

# Mosquitto: Getting started

*How to start using Mosquitto*

## Welcome on Stackhero's documentation!

Stackhero provides MQTT Mosquitto instances that are ready for production in just 2 minutes!

Including TLS encryption (aka HTTPS), customizable domain name, unlimited messages, authentication of devices, backups and updates in just a click.

Try our [managed MQTT Mosquitto cloud](#) in just 2 minutes

- [What are topics in MQTT](#)
  - [MQTT topics structures](#)
  - [MQTT topics wildcards](#)
- [How to connect to Mosquitto/MQTT using `mosquitto\\_pub` and `mosquitto\\_sub`](#)
- [Connect to MQTT using websockets](#)
- [MQTT/Mosquitto with Node.js usage examples](#)

Mosquitto is an MQTT server.

The MQTT protocol (Message Queuing Telemetry Transport) is perfect for communication between IoT objects and servers.

On Stackhero, Mosquitto uses TLS encryption and users authentication, providing a very high security level.

You can find some code examples showing how to use Mosquitto on <https://github.com/stackhero-io/mosquittoGettingStarted>

## What are topics in MQTT

---

The MQTT protocol works with "topics".

A topic is like a channel, where devices can push data. Pushing data to a topic is called "publishing".

Devices can listen to a topic too. It is called "subscribing".

Each device that subscribes to a topic will receive data that are published to.

### MQTT topics structures

A topic is composed of UTF-8 characters, is case sensitive and has to be composed of 1 character minimum.

You can define a hierarchy using the slash ("/") character.

Let's say you have IoT devices that collect temperature and humidity and that each device is identified by its MAC address.

You can publish the temperature and humidity to topics like this:

- IoT device with MAC address "00:00:00:00:00:00":
  - Temperature on topic "devices/00:00:00:00:00:00/temperature"
  - Humidity on topic "devices/00:00:00:00:00:00/humidity"
- IoT device with MAC address "11:11:11:11:11:11":
  - Temperature on topic "devices/11:11:11:11:11:11/temperature"
  - Humidity on topic "devices/11:11:11:11:11:11/humidity"

On your app side, you can now subscribe to the topic `devices/00:00:00:00:00:00/temperature` to get temperatures sent by the device with the MAC address "00:00:00:00:00:00".

## MQTT topics wildcards

When you subscribe to a topic, you can use wildcards to subscribe to multiple topics at a time.

For example, if you want to subscribe to every devices temperature topic, you can subscribe to this topic: `devices+/temperature`.

The "+" wildcard means that you subscribe to each topics composed by `devices/<something>/temperature`.

It's a single level wildcard in the way that it will match a topic like `devices/1/temperature` but it will not match `devices/1/2/temperature`.

Another wildcard is the "#" that is a multi level wildcard.

For example, you can subscribe to `devices/#`, meaning that you will subscribe to every topic starting with `devices/`.

It will match `devices/1/temperature`, `devices/2/humidity`, but also `devices/1/2/somethingElse`.

## How to connect to Mosquitto/MQTT using `mosquitto_pub` and `mosquitto_sub`

Mosquitto pub/sub CLIs are useful to make some tests.

If you want to use them, you will have first to disable the "Allow clear connections" option in your Mosquitto configuration (on Stackhero dashboard).

Once done, subscribe to your Mosquitto on topic `#`. It means that you will listen for every topics:

```
mosquitto_sub -L -d "mqtt://admin:<PASSWORD>@<STACKHERO_MOSQUITTO_HOST>:8883/#"
```

Then, in another console, publish the message `It works` to the topic `/sensor/a`:

```
mosquitto_pub -L -d "mqtt://admin:<PASSWORD>@<STACKHERO_MOSQUITTO_HOST>:8883/sensor/a" -m "It works"
```

You should see the message `It works` (and some debug logs because of the `-d` flag) appears in your first console.

# Connect to MQTT using websockets

You can use websockets connections to connect to your server directly from a browser.

You will find more informations on the "Websockets" documentation.

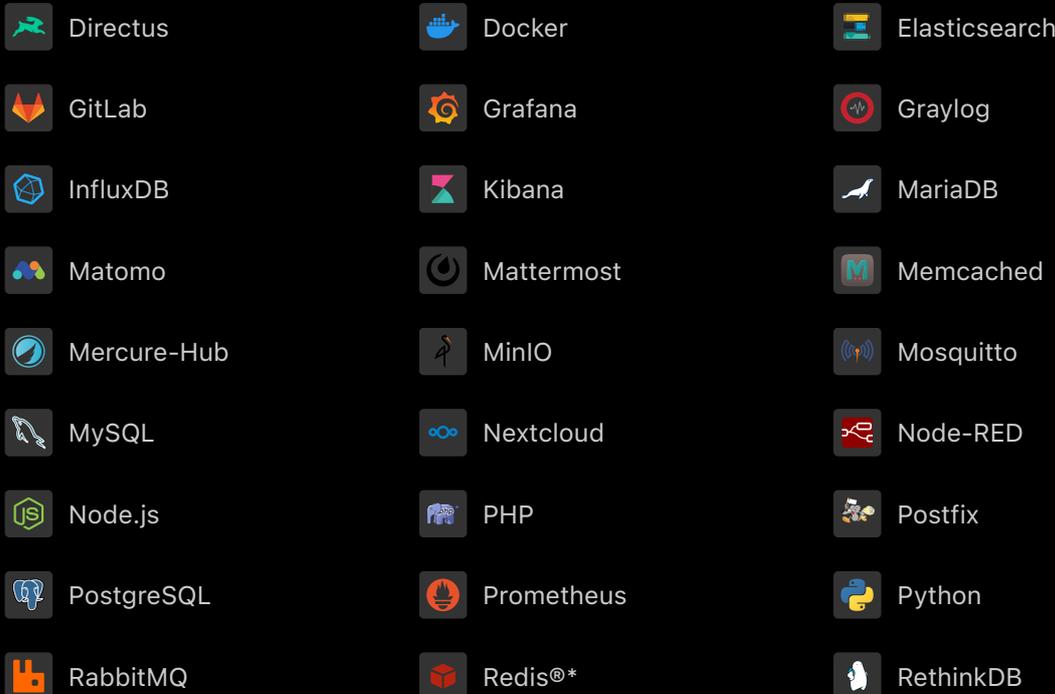
## MQTT/Mosquitto with Node.js usage examples

We have created some code examples in Node.js that are available on this repository: <https://github.com/stackhero-io/mosquittoGettingStarted>

## Other articles about Mosquitto that might interest you

- [Authentication of devices](#)  
How to authenticate thousands of devices using an external API
- [Websockets](#)  
How to connect to MQTT using websockets

### Our Managed Services



[Terms of Service](#)

[Privacy Policy](#)

[Documentations](#)

[Support](#)

[Status](#)

English

Global



Directus, Docker, Elasticsearch, GitLab, Grafana, Graylog, InfluxDB, Kibana, MariaDB, Matomo, Mattermost, Memcached, Mercure-Hub, MinIO, MongoDB, Mosquitto, MySQL, Nextcloud, Node-RED, Node.js, PHP, Postfix, PostgreSQL, Prometheus, Python, RabbitMQ, Redis®, RethinkDB are trademarks and property of their respective owners. All product and service names used on this website are for identification purposes of their open sourced products only and do not imply endorsement. Stackhero is not affiliated to these trademarks or companies.

\*Redis is a registered trademark of Redis Ltd. Any rights therein are reserved to Redis Ltd. Any use by Stackhero is for referential purposes only and does not indicate any sponsorship, endorsement or affiliation between Redis and Stackhero

