# Tutorial No.2

**Roll No-CS22D003**

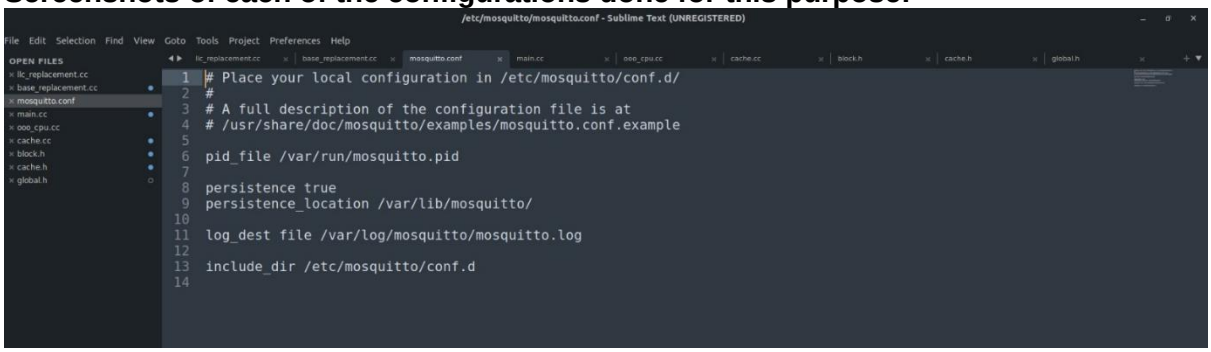
**Case Study:**
I am considering a simple case of an IKEA store. I am using a random number generator to generate a random number which shows the number of people in IKEA stores at the present time every 6 seconds.

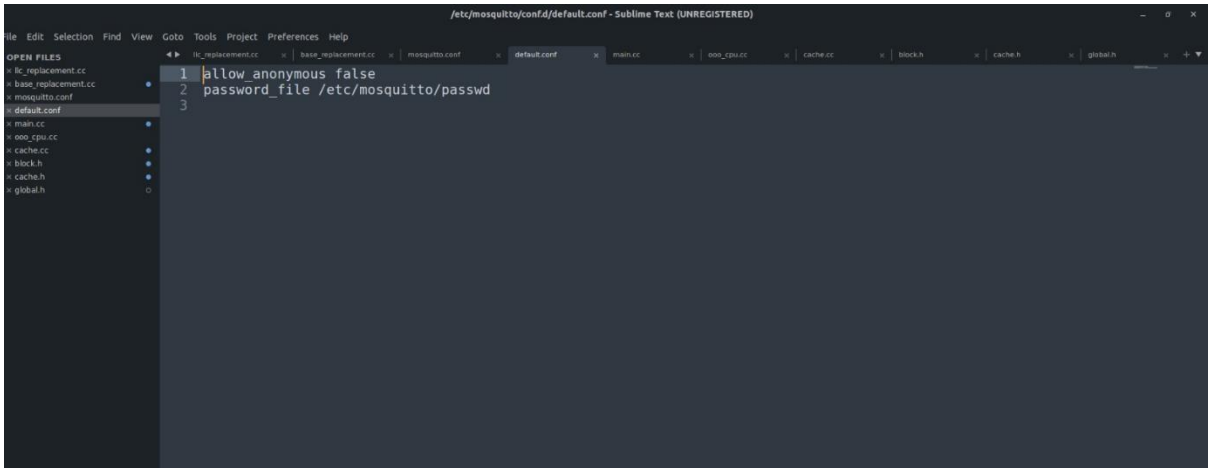
If the number of people in an IKEA store is greater than 900, the sensor senses it and it will publish a message “over crowded with” with the number of people to the subscribers. Otherwise, the number of people will be published as a message. People with health issues, old people or people with children can better plan and leave the IKEA store.

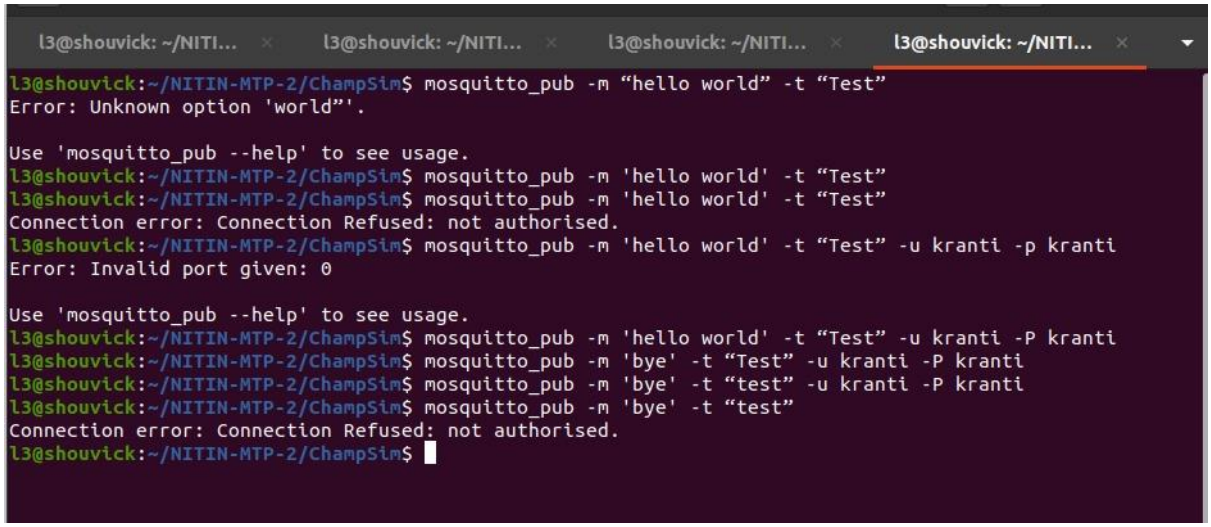
I have used Mosquitto which is an open-source message broker that implements the MQTT protocol. I wrote python code for publisher and subscriber.

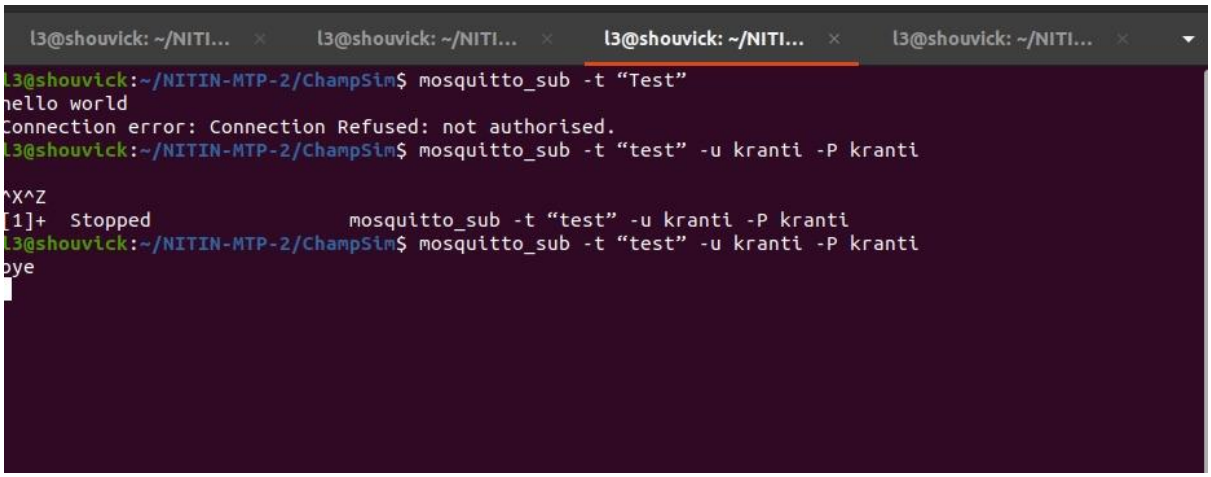
Authentication of subscriber and publisher is done using a pre-set username & password. MQTT has a utility to generate a password file. I have used mosquitto_passwd tool to create a password and created a password file. conf.d/default.conf is a password file and mosquitto.conf is edited to include this file with “include /etc/mosquitto/conf.d”.

**Screenshots of each of the configurations done for this purpose:**




l3@shouvick: ~/NITI...
l3@shouvick: ~/NITI...
l3@shouvick: ~/NITI...
l3@shouvick: ~/NITI...
l3@shouvick:~/NITIN-MTP-2/ChampSim$ mosquitto_pub -m “hello world” -t “Test”
Error: Unknown option 'world”'.
Use 'mosquitto_pub --help' to see usage.
l3@shouvick:~/NITIN-MTP-2/ChampSim$ mosquitto_pub -m 'hello world' -t “Test”
l3@shouvick:~/NITIN-MTP-2/ChampSim$ mosquitto_pub -m 'hello world' -t “Test”
Connection error: Connection Refused: not authorised.
l3@shouvick:~/NITIN-MTP-2/ChampSim$ mosquitto_pub -m 'hello world' -t “Test” -u kranti -p kranti
Error: Invalid port given: 0
Use 'mosquitto_pub --help' to see usage.
l3@shouvick:~/NITIN-MTP-2/ChampSim$ mosquitto_pub -m 'hello world' -t “Test” -u kranti -P kranti
l3@shouvick:~/NITIN-MTP-2/ChampSim$ mosquitto_pub -m 'bye' -t “Test” -u kranti -P kranti
l3@shouvick:~/NITIN-MTP-2/ChampSim$ mosquitto_pub -m 'bye' -t “test” -u kranti -P kranti
l3@shouvick:~/NITIN-MTP-2/ChampSim$ mosquitto_pub -m 'bye' -t “test”
Connection error: Connection Refused: not authorised.
l3@shouvick:~/NITIN-MTP-2/ChampSim$


l3@shouvick: ~/NITI...
l3@shouvick: ~/NITI...
l3@shouvick: ~/NITI...
l3@shouvick: ~/NITI...
l3@shouvick:~/NITIN-MTP-2/ChampSim$ mosquitto_sub -t “Test”
hello world
Connection error: Connection Refused: not authorised.
l3@shouvick:~/NITIN-MTP-2/ChampSim$ mosquitto_sub -t “test” -u kranti -P kranti
^X^Z
[1]+ Stopped mosquitto_sub -t “test” -u kranti -P kranti
l3@shouvick:~/NITIN-MTP-2/ChampSim$ mosquitto_sub -t “test” -u kranti -P kranti
bye

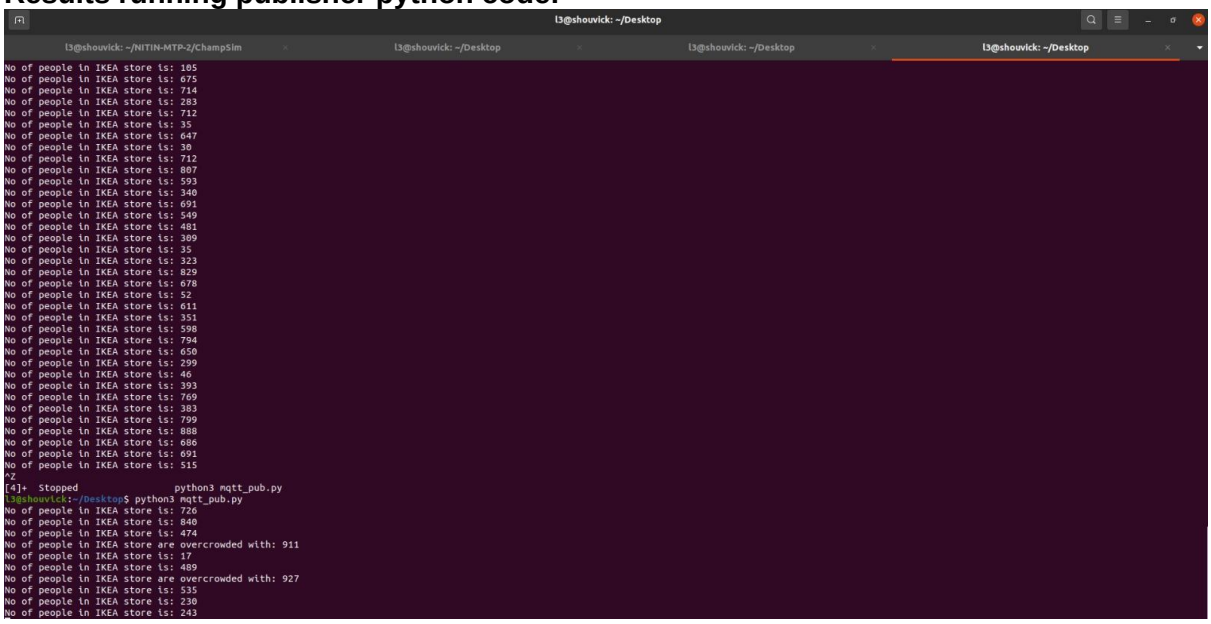
**Python Code for publisher:**

```
#Program for MQTT
import paho.mqtt.publish as publish
import sys
import random
import time

broker_url = "localhost"
broker_port = 1883
topic = "test/storeoccupancy"
username = "kranti"
password = "kranti"

while True:
    peoplecount = random.randint(0, 1000)
    if peoplecount >=900:
        print("No of people in IKEA store are overcrowded with: " +
str(peoplecount))
        msg="overcrowded with: " + str(peoplecount)
        publish.single(topic, msg, hostname=broker_url, port=broker_port,
auth={'username': username, 'password': password})
    else:
        print("No of people in IKEA store is: " + str(peoplecount))
        publish.single(topic, peoplecount, hostname=broker_url,
port=broker_port, auth={'username': username, 'password': password})
    time.sleep(5)
```

**Python Code for subscriber:**

```python
import paho.mqtt.client as mqtt


broker_url = "localhost"
broker_port = 1883
username = "kranti"
password = "kranti"

def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("test/storeoccupancy")

def on_message(client, userdata, msg):
    print(msg.topic+"- No of people in the IKEA store
now:  "+str(msg.payload))

client = mqtt.Client()
client.username_pw_set(username, password) # set the username and password
client.on_connect = on_connect
client.on_message = on_message

client.connect(broker_url, broker_port, 60)

client.loop_forever()
```

**Results running publisher python code:**



## Results running subscriber python code: