Установка PostgreSQL

Темы занятия

Установка PostgreSQL

Создание кластера БД

Запуск и останов сервера

Установка расширений

Установка пакета

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

Готовые пакеты – предпочтительный способ

FreeBSD, OpenBSD

- пакеты из Ports and Packages Collection

Red Hat, Debian, Ubuntu Linux

- входит в дистрибутив ОС + репозиторий (yum, apt)

Linux (другие версии)

- пакеты RPM, DEB

Solaris

Mac OS X

Windows

Другие варианты

Установка из исходных кодов

- собрать сервер со специфичными параметрами или на специфичной архитектуре
- (этот способ попробуем на практике)

Установка из репозитория git

- в первую очередь для разработчиков
- требует более широкого набора инструментов

Необходимое ПО

Postgres PROFESSIONAL

Обязательно

tar, gzip/bzip2, GNU make, компилятор Си (C89)

Используются, но можно отключить

библиотеки GNU Readline, zlib

Дополнительно

языки программирования Perl, Python, Tcl
для использования PL/Perl, PL/Python, PL/Tcl

Kerberos, OpenSSL, OpenLDAP, PAM
для аутентификации и шифрования

Отдельные инструменты при сборке из репозитория git

Исходный код

http://www.postgresql.org/ftp/source/

postgresql-9.4.4.tar.gz

```
$ gunzip postgresql-9.4.4.tar.gz
$ tar xf postgresql-9.4.4.tar
$ cd postgresql-9.4.4
```

postgresql-9.4.4.tar.bz2

```
$ bunzip2 postgresql-9.4.4.tar.bz2
$ tar xf postgresql-9.4.4.tar
$ cd postgresql-9.4.4
```

Конфигурация

Сброс (перед повторной конфигурацией)

```
$ make distclean
```

Конфигурация

```
$ ./configure
```

параметры	`--prefix`	каталог установки (/usr/local/pgsql/)
	`--enable-debug`	отладочная информация
	…	
переменные	`CC`	компилятор Си
	`CFLAGS`	параметры компилятора
	…	

проверить после установки: `$ pg_config`

Сборка и установка сервера

Сборка

`$ make`	All of PostgreSQL is successfully made. Ready to install.
или	
`$ make world`	PostgreSQL, contrib and HTML documentation successfully made. Ready to install.

Проверка

`$ make check`	All 145 tests passed.

Установка

`$ sudo make install`	PostgreSQL installation complete.
или	
`$ sudo make install-world`	PostgreSQL, contrib, and documentation installation complete.

Установка клиента

Установка только необходимого для клиента

```
$ sudo make -C src/bin install
$ sudo make -C src/include install
$ sudo make -C src/interfaces install
$ sudo make -C doc install
```

Настройка

Пользователь, под которым будет работать СУБД

```
$ sudo adduser postgres
```

Каталог для данных

```
$ sudo mkdir /usr/local/pgsql/data
$ sudo chown postgres /usr/local/pgsql/data
```

Далее все под пользователем postgres!

Создание кластера

Переменные окружения:

```
$ export PATH=/usr/local/pgsql/bin:$PATH
$ export MANPATH=/usr/local/pgsql/share/man:$MANPATH
$ export PGDATA=/usr/local/pgsql/data
```

вместо $PGDATA можно указывать путь в параметре -D

Создание

```
$ initdb -k -D /usr/local/pgsql/data
```

или

```
$ pg_ctl initdb -o "-k" -D /usr/local/pgsql/data
```

Запуск сервера

Запуск

```
$ postgres >logfile 2>&1 &
```

или

```
$ pg_ctl start -l logfile
```

Проверка

```
$ psql -c 'select now()'
```

Останов сервера

Останов

```
$ pg_ctl stop -m smart|fast|immediate
```

smart — ожидает завершения всех сеансов и выполняет контрольную точку (`kill -TERM` *pid*)

fast — принудительно завершает сеансы и выполняет контрольную точку (`kill -INT` *pid*)

immediate — принудительно завершает сеансы, при запуске потребуется восстановление (`kill -QUIT` *pid*)

Установка расширений

Поставляемые расширения

модули (45 штук)

клиентские и серверные программы (9 штук)

Сборка и установка

```
$ make
$ sudo make install
```

можно выполнить для всех расширений (из каталога contrib/)
или для отдельного (из каталога contrib/.../)

для модулей, добавляющих объекты SQL, дополнительно требуется выполнить команду CREATE EXTENSION

некоторые расширения имеют отдельную инструкцию по установке

Итоги

Познакомились с вариантами установки PostgreSQL

Собрали СУБД из исходных кодов

Создали кластер БД

Научились запускать и останавливать сервер

Познакомились с процедурой установки расширений

Практика

1. Собрать PostgreSQL без расширений и установить его
2. Создать кластер, запустить сервер
3. Убедиться, что сервер работает
4. Остановить сервер
5. Собрать и установить расширение oid2name

Авторские права
Курс «Администрирование PostgreSQL 9.4. Базовый курс» разработан в компании Postgres Professional (2015 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.

Темы занятия

Установка PostgreSQL

Создание кластера БД

Запуск и останов сервера

Установка расширений

Установка пакета

http://www.postgresql.org/download/
Готовые пакеты – предпочтительный способ

FreeBSD, OpenBSD
- пакеты из Ports and Packages Collection

Red Hat, Debian, Ubuntu Linux
- входит в дистрибутив ОС + репозиторий (yum, apt)

Linux (другие версии)
- пакеты RPM, DEB

Solaris

Mac OS X

Windows

Предпочтительным вариантом является использование готовых пакетов, так как в этом случае получается понятная, поддерживаемая и легко обновляемая установка.

Пакеты существуют для большинства широко распространенных систем, но в каждой из них имеются свои особенности установки, которые здесь не рассматриваются.

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

Другие варианты

Postgres PROFESSIONAL

Установка из исходных кодов

- собрать сервер со специфичными параметрами или на специфичной архитектуре
- (этот способ попробуем на практике)

Установка из репозитория git

- в первую очередь для разработчиков
- требует более широкого набора инструментов

Полезно иметь представление о том, как происходит установка из исходных кодов, чтобы в случае необходимости можно было собрать PostgreSQL с нестандартными параметрами или на специфичной архитектуре.

При установке из репозитория git требуется более широкий набор инструментов. Например, лексический и синтаксический анализатор сделан с помощью Flex и Bison, и в git хранятся именно исходные коды для этих инструментов. При сборке из них генерируются файлы на Си, которые затем компилируются. При этом в архиве с исходными кодами находится уже сгенерированные файлы Си.

Необходимое ПО

Обязательно

tar, gzip/bzip2, GNU make, компилятор Си (C89)

Используются, но можно отключить

библиотеки GNU Readline, zlib

Дополнительно

языки программирования Perl, Python, Tcl
для использования PL/Perl, PL/Python, PL/Tcl

Kerberos, OpenSSL, OpenLDAP, PAM
для аутентификации и шифрования

Отдельные инструменты при сборке из репозитория git

Для сборки требуется ряд программ и утилит.

Библиотека readline дает возможность редактировать командную строку, пользоваться историей команд и автодополнением. В серверной инсталляции может быть и не нужна, если не предполагается запускать psql в консоли.

Библиотека zlib используется для сжатия архивов pg_dump.

Исходный код

http://www.postgresql.org/ftp/source/

postgresql-9.4.4.tar.gz

```
$ gunzip postgresql-9.4.4.tar.gz
$ tar xf postgresql-9.4.4.tar
$ cd postgresql-9.4.4
```

postgresql-9.4.4.tar.bz2

```
$ bunzip2 postgresql-9.4.4.tar.bz2
$ tar xf postgresql-9.4.4.tar
$ cd postgresql-9.4.4
```


Исходный код доступен по указанному адресу в двух вариантах (gzip и bzip2).

Весь процесс установки выполняется обычным пользователем системы. Права системного администратора требуются только для последнего шага – установки уже собранного PostgreSQL.

Конфигурация

Postgres PROFESSIONAL

Сброс (перед повторной конфигурацией)

```
$ make distclean
```

Конфигурация

```
$ ./configure
```

параметры	`--prefix`	каталог установки (/usr/local/pgsql/)
	`--enable-debug`	отладочная информация
	…	
переменные	`CC`	компилятор Си
	`CFLAGS`	параметры компилятора
	…	

проверить после установки: `$ pg_config`

Конфигурация проверяет наличие необходимого инструментария, узнает особенности системы (такие, как занимаемый разными типами данных объем памяти) и настраивает скрипты сборки, учитывая также переданные параметры.

Различных параметров очень много, мы не будем их подробно рассматривать. В качестве примера: параметры, которые разработчики могут попросить выставить для поиска проблем: --enable_debug для компиляции с отладочной информации и CCFLAGS для отключения оптимизации.

Перед повторной конфигурацией следует выполнить очистку с помощью make distclean.

После установки можно посмотреть параметры с помощью утилиты pg_config.

http://www.postgresql.org/docs/9.4/static/install-procedure.html

Сборка и установка сервера

Сборка

`$ make`	All of PostgreSQL is successfully made. Ready to install.
или	
`$ make world`	PostgreSQL, contrib and HTML documentation successfully made. Ready to install.

Проверка

`$ make check`	All 145 tests passed.

Установка

`$ sudo make install`	PostgreSQL installation complete.
или	
`$ sudo make install-world`	PostgreSQL, contrib, and documentation installation complete.

Сборка выполняется с помощью GNU make. Можно собрать только СУБД, а можно и вместе с расширениями и документацией.

После сборки можно (но не обязательно) проверить результат с помощью тестов.

Собранный PostgreSQL необходимо установить (каталог установки указывается при конфигурировании). Для этого требуются права администратора.

http://www.postgresql.org/docs/9.4/static/install-procedure.html

Установка клиента

Установка только необходимого для клиента

```
$ sudo make -C src/bin install
$ sudo make -C src/include install
$ sudo make -C src/interfaces install
$ sudo make -C doc install
```


Если требуется установка только необходимого для подключения к внешней СУБД, нужно выполнить сборку полностью и установить только необходимые файлы.

http://www.postgresql.org/docs/9.4/static/install-procedure.html

Настройка

Пользователь, под которым будет работать СУБД

```
$ sudo adduser postgres
```

Каталог для данных

```
$ sudo mkdir /usr/local/pgsql/data
$ sudo chown postgres /usr/local/pgsql/data
```

Далее все под пользователем postgres!

СУБД рассчитана на работу под отдельным пользователем postgres.

http://www.postgresql.org/docs/9.4/static/postgres-user.html

Также необходимо создать каталог для данных. Он должен принадлежать пользователю postgres.

http://www.postgresql.org/docs/9.4/static/creating-cluster.html

Создание кластера

Переменные окружения:

```
$ export PATH=/usr/local/pgsql/bin:$PATH
$ export MANPATH=/usr/local/pgsql/share/man:$MANPATH
$ export PGDATA=/usr/local/pgsql/data
```

вместо $PGDATA можно указывать путь в параметре -D

Создание

```
$ initdb -k -D /usr/local/pgsql/data
```

или

```
$ pg_ctl initdb -o "-k" -D /usr/local/pgsql/data
```


PATH позволяет вызывать утилиты PostgreSQL из командной строки (путь можно уточнить командой pg_config --bindir).

MANPATH нужен для вызова справки, если сервер был собран и установлен с документацией (путь можно уточнить командой pg_config --mandir).

PGDATA указывает путь до каталога с данными, созданного на прошлом шаге. Эту переменную можно не задавать, но тогда путь придется указывать в параметрах всех команд.

На практике установку этих переменных следует автоматизировать.

Ключ -k утилиты initdb включает подсчет контрольной суммы страниц, что позволяет своевременно обнаружить повреждение данных.

http://www.postgresql.org/docs/9.4/static/install-post.html

http://www.postgresql.org/docs/9.4/static/creating-cluster.html

Запуск сервера

Запуск

```
$ postgres >logfile 2>&1 &
```

или

```
$ pg_ctl start -l logfile
```

Проверка

```
$ psql -c 'select now()'
```


Запуск сервера – это обычно запуск программы postgres в фоновом режиме. Вывод перенаправляется в журнал, туда же перенаправляется и поток ошибок. Но удобнее пользоваться pg_ctl.

И в том, и другом случае postgres «демонизируется» (отвязывается от породившего его процесса).

Чтобы проверить, что сервер запустился, можно выполнить простую команду (вывод текущего времени). Программа psql будет подробно рассмотрена в следующем занятии.

http://www.postgresql.org/docs/9.4/static/server-start.html

Останов сервера

Останов

```
$ pg_ctl stop -m smart|fast|immediate
```

smart	ожидает завершения всех сеансов и выполняет контрольную точку (`kill -TERM` *pid*)
fast	принудительно завершает сеансы и выполняет контрольную точку (`kill -INT` *pid*)
immediate	принудительно завершает сеансы, при запуске потребуется восстановление (`kill -QUIT` *pid*)

Есть три режима останова: дождаться всех сеансов и выполнить контрольную точку (неопределенно долго), принудительно завершить все сеансы и выполнить контрольную точку (быстрее) и просто принудительно завершить все сеансы (совсем быстро, но при старте серверу потребуется восстановление).

Можно воспользоваться pg_ctl, а можно вручную послать соответствующий сигнал процессу postmaster (его номер можно узнать с помощью pg_ctl status или посмотрев в файл $PGDATA/postmaster.pid).

Удобный способ, позволяющий уменьшить время простоя (при выполнении первой контрольной точки система работоспособна; вторая контрольная точка выполнится уже быстро):

```
$ psql -c CHECKPOINT && pg_ctl stop -m fast
```

http://www.postgresql.org/docs/9.4/static/server-shutdown.html

Установка расширений

Поставляемые расширения

модули (45 штук)

клиентские и серверные программы (9 штук)

Сборка и установка

```
$ make
$ sudo make install
```

можно выполнить для всех расширений (из каталога contrib/)
или для отдельного (из каталога contrib/.../)

для модулей, добавляющих объекты SQL, дополнительно требуется выполнить команду CREATE EXTENSION

некоторые расширения имеют отдельную инструкцию по установке

Кроме расширений, входящих в поставку, существует масса других расширений.

В практическом задании требуется установка расширений oid2name и pgcrypto, которые будут использоваться в теме «Управление базами данных». В той же теме рассматривается установка расширения с помощью CREATE EXTENSION.

Список расширений, входящих в поставку:

http://www.postgresql.org/docs/9.4/static/contrib.html

http://www.postgresql.org/docs/9.4/static/contrib-prog.html

Итоги

Познакомились с вариантами установки PostgreSQL

Собрали СУБД из исходных кодов

Создали кластер БД

Научились запускать и останавливать сервер

Познакомились с процедурой установки расширений

Практика

1. Собрать PostgreSQL без расширений и установить его
2. Создать кластер, запустить сервер
3. Убедиться, что сервер работает
4. Остановить сервер
5. Собрать и установить расширение oid2name