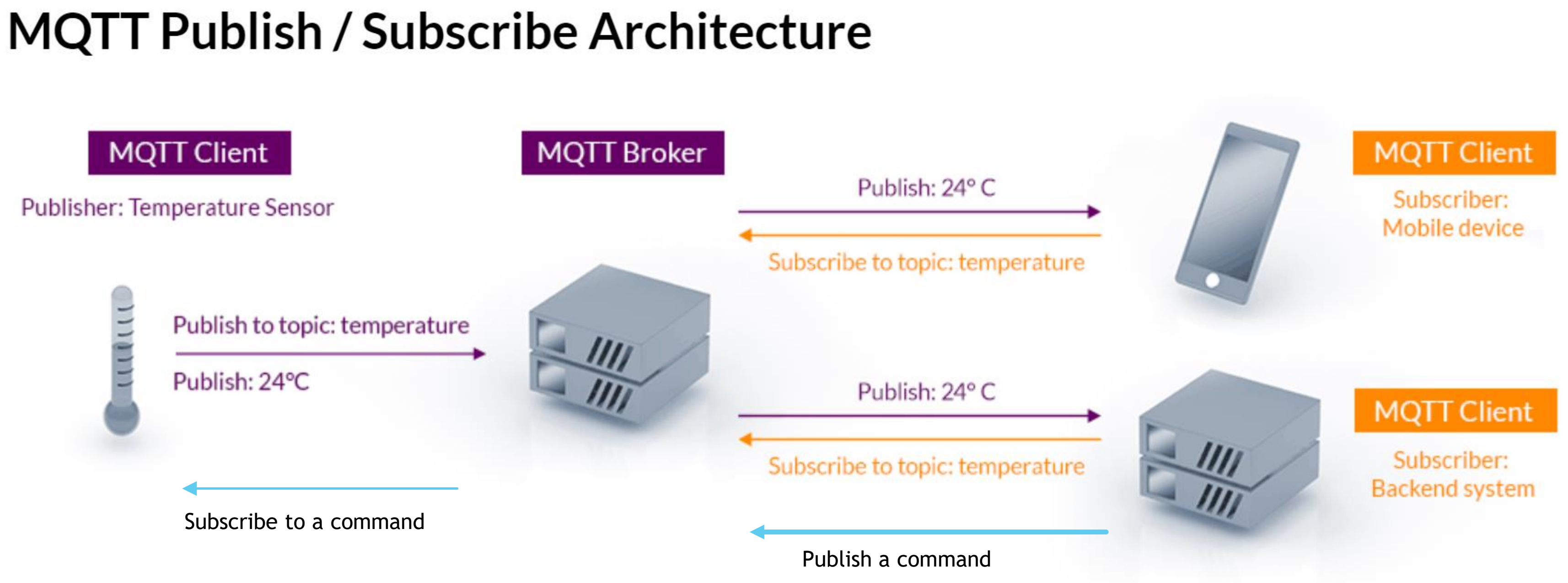
Internet of Things

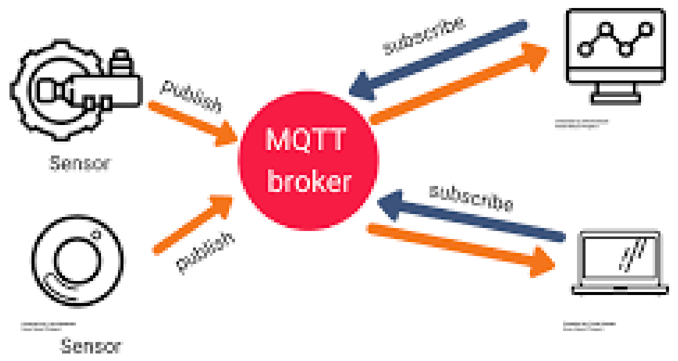
- What is IOT?
  - Deploying a system of smart devices (one to a million or more)
  - Connected over the internet
  - Monitor them remotely
  - Control them remotely
  - Access them securely
- IOT is not just about programming embedded systems
  - It’s a system and an infrastructure
  - That’s where the opportunities are

IOT System


MQTT Publish / Subscribe Architecture
MQTT Client
Publisher: Temperature Sensor
MQTT Broker
Publish to topic: temperature
Publish: 24°C
Subscribe to a command
Publish: 24° C
Subscribe to topic: temperature
Publish: 24° C
Subscribe to topic: temperature
Publish a command
MQTT Client
Subscriber: Mobile device
MQTT Client
Subscriber: Backend system

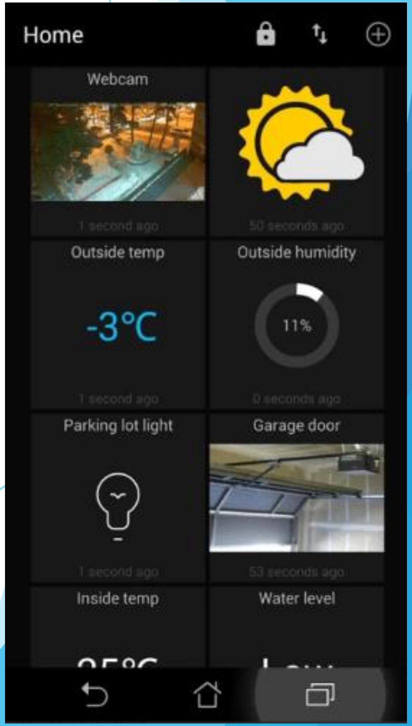
Broker

- An IOT Broker is:
  - Server software, visible on the interwebs
  - A middleman to decouple Clients
  - Receives and Sends Messages
  - Speaks MQTT – MQ Telemetry Transport
    - Publish/Subscribe
    - MQ Telemetry Transport
    - Lightweight protocol over TCP
  - Somewhere in the cloud
  - Lots of vendors



Clients

- IOT calls all the devices and computers ‘Clients’ of the Broker
  - Remote Devices
  - Workstations, Mobile, Back-end
- Clients contact the Broker
  - Clients are usually behind a firewall of some kind
    - Typically, don’t expose a static address and port on the internet (at least not the port)
  - Clients reach out to the broker and establish a connection
  - The Broker never goes out to the Clients
- Clients Publish and Subscribe their data as ‘Topics’ (in MQTT)
- Software
  - Broker Software usually has associated Client SDKs for most languages and runtimes
  - **usually Linux or RTOS, not so much Windows**
    - **You need basic Linux skills**



They Talk MQTT Protocol

- MQ Telemetry Transport – not a message queue ☺
  - Machine-to-machine communication
  - invented at IBM in 1999 by Andy Stanford-Clark and Arlen Nipper
    - Open-source spec released 2010
  - Simple
  - TCP/IP (clients initiate the connections)
  - QOS
  - Lightweight
  - Data agnostic (binary)
- Publish/Subscribe vs Client Server
- Version 3.x, Version 5 coming soon
- Standards based SDK’s are available for mainstream programming languages

MQTT

Some Solutions

- Major Cloud Providers
  - Google, Azure, AWS, IBM
    - All provide an end-to-end IOT infrastructure including broker, node and client support
    - Tie into all their other cloud services such as storage, database, analytics
    - Of course, large distributed data center infrastructure
- Hosted MQTT Brokers
    - These focus more on the core Client ⇔ Broker infrastructure
  - hivemq.com *
  - cloudmqtt.com
  - emqx.io *
- Open Source
  - mosquitto.org*
  - eclispse.org/paho*
- * has open source edition

Developer Opportunities In IOT

- Cloud Experts
  - The most important for a system designer and administrator
- Back End Devs : API’s, Databases, Analytics, Servers
  - Subscribe to data
  - Publish commands
- Front End Devs : Web, Mobile, Dashboards, Analytics, Apps
- Embedded Systems : Linux, RTOS, Devices

References

- mqtt.org/
  - standards
- **www.hivemq.com/mqtt-essentials/**
  - **https://youtu.be/jTeJxQFD8Ak (their mqtt-essentials YouTube series)**
  - **https://www.hivemq.com/mqtt-client-library-encyclopedia/**
- www.steves-internet-guide.com/
- azure.microsoft.com/en-us/overview/iot/
- amazon.com/iot/
- My IOT REPO
  - https://github.com/dmh2000/iot.git
  - Includes quickstarts for HiveMQ and Azure IOT
  - Slides from this presentation
    - But you don’t really need those ☺

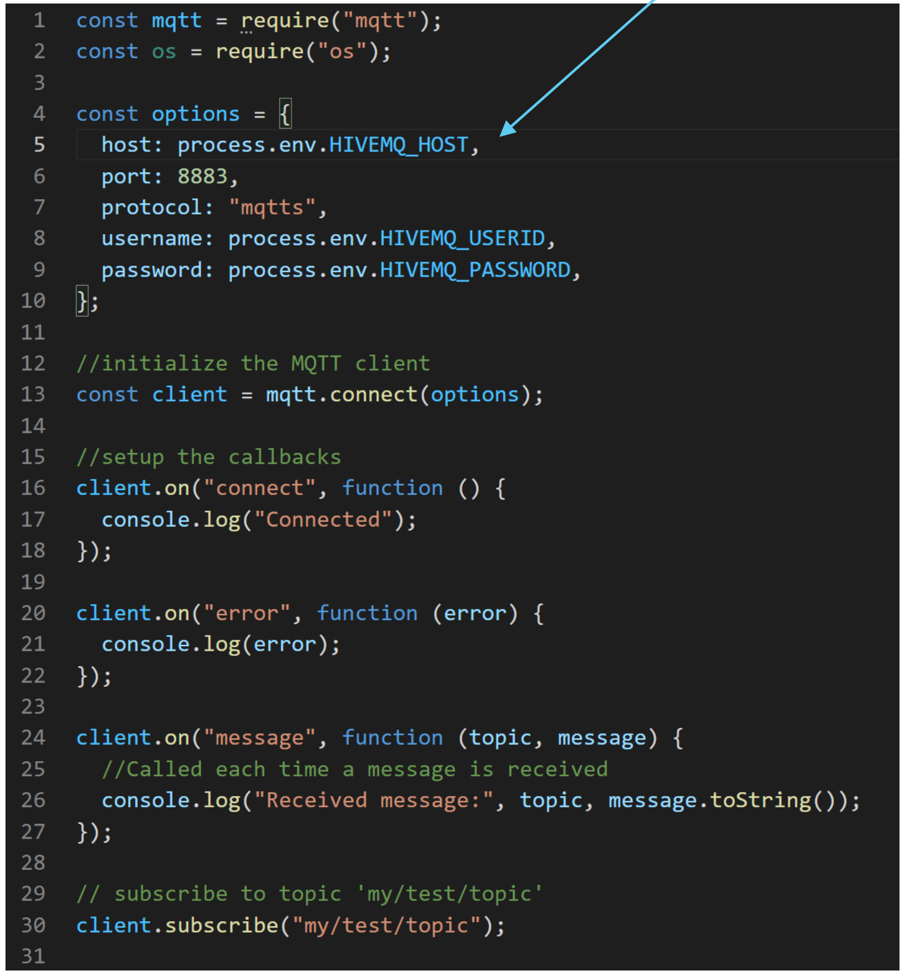
HiveMQ Quickstart at https://github.com/dmh2000/iot.git

https://www.hivemq.com/docs/hivemq/4.7/user-guide/getting-started.html

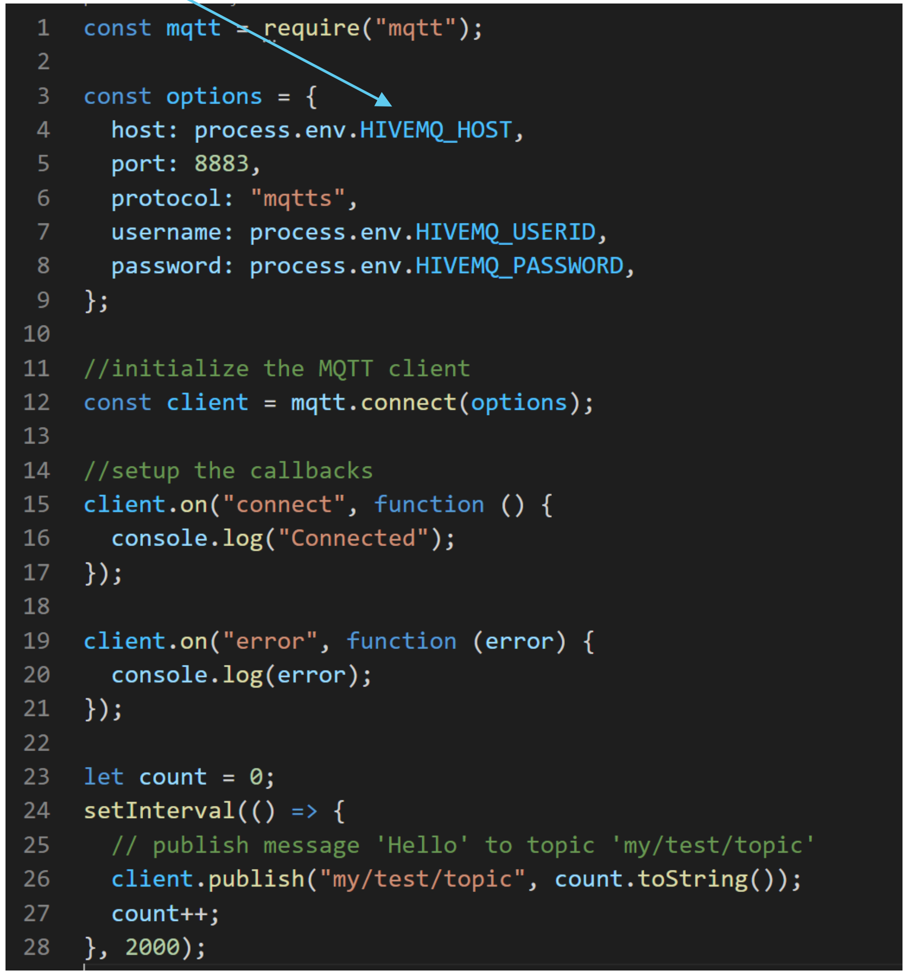
subscriber

You get these when you signup

publisher



```
const mqtt = require("mqtt");
const os = require("os");

const options = {
  host: process.env.HIVEMQ_HOST,
  port: 8883,
  protocol: "mqtts",
  username: process.env.HIVEMQ_USERID,
  password: process.env.HIVEMQ_PASSWORD,
};

//initialize the MQTT client
const client = mqtt.connect(options);

//setup the callbacks
client.on("connect", function () {
  console.log("Connected");
});

client.on("error", function (error) {
  console.log(error);
});

client.on("message", function (topic, message) {
  //Called each time a message is received
  console.log("Received message:", topic, message.toString());
});

// subscribe to topic 'my/test/topic'
client.subscribe("my/test/topic");

```



```
const mqtt = require("mqtt");

const options = {
  host: process.env.HIVEMQ_HOST,
  port: 8883,
  protocol: "mqtts",
  username: process.env.HIVEMQ_USERID,
  password: process.env.HIVEMQ_PASSWORD,
};

//initialize the MQTT client
const client = mqtt.connect(options);

//setup the callbacks
client.on("connect", function () {
  console.log("Connected");
});

client.on("error", function (error) {
  console.log(error);
});

let count = 0;
setInterval(() => {
  // publish message 'Hello' to topic 'my/test/topic'
  client.publish("my/test/topic", count.toString());
  count++;
}, 2000);
```