Building highly available, horizontally scalable MQTT Broker Clusters
About this document

MQTT is based on a publish/subscribe architecture that decouples MQTT clients and uses a central MQTT broker for distributing messages in a very performant and efficient manner. Traditionally, MQTT brokers are deployed as a single instance. This makes a broker deployment hard to scale and the single deployment is a Single Point of Failure, which means the whole MQTT communication is offline when the single broker fails.

This document discusses how you can eliminate these problems by creating a MQTT broker cluster that is scalable in a horizontal fashion (which means you can add any number of MQTT brokers at runtime) and eliminates the single point of failure by clustering multiple broker nodes to one logical MQTT broker. You will learn how you achieve true high availability and linear scalability with HiveMQ.

Table of Contents

1. Introduction ................................................................................................................. 2
2. The Only MQTT Broker that scales horizontally ......................................................... 2
3. The Distributed Cluster explained .............................................................................. 2
4. Availability - Paramount for the IoT ............................................................................ 3
5. Best in Class Performance and Scalability ................................................................. 4
6. Zero Downtime Upgrades ............................................................................................. 4
7. Discovery Mechanisms ................................................................................................. 4
8. Support for all Major OS and Cloud Providers ............................................................ 4
9. A Plugin System designed for MQTT Broker Clusters ................................................ 5
10. Conclusion .................................................................................................................. 5

© dc-square GmbH 2016. All rights reserved.
Introduction

HiveMQ is a reliable and resilient high-performance MQTT Broker, engineered in Germany for the professional use in companies and enterprises worldwide. It’s a dedicated MQTT Broker with a unique feature set, optimized for millions of concurrent MQTT connections. The broker is highly event-driven and multi-threaded to ensure lowest latency and highest throughput even for the most ambitious MQTT projects.

One of the most outstanding and unique features of HiveMQ is the ability to form resilient MQTT broker clusters that eliminate a Single Point of Failure for the MQTT communication by providing high availability and true horizontal scalability due to the distributed architecture of the HiveMQ cluster.

The only MQTT Broker that scales horizontally

HiveMQ is the only MQTT Broker that scales in a true horizontal fashion without trading availability or MQTT reliability guarantees. We all know from basic computer science theory that all systems eventually fail and this is why HiveMQ is built to be resilient and the cluster is able to self-heal in case any disastrous events happen. Network Splits are handled gracefully without losing the availability characteristics of certain cluster nodes; the cluster as a whole is fully operational, as long as at least a single node is still healthy.

HiveMQ is able to scale to a vast amount of cluster nodes and is truly horizontal scalable by adding (or even removing) cluster nodes at runtime without any reconfiguration of existing cluster nodes. It’s possible to achieve extreme numbers of concurrent MQTT clients by spawning additional HiveMQ instances as soon as you need them - and easily remove them if they are not needed anymore. All prevalent load balancing solutions, both hardware and software, are fully supported by HiveMQ.

The distributed Cluster explained

HiveMQ is designed with a true distributed and masterless cluster architecture, which means there is no single point of failure and the cluster can grow and shrink at runtime without losing data or trading availability. MQTT clients can (re-)connect to any
HiveMQ cluster node and can resume their MQTT session. The HiveMQ cluster implements a sophisticated and very efficient message routing that ensures the intra-cluster communication is at a minimum while all MQTT guarantees are maintained at lowest latency.

HiveMQ’s cluster is designed for both, high availability and scaling out use cases. For ambitious MQTT deployments with many millions of clients, any number of broker nodes can be deployed at runtime to scale out while maintaining stability and availability at any point of time. Full data replication between cluster nodes is possible but is not needed - data can be distributed with configurable replica counts.

The HiveMQ cluster is self-healing, which means that even if network splits occur or any kind of connectivity problem between nodes arise, the cluster as a whole is still available, as long as at least a single node is still healthy, and heals itself in error scenarios. No human interaction is required in such cases.

From a MQTT clients standpoint, the cluster is completely transparent, which means the MQTT client can interact with any node and the HiveMQ deployment always looks as “one broker”, even if a vast number of nodes is deployed. This is very important if a load balancer is in front of the nodes, since the load balancer can distribute the client to any broker node.

Availability - Paramount for the IoT

Availability is key for critical infrastructure components - this is especially true for MQTT broker clusters. Even if parts of the cluster fail, the system as a whole must be available to avoid service interruptions.

HiveMQ is a MQTT Broker that is resilient against network splits and strives to be always available, even in the presence of network partitions. This makes HiveMQ an eventually consistent system that strives towards consistency under normal operation and doesn’t trade availability at any time. An arbitrary number of broker nodes can fail or can be unavailable and the system as a whole is still online and fully functional.

This is one of the unique characteristics of HiveMQ that make it a highly resilient and scalable distributed MQTT Broker. A single MQTT broker node can potentially serve multiple clients.
hundreds of thousands and up to millions of open TCP connections, it would not be viable to trade availability characteristics and disconnect or halt these connected MQTT clients just because of a networking problem between brokers. HiveMQ never disconnects or halts MQTT clients just because the node they’re connected to is part of some kind of minority partition.

**Best in Class Performance and Scalability**

A single HiveMQ node is designed for multiple hundreds of thousands and even millions of concurrently connected MQTT clients and it’s easy to scale HiveMQ vertically by adding more resources (CPU & RAM).

The masterless and distributed architecture of the HiveMQ cluster allows to scale horizontally, which means you can add an arbitrary number of nodes (or even remove nodes) at runtime to serve more MQTT clients when needed. Deployments that experience traffic or usage spikes can scale up quickly if needed.

**Zero Downtime Upgrades**

HiveMQ supports Rolling Upgrades, which means that zero-downtime upgrades are possible when using a HiveMQ cluster. All nodes can be upgraded one-by-one.

When upgrading broker nodes, the cluster is still fully functional without any performance or reliability degradations. In case of unforeseen problems when upgrading you can stop the upgrade process and go back to the older version any time, no data is lost. DevOps will love the fact that the update can be done in a completely automated fashion.

**Discovery Mechanisms**

Multiple Cluster Discovery Mechanisms are supported by HiveMQ out-of-the-box. This means that you can adapt the cluster elasticity to your project needs.

Beside static cluster configuration it’s possible to use auto-discovery mechanisms like UDP Multicast, TCP Broadcast or even more sophisticated cluster discovery mechanisms like AWS S3 discovery or Consul Discovery with off-the-shelf plugins. You need to integrate with your own service registry...
application? No problem, the discovery mechanisms are pluggable via the powerful HiveMQ plugin API that is available to developers.

Support for all Major OS and Cloud Providers

When it comes to HiveMQ deployments, public cloud providers like AWS, Microsoft Azure and Google Cloud Platform are first-class citizens as well as hybrid or private clouds. HiveMQ runs on any operating system like Linux (e.g. RHEL, CentOS, Ubuntu, Debian) or Microsoft Windows Server.

It’s easy to scale out your cluster deployments with Docker and tools like Kubernetes or Mesos (with Marathon).

A Plugin System designed for MQTT Broker Clusters

The HiveMQ Plugin system allows to extend the broker with any custom business logic. All Plugin Services available for your own plugins are designed for interaction with both clustered and single HiveMQ instances. Very efficient asynchronous programming styles are supported as well as synchronous programming paradigms for optimal plugin performance when needed.

Conclusion

Due to HiveMQ’s sophisticated cluster design that is tailored specifically for MQTT, HiveMQ is able to form highly-available, elastically and horizontally scalable MQTT Broker clusters that avoid a Single Point of Failure. The cluster is linearly scalable and nodes can be added and removed at runtime without any service interruption.

HiveMQ is designed to handle network splits and the system is available, as long as at least a single node is still healthy, even if some parts of the infrastructure become unavailable. Advanced self-healing mechanisms protect from fatal disasters.

These unique characteristics of the cluster make HiveMQ a superior choice for ambitious MQTT projects.

Not sure if the HiveMQ cluster is suitable for your use case? Talk to one of our experts: contact@hivemq.com