




















































# Full comparison of open-source and Pro Mosquitto features























Last updated: 17.06.2023

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>	Mosquitto Management Center 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 Mosquitto 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>Administration</b>			
<b>Mosquitto Management Center (MMC)</b>	MMC 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 Mosquitto Management Center. <a href="#">Learn more.</a>	 (Limited to one broker node)	
<b>Standalone Web Application</b>	Mosquitto Management Center 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>	Mosquitto Management Center enables you to restart your (single) cloud broker instance.		

Security			
<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 Mosquitto 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 certificates to verify broker authenticity when connecting Mosquitto Management Center to a broker.		
<b>HTTPS/TLS termination at Mosquitto Management Center</b>	HTTPS/TLS terminates directly at Mosquitto Management Center, so no reverse proxy is required.	 (Manual configuration)	
Bridges			
<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>		

Monitoring			
<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>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 Mosquitto Management Center by using username/password authentication. <a href="#">Learn more.</a>		
<b>Unlimited number of users</b>	The number of MMC users.		
<b>User roles</b>	Configure Mosquitto 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>	Integrates the Mosquitto Management Center User Management with your SAML-based SSO provider. <a href="#">Learn more.</a>		

REST APIs			
<b>Dynamic Security</b>	This feature enables the management of MQTT clients, groups, and ACL access rights. <a href="#">Learn more.</a>		
<b>Topic Tree</b>	The Topic Tree REST API provides information about the topic tree for every broker connected to Mosquitto Management Center.		
<b>Mosquitto Management Center User management</b>	The User Management REST API allows for managing MMC users, groups, and their roles. <a href="#">Learn more.</a>		
<b>Cluster Management</b>	The Cluster Management REST API allows you to get detailed information about Mosquitto clusters, e.g., which clusters Mosquitto Management Center is connected to and which Mosquitto nodes belong to which cluster.		
<b>Connections</b>	The Connections REST API helps manage connections from Mosquitto brokers to the Mosquitto Management Center (e.g., creating, updating, or deleting them). <a href="#">Learn more.</a>		
<b>Application Tokens</b>	The Application Tokens REST API enables the management of application tokens with role-based access & expiration dates. <a href="#">Learn more.</a>		
<b>Monitoring</b>	The Monitoring REST API provides monitoring information about the broker instances connected to Mosquitto Management Center and the different metrics for each broker instance. <a href="#">Learn more.</a>		
<b>Access control by application tokens</b>	Each Rest API can be accessed using username, password or API token for authentication. Learn more about <a href="#">application tokens</a> and <a href="#">REST APIs</a> .		
MQTT protocol versions			
<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>		
Supported protocols			
<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>	The Sparkplug specification defines a standard for structuring topics and payloads, mainly in the Internet of Things (IoT) scope.		
<b>MQTT basics</b>			
<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.		
<b>Supported platforms</b>			
<b>Docker (Linux, Windows, macOS, Raspberry Pi)</b>	—		
<b>RPM (RedHat, CentOS, Rocky Linux)</b>	—		
<b>Additional features</b>			
<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 of management center UI</b>	Users can customize the logo and colors used in Mosquitto Management Center. <a href="#">Learn more.</a>	