

Задание: необходимо собрать образ и запустить из него контейнер.

- 1) основой образа должна быть alpine
- 2) установить необходимо mariaDB
- 3) уменьшить размер образа (способ обсуждался на лекции)
- 4) необходимо открыть порт для коммуникации с другими сущностями
- 5) для проверки решения необходимо подключить к такому контейнеру phpmyadmin (нужно, чтобы в нем вы увидели данные из вашей БД)
- 6) необходимо смонтировать внешнюю папку для хранения данных БД вне контейнера

- 1) `sudo -i`
- 2) `cd ..`
- 3) `mkdir GB && cd GB`
- 4) `nano Dockerfile`

```
GNU nano 6.4
FROM alpine

RUN apk add --no-cache mariadb mariadb-client mariadb-server-utils pwgen && \
    rm -f /var/cache/apk/*

ADD files/run.sh /scripts/run.sh
RUN mkdir /docker-entrypoint-initdb.d && \
    mkdir /scripts/pre-exec.d && \
    mkdir /scripts/pre-init.d && \
    chmod -R 755 /scripts

EXPOSE 3306

VOLUME ["/GB/var/lib/mysql"]

ENTRYPOINT ["/scripts/run.sh"]
```

- 5)
- 6) `mkdir -p /GB/var/lib/mysql/`
- 7) `mkdir files`
- 8) `nano files/run.sh`

```

GNU nano 6.4 files/r
#!/bin/sh

# execute any pre-init scripts
for i in /scripts/pre-init.d/*sh
do
    if [ -e "${i}" ]; then
        echo "[i] pre-init.d - processing $i"
        . "${i}"
    fi
done

if [ -d "/run/mysqld" ]; then
    echo "[i] mysqld already present, skipping creation"
    chown -R mysql:mysql /run/mysqld
else
    echo "[i] mysqld not found, creating..."
    mkdir -p /run/mysqld
    chown -R mysql:mysql /run/mysqld
fi

if [ -d /GB/var/lib/mysql/mysql ]; then
    echo "[i] MySQL directory already present, skipping creation"
    chown -R mysql:mysql /var/lib/mysql
else
    echo "[i] MySQL data directory not found, creating initial DBs"

    chown -R mysql:mysql /var/lib/mysql

    mysql_install_db --user=mysql --ldata=/var/lib/mysql > /dev/null

    if [ "$MYSQL_ROOT_PASSWORD" = "" ]; then
        MYSQL_ROOT_PASSWORD=$(pwgen 16 1)
        echo "[i] MySQL root Password: $MYSQL_ROOT_PASSWORD"
    fi

    MYSQL_DATABASE=${MYSQL_DATABASE:-""}
    MYSQL_USER=${MYSQL_USER:-""}
    MYSQL_PASSWORD=${MYSQL_PASSWORD:-""}

    tfile=$(mktemp)
    if [ ! -f "$tfile" ]; then
        return 1
    fi

    cat << EOF > $tfile
USE mysql;
FLUSH PRIVILEGES ;
GRANT ALL ON *.* TO 'root'@'%' identified by '$MYSQL_ROOT_PASSWORD' WITH GRANT OPTION ;

```

9)

```

GNU nano 6.4 files/run.sh
FLUSH PRIVILEGES ;
GRANT ALL ON *.* TO 'root'@'%' identified by 'MYSQL_ROOT_PASSWORD' WITH GRANT OPTION ;
GRANT ALL ON *.* TO 'root'@'localhost' identified by 'MYSQL_ROOT_PASSWORD' WITH GRANT OPTION ;
SET PASSWORD FOR 'root'@'localhost' PASSWORD 'MYSQL_ROOT_PASSWORD';
DROP DATABASE IF EXISTS test ;
FLUSH PRIVILEGES ;
EOF

if [ "$MYSQL_DATABASE" != "" ]; then
echo "[i] Creating database: $MYSQL_DATABASE"
if [ "$MYSQL_CHARSET" != "" ] && [ "$MYSQL_COLLATION" != "" ]; then
echo "[i] with character set [$MYSQL_CHARSET] and collation [$MYSQL_COLLATION]"
echo "CREATE DATABASE IF NOT EXISTS \"\$MYSQL_DATABASE\" CHARACTER SET \$MYSQL_CHARSET COLLATE \$MYSQL_COLLATION;" >> \$tfile
else
echo "[i] with character set: 'utf8' and collation: 'utf8_general_ci'"
echo "CREATE DATABASE IF NOT EXISTS \"\$MYSQL_DATABASE\" CHARACTER SET utf8 COLLATE utf8_general_ci;" >> \$tfile
fi

if [ "$MYSQL_USER" != "" ]; then
echo "[i] Creating user: $MYSQL_USER with password $MYSQL_PASSWORD"
echo "GRANT ALL ON \"\$MYSQL_DATABASE\".* to '$MYSQL_USER'@'%' IDENTIFIED BY '$MYSQL_PASSWORD';" >> \$tfile
fi

/usr/bin/mysqld --user=mysql --bootstrap --verbose=0 --skip-name-resolve --skip-networking=0 < \$tfile
rm -f \$tfile

for f in /docker-entrypoint-initdb.d/*; do
case "$f" in
*.sql) echo "$0: running $f"; /usr/bin/mysqld --user=mysql --bootstrap --verbose=0 --skip-name-resolve --skip-networking=0 < "$f"; echo ;;
*.sql.gz) echo "$0: running $f"; gunzip -c "$f" | /usr/bin/mysqld --user=mysql --bootstrap --verbose=0 --skip-name-resolve --skip-networking=0 < "$f"; echo ;;
*) echo "$0: ignoring or entrypoint initdb empty $f" ;;
esac
done
echo
echo 'MySQL init process done. Ready for start up.'
echo

echo "exec /usr/bin/mysqld --user=mysql --console --skip-name-resolve --skip-networking=0" "$@"
fi

# execute any pre-exec scripts
for i in /scripts/pre-exec.d/*.sh
do
if [ -e "$i" ]; then
echo "[i] pre-exec.d - processing $i"
. $i
fi
done

exec /usr/bin/mysqld --user=mysql --console --skip-name-resolve --skip-networking=0 $@

```

- 10)
- 11) docker build -t alpine\_mariadb .

```

root@ubuntu-server:/GB# docker build -t alpine_mariadb .
[+] Building 1.1s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 412B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/alpine:3.17.0
=> [1/4] FROM docker.io/library/alpine:3.17.0@sha256:8914eb54f968791faf6a8638949e480fef81e697984fba772b3976835194c6d4
=> [internal] load build context
=> => transferring context: 3.21kB
=> CACHED [2/4] RUN apk add --no-cache mariadb mariadb-client mariadb-server-utils pwgen & rm -f /var/cache/apk/*
=> CACHED [3/4] ADD files/run.sh /scripts/run.sh
=> CACHED [4/4] RUN mkdir /docker-entrypoint-initdb.d & mkdir /scripts/pre-exec.d & mkdir /scripts/pre-init.d & chmod -R 755 /scripts
=> exporting to image
=> => exporting layers
=> => writing image sha256:7ee630a4634a8521df1474d861478c481398f1279e88c0125ba480c280268e7f
=> => naming to docker.io/library/alpine_mariadb
root@ubuntu-server:/GB#

```

- 12)
- 13) docker run -d --name maria\_db -p 3306:3306 -v /GB/var/lib/mysql:/var/lib/mysql -e MYSQL\_DATABASE=db\_test -e MYSQL\_USER=geek -e MYSQL\_PASSWORD=123 -e MYSQL\_ROOT\_PASSWORD=geek123 alpine\_mariadb

```

root@ubuntu-server:~# docker run -d --name maria_db -p 3306:3306 -v /GB/var/lib/mysql:/var/lib/mysql -e MYSQL_DATABASE=db_test -e MYSQL_USER=geek -e MYSQL_PASSWORD=123 -e MYSQL_ROOT_PASSWORD=geek123 alpine_mariadb
3bd12df4108969accf45c0dc98516eab625c24626935ba346afa269325f17570
root@ubuntu-server:~# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
3bd12df41089   alpine_mariadb  "/scripts/run.sh"        4 seconds ago Up 3 seconds  0.0.0.0:3306->3306/tcp, :::3306->3306/tcp  maria_db

```

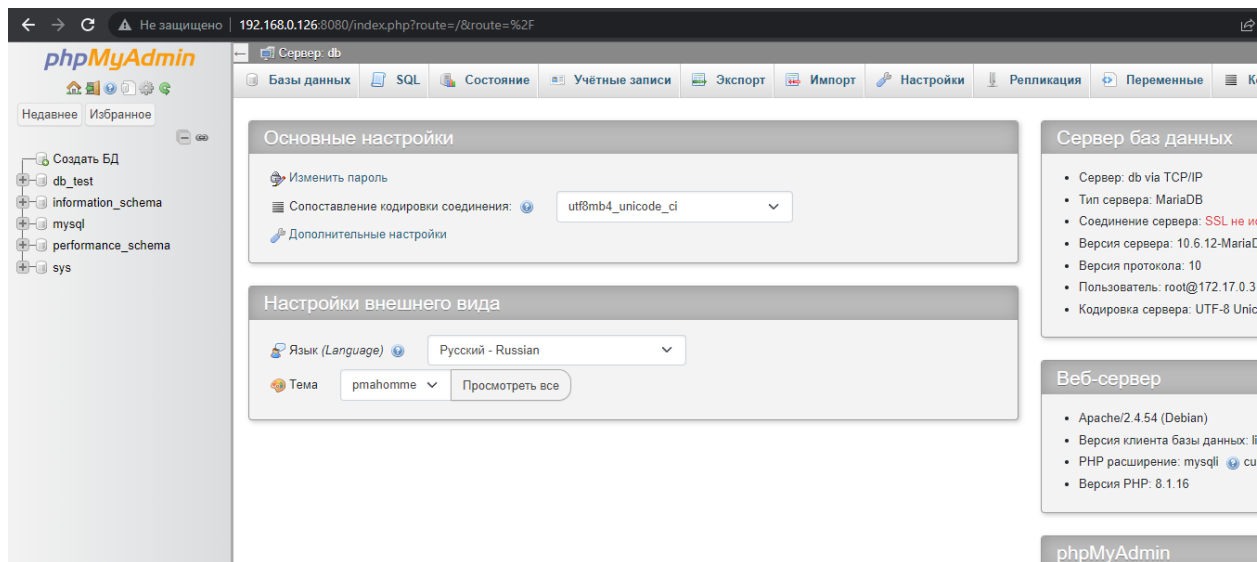
- 14)
- 15) docker run --name phpmyadmin -d --link maria\_db:db -p 8080:80 phpmyadmin

```

root@ubuntu-server:~# docker run --name phpmyadmin -d --link maria_db:db -p 8080:80 phpmyadmin
41d8fc9ced4a154dc7b475bb9eec2ffc54a6d3320232ff0a5d19f0ebe55d947f
root@ubuntu-server:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
41d8fc9ced4a   phpmyadmin  "/docker-entrypoint..."  8 seconds ago Up 7 seconds  0.0.0.0:8080->80/tcp, :::8080->80/tcp  phpmyadmin
3bd12df41089   alpine_mariadb  "/scripts/run.sh"        3 minutes ago Up 3 minutes  0.0.0.0:3306->3306/tcp, :::3306->3306/tcp  maria_db
root@ubuntu-server:~#

```

- 16)



17)

18) cd /GB/var/lib/mysql/

```

root@ubuntu-server:~# cd /GB/var/lib/mysql/
root@ubuntu-server:/GB/var/lib/mysql# ll
total 123344
drwxr-xr-x 6 _apt systemd-journal 4096 Mar 13 19:28 ./
drwxr-xr-x 3 root root 4096 Mar 13 19:21 ../
-rw-rw---- 1 _apt systemd-journal 2 Mar 13 19:28 3bd12df41089.pid
-rw-rw---- 1 _apt systemd-journal 417792 Mar 13 19:27 aria_log.00000001
-rw-rw---- 1 _apt systemd-journal 52 Mar 13 19:27 aria_log_control
drwx----- 2 _apt systemd-journal 4096 Mar 13 19:25 db_test/
-rw-rw---- 1 _apt systemd-journal 9 Mar 13 19:28 ddl_recovery.log
-rw-r----- 1 _apt systemd-journal 946 Mar 13 19:27 ib_buffer_pool
-rw-rw---- 1 _apt systemd-journal 12582912 Mar 13 19:27 ibdata1
-rw-rw---- 1 _apt systemd-journal 100663296 Mar 13 19:30 ib_logfile0
-rw-rw---- 1 _apt systemd-journal 12582912 Mar 13 19:28 ibtmp1
-rw-rw---- 1 _apt systemd-journal 0 Mar 13 19:25 multi-master.info
drwx----- 2 _apt systemd-journal 4096 Mar 13 19:25 mysql/
-rw-r--r-- 1 _apt systemd-journal 15 Mar 13 19:25 mysql_upgrade_info
drwx----- 2 _apt systemd-journal 4096 Mar 13 19:25 performance_schema/
drwx----- 2 _apt systemd-journal 12288 Mar 13 19:25 sys/
root@ubuntu-server:/GB/var/lib/mysql#

```

19)