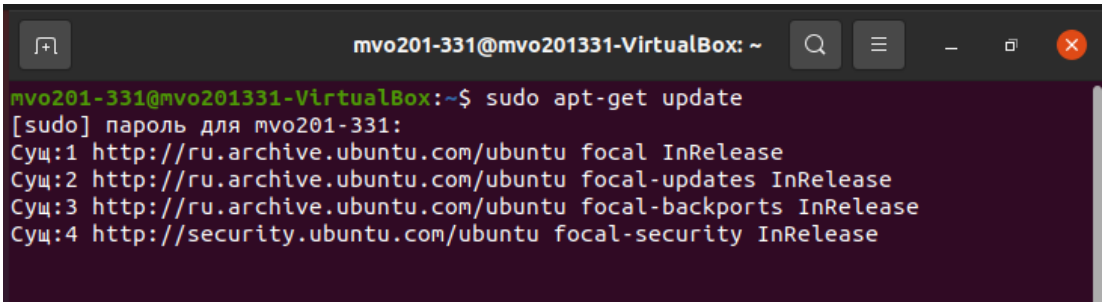
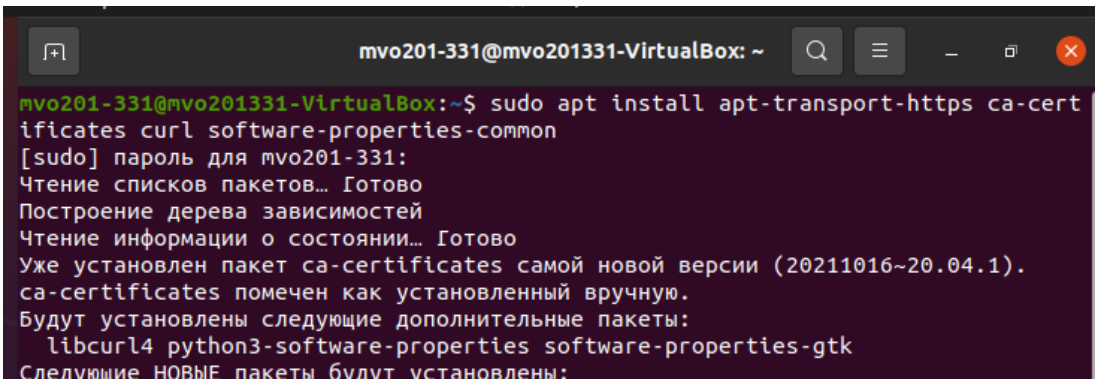
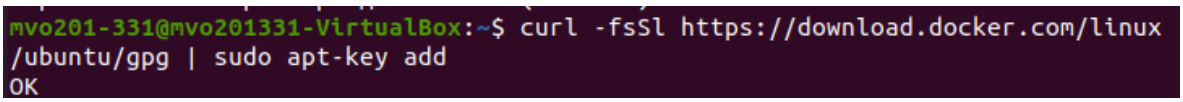
1. Установить Docker на всех развернутых в ходе ЛР2 виртуальных машинах

Обновим apt-get и установим пакеты, которые позволяют apt использовать пакеты через HTTPS:

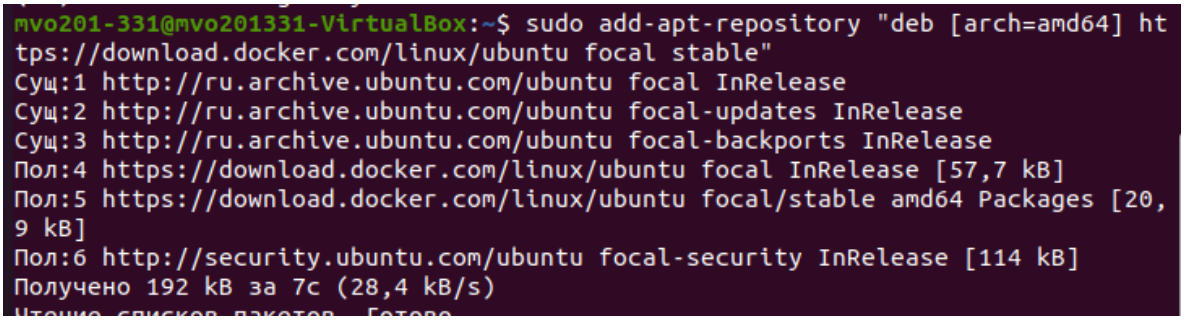




Добавим ключ GPG для официального репозитория Docker в систему:



Добавим репозиторий Docker в источники APT:



Проверим, что что установка будет выполняться из репозитория Docker

```
mvo201-331@mvo201331-VirtualBox:~$ apt-cache policy docker-ce
docker-ce:
  Установлен: (отсутствует)
  Кандидат:   5:20.10.21~3-0~ubuntu-focal
  Таблица версий:
     5:20.10.21~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Package
s
     5:20.10.20~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Package
s
     5:20.10.19~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Package
s
     5:20.10.18~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Package
s
     5:20.10.17~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Package
s
```

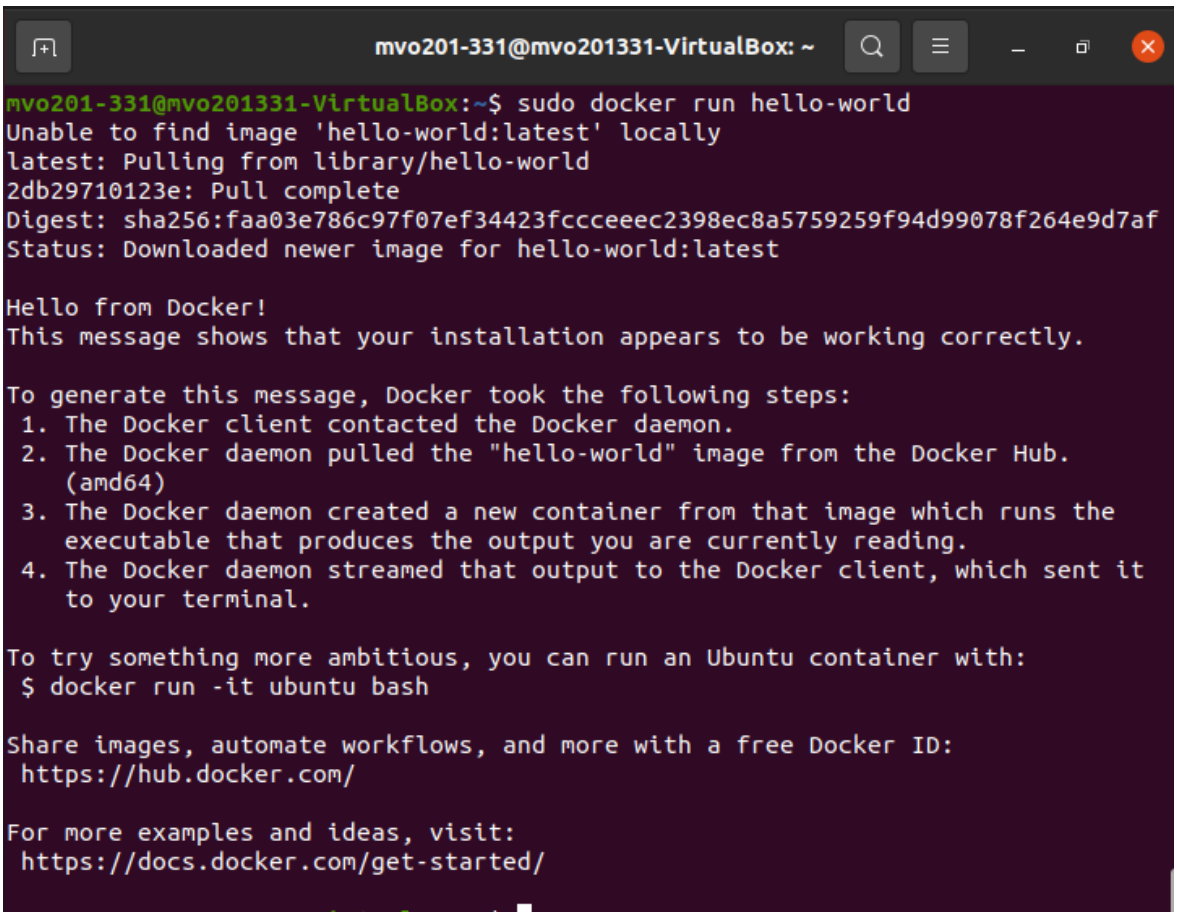
Установим Docker

```
mvo201-331@mvo201331-VirtualBox:~$ sudo apt install docker-ce
Чтение списков пакетов… Готово
Построение дерева зависимостей
Чтение информации о состоянии… Готово
Будут установлены следующие дополнительные пакеты:
  containerd.io docker-ce-cli docker-ce-rootless-extras docker-scan-plugin
  pigz slirp4netns
Предлагаемые пакеты:
  aufs-tools cgroupfs-mount | cgroup-lite
Следующие НОВЫЕ пакеты будут установлены:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras
  docker-scan-plugin pigz slirp4netns
Обновлено 0 пакетов, установлено 7 новых пакетов, для удаления отмечено 0 пакет
ов, и 156 пакетов не обновлено.
Необходимо скачать 102 MB архивов.
После данной операции объём занятого дискового пространства возрастёт на 383 MB
.
```

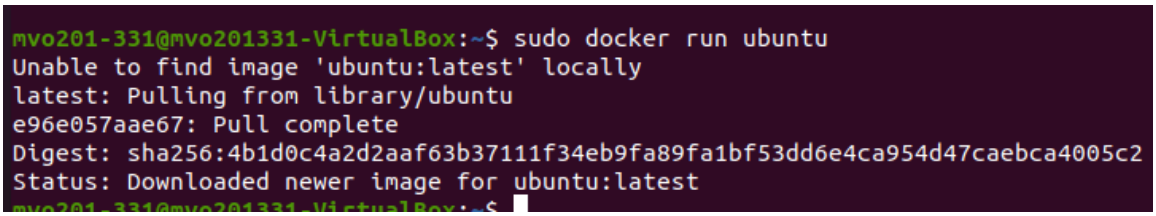
Проверим статус

```
mvo201-331@mvo201331-VirtualBox:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor prese>
     Active: active (running) since Sun 2022-12-04 22:03:14 MSK; 1min 33s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 7068 (dockerd)
      Tasks: 7
     Memory: 22.2M
     CGroup: /system.slice/docker.service
             └─7068 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con>
```

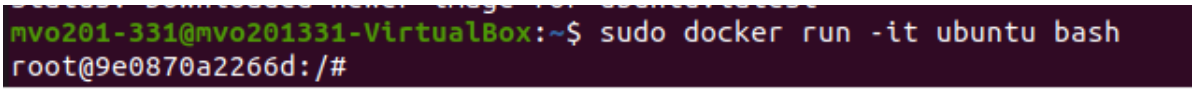
Проверим доступ к образам из Docker Hub



2. На усмотрение студента выбрать Docker-образ, который будет являться основой для разрабатываемого приложения

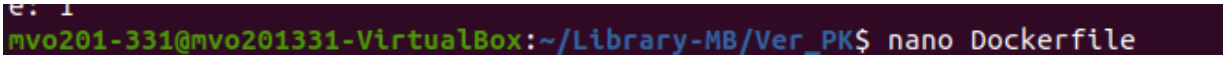


3. Получить образ на TEST-виртуальную машину

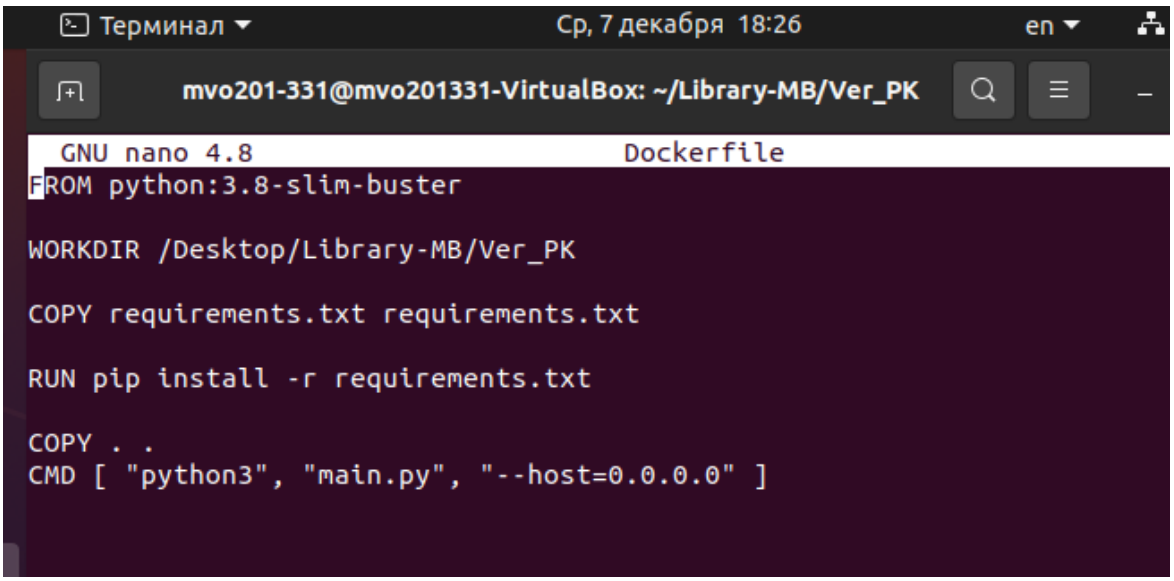


4. Добавить в корень проекта файл Dockerfile, внутри которого происходит сборка и запуск проекта из исходных файлов

   Создадим Dockerfile

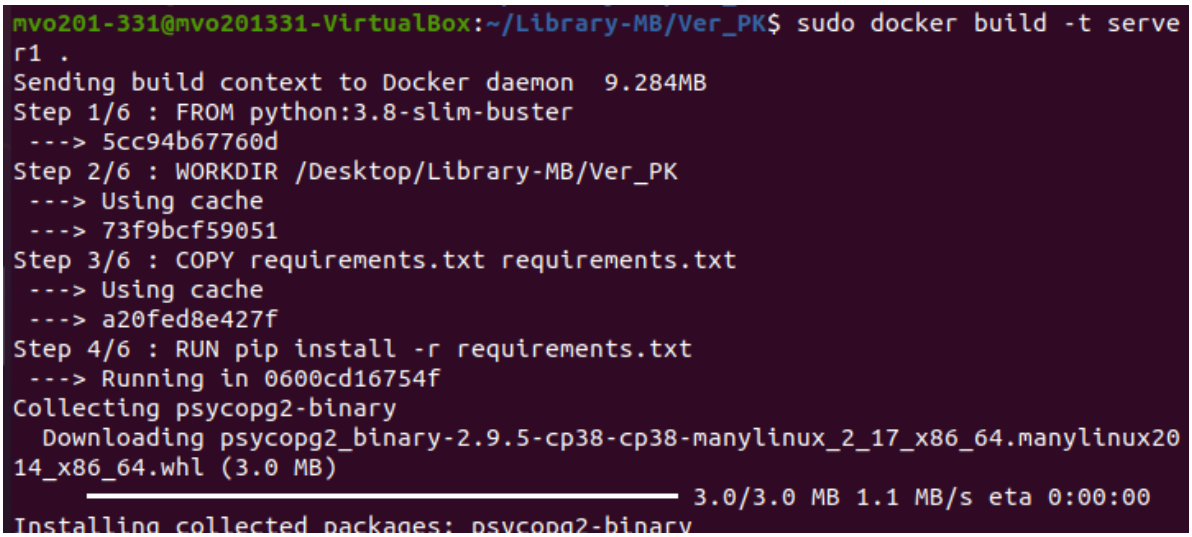


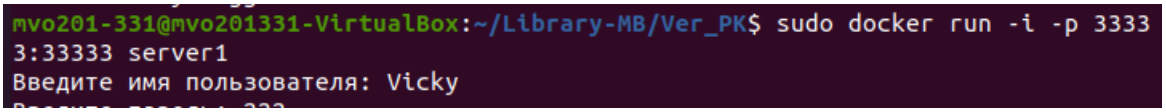
   Пропишем сборку и запуск проекта из исходных файлов (+ установка)



5. Собрать приложение внутри Docker, запустить контейнер из полученного образа

   Запустим





   Запуск приложения получился

   Часть 2

1. Установить на все ВМ утилиту docker-compose

   Установим утилиту, накинем права и выведем версию

```
mvo201-331@mvo201331-VirtualBox:~$ sudo curl -L "https://github.com/docker/comp
ose/releases/download/1.26.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/lo
cal/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 11.6M  100 11.6M    0     0   671k      0  0:00:17  0:00:17 --:--:--  801k
mvo201-331@mvo201331-VirtualBox:~$ sudo chmod +x /usr/local/bin/docker-compose
mvo201-331@mvo201331-VirtualBox:~$ docker-compose --version
docker-compose version 1.26.0, build d4451659
```

2. На выбор студента подобрать подходящую для приложения СУБД и получить с Dockerhub ее официальный (либо сторонний) образ. В случае, если приложение не использует БД, выбрать любой сторонний внешний сервис, используемый приложением (Redis, Minio и др.)

   Установим postgres

```
mvo201-331@mvo201331-VirtualBox:~$ sudo docker pull postgres:latest
[sudo] пароль для mvo201-331:
latest: Pulling from library/postgres
a603fa5e3b41: Pull complete
02d7a77348fd: Pull complete
16b62ca80c8f: Pull complete
fbd795da1fe1: Pull complete
9c68de39d930: Pull complete
2e441a95082c: Pull complete
1c97f440fe14: Pull complete
87a3f78bc5d1: Pull complete
264b18cba666: Pull complete
bea1513c492d: Pull complete
ed48cace97fa: Pull complete
e3c377e275ff: Pull complete
86fa351e30cb: Pull complete
Digest: sha256:766e8867182b474f02e48c7b1a556d12ddfa246138ddc748d70c891bf2873d82
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```

3. В корне проекта создать файл docker-compose.yaml, в котором описать параметры для сборки и запуска разрабатываемого приложения а также запуска стороннего сервиса, выбранного студентом. В конфигурации указать подключаемые директории (volume), в которых будут храниться файлы запущенных приложений (логи, файлы БД и другие).

   Создадим файл docker-compose.yaml

Dockerfile | main.py | docker-compose.yml

```
4
5   database:
6     container_name: db
7     hostname: db.local
8     image: postgres
9     restart: always
0     environment:
1       POSTGRES_PASSWORD: library
2       POSTGRES_USER: postgres
3       POSTGRES_DB: libraryMB
4     volumes:
5       - ./date_postgres:/var/lib/postgresql/data
6     ports:
7       - 33333:8080
8
9   app:
0     image: server1
1     hostname: app.local
2     container_name: libraryMB
3     build:
4       context: .
5       dockerfile: Dockerfile
6     depends_on:
7       - database
8     stdin_open: true
9     tty: true
```

YAML | Ширина табуляции: 8 | Стр 1

4. Запустить приложение и продемонстрировать его работу со сторонним сервисом.

   Запустим докер-компоус

```
mvo201-331@mvo201331-VirtualBox:~/Library-MB/Ver_PK$ sudo docker-compose up
Starting db ... done
Recreating libraryMB ... done
Attaching to db, libraryMB
db          |
db          | PostgreSQL Database directory appears to contain a database; Skip
ping initialization
db          |
db          | 2022-12-11 11:37:19.406 UTC [1] LOG:  starting PostgreSQL 15.1 (D
ebian 15.1-1.pgdg110+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 10.2.1-
6) 10.2.1 20210110, 64-bit
db          | 2022-12-11 11:37:19.434 UTC [1] LOG:  listening on IPv4 address "
0.0.0.0", port 5432
db          | 2022-12-11 11:37:19.434 UTC [1] LOG:  listening on IPv6 address "
::", port 5432
db          | 2022-12-11 11:37:19.535 UTC [1] LOG:  listening on Unix socket "/
var/run/postgresql/.s.PGSQL.5432"
db          | 2022-12-11 11:37:19.676 UTC [27] LOG:  database system was shut d
own at 2022-12-11 11:35:00 UTC
```

   Видим, что подключение к бд есть

```
db          | 2022-12-11 11:37:20.774 UTC [1] LOG:  database system is ready to
 accept connections
db          | 2022-12-11 11:42:19.773 UTC [25] LOG:  checkpoint starting: time
db          | 2022-12-11 11:42:19.955 UTC [25] LOG:  checkpoint complete: wrote
 3 buffers (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.021 s, s
ync=0.007 s, total=0.183 s; sync files=2, longest=0.004 s, average=0.004 s; dis
tance=0 kB, estimate=0 kB
```

Пробуем запустить

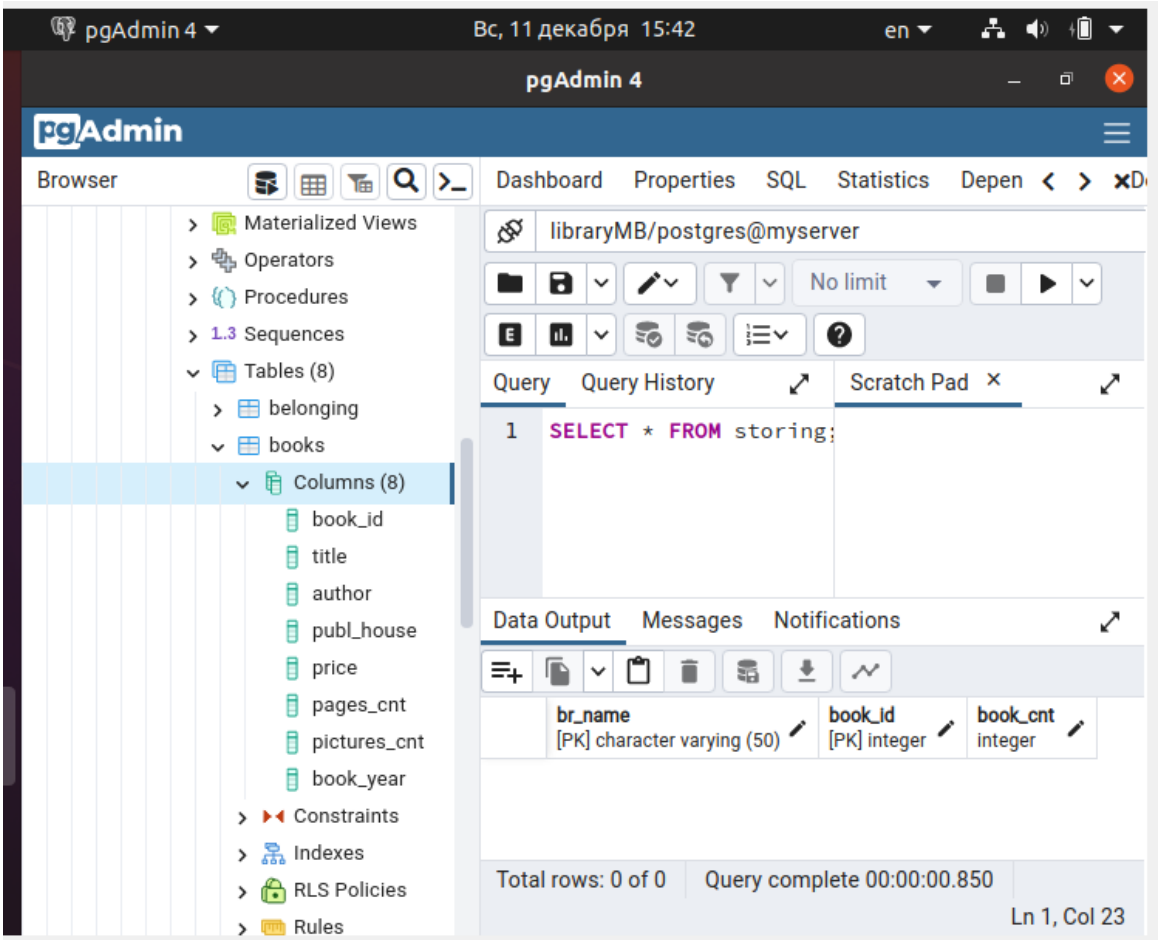
```
mvo201-331@mvo201331-VirtualBox:~/Library-MB/Ver_PK$ sudo docker attach library
MB
[sudo] пароль для mvo201-331:
postgres
Введите пароль: library
Вы подключены к -  ('PostgreSQL 15.1 (Debian 15.1-1.pgdg110+1) on x86_64-pc-lin
ux-gnu, compiled by gcc (Debian 10.2.1-6) 10.2.1 20210110, 64-bit',)


    Выполнить:
    0. Завершение работы
    1. Добавление информации о книгах в библиотеку
    2. Изменить информацию о книге в библиотеке
    3. Посмотреть информацию о всех книгах в библиотеке
    4. Добавить информацию о филиале
    5. Изменить информацию о филиале
    6. Для указанного филиала посчитать количество экземпляров указанной книги,
 находящихся в нем
    7. Для указанной книги посчитать количество факультетов, на которых она исп
ользуется в данном филиале, и вывести названия этих факультетов
```

Видим работу приложения

5. Внести изменения, требующие записи информации в файлы приложения (добавить записи в базу данных через приложение, создать файл и др.)

Воспользуемся командой 1 из приложения. Для этого покажем таблицу в базе данных.



Она пуста. Выполним команду 1

```
mvo201-331@mvo201331-VirtualBox:~/Library-MB/Ver_PK$ sudo docker attach library
MB
[sudo] пароль для mvo201-331:
postgres
Введите пароль: library
Вы подключены к -  ('PostgreSQL 15.1 (Debian 15.1-1.pgdg110+1) on x86_64-pc-lin
ux-gnu, compiled by gcc (Debian 10.2.1-6) 10.2.1 20210110, 64-bit',)


    Выполнить:
    0. Завершение работы
    1. Добавление информации о книгах в библиотеку
    2. Изменить информацию о книге в библиотеке
    3. Посмотреть информацию о всех книгах в библиотеке
    4. Добавить информацию о филиале
    5. Изменить информацию о филиале
    6. Для указанного филиала посчитать количество экземпляров указанной книги,
 находящихся в нем
    7. Для указанной книги посчитать количество факультетов, на которых она исп
ользуется в данном филиале, и вывести названия этих факультетов
    1
Введите через пробел: название филиала, id киги, кол-во книг
Электрозаводская 1 22

    Выполнить:
    0. Завершение работы
```

6. Остановить и запустить приложение еще раз, продемонстрировать, что изменения, внесенные в ходе работы, были сохранены.

Завершаем работу

```
Введите через пробел: название филиала, id киги, кол-во книг
Электрозаводская 1 22

    Выполнить:
    0. Завершение работы
    1. Добавление информации о книгах в библиотеку
    2. Изменить информацию о книге в библиотеке
    3. Посмотреть информацию о всех книгах в библиотеке
    4. Добавить информацию о филиале
    5. Изменить информацию о филиале
    6. Для указанного филиала посчитать количество экземпляров указанной книги,
 находящихся в нем
    7. Для указанной книги посчитать количество факультетов, на которых она исп
ользуется в данном филиале, и вывести названия этих факультетов
    0
Соединение с PostgreSQL закрыто
mvo201-331@mvo201331-VirtualBox:~/Library-MB/Ver_PK$
```

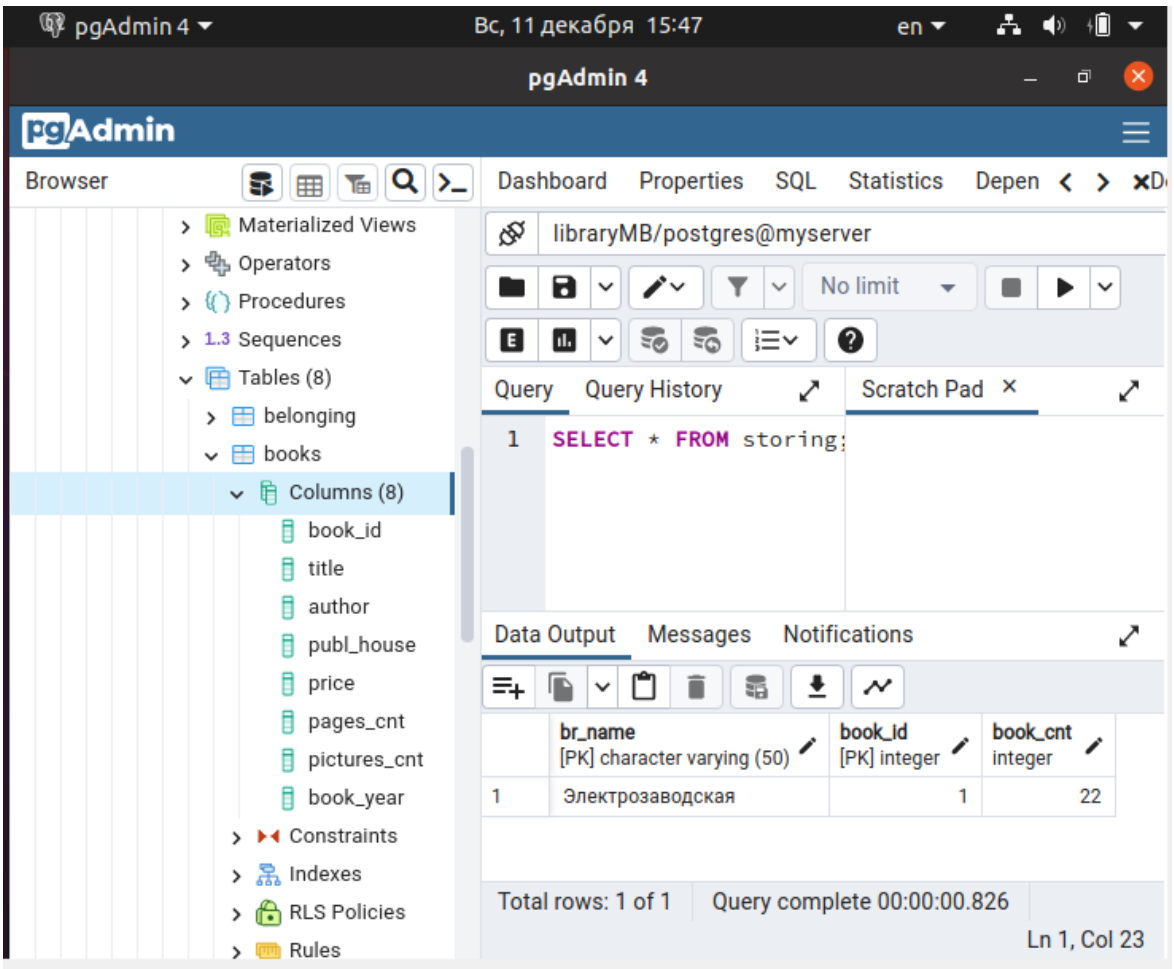
Запускаем докер-компоус

```
Терминал    Вс, 11 декабря 15:45    ru
mvo201-331@mvo201331-VirtualBox: ~/Library-MB/Ver_PK
mvo201-331@mvo201331-VirtualBox:~$ cd Library-MB/Ver_PK
mvo201-331@mvo201331-VirtualBox:~/Library-MB/Ver_PK$ sudo docker-compose up
[sudo] пароль для mvo201-331:
db is up-to-date
Starting libraryMB ... done
Attaching to db, libraryMB
db          |
db          | PostgreSQL Database directory appears to contain a database; Skip
ping initialization
db          |
db          | 2022-12-11 11:32:44.487 UTC [1] LOG:  starting PostgreSQL 15.1 (D
ebian 15.1-1.pgdg110+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 10.2.1-
6) 10.2.1 20210110, 64-bit
db          | 2022-12-11 11:32:44.506 UTC [1] LOG:  listening on IPv4 address "
0.0.0.0", port 5432
db          | 2022-12-11 11:32:44.506 UTC [1] LOG:  listening on IPv6 address "
::", port 5432
```

```
mvo201-331@mvo201331-VirtualBox:~/Library-MB/Ver_PK$ sudo docker attach library
MB
[sudo] пароль для mvo201-331:
postgres
Введите пароль: library
Вы подключены к -  ('PostgreSQL 15.1 (Debian 15.1-1.pgdg110+1) on x86_64-pc-lin
ux-gnu, compiled by gcc (Debian 10.2.1-6) 10.2.1 20210110, 64-bit',)


    Выполнить:
    0. Завершение работы
    1. Добавление информации о книгах в библиотеку
    2. Изменить информацию о книге в библиотеке
    3. Посмотреть информацию о всех книгах в библиотеке
    4. Добавить информацию о филиале
    5. Изменить информацию о филиале
    6. Для указанного филиала посчитать количество экземпляров указанной книги,
 находящихся в нем
```

Посмотрим, есть ли добавление



На данные вопросы необходимо ответить при защите лабораторной работы.

1. Каким образом docker обеспечивает изоляцию контейнеров от хостовой ОС и друг от друга?

   Приложение, запущенное в контейнере думает, что оно одно во всей ОС. Изоляция достигается за счет использования таких Linux-механизмов, как namespaces и control groups. Если говорить просто, то namespaces обеспечивают изоляцию в рамках ОС, а control groups устанавливают лимиты на потребление контейнером ресурсов хоста, чтобы сбалансировать распределение ресурсов между запущенными контейнерами

2. Какие параметры требуется передать контейнеру docker при запуске, чтобы его сетевой адаптер был тождественен сетевому адаптеру хостовой ОС и почему?

--net =host

При данном режиме контейнер создаёт совместные ресурсы своего сетевого пространства имён с хостом.

3. Какую последовательность команд требуется ввести, чтобы при помощи docker-compose сначала собрать приложение, затем запустить его в фоновом режиме и затем подключиться к его выводу stdout и stderr?

   • Docker compose build

   • Docker compose up -d

   • Docker compose exec