

IBM Integration Bus

Modifying MQTT nodes using Node Policy

Featuring:

Node Policy
MQTT
IBM MessageSight

January 2016

Hands-on lab built at product
Version 10.0.0.3

1. INTRODUCTION	3
1.1 MQTT SUPPORT IN IIB V10.....	3
1.2 NODE POLICY SUPPORT IN IIB V10	3
1.3 LAB GUIDE OVERVIEW	4
1.4 USE OF IBM WEBSphere MQ THROUGHOUT THIS LAB GUIDE	5
1.5 A NOTE ON IBM MESSAGEsIGHT TERMINOLOGY	5
2. LAB PREPARATION	6
2.1 FINDING YOUR IBM MESSAGEsIGHT APPLIANCE	6
3. MQ TELEMETRY SET UP	10
3.1 SET UP THE DEFAULT MQTT CONFIGURATION	10
3.2 TEST THE DEFAULT CONFIGURATION WITH A BASIC TEST	13
3.3 CONFIGURE A BASIC MQTT APPLICATION IN IIB	15
3.4 TEST THE TTPOLICYLAB APPLICATION.....	17
3.5 IBM MESSAGEsIGHT CONFIGURATION	22
3.6 TEST THE SIMPLE IBM MESSAGEsIGHT CONFIGURATION	29
4. MODIFYING THE IIB MQTT CONFIGURATION USING NODE POLICY	32
4.1 GENERATE AND STORE A POLICY.....	32
4.2 MODIFY THE POLICY IN THE INTEGRATION REGISTRY	35
4.3 TESTING THE RESULT OF A POLICY CHANGE	37
END OF LAB GUIDE	39

1. Introduction

This lab guide provides a “hands on” introduction to the IIB support of MQTT and how the IIB V10 Node Policy support can provide a useful way of changing connection specific configuration related to an MQTT node without the need to redeploy a message flow.

In addition, it showcases two MQTT server technologies:

- the one provided with **IBM WebSphere MQ**
- **IBM MessageSight**. This is an appliance designed for the Internet of Things (IoT) and mobile environments. It provides a secure, DMZ-ready channel for lightweight, rapid, bi-directional messaging.

1.1 MQTT support in IIB V10

MQ Telemetry Transport (MQTT) is a lightweight publish/subscribe messaging protocol. IBM Integration Bus provides built-in input and output nodes for processing MQTT messages.

The MQTT messaging protocol provides robust messaging features for communicating with remote systems and devices, and also minimizes network bandwidth and device resource requirements.

The protocol is designed for devices in constrained environments, such as embedded systems, cell phones, and sensors with limited processing ability and memory, and for systems that are connected to unreliable networks.

MQTT uses the publish/subscribe style of messaging that enables the information provider (publisher) to be decoupled from the consumer of the information (subscriber). Consequently, the MQTT protocol is ideal for the machine-to-machine, or Internet of Things, world of communication.

IBM Integration Bus V10 provides support for MQTT through two MQTT nodes:

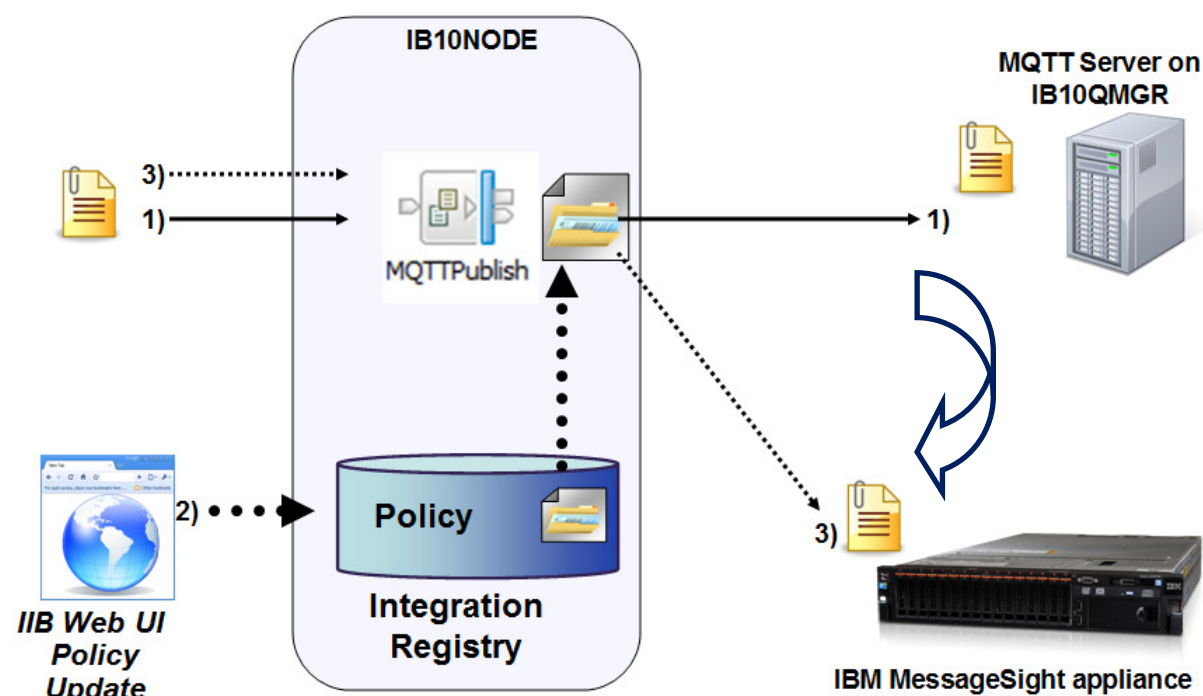
- **MQTTPublish:**
Provides an (outbound) interface for publishing data to MQTT endpoints (topic strings)
- **MQTTSubscribe:**
Provides an (inbound) subscription interface to obtain data from MQTT endpoints.

1.2 Node Policy support in IIB V10

The node policy support provided through IIB V10 allows for connection specific data applicable to nodes to be changed in real time without the need for development changes and redeployment of an application or Integration Service. The Node Policy can be generated from within Integration Toolkit.

1.3 Lab Guide overview

This lab guide provides an outline of how Node Policy support in IIB V10 can be used to change the connection specific data in MQTT nodes.



You will:

- Learn how to easily configure sample MQ Telemetry support in **WebSphere MQ**
- Learn how to configure an **IBM MessageSight** Appliance so that remote clients can publish and subscribe to the appliance.
- Create a simple message flow demonstrating how the IIB MQTT Publish node can be used.
- Generate an MQTT policy, and store it in the IIB Integration Registry (**2) in the above diagram**)
- Test how changes in the definition of a policy can affect message flows without the need to redeploy the flow. In **1) above**, MQTT data is first published to **WebSphere MQ**. After a change in policy, MQTT data is then published to **IBM MessageSight (3) above**).

1.4 Use of IBM WebSphere MQ throughout this lab guide

You will use the MQTT support in IBM WebSphere MQ as an example MQTT server. You will also use IBM WebSphere MQ to initiate the simple message flow you will create.

IIB V10 support of MQTT does not however have a pre-requisite to use IBM WebSphere MQ. Any MQTT Server can be used.

1.5 A note on IBM MessageSight terminology

In this lab guide you will use an IBM MessageSight appliance for the MQTT clients to communicate with. Full guidance is provided within the lab guide on how to configure the IBM MessageSight appliance. The following section gives a brief overview of the terminology that you will see when using the appliance:

Endpoints: are the entry points for clients to connect to the appliance on one or all configured ethernet interfaces, and a specific Port (the value of which can range between 1 and 65535). They are created on a message Hub, and must have at least one connection policy and one messaging policy defined/attached in order to work.

Connection policies: define how (and which type of) clients connect to the appliance for example using JMS, MQTT or both.

Messaging policies: define what clients can do once connected, which resources to use and how to use them. For example if they are able to access topics, or queues.

Message Hubs: are a method of organizing endpoints, connection policies and messaging policies. The ability to group endpoints and policies is intended to make management and monitoring of them easier.

2. Lab preparation

To run this lab you will need access to an IBM MessageSight appliance. On the June 2014 classroom machines a VMware copy of the IBM MessageSight appliance has been made available on your machines so each Lab machine will have its own private copy of the device.

2.1 Finding your IBM MessageSight appliance

This section is only relevant if you are following this lab guide using the supplied VMware machines for the IBM Integration Bus V10 beta class. If you are connecting to a remote IBM MessageSight appliance, you should already know the IP address of your appliance, continue to the next section.

The supplied VMware machines are designed for a classroom environment and configured to use DHCP. You can find the IP address of the IBM MessageSight appliance in your environment using the following steps:

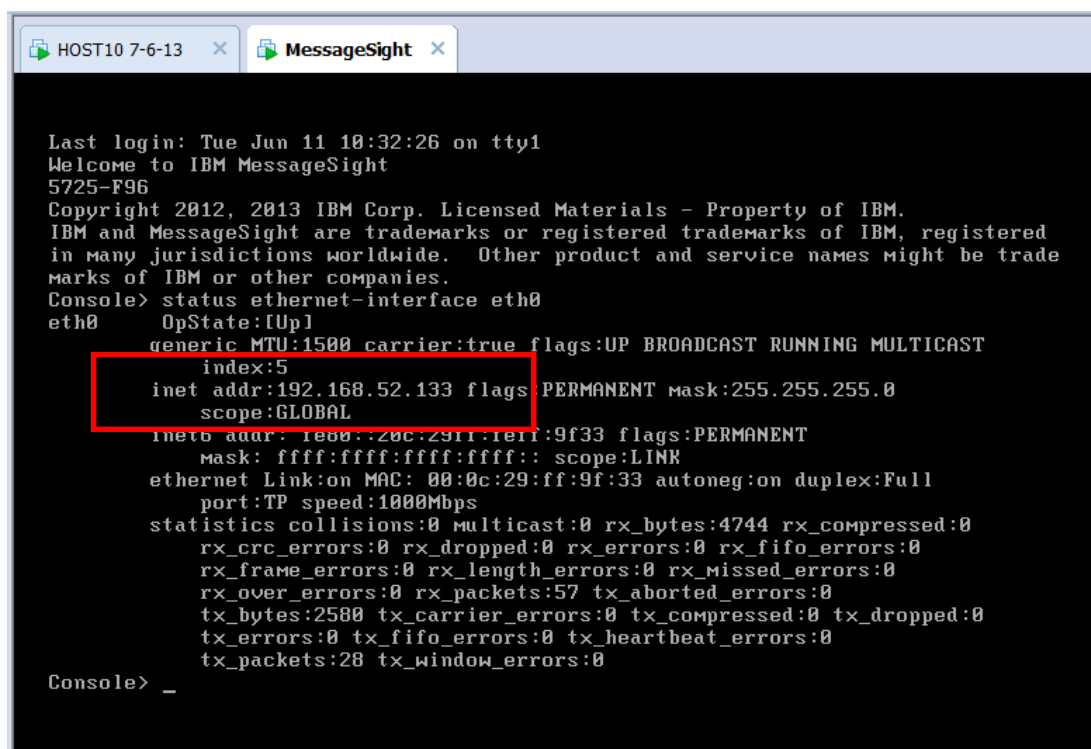
1. In VMware workstation, the IBM MessageSight appliance will be started (The tab will say "IBM MessageSight"), select this tab and click the middle of the black background.

If you are prompted to sign on, use "admin" with a password of "admin" to sign on.

At the **Console>** prompt, type:

```
status ethernet-interface eth0
```

The ip address you will need later in this lab is "inet addr:" in the following:



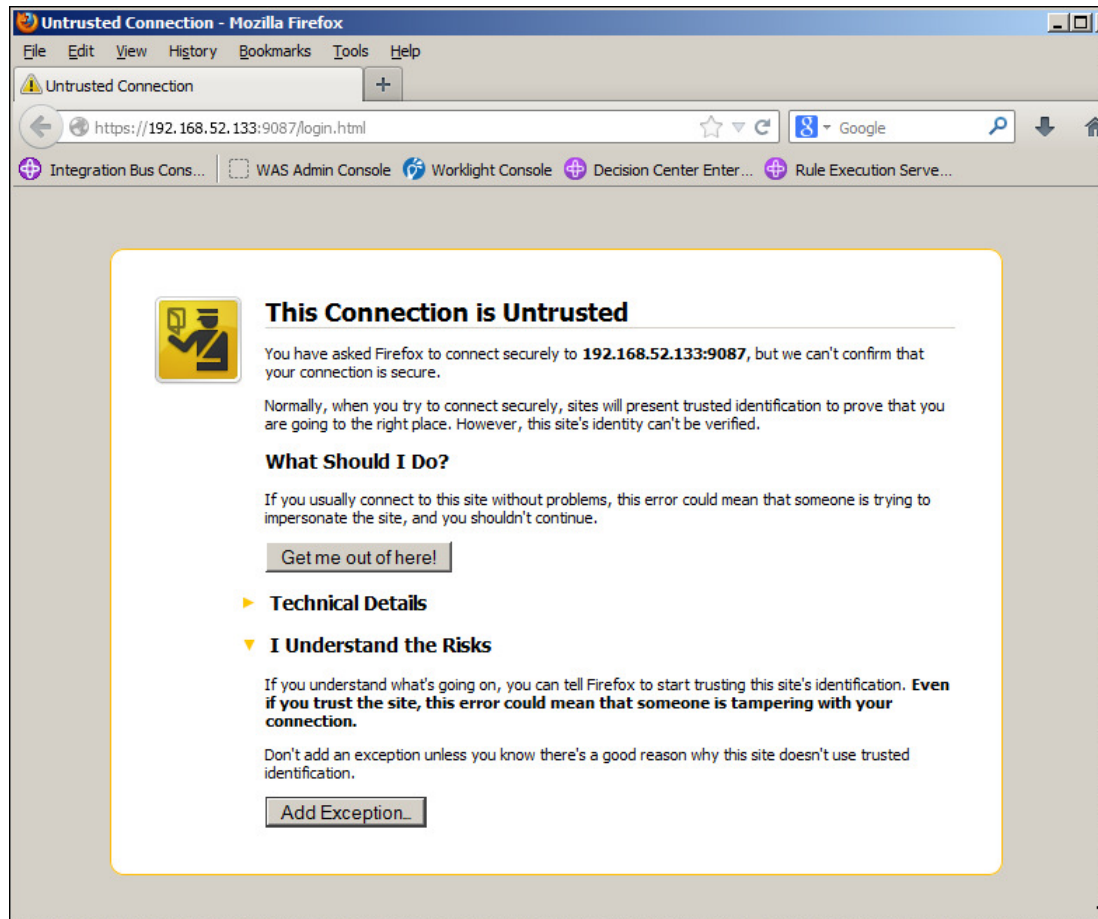
```
HOST10 7-6-13 x MessageSight x
Last login: Tue Jun 11 10:32:26 on tty1
Welcome to IBM MessageSight
5725-F96
Copyright 2012, 2013 IBM Corp. Licensed Materials - Property of IBM.
IBM and MessageSight are trademarks or registered trademarks of IBM, registered
in many jurisdictions worldwide. Other product and service names might be trade
marks of IBM or other companies.
Console> status ethernet-interface eth0
eth0    OpState:[Up]
generic MTU:1500 carrier:true flags:UP BROADCAST RUNNING MULTICAST
        index:5
        inet addr:192.168.52.133 flags:PERMANENT mask:255.255.255.0
        scope:GLOBAL
        inet6 addr: fe80::20c:29ff:fe9f:9f33 flags:PERMANENT
        mask: ffff:ffff:ffff:ffff:: scope:LINK
        ethernet Link:on MAC: 00:0c:29:ff:9f:33 autoneg:on duplex:Full
        port:TP speed:1000Mbps
        statistics collisions:0 multicast:0 rx_bytes:4744 rx_compressed:0
        rx_crc_errors:0 rx_dropped:0 rx_errors:0 rx_fifo_errors:0
        rx_frame_errors:0 rx_length_errors:0 rx_missed_errors:0
        rx_over_errors:0 rx_packets:57 tx_aborted_errors:0
        tx_bytes:2580 tx_carrier_errors:0 tx_compressed:0 tx_dropped:0
        tx_errors:0 tx_fifo_errors:0 tx_heartbeat_errors:0
        tx_packets:28 tx_window_errors:0
Console> _
```

2. Go back to the IIB V10 VMware machine and type the following in a Firefox browser (remember to replace your specific IP address if it is different and note its https):

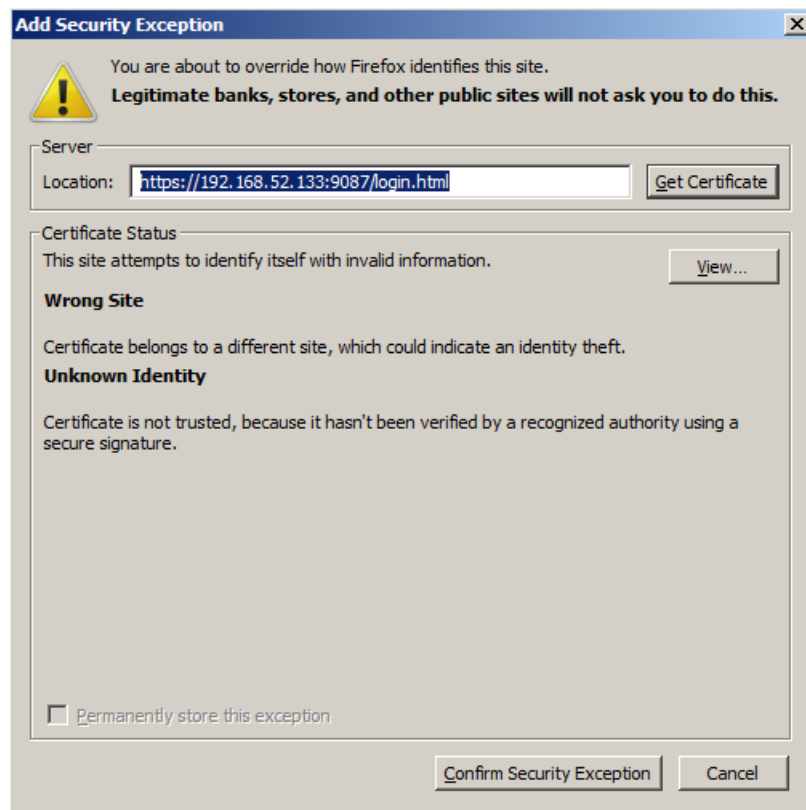
https://192.168.52.133:9087

NB before you can switch from the IBM MessageSight appliance VM, you may need to release the cursor by pressing <ctrl> and <Alt> keys simultaneously.

3. If you are prompted with the following, choose the option "I understand the Risks" and click the "Add Exception" button:



4. Click "Confirm Security Exception" on the "add security window":

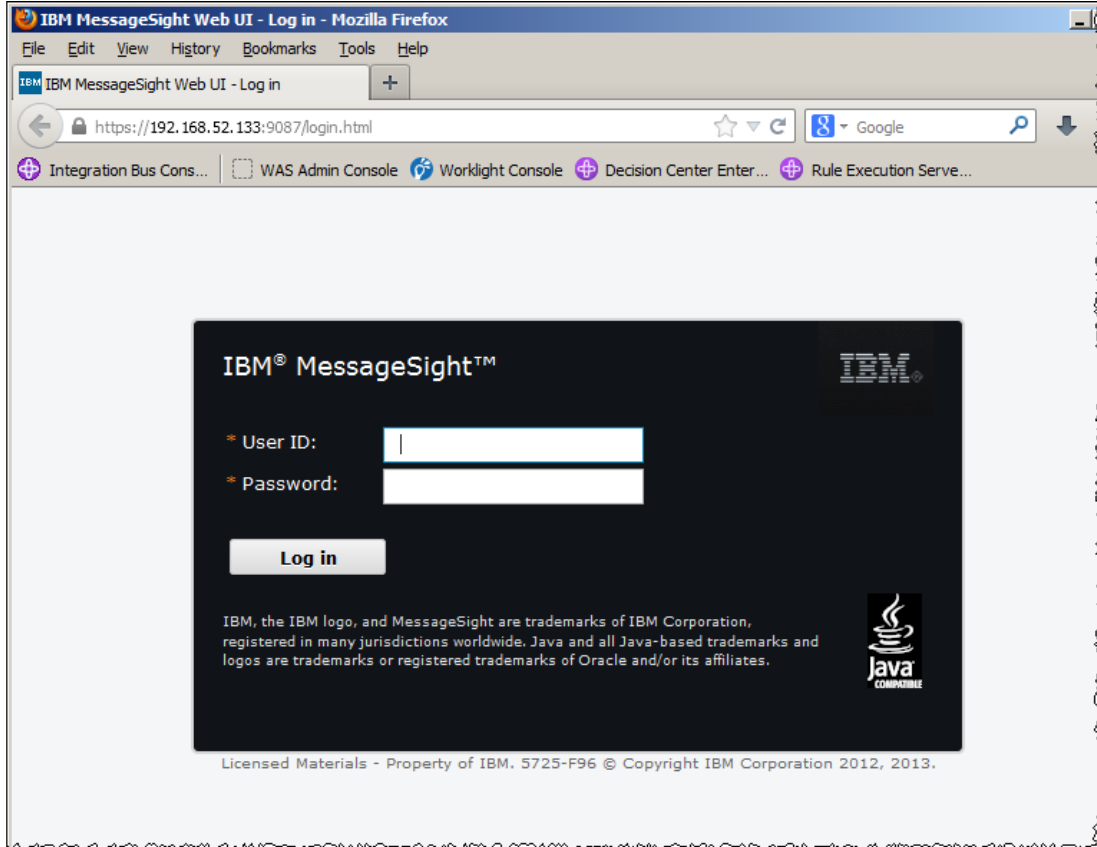


- 5. When you see the following sign-on screen, you have confirmed the IBM MessageSight appliance IP address, book mark the link and note the IP address here:

* * * * *

IBM MessageSight IP address: < _____ >

* * * * *



3. MQ Telemetry set up

IIB V10 support of MQTT does not pre-req using IBM WebSphere MQ as an MQTT server. The MQTT server in IBM WebSphere MQ gives a convenient and quick way of providing an MQTT environment. You will now configure the MQ Telemetry feature in IBM WebSphere MQ. We will use it as an example of an MQTT server for the IIB MQTT Publish node to communicate with.

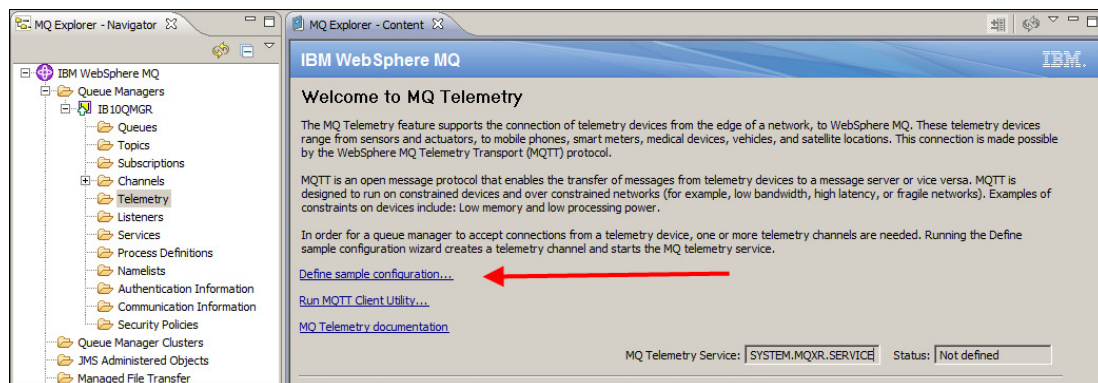
The MQ Telemetry feature in WebSphere MQ is enabled at a queue manager level. In this section you will use the MQ Explorer to enable MQ Telemetry and configure the MQTT Client Utility to communicate with the queue manager.

3.1 Set up the default MQTT configuration

WebSphere MQ provides a sample MQTT configuration through the WebSphere MQ Explorer. You will now define this sample configuration:

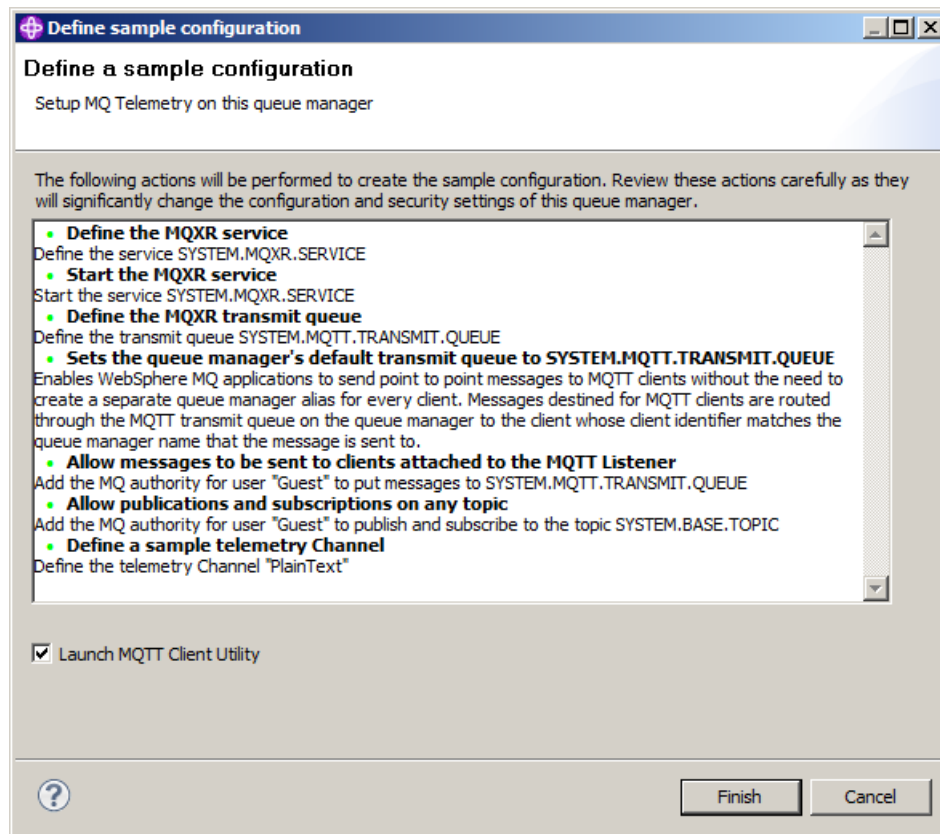
1. Open the WebSphere MQ Explorer.

Expand the <Queue Managers><IB10QMGR> and click on the Telemetry Folder:

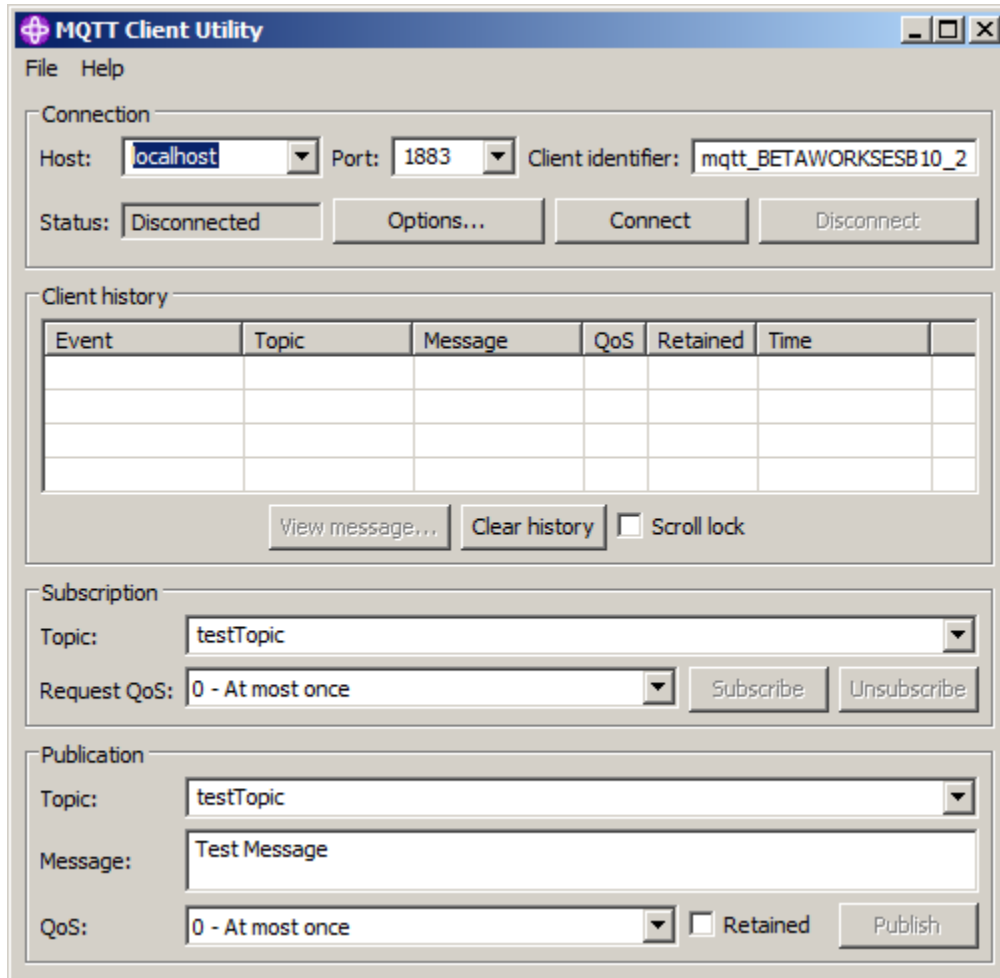


Click on “Define sample configuration”

- Review what the sample configuration will define and click Finish:



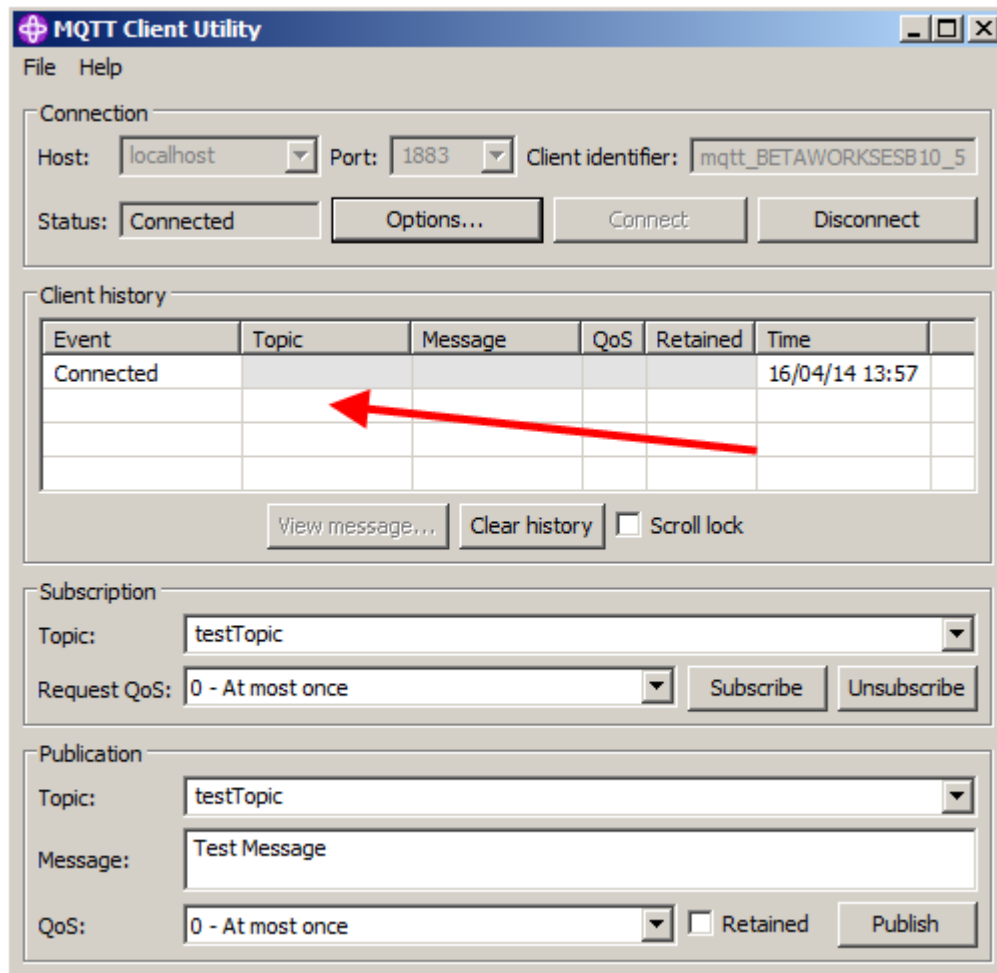
- When the configuration wizard has finished, the Status of the MQ Telemetry Service will change to "Started" and the MQTT Client Utility will open:



3.2 Test the default configuration with a basic test

The MQ TT Client Utility allows you to publish and subscribe to an MQTT server. You will now use this simple utility to test the default configuration.

1. Connect to the Queue manager by clicking the “Connect” button in the Utility. The Client Utility history table will be updated:



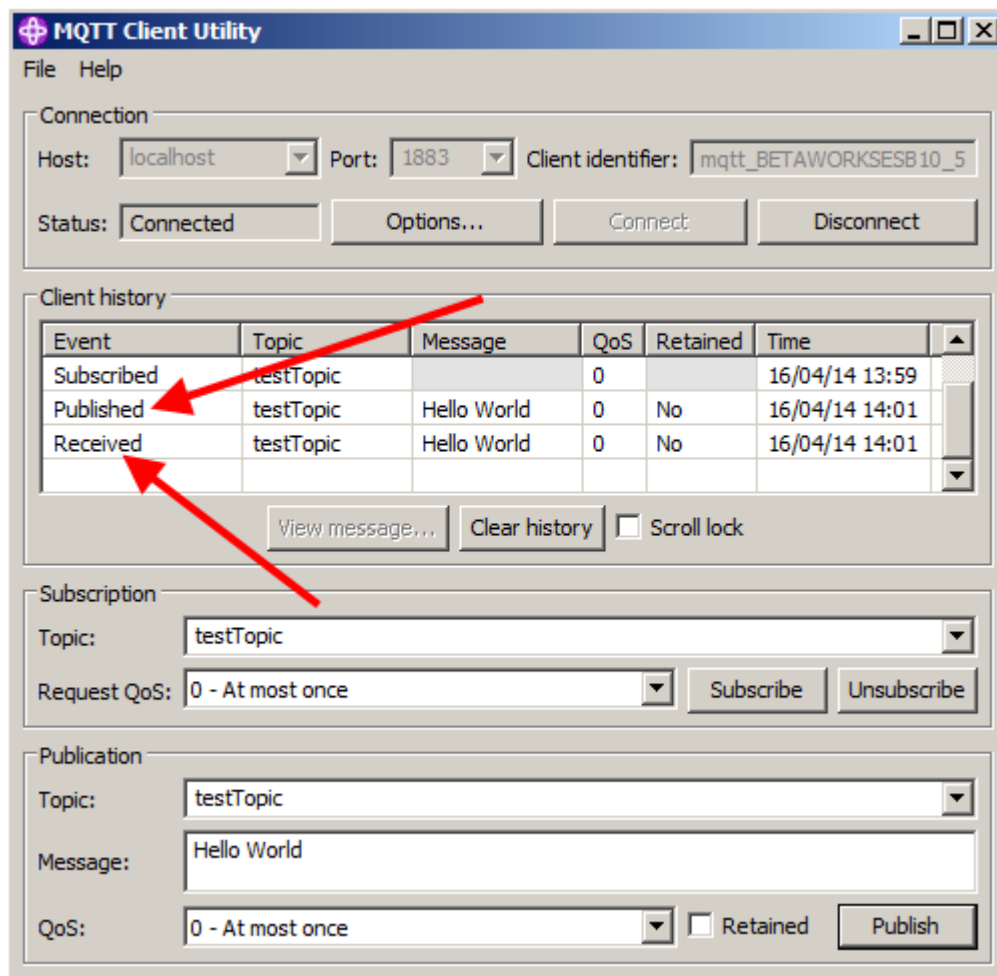
Note: the Port number that is shown the first time you open the MQTT Client Utility is correct for the latest set up in your environment. If you close the Client Utility and open it again, it will default to port 1883. You can check the default configuration port for your environment by looking for the Telemetry Channel called “PlainText” in <QMGrName><Telemetry><Channels>.

2. Click the “Subscribe” button (leave the Topic as “testTopic”)

Note the Client Utility history is updated with the status (subscribed).

- Write "Hello World" in the message box in the Publication section and click "Publish".

The Client Utility history is updated with the "Publish" and what was received as part of the "subscribe":



- The MQTT Client Utility and the queue manager is now ready to receive publications from IBM Integration Bus.

3.3 Configure a basic MQTT application in IIB

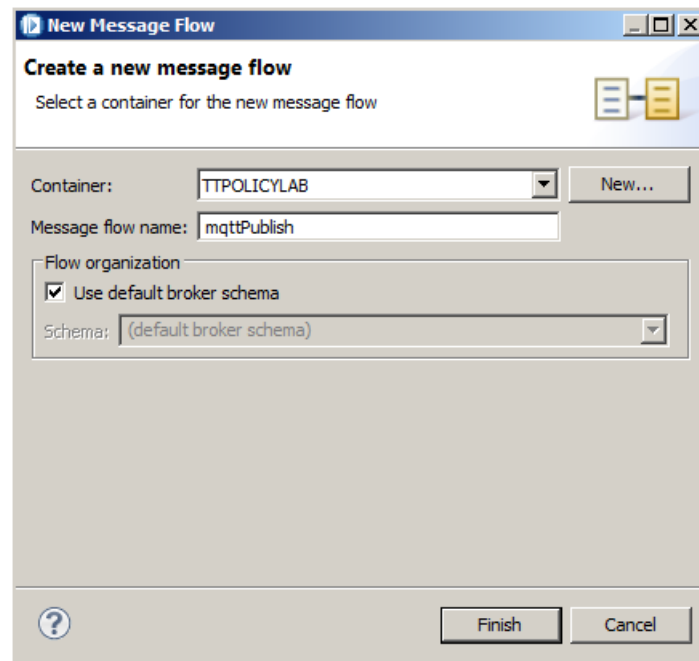
You will now use IBM Integration Toolkit to define a simple application to publish data to “testTopic”.

1. Open the IBM Integration Toolkit.

Create a new workspace called MQTTPolicy (use *File > switch workspace > other*)

Create an Application called “TTPOLICYLAB”.

2. Create a flow called “mqttPublish”:



3. Add an MQ Input node with queue name "MQTT.IN".

On the MQ Connection tab, set the Destination Queue Manager Name to IB10QMGR.

Add an MQTT Publish node with the following properties:

Client ID: **FromMQTT.IN**

Topic name: **testTopic**

Host name: **localhost**

Port: **1883** (note this may be different *in your environment*, you can check the default configuration port for your environment by looking for the Telemetry Channel called "PlainText" in <QMgrName><Telemetry><Channels>)

Wire the Output terminal of the MQ input node to the Input terminal of the MQTT Publish node:

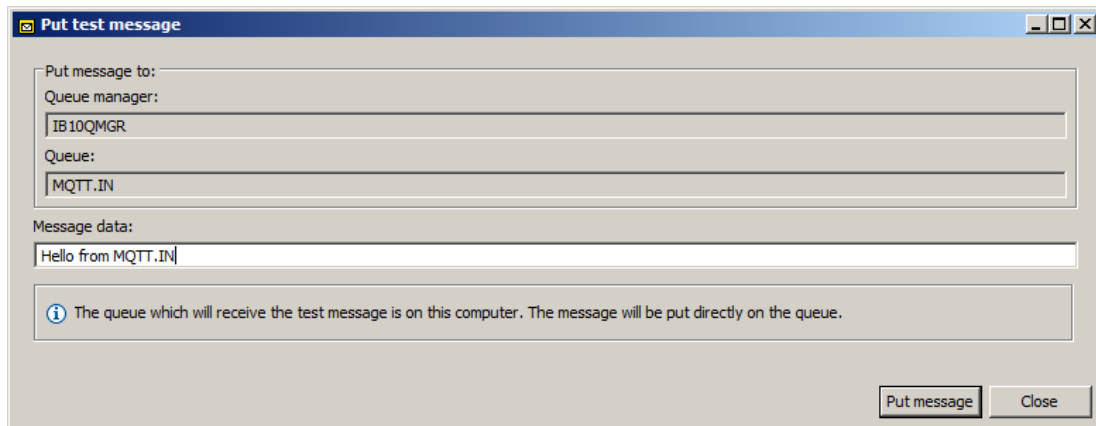
The screenshot displays the IBM Integration Bus Flow Designer interface. The top pane shows a flow graph with two nodes: 'MQ Input' and 'MQTTPublish'. An arrow connects the output terminal of the 'MQ Input' node to the input terminal of the 'MQTTPublish' node. The bottom pane shows the 'MQTTPublish Node Properties' dialog box, which is currently open to the 'User Defined Properties' tab. The dialog has several sections: 'Description', 'Basic', 'Validation', 'Policy', and 'Monitoring'. The 'Basic' section contains the following properties and values:

Section	Property	Value
Basic	Client ID*	FromMQTT.IN
	Topic name*	testTopic
Validation	Host name*	localhost
	Port*	1883
Policy	Quality of service*	0 - At most once
	Security identity	

4. Save the mqttPublish message flow.
5. Deploy the TTPOLICYLAB Application to the default server on the TESTNODE_iibuser node.

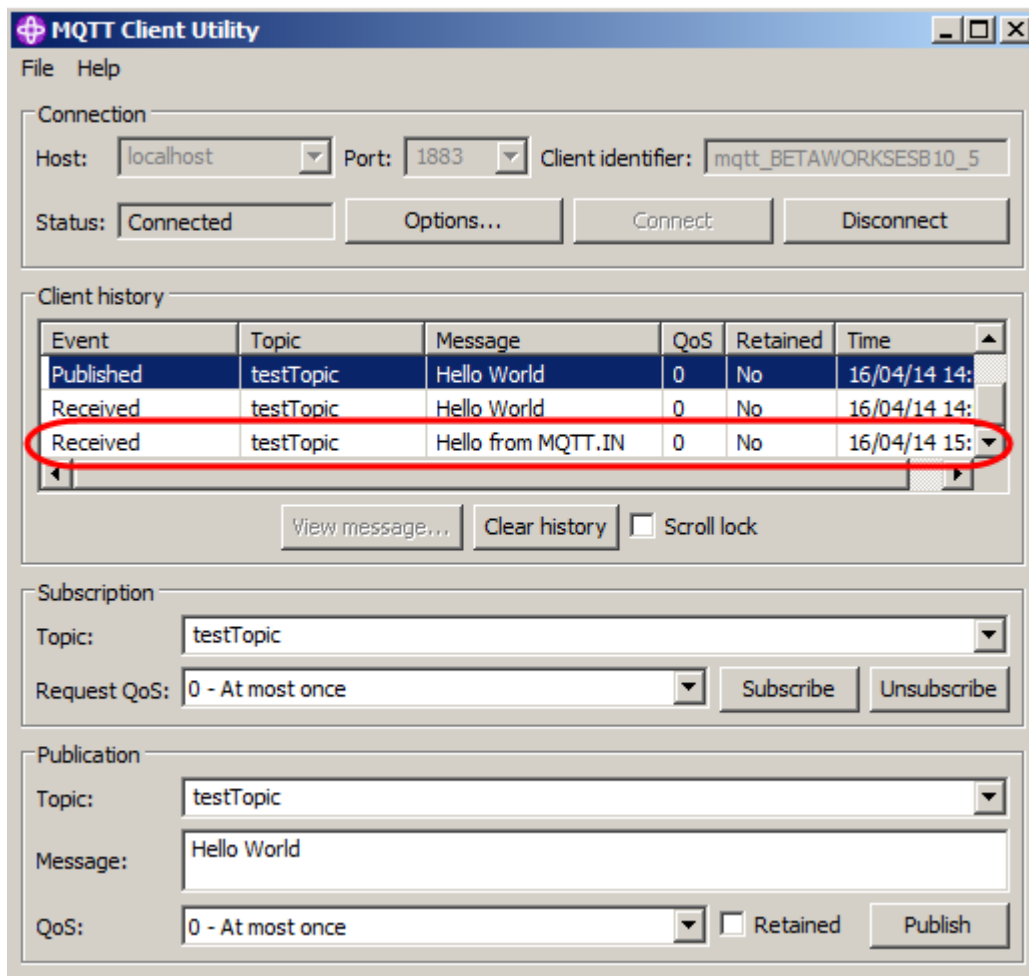
3.4 Test the TTPOLICYLAB Application

1. In WebSphere MQ Explorer, right click on the MQTT.IN queue and choose “Put test message” from the list of options.
2. Position the MQTT Client Utility where you can see it on the desktop. Ensure you are still subscribed to the “testTopic”
3. Write “Hello from MQTT.IN” in the message data:



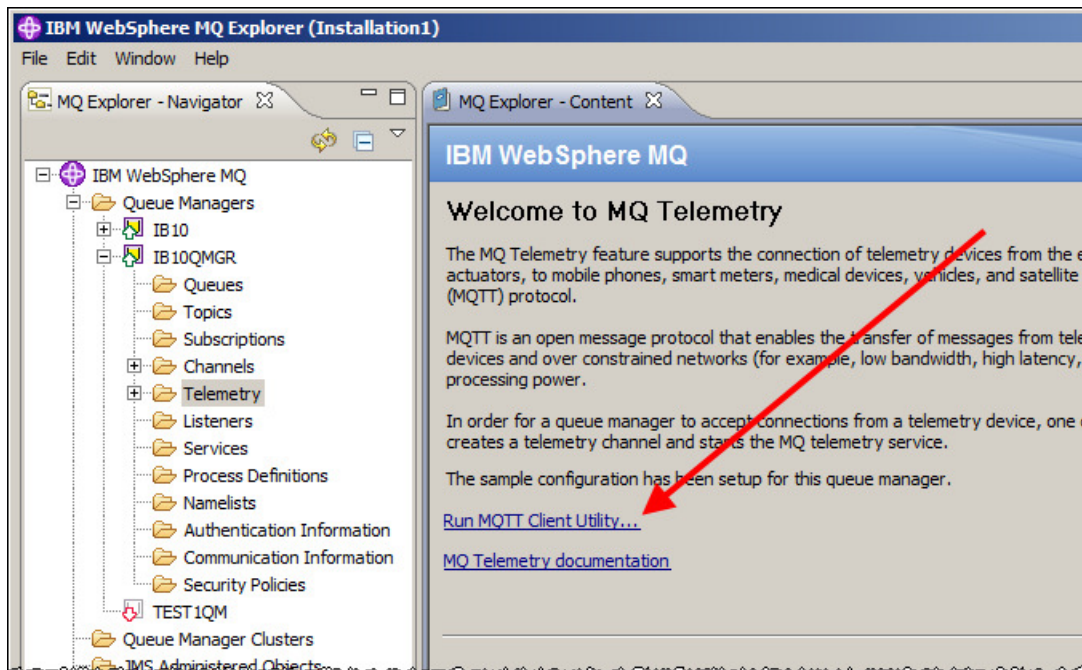
Click the “Put message” button.

- In the MQTT Client Utility the "Client history" will update with "Received" in the Event Column and the text you put to the queue `MQTT.IN` in the "message" column:

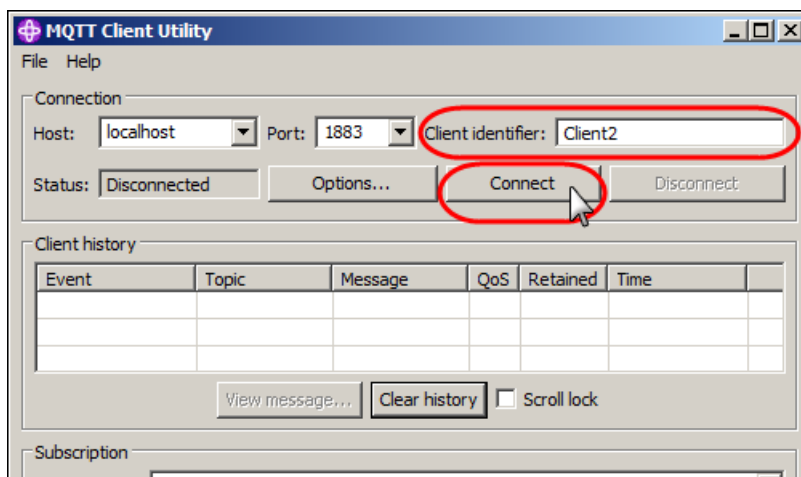


- Close the Put Test Message window.

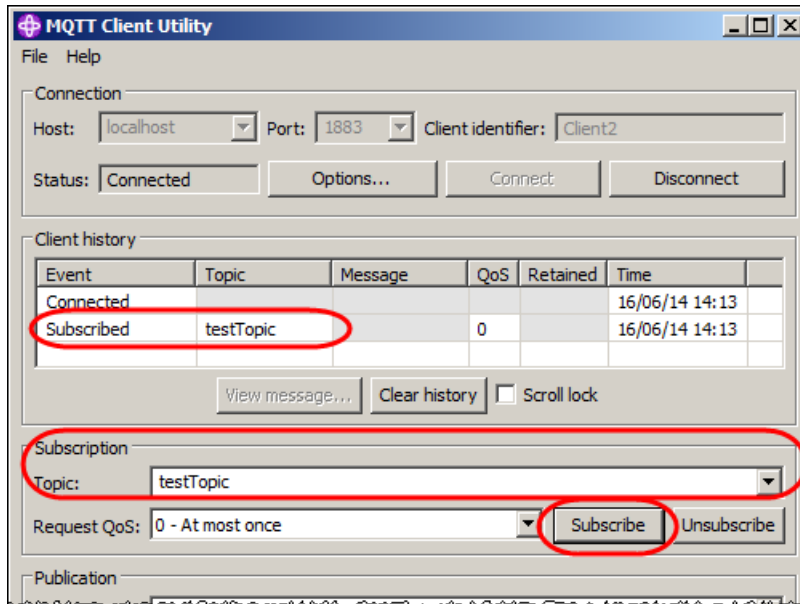
- Run another instance of the **MQTT Client Utility** by going to the MQ Explorer and navigating to IB10QMGR and clicking on the Telemetry folder, in the MQ Explorer – Content page there is a link to Run the MQTT Client Utility:



- Connect the Client Utility to the queue manager with a Client Identifier of "Client2"



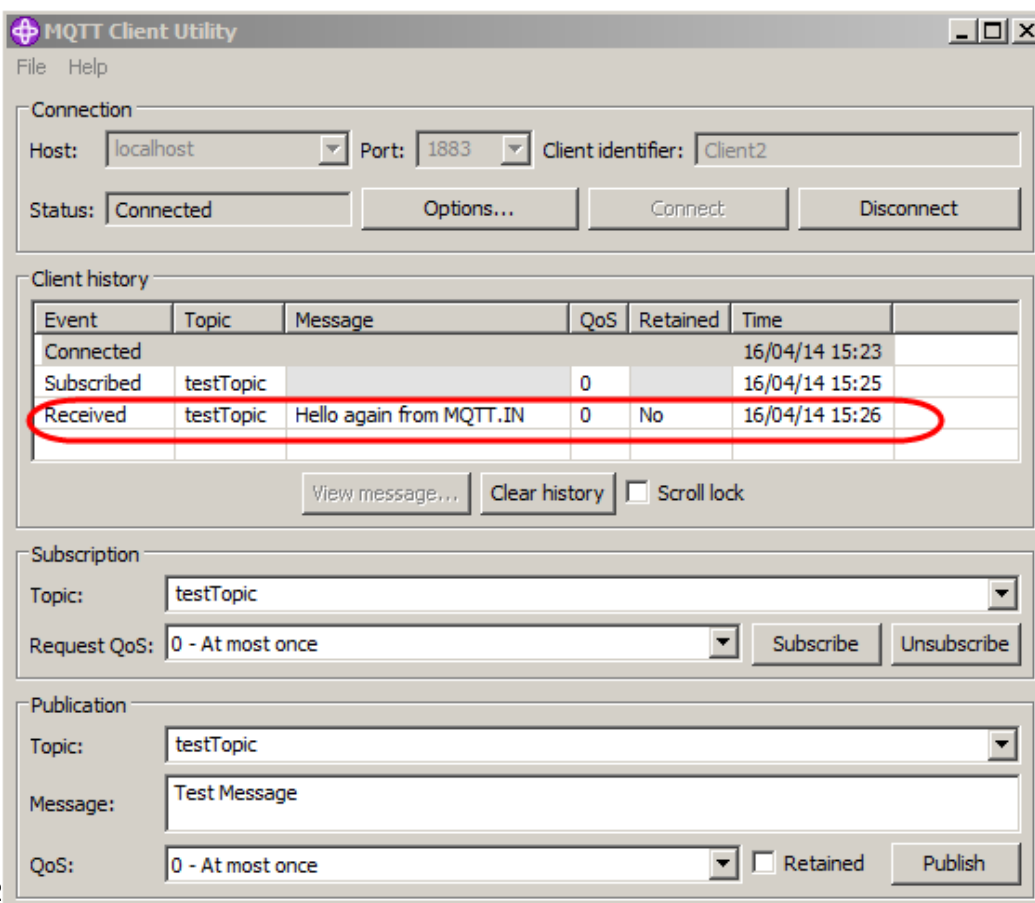
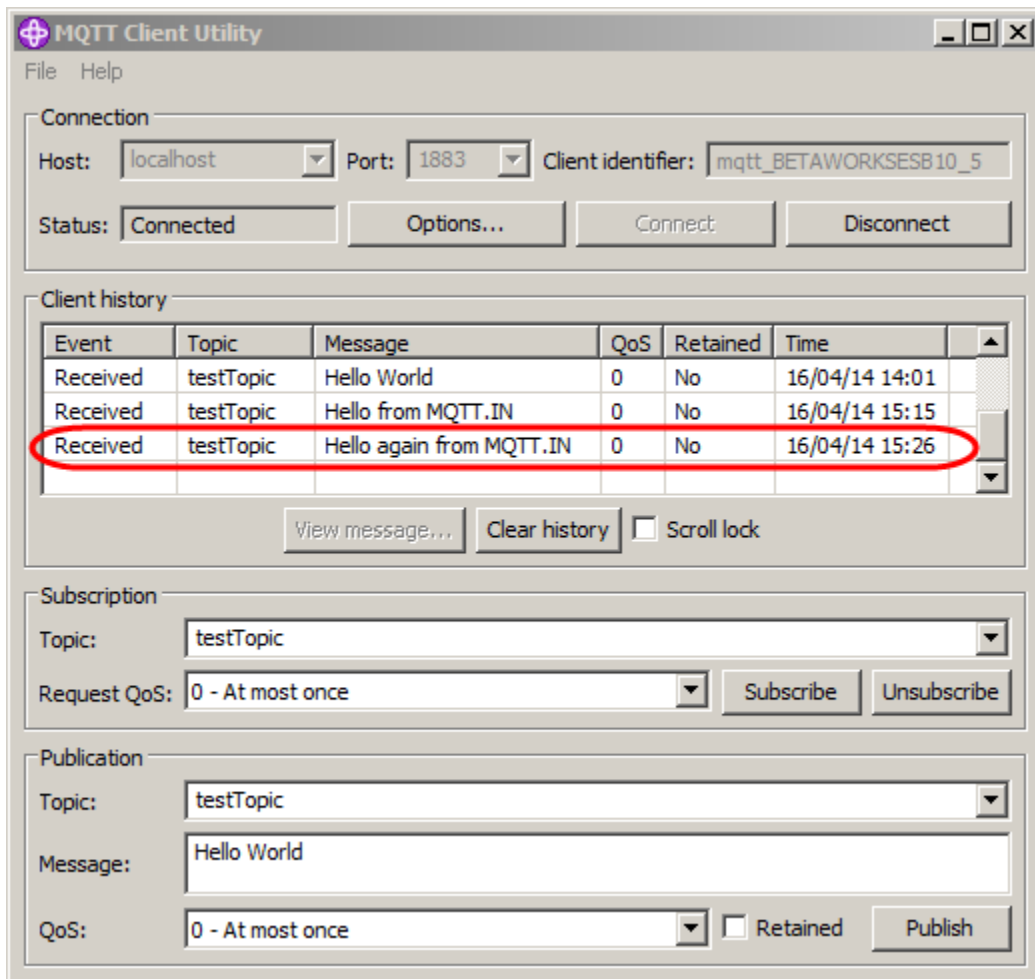
- Subscribe to "testTopic".



- Position the two Clients so that you can see them both at the same time.

Write **"Hello again from MQTT.IN"** to the MQTT.IN queue using the put Test Message data feature in the WebSphere MQ Explorer.

10. Both Clients will be updated with the data that you wrote to the queue:

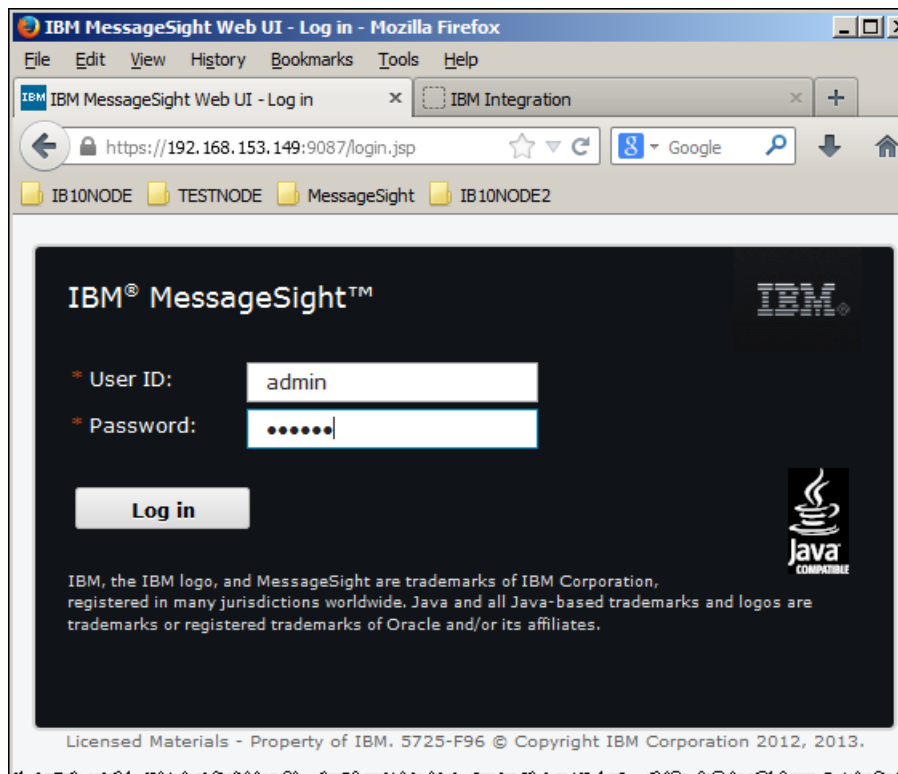


3.5 IBM MessageSight configuration

You will now define an IBM MessageSight Endpoint so that an instance of the MQTT Client Utility can subscribe to the endpoint.

1. In a browser, open the IBM MessageSight Administration UI by entering the IBM MessageSight ip address on port 9087 (*the ip address you found earlier in this lab*)

Sign on to the IBM MessageSight Administration UI (admin/admin)



2. The first screen you will see the “First Steps” configuration.

Cancel out of this configuration (don't change anything).

3. From the main menu choose <Messaging><Message Hubs>

From the “Message Hubs” page click the green plus sign to add a Message Hub, Call the Message Hub “**MQTTPolicyLab**”:

Add Message Hub
Define a message hub to manage appliance connections.

* Name:

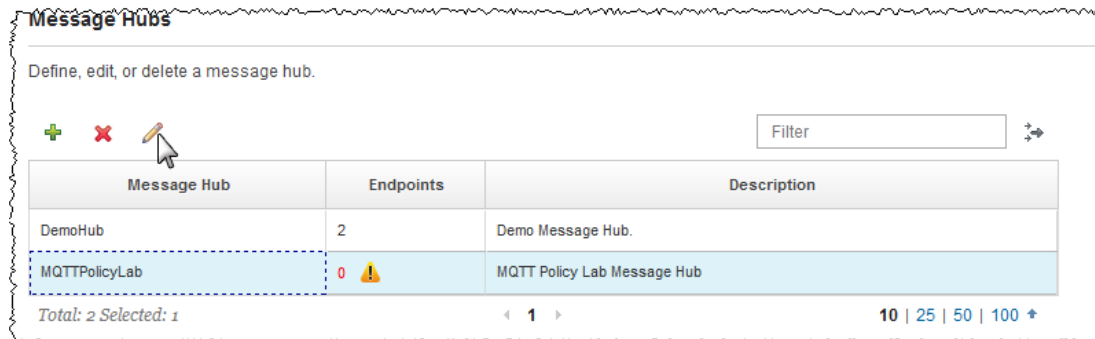
Description:

Click Save.

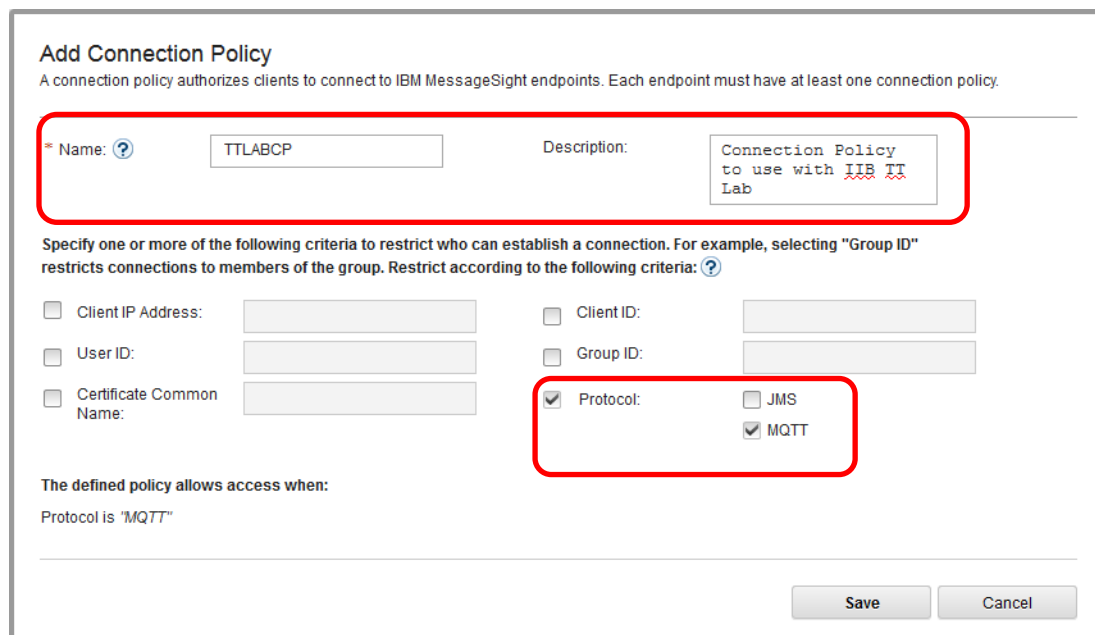
4. The Message Hub now requires three further definitions:

- 1) Connection Policy
- 2) Messaging Policy
- 3) an Endpoint.

Highlight the MQTTPolicyLab Message Hub you have just added and click the Edit “pencil”:

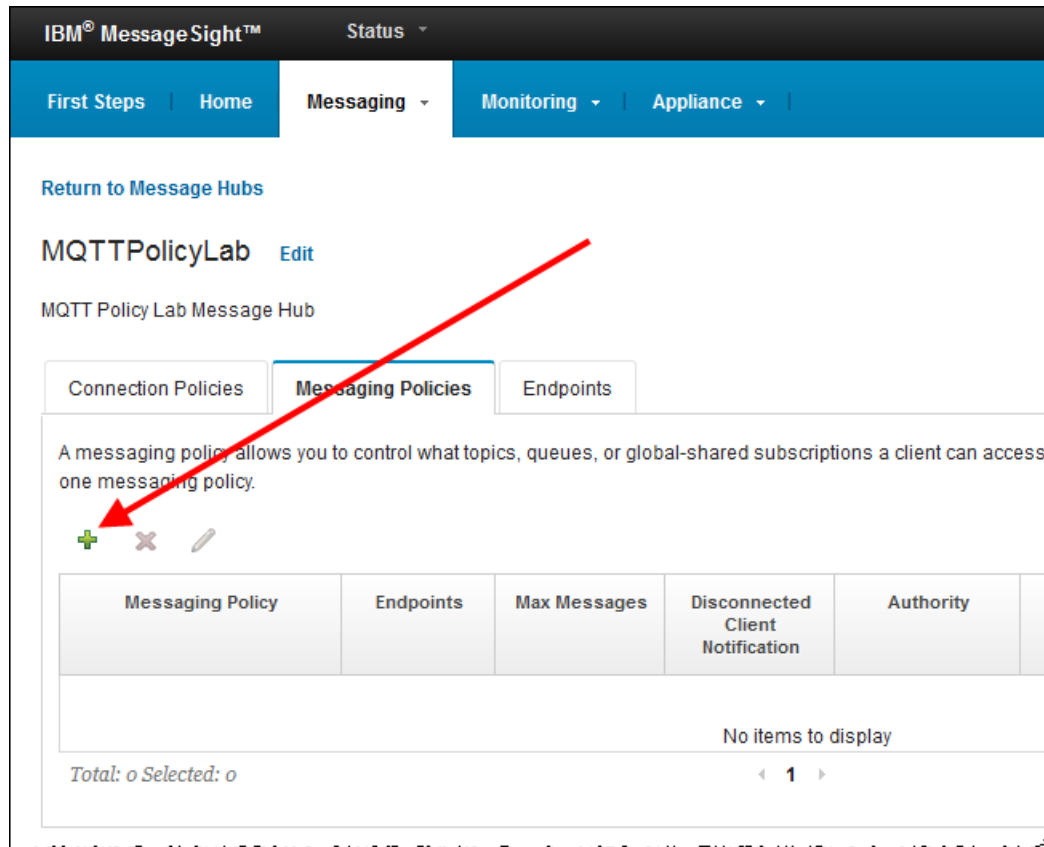


5. On the Connection Policies Tab, Click the green plus sign to add a new Connection Policy: Give it a name of TTLABCP and select the Protocol – and MQTT check boxes.



Click the Save button.

- Now switch to the Messaging Policy. Click the green plus sign to add a new **Messaging Policy**.



The screenshot shows the IBM MessageSight interface. At the top, there is a navigation bar with 'First Steps', 'Home', 'Messaging', 'Monitoring', and 'Appliance'. Below this, the page title is 'MQTTPolicyLab' with an 'Edit' link. The main content area is titled 'MQTT Policy Lab Message Hub' and has three tabs: 'Connection Policies', 'Messaging Policies', and 'Endpoints'. The 'Messaging Policies' tab is active. Below the tabs, there is a text box explaining that a messaging policy allows control over topics, queues, or global-shared subscriptions. Below the text are three icons: a green plus sign, a grey 'x', and a grey pencil. A red arrow points to the green plus sign. Below the icons is a table with five columns: 'Messaging Policy', 'Endpoints', 'Max Messages', 'Disconnected Client Notification', and 'Authority'. The table is currently empty, with the text 'No items to display' centered below it. At the bottom, there is a summary row showing 'Total: 0 Selected: 0' and a pagination control with '1' in the center.

7. Specify the following on the Messaging Policy:

Name: **TTLABMP**
 Destination Type: **Topic**
 Authority : (tick) **Publish** (&) **Subscribe**
 Tick "**Protocol**" then "**MQTT**"

Add Messaging Policy
 A messaging policy authorizes connected clients to perform specific messaging actions, such as which topics, queues, or global-shared subscriptions the client can access on IBM MessageSight. In a global-shared subscription, the work of receiving messages from a durable topic subscription is shared between multiple subscribers. Each endpoint must have at least one messaging policy.

* Name: * Authority: Publish Subscribe
 Description: Send
 * Destination Type: Browse
 * Destination: Receive
 * Max Messages: Control
 Disconnected Client Notification:

Specify one or more of the following criteria to restrict the messaging actions defined in this policy to specific connected clients. For example, selecting "Group ID" restricts the policy to members of the group. Restrict according to the following criteria: [?](#)

Client IP Address: Client ID:
 User ID: Group ID:
 Certificate Common Name: Protocol: JMS MQTT

The defined policy allows access when: Protocol is "MQTT"

Click the Save button

8. The Messaging Policy "TTLABMP" will appear in the list of messaging Policies:

MQTTPolicyLab [Edit](#)

MQTT Policy Lab Message Hub

Connection Policies **Messaging Policies** Endpoints

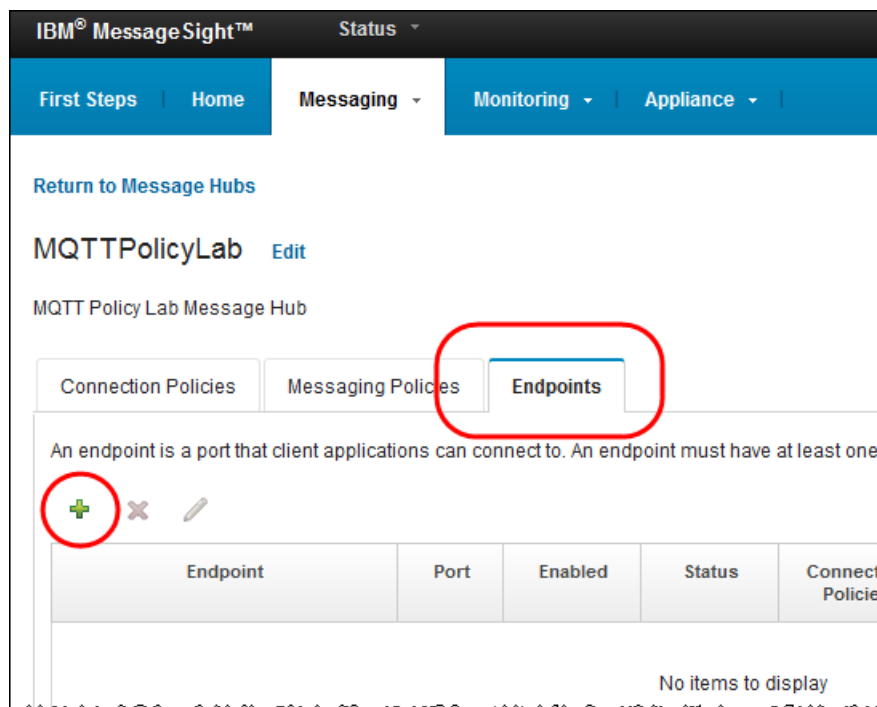
A messaging policy allows you to control what topics, queues, or global-shared subscriptions a client can access on IBM MessageSight. Each endpoint must have at least one messaging policy.

Messaging Policy	Endpoints	Max Messages	Disconnected Client Notification	Authority	Destination	Description
TTLABMP	0	5,000	False	Publish,Subscribe	*	Messaging Policy for TT Lab

Total: 1 Selected: 0 10 | 25 | 50 | 100 +

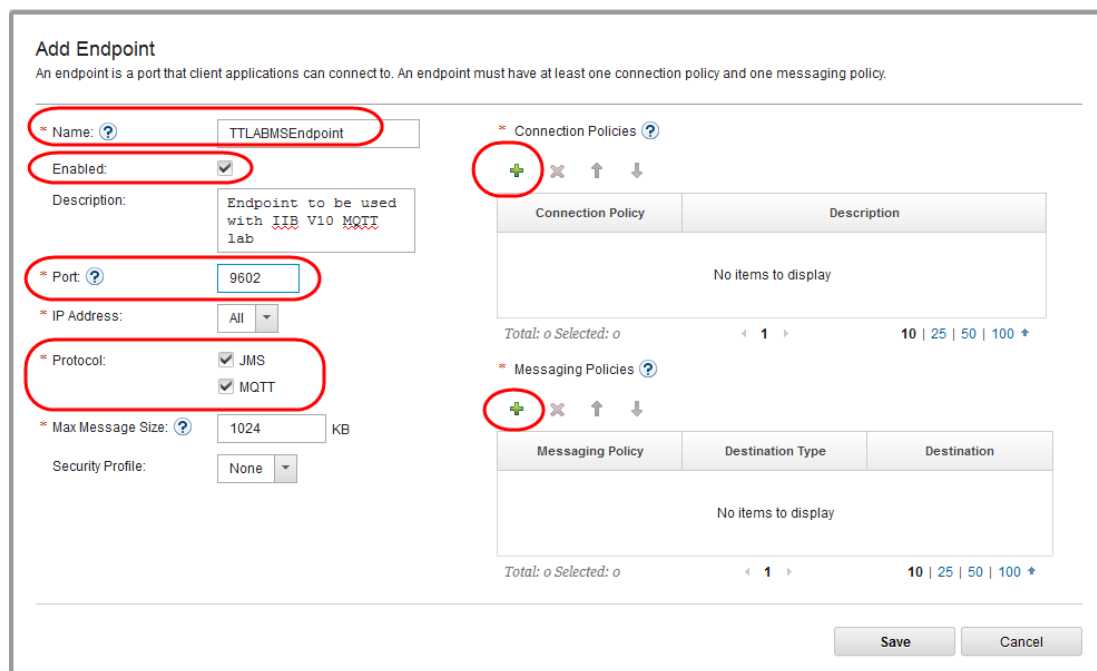
- Click the EndPoints Tab.

Click the green plus sign to add an Endpoint:

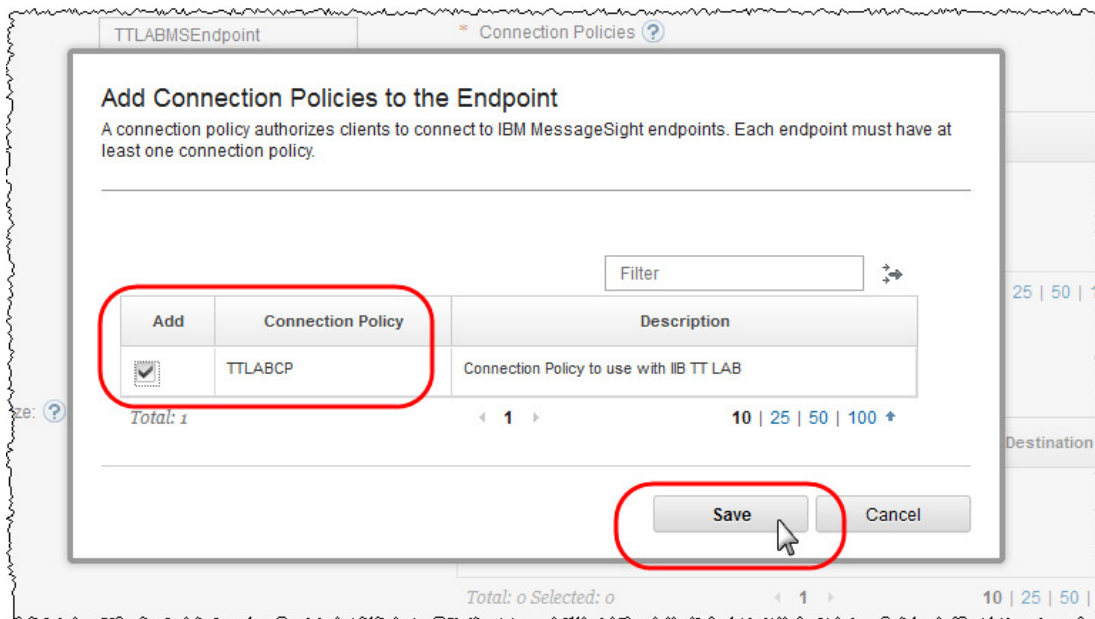


- On the pop up window specify the endpoint as follows. (Note: the screen layout may be slightly different to that shown).

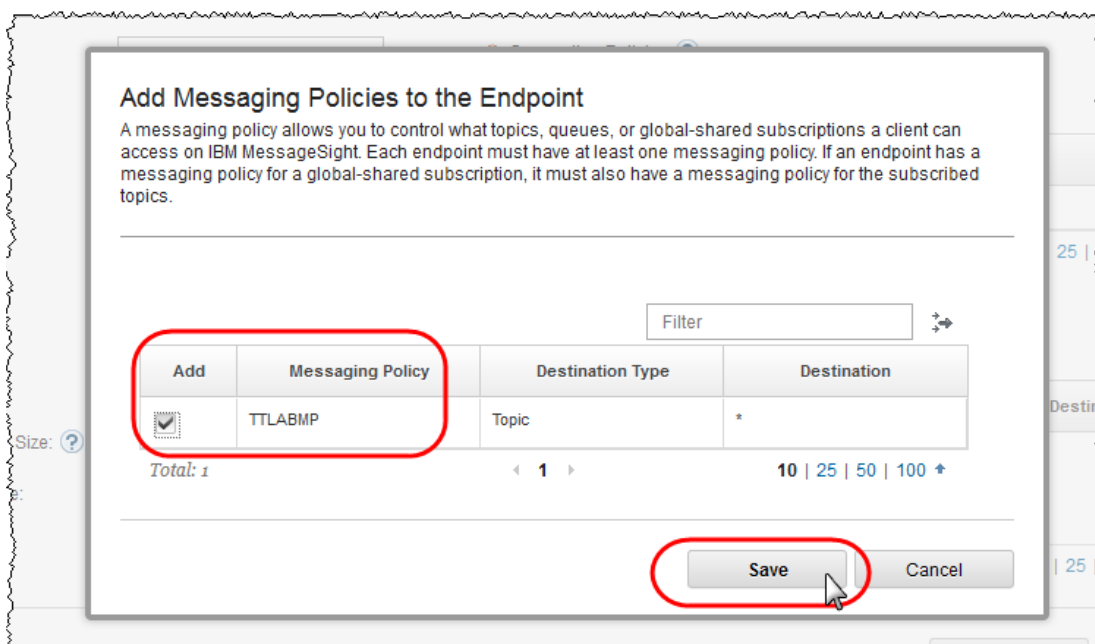
Name: **TTLABMSEndPoint**
 Enabled: (tick)
 Port: **9602**
 Protocol: (tick) JMS & MQTT



- Use the green plus signs to add **TTLABCP** Connection policy to the endpoint definition.



- Use the green plus signs to add **TTLABMP** messaging policy to the endpoint definition.



13. When the Endpoint definition looks like this, click the Save button:

Add Endpoint

An endpoint is a port that client applications can connect to. An endpoint must have at least one connection policy and one messaging policy.

Name:

Enabled:

Description:

Port:

IP Address:

Protocol: JMS MQTT

Max Message Size: KB

Security Profile:

Connection Policies

Connection Policy	Description
TTLABCP	Connection Policy to use with IIB TT LAB

Total: 1 Selected: 0

Messaging Policies

Messaging Policy	Destination Type	Destination
TTLABMP	Topic	*

Total: 1 Selected: 0

14. TTLABMSEndPoint will appear in the list of Endpoints:

MQTTPolicyLab [Edit](#)

MQTT Policy Lab Message Hub

Connection Policies | Messaging Policies | **Endpoints**

An endpoint is a port that client applications can connect to. An endpoint must have at least one connection policy and one messaging policy.

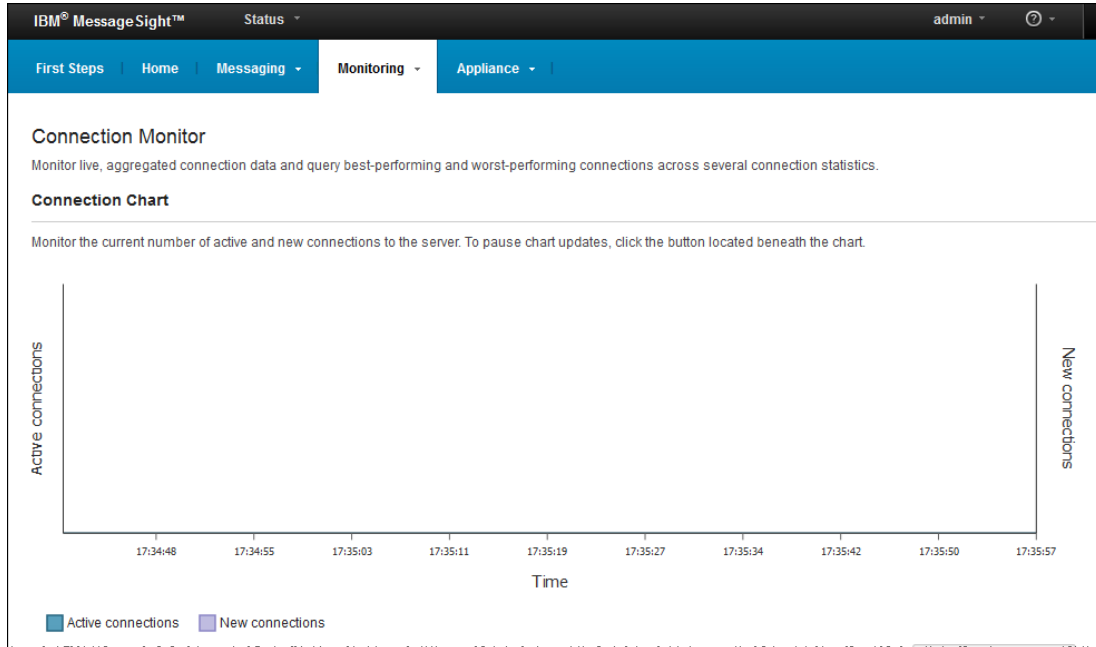
Endpoint	Port	Enabled	Status	Connection Policies	Messaging Policies	Description
TTLABMSEndPoint	9602	<input checked="" type="checkbox"/>	↑	? 1	? 1	Endpoint to be used with TT Lab

Total: 1 Selected: 0

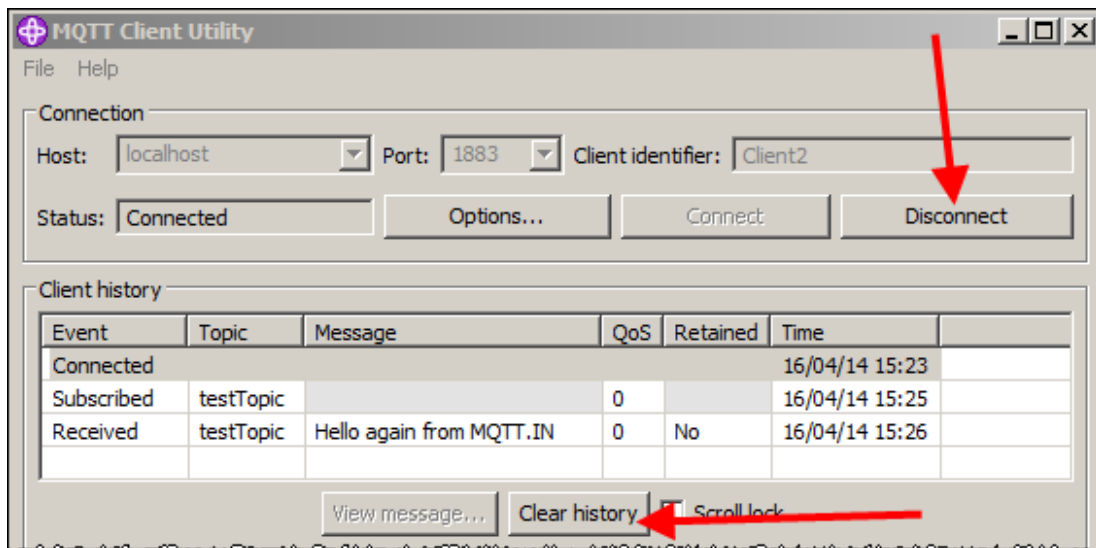
3.6 Test the Simple IBM MessageSight configuration

You will test the IBM MessageSight configuration using one of the MQTT Client Utility by connecting to the endpoint defined above:

1. In the IBM MessageSight administration UI, click <Monitoring><Connections> The Connection chart will not show any active connections:



2. In the MQTT Client Utility with Client identifier "Client2" disconnect from the localhost connection and clear the Client history:

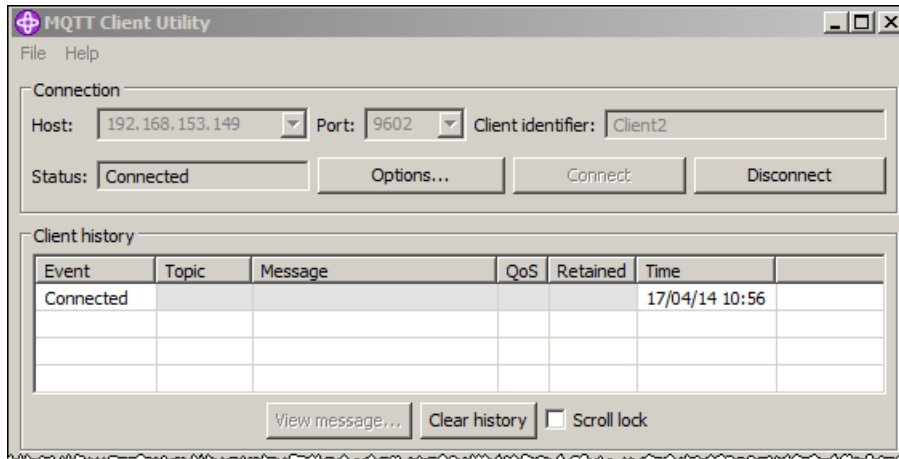


- Connect the MQTT Client Utility to the IBM MessageSight configuration that you just added:

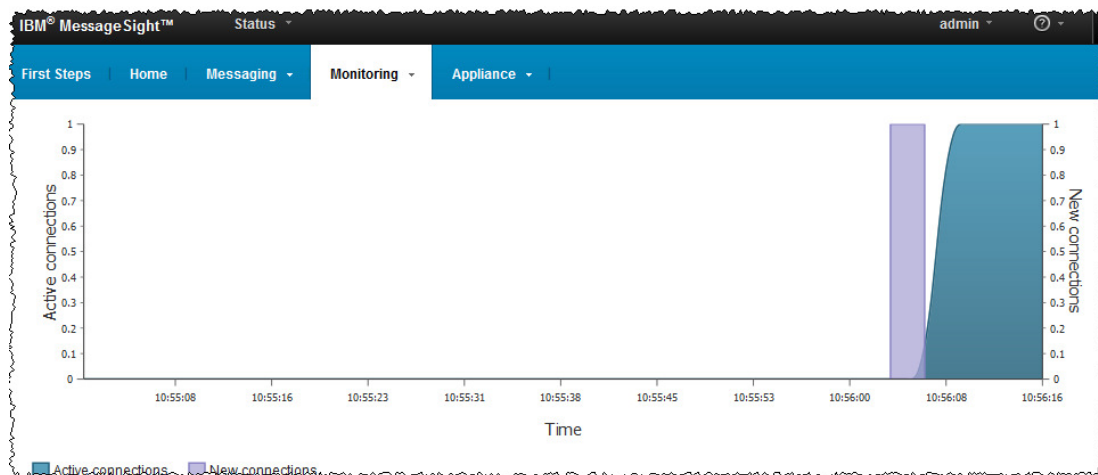
Specify:

- Host ip address of the IBM MessageSight appliance
- Port details of the Endpoint you defined above (9602)

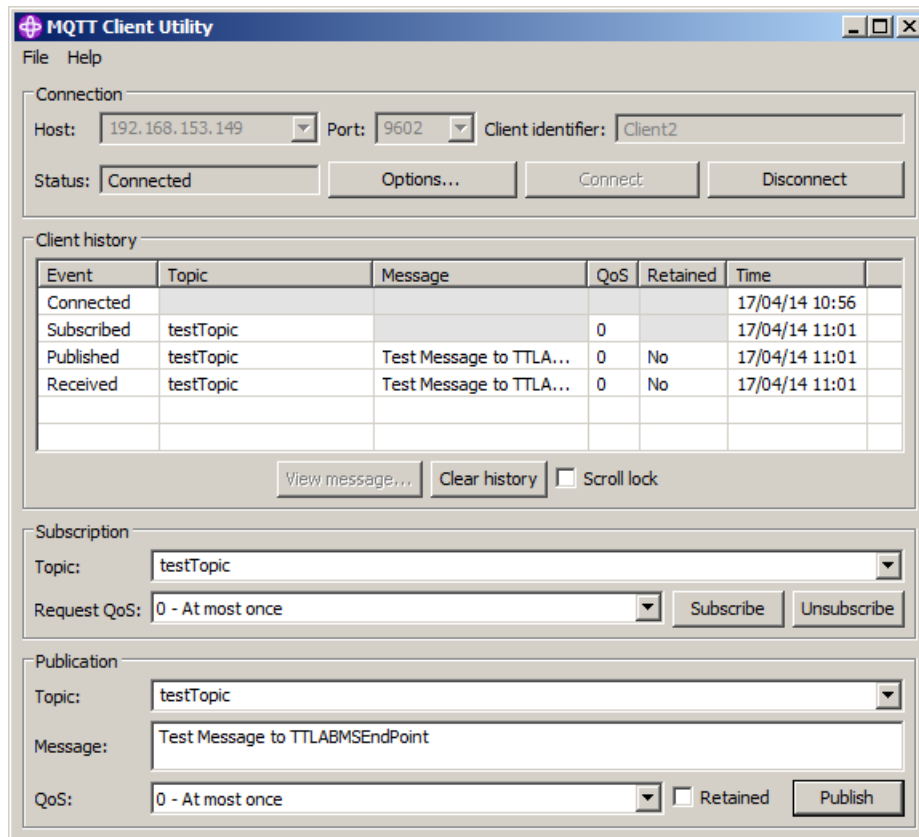
The Client History should be updated with “Connected” in the Event Column:



- The Connection monitor should now show a new connection:



5. In the MQTT Client Utility:
 - a) Subscribe to “testTopic” (note the Client History update when you do this)
 - b) Publish a message to the testTopic from the Client Utility (note two updates for this action in the Client History: “Published” to the testTopic; “Received” (via the subscription) with the details of the message received:



6. This verifies your IBM MessageSight configuration is working.
7. Close any open MQTT Client Utility.

4. Modifying the IIB MQTT configuration using Node Policy

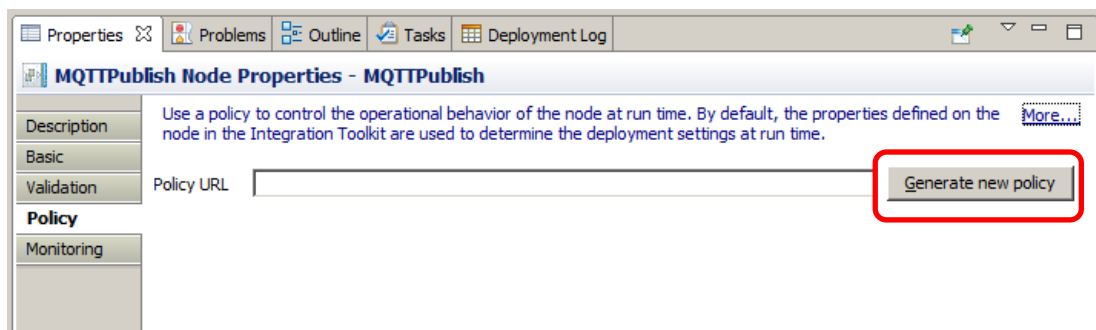
You will now define a node policy for the MQTT publish node that will allow you to dynamically change the connection specific details without the need to redeploy the message flow.

4.1 Generate and store a policy

So far the testing you have done to verify the MQTTPublish node is working has been based on properties statically defined in the MQTTPublish node. You will now generate a policy for the MQTTPublish node and store it in the Integration Registry associated with the TESTNODE_iibuser.

1. In the mqttPublish message flow, click the MQTTPublish node to show the properties window.

In the MQTTPublish node, (Select Policy section), click the “Generate new policy” button:

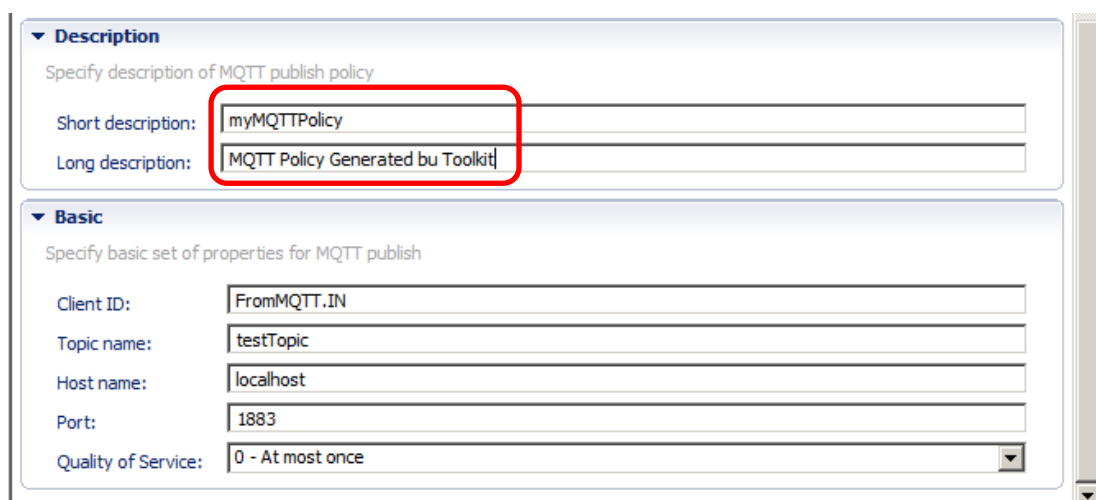


If you are prompted to save the mqttPublish.msgflow, reply Yes.

The Policy Editor will open.

2. The Policy Editor window will open showing the current settings of the MQTT Publish node.

Type **myMQTTPolicy** in the short description followed by something suitable in the Long Description.



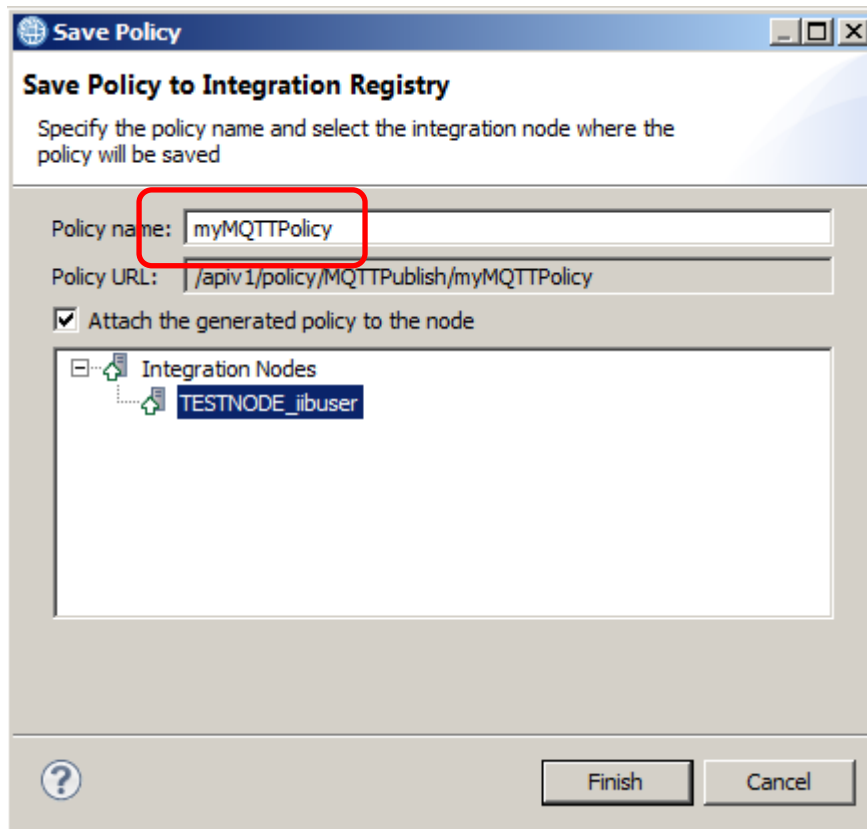
Ensure the details in the “Basic” section reflect the definition of your MQTT Publish node.

3. Click the save link.

Enter the Policy name “myMQTTPolicy”.

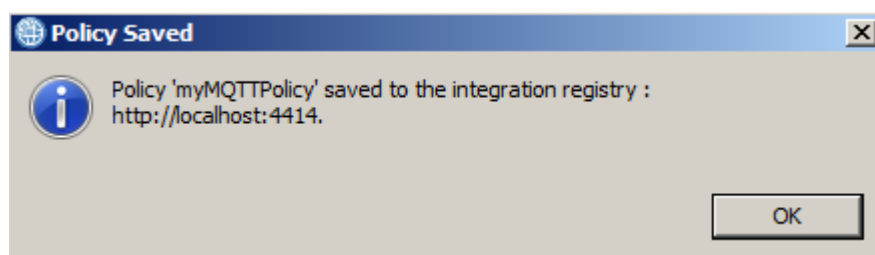
Select the node where the policy will be saved - TESTNODE_iibuser.

Click Finish.



IIB Toolkit communicates with the Integration Registry using the same port and value used by the web ui. These values are “localhost” and “4421” respectively if you are using the VMware provided for this class lab guide.

4. Click OK to dismiss the message that pops up when the policy has been saved.

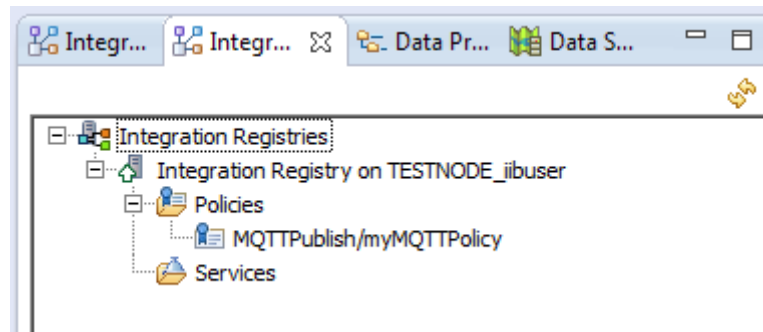


5. Save the message flow (ctrl s).

- Switch to the "Integration Registries" hierarchical view (bottom left of the Integration Toolkit).

Expand the Integration Registry on TESTNODE_iibuser to show "Policies" and "Services". If **myMQTTPolicy** is not showing in the list, Right click on the "Integration Registry on TESTNODE_iibuser" and click "Refresh".

MyMQTTPolicy will appear in the list:



- Redeploy the "TPOLICYLAB" application to the default server on TESTNODE_iibuser.
- Stop and restart the server "default" Integration Server.

4.2 Modify the Policy in the Integration Registry

We will now modify the stored policy in the Integration Registry associated with TESTNODE_ibuser, so that the MQTTPublish node, publishes to the IBM MessageSight device that you configured earlier instead of the MQ MQTT server on port 1883.

1. In a web browser, go the bookmarked IIB web administration utility, expand the Operational Policy section at the left under the TESTNODE_ibuser integration node, then locate and highlight the policy that was created earlier, myMQTTPolicy.

The screenshot displays the IBM Integration Bus web administration utility interface. The left-hand navigation pane shows a tree structure under 'TESTNODE_ibuser', with 'Operational Policy' expanded to 'MQTTPublish', where 'myMQTTPolicy' is highlighted with a red box. The main content area is titled 'Operational Policy - MQTTPublish : myMQTTPolicy' and shows the configuration for this policy. The 'Policy URL' is '/apiv1/policy/MQTTPublish/myMQTTPolicy'. The 'Description' section includes a 'Short description' of 'myMQTTPolicy'. The 'Basic' section contains the following fields:

Client ID	FromMQTT.IN
Topic name	testTopic
Host name	localhost
Port	1883
Quality of service	0 - At most once

2. Change the IP address to reflect the configuration of your IBM MessageSight and the port to **9602** (the port value specified for the Endpoint that you configured on the IBM MessageSight appliance).

Then click the save button. You are now ready to test the changes to the Policy.

The screenshot shows the IBM Integration Bus configuration interface for an Operational Policy named 'myMQTTPolicy'. The interface includes a left-hand navigation tree and a main configuration area. The 'Basic' section of the configuration is highlighted with a red box, showing the following fields:

Client ID	FromMQTT.IN
Topic name	testTopic
Host name	192.168.126.45
Port	9602
Quality of service	0 - At most once

The 'Save' button is also highlighted with a red box. The 'Policy URL' is set to '/apiv1/policy/MQTTPublish/myMQTTPolicy'. The 'Short description' is 'myMQTTPolicy'. The 'Long description' is empty. The 'Basic' section is expanded, and the 'Port' field is a dropdown menu.

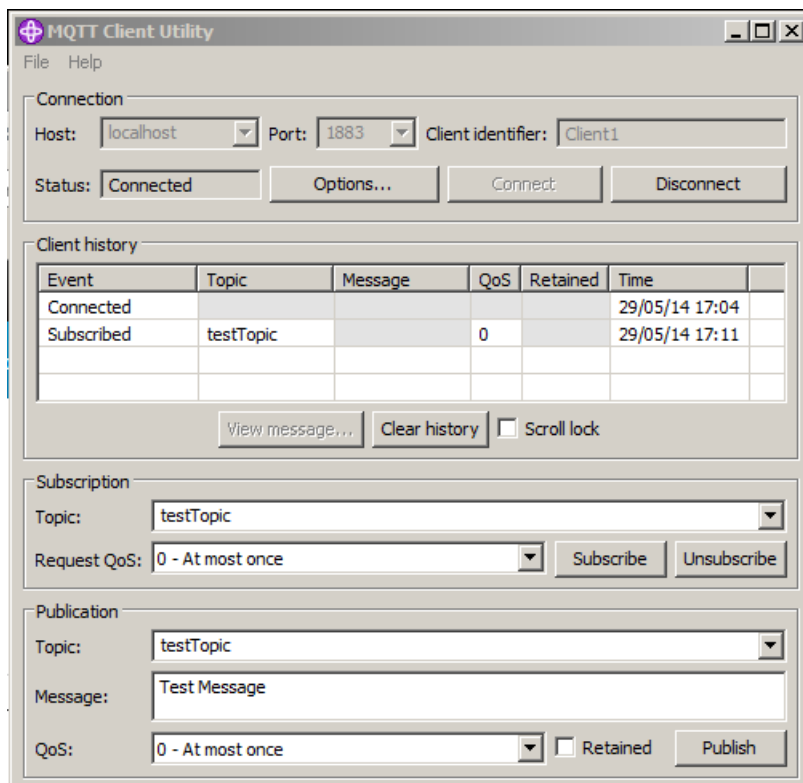
4.3 Testing the result of a Policy Change

1. Open two MQTT Clients by clicking Run MQTT Client Utility (on the content window of the Telemetry folder) in MQ Explorer as you did previously.

Using the Client Identifier field, call the clients **Client1** and **Client2** respectively.

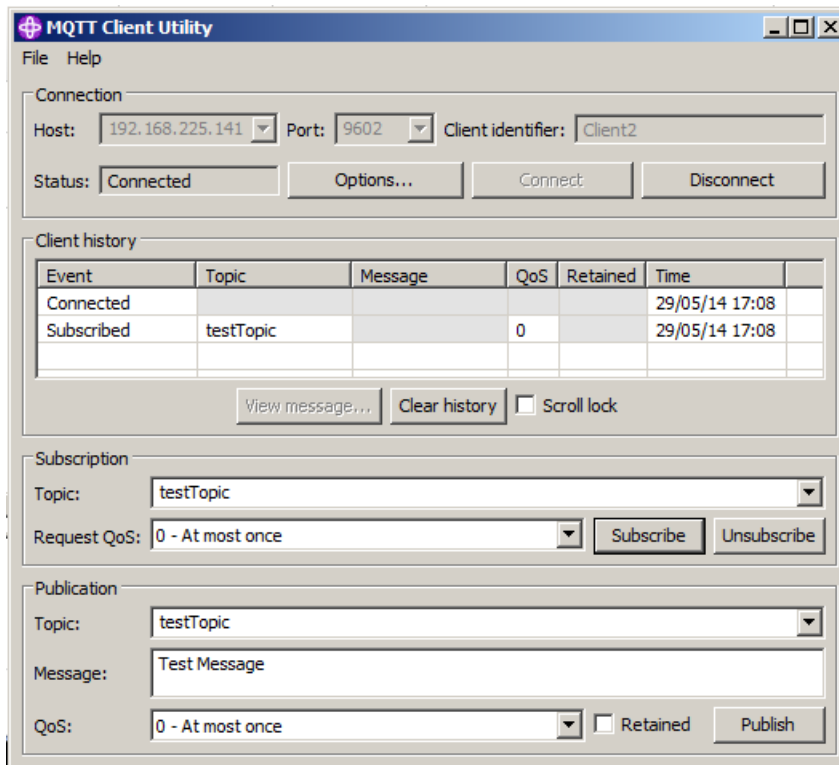
2. Connect Client1 to the default MQTT server on **IB10QMGR** by connecting to port **1883** on Localhost.

When connected subscribe to **“testTopic”**:

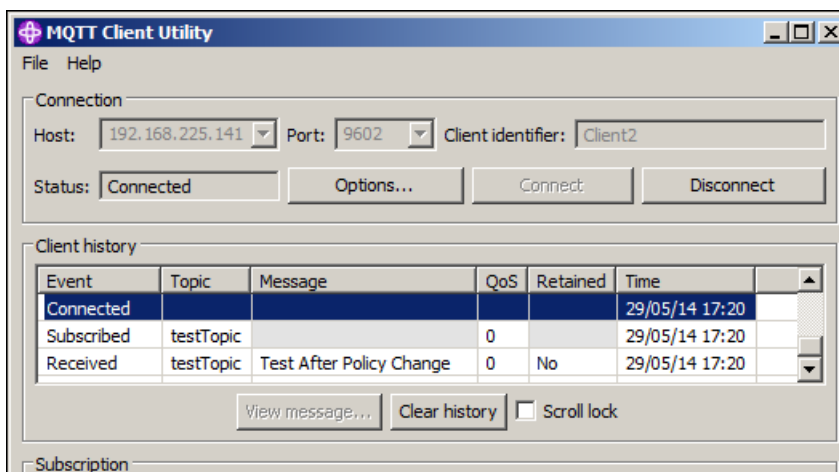


- Connect Client2 to your IBM MessageSight appliance by connecting to port **9602** on the ip address of your local IBM MessageSight.

When connected also subscribe to **“testTopic”** :



- Position the **Client 2** below **Client 1** so that you can see what each Client Utility receives.
- Using the MQ Explorer, “put test message” feature, initiate the flow by writing **“Test After Policy Change”** in a test message to **MQTT.IN**.
- Note that **Client2** (the Client Utility connected to the IBM MessageSight appliance) receives the publication and Client1 does not:



In Summary the Change in the “myMQTTpolicy” configuration to point to the IBM MessageSight Endpoint resulted in:

- a) the message being published to the IBM MessageSight Endpoint.
- b) this resulted in Client2 (which was connected and subscribing to the same IBM MessageSight Endpoint) to receive the message.

The change in connection was done using Node Policy configuration changes which did not require message flow deployment.

END OF LAB GUIDE