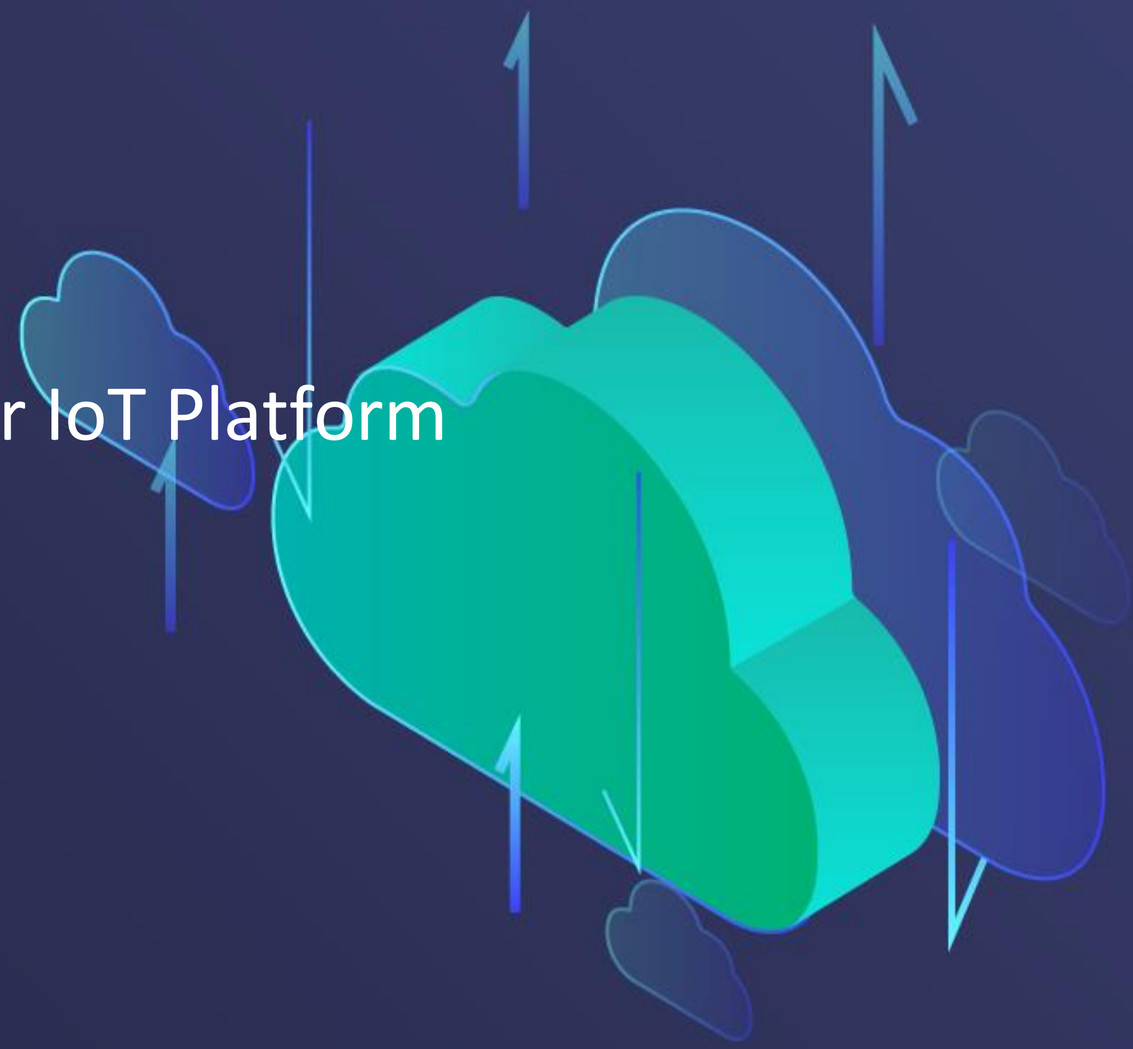
EMQ X Cloud Webinar

EMQ X Cloud

The Quick and Easy Way to Setup Your IoT Platform



December 16th
9:00am EST / 3:00pm CET / 2:00pm UTC

Speaker:
Kary Ware, Sales Engineer @EMQ

Agenda

- EMQ X Cloud benefits and features
- MQTT protocol advantages
- Getting started with EMQ X Cloud
- Demo: EMQ X Cloud in an IoT scenario
  - Process and stream data to MySQL and Kafka
- Summary and Q & A

EMQ X Cloud Benefits and Features

High Performance, Easy-to-Deploy Message Broker

EMQ X CLOUD

- High capacity and low latency
  - Tested at 10 million connections
  - Millions of messages per second
  - Latency < 100 ms (< 10 ms QoS 0)
- Fast deployment – setup in just a few clicks
- Easy to manage – we do the work for you
- Pay as you go – pay for what you need
- Cloud native
- Fully managed v3.1.1 and v5.0 MQTT broker service
- Other supported protocols include, MQTT-SN, CoAP, LwM2M

EMQ

Choose from Three Plans

EMQ X CLOUD

Standard

Start at **$ 0.18** per hour

Save 15% if you prepay annually

Create Now

- Single node
- Up to 10,000 connections
- Contains 100G traffic
- Support WebHook
- Support MQTT Bridge
- SLA 99%
- 8/5 technical support

Professional

Start at **$ 0.99** per hour

Save 15% if you prepay annually

Create Now

- Multiple node cluster
- Exclusive resources
- Up to 100,000 connections
- Contains up to 1T traffic
- Support Rule Engine
- Support VPC Peering
- SLA 99.99%
- 24/7 technical support

Unlimited

Contact business for customized solutions

Contact Us

- Physical resource isolation
- Device management
- Device model
- Device Shadow
- Cloud Edge Collaboration
- SLA 99.99%
- Consulting service

Choice of Cloud Platform, Region and Connections

EMQ X CLOUD

Example for Professional Plan

Choose Cloud Platform

AWS

Azure

Google Cloud

Region depends on cloud platform

Choose Region

US East (N. Virginia)

US West (Oregon)

EU (Ireland)

EU (Frankfurt)

Asia Pacific (Singapore)

Asia Pacific (Mumbai)

Choose Specification

5,000 CONN / 10,000 TPS

10,000 CONN / 20,000 TPS

50,000 CONN / 50,000 TPS

100,000 CONN / 100,000 TPS

> 100,000 CONN, Contact sales

Rule Engine: Process Messages in Real-Time

EMQ X CLOUD

All Plans

Modify and filter the topic messages in real time

Stream messages to external HTTP endpoints

Stream messages to MQTT brokers (bridge)

Professional and Unlimited

Persist messages to databases:

MySQL, MongoDB, InfluxDB, PostgreSQL, ClickHouse, etc.

Stream messages to other message servers:

Kafka, RabbitMQ, Pulsar, etc.

Security: Authentication and Authorization

Authentication: user/pw, HTTP, JWT, LDAP, or from databases:

Redis, MySQL, PostgreSQL, MongoDB. **Certificates:** SSL/TLS, PSK, X.509

Authorization: Fine granularity of control at topic level

Client ID

Client ID | Topic | Publish & Sul | Allow | + Add | + Import

Client ID	Topic	Action	Allowed	Actions
client3	sensor/#	Publish	Deny	
client3	sensor/#	Subscribe	Deny	
client2	sensor/+/data	Publish	Deny	
client2	sensor/+/data	Subscribe	Deny	
client1	sensor/1/data	Publish	Deny	
client1	sensor/1/data	Subscribe	Deny	

Dashboard: Performance Monitoring

EMQ X CLOUD

Instance status: Running

Running time: 167D, 04H, 13M

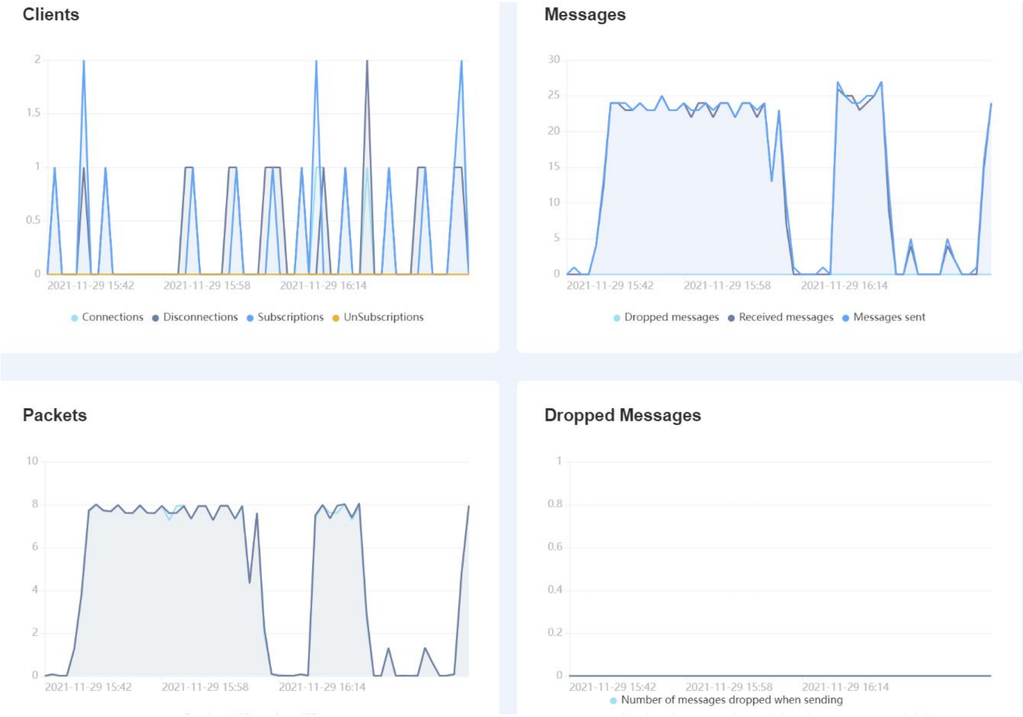
Number of messages sent: 210 per second

Number of messages received: 437 per second

Connections: 1931

Subscriptions: 6719

Topics: 10149



1 Hour | 1 Day | 7 Days | 1 Month | 1 Year

Dashboard: View Clients and Subscriptions

EMQ X CLOUD

Clients

sensor*

Username

Search | Reset | More

Client ID	Username	IP Address	Keepalive	Protocol	Connect Status	Connection At	
sensor3	sensor3	87.181.19.79:60146	60	MQTT	Connected	2021-11-30 11:16:17	Kick Out
sensor2	sensor2	87.181.19.79:60147	60	MQTT	Connected	2021-11-30 11:16:23	Kick Out
sensor1	sensor1	87.181.19.79:60145	60	MQTT	Connected	2021-11-30 11:16:13	Kick Out

Subscriptions

Client ID

Topic

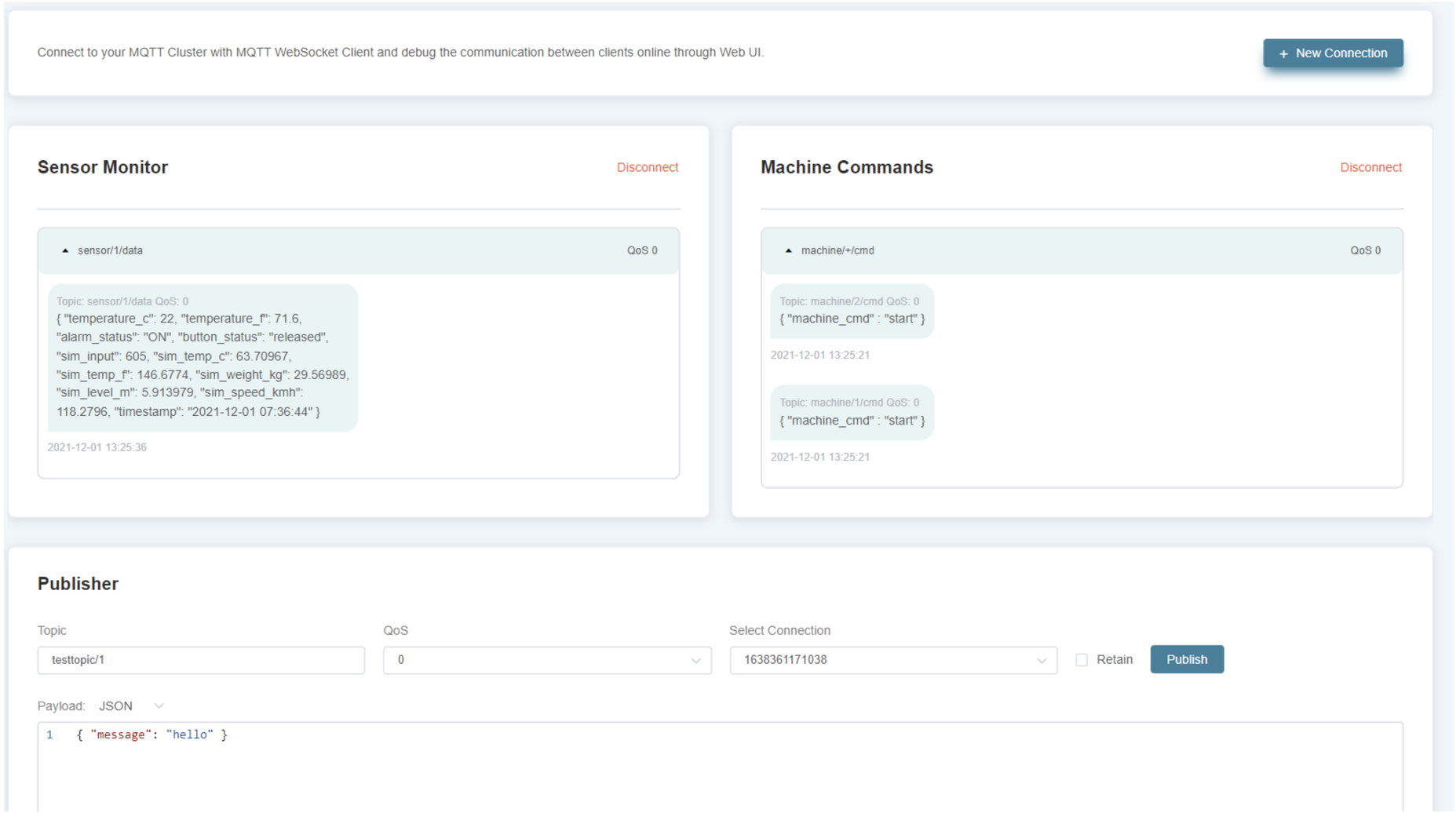
QoS

Shared subscription group name

Search | Reset | Collapse

Client ID	Topic	QoS
test1	sensor/1/data	0
test1	sensor/1/status	0
test3	sensor/-/data	0
test2	sensor/#	0

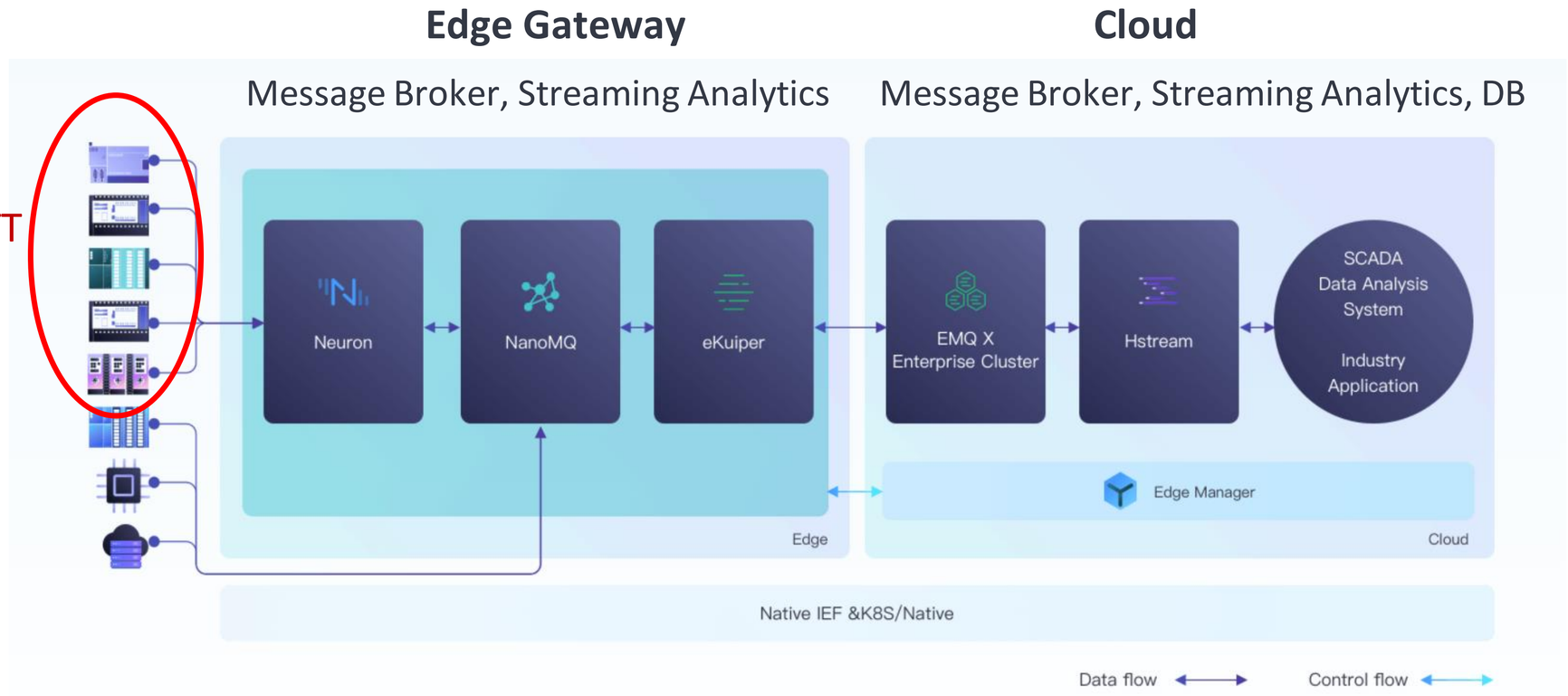
Dashboard: Websocket Based Clients



Great for testing !!

Full IoT Solutions for Edge to Cloud

EMQ X CLOUD



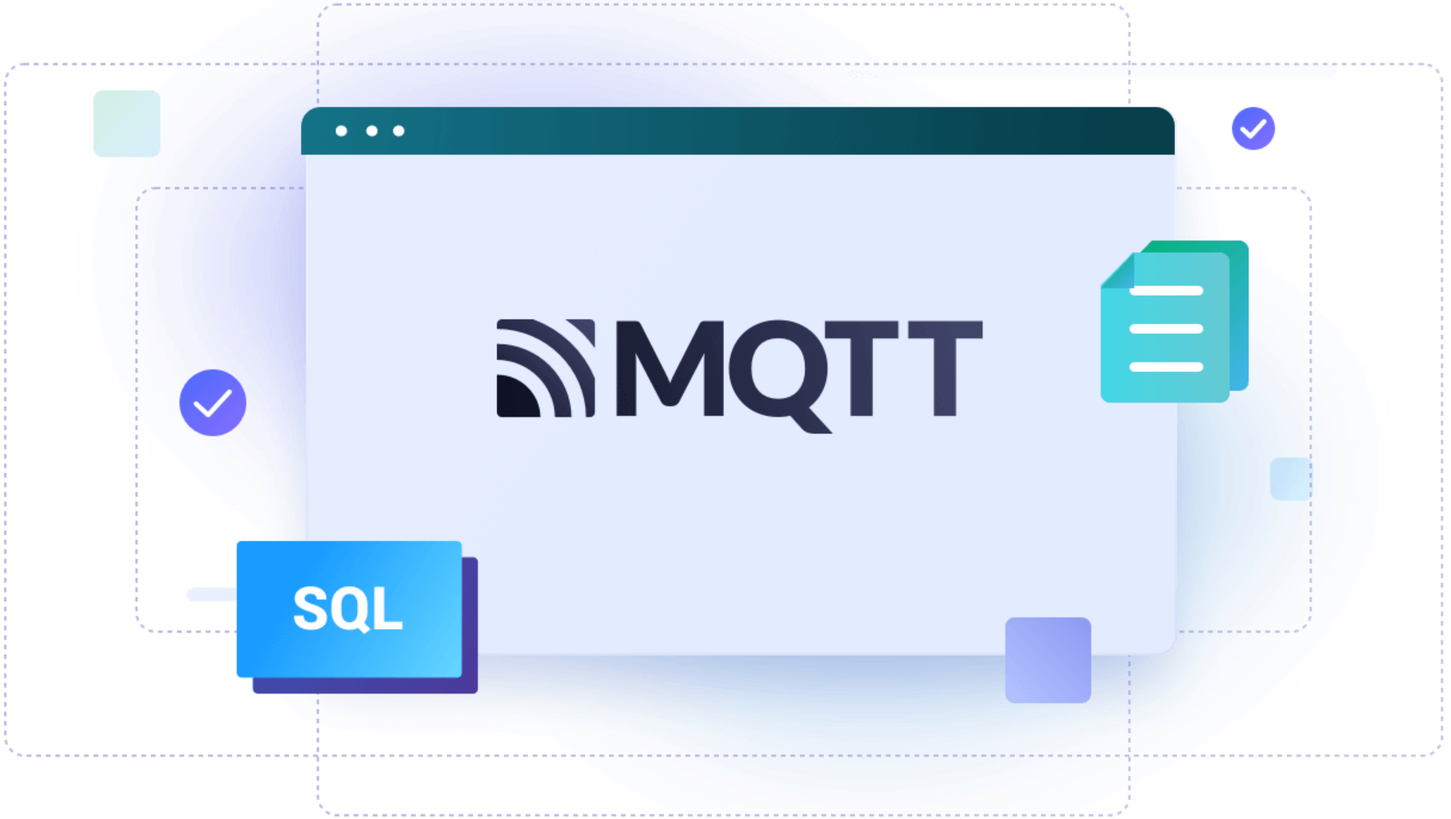
Neuron: Interface to non-MQTT industry protocols:
Modbus, OPCUA, IEC61850, IEC104, BACnet, Omron, Mitsubishi, Siemens, Allen-Bradley …

NanoMQ: Edge message broker. Supports MQTT, ZeroMQ, nanomsg/NNG

EKuiper: Edge real-time streaming analytics

HStreamDB: Cloud native streaming database with real-time analytics

MQTT Protocol Advantages

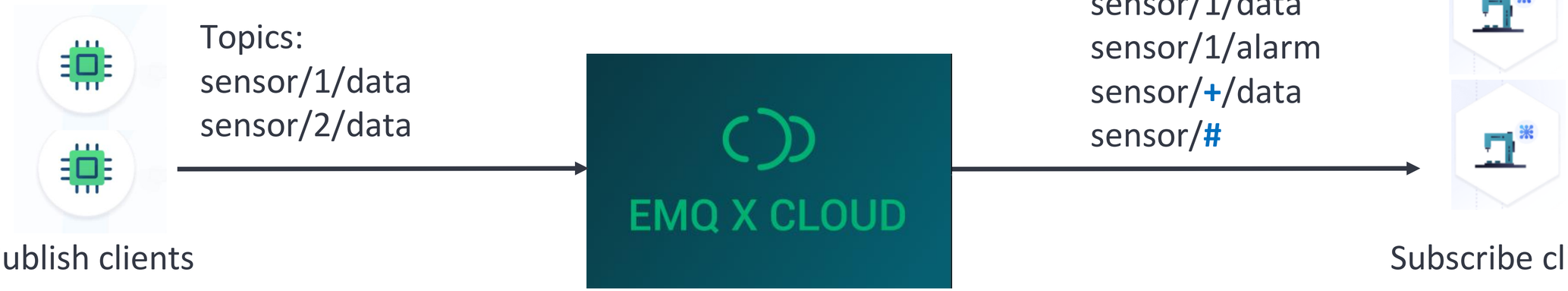


MQTT Advantages

Great for the Internet of Things

Based on Publish Subscribe Model

Decoupling between publishers and subscribers



Lightweight

Designed for devices
with limited hardware

Overhead is in bytes

Reliable Message Delivery over Unreliable Networks

Persistent sessions: Clean Session Flag: False

Retained message: Retain Flag: True

QoS with guaranteed delivery

Will message

Heartbeat mechanism KeepAlive: 60 secs (Example)

EMQ X has Full Support for MQTT

- 100% comptible with
  MQTT v3.1, v3.1.1 and v5.0 standards

- Full support for QoS 0, 1, 2
  MQTT message delivery

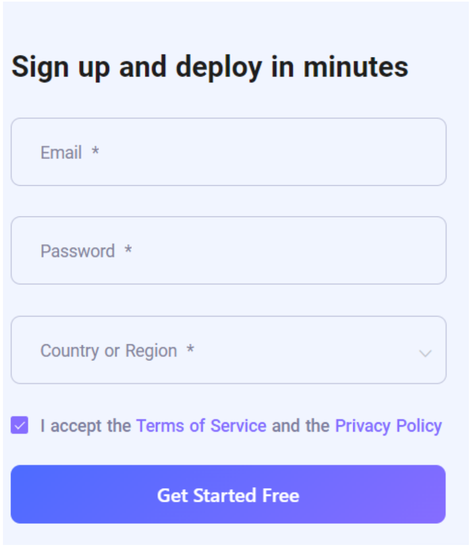
  QoS 0 = At most once – not guaranteed
  QoS 1 = At least once – may be duplicates
  QoS 2 = Exactly once

MQTT Applicable to Many Industries

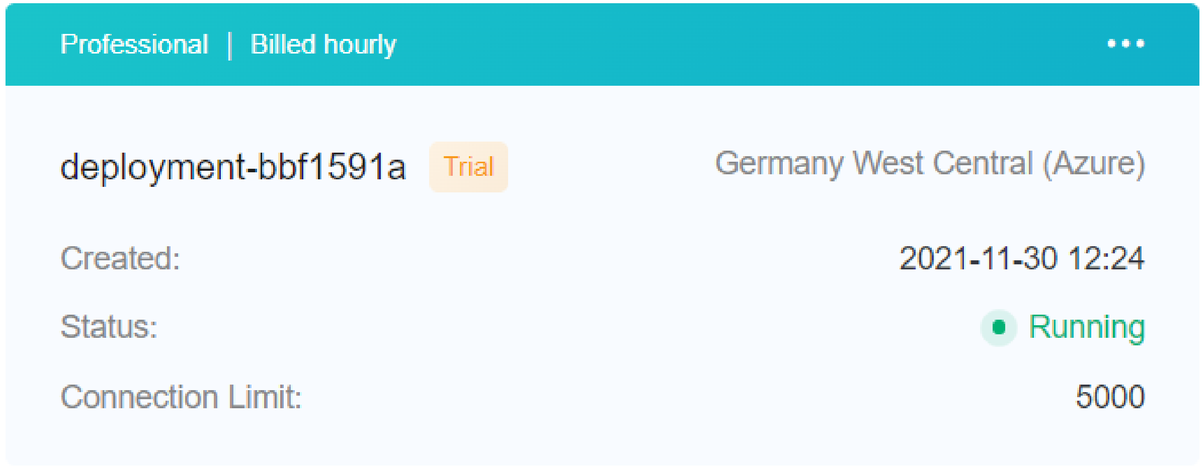
Some examples:

Unlimited possibilities and use cases!

Getting Started with EMQ X Cloud



Try Free →



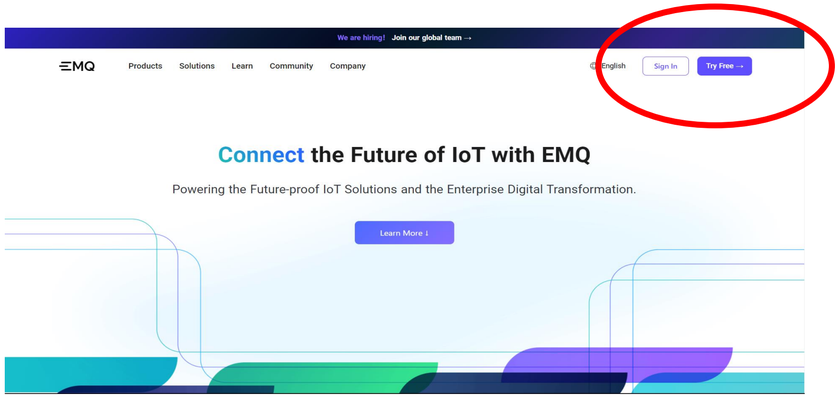
You don't need a credit card !!!

First Step to Getting a Free Trial

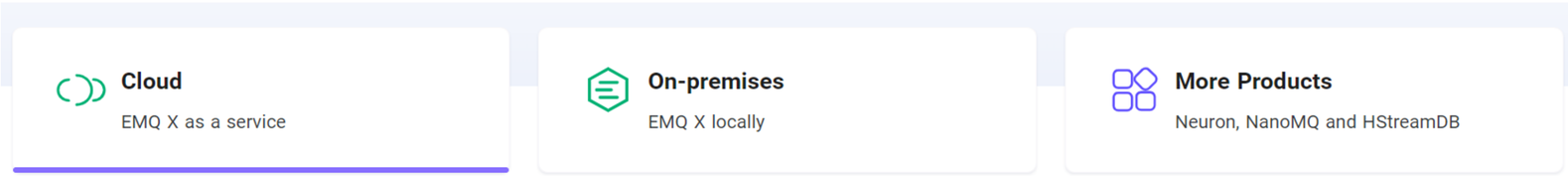
Try Free →

Go to www.emqx.com

and click **Try Free**

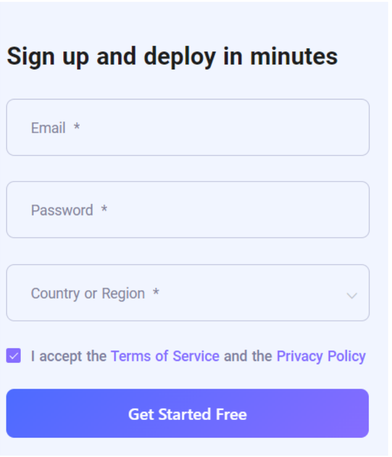


Choose **Cloud** option



Enter your email,

password, and region

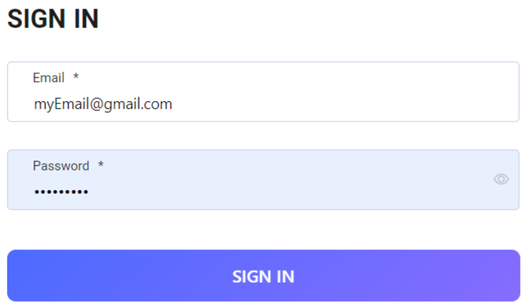


You don't need a credit card !!!

Sign in and Choose Plan

Try Free →

After verifying your email, you will be prompted to sign in.

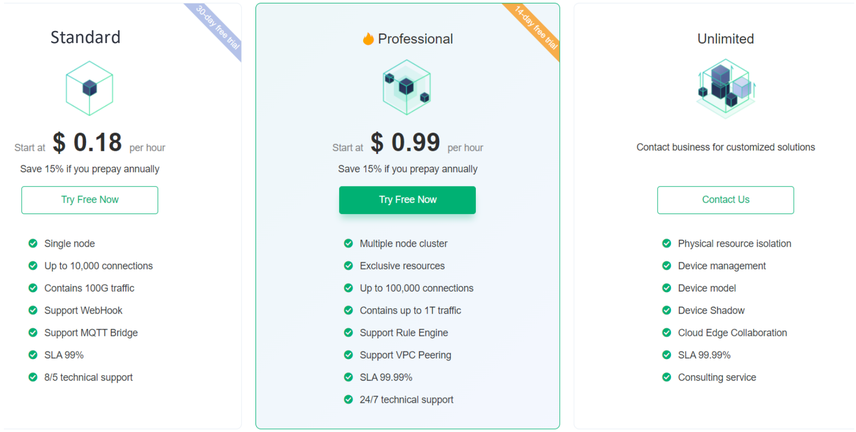


Trial length

Standard: 30 days

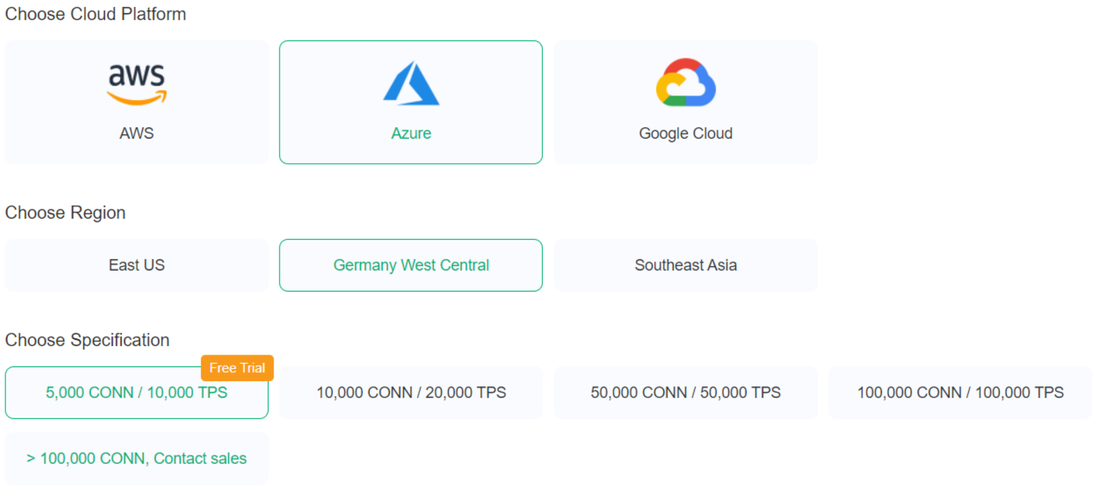
Professional: 14 days

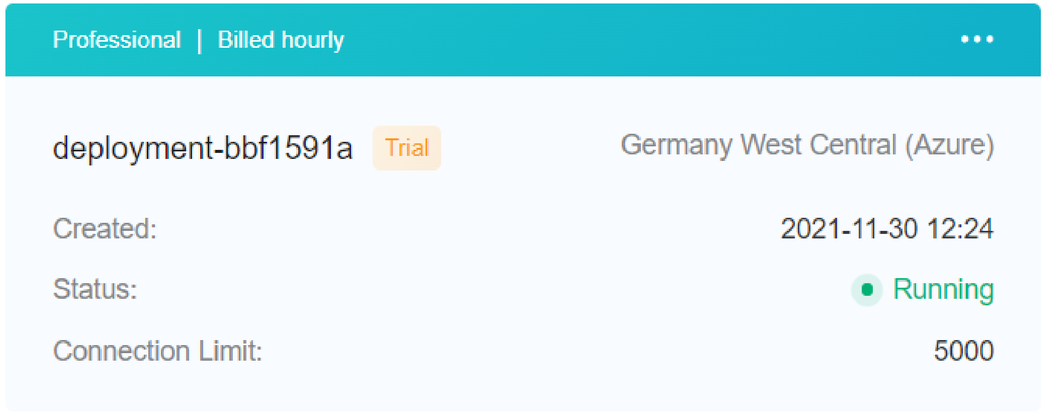
Then you can choose between **Standard** or **Professional**



Choose Cloud Platform and Region

Try Free →





For the trial,
you must choose
the least number of connections.

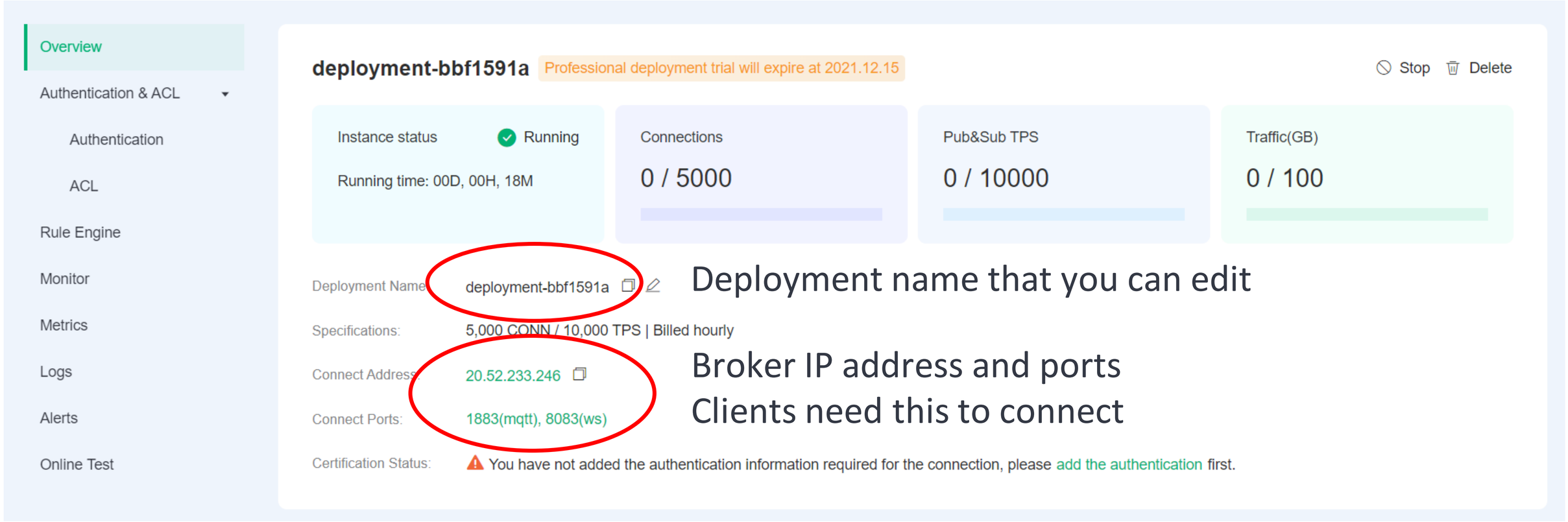
After agreeing to the license terms,
you will see your cloud deployment
starting up.

After about 10 minutes, it will be **Running**.

Click it to start using it.

Broker is Running: Overview

Try Free →



Deployment name that you can edit

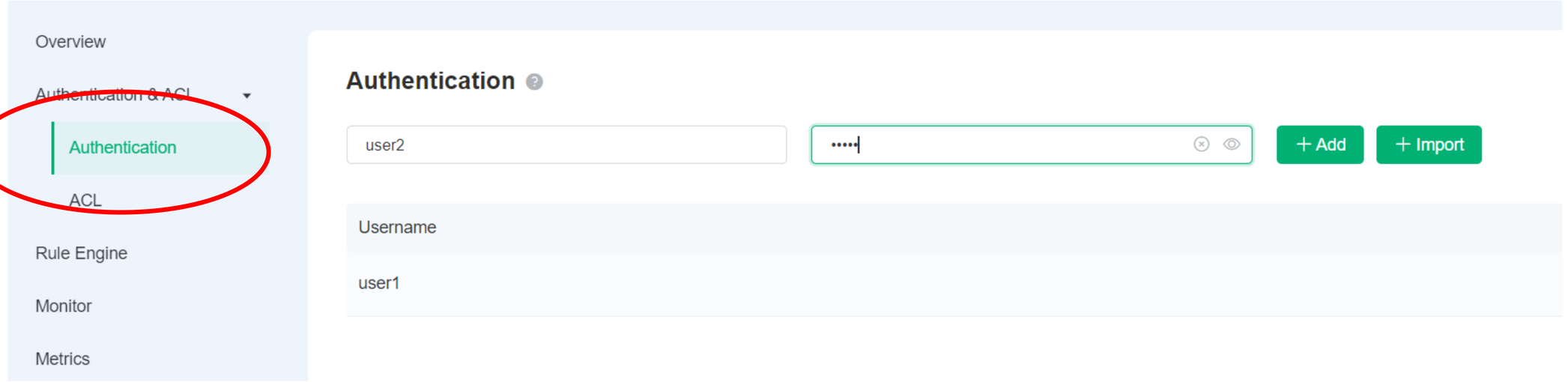
Broker IP address and ports
Clients need this to connect

Next is to add some users and then test the broker...

Broker Setup: Add a few Users

Try Free →

Add a few users that will act as publishing and subscribing cleints.



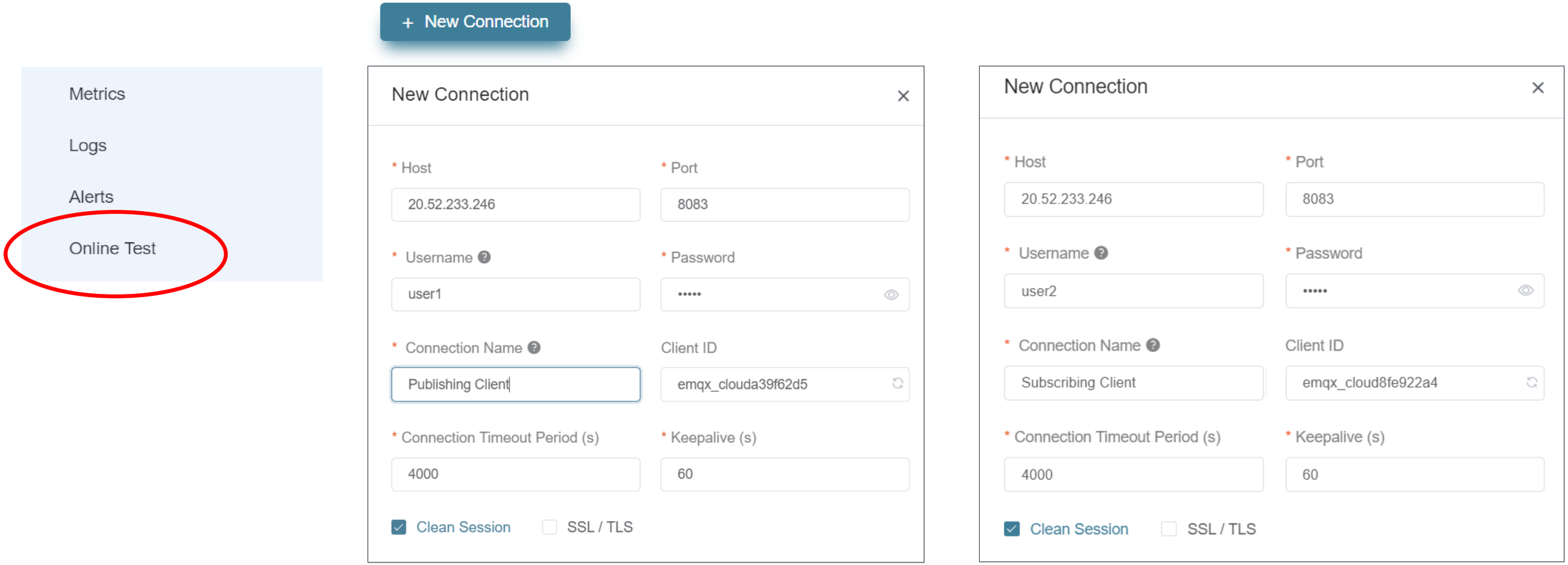
EMQ

Broker Setup: Add Two Clients

Try Free →

Click **Online Test** to open the online WebSocket client.

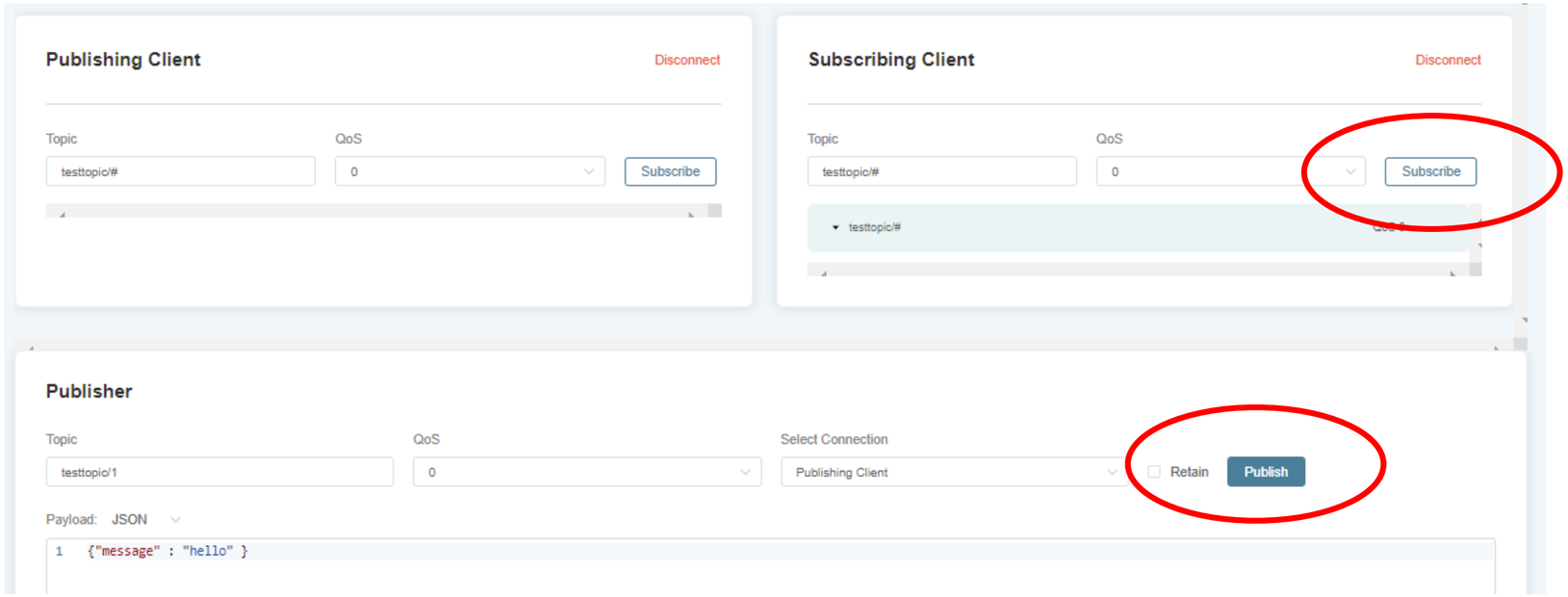
Add a publishing client and subscribing client.



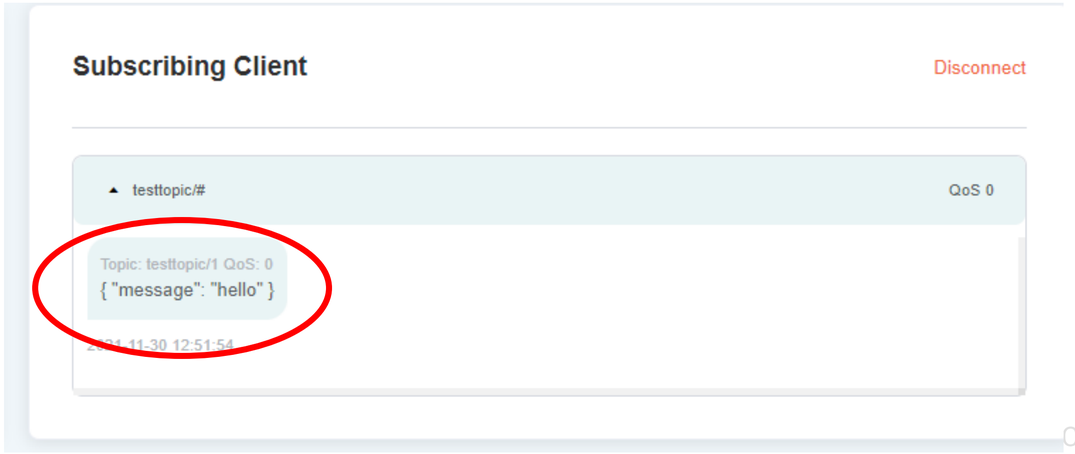
Test: Subscribe and Publish a Message

Try Free →

Click **Subscribe** and then **Publish**

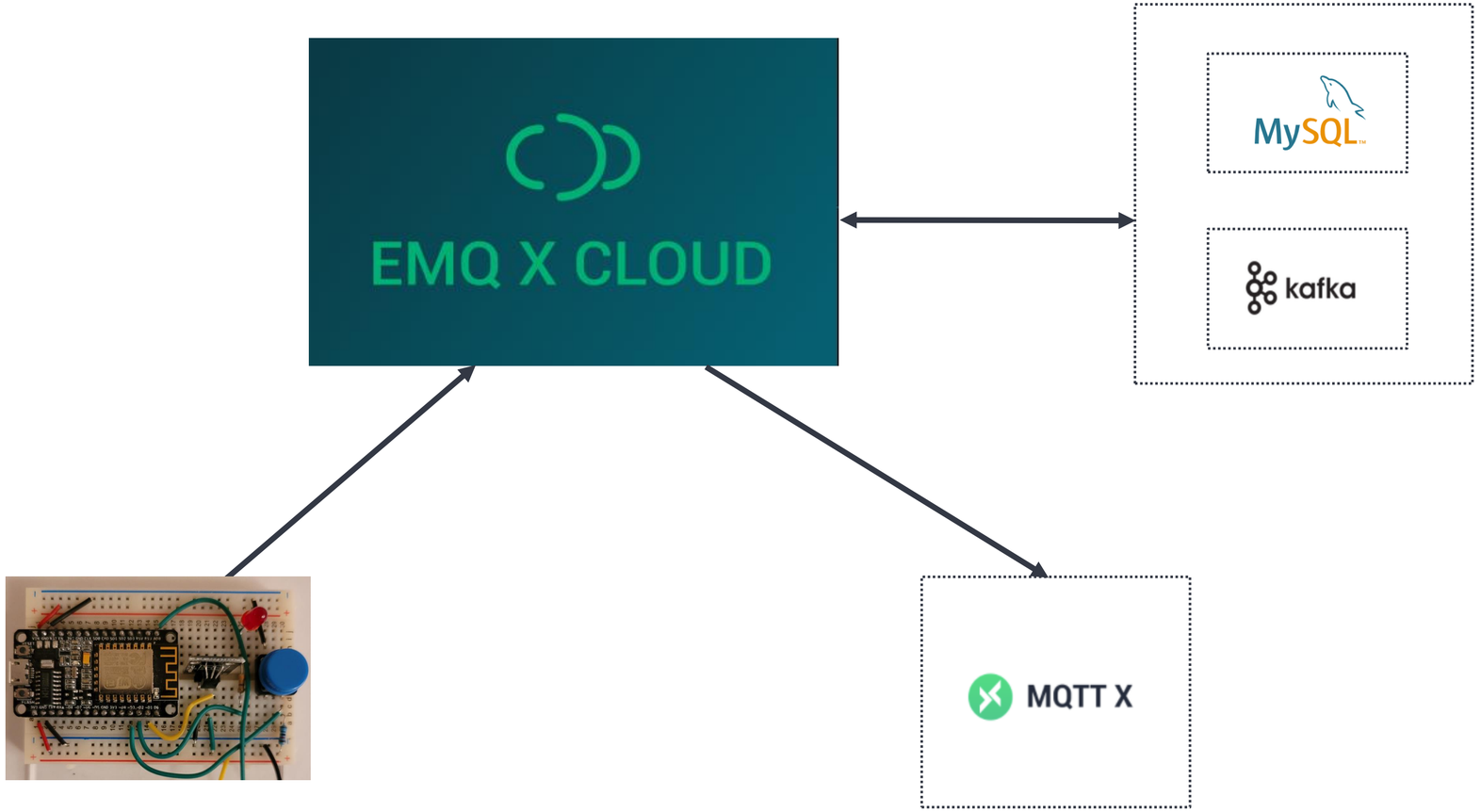


The subscribing client will receive the message.



Demo: EMQ X Cloud and IoT

Setup



Demo Setup

Demo

Need to create:

VPC Peering Connection

Resources

Resource ID	Resource Type
resource:7f7e05ed	Kafka
resource:4047d651	MySQL

Professional WebSocket Clients

EMQ X CLOUD

subscribe

VPC Peering

aws

Virtual, Linux

MySQL

Dockers

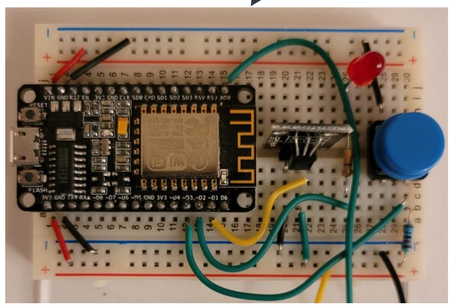
kafka

publish/subscribe

topics:
sensor/1/data
sensor/1/alarm

SSH

IoT Sensor



Laptop PC

MQTT X

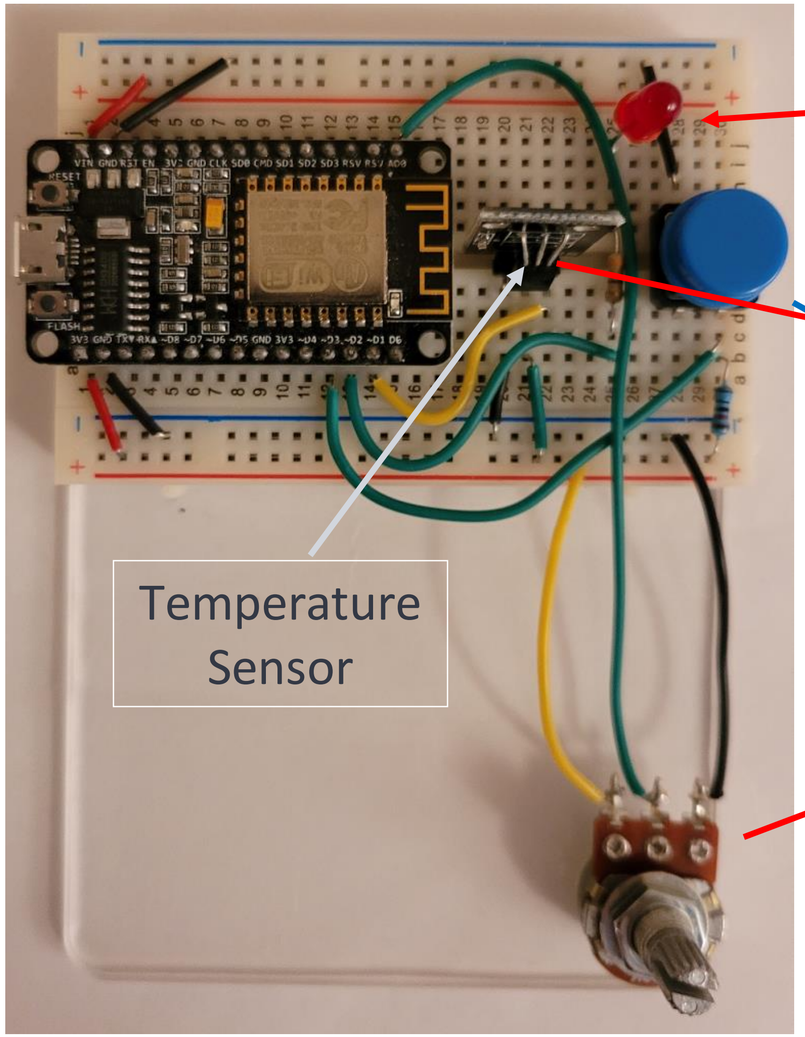
MQTT Clients

subscribe

IoT Sensor

Demo

NodeMCU
128K RAM
4MB Flash



LED

Button

Potentiometer
0 to 1023

Subscribe Topic: sensor/1/alarm

```
{
  "alarm_cmd" : "ON"
}
```

Publish Topic: sensor/1/data

```
{
  "temperature_c": 22,
"   temperature_f": 71.6,
  "alarm_status": "off",
 "button_status": "released",
  "sim_input": 813,
  "sim_temp_c": 94.20821,
   "sim_weight_kg": 39.73607,
  "sim_level_m": 7.947214,
  "sim_speed_kmh": 158.9443,
  "timestamp": "2021-11-24 13:50:41"
}
```

Stream Processing Using Rule Engine

Demo

EMQ X Cloud

Persist data to MySQL
Calculate sim_temp_f

Forward data to Kafka
Calculate sim_temp_f

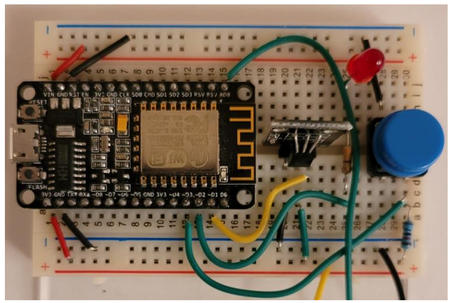
Turn On/Off Alarm
sim_temp_c > 50

Handle button events
pressed / released

Detect sensor
connect / disconnect

Publish:
sensor/1/data

Subscribe:
sensor/1/alarm



sensor/1/alarm
{ "alarm_cmd" : "ON" }

machine/1/cmd
machine/2/cmd
{ "machine_cmd" : "Start" }

sensor/1/status
{"event" : "disconnected"}

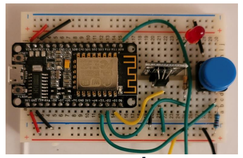
Clean Session vs. Persisted Session

Demo

ClientClean

ClientPersist

EMQ X CLOUD



Clean Session: True

Clean Session: False

Clients connected
Sensor disconnected

Connect,
Subscribe,
Disconnect

Topic: sensor/1/data

Connect, Subscribe, Disconnect

Clients disconnected
Sensor connected

Client ID	Topic
ClientPersist	sensor/1/data

Connect, Publish, Disconnect

Clients connected
Sensor disconnected

Connect

Connect, Subscribe

Messages Received

Demo

ClientClean - Clean Session Disconnect

sensor/1/data — Retain: False — QoS 0

No Data

No messages received

sensor/1/retain — Retain: True — QoS 0

Topic: sensor/1/retain QoS: 0

{ "temperature_c": 21, "temperature_f": 69.8, "alarm_status": "off", "button_status": "released", "sim_input": 5, "sim_temp_c": -24.26686, "sim_weight_kg": 0.244379, "sim_level_m": 0.048876, "sim_speed_kmh": 0.977517, "timestamp": "2021-12-09 12:37:37" }

2021-12-09 12:41:12

ClientPersist - Persisted Session Disconnect

sensor/1/data — Retain: doesn't matter — QoS 0

Topic: sensor/1/data QoS: 0

{ "temperature_c": 21, "temperature_f": 69.8, "alarm_status": "off", "button_status": "released", "sim_input": 5, "sim_temp_c": -24.26686, "sim_weight_kg": 0.244379, "sim_level_m": 0.048876, "sim_speed_kmh": 0.977517, "timestamp": "2021-12-09 12:37:32" }

2021-12-09 12:37:55

Topic: sensor/1/data QoS: 0

{ "temperature_c": 21, "temperature_f": 69.8, "alarm_status": "off", "button_status": "released", "sim_input": 5, "sim_temp_c": -24.26686, "sim_weight_kg": 0.244379, "sim_level_m": 0.048876, "sim_speed_kmh": 0.977517, "timestamp": "2021-12-09 12:37:37" }

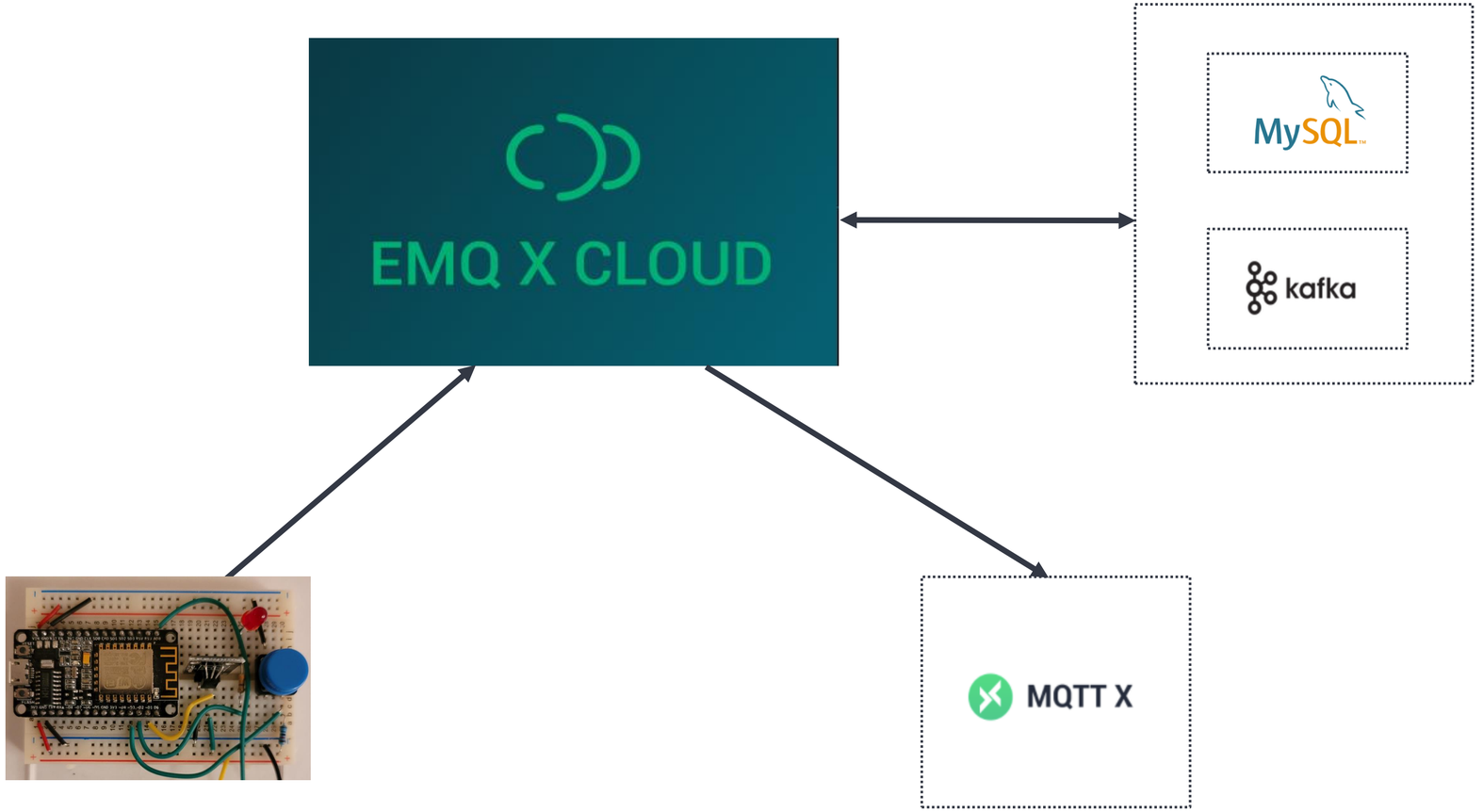
2021-12-09 12:37:55

Receives all messages published while it was disconnected.

Note:
Even QoS 0 messages

If messages published with retain = True:
Client would receive last published message

Demo:
EMQ X Cloud and IoT



EMQ

Summary

EMQ X Cloud is a high performance, high capacity, low latency, fully managed IoT platform that is fast and easy to deploy.

Sign up for your free trial today!

Try Free →

EMQ

Welcome to join EMQ X Community

slack

https://slack-invite.emqx.io/

GitHub

https://github.com/emqx/emqx

Forum

https://github.com/emqx/emqx/discussions

Happy to discuss with YOU

Questions and Answers

Q & A

Sign up for your free trial today!

Try Free →

Details

The Two Parts of Creating Rules

Demo

Use SQL syntax to define the data

Select

Choose or modify the data that you want available to the action

From

One or more topics

Where

The conditions when the action should trigger

Example: Save data to MySQL when temp > 50

```
SELECT
    clientid as client_id,   payload.temp_c as temp_c

FROM
    "sensor/1/data"

WHERE
    temp_c > 50
```

```
Topic: sensor/1/data
{
  "clientid": "sensor1",
  "payload":  { "temp_c":53.5 }
}
```

```
{
 "client_id": "sensor1",
 "temp_c": 53.5
}
```

Create one or more actions

For example:

- Persist to a database
- Forward data to a broker
- Publish to a different topic

```
insert into table_temp
(client_id, temp_c) values (${client_id}, ${temp_c})
```

```
 client_id | temp_c
-----------+--------
 sensor1   |   53.5
```