Openldap and postgresql – Successs – by Suranga De Silva

PostgreSQL Installation

./configure

gmake

su

gmake install

adduser postgres

mkdir /usr/local/pgsql/data

chown postgres /usr/local/pgsql/data

su - postgres

/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data

/usr/local/pgsql/bin/postmaster -D /usr/local/pgsql/data >logfile 2>&1 &

/usr/local/pgsql/bin/createdb test

/usr/local/pgsql/bin/psql test

iODBC installation

iODBC can be found at OpenLink ODBC Drivers. Here is what I use to install libiodbc:

```
./configure --with-iodbc-inidir=/etc
make
make install
```

PostgreSQL ODBC Driver installation

The PostgreSQL ODBC driver can be found at psqlODBC - The PostgreSQL ODBC Driver. Here is what I use to install libpsqlodbc:

execute the following command and configure the postgresql iodbc drivers.

[root@suranga suranga]#export PATH=$PATH:/usr/local/pgsql/bin

```
        ./configure --with-iodbc --with-odbcinst=/etc
        make
        make install
```

OpenLDAP installation

OpenLDAP can be found at http://www.openldap.org/. You need to compile OpenLDAP with the --enable-sql option. Here is what I use to install OpenLDAP:

```
        ./configure --enable-sql --without-cyrus-sasl --disable-bdb --enable-crypt
        make depend
        make
        make install
```

[root@suranga suranga]# more /etc/odbc.ini

```
;

;  odbc.ini

;

[ODBC Data Sources]

PgSQL=PostgreSQL



[PgSQL]

Driver=/usr/lib/libodbcpsql.so

Description=Connection to LDAP/POSTGRESQL

Servername=localhost

Port=5432
```

```
Protocol=6.4

FetchBufferSize=99

Username=test

Password=test

Database=pg_ldap

ReadOnly=no

Debug=1

CommLog=1


[ODBC]

InstallDir=/usr/local/lib

[root@suranga suranga]#
```

[root@suranga suranga]# more /etc/odbcinst.ini

```
;

;  odbcinst.ini

;

[PostgreSQL]

Description=ODBC for PostgreSQL

Driver=/usr/lib/libodbcpsql.so


[ODBC]

Trace=1

Debug=1

Pooling=No
```

PostgreSQL

We just need to create the test database and the test user.You must su to user postgres or any other PostgreSQL superuser to perform the following task.

Creating the test database and test user

Just run the following command to create the test database:

```
        createdb pg_ldap
```

Just run the following command to create the user 'test' with password 'test':

```
        createuser --no-createdb --no-adduser --password test
```

and give the password 'test' at prompt.

Creating the SQL backend for LDAP

To have OpenLDAP working with a SQL backend you must create the database structure and fill some information in it. All you need is to change directory to openldap-2.1.12/servers/slapd/back-sql/rdbms_depend/pgsql/ and run the following command as PostgreSQL superuser:

```
        psql pg_ldap  < backsql_create.sql
```

These tables are used by OpenLDAP to maintain all links between objects. This is the LDAP meta-structure.

Creating the test database schema

We have now to create a schema with table representing our test LDAP objects. This can be done by using the rdbms_depend/testdb_*.sql files and running the following commands:

```
        psql -d pg_ldap < testdb_create.sql
```

These tables are used to create our test directory objects and their attributes.

Creating the metadata

This part generate all links between the SQL backend and the stored object for the test database. Theses metainformation are used to translate LDAP queries to SQL queries. This part also generate all SQL function used by the metadata definition to create links between the SQL backend and the stored object for the test database and to store all attributes value.

```
        psql -d pg_ldap < testdb_metadata.sql
```

Insert data for testing

This part insert some data into the test database. This can be done by saving the following SQL code into a file named testdb_data.sql and running the following command:

```
        psql -d pg_ldap < testdb_data.sql
```

Set grant on the database objects

To be able to run SQL queries onto the test database we must give the grant to user 'test'. This can be done by saving the following SQL code into a file named testdb_grant.sql and running the following command:

```
        psql -d pg_ldap -c "GRANT ALL ON ldap_attr_mappings,ldap_entries,
ldap_entry_objclasses,ldap_oc_mappings,ldap_referrals TO test;"
        psql -d pg_ldap -c "GRANT ALL ON ldap_attr_mappings_id_seq,
ldap_entries_id_seq,ldap_oc_mappings_id_seq TO test;"
        psql -d pg_ldap -c "GRANT ALL ON authors_docs,documents,institutes,
persons,phones TO test;"
        psql -d pg_ldap -c "GRANT ALL ON documents_id_seq,institutes_id_seq,

persons_id_seq,phones_id_seq TO test;"
```

Testing the ODBC installation

To test the ODBC installation simply run odbctest and give our DSN. Output must be as follow. If you don't have the SQL prompt you may have problem with ODBC and OpenLDAP SQL backend should not work as wanted.

```
        /usr/local/bin/odbctest

        iODBC Demonstration program
        This program shows an interactive SQL processor

        Enter ODBC connect string (? shows list): DSN=PgSQL
        Driver: 07.02.0005

        SQL>
```

```
[root@suranga suranga]# vim /etc/ldap.conf
host 127.0.0.1
base dc=example,dc=com
```

```
[root@suranga suranga]# slappasswd
```

copy the value you get as the output in my case it was

```
{SSHA}X59lpHb78L7JaNunf7J12FyAMJgeaOdD

[root@suranga suranga]# more /usr/local/etc/openldap/slapd.conf

#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
password-hash {CRYPT}
include         /usr/local/etc/openldap/schema/core.schema
include         /usr/local/etc/openldap/schema/cosine.schema
include         /usr/local/etc/openldap/schema/inetorgperson.schema

# Define global ACLs to disable default read access.
# Define global ACLs to disable default read access.
access to *
#       by self write
        by * write
access to * by dn="cn=root,o=sql,c=RU" write
#defaultaccess none

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral       ldap://root.openldap.org

pidfile         /usr/local/var/slapd.pid
argsfile        /usr/local/var/slapd.args

#######################################################################
# sql database definitions
#######################################################################

database        sql
suffix          "dc=example,dc=com"
rootdn          "cn=root,dc=example,dc=com"
rootpw          {SSHA}X59lpHb78L7JaNunf7J12FyAMJgeaOdD
dbname          PgSQL
dbuser          test
dbpasswd        test
insentry_query  "insert into ldap_entries (id,dn,oc_map_id,parent,keyval) values ((select max(id)+1
from ldap_entries),?,?,?,?)"
upper_func      "upper"
strcast_func    "text"
concat_pattern  "?||?"
has_ldapinfo_dn_ru      no

lastmod off
[root@suranga suranga]#
```

Now you can run the ldap server in debug mode as follows

```
[root@suranga suranga]#/usr/local/libexec/slapd -d 1

[root@suranga suranga]# ldapsearch -b "dc=example,dc=com" "(objectClass=*)"
# extended LDIF
#
# LDAPv3
# base <dc=example,dc=com> with scope subtree
# filter: (objectClass=*)
# requesting: ALL
#

# search result
search: 2
result: 0 Success

# numResponses: 1


[root@suranga suranga]# more /tmp/root.ldif dn: cn=User suranga,dc=example,dc=com
objectClass: inetOrgPerson
sn: Desilva suranga user
cn: User suranga

[root@suranga suranga]# ldapadd -D "cn=root,dc=example,dc=com" -w secret -f /tmp/root.ldif
adding new entry "cn=User suranga,dc=example,dc=com"
```

Now I am going to modify my data

```
[root@suranga suranga]# more /tmp/modify.ldif
dn: cn=User suranga,dc=example,dc=com
changetype: modify
replace: sn
sn: suranga user
cn: User suranga

dn: cn=User suranga,dc=example,dc=com
changetype: modify
add: telephoneNumber
telephoneNumber: 123-4567
telephoneNumber: 765-4321


[root@suranga suranga]# ldapmodify -D "cn=root,dc=example,dc=com" -w secret -f /tmp/modify.ldif
modifying entry "cn=User suranga,dc=example,dc=com"

modifying entry "cn=User suranga,dc=example,dc=com"
```

Here I am going to add contact.ldif to the database

```
[root@suranga suranga]# more /tmp/contact.ldif
dn: cn=Gayan Suranga, dc=example, dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Gayan Suranga
gn: Suranga
sn: Silva
mail: gayan@example.com
physicalDeliveryOfficeName: TechCERT
postalAddress: PO BOX 55555
l: UOM
ou: addressbook
st: Moratuwa
postalCode: 70555
telephoneNumber: 555-555-5555
facsimileTelephoneNumber: 555-555-5556
pager: 555-555-5557
mobile: 555-555-5558
homePhone: 555-555-5559
[root@suranga suranga]#
```

```
[root@suranga suranga]# ldapadd -D 'cn=root,dc=example,dc=com' -f /tmp/contact.ldif -W
Enter LDAP Password:
adding new entry "cn=Gayan Suranga, dc=example, dc=com"
```