DigitalSTROM MQTTG Gateway in a Docker container

this is WIP this is how i set up Chriss Gross' MQTT bridge to run inside a node docker container. i have also slightly modified his script to use authentication and to allow setting the qos in the config file as well. in addition to that i have written a wrapper script in bash that will constantly (as in every second) check when the last contact was established to the dss server. if that's too long ago, it will restart the bridge. this was necessary because otherwise the bridge script would not recognize when it had lost the connection to the dss server (i.e. because of a network outage or because the dss has been restarted or whatever other reason there might be) and it would need a manual intervention. this script fixed this issue for me pretty reliably.

so here is how i set it up:

docker-compose.yml

```
# create ./app directory with uid and gid 1000 first
services:
  node:
    build: ./build
    restart: always
    user: 1000:1000
    working_dir: /app
    environment:
      - NODE_ENV=production
    volumes:
      - ./app:/app
```

```
mkdir ./app
chown 1000.1000 app
mkdir ./build
```

build/startup.sh

```
#!/bin/bash
USERNAME="myuser"
PASSWORD="mysecret"
MQTT_SERVER="mqtt.psuter.ch"

if [ -z "$(ls -A /app)" ]; then
    echo "installing dss mqtt bridge (see
https://github.com/cgHome/mqtt-dss-bridge)"
    cd /app
    npm install --save mqtt-dss-bridge
    echo "IMPORTANT: configure the bridge by editing
/app/node_modules/mqtt-dss-bridge/config.js"
```

```
    echo "installation completed"
fi
echo "Starting DSS to MQTT bridge"
# start the bridge in the background
cd /app
node /app/node_modules/mqtt-dss-bridge/index.js 2>&1 &
PID=$!
echo "DSS Bridge started with PID $PID"
# start a loop to monitor the status every second
errcntr=0
lastUpdate=999
while true; do
    lastUpdate=$(( \
            $(date +%s)\
            -\
            $(date -d "$(\
                mosquitto_sub -h $MQTT_SERVER -u $USERNAME -P $PASSWORD
-t dss/state/lastDiscovered -C 1 | \
                sed -E -e 's/"([0-9]{4}-[0-9]{2}-
[0-9]{2})T([0-9]{2}:[0-9]{2}:[0-9]{2}).*$/\1 \2Z/'\
            )" +%s)\
        ))
    if [ 10 -lt $lastUpdate ]; then
            if [ $errcntr -lt 10 ]; then
                let errcntr++
                echo "DSS Bridge state is too old (last updated
$lastUpdate seconds ago, max 10 allowed). This was failure $errcntr out
of 10 accepted failures"
            else
                echo "DSS bridge state is still too old ($lastUpdate
seconds) after 10 retries. killing mqtt-dss-bridge (pid $PID) and exit
with error status 1"
                kill $PID
                exit 1
            fi
    fi
    sleep 1
done
```

obviously you need to adjust the three variables at the top of the script to match your setup

```
chmod 755 build/startup.sh
```

build/Dockerfile

```
FROM node:22-bookworm
ARG DEBIAN_FRONTEND=noninteractive
ENV TZ=Europe/Zurich
```

```
# mqtt client
RUN apt-get update && apt-get install -y mosquitto-clients
#RUN npm install --save mqtt-dss-bridge
COPY startup.sh /startup.sh
CMD /startup.sh
```

```
docker compose build
docker compose up
```

press `ctrl+C` to stop the app again, then edit the config file

```
nano ./app/node_modules/mqtt-dss-bridge/config.js
```

now your setup might be different, but i wanted to be able to use a password authentication for mqtt and i wanted to set `qos` to 2, by default this is hardcoded to 1.

so in the settings in the `var config=` section i edited the `client` attribute to look like this:

```
    // MQTT-Client
    client: {
        url: 'mqtt://mqtt.psuter.ch:1883',
        baseTopic: dssTopic,
        qos: 2,
        username: 'myuser',
        password: 'mysecret'
    },
```

again of course with the correct username and password.

next i had to slightly modify the script in `app/node_modules/mqtt-dss-bridge/lib/dss.js` starting from `line 23` i have added the `qos` setting as well as the `username` and `password` settings, so this section now looks like this:

```
var clientSettings = {
    keepalive: 10,
    clientId: config.name,
    will: {
        topic: config.client.baseTopic + '/state/online',
        payload: new Buffer('{"$value": "false"}'),
        qos: config.client.qos,
        retain: true
    },
    username: config.client.username,
    password: config.client.password
};
```

with all this done, we can now start the container and hope for it to run ;)

```
docker compose up -d
```

From:
http://wiki.psuter.ch/ - **pswiki**

Permanent link:
http://wiki.psuter.ch/doku.php?id=digitalstrom_mqtt_gateway_in_a_docker_container&rev=1720251777

Last update: **06.07.2024 09:42**