

Установка PostgreSQL в CentOS 7



alexkn.ru/node/42

PostgreSQL, лично по моему мнению, не так популярна, как MySQL/MariaDB, но есть мнение, что более надёжна и масштабируема. Тем не менее, есть ПО, которое требует именно PostgreSQL, поэтому, как обычно, администратору вбить не приходится =) Давай-те ставить.

Самая новая версия из доступных в моём "стандартном" наборе репозиториях - 9.2.7 из ере! Но... "маловато, говорю! Маловато будет!" (с)

Идём на <http://yum.postgresql.org/repopackages.php>, где, конечно же, сможем найти самую последнюю версию.

На момент написания статьи это 9.4.0. Устанавливаем репу:

```
yum install http://yum.postgresql.org/9.4/redhat/rhel-7-x86_64/pgdg-centos94-9.4-1.noarch.rpm
```

Теперь устанавливаем сам PostgreSQL. Обратите внимания, что в названии пакета фигурируют цифры версии, которые, будут меняться при мажорных обновлениях (9.5, 9.6 и т.д.):

```
yum install postgresql94-server
```

Сразу включим его в автозагрузку

```
systemctl enable postgresql-9.4
```

Далее мы должны инициализировать кластер БД. Для тех, кому важно понимать - прошу под спойлер, нашёл очень хорошее описание на сайте postgresql-lab.blogspot.ru.

Spoiler: Highlight to view

Кластер БД - это набор баз данных, который управляется одним экземпляром запущенного сервера БД. После инициализации, кластер БД будет содержать БД с именем postgres, которая является БД по умолчанию, которую используют утилиты, пользователи и сторонние приложения. Сам сервер БД не нуждается в этой БД, но многие внешние приложения подразумевают, что она существует. Другая БД, создаваемая на этапе инициализации, называется template1. Как и следует из её имени, она будет использоваться как шаблон для создаваемых БД; она не должна использоваться для работы.

Сама команда:

```
/usr/pgsql-9.4/bin/postgresql94-setup  
initdb
```

После этого запускаем СУБД:

```
systemctl start  
postgresql-9.4
```

Теперь нам необходимо включить авторизацию и создать пользователей, так как по умолчанию подключение локально разрешено для любого авторизованного в системе пользователя. Для этого заходим в систему под пользователем postgres и запускаем консоль управления СУБД:

```
su postgres  
psql
```

Пользователь "postgres" является root'ом в СУБД. Поэтому начнём с того, что создадим для него пароль:

```
ALTER ROLE postgres WITH PASSWORD  
'your_super_password';
```

Теперь возвращаемся в консоль и обратно в root'a, дважды нажав сочетание клавиш "ctrl+d".

Пора внести изменения в схему авторизации СУБД. Для этого нам потребуется отредактировать файл, расположенный по адресу `/var/lib/pgsql/9.4/data/pg_hba.conf`. Как обычно любым текстовым редактором:

```
nano /var/lib/pgsql/9.4/data/pg_hba.conf
```

В конце файла видим подобные строки:

```
local    all             all  
peer  
host     all             all          127.0.0.1/32  
ident  
host     all             all          ::1/128  
ident
```

Это как раз и есть разрешение с localhost заходить без авторизации. Меняем это на
Приводим всё к виду:

```
local    all             all  
md5  
host     all             all          127.0.0.1/32  
md5
```

Да, подключение по IPv6 мы убрали. Считаете, что нужно? Оставляйте.

Первая строка разрешает подключение к СУБД для всех локальных пользователей, но при этом требует с них ввод пароля. Вторая делает практически то же самое, но мы эту строку оставим для того, чтобы подключаться к СУБД могли даже те пользователи, у кого нет системной учетной записи, а есть только роль в СУБД.

Ещё один момент. Если Вам нужно удаленное подключение, думаю, понятно, что необходимо действовать по аналогии и добавить строку:

```
host     all             all          123.456.678.91024  
md5
```

Дополнительно = необходимо будет указать СУБД прослушивать требуемые адреса. Делается это в файле конфигурации `postgresql.conf`. Находим строку:

```
#listen_addresses =  
'localhost'
```

и приводим к виду:

```
listen_addresses =  
'*'
```

Это заставит СУБД слушать все сетевые интерфейсы в поисках входящих подключений.

Теперь перезапускаем службу:

```
systemctl restart  
postgresql-9.4
```

Пробуем подключиться от имени пользователя postgres:

```
psql -U postgres
```

Кстати, для удобства, чтобы не вводить пароль, но и не светить его в консоли, можно в домашней директории создать файл (он будет с права на чтение только для Вас), в котором записать пароль к СУБД:

```
nano ~/.pgpass
```

И добавляем в него имя пользователя и пароль вида:

```
localhost:*:*:postgres:your_super_password
```

Теперь выставим права:

```
chmod 600 /root/.pgpass
```

Заходить всё равно придётся с указанием имени пользователя "postgres", но пароль больше вводить не нужно.

Если Вы пытаетесь зайти под другим пользователем, необходимо так же указывать базу данных, к которой Вы хотите подключиться, иначе PostgreSQL попытается подключиться к БД с тем же названием, что и имя пользователя, не найдет ее и выдаст сообщение об ошибке.

Дополнение №1: Не верная кодировка СУБД

Spoiler: Highlight to view

Столкнулся с тем, что СУБД ставится с кодировкой Encoding="SQL_ASCII", Collate="C", Ctype="C".

Чтобы это исправить, убейте СУБД (не забыв про бекап БД, если они уже созданы и нужны) и поставьте пакет заново:

```
yum remove postgresql94-server
rm -rf /var/lib/pgsql/
yum install postgresql94-
server
```

Далее нужно зайти под пользователя postgres

```
su -
postgres
```

И выполнить инициализацию указав соответствующие кодировки:

```
/usr/pgsql-9.4/bin/initdb --encoding UTF-8 --lc-collate='ru_RU.UTF-8' --lc-
ctype='ru_RU.UTF-8' -D /var/lib/pgsql/9.4/data/
```

Теперь можно запускать СУБД и добавлять её в автозагрузку.

Теперь полезные команды в консоли управления СУБД:

1. Создать пользователя с разрешением на вход в СУБД

```
CREATE ROLE username WITH PASSWORD 'user_password'
LOGIN;
```

2. Изменить пароль пользователя

```
\password username;
```

3. Создание новой БД с присвоением владельца

```
CREATE DATABASE databasename WITH OWNER
username;
```

4. Сменить владельца БД

```
ALTER DATABASE name OWNER TO
new_username;
```

5. Дать права доступа на БД для пользователя

```
GRANT ALL ON DATABASE databasename TO
username;
```

6. Переименовать БД

```
ALTER DATABASE name RENAME TO new_databasename
;
```

7. Посмотреть список БД

`\list` или просто `\l`

8. Подключиться к БД

`\c databasename`
и вывести список таблиц
`\dt`

9. Посмотреть список запросов к БД

```
select * from pg_stat_activity where  
datname='YourDatabase';
```

10. Удалить процесс по pid

```
select pg_terminate_backend(pid) from pg_stat_activity where  
datname='YourDatabase';
```

11. Сделать дамп БД

```
pg_dump -U postgres database >  
dump_file.sql
```

12. Залить дамп БД

```
psql -U postgres database <  
dump_file.sql
```

13. Соответственно работает связка

```
pg_dump -U postgres database | psql -U postgres  
database2
```

14. Сбросить все соединения к базе, удалить базу, создать базу

```
select pg_terminate_backend(pid) from pg_stat_activity where datname =  
'db_name'; drop database db_name; create database db_name;
```