

Tutorial - How to get to "Viele MQTT-Clients"

Von Aron & Fabian

1. Neue Gruppe

- Aron ist neu mit Fabian
- Atahan ist neu mit Leon
- Pascal immer noch mit Burak
- Gruppe Devrim wie immer gleich

2. Ubuntu VM

Du hast eine stabile Ubuntu version heruntergeladen, wie Jammy Jellyfish (22.04 oder 22.03)
Du bringst diese auf den neusten stand mit:

```
sudo apt update
```

und

```
sudo apt upgrade
```

Die **GuestAdditions** muss du auch installiert haben
Tutorial dafür geb ich dir nicht (keine Zeit, es ist 11:15)

3. Basis Systeme #Wichtig

Die Systeme die wir brauchen um diese Applikation laufen lassen zu können, sind die fundamentalen Systeme, die sehr wichtig sind.

1. Docker

- *Docker ist ein System, welches man auf Betriebssystemen installieren kann*
- *Die wichtigste Funktion von Docker sind die Container*
- *In Containern, kann man Code aller Art ausführen (isoliert lol)*
- *Wir brauchen Docker hier, um einen MQTT-Broker aufzusetzen (google: Broker)*

1. Docker installieren

Diesen code ins Terminal kopieren wird docker installieren müssen

```
sudo apt-get update && sudo apt-get install -y apt-transport-https ca-
certificates curl software-properties-common && curl -fsSL
https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg && echo "deb [arch=amd64
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null && sudo apt-get update && sudo
apt-get install -y docker-ce docker-ce-cli containerd.io && sudo systemctl
status docker && sudo docker run hello-world
```

uhm, if not ask Chatgpt lol

2. MQTT installieren

MQTT (Message Queuing Telemetry Transport) ist ein Nachrichtenprotokoll für eingeschränkte Netzwerke mit geringer Bandbreite und IoT-Geräte mit extrem hoher Latenzzeit. MQTT funktioniert nach dem Publisher- / Subscriber-Prinzip und wird über einen zentralen Broker betrieben. Das bedeutet, dass Sender und Empfänger keine direkte Verbindung haben.

Das MQTT Image installieren

Hier ist der link, wenn du es selber machen wilt

https://hub.docker.com/_/eclipse-mosquitto/

```
sudo docker pull eclipse-mosquitto
```

Überprüfe mit `ls -l` , ob ein neuer Ordner erstellt worden ist mit namen mosquitto und ob dieser 3 weitere Ordner enthält.

```
/mosquitto/config  
/mosquitto/data  
/mosquitto/log
```

Falls es diesen Ordner nicht gibt, ganz simple: erstelle sie mit `mkdir folder_name` oder mit diesem Command:

```
mkdir -p /mosquitto/config /mosquitto/data /mosquitto/log && touch  
/mosquitto/config/mosquitto.conf
```

Wenn du die folders erstellt hast, kannst du diesen Kommand ausführen

```
sudo docker run -it -p 1883:1883 -p 9001:9001 -v  
mosquitto.conf:/mosquitto/config/mosquitto.conf eclipse-mosquitto
```

Diese sachen sollen in dein config file:

```
listener 1883  
allow_anonymous true  
persistence true  
persistence_location /mosquitto/data/  
log_dest file /mosquitto/log/mosquitto.log
```

oder dies:

```
persistence true  
persistence_location /mosquitto/data/  
log_dest file /mosquitto/log/mosquitto.log
```

Wenn du dieses neue file gespeichert hast kannst du jetzt diesen Komand ausführen

```
docker run -it -p 1883:1883 -p 9001:9001 -v  
mosquitto.conf:/mosquitto/config/mosquitto.conf -v /mosquitto/data -v  
/mosquitto/log eclipse-mosquitto
```

Wenn du andere Pfade hast, was wahrscheinlich ist, gehe zum Ordner und benutze:

```
realpath "File/OrdnerName"
```

ich habe auch das noch gefunden: muss angepasst werden

```
sudo docker run -it -p 1883:1883 -p 9001:9001 \
-v
/home/aron/mosquitto/config/mosquitto.conf:/mosquitto/config/mosquitto.conf \
-v /home/aron/mosquitto/data:/mosquitto/data \
-v /home/aron/mosquitto/log:/mosquitto/log \
eclipse-mosquitto
```

Loops mit MQTT

Jetzt läuft unser MQTT broker und wir können anfangen mit subscriben und publishen

Erster sub und pub

```
sudo apt install mosquitto-clients
```

```
mosquitto_sub -h 192.168.56.107 -p 1883 -t "sensor"
```

```
mosquitto_pub -h 192.168.56.107 -p 1883 -t "sensor" -m "Hello, MQTT"
```

Loopen von Publishen

Mache ein neues Directory mit namen mqtt-loops und mache zwei files mit namen "sensor1.sh" und "sensor2.sh"

Tue diesen Code rein ins "sensor1.sh":

```
while true; do

    mosquitto_pub -t sensoren/sensor1 -m $((1 + RANDOM % 10))

    sleep 1;

done
```

Diesen in "sensor2.sh":

```
while true; do

    mosquitto_pub -t sensoren/sensor2 -m $((10 + RANDOM % 10))

    sleep 1;

done
```

Dannach müssen wir sie beide ausführbar machen

```
chmod +x sensor1.sh && chmod +x sensor2.sh
```

Und natürlich ausführen:

```
./sensor1.sh & ./sensor2.sh &
```

Aufgepasst!

Jetzt haben wir zwei Dateien am ausführen, die Nachrichten Publishen. Um diese Nachrichten zu sehen muss der Broker laufen (Neuen code noch einfügen) und ein Sub muss laufen: den sub macht man so:

```
mosquitto_sub -v -t "name_of_the_publishSource"
```

Unser Publishing Source ist in diesem Fall in den zwei Dateien Definiert worden, sube zu einen von beiden ("sensoren/sensor1" und "sensoren/sensor2")

Grafana

*Grafana ist eine Open-Source-Plattform für Überwachung, **#Visualisierung** und Analyse von Metriken und Logs. Es bietet eine leistungsstarke und flexible Benutzeroberfläche, die es Benutzern ermöglicht, **#Dashboards** zu erstellen und anzupassen, die **#Daten** aus verschiedenen Quellen in **#Echtzeit** visualisieren können.*

Einfach Grafana Enterprise installieren mit diesem Query :)

```
sudo apt-get install -y apt-transport-https software-properties-common wget && \
sudo mkdir -p /etc/apt/keyrings/ && \
wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee
/etc/apt/keyrings/grafana.gpg > /dev/null && \
echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com
stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list && \
sudo apt-get update && \
sudo apt-get install grafana-enterprise
```

Oder selber machen von der Webseite:

<https://grafana.com/docs/grafana/latest/setup-grafana/installation/debian/>

Grafana Schritt 2

Um die Daten im Browser zu sehen, müssen wir noch das MQTT-Plugin installieren
Das machen wir mit diesem Command:

```
sudo systemctl stop grafana-server && sudo grafana-cli plugins install grafana-mqtt-datasource && sudo systemctl start grafana-server
```

- Jetzt kannst du auf **localhost:3000** in Firefox zugreifen (in VM) und Grafana konfigurieren
- Wenn es nicht geht führe den Kommand evt nochmal aus...
- Wenn du nach einer Pause nicht auf localhost:3000 zugreifen kannst, kann es sein, dass beim ausschalten der VM der Grafana Server sich ausgeschaltet hat. Starte ihn dan mit:

```
sudo systemctl start grafana-server
```

Grafana Deinstallieren (wenn du willst)

```
sudo systemctl stop grafana-server && sudo apt-get remove -y grafana-enterprise && sudo rm -i /etc/apt/sources.list.d/grafana.list
```

Grafana Login

Logge dich mit Username: **#admin** und Password: **#admin** ein. Ändere dein Passwort, wenn es Grafana beansprucht und kopiere das Passwort in die VM description oder woanders so dass du es nicht vergisst.

Grafana Dashboard

Das Dashboard brauchst du um die Daten zu visualisieren. Ich erkläre wie du eins erstellst:

1. Drücke auf die 3 Striche (Menu)
2. Drücke auf "Dashboards"
3. Drücke auf "Create new Dashboard"
4. Drücke auf "Add visualization"
5. Drücke auf "Configure a new data source"

6. Suche nach "MQTT" und drücke drauf

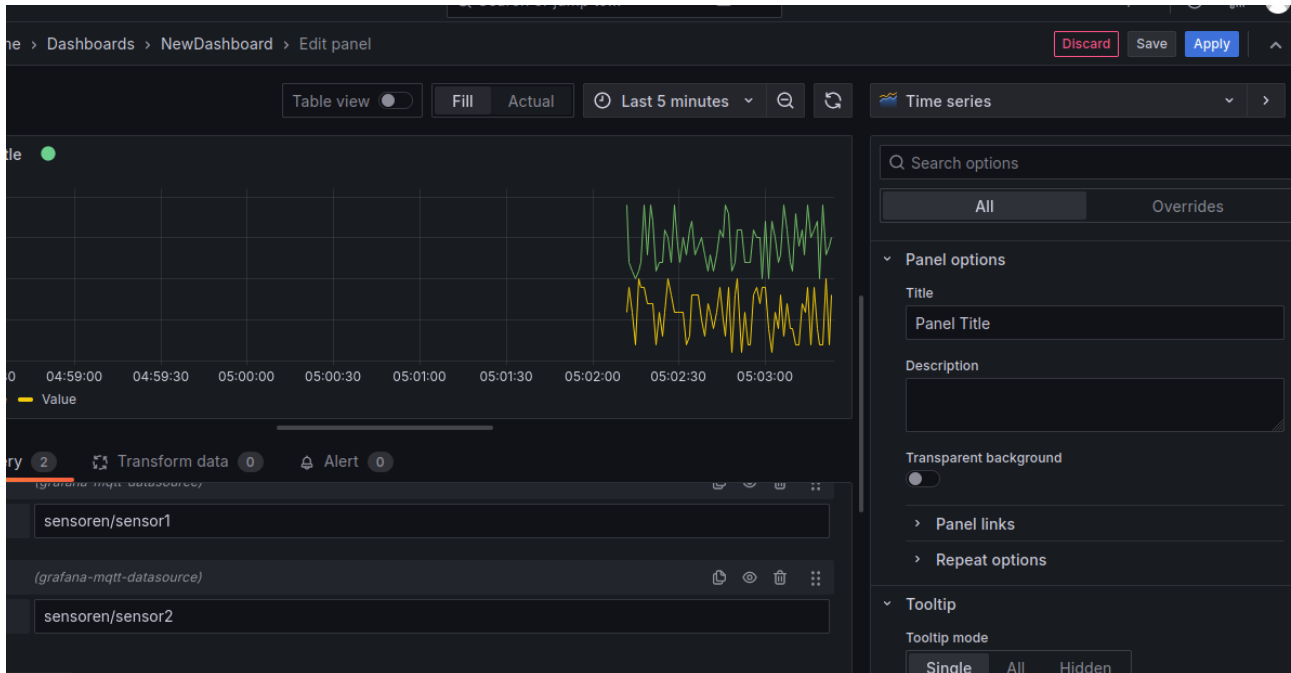
The screenshot shows the Grafana MQTT datasource configuration interface. At the top, the title is "grafana-mqtt-datasource". On the right, there are tabs for "Type" (MQTT), "Alerting" (Not supported), "Explore data", and "Build a dashboard". Below the title, it says "Type: MQTT". A "Settings" tab is selected. The "Name" field is "grafana-mqtt-datasource" and the "Default" toggle is turned on. The "Connection" section has a "URI" field set to "localhost:1883". The "Authentication" section has a "Username" field set to "admin" and a "Password" field set to "configured", with a "Reset" button. At the bottom, there are "Delete" and "Save & test" buttons.

7. Dann solltest du auf "Save & Test"drücken. Allfällig Daten anpassen

The screenshot shows a green confirmation message box with a checkmark icon. The text reads "MQTT Connected" and "Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#)."

8. Macht die gleichen Schritte um ein neues Dashoard zu erstellen, aber jetzt können wir bei Datasource die MQTT auswählen
9. Erst mal oben rechst auf Save drücken und dem Dashboard einen Namen geben
10. Bei edit im Dashboard kannst du deine Topics angeben wie "sensoren/sensor1" und das machen wir auch
11. Lass mal deine zwei sensoren laufen mit dem Befel von vorhin und schau, dass die die Gleichen Topic namen in Grafana eingibst

12. Resultat:



Machmal muss man auf den Refresh button Klicken oder in Terminal einen Sub machen um zu schauen ob e wirklich Sachen schickt. UND oben auf 5 min Klicken um schönere Daten zu sehen.

Side Quest: "Viele mqtt-Clients"

Voraussetzungen:

1. Du hast eine Host-only Connection mit deinem Host und kannst deine VM pingen
2. Dein Broker kann laufen auf port 1883 oder anders...
3. Grafana kann deine Topics visualisieren

Neues Java Project

Also, ich habe ein funktionierendes Projekt und mein Vorschlag ist, das ich dir einfach mein Github Link gebe. Du lädst das Projekt runter von meinem GIT und importierst es auf VS oder Eclipse oder IntelliJ. Wenn es schwierig wird, benutze VS Code, weil ich das benutzt habe. Es hat hilfreiche Kommentare zum verstehen vom Code.

<https://github.com/Ratedaron/IFZ-2224-013-M321-MQTT>

Sachen Anpassen im Projekt

1. Die Broker Address in allen vier files & evt die Topic auch anpassen
2. Die IP Adresse von deiner VM bekommst du mit dem Code

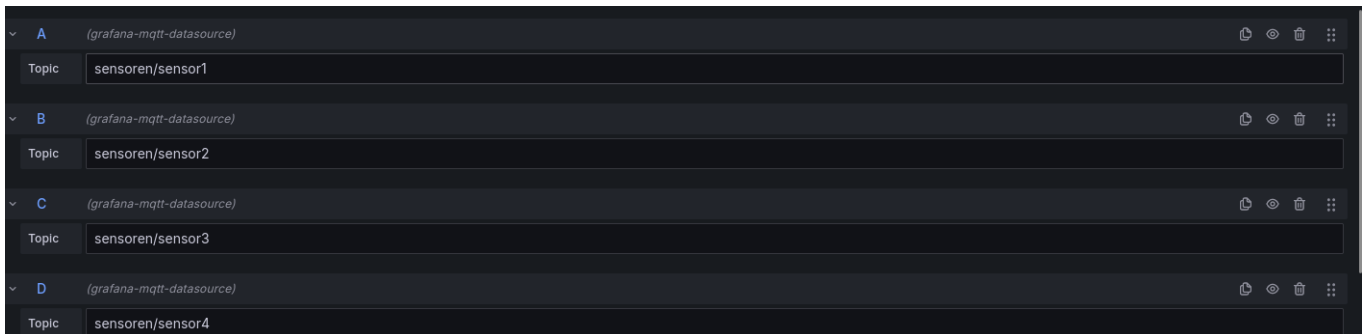
```
sudo ip a
```

oder mit

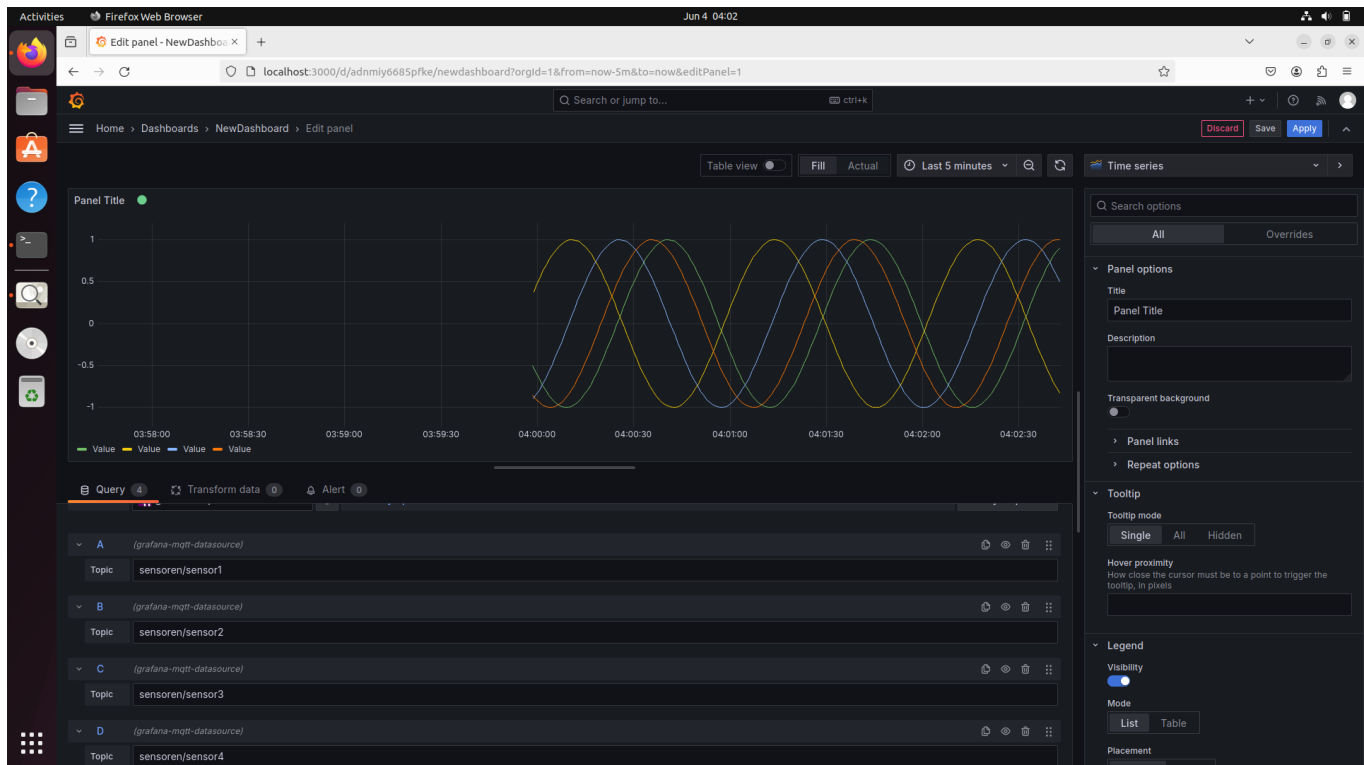
```
sudo ip addr show
```

```
Run | Debug
public static void main(String[] args) {
    String broker = "tcp://192.168.56.109:1883"; // MQTT Broker-Adresse
    String clientId = "demo_client"; // Client-ID
    String topic = "sensoren/sensor1"; // MQTT-Topic
    int pubQos = 1; // Qualitätsstufe der veröffentlichten Nachrichten
}
```

3. Alle deine angepassten Topics (auch die du gleich gelassen hast) in grafana reinschreiben



4. Resultat:



5. Falls du nichts im Grafana siehst, sube mal im Terminal zu einen von deinen Topics und schaue ob sie überhaupt Daten bekommen
6. Schaue auch immer ob dein Broker Container läuft, ob du die VM vom Host pingen kannst und in VS code die richtige ip hast und richtigen Port
7. Benutze mein GIT als Hilfe und frage mich wenn du nicht weiter kommst

aron.haubner@wiss-edu.ch

Zum Schluss nochmal mein GIT: <https://github.com/Ratedaron/IFZ-2224-013-M321-MQTT>

Ich habe dieses Forum Benutzt um das PDF zu stylen: [Page Breaks for PDFs - Help - Obsidian Forum](#)