Niagara MQTT with the Azure IoT Hub

- Introduction
- Architecture
- Configuration of Azure IoT Hub
 - Credential Setup
 - Required configuration for the Bluemix Application as a MQTT client
 - Testing the device (publisher configuration)
 - Testing the device (subscription configuration)

Introduction

MQ Telemetry Transport (MQTT), is a lightweight messaging protocol. MQTT is designed for the constrained devices, low-bandwidth, high-latency and the unreliable networks. MQTT uses publish/subscribe (push/pull) messaging transport to help the small footprint Machine to Machine (M2M) and Internet Of Things (IoT) platforms.

The following article demonstrates the use of Niagara MQTT driver to publish and subscribe Niagara points to Azure IoT hub. For details, refer to the link: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-mqtt-support

Architecture



The integration of Azure IoT Hub with the Niagara MQTT Client (AbstractMqttDriverNetwork) as shown below:



Configuration of Azure IoT Hub

Credential Setup

- To setup the credential, see Using the MQTT protocol directly. However, there are few guidelines to setup the credential for Niagara MQTT client.
- As described in the Using the MQTT protocol directly, you have to generate a SAS Token for the device to use it as a
 passphrase/password. The Token can be generated using a DeviceExplorer as described below:
- 1. Connect to the Azure IoT Hob using the DeviceExplorer.
- Check for the IoT Hub Connection String field. It contains connection sting, which is a combination of Hostname and SharedAccessKeyName separated by semicolon (;).
- 3. If required, update the connection sting, click on Update button to update the connection string.

🔡 Device Explore	Twin		×
Configuration	Management Data Messages To Device Call Method on Device		
Connection Ir	formation		
IoT Hub Con	nection String:		
HostName= devices.net;S szN8gB26G4	azure- haredAccessKeyName=iothubowner,SharedAccessKey=Z1lt0qwVdUVG3NNaqK5mBR TFhEHXYCzYY=		
Protocol Gat	ewav HostName:		
	,		
Undat			
opuar			
Shared Acce	ss Signature		
Key Name	iothubowner		
Key Value	Z1lt0qwVdUVG3NNaqK5mBRszN8gB26G4TFhEHXYCzYY=		
Target	GodrejIOTHub.azure-devices.net		
TTL (Days)	365 Generate SAS		
,			

4. Click on the Management tab, it shows the list of all the available devices from the IoT hub.

Configu Actio	ns Create	ement Data Me Refresh	essages To Devid	ce Call Method o	on Device	Token 1	win Props.
Tota	l: 11						
	ld	PrimaryKey	SecondaryKey	PrimaryThumbl	SecondaryThu	ConnectionStrin	ConnectionS
		gt1KbJpbMsi	4qfkpYjfyy62			HostName=G	Disconnecte
		jDjJcqo/N4p	V1VrPds6Ud			HostName=G	Disconnecte
		RwsYJjAymc	NXEvhzil6p0c			HostName=G	Disconnecte
				D81ADEFED		HostName=G	Disconnecte
				A031C46782		HostName=G	Disconnecte
				CE8966BFCD		HostName=G	Disconnecte
				7D70E599158		HostName=G	Disconnecte
				D4DE20D05	2CDCA6DB7	HostName=G	Disconnecte
	ЮТ			C7C672DDC3		HostName=G	Disconnecte
	Microsoft			948E1652586		HostName=G	Disconnecte
۱.	Test	emPyixFydR	hJvDaM8vyO			HostName=G	Disconnecte
*							

- 5. Click on the SAS Token button, the SASTokenForm window pops up.
- 6. In the SASTokenForm window, select the required device and set the TTL (Time to Live) to generate the token (password).
- 7. In case of the Azure IoT hub, generate the Username and Password .

The URI/Client ID is the Username.(Refer to the attached email for example).

onfiguration	Management	Data	Messages To Device	Call M	ethod or	n Dev	ice		
Antiona									
🔡 SASToker	nForm			-		×	SAS	Token	Twin Props.
DeviceID	Test					\sim	-		
DeviceKeys	emPyixFydR+tE	CDMCL6	'2iWBBiHRzhp2Xlgah/PldHw=	=		\sim	donyThu	ConnectionStrip	ConnectionS
	TTL (Dave)	0				÷.	luary mu	HeatNamor	Disconnections
	TTE (Days)							HostName=G	Disconnected
							-	HostName=G	Disconnecter
							-	HostName=G	Disconnecter
							-	HostName=G	Disconnected
							-	HostName=G	Disconnected
								HostName=G	Disconnected
							6DB7	HostName=G	Disconnected
								HostName=G	Disconnected
,								HostName=G	Disconnected
								HostName=G	Disconnected
	Genera	ate	Done						
		_		_	_		1		

The SharedAccessSignature is the Password for the device, see the highlighted text as shown below.

SASToker	ıForm				-	×	SAST	Token	Ти	in Props	
DeviceID	Test					~					_
20110012	TOSE					_					
DeviceKeys	emPyixFydR+t	ECDMCL6	2iWBBiHRz	hp2Xlgah/Pld	Hw=	~	1 71	0 1 0		0 1	_
	TTL (Dave)	0					dary i hui	ConnectionS	trin	Connectio)n:
	TTE (Days)							HostName=0	· ·	Jisconneo	ne
HostNam	a =/	37109-						HostName=0	1 L	Visconnec	te
devices.r	et;DeviceId=Tes	t;SharedA	cessSignatu	re=SharedAcc	essSignature			HostName=0	і с)isconner	-te
sr= 2FE8W1	azure-de. kihx8WWcpoVD	vices.net vMtsPrgyB	32Fdevices% BkrgM%3D&	2FTest&sig=Fil e=151116786	8V ImgjsxtO% 0			HostName=0	с)isconner	
								HostName=0	с)isconneo	te
								HostName=0)isconneo	te
							6DB7	HostName=0	i C	Disconneo	te
								HostName=0	i C	Disconneo	te
, ,								HostName=0	i C	Disconneo	te
								HostName=0	i C	Disconneo	te
	Gene	rate		Done							
L											

Note: Make sure that you copy the highlighed test as it is. For details, see Azure Documentation, page 334.

8. Configure the Niagara MQTT device/client.

Required configuration for the Bluemix Application as a MQTT client

Broker IP: xxx.azure-devices.net (This is the host name of your IoT Hub, xxx represents the name)

Client ID: Test (Client Id that is created in the IoT Hub, alternatively you can refer to the Management tab on the device explorer) UserName (API Key): {iothubhostname}/{device_id}/api-version=2016-11-14 (The iothubhostname is same as Broker IP and the device_id is same as Client ID)

Password (Token): SharedAccessSignature=SharedAccessSignature sr=xxx.azure-devices.net%2Fdevices%2FTest&sig=Fi8VTmgjsxtO%2FE8 W1Xihx8WWcpoVDyMtsPrgyB8krgM%3D&se=1511167860

devices/{device_id}/messages/events/ - Publisher (device_id is same as Client ID))

devices/{device_id}/messages/devicebound/# - Subscriber

Testing the device (publisher configuration)

Once the device configuration is done, you can test the publisher with the help of Device Explorer.

- 1. Click on the **Data** tab in the Device Explorer, Monitoring section appears.
- In the Monitoring section, select the required device the Device ID field to publish the messages. Note: The Event Hub details will be prepopulated with the Hub name.
- 3. When you click on the **Monitor** button, the Even Hub Data section shows the messages received by the IoT Hub that is published from the configured device.

Device Explor	rer Twin							-		
onfiguration	Management	Data	Messages To [Device	Call Method on Dev	rice				
Monitoring										
Event Hu	ıb:]
Device II	D: Test								~	
Start Tim	ne: 10/26/20)17 14:45	38]
Consum	er Group: \$Def	fault			Enable					
Mo	nitor		Cancel		Clear		Shov	v system p	propertie	s
Event Hub I 11/8/2017 9 'mqtt-retain 11/8/2017 9 'mqtt-retain 11/8/2017 9 'mqtt-retain	Data :20:22 PM> Dev :' 'true' :' 'true' :20:52 PM> Dev :' 'true'	ice: [Tes ice: [Tes ice: [Tes	tt], Data:[88.5123 tt.iftBalt], Data:[4 tt], Data:[3.09103	085825 1.58585 732067	0786]Properties: 346903406]Propertie 3943]Properties:	S:				

Testing the device (subscription configuration)

Once the device configuration is done, you can test the subscription with the help of Device Explorer.

- 1. Click on the **Messages To Device** tab in the Device Explorer, Send Message to Device section appears. **Note:** The IoT Hub details will be prepopulated with the IoT Hub name.
- 2. In the Send Message to Device section, select the required device in the Device ID field to send the message.
- 3. Key the messages you want to send and click on the Send button.

🔛 Device Explore	r Twin					—		×
Configuration	Management [Data	Messages To Device	Cal	Method on Device			
Send Messa	age to Device:							
IoT Hub:	GodrejlOTHul	b						
Device ID:	Test						,	~
Message:	10							
	🗌 Add Time S	Stamp	🗌 Monitor Fe	edba	ck Endpoint			
Properties:								
Ke	у				Value			
*								
								-
S	end		Clear					
Outrut		5						
Sent to Devi	ce ID: [Test] Mes	ssage."	10" message Id: d430g	0b9-	4c98-4110-8a27-0ab64573258c			
Contro Devi	ee ib. [resignies	Jougo.	in , message ia. a isot	.000				

- 4. Verify the subscription part from the Niagara after configuration of the device:
- Setup a subscription point in the Niagara MQTT device. See Abstract MQTT Driver Guide to create the subscription point.
 Verify the point value and make sure that you receive the message from device explorer after subscribing.