WEBINAR

Introducing HiveMQ Cloud

HIVEMQ

WELCOME

Dominik Obermaier

@dobermai

linkedin.com/in/dobermai/

- **HiveMQ CTO**
- Strong background in distributed and large scale systems architecture
- OASIS MQTT TC Member
- Author of „The Technical Foundations of IoT“
- Conference Speaker and Author
- Program committee member for German and international IoT conferences

Magi Erber

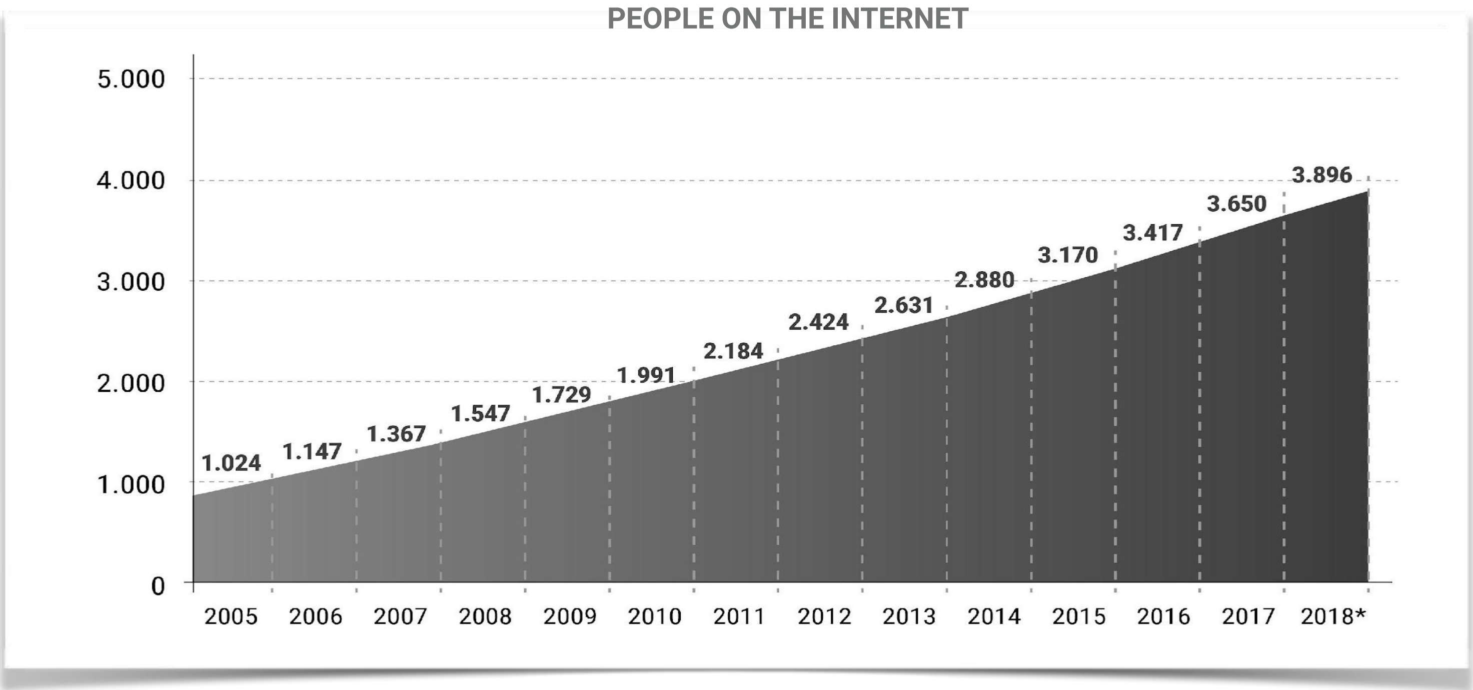
@ErberMagi

linkedin.com/in/margaretha-erber/

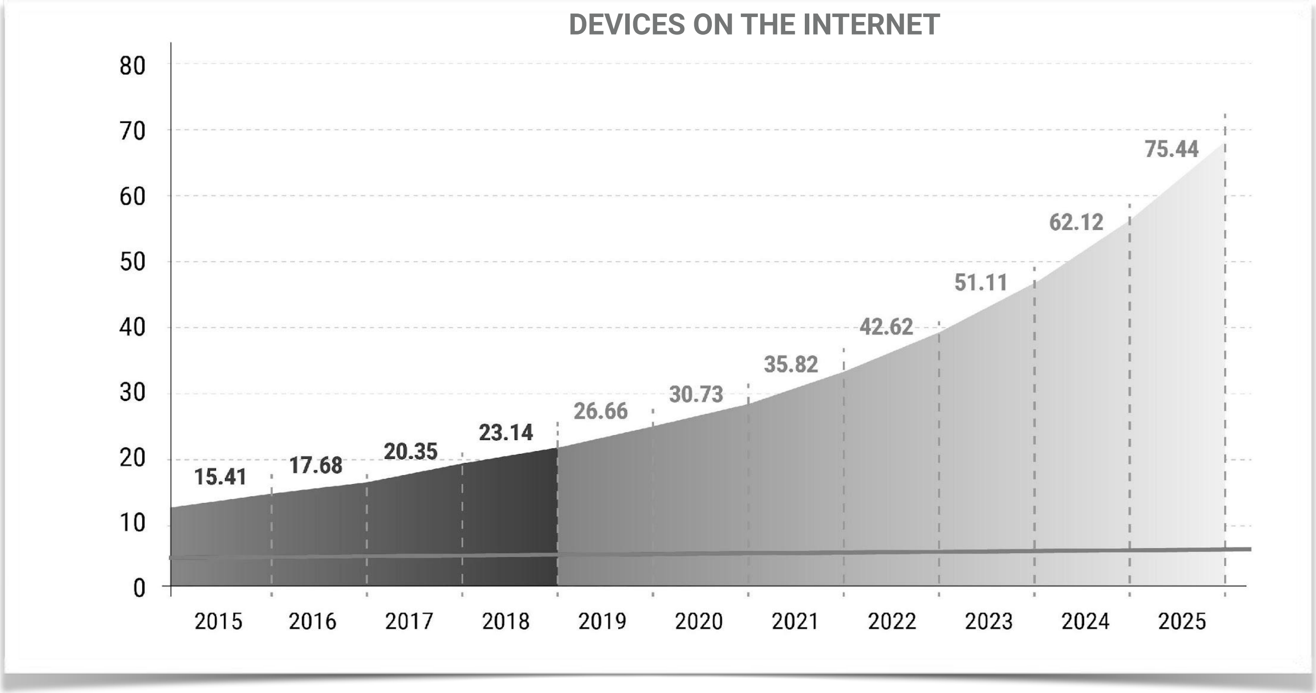
- **Product Manager @HiveMQ**
- Conference Speaker
- Author
- Expert for cloud native technologies and Apache Kafka

As we speak **millions of things** are newly connected to the internet

The Internet of Things is HUGE



The Internet of Things is HUGE



Technical IoT Challenges

Challenge 1 - Scalability

- Enterprise IT infrastructure is **not suitable** for IoT

- **Massive scalability required** for millions of devices

Challenge 2 - Instant Data Delivery required

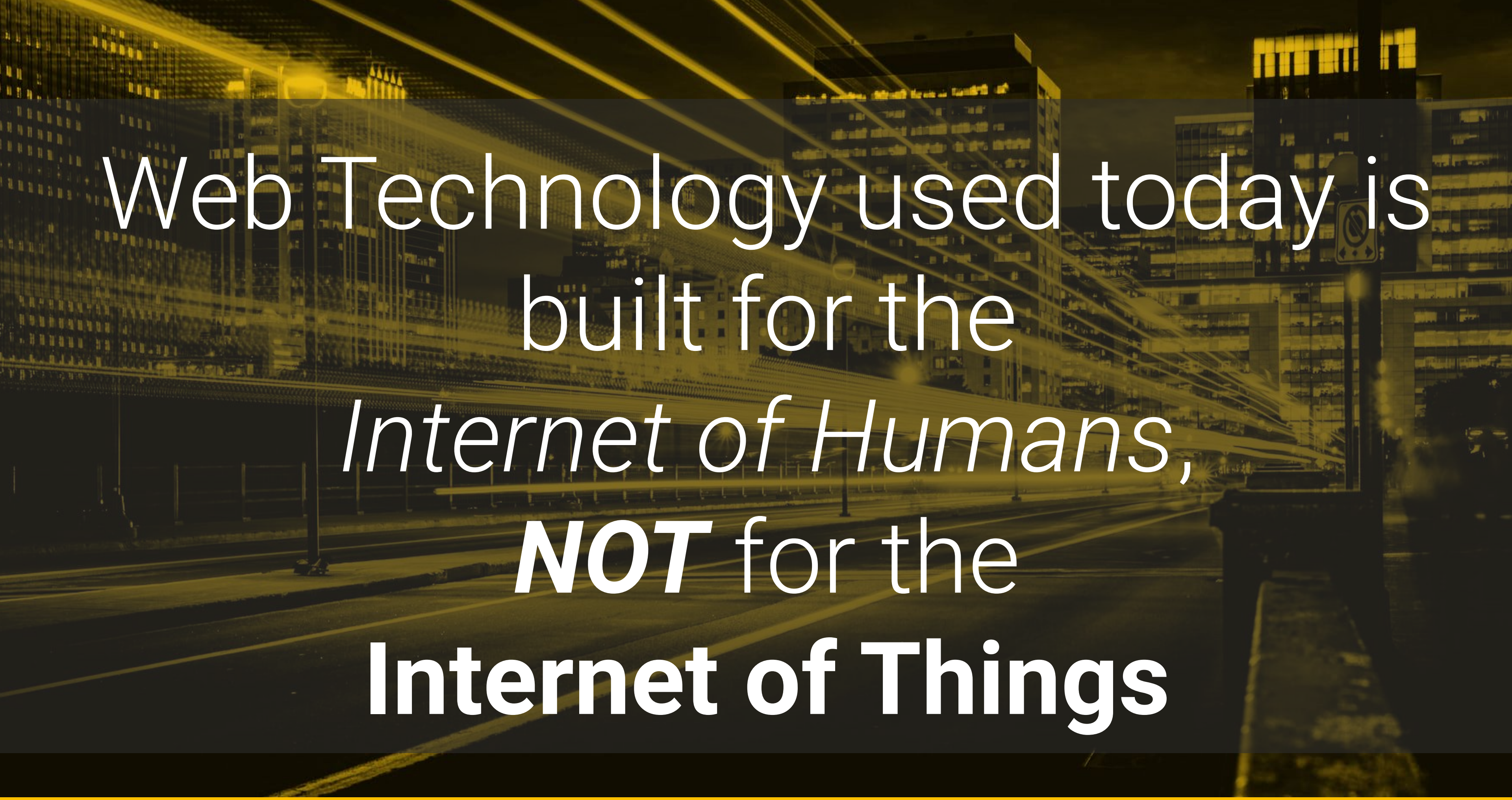
- **End customers are used to *instant* user experiences** like instant messaging with WhatsApp

- **Critical systems need reliable and *instant* data transfer** like manufacturing systems

Challenge 3 - Unreliable Networks

- **Customer experience for IoT apps and devices must be excellent even when internet connectivity is flaky**
  → Especially for moving "devices" like cars

- **Devices and apps must be easy to program and maintain, complexity should be in the cloud not on the device**
  → Cloud is easier to update than physical devices



Web Technology used today is built for the *Internet of Humans*, ***NOT*** for the **Internet of Things**

BUT: There are more challenges

Challenge 1: Flexibility

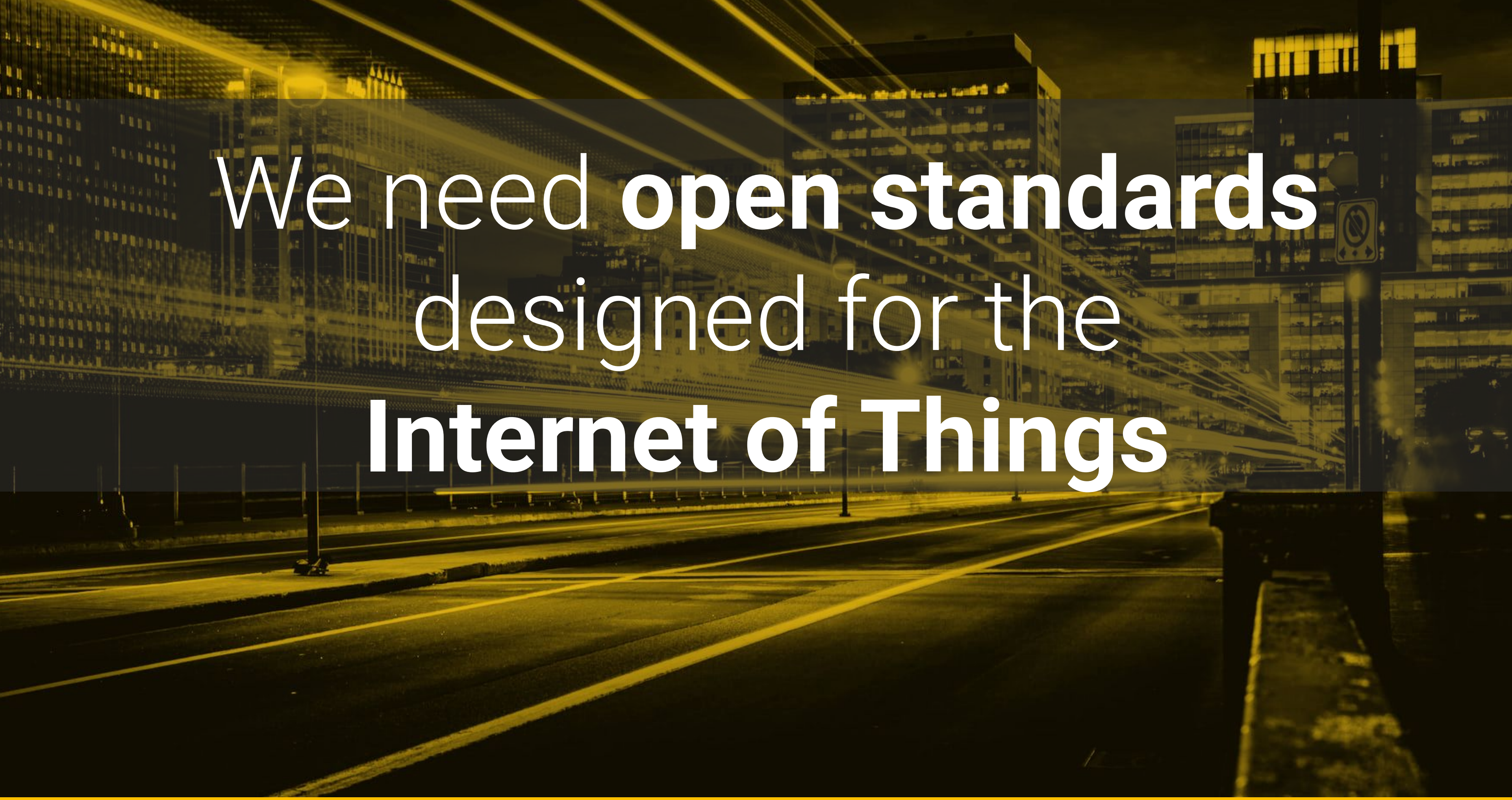
- **IoT Devices and Services may change and evolve in the future**
  → Important to add functionality easily without changing device to cloud communication protocols

- **Time to market is critical and lean/agile principles require flexibility on cloud and device side**
  → Secure bidirectional communication simplifies send/receive behavior changes

Challenge 2: Investment protection

- **IoT data communication enables digitization, new services and new business models**
  → But how to make sure I'm not tied to my cloud vendor?

- **Vendor lock-in for device-to-cloud communication is risky because of investment protection**
  → What if my vendor doesn't support my devices anymore? What If I'm forced to update devices in the field?



We need **open standards** designed for the **Internet of Things**

MQTT

What Is MQTT?

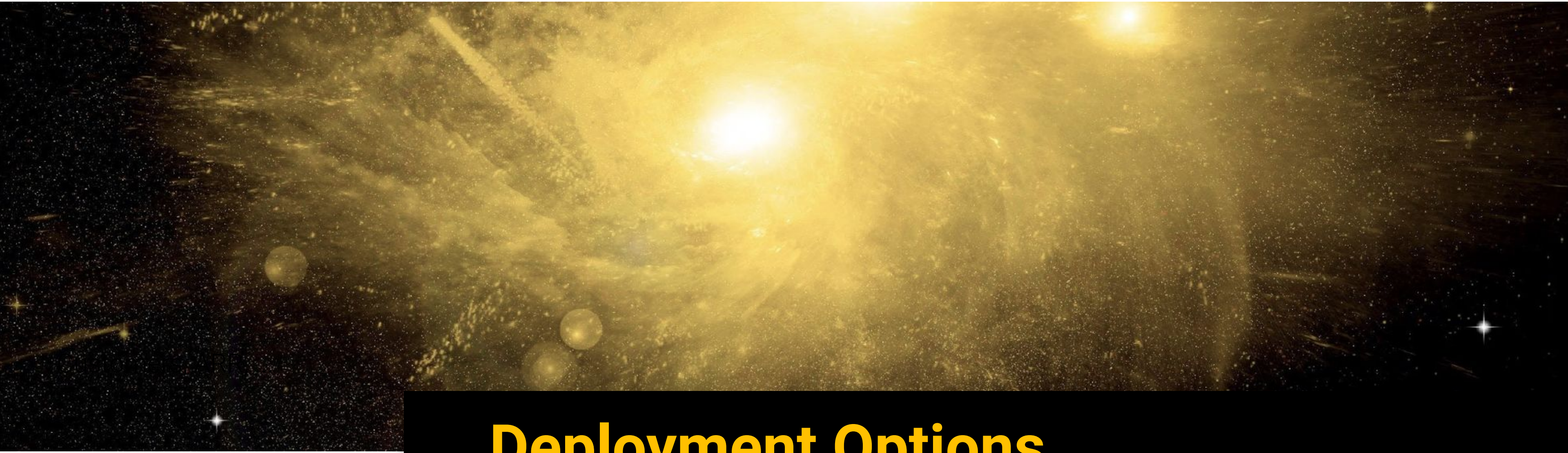
- (I)IoT Messaging Protocol
- Created for extreme scale and instant data exchange
- Publish/Subscribe based architecture
- Easy on the device side, pushes all implementation complexity to the server
- Built for machines (binary, data agnostic)
- Designed for reliable communication over unreliable channels

MQTT Use Cases

- Connected Car
- Industry 4.0 / IIoT
- Logistics / Transportation
- (IoT) Messaging Middleware
- Telecommunications

MQTT Use Cases

- Push Communication
- Reliable Communication over unreliable networks
- Constrained Devices
- Low Bandwidth and High Latency
- Industrial Message Bus



Deployment Options

Options for MQTT Deployments

1. Proprietary Cloud Vendor

2. Self-hosted solution

Self hosting an MQTT broker for IoT connectivity

Proof of concepts are easy, but production is hard

Many start with HiveMQ CE or mosquitto

➔ Difficult to scale for big use cases

Overwhelming complexity of managing the system

➔ Lot of technologies

➔ New challenges like millions of open tcp connections

Needs people and processes

➔ For running the system 24/7

Scalability and availability

➔ Downtime affects ALL customers in MQTT systems due to centralized architecture

Connecting external services

➔ Needs custom programming

Risks when using cloud platform for IoT connectivity

Cloud Services don't offer the MQTT ISO Standard

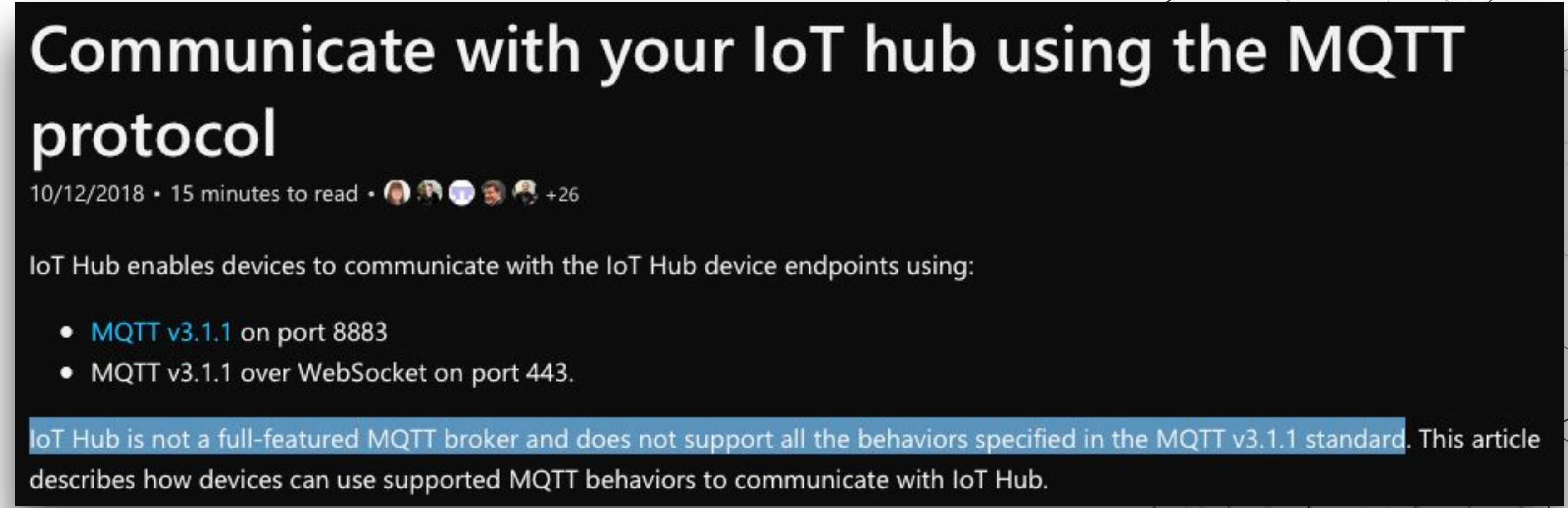
Only **subsets of the ISO standard protocol are supported** and MQTT can only be used in a very opinionated and **proprietary way**

Azure IoT Hub does not support the MQTT 5 specification and doesn't support all MQTT 3 features

AWS IoT does not support the MQTT 5 specification and doesn't support all MQTT 3 features

→ **Cloud vendor lock-in**

Azure IoT Hub - MQTT Restrictions and proprietary changes



Communicate with your IoT hub using the MQTT protocol

10/12/2018 • 15 minutes to read • +26

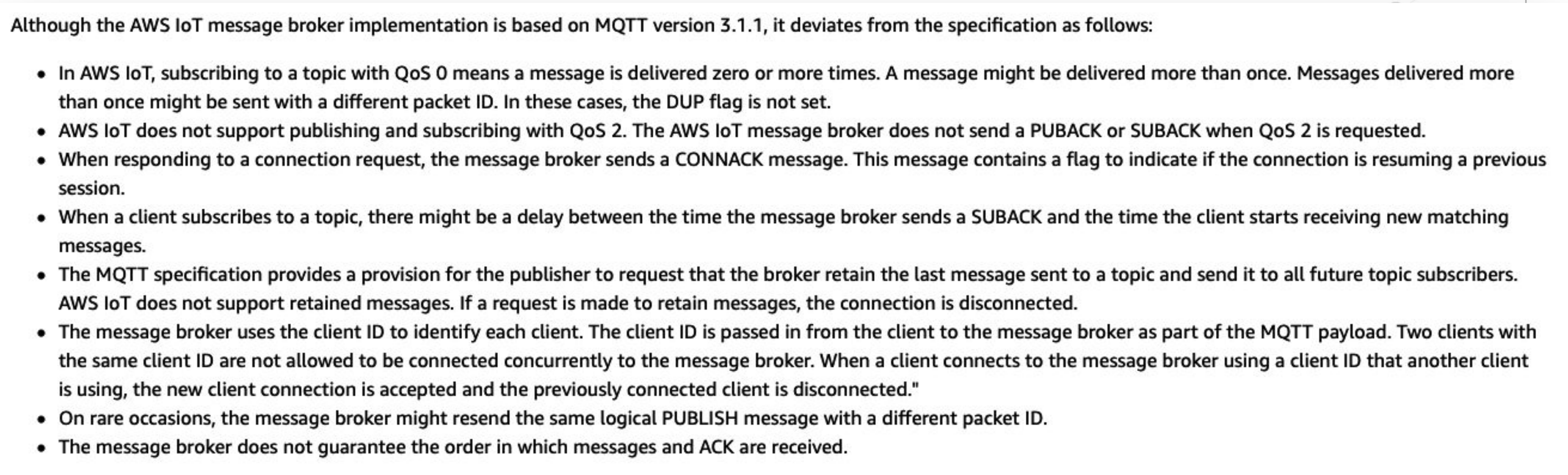
IoT Hub enables devices to communicate with the IoT Hub device endpoints using:

- MQTT v3.1.1 on port 8883
- MQTT v3.1.1 over WebSocket on port 443.

IoT Hub is not a full-featured MQTT broker and does not support all the behaviors specified in the MQTT v3.1.1 standard. This article describes how devices can use supported MQTT behaviors to communicate with IoT Hub.

Source: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-mqtt-support

AWS IoT - MQTT Restrictions and proprietary changes



Although the AWS IoT message broker implementation is based on MQTT version 3.1.1, it deviates from the specification as follows:

- In AWS IoT, subscribing to a topic with QoS 0 means a message is delivered zero or more times. A message might be delivered more than once. Messages delivered more than once might be sent with a different packet ID. In these cases, the DUP flag is not set.
- AWS IoT does not support publishing and subscribing with QoS 2. The AWS IoT message broker does not send a PUBACK or SUBACK when QoS 2 is requested.
- When responding to a connection request, the message broker sends a CONNACK message. This message contains a flag to indicate if the connection is resuming a previous session.
- When a client subscribes to a topic, there might be a delay between the time the message broker sends a SUBACK and the time the client starts receiving new matching messages.
- The MQTT specification provides a provision for the publisher to request that the broker retain the last message sent to a topic and send it to all future topic subscribers. AWS IoT does not support retained messages. If a request is made to retain messages, the connection is disconnected.
- The message broker uses the client ID to identify each client. The client ID is passed in from the client to the message broker as part of the MQTT payload. Two clients with the same client ID are not allowed to be connected concurrently to the message broker. When a client connects to the message broker using a client ID that another client is using, the new client connection is accepted and the previously connected client is disconnected."
- On rare occasions, the message broker might resend the same logical PUBLISH message with a different packet ID.
- The message broker does not guarantee the order in which messages and ACK are received.

Source: https://docs.aws.amazon.com/iot/latest/developerguide/mqtt.html

Risks when using cloud platforms for IoT connectivity

Unpredictable Pricing

- **May change at any time** for most cloud vendors (including AWS and Azure)
- **Usually consists of many variables** (connections per hours / messages with arbitrary counting decisions / traffic / …)
- Distinction between incoming and outgoing traffic and different MQTT packet types
- **Service Integrations with other services usually also adds more variables**
- Pricing for PoCs extremely low, but **actual production workload is expensive**

→ Hard to calculate costs upfront

MQTT 5 Support

MQTT 5 Support?

- **AWS IoT Core**: ***No*** MQTT 5 Support
- **Azure IoT Hub**: ***No*** MQTT 5 Support
- **Google Cloud IoT Core**: ***No*** MQTT 5 support

We live in a multi-cloud world,
so it is important
that IoT applications
can exist in a
multi-cloud world.

How to Fix?

- Use a vendor that supports 100% of the MQTT ISO standard in a cloud agnostic way
- Usually the connectivity layer (MQTT Broker) is HOSTED at a cloud provider but not proprietary cloud-specific services are used
- Use a MQTT broker that is built for high-availability and has industry references
- Don't use open source software like mosquitto for mission-critical deployments (fine for testing, though)
- Use a vendor that can integrate with the value-added services of your cloud providers (like Device Management, ...)
- If it's mission critical, make sure the vendor has good SLAs and is responsive

Operating an IoT infrastructure

Vendor lock-in when using big cloud providers

Hosting an MQTT broker needs expertise and resources

Options for MQTT Deployments

1. Proprietary Cloud Vendor

2. Self-hosted solution

Options for MQTT Deployments

3. Cloud Native IoT Messaging Service

Cloud Native IoT Messaging Service

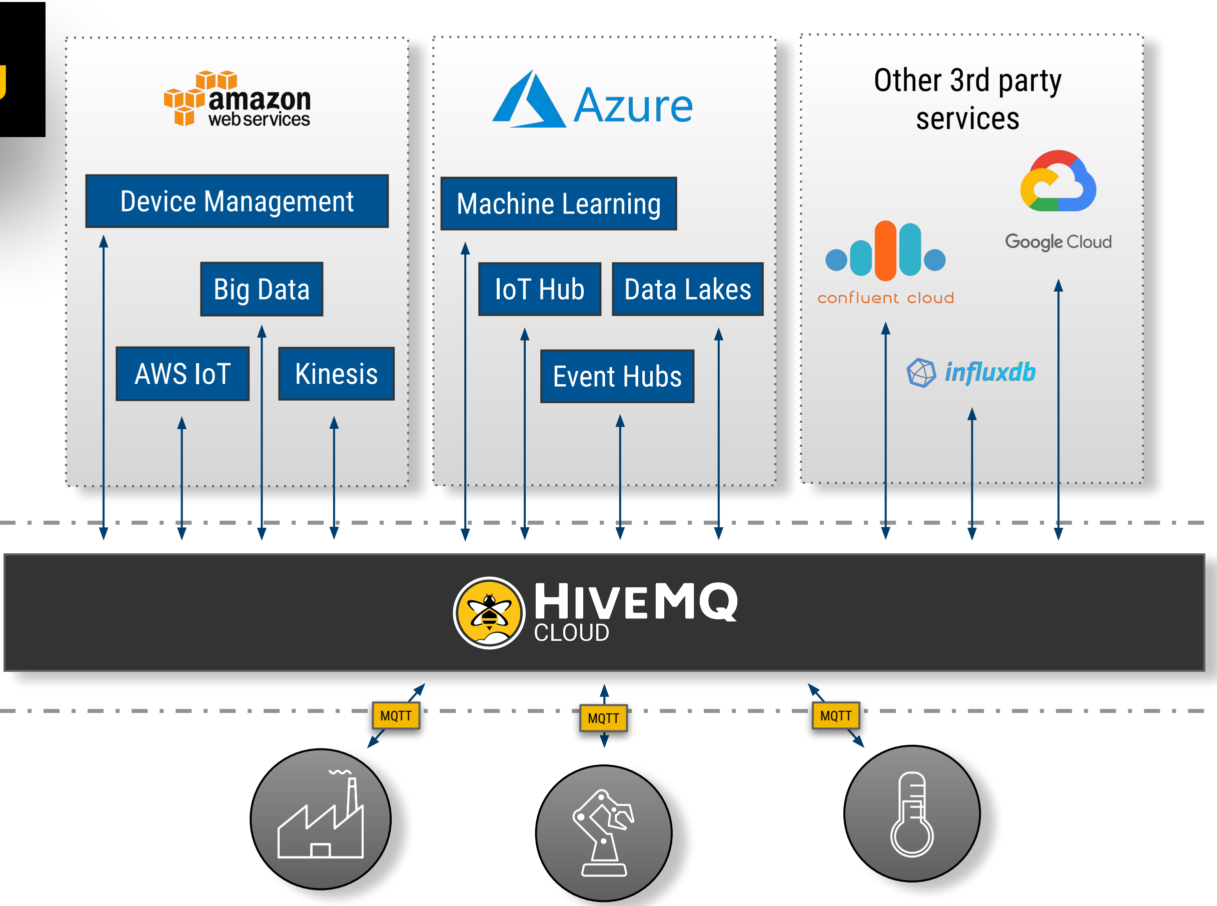
1. Based on open standards
2. Transparent elastic scalability
3. Transparent fault tolerance & high availability
4. Cloud Agnostic
5. Data integration with other services

Cloud Native IoT Messaging

Cloud Vendors & Third Party Services

Connectivity Layer running on any or multiple clouds

Devices, Factories and "Things"



amazon web services

Device Management

Big Data

AWS IoT

Kinesis

Azure

Machine Learning

IoT Hub

Data Lakes

Event Hubs

Other 3rd party services

confluent cloud

Google Cloud

influxdb

HIVEMQ CLOUD

MQTT

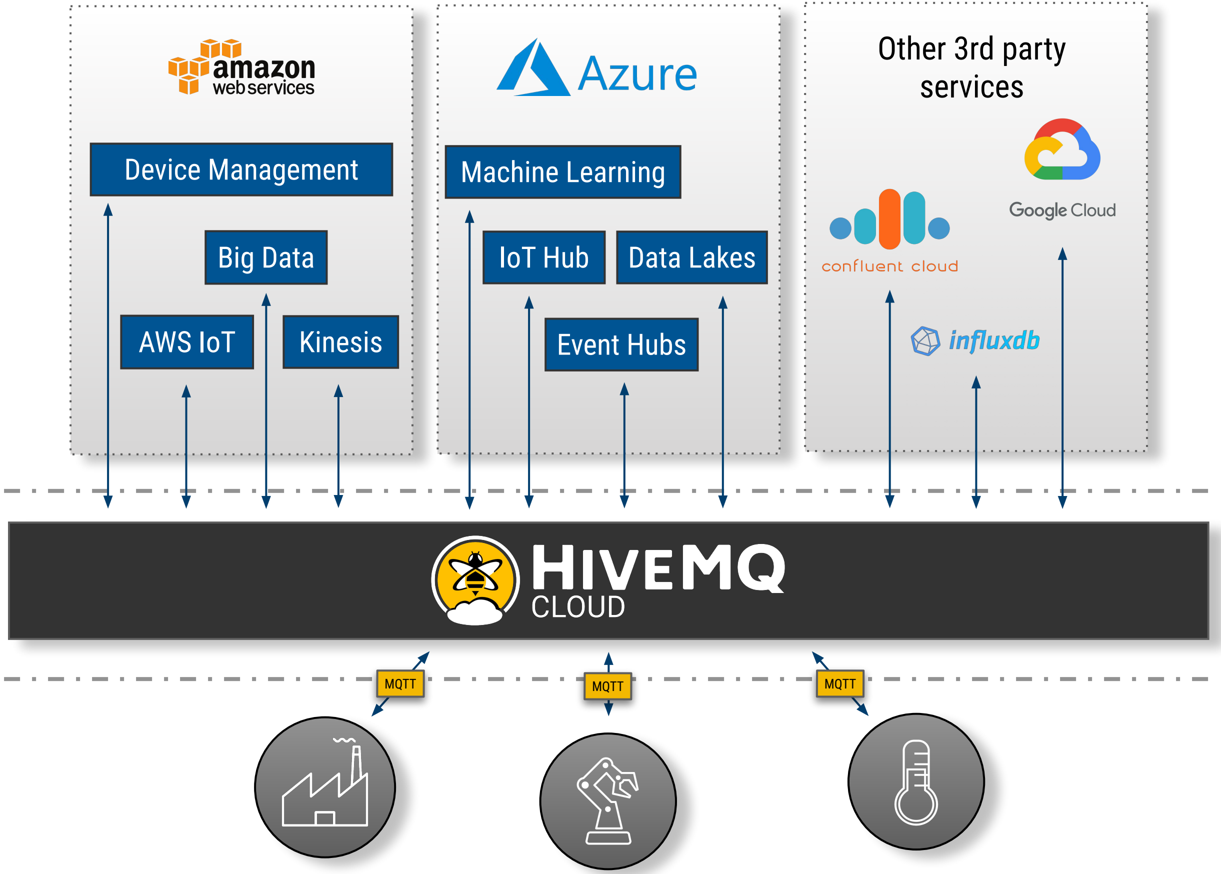
MQTT

MQTT

Cloud Vendors &
Third Party
Services

Connectivity
Layer
running on any or
multiple clouds

Devices, Factories
and "Things"



amazon web services

Device Management

Big Data

AWS IoT

Kinesis

Azure

Machine Learning

IoT Hub

Data Lakes

Event Hubs

Other 3rd party services

Google Cloud

confluent cloud

influxdb

HIVEMQ CLOUD

MQTT

MQTT

MQTT

*“As enterprises furiously adopt cloud as part of their overall architecture, **one of the choices they are forced to make will be whether to rely on services native to a single cloud vendor.***

*While these can seem more convenient in the short term, they have the **unfortunate side effect of locking buyers to a single platform.** This is why **customers are increasingly placing a high value on neutral, standards based software layers** that allow for seamless operation across different cloud providers.*

This is, in essence, what HiveMQ Cloud is built for.”

- Stephen O’Grady

Principal Analyst at RedMonk

HIVEMQ CLOUD

HiveMQ Cloud

HIVEMQ CLOUD

Completely managed MQTT broker service in the cloud with best in class third party integrations

Completely managed

Observability for IoT Devices

Built for Production

Automatic Scalability & Reliability

Enterprise-grade Security

Integration with Best of breed third party services

Cloud agnostic

Predictable Pricing

100% MQTT Specification compliant

Built for Production

Dedicated, highly available infrastructure

Deployed across 3 availability zones

Redundant load balancers

Connect hundreds of thousands of devices

High message throughput

24/7 professional HiveMQ support

Automatic Scalability & Reliability

Automatically scales to meet the demands of your devices

Unique clustering technology

Automatically managed cluster nodes

Enterprise-Grade Security

TLS secured communication

MQTT client authentication

HiveMQ Control Center user authentication

100% MQTT Specification Compliant

100% MQTT

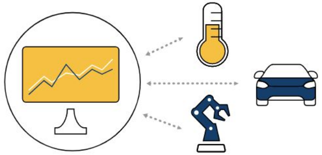
100% MQTT compliant to all MQTT protocol versions:

- MQTT 5
- MQTT 3.1.1
- MQTT 3.1

Support of all MQTT Features, like

- All Quality of Service (QoS)
- Shared Subscriptions
- User Properties
- Retained Messages
- Persistent Sessions
- ...

Observability for IoT Devices



Monitor each HiveMQ Cloud cluster with the HiveMQ Control Center in real time

Client overview and detail cluster view for each client session

Advanced analytics of irregular behaviour

Monitoring the health of the cluster

Predictable Pricing

BASIC HOURLY PRICE:

$ 7.50 per hour

ADDITIONAL DATA:

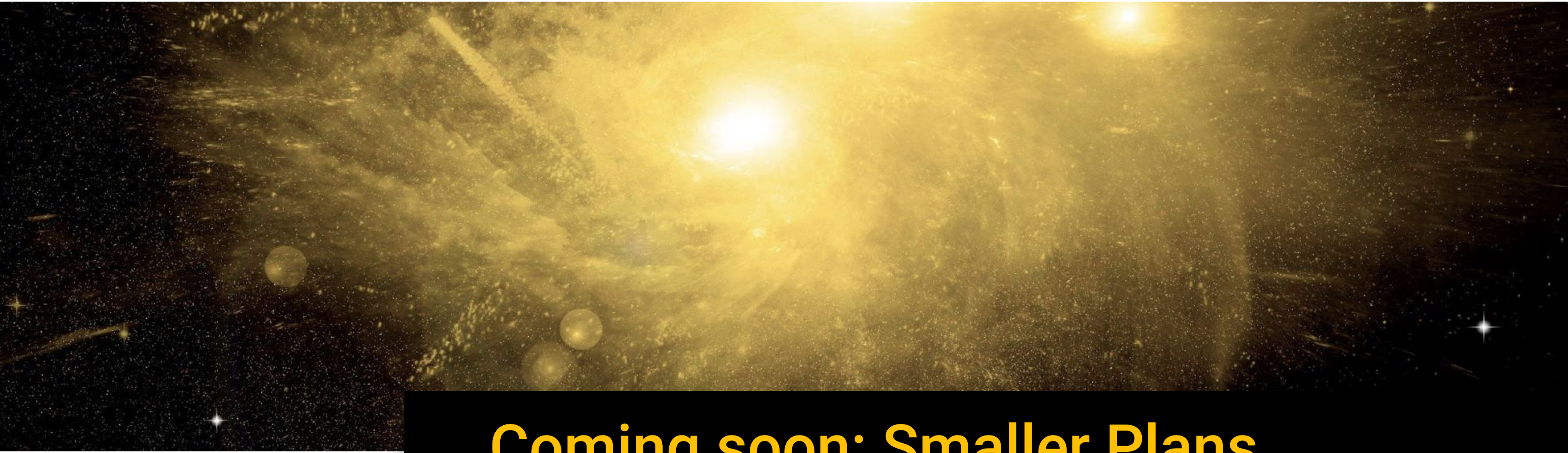
$ 0.15 per GB

HIVEMQ Cloud PRICING INCLUDES:

- Dedicated infrastructure
- 3 different data centers
- 100% MQTT specification compliant
- 1 TB data traffic free
- 7/24 world-class HiveMQ support

Take a Look

HIVEMQ CLOUD DEMO



Coming soon: Smaller Plans

Choose Your Location

Support for all big cloud providers

Google Cloud

amazon web services

Azure

Support for multiple regions

Third Party Integrations

okta

ForgeRock

External authentication providers

confluent

Kafka / Confluent Cloud

APACHE kafka

amazon web services

Google Cloud

Azure

Cloud Provider Services (like Device Management)

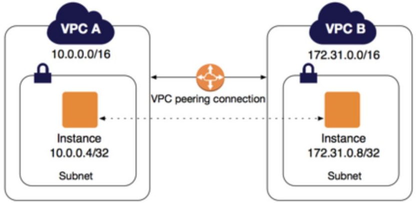
influxdb

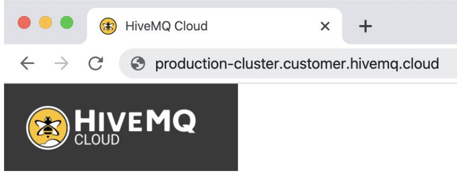
Other Best of Breed Services

Connect once, Integrate everywhere

Coming Soon...

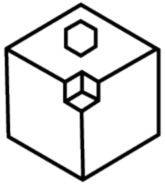
VPC Peering





Custom URLs

Custom + Pre-Built Extensions



Resources

HiveMQ

MQTT & MQTT 5 Essentials

[illegible]

Get Started with MQTT

MQTT Essentials Series

Evaluate HiveMQ Broker

Try HiveMQ Cloud

ANY QUESTIONS?

Reach out to community.hivemq.com

HIVEMQ

THANK YOU