

Cisco MQTT DSLink User Guide

Kinetic - Edge & Fog Processing Module (EFM)

Revised: February 2, 2020

Version 2.0.1

Contents

Introduction	2
Filesystem	2
Installation, Life cycle management and Logging.....	2
General Design	3
Node Hierarchy	3
Connector Link Actions	4
Mappings.....	6
Subscriptions.....	8
Endpoint Link Actions	10
Authentication	11
Clients.....	13
References for securing the EFM operating platform	14
Obtaining documentation and submitting a service request.....	15

Introduction

This document describes the Cisco MQTT DLink for the Cisco Kinetic Edge & Fog Processing Module. The Cisco MQTT Link offers functionality as an MQTT endpoint, bridging MQTT into the DSA world, and an MQTT client to connect the DSA world to MQTT.

Filesystem

The installed Cisco MQTT DLink consists of:

Artifact	Type	Description
bin	Folder	Contains the binary and start script
lib	Folder	Contains all depending libraries
.key	file	Contains the DSA broker token
dslink.json	file	Configuration for the initial link configuration and start
nodes.json	file	Configuration for the current link configuration and current metric information

Installation, Life cycle management and Logging

The Cisco MQTT DLink uses the standard EFM mechanisms for install, upgrade and logging. Please look into the [Cisco Kinetic EFM System Administrator User Guide](#) for details.

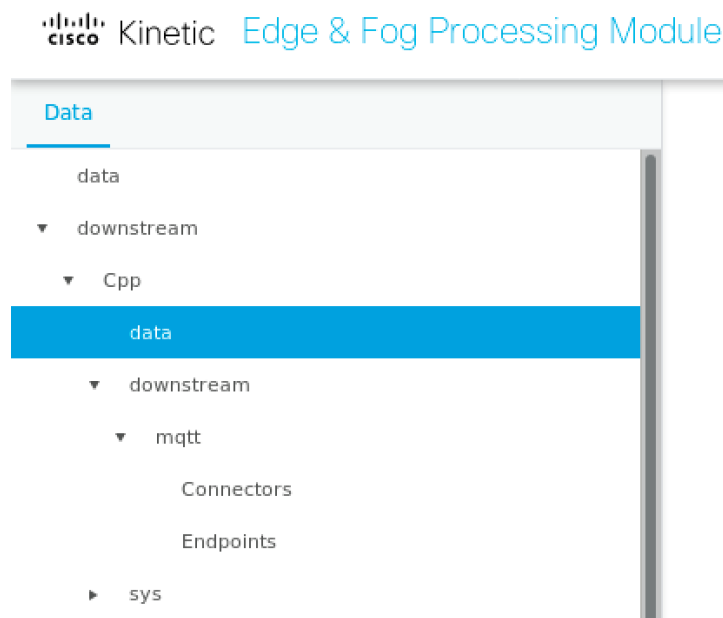
General Design

The Cisco MQTT Link can be used to open a connection to an MQTT server via a connector. A connector can be used to publish messages to an MQTT server and also to subscribe to topics in order to receive messages from the server.

The Cisco MQTT Link can act as an MQTT server for connecting clients. It does not implement a full MQTT server and is therefore called an MQTT endpoint and serves as an MQTT -> DSA bridge. The endpoint can be used by MQTT clients to connect and publish data through the link into the DSA world.

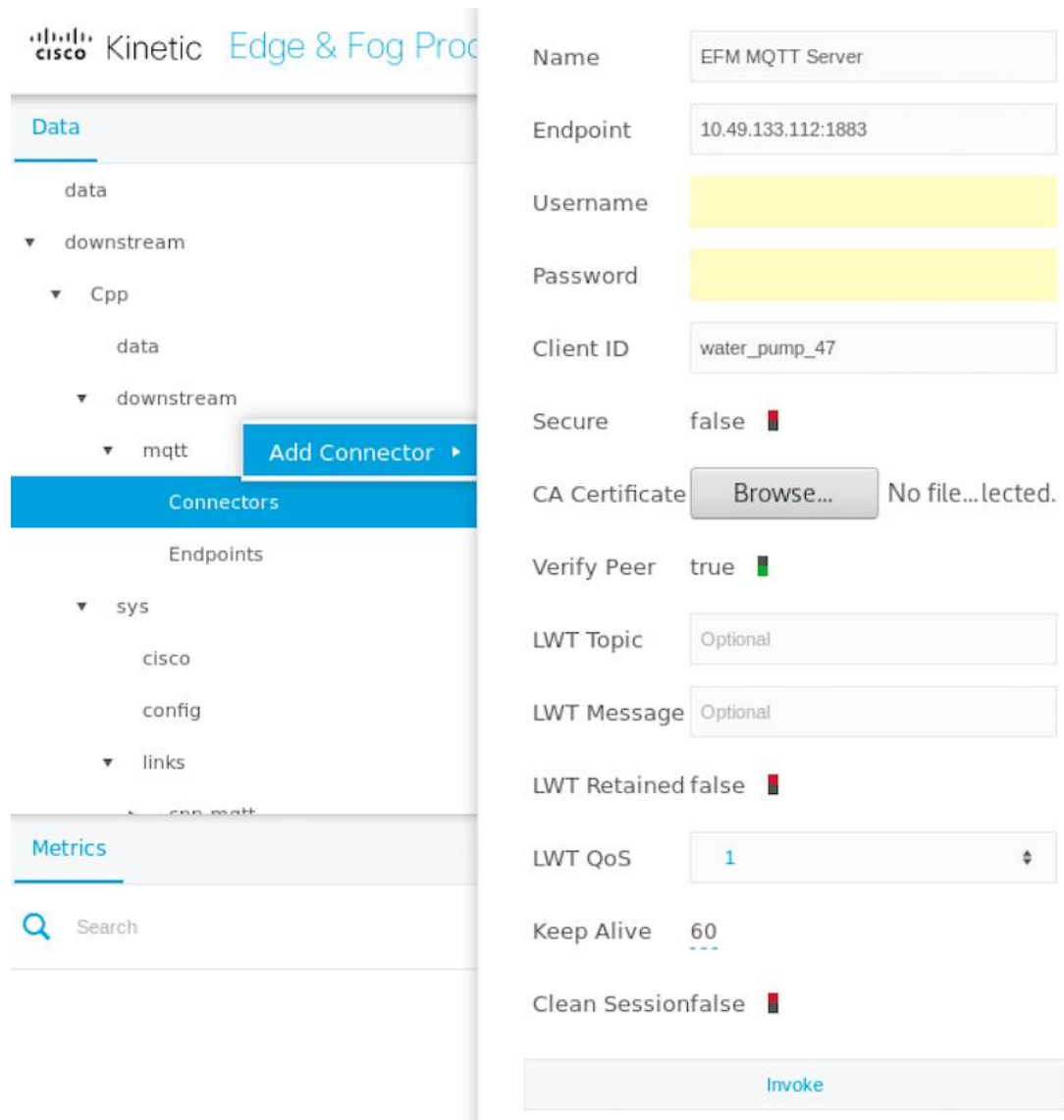
Node Hierarchy

The top hierarchy shows the Connectors and Endpoints nodes. Each has its own actions and subnodes.



Connector Link Actions

Using the **Add Connector** action, a new connector can be configured and created.



The screenshot shows the configuration page for an MQTT connector in the Cisco Kinetic Edge & Fog Platform. The left sidebar shows a navigation tree with 'Data' selected, and 'Connectors' highlighted. The main area displays the configuration for a connector named 'EFM MQTT Server'.

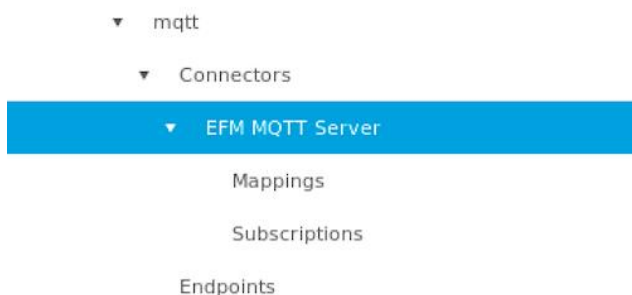
Name	EFM MQTT Server
Endpoint	10.49.133.112:1883
Username	[Redacted]
Password	[Redacted]
Client ID	water_pump_47
Secure	false <input type="checkbox"/>
CA Certificate	<input type="button" value="Browse..."/> No file...lected.
Verify Peer	true <input checked="" type="checkbox"/>
LWT Topic	Optional
LWT Message	Optional
LWT Retained	false <input type="checkbox"/>
LWT QoS	1
Keep Alive	60
Clean Session	false <input type="checkbox"/>

Each connector will be represented by a subnode under the Connectors node.

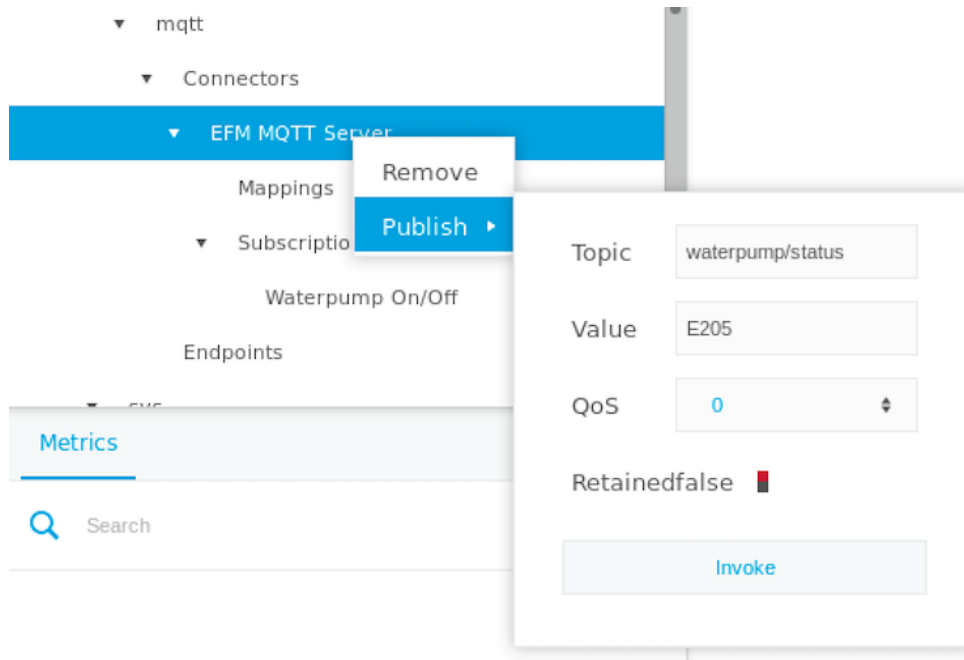
Parameter	Type	Description
Name	String	Unique name of the connector
Endpoint	String	Host followed by the port of the MQTT server to connect. The format is: <host>:<port> The host can either be a hostname or an IP address.
Username	String	If the MQTT server needs authentication, the username and password have to be specified. Has to be left blank for no authentication.
Password	String	If the MQTT server needs authentication, the username and password have to be specified. Has to be left blank for no authentication.
Client ID	String	The client id to use for the connection. If left blank no client id will be sent to the server.
Secure	Boolean	Set to true for a tls connection
CA Certificate	Binary	The CA certificate for a tls connection in PEM format
Verify Peer	Boolean	Set to true if the peer's certificate shall be verified. Optional for non tls connections.
LWT Topic	String	The last will and testament topic
LWT Message	String	The last will and testament message
LWT Retained	Boolean	If the last will and testament message shall be retained
LWT QoS	Enum	The last will and testament QoS
Keep Alive	Number	The keep alive message interval in seconds
Clean Session	Boolean	If the session should be cleaned with each successful connect

Each connector should have a meaningful name to be better distinguishable. The connector can connect either with or without encryption (TLS). If the connector shall use TLS to encrypt the communication, a CA certificate can be specified to verify the peer's identity.

After creating the connector, it is shown in the hierarchy as a subnode of the Connectors node.



A connector node has two actions: Remove and Publish. The **Remove** action on the connector removes the complete connector including all subnodes. The **Publish** action is used to publish arbitrary data to a specific topic.

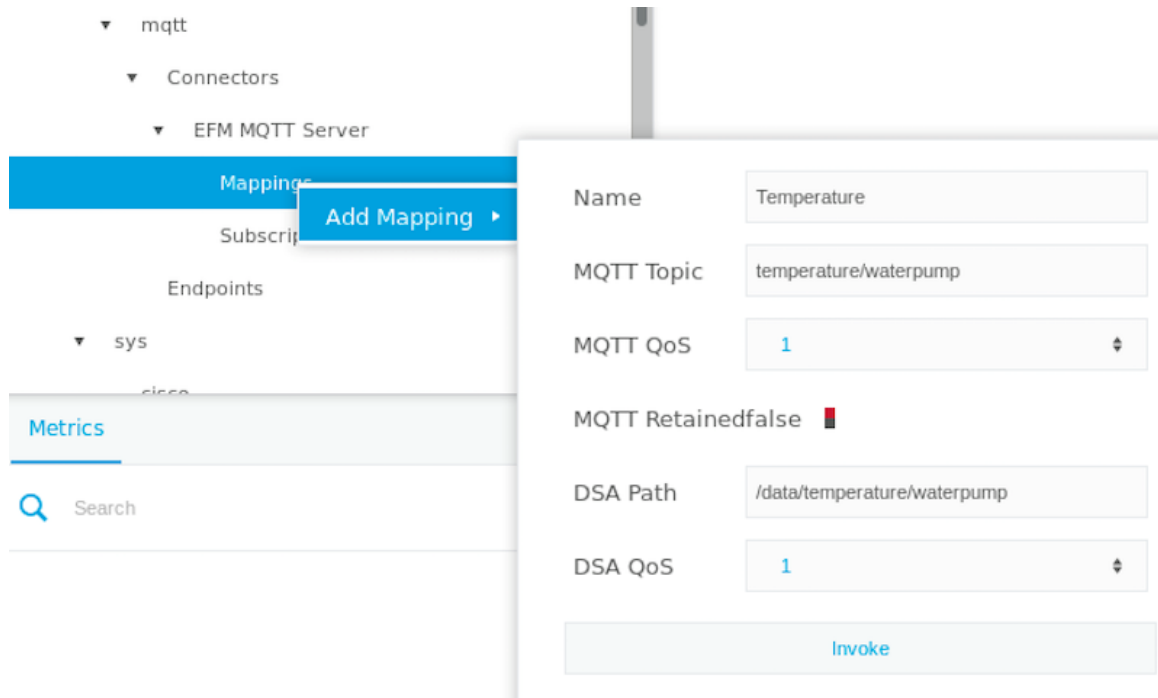


Parameter	Type	Description
Topic	String	The topic to publish the value
Value	String	The value to publish
QoS	Enum	The MQTT QoS to use for the publish
Retained	Boolean	If the message shall be retained

The **Publish** action is a one-shot action. It will just publish the given value once.

Upon creation, the connector will try immediately to open the connection. Each connector has two subnodes: Mappings and Subscriptions. Mappings are used to publish data from DSA paths to the MQTT server. Subscriptions are used to subscribe to MQTT server topics to receive data from the server.

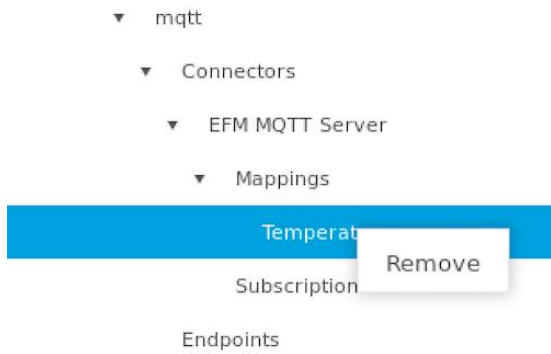
Mappings



A mapping can be added using the **Add Mapping** action.

Parameter	Type	Description
Name	String	Unique name of the mapping
MQTT Topic	String	The MQTT server's topic to publish data to
MQTT QoS	Enum	The MQTT QoS to use for the publishing
MQTT Retained	Boolean	If the message shall be retained in the server's session
DSA Path	String	The DSA path to map
DSA QoS	Enum	The DSA QoS to subscribe to the DSA path

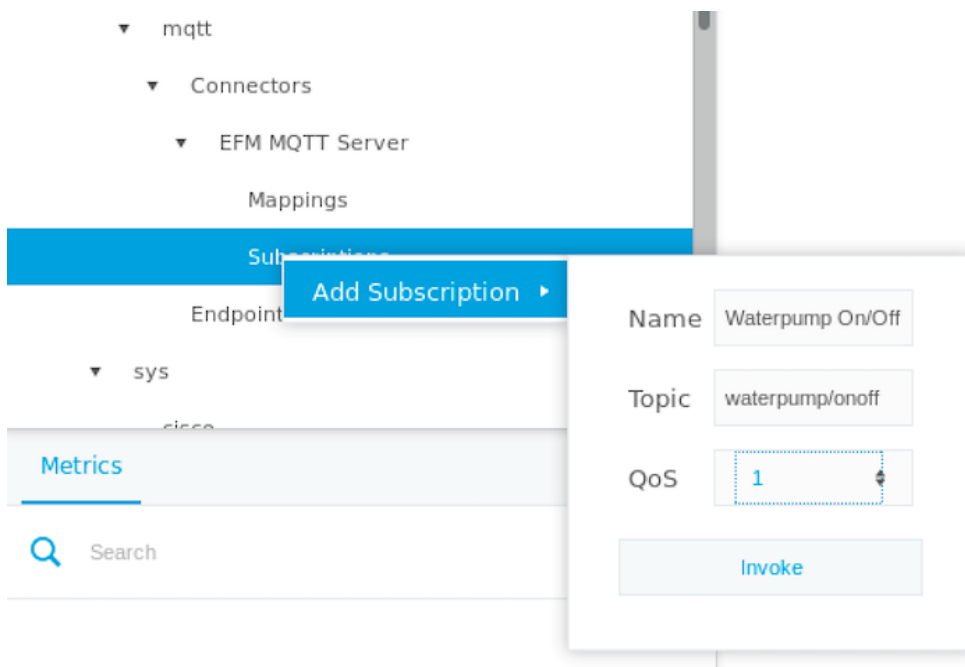
The mapping maps a DSA path to an MQTT topic. Whenever the value of the DSA path is updated, a message is published to the MQTT server for the configured topic. The mapping will be represented by a subnode below the Mappings node.



A mapping can be removed by the **Remove** action on the mapping node.

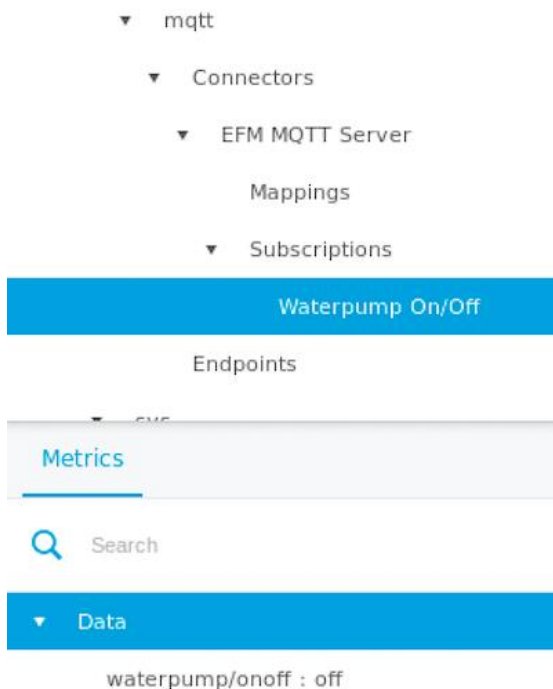
Subscriptions

A subscription can be added to the connector using the **Add Subscription** action.



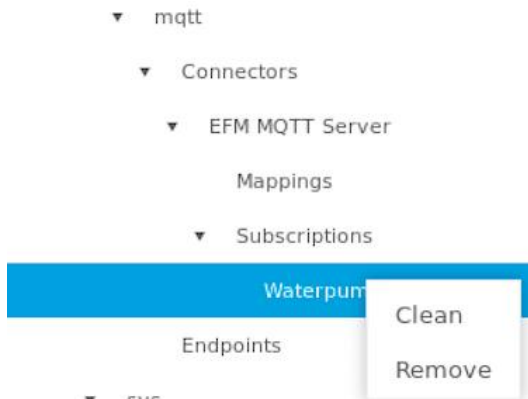
Parameter	Type	Description
Name	String	Unique name of the subscription
Topic	String	The MQTT server's topic to subscribe to
QoS	Enum	The MQTT QoS to use for the subscription

After the creation, the subscription will be represented as a subnode below the Subscriptions node. The subscription will subscribe to the MQTT server using the configured path. Incoming messages will be shown as metrics of the subscription node.



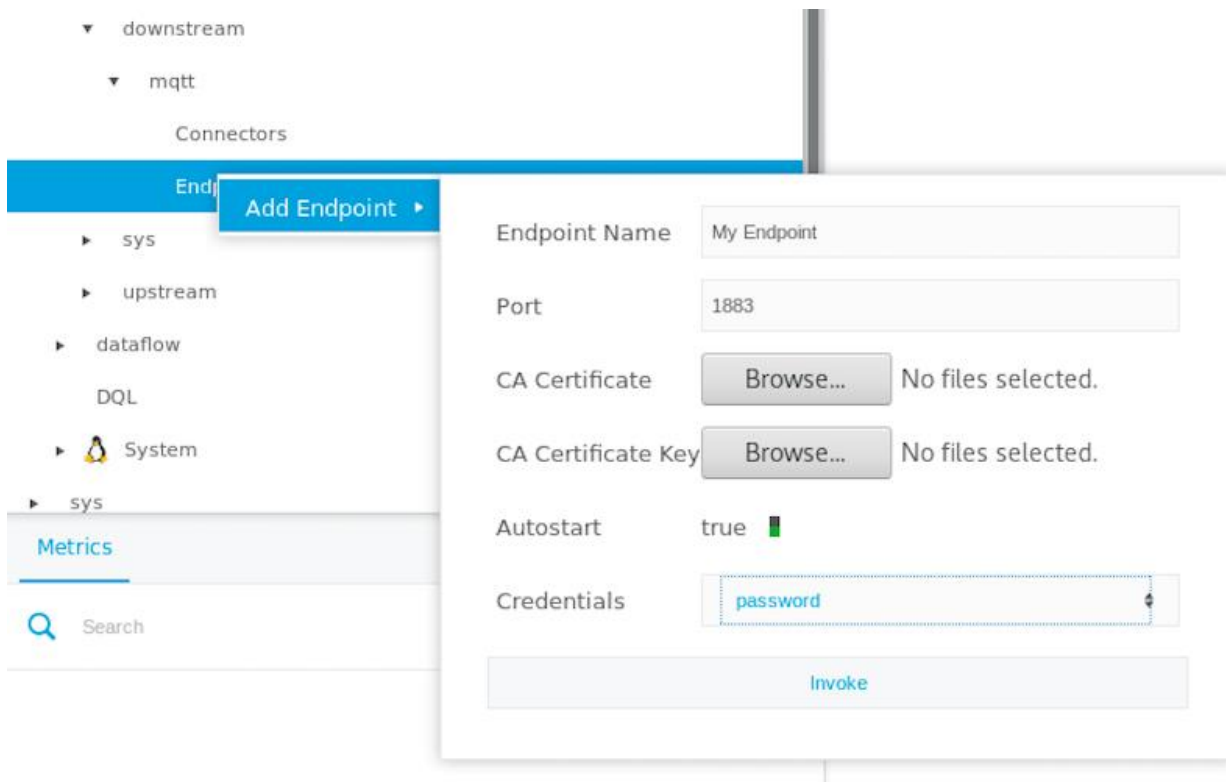
If wildcards are used in the subscription's topic, all incoming messages for every affected topic will be shown as metric. The topic of each message will be shown as own metric.

If the subscription's message metrics shall be cleaned, the **Clean** action has to be invoked on the subscription node. In order to remove a topic, the **Remove** action has to be invoked on the subscription node.



Endpoint Link Actions

Using the **Add Endpoint** action, a new endpoint can be configured and created.



Parameter	Type	Description
Endpoint Name	String	Unique name of the endpoint
Port	Number	The unique port number of the listen port
CA Certificate	Binary	The CA certificate for a tls connection in PEM format. Optional for non tls connections.
CA Certificate Key	Binary	The CA certificate key for a tls connection in PEM format. Optional for non tls connections.
Autostart	Boolean	Set to true if the endpoint shall start automatically
Credentials	Enum	Set to password if the server shall support authentication via username and password. Otherwise no authentication will take place.

The endpoint will enforce encryption via TLS if a CA certificate is set. Otherwise the communication will be unencrypted.

A specific port number can only be used for one endpoint. Specifying the same port number twice will lead to failing endpoint starts, as there cannot be multiple listeners on the same port.

The endpoint will be represented by a node below the Endpoints node. If autostart is set to true, the endpoint will start immediately after being created.

An endpoint can be removed with all its subnodes by invoking the **Remove** action on the endpoint node.

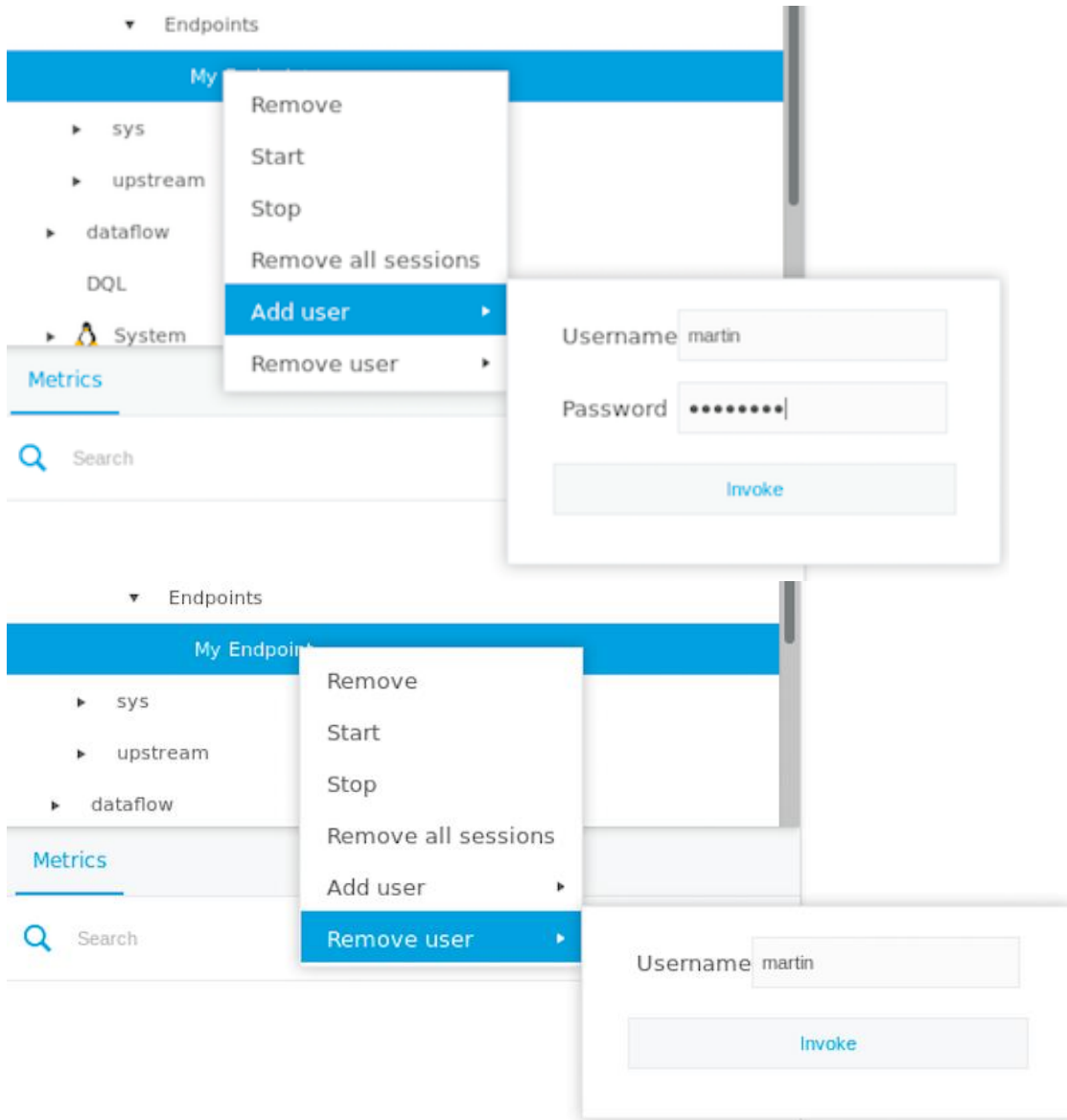
An endpoint can be started or stopped via the **Start** and **Stop** action on the endpoint node.

Authentication

If you need user authentication by the server, you have to select the password enum for the credentials. Users are then added using the **Add User** action on the endpoint node. The action can be called multiple times to add more than one user. Each username has to be unique.

Parameter	Type	Description
Username	String	Unique name of the user
Password	String	The password of the user. Will be hashed before storing it.

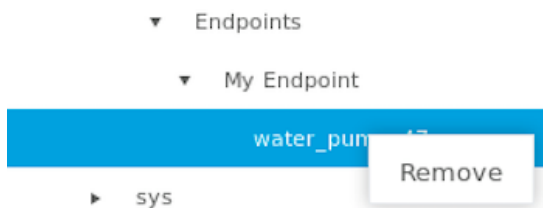
In order to remove a user, invoke the **Remove User** action. You have to specify the username to remove the user. Configured users can be seen in the metrics of the endpoint node.



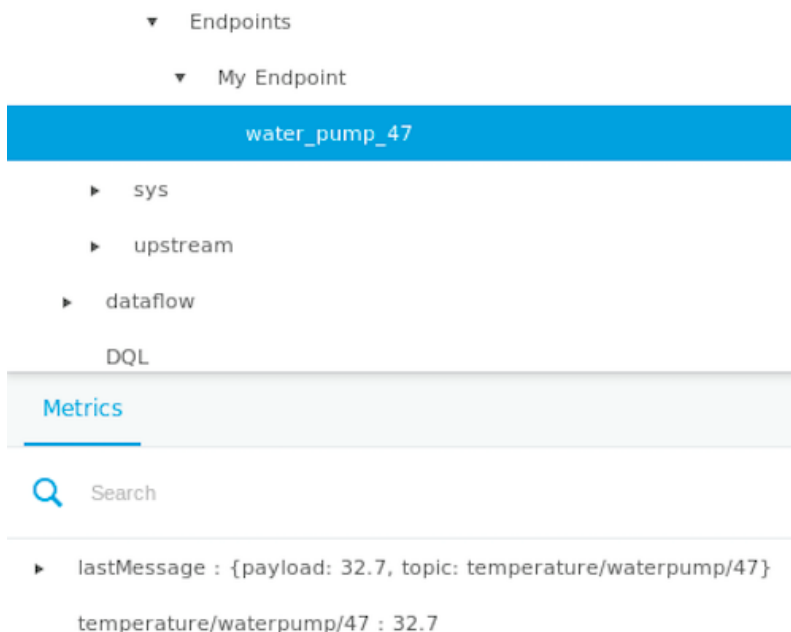
Clients

Whenever a client connects to the endpoint, it will be represented as node below the endpoint node. The name of the node will be the client id of the connecting client. Each client node stands for a session in the server. All current sessions can be forcibly removed by the **Remove All Sessions** action. **All** persistent information associated to the sessions will be **lost**.

A single session can be removed by invoking the **Remove** action on the client node.



Messages sent from the client will be shown as metrics of the client node. They will be shown with the topic as metric name. Only the last message of each topic will be shown on the topic node. Additionally, the last message send for any topic will be stored in the lastMessage metric.



References for securing the EFM operating platform

The EFM Installation guides describe the product configuration options, but the administrators may desire hardening the operating platform that the EFM and its components function.

RedHat: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/pdf/security_guide/Red_Hat_Enterprise_Linux-7-Security_Guide-en-US.pdf

Microsoft Windows: Server Hardening: Windows Server 2012 <https://technet.microsoft.com/en-us/security/jj720323.aspx>

Cisco Guide to Hardening Cisco IOS devices - <https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html>

Obtaining documentation and submitting a service request

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.