

# Internet of Things: Technologies and Applications – Lab 4

Tran Phong Binh\*

*Department of Computer Science, National Tsing Hua University*

November 12, 2021

## 1 Part I

We begin by publishing telemetry events through the gateway, illustrated step by step as follows:

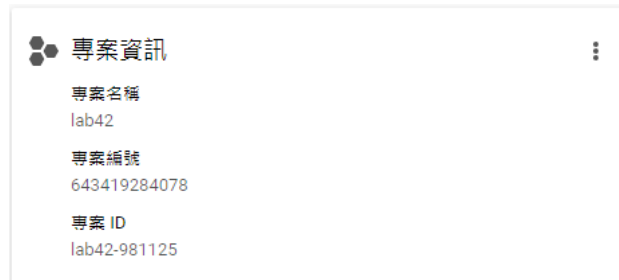


Figure 1: Create cloud project lab42 with ID lab42-981125.

---

\*Student ID: 110062421



Figure 2: Create cloud pub/sub topic `gateway-telemetry` with device state topic `gateway-state` under registry ID `lab42-registry`.



Figure 3: Create cloud gateway `lab42-gateway` under `lab42-registry` using the generated RSA public key.

```

bill-pi@raspberrypi:~$ ssh bill-pi@192.168.2.134
bill-pi@192.168.2.134's password:
Linux raspberrypi 5.10.63-v7+ #1459 SMP Wed Oct 6 16:41:10 BST 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Nov 12 19:43:36 2021 from 192.168.2.117
bill-pi@raspberrypi:~$ echo $GOOGLE_CLOUD_PROJECT
lab42-981125
bill-pi@raspberrypi:~$

```

Figure 4: Set local environment variable `GOOGLE_CLOUD_PROJECT` value to cloud project ID `lab42-981125`.

```

bill-pi@raspberrypi:~$ cd Desktop/1101-iot/lab-4/lab42/
bill-pi@raspberrypi:~/Desktop/1101-iot/lab-4/lab42$ cat run-gateway
#!/bin/bash

# Copyright 2018 Google LLC
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

python3 ./cloudiot_mqtt_gateway.py \
  --registry_id=lab42-registry \
  --gateway_id=lab42-gateway \
  --cloud_region=asia-east1 \
  --project_id=$GOOGLE_CLOUD_PROJECT \
  --private_key_file=rsa_private.pem \
  --algorithm=R5256 \
  --ca_cert=roots.pem \
  --mqtt_bridge_hostname=mqtt.googleapis.com \
  --mqtt_bridge_port=8883 \
  --let_expire_in_minutes=1200
bill-pi@raspberrypi:~/Desktop/1101-iot/lab-4/lab42$

```

Figure 5: Set local registry and gateway ID to `lab42-registry` and `lab42-gateway` respectively.

```

bill-pi@raspberrypi:~/Desktop$ cat run-gateway
#!/bin/bash

# Copyright 2018 Google LLC
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

python3 ./cloudiot_mqtt_gateway.py \
  --registry_id=lab42-registry \
  --gateway_id=lab42-gateway \
  --cloud_region=asia-east1 \
  --project_id=$GOOGLE_CLOUD_PROJECT \
  --private_key_file=rsa_private.pem \
  --algorithm=R5256 \
  --ca_cert=roots.pem \
  --mqtt_bridge_hostname=mqtt.googleapis.com \
  --mqtt_bridge_port=8883 \
  --let_expire_in_minutes=1200
bill-pi@raspberrypi:~/Desktop/1101-iot/lab-4/lab42$ wget https://pki.goog/roots.pem
2021-11-12 20:39:51-- https://pki.goog/roots.pem
Resolving pki.goog (pki.goog)... 216.239.32.29, 2001:4860:4860::32:1d
Connecting to pki.goog (pki.goog)|216.239.32.29|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 72116 (70k) [Application/x-pem-file]
Saving to: 'roots.pem'

roots.pem                               100%[=====] 69.54K  178K/s  in 0.4s

2021-11-12 20:39:52 (175 KB/s) = 'roots.pem' saved [72116/72116]
bill-pi@raspberrypi:~/Desktop/1101-iot/lab-4/lab42$

```

Figure 6: Get `roots.pem` from Google API.

```
bill-pi@raspberrypi:~/Desktop $ python3 -m venv env
bill-pi@raspberrypi:~/Desktop/1181-lot/lab-4/lab2 $ source env/bin/activate
(env) bill-pi@raspberrypi:~/Desktop/1181-lot/lab-4/lab2 $ pip3 install -r requirements-gateway.txt
Looking in indexes: https://pypi.org/simple, https://www.piheels.org/simple
Collecting cryptography==2.4.1 (from -r requirements-gateway.txt (line 1))
Using cached https://www.piheels.org/simple/cryptography/cryptography-2.4.1-cp37-cp37m-linux_armv7l.whl
Collecting paho-mqtt==1.4.0 (from -r requirements-gateway.txt (line 2))
Using cached https://www.piheels.org/simple/paho-mqtt/paho-mqtt-1.4.0-py3-none-any.whl
Collecting pyjwt==1.6.4 (from -r requirements-gateway.txt (line 3))
Using cached https://files.pythonhosted.org/packages/93/d1/3378cc8184a652dc9299399ee8b4c83847c567e298385dcf86987e00
3/PyJWT-1.6.4-py2.py3-none-any.whl
Collecting idna==2.1 (from cryptography==2.4.1->-r requirements-gateway.txt (line 1))
Using cached https://files.pythonhosted.org/packages/04/a2/d918dcd2235d8958fe113e1a3638137ef8c8b44859ade3863982eacd2a
4/idna-2.3-py3-none-any.whl
Collecting six==1.4.1 (from cryptography==2.4.1->-r requirements-gateway.txt (line 1))
Using cached https://files.pythonhosted.org/packages/d9/5a/e7c1adbe875f2abb91bd84cf2dc52d792b5a01586781dbcf25c91da1f
1/six-1.16.0-py2.py3-none-any.whl
Collecting asn1crypto==0.21.0 (from cryptography==2.4.1->-r requirements-gateway.txt (line 1))
Using cached https://files.pythonhosted.org/packages/b5/a8/56be92dc04a5bf1998785a904028249fe7c9a835b955fe93b6a3e5b829f
8/asn1crypto-1.4.0-py2.py3-none-any.whl
Collecting cffi==1.11.3, >=1.7 (from cryptography==2.4.1->-r requirements-gateway.txt (line 1))
Using cached https://www.piheels.org/simple/cffi/cffi-1.15.0-cp37-cp37m-linux_armv7l.whl
Collecting pycparser (from cffi==1.11.3, >=1.7->cryptography==2.4.1->-r requirements-gateway.txt (line 1))
Using cached https://files.pythonhosted.org/packages/62/d5/5f61e8be421e85889f2e5e33b7f9a6799bd982198517d912eb1c76e1a5
3/pycparser-2.21-py2.py3-none-any.whl
Installing collected packages: idna, six, asn1crypto, pycparser, cffi, cryptography, paho-mqtt, pyjwt
Successfully installed asn1crypto-1.4.0 cffi-1.15.0 cryptography-2.4.1 idna-2.3 paho-mqtt-1.4.0 pycparser-2.21 pyjwt-1.6
4 six-1.16.0
(env) bill-pi@raspberrypi:~/Desktop/1181-lot/lab-4/lab2 $
```

Figure 7: Install gateway's requirements.

```
bill-pi@raspberrypi:~/Desktop $ python3 -m venv env
bill-pi@raspberrypi:~/Desktop/1181-lot/lab-4/lab2 $ source env/bin/activate
(env) bill-pi@raspberrypi:~/Desktop/1181-lot/lab-4/lab2 $ pip3 install -r requirements-gateway.txt
Looking in indexes: https://pypi.org/simple, https://www.piheels.org/simple
Collecting cryptography==2.4.1 (from -r requirements-gateway.txt (line 1))
Using cached https://www.piheels.org/simple/cryptography/cryptography-2.4.1-cp37-cp37m-linux_armv7l.whl
Collecting paho-mqtt==1.4.0 (from -r requirements-gateway.txt (line 2))
Using cached https://www.piheels.org/simple/paho-mqtt/paho-mqtt-1.4.0-py3-none-any.whl
Collecting pyjwt==1.6.4 (from -r requirements-gateway.txt (line 3))
Using cached https://files.pythonhosted.org/packages/93/d1/3378cc8184a652dc9299399ee8b4c83847c567e298385dcf86987e00
3/PyJWT-1.6.4-py2.py3-none-any.whl
Collecting idna==2.1 (from cryptography==2.4.1->-r requirements-gateway.txt (line 1))
Using cached https://files.pythonhosted.org/packages/04/a2/d918dcd2235d8958fe113e1a3638137ef8c8b44859ade3863982eacd2a
4/idna-2.3-py3-none-any.whl
Collecting six==1.4.1 (from cryptography==2.4.1->-r requirements-gateway.txt (line 1))
Using cached https://files.pythonhosted.org/packages/d9/5a/e7c1adbe875f2abb91bd84cf2dc52d792b5a01586781dbcf25c91da1f
1/six-1.16.0-py2.py3-none-any.whl
Collecting asn1crypto==0.21.0 (from cryptography==2.4.1->-r requirements-gateway.txt (line 1))
Using cached https://files.pythonhosted.org/packages/b5/a8/56be92dc04a5bf1998785a904028249fe7c9a835b955fe93b6a3e5b829f
8/asn1crypto-1.4.0-py2.py3-none-any.whl
Collecting cffi==1.11.3, >=1.7 (from cryptography==2.4.1->-r requirements-gateway.txt (line 1))
Using cached https://www.piheels.org/simple/cffi/cffi-1.15.0-cp37-cp37m-linux_armv7l.whl
Collecting pycparser (from cffi==1.11.3, >=1.7->cryptography==2.4.1->-r requirements-gateway.txt (line 1))
Using cached https://files.pythonhosted.org/packages/62/d5/5f61e8be421e85889f2e5e33b7f9a6799bd982198517d912eb1c76e1a5
3/pycparser-2.21-py2.py3-none-any.whl
Installing collected packages: idna, six, asn1crypto, pycparser, cffi, cryptography, paho-mqtt, pyjwt
Successfully installed asn1crypto-1.4.0 cffi-1.15.0 cryptography-2.4.1 idna-2.3 paho-mqtt-1.4.0 pycparser-2.21 pyjwt-1.6
4 six-1.16.0
(env) bill-pi@raspberrypi:~/Desktop/1181-lot/lab-4/lab2 $ source run-gateway
Creating JWT using RS256 from private key file rsa_private.pem
on_publish: userdata: None, mid 1
Unable to find key 1
connect status False
on_connect: Connection Accepted.
on_subscribe: mid 2, qos (1,)
on_unsubscribe: mid 3, qos (0,)
Received message '' on topic '/devices/lab2-gateway/config' with Qos 1
Nobody subscribes to topic /devices/lab2-gateway/config
```

Figure 8: Run the given gateway program.

[← 裝置詳細資訊](#) [編輯裝置](#) [更新設定](#) [傳送指令](#) [訂閱通訊](#) [刪除](#)

**裝置 ID: thermostat**

數字 ID	登錄權	Cloud Logging	通訊
3188364267338418	lab42-registry	登錄權預設值 <a href="#">查看紀錄檔</a>	已核准

[詳細資料](#) [設定和狀態](#) [驗證](#)

**最新活動**

活動訊號 (僅限 MQTT)	—
上次收到遙測事件的時間	2021年11月12日 下午8:03:30
上次收到裝置狀態事件的時間	—
上次傳送設定的時間	—
可用基設定 ACK (僅限 MQTT)	—
雜誌	2021年11月12日 下午8:03:38
雜誌狀態和訊息	[9] mqtt: The connection broke or was closed by the client.

**裝置中繼資料**

您可以在裝置設定中新增或編輯中繼資料。 [編輯裝置](#)。

Figure 9: Create cloud device with ID thermostat under lab42-registry using the generated RSA public key and add it to gateway lab42-gateway.

```

bill-pi@raspberrypi ~/Desktop
bill-pi@raspberrypi:~$ cd Desktop/1101-lot/lab-4/lab02/
bill-pi@raspberrypi:~/Desktop/1101-lot/lab-4/lab02$ cat thermostat.py
# Copyright 2018 Google LLC
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the license at
# https://www.apache.org/licenses/LICENSE-2.0
# Unless required by applicable law or agreed to in writing, software
# distributed under the license is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
from __future__ import print_function
import random
import sys
import socket
import time
import Adafruit_DHT
from colors import bcolors
DHT_SENSOR_PIN = 4
ADDR = '192.168.2.134'
PORT = 18080

```

Figure 10: Set IP address in `thermostat.py` to that of the Raspberry Pi.

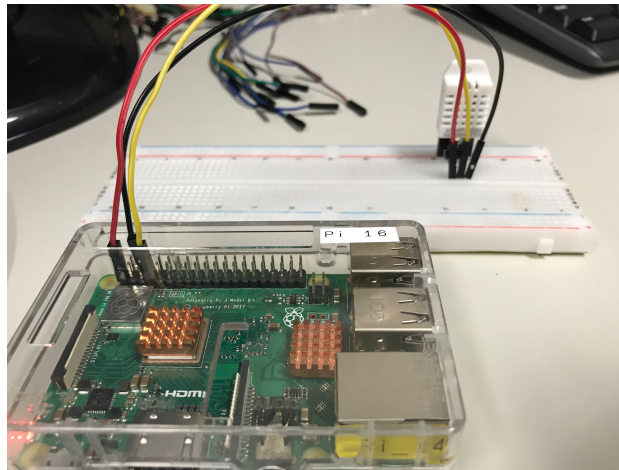


Figure 11: Connect the DHT22 sensor to Raspberry Pi's GPIO pin 4.

```

bill-pi@raspberrypi ~/Desktop
bill-pi@raspberrypi:~$ cat thermostat.py
t = "{:.3f}".format(t)
sys.stdout.write(
    '\r >' + bcolors.GREEN + bcolors.BOLD +
    'Temp: {}, Hum: {}'.format(t, h) + bcolors.ENDC + ' <<<')
sys.stdout.flush()

message = MakeMessage(
    device_id, 'event', 'temperature={}, humidity={}'.format(t, h))
SendCommand(client_sock, message, False)
time.sleep(2)

finally:
    print('closing socket', file=sys.stderr)
    client_sock.close()
bill-pi@raspberrypi:~/Desktop/1101-lot/lab-4/lab02$ source run-thermostat
Bringing up device thermostat
Bring up device 1
Send data: { "device": "thermostat", "action": "detach" }
sending: { "device": "thermostat", "action": "detach" }
waiting for response
received: { "device": "0", "command": "d", "status": "ok" }
Response { "device": "0", "command": "d", "status": "ok" }
Send data: { "device": "thermostat", "action": "attach" }
sending: { "device": "thermostat", "action": "attach" }
waiting for response
received: { "device": "thermostat", "command": "attach", "status": "ok" }
Response { "device": "thermostat", "command": "attach", "status": "ok" }
>>>

```

Figure 12: Run the given thermostat program.

## 2 Part II

In the second part of the experiment, we create a subscription to our telemetry topic to view data:



Figure 13: Execute pull device data command on the Cloud Shell.