

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**  
**Федеральное государственное автономное**  
**образовательное учреждение высшего образования**  
**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Лабораторная работа 3**  
**по дисциплине**  
**«Автоматизация процессов жизненного цикла программных средств»**

**Выполнил:**  
**ст. гр. 2xx-3xx**  
**Иванов И. И.**  
**Проверил:**  
**Красников А. С.**

**Москва – 2024**

## План выполнения лабораторной работы

### Часть 1

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

Обновим существующий список пакетов:

```
zaa201-331@alicia:~$ sudo apt update
Сущ:1 http://ru.archive.ubuntu.com/ubuntu jammy InRelease
Пол:2 http://ru.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Пол:3 http://ru.archive.ubuntu.com/ubuntu jammy-backports InRelease [99,8 kB]
Пол:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
```

Затем установим несколько необходимых пакетов, которые позволяют apt использовать пакеты через HTTPS:

```
zaa201-331@alicia:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
[sudo] пароль для zaa201-331:
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Уже установлен пакет ca-certificates самой новой версии (20211016).
ca-certificates помечен как установленный вручную.
```

Добавим ключ GPG для официального репозитория Docker в вашу систему:

```
zaa201-331@alicia:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

Добавим репозиторий Docker в источники APT:

```
zaa201-331@alicia:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
Репозиторий: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable'
Описание:
Archive for codename: focal components: stable
Дополнительные сведения: https://download.docker.com/linux/ubuntu
Добавление репозитория.
```

Убедимся, что установка будет выполняться из репозитория Docker, а не из репозитория Ubuntu по умолчанию:

```
zaa201-331@alicia:~$ apt-cache policy docker-ce
docker-ce:
  Установлен: (отсутствует)
  Кандидат:  5:20.10.18-3-0~ubuntu-focal
  Таблица версий:
    5:20.10.18-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.17-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
```

Установим Docker:

```
zaa201-331@alicia:~$ sudo apt install docker-ce
[sudo] пароль для zaa201-331:
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  containerd.io docker-ce-cli docker-ce-rootless-extras docker-scan-plugin
  libslirp0 pigz slirp4netns
Предлагаемые пакеты:
  aufs-tools cgroupfs-mount | cgroup-lite
```

Docker должен быть установлен, демон-процесс запущен, а для процесса активирован запуск при загрузке. Проверим, что он запущен:

```
zaa201-331@alicia:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Sat 2022-10-08 17:03:48 MSK; 1min 41s ago
    TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 8146 (dockerd)
     Tasks: 7
    Memory: 21.8M
      CPU: 162ms
     CGroup: /system.slice/docker.service
             └─8146 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Чтобы проверить, можно ли получить доступ к образам из Docker Hub и загрузить их, введем следующую команду:

```
zaa201-331@alicia:~$ sudo docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

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

Выберем Docker-образ ubuntu:

```
zaa201-331@alicia:~$ sudo docker run ubuntu
[sudo] пароль для zaa201-331:
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
2b55860d4c66: Pull complete
Digest: sha256:20fa2d7bb4de7723f542be5923b06c4d704370f0390e4ae9e1c833c8785644c1
Status: Downloaded newer image for ubuntu:latest
```

3. Получить образ на TEST-виртуальную машину

Получим образ на TEST-BM:

```
zaa201-331@alicia:/$ sudo docker run -it ubuntu bash
[sudo] пароль для zaa201-331:
root@1f3f7e2edd45:/#
```

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

Перейдем в корень проекта:

```
zaa201-331@alicia:/$ cd /home/server
zaa201-331@alicia:/home/server$ nano Dockerfile
```

Напишем Dockerfile, внутри которого происходит сборка и запуск проекта из исходных файлов:

GNU nano 6.2	Dockerfile
FROM ubuntu:20.04	
RUN apt update	
RUN apt install qt5-qmake -y	
RUN apt install cmake -y	
RUN apt install qt5-default -y	
RUN apt install build-essential -y	
COPY . /home/server	
WORKDIR /home/server	
RUN qmake server1.pro	
RUN make	
RUN make install	
CMD ["./server1"]	

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

Соберем сервер внутри Docker:

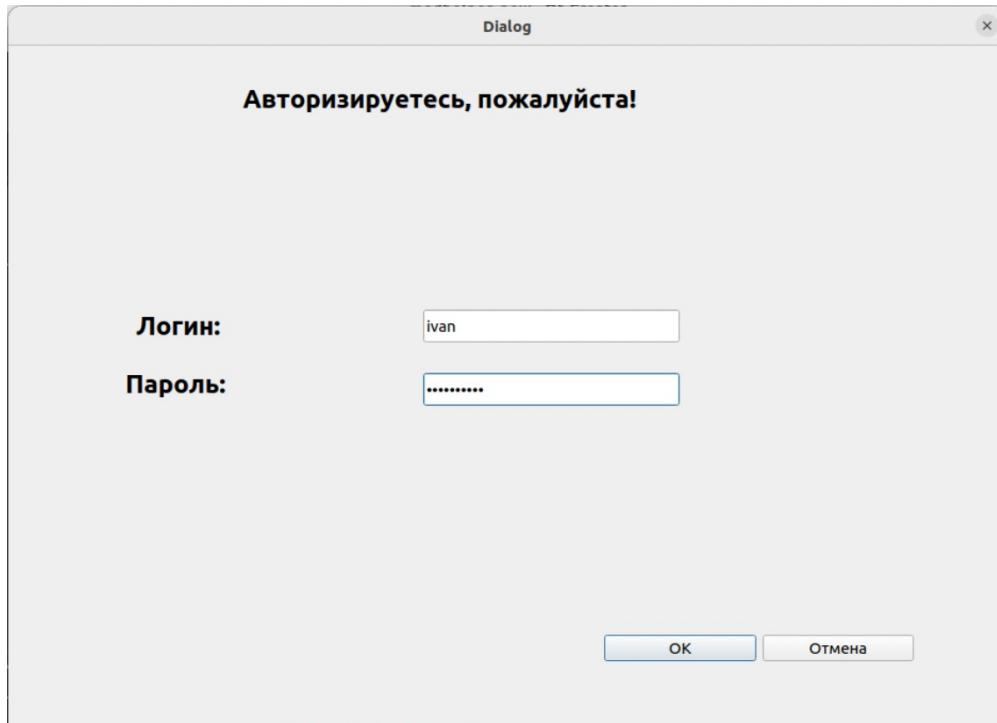
```
zaa201-331@alicia:/home/server$ sudo docker build -t server1 .
[sudo] пароль для zaa201-331:
Sending build context to Docker daemon 4.718MB
Step 1/12 : FROM ubuntu:20.04
--> a0ce5a295b63
Step 2/12 : RUN apt update
--> Using cache
--> de7861c4e47e
Step 3/12 : RUN apt install qt5-qmake -y
--> Using cache
--> 973059349698
Step 4/12 : RUN apt install cmake -y
--> Using cache
--> 21ecc49997d7

Step 11/12 : RUN make install
--> Running in 1b09049e7db1
/usr/lib/qt5/bin/qmake -install qinstall -exe server1 /opt/server1/bin/server1
strip /opt/server1/bin/server1
Removing intermediate container 1b09049e7db1
--> da84db470ab5
Step 12/12 : CMD ["./server1"]
--> Running in c38a2bed9c9a
Removing intermediate container c38a2bed9c9a
--> 13fc73cef418
Successfully built 13fc73cef418
Successfully tagged server1:latest
```

Запустим контейнер из полученного образа:

```
zaa201-331@alicia:/home/server$ sudo docker run -p 33333:33333 server1
server is started
```

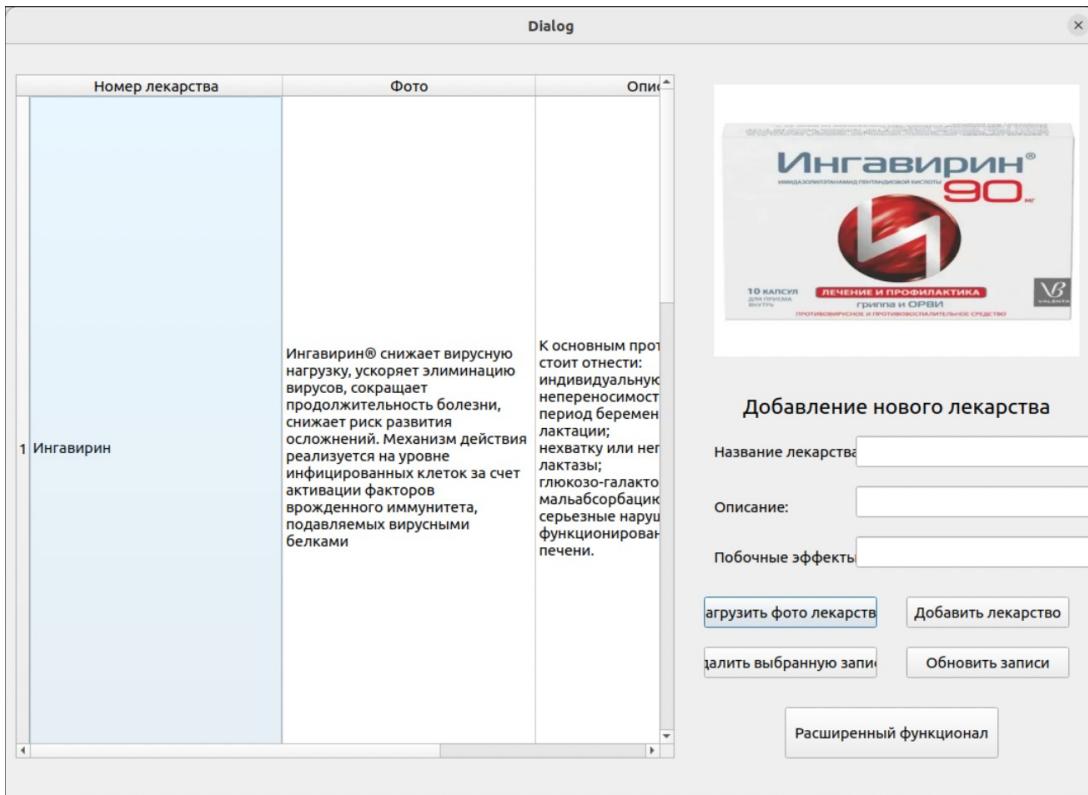
Попробуем авторизоваться:



Авторизация на сервере внутри контейнера успешна:

```
zaa201-331@alicia:/home/server$ sudo docker run -p 33333:33333 server1
server is started
"auth&ivan&Qwetty123!"
db is open
 QSqlQuery::value: not positioned on a valid record
login = "ivan" password = "Qwetty123!" result = "authorization error "
QSqlDatabasePrivate::addDatabase: duplicate connection name 'qt_sql_default_connection', old connection removed.
db is open
 QSqlQuery::value: not positioned on a valid record
"auth&ivan&Qwerty123!"
QSqlDatabasePrivate::addDatabase: duplicate connection name 'qt_sql_default_connection', old connection removed.
db is open
login = "ivan" password = "Qwerty123!" result = "authorization yes "
QSqlDatabasePrivate::addDatabase: duplicate connection name 'qt_sql_default_connection', old connection removed.
db is open
"Client is disconnected \n"
"Client is disconnected \n"
```

Открылась страница приложения после авторизации:



## Часть 2

### 1. Установить на все ВМ утилиту docker-compose

Установим docker-compose:

```
zaa201-331@alicia:~$ docker compose version
Docker Compose version v2.11.2
```

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.)

В качестве СУБД выберем sqlite3 и получим с Dockerhub ее образ:

```
zaa201-331@alicia:/home/server$ sudo docker pull keinos/sqlite3:latest
[sudo] пароль для zaa201-331:
latest: Pulling from keinos/sqlite3
213ec9aee27d: Pull complete
9d36b1e43f90: Pull complete
a647e5371462: Pull complete
4acf02aea91: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:fa7330d0beb7d6844f327ce5a0a8bf0aa2d722df1e0003d68870d367df1373
Status: Downloaded newer image for keinos/sqlite3:latest
docker.io/keinos/sqlite3:latest
```

Выведем версию sqlite3 из полученного контейнера:

```
zaa201-331@alicia:/home/server$ docker run --rm keinos/sqlite3 sqlite3 --version
3.39.4 2022-09-29 15:55:41 a29f9949895322123f7c38fbe94c649a9d6e6c9cd0c3b41c96d69
4552f26b309
```

Выведем данные из таблицы authorization нашей БД через контейнер:

```
zaa201-331@alicia:/home/server$ docker run --rm -it -v "$(pwd):/workspace" keino
s/sqlite3 sqlite3 /workspace/photolife.db -header -column 'SELECT * FROM autho
rization'
login    password      id  fullname
-----  -----
ivan     Qwerty123!   1   Иваныч
fedor   Asdfg123!    2   Феодосиф
```

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.

Перейдем в корень проекта:

```
zaa201-331@alicia:~$ cd /home/server
zaa201-331@alicia:/home/server$ nano docker-compose.yaml
```

Создадим docker-compose:

```
GNU nano 6.2                               docker-compose.yaml *
```

```
version: '3'

services:
  sqlite3: #Имя сервиса
    stdin_open: true
    tty: true
    image: keinos/sqlite3 #Образ из которого запускается контейнер
    restart: unless-stopped
  server:
    container_name: server_new
    hostname: server
    build:
      context: /home/server
      dockerfile: Dockerfile
    ports:
      - 33333:33333
    restart: always
    depends_on:
      - sqlite3
```

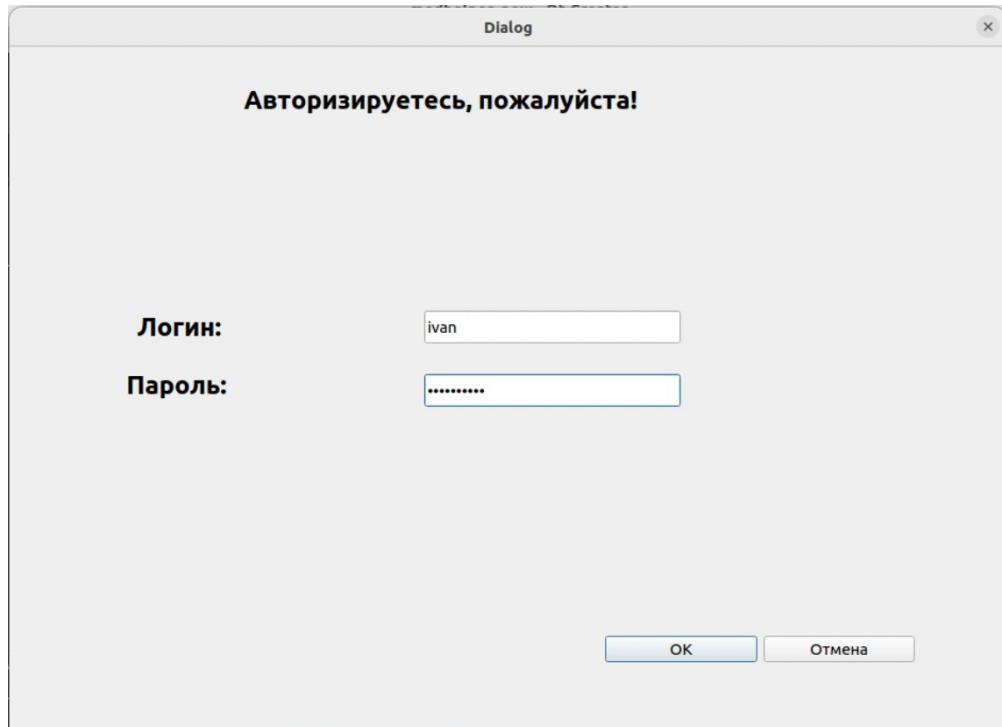
Запустим docker compose:

```
zaa201-331@alicia:/home/server$ docker compose up
[+] Building 41.3s (7/15)
=> [internal] load build definition from Dockerfile          0.0s
=> => transferring dockerfile: 299B                         0.0s
=> [internal] load .dockerignore                            0.0s
=> => transferring context: 2B                           0.0s
=> [internal] load metadata for docker.io/library/ubuntu:20.04 0.0s
=> [ 1/11] FROM docker.io/library/ubuntu:20.04            0.0s
=> [internal] load build context                          0.2s
=> => transferring context: 4.71MB                      0.1s
=> [ 2/11] RUN apt update                                9.9s
=> [ 3/11] RUN apt install qt5-qmake -y                  5.3s
=> [ 4/11] RUN apt install cmake -y                     25.9s
=> => # Selecting previously unselected package librhash0:amd64.
=> => # Preparing to unpack .../45-librhash0_1.3.9-1_amd64.deb ...
=> [ 5/11] Unpacking librhash0:amd64 (1.3.9-1)           0.0s
=> => Extracting dependency information...                0.0s
=> [ 6/11] Unpacking cmake (3.14.7-1)                    0.0s
=> => Extracting dependency information...                0.0s
=> [ 7/11] Unpacking qt5-qmake (5.14.1-1)               0.0s
=> => Extracting dependency information...                0.0s
=> [ 8/11] Setting environment for cmake...
=> [ 9/11] Setting environment for qt5-qmake...
=> [10/11] Setting environment for apt...
=> [11/11] RUN make install                            0.5s
=> exporting to image                                 7.4s
=> => exporting layers                             7.4s
=> => writing image sha256:f58c6d6f67e7b988cb85bb6a1a33f320d48699c47cee2 0.0s
=> => naming to docker.io/library/server-sever       0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
[+] Running 2/2
  :: Network server_default  Created                                0.1s
  :: Container server        Created                                0.2s
Attaching to server
server | server is started
```

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

Попробуем авторизоваться в приложении:



Авторизация успешна:

```
zaa201-33i@alicia:/home/server$ docker compose up
[+] Running 2/0
  :: Container server-sqlite3-1  Running                               0.0s
  :: Container server_new        Running                               0.0s
Attaching to server-sqlite3-1, server_new
server_new      | "Client is disconnected \n"
server_new      | "auth&ivan&Qwerty123!"
server_new      | QSqlDatabasePrivate::addDatabase: duplicate connection name
'qt_sql_default_connection', old connection removed.
server_new      | db is open
server_new      | login = "ivan" password = "Qwerty123!" result = "authoriz
ation yes "
server_new      | QSqlDatabasePrivate::addDatabase: duplicate connection name
'qt_sql_default_connection', old connection removed.
server_new      | db is open
```

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

Добавим еще одного пользователя в БД:

```
zaa201-331@alicia:~$ cd /home/server
zaa201-331@alicia:/home/server$ docker run --rm -it -v "$(pwd):/workspace" -w /workspace keinos/sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open ./photolife.db
sqlite> INSERT INTO authorization (login, password, id, fullname) VALUES ('elya', '123', 3, 'Alicia');
sqlite> select * from authorization;
ivan|Qwerty123!|1|Иваныч
fedor|Asdfg123!|2|Феодосий
elya|123|3|Alicia
sqlite> █
```

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

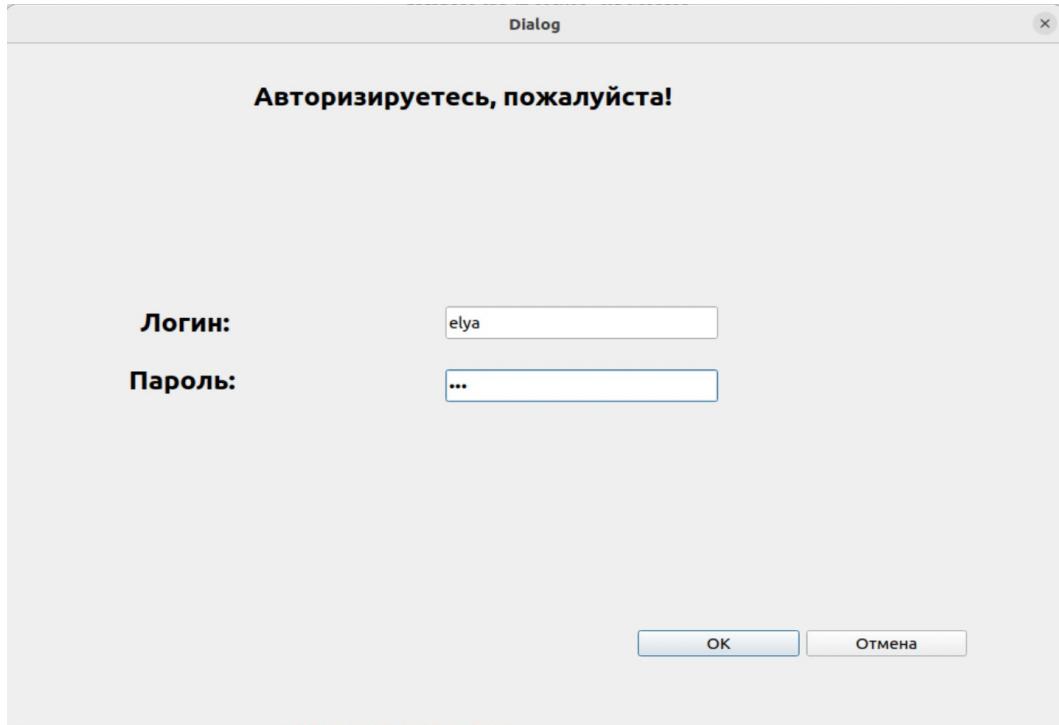
Остановим docker compose:

```
zaa201-331@alicia:/home/server$ docker compose down
[+] Running 3/3
  :: Container server_new      Removed          10.3s
  :: Container server-sqlite3-1 Removed          10.2s
  :: Network server_default   Removed          0.1s
zaa201-331@alicia:/home/server$ █
```

Запустим снова docker compose:

```
zaa201-331@alicia:/home/server$ docker compose up
[+] Running 3/3
  :: Network server_default    Created          0.1s
  :: Container server-sqlite3-1 Created          0.1s
  :: Container server_new      Created          0.0s
Attaching to server-sqlite3-1, server_new
server-sqlite3-1  | SQLite version 3.39.4 2022-09-29 15:55:41
server-sqlite3-1  | Enter ".help" for usage hints.
server-sqlite3-1  | Connected to a transient in-memory database.
server-sqlite3-1  | Use ".open FILENAME" to reopen on a persistent database.
server_new        | server is started
zaa201-331@alicia:/home/server$ █
```

Попробуем авторизоваться с данными нового пользователя:



The application interface shows a table of medications and a detailed view for 'Ингавирин'.

Номер лекарства	Фото	Опис
1 Ингавирин		<p>Ингавирин® снижает вирусную нагрузку, ускоряет элиминацию вирусов, сокращает продолжительность болезни, снижает риск развития осложнений. Механизм действия реализуется на уровне инфицированных клеток за счет активации факторов врожденного иммунитета, подавляемых вирусными белками</p> <p>К основным противопоказаниям относят: индивидуальную непереносимость периода беременности; недостаток или неглутказы; глюкозо-галакто мальабсорбацию; серьезные нарушения функционирования печени.</p>

On the right, a detailed view of 'Ингавирин 90 mg' is shown with its logo and text: 'Ингавирин 90 mg', 'ИНГАВИРИН', '90 mg', 'ЛЕЧЕНИЕ И ПРОФИЛАКТИКА', 'гриппа и ОРВИ', 'противовирусное и противовоспалительное средство', 'VALANTIA'.

**Добавление нового лекарства**

Fields for adding a new medicine: 'Название лекарства', 'Описание:', 'Побочные эффекты'.

Buttons: 'загрузить фото лекарства', 'Добавить лекарство', 'удалить выбранную запись', 'Обновить записи', 'Расширенный функционал'.

Авторизация успешна:

```
zaa201-331@alicia:/home/server$ docker compose up
[+] Running 3/3
  ● Network server_default      Created                         0.1s
  ● Container server-sqlite3-1  Created                         0.1s
  ● Container server_new        Created                         0.0s
Attaching to server-sqlite3-1, server_new
server-sqlite3-1  | SQLite version 3.39.4 2022-09-29 15:55:41
server-sqlite3-1  | Enter ".help" for usage hints.
server-sqlite3-1  | Connected to a transient in-memory database.
server-sqlite3-1  | Use ".open FILENAME" to reopen on a persistent database.
server_new       | server is started
server_new       | "auth&elya&123"
server_new       | db is open
server_new       | login = "elya" password = "123" result = "authorization y
es "
server_new       | QSqlDatabasePrivate::addDatabase: duplicate connection name
'qt_sql_default_connection', old connection removed.
server_new       | db is open
server_new       | "Client is disconnected \n"
```

### Вопросы для изучения

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

Приложение, запущенное в контейнере думает, что оно ...

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

docker run -p 127.0.0.1:80:80

Это эквивалентно режиму ...

**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?**

В данном случае необходимо ввести следующую последовательность команд ...