

Лабораторная работа №00: Установка PostgreSQL и базовое управление сервером

Цель работы: Освоить основные способы установки PostgreSQL (из пакетов и исходных кодов) в предустановленном окружении. Получить практические навыки создания кластера баз данных, запуска и остановки сервера, а также проверки его состояния. Научиться использовать утилиты `initdb`, `pg_ctl`, `pg_checksums` и `psql` для базового взаимодействия с СУБД.

Стек технологий:

- **ОС:** Xubuntu 24.04 64-bit (предустановлена в виртуальной машине)
- **СУБД:** PostgreSQL 16 (частично предустановлен из пакетов)
- **Утилиты:** `apt`, `pg_createcluster`, `initdb`, `pg_ctl`, `pg_checksums`, `psql`, `sudo`
- **Редакторы:** `nano`, `vim`, `gedit`, `mousepad` (на выбор)
- **Файловые менеджеры:** `mc`, `Thunar` (на выбор)

Настройка рабочего места: Установите систему виртуализации (необходим VirtualBox версии 6.1 или более новый), затем скачайте и импортируйте виртуальную машину <https://edu.postgrespro.ru/16/DBA1-16.ova>. Если у вас компьютер с процессором ARM64 (MacBook с процессором Apple Silicon), используйте VirtualBox версии 7.1 или выше и виртуальную машину <https://edu.postgrespro.ru/16/DBA1-16-arm64.ova>. Виртуальные машины по этим ссылкам обновляются с учетом изменений в учебном курсе. Обращаем внимание: установка виртуальной машины возможна только на компьютер с 64-битной операционной системой. Виртуальная машина содержит ОС Xubuntu 24.04 64-bit, в которой выполнены все настройки, необходимые для прохождения курса. Для работы вам потребуется минимум 1 Гбайт свободной оперативной памяти. Вход в систему выполняется под пользователем `student` (пароль `student`). Браузер Firefox виртуальной машины настроен на локальную документацию по PostgreSQL на русском языке. Для управления файлами можно использовать как командную строку (`ls`, `pwd`, `cd`, ...), так и графические менеджеры: `mc`, `Thunar` (встроенный файловый менеджер). Для редактирования файлов в системе предустановлены редакторы: `vim`, `nano`, `gedit`, `mousepad`.

Теоретическая часть (краткое содержание): Перед началом работы сервера PostgreSQL необходимо:

1. **Установить ПО:** Сервер можно установить из пакетов (предпочтительно для production) или собрать из исходных кодов (для кастомных сборок). В виртуальной машине часть пакетов уже предустановлена.
2. **Создать кластер БД:** Кластер баз данных — это совокупность баз данных, управляемых одним экземпляром сервера. Для его инициализации используется утилита `initdb`. Важной настройкой является включение контрольных сумм страниц данных (`--data-checksums`) для обнаружения повреждений.
3. **Запустить сервер:** Управление жизненным циклом сервера (запуск, останов, перезагрузка, проверка статуса) осуществляется утилитой `pg_ctl` (или `pg_ctlcluster` для пакетной установки).
4. **Подключиться к серверу:** Для проверки работоспособности и выполнения операций используется клиентская утилита `psql`.

Задание на практическую реализацию:

Работа состоит из двух независимых частей. Рекомендуется выполнить обе.

Часть 1: Установка из пакетов (Ubuntu) и управление кластером

1. **Установка:** Установите PostgreSQL 16 из официальных пакетов используя `apt`.
2. **Проверка кластера:** Убедитесь, что пакетный менеджер автоматически создал кластер `main`. Проверьте его статус с помощью `pg_lsclusters`.
3. **Включение контрольных сумм:** Остановите кластер. Проверьте, отключены ли контрольные суммы страниц данных с помощью `pg_checksums --check`. Включите расчет контрольных сумм командой `pg_checksums --enable`. Запустите кластер и убедитесь, что он работает.
4. **Управление сервером:** Используя утилиту `pg_ctlcluster`, выполните остановку, запуск и перезагрузку кластера. Проверяйте статус после каждой команды.
5. **Подключение:** Подключитесь к кластеру с помощью `psql` от имени пользователя `postgres` и выполните простой запрос (например, `SELECT version();`).

Часть 2: Установка из исходных кодов

1. **Подготовка:** Убедитесь, что установлены необходимые инструменты для сборки (`gcc`, `make`, `libreadline-dev` и др.).
2. **Сборка:** Скачайте архив с исходными кодами PostgreSQL 16. Распакуйте его. Сконфигурируйте сборку с указанием каталога установки (`--prefix`) и нестандартного порта (`--with-pgport`). Выполните сборку (`make`) и установку (`make install`) сервера.
3. **Создание кластера:** Создайте каталог для данных (`PGDATA`). Инициализируйте в нем новый кластер БД с помощью `initdb`, указав суперпользователя `postgres` и обязательно включив расчет контрольных сумм.
4. **Запуск и проверка:** Запустите сервер с помощью `pg_ctl`, указав файл для логов. Подключитесь к серверу через `psql`, указав нестандартный порт, и выполните проверочный запрос. Остановите сервер.

Требования к оформлению и отчету:

1. **Скрипты:** Предоставьте последовательность всех выполненных команд для обеих частей работы.
2. **Проверка:** Для каждой части приложите вывод команд, подтверждающих успешность действий:
 - Для Части 1: вывод `pg_lsclusters`, вывод `pg_checksums --check` (до и после), вывод `SELECT version();`.
 - Для Части 2: вывод `pg_ctl status`, вывод `SELECT version();` с указанием порта.
3. **Отчет:** Краткий отчет должен содержать:
 - Описание выполненных шагов.
 - Объяснение, зачем нужны контрольные суммы данных.
 - Сравнение двух способов установки (пакеты vs исходный код) по сложности и времени выполнения.

Критерии оценки:

- **Удовлетворительно:** Выполнена и успешно работает Часть 1 (пакетная установка и базовое управление).
- **Хорошо:** Выполнены и успешно работают обе части. Предоставлены все требуемые проверочные выводы команд.
- **Отлично:** В отчете присутствует осмысленное сравнение способов установки и объяснение важности контрольных сумм. В Части 2 сервер собран с нестандартными параметрами (порт, префикс).

Примечание: Для Части 2 будьте готовы к тому, что процесс компиляции может занять значительное время (10-30 минут).