Watson IoT Platform – T4 - NodeRed – Subscriber Client

Commissioning task 4

1. Introduction

In the earlier commissioning task we have created a connection between a real device and the IBM Cloud Watson IoT Platform. We have created a connection between a simulated client device and as well a real client device and the NodeRED server application running on IBM Cloud Watson.

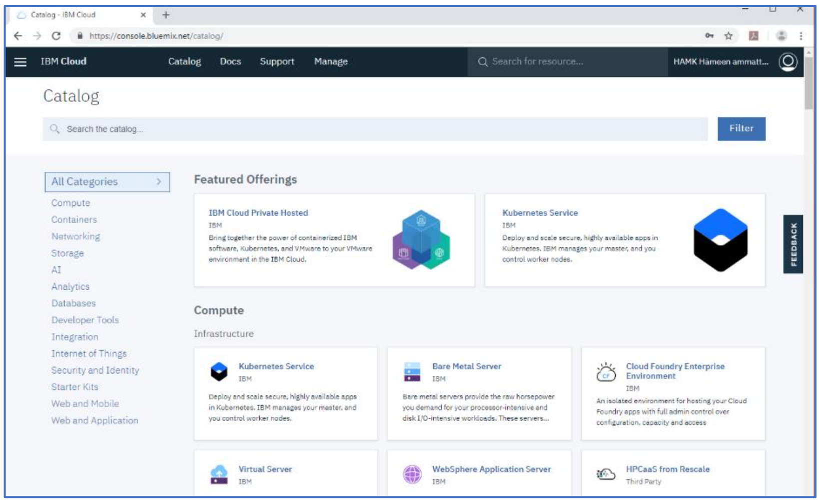
In the earlier commissioning tasks the client acted as a Publisher according to the mqtt protocol. In this commissioning task we will configure a client which will act as a Subscriber according to the mqtt protocol.

Both the Publisher and the Subscriber devices will authenticate themselves in the IBM Cloud with the Client ID user name and Authentication Token password. The Client ID and the Authentication Token have been created in the Watson IoT Platform. In addition to this authentication method both the publisher topic and the subscriber topic needs to be defined according to certain rules.

2. A client will be created both for publishing information and for subscribing to information

Please search with a browser for “IBM Cloud” . Select from proposed sites the https://www.ibm.com/cloud/ . Log in here with your IBMid credentials and further on top of the page select the “Catalog” . You can browse directly to address https://console.bluemix.net/catalog/ if your IoT Platform instance is in the older cloud platform. The relevant addresses can have been changed already when you are completing this task. These addresses are based on the situation in the summer 2019.

To be able to configure the client devices you need to have an IBM Cloud user account. This user account is the so called IBMid. You will need the academic user status to be able to complete the tasks in the NodeRED. The NodeRED will be installed in the IBM Cloud platform. Some other status higher than the free user id available for everyone will work as well. Instructions on how to upgrade the status from free to academic can be found in the earlier commissioning tasks. If you will try to complete the assignments built with NodeRED using the free account you might run out of resources.



Catalog - IBM Cloud

https://console.bluemix.net/catalog/

IBM Cloud · Catalog · Docs · Support · Manage · Search for resource... · HAMK Hämeen ammatt...

Catalog

Search the catalog... · Filter

All Categories: Compute, Containers, Networking, Storage, AI, Analytics, Databases, Developer Tools, Integration, Internet of Things, Security and Identity, Starter Kits, Web and Mobile, Web and Application

Featured Offerings

IBM Cloud Private Hosted
IBM
Bring together the power of containerized IBM software, Kubernetes, and VMware to your VMware environment in the IBM Cloud.

Kubernetes Service
IBM
Deploy and scale secure, highly available apps in Kubernetes. IBM manages your master, and you control worker nodes.

Compute

Infrastructure

Kubernetes Service
IBM
Deploy and scale secure, highly available apps in Kubernetes. IBM manages your master, and you control worker nodes.

Bare Metal Server
IBM
Bare metal servers provide the raw horsepower you demand for your processor-intensive and disk I/O-intensive workloads. These servers...

Cloud Foundry Enterprise Environment
IBM
An isolated environment for hosting your Cloud Foundry apps with full admin control over configuration, capacity and access

Virtual Server
IBM

WebSphere Application Server
IBM

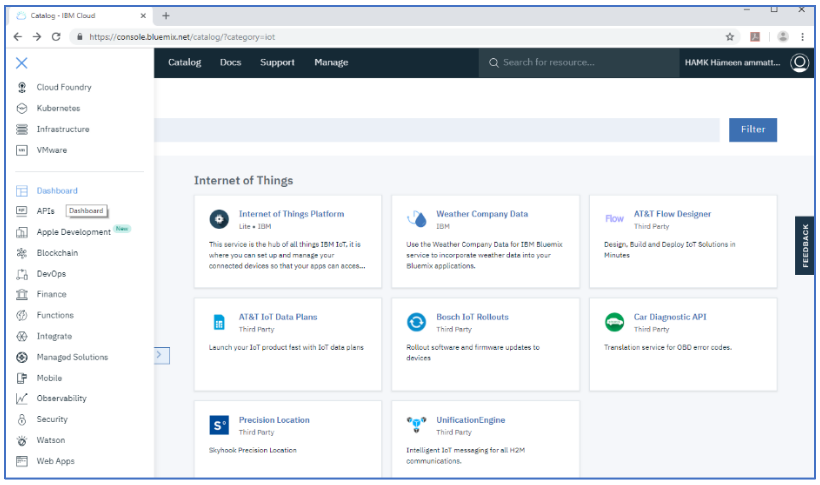
HPCaaS from Rescale
Third Party

FEEDBACK

Fig 2.1 Registerin the IBM Cloud service. Please select on top of the page the Catalog to browse to page IBM Cloud Catalog. The address for the Catalog page might be https://cloud.ibm.com/catalog.

If you haven’t already established the Internet of Things Platform service for your user account please do it now. Please select out of available packages the ”Internet of Things Platform Starter”. This is under subtitle Starter Kits.

If you already in the earlier assignments and tasks established the Internet of Things Platform service please browse to it by selecting in the left drop down menu the Dashboard.



Catalog - IBM Cloud

https://console.bluemix.net/catalog/?category=iot

Catalog · Docs · Support · Manage · Search for resource... · HAMK Hämeen ammatt...

Cloud Foundry, Kubernetes, Infrastructure, VMware

Dashboard, APIs (Dashboard), Apple Development (New), Blockchain, DevOps, Finance, Functions, Integrate, Managed Solutions, Mobile, Observability, Security, Watson, Web Apps

Filter

Internet of Things

Internet of Things Platform
Lite • IBM
This service is the hub of all things IBM IoT, it is where you can set up and manage your connected devices so that your apps can acces...

Weather Company Data
IBM
Use the Weather Company Data for IBM Bluemix service to incorporate weather data into your Bluemix applications.

AT&T Flow Designer
Third Party
Design, Build and Deploy IoT Solutions in Minutes

AT&T IoT Data Plans
Third Party
Launch your IoT product fast with IoT data plans

Bosch IoT Rollouts
Third Party
Rollout software and firmware updates to devices

Car Diagnostic API
Third Party
Translation service for OBD error codes.

Precision Location
Third Party
Skyhook Precision Location

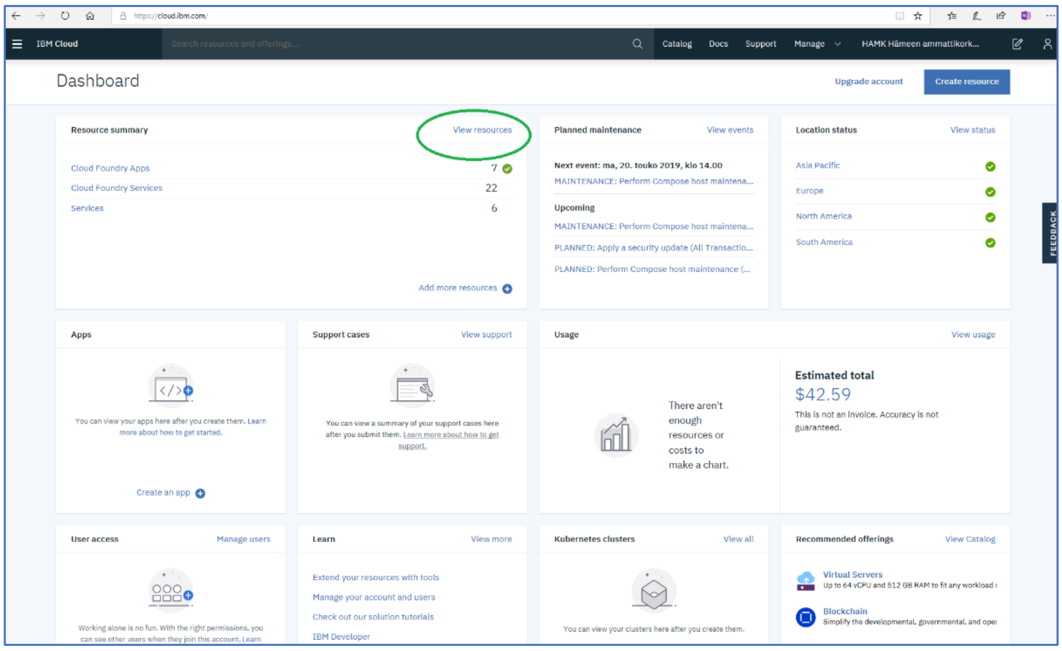
UnificationEngine
Third Party
Intelligent IoT messaging for all H2M communications.

FEEDBACK

Fig 2.2 Please select the Dashboard on the drop down menu.

The Dashboard will show you all the services and applications created earlier.

The latest version of the Cloud page might bring you to the page Dashboard.



Dashboard

Resource summary View resources

Cloud Foundry Apps	7 ✓
Cloud Foundry Services	22
Services	6

Add more resources

Planned maintenance View events

Next event: ma, 20. touko 2019, klo 14.00
MAINTENANCE: Perform Compose host maintena...

Upcoming
MAINTENANCE: Perform Compose host maintena...
PLANNED: Apply a security update (All Transactio...
PLANNED: Perform Compose host maintenance (...

Location status View status

Asia Pacific	✓
Europe	✓
North America	✓
South America	✓

Apps

You can view your apps here after you create them. Learn more about how to get started.

Create an app

Support cases View support

You can view a summary of your support cases here after you submit them. Learn more about how to get support.

Usage View usage

There aren't enough resources or costs to make a chart.

Estimated total
$42.59
This is not an Invoice. Accuracy is not guaranteed.

User access Manage users

Working alone is no fun. With the right permissions, you can see other users when they join this account. Learn

Learn View more

Extend your resources with tools
Manage your account and users
Check out our solution tutorials
IBM Developer

Kubernetes clusters View all

You can view your clusters here after you create them.

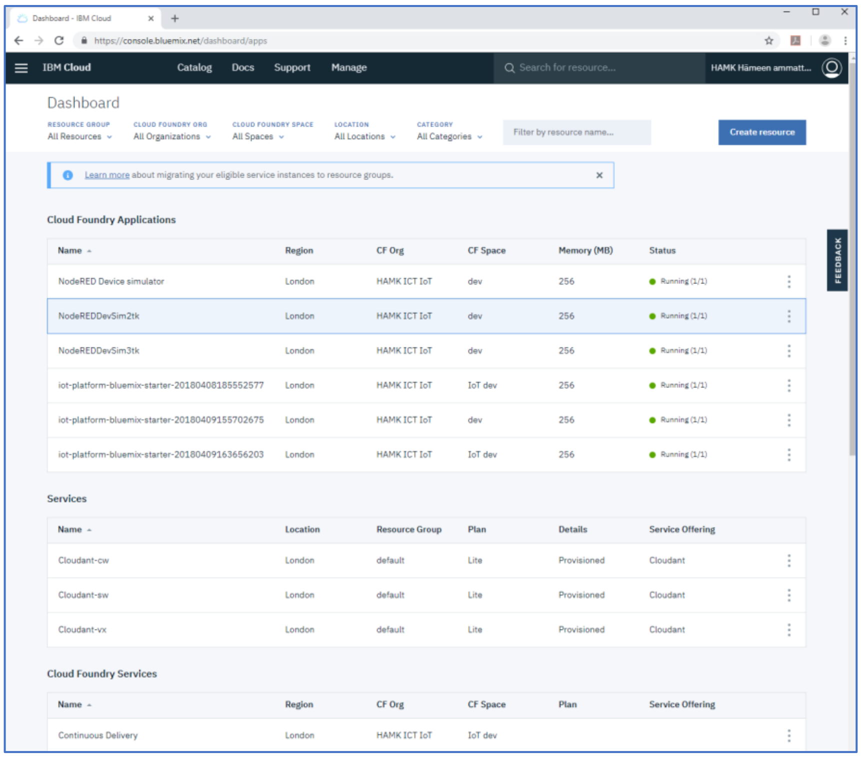
Recommended offerings View Catalog

Virtual Servers
Up to 64 vCPU and 512 GB RAM to fit any workload i

Blockchain
Simplify the developmental, governmental, and oper

Fig. 2.3 Dashboard. Please select Resource Summary and View resources.

Please select from this Dashboard page the View Resources.



Dashboard

RESOURCE GROUP: All Resources
CLOUD FOUNDRY ORG: All Organizations
CLOUD FOUNDRY SPACE: All Spaces
LOCATION: All Locations
CATEGORY: All Categories
Filter by resource name...

Learn more about migrating your eligible service instances to resource groups.

Cloud Foundry Applications

Name	Region	CF Org	CF Space	Memory (MB)	Status
NodeRED Device simulator	London	HAMK ICT IoT	dev	256	● Running (1/1)
NodeREDDevSim2tk	London	HAMK ICT IoT	dev	256	● Running (1/1)
NodeREDDevSim3tk	London	HAMK ICT IoT	dev	256	● Running (1/1)
iot-platform-bluemix-starter-20180408185552577	London	HAMK ICT IoT	IoT dev	256	● Running (1/1)
iot-platform-bluemix-starter-20180409155702675	London	HAMK ICT IoT	dev	256	● Running (1/1)
iot-platform-bluemix-starter-20180409163656203	London	HAMK ICT IoT	IoT dev	256	● Running (1/1)

Services

Name	Location	Resource Group	Plan	Details	Service Offering
Cloudant-cw	London	default	Lite	Provisioned	Cloudant
Cloudant-sw	London	default	Lite	Provisioned	Cloudant
Cloudant-vx	London	default	Lite	Provisioned	Cloudant

Cloud Foundry Services

Name	Region	CF Org	CF Space	Plan	Service Offering
Continuous Delivery	London	HAMK ICT IoT	IoT dev		

Fig. 2.4 All the service instances from the earlier assignments and tasks.

Depending on whether you are looking at your services in the old Bluemix cloud or in the new Cloud cloud the services will be visible either on address https://console.bluemix.net/dashboard/apps or on address https://cloud.ibm.com/resources .

A Client device can be created by opening the application created in the earlier Device – NodeRED assignment. That can be found under the subtitle Cloud Foundry Applications. In that application please open the …-iotf-service resource.

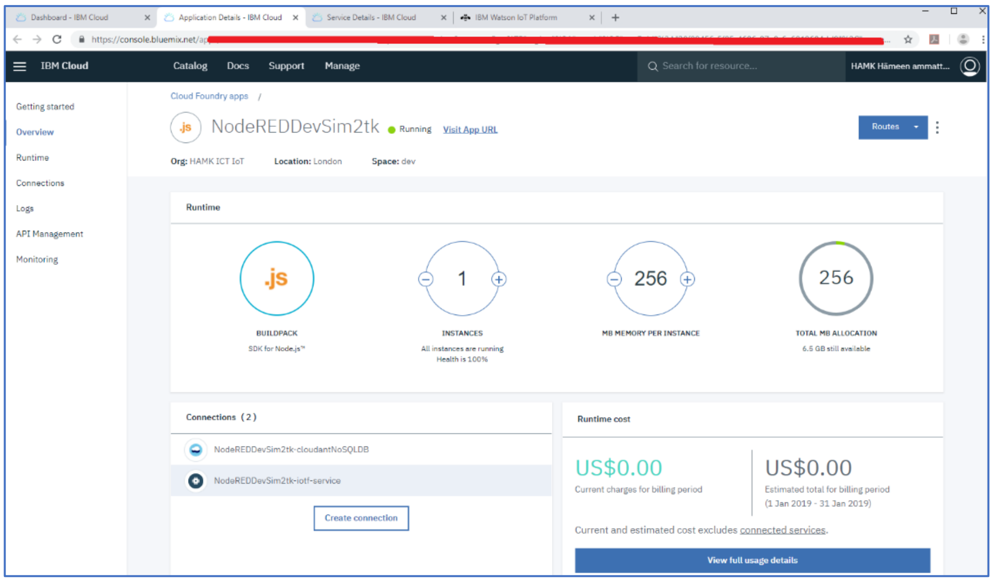


Fig. 2.5 IoT Service

The same resource can be reached by selecting in the Dashboard the subtitle Cloud Foundry Services and the same …. -iotf-service resourse.

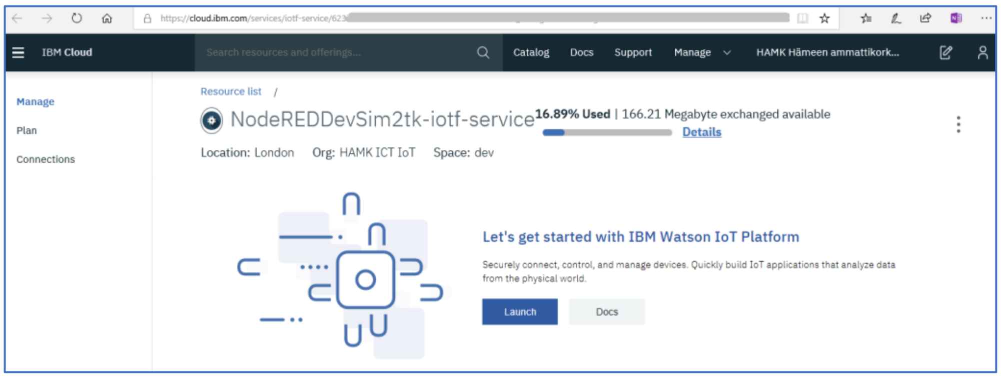

https://cloud.ibm.com/services/iotf-service/623[redacted]
IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
HAMK Hämeen ammattikork...
Manage
Plan
Connections
Resource list /
NodeREDDevSim2tk-iotf-service
16.89% Used | 166.21 Megabyte exchanged available
Details
Location: London Org: HAMK ICT IoT Space: dev
Let's get started with IBM Watson IoT Platform
Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.
Launch
Docs

Fig. 2.6 Please click the Launch button.

When the …. iotf-service opens please select the button Launch. You get into Browse Devices list. For this commissioning task you need to create two new device definitions. Click the Add Device and answer the questions. You can use similar names for the new devices as in this instruction text. Please let the service create automatically the Authentication Token.

The definitions for the two new devices:

Organization ID: 8yyyynp
Device Type: MonitoringClient
Device ID: MonitorC01
Authentication Method: use-token-auth

Authentication Token: BgF?xxxxxxxxQfLvfe

Organization ID: 8yyyynp
Device Type: MonitoringClient
Device ID: MonitorC02
Authentication Method: use-token-auth
Authentication Token: olh8xxxxxxxxN!oJ*W

The yyyy is hiding part of the organisation ID and xxxx is hiding part of the Authentication Token. Please copy and save the device specification and Authentication Token into a separate text file. We will need those later!

3. The testing will be done with Client App communicating according to mqtt protocol

This time we will not use a real device reading the sensor data. We will do the commissioning tests with a MQTT Client App running on a normal Windows computer. The app is the MQTTbox. Other similar apps can be used. MQTT Client apps can be found for Android and iOS smart phones, too.

If you are in the HAMK network in a local computer you can install the app from the Software Center. On your own computer it can be installed from http://workswithweb.com/mqttbox.html . For the installation the administrator rights will not be needed.

Whatever mqtt client app you are using the client needs to be configured with the following parameters:

MQTT Client Name: anything goes
MQTT Client Id: d:8yyyynp:MonitoringClient:MonitorC01
Protocol: mqtt/tcp
Host: 8zzzzp.messaging.internetofthings.ibmcloud.com
Username: use-token-auth
Password: BgF?xxxxxxxxQfLvfe
Append timestamp to MQTT client id: No

Topic to publish: iot-2/evt/Mon01/fmt/json
Payload Type: String/JSON/XML/Character
Payload: { "Opening":"This is Mon01", "fromNetwork":"Home", "fromLogicNetwork":"Home", "AttempNumber":1 }

MQTT Client Name: anything goes
MQTT Client Id: d:8yyyynp:MonitoringClient:MonitorC02
Protocol: mqtt/tcp
Host: 8zzzzp.messaging.internetofthings.ibmcloud.com
Username: use-token-auth
Password: olh8xxxxxxxxN!oJ*W

Append timestamp to MQTT client id: No

Topic to subscribe: iot-2/cmd/monitoringResponce/fmt/json

The xxxx, yyyy, zzzz are hiding part of the ids and tokens.

In the app you can give almost any name to the client. The Client ID must be exactly like described on the details above. In the Client ID you will type the Organization ID and the Device ID you have created earlier.

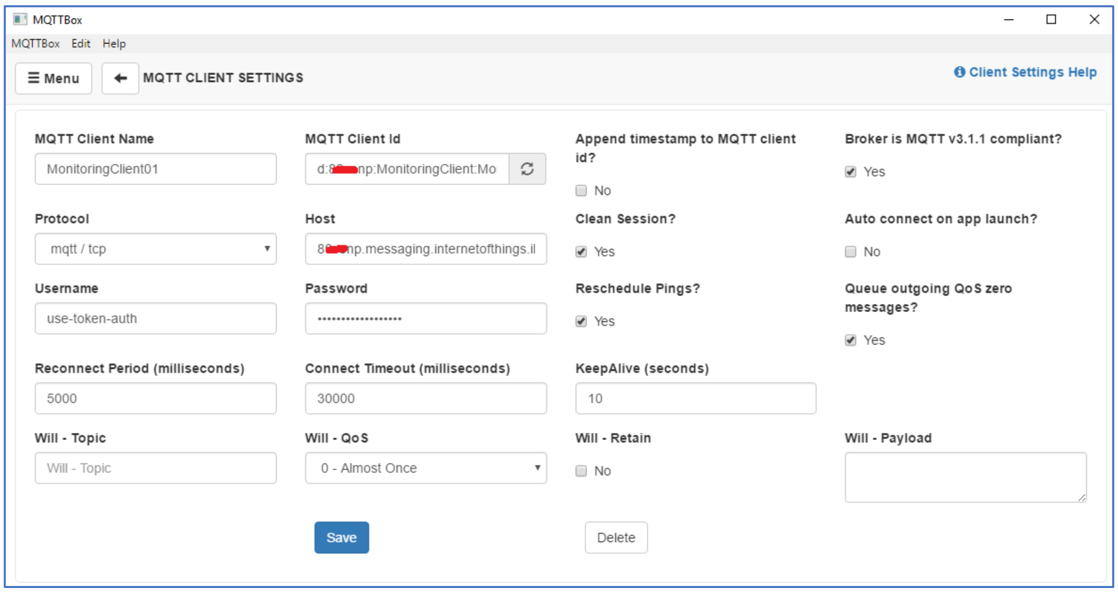
First part of the host address is again the Organisation ID you have created earlier for your organizations IoT Service.

Please remember to remove the selection to add a time stamp in the client ID. Typically a time stamp is added. This prevents the mistakes of accidentally creating two clients with exactly the same name.

In the Topic to Publish field there needs to be a text similar to the one given in the details above. In the details above there is a word Mon01. This one you can replace with an other name to the event.

In the Topic to Subscribe field please add a text similar to the one given in the details above. The word MonitortingResponce you will be using later when creating the answering service. Of course you can use an other name here if you like.

In the picture below the parameters can be seen filled in into the text boxes on the client app MQTTBox.



MQTT Client Name: MonitoringClient01

MQTT Client Id: d:8[redacted]np:MonitoringClient:Mo

Append timestamp to MQTT client id?: No

Broker is MQTT v3.1.1 compliant?: ✓ Yes

Protocol: mqtt / tcp

Host: 8[redacted]np.messaging.internetofthings.il

Clean Session?: ✓ Yes

Auto connect on app launch?: No

Username: use-token-auth

Password: ••••••••••••••••••

Reschedule Pings?: ✓ Yes

Queue outgoing QoS zero messages?: ✓ Yes

Reconnect Period (milliseconds): 5000

Connect Timeout (milliseconds): 30000

KeepAlive (seconds): 10

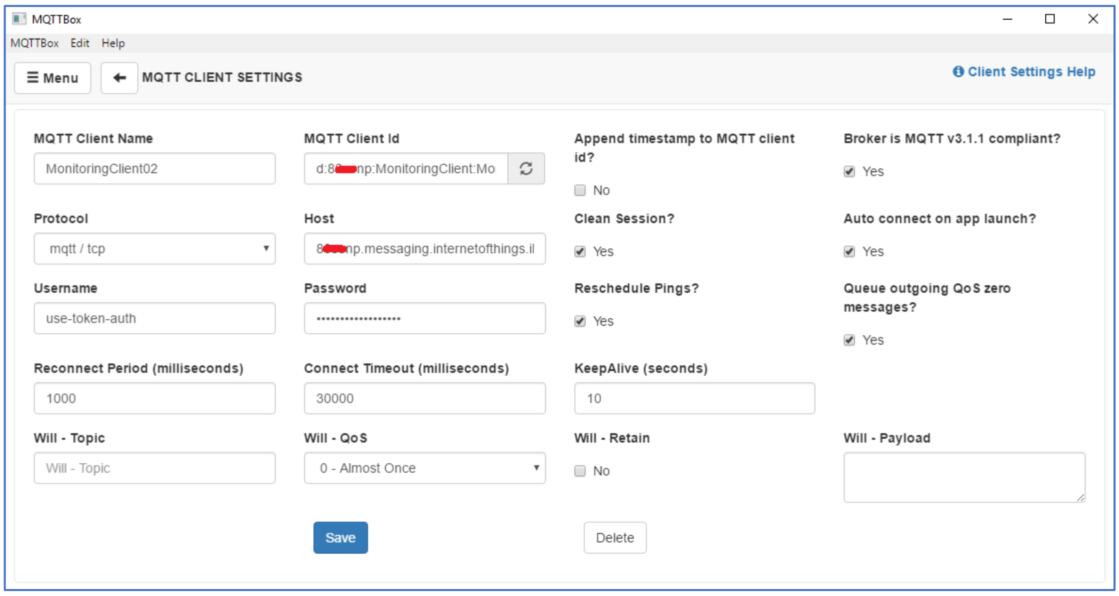
Will - Topic: Will - Topic

Will - QoS: 0 - Almost Once

Will - Retain: No

Will - Payload:

Fig 3.1 The parameters for the first mqtt client.

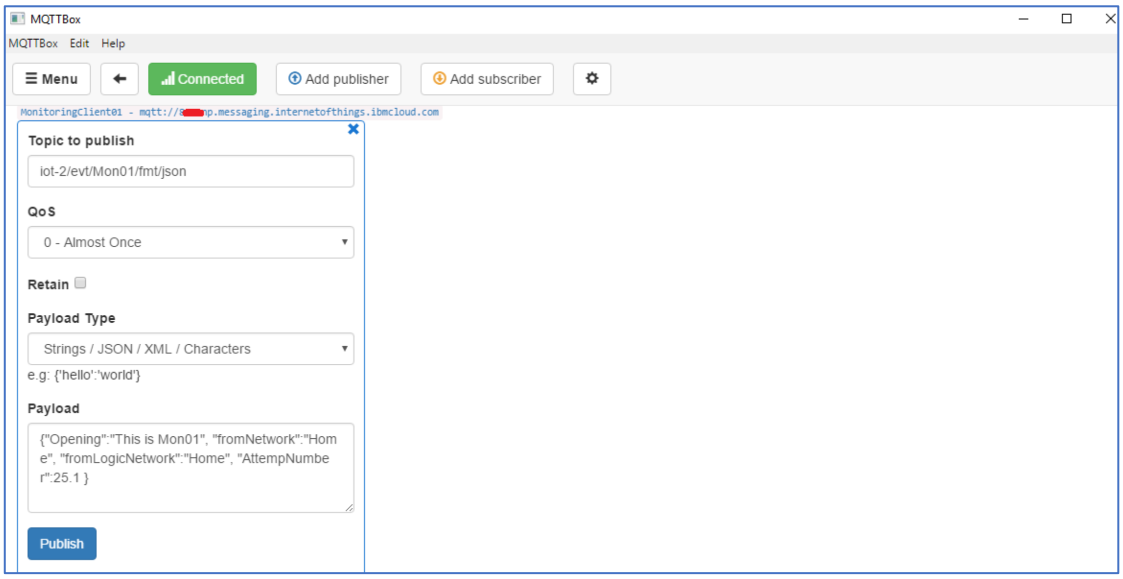

MQTTBox
MQTTBox Edit Help
☰ Menu ← MQTT CLIENT SETTINGS
Client Settings Help

MQTT Client Name	MQTT Client Id	Append timestamp to MQTT client id?	Broker is MQTT v3.1.1 compliant?
MonitoringClient02	d:8[redacted]np:MonitoringClient:Mo	No	✓ Yes
Protocol	Host	Clean Session?	Auto connect on app launch?
mqtt / tcp	8[redacted]np.messaging.internetofthings.i	✓ Yes	✓ Yes
Username	Password	Reschedule Pings?	Queue outgoing QoS zero messages?
use-token-auth	••••••••••••••••••	✓ Yes	✓ Yes
Reconnect Period (milliseconds)	Connect Timeout (milliseconds)	KeepAlive (seconds)	
1000	30000	10	
Will - Topic	Will - QoS	Will - Retain	Will - Payload
Will - Topic	0 - Almost Once	No	

Save Delete

Fig 3.2 The parameters for the second mqtt client

With the first client a message according to mqtt protocol will be sent to the Watson IoT platform. As a content of the message the payload field can be filled in with almost anything as long as it is in the JSON format.


MQTTBox
MQTTBox Edit Help
☰ Menu ← Connected Add publisher Add subscriber

MonitoringClient01 - mqtt://8[redacted]np.messaging.internetofthings.ibmcloud.com

Topic to publish: iot-2/evt/Mon01/fmt/json

QoS: 0 - Almost Once

Retain: ☐

Payload Type: Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload:

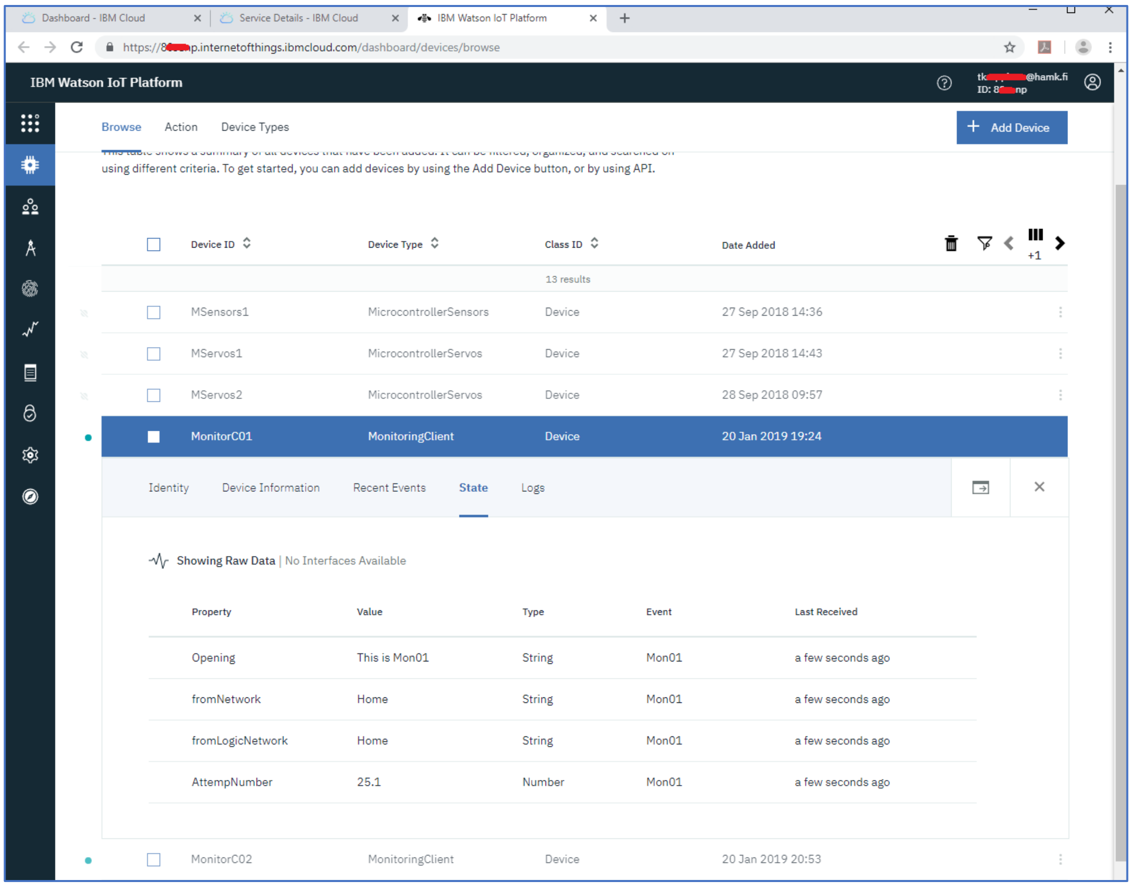
```
{"Opening":"This is Mon01", "fromNetwork":"Home", "fromLogicNetwork":"Home", "AttempNumber":25.1 }
```

Publish

Fig 3.3 The payload of the message will be written in the JSON format.

In the IBM Cloud Watson IoT in the Device View the corresponding device will be opened and the State view opened.

Please send a message from the MQTTBox application by clicking the button Publish.



Dashboard - IBM Cloud | Service Details - IBM Cloud | IBM Watson IoT Platform

https://8[redacted]p.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform

tk[redacted]@hamk.fi
ID: 8[redacted]np

Browse | Action | Device Types | Add Device

using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Device ID	Device Type	Class ID	Date Added
13 results			
MSensors1	MicrocontrollerSensors	Device	27 Sep 2018 14:36
MServos1	MicrocontrollerServos	Device	27 Sep 2018 14:43
MServos2	MicrocontrollerServos	Device	28 Sep 2018 09:57
MonitorC01	MonitoringClient	Device	20 Jan 2019 19:24

Identity | Device Information | Recent Events | State | Logs

Showing Raw Data | No Interfaces Available

Property	Value	Type	Event	Last Received
Opening	This is Mon01	String	Mon01	a few seconds ago
fromNetwork	Home	String	Mon01	a few seconds ago
fromLogicNetwork	Home	String	Mon01	a few seconds ago
AttempNumber	25.1	Number	Mon01	a few seconds ago

MonitorC02	MonitoringClient	Device	20 Jan 2019 20:53

Fig 3.4 In the Watson Dashboard Devices view it is possible to see the JSON formatted payload opened as an object.

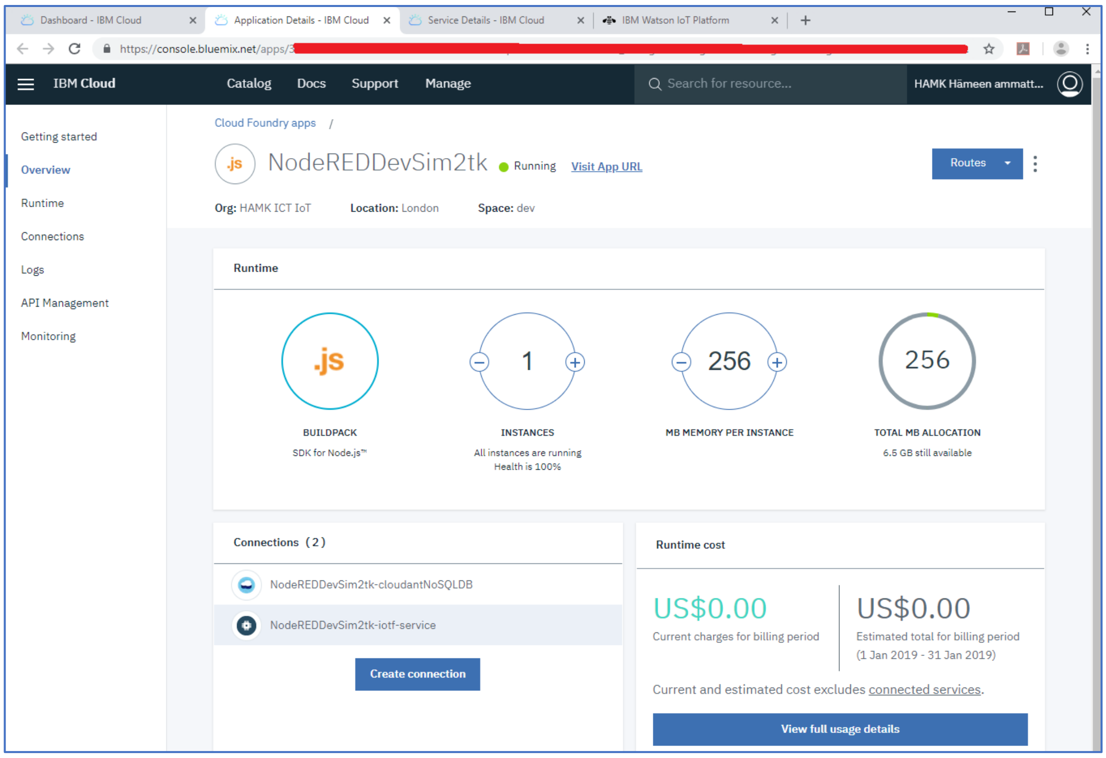
Assignment 1. Please write in the mqtt client in the payload field in the JSON format a number in the following formats: integer number 25, decimal number 25.1, a number in a quote “25.1”. Please try as well some special characters. What are the capabilities in Watson IoT in opening the JSON formatted message?

4. Sending a message from the Watson IoT to the mqtt client; settings for the NodeRED flow

The message data analysis and the answering function will be created in the NodeRED.

The NodeRED editing environment was created already in the earlier commissioning task.

You will be opening the already existing NodeRED flow. In the IBM Cloud Watson Dashboard please open the same Cloud Foundry Application which was used in the previous chapter.


https://console.bluemix.net/apps/3[redacted]

Cloud Foundry apps /

NodeREDDevSim2tk Running Visit App URL

Org: HAMK ICT IoT Location: London Space: dev

Runtime

BUILDPACK	INSTANCES	MB MEMORY PER INSTANCE	TOTAL MB ALLOCATION
.js	1	256	256
SDK for Node.js™	All instances are running Health is 100%		6.5 GB still available

Connections (2)

NodeREDDevSim2tk-cloudantNoSQLDB

NodeREDDevSim2tk-iotf-service

Runtime cost

US$0.00 Current charges for billing period

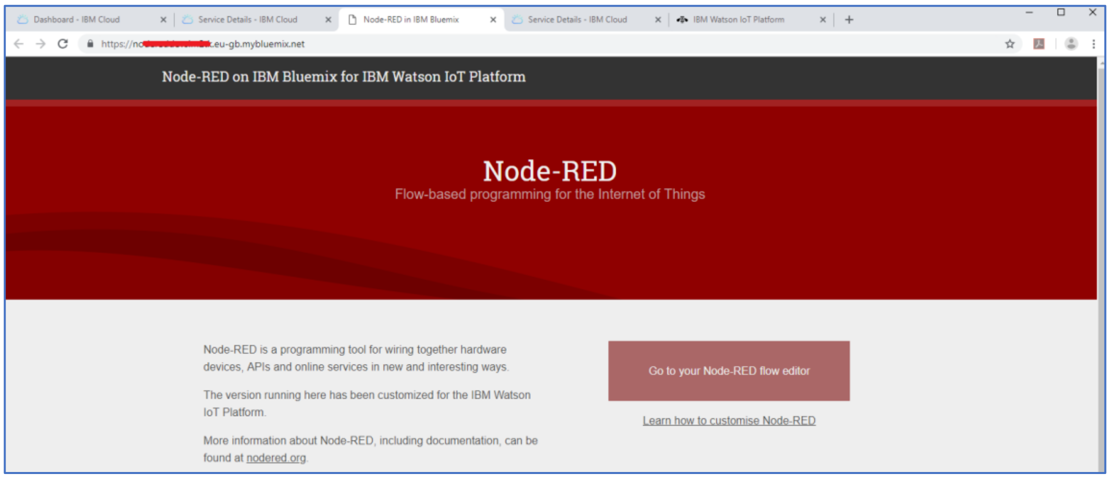
US$0.00 Estimated total for billing period (1 Jan 2019 - 31 Jan 2019)

Current and estimated cost excludes connected services.

Fig 4.1 The Cloud Foundry Application includes a CloudantNoSQLDB database and an iotf-service IoT platform..

The CloudantNoSQLDB includes the NodeRED environment created in the earlier commissioning tasks. If you haven’t got this please go back to the commissioning task ” IBM Watson IoT – T2 – Platform – NodeRED flow”. With that set of instructions you will be able to create a suitable NodeRED flow.

Please open in the connections view the ….-cloudantNoSQLDB and in the next view your application and in the opening window you will be giving the credentials for your NodeRED editor.


https://no[redacted].eu-gb.mybluemix.net

Node-RED on IBM Bluemix for IBM Watson IoT Platform

Node-RED

Flow-based programming for the Internet of Things

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

The version running here has been customized for the IBM Watson IoT Platform.

More information about Node-RED, including documentation, can be found at nodered.org

Dashboard - IBM Cloud
Service Details - IBM Cloud
Node-RED in IBM Bluemix
Service Details - IBM Cloud
IBM Watson IoT Platform
Go to your Node-RED flow editor
Learn how to customise Node-RED

Fig 4.2 Please fill in your username and the password for the NodeRED editor.

Please create a flow similar to the one seen in the picture below. The necessary nodes you will recognize in the menu left by the colour and the small symbol of the node. To make it easier to follow these instructions please use the same names for the nodes as seen in the picture.

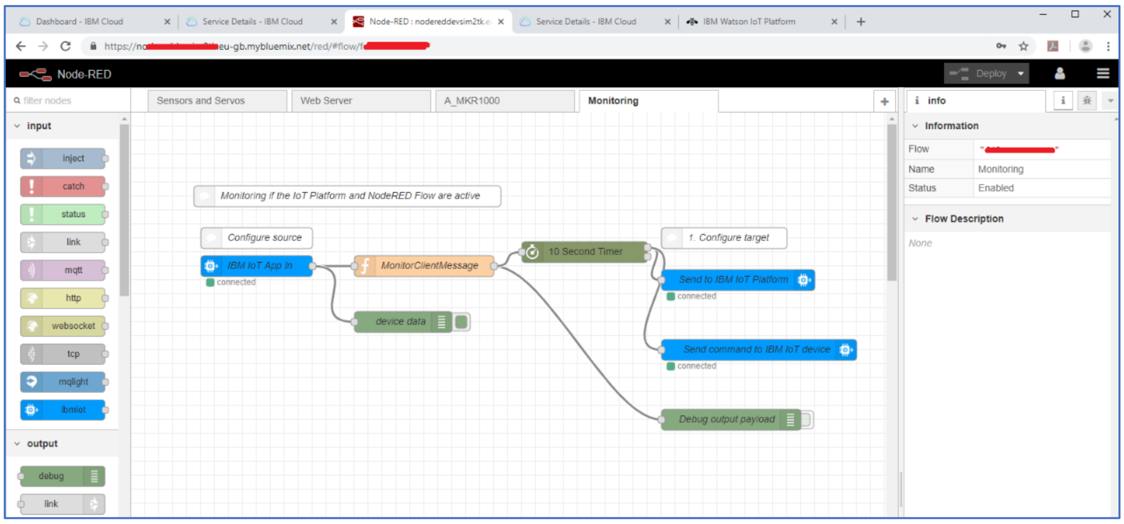
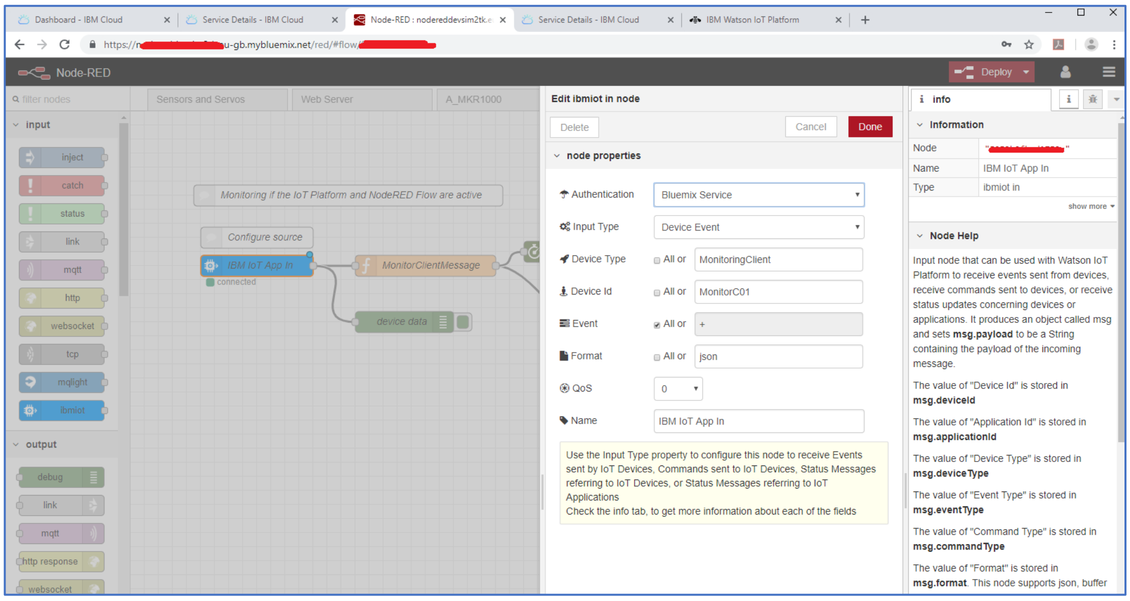


Fig 4.3 A NodeRED flow creating an answer message to the device as a mqtt message.

The leftmost node is the one used for connecting to the IoT Device or to another mqtt client.


Dashboard - IBM Cloud
Service Details - IBM Cloud
Node-RED : nodereddevsim2tk
Service Details - IBM Cloud
IBM Watson IoT Platform
https://n[redacted]u-gb.mybluemix.net/red/#flow/[redacted]
Node-RED
Deploy
Edit ibmiot in node
Delete
Cancel
Done
node properties

Authentication		Bluemix Service
Input Type		Device Event
Device Type	☐ All or	MonitoringClient
Device Id	☐ All or	MonitorC01
Event	☑ All or	+
Format	☐ All or	json
QoS		0
Name		IBM IoT App In

Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to IoT Applications
Check the info tab, to get more information about each of the fields

info

Information

Node	"[redacted]"
Name	IBM IoT App In
Type	ibmiot in

show more

Node Help

Input node that can be used with Watson IoT Platform to receive events sent from devices, receive commands sent to devices, or receive status updates concerning devices or applications. It produces an object called msg and sets **msg.payload** to be a String containing the payload of the incoming message.

The value of "Device Id" is stored in **msg.deviceId**

The value of "Application Id" is stored in **msg.applicationId**

The value of "Device Type" is stored in **msg.deviceType**

The value of "Event Type" is stored in **msg.eventType**

The value of "Command Type" is stored in **msg.commandType**

The value of "Format" is stored in **msg.format**. This node supports json, buffer

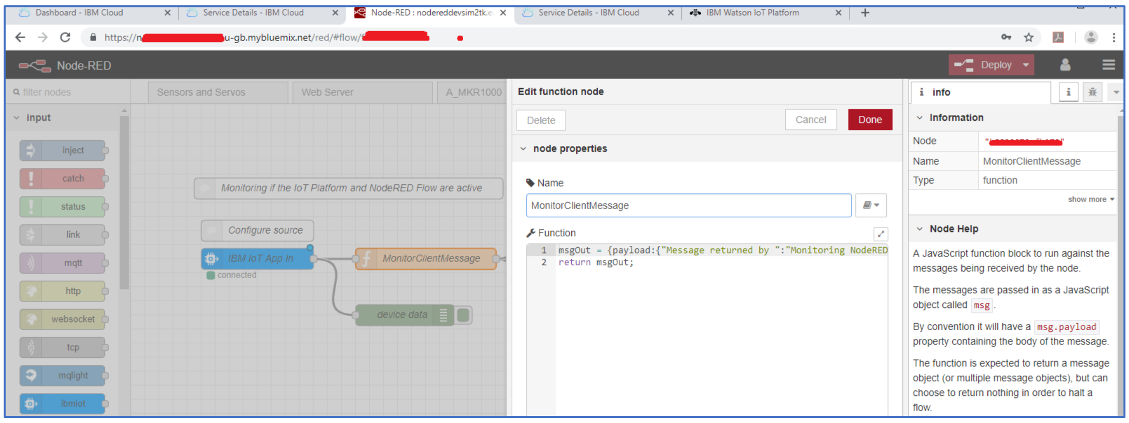
Fig 4.4 The IBM IoT App In node and the parameters.

Opening the node you will see the info frame and the Help text. In the parameter fields all the same information must be filled as earlier in defining the device in the IoT platform:

Authentication: Bluemix Service
Input Type: Device Event
Device Type: MonitorinClient …. Your previously defined Device Type

Device Id: MonitorC01 ….. Your previously defined Device Id
Event: select All
Format: json
QoS: 0
Name: ….. a new name can be given

Similarly please open the function node.



Name: MonitorClientMessage

Function:

```
msgOut = {payload:{"Message returned by ":"Monitoring NodeRED
return msgOut;
```

Node	[redacted]
Name	MonitorClientMessage
Type	function

A JavaScript function block to run against the messages being received by the node.

The messages are passed in as a JavaScript object called `msg`.

By convention it will have a `msg.payload` property containing the body of the message.

The function is expected to return a message object (or multiple message objects), but can choose to return nothing in order to halt a flow.

Fig 4.5 Function node seen opened

As a function you can write:

msgOut = {payload:{"Message returned by ":"Monitoring NodeRED flow","received from device of type ":msg.deviceType," of ID ":msg.deviceId," payload ":msg.payload}};
return msgOut;

For the msgOut a JSON formatted payload will be created.

The object msq consists of the values from the incoming message: msg.deviceType, msg.deviceId, msg.payload.

The node Send to IBM IoT Platform will be opened from top right. Parameters will be given as:

Authentication: Bluemix Service
Output Type: Device Event
Device Type: MonitoringClient …. Your previously defined Device Type
Device Id: MonitorC02 …. Your previously defined Device Id
Event Type: update
Format: json
Data: temp:10 …. this will be replaced by a value created in the function
QoS: 0
Name: Send to IBM IoT Platform … or a new name can be given

And further the node Send command to IBM IoT Device will be opened from top right. Parameters will be given as:

Authentication: Bluemix Service
Output Type: Device Command
Device Type: MonitoringClient …. Your previously defined Device Type
Device Id: MonitorC02 …. Your previously defined Device Id
Command Type: monitoringResponce … In here a type of your own can be written
Format: json
Data: {"data":"data"} …. this will be replaced by a message created in the function
QoS: 0
Name: Send command to IBM IoT device … or a new name can be given

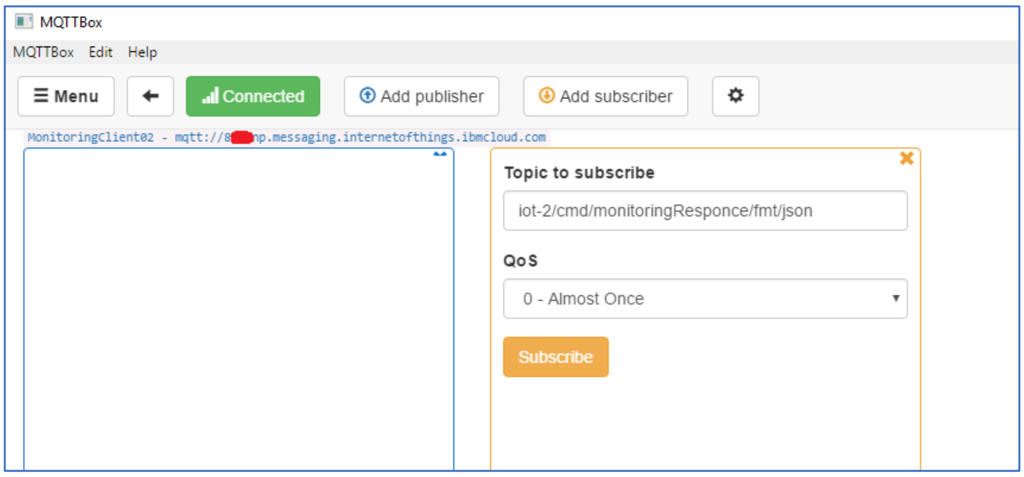
The timer 10 seconds will be add. With this it will be easier to recognize which one is the original message and which one is the command sent as a response.

The nodes and parameters in the NodeRED flow editor are done now.

5. Sending a message from the Watson IoT to the mqtt client; settings for the MQTTBox client application

The operation will be tested with a MQTTBox client application.

Parameters for the subscriber will be written in the second client you created earlier in the MQTTBox. .



MonitoringClient02 - mqtt://8[redacted]np.messaging.internetofthings.ibmcloud.com

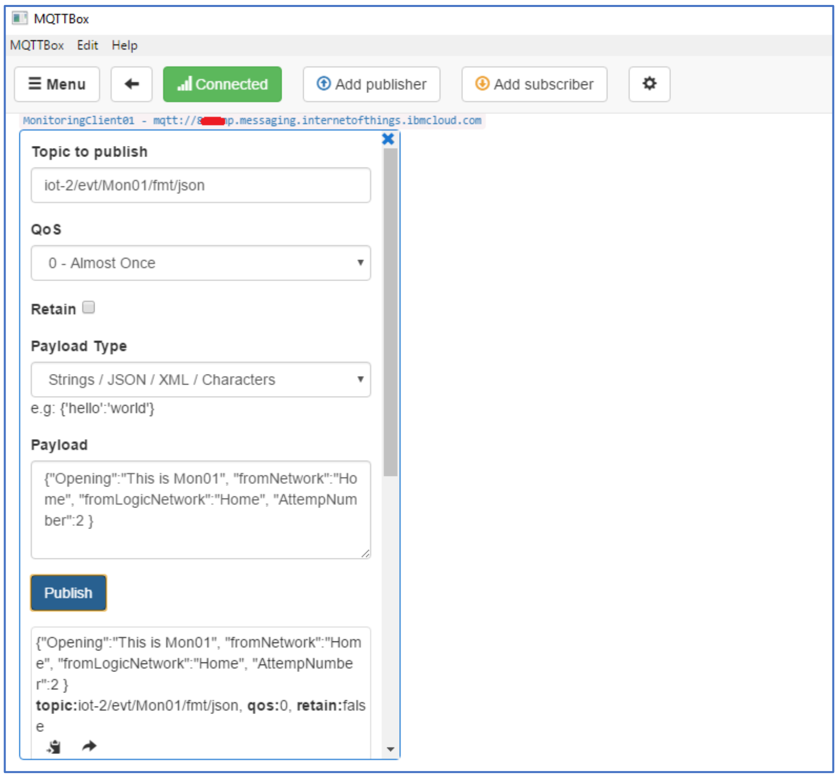
Topic to subscribe
iot-2/cmd/monitoringResponce/fmt/json
QoS
0 - Almost Once
Subscribe

Fig. 5.1 Parameters for the subscriber will be written

In the subscribe settings:

Topic to subscribe: iot-2/cmd/monitoringResponce/fmt/json
QoS: 0

Please open the sending client and send a message by clicking the publish button.


MQTTBox
MQTTBox Edit Help
Menu
Connected
Add publisher
Add subscriber
MonitoringClient01 - mqtt://[redacted]p.messaging.internetofthings.ibmcloud.com

Topic to publish: iot-2/evt/Mon01/fmt/json

QoS: 0 - Almost Once

Retain ☐

Payload Type: Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload

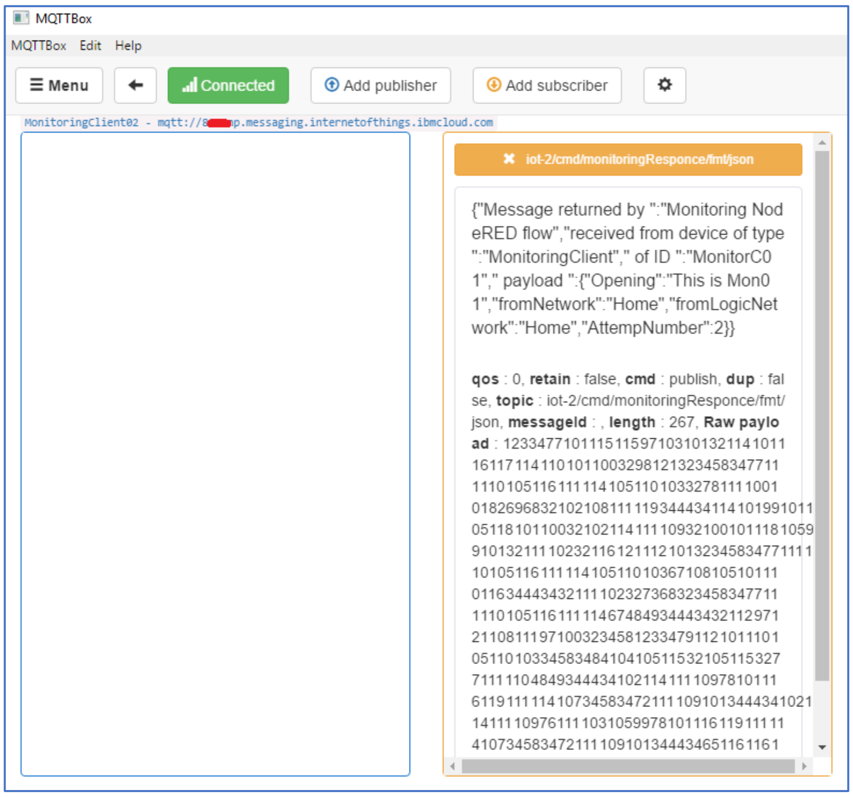
```
{"Opening":"This is Mon01", "fromNetwork":"Home", "fromLogicNetwork":"Home", "AttempNumber":2 }
```

Publish

```
{"Opening":"This is Mon01", "fromNetwork":"Home", "fromLogicNetwork":"Home", "AttempNumber":2 }
topic:iot-2/evt/Mon01/fmt/json, qos:0, retain:false
```

Fig 5.2 A message will be sent .

A delay was defined in the NodeRED flow. You should receive a response after the defined delay.


MQTTBox
MQTTBox Edit Help
Menu
Connected
Add publisher
Add subscriber
MonitoringClient02 - mqtt://[redacted]p.messaging.internetofthings.ibmcloud.com
iot-2/cmd/monitoringResponce/fmt/json

```
{"Message returned by ":"Monitoring NodeRED flow","received from device of type ":"MonitoringClient"," of ID ":"MonitorC01"," payload ":{"Opening":"This is Mon01","fromNetwork":"Home","fromLogicNetwork":"Home","AttempNumber":2}}

qos : 0, retain : false, cmd : publish, dup : false, topic : iot-2/cmd/monitoringResponce/fmt/json, messageId : , length : 267, Raw payload : 1233477101115115971031013211410111611711411010110032981213234583477111101051161111141051101033278111001018269683210210811111934443411410199101105118101100321021141111093210010111810599101321111023211612111210132345834771111010511611111410511010367108105101110116344434321111023273683234583477111101051161111146748493444343211297121108111971003234581233479112101110105110103345834841041051153210511532771111104849344434102114111109781011161191111141073458347211110910134443410211411110976111103105997810111611911111410734583472111109101344434651161161
```

Fig. 5.3 A device command is received.

6. The assignments

Please note the number of symbols in the received message. In our command message the number of symbols is 267.

The length of the message can be a limiting factor when receiving commands with some micro controller and a mqtt subscriber client.

In the Arduino pubsubclient: 90 bytes / https://pubsubclient.knolleary.net/

In the Arduino mqttclient by Joël Gähviler: 128 bytes / https://github.com/256dpi/arduino-mqtt

It is possible to increase the number of characters received. This is done in the above mentioned class libraries by changing a parameter value. Please have a look in the class library documentation.

Assignment 2. Please modify the content of the command message to make the length of the message less than 90 bytes. You need to try a few times. The number of symbols giving certain number of bytes is not that easy to count.

Assignment 3. Please modify the subscriber client settings and the topic definition. The message should be received only from exactly this device type and this device id device. It should receive all command types.