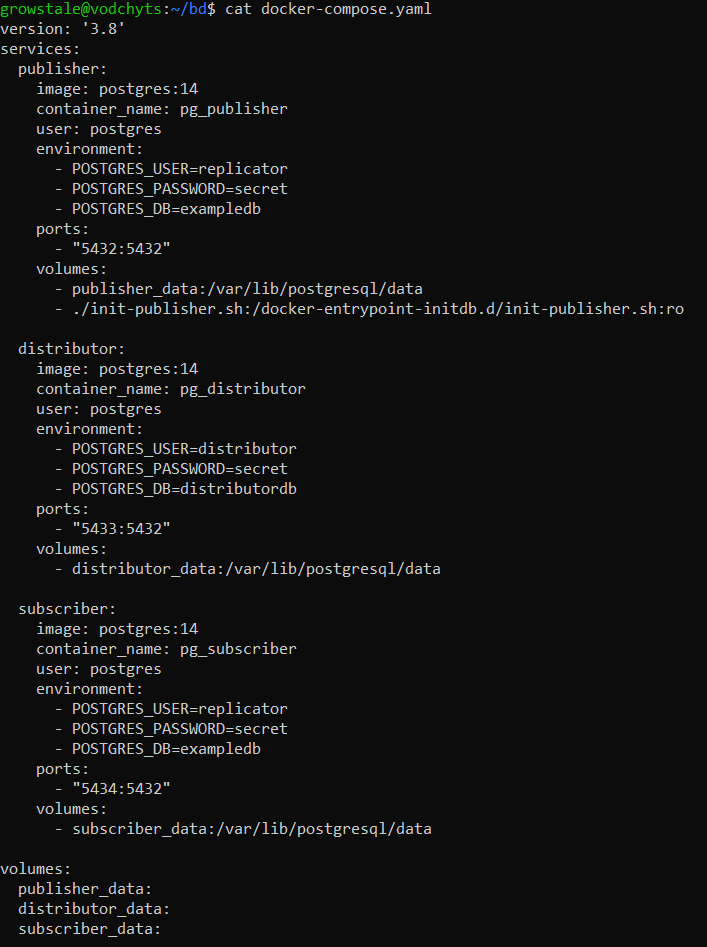
Команда docker compose up используется для запуска и создания сервисов, описанных в файле конфигурации **docker-compose.yml**.

docker-compose.yaml файл описывает конфигурацию для развертывания трёх контейнеров с PostgreSQL, которые участвуют в настройке репликации.



* **Три сервиса**: publisher, distributor, subscriber – это три отдельных контейнера PostgreSQL.
* **Три именованных тома**: publisher\_data, distributor\_data, subscriber\_data – используются для хранения данных БД, чтобы они сохранялись при перезапуске контейнеров.

**1. Publish (Издатель):** Это основная база данных, из которой будут распространяться изменения (репликация).

* Образ: postgres:14 (используется PostgreSQL версии 14).
* Контейнерное имя: pg\_publisher.
* Пользователь: replicator (важно для настройки репликации).
* Пароль: secret.
* База данных: exampledb.
* Порт: 5432:5432.

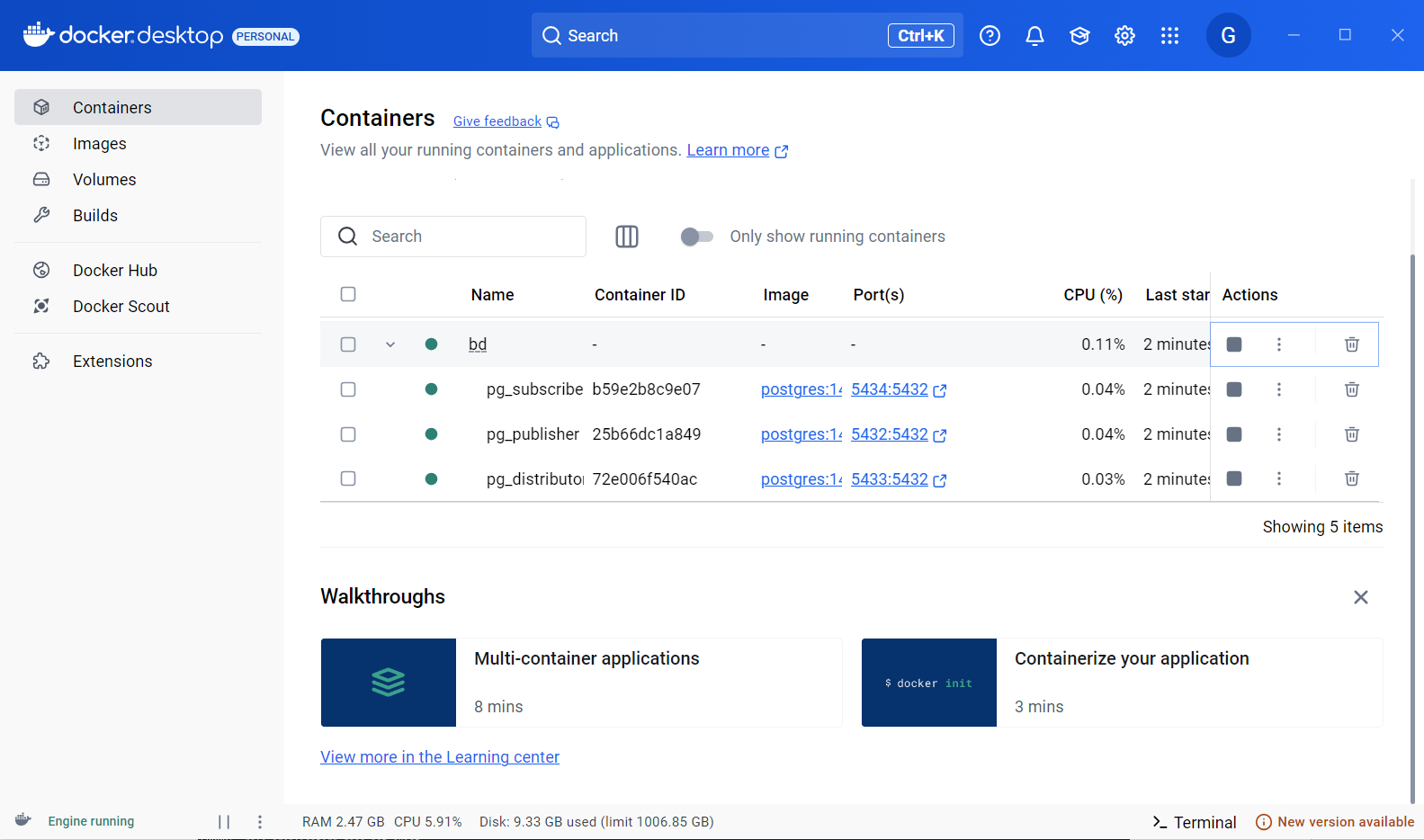
**2. Distributor (Распределитель):** Этот сервис используется как сервер распределения в репликации.

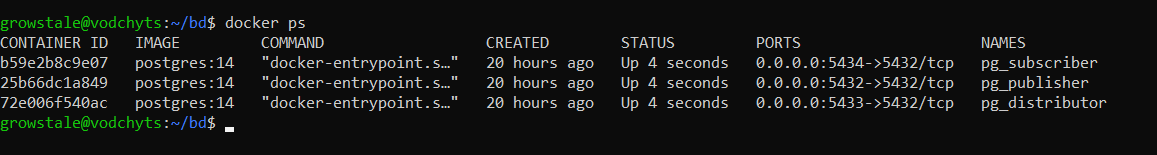
* Контейнерное имя: pg\_distributor.
* Пользователь: distributor.
* Пароль: secret.
* База данных: distributordb.
* Порт: 5433:5432 (то есть PostgreSQL внутри контейнера работает на 5432, но наружу он проброшен на 5433).
* Том: distributor\_data → хранит данные.

**3. Subscriber (Подписчик):** Это реплицируемая база данных, которая будет получать обновления от publisher.

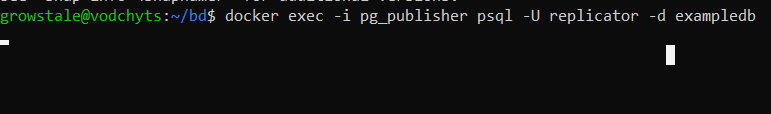
* Контейнерное имя: pg\_subscriber.
* Пользователь: replicator (такой же, как у publisher, что важно для репликации).
* Пароль: secret.
* База данных: exampledb (совпадает с publisher, что логично).
* Порт: 5434:5432.

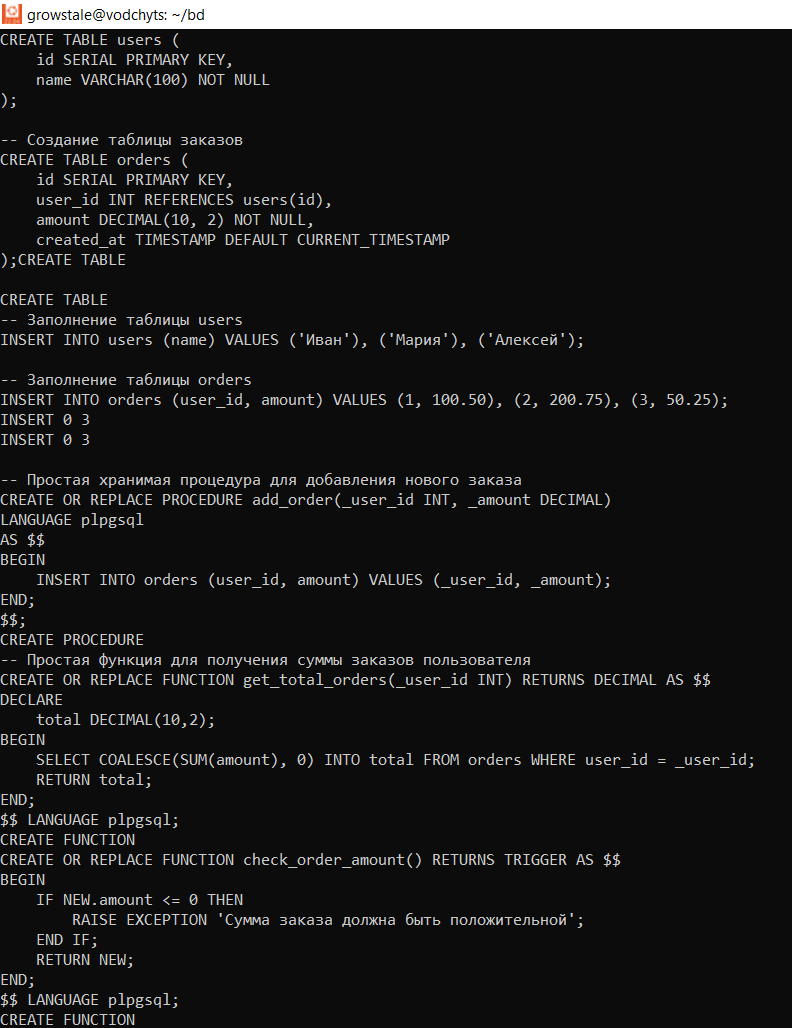
Отображаем список запущенных контейнеров в Docker:





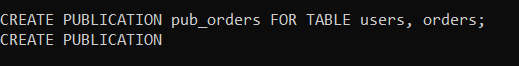
Эта команда выполняется в **терминале Linux** и используется для подключения к **PostgreSQL**, работающему в контейнере Docker.





Команда CREATE PUBLICATION используется для создания публикации в логической репликации.

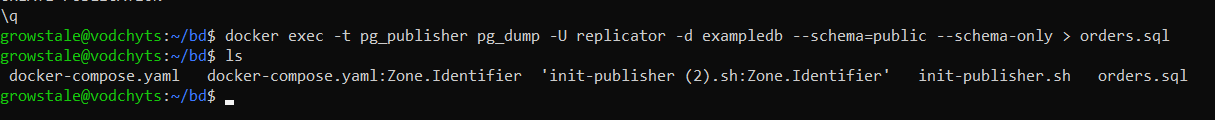
Логическая репликация в PostgreSQL — это механизм передачи изменений на уровне строк (DML-операции: INSERT, UPDATE, DELETE) между базами данных.



Эта команда выполняется в терминале Linux и используется для **экспорта схемы** (структуры) базы данных PostgreSQL из контейнера Docker.

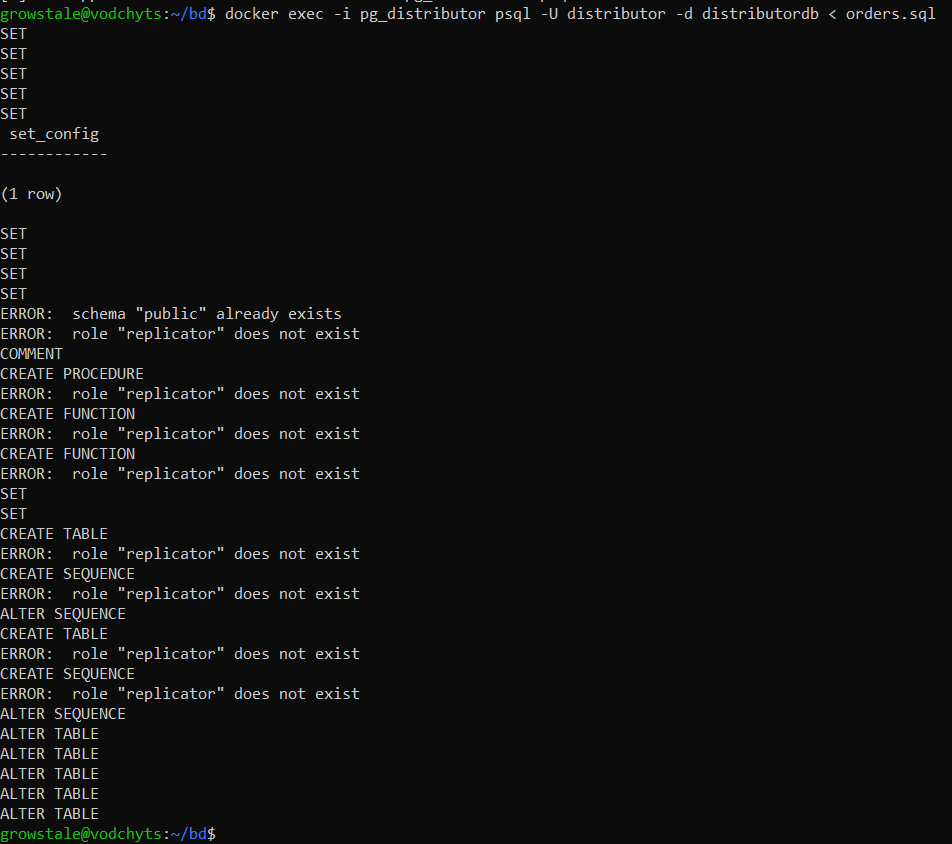
Запускает утилиту pg\_dump внутри контейнера, выполняя дамп базы exampledb от имени пользователя replicator.

Сохраняем результат в файл orders.sql на **хост-машине** (за пределами контейнера).



Эта команда выполняется в терминале Linux и используется для **импорта структуры БД PostgreSQL** в контейнере Docker, подключаясь к базе distributordb под пользователем distributor.

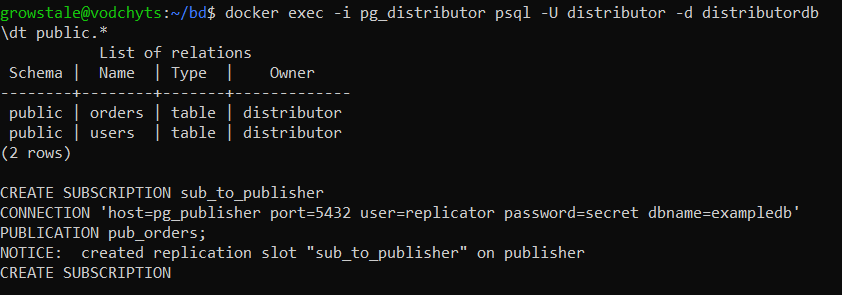
Перенаправляет содержимое файла orders.sql в psql, чтобы выполнить его команды.



Запускает **PostgreSQL**, подключаясь к базе данных distributordb под пользователем distributor.

Команда \dt public.\* используется в **PostgreSQL** для вывода списка таблиц в схеме public.

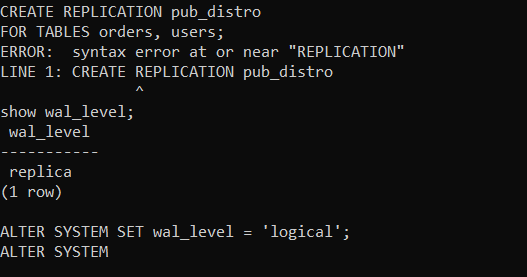
Далее создаём **подписку** (SUBSCRIPTION) с именем sub\_to\_publisher для логической репликации. Для этого подключается к **издателю** и подписывается на изменения из **публикации pub\_orders**.



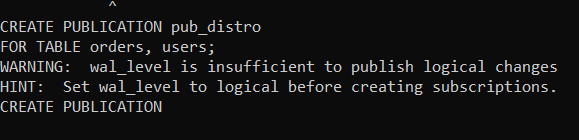
Команда **SHOW wal\_level** в PostgreSQL отображает текущий уровень записи **Write-Ahead Logging (WAL)**.

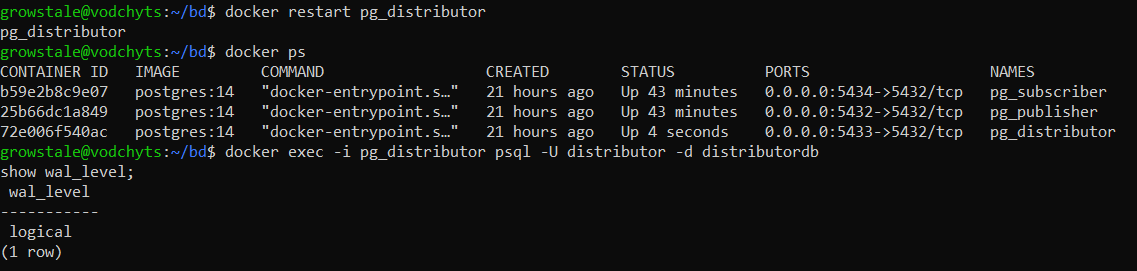
**wal\_level** — параметр, определяющий, сколько данных записывается в WAL (журнал предзаписи).

|  |  |
| --- | --- |
| minimal | Минимальный уровень, используется для восстановления после сбоев, но не поддерживает репликацию. |
| replica | Используется для потоковой (физической) репликации, включает данные для воспроизведения операций. |
| logical | Требуется для **логической репликации**, включает всю информацию о DML (INSERT, UPDATE, DELETE). |

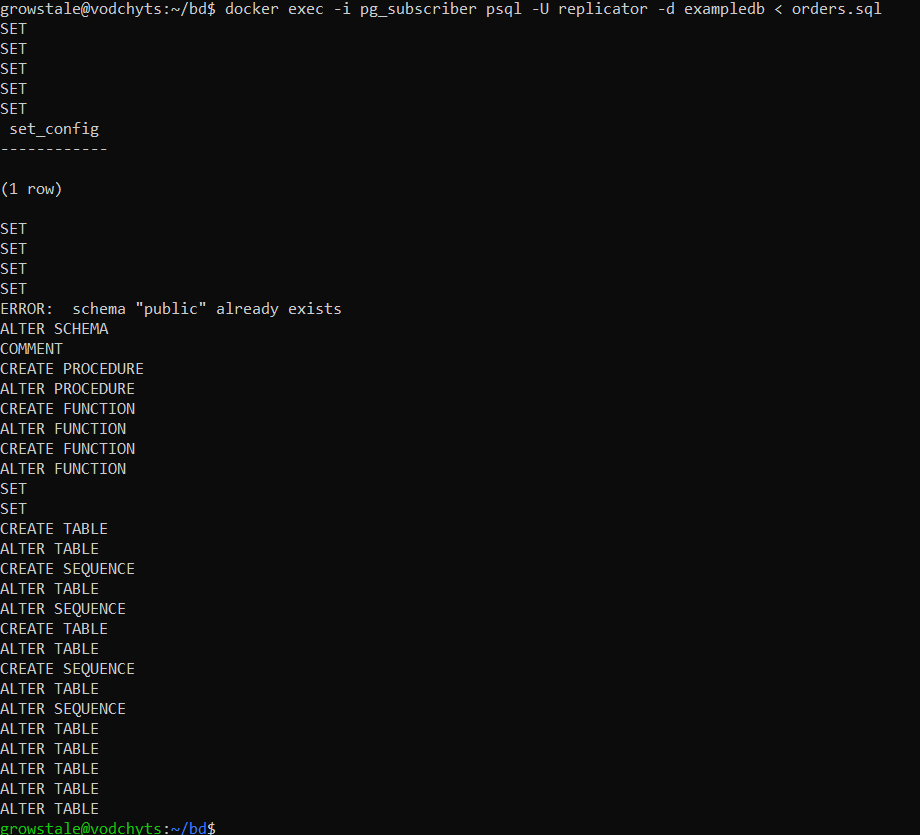


Создаём публикацию от дистрибьютера.

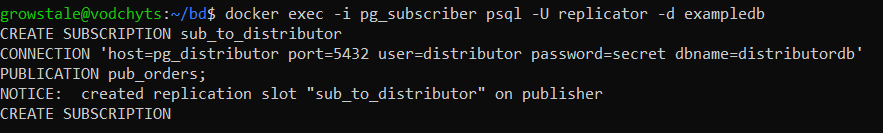




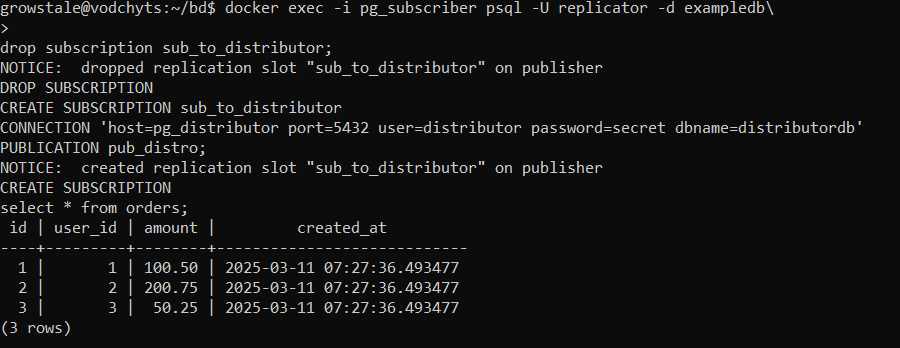
Импортируем структуру из файла orders.sql в базу данных exampledb внутри контейнера pg\_subscriber.



Открывает интерактивную сессию в psql для работы с базой данных exampledb в контейнере pg\_subscriber.



Подписываемся на публикацию дистрибьютера.



Этот запрос в PostgreSQL извлекает информацию о функциях, определённых в схеме public базы данных, из системных таблиц.

