MQTT PROTOCOL

-**Kunal bharti**

Introduction

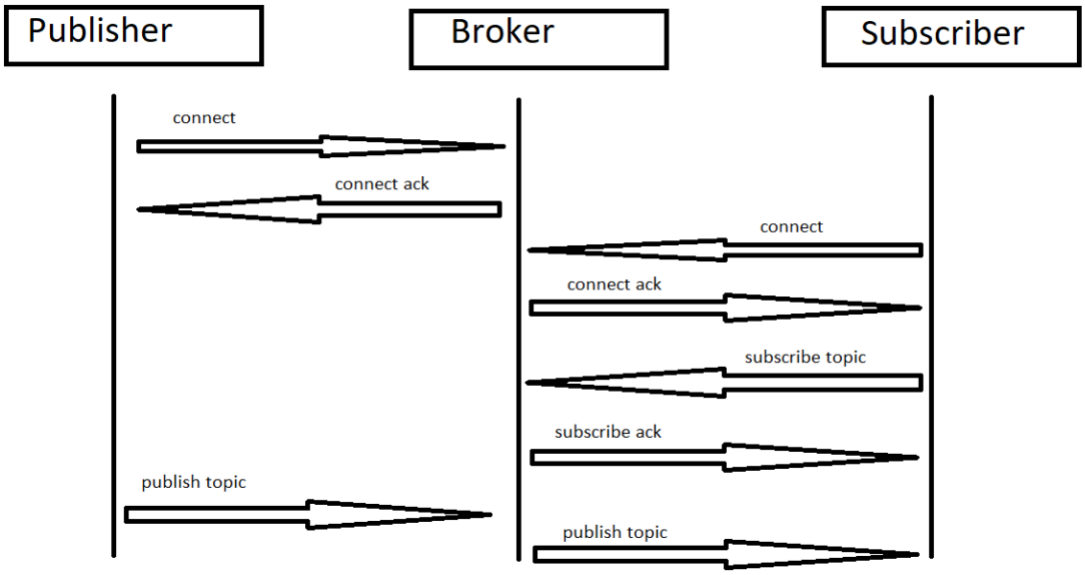
IOT (internet of things) is all about network of interconnecting devices, collecting and exchanging data .MQTT (message queuing telemetry transport) is one of the ways to do it in a secure manner

MQTT is a is an open OASIS and ISO standard (ISO/IEC PRF 20922)[3] lightweight, simple machine to machine TCP/IP based protocol which can be used for communication between the devices(wired and wireless both).

It is a **Publish-Subscribe** architecture based protocol , it does not require direct connection between devices as it relays all the massages via Broker (central server) , which makes it very appropriate for the IOT technology as new devices can be easily added to the existing setup and also does not require compatibility with other devices.

Working mechanism

In MQTT protocol data flows in both direction from publisher to subscriber and from subscriber to publisher but in both cases the data is relayed through a central server called **Broker** .The Broker is just a software running on an operating system, it works as a post office regulating all the data traffic from one device to another and decides what data flows to which device but instead of using address of the device it uses **topic** for unique identification. The subscriber has to subscribe to a topic relevant to its requirement, It allows both ***one –to-many*** and ***many-to-one*** model , i.e. a subscriber can subscribe to any number of topics or any number of publisher can publish data to one or any number of topics . It is bi-directional so a device can send the message and at the same time can receive configuration data or control commands simultaneously.



QoS (quality of service)

- QoS 0 (at most once) - in this service level the message is sent only once and does not required any acknowledgment of whether the message is received/delivered or not.
- QoS 1 (at least once) - in the service level the message is multiple times till it receives the acknowledgment of delivery .There is a chance of duplicity in case of acknowledgement message is lost.
- QoS 3 (exactly once) - sender and receiver engage in 2 level handshake to ensure only one copy of the message is delivered.

MQTT packet stucture

Fixed header (always present) Size : 2 byte	**Variable header** (optional) Size: variable	**Payload** (optional) Size: variable

Wild card

when a client want to subscribe to a topic it can subscribe to a perticular topic or it can subscribe to multiple topics simultaneously , it can be done by using wildcards. it can be used only for the subscription activities not publishing .

- Single level wildcard(+) : it can replace only one topic level

  Eg. Myhome/1stfloor/room1/+/temperature

- multilevel wildcards(#) : it can replace multilevel topics

  Eg. Myhome/#/

Miscellaneous

We can send the payload/data of maximum of 268435455 bytes (~268Mb) .The MQTT protocol has no limits on the number of subscriptions. However you will be restrained by available memory.

POC

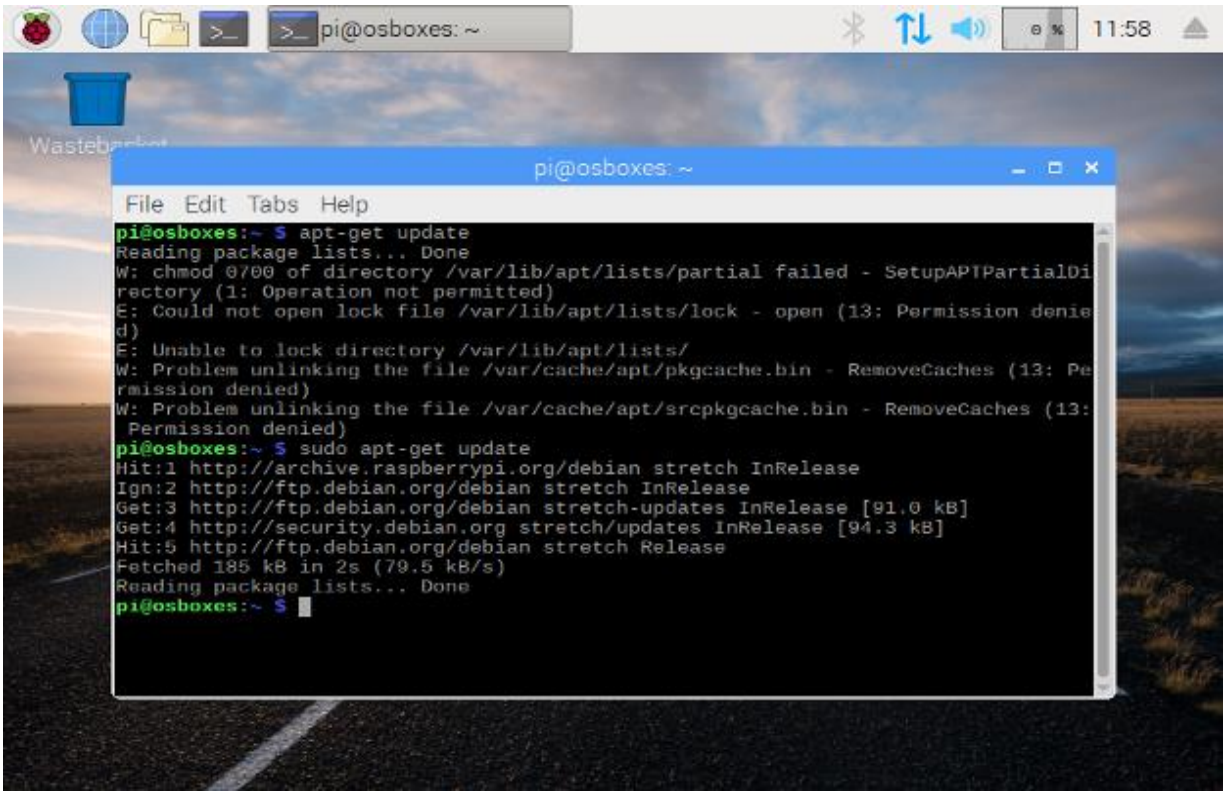
We are taking Raspberry pie OS as our broker and Kali Linux as subscriber . we are using ***mosquito*** tool in both the OS to interact using MQTT protocol

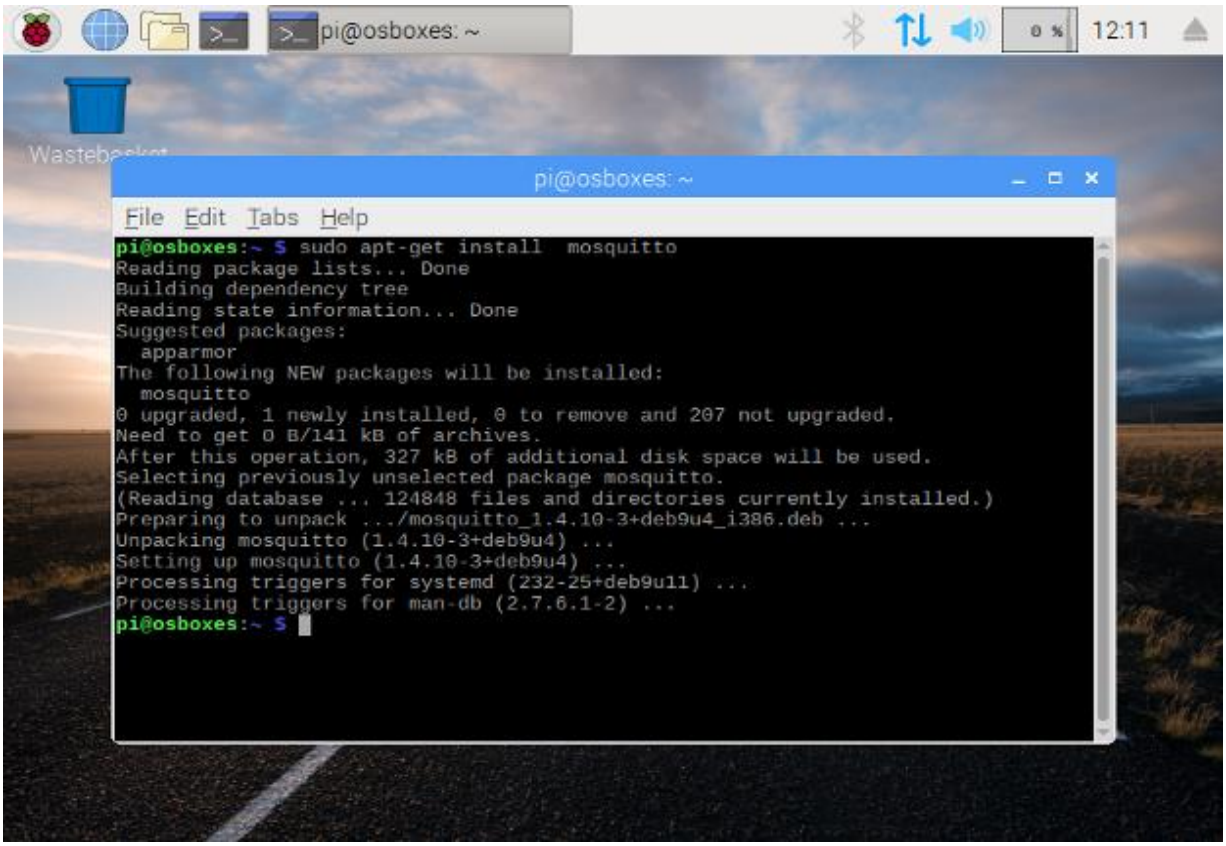
Open the command Terminal and run:

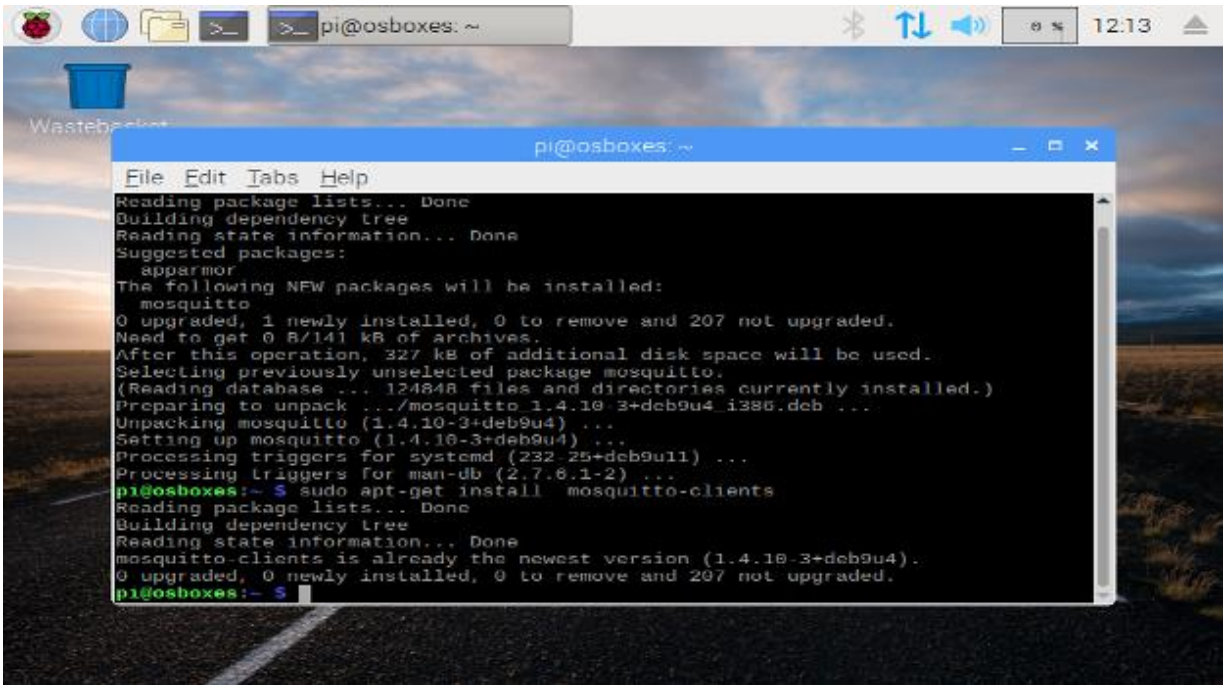
Step1.sudo apt-get update

Step2.sudo apt-get install mosquitto

Step3. sudo apt-get install mosquitto-clients


pi@osboxes: ~
11:58
pi@osboxes: ~
File Edit Tabs Help
pi@osboxes:~ $ apt-get update
Reading package lists... Done
W: chmod 0700 of directory /var/lib/apt/lists/partial failed - SetupAPTPartialDirectory (1: Operation not permitted)
E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)
E: Unable to lock directory /var/lib/apt/lists/
W: Problem unlinking the file /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permission denied)
W: Problem unlinking the file /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permission denied)
pi@osboxes:~ $ sudo apt-get update
Hit:1 http://archive.raspberrypi.org/debian stretch InRelease
Ign:2 http://ftp.debian.org/debian stretch InRelease
Get:3 http://ftp.debian.org/debian stretch-updates InRelease [91.0 kB]
Get:4 http://security.debian.org stretch/updates InRelease [94.3 kB]
Hit:5 http://ftp.debian.org/debian stretch Release
Fetched 185 kB in 2s (79.5 kB/s)
Reading package lists... Done
pi@osboxes:~ $

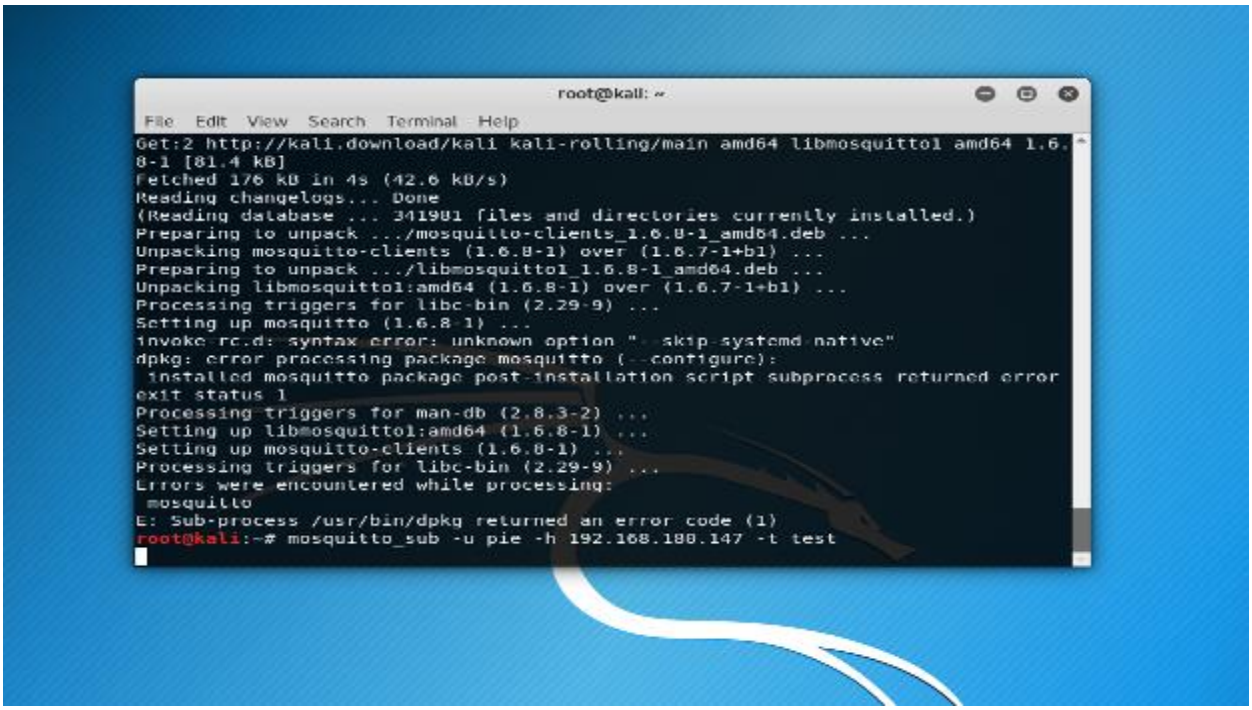

pi@osboxes: ~
12:11
pi@osboxes: ~
File Edit Tabs Help
pi@osboxes:~ $ sudo apt-get install mosquitto
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
apparmor
The following NEW packages will be installed:
mosquitto
0 upgraded, 1 newly installed, 0 to remove and 207 not upgraded.
Need to get 0 B/141 kB of archives.
After this operation, 327 kB of additional disk space will be used.
Selecting previously unselected package mosquitto.
(Reading database ... 124848 files and directories currently installed.)
Preparing to unpack .../mosquitto_1.4.10-3+deb9u4_i386.deb ...
Unpacking mosquitto (1.4.10-3+deb9u4) ...
Setting up mosquitto (1.4.10-3+deb9u4) ...
Processing triggers for systemd (232-25+deb9u11) ...
Processing triggers for man-db (2.7.6.1-2) ...
pi@osboxes:~ $



#Above commands are to be run on both the OS before moving forward.#

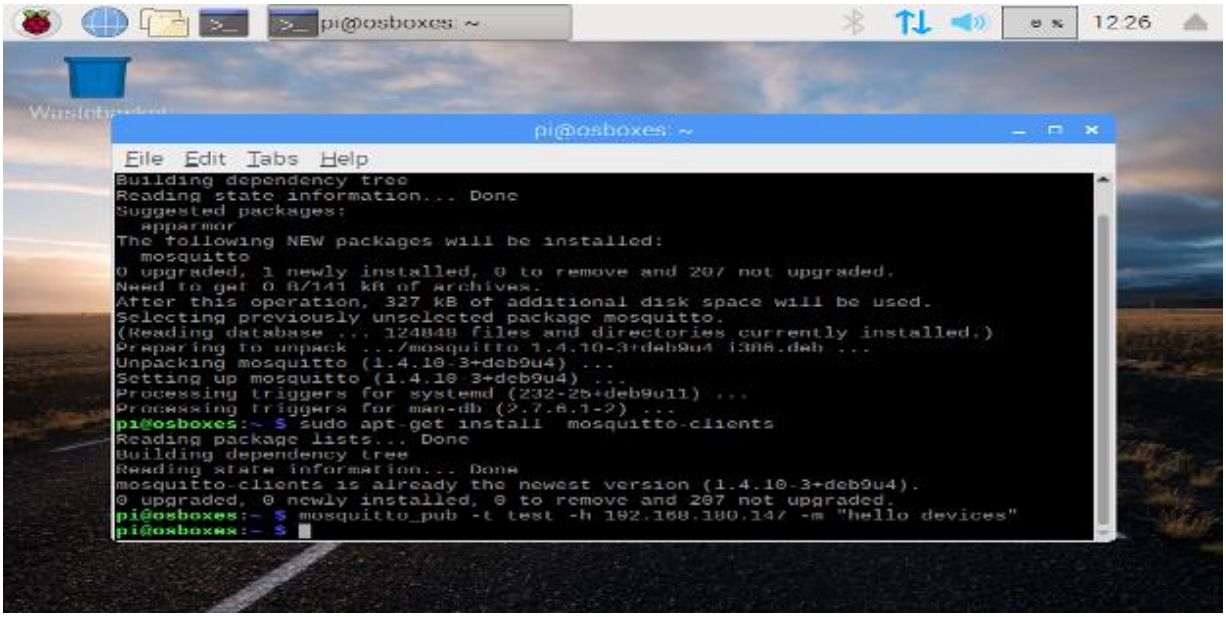
Now in kali terminal **(subscriber**) we are subscribing to a topic called test.

Step4- **mosquitto_sub –u pie –h 192.168.180.147 –t test**

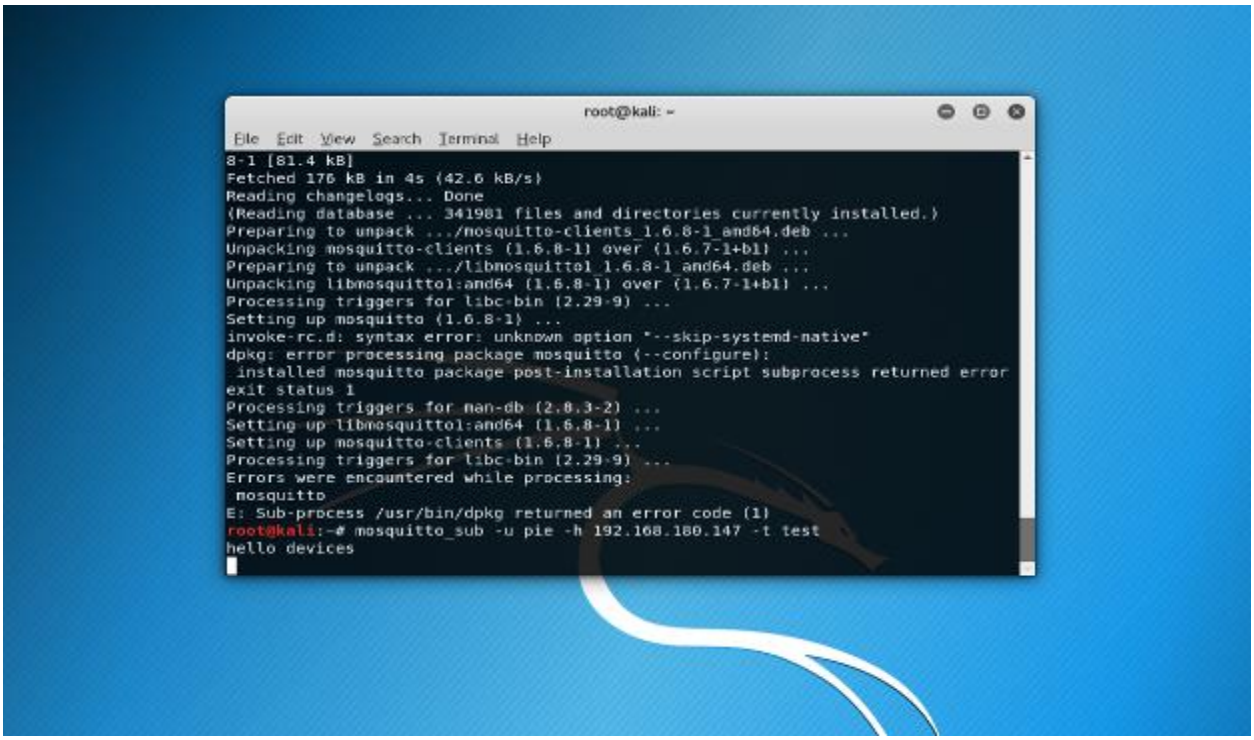


In Pie 0S terminal(**Broker**) we will try to publish some mesage in the topic test .

Step5- mosquitto_pub –t test –h 192.168.180.174 –m “hello devices”



Here in the kali terminal (subscriber) we can see the delivered message.



References

[1] http://www.scalagent.com/IMG/pdf/Benchmark_MQTT_servers-v1-1.pdf