Dockerfile run script

dockerfile

/via:https://github.com/chio-nzgft/docker-MariaDB-with-SSL/blob/master/run.sh

destacable

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

completo

```
#!/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 /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 > /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 > $tfile
USE mysql;
FLUSH PRIVILEGES;
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' identified by '$MYSQL_ROOT_PASSWORD' WITH
GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' identified by
'MYSQL_ROOT_PASSWORD' WITH GRANT OPTION;
UPDATE user SET password=PASSWORD("") WHERE user='root' AND host='localhost';
EOF

        if [ "$MYSQL_DATABASE" != "" ]; then
            echo "[i] Creating database: $MYSQL_DATABASE"
            echo "CREATE DATABASE IF NOT EXISTS \`$MYSQL_DATABASE\` CHARACTER SET
utf8 COLLATE utf8_general_ci;" >> $tfile

            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
        fi

        /usr/bin/mysqld --user=mysql --bootstrap --verbose=0 < $tfile
        rm -f $tfile
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

cp /etc/newcerts/server-key.pem /etc/mysql/server.key
cp /etc/newcerts/server-cert.pem /etc/mysql/server.crt
cp /etc/newcerts/ca-cert.pem /etc/mysql/CA.crt

if [ "$SERVER_KEY" ]; then
  echo $SERVER_KEY | sed "s/\\$/\n/g" | sed "s/^ //g" >/etc/mysql/server.key
  echo $SERVER_CERT | sed "s/\\$/\n/g" | sed "s/^ //g" >/etc/mysql/server.crt
  echo $CA_CERT | sed "s/\\$/\n/g" | sed "s/^ //g" >/etc/mysql/CA.crt
  export MYSQLD_SSL_KEY=/etc/mysql/server.key
```

```
  export MYSQLD_SSL_CERT=/etc/mysql/server.crt
  export MYSQLD_SSL_CA=/etc/mysql/CA.crt
fi

if  [ ! -e "/etc/mysql/my1.cnf" ]; then
  sed 's/\[mysqld\]/\[mysqld\]\n\ssl-key=\/etc\/mysql\/server.key/g'
/etc/mysql/my.cnf > /etc/mysql/my1.cnf
  sed 's/\[mysqld\]/\[mysqld\]\n\ssl-cert=\/etc\/mysql\/server.crt/g'
/etc/mysql/my1.cnf > /etc/mysql/my.cnf
  sed 's/\[mysqld\]/\[mysqld\]\n\ssl-ca=\/etc\/mysql\/CA.crt/g' /etc/mysql/my.cnf >
/etc/mysql/my1.cnf
  cp -rf /etc/mysql/my1.cnf /etc/mysql/my.cnf
fi

exec /usr/bin/mysqld --user=mysql --console
```