

# Azure IoT Hub with RUT routers

---

## Table of Contents

Azure account creation .....	2
Managing Azure services.....	2
Configuring RUT955 Azure IoT Hub.....	5
Checking if Data reaches Azure IoT Hub .....	6
Setting router to Forward MQTT messages/commands to Azure IoT Hub.....	9
Checking if MQTT messages are being forwarded to Azure IoT Hub.....	10
Setting Modbus Data to Server to Azure IoT Hub .....	10
Checking if Modbus Data to Server is sent to Azure IoT Hub.....	11

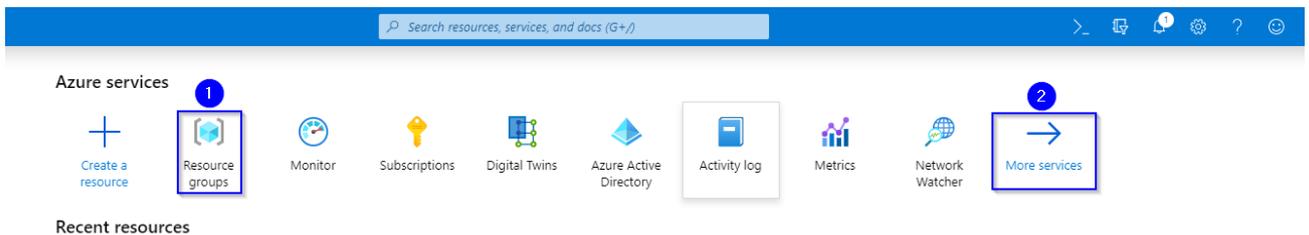
## Azure account creation

Visit <https://azure.microsoft.com/en-us/> and create an account that will suit your needs, for testing purposes we will be using free Azure account.

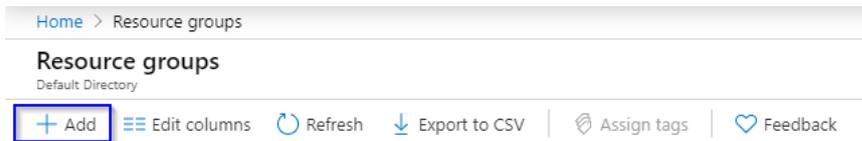
## Managing Azure services

First you will want to create a Resource group for easier management of resources that you will add later. In Microsoft Azure home page:

1. Select **Resource groups**
2. If it is not in very first page, click **More services** and locate it there.



In new window, select **Add**.



And then finish creating yours Resource group

1. Select your subscription, we are using **Free Trial** for this test.
2. Name your group
3. Finally, choose server location for meta data. We will choose one of the closest available locations **North Europe** and will use it during test where available.

**Project details**

Subscription \* ⓘ 1 Free Trial ✓

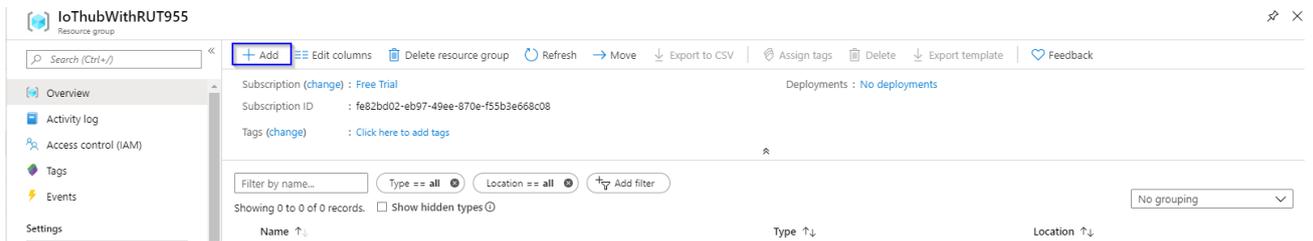
Resource group \* ⓘ 2 IoThubWithRUT955 ✓

**Resource details**

Region \* ⓘ 3 (Europe) North Europe ✓

At this moment we will skip adding Tags since we will be able to do that later if needed, so simply press Review + create at the bottom of screen and then click Create to finish setup.

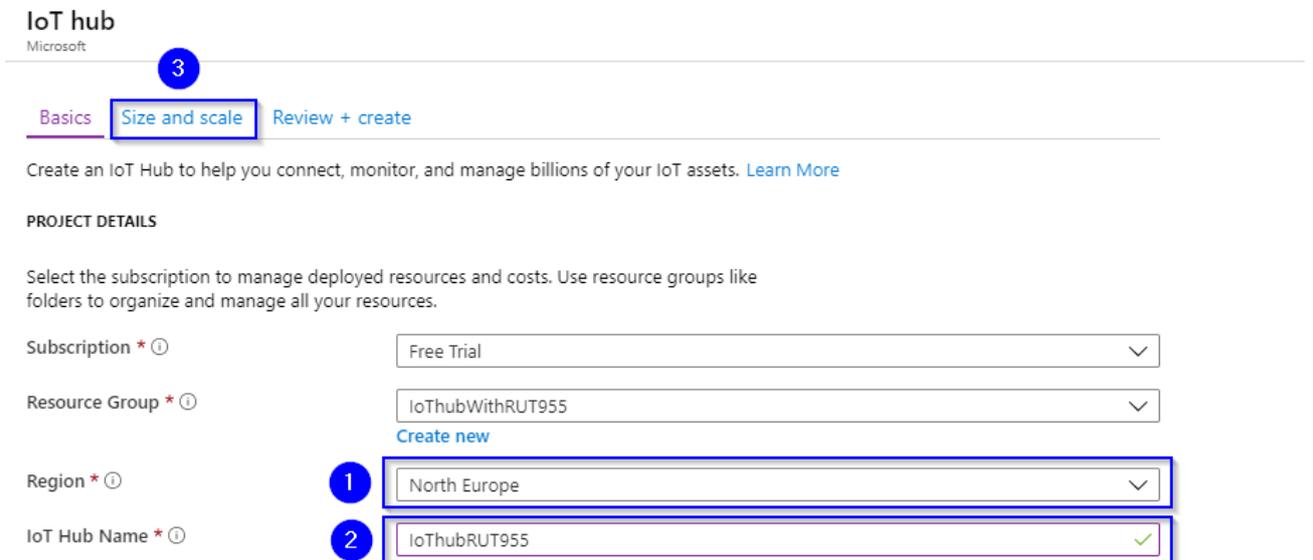
You will be redirected to Homepage, there click on **Resource groups**. You should see yours newly created group, select it, and press **Add**.



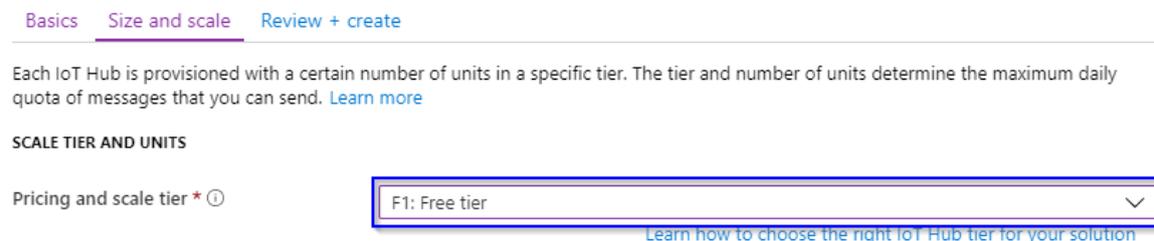
Select **Internet of Things** or simply search **IoT Hub** and press **Create**.

We leave default subscription and resource group and choose:

1. Region – North Europe as before
2. Create a name for IoT Hub
3. Then go to **Size and scale tab**



For testing purposes, we are using **F1: Free tier**

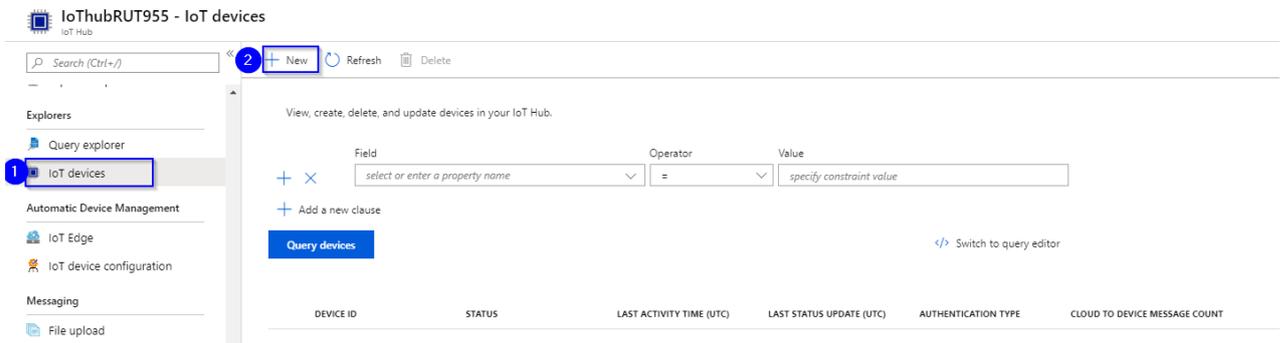


And finally, at the bottom of the screen **Review + create >> Create**

Wait until resource deploys and press **Go to resource**

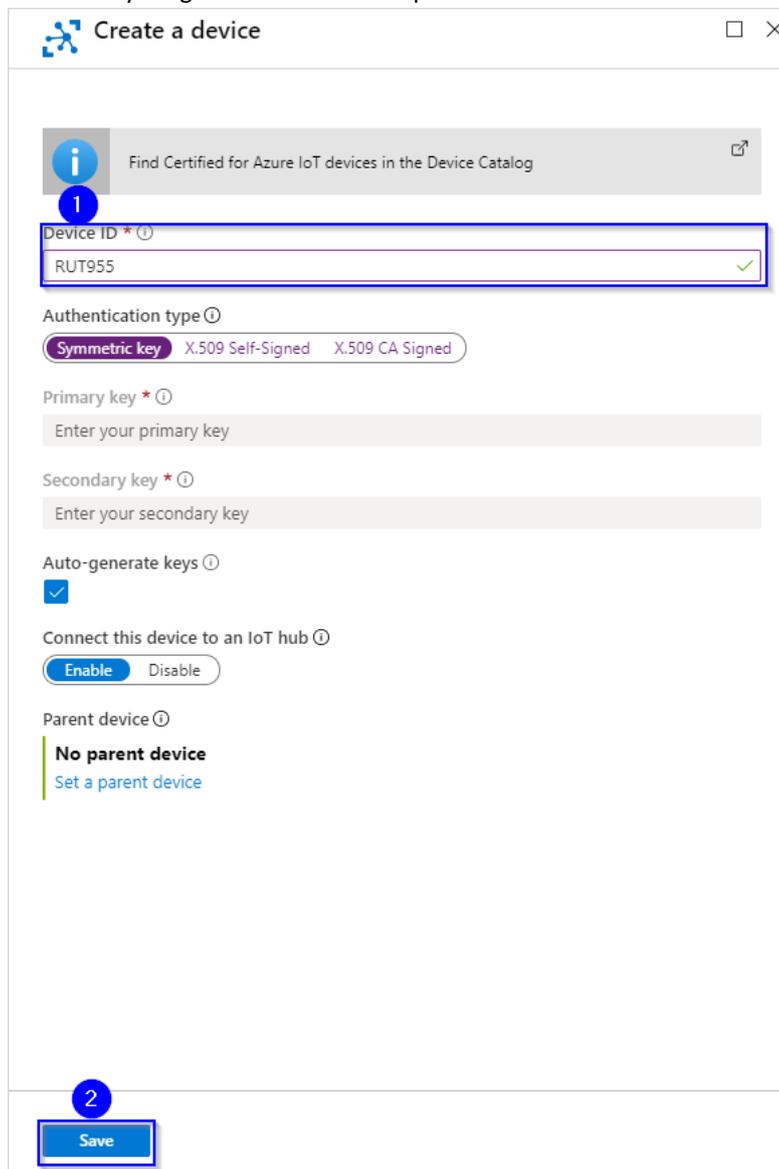
Inside IoT Hub list:

1. Scroll down to **Explorers** and select **IoT devices**
2. Press **New**



In new device creation

1. Enter Device ID
2. Leave everything else on default and press **Save**



After you finish creation, you will be redirected back to IoT devices select yours newly created **Device ID**

✓	DEVICE ID	STATUS	LAST ACTIVITY TIME (UTC)	LAST STATUS UPDATE (UTC)	AUTHENTICATION TYPE	CLOUD TO DEVICE MESSAGE COUNT
✓	RUT955	Enabled	--	--	Sas	0

In your device window you will find information needed to connect RUT devices to Azure IoT Hub. For now, we will only need connection string. Copy Primary Connection string by pressing copy icon next to it.

## Configuring RUT955 Azure IoT Hub

First open router WebUI, go to **System > Package Manager** and install Azure IoT Hub package

Azure IoT Hub	-	0.0.2 (372 KB)	Available	<button>Install</button>
---------------	---	----------------	-----------	--------------------------

Now navigate to **Services > IoT Platforms**

1. Select **Enable Azure IoT Hub monitoring**
2. Paste previously copied Connection String (For this test we leave other values as default)
3. Select what kind of information you want to send to Azure IoT Hub
4. Press **Save**

## Checking if Data reaches Azure IoT Hub

From router side, connect to it with SSH client and write in command **azure\_iotHub** and press Enter

```
root@Teltonika:~# azure_iotHub
```

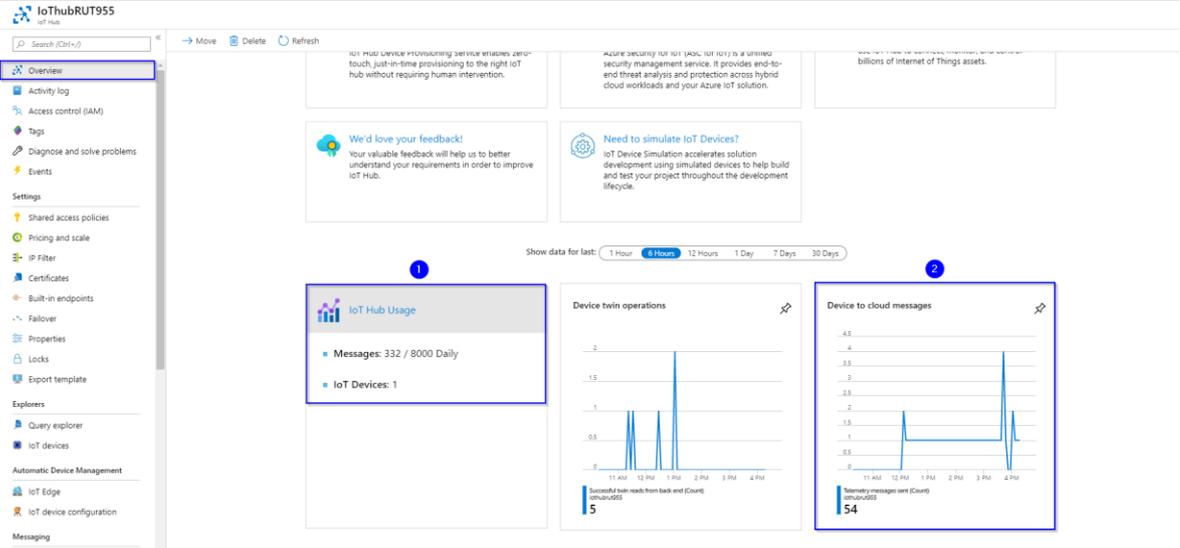
You should get answer that looks something like that, depending on what information you chose to send.:

```
root@Teltonika:~# azure_iotHub
This sample simulates a Chiller device connected to the Remote Monitoring solution accelerator
{
  "ip": "84.15.123.1",
  "bytes_sent": "11316700",
  "bytes_received": "664240493",
  "connection_state": "connected",
  "signal": "-57"
}Confirmation callback received for message 1 with result IOTHUB_CLIENT_CONFIRMATION_OK
Device Twin reported properties update completed with result: 204
^C
root@Teltonika:~#
```

From Azure IoT Hub side you can check if it receives data. Go to IoT Hub that you created previously. Select **Overview**, there you can see:

1. How many devices are connected to hub, and how many messages it sent during chosen period of time.
2. Device to cloud messages, that your router is sending.

It should look something like that if IoT Hub is receiving data.



To capture logs you will need Device Explorer for IoT Hub Devices. For Windows you can get here: <https://github.com/Azure/azure-iot-sdk-csharp/releases/tag/2019-1-4>

Scroll down to Assets, download and install SetupDeviceExplorer.msi

▼ Assets 12

Microsoft.Azure.Devices.1.17.2.symbols.nupkg	876 KB
Microsoft.Azure.Devices.Client.1.19.0.symbols.nupkg	1.28 MB
Microsoft.Azure.Devices.Provisioning.Client.1.2.2.symbols.nupkg	46.7 KB
Microsoft.Azure.Devices.Provisioning.Security.Tpm.1.1.4.symbols.nupkg	39.6 KB
Microsoft.Azure.Devices.Provisioning.Service.1.3.1.symbols.nupkg	174 KB
Microsoft.Azure.Devices.Provisioning.Transport.Amqp.1.1.5.symbols.nupkg	89.3 KB
Microsoft.Azure.Devices.Provisioning.Transport.Http.1.1.4.symbols.nupkg	82.3 KB
Microsoft.Azure.Devices.Provisioning.Transport.Mqtt.1.1.6.symbols.nupkg	78.8 KB
Microsoft.Azure.Devices.Shared.1.15.2.symbols.nupkg	103 KB
SetupDeviceExplorer.msi	2.4 MB
Source code (zip)	
Source code (tar.gz)	

Now you will need connection string of yours Azure IoT Hub, **Not device**. Navigate to IoT hub in your browser, then:

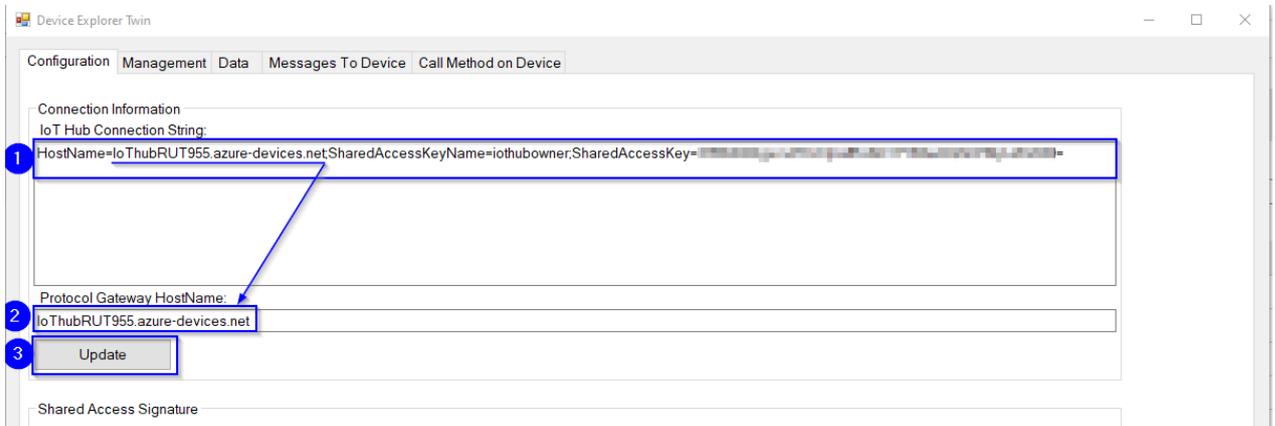
1. Click **Shared access policies**
2. Next choose **iothubowner**
3. And copy **Connection string – primary key**

The screenshot shows the Azure IoT Hub management console for 'IoThubRUT955'. The left sidebar has 'Shared access policies' selected. The main area shows a table of policies with 'iothubowner' selected. The right-hand pane shows the details for the 'iothubowner' policy, including permissions and shared access keys. The 'Connection string—primary key' is highlighted with a blue box and a circled '3'.

Policy	Permissions
iothubowner	registry write, service connect, device connect
service	service connect
device	device connect
registryRead	registry read
registryReadWrite	registry write

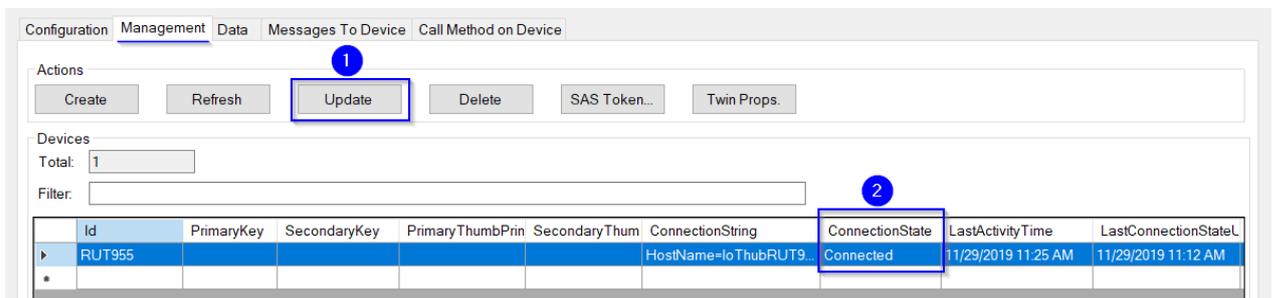
After that go back to Device Explorer:

1. In Configuration tab paste in Connection string that you just copied
2. Copy HostName part from connection string and paste it in **Protocol Gateway HostName**
3. Click **Update**



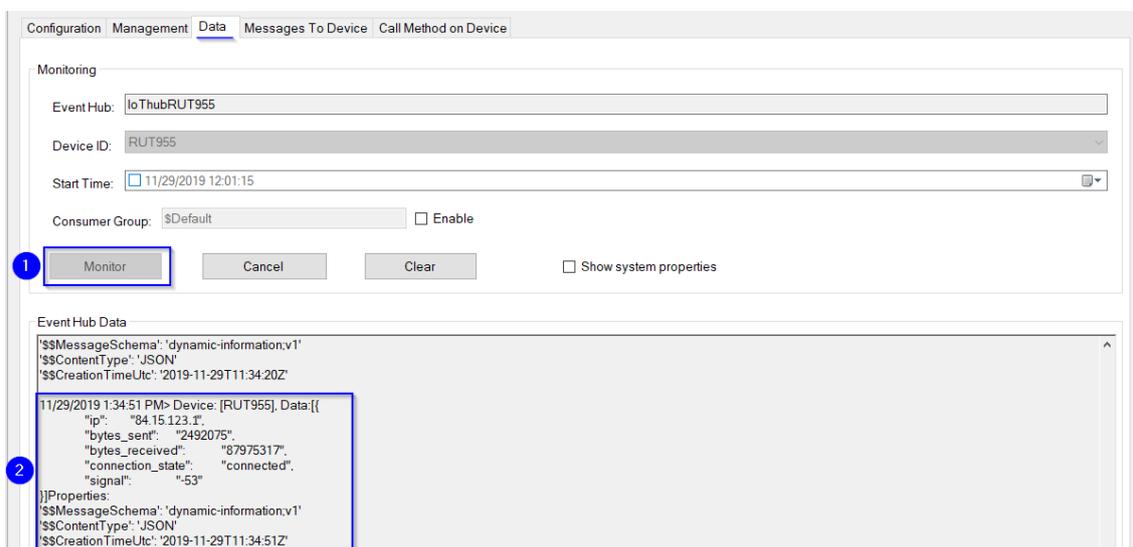
Open Management tab

1. Click Update
2. You should see your device in the list below and **Connection state**



Go to Data tab

1. Click monitor, and wait for **Event Hub Data** to update (Depends on yours chosen interval)
2. Messages like this should start appearing.

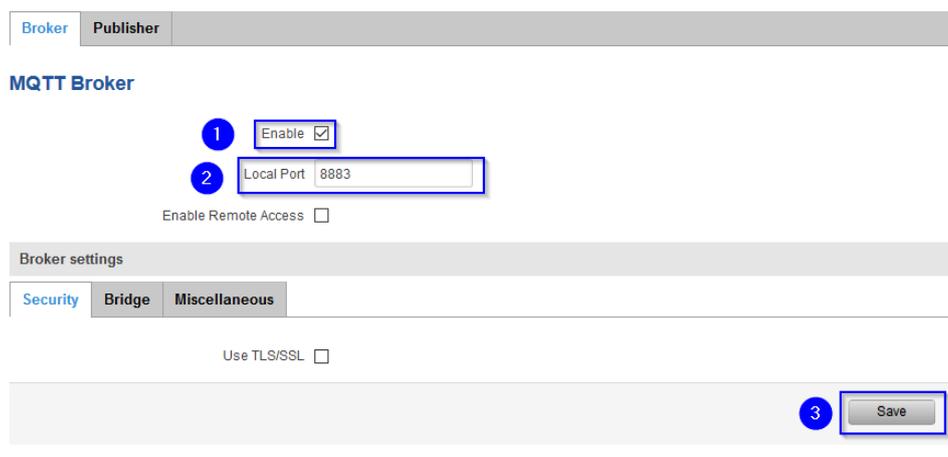


## Setting router to Forward MQTT messages/commands to Azure IoT Hub

First you will need MQTT broker to subscribe to, for testing purposes we will set MQTT Broker in same router, and will use PC from LAN to sent MQTT messages.

Go to Services > MQTT

1. Click **Enable**
2. Use same port in MQTT Broker and Azure IoT Hub settings.
3. Press Save

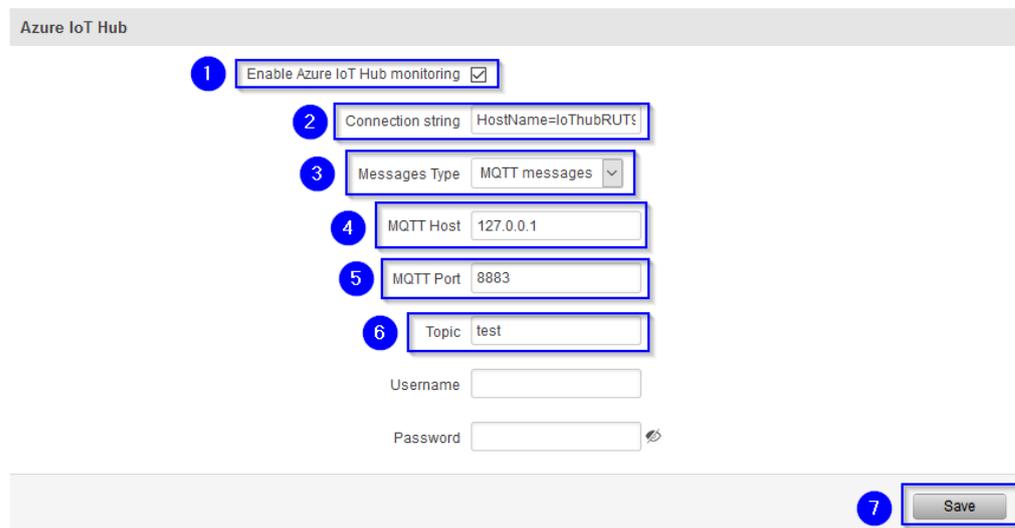


The screenshot shows the MQTT Broker configuration page. At the top, there are tabs for 'Broker' and 'Publisher'. Under the 'MQTT Broker' section, there is a checkbox for 'Enable' (checked) and a text input for 'Local Port' (8883). Below this is a checkbox for 'Enable Remote Access' (unchecked). Under the 'Broker settings' section, there are tabs for 'Security', 'Bridge', and 'Miscellaneous'. Under the 'Miscellaneous' tab, there is a checkbox for 'Use TLS/SSL' (unchecked). At the bottom right, there is a 'Save' button.

Go to Service > IoT Platforms

1. Enable monitoring
2. Use same Connection string as before (GSM values configuration)
3. Messages Type choose **MQTT messages**
4. Enter MQTT Host address, we are using 127.0.0.1 since our broker is set up on same router.
5. Port **Same as MQTT Broker**
6. And Topic under which router will subscribe to MQTT Broker
7. Press Save, we will not need username or password.

### Azure IoT Hub Settings



The screenshot shows the Azure IoT Hub Settings configuration page. At the top, there is a tab for 'Azure IoT Hub'. Under this tab, there is a checkbox for 'Enable Azure IoT Hub monitoring' (checked). Below this are several text input fields: 'Connection string' (HostName=IoTHubRUT...), 'Messages Type' (MQTT messages), 'MQTT Host' (127.0.0.1), 'MQTT Port' (8883), and 'Topic' (test). Below these are fields for 'Username' and 'Password'. At the bottom right, there is a 'Save' button.

## Checking if MQTT messages are being forwarded to Azure IoT Hub

Connect to router with SSH, in logread you should see Router establishing connection to Azure IoT Hub:

```

root@Teltonika-RUT955:~# logread -f
Fri Nov 29 14:17:03 2019 user.info Azure IoThub: Creating IoThub handle
Fri Nov 29 14:17:03 2019 user.info Azure IoThub: Subscribing to: test
Fri Nov 29 14:17:04 2019 local1.info gsmd[2331]: gsmd send: 'AT+QCSQ' (8)
Fri Nov 29 14:17:04 2019 local1.info gsmd[2331]: gsmd get: '+QCSQ: "LTE",58,-89,157,-12' (27)
Fri Nov 29 14:17:04 2019 local1.info gsmd[2331]: gsmd send: 'AT+CREG?' (9)
Fri Nov 29 14:17:04 2019 local1.info gsmd[2331]: gsmd get: '+CREG: 2,1,"0078","0126A16",7' (29)
Fri Nov 29 14:17:04 2019 user.info Azure IoThub: The device client is connected to iotHub
Fri Nov 29 14:17:06 2019 local1.crit luci-reload[21778]: START==1
Fri Nov 29 14:17:06 2019 local1.crit luci-reload[21778]: IF
  
```

Open terminal and publish to MQTT Broker message with previously chosen topic. For our example we are using example: `mosquitto_pub -h 192.168.1.1 -p 8883 -m 'testing Azure MQTT messages' -t test`

```

~$ mosquitto_pub -h 192.168.1.1 -p 8883 -m 'testing Azure MQTT messages' -t test
  
```

If everything was configured correctly in Device Explorer Data tab you should receive message like:

```

11/29/2019 1:59:33 PM> Device: [RUT955]. Data:{
  "value": "testing Azure MQTT messages"
}Properties:
'$$MessageSchema': 'dynamic-information:v1'
'$$ContentType': 'JSON'
'$$CreationTimeUtc': '2019-11-29T11:59:32Z'
  
```

This means our router Forwards MQTT messages to Azure IoT Hub.

## Setting Modbus Data to Server to Azure IoT Hub

First go to Services > IoT Platforms and disable it.

Second you will need to configure Modbus Master and Slave, for this example we already setup TCP Master and Slave to send data to IoT Hub.

After that go to Services > Modbus > Modbus Data to Server and add **New modbus data sender**

1. Choose **Azure MQTT** Protocol
2. Paste device **Connection String** from Azure IoT Hub
3. And enter Period, how often data will be sent
4. Click **Add**

**New modbus data sender**

Protocol	URL / Host / Connection string	Period	
1 <input type="text" value="Azure MQTT"/>	2 <input type="text" value="HostName=IoTHubRUT955"/>	3 <input type="text" value="20"/>	4 <input type="button" value="Add"/>
			<input type="button" value="Save"/>

## In Advanced Sender Settings

1. Select Enable
2. Enter **Device ID** as Name (copy it from Connection String)
3. Leave everything else on default and press Save

### Advanced sender settings

Here you can configure advanced settings for the data sender

**Data sender configuration**

1  Enabled

2  RUT955

Protocol

JSON format 
Modbus slave ID - %i  
Modbus slave IP - %p  
Date (Linux timestamp) - %t  
Date (Day/Month/Year Hour:Minute:Second) - %d  
Start register - %s  
Register data - %a

Segment count

URL / Host / Connection string

Period

Data filtering

Retry on fail

3

## Checking if Modbus Data to Server is sent to Azure IoT Hub

Open SSH client and connect to router, write in command `azure_iotHub` and you should receive confirmation when Data is sent to Azure IoT Hub:

```

root@Teltonika-RUT955:~# azure_iotHub
This sample simulates a Chiller device connected to the Remote Monitoring solution accelerator
{
}Confirmation callback received for message 1 with result IOTHUB_CLIENT_CONFIRMATION_OK
Device Twin reported properties update completed with result: 204
{
}Confirmation callback received for message 2 with result IOTHUB_CLIENT_CONFIRMATION_OK
{
}Confirmation callback received for message 3 with result IOTHUB_CLIENT_CONFIRMATION_OK

```

Open Device explorer and open Data tab, you should see that data is coming to IoT Hub:

```

- Event Hub Data
11/29/2019 2:59:39 PM> Device: [RUT955]. Data:{{
}}Properties:
'$$MessageSchema': 'dynamic-information;v1'
'$$ContentType': 'JSON'
'$$CreationTimeUtc': '2019-11-29T12:59:38Z'

11/29/2019 3:04:37 PM> Device: [RUT955]. Data:{{
}}Properties:
'$$MessageSchema': 'dynamic-information;v1'
'$$ContentType': 'JSON'
'$$CreationTimeUtc': '2019-11-29T13:04:36Z'

```