



ЭНЕРГЕТИКА
МИКРОЭЛЕКТРОНИКА
АВТОМАТИКА

РАСПРЕДЕЛЕННАЯ СИСТЕМА ДИСПЕТЧЕРСКОГО УПРАВЛЕНИЯ

УСТАНОВКА POSTGRES PRO
STANDARD 11

Руководство администратора

РАСПДУ



Содержание

1 Общие сведения	2
1.1 Необходимые условия	2
2 Установка Postgres Pro	3
2.1 Настройка параметров ядра	3
2.2 Подключение репозитория с дистрибутивом	3
2.3 Установка пакетов	3
2.4 Задание пароля пользователю	4
2.5 Настройка переменных окружения.....	4
2.6 Запуск инициализации «кластера»	4
2.7 Постинсталляционные действия	4
2.7.1 Логирование	5
2.7.2 Настройка подключения клиентов.....	5
2.7.3 Настройки для резервного копирования и репликации	6
2.7.4 Прочие настройки	6
3 Запуск Postgres Pro	8
3.1 Проверка подключения к серверу Postgres	8
3.2 Проверка работы серверу Postgres.....	8
3.3 Останов сервера Postgres.....	9
3.4 Настройка клиента.....	9
3.5 Среда разработки.....	9

1 Общие сведения

В данном руководстве описываются действия, необходимые для установки и настройки СУБД Postgres Pro Standard 11.

1.1 Необходимые условия

Перед переходом к действиям, описанным ниже в руководстве, должны быть выполнены следующие необходимые условия:

- ✓ сервер, предназначенный для установки СУБД, должен удовлетворять минимальным или рекомендованным требованиям к аппаратному обеспечению:
 - минимальные требования к аппаратному обеспечению:
 - процессор 2 ядра x64 1.6 ГГц;
 - оперативная память 2 Гб;
 - дисковое пространство 30Гб;
 - рекомендованные требования к аппаратному обеспечению:
 - процессор 4 ядра x64 2+ ГГц;
 - оперативная память 16+ Гб;
 - дисковое пространство 50+ Гб.
- ✓ на сервере, предназначенном для установки СУБД, должна быть установлена ОС SLES12 SP1 или старше, версия x64. Проверить версию ОС можно следующей командой:

```
cat /etc/issue
```

Должно отобразиться сообщение следующего вида:

```
Welcome to SUSE Linux Enterprise Server 12 SP1 (x86_64) - Kernel \r
(\l).
```

2 Установка Postgres Pro

ВНИМАНИЕ!

Все команды ниже приведены для ПО следующих версий:

- ✓ SLES 12 SP3 x64
- ✓ Postgres Pro Standard 11

2.1 Настройка параметров ядра

Значения параметров ядра в Linux по умолчанию имеют вполне подходящие значения, так что их обычно менять не нужно. Единственно, что может потребоваться настроить, – это размер разделяемой памяти.

Параметры размера разделяемой памяти можно изменить, воспользовавшись командой `sysctl`. Например, так можно выделить 16 ГБ для разделяемой памяти:

```
$ sysctl -w kernel.shmmax=17179869184
$ sysctl -w kernel.shmall=4194304
```

Примечание. Подробнее о настройках ресурсов ядра: <https://postgrespro.ru/docs/postgrespro/10/kernel-resources#LINUX-MEMORY-OVERCOMMIT>.

2.2 Подключение репозитория с дистрибутивом

Необходимо подключить репозиторий с дистрибутивом Postgres Pro Standard 11:

```
# sudo rpm --import http://repo.postgrespro.ru/pgpro-11/keys/SUSE-GPG-KEY-
POSTGRESPRO
# sudo zypper addrepo http://repo.postgrespro.ru/pgpro-11/sles/12.1 pgpro
Adding repository 'pgpro'
.....
..... [done]
Repository 'pgpro' successfully added

URI           : http://repo.postgrespro.ru/pgpro-11/sles/12.1
Enabled       : Yes
GPG Check     : Yes
Autorefresh   : No
Priority      : 99 (default priority)

Repository priorities are without effect. All enabled repositories share the
same priority.
```

2.3 Установка пакетов

Необходимо установить пакет `postgrespro-std-11`:

```
# sudo zypper install postgrespro-std-11
Building repository 'pgpro' cache
.....
..... [done]
Loading repository data...
Reading installed packages...
Resolving package dependencies...
...
Updating /etc/manpath.config
Initializing database ... OK
Created symlink from /etc/systemd/system/multi-user.target.wants/postgrespro-std-11.service to /usr/lib/systemd/system/postgrespro-std-11.service.
```

Установить пакет `postgrespro-std-11-contrib` для работы функции, генерирующей GUID:

```
# sudo zypper install postgrespro-std-11-contrib
```

Примечание. Список всех пакетов: <https://postgrespro.ru/docs/postgrespro/11/binary-installation-on-linux>.

2.4 Задание пароля пользователю

Задать пароль пользователю `postgres`:

```
# passwd postgres
```

2.5 Настройка переменных окружения

Необходимо задать значение переменной окружения `PGDATA`, а также дополнить `PATH` путем к исполняемым модулям. Для этого нужно создать файл (например, `postgres.sh`) в каталоге `profile.d` следующего содержания:

```
export PGDATA=/var/lib/pgpro/std-11/data
export PATH=/opt/pgpro/std-11/bin:$PATH
```

2.6 Запуск инициализации «кластера»

Выполнить `initdb` из-под пользователя `postgres`, с указанием каталога, который был создан по умолчанию при установке.

Примечание. Если располагать файлы в другом месте, необходимо сделать владельцем каталога `PGDATA` пользователя `postgres`.

```
# initdb -D /path/to/PGDATA -encoding UTF8 --locale=ru_RU.UTF-8
```

2.7 Постинсталляционные действия

Выполнить базовые настройки сервера Postgres, приведенные ниже.

2.7.1 Логирование

В файле `$PGDATA/postgresql.conf` в секции «LOGGING AND REPORTING» необходимо настроить параметры логирования. По умолчанию сообщения выдаются в `stderr`, `logging_collector` может перехватывать эти сообщения и складывать в файл. Настроек логирования много, что позволяет достаточно гибко настроить что, как и куда писать.

Минимальные настройки:

```
log_destination = stderr
logging_collector = on
log_directory = 'log' #абсолютный путь к каталогу или относительный (от
$PGDATA)
```

2.7.2 Настройка подключения клиентов

Секция «CONNECTIONS AND AUTHENTICATION» файла «`postgres.conf`» представлена ниже:

```
listen_addresses = '*'
port = 5432
max_connections = 100
```

В параметре `listen_addresses` может быть указан список адресов, `localhost` или «*» для всех.

Параметр `max_connections` определяет количество одновременных подключений к серверу `postgres`.

В файле `pg_hba.conf` необходимо сделать запись для соответствующей подсети:

```
# TYPE      DATABASE        USER            ADDRESS          METHOD
# "local" is for Unix domain socket connections only
local      all             all             peer
# IPv4 local connections:
host       all             all             127.0.0.1/32    md5
host       all             all             192.168.50.0/24 md5
...
```

- ✓ поле `ADDRESS` – указывает адрес (или адреса) клиентской машины, которой соответствует данная запись. Это поле может содержать или имя компьютера, или диапазон IP-адресов, или одно из ключевых слов (`all`, чтобы указать любой IP-адрес, `samehost`, чтобы указать любые IP-адреса данного сервера, или `samenet`, чтобы указать любой адрес любой подсети, к которой сервер подключён напрямую);
- ✓ поле `METHOD` – указывает способ аутентификации и может принимать значения `peer`, `md5` и `trust`.

2.7.3 Настройки для резервного копирования и репликации

2.7.3.1 Резервное копирование

Для настройки резервного копирования используется секция «WRITE-AHEAD LOG» файла «postgresql.conf».

Параметр `wal_level` определяет, как много информации записывается в WAL (журнал предзаписи). Каждый последующий уровень включает информацию, записываемую на всех уровнях ниже. Минимальный журнал не будет содержать достаточно информации для восстановления данных из базовой копии и журналов, поэтому для реализации стратегии архивации WAL и потоковой репликации необходим уровень `replica` или более высокий.

```
wal_level = replica
```

Т.к. базовая резервная копия достаточно объемная, то выгоднее делать ее не каждый день. В таком случае, для восстановления понадобятся также архивные копии журналов. Поэтому при включении режима архивирования указываем действие для `archive_command`: в качестве значения для `archive_command` можно указывать и скрипт, в котором будет организована более сложная логика. Скрипт, в случае успешной работы, должен возвращать 0.

```
archive_mode = on
archive_command = 'cp %p /var/lib/pgpro/std-11/data/archive/%f'
```

Примечание 1. Если значение `archive_command` — пустая строка, но `archive_mode` включён, архивация WAL временно отключается, но сервер продолжает накапливать файлы сегментов WAL в ожидании, что команда будет вскоре определена.

Примечание 2. Подробнее про журнал предзаписи можно посмотреть в документации: <https://postgrespro.ru/docs/postgrespro/11/runtime-config-wal>.

2.7.3.2 Репликация

Для настройки репликации используется секция `REPLICATION` файла «postgresql.conf».

Параметр `max_wal_senders` — максимальное количество одновременно работающих процессов передачи WAL. Для работы `pg_basebackup` необходимо 2 подключения. Процессы-передатчики WAL учитываются в общем числе соединений, так что этот параметр не может превышать `max_connections`. Можно оставить значение по умолчанию: `max_wal_senders = 10`

2.7.4 Прочие настройки

В секции «RESOURCE USAGE» файла «postgresql.conf» необходимо задать объём памяти, который будет использовать сервер баз данных для буферов в

разделяемой памяти: в системах с объемом ОЗУ 1 ГБ и более, рекомендуемым начальным значением `shared_buffers` будет 25% от объема памяти:

```
shared_buffers = RAM/4
```

Примечание. Подробнее в документации:
<https://postgrespro.ru/docs/postgrespro/11/runtime-config-resource>.

3 Запуск Postgres Pro

Запуск выполняется под пользователем postgres:

```
$ pg_ctl -D $PGDATA start
```

Если логирование не настроено, нужно задать указать путь к лог-файлу:

```
$ pg_ctl -D $PGDATA -l /path/to/logfile start
```

3.1 Проверка подключения к серверу Postgres

Под пользователем postgres подключиться, используя утилиту psql:

```
$ psql
```

или

```
$ psql -U username -d databasename -h host_address
```

Проверка возможности удаленного подключения:

```
$ sudo netstat -pant | grep postgres
tcp        0      0 0.0.0.0:5432          0.0.0.0:*           LISTEN
52437/postgres
...
```

Если указан адрес 127.0.0.1, то подключение к консоли возможно только с localhost.

3.2 Проверка работы сервера Postgres

Статус сервера проверяется командой:

```
$ pg_ctl status
pg_ctl: server is running (PID: 52437)
```

Если сервер Postgres стартовал успешно, в каталоге `/var/lib/pgpro/std-11/data/` создается файл «`postmaster.pid`».

Запущенные процессы Postgres:

```
ps -ax | grep postgres
 1277 ?        Ss      0:03 /usr/lib/postfix/master -w
 52437 ?        S       0:08 /opt/pgpro/std-11/bin/postgres -D
/var/lib/pgpro/std-11/data
 52438 ?        Ss      0:00 postgres: logger
 52440 ?        Ss      0:00 postgres: checkpointer
 52441 ?        Ss      0:03 postgres: background writer
 52442 ?        Ss      0:03 postgres: walwriter
 52443 ?        Ss      0:07 postgres: autovacuum launcher
 52444 ?        Ss      0:00 postgres: archiver last was ...
 52445 ?        Ss      0:21 postgres: stats collector
```

3.3 Останов сервера Postgres

```
pg_ctl stop
```

Параметры команды:

- ✓ `w`, `--wait` – ждет завершения операции (по умолчанию);
- ✓ `W`, `--no-wait` – не ждет завершения операции;
- ✓ `m`, `--mode=MODE` может принимать значения `smart`, `fast`, `or immediate`. По умолчанию режим `fast`.

3.4 Настройка клиента

Установка клиентской части производится из полного дистрибутива, который можно скачать на сайте <https://postgrespro.ru/>.

3.5 Среда разработки

Бесплатную кросс-платформенную среду разработки с web-интерфейсом, ориентированную на Postgres, – PGAdmin – можно скачать с сайта <https://www.pgadmin.org/>, через меню Downloads, выбрав операционную систему, на которую производить установку.

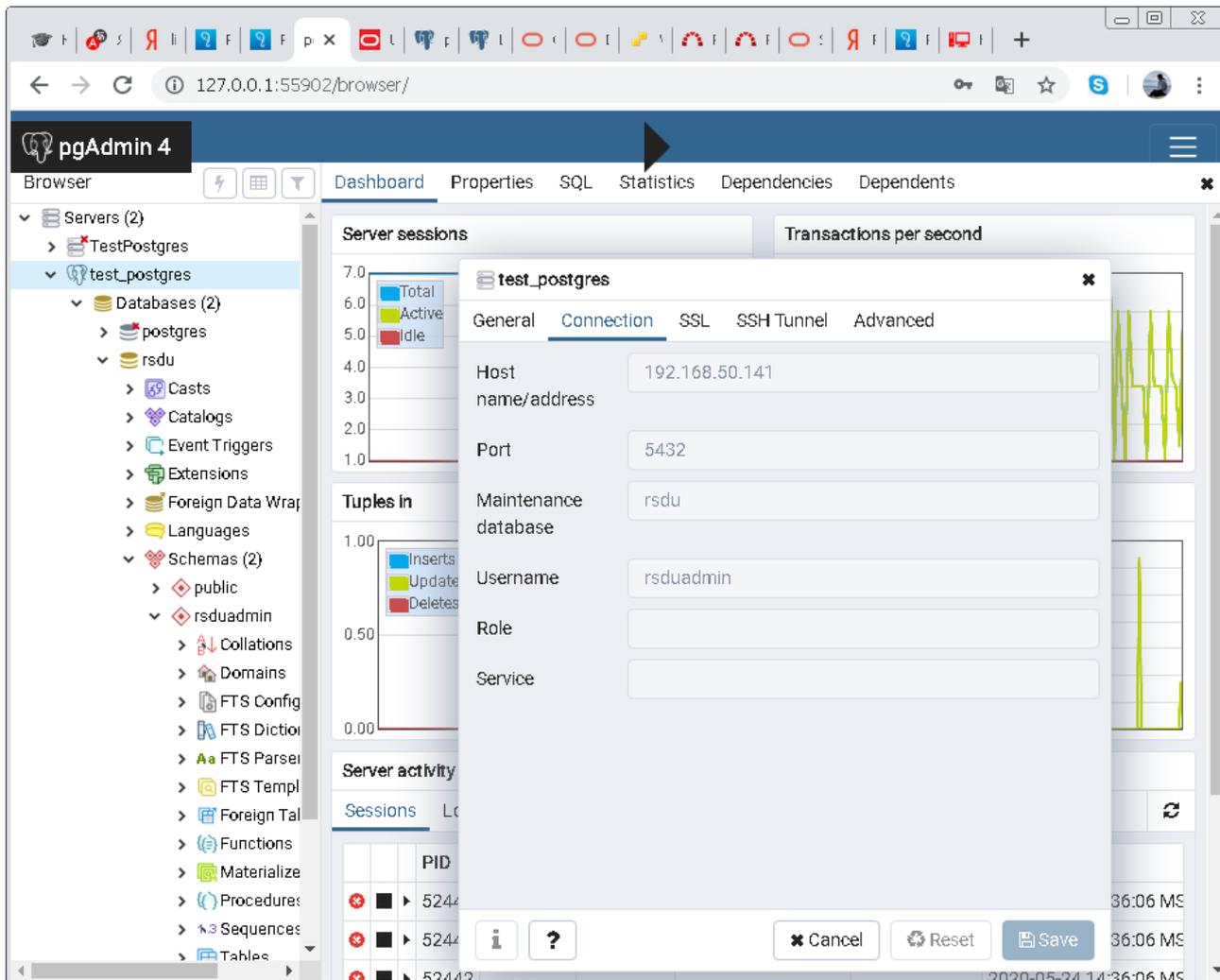


Рисунок 1 - Пример среды разработки PGAdmin

Также существует альтернативная среда разработки для ОС Windows – HeidiSQL. Ее можно скачать с сайта <https://www.heidisql.com>.

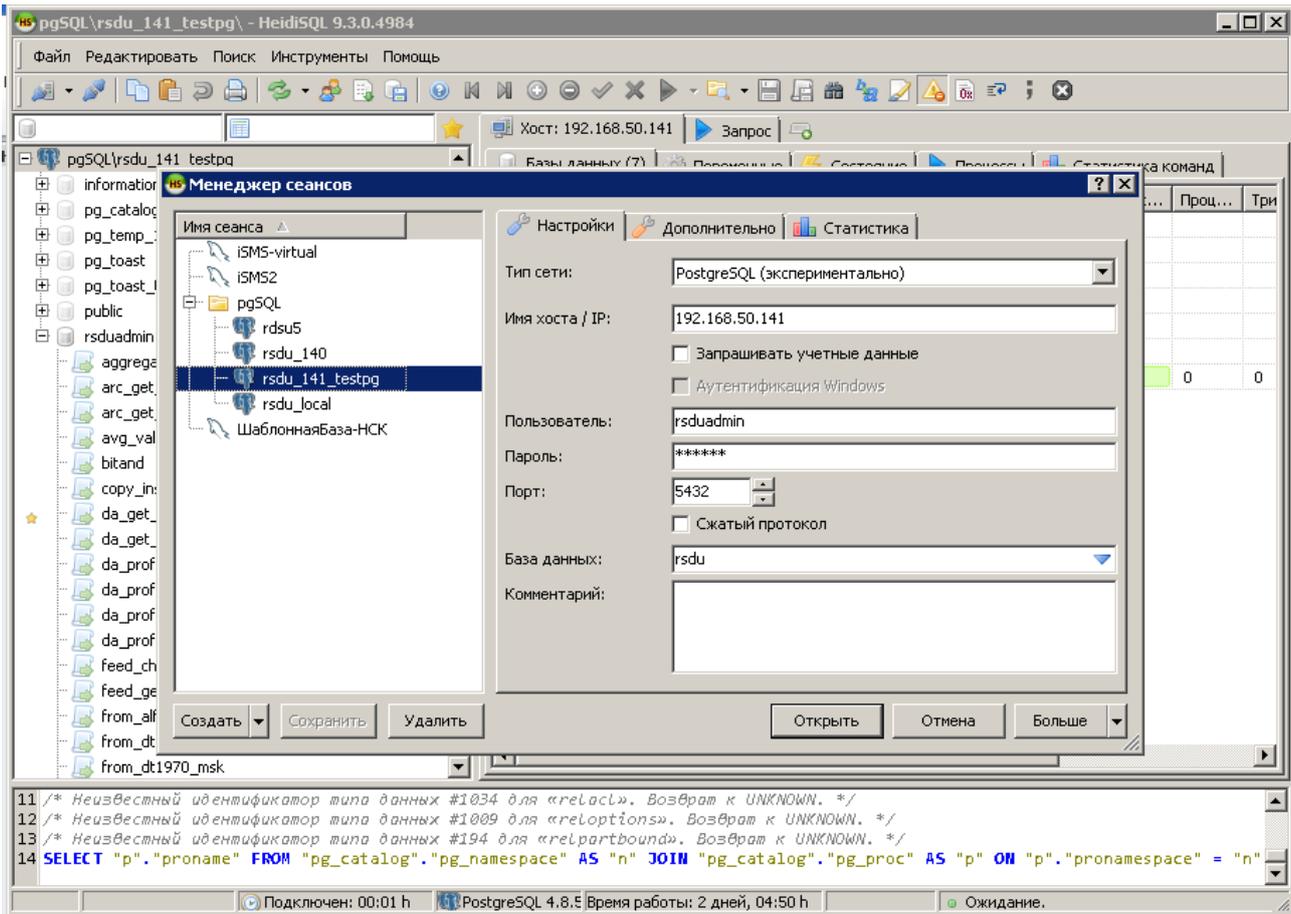


Рисунок 2 - Пример среды разработки HeidiSQL