

Instalacija PostgreSQL (8.3.6) na Debianu 5.0 (Lenny)

Postoje najmanje tri metode instalacije PostgreSQL DBMS-a, (ovdje konkretno verzije 8.3.6) na Debian GNU/Linux (ovdje konkretno verzije 5.0 = Lenny):

(1) Kompilacija i konfiguriranje odgovarajućeg source paketa (postgresql-8.3.6.tar.gz) skinutog sa PostgreSQL web sitea (www.postgresql.org)

(2) Instalacija potrebnih deb paketa u Lenny distribuciji nakon instalacije Lennyja

(3) Instalacija pomoću programa taskel koji se može pokrenuti tokom ili nakon instalacije Lennyja

Zanimljivo je da se u nekim slučajevima pri instalaciji potrebnih deb paketa u (2) i (3) (iz nekih nepoznatih razloga) ne uspijeva automatski kreirati DB-cluster i pokrenuti DB-server, odnosno podesiti automatsko pokretanje DB-servera pri dizanju sustava. U ovom tekstu popisano je sve što treba napraviti da bi DB-server radio normalno, ako se dogodi ovakav slučaj. Ovi su koraci ujedno potrebni i pri instalaciji pod (1).

Pri instalaciji PostgreSQL-a pod (2) najprije je potrebno pokrenuti naredbu

```
apt-get install postgresql
```

s tim da je korisno imati PostgreSQL dokumentaciju i neke dodatne opcije, što se instalira sa

```
apt-get install postgresql-contrib postgresql-doc
```

Pri instalaciji PostgreSQL-a pod (3) treba u tasksel formi izabrati opciju SQL database. Pritom se instaliraju slijedeći paketi (isto kao i pri instalaciji pod (2)):

- postgresql (8.3.6-1) - object-relational SQL database (supported version)
- postgresql-8.3 (8.3.6-1) - object-relational SQL database, version 8.3 server
- postgresql-client (8.3.6-1) - front-end programs for PostgreSQL (supported version)
- postgresql-client-8.3 (8.3.6-1) - front-end programs for PostgreSQL 8.3
- postgresql-client-common (94lenny1) - manager for multiple PostgreSQL client versions
- postgresql-common (94lenny1) - PostgreSQL database-cluster manager
- postgresql-contrib (8.3.6-1) - additional facilities for PostgreSQL (supported version)
- postgresql-contrib-8.3 (8.3.6-1) - additional facilities for PostgreSQL
- postgresql-doc (8.3.6-1) - documentation for the PostgreSQL database management system
- postgresql-doc-8.3 (8.3.6-1) - documentation for the PostgreSQL database management system
- libpq5 (8.3.6-1) - PostgreSQL C client library
- libosp-uuid15 (1.5.1-1.1+b1) - OSSP uuid ISO-C and C++ - shared library

Tokom ove instalacije automatski se kreira user postgres sa home direktorijem `/var/lib/postgresql/`.

Ovaj je direktorij prazan, pa je zgodno tu kreirati inicijalizacijske fileove `.profile`, `.bash_login`, `.bash_logout` i `.bashrc` (odnosno kopirati fileove iz home direktorija nekog drugog korisnika).

Pošto ne znamo password korisnika postgres zgodno ga je promijeniti naredbom

```
passwd postgres
```

koju pokrećemo kao korisnik root (`su - root`) nakon čega upisujemo novi password za korisnika postgres.

Zatim je zgodno je kreirati simbolički link na home direktorij korisnika postgres u direktoriju `/home/`

```
ln -s /var/lib/postgresql /home/postgres
```

što pokrećemo kao korisnik postgres (`su - postgres`). Kao isti korisnik kreiramo direktorij `data` u njegovom home direktoriju (`/home/postgres/ = /var/lib/postgresql/`) koji će služiti za podatke u PostgreSQL DB-clustera koji želimo kreirati (i sve ostale naredbe izvršit ćemo kako korisnik postgres)

```
mkdir /var/lib/postgresql/data/
```

Korisno je redefinirati environment varijablu `PATH` i definirati varijable `PGDATA` i `PGLOGFILE` u inicijalizacijskom fileu `.profile` za korisnika postgres (koji se nalazi u njegovom home direktoriju) - dodati

```
PATH=/usr/lib/postgresql/8.3/bin:$PATH
```

```
PGDATA=/var/lib/postgresql/data; export PGDATA
```

```
PGLOGFILE=/var/lib/postgresql/postgresql.log; export PGLOGFILE
```

Sad možemo kreirati DB-cluster (ako već nije kreiran) - otvorimo novi terminal, prijavimo se kao korisnik postgres i pokrenemo naredbu

```
initdb -D /var/lib/postgresql/data --pwprompt
```

Pri kreiranju DB-clustera treba upisati password za superusera (postgres). Ako je kreiranje prošlo uspješno sustav će na kraju ispisati poruku

Success. You can now start the database server using:

```
postgres -D /var/lib/postgresql/data
```

or

```
pg_ctl -D /var/lib/postgresql/data -l logfile start
```

Prije pokretanja baze treba namjestiti kernel parametre OS-a i konfiguracijske parametre baze (tj. DBMS-a). Defaultne vrijednosti kernel parametara za Debian GNU/Linux su uglavnom odgovarajuće, jedino je za veće baze potrebno povećati parametar **SHMMAX** (maksimalna veličina shared memory segmenta tj. prostora u RAM-u koji koriste procesi DBMS-a) čija je defaultna vrijednost 32MB. Ovu vrijednost smo povećali na 128MB tako da smo u file **/etc/sysctl.conf** dodali liniju

```
kernel.shmmax = 134217728
```

(za baze u produkciji preporuča se za vrijednost **SHMMAX** staviti polovicu ukupne veličine RAM-a).

Konfiguracijski parametri baze nalaze se u fileu **/var/lib/postgresql/data/postgresql.conf**.

Parametri koje smo promijenili su:

```
listen_addresses = * (definira adrese sa kojih je dozvoljeno spajanje na server - default localhost)
```

```
ssl = on (omogućava SSL konekcije - default off)
```

```
shared_buffers = 128MB (memorijski prostor za shared memory buffere - default 24MB)
```

```
work_mem = 2MB (memorijski prostor za pojedinačna sortiranja i hash tablice - default 1MB)
```

Konfiguracijski parametri baze mogu se vidjeti na **psql** promptu pomoću naredbe **show all**.

U fileu **/var/lib/postgresql/data/pg_hba.conf** navode se klijenti za PostgreSQL bazu i način njihovog spajanja. Zgodno je umjesto defaultne (**trust**) autentikacije staviti **ident** ili **MD5**.

Da bi se baza automatski podigla pri bootanju, odnosno spustila na regularan način pri shutdownu ili rebootu sustava, treba u direktorij **/etc/init.d** staviti odgovarajuću bash skriptu i u direktorijima **rcN.d** (N=0,...,6) kreirati simboličke linkove na tu skriptu prema potrebi. Bash skriptu pod nazivom **postgresql** navodimo u dodatku ovoga teksta, a potrebne linkove kreiramo naredbama

```
ln -s /etc/init.d/postgresql /etc/rc0.d/K20postgresql
```

```
ln -s /etc/init.d/postgresql /etc/rc6.d/K20postgresql
```

```
ln -s /etc/init.d/postgresql /etc/rc2.d/S20postgresql
```

(pretpostavljamo defaultni runlevel 2, inače se može definirati podizanje baze i za druge runlevele).

Ovim smo uglavnom završili konfiguriranje baze (finija podešavanja mogu se po potrebi provesti naknadno), pa možemo rebootati sustav ili podići bazu (kao korisnik postgres) naredbom

```
pg_ctl -l $PGLOGFILE start
```

Sad se napokon možemo spojiti na bazu tj. DB-cluster (kao korisnik postgres) naredbom

```
psql template1
```

i početi sa administriranjem. Za početak zgodno je kreirati korisnika s administratorskim ovlastima na nivou instance baze

```
CREATE USER oam WITH PASSWORD 'oam' CREATEDB CREATEUSER;
```

Zatim kao ovaj novi korisnik kreiramo tablespace za korisničku instancu baze, kao i samu tu instancu (u postgresql terminologiji bazu)

```
\c template1 oam
```

```
CREATE TABLESPACE exactbasdata LOCATION '/var/lib/postgresql/data/pg_tblspc';
```

```
CREATE DATABASE exactbas OWNER oam ENCODING 'UTF8' TEMPLATE template0 TABLESPACE exactbasdata;
```

Često je puta zgodno kreirati i shemu čije je ime isto kao i ime korisnika kojega smo kreirali - svi objekti koje kreira ovaj korisnik naći će se u toj shemi osim ako se eksplicitno navede da se kreiraju negdje drugdje

```
CREATE SCHEMA oam;
```

Nakon ovoga slijedi kreiranje "običnih korisnika" i grantanje ovlasti već prema potrebi. Ako baza sadrži bitne podatke još treba napisati skriptu za backup podataka i podesiti njeno redovno izvršavanje.

Dodatak - skripta postgresql

```
#!/bin/sh

DAEMON=/usr/lib/postgresql/8.3/bin/postgres
NAME=postgres

# Exit if the package is not installed
[ -x "$DAEMON" ] || exit 0

# Read configuration variable file if it is present
[ -r /etc/default/$NAME ] && . /etc/default/$NAME

#
# Function that starts the daemon/service
#
do_start()
{
    echo "PostgreSQL 8.3.6 Database on NEPTUN8"
    su - -c 'pg_ctl -D $PGDATA -l $PGLOGFILE start' postgres
}
#
# Function that stops the daemon/service
#
do_stop()
{
    echo "PostgreSQL 8.3.6 Database on NEPTUN8"
    su - -c 'pg_ctl -D $PGDATA stop' postgres
}
#
# Function that sends a SIGHUP to the daemon/service
#
do_reload()
{
    su - -c 'pg_ctl -D $PGDATA reload' postgres
}

case "$1" in
    start)
        do_start
        ;;
    stop)
        do_stop
        ;;
    reload|force-reload)
        do_reload
        ;;
    restart|force-reload)
        do_stop
        sleep 5
        do_start
        ;;
    *)
        echo "Usage: /etc/init.d/postgresql {start|stop|restart|force-reload}"
        exit 3
        ;;
esac
```