Mosquitto- Mosquittto is an open-source message broker service that uses the MQTT protocol to send and receive messages, typically with IOT (Internet of Things) devices.

Few imp points –

Token- it is used for authentication and will be mailed to end user as soon as he/she signup in our portal. This is used will activating the device.

Deviceid- it will be unique for each and every device

Note\* for sending payload we need to always append the device id so that broker can understand to which device command should be send.

How to take remote access of AWS ubuntu

* Install the software putty
* Open up the Aws account and copy the public DNS
* paste the public DNS instead of Host name(or IP address)
* now go to SSH and click on auth-🡪 tick the Display authentication and untick the bypass authentication
* now provide the private key file for authentication( already provided on drive—just download it and select the file by clicking on browse button)
* On click on open and black window opens. Now login with ubuntu

How to setup WinScp for transferring files from local machine to server (Ubuntu)

* Install software WinScp
* In hostname write down the IP address of the the AWS iot Instance
* On browse the private key ( same key will be applied as applied in putty)
* Select the file protocol SFTP
* Login the session
* For authentication popup select yes to all

**Note\* - make sure AWS inbounds rules are set for 22 port no. access is granted to your local machine.**

How to setup mosquitto on ubuntu

Login to your ubuntu OS with non root user (for AWS mainly with ubuntu)

***$ sudo apt-get install mosquitto mosquitto-clients***

By default, Ubuntu will start the Mosquitto service after install.

Few important commands for mosquitto.

* How to stop mosquitto

***$ pkill mosquitto***

* How to start mosquitto with customized conf file ( how to design conf file try to study the conf file of mosquittio else through winscp software go to /***etc/mosquitto/conf.d/mosquitto.conf*** file and try to read already customized conf file )

***$ mosquitto –c /etc/mosquitto/conf.d/mosquitto.conf –d*** ( d is used for running the process in background)

Note\* Now this already customized file contain mosquito on 1884 port and while currently we are doing testing on 1883 port

**For running default mosquito-- $ mosquito –c**

* How to check which all instance are running

$ netstat –an

**Note\* - default mosquitto runs on 1883 port and already customized port is 1884( with authentication and authorization), web sockets are on 8083 port( for all these study the conf file well)**

**Mosquito 1884(this customized port authenticate and authorized through redis db—this authentication and authorization will be used while activating hardware)**

Authentication and authorization

Redis db is responsible for auth token mapping.

Encrypt- PBKDF2

Already done with the encrypt with existing backend code.

* **For superuser-** study the config file [/etc/mosquitto/conf.d/mosquitto.conf], last line is related to super user( just check the last query of auth plugin in conf file which topic is given super user access)
* How to change the password manually for any topic for authentication

Open linux(putty instance) ---***$ np***🡪enter the password you want to set and just copy that

Now open another session of putty and write

***$redis-cli***

***Del NAME\_OF\_Topic\_You\_Want\_To\_Change\_the\_password***

***Set Name\_of\_the\_topic PASTEthe\_copy\_term***

* For authorization- we need to set the token with topic access to that token separated by “- ”

Ex- 12345-qlikAway\_Smartswitch 2

12345- this token

qlikAway\_Smartswitch- topic

2- read/write access( 0 for read only access , 1 for write only acess)

{Just fire the command set 12345-qlikAway\_Smartswitch 2}

Note\* for hardware side it’s already done and code is mailed to aniket.

**Few important code of line**

**For updates in software side while switching on port 1884**

1. please add the following lines before  connectivity to broker and do change the port no.
2. /////////////////////////////////////////////////  
   var options={  
       clientId: ' append DEVICE\_ID ',  
       username:  ' append email\_ID ',  
       password:  ' append the token ',  
   };  
     
   //////////////////////////////////////////////////////
3. var client = mqtt.connect('mqtt://[13.126.36.205](http://13.126.36.205/):**1884**',options);
4. following line will modified ---adding up the end users required Quality of services and retaining the message
5. /////////////////////////////////////////////////////////////////////////////////////
6. client.publish('hello/demo', 'Hello mqtt',{qos:"2",retain :true});
7. client.subscribe('hello/demo',{qos:"2"});

**For device Activation through adnate token**

1) end user need to signup for in our portal (either web or app)

2) after signing up welcome mail will be send with his/her *creditional*

*3) Then conf of device is required*

*4) conf--> ssid-->password--> token*

*5) acls, userid and password  will be maped at redis*

*6) after saving the creditional user need not to save the token again and again for the same device.(toking editing can be done in case, token input is incorrect)*

*7) after successful activation of device, offline and online mode will be turned on.*