

Обзор базового инструментария Установка и управление, psql



Авторские права

© Postgres Professional, 2017 год.

Авторы: Егор Рогов, Павел Лузанов

Использование материалов курса

Некоммерческое использование материалов курса (презентации, демонстрации) разрешается без ограничений. Коммерческое использование возможно только с письменного разрешения компании Postgres Professional. Запрещается внесение изменений в материалы курса.

Обратная связь

Отзывы, замечания и предложения направляйте по адресу:
edu@postgrespro.ru

Отказ от ответственности

Компания Postgres Professional не несет никакой ответственности за любые повреждения и убытки, включая потерю дохода, нанесенные прямым или непрямым, специальным или случайным использованием материалов курса. Компания Postgres Professional не предоставляет каких-либо гарантий на материалы курса. Материалы курса предоставляются на основе принципа «как есть» и компания Postgres Professional не обязана предоставлять сопровождение, поддержку, обновления, расширения и изменения.

Установка и управление сервером

Использование psql

Варианты установки

Управление сервером

Журнал сервера

Настройка параметров

Где взять PostgreSQL, как запустить или остановить сервер, где посмотреть журнал работы, как проверить и установить новые значения конфигурационным параметрам.

Минимальные сведения по этим вопросам необходимы каждому работающему с PostgreSQL.

Подробно темы установки и управления сервером рассматриваются в курсах для администраторов БД.

Варианты установки

готовые пакеты — предпочтительный способ
установка из исходных кодов

Расширения

дополнительный функционал
устанавливаются отдельно
в поставке с сервером — модули и программы (50 штук)

Предпочтительный вариант установки PostgreSQL это использование готовых пакетов. В этом случае получается понятная, поддерживаемая и легко обновляемая установка. Пакеты существуют для большинства широко распространенных систем.

Другой вариант установки – сборка системы из исходных кодов. Такой вариант может понадобиться при сборке с нестандартными параметрами или при использовании платформы для которой нет готового пакета.

Готовые пакеты и исходные коды расположены здесь:

<http://www.postgresql.org/download/>

В курсе мы будем использовать виртуальную машину с ОС Ubuntu и сервер PostgreSQL, установленный из пакета для этой ОС.

Для PostgreSQL существует большое количество расширений, которые подключают новый функционал к СУБД «на лету», без изменения ядра системы.

В состав дистрибутива PostgreSQL входит примерно 50 различных расширений. Их описание включено в документацию:

<https://postgrespro.ru/docs/postgresql/9.6/contrib.html>

<https://postgrespro.ru/docs/postgresql/9.6/contrib-prog.html>

Список доступных расширений и статус их установки можно посмотреть в представлении `pg_available_extensions`.

Утилита для управления

`pg_ctlcluster`  `pg_ctl`

Основные задачи

- запуск сервера
- останов сервера
- обновление параметров конфигурации

К основным операциям управления сервером относятся инициализация или создание кластера БД, запуск/останов сервера, обновление конфигурации и некоторые другие. Для выполнения этих действий предназначена утилита `pg_ctl`, идущая в составе сервера.

В пакетном дистрибутиве для Ubuntu доступ к утилите `pg_ctl` осуществляется не напрямую, а через специальную обертку `pg_ctlcluster`. Справку по использованию `pg_ctlcluster` можно получить командой:

```
$ man pg_ctlcluster
```

Более подробная информация об управлении сервером для администраторов баз данных:

<https://postgrespro.ru/docs/postgresql/9.6/app-pg-ctl.html>

<https://postgrespro.ru/docs/postgresql/9.6/runtime.html>

В журнал записываются

- служебные сообщения сервера
- сообщения пользовательских сеансов
- сообщения приложений

Настройка журнала

- расположение
- формат записей
- какие события регистрировать

Информация о ходе работы СУБД записывается в журнал сервера. Сюда попадают сведения о запуске/останове сервера, различные служебные сообщения о возникающих проблемах.

Также сюда могут попадать сообщения о работе пользовательских процессов: какие команды и сколько времени выполняются, возникающие блокировки и пр. Это позволяет выполнять трассировку работы пользовательских сеансов.

Разработчики приложений могут записывать в журнал свои собственные сообщения.

Настройки PostgreSQL позволяют гибко определять какие именно сообщения и в каком формате должны попадать в журнал сервера.

Например, вывод в формате csv удобен для автоматизации анализа журнала.

<https://postgrespro.ru/docs/postgresql/9.6/runtime-config-logging.html>

Параметры конфигурации

основной файл параметров — postgresql.conf

ALTER SYSTEM — postgresql.auto.conf

Внесение изменений для всего экземпляра

изменить postgresql.conf или выполнить ALTER SYSTEM

обновить конфигурацию

проверить — pg_settings

У сервера PostgreSQL есть большое число параметров конфигурации для настройки тех или иных возможностей: управление потреблением ресурсов, настройка служебных процессов и пользовательских сеансов, управление журналом сервера и многое другое. В ходе курса мы будем встречаться с некоторыми параметрами. А сейчас важно разобраться с тем, как проверить текущие значения и установить новые.

Настройки сервера обычно задаются в конфигурационных файлах. Основной конфигурационный файл — postgresql.conf. Помимо этого, есть команда ALTER SYSTEM, которая позволяет вносить изменения во второй конфигурационный файл postgresql.auto.conf. Параметры установленные через ALTER SYSTEM имеют приоритет над параметрами в postgresql.conf.

Типовой сценарий изменения параметров для всей системы выглядит следующим образом:

1. Вносим изменения в файл postgresql.conf или выполняем команду ALTER SYSTEM SET *parameter* TO *new_value*;
2. Обновляем конфигурацию (pg_ctlcluster reload или вызываем функцию pg_reload_conf). Для некоторых параметров требуется перезагрузка сервера.
3. Проверяем, что изменения применились. Представление pg_settings содержит актуальные значения всех параметров.

Варианты установки и управления параметрами:

<https://postgrespro.ru/docs/postgresql/9.6/config-setting.html>

Описание представления pg_settings:

<https://postgrespro.ru/docs/postgresql/9.6/view-pg-settings.html>

Установка во время выполнения

SET/RESET
set_config()

Просмотр

SHOW
current_setting()

Большинство параметров конфигурации допускает изменение значений в пользовательских сеансах прямо во время выполнения.

Изменить параметры можно командой SET или функцией set_config.

Для получения текущих значений можно воспользоваться командой SHOW или функцией current_setting.

Терминальный клиент для работы с PostgreSQL

Поставляется вместе с СУБД

Используется администраторами и разработчиками для интерактивной работы и выполнения скриптов

Для работы с СУБД PostgreSQL существуют различные сторонние инструменты, рассмотрение которых не входит в рамки курса.

В курсе мы будем использовать терминальный клиент psql:

1. psql — это единственный клиент, поставляемый вместе с СУБД.
2. Навыки работы с psql пригодятся разработчикам и администраторам БД вне зависимости от того, с каким инструментом они будут работать дальше.

Для интерактивной работы в psql встроена поддержка readline, программ постраничного просмотра результатов запросов (more, less). Возможности psql позволяют взаимодействовать с ОС, просматривать содержимое системного каталога, создавать скрипты для автоматизации повторяющихся задач.

Полное описание psql:

<https://postgrespro.ru/docs/postgresql/9.6/app-psql.html>

Запуск

```
$ psql -d база -U роль -h узел -p порт
```

Новое подключение в psql

```
=> \c[onnect] база роль узел порт
```

Информация о текущем подключении

```
=> \conninfo
```

При запуске psql нужно указать параметры подключения.

К обязательным параметрам подключения относятся: имя базы данных, имя пользователя, имя сервера, номер порта. Если эти параметры не указаны, psql попытается подключиться, используя значения по умолчанию:

- *база* — совпадает с именем пользователя
- *роль* — совпадает с именем пользователя ОС
- *узел* — локальное соединение
- *порт* — обычно 5432

Настройки сделанные для целей курса позволяют подключаться к PostgreSQL без указания параметров.

Если требуется выполнить новое подключение, не выходя из psql, то нужно выполнить команду `\connect`.

Команда `\conninfo` выдает информацию о текущем подключении.

Дополнительная информация о возможностях настройки подключения:

<https://postgrespro.ru/docs/postgresql/9.6/libpq-envvars.html>

<https://postgrespro.ru/docs/postgresql/9.6/libpq-pgservice.html>

<https://postgrespro.ru/docs/postgresql/9.6/libpq-pgpass.html>

В командной строке ОС

```
$ psql --help  
$ man psql
```

В psql

=> \?	список команд psql
=> \? variables	переменные psql
=> \h[elp]	список команд SQL
=> \h <i>команда</i>	синтаксис команды SQL
=> \q	выход

Справочную информацию по psql можно получить не только в документации, но и прямо в системе.

psql с ключом `--help` выдает справку по запуску. А если в системе была установлена документация, то справочное руководство можно получить командой `man psql`.

psql умеет выполнять команды sql и свои собственные команды.

Внутри psql есть возможность получить список и краткое описание команд psql. Все команды psql начинаются с обратной косой черты — \.

Командой `\help` выдает список команд SQL, которые поддерживает сервер, а также синтаксис конкретной команды SQL.



Установка PostgreSQL из готовых пакетов —
предпочтительный способ установки

Пакетные дистрибутивы учитывают особенности ОС,
которые нужно знать

- как запускать/останавливать сервер
- расположение файлов конфигурации
- расположение журнала сервера

psql — клиент для работы с PostgreSQL

1. Установите в `postgresql.conf` для параметра `work_mem` значение 8 мегабайт.
2. Обновите конфигурацию и проверьте, что изменения вступили в силу.
3. Запишите в файл `ddl.sql` команду `CREATE TABLE` на создание любой таблицы.
4. Запишите в файл `populate.sql` команды на вставку строк в эту таблицу.
5. Войдите в `psql`, выполните оба скрипта и проверьте, что таблица создана и в ней появились записи.
6. Найдите в журнале сервера все строки за сегодняшний день.

Для выполнения практических заданий нужно войти в операционную систему под пользователем `student` (пароль `student`).

Для запуска `psql` в окне терминала наберите `psql` без параметров. Для подключения будут использованы настройки по умолчанию:

```
student@student:~$ psql
psql (9.6.3)
Type "help" for help.
```

```
student=# \conninfo
You are connected to database "student" as user "student" via
socket in "/var/run/postgresql" at port "5432".
student=#
```