BIG-IP® Local Traffic Manager™: Internet of Things Administration

Version 13.1

Table of Contents

Configuring MQTT Functionality

Overview: Creating an MQTT configuration

You can use a Message Queuing Telemetry Transport (MQTT) configuration to optimize the performance and bandwidth of mobile environments. Because the MQTT protocol is designed for lightweight publish-and-subscribe messaging, it reduces or eliminates the disadvantages of the commonly used HTTP request-response protocol, especially in mobile environments. For example, you will want to use an MQTT configuration when devices use intermittent connectivity, when bandwidth is at a premium, when an enterprise application interacts with multiple mobile device applications, or when mobile device applications send data reliably without requiring retries.

In an MQTT configuration, clients publish messages, and the BIG-IP® system validates and manages those messages through a pool of message brokers, which then transport and route the messages to subscribing servers. You can examine statistics specific to MQTT parameters through the Profiles Summary.

A typical BIG-IP MQTT configuration includes:

- MQTT pool of message brokers
- iRules for MQTT
- Client SSL profile
- MQTT profile (configured in TMSH)
- Virtual server configured to use MQTT functionality

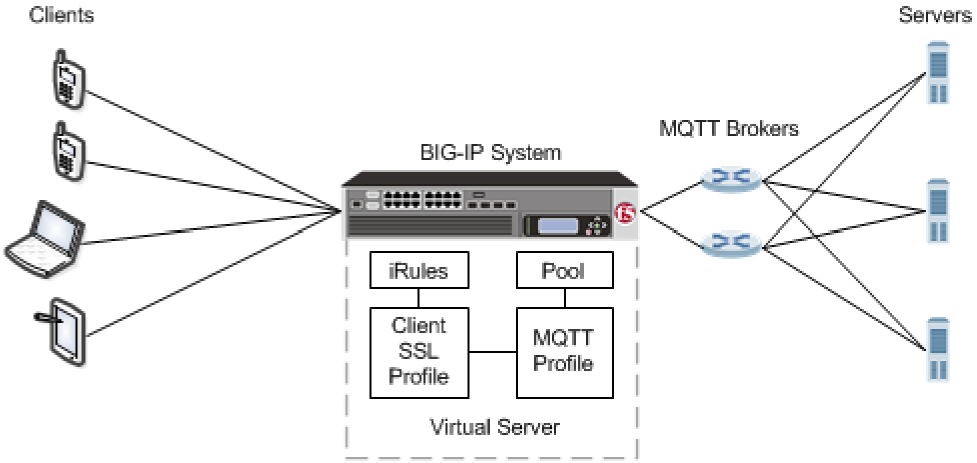

Figure 1: An MQTT configuration

About the MQTT profile

The Message Queuing Telemetry Transport (MQTT) profile supports MQTT protocol functionality, enabling you to configure a publish-and-subscribe environment to manage devices in an Internet of Things (IoT) context. The BIG-IP® system includes a default MQTT profile, configured in TMSH, that you assign to a virtual server.

Example iRule to log MQTT messages

This example iRule shows how to log MQTT messages.

```
ltm rule mqtt_rule {
    when MQTT_CLIENT_INGRESS {
        log local0. "Client message type [MQTT::type]"
        switch [MQTT::type] {
            CONNECT {
                log local0. "   protocol-name    [MQTT::protocol_name]"
                log local0. "   protocol-version [MQTT::protocol_version]"
                log local0. "   client-id        [MQTT::client_id]"
                log local0. "   keep-alive       [MQTT::keep_alive]"
                log local0. "   username         [MQTT::username]"
                log local0. "   password         [MQTT::password]"
            }
            PUBLISH {
                log local0. "   qos              [MQTT::qos]"
                log local0. "   message-id       [MQTT::message_id]"
                log local0. "   topic            [MQTT::topic]"
            }
            PUBREL {
                log local0. "   message-id       [MQTT::message_id]"
            }
            SUBSCRIBE {
                log local0. "   message-id       [MQTT::message_id]"
                set count [MQTT::topic count]
                for {set i 0} {$i < $count} {incr i} {
                    set topic [MQTT::topic index $i]
                    log local0. "   topics index $i      $topic"
                    log local0. "   topics index $i qos  [MQTT::message topics qos $topic]"
                }
            }
            UNSUBSCRIBE {
                log local0. "   message-id       [MQTT::message_id]"
                set count [MQTT::topic count]
                for {set i 0} {$i < $count} {incr i} {
                    log local0. "   topics index $i   [MQTT::topic index $i]"
                }
            }
        }
    }
    when MQTT_SERVER_INGRESS {
        log local0. "Server message type [MQTT::type]"
        switch [MQTT::type] {
            CONNACK {
                log local0. "   return-code      [MQTT::return_code]"
            }
            PUBLISH {
                log local0. "   message-id       [MQTT::message_id]"
                log local0. "   topic            [MQTT::topic]"
            }
            PUBACK {
                log local0. "   message-id       [MQTT::message_id]"
            }
            PUBREC {
                log local0. "   message-id       [MQTT::message_id]"
            }
            PUBCOMP {
                log local0. "   message-id       [MQTT::message_id]"
            }
            SUBACK {
                log local0. "   message-id       [MQTT::message_id]"
            }
            UNSUBACK {
                log local0. "   message-id       [MQTT::message_id]"
            }
        }
    }
}
```

Example iRule to pass client certificate common name

This example iRule shows how to pass the common name for a client certificate to an MQTT server through the username field in the CONNECT message.

```
when CLIENT_ACCEPTED {
    set cn ""
}

when CLIENTSSL_CLIENTCERT {
    set cn [ lindex [ split [lindex [ split [X509::subject [SSL::cert 0]] "," ] 0 ] "=" ]
1 ]
    log local0. "Client Cert Common Name : $cn"
}

when MQTT_CLIENT_INGRESS {
    if {[MQTT::type] == "CONNECT"} {
        if {$cn == ""} {
            # if we didn't see a client cert, return an authentication error

            MQTT::drop                                    # drop current message
            MQTT::respond type CONNACK return-code 5      # send a CONNACK
            MQTT::disconnect                              # and disconnect
        } else {
            MQTT::username $cn                    # fill-in username field
        }
    }
 }
```

Task Summary

Complete these tasks to configure the BIG-IP system to use MQTT functionality.

Task list

Creating an MQTT monitor

Creating a pool

Creating an iRule for MQTT publishing

Creating a Client SSL profile

Creating a virtual server

Viewing MQTT statistics

Creating an MQTT monitor

You can create an MQTT monitor to monitor MQTT brokers that are configured as pool members in an LTM® pool.

1. On the Main tab, click **Local Traffic** > **Monitors**.
   The Monitors List screen opens.
2. Click **Create**.
   The New Monitor screen opens.
3. In the **Name** field, type a name for the monitor.
4. From the **Type** list, select **MQTT**.
   The screen refreshes, and displays the configuration options for the **MQTT** monitor type.
5. From the **Configuration** list, select **Advanced**.

   This selection makes it possible for you to modify additional default settings.

6. (Optional) In the **Interval** field, type a number that indicates, in seconds, how frequently the system issues the monitor check.

   The default is `5` seconds.
7. (Optional) To specify a different interval for health checking, from the **Up Interval** list, select **Enabled**, and, in the **Up Interval** field, type the number of seconds for the interval.

   The default is **Disabled**.
8. (Optional) In the **Time Until Up** field, type the number of seconds to wait after a resource first responds correctly to the monitor before setting the resource to up.

   The default is `0` seconds.
9. (Optional) In the **Timeout** field, type the number of seconds that the target has in which to respond to the monitor request.

   The default is `16` seconds.
10. (Optional) To specify that you must manually re-enable the resource after an unsuccessful monitor check, for **Manual Resume**, select **Yes**.

    The default is **No**.
11. (Optional) From the **MQTT Version** list, select the protocol version that the monitor will use to communicate with the monitoring object.

    The default is **3.1.1**.
12. (Optional) In the **Client ID** field, type the Client ID that the monitor will send to communicate with the monitoring object.
13. (Optional) If the monitored object requires authentication, type a **User Name**.
14. (Optional) If the monitored target requires authentication, type a **Password**.
15. (Optional) In the **Alias Address** field, type an alias IP address for the monitor to check, on behalf of the pools or pool members with which the monitor is associated.

    The default is `* All Addresses`.
16. (Optional) In the **Alias Service Port** field, type an alias port or, from the **Alias Service Port** list, select a service for the monitor to check, on behalf of the pools or pool members with which the monitor is associated.

    The default is `* All Ports`.
17. Click **Finished**.

Creating a pool

Before you can assign an MQTT health monitor to a pool, you need to create the MQTT monitor.

You can create a pool of servers that you can group together to receive and process traffic. After the pool is created, you can associate the pool with a virtual server.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. (Optional) For the **Health Monitors** setting, select an MQTT health monitor from the **Available** field, and move it to the **Active** field.
5. For each pool member, in the **New Members** setting, select one of the options, and then follow the steps to configure the applicable settings.

<table>
<tr><th>Option</th><th>Steps</th></tr>
<tr><td><b>New Node</b></td><td>
<ol>
<li>In the <b>Node Name</b> field, type a name for the node portion of the pool member.</li>
<li>In the <b>Address</b> field, type the IP address of the server.</li>
<li>For the <b>Service Port</b> setting, type one of the following port numbers, or select one of the following services from the list.
<table>
<tr><th>Port Number</th><th>Service Name</th></tr>
<tr><td><code>1883</code></td><td><b>MQTT</b>. The Internet Assigned Numbers Authority (IANA) registered port for MQTT service.</td></tr>
<tr><td><code>8883</code></td><td><b>MQTT-TLS</b>. The IANA registered port for secure MQTT service over a Transport Layer Security (TLS) network.</td></tr>
</table></li>
<li>Click <b>Add</b>.</li>
</ol>
</td></tr>
<tr><td><b>New FQDN Node</b></td><td>
<ol>
<li>In the <b>Node Name</b> field, type a name for the node portion of the pool member.</li>
<li>In the <b>FQDN</b> field, type the FQDN of the server.
<p><i><b>Note:</b> To use FQDNs instead of IP addresses, you should still type at least one IP address. Typing one IP address ensures that the system can find a pool member if a DNS server is not available.</i></p></li>
<li>For the <b>Service Port</b> setting, type one of the following port numbers, or select one of the following services from the list.
<table>
<tr><th>Port Number</th><th>Service Name</th></tr>
<tr><td><code>1883</code></td><td><b>MQTT</b>. The Internet Assigned Numbers Authority (IANA) registered port for MQTT service.</td></tr>
<tr><td><code>8883</code></td><td><b>MQTT-TLS</b>. The IANA registered port for secure MQTT service over a Transport Layer Security (TLS) network.</td></tr>
</table></li>
<li>From the <b>Auto Populate</b> list, select <b>Enabled</b> to automatically create ephemeral nodes, using the IP addresses returned by the resolution of a DNS query for the pool member defined by the FQDN.</li>
<li>Click <b>Add</b>.</li>
</ol>
</td></tr>
</table>

6. Click **Finished**.
   The screen refreshes, and you see the new pool in the Pool list.

Creating an iRule for MQTT publishing

You can create iRules for MQTT functionality, for example to log the messages that the BIG-IP system passes, or to pass the client certificate's common name in the CONNECT message.

1. On the Main tab, click **Local Traffic** > **iRules**.
   The iRule List screen opens, displaying any existing iRules.
2. Click **Create**.
   The New iRule screen opens.
3. In the **Name** field, type a unique name for the iRule.

   The full path name of the iRule cannot exceed 255 characters.

4. In the **Definition** field, type an iRule.
5. Click **Finished**.
   The new iRule appears in the list of iRules on the system.

The BIG-IP system includes the iRules for MQTT functionality that you've created.

Creating a Client SSL profile

You create a Client SSL profile when you want the BIG-IP® system to authenticate and decrypt/encrypt client-side application traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client SSL profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. Configure all profile settings as needed.
4. Click **Finished**.

After creating the Client SSL profile and assigning the profile to a virtual server, the BIG-IP system can apply SSL security to the type of application traffic for which the virtual server is configured to listen.

Creating a virtual server

Before creating a virtual server, verify that you have created the pool to which you want this virtual server to send traffic.

When you create a virtual server, you specify a destination IP address and service port. All other settings on the virtual server have default values. You can change the default values of any settings to suit your needs.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *The IP address for this field needs to be on the same subnet as the external self-IP address.*

5. In the **Service Port** field, type one of the following port numbers, or select one of the following services from the list.

Port Number	Service Name
`1883`	**MQTT**. The Internet Assigned Numbers Authority (IANA) registered port for MQTT service.
`8883`	**MQTT-TLS**. The IANA registered port for secure MQTT service over a Transport Layer Security (TLS) network.

6. From the **Configuration** list, select **Advanced**.

7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created and move the name to the **Selected** list.
8. Select the **MQTT** check box.
9. Configure any other settings that you need.
10. In the Resources area, for the **iRules** setting, from the **Available** list, select the name of the iRule that you want to assign, and move the name into the **Enabled** list.
11. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
12. Click **Finished**.

After performing this task, you have a virtual server that listens for application traffic and acts according to the values configured within the virtual server.

Viewing MQTT statistics

Ensure that an MQTT profile is assigned to at least one virtual server.

You can see how the BIG-IP® system is handling MQTT messages by viewing statistics per MQTT profile.

1. On the Main tab, click **Statistics** > **Module Statistics** > **Local Traffic**.
   The Local Traffic statistics screen opens.
2. From the **Statistics Type** list, select **Profiles Summary**.
3. In the Details column for the MQTT profile, click **View** to display detailed statistics about MQTT messages.

Legal Notices

Legal notices

Publication Date

This document was published on November 15, 2017.

Publication Number

MAN-0654-01

Copyright

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

Trademarks

For a current list of F5 trademarks and service marks, see *http://www.f5.com/about/guidelines-policies/trademarks*.

All other product and company names herein may be trademarks of their respective owners.

Patents

This product may be protected by one or more patents indicated at: *https://f5.com/about-us/policies/patents*.

Link Controller Availability

This product is not currently available in the U.S.

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Index