

Setup Kodi to use (MariaDB) SQL Database running in Docker Container

I recently had to rebuild my Kodi setup and found that Kodi says to "GRANT ALL ON *.* to KODI"!!!! That is the most insecure thing possible, so I decided to use an Docker MariaDB Container for my SQL Backend.

Note: Because Kodi also says not to create the DB's and I'm using a standalone SQL server I use the root account in Kodi and non-standard SQL Ports.

Docker Run

```
WD=/opt/kodisql
mkdir -p $WD/{sql_data,mnt,setup}
cd $WD/setup
cat << 'EOF' >kodi_sql_docker.sh
docker run -d --name="kodi-sql" \
-p 3506:3306 \
-e TZ="America/Whitehorse" \
-e MYSQL_ROOT_PASSWORD=kodi \
-v "/opt/kodisql/sql_data":"/var/lib/mysql" \
-v "/opt/kodisql/mnt":"/mnt" \
mariadb:latest
EOF
chmod +x kodi_sql_docker.sh
./kodi_sql_docker.sh
```

Docker-Compose:

```
WD=/opt/kodisql
mkdir -p $WD/{mnt,setup}
cd $WD/setup
cat << 'EOF' >docker-compose.yaml
version: '3.7'

services:
  kodi-sql:
    container_name: kodi-sql
    hostname: kodi-sql
    image: mariadb
    ports:
      - '3506:3306'
    volumes:
      - type: bind
        source: /opt/kodisql/sql_data/
        target: /var/lib/mysql
      - type: bind
        source: /opt/kodisql/mnt/
        target: /mnt
    environment:
      - 'TZ=America/Whitehorse'
      - 'MYSQL_ROOT_PASSWORD=kodi'

EOF
chmod +x docker-compose.yaml
```

Kodi Setup:

advancedsettings.xml

Note: UNAME=root and "non-standard" port 3506

```

<advancedsettings>
  <videodatabase>
    <type>mysql</type>
    <host>192.168.1.1</host>
    <port>3506</port>
    <user>root</user>
    <pass>kodi</pass>
  </videodatabase>
  <musicdatabase>
    <type>mysql</type>
    <host>192.168.1.1</host>
    <port>3506</port>
    <user>root</user>
    <pass>kodi</pass>
  </musicdatabase>
  <videolibrary>
    <importwatchedstate>true</importwatchedstate>
    <importresumepoint>true</importresumepoint>
  </videolibrary>
</advancedsettings>

```

SystemD Integration:

Docker-Run:

```

cat << 'EOL' >/lib/systemd/system/kodisql.service
[Unit]
Description=KodiSQL_Docker_Container
Requires=docker.service network-online.target

[Service]
Restart=on-abnormal
ExecStart=/usr/bin/docker start -a kodi-sql
ExecStop=/usr/bin/docker stop -t 2 kodi-sql

[Install]
WantedBy=multi-user.target
EOL
systemctl enable kodisql
systemctl restart kodisql
systemctl status kodisql

```

Docker-Compose:

```

WD=/opt/kodisql/setup
cat << EOF >$WD/kodi-sql.service.setup.sh
cat << EOL >/lib/systemd/system/kodi-sql.service
[Unit]
Description=kodi-sql_Docker
Requires=docker.service network-online.target

[Service]

Restart=on-abnormal
ExecStart=/usr/bin/docker-compose --project-name kodi-sql --project-directory $WD -f $WD/docker-compose.yaml up
ExecStop=/usr/bin/docker-compose --project-name kodi-sql --project-directory $WD -f $WD/docker-compose.yaml stop

[Install]
WantedBy=multi-user.target
EOL
systemctl enable kodi-sql
systemctl start kodi-sql
EOF
chmod +x $WD/kodi-sql.service.setup.sh
$WD/kodi-sql.service.setup.sh

```

CLI Access:

```
echo 'alias kodiqlcli="docker exec -it kodi-sql /bin/bash"' >> ~/.bashrc  
source ~/.bashrc
```

Add to XYMon:

```
nano /etc/xymon/analysis.cfg && systemctl restart xymon xymon-client
```

```
PROC kodi-sql
```