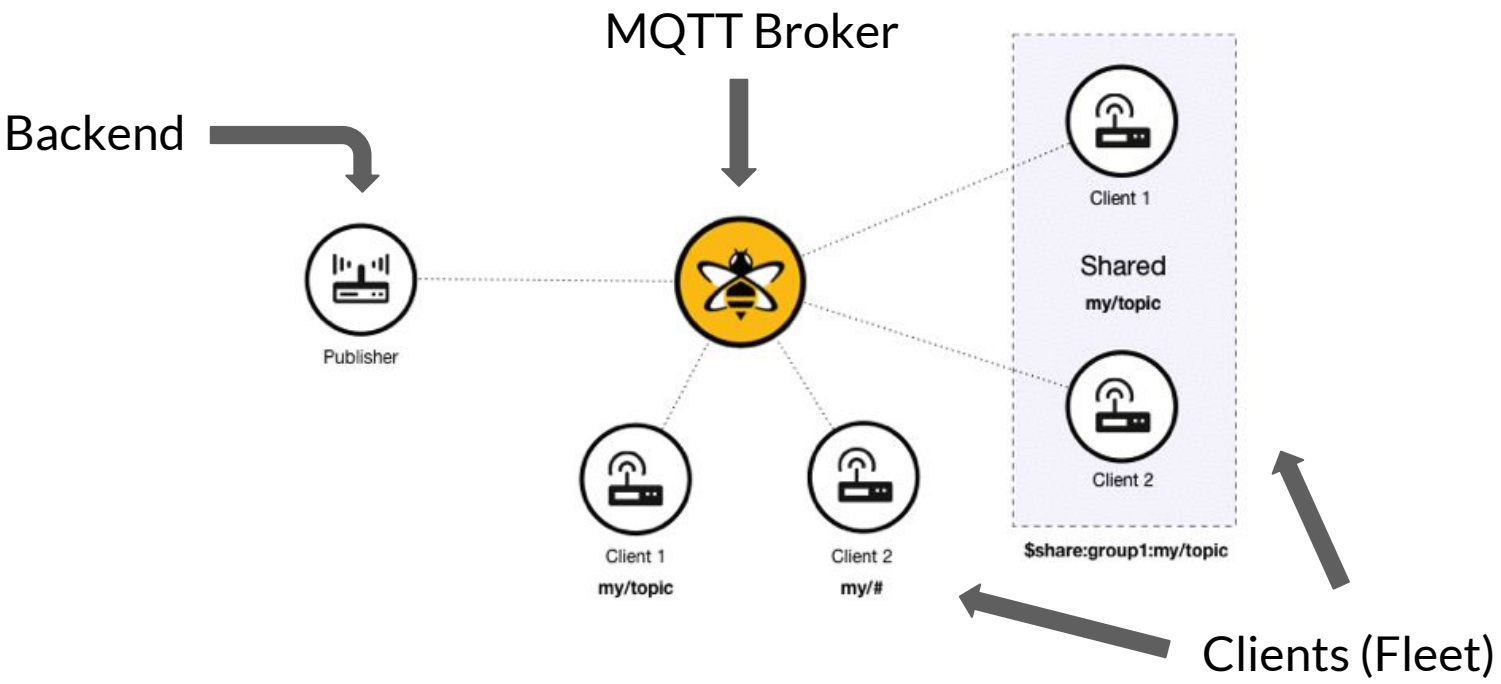
inovex

The journey of shifting the MQTT message broker HiveMQ to Kubernetes

Arnold Bechtoldt

Background

Use Case: Fleet Management



Design Flaws

- MQTT clients are *everywhere*
- One *fat* pet (cluster)
- *Manual* MQTT broker deployment
- Painful up-scaling
- Antiqued configuration management

The Plan

Docker!

Docker + Kubernetes!

Kubernetes works best with an app that is ...

- Stateless
- Web-based (HTTP interfaces)
- Cloud-ready

HiveMQ wasn‘t ...

- Stateless
- Web-based (HTTP interfaces)
- Cloud-ready

END

Kubernetes because ...

- Already in use (for microservices)
- High flexibility
- A lot of knowledge and skills at inovex :-)
- Close collaboration with HiveMQ support
- *Easy is not an option!*

Facing the Truth: Challenges

Challenge: HiveMQ Cluster Discovery (1)

- Existing mechanisms not cloud-ready
- Kubernetes API integration was needed
- We've created a new HiveMQ plugin
- Final completion & open sourcing by dc-square

Challenge: Loadbalancing Capabilities (2)

- Cloud's main *language* is HTTP
- MQTT works different
- TLS required
- Proxy protocol (v2) for authn & authz

Challenge: Determining App Health/Readiness (3)

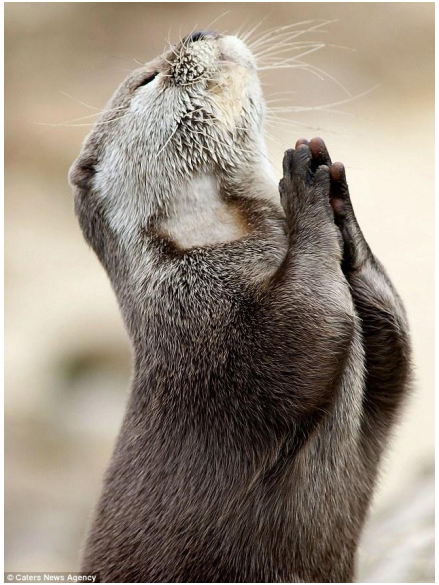
- Health: Is it working properly?
- Readiness: Is it ready for clients/traffic?
- No smooth way to find out
- Log event available only
- New HTTP interface in the pipeline (WIP)

Challenge: Missing HiveMQ Application Insights (4)

- Kubernetes won’t simplify your IT architecture
- Applications must be verbose (see 12factor app guide)
- Your containers *will die*
- Overall troubleshooting becomes more complicated
- Already work in progress

Hands on: Demo

Demo Time



You know what happens with **live demos**?
Right, they always **fail**!

Demo Time: Video!

youtu.be/Ly_f221O6Fo

Arnold Bechtoldt

inovex GmbH

Feedback & Questions:

sayat.me/arbe

arnold.bechtoldt@inovex.de

---

github.com/bechtoldt

youtube.com/inovexGmbH

arbe.io

Q&A