Vietnam National University of Ho Chi Minh City
Ho Chi Minh University of Technology
Computer Science - Engineering Department

Multi-disciplinary Project

Tutorial

MQTT Broker installation on Azure

along with Publish - Subscriber simulation

Instructor: Assoc. Prof. Dr. Quản Thành Thơ
Contributors: Nguyễn Dương Minh Tâm Đạt - 1710059
Hồ Minh Hoàng - 1710094
Nguyễn Việt Long - 1712025
Nguyễn Ngọc Thu Phương - 1712725
Dương Đức Tín - 1710332
Nguyễn Trần Công Duy - 1710043
Nguyễn Thành Thông - 1710313
Ngô Trọng Khôi - 1710150
Cao Đức Hùng - 1710114
Trần Huy - 1711552

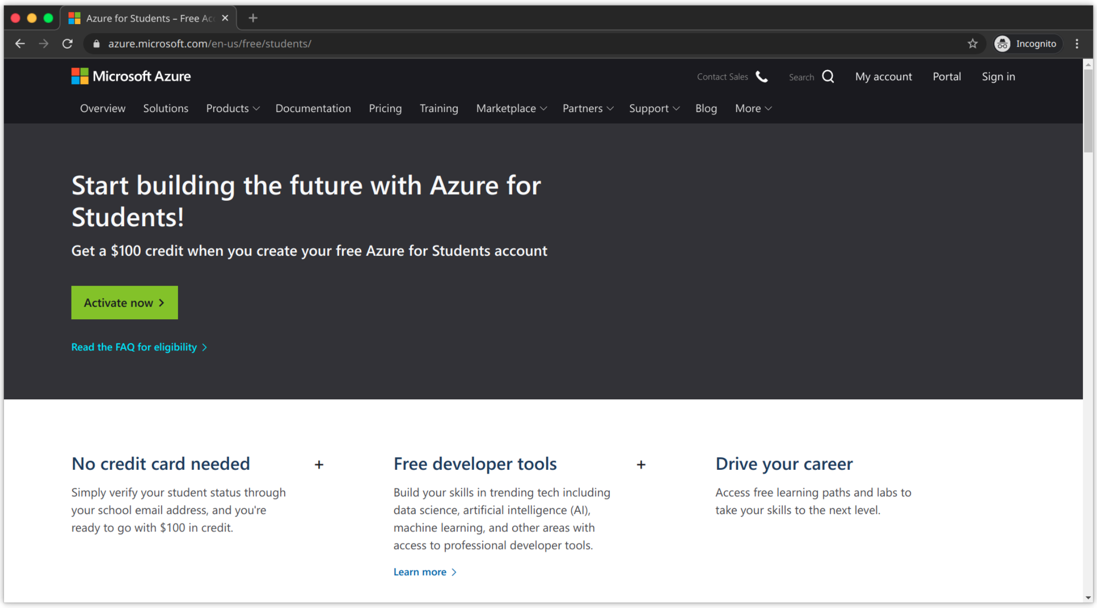
Ho Chi Minh City, May 2020

Contents

1 Server Setup

1.1 Azure for Students

At first, access Azure Student's website: https://azure.microsoft.com/en-us/free/students/.



Start building the future with Azure for Students!

Get a $100 credit when you create your free Azure for Students account

No credit card needed

Simply verify your student status through your school email address, and you're ready to go with $100 in credit.

Free developer tools

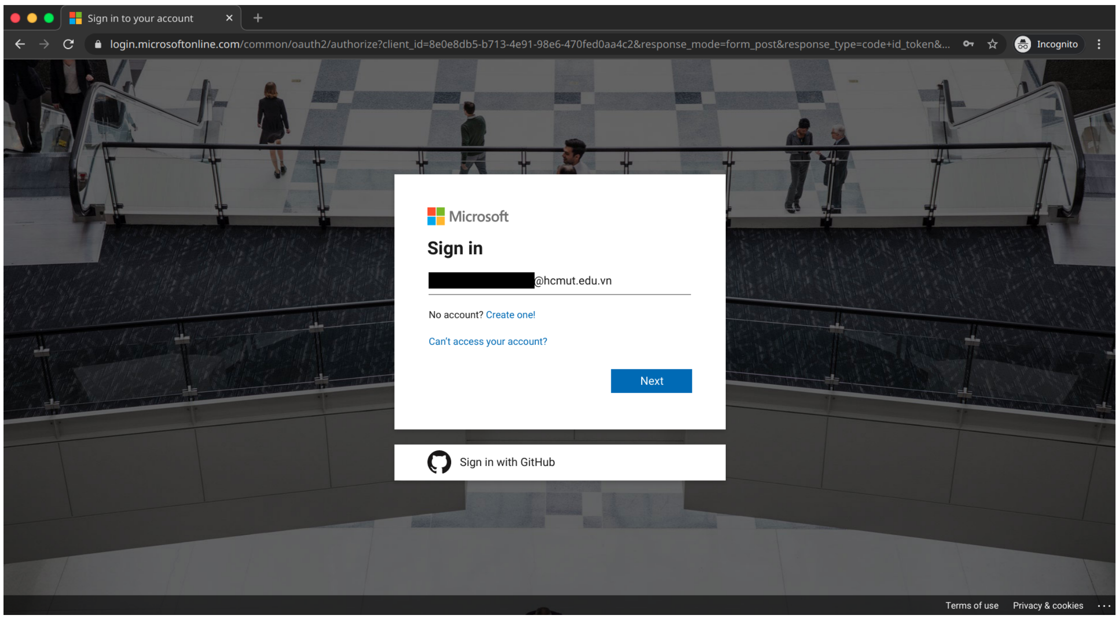
Build your skills in trending tech including data science, artificial intelligence (AI), machine learning, and other areas with access to professional developer tools.

Drive your career

Access free learning paths and labs to take your skills to the next level.

Figure 1: Azure Student User Interface

In this step, you will register an account pertaining to your university email.

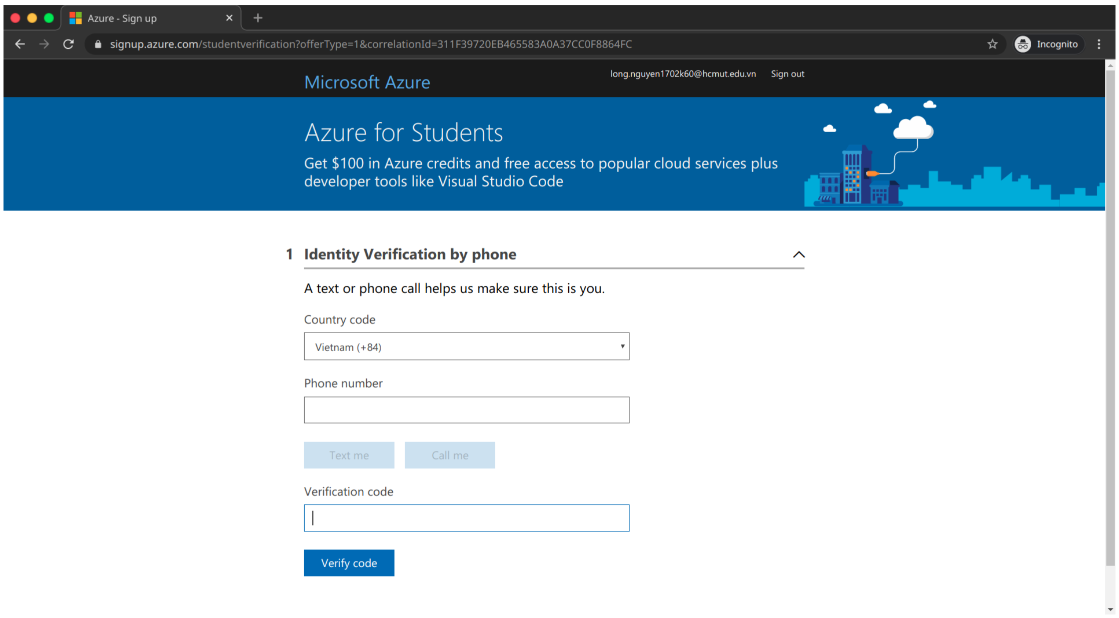


Sign in to your account
login.microsoftonline.com/common/oauth2/authorize?client_id=8e0e8db5-b713-4e91-98e6-470fed0aa4c2&response_mode=form_post&response_type=code+id_token&...

Microsoft
Sign in
[redacted]@hcmut.edu.vn
No account? Create one!
Can't access your account?
Next
Sign in with GitHub

Figure 2: Fill in your email

After successfully signing up, you are requested to verify your phone number.



Azure - Sign up
signup.azure.com/studentverification?offerType=1&correlationId=311F39720EB465583A0A37CC0F8864FC

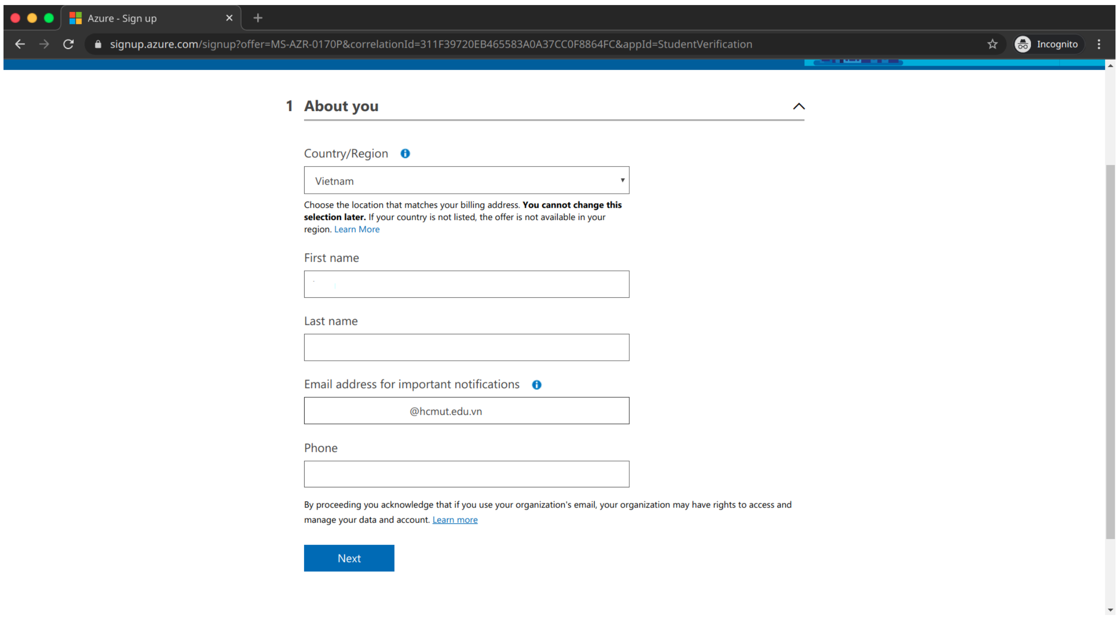
Microsoft Azure
long.nguyen1702k60@hcmut.edu.vn Sign out

Azure for Students
Get $100 in Azure credits and free access to popular cloud services plus developer tools like Visual Studio Code

1 Identity Verification by phone
A text or phone call helps us make sure this is you.
Country code: Vietnam (+84)
Phone number
Text me
Call me
Verification code
Verify code

Figure 3: Phone Number verification

Afterwards, let's enter some personal information and obviously, absolutely agree with all Microsoft's terms and conditions!


Azure - Sign up

signup.azure.com/signup?offer=MS-AZR-0170P&correlationId=311F39720EB465583A0A37CC0F8864FC&appId=StudentVerification

1 About you

Country/Region: Vietnam

Choose the location that matches your billing address. **You cannot change this selection later.** If your country is not listed, the offer is not available in your region. Learn More

First name

Last name

Email address for important notifications: @hcmut.edu.vn

Phone

By proceeding you acknowledge that if you use your organization's email, your organization may have rights to access and manage your data and account. Learn more

Figure 4: Enter personal information

1.2 SSH key generation

After creating our own account, we need to generate the key so that the server can accept our connection through Secure Shell (SSH). At the outset, open up your terminal (Linux), and execute the following command:

```
ssh-keygen -t rsa -b 4096
```

By default, your key is stored in `~/.ssh/`. Open *.pub comprising your public key, copy all of the information, which is on demand for the future steps.

1.3 Virtual Machine Setup

Microsoft provides a quick and simple interface to create our own VM.

- Enter Azure portal: https://portal.azure.com.

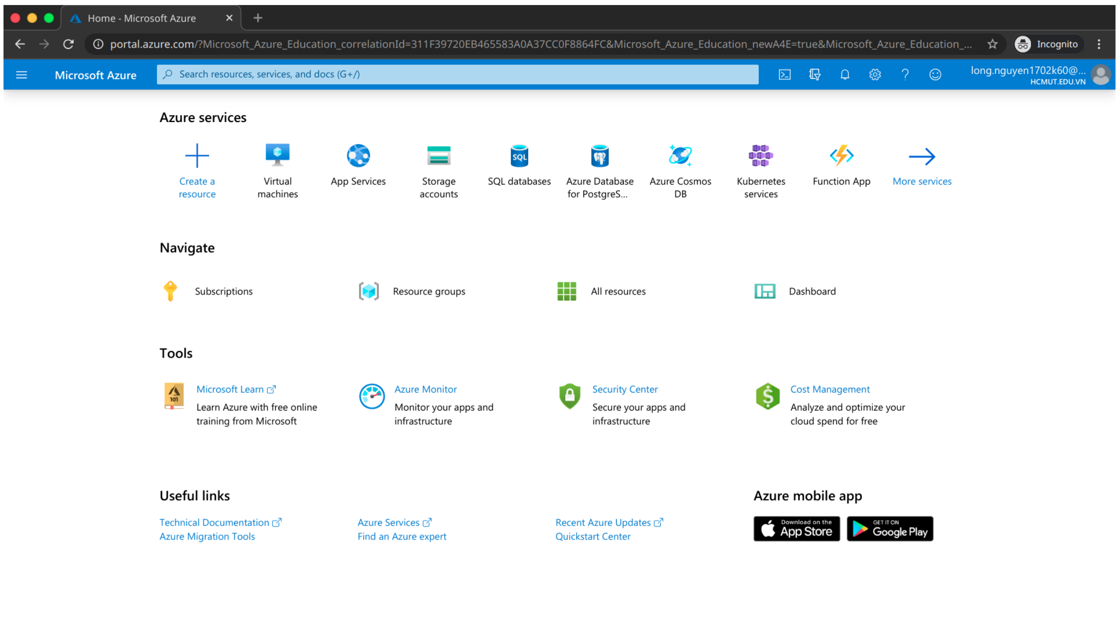
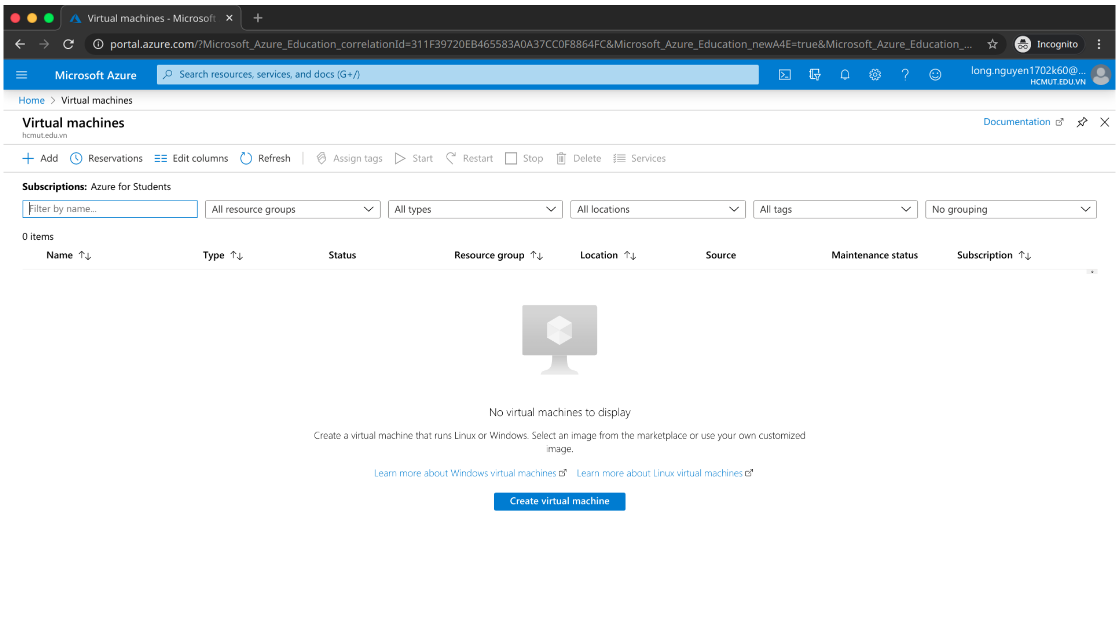


Figure 5: Azure portal

- In the Azure portal, search for and select **Virtual Machines**.



portal.azure.com/?Microsoft_Azure_Education_correlationId=311F39720EB465583A0A37CC0F8864FC&Microsoft_Azure_Education_newA4E=true&Microsoft_Azure_Education_...

Virtual machines
hcmut.edu.vn

Subscriptions: Azure for Students

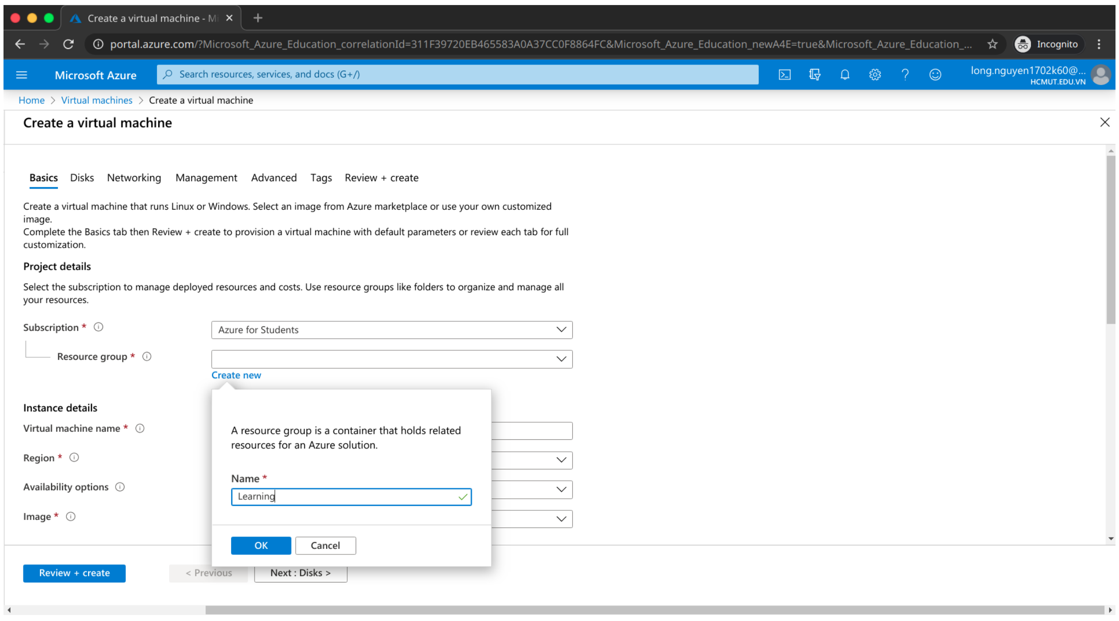
0 items

No virtual machines to display

Create a virtual machine that runs Linux or Windows. Select an image from the marketplace or use your own customized image.

Figure 6: Virtual Machine management

- Set up necessary information for your machine. One thing to note is that you shall pick B1s as your **VM size**, since your account belongs to Student package.



portal.azure.com/?Microsoft_Azure_Education_correlationId=311F39720EB465583A0A37CC0F8864FC&Microsoft_Azure_Education_newA4E=true&Microsoft_Azure_Education_...

Create a virtual machine

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image.
Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Azure for Students

Resource group * :

Instance details

Virtual machine name *

Region *

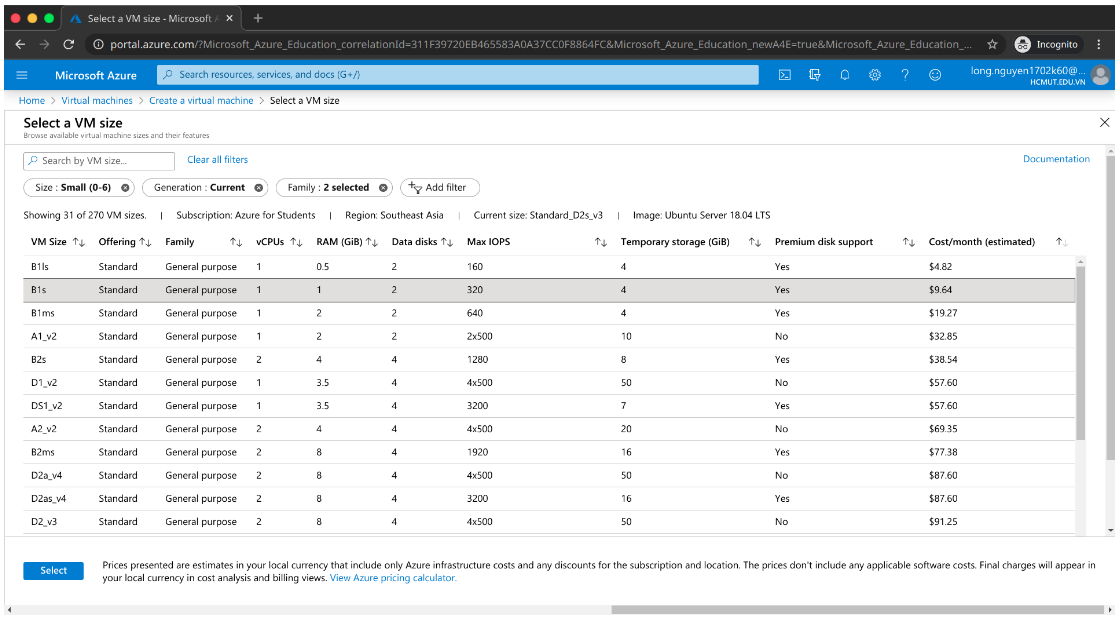
Availability options

Image *

A resource group is a container that holds related resources for an Azure solution.

Name * : Learning

Figure 7: Setup your machine



Select a VM size - Microsoft

portal.azure.com/?Microsoft_Azure_Education_correlationId=311F39720EB465583A0A37CC0F8864FC&Microsoft_Azure_Education_newA4E=true&Microsoft_Azure_Education_...

Home > Virtual machines > Create a virtual machine > Select a VM size

Select a VM size

Browse available virtual machine sizes and their features

Size : Small (0-6) | Generation : Current | Family : 2 selected

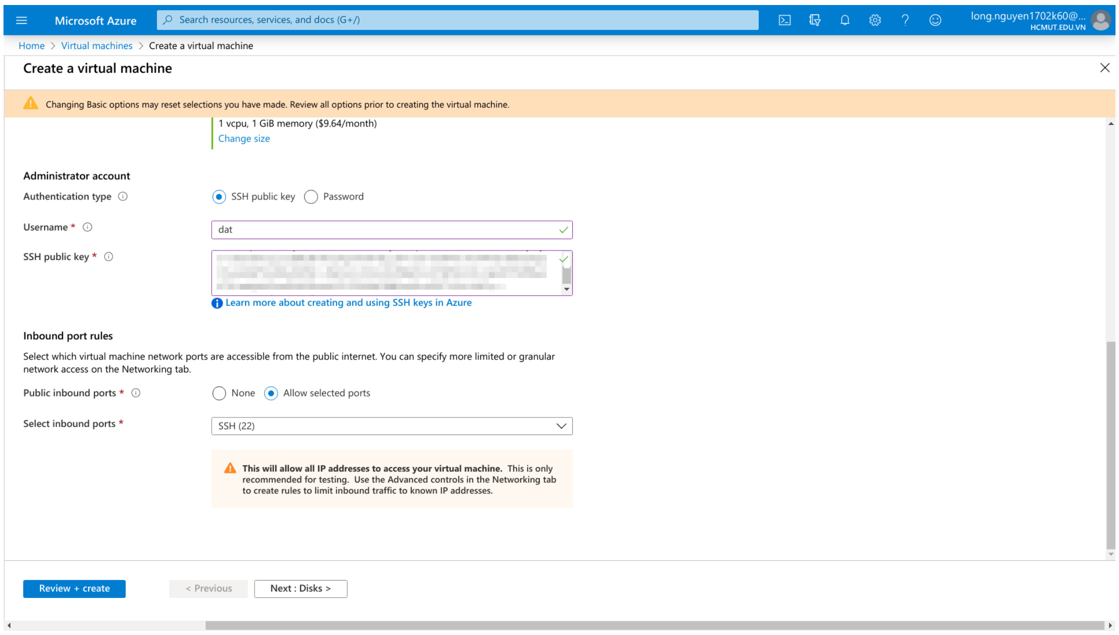
Showing 31 of 270 VM sizes. | Subscription: Azure for Students | Region: Southeast Asia | Current size: Standard_D2s_v3 | Image: Ubuntu Server 18.04 LTS

VM Size	Offering	Family	vCPUs	RAM (GiB)	Data disks	Max IOPS	Temporary storage (GiB)	Premium disk support	Cost/month (estimated)
B1ls	Standard	General purpose	1	0.5	2	160	4	Yes	$4.82
B1s	Standard	General purpose	1	1	2	320	4	Yes	$9.64
B1ms	Standard	General purpose	1	2	2	640	4	Yes	$19.27
A1_v2	Standard	General purpose	1	2	2	2x500	10	No	$32.85
B2s	Standard	General purpose	2	4	4	1280	8	Yes	$38.54
D1_v2	Standard	General purpose	1	3.5	4	4x500	50	No	$57.60
DS1_v2	Standard	General purpose	1	3.5	4	3200	7	Yes	$57.60
A2_v2	Standard	General purpose	2	4	4	4x500	20	No	$69.35
B2ms	Standard	General purpose	2	8	4	1920	16	Yes	$77.38
D2a_v4	Standard	General purpose	2	8	4	4x500	50	No	$87.60
D2as_v4	Standard	General purpose	2	8	4	3200	16	Yes	$87.60
D2_v3	Standard	General purpose	2	8	4	4x500	50	No	$91.25

Prices presented are estimates in your local currency that include only Azure infrastructure costs and any discounts for the subscription and location. The prices don't include any applicable software costs. Final charges will appear in your local currency in cost analysis and billing views. View Azure pricing calculator.

Figure 8: Pick your VM's size

Remember your public key? Now is the time it comes into play.



Home > Virtual machines > Create a virtual machine

Create a virtual machine

Changing Basic options may reset selections you have made. Review all options prior to creating the virtual machine.

1 vcpu, 1 GiB memory ($9.64/month)
Change size

Administrator account

Authentication type: [✓ SSH public key] Password

Username *: dat

SSH public key *: [redacted]

Learn more about creating and using SSH keys in Azure

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

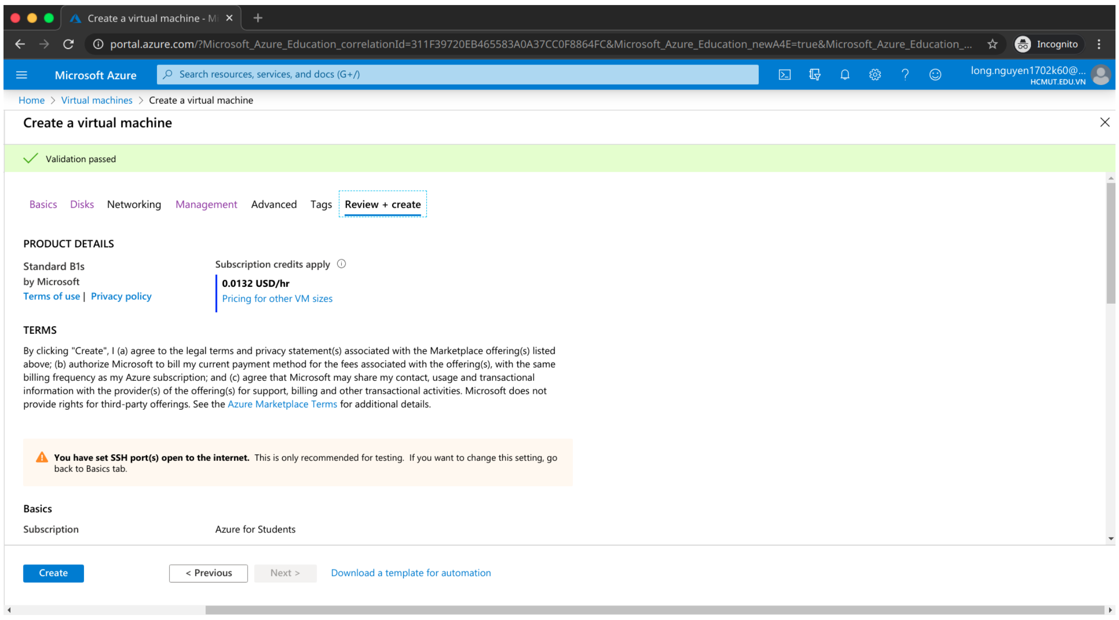
Public inbound ports *: None [✓ Allow selected ports]

Select inbound ports *: SSH (22)

This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Figure 9: Create username and paste your public key

- Proceed to **Networking**. At **Public IP** and **Assignment**, choose **Create new** and **Static**, respectively.



Create a virtual machine

Validation passed

Basics Disks Networking Management Advanced Tags Review + create

PRODUCT DETAILS

Standard B1s
by Microsoft
Terms of use | Privacy policy

Subscription credits apply
0.0132 USD/hr
Pricing for other VM sizes

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the Azure Marketplace Terms for additional details.

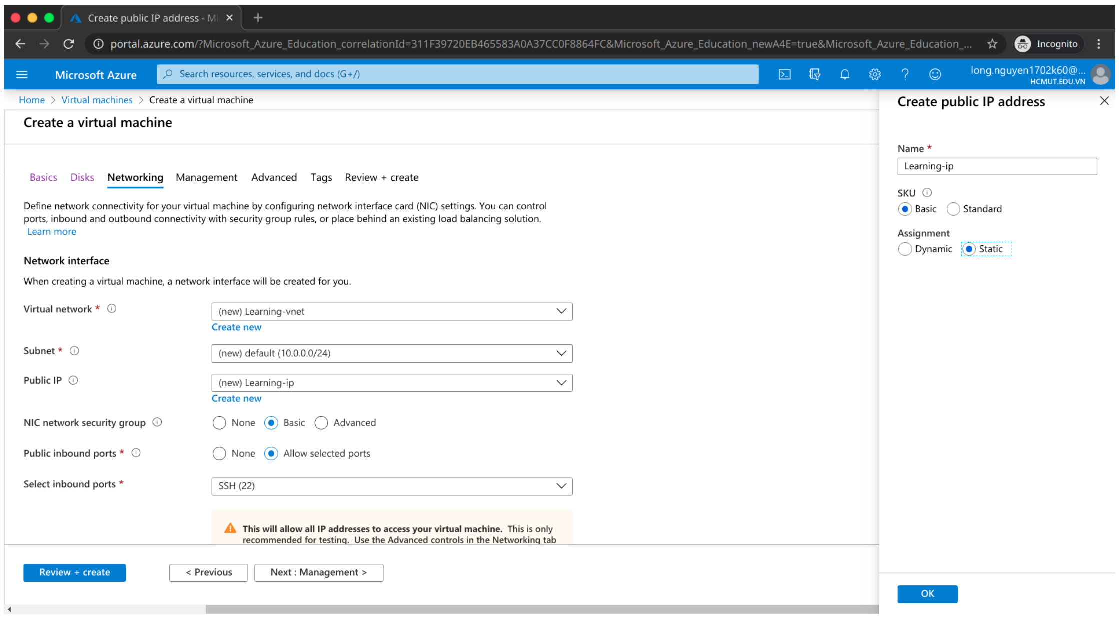
You have set SSH port(s) open to the internet. This is only recommended for testing. If you want to change this setting, go back to Basics tab.

Basics

Subscription Azure for Students

Create < Previous Next > Download a template for automation

Figure 11: Finish VM creation



Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution.
Learn more

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network * (new) Learning-vnet
Subnet * (new) default (10.0.0.0/24)
Public IP (new) Learning-ip
NIC network security group: None [✓ Basic] Advanced
Public inbound ports *: None [✓ Allow selected ports]
Select inbound ports * SSH (22)

This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab

Review + create < Previous Next : Management >

Create public IP address

Name * Learning-ip
SKU: [✓ Basic] Standard
Assignment: Dynamic [✓ Static]

OK

Figure 10: Pick **Static Assignment**

- Confirm your creation process by clicking **Review + create**.

You will be directed back to Virtual Machine main page. Please don't forget to note down your **Public IP Address**.

1.4 Virtual Machine setup

Let's utilize SSH to connect to your recently created machine.

```
ssh user@ip
```

Where *user* is your username and *ip* is the **Public Ip Address**. As you can see, don't miss out any details! Just a few small steps before we start enjoying our server. Please update and install Docker.

```
sudo apt update && sudo apt upgrade -y
sudo apt install docker.io -y
```

Follow instructions in this link https://docs.docker.com/engine/install/linux-postinstall/ to finish up your *Docker*.
Install MQTT Broker

```
docker run -d --name emqx -p 18083:18083 -p 8083:8083 -p 1883:1883
    emqx/emqx:lastest
```

For further information, feel free to check in this location: https://docs.emqx.io/broker/latest/en/

2 Pubisher-Subscriber simulation

Below are some typical tools which assist in publisher - subscriber and even broker simulation.

2.1 Mosquitto

Figure 12: Mosquitto

Please don't confuse yourself between mosquito and "mosquitto"! Mosquitto helps simulate a broker right on your personal computer. By default, it applies MQTT protocol. Furthermore, Mosquitto's in capable of playing Publisher or Subscriber role.

Installation link: https://mosquitto.org/

Turn on mosquitto by taking execution of mosquitto.exe.

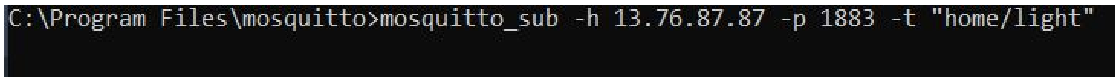
With a view to creating a subscriber, run the command below in your command line:

```
mosquitto_sub -h [host_name] -p [port] -t [topic]
```

From now on, you're able to publish your payload.

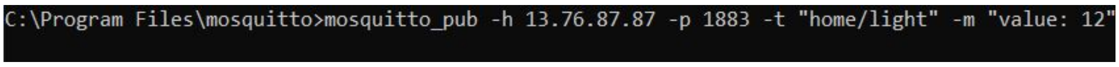
```
mosquitto_sub -h [host_name] -p [port] -t [topic] -m [message
```

For instance, below images illustrate a process connecting to the server possessing 13.76.87.8, 1883, "home/light" and "value: 12" as its hostname, port, topic and the message of the topic.



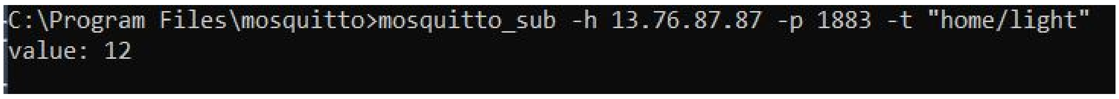
```
C:\Program Files\mosquitto>mosquitto_sub -h 13.76.87.87 -p 1883 -t "home/light"
```

Figure 13: A subscriber with "home/light" topic



```
C:\Program Files\mosquitto>mosquitto_pub -h 13.76.87.87 -p 1883 -t "home/light" -m "value: 12"
```

Figure 14: Publish message "value: 12" to topic "home/light"



```
C:\Program Files\mosquitto>mosquitto_sub -h 13.76.87.87 -p 1883 -t "home/light"
value: 12
```

Figure 15: Message arrived

You are strongly encouraged to research for a wider range of information by discovering mosquitto's documentation or just executing this in your command line:

```
mosquitto --help
```

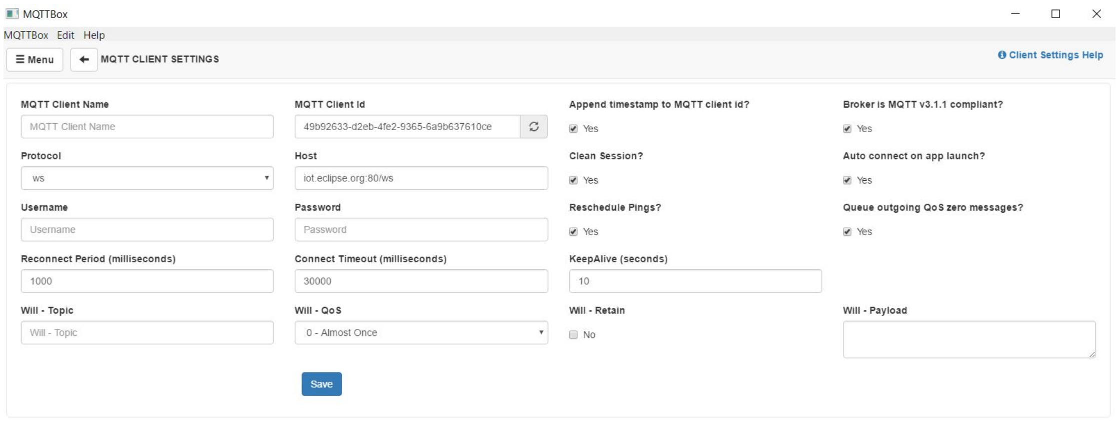
2.2 MQTTBox

Figure 16: MQTT logo

MQTTBox is an incredibly useful tool imitating a publisher or subscriber, with attractive and straightforward user interface.

You can attain access to MQTTBox at http://workswithweb.com/html/mqttbox/downloads.html

After the installing procedure, your application would have this appearance.



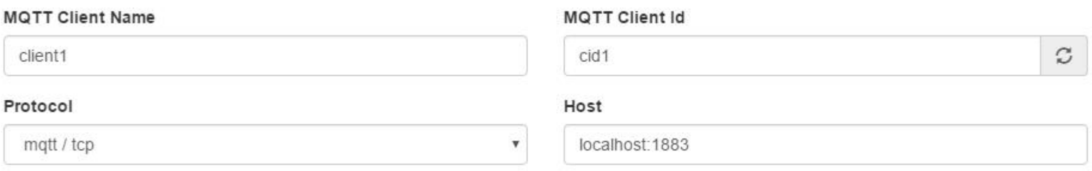
MQTT Client Name	MQTT Client Name
MQTT Client Id	49b92633-d2eb-4fe2-9365-6a9b637610ce
Append timestamp to MQTT client id?	✓ Yes
Broker is MQTT v3.1.1 compliant?	✓ Yes
Protocol	ws
Host	iot.eclipse.org:80/ws
Clean Session?	✓ Yes
Auto connect on app launch?	✓ Yes
Username	Username
Password	Password
Reschedule Pings?	✓ Yes
Queue outgoing QoS zero messages?	✓ Yes
Reconnect Period (milliseconds)	1000
Connect Timeout (milliseconds)	30000
KeepAlive (seconds)	10
Will - Topic	Will - Topic
Will - QoS	0 - Almost Once
Will - Retain	☐ No
Will - Payload	

Figure 17: MQTTBox User Interface

In order to create an MQTT client, all you have to do is to complete these fields

- MQTT Client Name
- MQTT Client Id
- Protocol

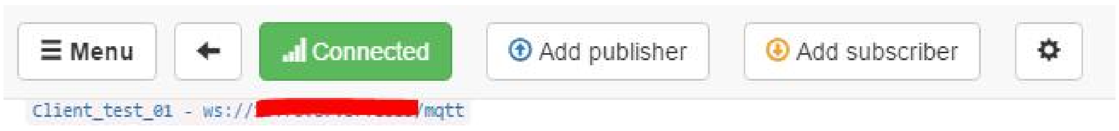
- Host



MQTT Client Name: client1
MQTT Client Id: cid1
Protocol: mqtt / tcp
Host: localhost:1883

Figure 18: Connect to the server with mosquitto

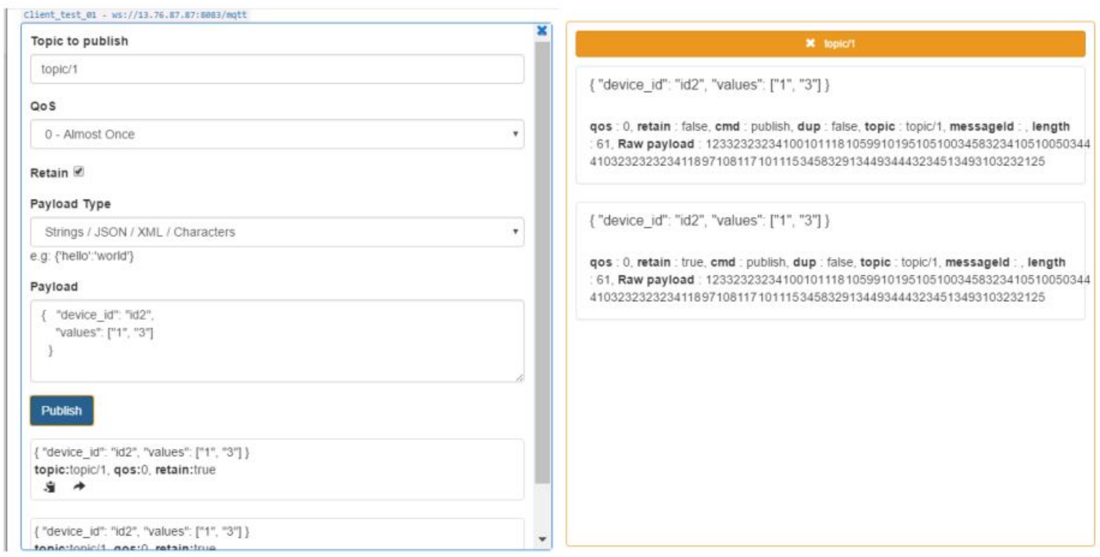
After double-clicking **Save**, you should realize the **Connected** state.



Client_test_01 - ws://[redacted]/mqtt

Figure 19: Server connected

You can also generate more than 1 publisher or subscriber on your machine. For an abundance of details, look up in the MQTTBox documentation.



Client_test_01 - ws://13.76.87.87:8083/mqtt

Topic to publish: topic/1
QoS: 0 - Almost Once
Retain ☑
Payload Type: Strings / JSON / XML / Characters
e.g: {'hello':'world'}
Payload:
```
{  "device_id": "id2",
   "values": ["1", "3"]
 }
```

{ "device_id": "id2", "values": ["1", "3"] }
topic:topic/1, **qos:**0, **retain:**true

{ "device_id": "id2", "values": ["1", "3"] }
topic:topic/1, **qos:**0, **retain:**true

{ "device_id": "id2", "values": ["1", "3"] }

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : topic/1, **messageId** : , **length** : 61, **Raw payload** : 12332323234100101118105991019510510034583234105100503444103232323234118971081171011153458329134493444323451349310323212 5

{ "device_id": "id2", "values": ["1", "3"] }

qos : 0, **retain** : true, **cmd** : publish, **dup** : false, **topic** : topic/1, **messageId** : , **length** : 61, **Raw payload** : 123323232341001011181059910195105100345832341051005034441032323232341189710811710111534583291344934443234513493103232125

Figure 20: MQTTBox Publisher and Subscriber

2.3 Paho - Python

One day, you might have an interest to work with MQTT on Python. As a result, Python API developers have provided a library dubbed as Paho to fulfill our wishes as Python developers.

To get embarked on Paho, you shall install it.

```
pip install paho-mqtt
```

A majority of our work will be executed on 2 modules named `paho.mqtt.client` and `paho.mqtt.subscribe`
In the latest section, we have made a connection to 13.76.87.87 broker and published a message along with "home/light01" topic. At the moment, instead of co-operating with MQTTBox, let's breathe some fresh air and overtake it on Paho.

Nonetheless, first, a Subscriber must be created to receive information, which you might want to do it on MQTTBox.

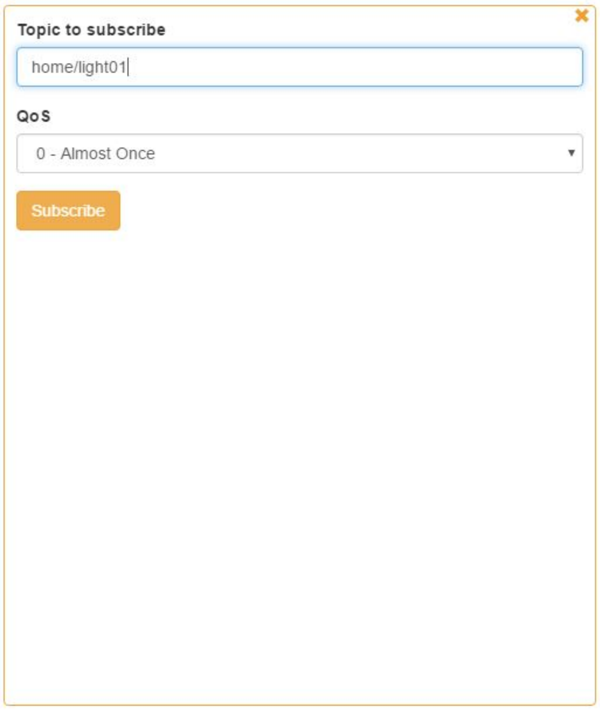
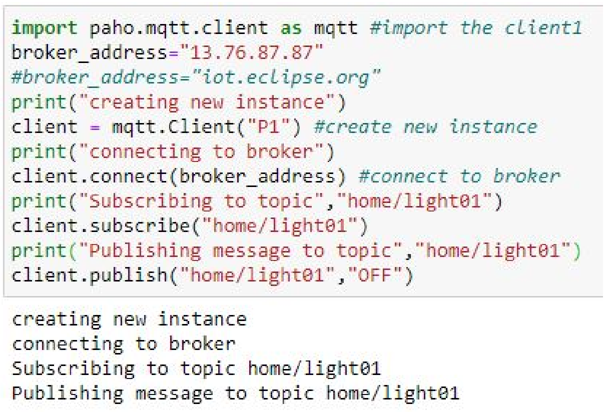

Topic to subscribe: home/light01
QoS: 0 - Almost Once

Figure 21: Make a Subscriber pertaining tp topic home/light01

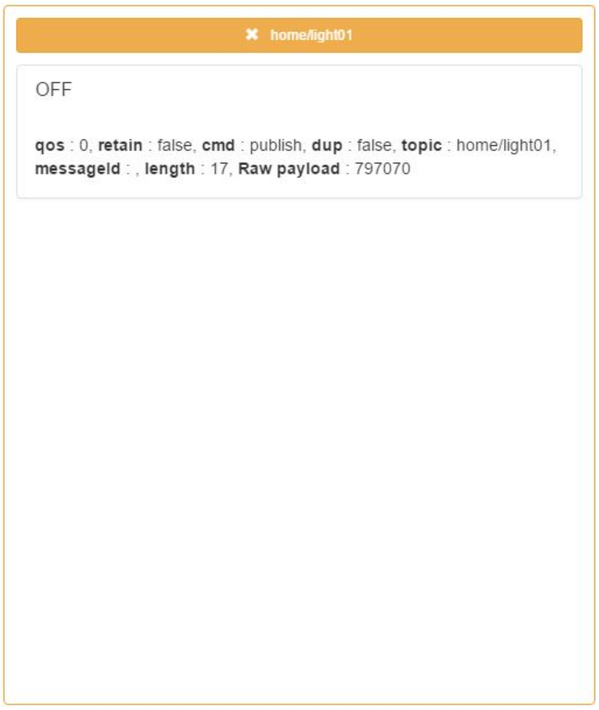
The below image includes code segments opening a connection and publishing a message to the Subscriber.



```
import paho.mqtt.client as mqtt #import the client1
broker_address="13.76.87.87"
#broker_address="iot.eclipse.org"
print("creating new instance")
client = mqtt.Client("P1") #create new instance
print("connecting to broker")
client.connect(broker_address) #connect to broker
print("Subscribing to topic","home/light01")
client.subscribe("home/light01")
print("Publishing message to topic","home/light01")
client.publish("home/light01","OFF")
```

```
creating new instance
connecting to broker
Subscribing to topic home/light01
Publishing message to topic home/light01
```

Figure 22: Paho aiding in connection and publishing a message to topic "home/light01"



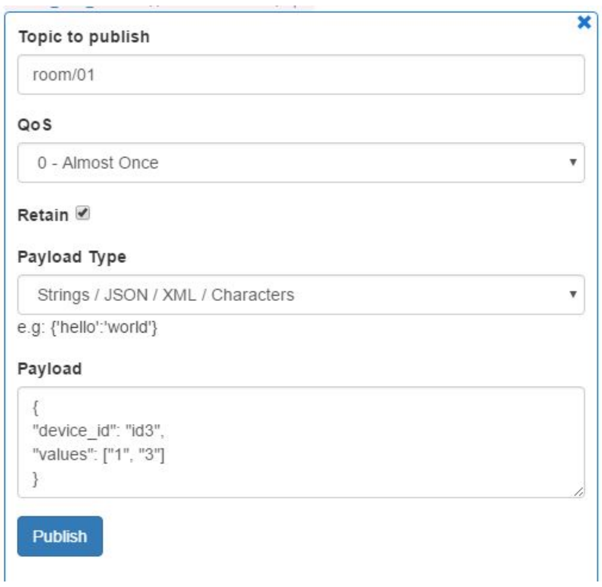
```
home/light01

OFF

qos : 0, retain : false, cmd : publish, dup : false, topic : home/light01,
messageId : , length : 17, Raw payload : 797070
```

Figure 23: Subcriber successfully received the message

The underneath code helps you construct a Subscriber to get a message from topic "room/01". This time, open your MQTTBox and experiment publishing a message (JSON format).



Topic to publish

room/01

QoS

0 - Almost Once

Retain ✓

Payload Type

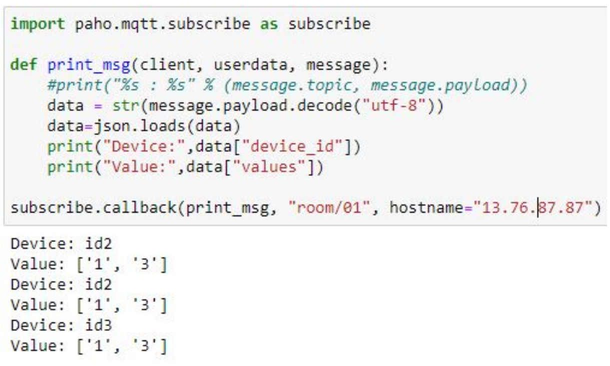
Strings / JSON / XML / Characters

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

Payload

```
{
"device_id": "id3",
"values": ["1", "3"]
}
```

Figure 24: Mqttbox publish message lên topic room/01



```
import paho.mqtt.subscribe as subscribe

def print_msg(client, userdata, message):
    #print("%s : %s" % (message.topic, message.payload))
    data = str(message.payload.decode("utf-8"))
    data=json.loads(data)
    print("Device:",data["device_id"])
    print("Value:",data["values"])

subscribe.callback(print_msg, "room/01", hostname="13.76.87.87")
```

```
Device: id2
Value: ['1', '3']
Device: id2
Value: ['1', '3']
Device: id3
Value: ['1', '3']
```

Figure 25: Connect to topic "room/01" and attain the message