Задание: необходимо создать Dockerfile, основанный на любом образе (вы в праве выбрать самостоятельно).  
В него необходимо поместить приложение, написанное на любом известном вам языке программирования (Python, Java, C, С#, C++).  
При запуске контейнера должно запускаться самостоятельно написанное приложение.

Семинар 4

## Решение

Контейнер был собран на базе **Alpine Linux** и **Mariadb 10.6.12**.

Также был написан sql-файл для добавления пользователей с доступом по сети.

Привожу содержимое Dockerfile, скриптов и терминалов, на последних страницах скриншоты с БД.

Содержимое Dockerfile:

FROM alpine:latest

# последняя версия дистрибутива алпайн

COPY ./entry.sh /base\_entry/entry.sh

# копирование в контейнер скрипта для точки входа

COPY ./initial\_script.sql /base\_entry/initial\_script.sql

# копирование в контейнер sql-скрипта для создания пользователей

# с доступом по сети

RUN apk update && apk --no-cache add mariadb mariadb-client mariadb-server-utils bash && \

mkdir -p /auth\_pam\_tool\_dir/auth\_pam\_tool && \

chmod 777 /base\_entry/entry.sh

# запуск пакетного менеджера алпайн с командой обновления и

# затем установки необходимых компонентов БД, так же добавлен bash

# изменение атрибутов файла точки входа

ENTRYPOINT ["/base\_entry/entry.sh"]

# назначение точки входа

Содержимое initial\_script.sql:

USE mysql;

# переключение на служебную БД

CREATE USER 'root'@'172.17.%' IDENTIFIED BY 'toor';

# создание пользователя root с доступом по сети с адресов из подсети

# докера

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'172.17.%';

# выдать права новому суперпользователю

CREATE USER user123@'172.17.%' IDENTIFIED BY 'toor';

# создание безвредного пользователя с доступом по сети с адресов из

# подсети докера

GRANT ALL PRIVILEGES ON \*.\* TO user123@'172.17.%';

# назначение прав пользователя с доступом по сети с адресов из

# подсети докера

FLUSH PRIVILEGES;

# записать права

Содержимое файла entry.sh

#! /bin/bash

mkdir -p /run/mysqld/ && chown -R mysql:mysql /run/mysqld/

# создать директорию для размещения сокета и выдать права доступа

# системному пользователю mysql

if [ ! -d /var/lib/mysql/mysql ]; then

# проверка, существует ли директория на случай если подключен внешний

# том с БД

echo 'creating and filling directories'

mkdir -p /var/lib/mysql && chown -R mysql:mysql /var/lib/mysql

mkdir -p /var/log/mysql && \

chown -R mysql:mysql /var/log/mysql

mariadb-install-db --user=mysql --basedir=/usr --datadir=/var/lib/mysql

# создание директорий, необходимых для работы БД и заполнение их

# утилитой из дистрибутива БД

/usr/bin/mariadbd -uroot &

# временно запустить демон БД для выполнения скрипта (требуется сокет)

sleep 10

# без этого не работало.

/usr/bin/mariadb -uroot < /base\_entry/initial\_script.sql

# передать в базу sql-скрипт

/usr/bin/mysqladmin shutdown -uroot -ptoor

# погасить временный демон

else

echo 'directories are found'

fi

mariadbd --user=root --skip-name-resolve=ON --port=3306 --skip-networking=OFF

# окончательная команда запуска скрипта

Опции:

--skip-name-resolve=ON -- отключить разрешение доменных имен

--port=3306 -- назначение номера порта для работы по TCP

--skip-networking=OFF -- разрешить доступ по сети

Команда для сборки контейнера с БД:

docker build -t hw04\_mar-alp4 .

Команда для запуска контейнера с БД:

docker run -d --name hw04\_mar\_alp \

запуск в режиме сервиса, назначение имени контейнера

-v /home/HomeWork04/fourth/hw04\_localdb/:/var/lib/mysql:rw \

смонтировать внешнюю директорию с готовой БД, или директорию где БД будет создана средствами СУБД.

-p 3360:3306 hw04\_mar-alp4

проброс порта для доступа к БД

Команда для запуска PhpMyAdmin:

docker run --name hw04\_my-phpmyadmin -d \

запустить приложение в режиме сервиса, назначить имя контейнеру

--link hw04\_mar\_alp:db \

связать с контейнером, в котором запущена БД

-p 8081:80 phpmyadmin/phpmyadmin

проброс порта для доступа к приложению

Команда для подключения к БД через CLI:

mysql -h 172.17.0.1 -P 3360 -u user123 -p

-h -- адрес хоста, в данном случае контейнера

-u -- имя юзера -P -- порт -p -- запрос пароля

