FreeBSDのパッケージでPostgreSQLをクイックセットアップ

2017/03/05
FreeBSDのpkgコマンドでお手軽にPostgreSQLをインストール、セットアップする portsでソースからコンパイルするより速いし、ちゃちゃっと準備するにはおすすめ。

今回は ezjail でJailをこしらえてその中にインストールします。~~Jailではなくても可能だけどできれば分離することをお勧めします。~~

PostgreSQLインストール前の準備

Jailにインストールする場合、事前に設定しておく内容。
既にJailインスタンス pg が作成され起動しているものとします。

Jailの通信許可と共有メモリ利用許可を出します。

FreeBSDのJailホストJailで設定する項目のメモを参照。

PostgreSQLインストール

Jailにインストールする場合はJailにログインしておきます。

PostgreSQLのパッケージ名確認

pkgコマンドの searchサブコマンドで postgres を検索します。

```
root@pg:~ # pkg search postgres
exim-postgresql-4.88           High performance MTA for Unix systems on the
Internet
fpc-postgres-3.0.0_1           Free Pascal interface to PostGreSQL
libgda4-postgresql-4.2.12_2    Provides PostgreSQL access for the libgda4
library
libgda5-postgresql-5.2.4       Provides postgresql access for the libgda5
library
luasql-postgres-2.3.3          Lua interface to PostgreSQL
mysqlphp2postgres-0.95         Convert MySQL calls in a PHP page into
PostgreSQL calls
nagios-check_postgres-2.22.0   Monitor various attributes of your PostgreSQL
database
nagios-check_postgres_replication-2010.12.16 Nagios plugin to check the lag
between a postgresql replica and master
opensmtpd-extras-table-postgresql-201606230001 PostgreSQL table support for
OpenSMTPD
```

```
p5-PostgreSQL-PLPerl-Call-1.006_1 Simple interface for calling SQL functions
from PostgreSQL PL/Perl
p5-PostgreSQL-PLPerl-Trace-1.001_1 Simple way to trace execution of Perl
statements in PL/Perl
p5-Test-postgresql-0.09_1      Perl extension of postgresql runner for tests
pgtcl-postgresql92-2.0.0_1     TCL extension for accessing a PostgreSQL
server (PGTCL-NG)
pgtcl-postgresql93-2.0.0_1     TCL extension for accessing a PostgreSQL
server (PGTCL-NG)
pgtcl-postgresql94-2.0.0_1     TCL extension for accessing a PostgreSQL
server (PGTCL-NG)
pgtcl-postgresql95-2.0.0_1     TCL extension for accessing a PostgreSQL
server (PGTCL-NG)
pgtcl-postgresql96-2.0.0_1     TCL extension for accessing a PostgreSQL
server (PGTCL-NG)
postgresql-jdbc-9.2.1004       The Java JDBC implementation for PostgreSQL
postgresql-libpgeasy-3.0.4_1   Easy-to-use C interface to PostgreSQL
postgresql-libpqxx-4.0.1_1     New C++ interface for PostgreSQL
postgresql-libpqxx3-3.1.1_1    New C++ interface for PostgreSQL
postgresql-odbc-09.05.0400     PostgreSQL ODBC driver
postgresql-plproxy-2.5_1       PL/Proxy - database partitioning system
postgresql-relay-1.3.2_1       Multiplex multiple PostgreSQL databases to
one relay
postgresql-repmgr-3.1.5        PostgreSQL replication manager
postgresql-repmgr2-2.0.3_1     PostgreSQL replication manager
postgresql92-client-9.2.19_1   PostgreSQL database (client)
postgresql92-contrib-9.2.19_1  The contrib utilities from the PostgreSQL
distribution
postgresql92-docs-9.2.19_1     The PostgreSQL documentation set
postgresql92-plperl-9.2.19_1   Write SQL functions for PostgreSQL using
Perl5
postgresql92-plpython-9.2.19_1 Module for using Python to write SQL
functions
postgresql92-pltcl-9.2.19_1    Module for using Tcl to write SQL functions
postgresql92-server-9.2.19_1   PostgreSQL is the most advanced open-source
database available anywhere
postgresql93-client-9.3.15_1   PostgreSQL database (client)
postgresql93-contrib-9.3.15_1  The contrib utilities from the PostgreSQL
distribution
postgresql93-docs-9.3.15_1     The PostgreSQL documentation set
postgresql93-plperl-9.3.15_1   Write SQL functions for PostgreSQL using
Perl5
postgresql93-plpython-9.3.15_1 Module for using Python to write SQL
functions
postgresql93-pltcl-9.3.15_1    Module for using Tcl to write SQL functions
postgresql93-plv8js-1.4.8      PL/v8js procedural language for PostgreSQL
database
postgresql93-server-9.3.15_1   PostgreSQL is the most advanced open-source
database available anywhere
postgresql94-client-9.4.10_1   PostgreSQL database (client)
postgresql94-contrib-9.4.10_1  The contrib utilities from the PostgreSQL
```

```
distribution
postgresql94-docs-9.4.10_1     The PostgreSQL documentation set
postgresql94-plperl-9.4.10_1   Write SQL functions for PostgreSQL using
Perl5
postgresql94-plpython-9.4.10_1 Module for using Python to write SQL
functions
postgresql94-pltcl-9.4.10_1    Module for using Tcl to write SQL functions
postgresql94-server-9.4.10_1   PostgreSQL is the most advanced open-source
database available anywhere
postgresql95-client-9.5.5_1    PostgreSQL database (client)
postgresql95-contrib-9.5.5     The contrib utilities from the PostgreSQL
distribution
postgresql95-docs-9.5.5_1      The PostgreSQL documentation set
postgresql95-plperl-9.5.5_1    Write SQL functions for PostgreSQL using
Perl5
postgresql95-plpython-9.5.5_1  Module for using Python to write SQL
functions
postgresql95-pltcl-9.5.5_1     Module for using Tcl to write SQL functions
postgresql95-server-9.5.5_1    PostgreSQL is the most advanced open-source
database available anywhere
postgresql96-client-9.6.1      PostgreSQL database (client)
postgresql96-contrib-9.6.1     The contrib utilities from the PostgreSQL
distribution
postgresql96-docs-9.6.1_1      The PostgreSQL documentation set
postgresql96-plperl-9.6.1_1    Write SQL functions for PostgreSQL using
Perl5
postgresql96-plpython-9.6.1_1  Module for using Python to write SQL
functions
postgresql96-pltcl-9.6.1_1     Module for using Tcl to write SQL functions
postgresql96-server-9.6.1_1    PostgreSQL is the most advanced open-source
database available anywhere
proftpd-mod_sql_postgres-1.3.5b PostgreSQL ProFTPD module
py35-postgresql-1.1.0_3        Python 3 compatible PostgreSQL database
driver and tools
rubygem-dm-postgres-adapter-1.2.0 PostgreSQL Adapter for DataMapper
rubygem-do_postgres-0.10.13    Ruby DataObjects driver for PostgreSQL
rubygem-postgres_ext-3.0.0     PostgreSQL data types extension for
ActiveRecord
root@pg:root@pg:~ #
```

指定パッケージをインストール

古いバージョンを使う理由がないならこのリストで最新の postgresql96-server-9.6.1_1 をインストールしましょう。 依存関係のあるパッケージも同時にインストールされます。

```
root@pg:~ # pkg install postgresql96-server-9.6.1_1
Updating FreeBSD repository catalogue...
FreeBSD repository is up-to-date.
All repositories are up-to-date.
The following 8 package(s) will be affected (of 0 checked):
```

```
New packages to be INSTALLED:
        postgresql96-server: 9.6.1_1
        libxml2: 2.9.4
        icu: 58.2,1
        gettext-runtime: 0.19.8.1_1
        indexinfo: 0.2.6
        postgresql96-client: 9.6.1
        perl5: 5.24.1.r4_1
        readline: 6.3.8

Number of packages to be installed: 8

The process will require 130 MiB more space.
30 MiB to be downloaded.

Proceed with this action? [y/N]: y
[pg] Fetching postgresql96-server-9.6.1_1.txz: 100%    4 MiB   1.3MB/s
00:03
[pg] Fetching libxml2-2.9.4.txz: 100%  802 KiB 821.1kB/s    00:01
[pg] Fetching icu-58.2,1.txz: 100%    9 MiB   2.4MB/s    00:04
[pg] Fetching gettext-runtime-0.19.8.1_1.txz: 100%  148 KiB 151.1kB/s
00:01
[pg] Fetching indexinfo-0.2.6.txz: 100%    5 KiB   5.3kB/s    00:01
[pg] Fetching postgresql96-client-9.6.1.txz: 100%    2 MiB   1.2MB/s
00:02
[pg] Fetching perl5-5.24.1.r4_1.txz: 100%   13 MiB   2.3MB/s    00:06
[pg] Fetching readline-6.3.8.txz: 100%  318 KiB 325.7kB/s    00:01
Checking integrity... done (0 conflicting)
[pg] [1/8] Installing indexinfo-0.2.6...
[pg] [1/8] Extracting indexinfo-0.2.6: 100%
[pg] [2/8] Installing gettext-runtime-0.19.8.1_1...
[pg] [2/8] Extracting gettext-runtime-0.19.8.1_1: 100%
[pg] [3/8] Installing perl5-5.24.1.r4_1...
[pg] [3/8] Extracting perl5-5.24.1.r4_1: 100%
[pg] [4/8] Installing readline-6.3.8...
[pg] [4/8] Extracting readline-6.3.8: 100%
[pg] [5/8] Installing libxml2-2.9.4...
[pg] [5/8] Extracting libxml2-2.9.4: 100%
[pg] [6/8] Installing icu-58.2,1...
[pg] [6/8] Extracting icu-58.2,1: 100%
[pg] [7/8] Installing postgresql96-client-9.6.1...
[pg] [7/8] Extracting postgresql96-client-9.6.1: 100%
[pg] [8/8] Installing postgresql96-server-9.6.1_1...
===> Creating groups.
Creating group 'postgres' with gid '770'.
===> Creating users
Creating user 'postgres' with uid '770'.

  =========== BACKUP YOUR DATA! =============
  As always, backup your data before
  upgrading. If the upgrade leads to a higher
  minor revision (e.g. 8.3.x -> 8.4), a dump
  and restore of all databases is
  required. This is *NOT* done by the port!
  ===========================================
[pg] [8/8] Extracting postgresql96-server-9.6.1_1: 100%
Message from perl5-5.24.1.r4_1:
The /usr/bin/perl symlink has been removed starting with Perl 5.20.
For shebangs, you should either use:

#!/usr/local/bin/perl

or

#!/usr/bin/env perl

The first one will only work if you have a /usr/local/bin/perl,
the second will work as long as perl is in PATH.
Message from postgresql96-client-9.6.1:
The PostgreSQL port has a collection of "side orders":

postgresql-docs
  For all of the html documentation

p5-Pg
  A perl5 API for client access to PostgreSQL databases.

postgresql-tcltk
  If you want tcl/tk client support.

postgresql-jdbc
  For Java JDBC support.

postgresql-odbc
  For client access from unix applications using ODBC as access
  method. Not needed to access unix PostgreSQL servers from Win32
  using ODBC. See below.

ruby-postgres, py-PyGreSQL
  For client access to PostgreSQL databases using the ruby & python
  languages.

postgresql-plperl, postgresql-pltcl & postgresql-plruby
  For using perl5, tcl & ruby as procedural languages.

postgresql-contrib
  Lots of contributed utilities, postgresql functions and
  datatypes. There you find pg_standby, pgcrypto and many other cool
  things.

etc...
Message from postgresql96-server-9.6.1_1:
```

```
For procedural languages and postgresql functions, please note that
you might have to update them when updating the server.

If you have many tables and many clients running, consider raising
kern.maxfiles using sysctl(8), or reconfigure your kernel
appropriately.

The port is set up to use autovacuum for new databases, but you might
also want to vacuum and perhaps backup your database regularly. There
is a periodic script, /usr/local/etc/periodic/daily/502.pgsql, that
you may find useful. You can use it to backup and perform vacuum on all
databases nightly. Per default, it performs `vacuum analyze'. See the
script for instructions. For autovacuum settings, please review
~pgsql/data/postgresql.conf.

If you plan to access your PostgreSQL server using ODBC, please
consider running the SQL script /usr/local/share/postgresql/odbc.sql
to get the functions required for ODBC compliance.

Please note that if you use the rc script,
/usr/local/etc/rc.d/postgresql, to initialize the database, unicode
(UTF-8) will be used to store character data by default.  Set
postgresql_initdb_flags or use login.conf settings described below to
alter this behaviour. See the start rc script for more info.

To set limits, environment stuff like locale and collation and other
things, you can set up a class in /etc/login.conf before initializing
the database. Add something similar to this to /etc/login.conf:
---
postgres:\
        :lang=en_US.UTF-8:\
        :setenv=LC_COLLATE=C:\
        :tc=default:
---
and run `cap_mkdb /etc/login.conf'.
Then add 'postgresql_class="postgres"' to /etc/rc.conf.

======================================================================

To initialize the database, run

  /usr/local/etc/rc.d/postgresql initdb

You can then start PostgreSQL by running:

  /usr/local/etc/rc.d/postgresql start

For postmaster settings, see ~pgsql/data/postgresql.conf

NB. FreeBSD's PostgreSQL port logs to syslog by default
    See ~pgsql/data/postgresql.conf for more info
```

```
======================================================================

To run PostgreSQL at startup, add
'postgresql_enable="YES"' to /etc/rc.conf
root@pg:~ #
```

ユーザ postgres が作成されています□PostgreSQLの設定ファイルの修正等はこのユーザにログインして実施します。

/etc/rc.confに追記

指示に従い /etc/rc.conf に追記します。

/etc/rc.conf

```
postgresql_enable="YES"
```

データベース初期化

PostgreSQLのインストール後の説明は一部古い部分があるので適宜置き換えして実行してください。もちろん、説明のまま実行しても問題はありませんが。

以下は説明中にある

```
/usr/local/etc/rc.d/postgresql initdb
```

に対応するものです。

```
root@pg:~ # service postgresql initdb
The files belonging to this database system will be owned by user
"postgres".
This user must also own the server process.

The database cluster will be initialized with locale "C".
The default text search configuration will be set to "english".

Data page checksums are disabled.

creating directory /var/db/postgres/data96 ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok
```

```
WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    /usr/local/bin/pg_ctl -D /var/db/postgres/data96 -l logfile start

root@pg:~ #
```

/var/db/postgres/data96 にデータベースが作成されたようです。

なお、共有メモリの利用許可がないとこんなエラーが出ます。

```
root@pg:/usr/local/etc/rc.d # service postgresql initdb
The files belonging to this database system will be owned by user
"postgres".
This user must also own the server process.

The database cluster will be initialized with locale "C".
The default text search configuration will be set to "english".

Data page checksums are disabled.

creating directory /var/db/postgres/data96 ... ok
creating subdirectories ... ok
selecting default max_connections ... 10
selecting default shared_buffers ... 400kB
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
running bootstrap script ... FATAL:  could not create shared memory segment:
Function not implemented
DETAIL:  Failed system call was shmget(key=1, size=48, 03600).
child process exited with exit code 1
initdb: removing data directory "/var/db/postgres/data96"
root@pg:/usr/local/etc/rc.d #
```

pg_hba.conf の編集

デフォルトでは外部からの接続ができないので pg_hba.conf を編集して接続許可を出します。

```
root@pg:/var/db/postgres/data96 # ll
total 120
-rw-------  1 postgres  postgres     4 Mar  4 19:12 PG_VERSION
drwx------  5 postgres  postgres   512 Mar  4 19:12 base/
drwx------  2 postgres  postgres  1024 Mar  4 19:31 global/
drwx------  2 postgres  postgres   512 Mar  4 19:12 pg_clog/
drwx------  2 postgres  postgres   512 Mar  4 19:12 pg_commit_ts/
drwx------  2 postgres  postgres   512 Mar  4 19:12 pg_dynshmem/
-rw-------  1 postgres  postgres  4608 Mar  4 19:29 pg_hba.conf
```

```
-rw-------  1 postgres  postgres   1636 Mar  4 19:12 pg_ident.conf
drwx------  4 postgres  postgres    512 Mar  4 19:12 pg_logical/
drwx------  4 postgres  postgres    512 Mar  4 19:12 pg_multixact/
drwx------  2 postgres  postgres    512 Mar  4 19:31 pg_notify/
drwx------  2 postgres  postgres    512 Mar  4 19:12 pg_replslot/
drwx------  2 postgres  postgres    512 Mar  4 19:12 pg_serial/
drwx------  2 postgres  postgres    512 Mar  4 19:12 pg_snapshots/
drwx------  2 postgres  postgres    512 Mar  4 19:31 pg_stat/
drwx------  2 postgres  postgres    512 Mar  5 07:11 pg_stat_tmp/
drwx------  2 postgres  postgres    512 Mar  4 19:12 pg_subtrans/
drwx------  2 postgres  postgres    512 Mar  4 19:12 pg_tblspc/
drwx------  2 postgres  postgres    512 Mar  4 19:12 pg_twophase/
drwx------  3 postgres  postgres    512 Mar  4 19:12 pg_xlog/
-rw-------  1 postgres  postgres     88 Mar  4 19:12 postgresql.auto.conf
-rw-------  1 postgres  postgres  22330 Mar  4 19:12 postgresql.conf
-rw-------  1 postgres  postgres     55 Mar  4 19:31 postmaster.opts
-rw-------  1 postgres  postgres     81 Mar  4 19:31 postmaster.pid
root@pg:/var/db/postgres/data96 # su postgres
$ vi pg_hba.conf
```

以下の2行を追加してください。

/var/db/postgres/data96/pg_hba.conf

```
host    all             all             192.168.1.30/32         trust
host    all             all             192.168.1.0/24          trust
```

最初の行はJailに割り当てたIPアドレスそのもの。
次の行はPostgreSQLに接続可能なIPアドレス範囲です。各環境に合わせて変更してください。

PostgreSQLの起動

以下は説明中にある

```
/usr/local/etc/rc.d/postgresql start
```

に対応するものです。

```
$ ^D
root@pg:~ # service postgresql start
LOG:  could not create IPv6 socket: Protocol not supported
LOG:  ending log output to stderr
HINT:  Future log output will go to log destination "syslog".
root@pg:~ #
```

PostgreSQLにユーザとデータベースを定義する

接続に使うユーザ kamo と、データベース kamonohashi を作成してみます。

```
root@pg:~ # su postgres
$ createuser --help
createuser creates a new PostgreSQL role.

Usage:
  createuser [OPTION]... [ROLENAME]

Options:
  -c, --connection-limit=N  connection limit for role (default: no limit)
  -d, --createdb            role can create new databases
  -D, --no-createdb         role cannot create databases (default)
  -e, --echo                show the commands being sent to the server
  -E, --encrypted           encrypt stored password
  -g, --role=ROLE           new role will be a member of this role
  -i, --inherit             role inherits privileges of roles it is a
                            member of (default)
  -I, --no-inherit          role does not inherit privileges
  -l, --login               role can login (default)
  -L, --no-login            role cannot login
  -N, --unencrypted         do not encrypt stored password
  -P, --pwprompt            assign a password to new role
  -r, --createrole          role can create new roles
  -R, --no-createrole       role cannot create roles (default)
  -s, --superuser           role will be superuser
  -S, --no-superuser        role will not be superuser (default)
  -V, --version             output version information, then exit
  --interactive             prompt for missing role name and attributes
rather
                            than using defaults
  --replication             role can initiate replication
  --no-replication          role cannot initiate replication
  -?, --help                show this help, then exit

Connection options:
  -h, --host=HOSTNAME       database server host or socket directory
  -p, --port=PORT           database server port
  -U, --username=USERNAME   user name to connect as (not the one to create)
  -w, --no-password         never prompt for password
  -W, --password            force password prompt

Report bugs to <pgsql-bugs@postgresql.org>.
$ createuser kamo
$ createdb --help
createdb creates a PostgreSQL database.

Usage:
  createdb [OPTION]... [DBNAME] [DESCRIPTION]

Options:
  -D, --tablespace=TABLESPACE  default tablespace for the database
  -e, --echo                   show the commands being sent to the server
  -E, --encoding=ENCODING      encoding for the database
  -l, --locale=LOCALE          locale settings for the database
      --lc-collate=LOCALE      LC_COLLATE setting for the database
      --lc-ctype=LOCALE        LC_CTYPE setting for the database
  -O, --owner=OWNER            database user to own the new database
  -T, --template=TEMPLATE      template database to copy
  -V, --version                output version information, then exit
  -?, --help                   show this help, then exit

Connection options:
  -h, --host=HOSTNAME          database server host or socket directory
  -p, --port=PORT              database server port
  -U, --username=USERNAME      user name to connect as
  -w, --no-password            never prompt for password
  -W, --password               force password prompt
  --maintenance-db=DBNAME      alternate maintenance database

By default, a database with the same name as the current user is created.

Report bugs to <pgsql-bugs@postgresql.org>.
$ createdb -E utf-8 -O kamo kamonohashi
$
```

データベースのユーザ kamo にはパスワードを設定していません。
必要ならpsqlコマンドで接続して \password コマンドで設定するか createuser コマンドに -P オプションを付けてユーザ作成時にパスワードも指定してください。
データベース kamonohashi のオーナーを kamo にしてあります。

psqlコマンドでデータベース kamonohashi へ接続してみます。

```
$ psql -n kamonohashi kamo
psql (9.6.1)
Type "help" for help.

kamonohashi=> \d
No relations found.
kamonohashi=> create table demo(
kamonohashi(>   id numeric(10,1),
kamonohashi(>   str varchar(10)
kamonohashi(> );
CREATE TABLE
kamonohashi=> \d
       List of relations
 Schema | Name | Type  | Owner
--------+------+-------+-------
 public | demo | table | kamo
```

```
(1 row)

kamonohashi=> insert into demo values(1,'ABCD');
INSERT 0 1
kamonohashi=> select * from demo;
 id  | str
-----+------
 1.0 | ABCD
(1 row)

kamonohashi=> insert into demo values(1,'あいうえ');
ERROR:  invalid byte sequence for encoding "UTF8": 0xa4
kamonohashi=> \encoding euc-jp
kamonohashi=> insert into demo values(1,'あいうえ');
INSERT 0 1
kamonohashi=> select * from demo;
 id  |   str
-----+----------
 1.0 | ABCD
 1.0 | あいうえ
(2 rows)

kamonohashi=>
```

端末がeuc-jpなので□\encoding コマンドでエンコーディングを指定しています。

技術資料, FreeBSD, Jail, PostgreSQL, database

From:
https://wiki.hgotoh.jp/ - 努力したWiki

Permanent link:
https://wiki.hgotoh.jp/documents/quick/quick-0018

Last update: **2024/11/01 16:25**