2.1.1.4 MQTT Broker

How It Works

Imagine your smart home devices (lights, thermostats, door locks, etc.) are family members who need to communicate. Instead of each device talking directly to every other device (which would be messy), they all use a central "message board" called an MQTT broker. When one device needs to send information or receive a command, it posts the message to this board, and other devices that care about that message can read it.

A Practical Example

Let's say you have:

- A smart light in your living room
- A motion sensor in your hallway
- Home Assistant (your smart home hub)

When the motion sensor detects movement, it doesn't directly tell the light to turn on. Instead, it posts a message saying "motion detected in hallway" to the central message board. Home Assistant reads this message, thinks "I have a rule that says turn on the living room light when motion is detected," and then posts a message saying "turn on living room light." The light reads this instruction and turns on.

Why It's Useful

- Devices don't need to know about each other — they just post and read messages
- It's lightweight — doesn't use much internet bandwidth or processing power
- It works reliably even if connections are slow — messages get delivered when devices reconnect
- You can automate things — Home Assistant acts as the "smart controller" that makes decisions based on the messages it receives

That's essentially what MQTT does in Home Assistant—it's the communication backbone that lets your devices talk to each other through a central hub.

Installation

Now, here we will assume that you have installed ha core via docker and not the HA OS. The HA OS mqtt installation is a lot simpler but you gain some and you loose some ;) So we will basically create another mqtt container that will serve as our mqtt broker.

Source: Setup MQTT broker in docker

The source from above explains everything pretty well but I want to add something to it that made the setup easier for me. That is simply setting the PUID and PGID to my user. By default the first user runs on PUID=PGID=1000 so we add this to the docker compose file.

But first assuming you use Dockge again we make the following folders and items before we deploy our compose files

```
mkdir -p /opt/stacks/mqtt/{config,data,log}
cd /opt/stacks/mqtt/
```

We will then create the docker compose file inside the mqtt folder

Compose file

```
services:
  mosquitto:
    container_name: mqtt
    image: eclipse-mosquitto:2
    environment:
      - PUID=1000
      - PGID=1000
    ports:
      # Standard MQTT protocol
      - 1883:1883
      # MQTT over WebSocket
      - 9001:9001
    volumes:
      # Mount configuration directory
      - ./config:/mosquitto/config
      # Persist message data
      - ./data:/mosquitto/data
      # Persist log files
      - ./log:/mosquitto/log
    restart: unless-stopped
    networks:
```

```
      mosquitto:

networks:
  mosquitto:
```

After that is done we will move inside config folder and create our mosquitto.conf file and add the following content to it

mosquitto.conf

```
# mosquitto-docker/config/mosquitto.conf - Basic Mosquitto configuration

# Listen on all interfaces for standard MQTT
listener 1883
protocol mqtt

# WebSocket listener on port 9001
listener 9001
protocol websockets

# Do not allow anonymous connections
allow_anonymous false
password_file /mosquitto/config/passwd

# Persistence settings - retain messages across restarts
persistence true
persistence_location /mosquitto/data/

# Log settings
log_dest file /mosquitto/log/mosquitto.log
log_dest stdout
log_type all
```

After that is done we will create the passwd file inside the config directory

```
touch passwd
```

Then before we deploy everything we need to make sure that all the files are owned by your user. To make sure of this we will use the following command

```
sudo chown user:user -R /opt
```

This recursively makes all items inside of /opt recursively ownership by PUID and GPID of "user". Finally make sure that the passwd has 700 as permissions. You can make sure of this by entering the config directory and use the command

```
chmod 700 passwd
```

If this is all done we can go back to our Dockge display and reload it. Now an mqtt container should be ready to be deployed. Just simply hit deploy and watch the logs.

If it is all up and running we need to make some users of the mosquitto broker. We can do this by entering the moquitto container with

```
docker exec -it mqtt sh
```

Then we use the following command to make the first user

```
# Create a password file with an initial user
mosquitto_passwd -c /mosquitto/config/passwd device1

# Add more users without the -c flag (which creates a new file)
mosquitto mosquitto_passwd /mosquitto/config/passwd device2
mosquitto mosquitto_passwd /mosquitto/config/passwd webapp
```

You can get this error message is your user is not root

> Warning: File /mosquitto/config/passwd owner is not root. Future versions will refuse to load this file.To fix this, use `chown root /mosquitto/config/passwd`

But this should be fine. Otherwise you can set it temporarily to root and change the ownership of passwd back to "user" after creation.

You will be prompted by setting up a password for the created users. Make sure you note them down since you will need it for authentication in HA and the apps that are using the MQTT broker.

Sweet!!! So now you finally have Mosquitto up and running.

Revision #8
Created 2026-06-09 18:45:17 UTC by dpasn
Updated 2026-06-10 01:19:44 UTC by dpasn