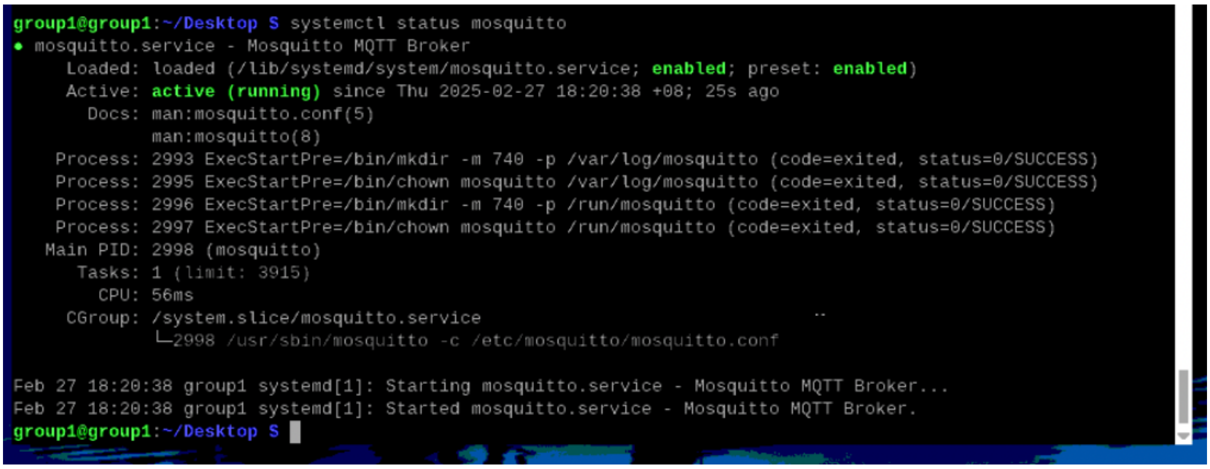
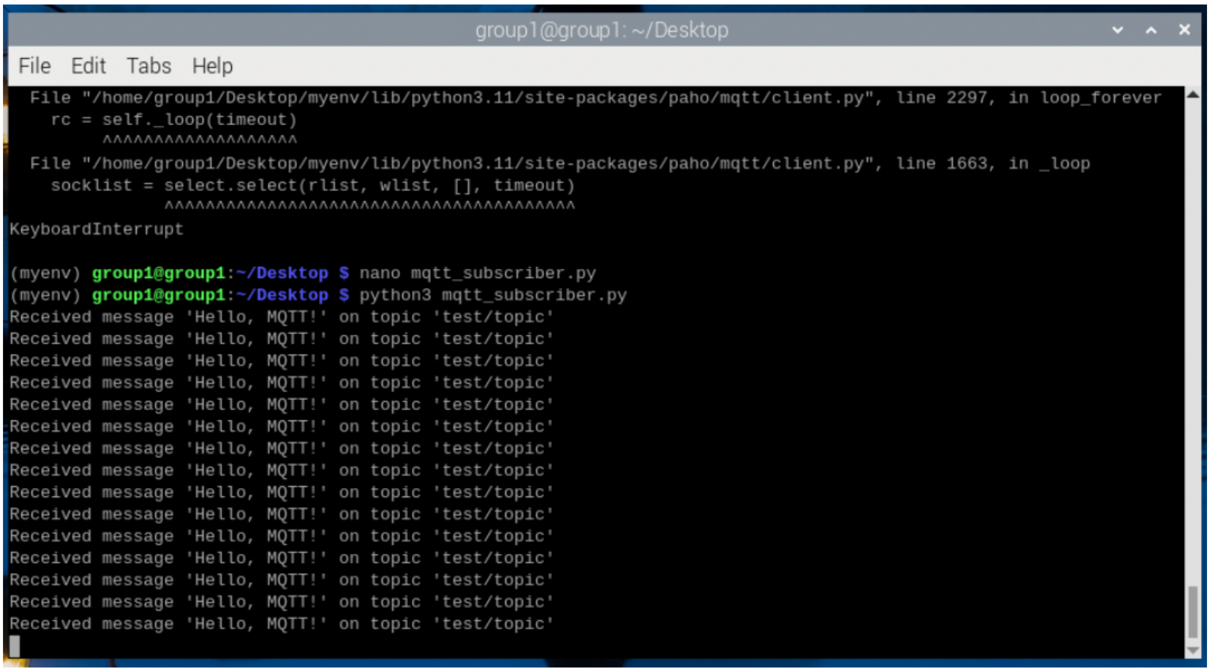
Verify that Mosquitto is running



```
group1@group1:~/Desktop $ systemctl status mosquitto
● mosquitto.service - Mosquitto MQTT Broker
     Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-02-27 18:20:38 +08; 25s ago
       Docs: man:mosquitto.conf(5)
             man:mosquitto(8)
    Process: 2993 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=exited, status=0/SUCCESS)
    Process: 2995 ExecStartPre=/bin/chown mosquitto /var/log/mosquitto (code=exited, status=0/SUCCESS)
    Process: 2996 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exited, status=0/SUCCESS)
    Process: 2997 ExecStartPre=/bin/chown mosquitto /run/mosquitto (code=exited, status=0/SUCCESS)
   Main PID: 2998 (mosquitto)
      Tasks: 1 (limit: 3915)
        CPU: 56ms
     CGroup: /system.slice/mosquitto.service
             └─2998 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Feb 27 18:20:38 group1 systemd[1]: Starting mosquitto.service - Mosquitto MQTT Broker...
Feb 27 18:20:38 group1 systemd[1]: Started mosquitto.service - Mosquitto MQTT Broker.
group1@group1:~/Desktop $ 
```

MQTT Subscriber receiving messages every 5 seconds



```
  File "/home/group1/Desktop/myenv/lib/python3.11/site-packages/paho/mqtt/client.py", line 2297, in loop_forever
    rc = self._loop(timeout)
         ^^^^^^^^^^^^^^^^^^^
  File "/home/group1/Desktop/myenv/lib/python3.11/site-packages/paho/mqtt/client.py", line 1663, in _loop
    socklist = select.select(rlist, wlist, [], timeout)
               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
KeyboardInterrupt

(myenv) group1@group1:~/Desktop $ nano mqtt_subscriber.py
(myenv) group1@group1:~/Desktop $ python3 mqtt_subscriber.py
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
Received message 'Hello, MQTT!' on topic 'test/topic'
```

Lab Assignment

mqtt_subscriber:

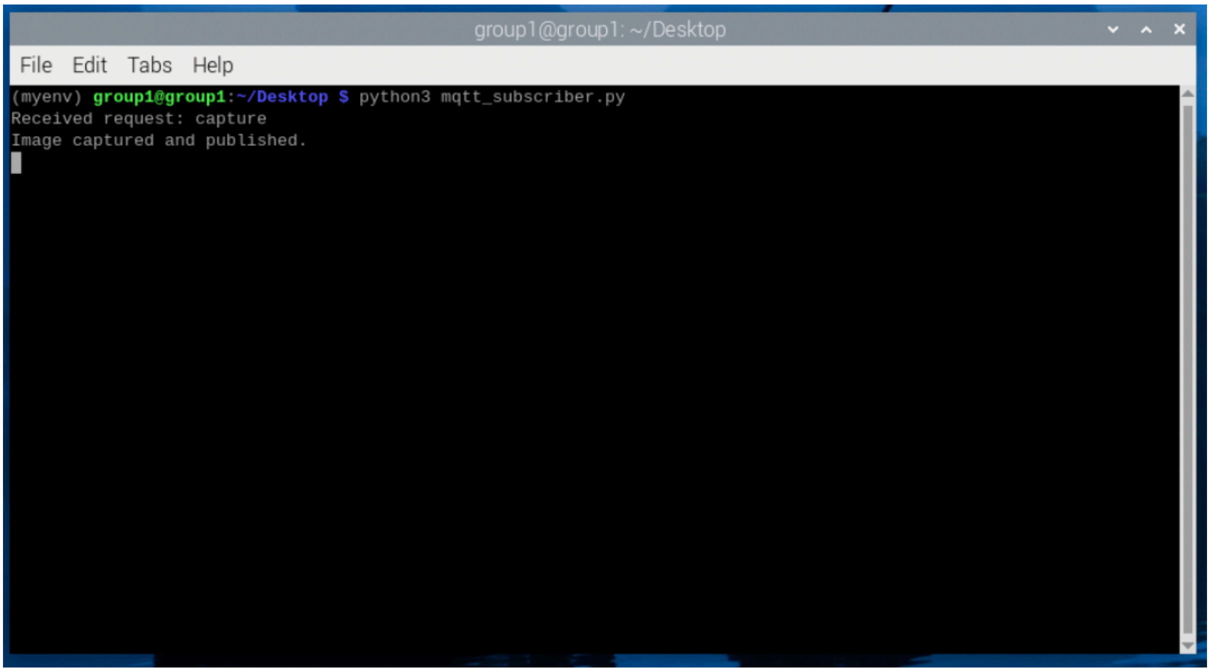
```
import paho.mqtt.client as mqtt
import cv2
import base64
import numpy as np

# MQTT Configuration
BROKER = "172.20.10.4"
REQUEST_TOPIC = "image/request"
IMAGE_TOPIC = "image/data"
```

```
def on_message(client, userdata, message):
    print(f"Received request: {message.payload.decode()}")

    # Capture image from webcam
    cap = cv2.VideoCapture(0)  # 0 is the default webcam
    ret, frame = cap.read()
    cap.release()

    if ret:
        # Encode image as Base64
        _, buffer = cv2.imencode('.jpg', frame)
        img_base64 = base64.b64encode(buffer).decode()

        # Publish image data
        client.publish(IMAGE_TOPIC, img_base64)
        print("Image captured and published.")
    else:
        print("Failed to capture image.")

# Initialize MQTT Client
client = mqtt.Client(client_id="Camera",
callback_api_version=mqtt.CallbackAPIVersion.VERSION2)
client.on_message = on_message
client.connect(BROKER, 1883)

# Subscribe to the request topic
client.subscribe(REQUEST_TOPIC)
client.loop_forever()
```

mqtt_publisher:

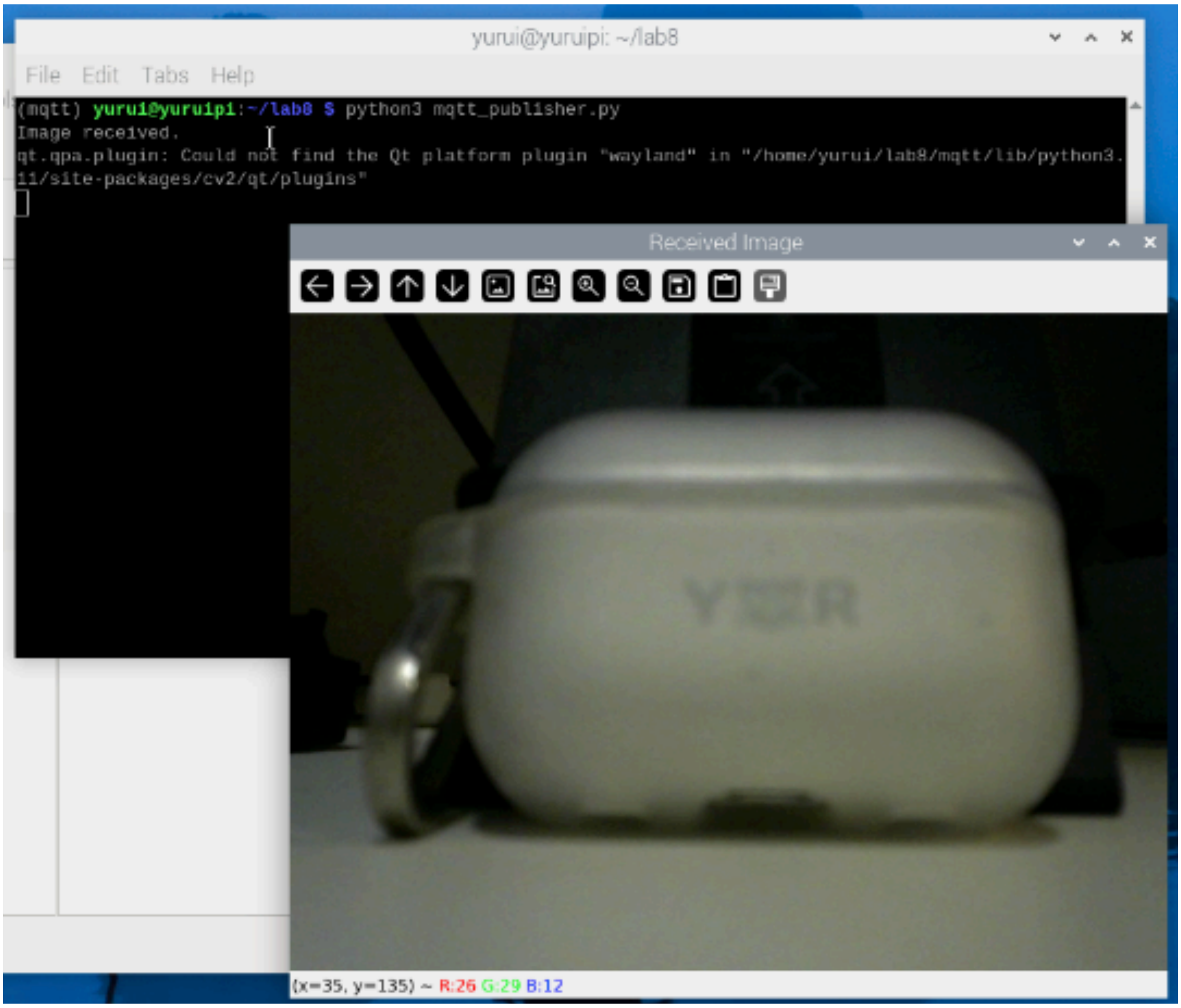
```
import paho.mqtt.client as mqtt
import time
import cv2
import base64
import numpy as np

# MQTT Configuration
BROKER = "172.20.10.4"
REQUEST_TOPIC = "image/request"
IMAGE_TOPIC = "image/data"

def on_message(client, userdata, message):
    print("Image received.")

    # Decode Base64 image
    img_data = base64.b64decode(message.payload)
    np_arr = np.frombuffer(img_data, np.uint8)
    frame = cv2.imdecode(np_arr, cv2.IMREAD_COLOR)

    # Show image
    cv2.imshow("Received Image", frame)
    cv2.waitKey(5000)  # Display for 5 seconds
    cv2.destroyAllWindows()

# Initialize MQTT Client
client = mqtt.Client(client_id="Viewer",
callback_api_version=mqtt.CallbackAPIVersion.VERSION2)
client.on_message = on_message
client.connect(BROKER, 1883)

# Subscribe to the image data topic
client.subscribe(IMAGE_TOPIC)

# Send an image request
client.publish(REQUEST_TOPIC, "capture")

# Start loop to receive the image
client.loop_forever()
```

<u>mqtt_subscriber receiving request and sending an image</u>



```
(myenv) group1@group1:~/Desktop $ python3 mqtt_subscriber.py
Received request: capture
Image captured and published.
```

<u>mqtt_publisher sending request and receiving the image</u>



```
(mqtt) yurui@yuruipi:~/lab8 $ python3 mqtt_publisher.py
Image received.
qt.qpa.plugin: Could not find the Qt platform plugin "wayland" in "/home/yurui/lab8/mqtt/lib/python3.
11/site-packages/cv2/qt/plugins"
```