



MQTT - Broker connection

Technical manual

NOTE:

The Information in this manual is in a “draft” state, meaning, that the contents may be incomplete or not completely up to date with the myGEKKO OS Version you may be using.

Table of Contents

Safety and Warranty.....	2
Safety instructions	2
Warranty.....	2
Purpose.....	3
Benefits	3
Realization in 3 steps.....	3
Configuration and connection.....	4
Basic configuration and connection.....	4
SSL/TLS configuration and connection.....	8
Connection status messages.....	12
Topics and elements.....	13
By default topic structure	13
Extended topic structure.....	15
States updates.....	16
Element Discovery.....	21
Sending commands	25
Topic subscription.....	26
Demos.....	27
Demo 1: connecting myGEKKO with HiveMQ Broker running online (HiveMQ cloud)	27
Demo 2: creating your own client certificate.....	55
Useful links and bibliography	56



Safety and Warranty

The devices are manufactured in a safe and workmanlike manner in compliance with current regulations and are operationally reliable. They have been tested and have left the factory in perfect condition.

However, there are residual risks. Read and comply with safety instructions to avoid risks.

Ekon GmbH assumes no liability for damages caused by non compliance with the safety instructions.

Safety instructions

Safety instructions about the product described in this document are listed below. Please observe them carefully while using the product.

Electric Voltage

Danger to life and fire hazard from electrical voltage

Inside the device are unprotected live component parts. Observe the VDE regulations.

Disconnect all cables to be installed and take safety precautions against unintentional switching on. Do not operate the device if damaged or not functioning properly. Decommission the device or system and secure it against accidental operation, if any operation without risk may supposedly not be guaranteed.

Device damage from external influences!

Moisture and dirt on the equipment can lead to its destruction.

Protect devices from moisture, dirt and damage during transport, storage and operation.

Warranty

The device is intended solely for proper use. Any inappropriate change or non compliance with the operating instructions invalidates any warranty or guarantee. After unpacking, check the device immediately for any mechanical damage. In case of damage caused during transportation please immediately inform the supplier. The device must only be operated as a fixed installation, i.e. only when installed and after all of the installation and commissioning work has been completed, and exclusively in the environment it is intended for. Ekon GmbH shall not be liable for any modifications to standards and norms after publication of the operating instructions.



Purpose

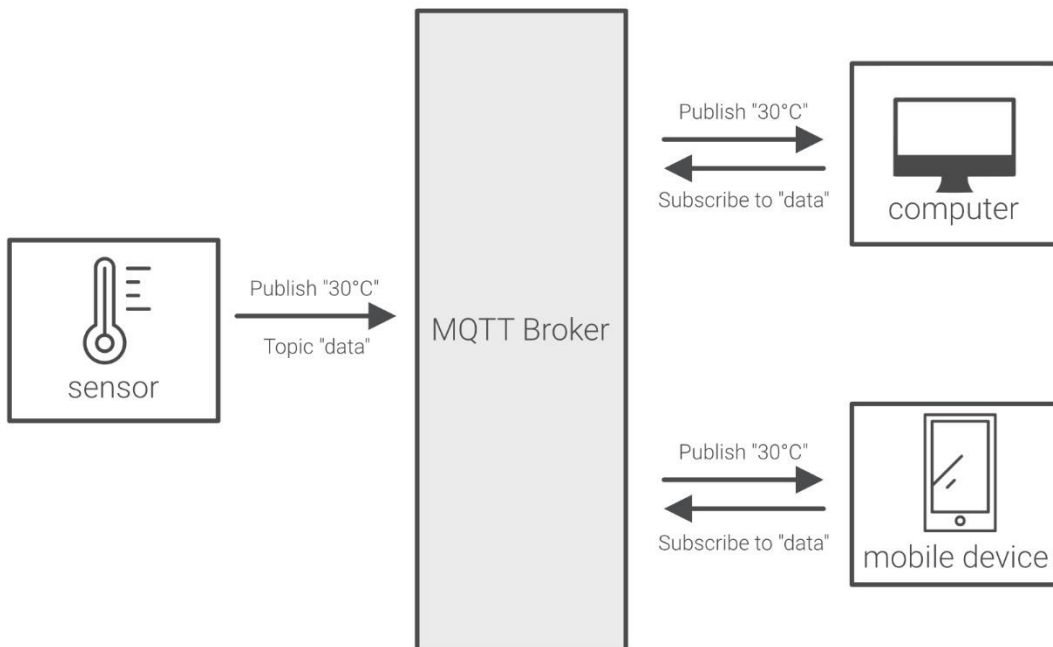
The MQTT Broker connection enables the communication of the myGEKKO controller with any MQTT broker (MQTT version 3.1 or 3.1.1). Elements from all systems such as "Light", "Blind", "Ventilation", etc. can be pushed directly to the broker as a separate topic. It is also possible to control the system elements using commands.

Benefits

- Simple way to access the myGEKKO System
- Alternative to the REST-API
- Use a MQTT Broker of your own choice

Realization in 3 steps

1. Set up a MQTT Broker
2. Connect the myGEKKO controller with the broker
3. Subscribe to the topic you want access to



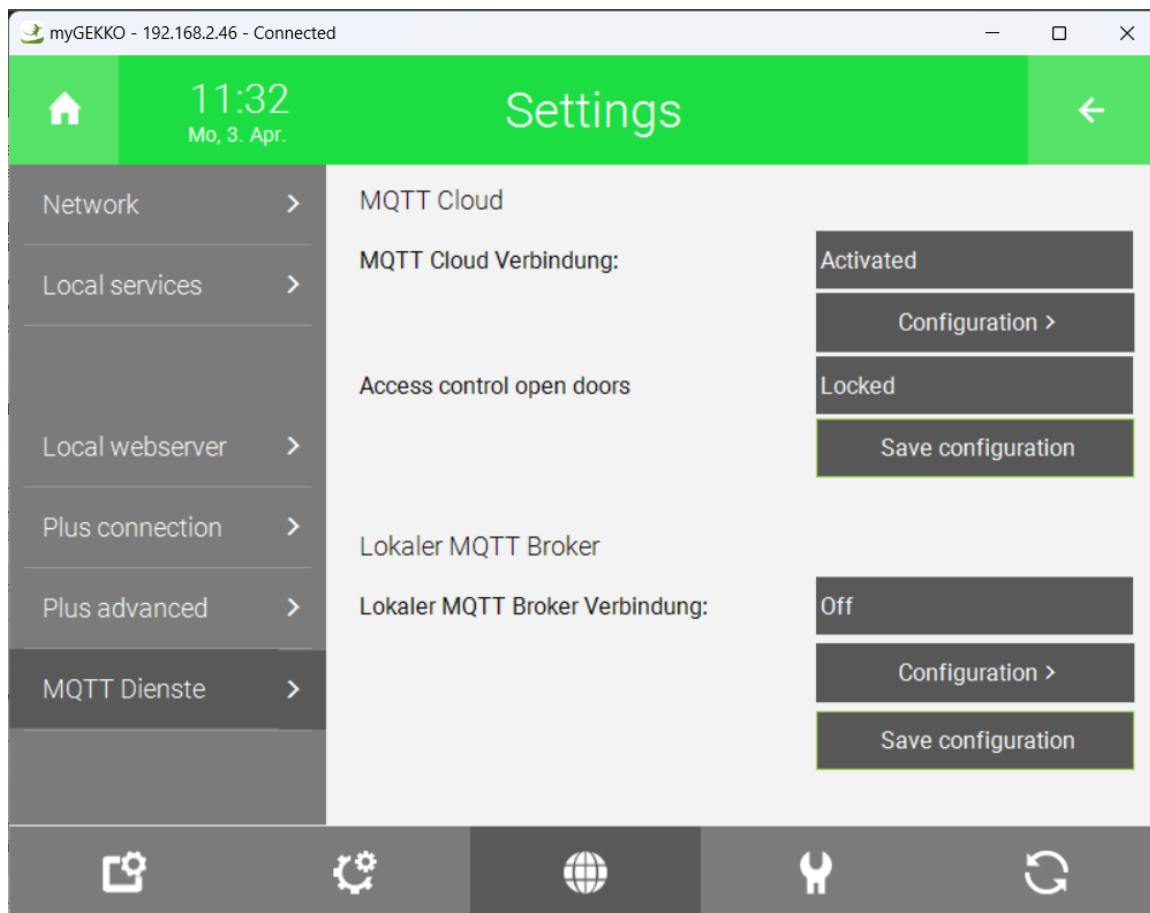
Configuration and connection

This section shows, in general, how to configure and set a connection from the myGEKKO device to a MQTT broker.

Basic configuration and connection

Log in to the myGEKKO building controller as a configurator and go to the settings (gear icon at the top -> Settings). Go to the network settings (globe icon at the bottom).

Select "MQTT Dienste" in the selection list on the left. On the right, the "MQTT Cloud" menu is shown.



To set the MQTT cloud connection with the broker chosen by the user, press the "Configuration" button on the right, in the menu MQTT Broker connection.

The following settings are available in the configuration menu:

- **Broker address:** The address of the MQTT broker (URL or IP address).
- **Port:** The port used by the broker (1883 by default, 8883 if using SSL/TLS).



- **Username:** The username for authentication, if available or required by the broker.
- **Password:** The password for authentication, if available or required by the broker.
- **Permission:** Selection of the permission from the broker side to the device (read only - read & write). By default, this option is set to “read only”.
- **SSL Connection:** Selection to enable SSL over MQTT (not activated - activated). By default, this option is set to “not activated”.
- **Use client certificate:** Selection to enable the use of the certificate in the client side. By default, this option is set to “not activated”. **NOTE:** if this option is enabled, it is mandatory to upload a valid client certificate.
- **Check server certificate:** Selection to enable the check of the server-side certificate when creating the connection. By default, this option is set to “not activated”.
- **Upload certificate:** Copy the MQTT TLS certificate from either the USB stick or the computer.
- **Encryption (not available yet):** Selection of the block encryption used (off - on). By default, this option is set to “off”.
- **Topic header:** an alphanumeric ID that is added at the beginning of the myGEKKO MQTT topic format.

The connection status is shown in the field at the bottom left. After setting up all the options, it is needed to go back to the MQTT configuration screen in network configuration to activate the connection:

Connection to MQTT Broker → Activated.

Check the Connection Status in the configuration screen to verify that the connection with the server is working.



myGEKKO - 192.168.2.46 - Connected

MQTT Cloud Konfiguration

Broker Adresse	cbce363fd16942528fddeb7a031f6f3b.s2.eu.hivemq.cloud		
Port	8883	Rechte	Lesen+Schreiben
Benutzername	myGEKKO_manager	Passwort	*****
SSL Verbindung	Aktiviert	Zertifikat importieren >	
Topic header	Aus		

Verbindungsstatus

Ok Zurück



myGEKKO - 192.168.2.46 - Connected

MQTT Cloud Konfiguration

Broker Adresse	cbce363fd16942528fd16942528fdbb7a031f6f3b.s2.eu.hivemq.cloud		
Port	8883	Rechte	Lesen+Schreiben
Benutzername	myGEKKO_manager	Passwort	*****
SSL Verbindung	Aktiviert	Zertifikat importieren >	
Topic header	Aktiviert	VillaFranzelin	

Verbindungsstatus
Ok

Zurück



SSL/TLS configuration and connection

To use SSL/TLS connection, required sometimes by the MQTT broker provider, a certificate may be needed. The certificate can be provided by the server side. This means that different MQTT brokers could provide different certificates. Therefore, the user must upload the correct certificate to the myGEKKO controller according to the chosen broker. However, it is also possible to generate your own certificate (advanced knowledge is needed, see Demo 2) and use it as client certificate.

After downloading/generating the root certificate, two options are available:

1. Import by using a USB stick.
2. Import from the computer.

NOTE: In case that multiple MQTT certificates are being used, the user must copy the content of the certificate to the **mqtt_cloud_cert.pem** file, since a .pem file is a container format for digital certificates and keys. This is a file format that contains a bunch of certificate files.

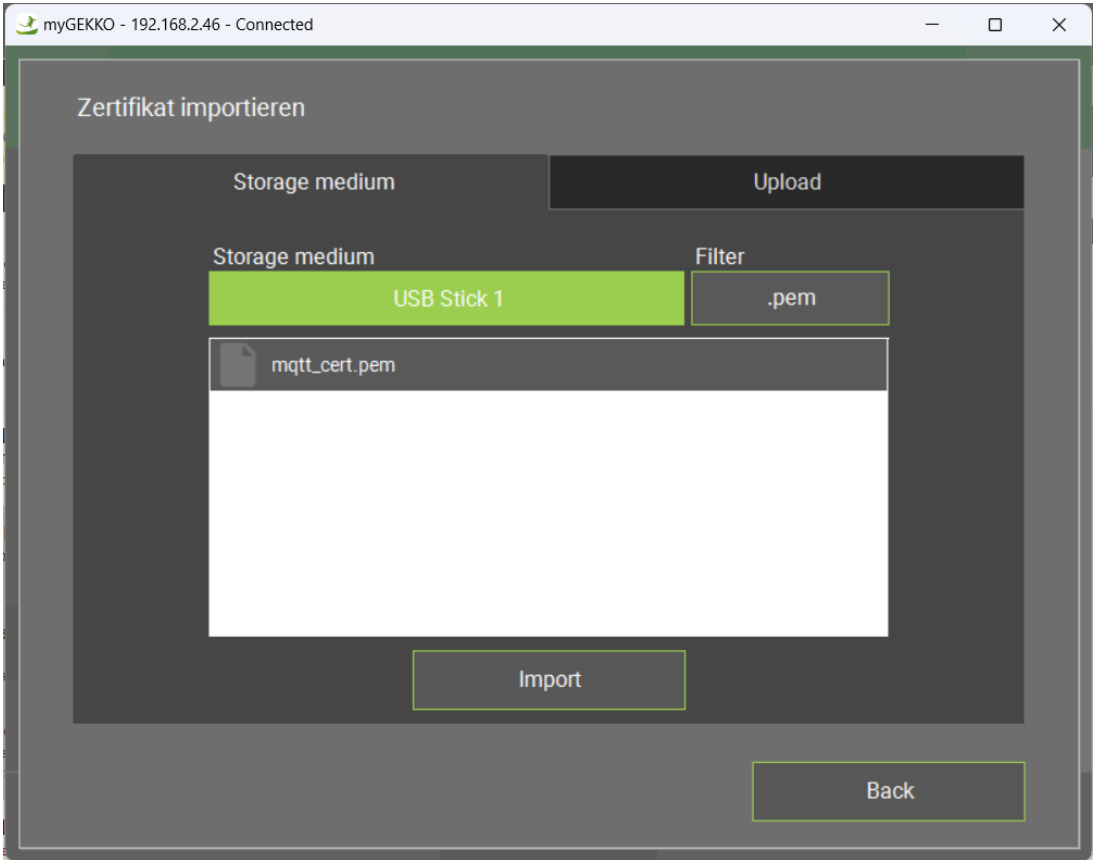
In any case, click once on the button “Import certificate”, and then a pop-up menu is shown. The two options mentioned above are shown at the top bar.

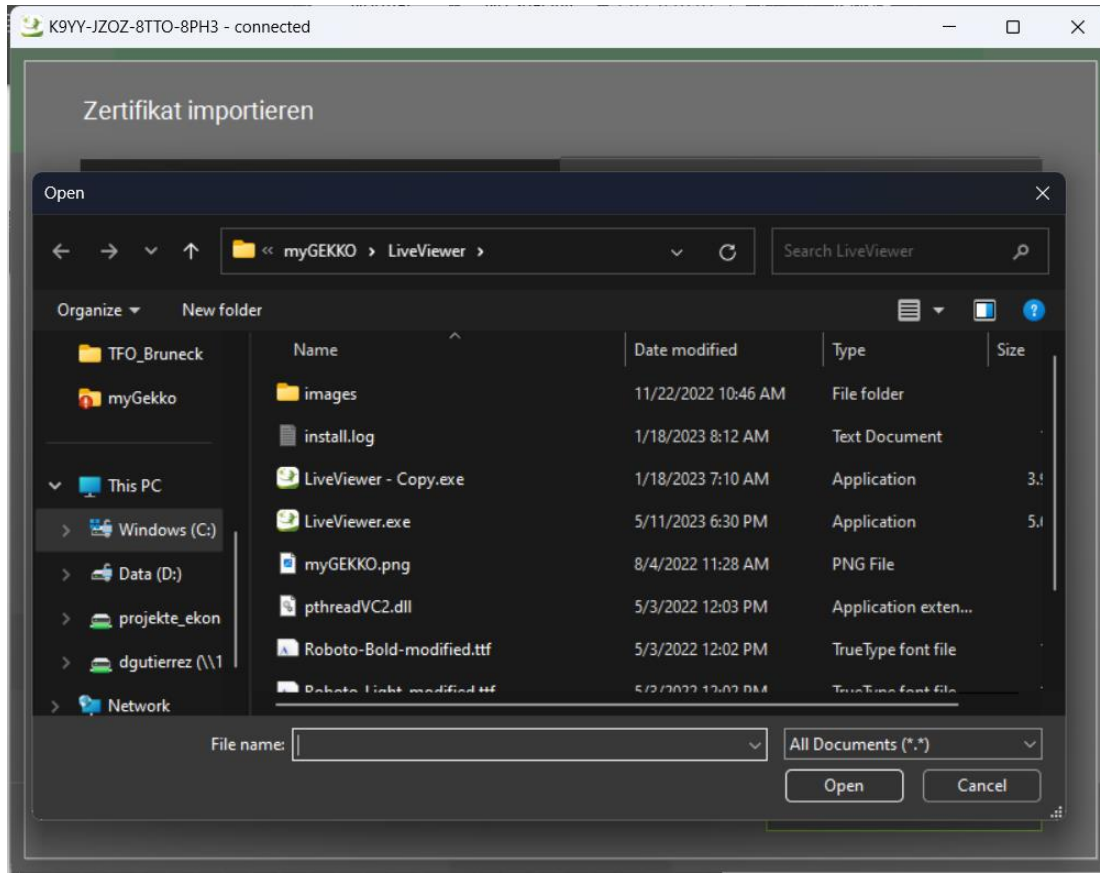
In case you want to use the USB stick, rename the file to **mqtt_cloud_cert.pem** and copy it into a USB stick. Then, plug-in the USB stick into the myGEKKO device.

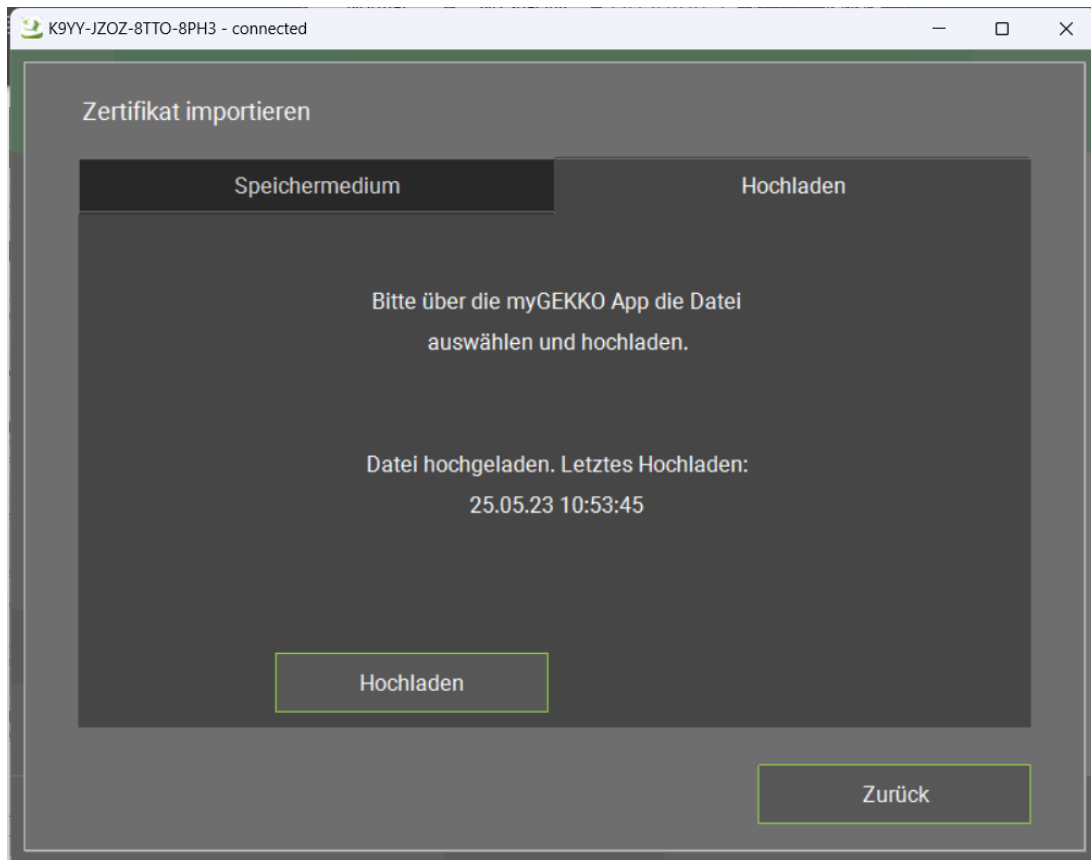
If you want to use the USB stick, click on the “Storage medium” tab, then click on the USB textbox to select the correct USB stick. The filter is preconfigured to be .pem, and should not be changed. Then, if the file was copied successfully to the USB stick, it will be shown in the viewer. Select the correct file and click on “Import”. The certificate will be copied automatically from the USB stick to the myGEKKO device, overwriting the previous **mqtt_cloud_cert.pem** file. If everything was ok, a confirmation message will be shown.

On the other hand, if you prefer to upload the certificate from the computer by using the Viewer, click on the “Upload” button, and then click on the new “Upload” button shown at the bottom left corner. The file explorer will be shown. There, go to the folder where the certificate was downloaded, click on it, and click on “Open”. The uploading process will start. If everything was ok, a confirmation message will be shown, and the date and time of the update will be also shown.









After uploading the certificate, and properly configuring the address, port, username, and password, enable the SSL connection:

SSL connection -> Activated.

Then, enable the option "Use client certificate" to use the uploaded certificate, and also the "Check server certificate" option.

WARNING: sometimes the broker requires a SSL connection but does not require the use of a client certificate. In that case, please make sure that the "Use client certificate" option is disabled. In addition, if a user certificate is not required, it is probably not needed to check the certificate of the server side. Therefore, the option "Check server certificate" may need to be disabled as well.

Check the Connection Status to verify that the connection with the server is working.



Connection status messages

The status field in the MQTT Cloud configuration menu informs about the connection status of the myGEKKO device with the MQTT broker. The following table summarizes the set of status in which the connection can be:

Status display	Status description	Possible solutions
Ok	The station is connected and ready to receive messages	-
Nicht verbunden	The device is not connected	Enter the connection parameters and set the cloud connection to Enabled.
Keine Antwort	The broker does not provide a response to the connection request	Check the used broker for errors.
Benutzer/Passwort falsch	The entered password and/or username are incorrect	Checking the input fields and the broker authentication settings.
Host Adresse falsch/Ungültiges Zertifikat	The specified address cannot be reached	Check address for correctness. Check if the selected broker is active. Check the port used (by default: 1883) Check the certificate was successfully installed
Sys 1-5	An internal system-related error has occurred	Contact the support department if this occurs.



Topics and elements

This section describes the MQTT topics and the structure of the information contained in the elements.

By default topic structure

The topic for a system element is structured as follows:

GEKKOID/SYSTEM/itemX

Where:

1. **GEKKOID:** 16-digit unique identification number separated by - (XXX-XXXX-XXXX-XXXX)
2. **SYSTEM:** The name of the system in which the element is located (lights, blinds, etc.).
3. **itemX:** The item number (item0, item1, ...)

For example:

7998-89YZ-8XUT-5V43/lights/item0

The table on the next page summarizes the SYSTEM names with the system elements of the myGEKKO device:



MQTT system element name	myGEKKO system element name	myGEKKO system element index
globals	-	-
lights	Light	0
blinds	Blinds	1
vents	Ventilation	2
loads	Device	3
roomtemps	Room control	4
heatingsystems	H/C Producer	5
heatingcircuits	H/C Circuit	6
energycosts	Energy meter	7
alarms_logics	Logics	8
accessdoors	Access	9
energymanager	Energy manager	10
alarmsystem	Alarm system	11
multirooms	Music	12
cams	Video	13
smsemail	Mail/Sms	14
actions	Actions	15
clocks	Timer	16
controlcircuits	Control circuits	19
air_handling_unit	Air condition	20
saunas	Sauna	22
pools	Pool	23
trends	Analysis	27
emobils	E-Mobility	28
hotwater_systems	H/C Storage	39
hotwater_circulations	H/C Circulation	40
stoves	Fireplace	41
door_intercom	Door intercom	45



Extended topic structure

The extended topic structure is enabled when the “topic header” option is enabled. The topic header option allows you to include a custom ID at the beginning of the topic, thus extending it.

For example, if you have a building with multiple myGEKKO controllers, all of them publishing to the same MQTT broker, the building manager could include the name of the building at the beginning of the default topic structure to better organize all the controllers. Therefore, instead of having:

```
7998-89YZ-8XUT-5V43/lights/item0
```

We could have

```
myBuilding-01/7998-89YZ-8XUT-5V43/lights/item0
```

This way, it will be much easier to the user to subscribe to the building topic:

```
myBuilding-01/#
```

In order to receive all the packages from all the controllers instead of subscribing one by one to all the individual topics from each device.

There are two limitations when using the extended topic structure:

1. Discovery limited to the entire topic header: this means that even if the topic header has multiple levels, for example, “Brunico/TFO_Bruneck/Floor1/”, the discovery command, is limited to the full topic header. For example, the user could send the discovery command to the topic “Brunico/TFO_Bruneck/Floor1/”, but not to the topic “Brunico/”.
 - a. “Brunico/TFO_Bruneck/Floor1/discovery” → correct
 - b. “Brunico/discovery” → incorrect
2. The maximum number of characters is limited to 255 characters.

And there are also two important considerations:

1. The character “/” is added automatically: when writing the topic header, the user does not need to add the “/” to indicate the end of the topic header. This character is added automatically after normalizing the string and converting all the not allowed characters into allowed characters. However, it is not a problem if the user adds the “/” at the end of the topic header since the system will recognize it.
2. If the topic header is enabled but the textbox is empty, then no topic header is added by default, and it will have the same effect as the topic header is not enabled.



States updates

After the MQTT connection is established for the first time, a packet is sent once for each system element. After that, only the elements that have changed their states are sent out at regular intervals.

The “*retain flag*” is enabled by default so that new subscribers always immediately receive the latest current state of the system element from the broker.

WARNING: this has to be taken into account since it will also be shown the states of the commands even though the new user does not have write permission.

The payload consists of a JSON list containing the parameters’ values that this element has.

For example:

The lamp at index 0 is dimmable and has RGB light. After the state of the lamp has changed, the following message is sent via MQTT:

Topic: J9Y8-X9NZ-8TUD-5EG3/lights/item0

Payload:

```
{
  "page": "groundfloor",
  "name": "Lamp 0",
  "currentState": 1,
  "dimLevel": 100,
  "rgbColor": 16777215
}
```

The available states list for a system element can be read out via Element Discovery (see separate chapter on section “



Element Discovery”).

If the topic header option is enabled, with the topic header “myBuilding-01”, then message will be exactly the same, and the topic will include the topic header:

Topic: myBuilding-01/J9Y8-X9NZ-8TUD-5EG3/lights/item0

Payload:

```
{
  "page": "groundfloor",
  "name": "Lamp 0",
  "currentState": 1,
  "dimLevel": 100,
  "rgbColor": 16777215
}
```

Additionally, there is an option for sending a request to receive the latest status of all the active system elements. This can be done by appending **/status** at the end of the topic. Then, depending on the topic’s level in which the /status was set, the systems’ elements will send a message including its variables and their current state.

For example:

You have a myGEKKO controller with ID J9Y8-X9NZ-8TUD-5EG3. As system elements, we have a lamp at index 0, which is dimmable and has RGB light, and a plug with the basic configuration.

To request the status of the entire system, the following message is sent via MQTT:

Transmit:

Topic: J9Y8-X9NZ-8TUD-5EG3/status

Payload: “ ” (a white space)

Received:

Topic: J9Y8-X9NZ-8TUD-5EG3/lights/item0

Payload:

```
{
  "page": "groundfloor",
  "name": "Lamp 0",
  "currentState": 1,
```



```
    "dimLevel": 100,  
    "rgbColor": 16777215  
}
```

Topic: J9Y8-X9NZ-8TUD-5EG3/globals/meteo

Payload:

```
{  
    "twilight":255  
}
```

Topic: J9Y8-X9NZ-8TUD-5EG3/globals/network

Payload:

```
{  
    "hardware": "BASE R07 X6 series",  
    "version": 680007,  
    "gekkoname": "myGEKKO",  
    "language": 2  
}
```

Topic: J9Y8-X9NZ-8TUD-5EG3/loads/item0

Payload:

```
{  
    "currentState": 0,  
    "elementInfo":0  
}
```

To request the status of just the item0 of the lights, the following message is sent via MQTT:

Transmit:

-Topic: J9Y8-X9NZ-8TUD-5EG3/lights/item0/status

-Payload: " " (a white space)

Received:



-Topic: J9Y8-X9NZ-8TUD-5EG3/lights/item0

-Payload:

```
{  
  "page": "groundfloor",  
  "name": "Lamp 0",  
  "currentState": 1,  
  "dimLevel": 100,  
  "rgbColor": 16777215  
}
```

In the same way, to request the status of all the lights, the following message is sent via MQTT:

Transmit:

-Topic: J9Y8-X9NZ-8TUD-5EG3/lights /status

-Payload: " " (a white space)

Received:

-Topic: J9Y8-X9NZ-8TUD-5EG3/lights/item0

-Payload:

```
{  
  "page": "groundfloor",  
  "name": "Lamp 0",  
  "currentState": 1,  
  "dimLevel": 100,  
  "rgbColor": 16777215  
}
```

-Topic: J9Y8-X9NZ-8TUD-5EG3/lights/item1

-Payload:

```
{  
  "page": "groundfloor",  
  "name": "Living room",  
  "currentState": 1,  
}
```



```
"dimLevel": 50  
"rgbColor": 16777215  
}
```

NOTE: the request can be done either to the main topic (the myGEKKO ID), or in case that the topic header option is enabled, to the topic header followed by the myGEKKO ID. For example:

J9Y8-X9NZ-8TUD-5EG3/status → Correct

myBuilding-01/J9Y8-X9NZ-8TUD-5EG3/status → Correct

myBuilding-01/status → Incorrect

myBuilding-01/lights/status → Incorrect



Element Discovery

To know which system's elements are present and what values they return, a discovery request can be sent to the myGEKKO device. For a discovery request, a message containing a white space " " is sent to the desired topic, with a **/discovery** appended.

In response, a JSON list is sent to the requested topic + **/info**.

For example:

Transmit:

-Topic: J9Y8-X9NZ-8TUD-5EG3/lights/item0/discovery

-Payload: " "

Received:

-Topic: J9Y8-X9NZ-8TUD-5EG3/lights/item0/info

-Payload:

```
{
  "dimLevel": {
    "permission": "READ",
    "description": "Current value of the intensity of the light,
where 0.00% means completely off, and 100.00% means completely
on",
    "format": "float[0.00,100.00](%)",
    "type": "AO"
  },
  "setDimLevel": {
    "permission": "WRITE ",
    "description": "Set the current value of the dimLevel",
    "format": "float[0.00,100.00](%)",
    "type": "AV"
  },
  "currentState": {
    "permission": "READ ",
    "description": "Current state of the light",
    "format": "enum[0=off,1=on]()",
```



```

        "type": "A0"
    },
    "setRGBColor": {
        "permission": " WRITE ",
        "description": "Modify the current RGB color of the RGB light.
Example: green color is desired --> R=0, G=255, B=0 --> R=0x00,
G=0xFF, B=0x00 --> colorRGB=0x00FF00 --> colorRGB=65280",
        "format": "int[24bit->R:23-16|G:15-8|B:7-0]()",
        "type": "AV"
    },
    "setCurrentState": {
        "permission": " WRITE ",
        "description": "Set the current state of the light",
        "format": "enum[0=off,1=on]()",
        "type": " AV"
    },
    "toggleCurrentState ": {
        "permission": "WRITE",
        "description": "Toggle the current state of the light",
        "format": "enum[1=toggle]()",
        "type": " AV"
    },
    "rgbColor": {
        "permission": " READ ",
        "description": "Current RGB color of the light. Example:
colorRGB=16721246 --> colorRGB=0xFF255E --> R=0xFF, G=0x25,
B=0x5E --> R=255, G=37, B=94",
        "format": "int[24bit->R:23-16|G:15-8|B:7-0]()",
        "type": "A0"
    }
}
}

```



Four properties are shown for each element:

- **Permission:** it indicates if the element is a state (“READ”) or a command (“WRITE”).
- **Description:** it provides a short explanation of the meaning of the state. If needed, it also provides an example.
- **Format:** it always has the same structure: datatype[range](unit). First, the data type is indicated (int, float, string, enum...). Then, inside the [], the range is shown, indicating the lower limit, separated by a comma, and the upper limit (for example [-100.00,100.00]). And finally, the unit, shown inside the brackets. For example, meters will be indicated as (m), always using the International System (I.S.).
- **Type:** it indicates the type of variable from the point of view of the myGEKKO device. It could be AI (analog input, from outside to the device), AO (analog output: from the device to outside), AV (analog value, used as input parameter when sending commands), and STRING.

Discovery can also be requested for an entire system class or for all elements on the device:

```
J9Y8-X9NZ-8TUD-5EG3/lights/discovery  
J9Y8-X9NZ-8TUD-5EG3/discovery
```

And it will return one main JSON object containing a node for each element. For example, if two lights are enabled, it will return the same JSON node for both of them.

WARNING: if the Element Discovery is called while the device is in Read+Write mode, and then the mode is changed to Read only, a new subscriber will be able to see all the commands when it connects with the broker since the message had the retain flag activated. However, even though if we send a “set” command, it will be ignored by the device.

However, if the topic header is enabled, the user may want to do a discovery of the topic header. Originally, the same info package was sent by all the devices which have the same topic header.

The problem is that the “info” package does not contain the myGEKKO ID, neither the topic, since the reply to “myBuilding-01/discovery” will be “myBuilding-01/info”, and we will receive as many packages with the same topic as myGEKKO controllers we have with that topic.

Therefore, when a topic header discovery message is sent, each myGEKKO controller will reply with a message just including its myGEKKO ID instead of the elements that are enabled on it. This way, we will have a list of myGEKKO IDs, and then we can do a discovery using a specific myGEKKO ID to have the active elements in that myGEKKO controller.

For example:

Transmit:

-Topic: myBuilding-01/discovery

-Payload: “ ”

Received:

-Topic: myBuilding-01/info



-Payload:

```
{  
  "myGEKKO_ID": "79Y8-Y9YZ-8TUV-5A43"  
}
```

NOTE: in that case, the user would need to know the correlation between the myGEKKO ID and the location of the myGEKKO controller, since the name is not included in the payload.



Sending commands

Command messages are also available to modify some states' values. As with the status updates, the payload is sent in the form of a JSON list.

The standard topic structure with an appended **/set** is used as the topic.

For example:

The lamp at index 0 is to be switched on and dimmed to 50%. Therefore,

-Topic: J9Y8-X9NZ-8TUD-5EG3/lights/item0/set

-Payload:

```
{  
  " setCurrentState " : 1,  
  "setDimLevel " : 50  
}
```

If the topic header is enabled, then the topic header must be included in the topic when sending a command:

-Topic: myBuilding01/J9Y8-X9NZ-8TUD-5EG3/lights/item0/set

-Payload:

```
{  
  " setCurrentState " : 1,  
  "setDimLevel " : 50  
}
```

NOTE: The available commands for a system element can be read out via Element Discovery only if the "Permission" field is set to "Read+Write". If "Permission" is set to "Only Read", only the status will be shown via Element Discovery.



Topic subscription

In MQTT we can find two profiles: the publisher and the subscriber. The publisher can be any device that wants to post a message into the broker. As we saw, the message has a fixed structure, with a topic and a payload. The subscriber can be any device that wants to receive a message from the broker. To receive a message, the subscriber needs to connect to the broker, and add a subscription to a topic. This means that only messages within that topic will be received by the subscriber.

The subscriptions can be done to a generic element (for example, lights), the entire system, or just one specific element.

To subscribe to all, you will need to simply add "#". Alternatively, it could also subscribe to

```
J9Y8-X9NZ-8TUD-5EG3/#
```

If the topic header is enabled, you can subscribe to the topic header to receive the packages from all the myGEKKO controllers configured with that topic header:

```
myBuilding-01/#
```

If the topic header has multiple levels, you can also subscribe to part of the topic header:

```
Brunico/#
```

Another example, if I want to receive all the messages posted by all the lights connected to the myGEKKO device, you will need to subscribe to

```
J9Y8-X9NZ-8TUD-5EG3/lights/#
```

Where the first part is the system ID, then the system element name, and finally the #.

WARNING: if only `J9Y8-X9NZ-8TUD-5EG3/lights` or `J9Y8-X9NZ-8TUD-5EG3/lights/` is added as subscription, it will not work.

However, if I want to subscribe to the messages coming from one specific light, the item0, then the subscription will be to

```
J9Y8-X9NZ-8TUD-5EG3/lights/item0
```

If the character '#' is added, then also the /info messages from that item is received.

```
J9Y8-X9NZ-8TUD-5EG3/lights/item0/#
```

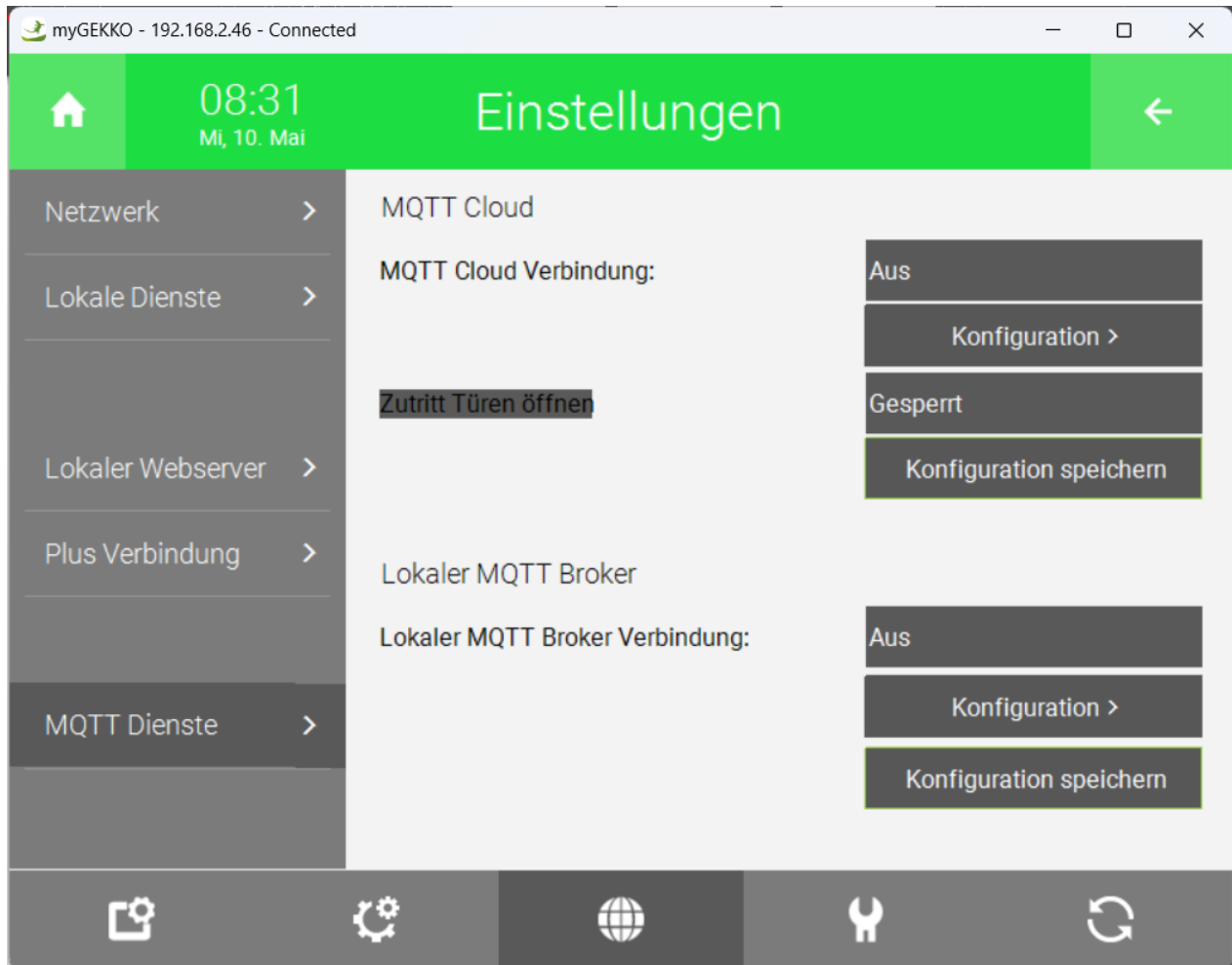
WARNING: if an extra '/' is added `J9Y8-X9NZ-8TUD-5EG3/lights/item0/` then it is not working.



Demos

Demo 1: connecting myGEKKO with HiveMQ Broker running online (HiveMQ cloud)

Starting from the “*network settings*”, in the MQTT Configuration menu:



Click on configuration of the MQTT Broker connection to go into the MQTT Broker connection configuration menu:



myGEKKO - 192.168.2.46 - Connected

MQTT Cloud Konfiguration

Broker Adresse	<input type="text"/>		
Port	<input type="text"/>	Rechte	Nur Lesen
Benutzername	<input type="text"/>	Passwort	*****
SSL Verbindung	Aus	Zertifikat importieren >	
Topic header	Aus		

Verbindungsstatus
Nicht verbunden

Zurück

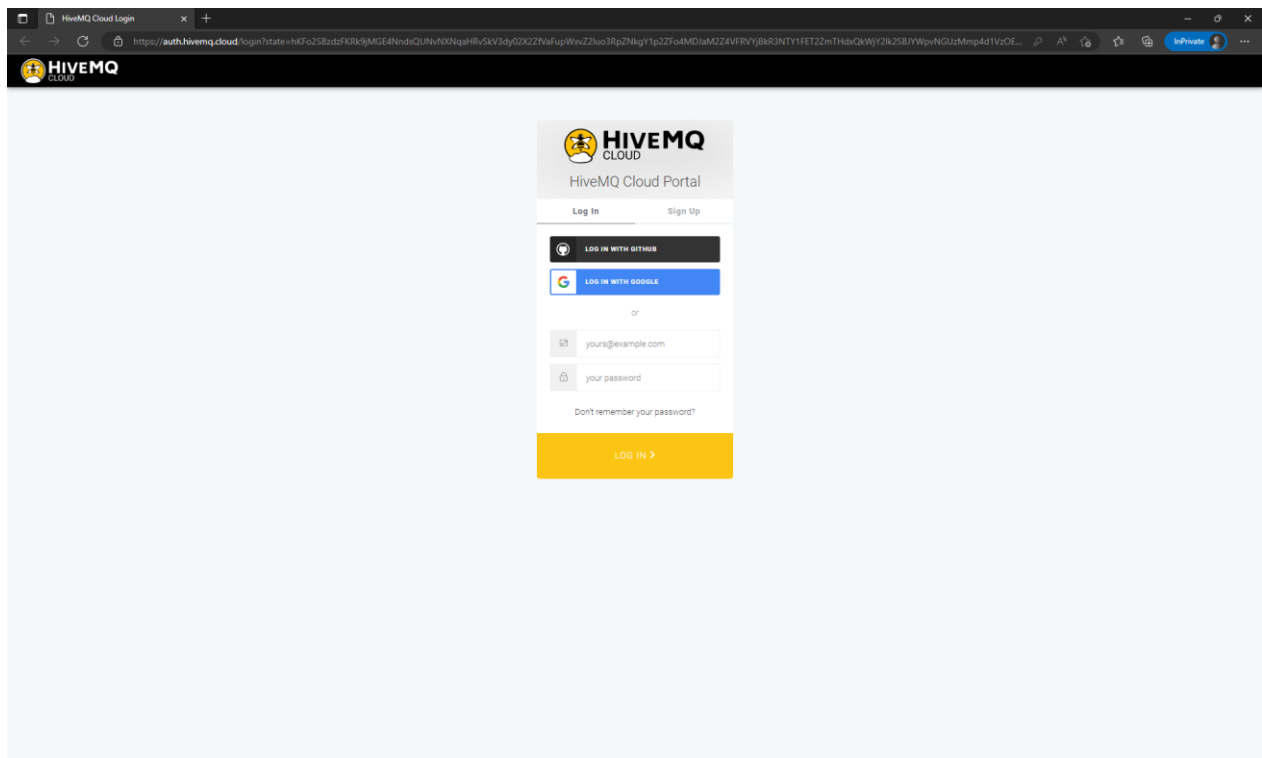
We need to add the information of the broker to configure the connection between the myGEKKO device and the cloud broker. For this demo, we have chosen HiveMQ, which offers a free version with some limitations.

Go to <https://www.hivemq.com/mqtt-cloud-broker/> and click on Try out for free.





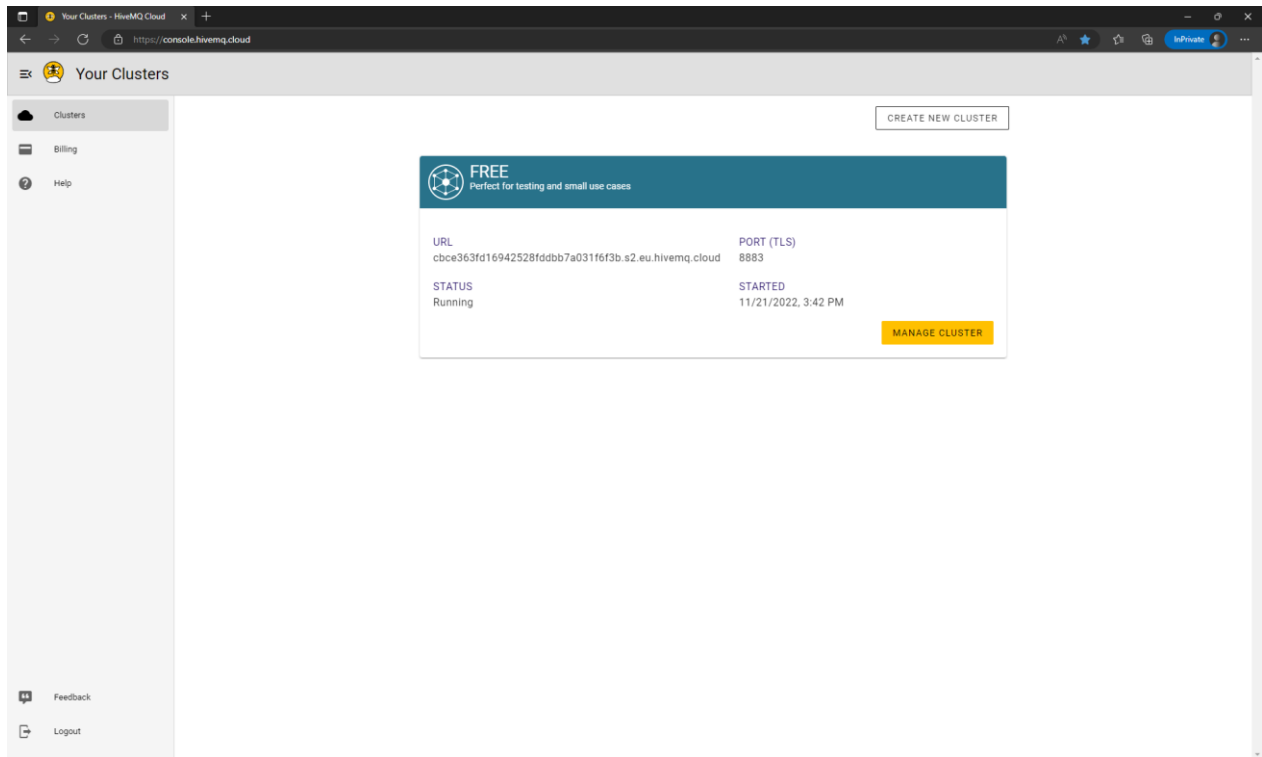
Then, the next screen is shown, asking for the log in information.



Write the credentials:



- User: sholzer@my-gekko.com
- Password: H?FQSPqYx#Y884rM



From the welcome screen, go to "Manage cluster". Then, in the "Overview" page, the cluster details screen appears, showing the cluster URL details, as well as the connection port.



Both the cluster URL and the Port will be needed to configure myGEKKO. Copy the “Cluster URL” value from the HiveMQ website to the field “Address” in myGEKKO MQTT Broker configuration. Copy the “Port” value from HiveMQ website to the field “Port” in myGEKKO MQTT Broker configuration.

- cbce363fd16942528fddb7a031f6f3b.s2.eu.hivemq.cloud
- 8883



myGEKKO - 192.168.2.46 - Connected

MQTT Cloud Konfiguration

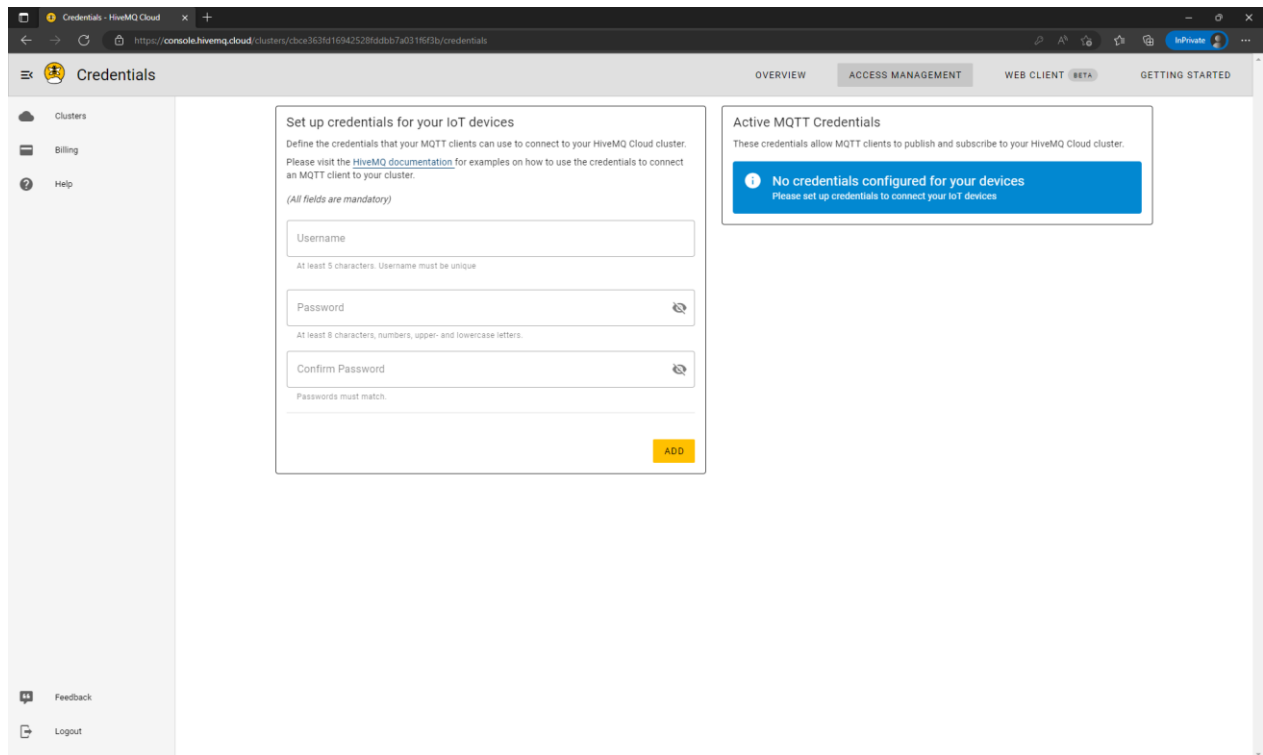
Broker Adresse	cbce363fd16942528fddb7a031f6f3b.s2.eu.hivemq.cloud		
Port	8883	Rechte	Nur Lesen
Benutzername		Passwort	*****
SSL Verbindung	Aus	Zertifikat importieren >	
Topic header	Aus		

Verbindungsstatus
Nicht verbunden

Zurück

Then, we need to manage the access to the broker. For that, click on “Access management” button located at the top right corner of the HiveMQ screen.





No credentials are created by default. Two access credentials will be created:

1. myGEKKO_device: to be used by the device.
2. myGEKKO_test: to be used by the user as client through the HiveMQ web client or any other external MQTT client, like MQTTX.

To create a new credential, fill up the fields of the box “Set up credentials for your IoT devices”.

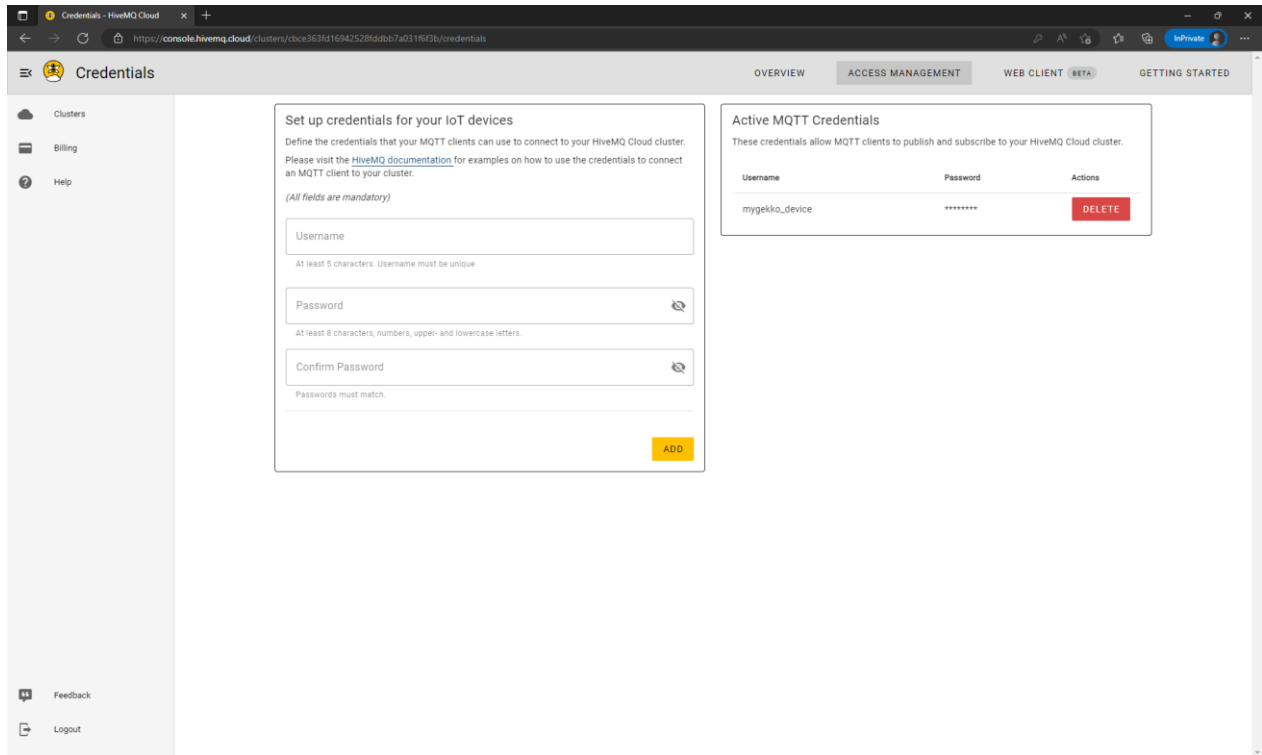
For myGEKKO_device, write:

- Username: mygekko_device
- Password: mygekko_device
- Confirm Password: mygekko_device

And click on the “Add” button.

Then, the “Active MQTT Credentials” box, shown at the right, will be updated and it should look like:





Then, for myGEKKO_test, write:

- Username: mygekko_test
- Password: mygekko_test
- Confirm Password: mygekko_test

And click on the “Add” button.



The screenshot shows the 'Credentials' page in the HiveMQ Cloud console. The page has a sidebar on the left with navigation options: Clusters, Billing, Help, Feedback, and Logout. The main content area is split into two panels. The left panel, titled 'Set up credentials for your IoT devices', provides instructions and a form to create new credentials. The form includes fields for Username (with a note: 'At least 5 characters. Username must be unique'), Password (with a note: 'At least 8 characters, numbers, upper- and lowercase letters'), and Confirm Password (with a note: 'Passwords must match'). An 'ADD' button is located at the bottom right of the form. The right panel, titled 'Active MQTT Credentials', shows a table of existing credentials. The table has three columns: Username, Password, and Actions. Two credentials are listed: 'mygekko_device' and 'mygekko_test', both with masked passwords and a 'DELETE' button in the Actions column.

Username	Password	Actions
mygekko_device	*****	DELETE
mygekko_test	*****	DELETE

After creating the credentials, copy the username and password in the myGEKKO MQTT Broker connection configuration.



myGEKKO - 192.168.2.46 - Connected

MQTT Cloud Konfiguration

Broker Adresse	cbce363fd16942528fddeb7a031f6f3b.s2.eu.hivemq.cloud		
Port	8883	Rechte	Nur Lesen
Benutzername	myGEKKO_manager	Passwort	*****
SSL Verbindung	Aus	Zertifikat importieren >	
Topic header	Aus		

Verbindungsstatus
Nicht verbunden

Zurück

Next, click on the button "Import certificate" to copy the SSL certificate from a USB stick or the computer to the myGEKKO device. If you do not have the certificate, please download it from <https://community.hivemq.com/t/frequently-asked-questions/514>





Frequently Asked Questions

HiveMQ Cloud



hivemq-support

13 Apr '21

This thread contains a list of frequently asked questions about HiveMQ Cloud

- ▶ Supported Features in Free, PAYG and Dedicated model
- ▶ How are the connections counted in HiveMQ Cloud?
- ▶ Is there a price difference for PAYG customers for AWS vs. Azure?
- ▶ Can customers create more than 2 clusters in HiveMQ Cloud
- ▶ Can I manually clear unused client connections?
- ▼ My device requires a server CA file to connect via TLS. How can I generate this for my HiveMQ Cloud instance?

You can download the root certificate [here](#). This will create a file called "isrgrootx1.pem", which you can use as "Server Certificate".

- ▶ TLS-SNI extension (Server Name Indication)
- ▶ How does the billing for HiveMQ Cloud PayAsYouGo work?
- ▶ My device does no longer (since September 30 2021) trust HiveMQ Cloud. What to do?
- ▶ How long is Data retention and message queue size?
- ▶ I have started an upgrade from Free to Pay-As-You-Go offer, for how long should I wait? Is there any downtime event such as client disconnections on the broker side?



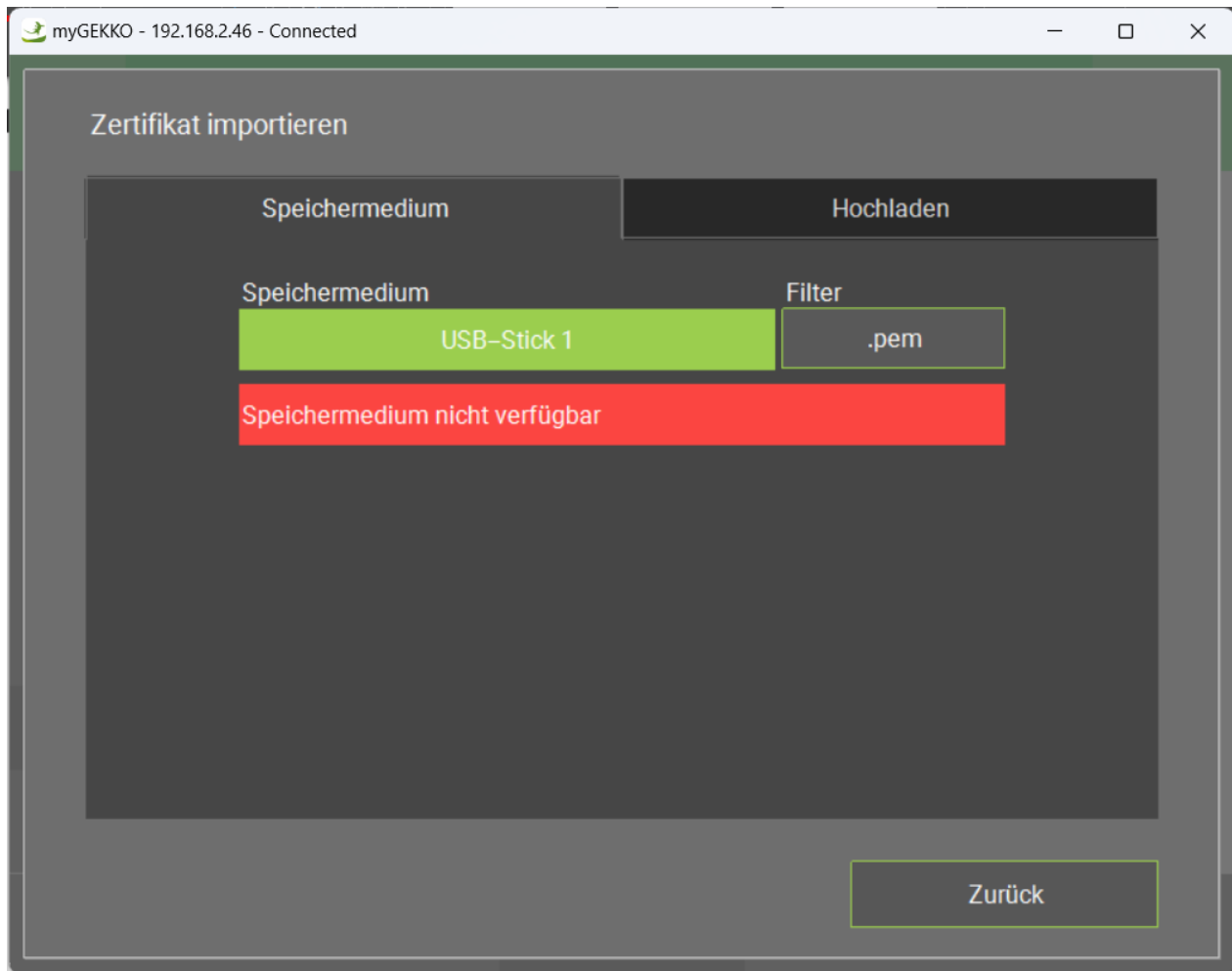
Apr 2021

1 / 2
Apr 2021

Sep 2021

[NOTE: see Demo 2 to create your own certificate.](#)





It needs to have format .pem, and it has to be renamed to "mqtt_certificate.pem". Then, switch the "SSL connection" field to "Activated" to enable the SSL connection.

Since we imported a certificate, we may also need to enable the options "Use client certificate" and/or "Check server certificate". It depends on the service you are using. In our case, for HiveMQ, we need to enable both of them.

(include a picture with these two options enabled)



myGEKKO - 192.168.2.46 - Connected

MQTT Cloud Konfiguration

Broker Adresse	cbce363fd16942528fd8bb7a031f6f3b.s2.eu.hivemq.cloud		
Port	8883	Rechte	Nur Lesen
Benutzername	myGEKKO_manager	Passwort	*****
SSL Verbindung	Aktiviert	Zertifikat importieren >	
Topic header	Aus		

Verbindungsstatus
Nicht verbunden

Zurück

If you want to use the topic header, then enable the topic header option. A textbox will appear, where the topic should be written. We can write "Villa_Franzelin" in the textbox, for example:



myGEKKO - 192.168.2.46 - Connected

MQTT Cloud Konfiguration

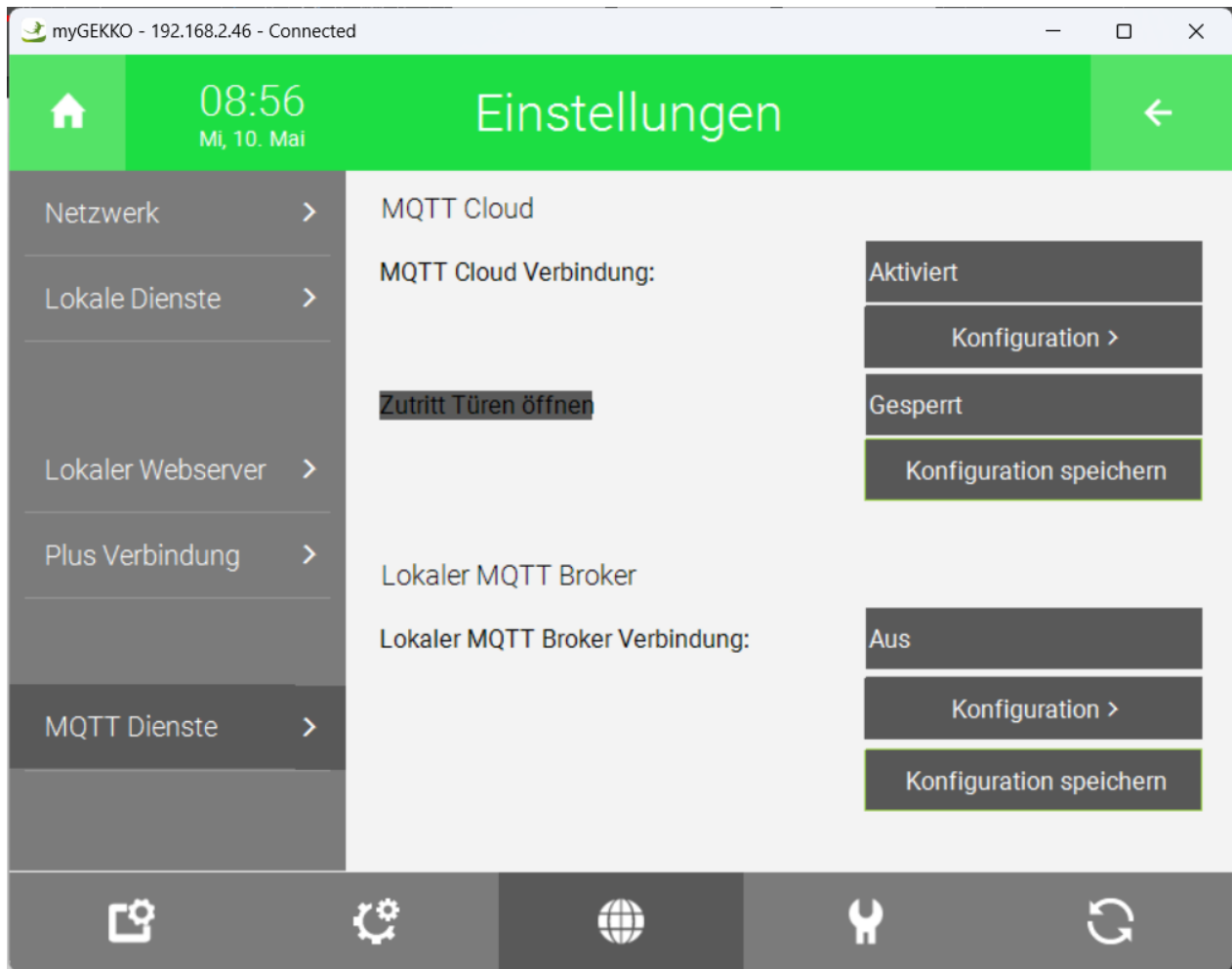
Broker Adresse	cbce363fd16942528fddb7a031f6f3b.s2.eu.hivemq.cloud		
Port	8883	Rechte	Nur Lesen
Benutzername	myGEKKO_manager	Passwort	*****
SSL Verbindung	Aktiviert	Zertifikat importieren >	
Topic header	Aktiviert	Villa_Franzelin	

Verbindungsstatus

Ok Zurück

Finally, go back to the settings, and enable the MQTT Broker connection by switching the option from "Off" to "Activated".





Go again to the configuration to check that the connection status is OK.



myGEKKO - 192.168.2.46 - Connected

MQTT Cloud Konfiguration

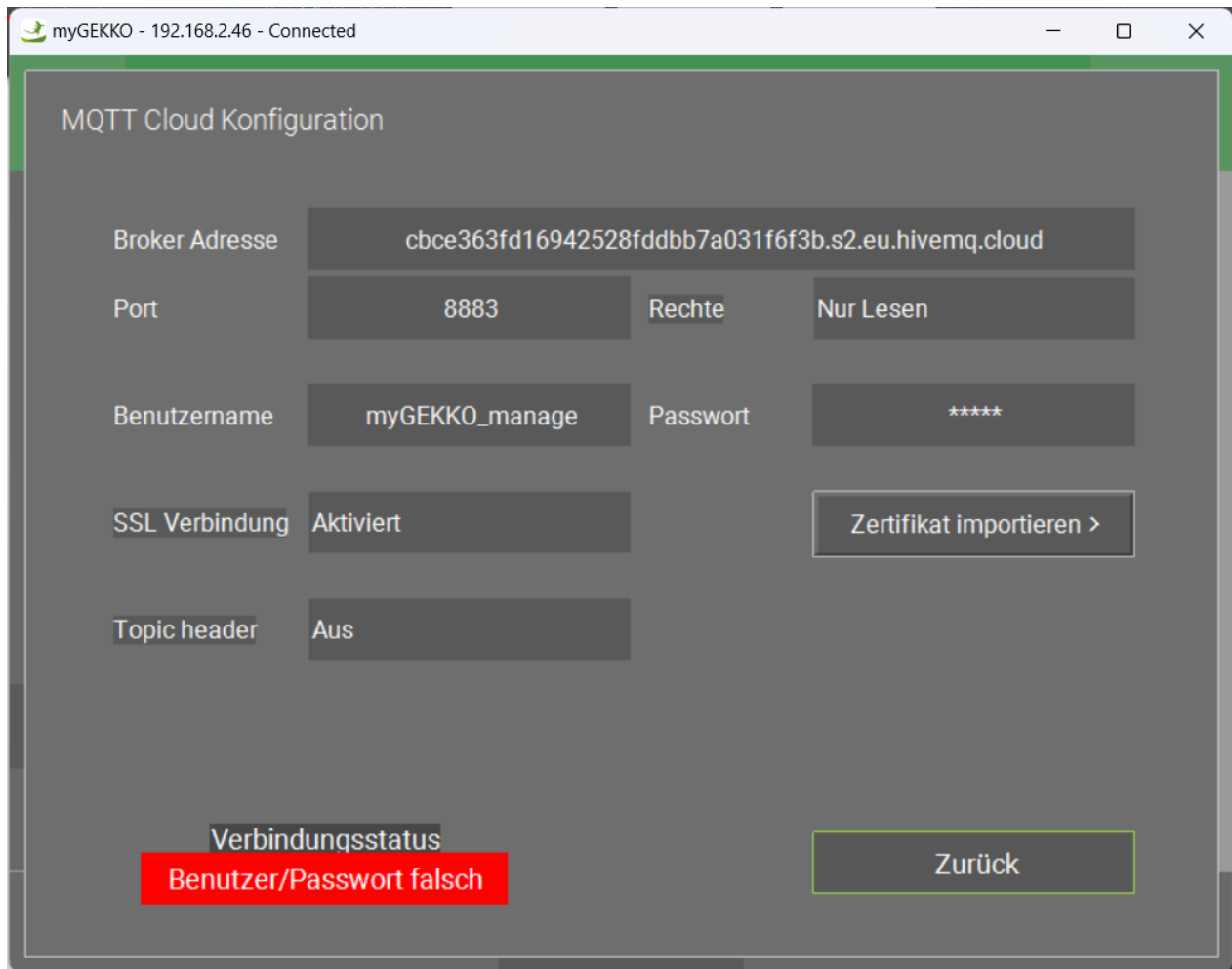
Broker Adresse	cbce363fd16942528fd8bb7a031f6f3b.s2.eu.hivemq.cloud		
Port	8883	Rechte	Nur Lesen
Benutzername	myGEKKO_manager	Passwort	*****
SSL Verbindung	Aktiviert	Zertifikat importieren >	
Topic header	Aus		

Verbindungsstatus

Ok Zurück

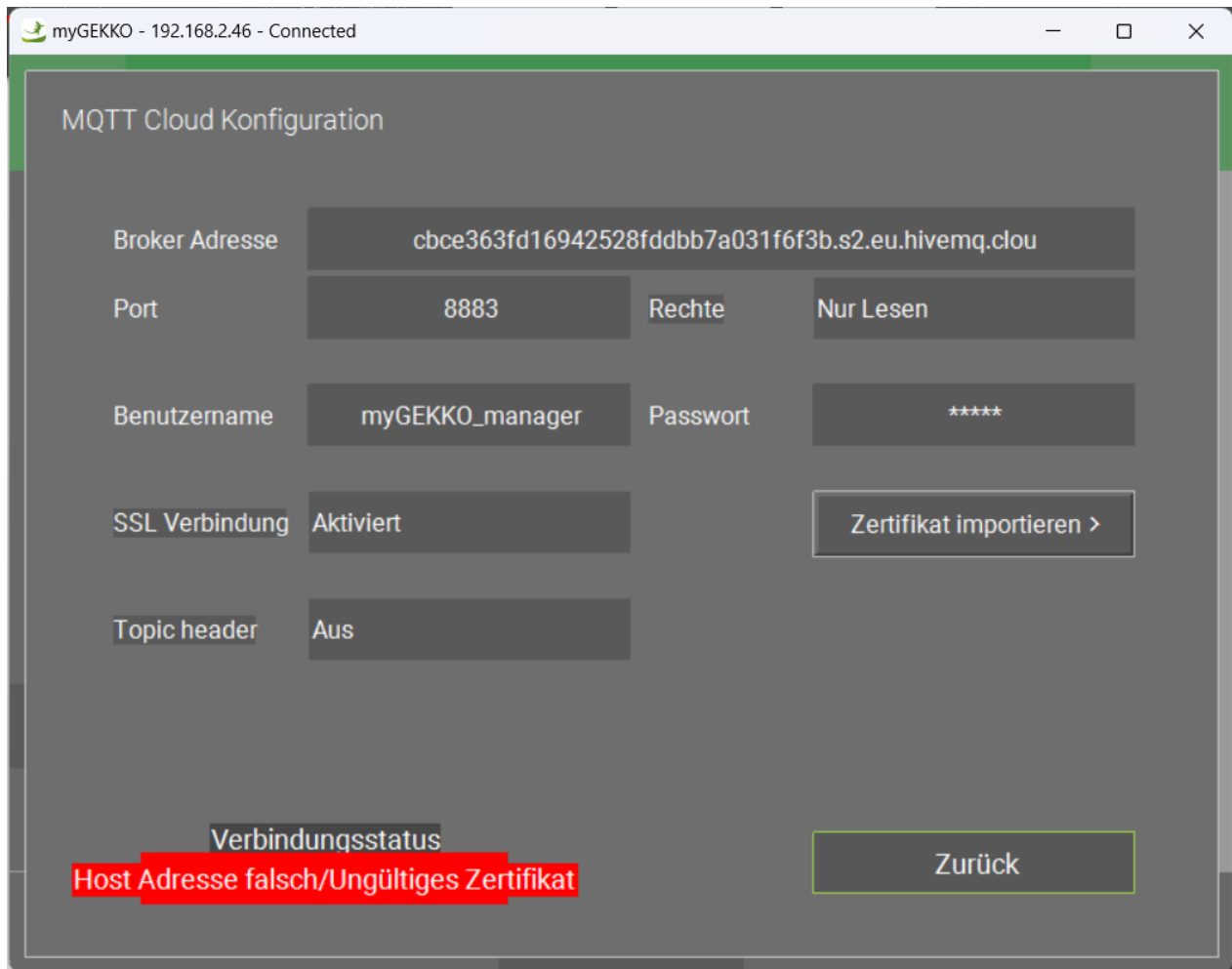
To double check the connection and that everything is working, try to change the username. For example, by removing the last "e" (mygekko_devic)





Then, we get an error message saying "Wrong pass/user". Now, if we change the address:





Then, we get an error message saying "Host not found". Now, if we change the port number:



myGEKKO - 192.168.2.46 - Connected

MQTT Cloud Konfiguration

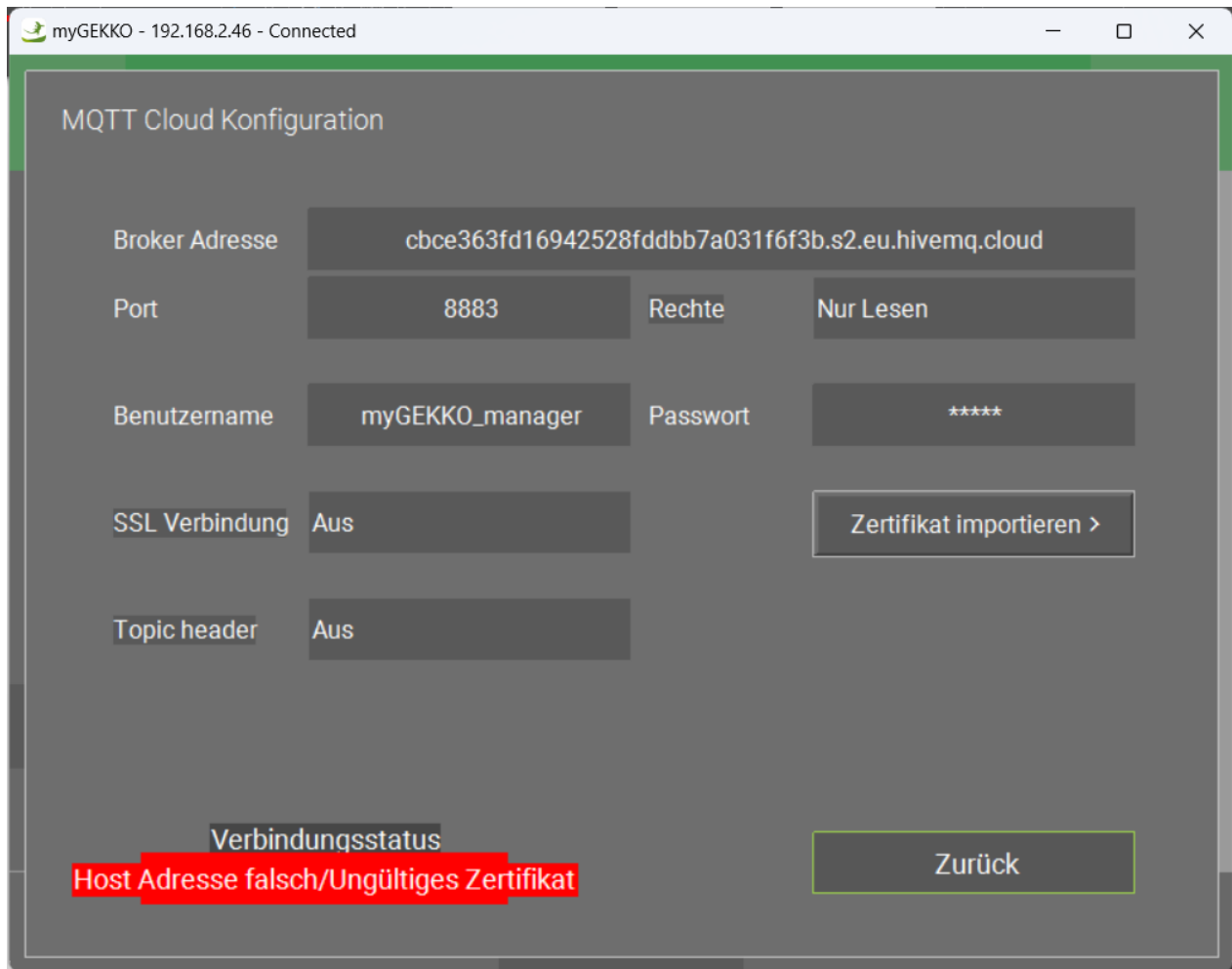
Broker Adresse	cbce363fd16942528fddeb7a031f6f3b.s2.eu.hivemq.cloud		
Port	888	Rechte	Nur Lesen
Benutzername	myGEKKO_manager	Passwort	*****
SSL Verbindung	Aktiviert	Zertifikat importieren >	
Topic header	Aus		

Verbindungsstatus
Nicht verbunden

Zurück

Then, we get an error message saying "Not connected". Now, if we disable the SSL connection:



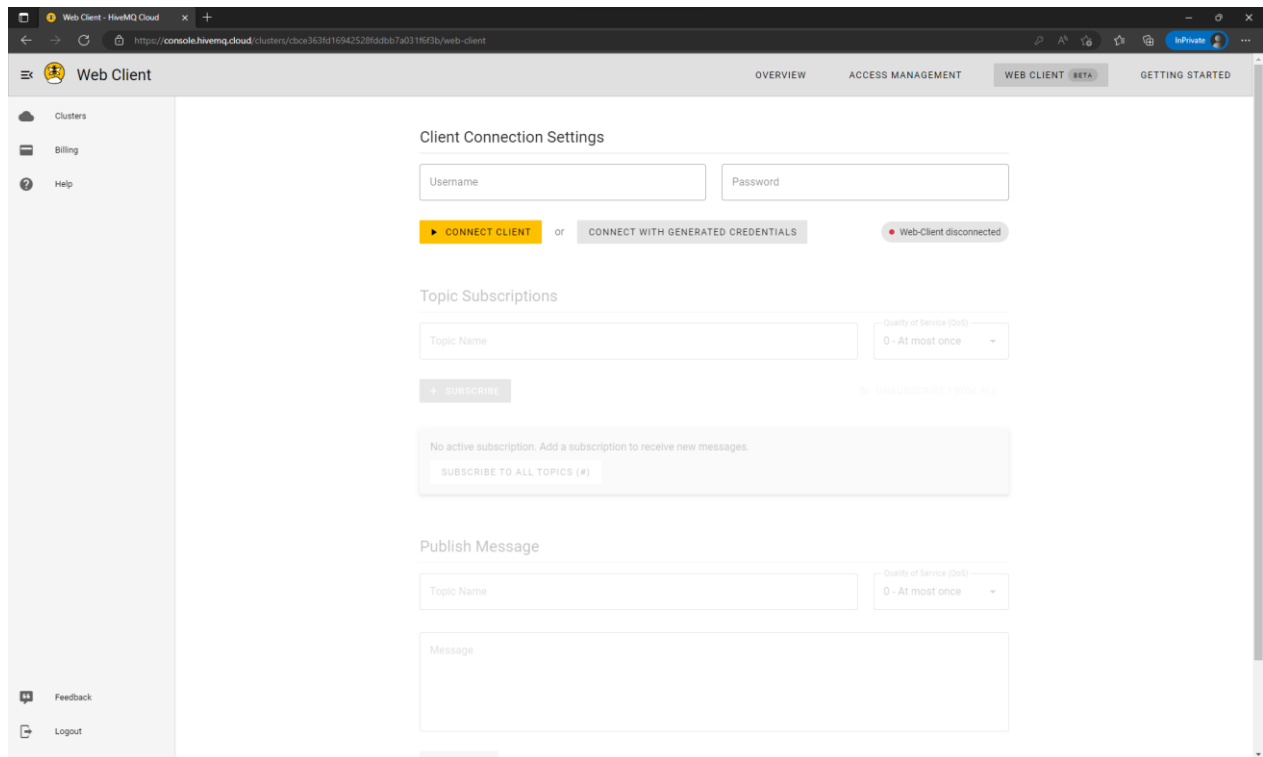


Then, we get an error message saying "Host not found".

If we want to be clients and subscribe to a topic, we need to create the connection to the broker through either a local or web clients.

To do it online, in the HiveMQ Cloud, go to Web Client.





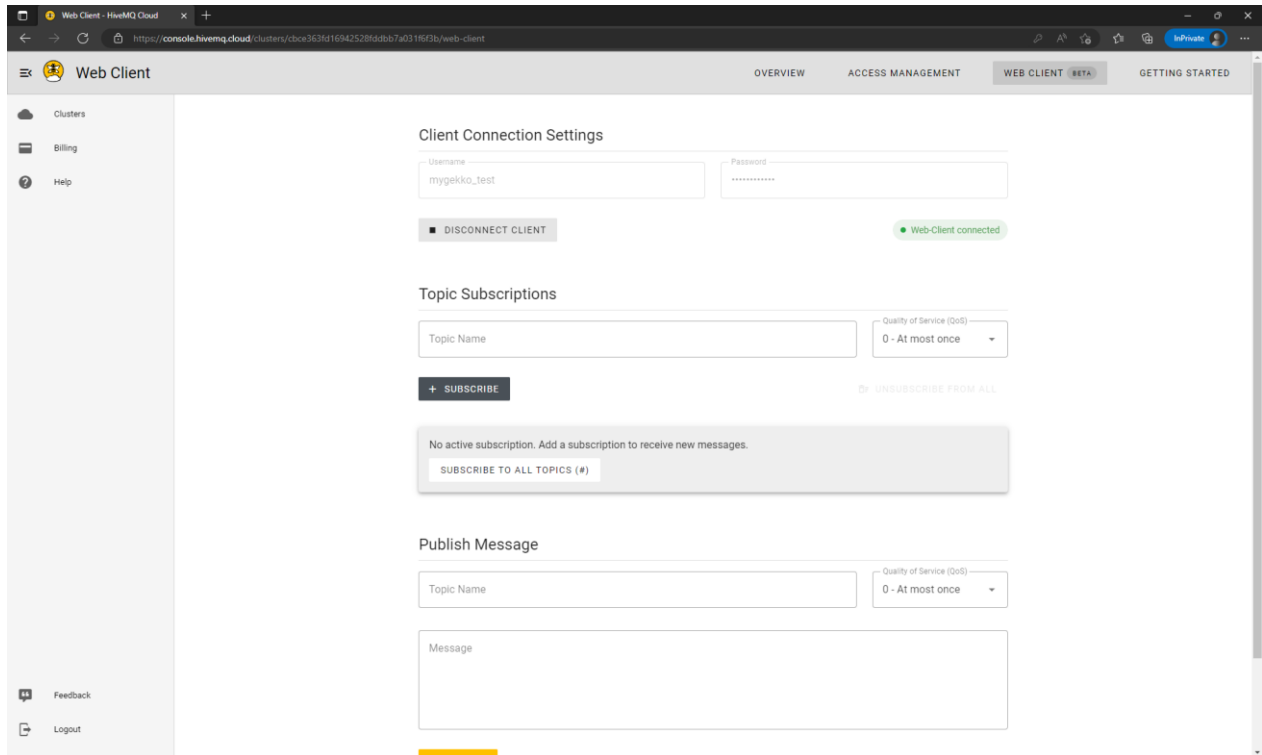
Then, in “Client Connection Settings”, add the username and password that we already created before:

- Username: mygekko_test
- Password: mygekko_test

And click on the yellow button “Connect client”.

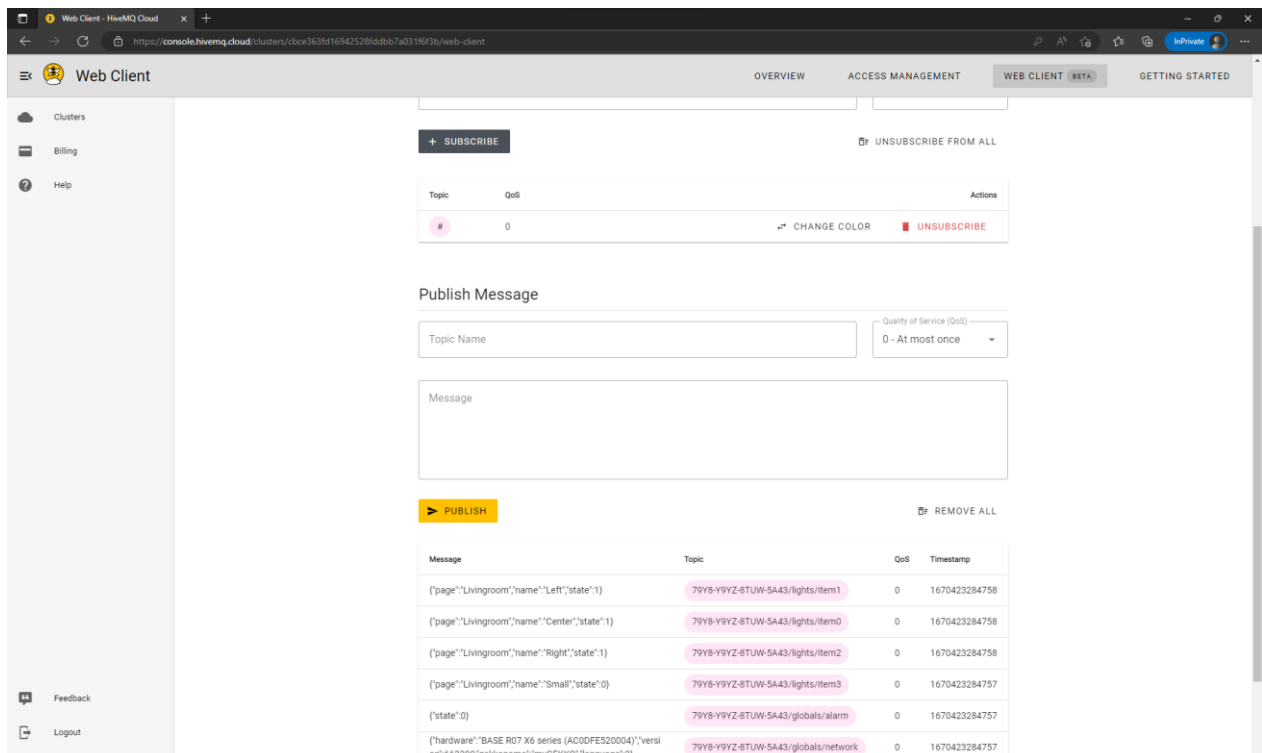
Then, if everything was ok, the connection is established:



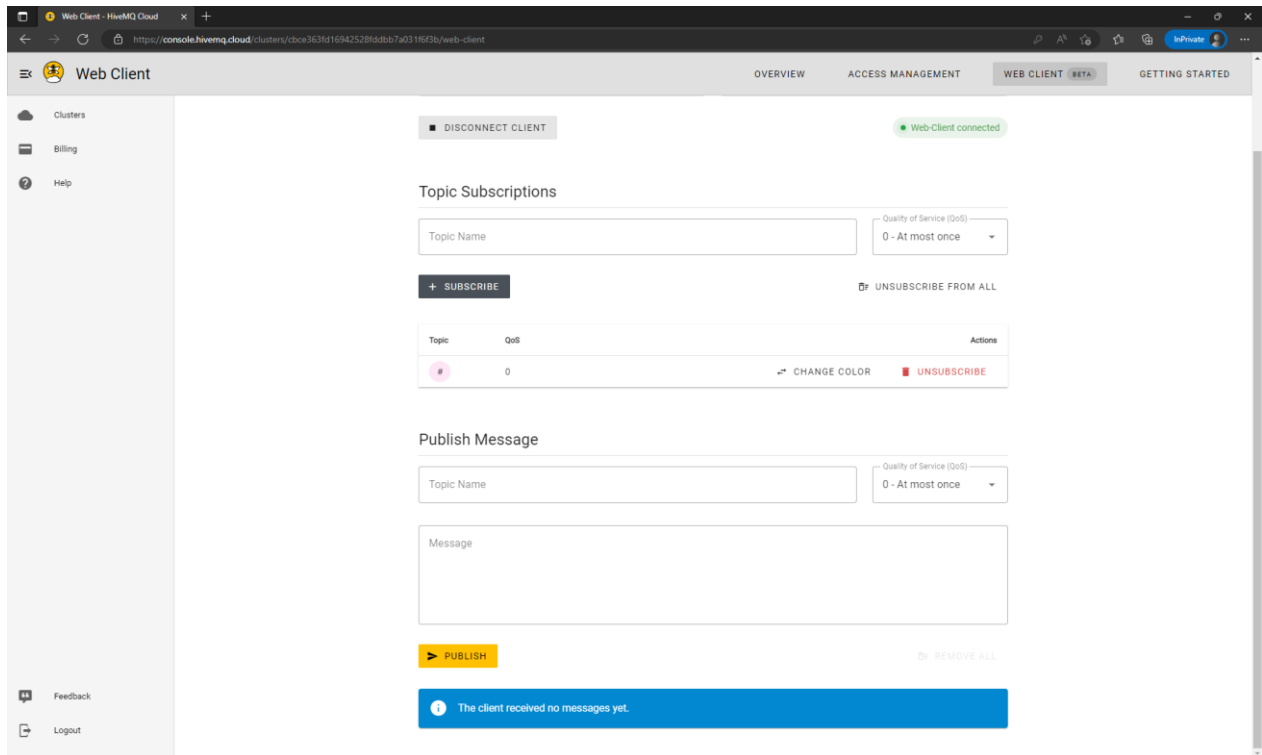


To get all the messages that myGEKKO has published to the broker, click on “Subscribe to all topics(#)”.

Then, a list of subscribed topics will appear. In this case, the topic “#” is the only one that we are subscribed. At the bottom of the screen, the messages retained at the broker should be printed:

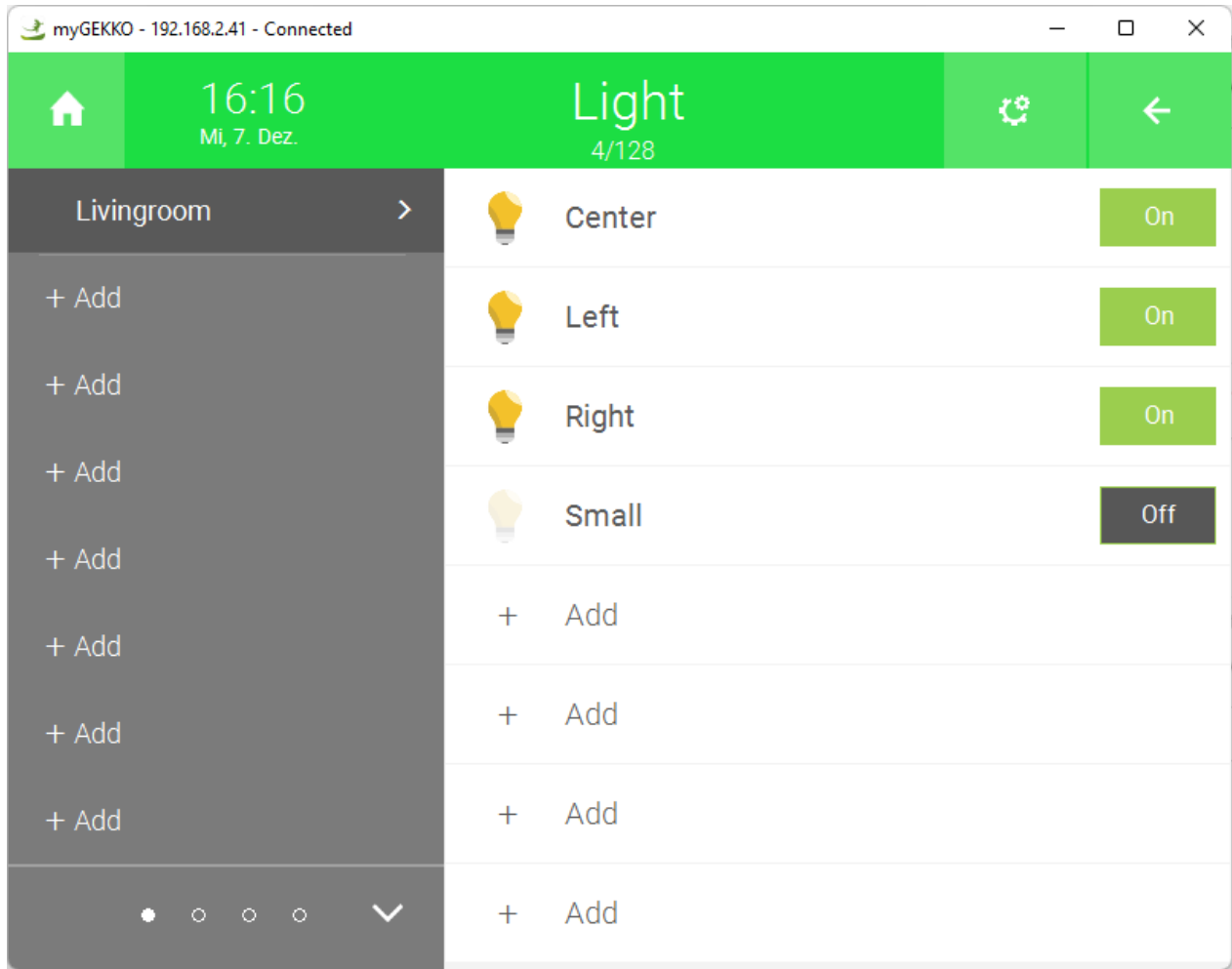


To check that the connection is working, we first clear this messages by clicking on “Remove all”.



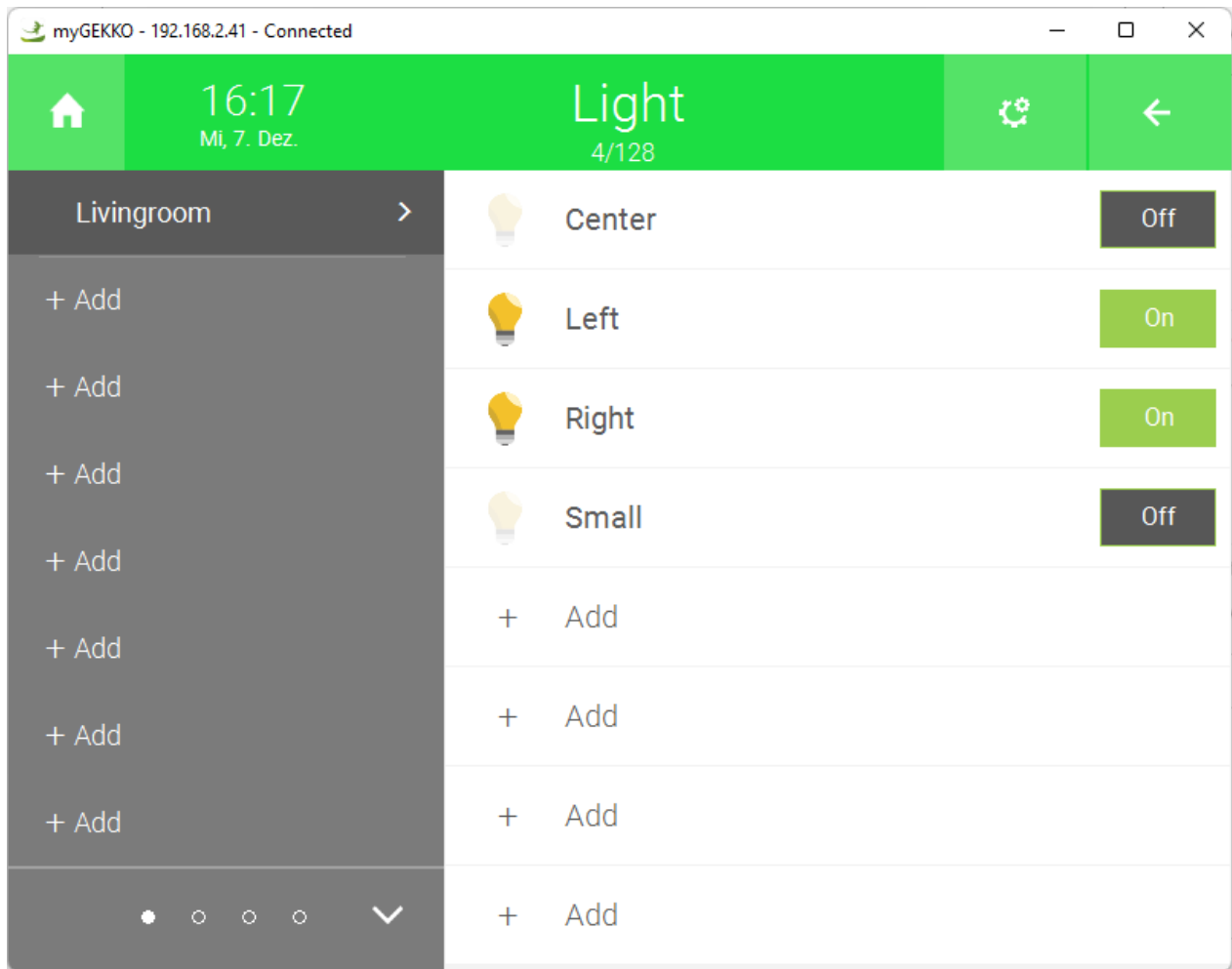
Then, go to myGEKKO, and change the state of a light. For example, the Center light of the Livingroom is ON:





And we switch it off.





Then, the client at HiveMQ cloud Web client should have received this message. Checking the messages:



The screenshot shows the HiveMQ Web Client interface. The top navigation bar includes 'OVERVIEW', 'ACCESS MANAGEMENT', 'WEB CLIENT BETA', and 'GETTING STARTED'. The main content area is divided into two sections: 'Topic Subscriptions' and 'Publish Message'. The 'Topic Subscriptions' section has a 'DISCONNECT CLIENT' button and a 'Web-Client connected' status indicator. It features a 'Topic Name' input field, a 'Quality of Service (QoS)' dropdown set to '0 - At most once', a '+ SUBSCRIBE' button, and an 'UNSUBSCRIBE FROM ALL' button. Below this is a table with columns for 'Topic', 'QoS', and 'Actions'. The table contains one entry with a pink topic name and a '0' QoS, with 'CHANGE COLOR' and 'UNSUBSCRIBE' actions. The 'Publish Message' section has a 'Topic Name' input field, a 'Quality of Service (QoS)' dropdown set to '0 - At most once', a 'Message' text area, a 'PUBLISH' button, and a 'REMOVE ALL' button. Below this is a table with columns for 'Message', 'Topic', 'QoS', and 'Timestamp'. The table contains one entry with a JSON message, a pink topic name, a '0' QoS, and a timestamp.

Topic	QoS	Actions
#	0	CHANGE COLOR UNSUBSCRIBE

Message	Topic	QoS	Timestamp
["page":"Livingroom","name":"Center","state":0]	79Y8-Y9YZ-8TUV-5A43/lights/Item0	0	1670423508351

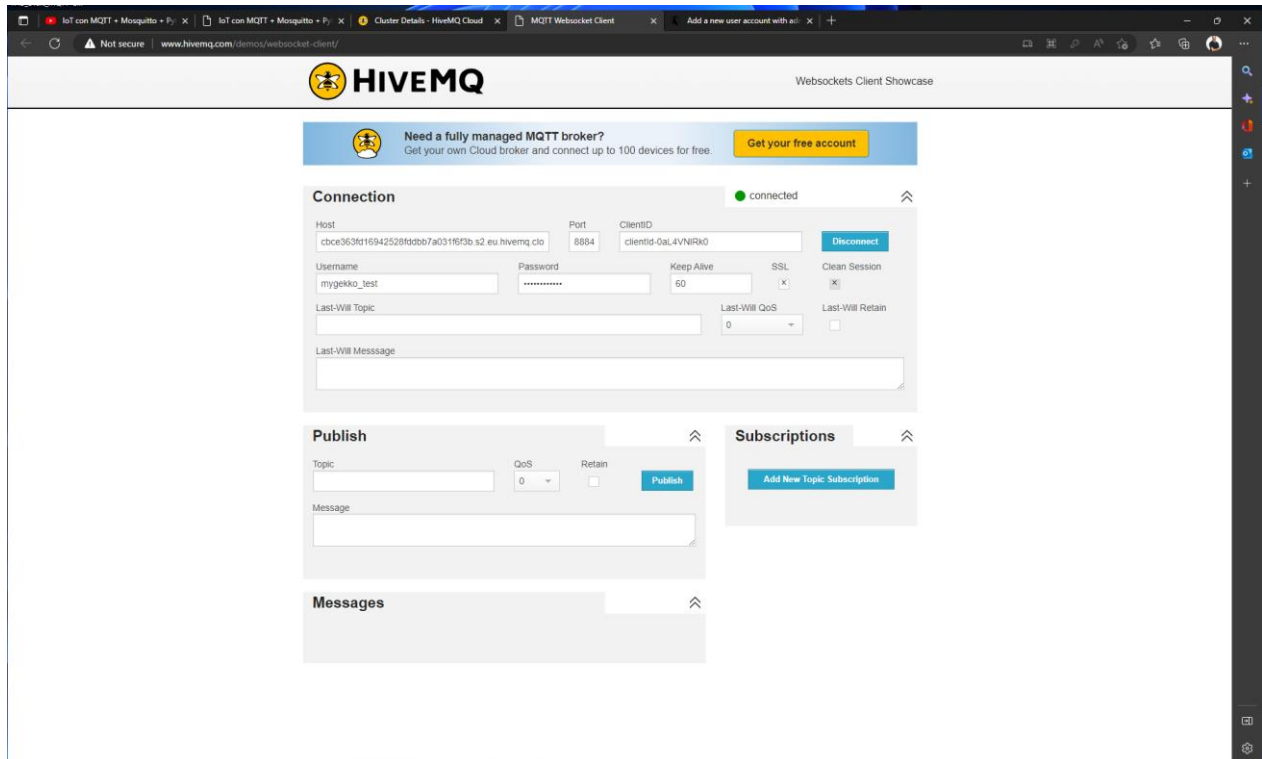
We can also run the discovery. Write the message in the Topic:

79Y8-Y9YZ-8TUV-5A43/lights/discovery

Alternatively to the web client, HiveMQ also offers a websocket client

<http://www.hivemq.com/demos/websocket-client/>



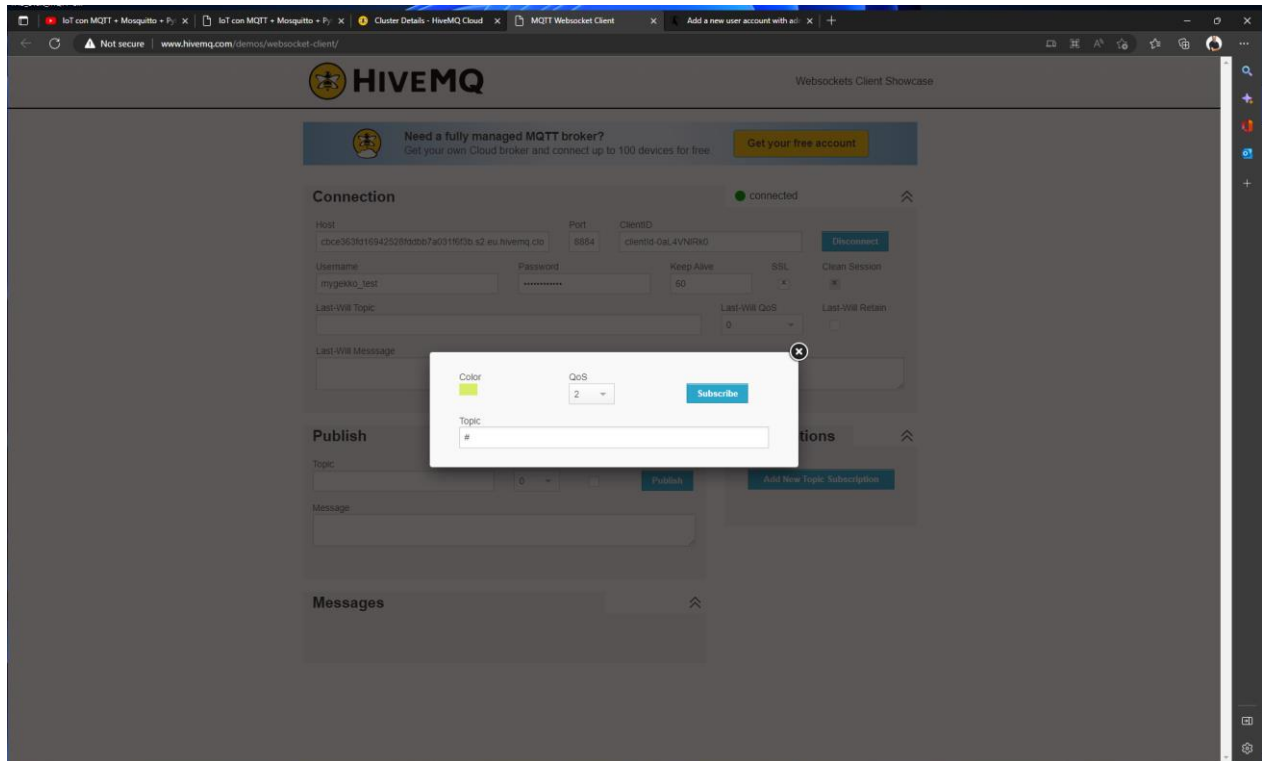


To connect to myGEKKO:

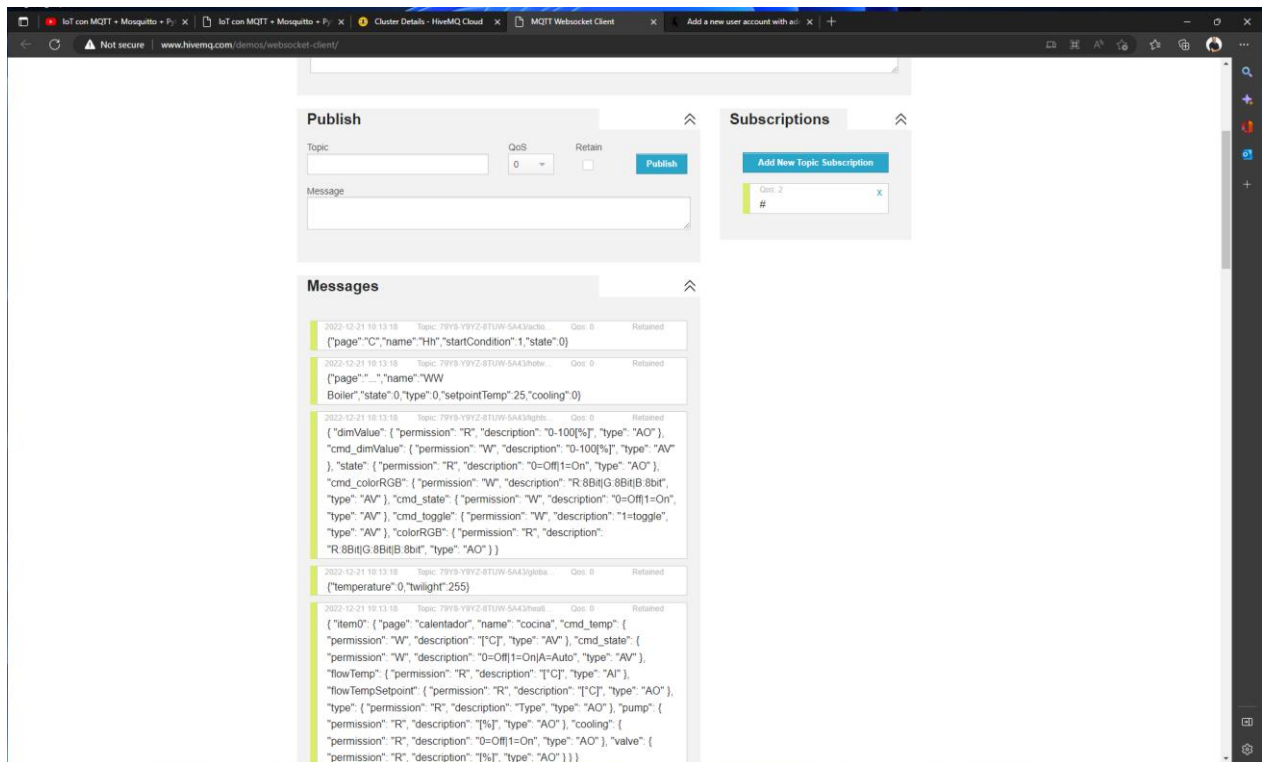
- Host: cbce363fd16942528fddb7a031f6f3b.s2.eu.hivemq.cloud
- Port: 8884
- ClientID: test (it is not important)
- Username: mygekko_test
- Password: mygekko_test
- Keep Alive: 60 (by default)
- SSL: activated by checking on it (X)
- Rest of parameters: empty.

Then, if we want to subscribe to a topic, click on Add new topic subscription and add "#":





Then, all the messages will appear:



Demo 2: creating your own client certificate



Useful links and bibliography

Client on windows

<https://www.hivemq.com/blog/mqtt-cli/>

Create your own certificates

Setting up TLS for Your Cloud-based MQTT Broker (youtube video:

<https://www.youtube.com/watch?v=LZ2U-IE0QFo>)

<https://www.hivemq.com/docs/hivemq/4.9/user-guide/howtos.html>

FAQ about the certificates

<https://community.hivemq.com/t/frequently-asked-questions/514>

Documentation about the MQTT library and certificates

https://www.ibm.com/docs/en/ibm-mq/7.5?topic=SSFKSJ_7.5.0/com.ibm.mq.javadoc.doc/WMQMQxrCClasses/struct_m_q_t_t_client__s_s_l_options.htm

Documentation about Paho

<https://wiki.eclipse.org/Paho#Development>

HiveMQ connect client through Command Line Interface

<https://www.hivemq.com/docs/hivemq-cloud/introduction.html#connect-client>

IoT con MQTT + HiveMQ + BASH (youtube video):

<https://www.youtube.com/watch?v=scTYxjzRZJI>

Mosquitto download

<https://mosquitto.org/download/>

<https://mosquitto.org/documentation/>

Mosquitto user problem when running on Linux

<https://stackoverflow.com/questions/35577615/mosquitto-auth-plugin-database-issue#:~:text=Error%3A%20Invalid%20user%20%27mosquitto%27.%20is%20because%20you%20are,user%20defined%20on%20the%20system%20hence%20the%20error.>

Using The Mosquitto_pub and Mosquitto_sub MQTT Client Tools- Examples (SUPER USEFUL)

http://www.steves-internet-guide.com/mosquitto_pub-sub-clients/



How to install and secure mosquitto on Ubuntu

<https://www.arubacloud.com/tutorial/how-to-install-and-secure-mosquitto-on-ubuntu-20-04.aspx>

<https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-the-mosquitto-mqtt-messaging-broker-on-ubuntu-16-04>



myGEKKO | Ekon GmbH

St. Lorenznerstraße 2
I-39031 Bruneck (BZ)

Tel. +039 0474 551820

info@my.gekko.com

www.my-gekko.com

Stand: 21.12.2023

