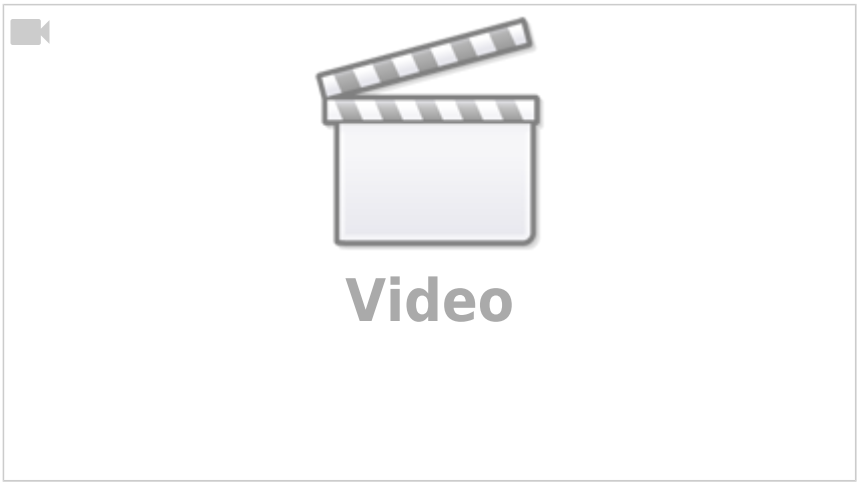
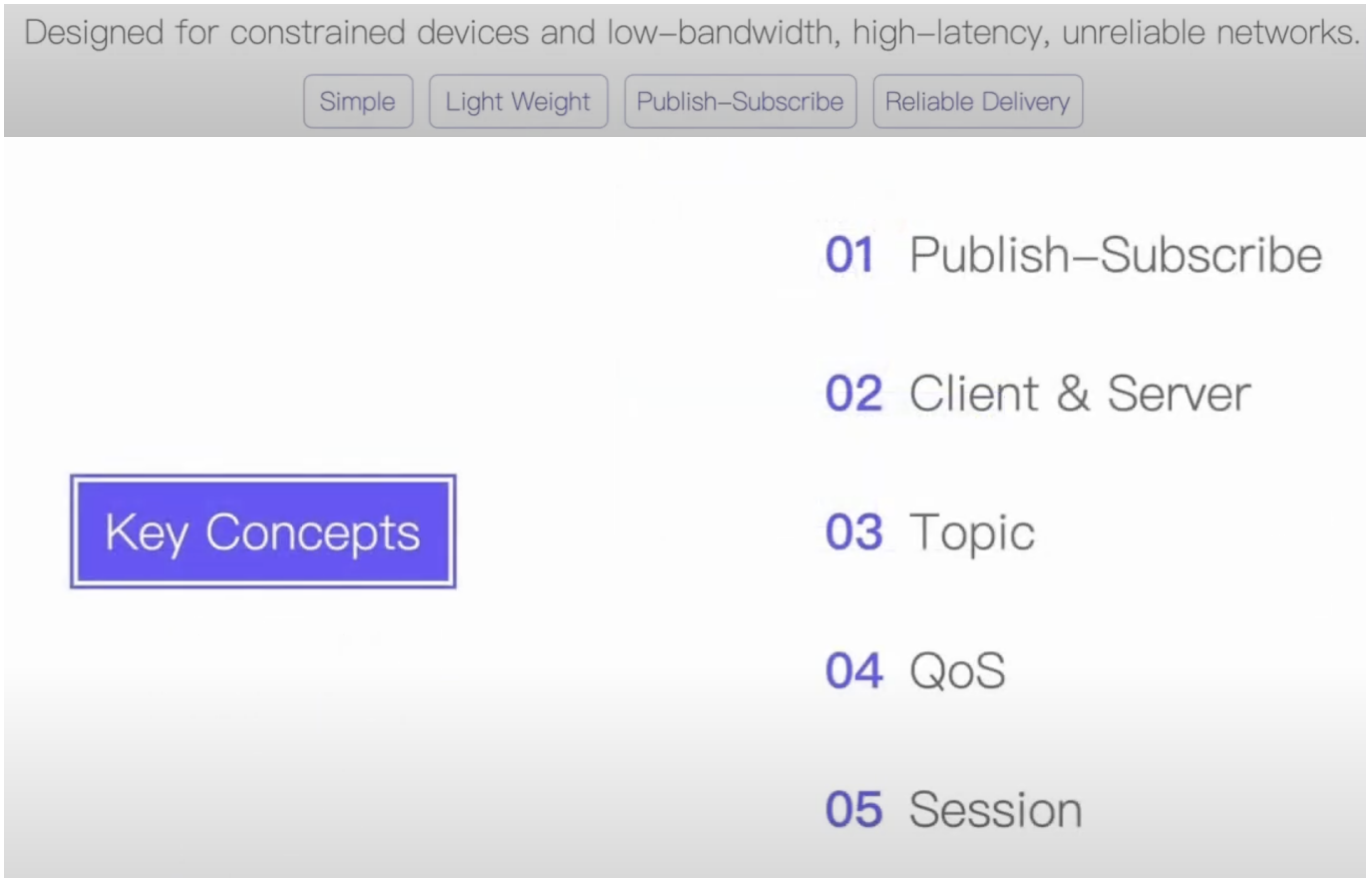
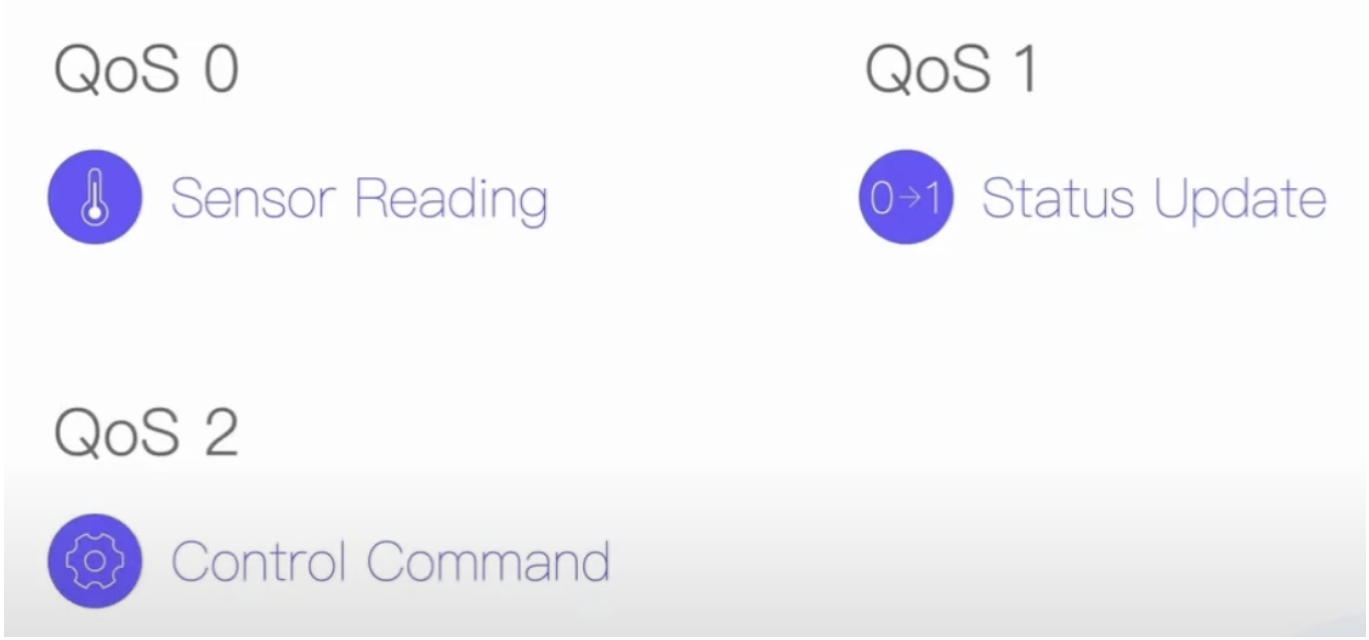
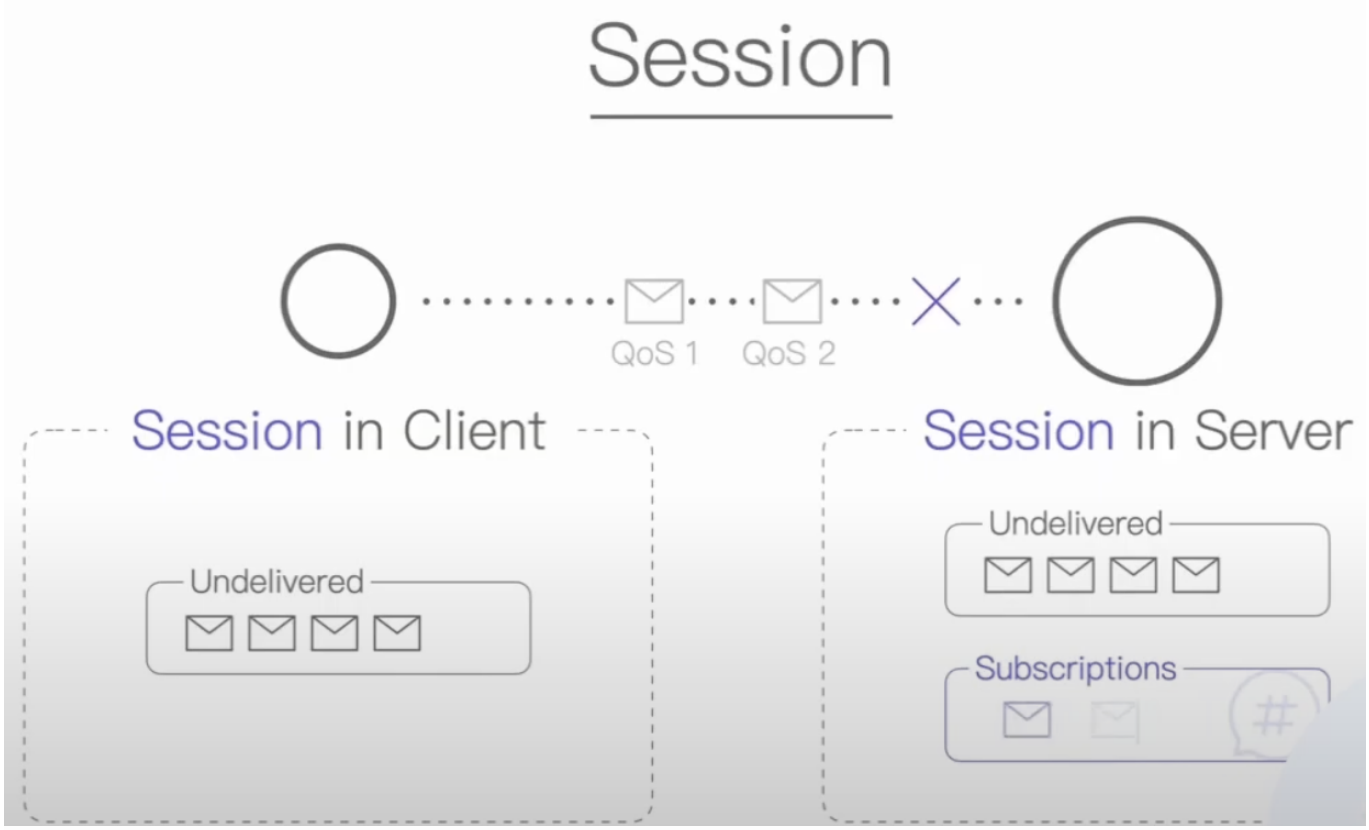
MQTT notes

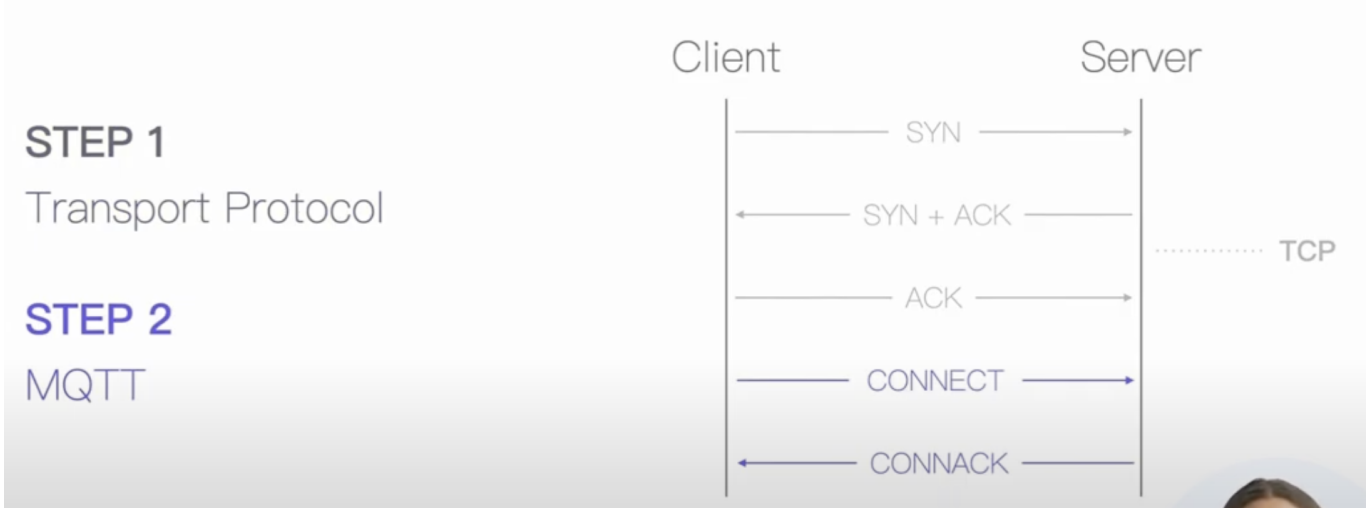
PubSub protocol

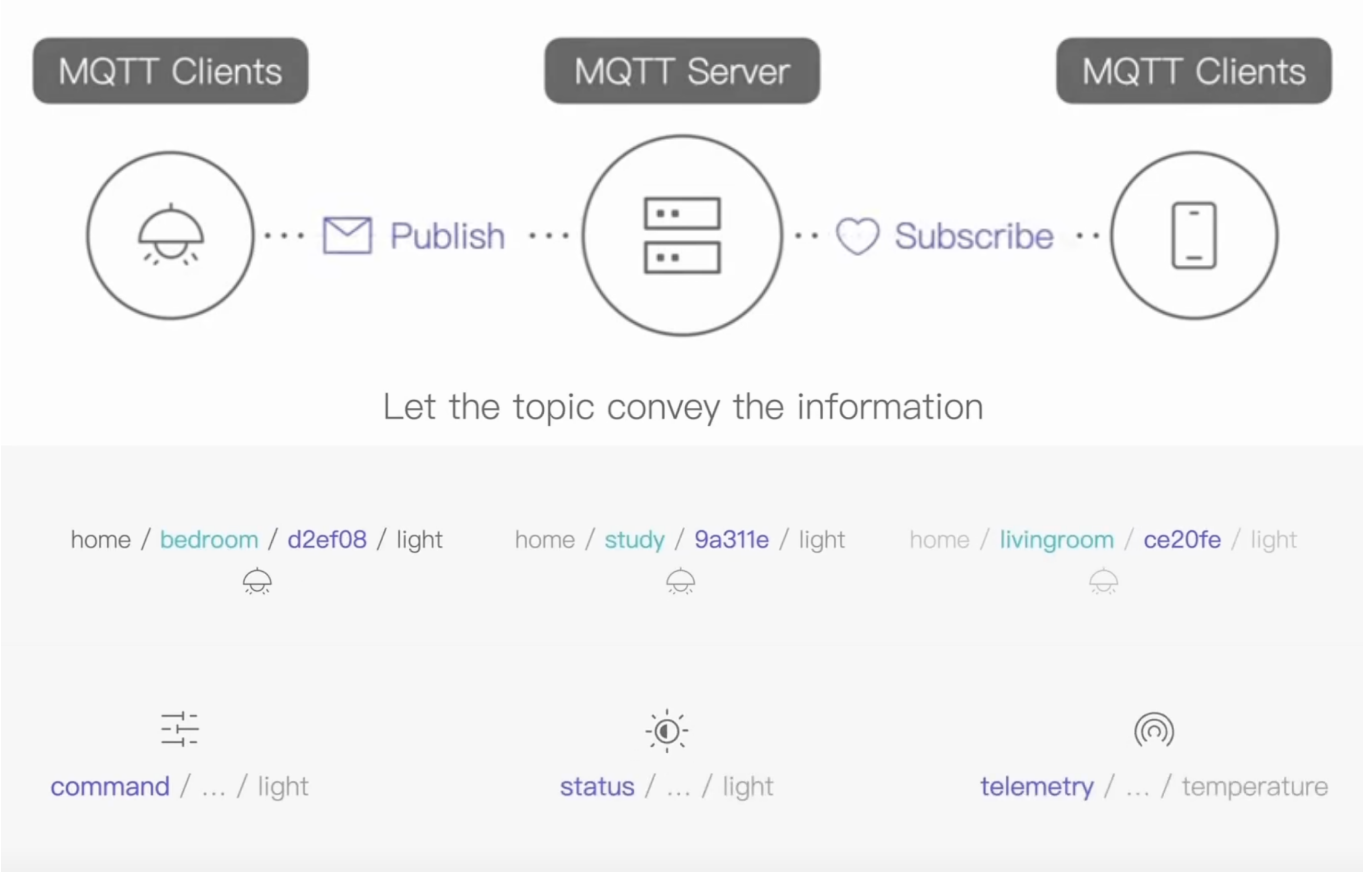





QoS 0
Sensor Reading
QoS 1
0→1
Status Update
QoS 2
Control Command


Session
QoS 1
QoS 2
Session in Client
Undelivered
Session in Server
Undelivered
Subscriptions
#


STEP 1
Transport Protocol
STEP 2
MQTT
Client
Server
SYN
SYN + ACK
ACK
TCP
CONNECT
CONNACK


MQTT Clients
MQTT Server
MQTT Clients
Publish
Subscribe
Let the topic convey the information
home / bedroom / d2ef08 / light
home / study / 9a311e / light
home / livingroom / ce20fe / light
command / ... / light
status / ... / light
telemetry / ... / temperature

EMQX ports

Type	Bind
ssl	0.0.0.0:8883
tcp	0.0.0.0:1883
ws	0.0.0.0:8083
wss	0.0.0.0:8084

Mqtt template to write to Influxdb

```
{
  "devices/+/messages": {
    "measurement": "devices",
    "tags": {
      "client_id": "$client_id"
    },
    "fields": {
      "temperature": ["$payload", "temperature"],
      "humidity": ["$payload", "humidity"]
    },
    "timestamp": "$timestamp"
  }
}
```

InfluxDB

Data organization

The InfluxDB data model organizes time series data into buckets and measurements. A bucket can contain multiple measurements. Measurements contain multiple tags and fields.

- **Bucket**: Named location where time series data is stored. A bucket can contain multiple *measurements*.
  - **Measurement**: Logical grouping for time series data. All *points* in a given measurement should have the same *tags*. A measurement contains multiple *tags* and *fields*.
    - **Tags**: Key-value pairs with values that differ, but do not change often. Tags are meant for storing metadata for each point–for example, something to identify the source of the data like host, location, station, etc.
    - **Fields**: Key-value pairs with values that change over time–for example: temperature, pressure, stock price, etc.
    - **Timestamp**: Timestamp associated with the data. When stored on disk and queried, all data is ordered by time.

> **Fields aren't indexed:** Fields are required in InfluxDB data and are not indexed. Queries that filter field values must scan all field values to match query conditions. As a result, queries on tags > are more performant than queries on fields. **Store commonly queried metadata in tags.**

> Tags containing highly variable information like UUIDs, hashes, and random strings will lead to a large number of unique series in the database, known as **high series cardinality**. High series cardinality is a primary driver of high memory usage for many database workloads. See series cardinality for more information.

Influxdb config

Set up InfluxDB

The initial setup process for an InfluxDB instance creates the following:

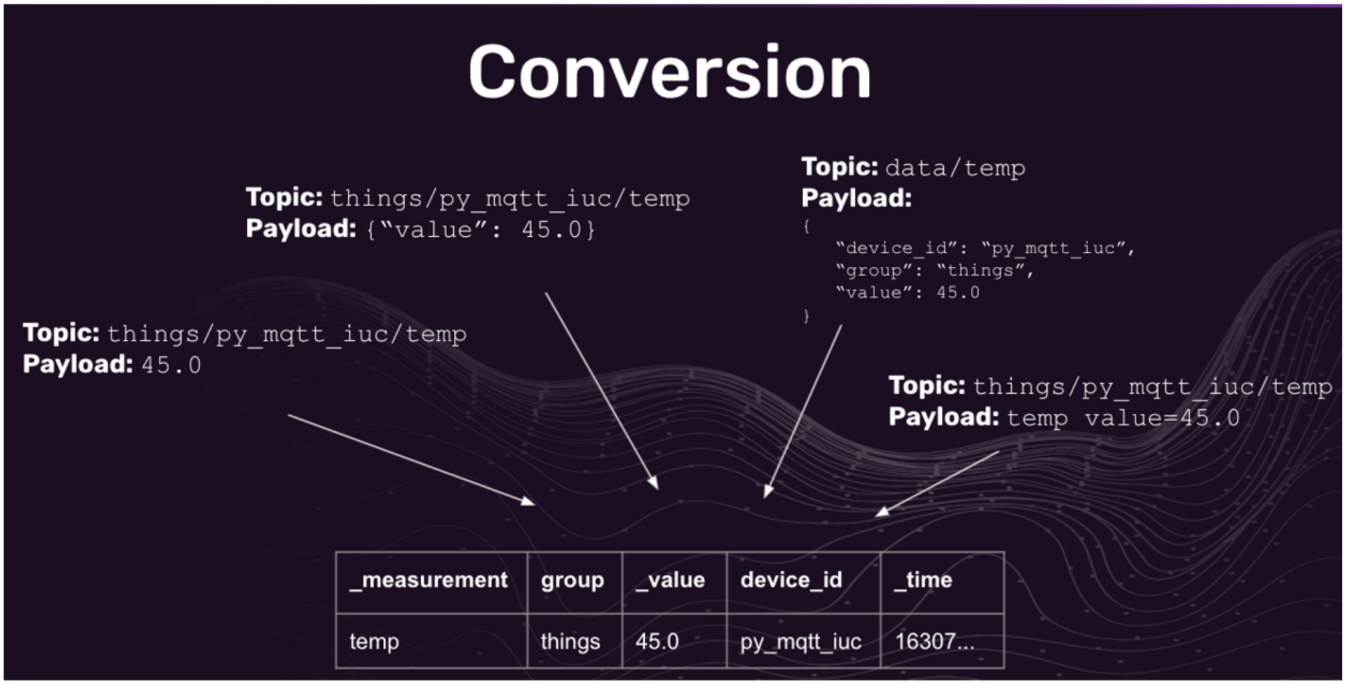
- An organization with the name you provide.
- A primary bucket with the name you provide.
- An admin authorization with the following properties:
  - The username and password that you provide.
  - An API token (*operator token*).
  - Read-write permissions for all resources in the InfluxDB instance.

To run an interactive setup that prompts you for the required information, use the InfluxDB user interface (UI) or the `influx` command line interface (CLI).

To automate the setup–for example, with a script that you write– use the `influx` command line interface (CLI) or the InfluxDB `/api/v2` API.

Telegraf parsing

https://www.influxdata.com/blog/mqtt-topic-payload-parsing-telegraf/
https://www.influxdata.com/blog/how-parse-json-telegraf-influxdb-cloud/



From:
https://wiki.unloquer.org/ -

Permanent link:
https://wiki.unloquer.org/personas/brolin/mqtt?rev=1696952286

Last update: **2023/10/10 15:38**