MQTT Exercises

Ing. Stabili Dario
dario.stabili@unimore.it
http://weblab.ing.unimo.it/people/stabili

Requirements

- Install **Eclipse Mosquitto** as Broker
  https://mosquitto.org/download/
- Install **Eclipse PAHO MQTT** client for python
  https://pypi.org/project/paho-mqtt/

Mosquitto

MQTT Exercises

Mosquitto broker

- Eclipse Mosquitto is the message broker used for set up the pub/sub network.
- By default the broker listens to port **1883** (both IPv4 and IPv6).
- It is possible to configure it following the documentation provided at https://mosquitto.org/man/mosquitto-conf-5.html
- Conf file is used to limit accessibility to topics based on different rules (example: topic available only to authenticated devices in write, no restrictions on read, …)

Mosquitto Conf

- Mosquitto conf file to limit access to the broker to only authenticated users, listening to port **8585**, limiting topic access

```
acl_file mosquitto.acl_file
port 8585
allow_anonymous false
password_file ./mosquitto.password
```

- Mosquitto.password contains the list of users and password in the

  user:password

  format. For generate this file, look for the command

  mosquitto_passwd

Mosquitto ACL File

- Mosquitto ACL file is a particular file that allows to specify restriction access to the topics on the broker.
- Allows to make some topics available only to particular users in particular modes.

```
[user <username>]
topic [read|write|readwrite]
<topic>
```

Mosquitto ACL File

- First topics are always related to anonymous users (allow_anonymous MUST be True)
- <topic> can contain wildcards
- Default access type for all topics is readwrite

```
topic read myHome/+/+/brightness

user mom
topic write myHome/#

user admin
topic readwrite myHome/#
topic readwrite adminRoom/#
```

Mosquitto

- Using the command line interface it is possible to specify
  - **-c :** broker config file
  - **-d :** put the broker in background
  - **-h :** display the help
  - **-p :** specify the listening port
  - **-v :** verbose

$ mosquitto -c mosquitto.conf -v

Paho MQTT Client

MQTT Exercises

Paho MQTT

- Paho MQTT is a Python client library which implements versions 3.1 and **3.1.1** of the MQTT protocol.
- Provides a client class to connect to an MQTT broker to publish messages, subscribe to topics and receive published messages.
- Complete documentation available at
  https://pypi.org/project/paho-mqtt/

Paho MQTT Client

- The Client class is the main class used for communicating with an MQTT broker.
- General usage flow:
  - connect to a broker
  - maintain network traffic flow with the broker
  - [subscribe/publish] to a topic
  - disconnect to a broker

Paho MQTT Client

- The class relies on callbacks to receive data back from the broker.
- Available callbacks:
  - on_connect
  - on_disconnect
  - on_message
  - on_publish
  - on_subscribe
  - on_unsubscribe
  - on_log

Paho MQTT Sub

- Example of a sub that subscribes to the **$SYS/#** topic (generic topics that contains statistics collected by the broker, they are read-only)

  (sub.py)

```
import paho.mqtt.client as mqtt

def on_connect(clinet, userdata, flags, rc):
    print("Connected with resultus code: {}".format(rc))

    client.subscribe("$SYS/#")

def on_message(client, userdata, msg):
    print("{}::{}".format(msg.topic, msg.payload))

client = mqtt.Client()
client.username_pw_set("sar", "password")
client.on_connect = on_connect
client.on_message = on_message

client.connect("localhost", 8585, 60)

client.loop_forever()
```

Paho MQTT Pub

- Example of a pub that publishes to the **sar/example** topic

(pub.py)

```
import paho.mqtt.client as mqtt

TOPIC = u"sar/example"

def on_connect(clinet, userdata, flags, rc):
            print("Connected with resultus code: {}".format(rc))



def on_publish(client, userdata, mid):
            print("{}::{}::{}".format(client, userdata, mid))

client = mqtt.Client()

client.username_pw_set("sar", "password")
client.on_connect = on_connect
client.on_publish = on_publish
client.connect("localhost", 8585, 60)
client.publish(TOPIC, u"Hello World")
```

Try it yourself

MQTT Exercises