HiveMQ - The Enterprise MQTT Platform

HiveMQ is the MQTT platform for the connected enterprise: The puzzle piece between constrained devices and enterprise systems. Besides its best-in-class performance and scalability, custom Enterprise Integrations enable your business to handle huge message throughput at lowest latency and further process your data. HiveMQ is scalable, secure and simple by leveraging state-of-the-art technology.

HiveMQ empowers your organization to connect all your devices and services with minimal effort by using the de-facto IoT standard protocol MQTT. HiveMQ’s natural habitat is in the frontier between devices and enterprise systems. It makes sure all your devices and services share information via instant, bi-directional push messages and can send their current data to your backend systems in real time.

Where is HiveMQ used?

HiveMQ is used in a variety of different industries, including:

- Automotive
- Automotive Supplier
- Chat applications
- Connected Vehicle
- Device Monitoring
- Energy Monitoring and Management
- Healthcare
- Industry 4.0
- Logistics
- Manufacturing
- Military
- Mobile communications
- Notification Services
- Retail
- Sensor Networks
- Smart Home

These industries have one thing in common: they need a reliable, secure and scalable foundation for their communication needs. HiveMQ enables these customers to build their solutions on top of MQTT.
HiveMQ Characteristics

Scalable and Cloud Native

HiveMQ has been built to handle hundreds of thousands of devices with extreme message throughput on a single node and millions of devices in cluster mode. Benchmarks proof the best-in-class performance and unique features like MQTT Broker Clustering bring extraordinary stability and speed to mission-critical MQTT deployments. Single HiveMQ nodes are vertically scalable while the deployment is also scalable horizontally. HiveMQ is designed for the cloud native environments (e.g. Docker on Kubernetes, OpenShift, or any of the major cloud platforms like GCP, AWS or Azure) and is easy to auto-scale. Traditional Enterprise IT landscapes are supported as well by using bare metal servers or virtual machines.

Secure

HiveMQ is built for mission-critical deployments with highest security requirements. Advanced Authentication and authorization mechanisms are key elements of secure MQTT deployments and even the most advanced security mechanisms like OAuth 2.0 can be easily plugged into HiveMQ with the Java based extension system. All state-of-the-art security standards like TLS 1.0, TLS 1.1 and TLS 1.2 are supported to keep your data safe while being transferred to and from HiveMQ. HiveMQ is typically deployed in environments where security is critical.

Simple

Simplicity is HiveMQ’s main philosophy. The broker can be downloaded and started without complicated installation steps. For production deployments, startup scripts for Linux systems are provided and a Windows Service is available. An extendable, holistic monitoring approach using industry standards like JMX allows the plugging in of your own business metrics.

Reliable & Resilient

The most important characteristic of HiveMQ for production use cases is its superior reliability. MQTT brokers are the heart of the communication between backends and devices. A broker outage would be fatal and HiveMQ avoids a loss in availability with sophisticated mechanisms in both, single node and cluster modes. When restarting a broker instance, all prior states are recovered due to disk persistence for single node deployments and due to replication mechanisms in the cluster. The cluster is able to heal itself for most failure scenarios and the first-of-its-kind Cluster Overload Protection prevents cascading failures induced by MQTT clients.
HiveMQ Key Features

100% MQTT Compliant

HiveMQ supports all features of the MQTT 3.1, MQTT 3.1.1 and MQTT 5 specification with 100% specification compliance. Different MQTT versions can be used simultaneously and full interoperability between these standards is ensured at the broker level. All major MQTT client libraries (like Eclipse Paho and HiveMQ MQTT Client) are supported out-of-the-box. The HiveMQ team is involved in the MQTT specification committee at OASIS and is committed to supporting future versions of MQTT from day one in a backwards compatible way.

Elastic Clustering

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 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 problems between nodes arise, the cluster as a whole is available and heals itself in error scenarios.

From an MQTT client’s standpoint, the cluster is completely transparent, which means the MQTT client can interact with any node and the HiveMQ deployment always appears as “one broker”, even if a vast number of nodes is deployed.

HiveMQ allows configuring static clusters with pre-defined servers or elastic cluster forming with UDP Multicast, Broadcast, and custom discovery mechanisms via the extension system. Various off-the-shelf extensions - like an AWS S3 discovery - are available for free and it’s trivial to integrate your own discovery mechanisms with custom extensions.
Java based extension system

Enterprise MQTT projects rarely start with a green field. Beside off-the-shelf extensions and Enterprise Integrations, HiveMQ allows hooking into virtually every aspect of the Broker and MQTT logic to cover your project’s use case. A free and open source Java-based SDK is available together with extensive documentation and tutorials. HiveMQ is the most extendible MQTT broker and allows your developers to integrate your existing systems with ease. Professional support and extension development directly from the HiveMQ team is available for mission-critical deployments.

Native Websockets

HiveMQ was one of the pioneers for bringing MQTT to the web. Modern web applications profit from MQTT over websockets by bringing the event-driven and real-time MQTT push communication paradigm to the browser. No additional third party software is needed for operating HiveMQ with websockets. All major JavaScript libraries for MQTT web development, like Eclipse Paho or MQTT.js, are supported and HiveMQ provides interoperability with standard MQTT clients. Using MQTT over websockets with TLS is fully supported.

Modern Control Center

HiveMQ has a built-in Control Center for monitoring, observability and advanced administrative actions. A powerful and modern dashboard gives a complete overview of the broker cluster and the health of the brokers as well as the most important monitoring metrics. Detailed information about all MQTT clients can be displayed together with administrative actions like modifying subscriptions and disconnecting a specific client. The Control Center works seamlessly even for millions of concurrent connections. Client Trace Recordings can be created manually and advanced analysis functionality is available (like monitoring dropped messages).

Deploy Everywhere

Deploying HiveMQ to your environment is a breeze. HiveMQ runs without any complex installation routines and works on Linux, Windows and OS X. Deploy HiveMQ with ease on server hardware, Virtual Machines or Containers (like Docker) in your private, hybrid or public cloud. We don’t force you to use a (virtual) appliance, you can install HiveMQ virtually everywhere: In your own computing center, on your local machine or on IaaS providers like AWS, Azure, or Google Cloud Platform.
Other notable features

- **Enterprise-grade security**: SSLv3, TLS 1.0, TLS 1.1 and TLS 1.2 are supported via the JVM SSL implementation or via native SSL by utilizing BoringSSL. Cipher Suite Whitelists are supported. Any authentication and authorization mechanism (like OAuth 2.0, database, LDAP or microservice backed authentication) can be plugged in. HiveMQ supports Blacklist and Whitelists for authorization. MQTT clients can get authenticated based on username/password, client id and IP address. Global throttling and client-based throttling is supported. MQTT message size can be restricted globally or on a per-client basis.

- **Shared Subscriptions**: Arbitrary MQTT clients can share the same topic subscriptions and the messages are distributed in a round-robin fashion between subscribers. Shared subscriptions are also available for HiveMQ clusters and multiple optimizations guarantee best performance for a distributed MQTT cluster.

- **Holistic Monitoring**: The holistic monitoring approach allows adding custom metrics easily to the 800+ built-in metrics. All metrics can be exposed effortlessly to your monitoring systems. There are off-the-shelf extensions available for InfluxDB and Prometheus. Integrating custom monitoring systems with custom extensions is straight forward.

- **Best-in-class performance**: HiveMQ instances scale with the underlying hardware. The non-blocking and multi-threaded approach allows scaling to hundreds of thousands of concurrent MQTT connections and extreme message throughput with minimal latency. Adding additional RAM and CPUs to the machine increases the scalability.

- **Advanced Analysis and Observability features**: Get instant insights on the health and condition of your MQTT deployment with advanced analysis like a dropped message monitoring. Trace Recordings allow filtering the MQTT traffic and client interactions and log them at scale to find the needle in the haystack when needed.

- **Cluster Overload Protection**: HiveMQ has a first-of-its-kind Cluster Overload Protection Mechanism built-in. The whole cluster is able to spot MQTT clients that cause overload on nodes or the cluster as a whole and protects itself from the overload. This mechanism also protects the deployment from cascading failures due to slow or failing underlying hardware (as sometimes seen on cloud providers).
System Requirements

HiveMQ is highly portable and runs on any server hardware and OS that runs a Java Virtual Machine. The broker scales with the underlying hardware and scales vertically.

For production environments - depending on the use case - we recommend at least:

- 4 CPUs
- 4 GB RAM
- 100 GB or more disk space
- OpenJDK JRE 1.11 (or higher)

HiveMQ clusters are designed to run in the same computing center and require good (at least 1Gbit or more) connectivity between cluster nodes.
Technical Aspects

MQTT

MQTT Standard Features
• MQTT 3.1, MQTT 3.1.1 and MQTT 5 supported. The protocols are supported in an interoperable way
• QoS 0, QoS 1, QoS 2 fully supported
• Retained Messages are fully supported
• MQTT Ordered Topic Guarantees for QoS 1 and 2
• Dynamic Topics
• Message Queuing
• Clean and Persistent sessions
• Last Will and Testament
• Wildcard Subscriptions
• Keep Alive
• Client Takeover
• Message Support up to 256MB
• Topic Length is restricted to 65535 UTF-8 characters and topic depth is restricted to 1000
• Unlimited Client Subscriptions
• Maximum Client identifier length: 65535
• Username / Password fields supported
• Shared Subscriptions for MQTT Client Load Balancing
• Client Session and Message Expire
• Key-Value Header for MQTT packages (User Properties)
• Negative Acknowledgements
• Server-side initiated disconnect
• Flow-Control
• Client- and Server-side indication of maximum supported message size
• Server-side overwriting of client parameters
• Topic Alias
• Reason Codes

Additional MQTT Features
• MQTT over HTML5 websocket support
• Secure HTML5 websockets with TLS

Configuration
• XML based, single file configuration
• Configuration file variable Interpolation with environment variables
• Reconfiguration at runtime via extensions
• Diagnostics Mode
• Support for multiple NICs
• Multiple simultaneous listeners:
  ▪ MQTT over TCP
  ▪ MQTT over TLS
  ▪ MQTT over Websockets
  ▪ MQTT over Secure Websockets
• Disk Persistence and In-Memory Mode
• Global and client based throttling
• Global and client based maximum MQTT message size
• Global maximum connection limit
• Configurable maximum client identifier length
• Configurable Retry Interval
• Configurable Connection Timeouts
• Configurable file persistence
• Configurable maximum queued messages per client
• Configurable Logging
• Windows Service available
• Linux Autostart Scripts: Debian, Ubuntu, CentOS, RHEL, systemd
• PROXY Protocol support (v1 and v2) with additional TLVs *
• Auto reloadable Logging configuration
• Additional shared subscription syntax that is compatible with most MQTT libraries
• Configurable Socket Send and Receive Buffer size
• HiveMQ folder configuration via system environment variables

* only available in HiveMQ EE
Control Center

- Modern web user interface
- A single Control Center instance shows the state of the whole cluster
- The Control Center is embedded with HiveMQ, no additional setup is required
- Powerful dashboard with basic monitoring options for connections, MQTT message rate (in/out), subscriptions, Retained Messages, Queued Messages and cluster nodes.
- Attention log for administrators
- Cluster node detail information with Memory, Disk Space, and Network statistics
- License information
- Advanced Analysis for dropped messages
- List of installed extensions
- Client Trace Recordings *
- Multiple users configurable
- User session management
- List of all connected clients in the whole cluster. Supports > 1.000.000 concurrent client connections
- Help texts for administrative actions
- Detailed Client overview with information about the MQTT session, the TCP connection, TLS, LWT, Client Restrictions and Proxy Protocol
- Ability to add and remove MQTT client subscriptions for individual clients
- Ability to add and remove MQTT client shared subscriptions for individual clients
- Ability to invalidate and delete MQTT Client Sessions
- Cluster-wide backups
- Desaster recovery by importing backups
- Role Bases Access Control for the Control Center

Trace Recordings *

- Detailed recordings of MQTT traffic and client behaviour
- Cluster wide recordings
- Time limited with a configurable end time
- Ad-hoc executions and scheduled executions
- Filterable by MQTT message type
- Filterable by client identifier Regular Expressions
- Filterable by MQTT Topic Regular Expressions
- Human Readable text file recordings
- Ability to download zipped versions of Trace Recordings of all Cluster nodes via the Control Center

Environment

- Linux, Windows, Mac OSX supported for development environments.
- All Linux distributions supported for production. CentOS 7 or other RHEL based distributions are recommended.
- Docker support
- Public Cloud Support: AWS EC2, Google Compute Engine, Microsoft Azure, Rackspace, Profitbricks, OpenStack. Most other IaaS vendors are also supported.
- Virtual Machine (Supported Hypervisors: VMWare ESXi, Xen, KVM, Microsoft Hyper-V) and Bare Metal Support

* only available in HiveMQ EE
Cluster

- Masterless cluster architecture
- Distributed cluster; linearly scalable
- Configurable Replica Count
- Transparent: A MQTT client can reconnect to any cluster node anytime and can resume its session and receives all messages.
- Hot-Standby supported by configuration of full replication
- Static and Elastic cluster discovery mechanisms (UDP Multicast, Broadcast)
- Free off-the-shelf extensions for advanced discovery mechanisms (e.g. AWS S3)
- Network Partitions are supported. Self healing mechanisms are triggered after Split-Brain scenarios
- No loss in availability when network partitions occur. All MQTT clients stay connected and can resume operation, even in minority partitions
- Eventually consistent in Split-Brain scenarios
- All L4 load balancers are supported
- Rolling upgrades
- Pluggable cluster discovery mechanisms
- TCP and UDP transports available
- Configurable Failure Detection Mechanisms
- Configuration of external IP addresses for cloud and container environments
- Secure cluster communication with TLS
- Ability to start cluster nodes statefully
- Configurable Topic Level Distribution for Topic Trees
- Dynamic Replicates
- Node Stress Level Metrics
- Cluster Overload Protection

Security

- SSLv3 (disabled by default), TLS 1.0, TLS 1.1, TLS 1.2 support
- All JVM cipher suites are supported
- Cipher Suite Whitelists
- JKS Key- and Truststores
- X509 client certificate authentication support
- Configurable TLS Handshake Timeouts
- Pluggable authentication and authorization mechanisms
- Authorization with mixed Black- and Whitelists possible.
- Fine-grained client authorization that allows restricting:
  - Activity (Publish / Subscribe)
  - QoS levels
  - Retained Messages
  - Topic
- Restrictions for bandwidth throttling, message size throttling
- Authentication based on combinations of:
  - Client Identifier
  - Client IP
  - Username
  - Password
  - X509 client Certificate
- Free off-the-shelf extensions for authentication and authorization are available.
- Commercial Enterprise Integrations possible (e.g. OAuth 2.0, LDAP, databases, …)
- Support for chaining multiple authentication and authorization extensions
- Reloadable key- and truststores
- Native SSL integration (BoringSSL) *
- OCSP Stapling support *
- Configurable limitation of concurrent TLS handshakes *
- Audit logs *
Extension System

- Free and Open Source Extension System that allows extending HiveMQ for your business requirements
- X509 client certificate usable for custom authentication and authorization logic
- Free extension developer guide available
- Free and open source off-the-shelf extensions available
- Integration with Apache Kafka (as commercial Enterprise Extension available)
- Installing, enabling and disabling of Extensions at runtime (Hot-Reload)
- Maven Plugin Archetype available
- Debug your extension from Eclipse and IntelliJ IDEA.
- Commercial Enterprise Integrations
- Customized Authentication and Authorization
- Lifecycle support
- Support for adding and removing interceptors at runtime
- Multiple Services for interacting with the HiveMQ core:
  - Retained Message Store
  - Subscription Store
  - Client Service
  - Publish Service
  - Cluster Service
  - Managed Extension Executor Service
  - Initializer Registry
  - Security Registry
  - Event Registry
  - Metric Registry
- Interceptors available that allow modifying MQTT behaviour
- Configurable Extension Service Overload Protection which provides rate-limiting for custom extensions
- Enterprise SDK for commercial extensions

Monitoring

- JMX
- 800+ Metrics
- Log files with time based rolling policy
- Different log levels configurable: TRACE, DEBUG, INFO, WARN and ERROR
- Holistic monitoring interface which enables customers to integrate their own business metrics to the standard monitoring interface.
- Free off-the-shelf monitoring extensions available: InfluxDB, Prometheus
- Custom Monitoring extensions are pluggable
- Diagnostic startup script for starting HiveMQ in diagnostic mode
- Native System Metrics: HiveMQ process metrics for Linux Systems

* only available in HiveMQ EE
More Information

You can find more information about the Enterprise MQTT Broker HiveMQ, extension development and commercial services at http://www.hivemq.com.

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