





























# Full comparison of open-source and Pro Mosquitto features



















Last updated: 21.01.2025

Feature name	Feature description	Open-source Mosquitto	Pro Mosquitto
<b>High Availability &amp; Clustering</b>			
<b>Clustering for High Availability</b>	The High Availability feature connects multiple brokers and syncs their internal status, configuration, and data - providing fail-safety against hardware defects or network outages. Clustering for High Availability runs a cluster of Mosquitto nodes with a single leader and multiple followers, ensuring uninterrupted service even if a single node goes down. Users can choose between two cluster modes - Full Sync and Dynamic-Security Sync. <a href="#">Learn more.</a>	✗	✓
<b>Cluster Management UI</b>	Management Center for Mosquitto provides a UI for managing broker clusters. <a href="#">Learn more.</a>	✗	✓
<b>Load balancer</b>	The Mosquitto MQTT cluster relies on a load balancer to monitor server availability and closes ports for inactive nodes. If the leader server fails, the cluster reorganizes and appoints a new leader, by which the load balancer will direct all clients to the new leader. <a href="#">Learn more.</a>	✗	✓
<b>Cluster Management REST API</b>	The Cluster Management REST API provides access to detailed Mosquitto cluster information, e.g., which HA clusters the Management Center is connected to and which broker nodes belong to which cluster.	✗	✓
<b>Dynamic-Security Cluster Mode</b>	An HA cluster mode with all nodes available for load distribution, but these nodes only sync their cluster state and dynamic-security-related changes (e.g., adding a client). <a href="#">Learn more.</a>	✗	✓
<b>Enhanced High Availability (HA) Monitoring</b>	The enhanced HA monitoring provides users with real-time output metrics such as voting node count, leader connectivity status, and cluster status indication for deeper insights into the HA cluster's health and resilience. <a href="#">Learn more.</a>	✗	✓
<b>Administration</b>			
<b>Cedalo Cloud Account Management</b>	Cloud Account Management provides a centralized interface for managing all Cedalo services, including cloud and on-premises deployments. <a href="#">Learn more.</a>	✗	✓
<b>Management Center for Mosquitto</b>	The Management Center provides a web-based user interface for managing Mosquitto brokers. <a href="#">Learn more.</a>	✓ (Limited to the OS functionality)	✓
<b>Maximal connected brokers</b>	Connects to and manages multiple broker instances with the Management Center. <a href="#">Learn more.</a>	✗ (Limited to one broker node)	✓











<b>Standalone Web Application</b>	Management Center for Mosquitto can run independently and does not need to be on the same server as the connected Mosquitto broker instances (including authentication with a TLS client certificate).	 (in general)	 (TLS)
<b>Broker restart</b>	The Management Center enables you to restart your (single) cloud broker instance.		
<b>Admin Board</b>	The Admin Board serves as a centralized interface for managing and configuring users, administrators, brokers, projects, and settings for customizing MQTT broker operations. <a href="#">Learn more.</a>		
<b>Security</b>			
<b>OAuth 2.0 / JWT Authentication</b>	The OAuth 2.0 / JWT Authentication feature introduces OAuth-based authentication for MQTT clients. As a result, clients can use token-based authentication mechanisms, which significantly enhance security and simplify user permissions management. <a href="#">Learn more.</a>		
<b>LDAP Authentication</b>	The LDAP Authentication feature enables LDAP-based verification for MQTT clients, offering a standardized and efficient way to authenticate client identities and enhance security through directory-based mechanisms. <a href="#">Learn more.</a>		
<b>Client Authentication: ID+password</b>	Basic authentication, where clients provide a username and a password.		
<b>Client Authentication by Certificate</b>	MQTT clients have to use the encrypted connection via TLS and provide a client certificate to authenticate.		
<b>Client Authentication by Pre-Shared Key (PSK)</b>	The client and the broker both have a Pre-Shared Key, which is used for authentication and verification. Here, no certificate management is needed.		
<b>Dynamic security plug-in</b>	This broker extension provides dynamic user management, authentication, and authorization via configuration file or a control API for run-time changes. <a href="#">Learn more.</a>		
<b>Access Control Lists (ACLs) based on client, role, and group levels</b>	Access Control Lists (ACLs) can be used for authorization based restrictions on configured topics.		
<b>Anonymous client access</b>	Allows clients to access and connect to the broker anonymously without a username or password.		
<b>Login rate limit</b>	Limits the number of login attempts for Management Center users to increase security.		
<b>Certificate Management</b>	Management of Client Certificate -based Authentication. <a href="#">Learn more.</a>		

<b>Custom CAs</b>	Custom CA verifies broker authenticity when connecting Mosquitto Management Center to a broker.		
<b>HTTPS/TLS termination at Management Center for Mosquitto</b>	HTTPS/TLS terminates directly at the Management Center, eliminating the need for a reverse proxy.	 (Manual configuration)	
<b>Audit Trail</b>	Audit Trail tracks critical MQTT broker and Management Center activities for better visibility, security, and operational transparency. <a href="#">Learn more.</a>		
<b>Integrations</b>			
<b>MQTT Bridge (In, Out, or Bidirectional Topic remapping)</b>	An MQTT Bridge connects a broker with another MQTT broker or service. A bridge defines topics the source broker forwards (publishes) or subscribes to on the target broker. When a bridge defines topics for subscribing and publishing, it enables bi-directional communication. <a href="#">Learn more.</a>		
<b>HTTP Bridge</b>	An HTTP Bridge forwards data from a broker to a web service that provides an HTTP endpoint. <a href="#">Learn more.</a>		
<b>Amazon Redshift Bridge</b>	The Amazon Redshift Bridge allows the automatic export of data from Pro Mosquitto to the Amazon Redshift database. <a href="#">Learn more.</a>		
<b>CockroachDB Bridge</b>	The CockroachDB Bridge allows the automatic export of data from Pro Mosquitto to the CockroachDB database. <a href="#">Learn more.</a>		
<b>Kafka Bridge</b>	The Kafka bridge enables streaming MQTT data into the Kafka ecosystem by establishing a unidirectional connection between the broker and Kafka server. Users can specify the server, select topics to publish, and map MQTT topics to Kafka topics. <a href="#">Learn more.</a>		
<b>MongoDB Bridge</b>	MongoDB bridge establishes a unidirectional flow of MQTT-generated data from the broker to MongoDB. Users can specify the database and collection for importing, select topics and message fields to import, map topics to specific collections, and choose the fields to be included in the collections. <a href="#">Learn more.</a>		
<b>Prometheus Metrics Exporter</b>	Prometheus Exporter allows Prometheus to query Pro Mosquitto for metrics to be monitored directly. The metrics set encompasses the number of clients connected, the count of individual MQTT message types sent and received, and more. <a href="#">Learn more.</a>		
<b>Google Pub/Sub Bridge</b>	The Google Pub/Sub bridge enables one-way data transfer from MQTT topics to Google Pub/Sub topics. Users can specify the Google Pub/Sub server, select MQTT topics to publish, map MQTT topics to Google Pub/Sub topics, select Google Pub/Sub topics for consuming, and map Google Pub/Sub topics to MQTT topics. <a href="#">Learn more.</a>		

















<b>Google AlloyDB Bridge</b>	Google AlloyDB bridge establishes a one-way data transfer from the broker to Google AlloyDB databases for efficient storage and analysis of high volumes of data. Users can specify the target Google AlloyDB database and table for data import, and map MQTT topics to corresponding Google AlloyDB tables for organized data storage. <a href="#">Learn more.</a>		
<b>MySQL Bridge</b>	The MySQL bridge facilitates a one-way flow of MQTT-generated data from the broker to MySQL databases. Users can specify the desired database and table for data import, choose specific MQTT topics and message fields, and map MQTT topics to corresponding MySQL tables for well-organized data storage. <a href="#">Learn more.</a>		
<b>PostgreSQL Bridge</b>	The PostgreSQL bridge seamlessly integrates MQTT data into PostgreSQL databases unidirectionally and offers an effective approach to data management and analytics capabilities. Users can specify the target PostgreSQL database and table for data import, map MQTT topics to corresponding PostgreSQL tables for organized data storage, and more. <a href="#">Learn more.</a>		
<b>TimescaleDB Bridge</b>	The Timescale DB bridge incorporates MQTT data into TimescaleDB databases to ensure efficient data storage and enhanced analytics capabilities. This bridge empowers users to define the target TimescaleDB database for seamless data ingestion, establish mapping between MQTT topics and TimescaleDB tables, and configure specific fields for insertion into TimescaleDB tables based on individual requirements. <a href="#">Learn more.</a>		
<b>MariaDB Bridge</b>	The MariaDB Bridge allows the automatic export of data from Pro Mosquitto to the MariaDB database. <a href="#">Learn more.</a>		
<b>Microsoft SQL Server Bridge</b>	The Microsoft SQL Server Bridge enables the seamless and automatic transfer of data from Pro Mosquitto to the Microsoft SQL Server database. <a href="#">Learn more.</a>		
<b>MongoDB Atlas Bridge</b>	The MongoDB Atlas Bridge extends the capabilities of the MongoDB Bridge to enable one-way data transfer from MQTT to MongoDB Atlas. Users can generate analytics by specifying the MongoDB Atlas database and collection for data import, selecting MQTT topics and message fields to import into MongoDB, mapping MQTT topics to specific collections within MongoDB, and choosing which message fields to insert into MongoDB collections. <a href="#">Learn more.</a>		
<b>Oracle DB Bridge</b>	The Oracle DB Bridge allows the automatic export of data from Pro Mosquitto to the Oracle database. <a href="#">Learn more.</a>		


<b>InfluxDB Metrics Exporter</b>	The InfluxDB Metrics Exporter exports Pro Mosquitto's operation metrics, including the number of connected clients, MQTT message transmissions, and other relevant performance indicators, to InfluxDB. Users can promptly detect and respond to abnormal behavior, ensuring optimal performance from the Pro Mosquitto broker. <a href="#">Learn more.</a>		
<b>Snowflake Bridge</b>	The Snowflake Bridge allows the automatic export of data from Pro Mosquitto to the Snowflake database. <a href="#">Learn more.</a>		
<b>Monitoring</b>			
<b>Broker Insights</b>	Broker Insights provides a dashboard with detailed metrics on broker performance, client connections, message flows, and system health for a deeper insight into your MQTT environment.		
<b>Listing of currently connected clients on a node</b>	The broker provides a control API with information about currently and previously connected clients and properties like MQTT protocol versions.		
<b>Client inspection</b>	Detailed information on each connected client, such as: <ul style="list-style-type: none"> <li>- connection status</li> <li>- connect/disconnect time</li> <li>- protocol information</li> <li>- IP address</li> <li>- TLS encryption information</li> <li>- last will message</li> <li>- message queue usage</li> <li>- subscribed topics</li> </ul> <a href="#">Learn more</a>		
<b>Client Control</b>	Client Control allows managing connected clients via a central MQTT API, e.g., subscribe to and unsubscribe MQTT clients from topics, disconnect them, etc. <a href="#">Learn more.</a>		
<b>PROXY Protocol v2</b>	The support for PROXY Protocol v2 in Pro Mosquitto enables the transmission of the client connection information through a proxy server, providing additional context for each connection. This feature is essential for deployments where Pro Mosquitto stands behind a load balancer or reverse proxy, ensuring that the original client IP address and connection details are preserved and accessible. <a href="#">Learn more.</a>		
<b>Broker status</b>	The broker status provides information about the status of the connected brokers. <a href="#">Learn more.</a>		
<b>Topic tree with drill-down</b>	The topic tree gives you an overview of the topic hierarchy for a given broker. It lets you inspect which topics are used, which messages are sent to those topics, and more drill-down metrics. <a href="#">Learn more.</a>		























## User management







<b>Authentication</b>	Secured access to the Management Center for Mosquitto by using username/password authentication. <a href="#">Learn more.</a>		
<b>Unlimited number of users</b>	The number of Management Center users.		
<b>User roles</b>	Configure Management Center users and assign roles to grant or deny them access to specific features and functionalities.		
<b>Role-Based Access Control (RBAC)</b>	Role-Based Access Control (RBAC) defines broker groups that are visible for certain users, roles, or user groups. <a href="#">Learn more.</a>		
<b>Single Sign-On (SSO)</b>	Single Sign-On (SSO) enhances secure user authentication by integrating with identity providers such as Okta, Active Directory, and Keycloak, while supporting protocols like OAuth 2.0, OpenID Connect, SAML, and LDAP, simplifying access control and enhancing security. Learn more about <a href="#">SAML</a> and <a href="#">LDAP</a> .		

## REST APIs

<b>Projects</b>	The Projects REST API allows for managing and organizing projects. <a href="#">Learn more.</a>		
<b>User</b>	The User REST API allows for managing and organizing users. <a href="#">Learn more.</a>		
<b>Broker</b>	The Broker REST API provides access and control of MQTT broker instances, connection settings, and licenses. <a href="#">Learn more.</a>		
<b>Clusters</b>	The Cluster Management REST API provides detailed information on MQTT broker clusters, enabling the efficient management of high availability broker setups. <a href="#">Learn more.</a>		
<b>Security - Access Control</b>	This Rest API controls the dynamic security feature for secure access control. <a href="#">Learn more.</a>		
<b>Security - Application Tokens</b>	The Application Tokens REST API enables the creation, management, retrieval, and revocation of tokens for secure REST API access. <a href="#">Learn more.</a>		
<b>Monitoring - Client Inspect</b>	The Client Inspect REST API enables you to retrieve detailed diagnostic information about connected clients. <a href="#">Learn more.</a>		
<b>Monitoring - Topic Tree</b>	The Topic Tree REST API provides information about the MQTT broker's topic tree for a better understanding and optimization of topic structures. <a href="#">Learn more.</a>		

<b>Streams</b>	The Streams REST API allows you to manage data streams on the MQTT broker, with functions for creating, updating, and replaying streams. <a href="#">Learn more.</a>		
<b>Project &amp; User Management</b>			
<b>Project Management</b>	Project management allows users to efficiently organize and manage multiple brokers in the Management Center by creating projects and group brokers within those projects. <a href="#">Learn more.</a>		
<b>Multi-Tenancy Support</b>	Multi-tenancy enables multiple teams or customers to work independently by creating isolated projects with dedicated users, brokers, and configurations. <a href="#">Learn more.</a>		
<b>User Management</b>	User management enables you to efficiently control user access and permission with Role-Based Access Control (RBAC), seamless integration with identity management systems (e.g., Active Directory, OpenLDAP), and a centralized interface for user onboarding and management. <a href="#">Learn more.</a>		
<b>MQTT protocol versions</b>			
<b>MQTT V3.1.1</b>	An older protocol version that is still widely used and a well-supported standard of MQTT. <a href="#">Learn more.</a>		
<b>MQTT V5</b>	The current MQTT standard provides additional features like session expiry, topic alias, reason codes, shared subscriptions, and more. <a href="#">Learn more.</a>		
<b>Supported protocols</b>			
<b>MQTT over TLS (MQTTS)</b>	Transport Layer Security (TLS) encrypted connection to protect the data sent and received by the broker. <a href="#">Learn more.</a>		
<b>WebSockets (WS)</b>	MQTT connection over WebSockets is a standard protocol that establishes persistent connections based on HTTP. Websockets can help create connections through restricted networks and firewalls. <a href="#">Learn more.</a>		
<b>WebSockets over TLS (WSS)</b>	TLS encrypted MQTT over WebSocket connections.		
<b>Sparkplug</b>	Sparkplug is a communication protocol, providing a standardized way for IoT clients to exchange real-time data efficiently and securely.		
<b>Sparkplug Decoding</b>	Sparkplug Decoder enables the broker to natively decode Sparkplug B payloads, enhancing monitoring and control of industrial devices. <a href="#">Learn more.</a>		

MQTT basics			
<b>Quality of Service (QoS) Levels</b>	MQTT QoS is a level of service that serves as a consensus between a publisher and a broker, as well as a broker and a subscriber. Regarding the latter, it guarantees to dispatch an MQTT message successfully. <a href="#">Learn more.</a>		
<b>Last Will messages</b>	The message that gets published on a specific broker topic if the client unexpectedly loses its connection to the broker. <a href="#">Learn more.</a>		
<b>Retained messages</b>	The broker keeps the last message received on a given topic. Clients will immediately receive these once they subscribe to that topic.		
<b>Persistent connections</b>	The broker will maintain a persistent session with the client, even if the client disconnects.		
<b>Mount points</b>	Mount points are topic prefixes that can create isolated topic trees per listener.		
Supported platforms			
<b>Windows</b>	Windows Deployment support enables users to install and run Pro Mosquitto natively on Windows operating systems.		
<b>Docker (Linux, Windows, macOS, Raspberry Pi)</b>	Docker support allows users to deploy an MQTT broker using Docker containers. Additionally, Docker Compose users can define and manage multi-container setups for faster, streamlined deployment. Learn more for a <a href="#">Single-node</a> , or a <a href="#">High Availability</a> Docker setup.		
<b>RPM (RedHat, CentOS, Rocky Linux)</b>	RPM package support enables users to deploy an MQTT broker with full control over installation and configuration in environments that use bare-metal servers or virtual machines. Learn more about the <a href="#">MQTT broker's RPM package</a> and the <a href="#">Management Center's RPM package</a> .		
<b>Kubernetes (Linux, Windows, macOS)</b>	Kubernetes support allows users to natively deploy an MQTT broker with Kubernetes container orchestration to effortlessly deploy, manage, and scale Mosquitto instances within Kubernetes clusters. <a href="#">Learn more.</a>		
<b>OpenShift</b>	OpenShift enables users to deploy and manage the MQTT broker in containerized environments, ensuring secure and scalable functionality of applications and services. <a href="#">Learn more.</a>		
Additional features			
<b>MQTT Client UI</b>	MQTT Client UI enables users to test and manage MQTT clients interactions directly in the Management Center without the need for third-party tools. <a href="#">Learn more.</a>		

<b>Persistent Queueing</b>	Persistent queueing in Pro Mosquitto stores messages to disk, thus increasing the broker's storage capacity. It further improves outgoing message retention per client and MQTT bridge and enhances reliable delivery even in environments with unstable connectivity. <a href="#">Learn more.</a>		
<b>Support of LMDB persistence</b>	The LMDB (Lightning Memory Mapped Database) provides Mosquitto with an even faster ability to save client sessions, retain message information to disk, and recover them in case of a restart.		
<b>Processing streams</b>	The number of processing streams that can be defined (e.g., selective redressing, persistence, and replay on separate topics). <a href="#">Learn more.</a>		
<b>White labeling</b>	Enterprise users can customize logos, color schemes, and the overall Management Center interface to align with their corporate identity. <a href="#">Learn more.</a>	