**Question 2.3 (3%)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Amazon Web Services IoT (IoT Core) | IBM Watson IoT Platform | Microsoft Azure IoT Hub | Google Cloud IoT | Thingspeak IoT Platform |
| Fault management | IoT rules engine:  an application can enable an IoT error action. | Monitoring and reporting: Message Gateway | Azure Monitor, Azure Event Grid | Cloud Security Alliance (CSA):  provides the Cloud Controls Matrix (CCM) | State Machines |
| Discovery and registrations of devices | Info: registry as JSON data.  Things: identified by a name. | Recognized by the Capture program | Command:  az group delete --name MyResourceGroup | Google Cloud Architecture Framework,  IAM addresses | globally unique MAC address |
| Troubleshooting | Device advisor | Management Console Support Assistant, dmsupportinfo command | Azure Monitor, Azure Event Grid | Test locally |  |
| Network performance monitoring | CloudWatch:  collects and processes raw data from AWS IoT into readable, near real-time metrics. | Live and Snapshot | IoT hub:  managed service hosted in the cloud that acts as a central message hub for communication between an IoT application and its attached devices. | SLO monitoring | Wayrest |
| Security management | All traﬃc to and  from AWS IoT is sent securely over Transport Layer Security (TLS). | TLS encryption | High:  IoT\_SharedCredentials,  IoT\_PrivilegedDockerOptions | mTLS-encrypted connection | Encryption, API key |
| Data storages | AWS S3, DynamoDB | Storage Manager | SMB serve | Google Cloud | central location in the cloud |
| Data processing | Streaming:  AWS Lambda, Kinesis  Edge processing:  Greengrass | Capture program | REST API-based cloud service | Cloud SQL for MySQL,  BigQuery | REST API, MQTT API |

1. To present your review results, you can summarise how the following management functions are achieved in each of the above platforms in a table (just providing some notes/comments).

If you cannot find related information, leave the space blank.

1. To compare their features, you can also generate a table to include the following aspects:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Amazon Web Services IoT (IoT Core) | IBM Watson IoT Platform | Microsoft Azure IoT Hub | Google Cloud IoT | Thingspeak IoT Platform |
| Operating systems used | Linux, other | Linux, Windows | Linux, Windows, RTOS | Linux, other | Linux |
| Applications protocols adopted | MQTT, HTTP, WebSockets | MQTT, HTTP, WebSockets | MQTT, AMQP, HTTPS | MQTT, HTTP. | MQTT, HTTP. |
| Wireless networks supported | Wi-Fi, LTE, Zigbee, cellular, ble | Wi-Fi, LTE, Wi-Fi, Zigbee, cellular, ble | Wi-Fi, LTE, LoRaWAN, Zigbee | Wi-Fi, LoRaWAN, ble | Wi-Fi |
| Typical connectable things (types of end devices, gateways, data sources etc) | Edge devices  Sensors, gateway, drones, microcontrollers | Sensors, gateway, machines, cameras | Sensors, machines, home devices, monitor | home devices, sensors, edge devices | Sensors, microcontrollers |
| Security types | mutual TLS | TLS, role-based access, API key | TLS encryption, SAS tokens | TLS | TLS encryption, API key access control |
| Services (eg PaaS, IaaS, and SaaS) | PaaS | PaaS | PaaS | PaaS | PaaS |
| Costs of use of the platform | Free tier with limited storage within 12-mounth | Pay-as-you-go account:  Pay what you exactly use | Basic tier: $10/50/100  Standard: free/ $25/250/2500 | Free: 250mb/  $0.0045 per MB | Free tier/  USD $79.00/m |
| 2 examples of established applications | Smart home,  Humidity detector | Motion detect light,  Remote control electronics | Vehicle monitoring,  Monitor camera | Soil moisture level detector, geospatial analysis | Smart Refrigerator,  Agricultural IoT |

3) You can provide a brief summary about your overall findings and remarks about how you may choose an IoT platform for a specific IoT market or application.

**References (you can find/add your own sources)**

[https://www.softwaretestinghelp.com/best-iot-platforms/Links to an external site.](https://www.softwaretestinghelp.com/best-iot-platforms/)

[https://www.g2.com/categories/iot-platformsLinks to an external site.](https://www.g2.com/categories/iot-platforms)

[https://www.postscapes.com/internet-of-things-platforms/Links to an external site.](https://www.postscapes.com/internet-of-things-platforms/)

<https://docs.aws.amazon.com/wellarchitected/latest/iot-lens/failure-management.html>

<https://docs.aws.amazon.com/pdfs/iot/latest/developerguide/iot-dg.pdf#iot-core-troubleshooting>

<https://www.ibm.com/docs/pl/wip-mg/5.0.0.1?topic=gateway-monitoring-reporting>

<https://learn.microsoft.com/pdf?url=https%3A%2F%2Flearn.microsoft.com%2Fen-us%2Fazure%2Fiot-hub%2Ftoc.json>

<https://au.mathworks.com/help/thingspeak/>

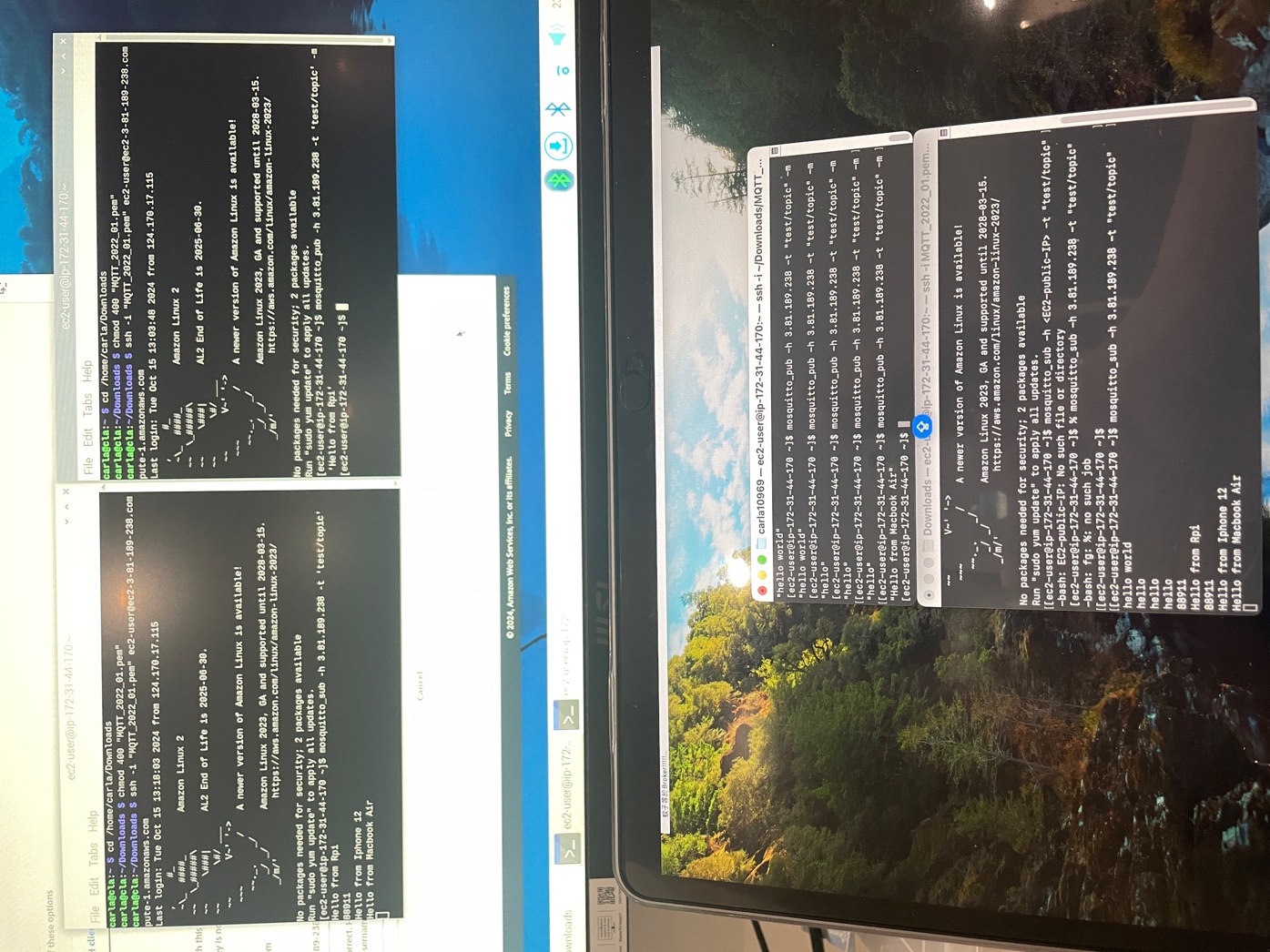
<https://blogs.mathworks.com/iot/2011/03/25/device-registration-and-mac-addresses/>

<https://www.softwareadvice.com.au/software/241427/thingspeak#:~:text=ThingSpeak%20has%20a%20free%20version,starts%20at%20USD%2079.00%2Fmonth>.

<https://www.ibm.com/topics/internet-of-things>

**Question 2.4 (3%)**

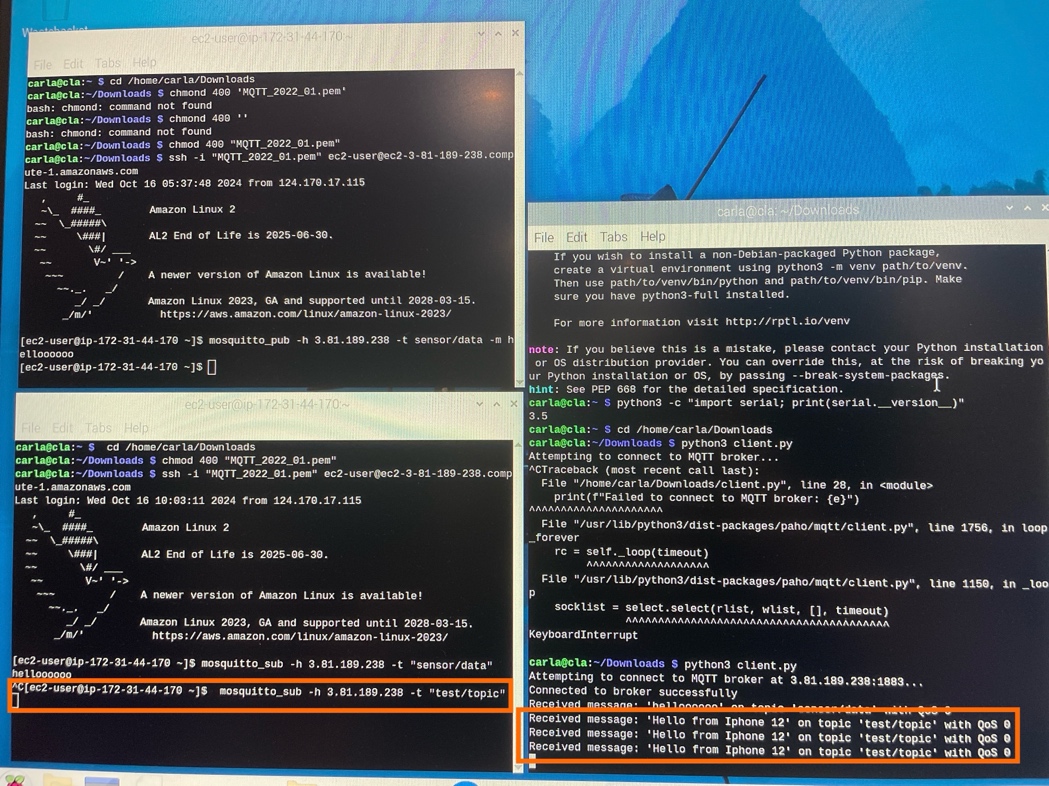
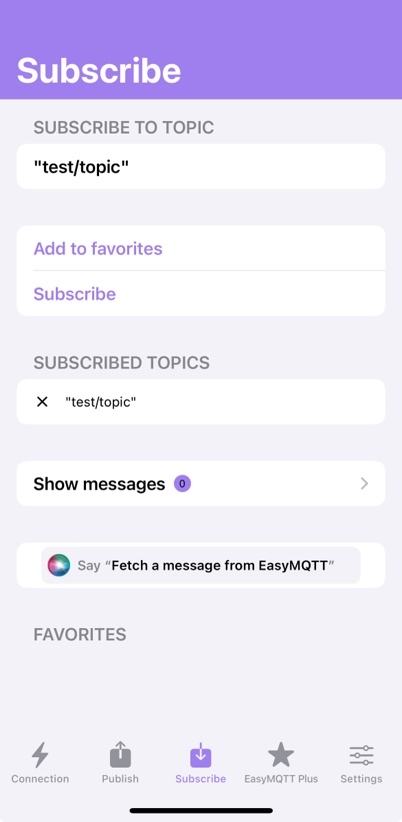
Using MQTT to connect between a laptop computer (client) and a self-hosted or cloud MQTT broker. You can follow “Beginners Guide To The MQTT Protocol” in the link: [http://www.steves-internet-guide.com/mqtt/Links to an external site.](http://www.steves-internet-guide.com/mqtt/) or other online tutorials to complete the connection. There are free self-hosted brokers , the most popular being [Mosquitto Links to an external site.](https://mosquitto.org/" \t "_blank)and commercial ones like [HiveMQ.Links to an external site.](http://www.hivemq.com/" \t "_blank) Please present the following contents as your assignment report.



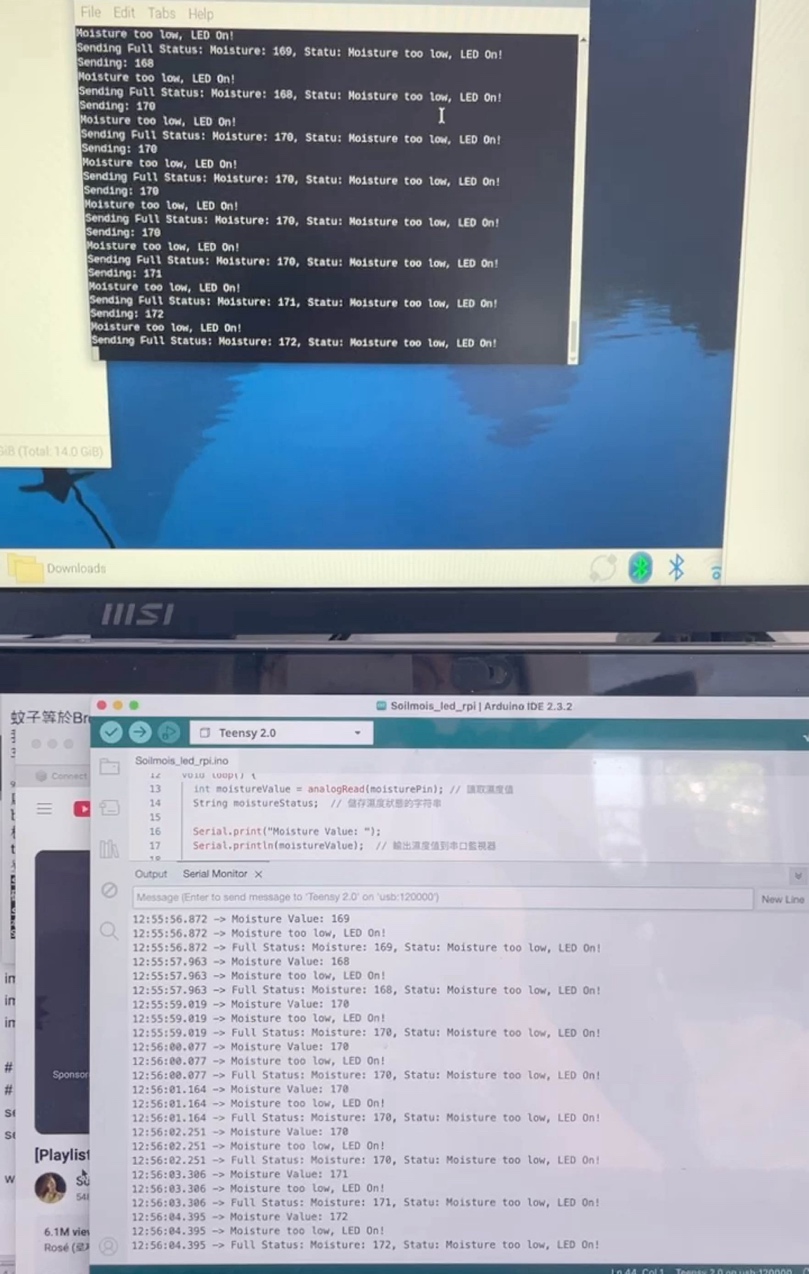
* Your chosen broker

MQTT broker

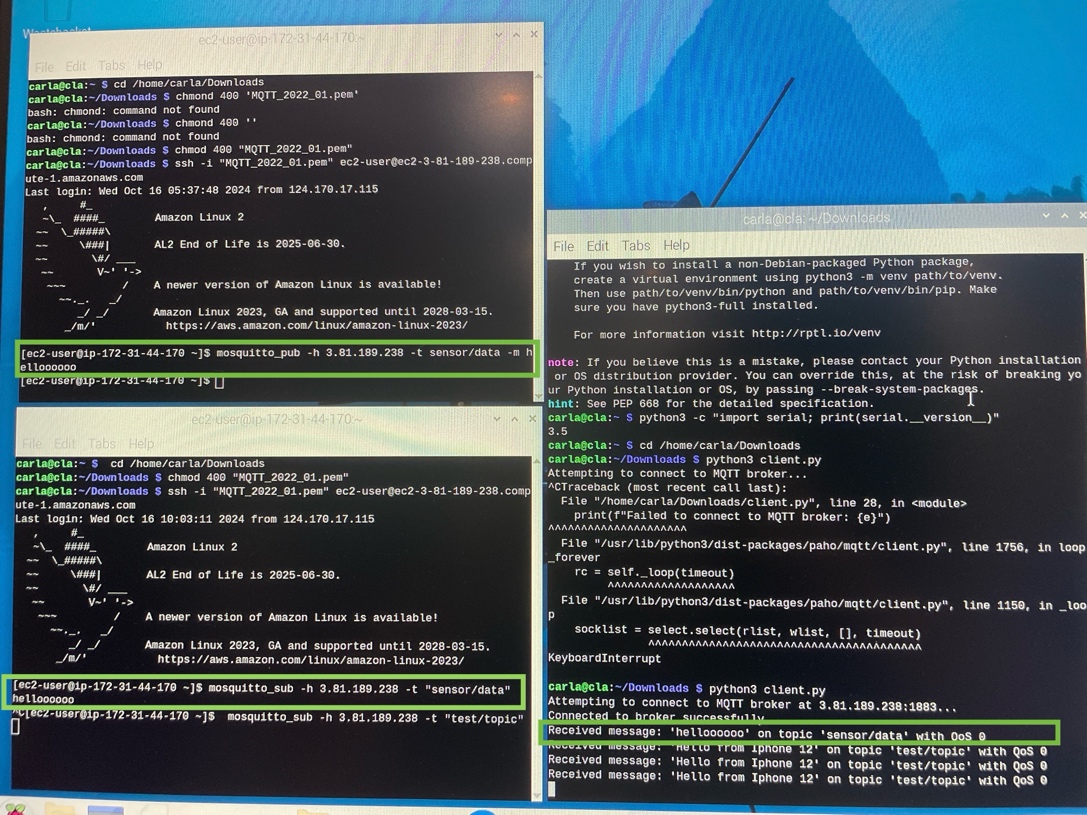
* How you set up MQTT client and MQTT broker -show the screenshots and explain
* How you set up the Topics for performing publish/subscribe-show your set up with 2 examples.
* How you demonstrate your success connections by publishing and subscribing the messages of two different sizes, such as 10Kbytes and 100Kbytes-show your screenshots or evidence that you can receive (by subscription) the message you send (publish).
  1. ***‘Test/topics’***
* *Iphone 12 published to Rpi*



* *Soil moisture sensor + BLE module published to Rpi*



* 1. *‘sensor/data’*
* Published ($ mosquito\_pub -h) & subscribe ($ mosquito\_sub -h)on Rpi



* How you compute the round-trip time (RTT) as a latency measure using your computer timestamps

***RTT= End Time−Start Time***

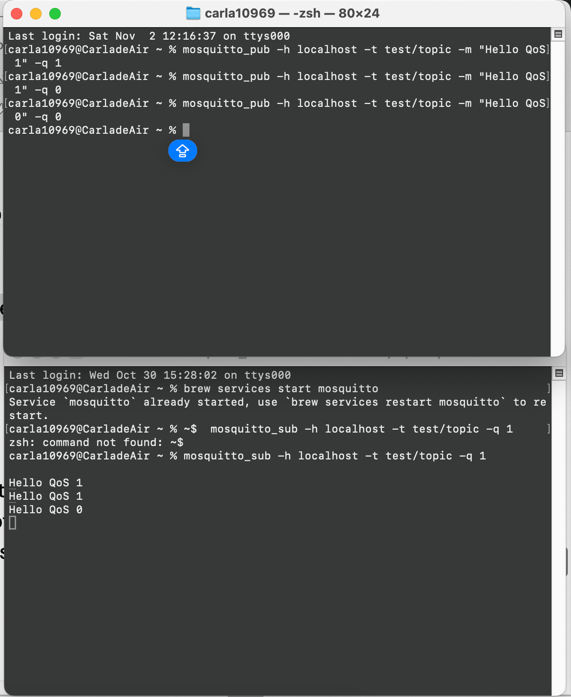
Both sending form iPhone and Rpi terminal are receive almost immediately.

* How you set up QoS levels-show your setting of QoS level 0, QoS level 1.

Use the following terminal command:

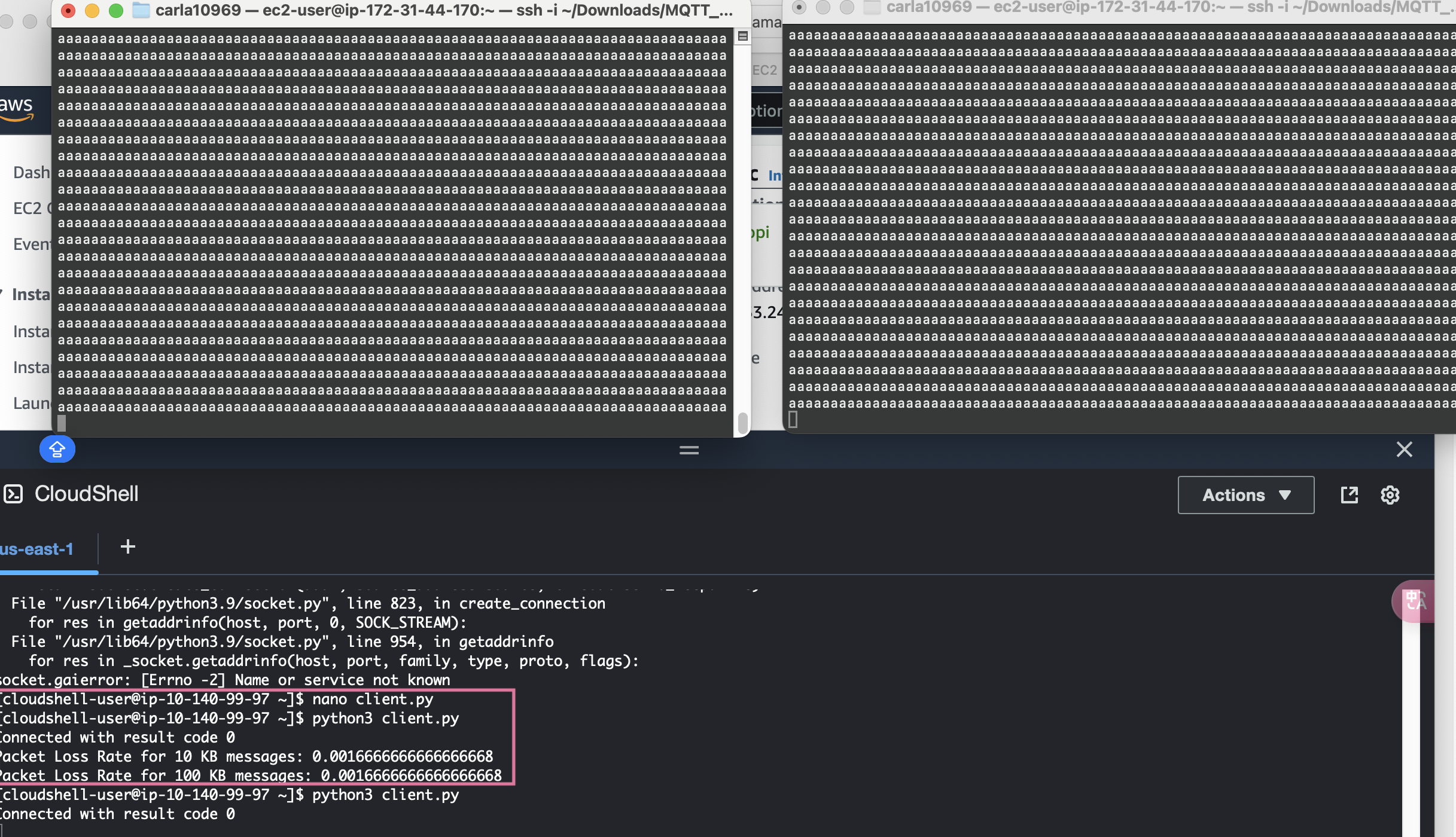
~$ mosquitto\_sub -h localhost -t test/topic -q 1

~$ mosquitto\_pub -h localhost -t test/topic -m "Hello QoS 1" -q 1



* Given the setting QoS level 1, compute the packet-loss rate over the period of 600 seconds by sending and receiving the messages at the sizes of 10Kbytes and 100Kbytes every second. what are they?   Are the RTTs and packet-loss rates related to the message sizes?

After running the python file on EC2 my broker, packet-loss rate for both 10 KB and 100 KB messages is approximately 0.00167/ 0.167%. This result means that the RTTs and packet-loss rates are not related to the message sizes as well as both my MQTT client and MQTT server on EC2 network are reliable to handle wither 10kb or 100kb messages.



**Question 2.5 (2%)**

From my point of view, general IoT technology would be one of the most popular information technology concepts in modern society. Most people cannot live without a smart phone or relevant computer devices for either personal time or working hours since the widespread use of 3G at the time I was very young started to contact the outside world around 2010. Everything around computer devices is a key target that IT aims to develop as creatively and fast as possible, providing this society with a more civilised life adventurously, such as popular topics such as smart homes, integration of portable devices, and industrial monitoring systems. This kind of possibility of integration diversity has brought me into the world of mobile networks and computer networks, allowing me to gain an important overview of whole IoT architecture with base theory while developing my own Arduino project in unit IFN649, which makes me complete my own beginner hands-on IoT program. Although it was a very challenging task for me, I still did the basic requirements and decided to study more about embedded systems in my long vacation for future jobs.