## References

* Microsoft Azure IoT SDKs <https://github.com/Azure/azure-iot-sdks>
* Microsoft Azure IoT device SDK for C <https://github.com/Azure/azure-iot-sdks/blob/master/c/readme.md>
* Azure IoT SDKs releases <https://github.com/Azure/azure-iot-sdks/releases>

## Requirements

* Finished the part 1 of HOL
* PuTTY
* Download and install Device Explorer tool (Windows)

<https://github.com/Azure/azure-iot-sdk-csharp/releases>

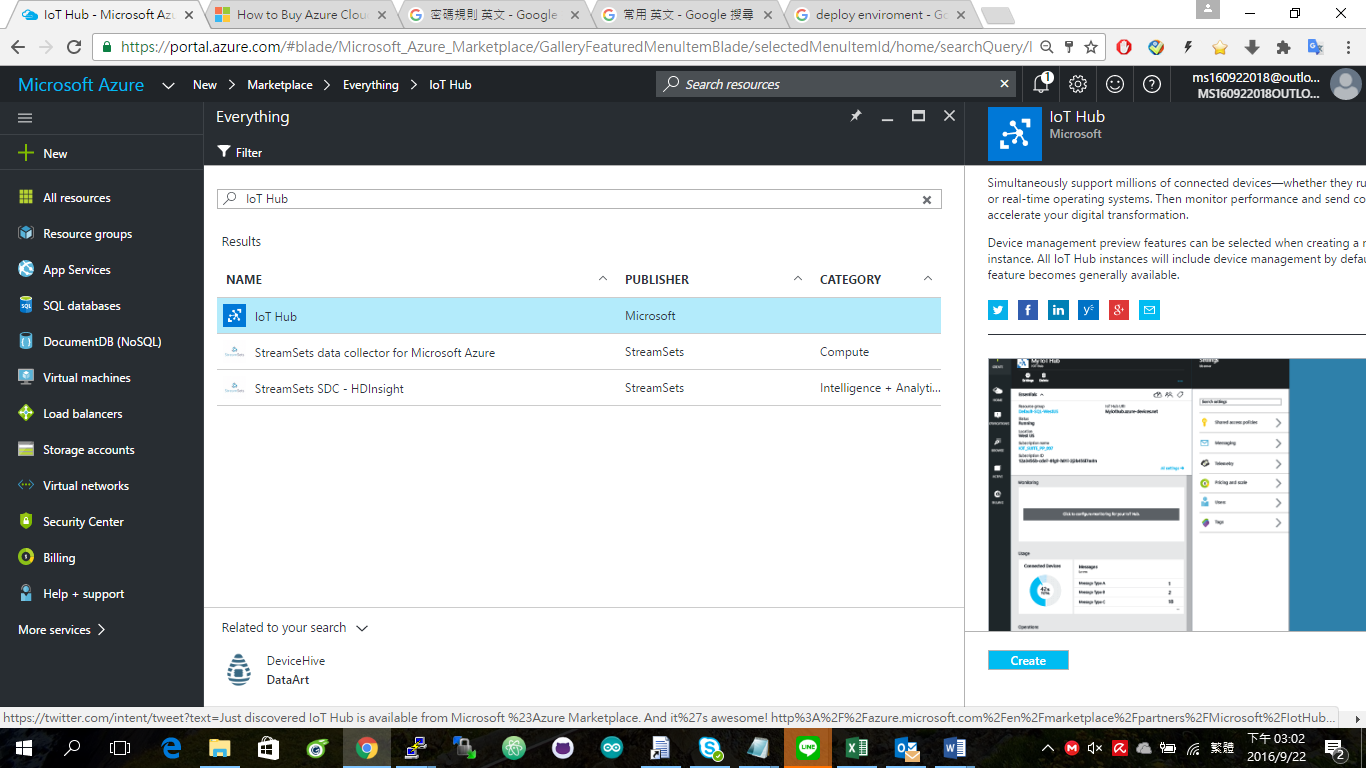
<https://github.com/Azure/azure-iot-sdk-csharp/releases/download/2017-6-30/SetupDeviceExplorer.msi>

## Goals

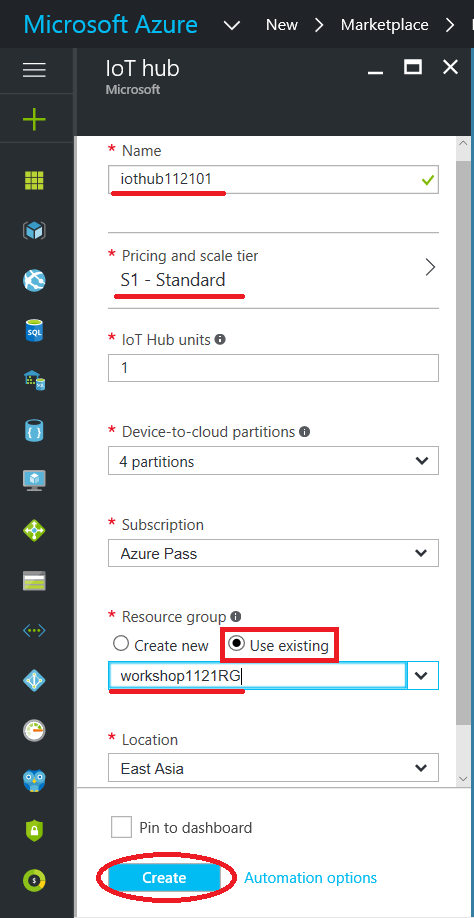
* Provision an Azure IoT Hub
* Setup the IoT SDK deployment environment of Ubuntu
* Build the IoT SDK
* Execute AMQP sample code and connect the data stream between D2C and C2D through the Device Explorer.

## Step 1: Provision an Azure IoT Hub

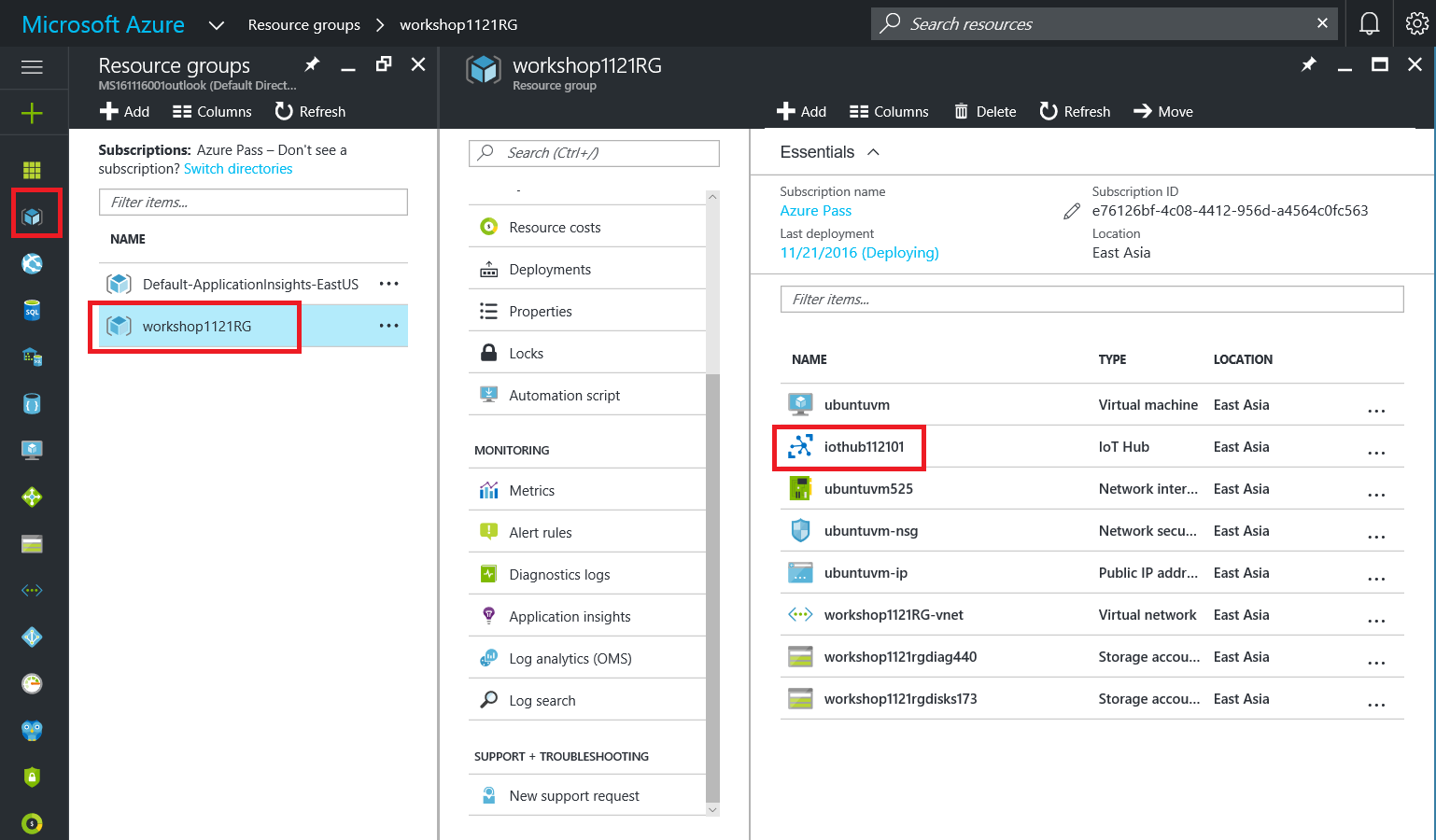
* Login <https://portal.azure.com>
* Search IoT Hub and create new **IoT Hub**

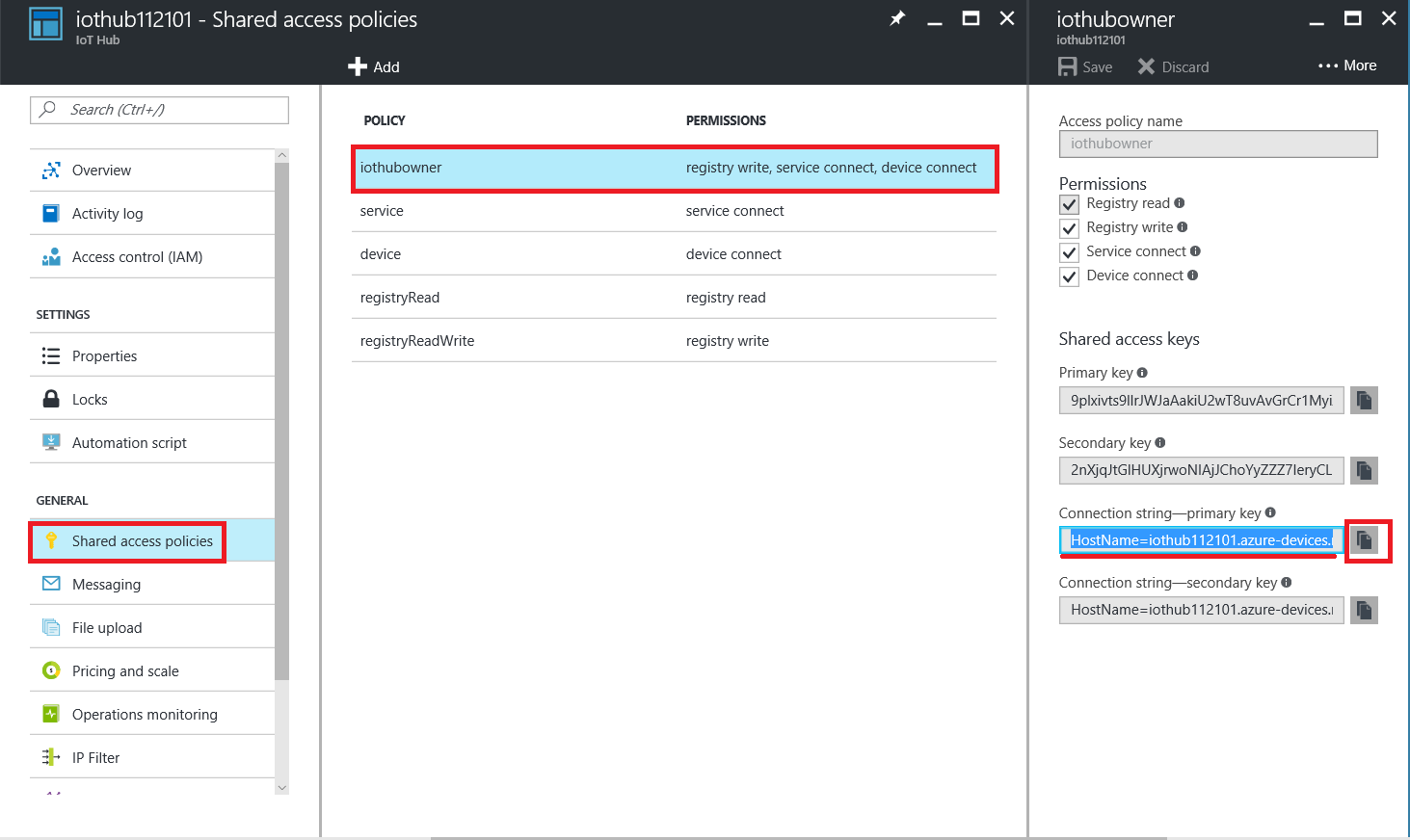


* Input basic information. Here we select **S1 Standard** for workshop (Deploying IOT hub may takes few minutes)
  + Suggest you to use the same resource group where your VM exists, in order to easy to manage your solutions.
  + Suggest you to select same location where your VM exists, like **East Asia**.

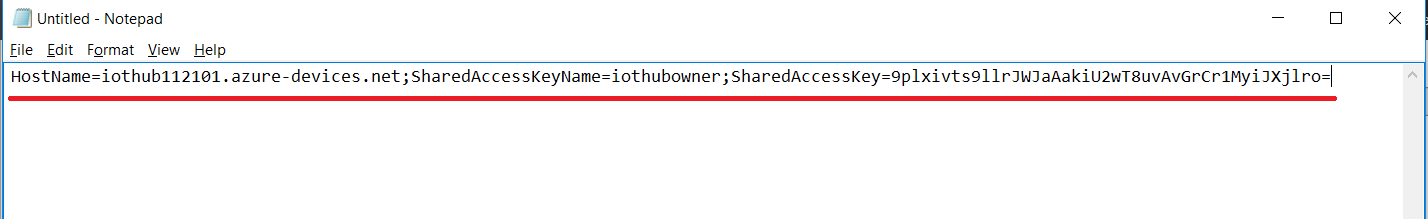


* Here you can get the connection string of IoT Hub
  + Select the **Share access policies**
  + Copy the **Connection String - primary key**





* + Save the **Connection String - primary key** for the later used.



## Step 2: Setup the deployment environment of Ubuntu

* Development tools
  + cmake version 3.x or higher
  + gcc version 4.9 or higher
  + git
* install the cmake
  + SSH connect to Ubuntu, and update the apt-get package.

(copy the marked commands in yellow and go to Putty, right-click on your mouse to paste)

sudo apt-get install software-properties-common

sudo add-apt-repository ppa:george-edison55/cmake-3.x

<Press [ENTER] to continue>

sudo apt-get update

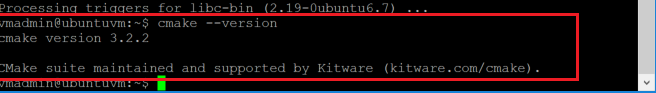
* + Install the cmake

sudo apt-get install cmake

<Press [y] to continue>

* + Check the version of cmake

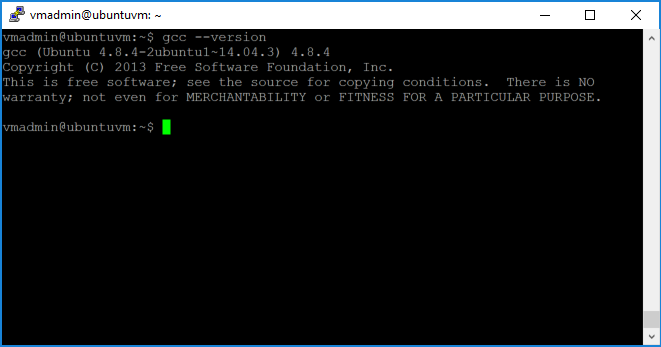
cmake --version



* Install the gcc
  + Check the version of gcc

gcc --version

<default 4.8.4>



* + Update the version to 4.9

sudo add-apt-repository ppa:ubuntu-toolchain-r/test

<Press [ENTER] to continue>

sudo apt-get update

sudo apt-get install gcc-4.9 g++-4.9

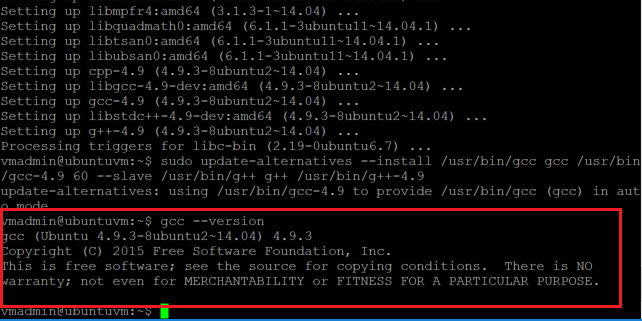
<Press [y] to continue>

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.9 60 --slave /usr/bin/g++ g++ /usr/bin/g++-4.9

* + Successfully installed, check the version.

gcc --version

<Current available version online is 4.9.4, below shows 4.9.3>



* Install the git

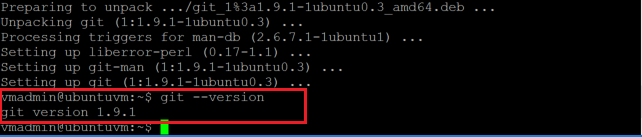
sudo apt-get install git

<Press [y] to continue>

* + Successfully installed, check the version

git --version

< version 1.9.1>

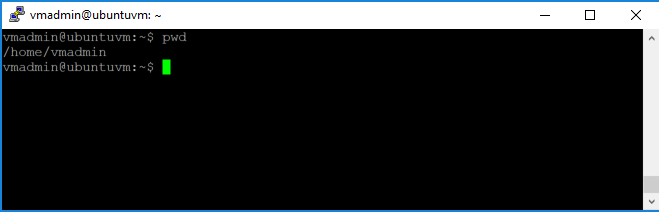


## Step 3: Build Azure IoT Device SDK for C

* Clone the latest Azure IoT SDK on GitHub
  + Confirm the path of repo, the default would be /home/vmadmin

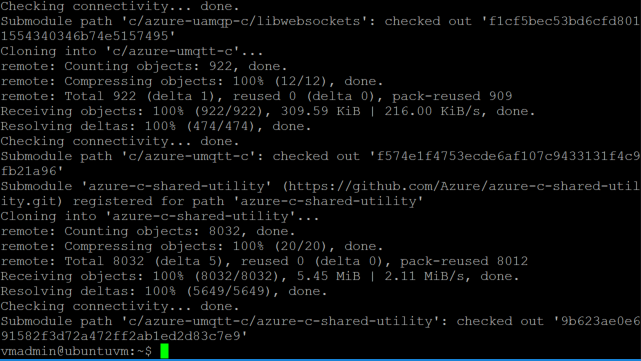
pwd

<default:/home/vmadmin>



* + Clone the source from GitHub

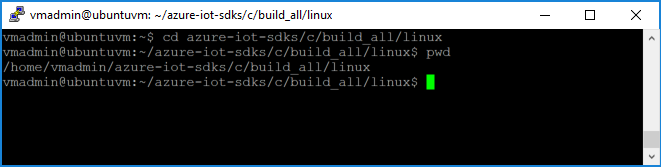
git clone --recursive https://github.com/Azure/azure-iot-sdk-c.git



* Set up the deployment environment
  + Change the directory to build folder

cd azure-iot-sdk-c/build\_all/linux

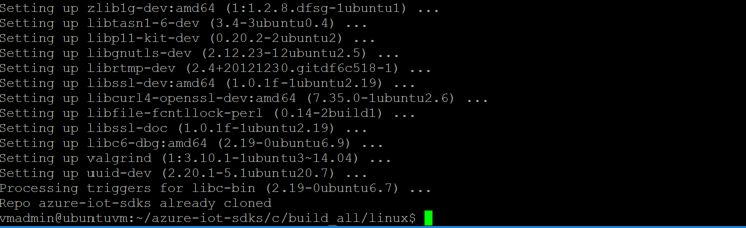
<default:/home/vmadmin/azure-iot-sdk-c/build\_all/linux>



* Set up the variables and environment of development

sudo ./setup.sh

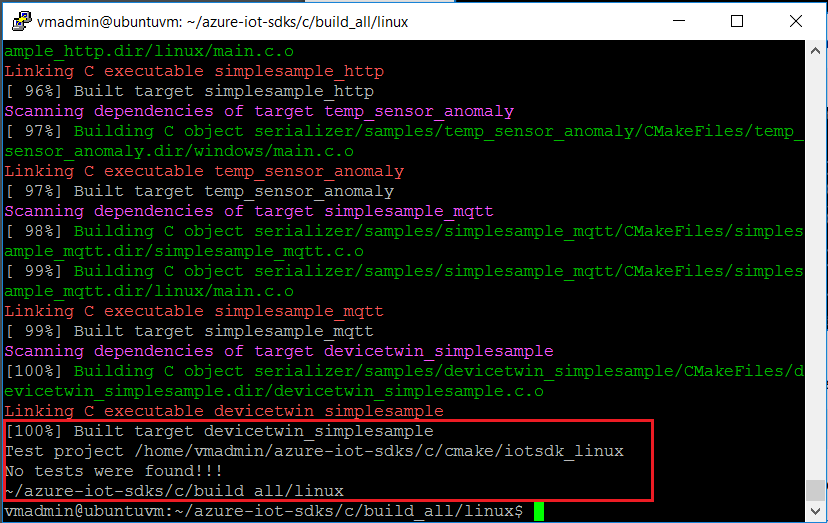
< Check the result of output >



* Build the code and skip the unit tests. (Please wait a few minutes)

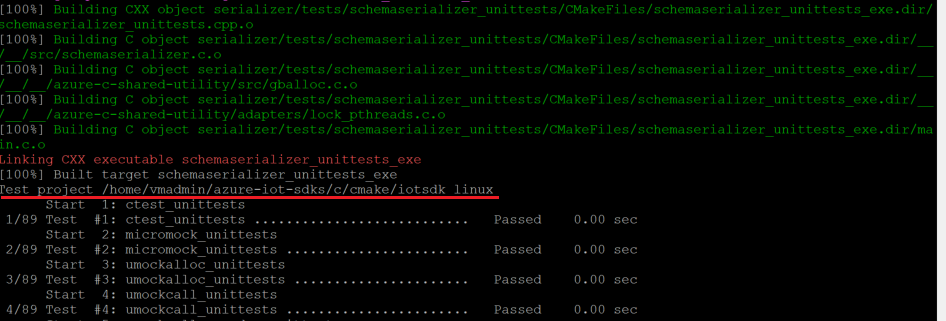
sudo ./build.sh

<View the output of build>

* + Check the output of build 
  + The output directory of build:

**/home/<user>/azure-iot-sdk-c/cmake/iotsdk\_linux**

* + - You can change output directory in build.sh.

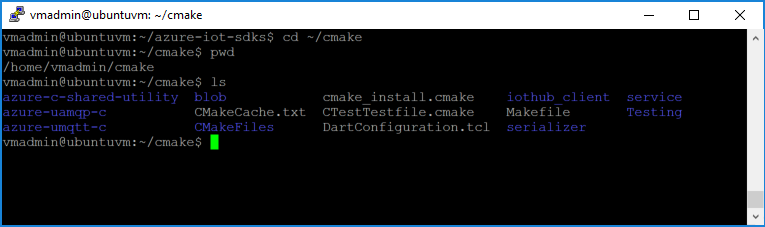


* Switch to output directory of build and confirm the result.

cd ~/azure-iot-sdk-c/cmake/iotsdk\_linux

ls

< Confirm result successful >

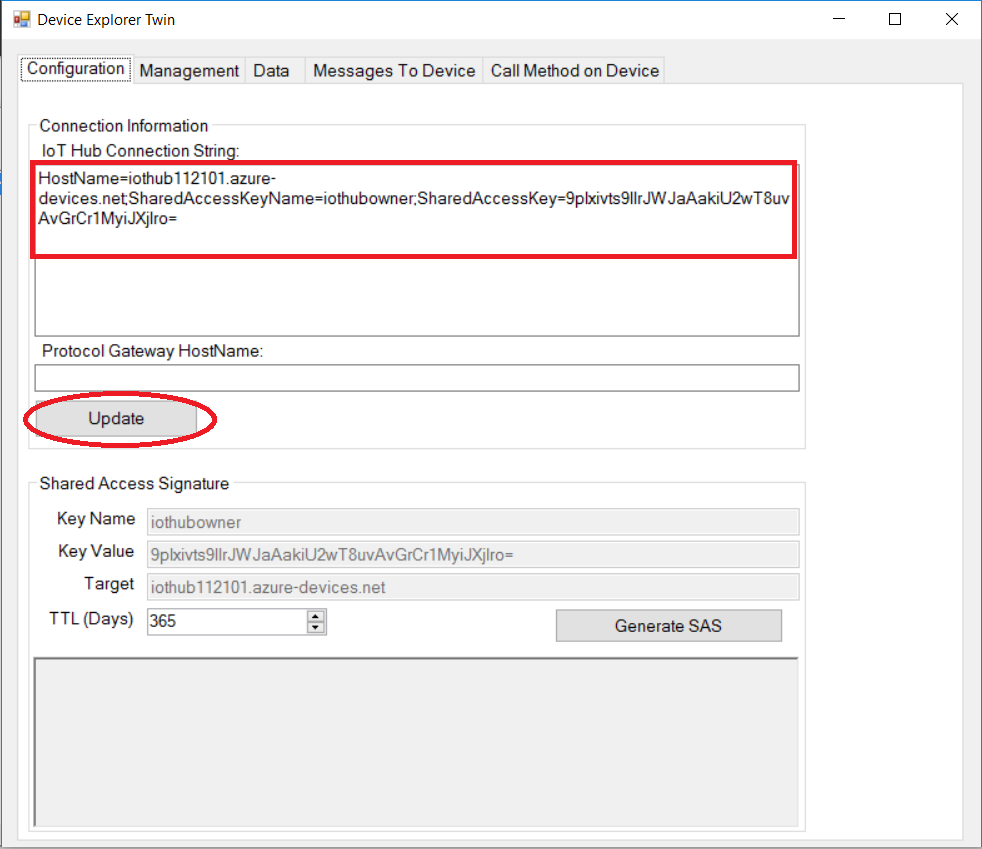


**Note: Every time you run build.sh, it deletes and then recreates the "cmake" folder in your Azure IoT SDK directory.**

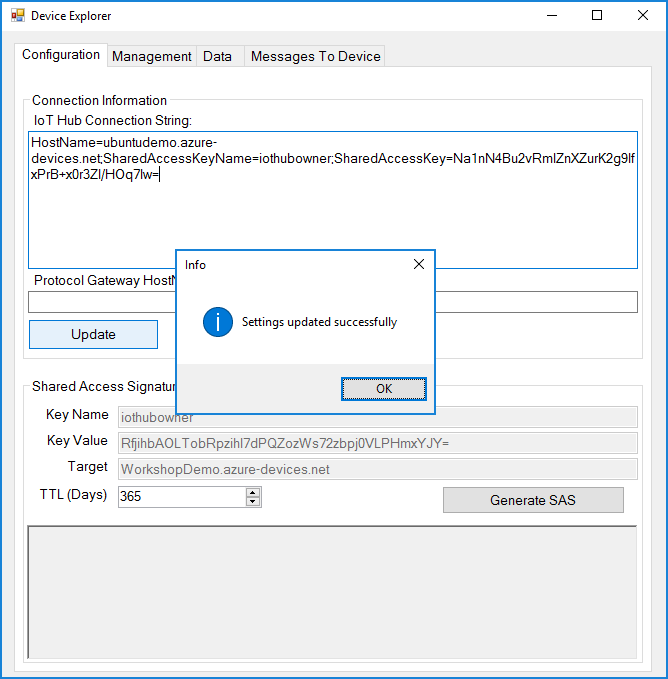
**Note: You will not be able to run the samples until you configure them with a valid IoT Hub device connection string.**

## Step 4: Execute the C sample code of AMQP

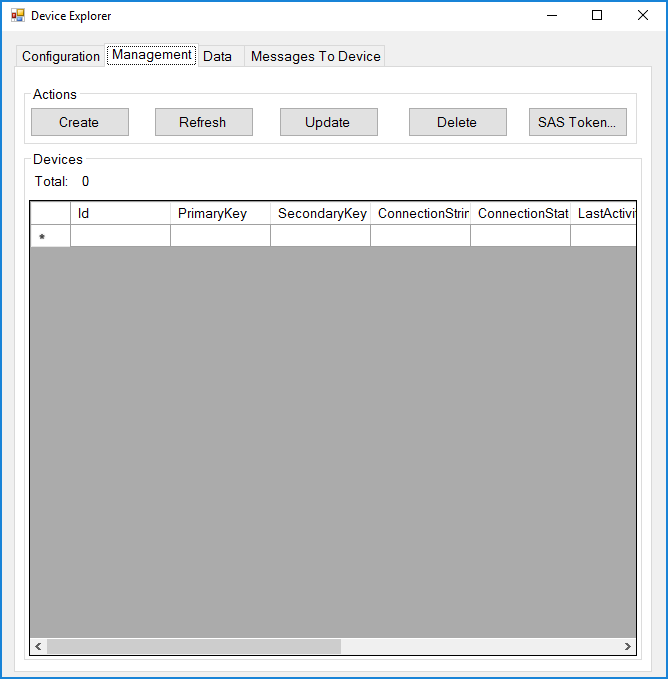
* Before the execution, you have to register your device in IoT hub, and get the connection string of device (for the device connected to Azure). You can go through the either way as below:
  + Azure IoT Service SDK ([C#](https://github.com/Azure/azure-iot-sdks/blob/master/csharp/service/README.md), [Java](https://github.com/Azure/azure-iot-sdks/blob/master/java/service/readme.md), [Node.js](https://github.com/Azure/azure-iot-sdks/blob/master/node/service/README.md))
  + tool
    - The cross-platform (Node.js language), command-line based, [iothub-explorer](https://github.com/Azure/azure-iot-sdks/blob/master/doc/manage_iot_hub.md#iothub-explorer) tool.
    - [Device Explorer](https://github.com/Azure/azure-iot-sdks/blob/master/doc/manage_iot_hub.md#device-explorer), a Windows-only graphical tool.
* This time, we use the **Device Explorer** in this workshop. First, we need to provision a device from the Device Explorer tool.
  + In the **Configuration** tab, paste the Connection string of IoT Hub then click **Update**.



* + Update successfully

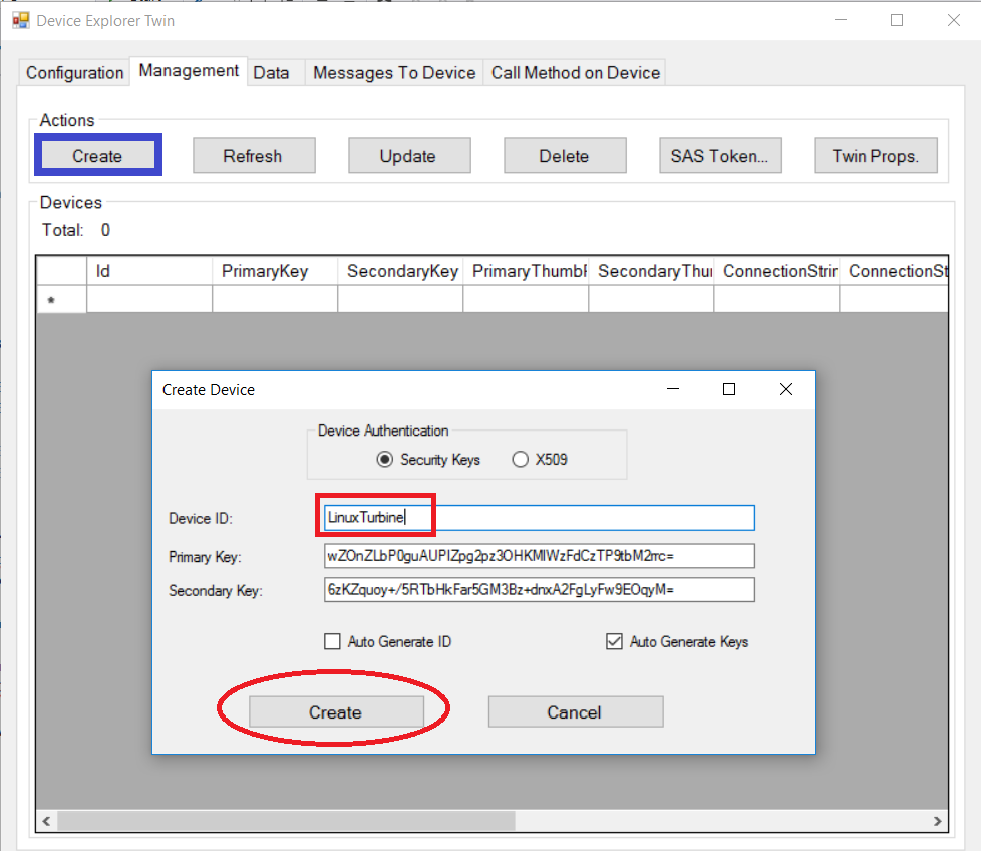


* + In the **Management** tab, you can manage the devices which were connected to IoT Hub.

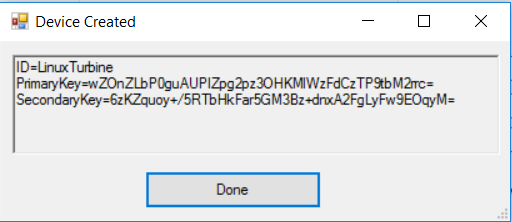


* + **Create** the simulated device of Linux, and named **LinuxTurbine**.

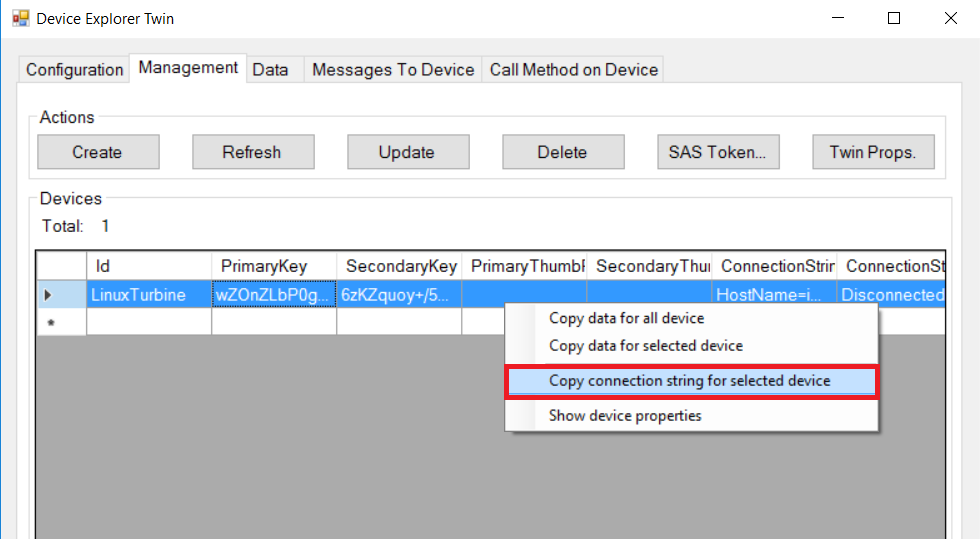
(**LinuxTurbine**, the name **must be fixed** in this workshop)



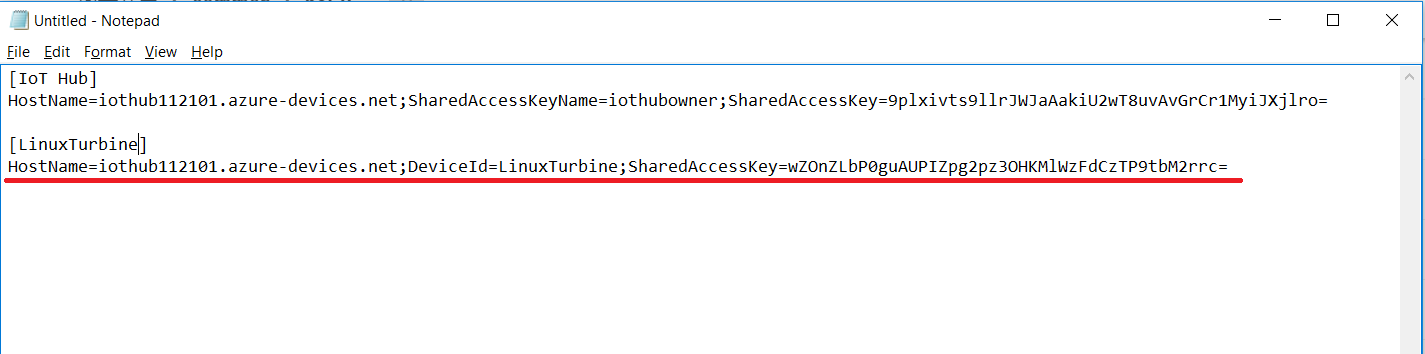
* + Created successfully



* + Wait for the list was updated, select the **LinuxTurbine** and right click to copy the connection string of device.



* + Save the connection string of device, and AMQP simulated device will use it later on.

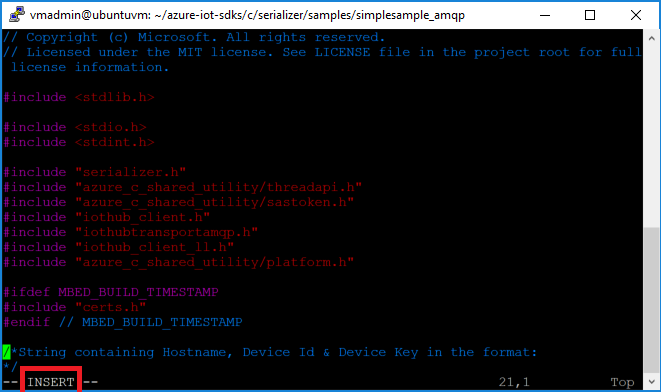


* Modify the C simple sample of AMQP in Azure IoT Device SDK
  + Edit the **simplesample\_amqp.c**

cd ~/azure-iot-sdk-c/serializer/samples/simplesample\_amqp

vi simplesample\_amqp.c

* + Type “a” to switch the editor mode.

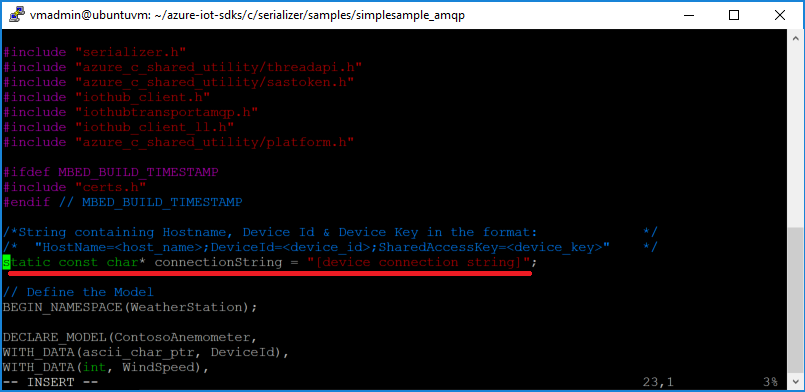


* + Update your connection string of device. Please search the code as below.

/\*String containing Hostname, Device Id & Device Key in the format: \*/

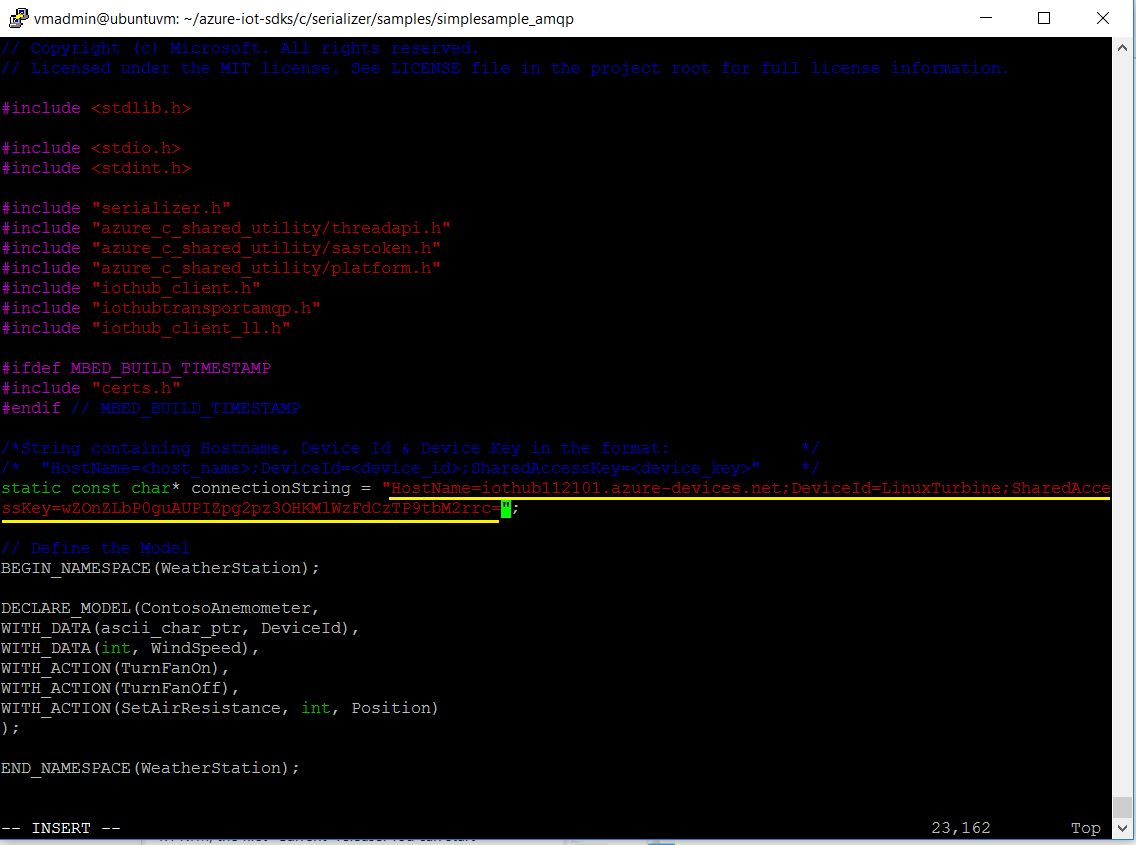
/\* "HostName=<host\_name>;DeviceId=<device\_id>;SharedAccessKey=<device\_key>" \*/

static const char\* connectionString = "[device connection string]";

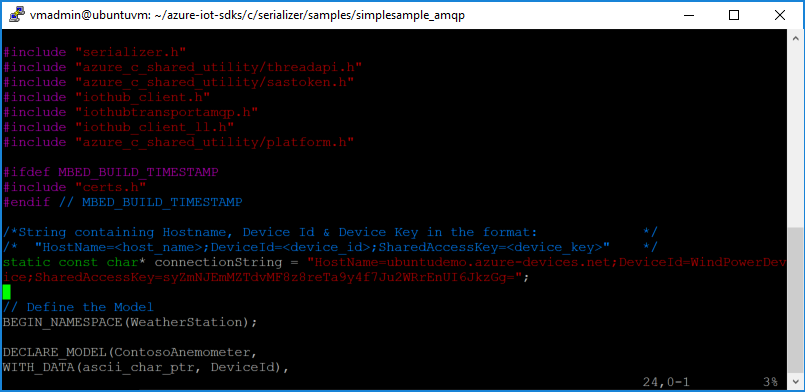


Paste your connection string of device (LinuxTurbine) which was copied from the previous.

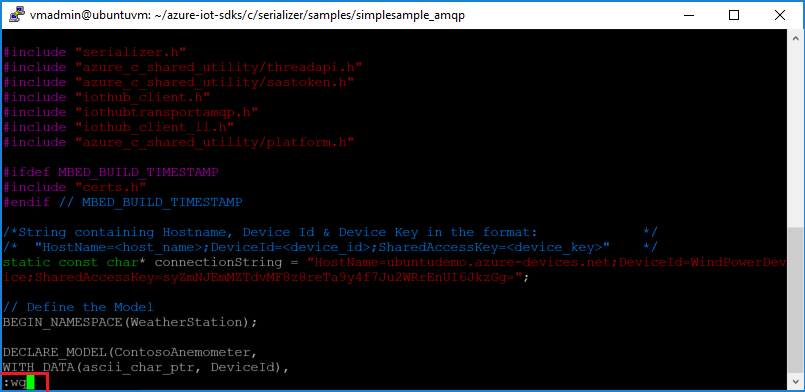
static const char\* connectionString = "HostName=iothub112101.azure-devices.net;DeviceId=LinuxTurbine;SharedAccessKey=wZOnZLbP0guAUPIZpg2pz3OHKMlWzFdCzTP9tbM2rrc=";



* + Press the Esc key to back the normal mode.



* + Type the :wq, save it then exit.

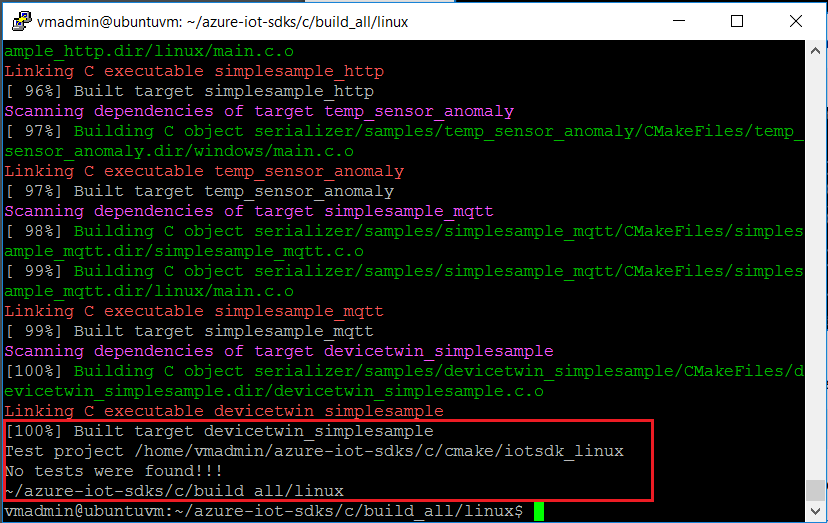


* + Build the code again

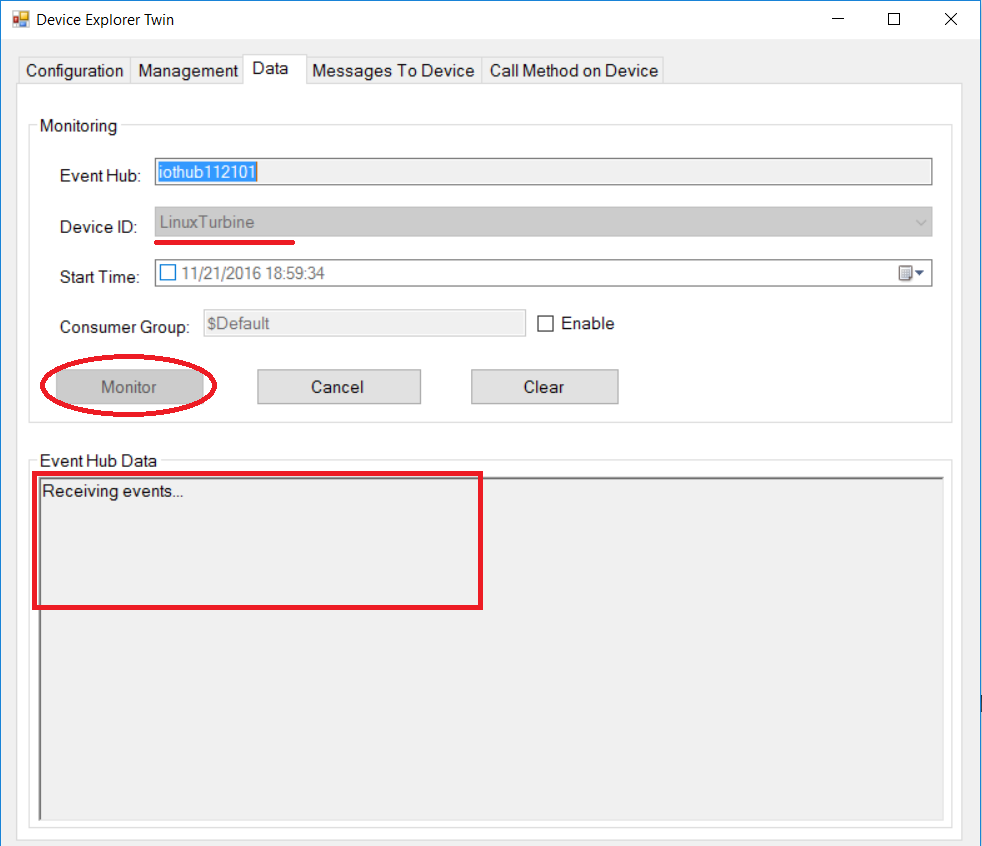
cd ~/azure-iot-sdk-c/build\_all/linux

sudo ./build.sh

* + Check the output of build



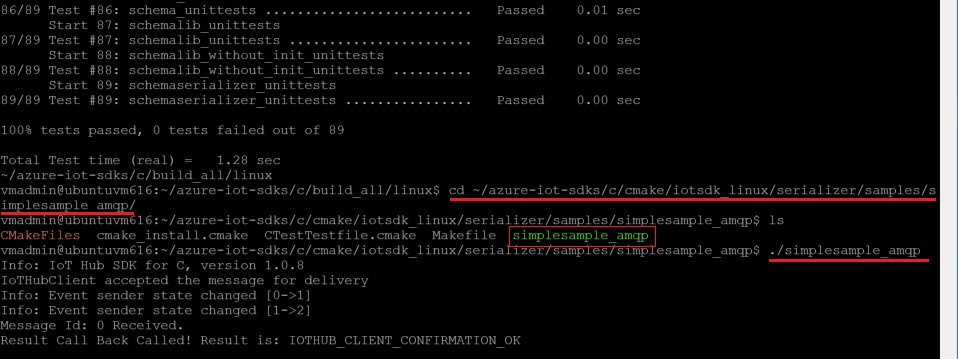
* Open the **Device Explorer** to monitor the data of device
  + In the **Data** tab, select the device which we wanted, then click the **Monitor** button.



* Run AMQP Simple Sample in C SDK

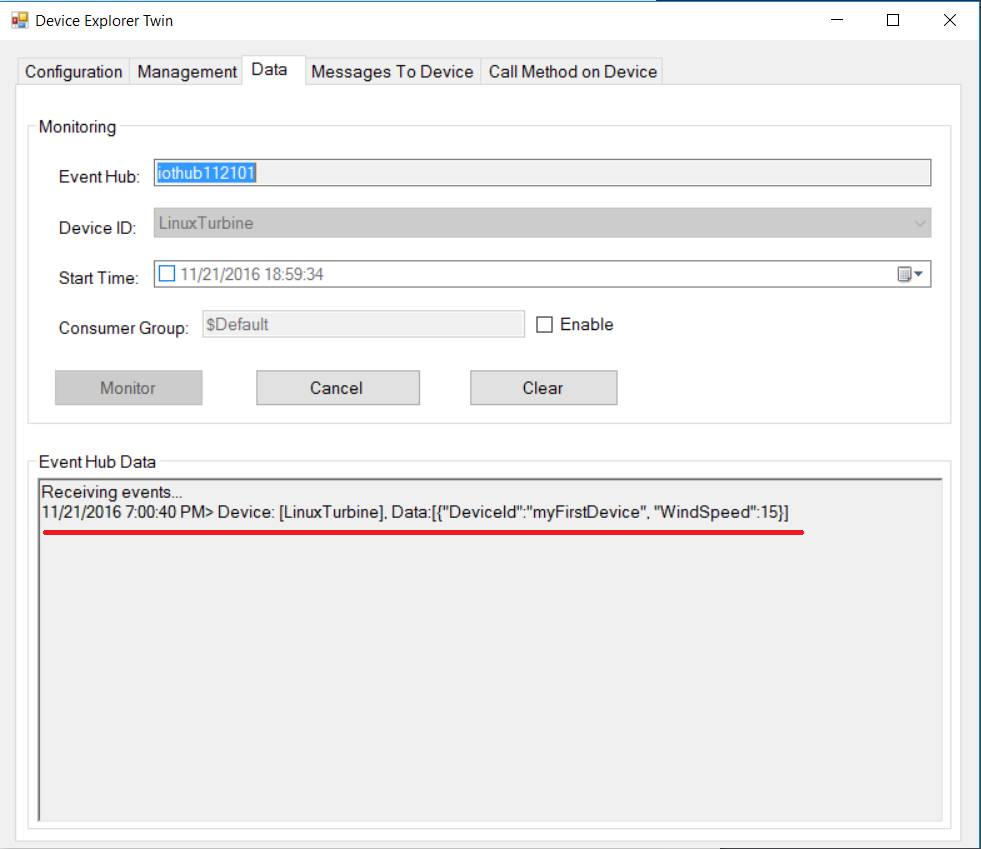
cd ~/azure-iot-sdk-c/cmake/iotsdk\_linux/serializer/samples/simplesample\_amqp/

./simplesample\_amqp

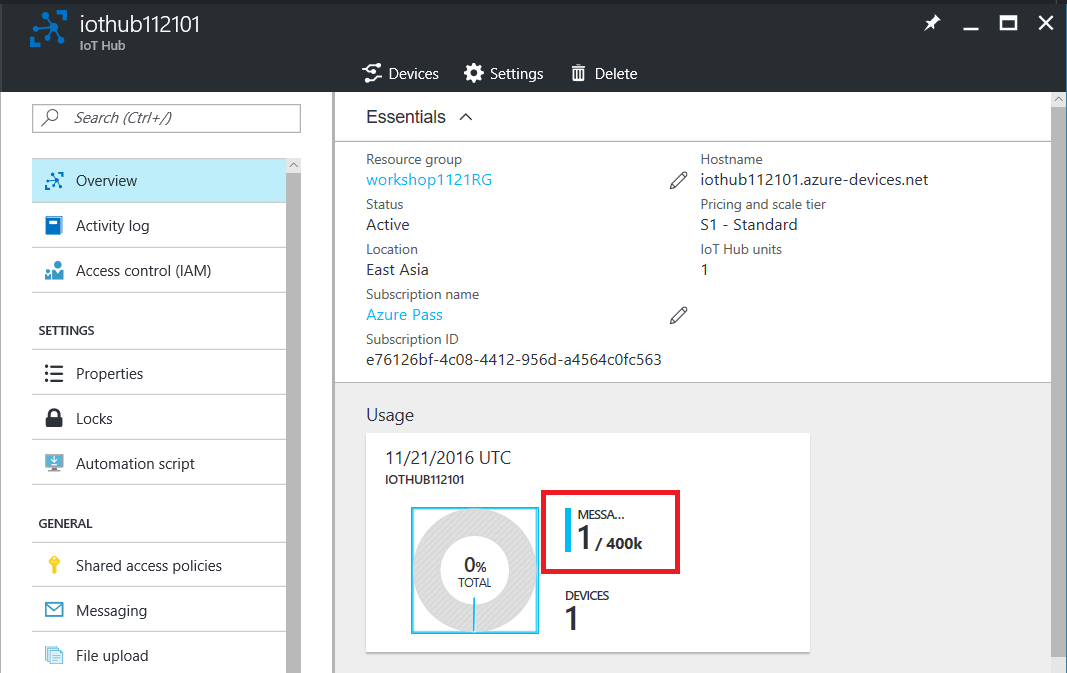


**Note: Sample code Only send one-time data to IoT Hub when it executes.**

* Check the receiving data from the device (Device to Cloud)
  + Device Explorer



* + Azure Portal

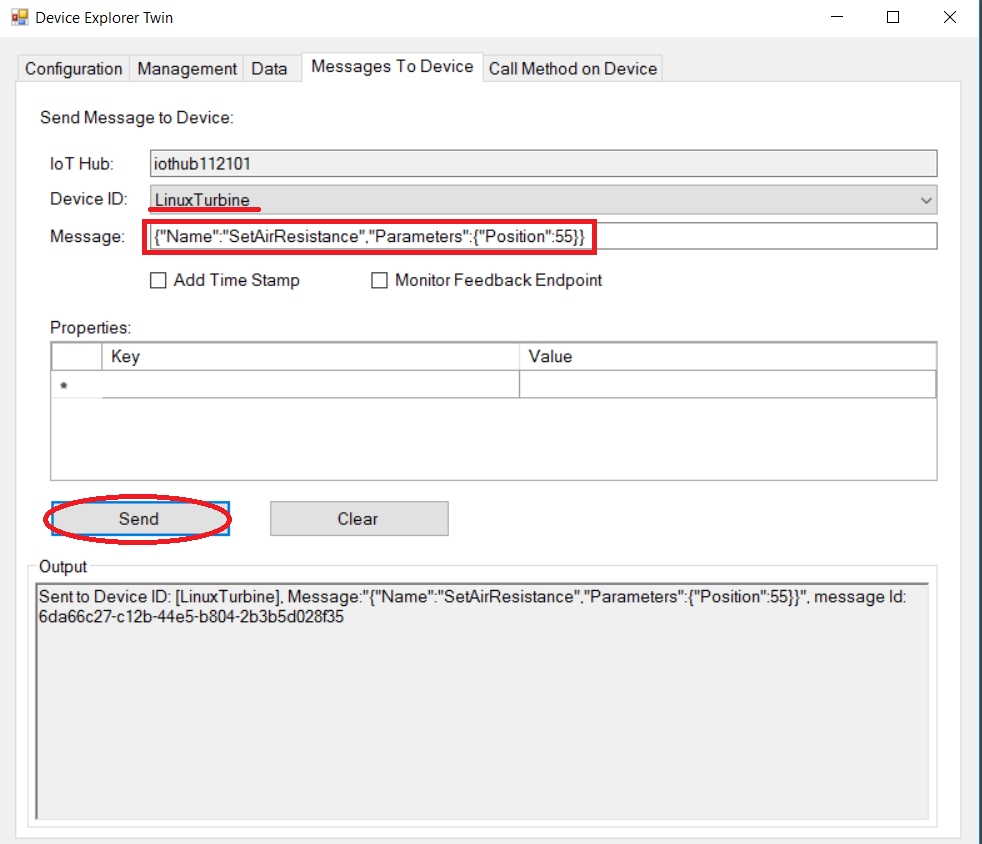


* Send the command to device (Cloud to Device)
  + Open the Device Explorer
    - In the **Message To Device** tab, you can try these commands in the Message field as below (JSON format).

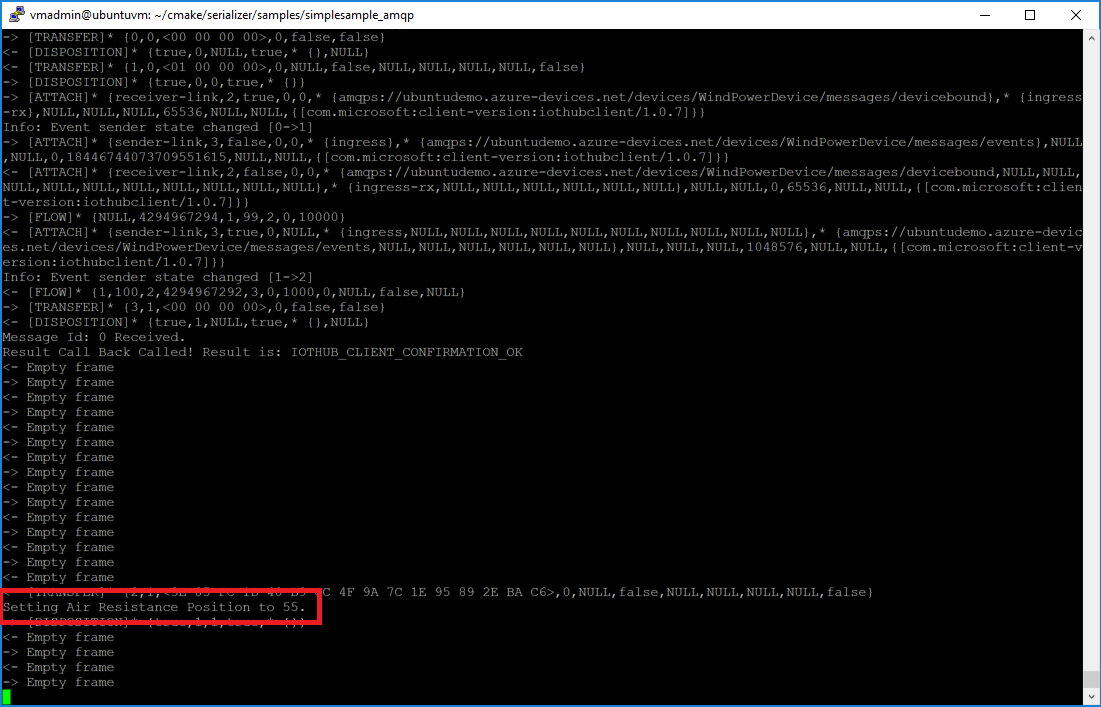
{"Name":"SetAirResistance","Parameters":{"Position":55}}

{"Name":"TurnFanOn","Parameters":{}}

{"Name":"TurnFanOff","Parameters":{}}



* + Confirm the received message from the console output of device



* *The HOL 2 has been completed. Now you can send and receive the data between device and cloud through Azure IoT Hub. Also you are ready to get* ***Azure Certified for IoT****.*

## Q & A

* How to install cmake 3.2 on ubuntu 14.04?
  + Here is the [link](http://askubuntu.com/questions/610291/how-to-install-cmake-3-2-on-ubuntu-14-04).
* How do I use the latest GCC on Ubuntu 14.04?
  + Here is the [link](http://askubuntu.com/questions/466651/how-do-i-use-the-latest-gcc-on-ubuntu-14-04).