом, написаний на PHP з використанням CSS і XHTML і технології AJAX. RoundCube Webmail встановлюється практично на будь-який сервер з підтримкою PHP і MySQL і надає можливість роботи з поштовими скриньками по протоколах IMAP і SMTP.

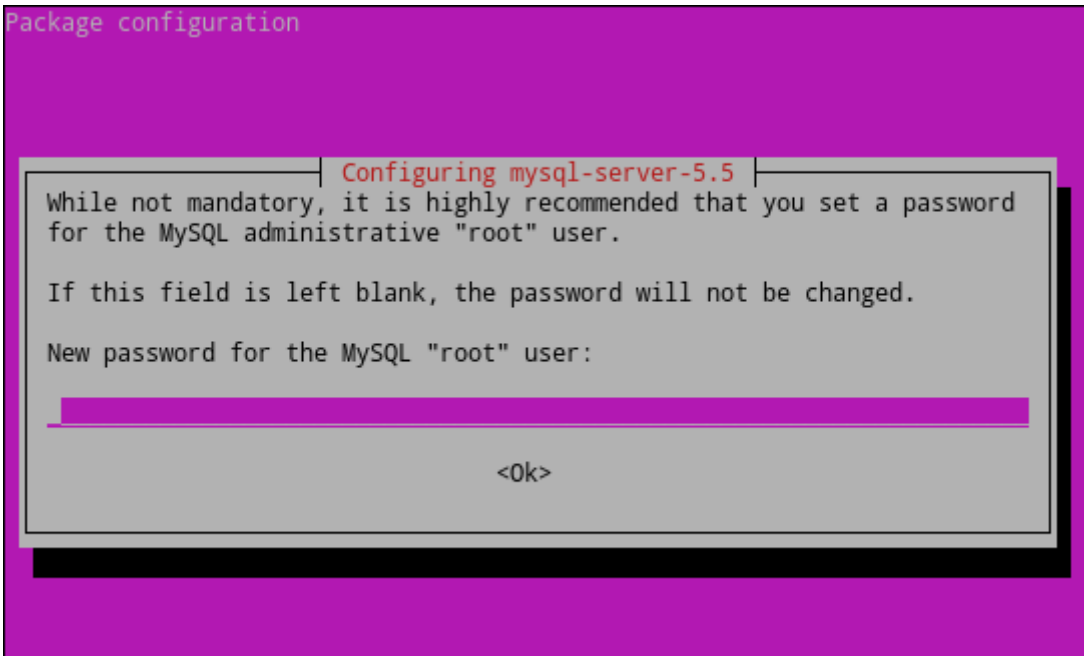
Проект був заснований 18 травня 2005 року. Тоді RoundCube Webmail являв собою скромний клієнт для роботи з електронною поштою. Зараз це потужна поштова програма, яка майже нічим не поступається звичайним поштовим клієнтам.

RoundCube Webmail випускається під ліцензією GPL і є вільним програмним забезпеченням.

Встановлення RoundCube

Для своєї роботи, а точніше для зберігання даних, кешувати пошти та внутрішньої адресної книги RoundCube вимагає наявності MySQL-сервера. Так як у нас його в мережі поки що немає, то встановимо його локально:

 # apt-get install mysql-server mysql-client

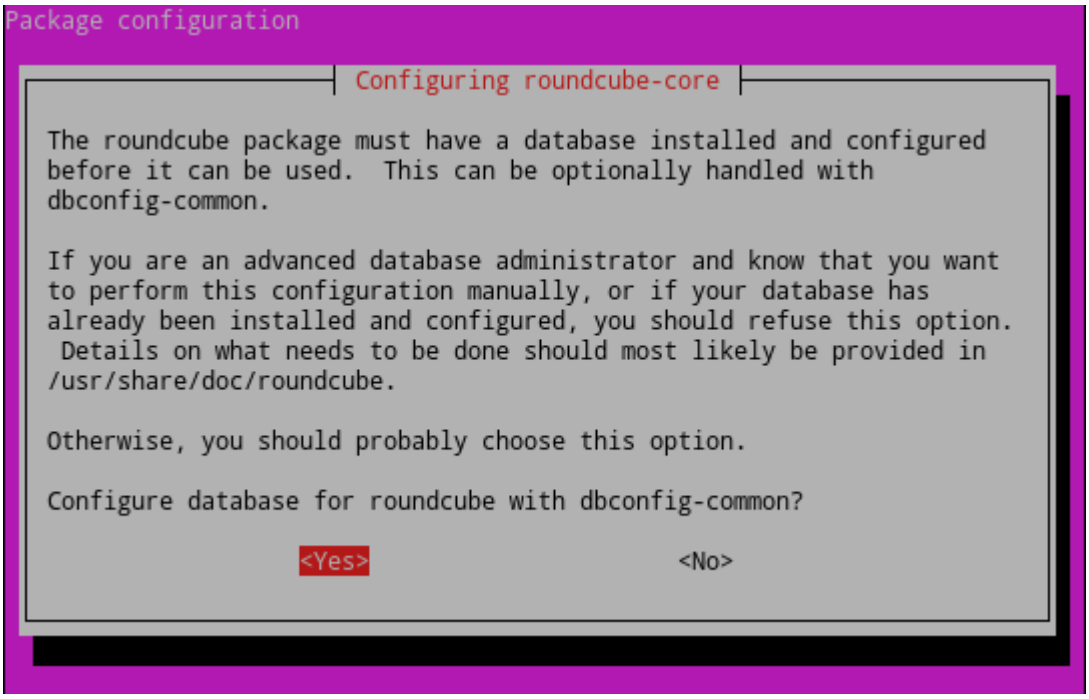


У процесі встановлення введемо пароль для користувача root нашого MySQL-сервера.

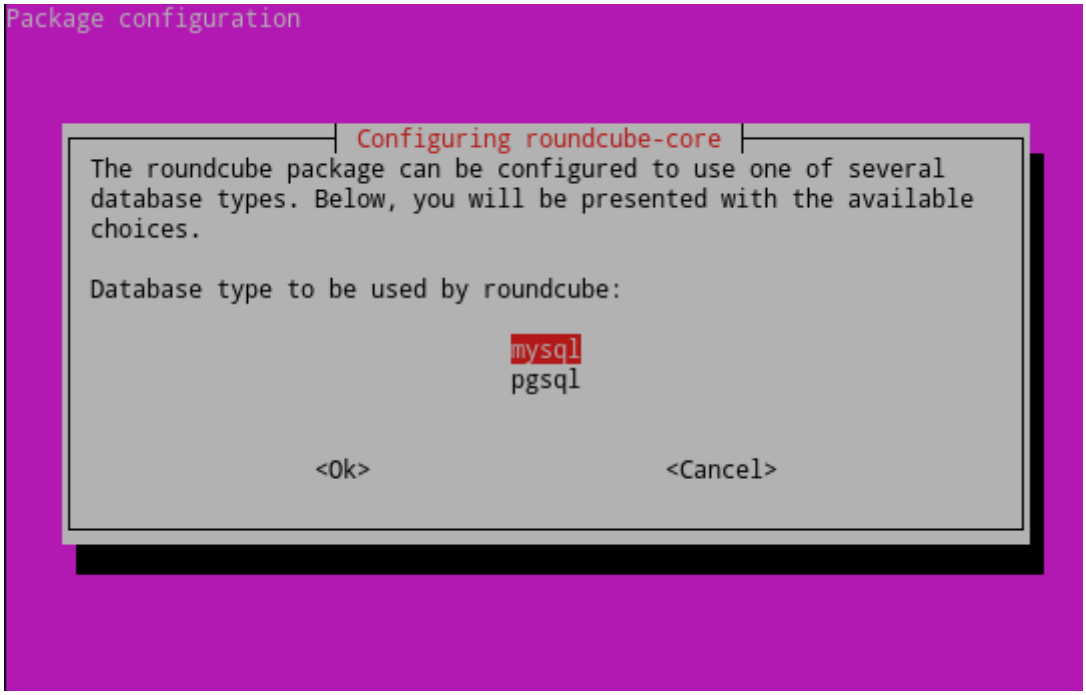
Тепер можна встановлювати безпосередньо і сам RoundCube:

 # apt-get install roundcube

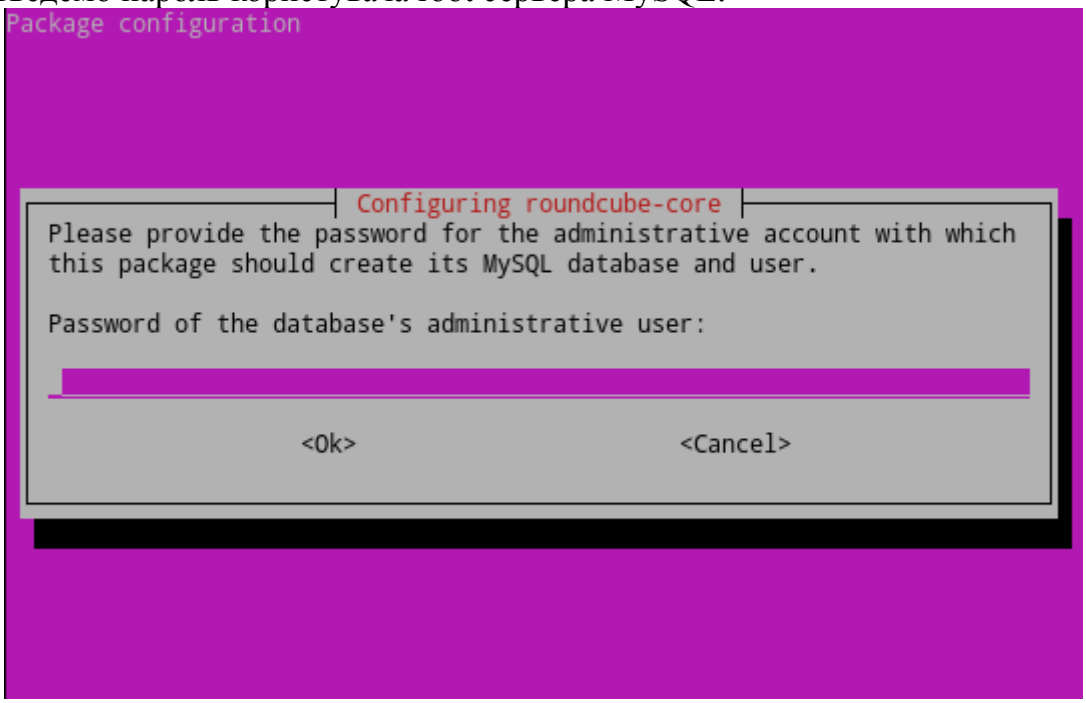
У процесі встановлення нам буде запропоновано автоматично налаштувати базу даних для RoundCube за допомогою dbconfig-common:



Оберемо тип бази даних — mysql:



І введемо пароль користувача root сервера MySQL.



Потім нам запропонують ввести пароль для користувача roundcube. Його можна залишити порожнім. У цьому випадку пароль буде згенерований випадковим чином.

На цьому встановлення завершено.

Зараз необхідно налаштувати RoundCube для коректної роботи з нашим поштовим сервером.

Налаштування RoundCube

Відкриємо файл конфігурації RoundCube — */etc/roundcube/main.inc.php*
У ньому необхідно змінити кілька рядків:

В розділі imap
$rcmail_config['default_host'] = array("127.0.0.1");

В розділі smtp
$rcmail_config['smtp_server'] = '127.0.0.1';

В розділі system
$rcmail_config['mail_domain'] = 'study.local';

Налаштування сервера Apache

Налаштуємо наш веб-сервер, для того, що б поштовий інтерфейс працював по захищеному протоколу https.

Для цього підключимо до Apache необхідні розширення

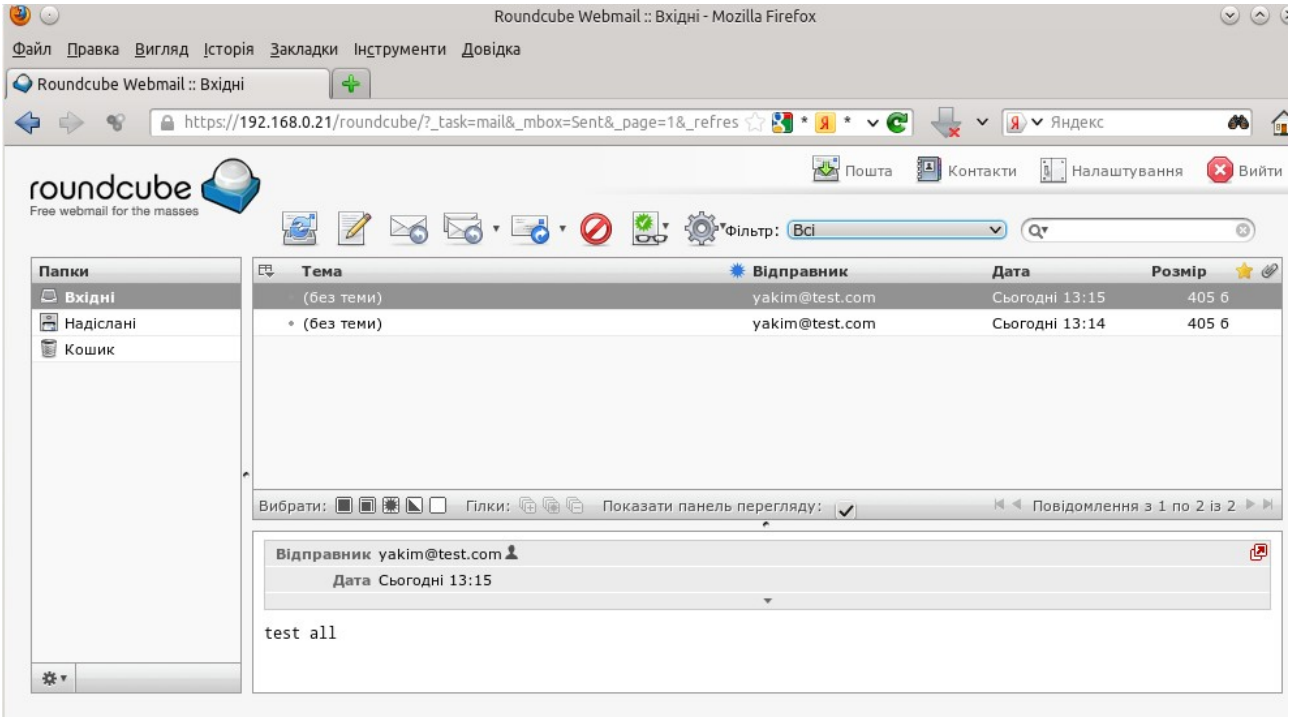
 # ln -s /etc/apache2/mods-available/ssl.conf /etc/apache2/mods-enabled/ssl.conf
 # ln -s /etc/apache2/mods-available/ssl.load /etc/apache2/mods-enabled/ssl.load

Тепер підключимо https-сайт за замовчанням
 # ln -s /etc/apache2/sites-available/default-ssl /etc/apache2/sites-enabled/default-ssl

Та ще в файлі */etc/apache2/conf.d/roundcube* розкоментуємо рядок
Alias /roundcube /var/lib/roundcube

Після цих дій необхідно перезапустити веб-сервер
#service apache2 restart

Тепер веб-інтерфейс до нашого поштового сервера доступний за посиланням *https://ip-addr/roundcube*



Налаштування проксі-сервера Squid

Squid — програмний пакет, що реалізує функцію кешуючого проксі-сервера для протоколів HTTP, FTP і HTTPS. Розроблений співтовариством як програма з відкритим вихідним кодом (поширюється відповідно до ліцензії GNU GPL).

Використовується в UNIX системах і в операційних системах сімейства Windows NT. Має можливість взаємодії з Active Directory шляхом аутентифікації через LDAP, що дозволяє використовувати розмежування доступу до інтернет ресурсів користувачів, що мають облікові записи на Windows Server, також дозволяє організувати «нарізку» інтернет трафіку для різних користувачів.

Налаштуванняа інтеграції з OpenLDAP

Після встановлення потрібно налаштувати Squid на те, що б він брав облікові записи користувачів з нашого домену.

Для авторизації користувача в LDAP, у Squid є спеціальний модуль *squid_ldap_auth*

Для початку перевіримо, чи працює взагалі у нас авторизація в домені.

Виконаємо таку команду, і в наступному рядку введемо логін і пароль доменного користувача через пробіл:

#/usr/lib/squid3/squid_ldap_auth -u uid -b "ou=Users,dc=study,dc=local" 192.168.56.10

mailadmin mailadmin

OK

Якщо в результаті вивелося OK — значить все нормально, і можна продовжувати перевірку.

Далі ми перевіримо, чи працює визначення групи користувача в домені.

Виконаємо таку команду, і в наступному рядку введемо ім'я користувача, і групу, належність до якої хочемо перевірити.

 # /usr/lib/squid3/squid_ldap_group -R -b "dc=study,dc=local" -D uid=mailadmin,ou=Users,dc=study,dc=local -w "mailadmin" -f "(&(memberUid=%v)(cn=%a))" -h 192.168.56.10

user2 all

OK

І, для перевірки, повторимо цю ж команду, але для іншого облікового запису:

 # /usr/lib/squid3/squid_ldap_group -R -b "dc=study,dc=local" -D uid=mailadmin,ou=Users,dc=study,dc=local -w "mailadmin" -f "(&(memberUid=%v)(cn=%a))" -h 192.168.56.10

admin all
ERR

Як ми бачимо, група користувача визначається правильно.

Зверніть увагу на рядок фільтра *-f "(&(memberUid=%v)(cn=%a))"*. У нашому випадку **cn** містить ім'я групи, а **memberUid** містить члена групи.

Як ми бачимо, група користувача визначається правильно.
Тепер відкриємо на редагування файл */etc/squid3/squid.conf*
Та опишемо наші нові acl:

 # Параметри авторизації в домені
auth_param basic program /usr/lib/squid3/squid_ldap_auth -u uid -b "ou=Users,dc=study,dc=local" 192.168.56.10
auth_param basic children 5
auth_param basic realm study.local
auth_param basic credentialsttl 2 hours

Далі пропишемо перевірку на наявність користувача в групі:
external_acl_type ldapgr children=5 %LOGIN /usr/lib/squid3/squid_ldap_group -R -b "dc=study,dc=local" -D uid=mailadmin,ou=Users,dc=study,dc=local -w "mailadmin" -f "(&(memberUid=%v)(cn=%a))" -h 192.168.56.10

Зажадаємо авторизацію:
acl auth proxy_auth REQUIRED

Та перевіримо, чи є користувач в групі all:
acl myinet external ldapgr all

І далі, в розділі дозволів, пропишемо дозвіл виходу в інтернет авторизованим користувачам, що входять в доменну групу all.
http_access allow myinet

Перезапустимо Squid:
#/etc/init.d/squid3 restart

Та можна перевіряти.
Тепер в інтернет зможуть вийти тільки користувачі, що входять в доменну групу myinet.

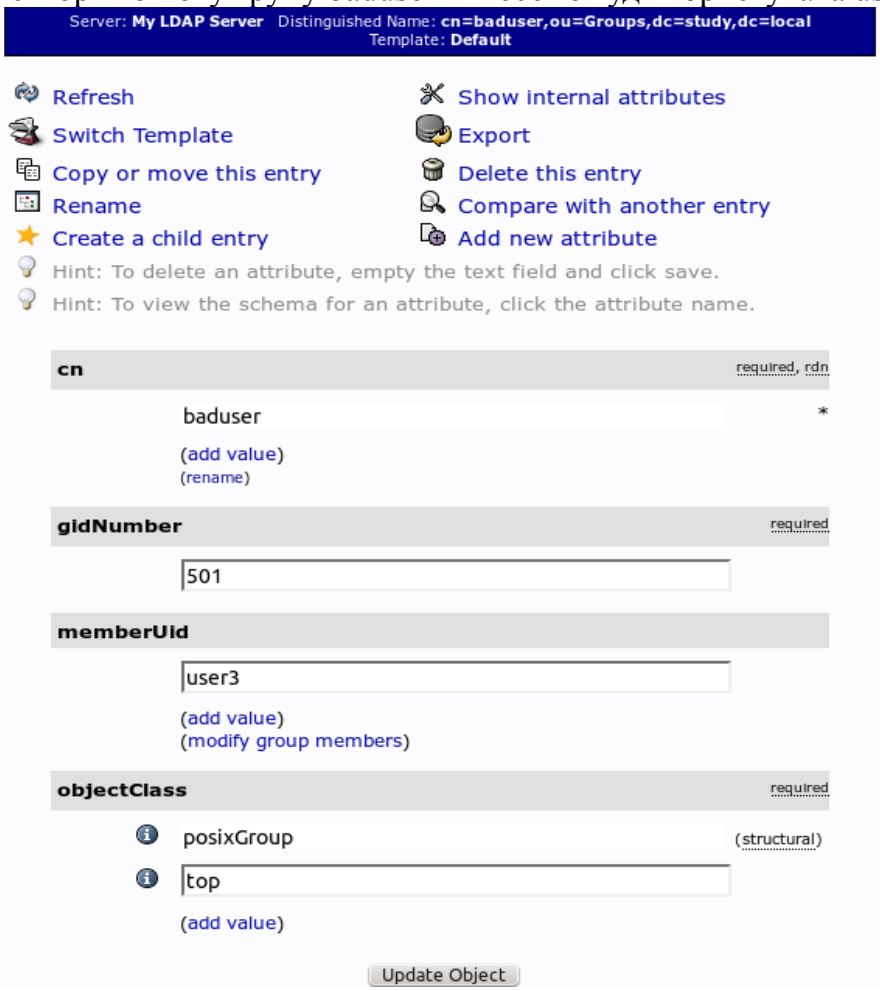
Налаштування ширини каналу для користувачів

Для обмеження ширини використовуваного користувачами інтернет-каналу в проксі-сервері Squid використовуються delay_pool'и. Вони бувають трьох класів:

1. 1. Індивідуальні — задається ширина каналу для користувача в байтах в секунду delay_parameters 1 32000/32000
2. Delay pools другого класу. Призначені для невеликих мереж. У параметрах тепер задається шейп на всю мережу перша пара і шейп на робочу станцію, це друга пара цифр delay_parameters 1 8000/8000 4000/4000
3. Delay pools третього класу призначені для великих мереж, що включають підмережі. Тепер параметри пулу описуються трьома парами цифр. Перша пара описує загальний, для всіх підмереж шейп, друга пара — шейп виділений для підмережі, а третя пара — для хоста delay_parameters 1 32000/32000 16000/16000 4000/4000

Створимо обмеження по швидкості для одного користувача, тобто створимо delay pool 1-го класу.

У домені створимо нову групу baduser і внесемо туди користувача user3



У цю групу можна додавати, природно різних доменних користувачів.

Опишемо визначення acl buser зі співставленням його з доменної групою baduser:
acl buser external ldapgr baduser

Тепер опишемо *delay_pool* для *acl buser* і встановимо обмеження пропускної здатності в 2 кб/с:

delay_pools 1
delay_class 1 1
delay_access 1 allow buser
delay_access 1 deny all
delay_parameters 1 2000/2000
http_access allow buser

Фільтрування завантажень файлів за розширенням

Проксі-сервер Squid дозволяє забороняти завантаження різних типів файлів, виходячи з їх розширення. Це дозволяє знизити завантаження каналу, заборонивши різний мультимедіа-контент.

Для того, що б зробити відповідні налаштування, в файл конфігурації в блок опису acl впишемо рядок:
acl iso urlpath_regex -i \.iso$

Тепер перед рядком *http_access allow myinet* додамо:
http_access deny myinet iso

При таких налаштуваннях у нас вийде, що користувачі, що входять в acl buser (відповідно в доменну групу baduser) можуть качати все, але повільно, а всі інші користувачі мають швидкий доступ в інтернет, але не можуть завантажувати iso-файли.

Налаштування блокування сайтів

Крім всього іншого, Squid дозволяє обмежувати доступ користувачів виходячи з імені сайту. Звичайно зробити з цього проксі-сервера повноцінну систему контентної фільтрації не вийде. Для цього існують спеціалізовані продукти, наприклад DansGuardian. Але ми все одно розглянемо, як здійснюється фільтрація такого типу.

В теці */etc/squid3* створимо файл block і заповнимо його наступними даними:
korrespondent
http://yakim.org.ua

Далі опишемо новий список контролю доступу (acl):
acl blocked url_regex "/etc/squid3/block"

і вставимо нове правило доступу перед визначенням *delay_pool*
http_access deny blocked

Далі, звісно, необхідно перечитати конфігурацію Squid

#service squid reload

Після цього всім користувачам буде заблокував доступ до будь-якого сайту, в адресному рядку якого є слово *korrespondent* і до сайту *http://yakim.org.ua*, але при цьому сайт *https://yakim.org.ua* буде нормально відкриватися.

Інтеграція Squid з Microsoft AD

Інтеграція нашого проксі-сервера з контролером домену на Windows робиться практично так само, як і з сервером OpenLDAP.

Рядок для перевірки логіна-пароля користувача буде виглядати наступним чином:
auth_param basic program /usr/lib/squid3/squid_ldap_auth -u cn -b "cn=Users,dc=study,dc=local" 192.168.0.10

А ось перевірка наявності користувача в групі вже зміниться:
external_acl_type ldapgr children=5 %LOGIN /usr/lib/squid3/squid_ldap_group -R -b "dc=study,dc=local" -f "(&(sAMAccountName=%v)(memberOf=CN=%a,CN=Users,dc=study,dc=local))" -D mailadmin@study.local -w "mailadmin" 192.168.0.10

Зверніть увагу, по-перше ім'я користувача береться вже з поля *sAMAcountName*, а по друге фільтр перевірки робиться по полю *MemberOf*.

Всі інші налаштування не змінюються взагалі. Вони не залежать від типу LDAP-сервера, що використовується.

Jabber-сервер з інтеграцією в домені

Jabber — це відкритий протокол, який використовує XML, для швидкого обміну повідомленнями та інформацією про присутність між будь-якими двома абонентами в Інтернет. Першим застосуванням технології Jabber стала поява асинхронної і розширюваної платформи для обміну миттєвими повідомленнями та мережі обміну миттєвими повідомленнями (від англ. IM — Instant Messaging), схожою за можливостями з комерційними системами IM, такими, як AIM, ICQ, MSN і Yahoo. Однак, Jabber має ряд переваг у порівнянні з комерційними системами IM:

- Відкритість — протокол Jabber є вільним (від ліцензування), відкритим, загальнодоступним і, крім того, легкий для розуміння; існує безліч реалізацій серверів і клієнтів, а також бібліотек з відкритим вихідним кодом.
- Розширюваність — за допомогою просторів імен в XML можна розширити протокол Jabber для виконання необхідних завдань та для забезпечення підтримки взаємодії між різними системами. Загальні розширення розробляються під контролем Jabber Software Foundation.
- Децентралізованість — хто завгодно може запустити свій власний сервер Jabber, це дозволить організаціям та приватним особам займатися будь-якими експериментами з IM.
- Безпека — будь-який сервер Jabber може бути ізольований від загальнодоступної мережі Jabber, багато з варіантів реалізації сервера використовують SSL при обміні між клієнтом і сервером, і чимало клієнтів підтримують шифрування за допомогою PGP / GPG усередині протоколу.

Jabber-сервер OpenFire

Розглянемо розгортання корпоративного jabber-сервера на прикладі OpenFire.

Openfire (раніше відомий як **Wildfire Server** и **Jive Messenger**) — це jabber-сервер, написаний на Java.

Більша частина адміністрування сервера робиться через веб-інтерфейс, який зроблений на основі Jetty і запущений на портах 9090 (HTTP) і 9091 (HTTPS) за замовчуванням. Адміністратори можуть зайти звідки завгодно і редагувати налаштування сервера, додавати і видаляти користувачів, кімнати конференцій і так далі.

Openfire має наступні особливості:

- Веб-панель адміністрування.
- Підтримка плагінів.
- Підтримка SSL/TLS.
- Робота з базами даних для зберігання повідомлень і профілів користувачів через JDBC, а це означає, що можна використовувати Oracle, MSSQL, Postgres, DB2, Sybase ASE, MySQL, або вбудовану СУБД - HSQLDB.

- Взаємодія з LDAP (учасниками мережі можуть бути користувачі Active Directory, а вибрані групи можна автоматично публікувати в списках контактів Jabber-сумісного клієнта).
- Аутентифікація користувачів за допомогою сторонніх джерел даних.
- Платформно-незалежний, в зв'язку з використанням для розробки Java

Підготовчі роботи

Для роботи даного сервера необхідно спочатку в системі встановити Java. На жаль, OpenJRE, який входить в дистрибутив, для роботи сервера не підходить. Тому є необхідність встановлення повноцінного варіанту від Oracle.

Можна, звичайно, завантажити його з офіційного сайту і збирати руками, але це довго і важко.

Ubuntu Linux крім звичайних, підтримує ще й неофіційні ppa-репозиторії, а так само має інструменти керування ними.

Встановимо пакунок для роботи з ppa-репозиторіями:

 # apt-get install python-software-properties

Та додамо репозиторій з java:

#add-apt-repository ppa:webupd8team/java -y

Після цього потрібно оновити список доступних пакетів і встановити, власне, java:

#apt-get update

#apt-get install oracle-java7-installer

Тепер завантажимо OpenFire, який так само не входить в стандартний дистрибутив.

 # wget http://download.igniterealtime.org/openfire/openfire_3.7.1_all.deb

Та встановимо його:

 # dpkg -i openfire_3.7.1_all.deb

Для роботи OpenFire ще потрібно буде встановити змінну оточення JAVA_HOME. Для цього виконаємо:

 # export JAVA_HOME=/usr/lib/jvm/java-7-oracle

Що б кожен раз при перезавантаженні сервера не доводилося це робити вручну, відкриємо файл */etc/init.d/openfire* та в його початок (звісно, після *#!/bin/sh*) допишемо:

JAVA_HOME=/usr/lib/jvm/java-7-oracle

export JAVA_HOME

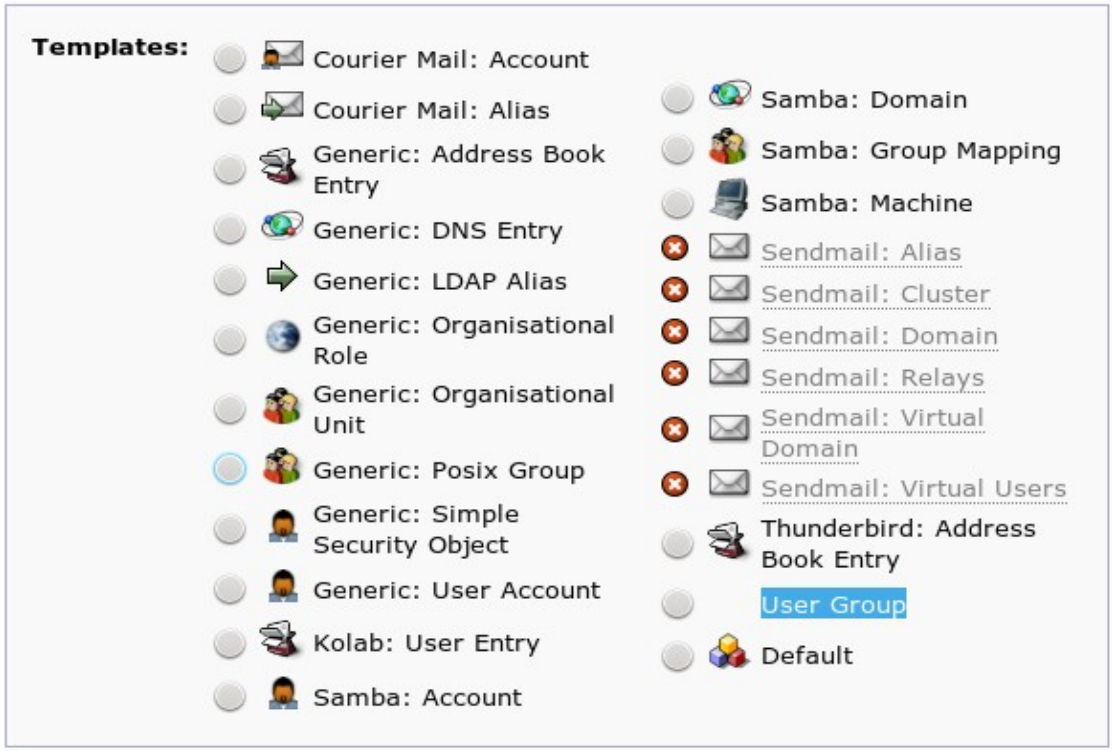
Тепер наш jabber-сервер буде стартувати без проблем.

Налаштування OpenLDAP для роботи з jabber-сервером

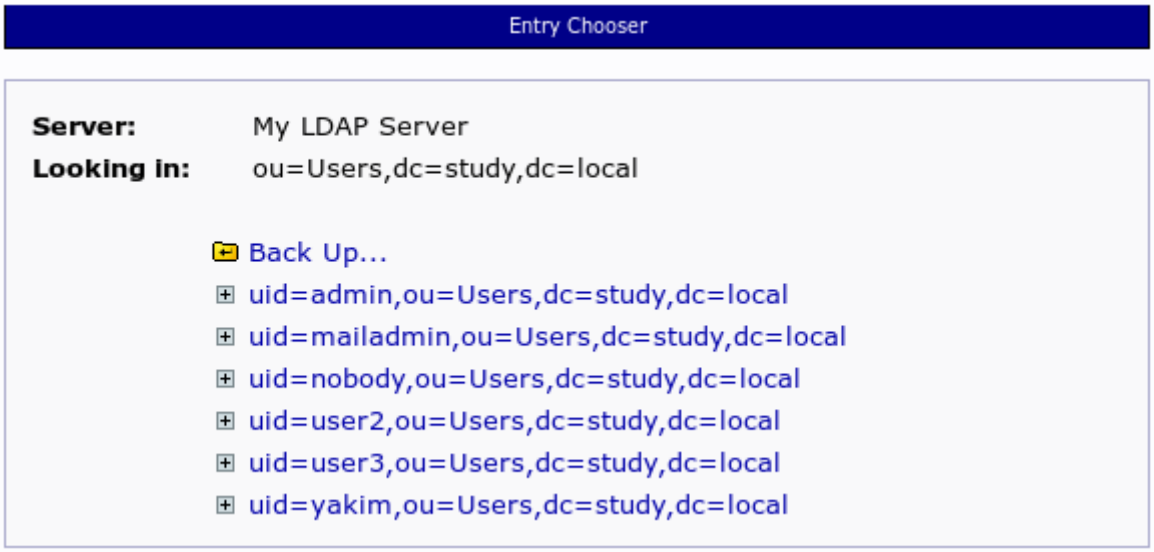
Для того, що б було простіше групувати користувачів при роботі з jabber-сервером додамо в наш домен нову гілку — ou jabber.

Далі всередині цієї сутності створимо необхідні нам групи.

При створенні груп за допомогою phpldapadmin в якості темплейтів слід вибрати «User Group».

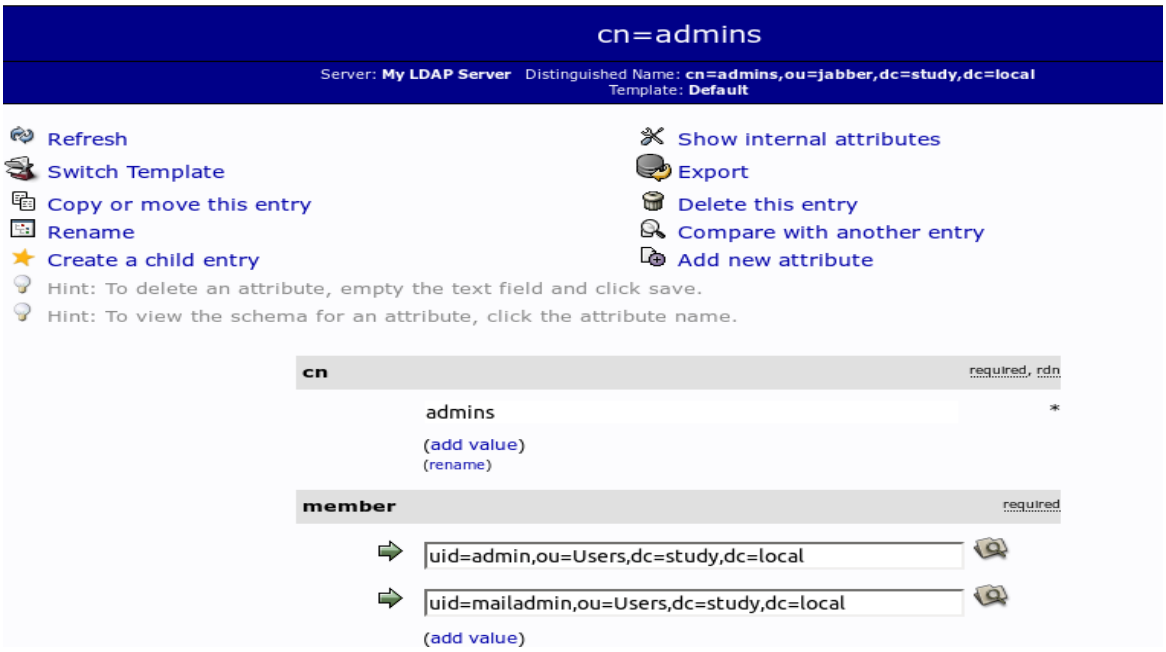


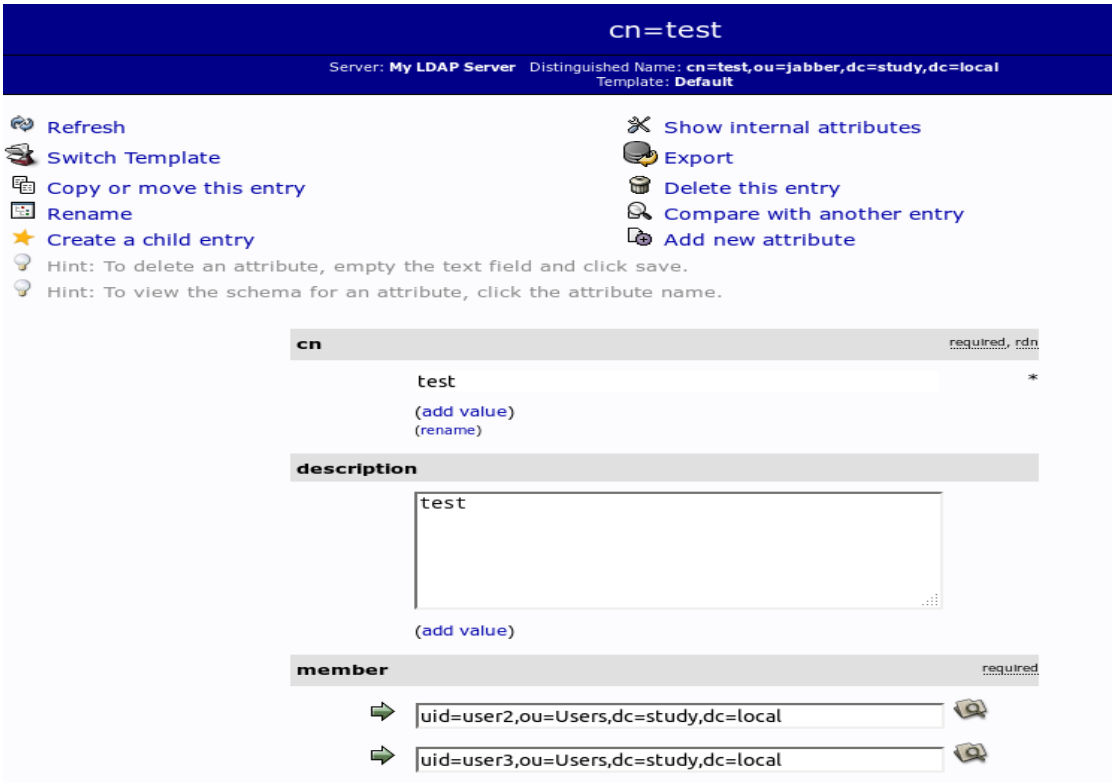
Так само відразу при створенні групи туди необхідно додати хоча б одного користувача. Для цього відкривається графічне вікно і ми вибираємо одного з існуючих користувачів.



Зверніть увагу! В даному випадку в поле «member» прописується не доменне ім'я користувача, а повний шлях опису користувача в дереві LDAP.

Таким чином ми створимо дві групи — admins і test та внесемо в них, відповідно, користувачів admin, mailadmin і user2, user3.





На цьому налаштування OpenLDAP-сервера закінчуємо, і починаємо налаштовувати безпосередньо сам jabber-сервер OpenFire.

Початкове налаштування OpenFire

Початкове налаштування сервера робиться через веб-інтерфейс. Він слухає з'єднання на порту 9090, але тільки на loopback інтерфейсі. Для того, що б нормально його відконфігурувати по мережі, відкриємо файл /etc/openfire/openfire.xml і в рядку

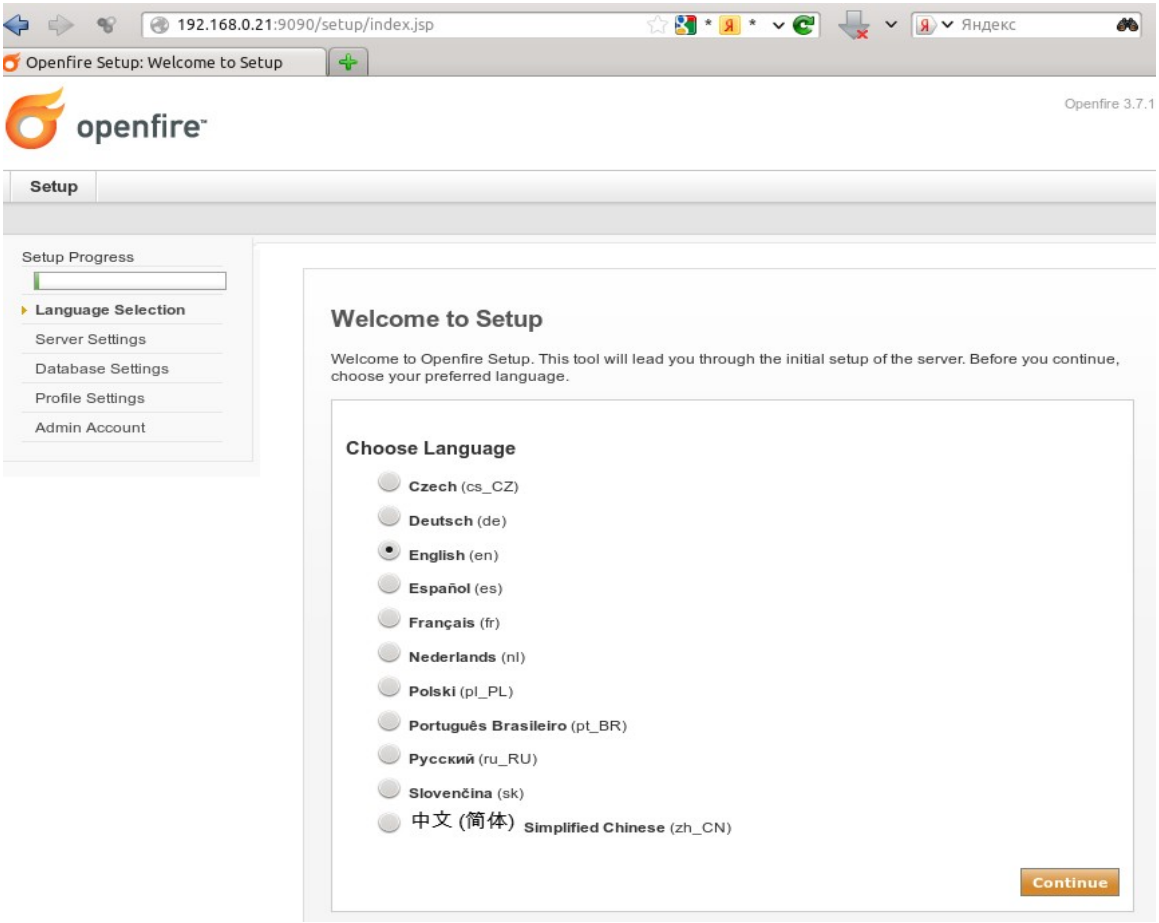
<interface></interface>

між тегами пропишемо зовнішню адресу сервера, наприклад:

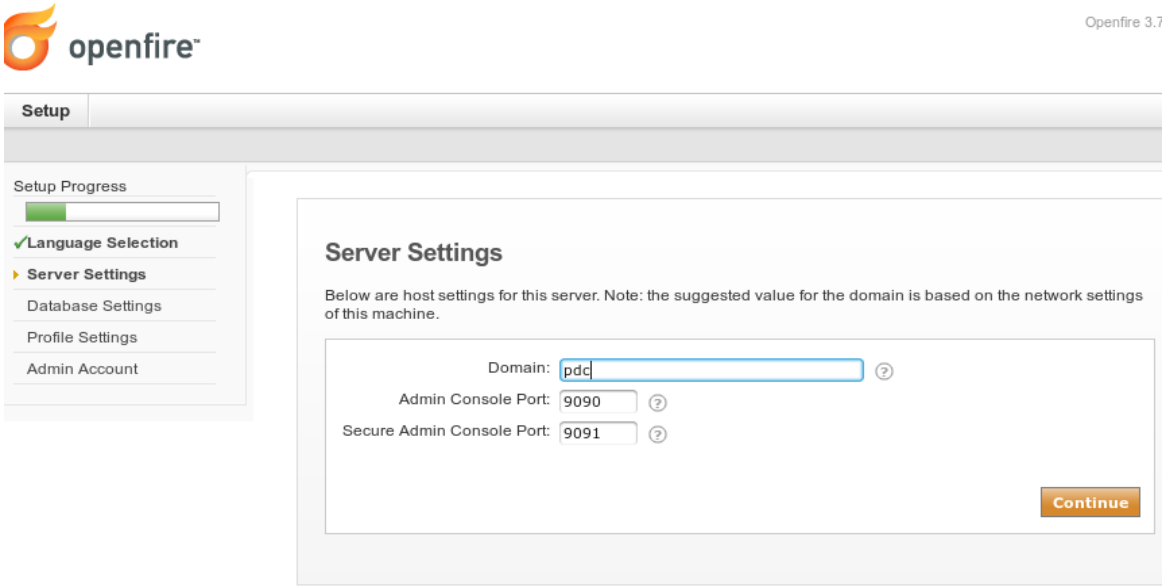
<interface>192.168.56.10</interface>

Тепер відкриємо в браузері посилання http://ip-addr:9090 та почнемо конфігурування.

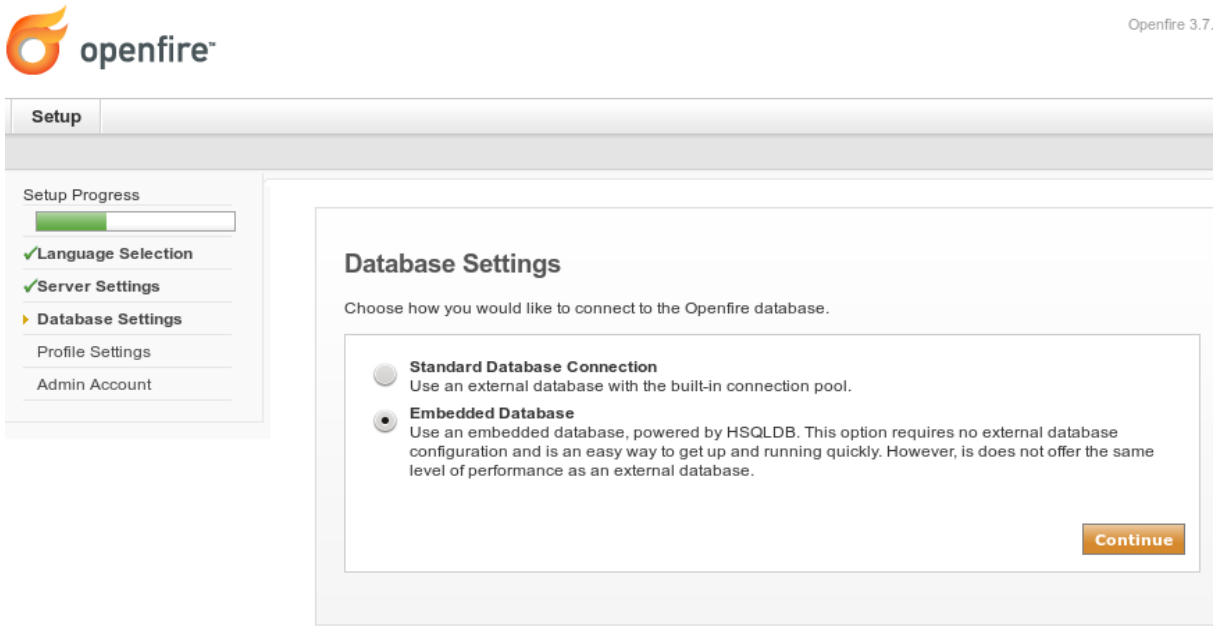
На першому кроці необхідно вибрати мову інтерфейсу. Дуже рекомендую залишити англійську. Російська локалізація зроблена, схоже, машинним перекладом та дуже важка для розуміння.



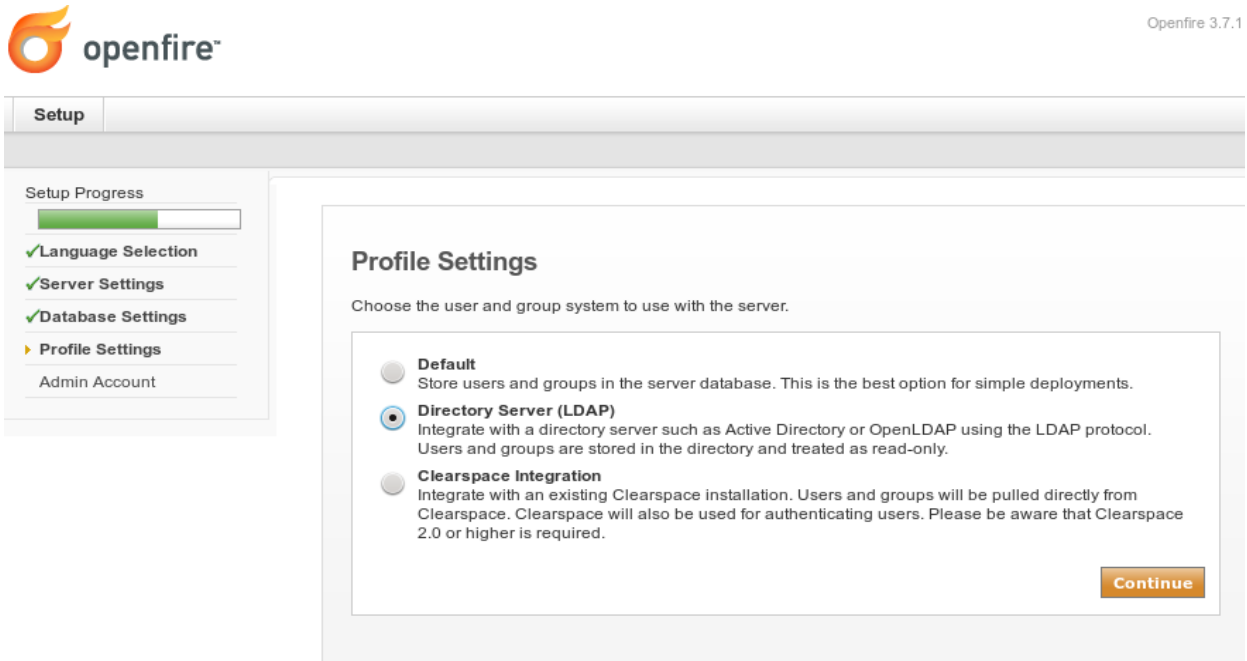
На другому кроці необхідно вказати доменне ім'я нашого сервера. Залишимо як PDC. Якщо планується робота jabber не тільки у внутрішній мережі, то потрібно вказувати FQDN, доступний зі світу.



На третьому кроці обираємо тип бази даних, що буде використовуватися. Оберемо внутрішню базу.

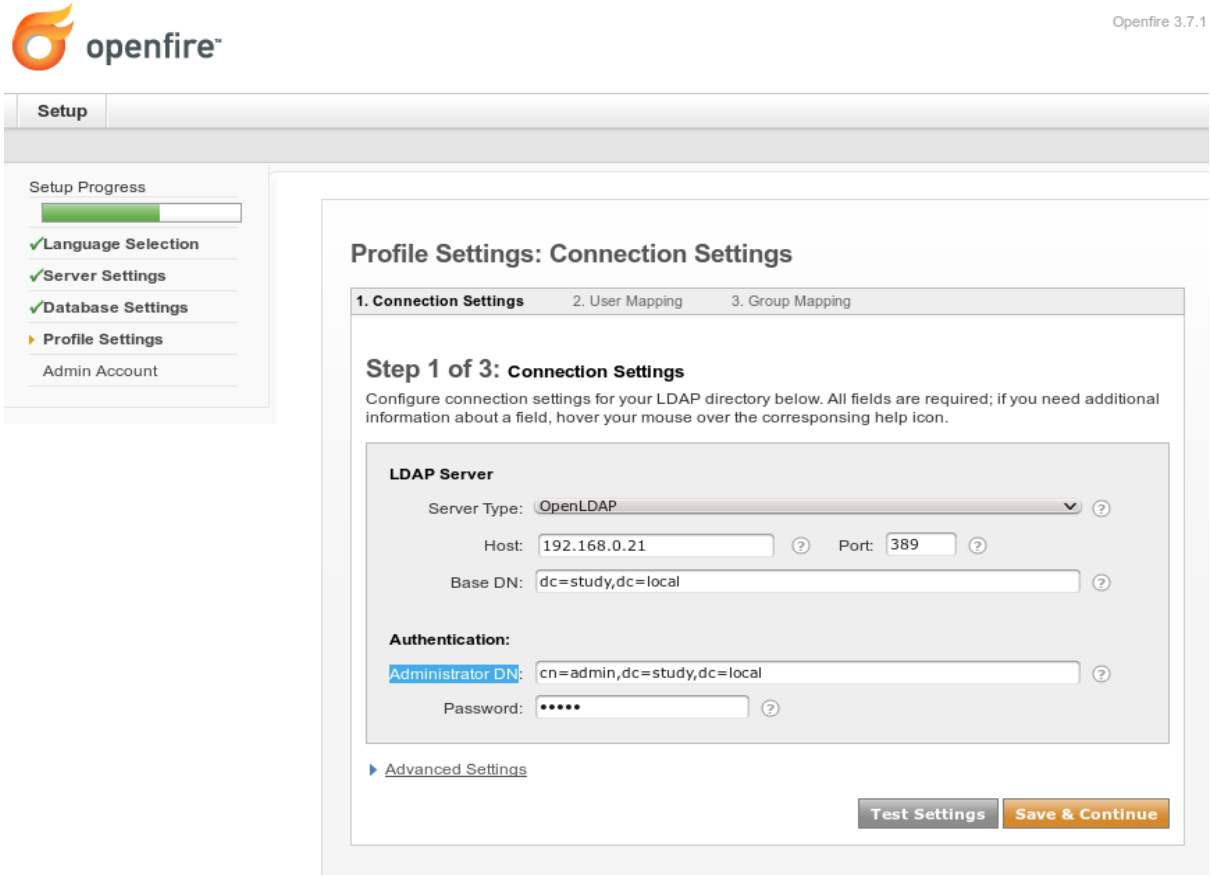


У зв'язку з тим, що всі наші користувачі зберігаються в OpenLDAP, то на цьому кроці вкажемо використання LDAP.

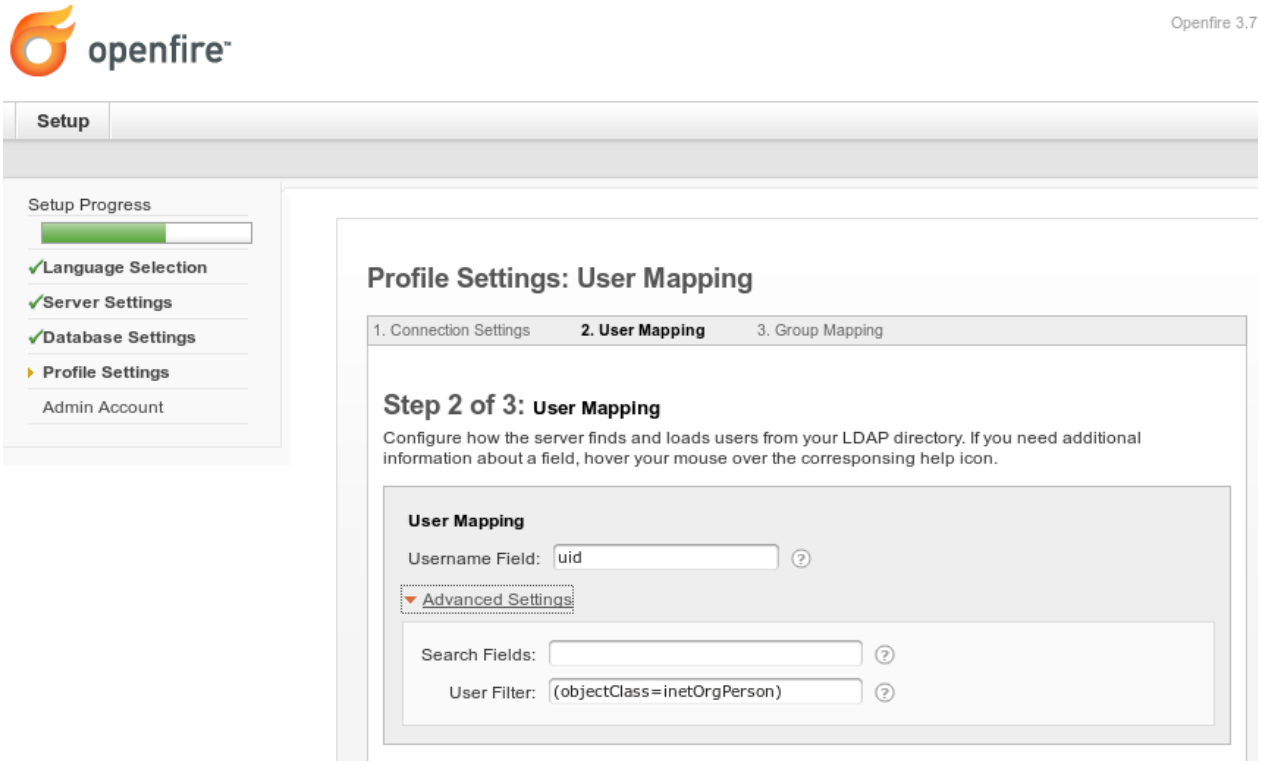


Далі йде налаштування параметрів з'єднання з OpenLDAP.

На першому кроці ми вказуємо тип LDAP-сервера, його адресу та обліковий запис для підключення. Зверніть увагу, що поля "Base DN" та "Administrator DN" вказуються зовсім не так, як звично при роботі з Microsoft AD



Якщо всі поля заповнені правильно, то після натискання на кнопку «Test Settings» має з'явитися повідомлення про успішне з'єднання.

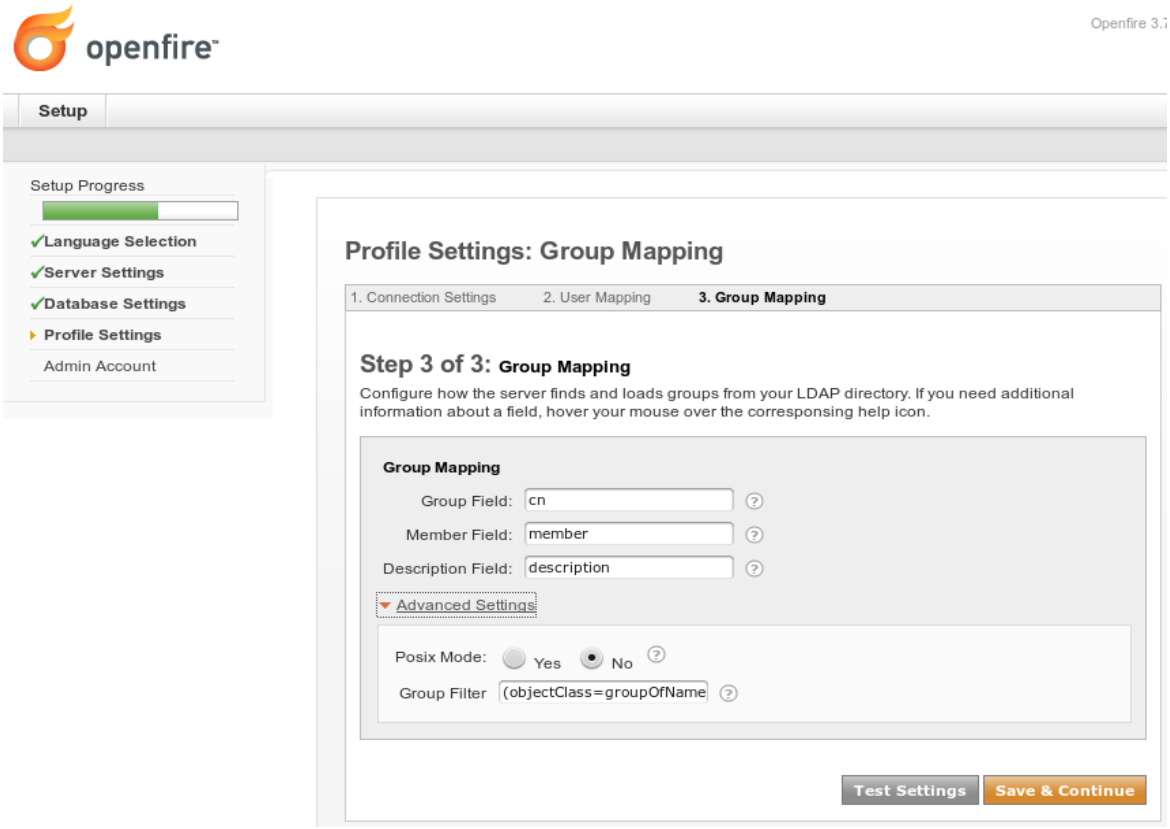


На другому кроці ми налаштовуємо вибірку користувачів домену. Зверніть увагу, що в Username Field повинно бути прописано uid, а в User Filter — (objectClass = inetOrgPerson). Якщо поле User Filter залишити порожнім, то в список користувачів потраплять навіть комп'ютери домену.

На третьому кроці налаштування взаємодії з OpenLDAP ми налаштуємо вибірку груп користувачів.

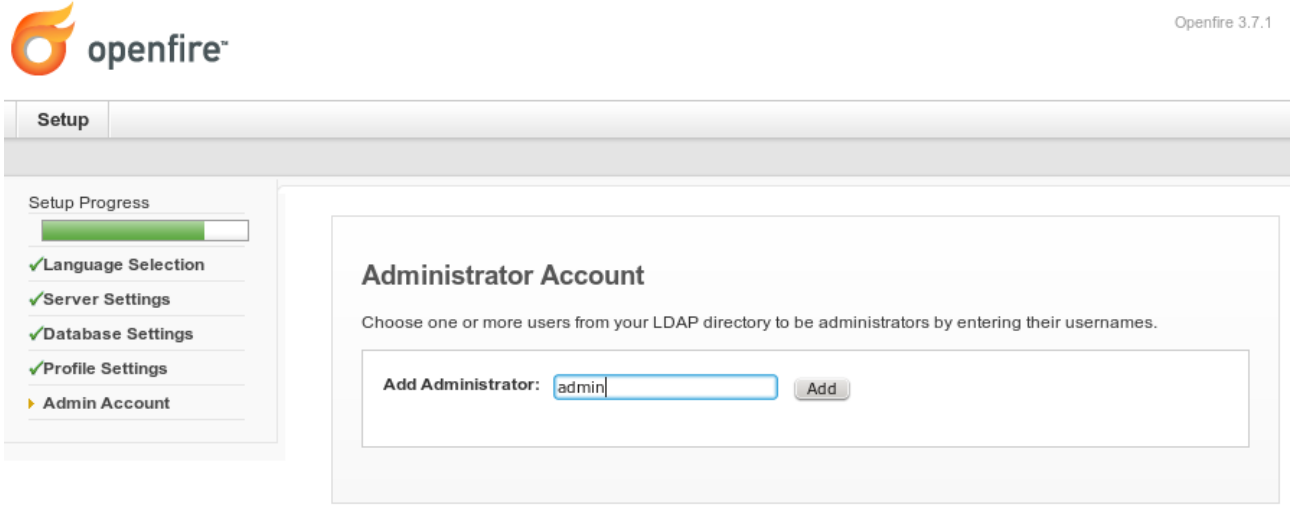
Знову ж таки необхідно правильно налаштувати поля.

Group Field: cn
Member Field: member
Description Field: description
Group Filter (objectClass=groupOfNames)

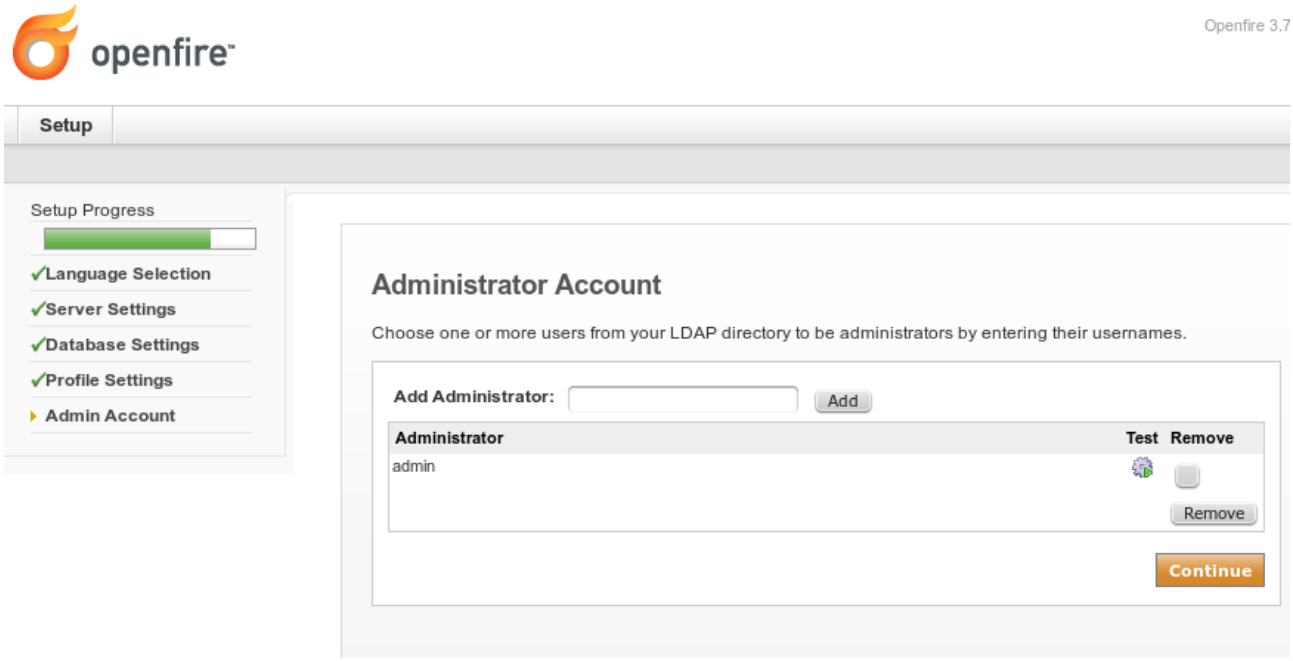


Залишилося додати адміністратора системи.

Зверніть увагу, що в якості адміністратора використовується один з облікових записів OpenLDAP.



Далі, при натисканні на іконку «test», можна перевірити правильність пароля.



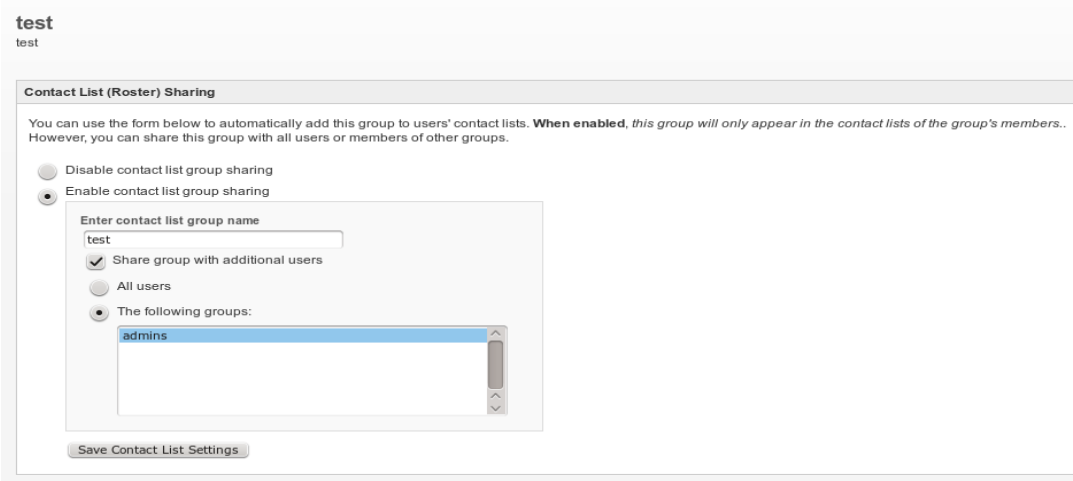
На цьому базове налаштування завершено і можна перемикатися в адміністраторську консоль сервера.

Налаштування OpenFire

Користувачі та групи автоматично додаються з сервера OpenLDAP, тому локальне керування ними не можливе.

Єдине що можна зробити з групами через інтерфейс OpenFire — це автоматично додавати їх до контактів користувачів.

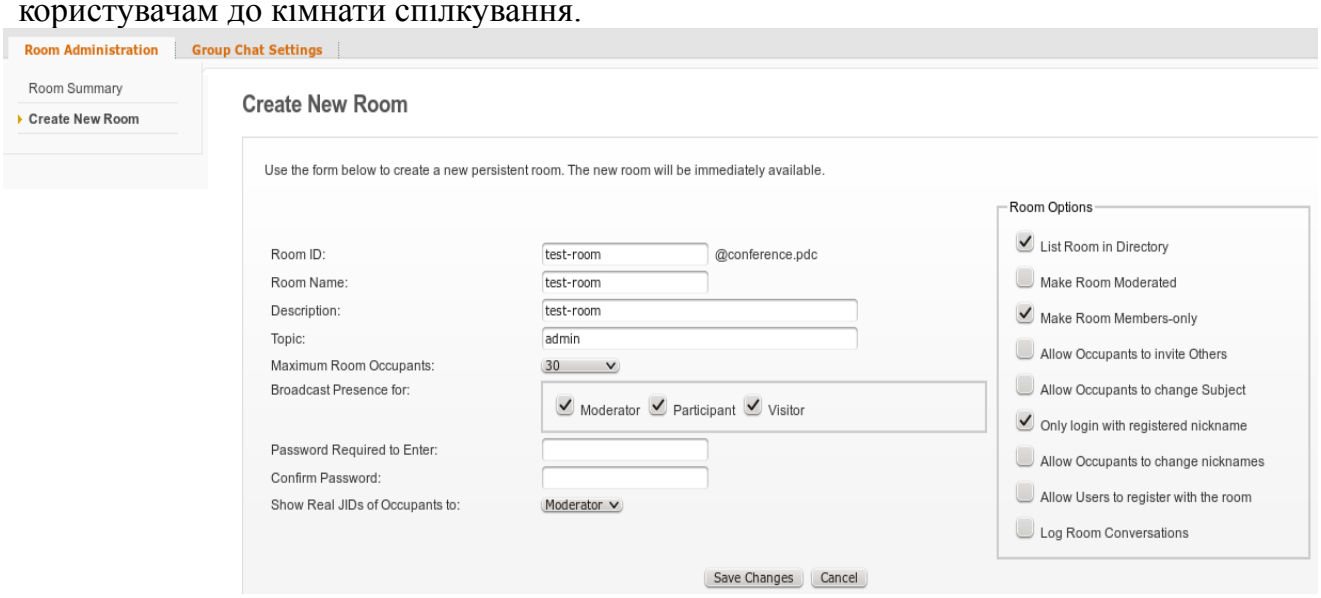
Для цього в табі «Users/Groups» на вкладці «Groups» можна зайти до параметрів обраної групи і налаштувати автоматичне додавання цієї групи певним іншим користувачам.



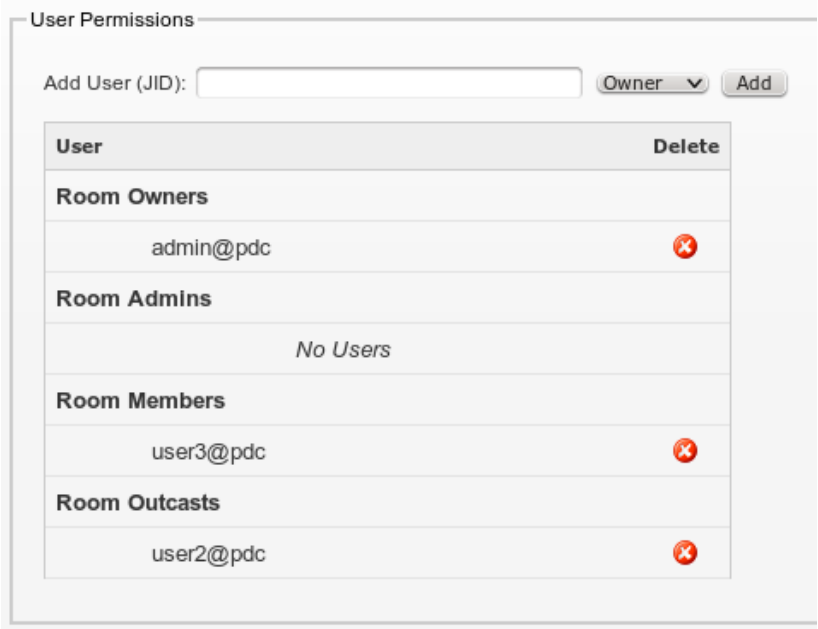
Створення кімнат спілкування

Якщо перейти на закладку «Group Chat» можна створити кімнату спілкування, а так само налаштувати її необхідним чином.

Кімната може бути доступна для всіх, тільки для зареєстрованих користувачів, а так само є можливість у явному вигляді дозволяти або забороняти доступ користувачам до кімнати спілкування.



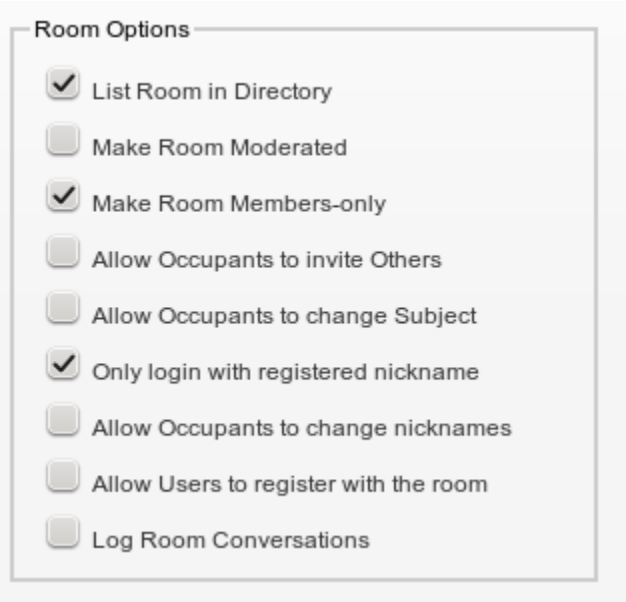
Якщо перейти на пункт меню «User Permissions», то ми побачимо, як в явному вигляді можна дозволяти або забороняти доступ користувачів до кімнати.



На показаному скріншоті видно, що користувачеві user2 доступ в кімнату заборонений, а користувачеві user3 — дозволений.

З налаштуванням прав користувачів є нюанс. Не можна помістити в заборонені

власника (творця) кімнати і адміністратора кімнати. Вони автоматично вносяться до списку дозволених користувачів. Крім цього, список заборонених користувачів працює завжди, а список дозволених — тільки в тому випадку, якщо в налаштуванні кімнати встановлений прапорець «Make room Members-Only»



Резервне копіювання серверів та робочих станцій в офісній мережі

Всі системні адміністратори діляться на дві категорії — ті хто не робить резервне копіювання і ті, хто вже робить.

Необхідність резервного копіювання зрозуміла практично всім адміністраторам. Але, на жаль, дуже часто трапляється так, що не вдається пояснити необхідність закупівлі комерційного програмного забезпечення для цих цілей.

Звичайно існує велика кількість комерційних рішень для створення резервних копій чого завгодно: як окремих файлів, так і конкретно поштових баз, структури AD або навіть всього сервера відразу. Так само є чимало і вільних і безкоштовних рішень для цього. Наприклад Bacula, Amanda, dar, BackUpPC та інші.

Найбільш потужним і універсальним рішенням є, звичайно, Bacula. Але вона ж і найбільш складна в налаштуваннях. Для більшості ситуацій буде достатньо використовувати BackUpPC.

BackupPC — це вільне ПЗ (розповсюджується під GNU General Public License) для резервного копіювання даних з управлінням через веб-інтерфейс. Багатоплатформовий програмний сервер може працювати на будь-якому сервері під управлінням GNU/Linux, Solaris або UNIX. Немає необхідності в клієнтській частині, так як сервер сам по собі є клієнтом для декількох протоколів, що підтримуються рідними службами клієнтської ОС.

Наприклад, BackupPC є SMB-клієнтом, що може використовуватися для резервного копіювання спільно використовуваних даних в мережевих папках на комп'ютерах з Microsoft Windows. Подібний сервер BackupPC може бути встановлений за міжмережевим екраном, що виконує функції мережевої трансляції адрес (NAT), коли Windows-комп'ютер має публічну IP адресу. Так як це не рекомендується через велику кількість SMB трафіку, то більш зручним є використання веб-серверів, що підтримують SSH і можливість роботи з tar і rsync, що дозволяє серверу BackupPC знаходитися в підмережі відокремленою від веб-серверів демілітаризованою зоною.

Встановлення системи резервного копіювання BackUpPC

Так як управління BackupPC проводиться через веб-інтерфейс, то спочатку необхідно встановити веб-сервер:

 # apt-get install apache2

Далі встановимо сам сервер резервного копіювання і пакунок smbclient для роботи з windows-комп'ютерами:

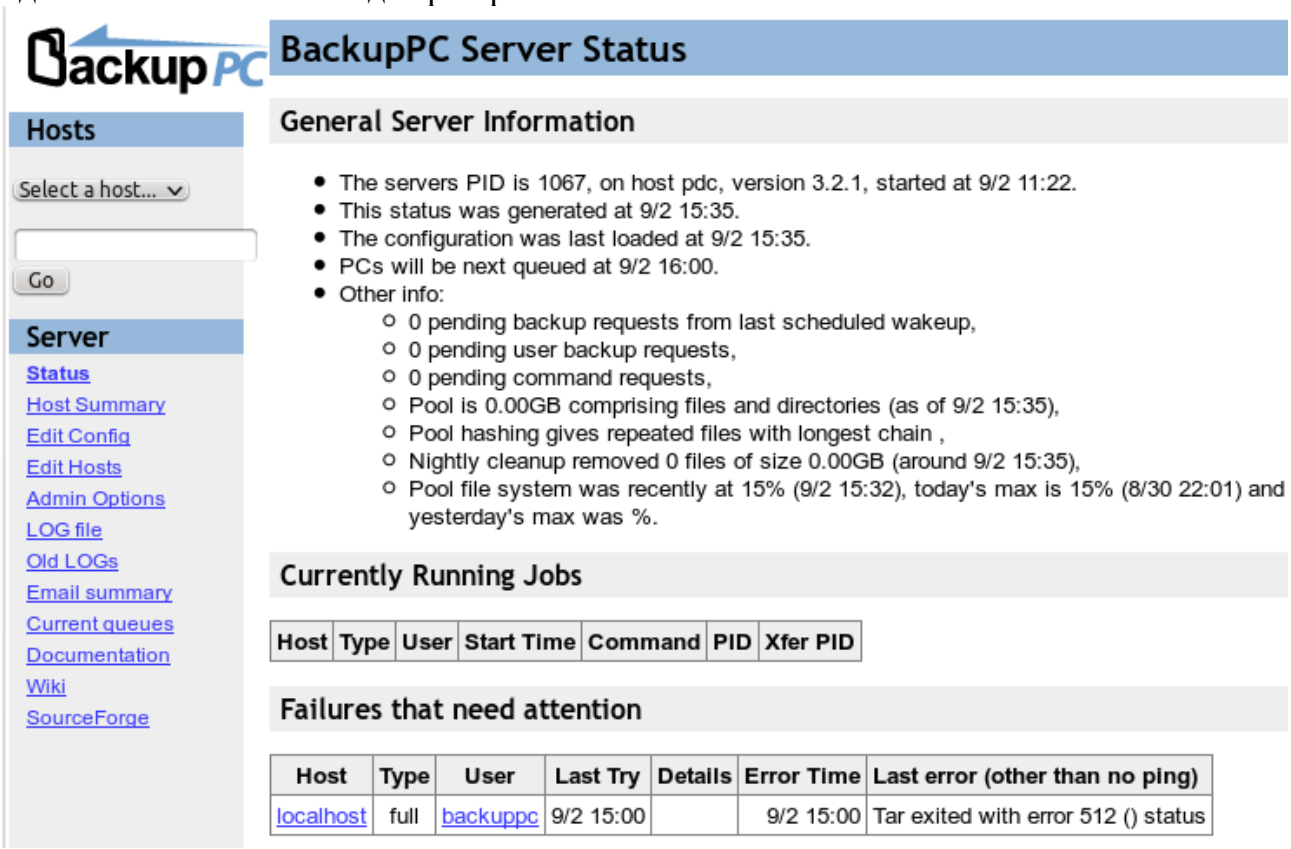
 # apt-get install backuppc smbclient

Після закінчення встановлення його веб-інтерфейс буде доступний за адресою *http://ip-addr/backuppc*

Для входу в систему будемо використовувати логін backuppc і пароль password.
Для того, що б змінити пароль для цього користувача слід ввести команду:
 # htpasswd /etc/backuppc/htpasswd backuppc

Додати нового користувача (наприклад admin) і поставити йому пароль можна командою:
 # htpasswd /etc/backuppc/htpasswd admin

Після першого логіна в систему ми побачимо вікно, в якому прописана тільки одна клієнтська машина для резервного копіювання — localhost



Файли та шляхи, що використовуються в BackupPC

Основний файл налаштувань даної системи резервного копіювання — */etc/backuppc/config.pl*

У цій же теці знаходиться файл *hosts*, в якому прописані всі клієнтські комп'ютери з яких буде проводиться резервне копіювання.

Так же в цій теці повинні знаходитися файли з розширенням pl, що називаються по іменах клієнтських комп'ютерів. Саме в них і будуть зберігатися всі індивідуальні налаштування резервного копіювання.

У теці */var/lib/backuppc/pc* будуть зберігатися резервні копії всіх серверів. Тому необхідно або передбачити достатню кількість дискового простору по цьому шляху, або перевизначити його у файлі */etc/backuppc/config.pl*

Ще один важливий шлях — */usr/share/backuppc/lib/BackupPC/Lang*

Саме там зберігаються всі файли локалізації інтерфейсу. Переклад є на англійську, чеську, іспанську та деякі інші мови. На жаль ні український, ні російський переклади туди не входять.

Виправлення помилок після встановлення програми

У зв'язку з тим, що в Linux багато програм пов'язані між собою, після чергового оновлення модулів perl при запуску BacupPC з'являються помилки:

Use of qw(...) as parentheses is deprecated at /usr/share/backuppc/lib/BackupPC/Storage/Text.pm line 302.

Use of qw(...) as parentheses is deprecated at /usr/share/backuppc/lib/BackupPC/Lib.pm line 1425.

Для їх виправлення необхідно у файлі */usr/share/backuppc/lib/BackupPC/Storage/Text.pm* замінити рядок

foreach my $param qw(BackupFilesOnly BackupFilesExclude) {

на

foreach my $param (qw(BackupFilesOnly BackupFilesExclude)) {

та в файлі */usr/share/backuppc/lib/BackupPC/Lib.pm* рядок

foreach my $param qw(BackupFilesOnly BackupFilesExclude) {

замінити на

foreach my $param (qw(BackupFilesOnly BackupFilesExclude)) {

Конфігурування клієнтського Linux-хоста

Додамо в кінець файлу */etc/backuppc/hosts* рядок:

linuxsrv1 0 backuppc

Перше поле, це зрозуміле нам ім'я хоста, друге — включення (1) або відключення (0) розкладу резервного копіювання, і третє — користувач, який має доступ до управління даним хостом.

Далі створимо файл конфігурації цього сервера:

#nano /etc/backuppc/linuxsrv1.pl

Та заповнимо його наступним чином:

```
$Conf{BackupsDisable} = '1';
$Conf{ClientNameAlias} = '192.168.56.20';
$Conf{BackupFilesOnly} = {
  '*' => [
    '/etc'
  ]
};

$Conf{XferMethod} = 'rsync';
$Conf{RsyncClientPath} = '/usr/bin/rsync';
$Conf{RsyncClientCmd} = '$sshPath -q -x -l bcuser $host sudo $rsyncPath $argList+';
$Conf{RsyncClientRestoreCmd} = '$sshPath -q -x -l bcuser $host sudo $rsyncPath $argList+';
$Conf{RsyncShareName} = [
  '/'
];
$Conf{RsyncCsumCacheVerifyProb} = '0.01';
$Conf{XferLogLevel} = '9';
$Conf{ClientCharsetLegacy} = 'utf8';
$Conf{BackupFilesExclude} = {
  '/' => [
    '/etc/ssh'
  ]
};
```

У цьому файлі ми вказуємо адресу клієнтського сервера

```
$Conf{ClientNameAlias} = '192.168.56.20';
```

Шлях копіювання:

```
$Conf{BackupFilesOnly} = {  '*' => [    '/etc'  ]};
```

Шлях виключень:

```
$Conf{BackupFilesExclude} = {  '/' => [    '/etc/ssh'  ]};
```

Та тип з'єднання:
$Conf{XferMethod} = 'rsync';

Далі потрібно зробити можливість нашому серверу резервного копіювання авторизуватися на віддаленій системі по протоколу ssh не за паролем, а за ключем.

Створюємо ключ для користувача backuppc, з правами якого і працює наш сервер резервного копіювання:

 # sudo -u backuppc ssh-keygen -t rsa

При запиті пароля на ключ, залишаємо його порожнім.

Далі на клієнтському сервері створюємо користувача bcuser і в налаштуваннях sudo (*# visudo*) прописуємо:

bcuser ALL=(ALL) NOPASSWD:/usr/bin/rsync

Тобто цьмоу користувачу ми даємо можливість без пароля запускати rsync з підвищеними правами.

Тепер повертаємося на наш сервер резервного копіювання. Ключ для авторизації на ssh ми вже створили і тепер потрібно передати його на клієнтський сервер.

 # sudo -u backuppc ssh-copy-id -i /var/lib/backuppc/.ssh/id_rsa.pub bcuser@192.168.56.20

де 192.168.56.20 — IP-адреса клієнтського Linux-сервера.

Намагаємося з'єднатися з авторизацією за ключем:

 # sudo -u backuppc ssh bcuser@192.168.0.27

Якщо з'єднання пройшло успішно, значить на цьому налаштування завершено.

Перезапустимо наш сервер резервного копіювання:

#service backuppc restart

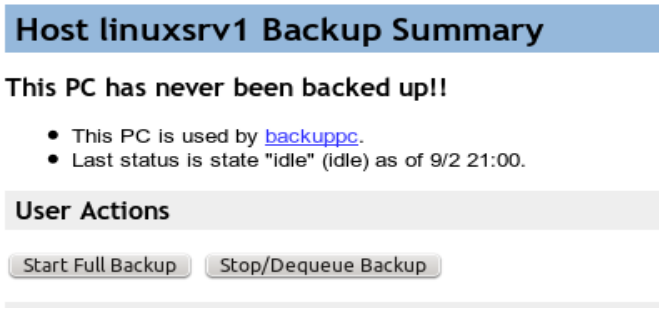
Відкривши його веб-інтерфейс і перейшовши на закладку «Host Summary» в списку клієнтських серверів ми побачимо наш тільки що описаний сервер linuxsrv1.

Hosts with no Backups

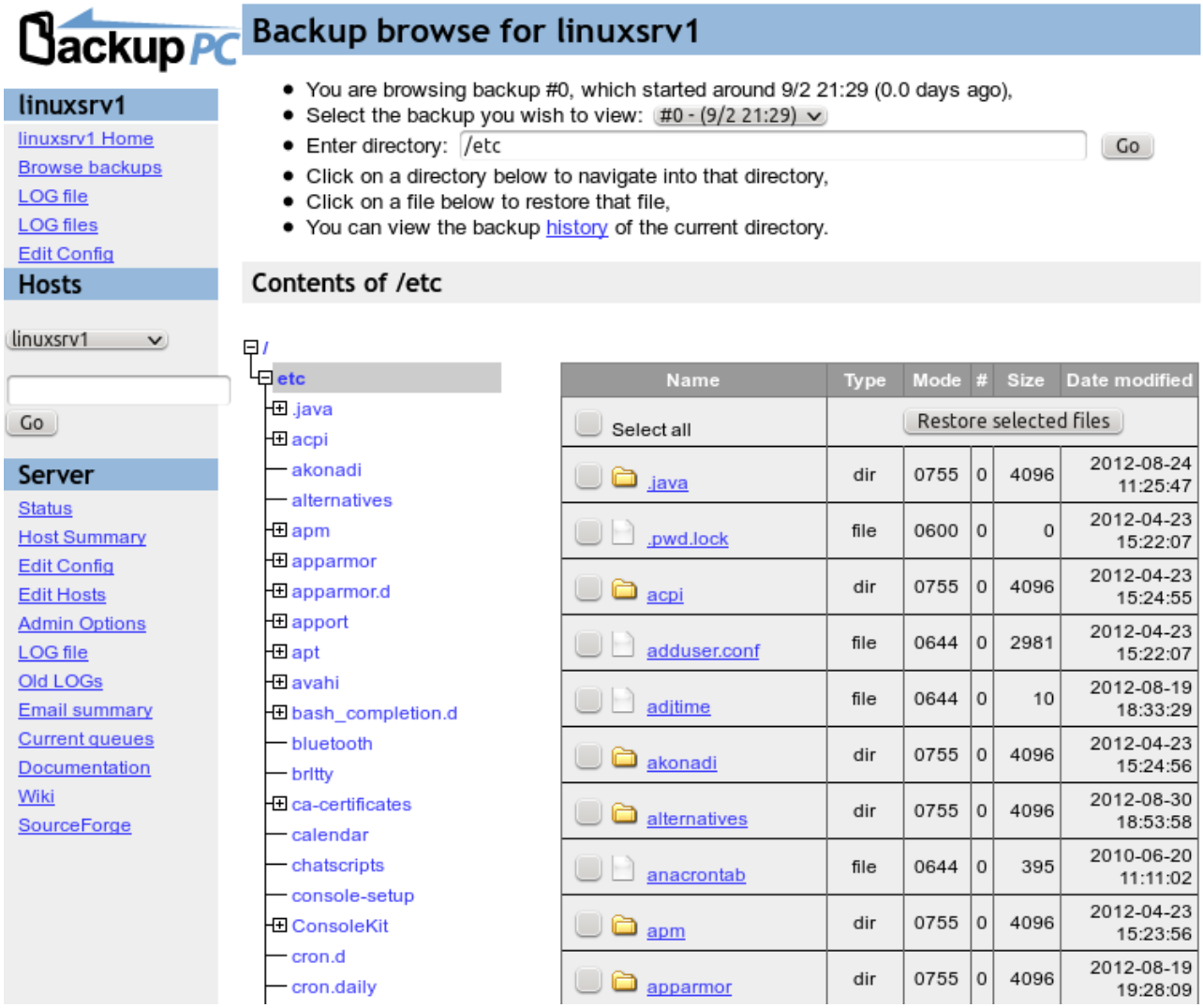
There are 1 hosts with no backups.

Host	User	#Full	Full Age (days)	Full Size (GB)	Speed (MB/s)	#Incr	Incr Age/days	Last Backup (days)	State	#Xfer errs	Last attempt
linuxsrv1	backuppc	0		0.00		0			auto disabled		

Зайдемо всередину цього хоста і натиснемо кнопку «Start Full Backup»



Через деякий (в нашому випадку близько 15-20 секунд) час, перейшовши на закладку «Browse backups» Ми зможемо побачити все дерево резервної копії з підлеглого сервера.



Якщо уважно подивитися на список вкладених тек, то ми побачимо, що теки /etc/ssh там немає, що говорить про те, що списки виключення працюють коректно.

Конфігурування клієнтського Windows-хоста

Резервне копіювання Windows-машин проводиться по протоколу SMB.

Для додавання нового клієнтського хоста в систему додамо в кінець файлу */etc/backuppc/hosts* рядок:

winxp 0 backuppc

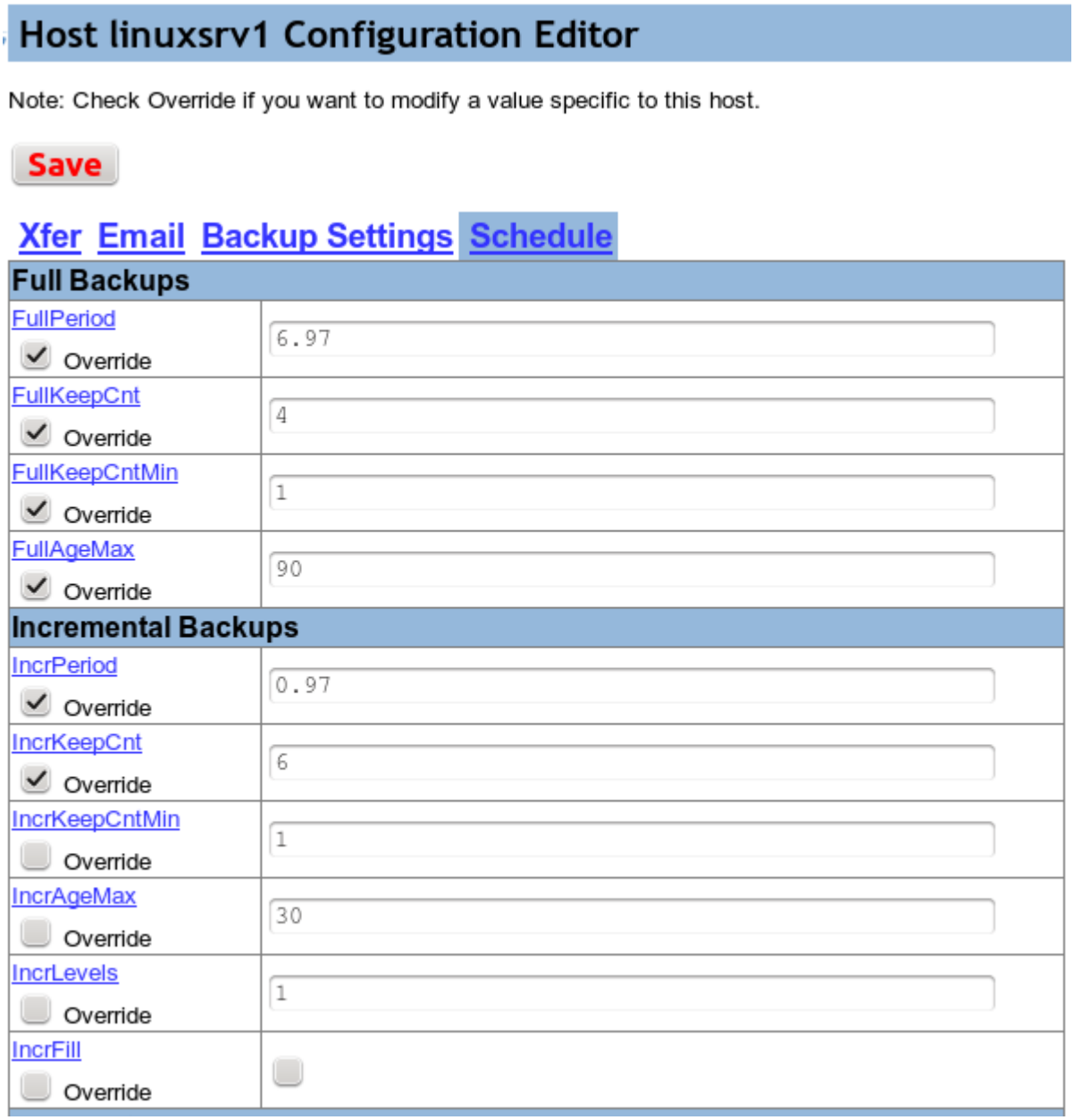
Та створимо файл */etc/backuppc/winxp.pl* з таким вмістом:

```
$Conf{ClientNameAlias} = '192.168.0.12';
$Conf{BackupFilesOnly} = {
 'C$' => ['/Documents and Settings/User/Мои документы'],
};
$Conf{XferMethod} = 'smb';
$Conf{SmbShareName} = [ 'C$' ];
$Conf{SmbShareUserName} = 'STUDY\\admin';
# Password should be configured on client
$Conf{SmbSharePasswd} = 'admin';
$Conf{PingMaxMsec} = '900';
$Conf{SmbClientFullCmd} = '$smbClientPath \\\\$host\\$shareName $I_option -U $userName -E -d 1 -c tarmode\\ full -Tc$$
$Conf{SmbClientIncrCmd} = '$smbClientPath \\\\$host\\$shareName $I_option -U $userName -E -d 1 -c tarmode\\ full -TcN$
$Conf{SmbClientRestoreCmd} = '$smbClientPath \\\\$host\\$shareName $I_option -U $userName -E -d 1 -c tarmode\\ full -$
$Conf{ClientCharset} = 'cp1252';
```

Принцип створення файлу такий же самий, як і для Linux-сервера.

Створення розкладу автоматичного копіювання.

Налаштування розкладу резервного копіровнія проводиться у властивостях конкретного хоста на вкладці «Shedule»



Тут маються наступні поля налаштувань:
Повна резервна копія:

FullPeriod — мінімальний час в днях між повними бекапами

FullKeepCnt — скільки повних бекапів необхідно зберігати

FullKeepCntMin — мінімальна кількість збережених повних бекапів

FullAgeMax — максимальний вік збереженого повного бекапа

Інкрементальна резервна копія:

IncrPeriod — мінімальний час в днях між інкрементальними бекапами

IncrKeepCnt — скільки інкрементальних бекапів необхідно зберігати

IncrKeepCntMin — мінімальна кількість збережених інкрементальних бекапів

IncrAgeMax — максимальний вік збереженого інкрементального бекапу

IncrLevels — рівень інкрементального бекапу

IncrFill — використовувати в системі хард-лінки, що б інкрементальний бекап виглядав повним.

Переклад інтерфейсу

Для того, що б перекласти веб-інтерфейс українською мовою необхідно спочатку взагалі додати його підтримку в BackupPC.

Для цього відкриємо файл
/usr/share/backuppc/lib/BackupPC/Config/Meta.pm

і замінимо в ньому рядок
values => [qw(cz de en es fr it nl pl pt_br zh_CN)],

на

values => [qw(cz de en es fr it nl pl pt_br zh_CN uk)],

Тепер завантажимо файл підтримки української мови
http://yakim.org.ua/images/stories/articles/backuppc_uk.tgz

та розпакуємо його вміст в */usr/share/backuppc/lib/BackupPC/Lang*
Далі в файлі */etc/backuppc/config.pl* змінюємо в рядку

$Conf{Language} = 'en';

мову на потрібну нам. В даному випадку на
$Conf{Language} = 'uk';

Тепер перечитуємо конфігурацію сервера
 # service backuppc reload

Ось і все. Ми маємо український інтерфейс.

BackupPC

Хости

Виберіть хост...

Перейти

Сервер

- Статус
- **Зведена інформація по хостам**
- Правити конфігурацію
- Правити хости
- Адмінські налаштування
- LOG файл
- Старі LOGи
- Поштові налаштування
- Поточні черги
- Документація (англ)
- Wiki
- SourceForge

BackupPC: Зведена інформація по хостах

- Цей статус було згенеровано 9/2 22:25.
- Файлова система пула зайнята на 15% (9/2 22:18), сьогодняшній максимум 15% (8/30 22:01) вчорашній максимум %.

Хости, що мають резервні копії

Загалом 2 хостів, що містять:

- 2 загальний розмір повних резервних копій 0.01GB (до об'єднання та стискання),
- 0 загальний розмір інкрементальних резервних копій 0.00GB (до об'єднання та стискання).

Хост	Користувач	#Кіль-ть повн. копій	Вік повн. копій (дні)	Повний розмір (GB)	Швидкість (MB/s)	#Кіль-ть інкр. копій	Вік інкр. копій (дні)	Остання копія (days)	Стан	#Xfer помилки	Остання дія
linuxsrv1	backuppc	1	0.0	0.01	0.50	0		0.0	auto відключене	0	
winxp	backuppc	1	3.0	0.00	1.24	0		3.0	бездія	0	no ping (no ping response)

Хости без резервних копій

Загалом 0 хостів без резервних копій.

Хост	Користувач	#Кіль-ть повн. копій	Вік повн. копій (дні)	Повний розмір (GB)	Швидкість (MB/s)	#Кіль-ть інкр. копій	Вік інкр. копій (дні)	Остання копія (days)	Стан	#Xfer помилки	Остання дія

Фільтрація трафіку за допомогою L7-Filter

7-filter — програмний пакет, який представляє собою класифікатор для підсистеми Netfilter в ОС Linux, який може розподіляти по категоріях IP-пакети, базуючись на даних прикладного рівня. Основна мета цього інструменту полягає в тому, щоб зробити можливим виявлення трафіку файлообмінних мереж (також p2p), клієнти яких використовують непередбачуване число портів.

Існують дві версії цього програмного продукту. Перша реалізована у вигляді модуля для ядра Linux 2.4 і 2.6. Друга, експериментальна версія, була випущена в грудні 2006 року в якості userspace-програми і для класифікації спирається на простір користувача бібліотек netfilter.

L7-filter дозволяє Netfilter ідентифікувати пакет на прикладному рівні даних, ґрунтуючись на його вмісті, і класифікувати пакети за їх призначенням, без прив'язки до номера порту. В даний час підтримуються протоколи HTTP і FTP; P2P мережі (Kazaa, BitTorrent, eDonkey2000, FastTrack); IM-системи (AIM / Jabber / IRC / MSN); VoIP / Skype; VPN; гри (Battlefield, CS, Doom3, WoW) ; файли (exe, mp3) і навіть черв'яки на зразок Code Red і Nimda.

Встановлення l7-filter userspace

Для встановлення необхідно виконати команду:
 # apt-get install l7-filter-userspace l7-protocols

Цим ми встановлюємо пакет *l7-filter-userspace* і бібліотеку описів протоколів, які він використовує.

Якщо перейти до теки / etc/l7-protocols то там можна побачити описи різних протоколів, що можуть визначатися за допомогою l7-filter.

Налаштування протоколів для l7-filter

Відкриємо файл */etc/l7_filter.conf*
Та запишемо в нього:
ssh 5
http 7

Тут ми описали 2 типи протоколів, які будуть визначатися за допомогою l7-filter і для кожного з них визначили певне числове значення (мітку) яка буде присвоюватися

пакету і по якому його згодом можна буде відфільтрувати та заблокувати.

Налаштування IPTables для роботи з l7-filter

Для початку побудуємо найпростіший варіант NAT
 # echo 1 > /proc/sys/net/ipv4/ip_forward
 # iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
 # iptables -t nat -A POSTROUTING -o eth0 -s 192.168.56.0/24 -j MASQUERADE
 # iptables -A FORWARD -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT

Далі завантажимо модуль, що потрібний для роботи l7-filter
 # modprobe ip_conntrack_netlink

Тепер створимо правило для IPTables, яке буде передавати всі транзитні пакети до черги №1
 # iptables -t mangle -I FORWARD -j NFQUEUE --queue-num 1

І створимо правило, яке буде видаляти пакети з міткою 7, тобто пакети, які будуть відноситися до протоколу HTTP.
 # iptables -t filter -I FORWARD -m mark --mark 7 -j DROP

Зверніть увагу, що в даний момент трафік крізь наш тестовий шлюз не ходить. Це відбувається тому, що всі транзитні пакети передаються в чергу, яку ніхто не обробляє і, відповідно, не повертає керування назад в ланцюжки IPTables.

Тепер запустимо сам l7-filter
 # l7-filter -vv -f /etc/l7_filter.conf -q 1

У цьому рядку ми використовуємо наступні параметри:
- vv — виводити докладну інформацію про пакети, що перевіряються
-f — вказуємо, який файл конфігурації використовувати
-q — вказуємо номер черги пакетів, яку потрібно обробляти.

Тепер можна перевірити зроблені налаштування з комп'ютера внутрішньої мережі.
Ми можемо переконатися, що, наприклад, пінги в зовнішню мережу у нас проходять, а по протоколу HTTP з'єднання буде обриватися по таймауту.

Епілог

Я спробував в цьому навчальному курсі розглянути моменти, які найбільш часто зустрічаються в роботі з доменною мережею.

Дуже сподіваюся, що інформація, яка наведена вище допоможе Вам налаштувати свою мережу з мінімальними зусиллями.

Не дивлячись на те, що в основу курсу покладений дистрибутив Ubuntu Server 12.04, практично всі наведені приклади будуть працювати і на будь-якому іншому дистрибутиві.

Використані матеріали

При підготовці цього навчального курсу і, відповідно, навчального матеріалу, широко використовувалася інформація з різних сайтів, які доступні в інтернеті. Зараз спробую перерахувати їх.

Якщо когось випадково забув — пишіть і я включу ваш сайт в список використаних матеріалів.

http://www.wikipedia.org/ — на різних мовах

http://opennet.ru/ — там взагалі багато цінної інформації

https://help.ubuntu.com — дуже корисний ресурс

http://wiki2.dovecot.org — документація з dovecot

http://yakim.org.ua/ — ну як же я міг не взяти матеріали з власного сайту.