KENNESAW STATE UNIVERSITY

CPE 4040: Data Collection and Analysis, Spring 2024

Laboratory Report #2

MQTT PUB/SUB on Raspberry Pi

Team Members: Anindita Deb and Damisi Kayode

Electrical and Computer Engineering

Kennesaw State University

Faculty: Dr. Jeffrey L Yiin

Date of Lab Session: January 29, 2024

I. Objective

1. Learn to create MQTT Broker on Raspberry Pi
2. Understand basic PUB/SUB messaging using Mosquitto MQTT package
3. Understand how to use MQTT messaging options (Topics) with Mosquitto

II. Material List

Software:

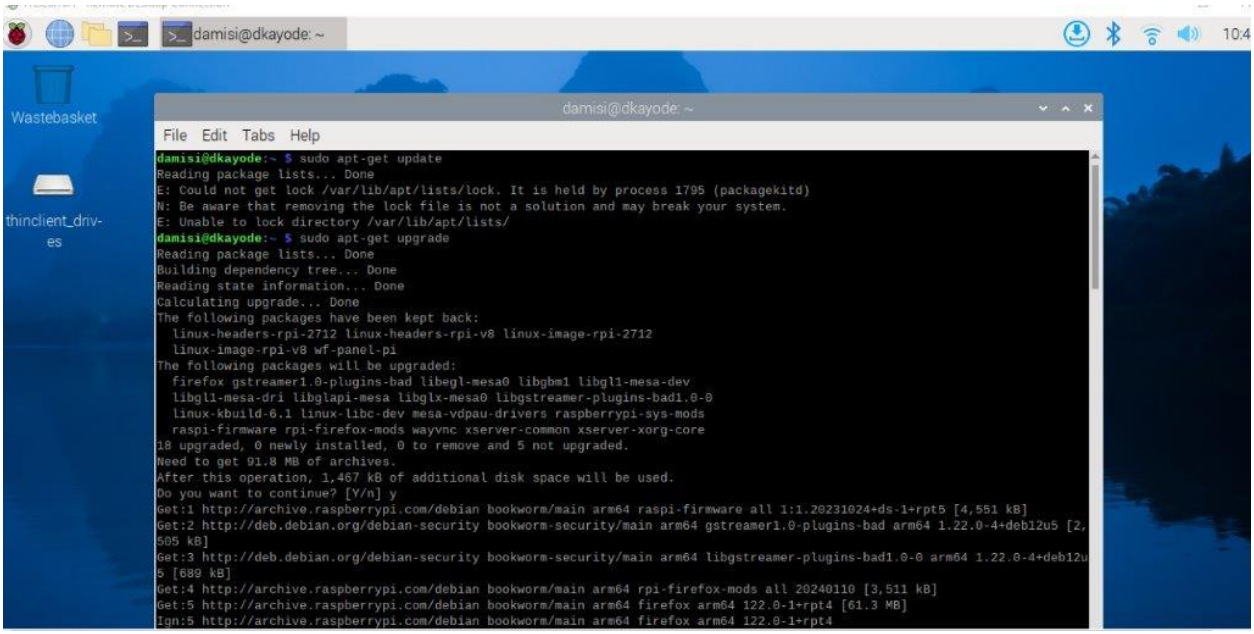
- MQTT
- PuTTY
- VNC Viewer

Hardware:

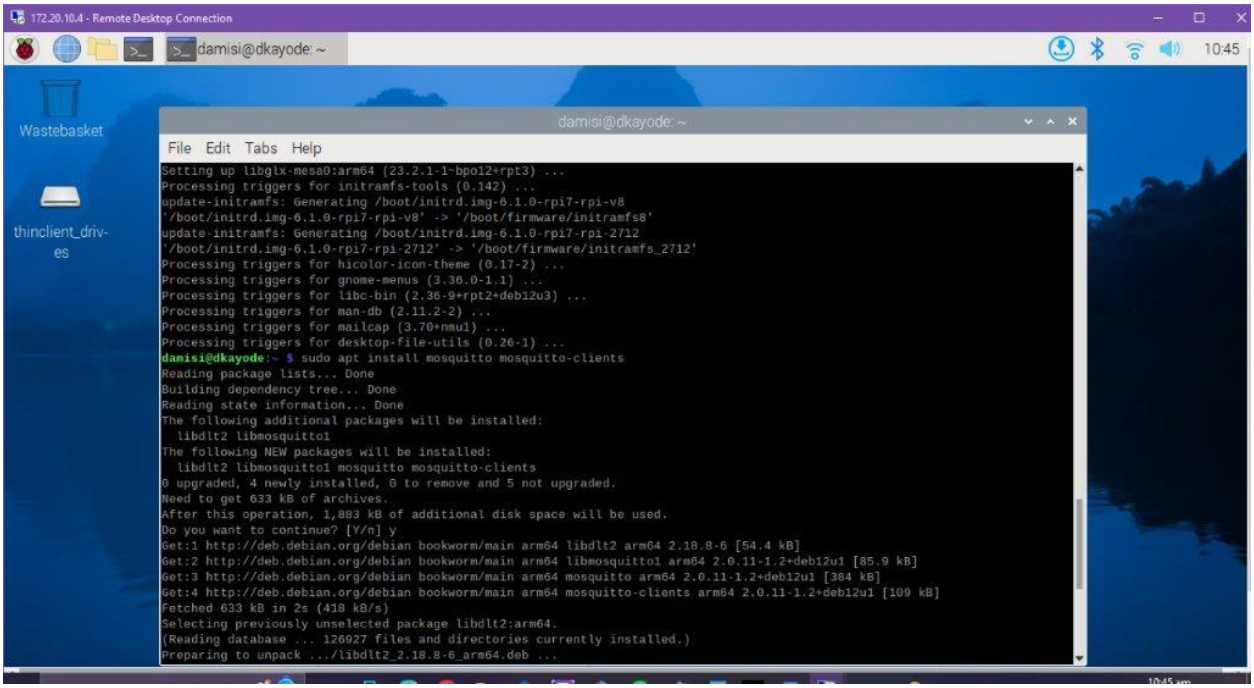
- Raspberry Pi 4
- Power cord
- PC

III. Lab Procedures and Results

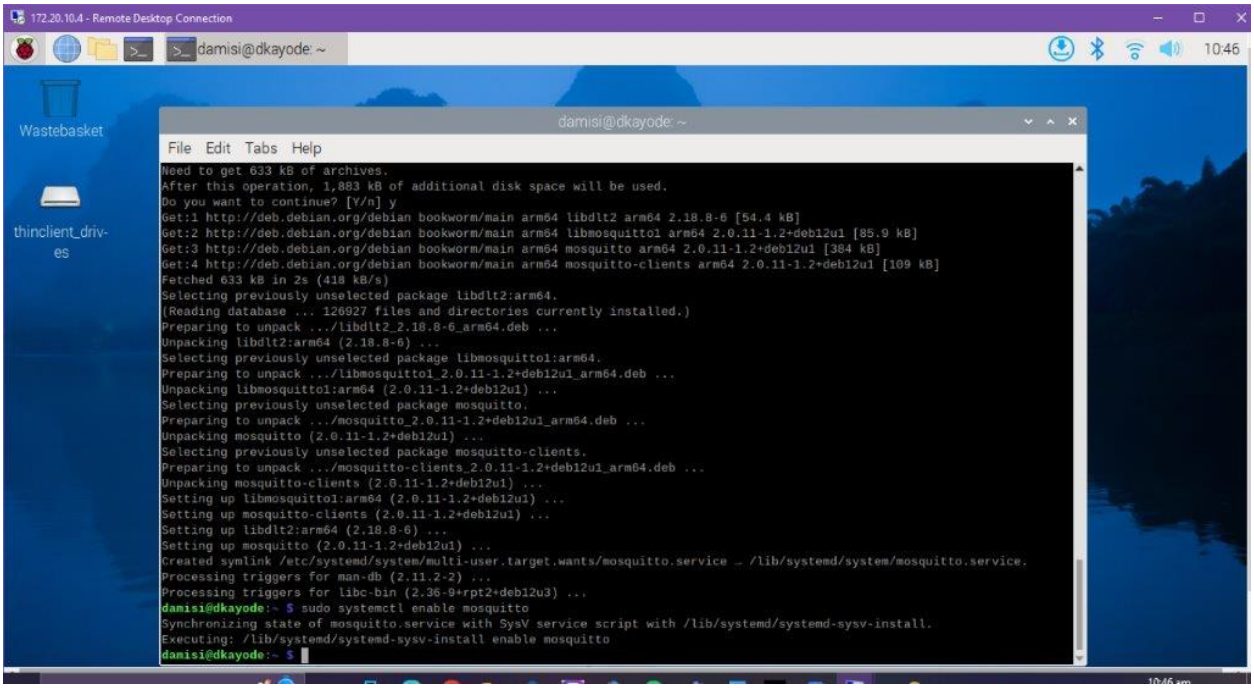
1. Power up Pi and open Remote Desktop Connection on your laptop and connect to the raspberry Pi.
2. After login, we open the terminal window and update and upgrade the Linux packages.



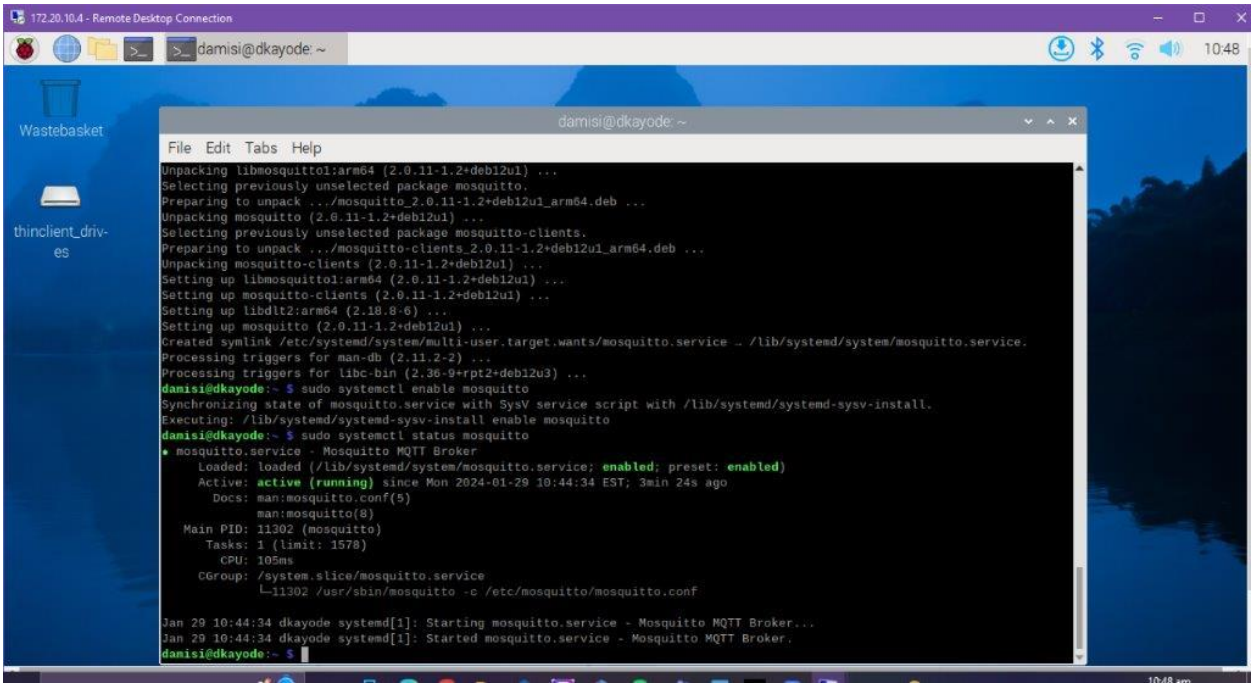
3. We need to create a MQTT Client/Broker (Publisher-subscriber) architecture with one broker and one publisher and 2 subscribers on the same Raspberry Pi.
   a) “sudo apt install mosquito-clients”



   b) Enable the broker and allow it to auto-start after reboot (Question: What does “systemctl” do? The systemctl is used for managing system services.
   “sudo systemctl enable mosquito”



c) The broker should be running. You can confirm by entering “sudo systemctl status mosquito”.

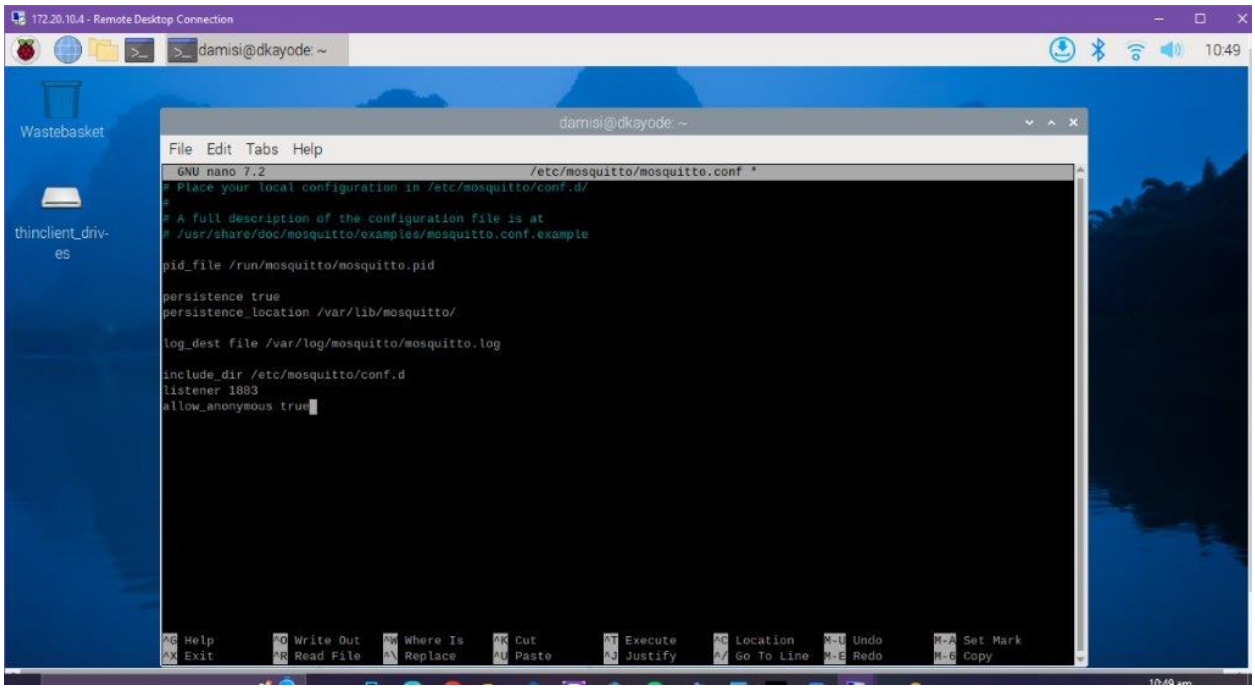


d) Before publishing and sending messages to the broker, a few settings need to be configured. “sudo nano /etc/mosquitto/mosquitto.conf”

e) Once in the file, navigate to the bottom of the file. Add the following two lines of code at the bottom:

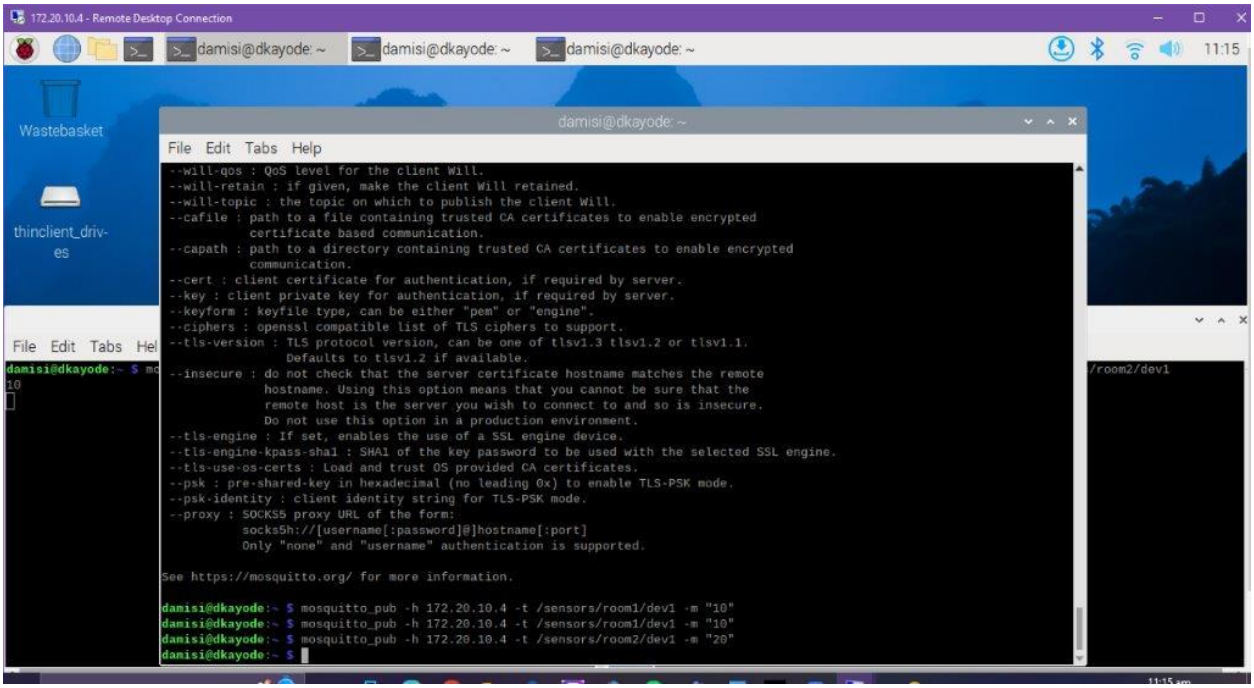
`"listener 1883"`

`"allow_anonymous true"`

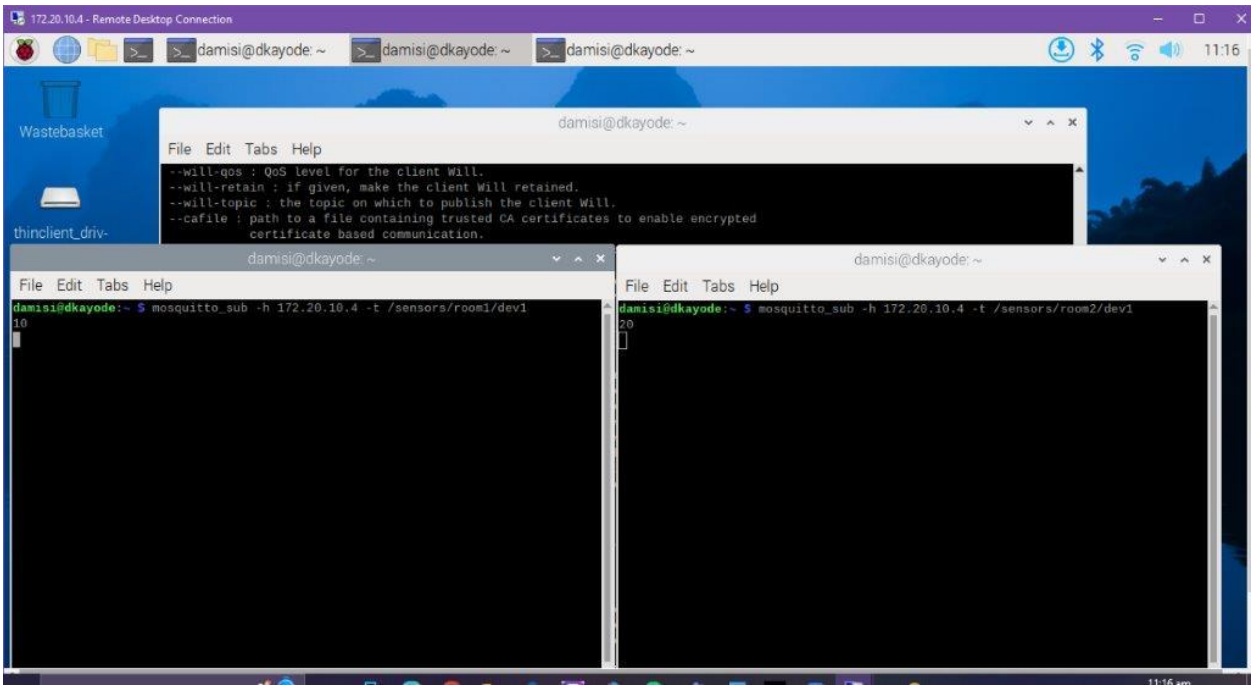


f) Save and execute the following line "`sudo systemctl restart mosquito`"

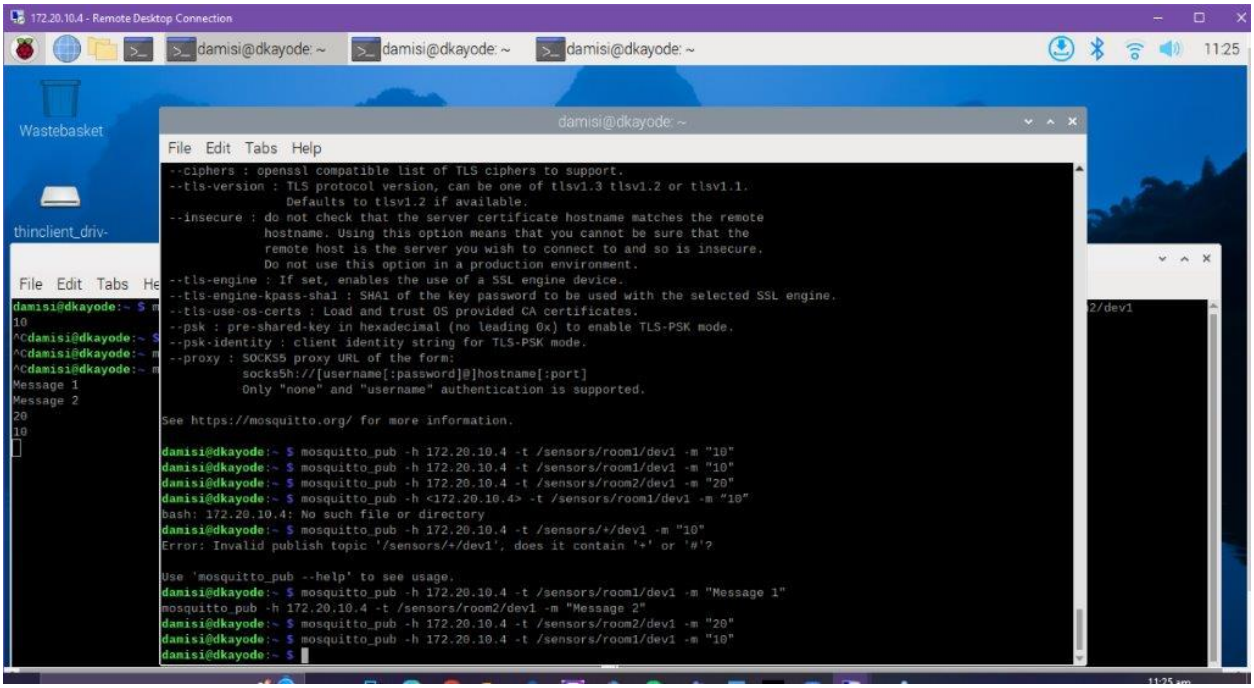
4. Open two additional Terminal windows. Now you should have three open terminals. Make a note that one of them is for "Publisher" and the other two are for "Subscriber 1" and "Subscriber 2", respectively.
5. We set up subscriber 1 in terminal 1 to a "topic": mosquitto_sub -h -t /sensors/room1/dev1
6. We repeat the steps from 5 to terminal 2 for subscriber 2 now: mosquitto_sub -h -t /sensors/room2/dev1
7. In the Publisher Terminal we start to publish messages to the newly created topics

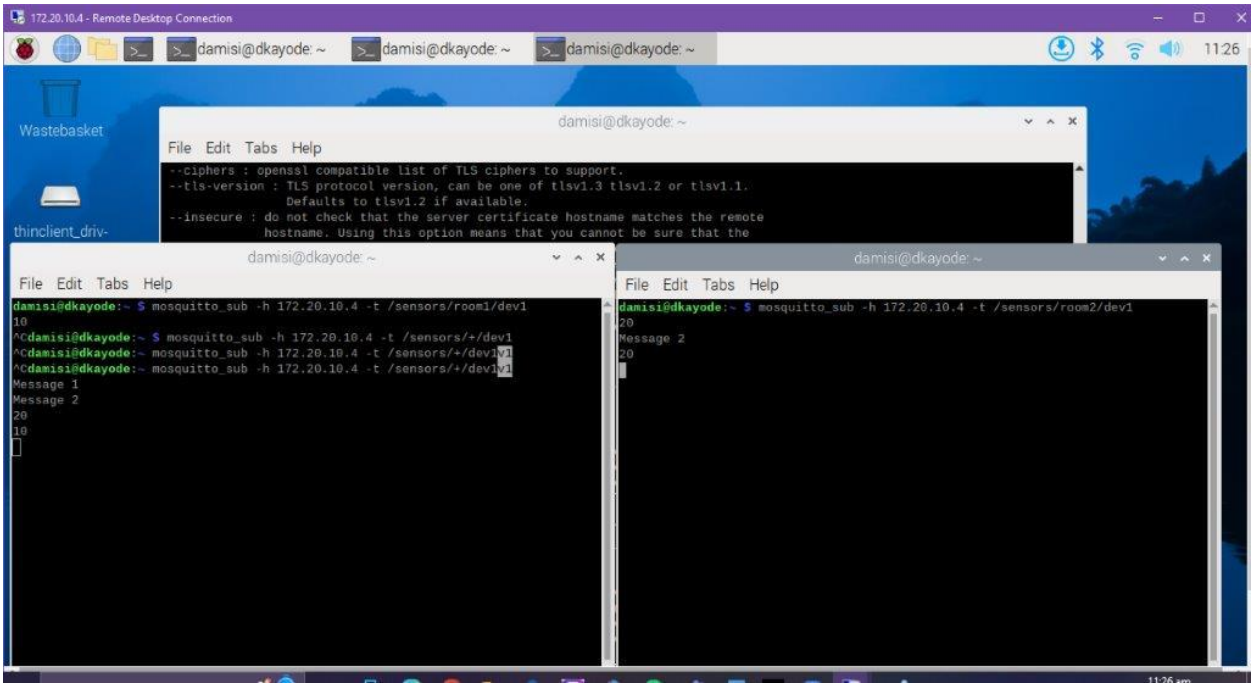


8. We view if the messages were received by terminals for subscriber 1 and subscriber 2.



9. Topic Wildcard: We go back to subscriber 1 and change topic to “/sensors/+/dev1”. Subscriber 1 should receive both messages while Subscriber 2 only receives the second message.





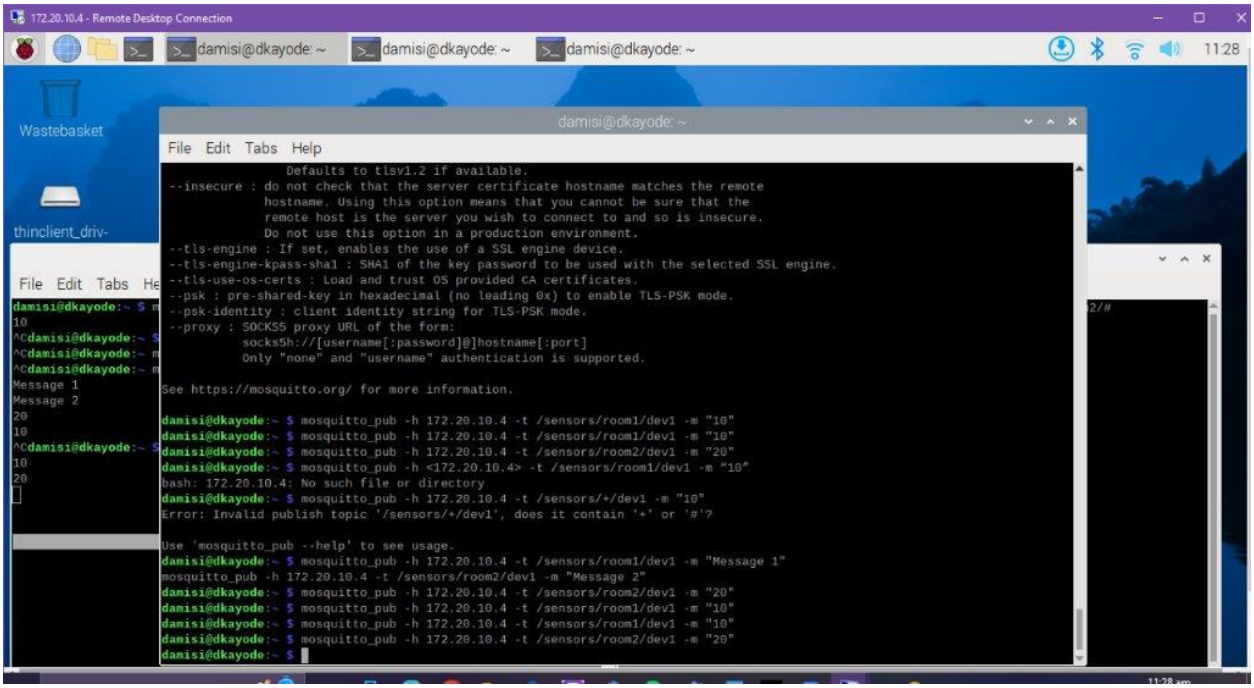
10. Topic Wildcard 2: For Subscriber 1, subscribe to the topic: /sensors/#. Then foR Subscriber 2, subscribe to the topic: /sensors/room2/#.

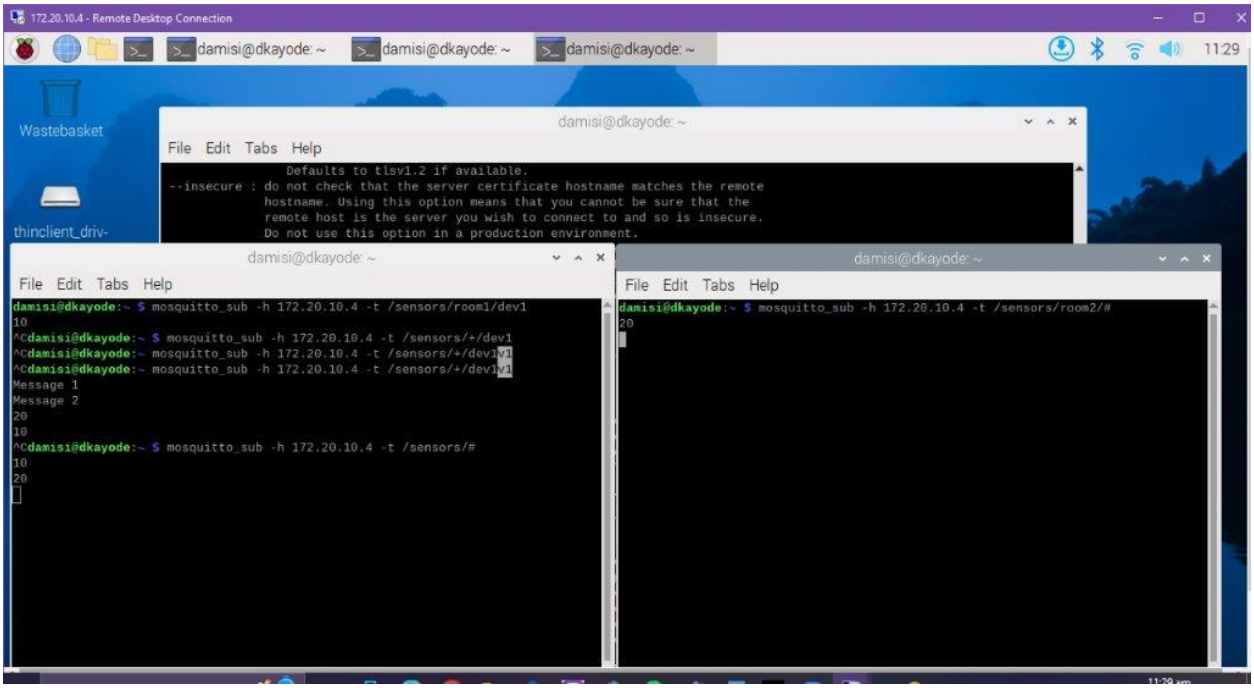
We will send messages similar to those in Step 8, however, using different room numbers and device numbers in each case. Observe and explain the results for both subscribers.

Explain the Results found in this step.

Subscriber 1 receives the message "10" and “20” because it subscribed to the topic /sensors/room1/dev1.

Subscriber 2 will receive the message "20" because it subscribed to the topic /sensors/room2/dev1.





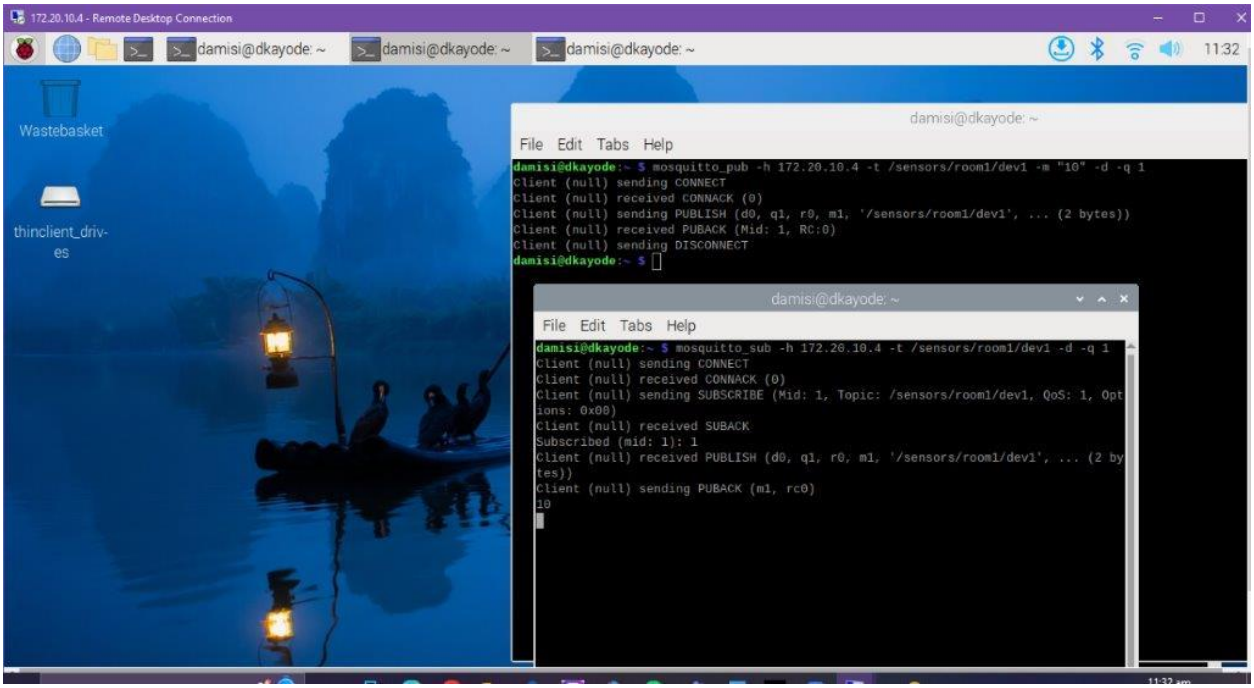
11. Handshaking messages and QoS: For Publisher and Subscriber 1, enable debug messages using the option "-d" to observe handshaking messages on both sides. Use option "-q" to change the QoS flags (0, 1, 2) on both Publisher and Subscriber and observe the result.

    mosquitto_sub -h -t /sensors/room1/dev1 -d -q 1

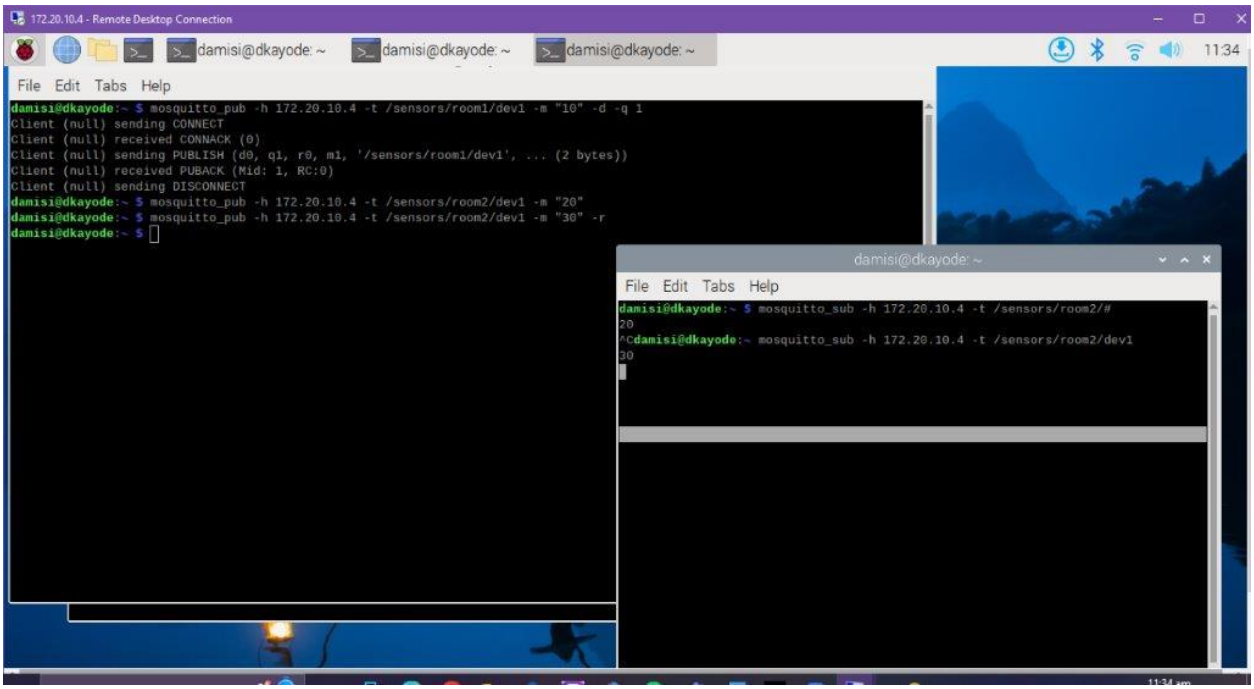
    mosquitto_pub -h -t /sensors/room1/dev1 -m "10" -d -q 1

    Explain the differences between the message exchange.

    By choosing QoS level 1 and adding the -d option, you make sure messages are sent more reliably between the subscriber and publisher. QoS 1 ensures messages are delivered at least once, and the -d option helps in understanding the communication process by showing extra details about the message exchange. It is like a safety measure for more dependable communication.

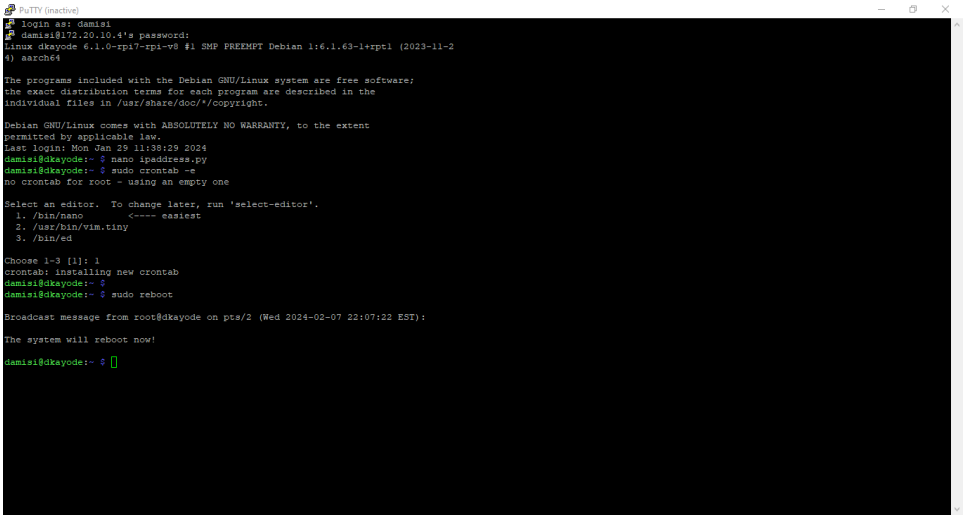


12. Retained messages: Go back to Subscriber 2 Terminal and stop the client. We will now go to the Publisher and send the same message from previous step with the addition of -r. mosquitto_pub -h -t /sensors/room2/dev1 -m “30” -r



Post Lab Exercise:

https://community.element14.com/members-area/personalblogs/b/blog/posts/get-an-email-from-your-raspberry-pi-with-its-ip-address



IV. Conclusion

The overall lab was not hard to complete step by step. The directions were clear in what we had to document and the various screenshots we needed to show we had achieved the goal. If I had to improve the lab directions, I would make it clear with a marker or indicator when screenshots were required. We paused every step to determine whether it was necessary or not and how many screenshots would be needed to display all information was covered.

We did run into problems with the post lab section. We were not able to send the IP address through email from our Pi. Terminal would open the file on PuTTY but not give us the IP address given the downloaded file contents.