

CS5453 Internet of Things

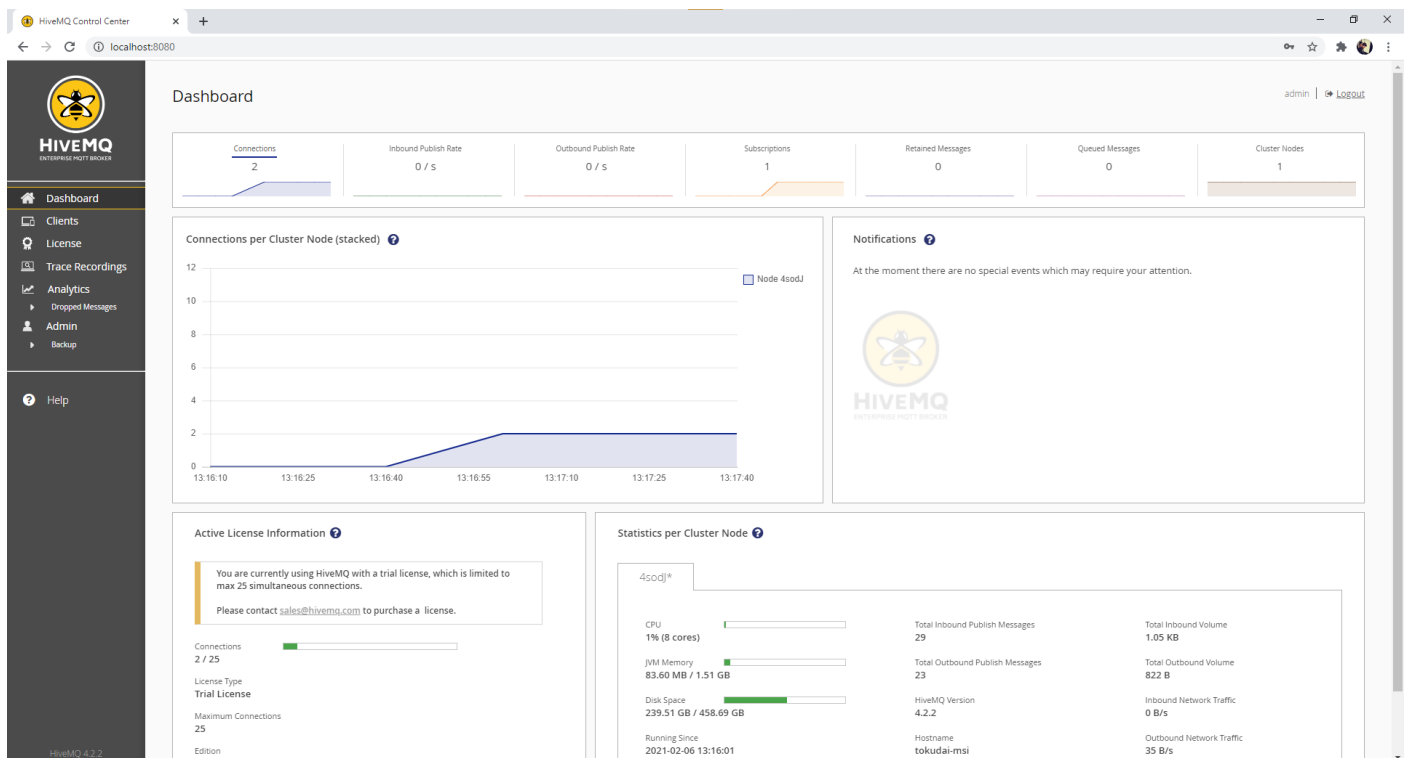
Tutorial 1

Submitted by Vishal Singh Yadav | CS20MTECH01001

Activity 1

In this activity, we have to run the subscriber and publisher code and show the hive dashboard results.

We changed the code in sub.py to subscribe to "location/pub" to subscribes to the publisher. We then run the code and get the following output in the Hive dashboard.



We see the connection count is 2 as only one publisher and one subscriber is connected to Hive as the point. Once we both the pub and sub, we start receiving messages from publisher to subscriber. These messages are printed in the terminal, as shown below.

```
source /home/tokudai/miniconda3/bin/activate
(base) tokudai@tokudai-nsi:/mnt/c/Users/visha/Documents/Assignments$ source /home/tokudai/miniconda3/bin/activate
(base) tokudai@tokudai-nsi:/mnt/c/Users/visha/Documents/Assignments$ conda activate assignments
(assignments) tokudai@tokudai-nsi:/mnt/c/Users/visha/Documents/Assignments$ cd IoT/Tutorial\ 1/Iot_activity1/
(assignments) tokudai@tokudai-nsi:/mnt/c/Users/visha/Documents/Assignments/IoT/Tutorial 1/Iot_activity1$ python pub.py
y
Connected to broker
[]

source /home/tokudai/miniconda3/bin/activate
(base) tokudai@tokudai-nsi:/mnt/c/Users/visha/Documents/Assignments/IoT/Tutorial 1/Iot_activity1$ source /home/tokudai/miniconda3/bin/activate
(base) tokudai@tokudai-nsi:/mnt/c/Users/visha/Documents/Assignments/IoT/Tutorial 1/Iot_activity1$ conda activate assignments
(assignments) tokudai@tokudai-nsi:/mnt/c/Users/visha/Documents/Assignments/IoT/Tutorial 1/Iot_activity1$ python sub.py
y
Connected to broker
Message Received is : b"{'x': 115, 'y': 79}"
Message Received on Topic location/pub
Message Received is : b"{'x': 42, 'y': 182}"
Message Received on Topic location/pub
Message Received is : b"{'x': 231, 'y': 136}"
Message Received on Topic location/pub
Message Received is : b"{'x': 49, 'y': 86}"
Message Received on Topic location/pub
Message Received is : b"{'x': 92, 'y': 198}"
Message Received on Topic location/pub
Message Received is : b"{'x': 43, 'y': 44}"
Message Received on Topic location/pub
Message Received is : b"{'x': 89, 'y': 222}"
Message Received on Topic location/pub
Message Received is : b"{'x': 125, 'y': 119}"
Message Received on Topic location/pub
Message Received is : b"{'x': 41, 'y': 140}"
Message Received on Topic location/pub
Message Received is : b"{'x': 224, 'y': 138}"
Message Received on Topic location/pub
Message Received is : b"{'x': 23, 'y': 86}"
Message Received on Topic location/pub
Message Received is : b"{'x': 117, 'y': 112}"
Message Received on Topic location/pub
Message Received is : b"{'x': 206, 'y': 194}"
Message Received on Topic location/pub
Message Received is : b"{'x': 209, 'y': 225}"
Message Received on Topic location/pub
```

We can see the publisher is connected to the broker, and the subscriber is receiving messages from the publisher and printing them on screen.

Activity 2

1. In the first subtask, we need to get the client name from the command line. We use `client_name = sys.argv[1]` to get the client name.
2. We now have to set client name as name for client object. We use `client = mqttClient.Client(client_name)` to set client object name as client_name
3. In third subtask we have to subscribe to all clients except the current one. We use the following code to do the same.

```
for cl in all_clients:
    if client_name != cl:
        client.subscribe("location/" + str(cl))
```

4. In this subtask we have to publish current location. We use the following code:

```
while time.time() < end_time:
    # Task-4 Write code here

    client.publish("location/" + client_name, str(curr))
    time.sleep(2)
    curr = location_generator()
```

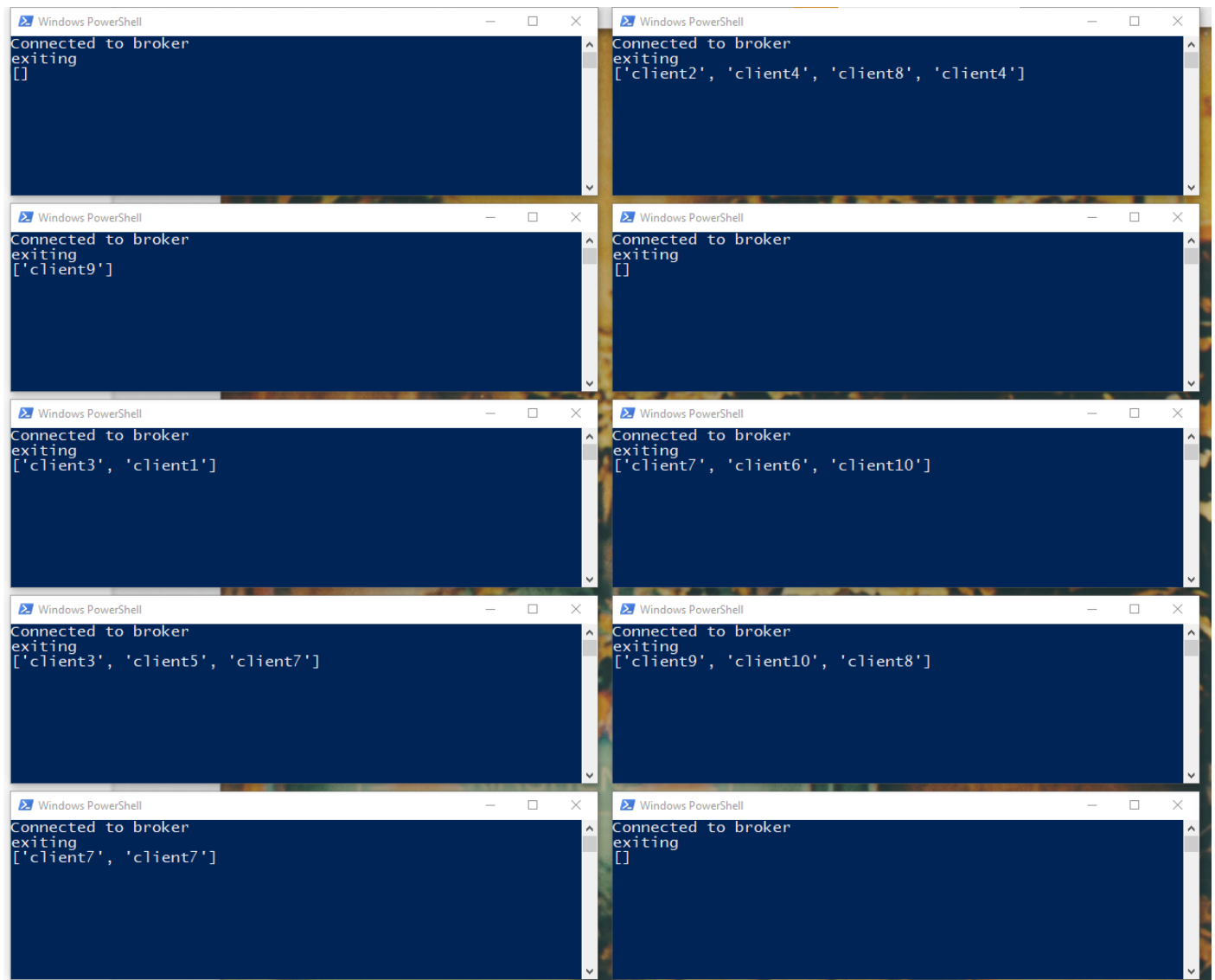
5. We have to complete `on_message()` function. The completed function is given below.

```
def on_message(client, userdata, message):
    # Task-5 Write code here
    location_data = ast.literal_eval(message.payload.decode("UTF-8"))
    cl_name = message.topic.split("/")[-1]

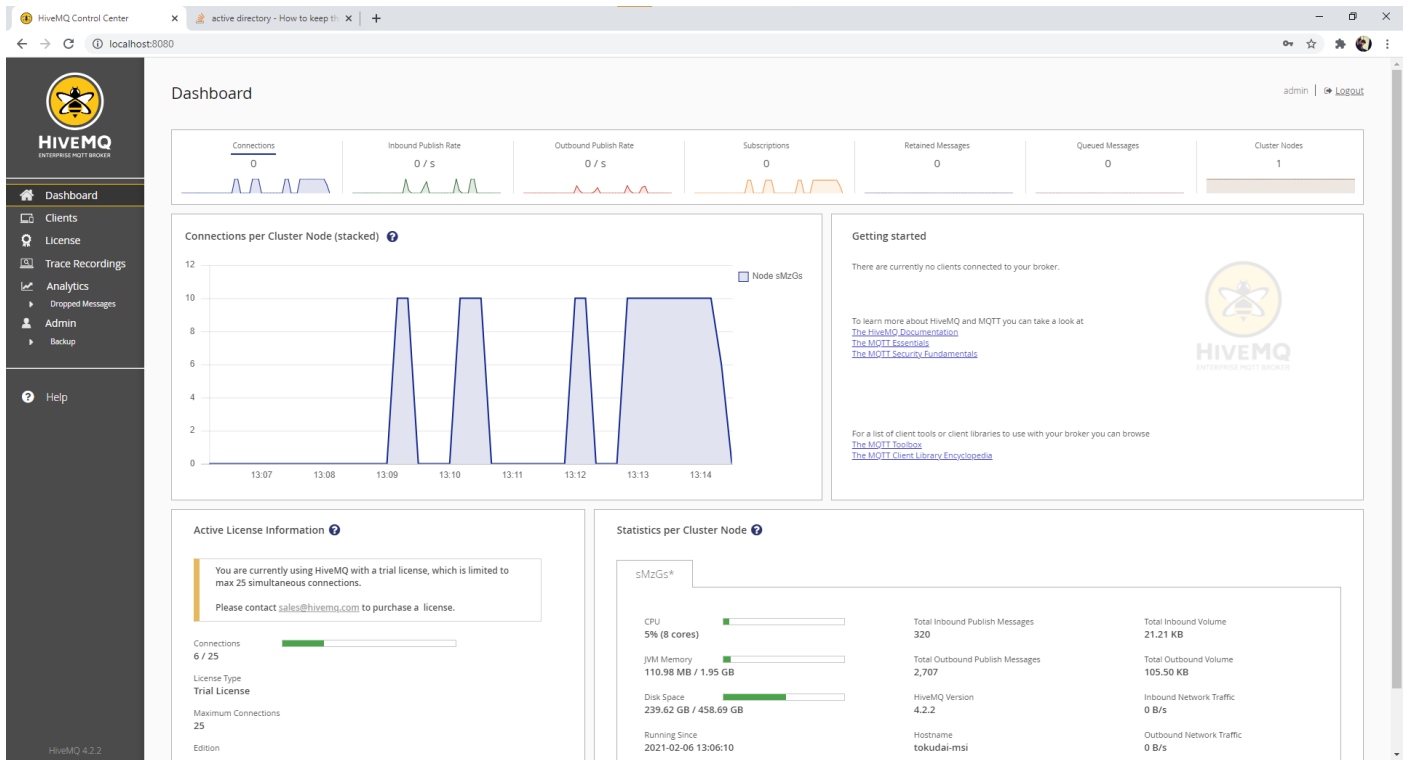
    d = distance(curr, location_data)
    if d < 20:
```

```
contact.append(cl_name)
```

At the end of execution, we print the list of all clients who had a distance of less than 20 to the current clients. The screenshot below shows the output of the same.



The hive dashboard output is given below.



We can see from the dashboard that 10 connections were connected. The number of connections in the screenshot is 0, as the screenshot was taken after the execution had stopped.