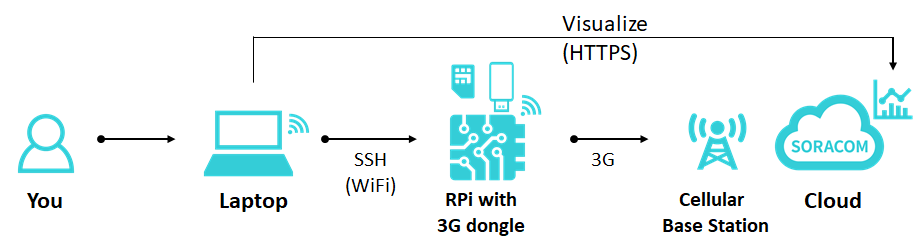
Online Instruction: https://goo.gl/GiEqyP

Scenario 1: Cellular connected Raspberry Pi

In this scenario, you will be using your own laptop to configure Raspberry Pi with Huawei MS2131 USB 3G modem and send live sensor data to SORACOM Harvest for visualization.

If you are not familiar with Linux commands, please study this page briefly before starting the workshop(https://www.raspberrypi.org/documentation/linux/usage/commands.md)

Overall Connection Diagram



1.Check to see if the cellular IoT workshop Kit is complete



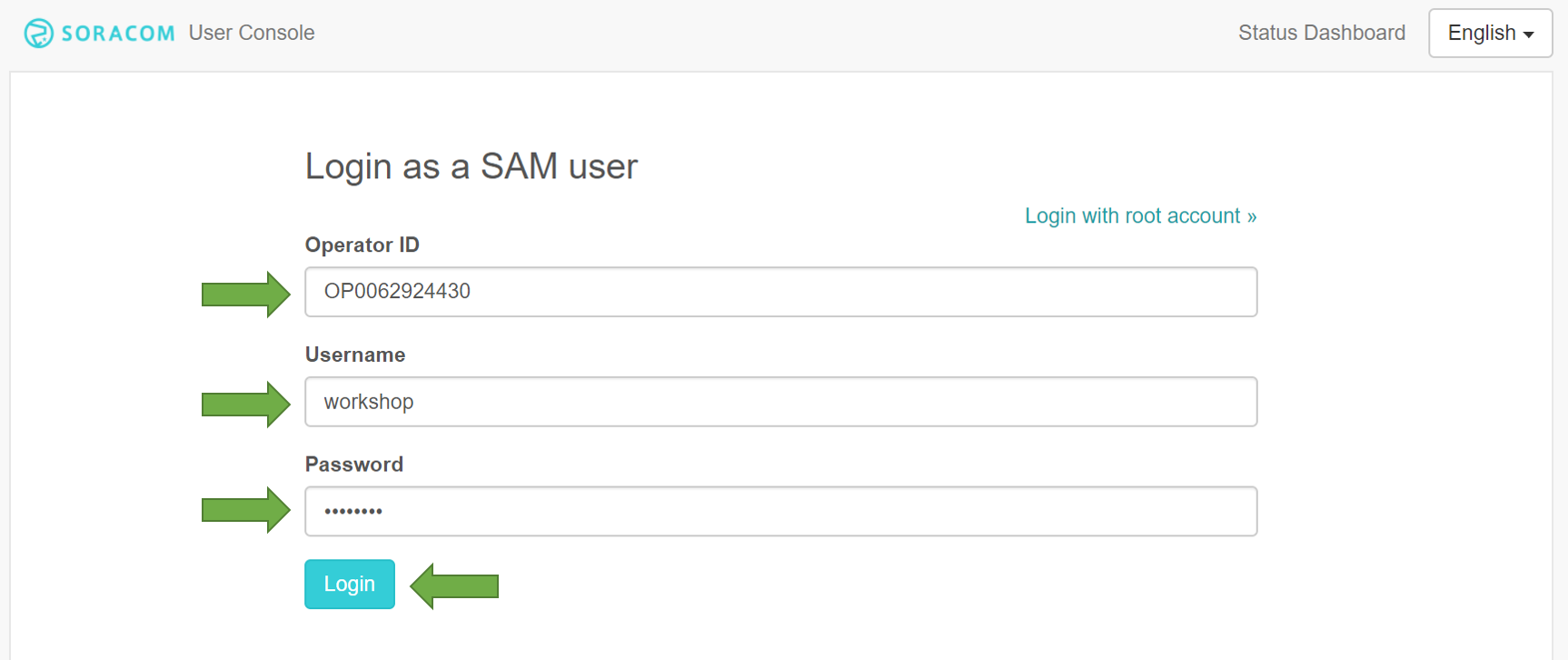
# 2.Logging onto the pre-registered SORACOM Account.

We have created a pre-registered SORACOM account for this workshop. Please follow the information below to log onto the SORACOM account.

Visit the below link by below for the SORACOM SAM user console.

[**https://console.soracom.io/#/sam\_login**](https://console.soracom.io/#/sam_login?coverage_type=g)

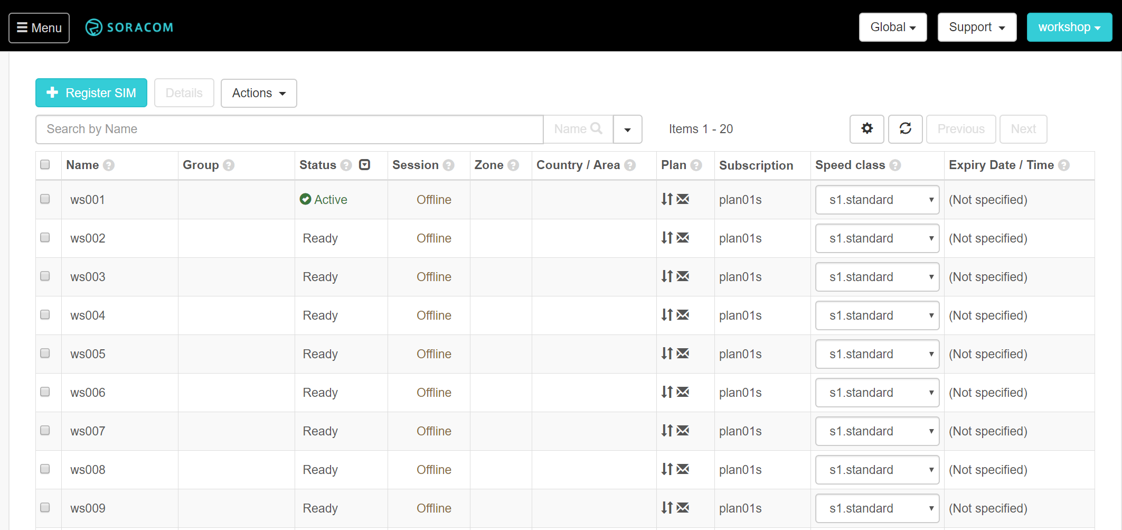
If you are reading this guide online, “Ctrl/Cmd + Click” to open a new browser tab, so that you can keep this guide open on your previous tab.



Please enter the following information to log in:

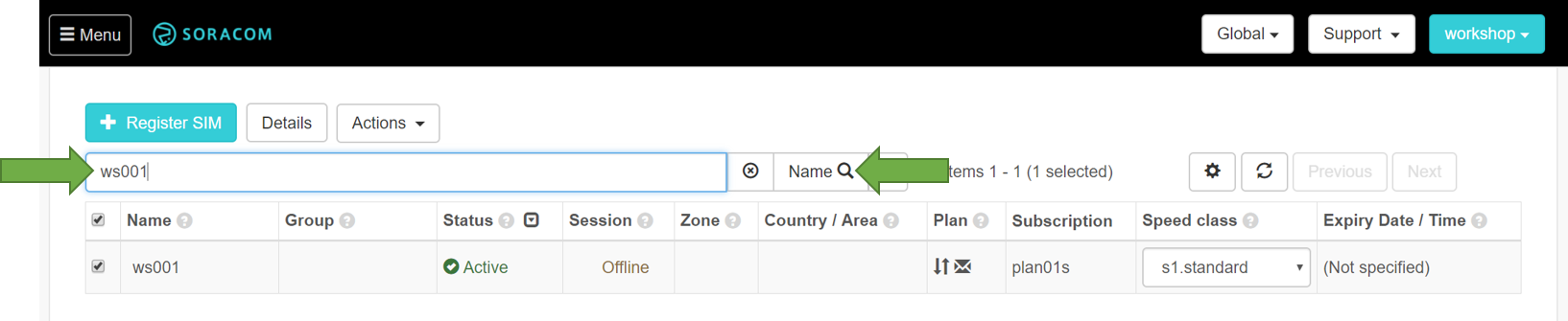
* Operator ID: OP0062924430
* Username: workshop
* Password: <Displayed on the screen>

Click “Menu” and select “SIM management” if you are not on below page.



You will find the sticker on the back of Huawei MS2131i-8 3G USB Dongle you received. Enter the name of your dongle in the search box and click Search button next to the search box.



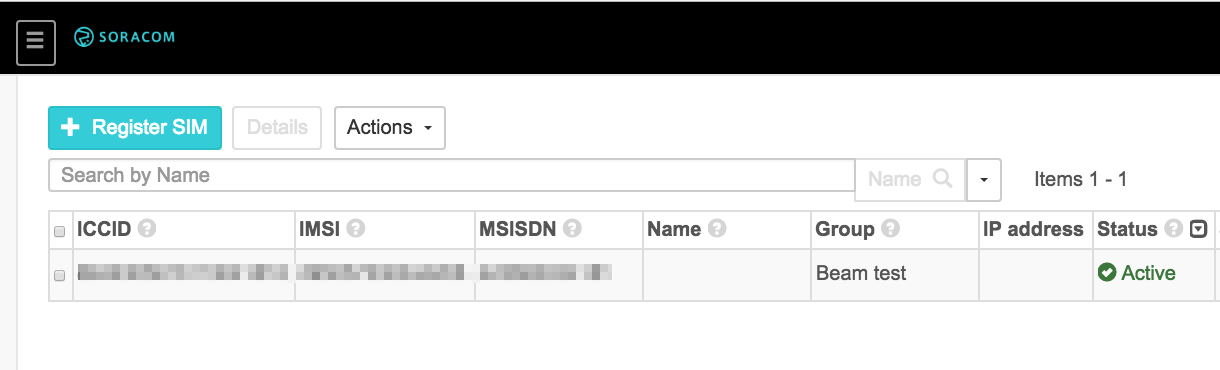


This is your SIM name for you to use during this workshop. Please do not use other name to avoid interrupting other people’s SIM.

# 3.SIM administration

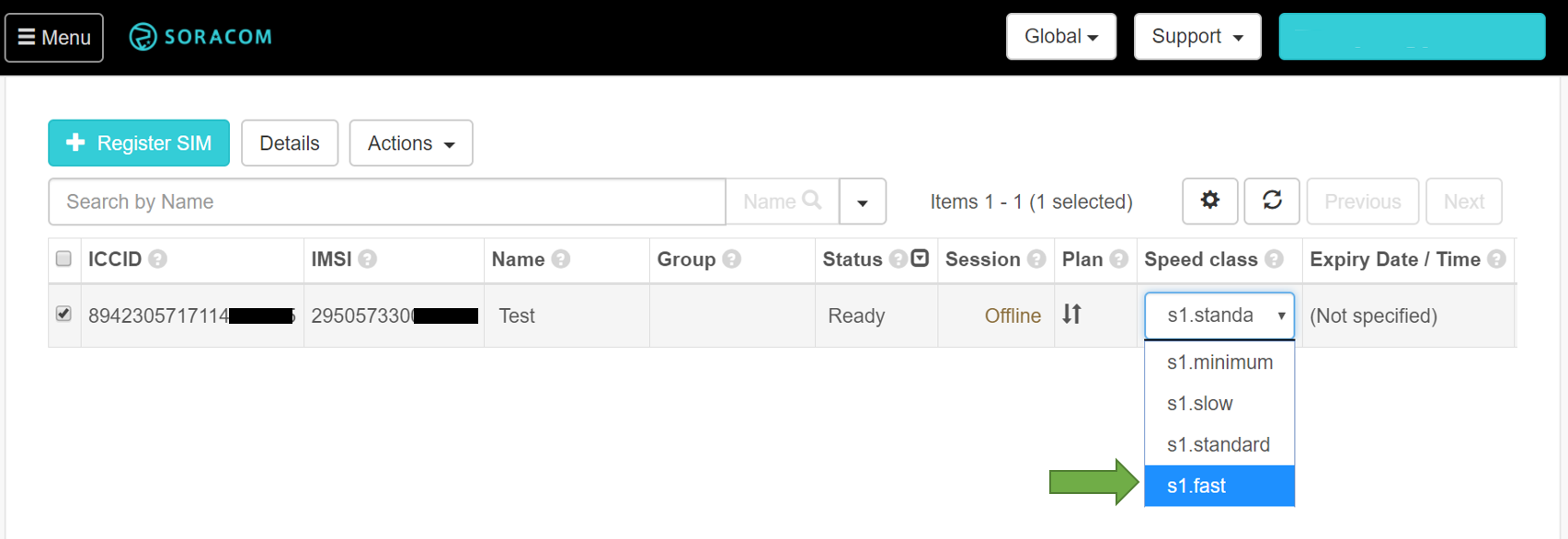
The SORACOM platform let you become a mobile network operator. This section explains how to navigate SORACOM User Console to manage and administer your SORACOM Air SIMs.

After logging onto the User Console, you will see the list of currently registered SORACOM Air SIMs.



## Changing your Air SIM speed class (Optional)

Changing your Air SIM speed class allows you to set threshold to your device network bandwidth without contacting your mobile provider. This allows you to avoid sudden traffic spike and save cost on the cellular data for your IoT devices.



For more information on SORACOM User Console Capabilities, please refer to <https://dev.soracom.io/en/start/console/>

## 4.Turn on your Raspberry Pi and connect to workshop Wi-Fi network

Please turn on your Raspberry Pi and connect your laptop to the Wi-Fi network labeled on your Raspberry Pi. Connecting Micro USB cable into your Laptop’s USB port will turn your Raspberry Pi on.



Select the SSID(soracom-event-xx) from the Wi-Fi Access point list in your laptop’s Wi-Fi menu and type in the password “**soracom7**”



## 5. Install SSH client to your laptop

For Windows users, please install an SSH client if you do not have one already. We recommend using “PuTTY” (<http://www.putty.org/>) for this workshop. Please download PuTTY from the web page and open it in your laptop.

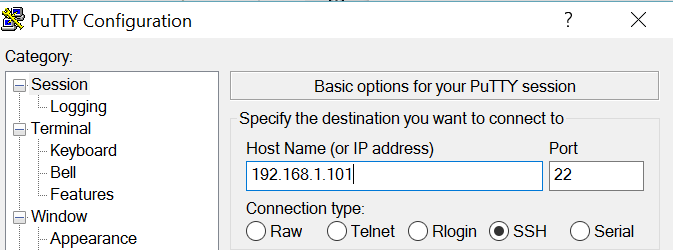
For Mac/Linux users we recommend using built in “Terminal” program. You can launch it from Spotlight/Launchpad by typing “Terminal” into the Search box at the Top-Center of the Launchpad screen.

## 6. Connecting to your Raspberry Pi

For this workshop, your Raspberry Pi is preconfigured to connected to “soracom-event-XX” Wi-Fi network with a fixed IP address labeled on your Raspberry Pi. Please check the label and connect over SSH following the guide below.

For more information on Wi-Fi configuration, please refer to this external site(https://scottlinux.com/2016/01/17/configure-wifi-on-raspberry-pi-raspbian-debian-linux-via-command-line/)

## Putty (For Windows users)



When prompted, type in “**pi**” as your user name and “**raspberry**” (NOT “rasberry” without the “p”) as your password

## **Command** (For Mac/Linux users)

SSH pi@192.168.xx.xx

\*when prompted for your password, type it in “raspberry” (NOT “rasberry” without the “p”)

# 7. Configuring Huawei MS2131i-8 3G USB Dongle

You will be installing “network-manager” software package into your Raspberry Pi over SSH. Please make sure you are logged on to Raspberry Pi in your Putty/Terminal app.

## **Command**

sudo apt-get update && sudo apt-get install network-manager

## **Output**

...

Need to get 7,709 kB of archives.

After this operation, 20.8 MB of additional disk space will be used.

Do you want to continue? [Y/n] y

Type 'y' and enter

...

Setting up ppp (2.4.6-3.1) ...

Setting up iputils-aRaspberry Ping (3:20121221-5) ...

Setcap worked! aRaspberry Ping is not suid!

Processing triggers for libc-bin (2.19-18+deb8u7) ...

Processing triggers for dbus (1.8.22-0+deb8u1) ...

Processing triggers for systemd (215-17+deb8u7) ...

## You are now setting up an APN(Access Point Name) for SORACOM Air SIM to connect to SORACOM mobile network.

## **Command**

sudo nmcli con add type gsm ifname "\*" con-name soracom apn soracom.io user sora password sora

## **Output**

Connection 'soracom' (3cbecb73-2f6c-48f9-819a-3e233408d4a0) successfully added.

To make above configuration active, you need to reboot your Raspberry Pi

## **Command**

sudo reboot

Your Raspberry Pi will be rebooted and your current SSH session will be disconnected. Please reconnect to your Raspberry Pi over SSH again by following Step 6.

## Putty (For Windows users)

Plug in your Huawei USB Dongle. You should now see “ppp0” interface in your “ifconfig” output. This means your cellular connection is successfully established. USB dongle LED indicator should be lit solid blue.



## Command

ifconfig

## Output

ppp0 Link encap:Point-to-Point Protocol

inet addr:10.237.184.147 P-t-P:10.64.64.64 Mask:255.255.255.255

UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1

RX packets:7 errors:0 dropped:0 overruns:0 frame:0

TX packets:9 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:3

RX bytes:106 (106.0 B) TX bytes:171 (171.0 B)

If you only see wwan0, it means your cellular connectivity is not successfully established. Please repeat Step 7. If you still fail to establish, please raise your hand for help.

Your Raspberry Pi will automatically dial up each time your 3G USB modem is plugged into this Raspberry Pi, or when the connection is interrupted for some reason (e.g. Out of Cellular Range, Carrier Timeouts, or unexpected modem/device resets)

By default, Cellular connection has lower routing priority than Wi-Fi or Ethernet. This will sometimes affect your SSH access to your device over Wi-Fi or Ethernet. To solve this, you can update the default network route by editing the IP routing table. You can save the command as a Network Manager script and have it run each time you connect a USB modem or you reboot your Raspberry Pi.

To do this, run the route -n command to see the current IP routing table:

## Command

route -n

## Output

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

0.0.0.0 192.168.1.1 0.0.0.0 UG 303 0 0 wlan0

0.0.0.0 10.64.64.64 0.0.0.0 UG 1024 0 0 ppp0

10.64.64.64 0.0.0.0 255.255.255.255 UH 0 0 0 ppp0

169.254.0.0 0.0.0.0 255.255.0.0 U 304 0 0 wwan0

192.168.1.0 0.0.0.0 255.255.255.0 U 303 0 0 wlan0

By default, wlan0 interface is set to highest priority. This means the traffic will pass wlan0 (Wi-Fi) interface even when you try to send data to SORACOM network.

Next step, download and install Soracom ppp route metric script:

Command

sudo curl -o /etc/NetworkManager/dispatcher.d/90.set\_ppp\_route\_metric https://soracom-files.s3.amazonaws.com/handson/90.set\_ppp\_route\_metric

Then, change the binary to executable

## Command

sudo chmod +x /etc/NetworkManager/dispatcher.d/90.set\_ppp\_route\_metric

Last but not least, you register them as Network Manager script to run when the USB modem is plugged in or the device restarts.

## Command

sudo /etc/NetworkManager/dispatcher.d/90.set\_ppp\_route\_metric ppp0 up

After running the script, run route -n to see the routing table:

## Command

route -n

## Output

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

0.0.0.0 10.64.64.64 0.0.0.0 UG 0 0 0 ppp0

0.0.0.0 192.168.1.1 0.0.0.0 UG 303 0 0 wlan0

0.0.0.0 10.64.64.64 0.0.0.0 UG 1024 0 0 ppp0

10.64.64.64 0.0.0.0 255.255.255.255 UH 0 0 0 ppp0

169.254.0.0 0.0.0.0 255.255.0.0 U 304 0 0 wwan0

192.168.1.0 0.0.0.0 255.255.255.0 U 303 0 0 wlan0

Notice that ppp0 is now the highest priority in the routing table. So for all network traffic, the ppp0 interface will be used.

8.Wiring up Ultra Sonic Sensor

Description for HC-SR04 Ultrasonic Sensor in [User's Manual](https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit)

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats or dolphins do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. From 2cm to 400 cm or 1” to 13 feet. It operation is not affected by sunlight or black material like Sharp rangefinders are (although acoustically soft materials like cloth can be difficult to detect). It comes complete with ultrasonic transmitter and receiver module.

To measure the distance between the object and the sensor, you need to perform below steps.

1. Send a pulse of HIGH (5V) to "Trig" pin at least 10 us (=0.00001 seconds).
2. Start measuring time when "Echo" pin becomes HIGH (5V). (=when ultrasonic waves sent)
3. Stop measuring time when "Echo" pin becomes LOW(0V). (=when reflected waves received)

If the measured time was t (sec), distance to object can be obtained by the following formula.

distance(m) = 340(m/sec) x t (sec) / 2 (because of round trip)

speed of sound depends on temperature, but roughly around 340 m/sec in the normal air.

**DO NOT connect red wire and/or connect any wires to Red GPIO pins until you connect everything else. Failing to follow may result in burning the sensor and Raspberry Pi circuits.**

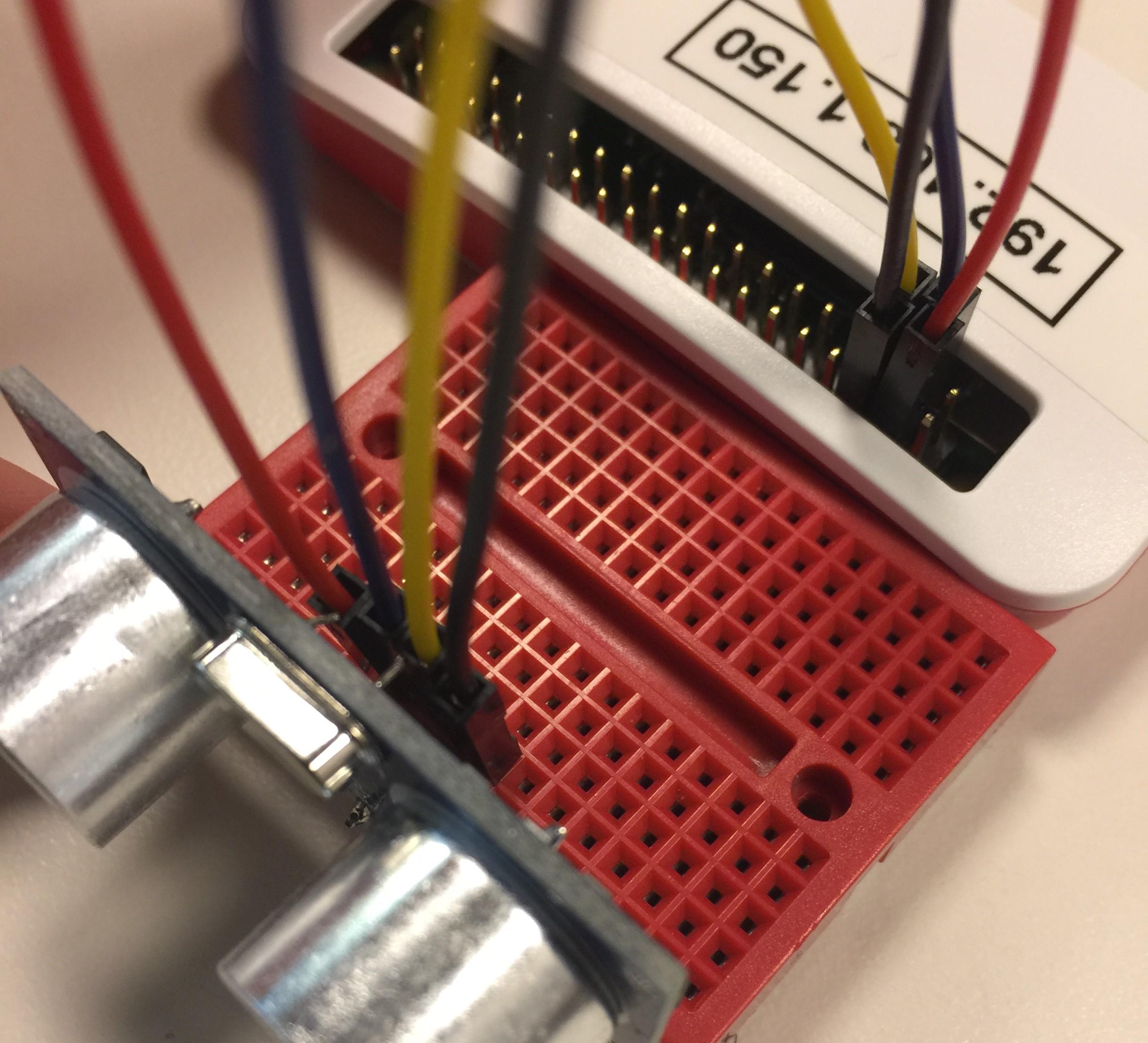
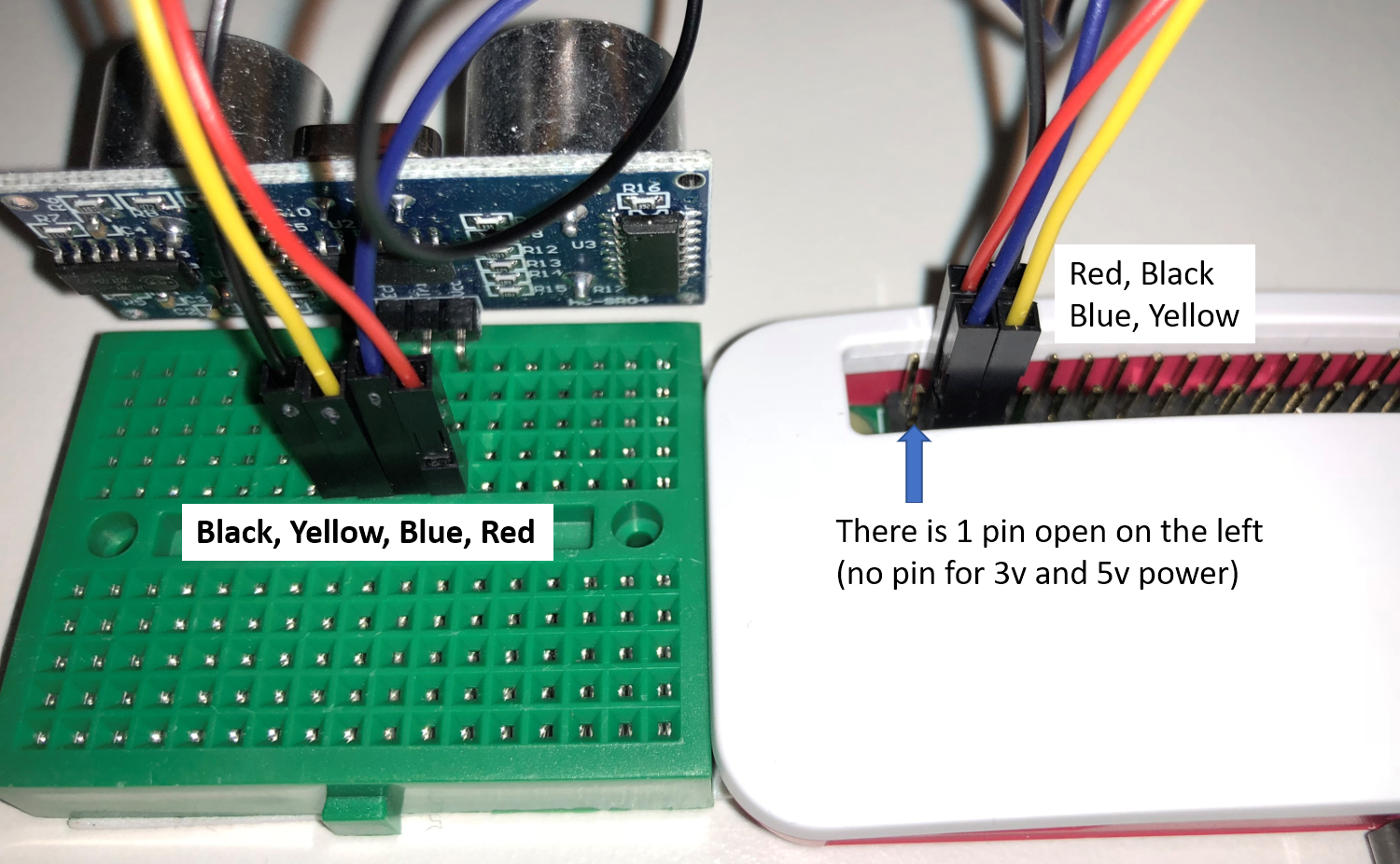
Raspberry Pi GPIO pinout



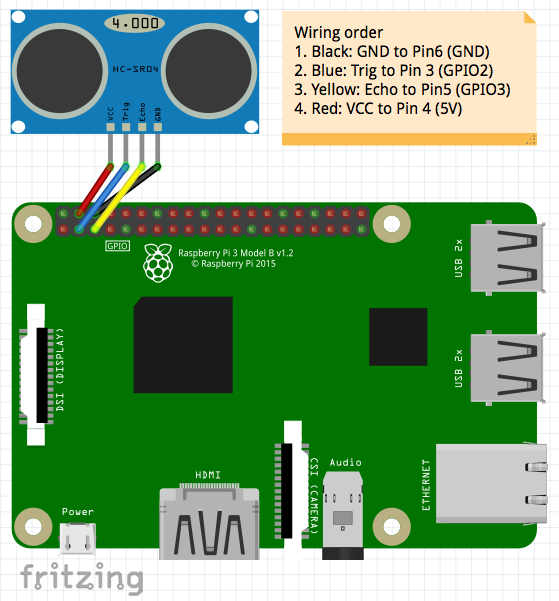
**If your sensor or Raspberry Pi feels hot or smoky, unplug all wires from your Raspberry Pi and raise your hand for help**

This Ultrasonic sensor has 4 pins. Please refer to the image and wiring sketch to correctly wire 4 wires from Raspberry PI GPIO pins to your Ultrasonic sensor.

**Correct Wiring image**



**Wiring Sketch**

.

9.Testing your sensor

Once you wired up the ultrasonic sensor, please download this test script (https://soracom-files.s3.amazonaws.com/handson/hcsr04.py) written in Python. To download and test it on Raspberry Pi, enter following command in the terminal.

**Command**

curl -O https://soracom-files.s3.amazonaws.com/handson/hcsr04.py

python hcsr04.py

**Output**

pi@raspberrypi:~ $ curl -O https://soracom-files.s3.amazonaws.com/handson/hcsr04.py

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 1752 100 1752 0 0 2650 0 --:--:-- --:--:-- --:--:-- 2650

pi@raspberrypi:~ $ python hcsr04.py

distance: 46.0 cm

distance: 46.4 cm

distance: 46.0 cm

distance: 47.2 cm

:

: (Hit Ctrl+C to stop)

**If you see the following error, most likely the script download was not completed. Please try downloading the script again.**

**Error 1**

pi@raspberrypi:~ $ python hcsr04.py

python: can't open file 'hcsr04.py': [Errno 2] No such file or directory

**Error 2**

pi@raspberrypi:~ $ python hcsr04.py

File "hcsr04.py", line 1

<?xml version="1.0" encoding="UTF-8"?>

^

SyntaxError: invalid syntax

**Try Tab completion for file names to avoid typing mistakes.**

**If it does not display anything just like below error output, please hit(Ctrl+C) and stop the process. Check to make sure your wiring is correct.**

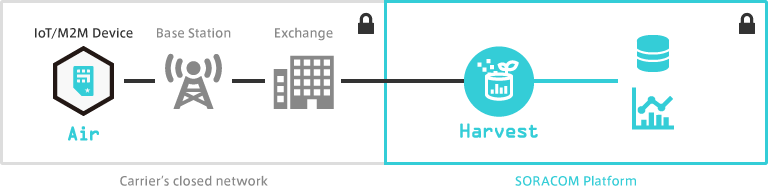
**Error output**

pi@raspberrypi:~ $ python hcsr04.py

(nothing)

**SORACOM Harvest**

SORACOM Harvest (Harvest) is a simple data visualization and storage service. To easily check if the data is properly sent from your devices in a right format, Harvest is a good service to try. Harvest holds your data for 40 days and this is suitable for proof of concept, testing of your first IoT device data ingestion and adjusting your data format for analytics.

By using Harvest, you can easily visualize the data without having to configure a backend server or storage in the beginning of your IoT project. You can send data to Harvest endpoint over HTTP, TCP, or UDP without the need for encryption. 

10. Log onto Harvest Console

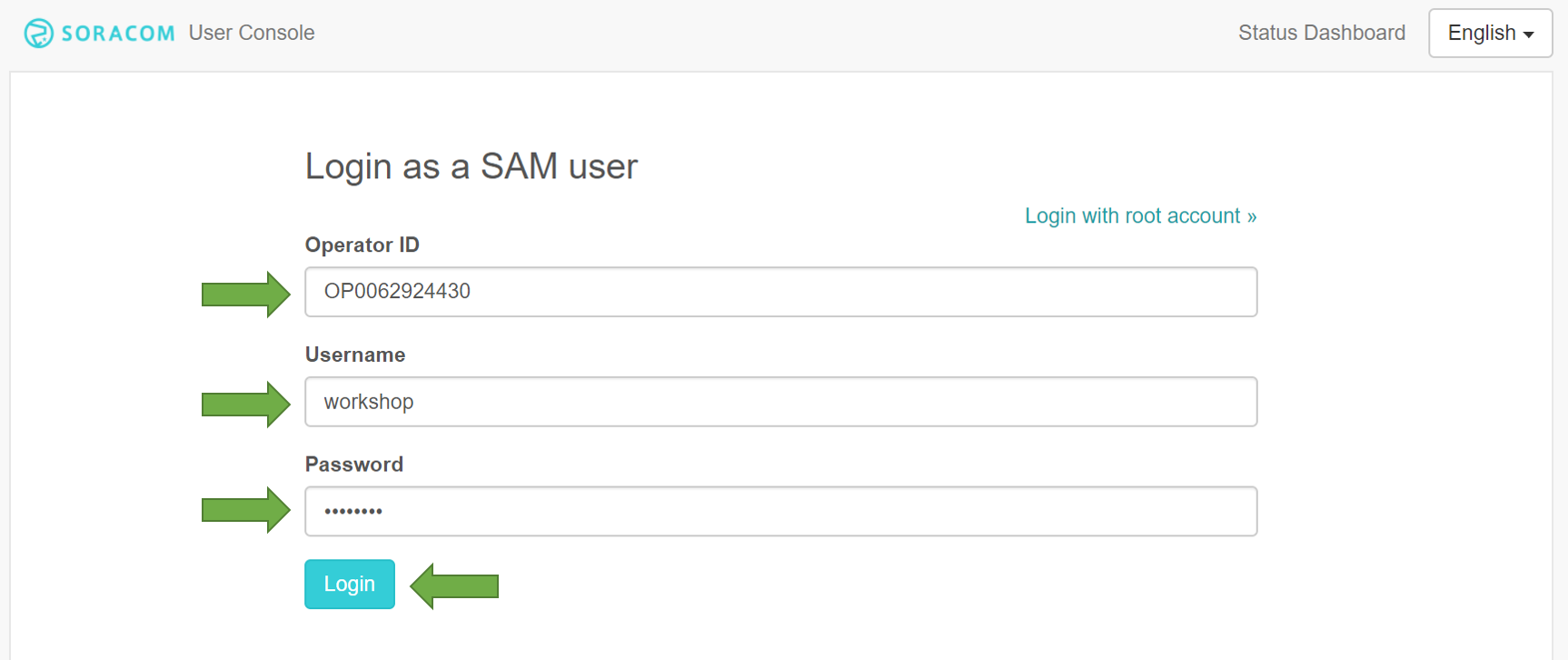
The dashboard is integrated into your User Console. You can view your graph by selecting your SIM and click “Harvest Data”

For more information, please refer to Harvest Getting Started Guide

Type in below URL in your browser to access our User Console

[**https://console.soracom.io/#/sam\_login**](https://console.soracom.io/#/sam_login?coverage_type=g)

“Ctrl/Cmd+Click” to open a new browser tab if you are reading this guide online.



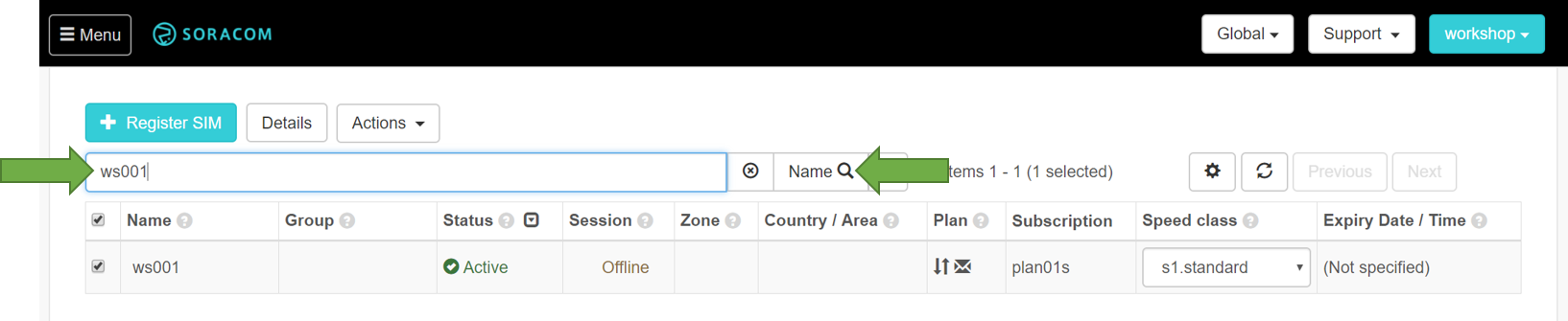
Please enter the following information to log in:

* Operator ID: OP0062924430
* Username: workshop
* Password: <Will be displayed on the main screen>

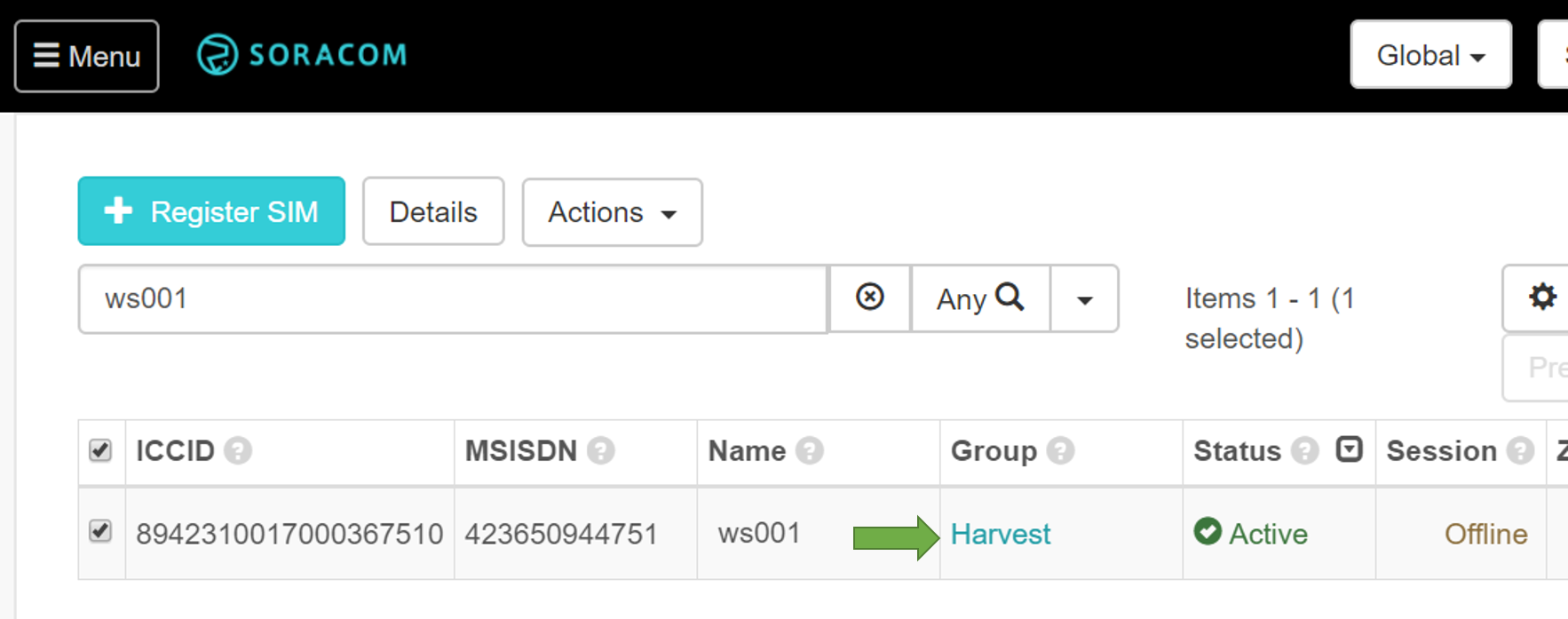
11.Select your device to view Harvest Graph

On SIM management screen, please enter the name in the “Search by any attribute” box to get to your workshop SIM.

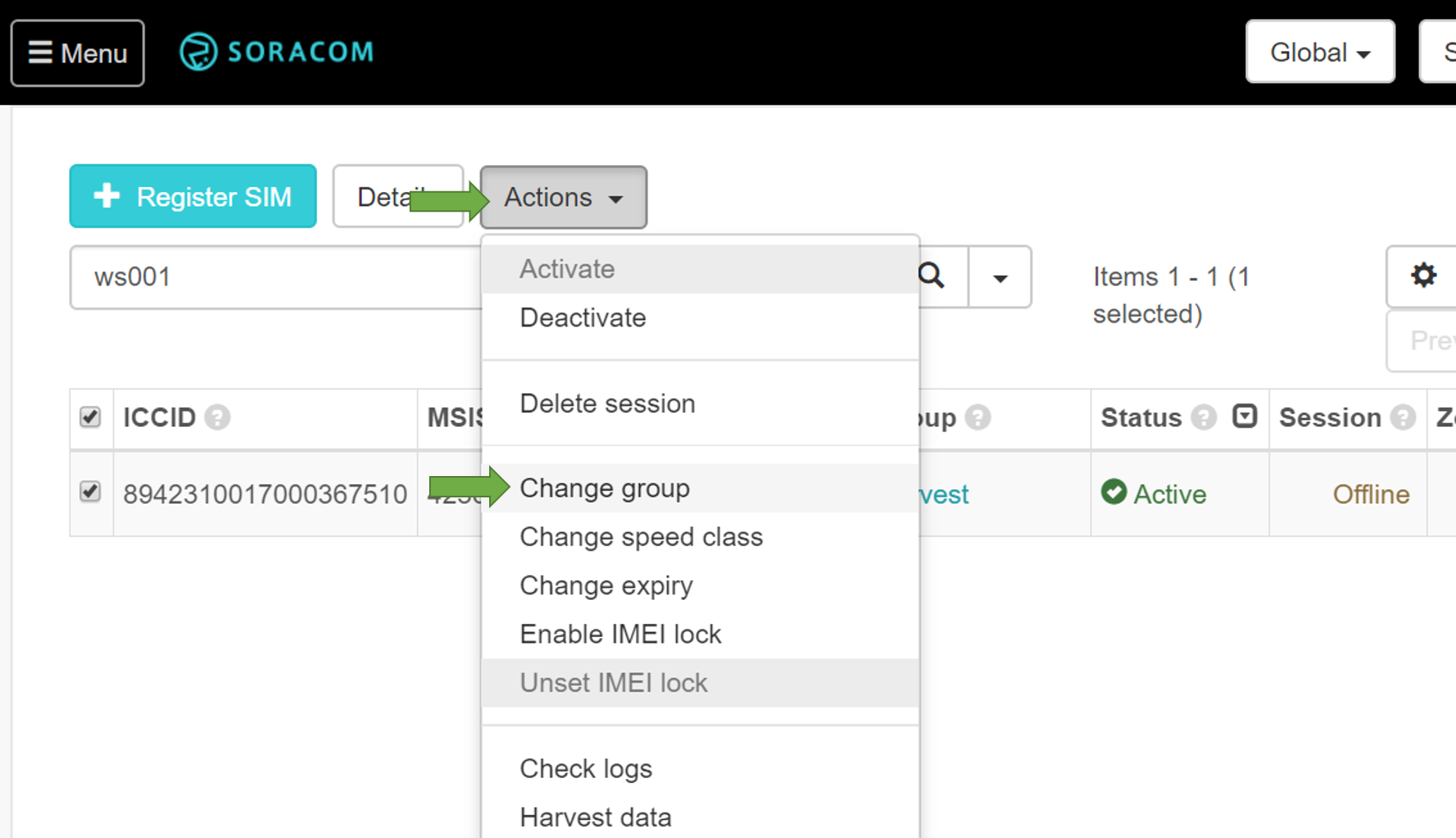




Please make sure your SIM is in “Harvest” group. This “Group” holds the Harvest configuration. All SIM in the same group will be allowed to send data to Harvest services.



If the SIM is not in “Harvest” group, please follow below screen and change the group to “Harvest”



12.Send test data to Harvest

Use with “curl” command and test the connectivity to Harvest server before sending the live data from your sensor.

**Command**

curl -v -d "Hello Harvest!" harvest.soracom.io

**Output**

pi@raspberrypi:~ $ curl -v -d "Hello Harvest!" harvest.soracom.io

\* Rebuilt URL to: harvest.soracom.io/

\* Hostname was NOT found in DNS cache

\* Trying 100.127.111.111...

\* Connected to harvest.soracom.io (100.127.111.111) port 80 (#0)

> POST / HTTP/1.1

> User-Agent: curl/7.38.0

> Host: harvest.soracom.io

> Accept: \*/\*

> Content-Length: 14

> Content-Type: application/x-www-form-urlencoded

>

\* upload completely sent off: 14 out of 14 bytes

< HTTP/1.1 201 Created

< Date: Tue, 09 May 2017 15:55:55 GMT

\* Server nginx/1.6.2 is not blacklisted

< Server: nginx/1.6.2

< Content-Length: 0

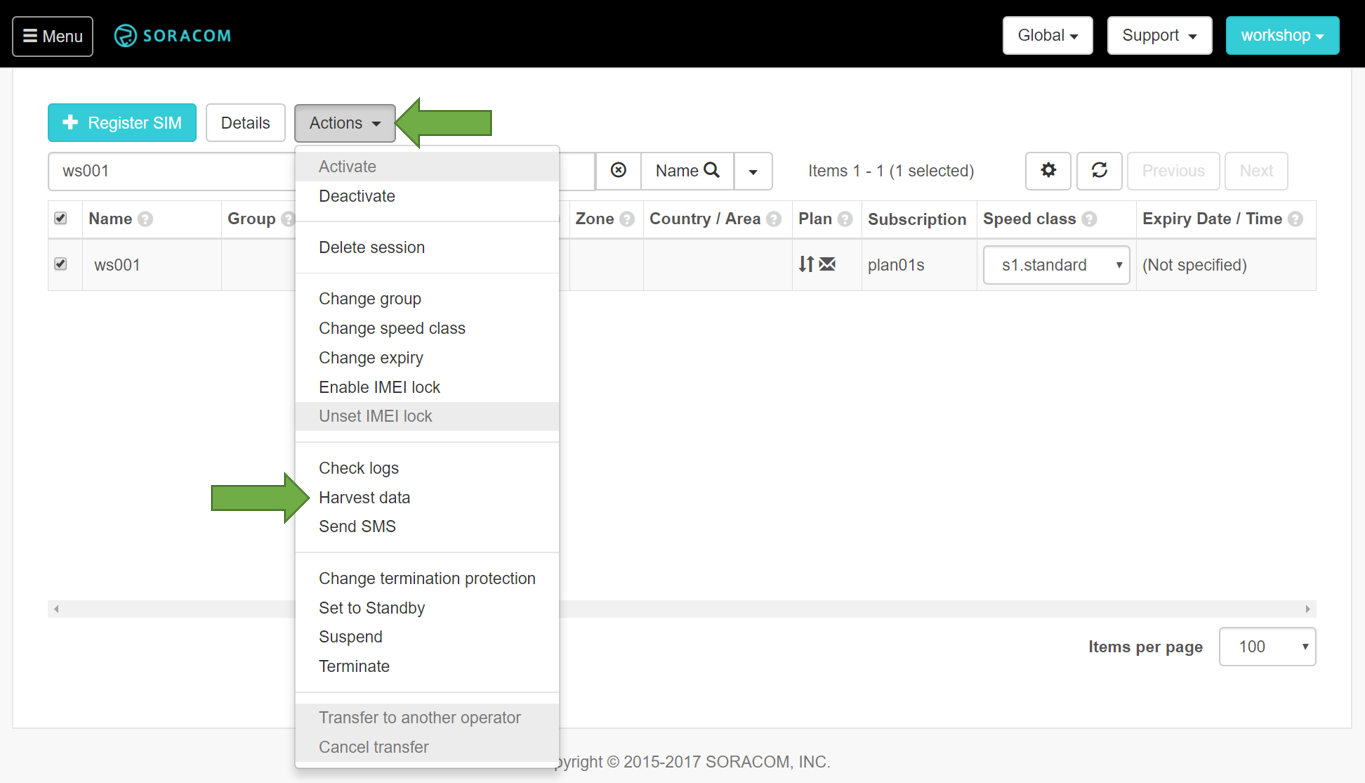
< Connection: Close

<

\* Closing connection 0

If you do not see above, please raise your hand and ask for help.

In the User Console, click “Actions” and “Harvest Data”



It will display uploaded data at the bottom. If the test upload is successful, you should see an entry with "Hello Harvest!" in “Content” column with other metadata.

**Notice: You need to scroll down to the bottom to check this message**



13. Send your ultrasonic sensor data to Harvest

Download another python data upload script(ultrasonic\_harvest.py) and run it in your Raspberry Pi to upload live sensor data to Harvest. You will need to install “python-requests” software package into your Raspberry Pi to make HTTP requests to Harvest from your python code.

**Command**

sudo apt-get install -y python-requests

curl -O http://soracom-files.s3.amazonaws.com/handson/ultrasonic\_harvest.py

python ultrasonic\_harvest.py

**Output**

pi@raspberrypi:~ $ sudo apt-get install -y python-requests

:

:

pi@raspberrypi:~ $ curl -O http://soracom-files.s3.amazonaws.com/handson/ultrasonic\_harvest.py

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 1169 100 1169 0 0 3636 0 --:--:-- --:--:-- --:--:-- 3641

pi@raspberrypi:~ $ python ultrasonic\_harvest.py

- reading distance

distance: 4.6 cm

- sending data to Harvest

<Response [201]>

- reading distance

distance: 4.7 cm

- sending data to Harvest

<Response [201]>

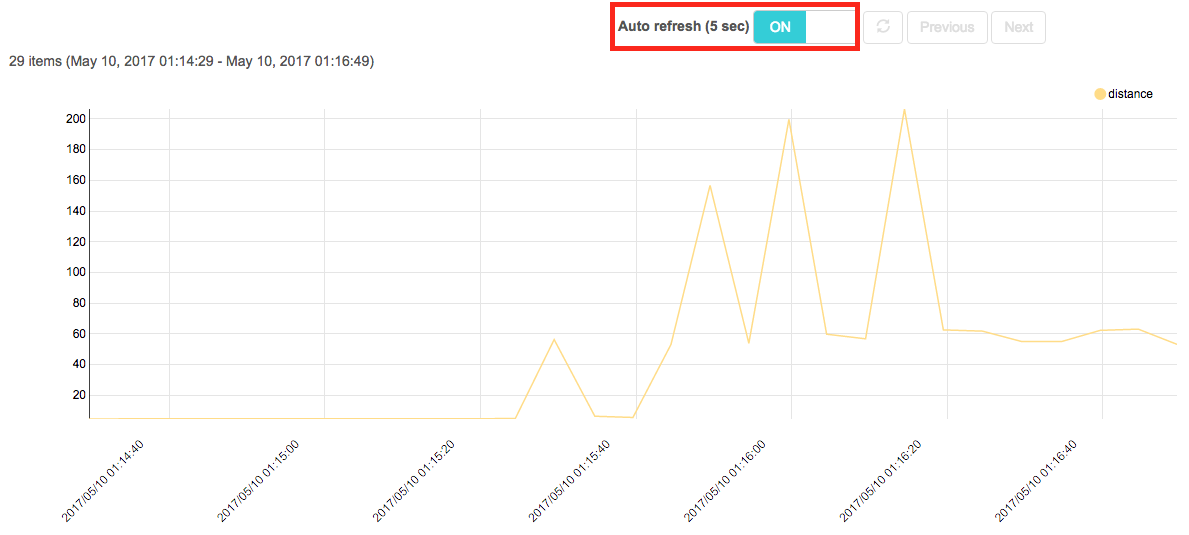
:

:

If you do not see the output above, please try downloading the script again.

14.Check the visualized live data on the User Console

Please enable “Auto refresh” by clicking the slider. Place your hand in front of the sensor, move your hand away from the sensor and then move closer the sensor a few times repeatedly to see if the “distance” data changes in the graph.



This concludes our Basic Cellular to SORACOM Harvest workshop scenario. If you have some time left, please move onto advanced scenario below.

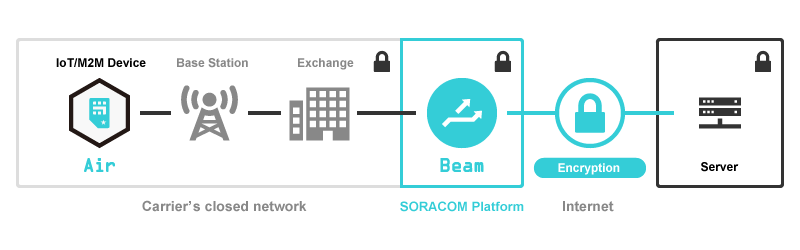
If you have any questions regarding your IoT project, please come to soracom.io and use “contact us” form to get to us

2. Advanced Scenario

In this workshop, you will learn how to send data from your device to backend servers or cloud services in a simple and secure fashion without your device encrypting the data.

# What is SORACOM Beam

SORACOM Beam (“Beam”) translates lightweight device friendly protocols like HTTP, MQTT, TCP or UDP into secure protocols like HTTPS, MQTTS or TCPS. Because SSL/TLS adds extra 100s of bytes overhead to each data upload, you end up saving approximately 40 to 60% of data transfer in typical IoT use cases. Beam also helps simple sensors or microcontrollers with no encryption capabilities to connect to cloud services like AWS IoT requiring HTTPS or MQTTS.

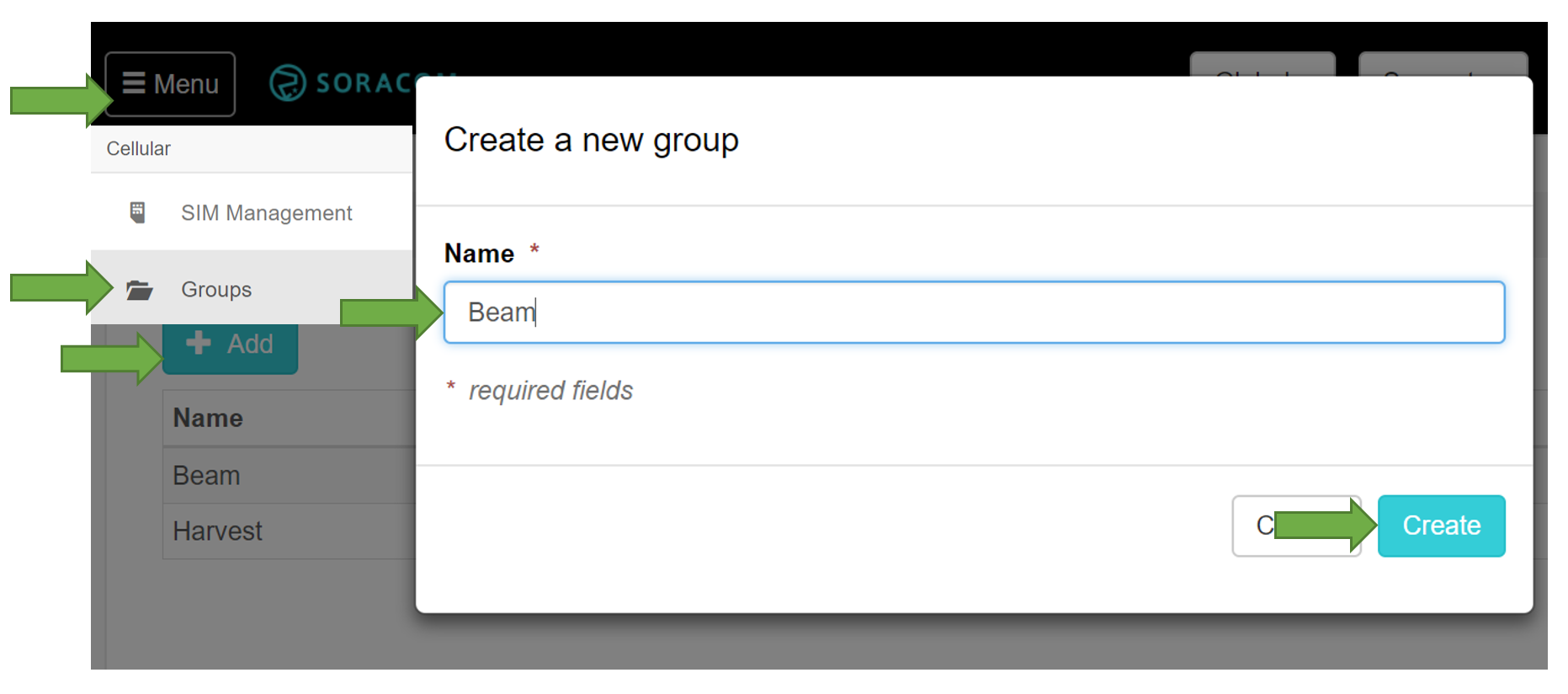
First, your data will be sent to Beam end point via 3G/LTE network to SORACOM Platform. Your data will then be encrypted by SORACOM and be forwarded to the destination of your choice. 

Currently, Beam supports the following protocol combinations.

|  |  |
| --- | --- |
| Original Protocol | Translated Protocol |
| HTTP | HTTP/HTTPS |
| MQTT | MQTT/MQTTS |
| TCP | TCP/TCPS |
| TCP | HTTP/HTTPS |
| UDP | HTTP/HTTPS |

To use Beam, you need to create a group, configure Beam parameter and move your SIM into the group you configured Beam with.

## 1.Create a Group

Click "Menu" in top left corner, and select "Groups". In the "Groups" screen, create a group by clicking "+ Add" button. You give it a name and hit “Create” 

## 2.Configure Group

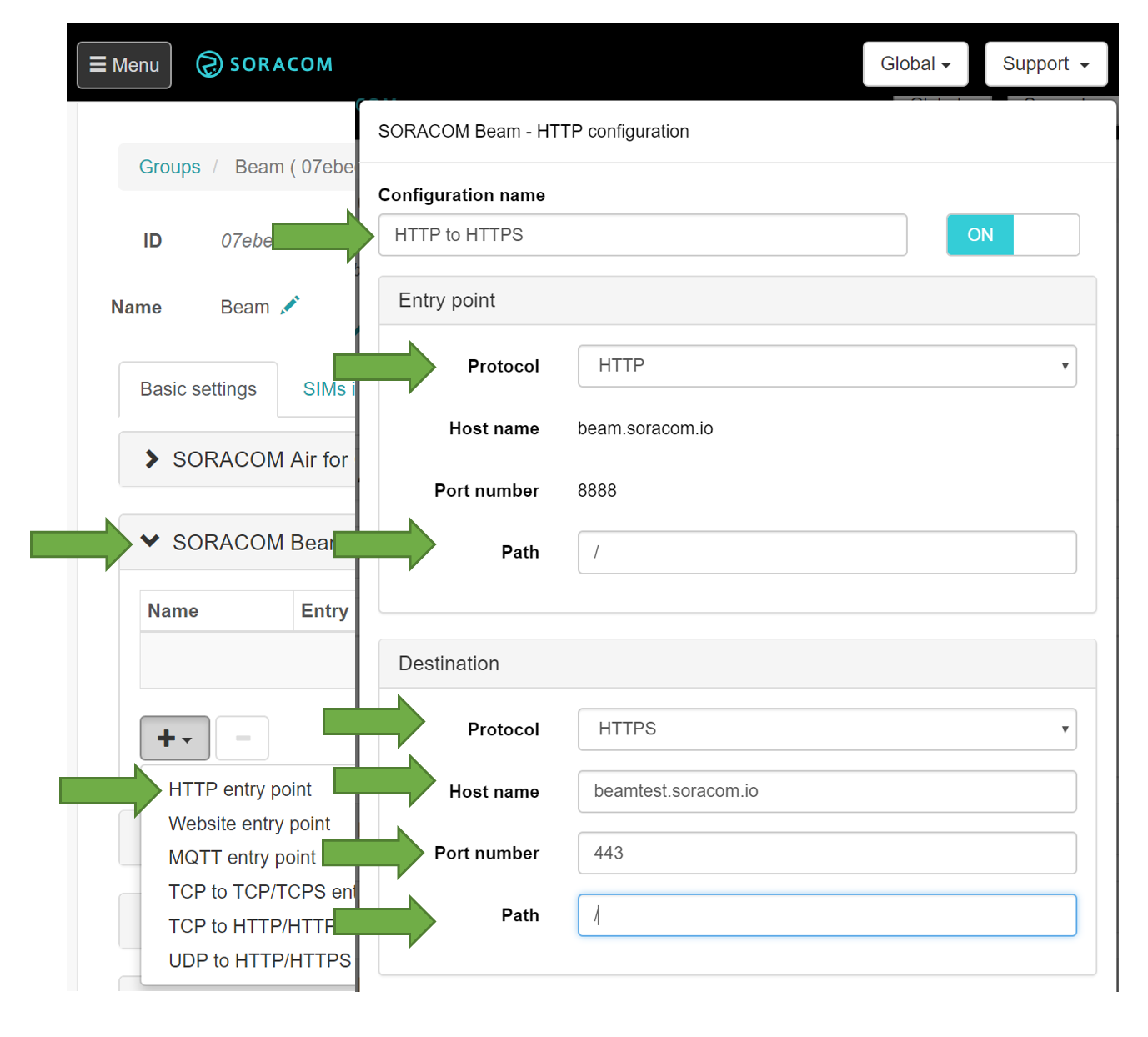
1.Click the group name in the list to open group detail.

2.Expand "SORACOM Beam" section.

3.Click on "+" button choose HTTP entry point from protocol list

4.Fill in the information below when "SORACOM Beam - HTTP configuration" dialog comes up

* Configuration name: You can give an arbitrary name. For this workshop, name "HTTP to HTTPS"
* Entry point -> Path: specify "/"
* Destination -> Host name: specify "beamtest.soracom.io" (this is preconfigured HTTPS endpoint for this workshop)
* Destination -> Port number: leave it blank or specify "443" (You can skip "Port number" if you use default port for HTTPS, which is 443/tcp)
* Destination -> Path: specify "/"

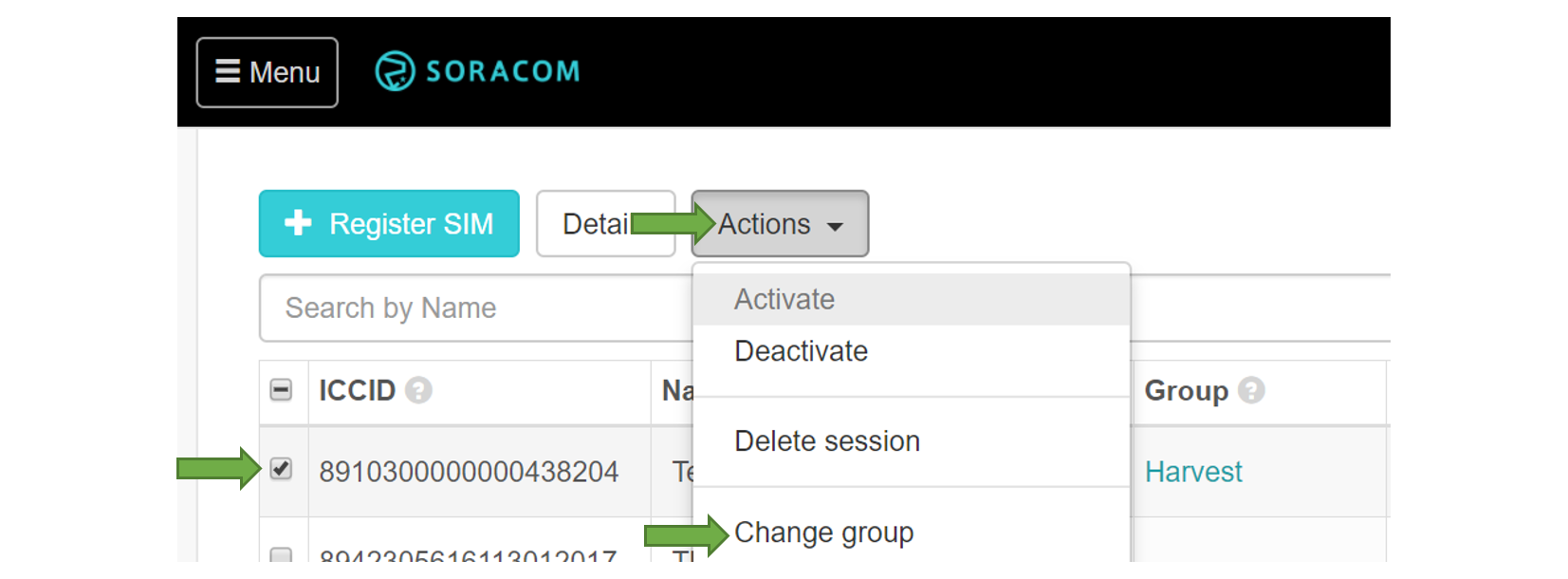
Scroll down to the bottom and click "Save".

## 3.Change group in your SIM management page

1.Click "Menu" and go back to "SIM Management" and search for the SIM name.

2.Check the box next to your SIM name and click "Actions" -> "Change group".

3.Click on ▼ and choose the group you created.



4.Click "Update" to apply.

## 4.Test the connection from Raspberry Pi

Login to the Raspberry Pi and type the following command.

### **Command**

curl https://beamtest.soracom.io

curl http://beam.soracom.io:8888

### **Output**

pi@raspberrypi:~ $ curl https://beamtest.soracom.io

Hello Unknown Client...

== HTTP Headers ==

pi@raspberrypi:~ $ curl http://beam.soracom.io:8888

Hello Unknown Client...

== HTTP Headers ==

You will see the same output for both commands although the first command is accessing the endpoint over HTTP. This means the HTTP request was translated into HTTPS and was forwarded to test server as an HTTPS request.

## 5.Append SIM info

As SORACOM identifies the SIM by its unique IDs, SORACOM Beam can append SIM information when forwarding the data to your endpoint. Please follow the step below to append Pre-Shared Key for your device to access restricted backend applications.

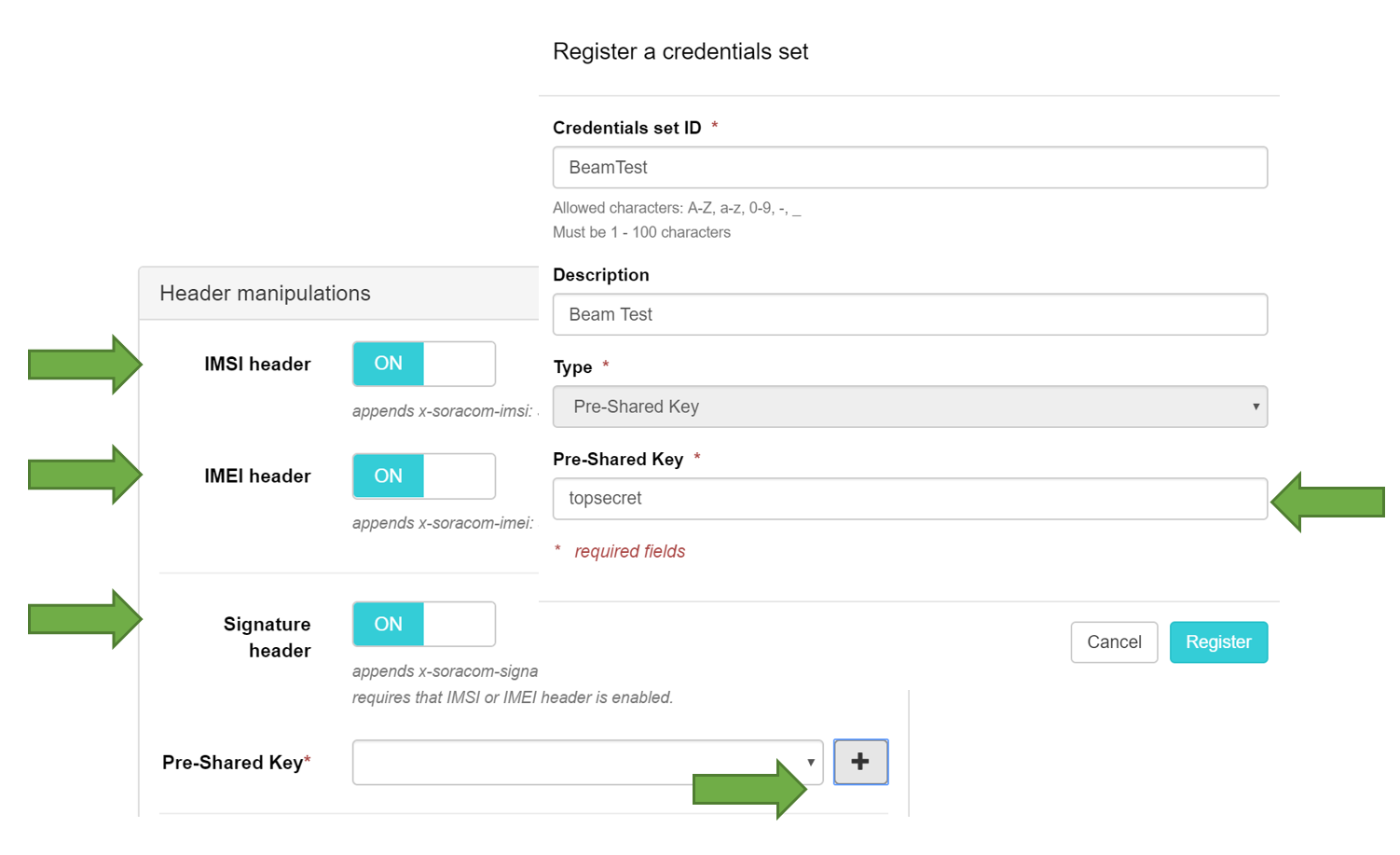
1.Go back to "Group" setting by clicking the group name on your SIM and expand SORACOM Beam configuration section.

2.Click "..." button on the right.

3.You will see "Header manipulations" at the bottom. Turn on "IMSI header", "IMEI header", and "Signature header".

4.Press [+] button next to "Pre-Shared Key".

5.You can give any arbitrary name, but for this workshop, type in "topsecret" in "Pre-Shared Key". "topsecret" is the password already configured in Beam test server for signature matching example in this workshop.



6.Click "Register" to get back to "HTTP configuration" and click "Save" to apply the setting.

7.Execute the following command on your Raspberry Pi.

### **Command**

curl http://beam.soracom.io:8888

### **Output**

pi@raspberrypi:~ $ curl http://beam.soracom.io:8888

Hello SORACOM Beam Client 89423xxxxxxxxxx !

== HTTP Headers ==

HTTP\_X\_SORACOM\_IMEI = 35636xxxxxxxxxx

HTTP\_X\_SORACOM\_IMSI = 89423xxxxxxxxxx

HTTP\_X\_SORACOM\_SIGNATURE = ccafbf350fffd01f5f4fd0dac0b34da80870864b54c621c97c1125ba30f1e4a7

HTTP\_X\_SORACOM\_SIGNATURE\_VERSION = 20151001

HTTP\_X\_SORACOM\_TIMESTAMP = 1494662049112

= Signature Verification =

Pre shared key = topsecret

stringToSign:

x-soracom-imei=35636xxxxxxxxxxx-soracom-imsi=89423xxxxxxxxxxx-soracom-timestamp=1494662049112

calculated\_signature:

SHA256('topsecret'+stringToSign) = ccafbf350fffd01f5f4fd0dac0b34da80870864b54c621c97c1125ba30f1e4a7

provided\_signature:

ccafbf350fffd01f5f4fd0dac0b34da80870864b54c621c97c1125ba30f1e4a7

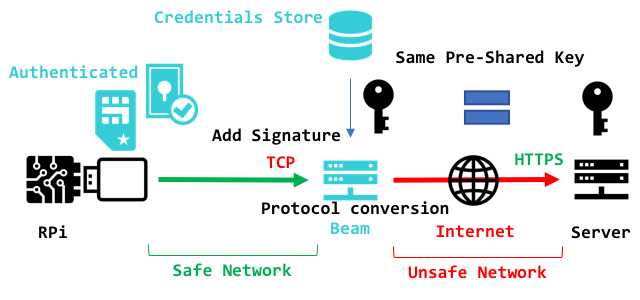
signature:

Match!

The output should look different now. Beam test server identifies this request coming from the SIM card with IMSI(89423xxxxxxxxxx). Beam test server understands this information from HTTP\_X\_SORACOM\_IMSI header.

To prevent unauthorized accesses, you should also check the request signature. Beam test server can calculate it from "Pre-Shared Key", “IMSI”, “IMEI” and its timestamp. If the signature matches, you can prove that this access is from the SIM card on your IoT device.

## 6.TCP to HTTPS translation



If your device is not capable of handling HTTP or HTTPS, Beam can even translate TCP into HTTPS for secure data transmission. Let's try TCP to HTTPS translation in this section.

## 7. Configure Group

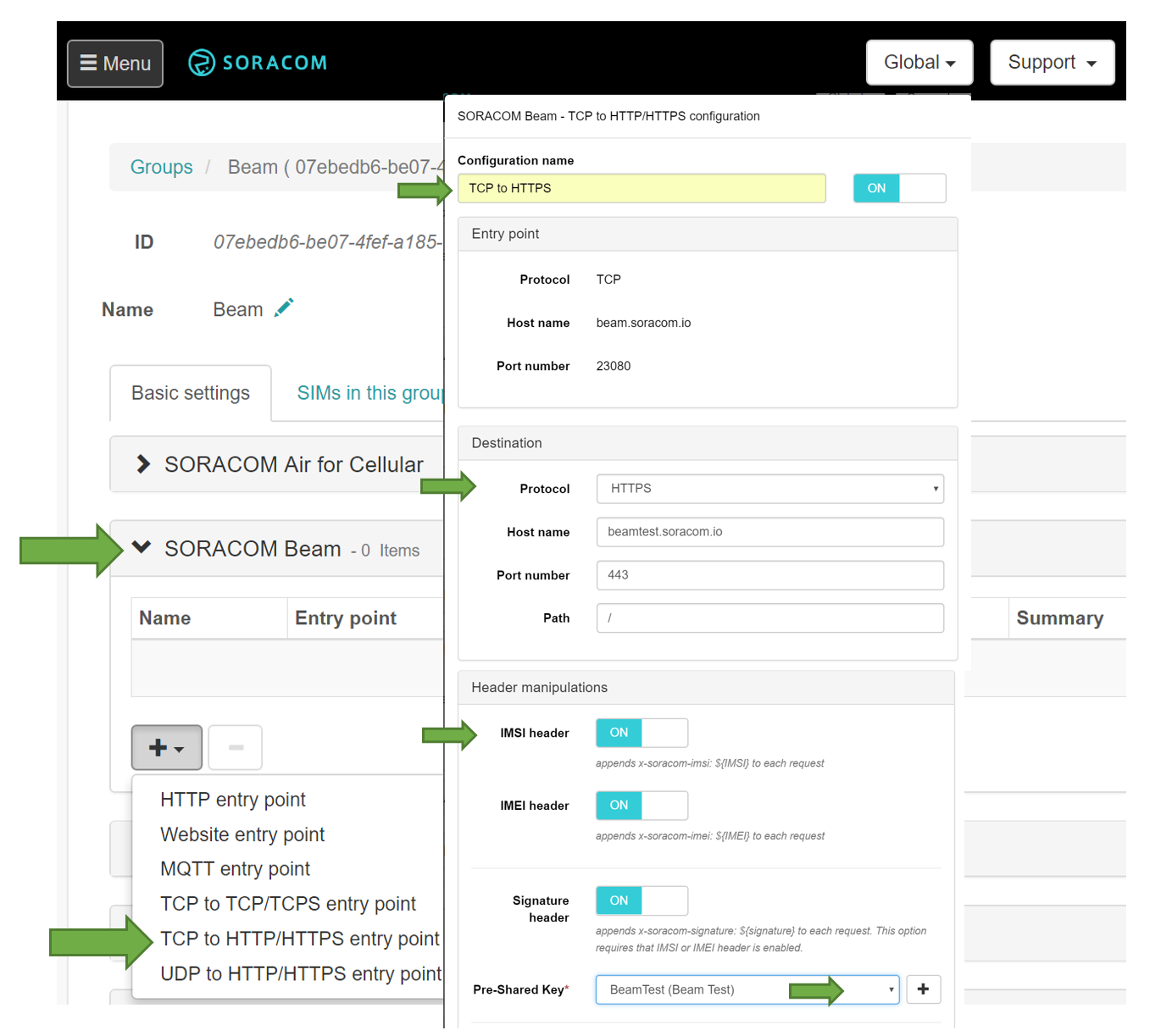
1.Click “Menu” -> "Groups" and click the group name.

2.Expand "SORACOM Beam"

3.Click on "+" button and select "TCP to HTTP/HTTPS" entry point.

4.When "SORACOM Beam - TCP to HTTP/HTTPS configuration" dialog comes up, please fill in the information below

* Configuration name: Any arbitrary name but let's give "TCP to HTTPS"
* Host name: specify "beamtest.soracom.io" (this is our test HTTPS endpoint for this workshop)
* Port number: keep it blank or specify "443" (You can leave "Port number" blank if you use default TCP:443)
* Path: specify "/"
* Header manipulations: IMSI header/IMEI header/Signature header -> ON
* Header manipulations -> Pre-Shared Key: choose the same key which you created in the previous HTTP to HTTPS section.



## 8.Test the connection from Raspberry Pi

You can use nc(netcat) command to send data into raw TCP socket. Please specify beam endpoint and port number in your command parameter.

### Command

nc beam.soracom.io 23080

foobar (or arbitrary string you want to send)

(Ctrl+C to exit)

### Result

pi@raspberrypi:~ $ nc beam.soracom.io 23080

foobar

200 Access Authorized: {"payload"=>"Zm9vYmFyCg=="} => foobar

The server authenticated your SIM by checking the signature and accepted your data in base64 encoded format. If you decode this string, it will match the original string you sent over Beam.

This will end the advanced scenario.

Thank you so much for reaching this far.

If you have any questions regarding your IoT project, please come to soracom.io and use “contact us” form to get to us