Contents

[1. Configure WiFi AP on Raspberry Pi 3B 1](#_Toc496596953)

[1.1 Install Raspbian Jessie latest version 1](#_Toc496596954)

[1.2 Enable ssh on Raspberry Pi 1](#_Toc496596955)

[1.3 Connect to RPI using putty 1](#_Toc496596956)

[1.4 Install required packages on Raspbian 3](#_Toc496596957)

[1) Install hostapd and isc-dhcp-server packages. 3](#_Toc496596958)

[2) Set up DHCP server 4](#_Toc496596959)

[3) Set up wlan0 for static IP 9](#_Toc496596960)

[4) Configure Access Point 11](#_Toc496596961)

[5) Configure Network Address Translation 14](#_Toc496596962)

[2. Install and Configure Mosquitto Mqtt Broker on RPI 3B 16](#_Toc496596963)

[2.1 Installation of mosquito mqtt broker 16](#_Toc496596964)

[2.2 Generating of certification files for TLS between RPI and ESP8266 17](#_Toc496596965)

[1) Certificates for mqtt server 17](#_Toc496596966)

[2) Generate client certificates 23](#_Toc496596967)

[2.3 Configure TLS on mosquitto mqtt broker 25](#_Toc496596968)

[3. Generate Der format certificate for ESP8266(node MCU) 27](#_Toc496596969)

[4. Generating certificate files for AWS IOT client on Raspberry Pi 29](#_Toc496596970)

[4.1 Get connection package from WAS IOT 29](#_Toc496596971)

[4.2 Install aws iot python sdk and getting server root file 32](#_Toc496596972)

[5. Configure DynamoDB and make db tables 33](#_Toc496596973)

[6. Install packages for RPI 33](#_Toc496596974)

[7. Deploying gateway program on RPI 34](#_Toc496596975)

[8. Deploy program on iot devices. 35](#_Toc496596976)

# Configure WiFi AP on Raspberry Pi 3B

## 1.1 Install Raspbian Jessie latest version

* Download Raspbian jessie image from raspberrypi.org.

<https://downloads.raspberrypi.org/raspbian/images/raspbian-2017-07-05/2017-07-05-raspbian-jessie.zip>

* Extract image from zip file and install sd card of RPI.

There are several methods online. I have been using etcher or Win32DiskImager to install Raspbian on SD card.

Notes: Nowadays, Raspbian stretch is released, but it has still some issues in sub processing. I have confirmed several times in this testing, so we can use jessie until they fix them on stretch version.

## 1.2 Enable ssh on Raspberry Pi

We have to enable SSH on Raspbian in order to install, configure, test software.

By default, the current version of Raspbian has SSH disabled.

Create an empty file in the boot directory called **ssh**

Make sure that the file has no file extension, i.e. remove the **.txt** if you created the file by selecting New > Text File.

enable-ssh-raspberry-pi

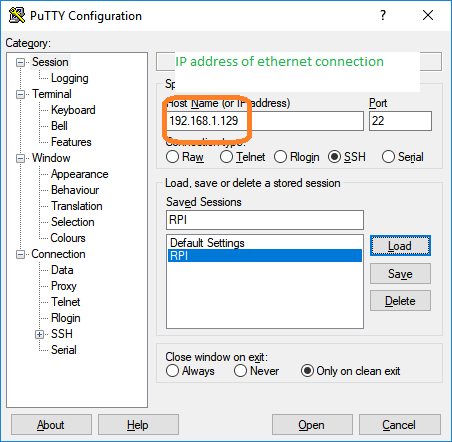
## 1.3 Connect to RPI using putty

I have been using putty on windows in order to connect and control pi on your desktop.

If you didn’t install putty on your desktop, download putty and install.

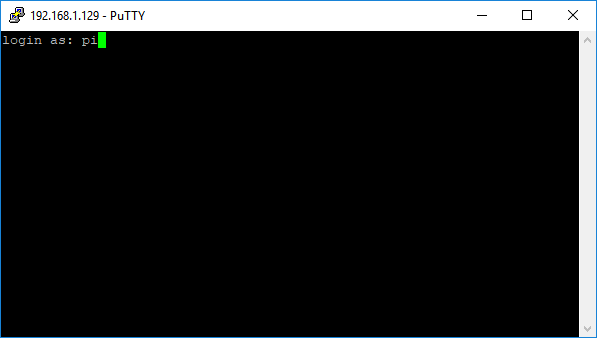
<https://the.earth.li/~sgtatham/putty/latest/w64/putty-64bit-0.70-installer.msi>

Open putty software.

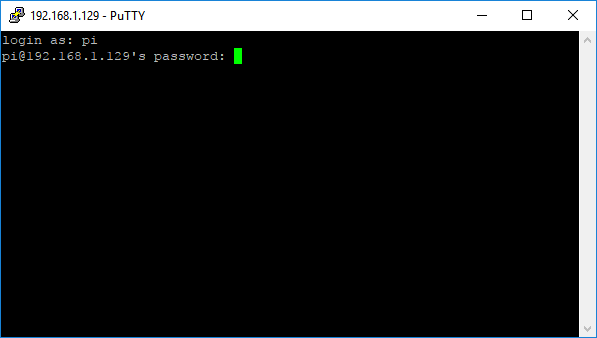


You have to know ip of ethernet connection of RPI. ( you can get ip easily on RPI desktop or several software such as Advanced Port Scanner.)

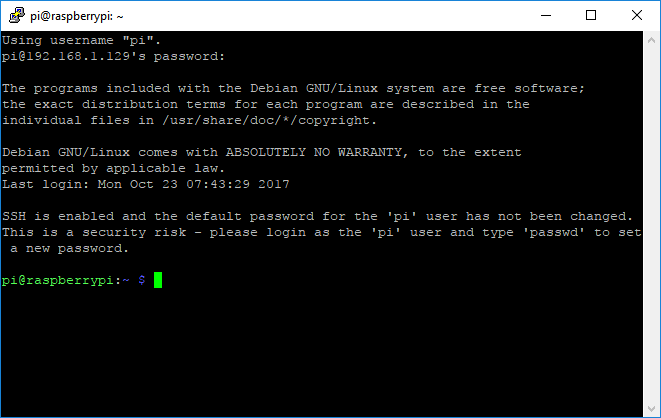
* Push open button, putty windows is showed.



* Input user name as “**pi**”, press enter.



* Default password is “**raspberry”.**



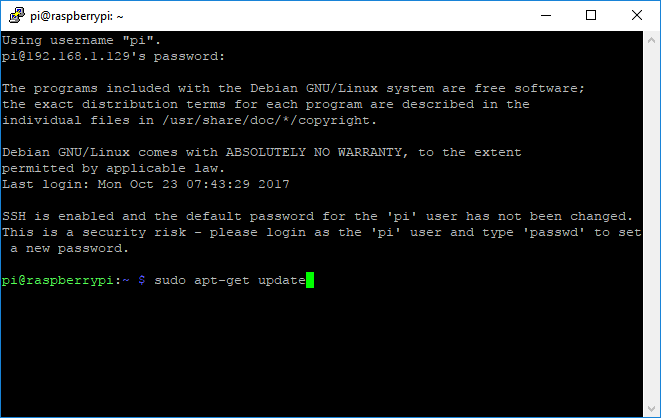
Connected status.

After that we can install packages via putty.

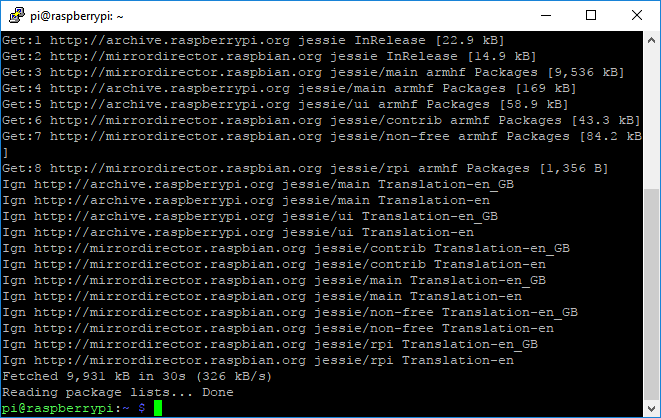
## Install required packages on Raspbian

### Install hostapd and isc-dhcp-server packages.

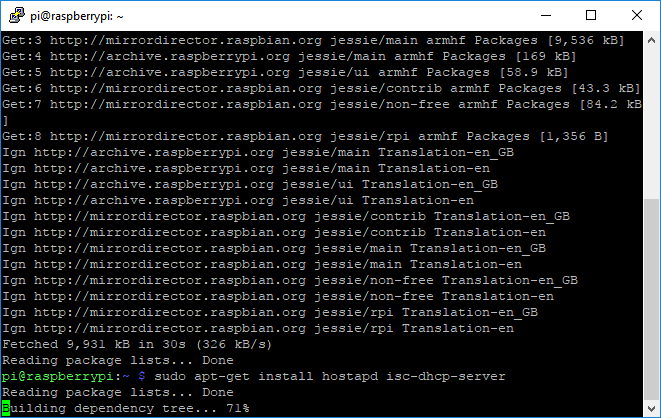
sudo apt-get update



Updating is finished,



sudo apt-get install hostapd isc-dhcp-server



### Set up DHCP server

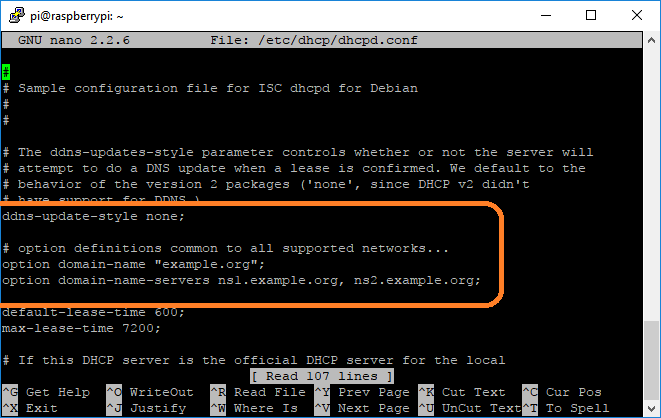
In putty window,

sudo nano /etc/dhcp/dhcpd.conf

Find the lines that say

option domain-name "example.org";

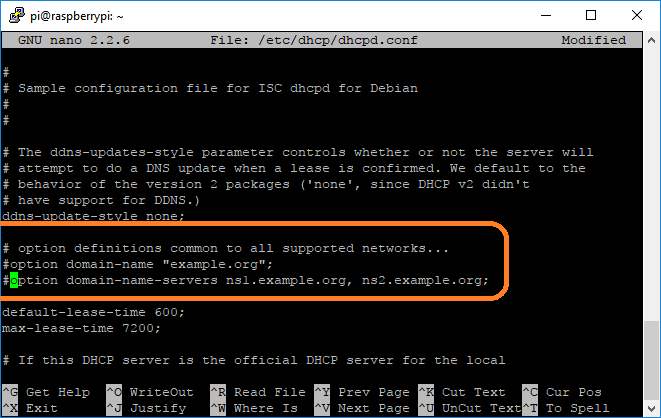
option domain-name-servers ns1.example.org, ns2.example.org;



Change them to add a # in the beginning so they say.

#option domain-name "example.org";

#option domain-name-servers ns1.example.org, ns2.example.org;

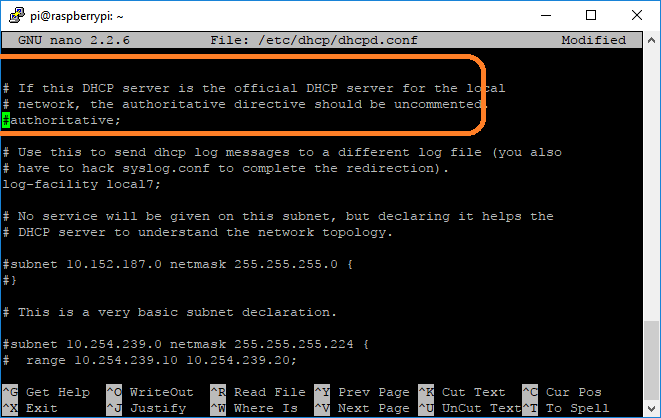


Find the lines that say

#If this DHCP server is the official DHCP server for the local

#network, the authoritative directive should be uncommented.

#authoritative;

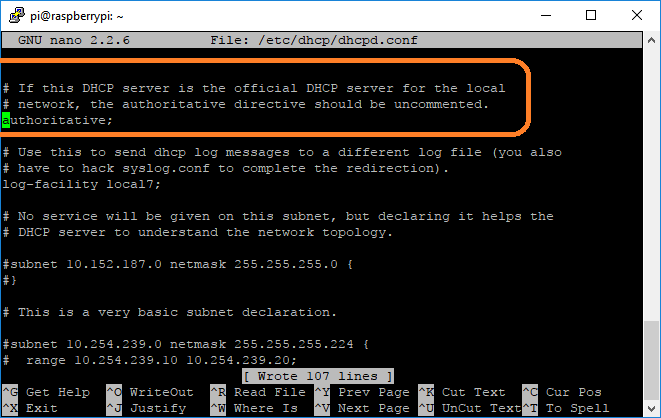


And remove the # so it says

# If this DHCP server is the official DHCP server for the local

# network, the authoritative directive should be uncommented.

authoritative;



Then scroll down to the bottom and add the following lines

subnet 192.168.42.0 netmask 255.255.255.0 {

range 192.168.42.10 192.168.42.50;

option broadcast-address 192.168.42.255;

option routers 192.168.42.1;

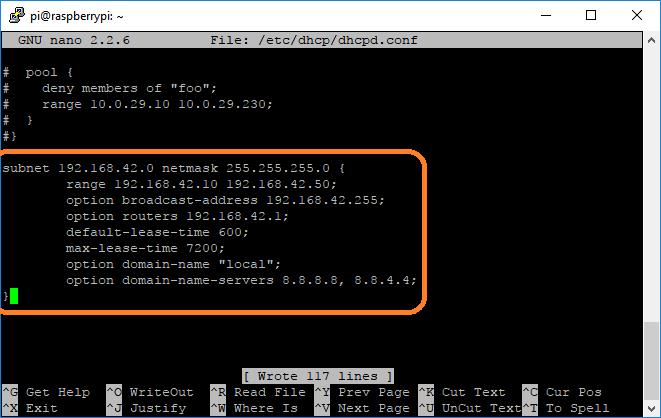
default-lease-time 600;

max-lease-time 7200;

option domain-name "local";

option domain-name-servers 8.8.8.8, 8.8.4.4;

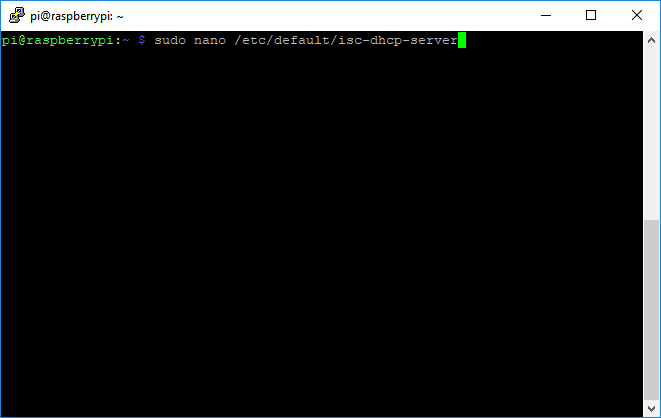
}



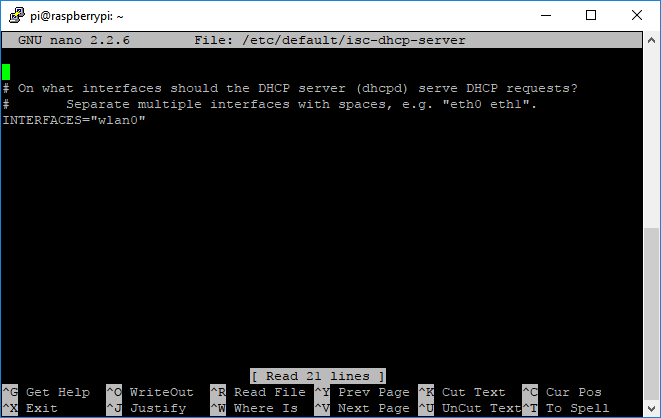
And then press CTRL+O to save config.

Press CTRL+X to exit from nano.

sudo nano /etc/default/isc-dhcp-server

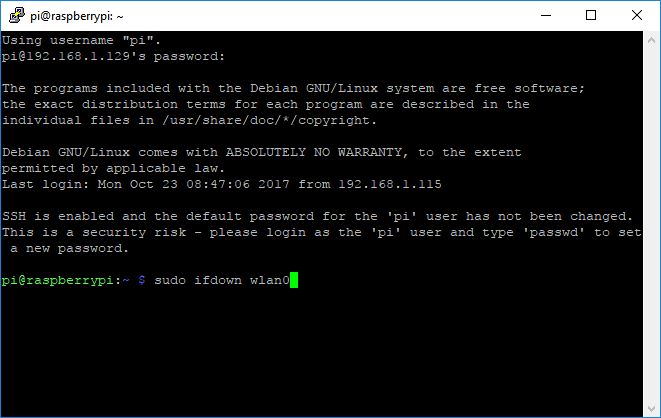


Scroll down to **INTERFACES=""**and update it to say **INTERFACES="wlan0"**

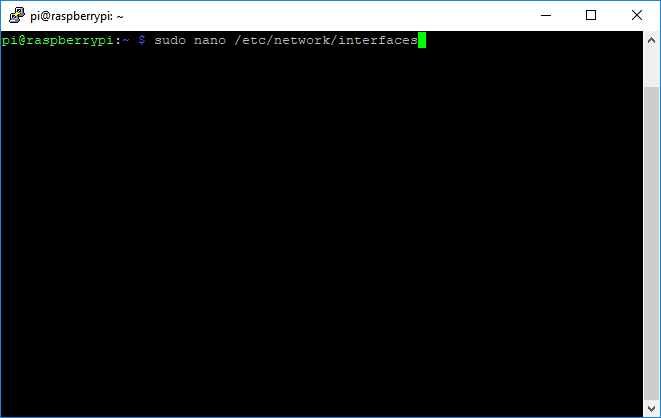


### Set up wlan0 for static IP

**sudo ifdown wlan0**



sudo nano /etc/network/interfaces

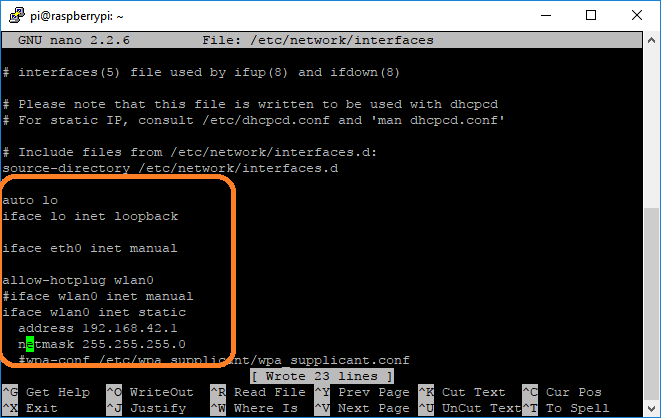


After **allow-hotplug wlan0** - see below for an example of what it should look like.  Any other lines afterwards should have a **#** in front to disable them

iface wlan0 inet static

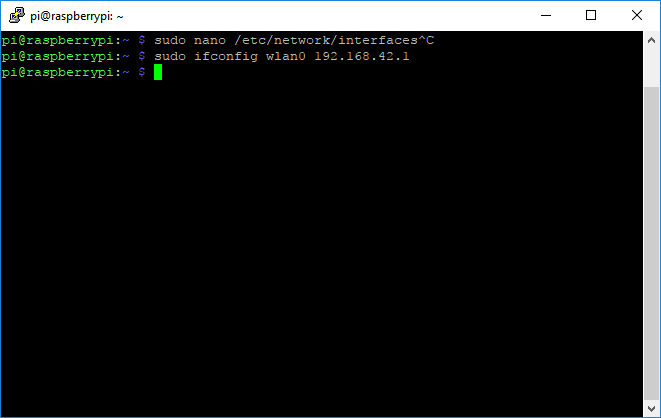
address 192.168.42.1

netmask 255.255.255.0



Save and close. (CTRL+O and CTRL+X)

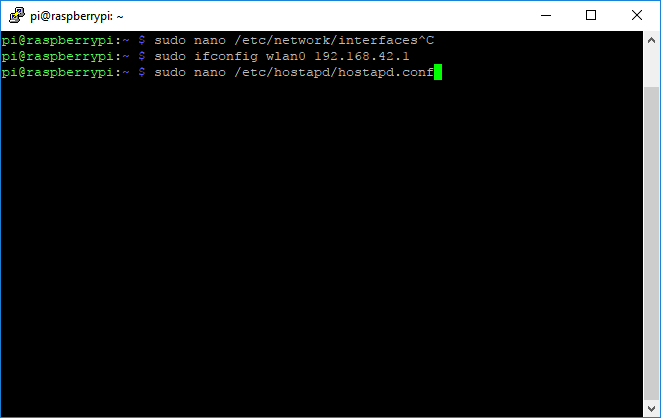
sudo ifconfig wlan0 192.168.42.1



### Configure Access Point

In putty,

sudo nano /etc/hostapd/hostapd.conf



Paste the following in, you can change the text after **ssid=** to another name, that will be the network broadcast name. The password can be changed with the text after **wpa\_passphrase=**

interface=wlan0

driver=nl80211

ssid=RPI\_AP\_001E

country\_code=US

hw\_mode=g

channel=6

macaddr\_acl=0

auth\_algs=1

ignore\_broadcast\_ssid=0

wpa=2

wpa\_passphrase=raspberry

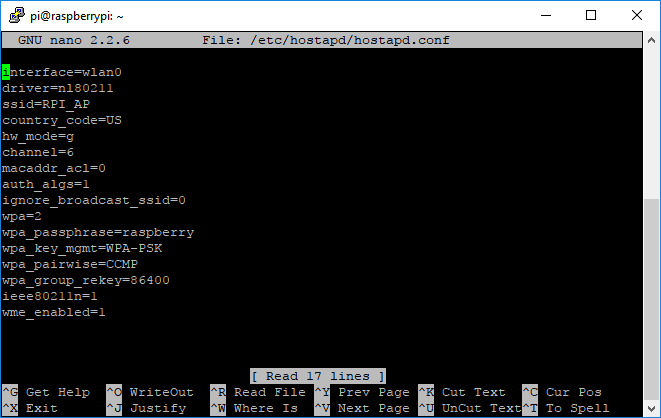
wpa\_key\_mgmt=WPA-PSK

wpa\_pairwise=CCMP

wpa\_group\_rekey=86400

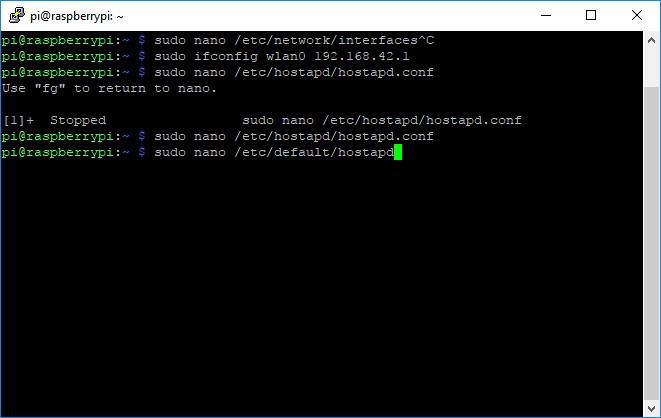
ieee80211n=1

wme\_enabled=1

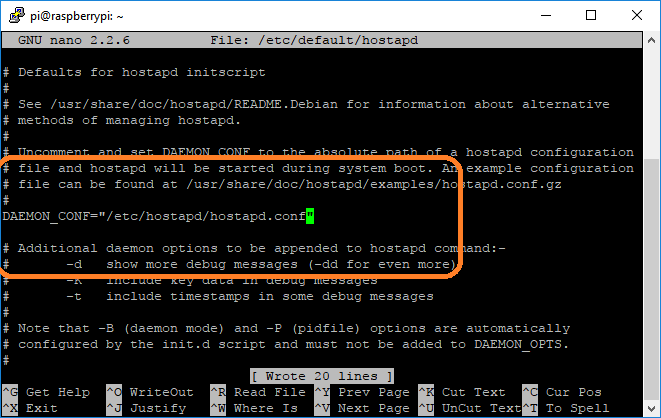


Save and exit.

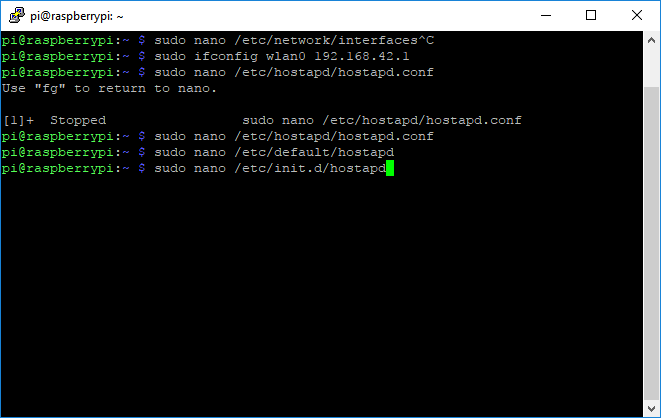
**sudo nano /etc/default/hostapd**



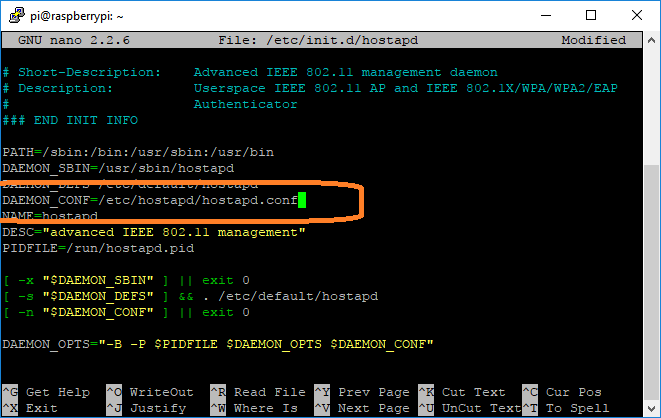
Find the line #DAEMON\_CONF="" and edit it so it says DAEMON\_CONF="/etc/hostapd/hostapd.conf".  
Don't forget to remove the # in front to activate it!

****

**sudo nano /etc/init.d/hostapd**

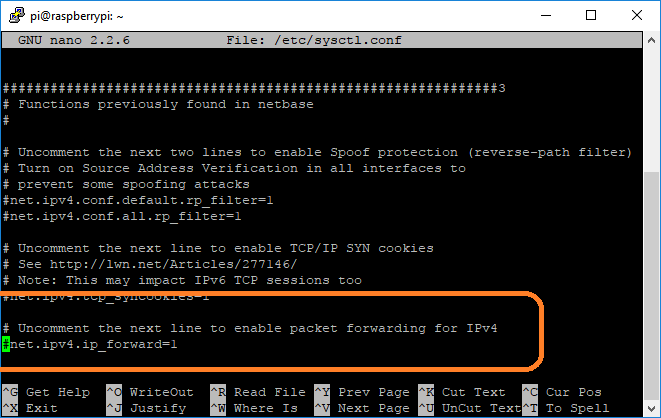


Find the line DAEMON\_CONF= and change it to DAEMON\_CONF=/etc/hostapd/hostapd.conf

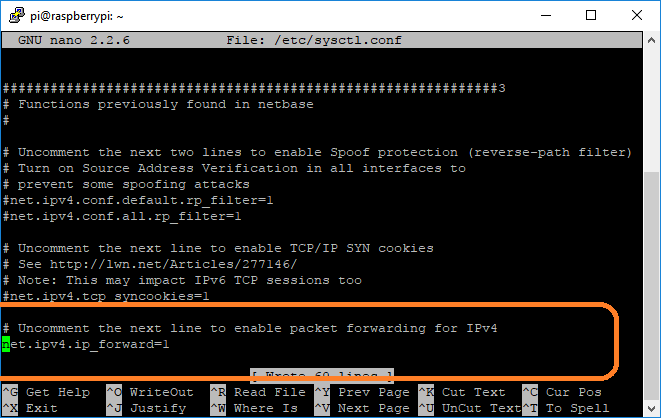


### Configure Network Address Translation

sudo nano /etc/sysctl.conf



Find that and remove #.



Save and exit

In putty, run following commands.

sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip\_forward"

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT

sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT

sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"

sudo nano /etc/rc.local and then just above the line exit 0, add the following line:

iptables-restore < /etc/iptables.ipv4.nat

sudo service hostapd start

sudo service dnsmasq start

and then reboot RPI using **sudo reboot.**

**So, RPI will be act as wifi ap (ssid is RPI\_AP and password is raspberry).**

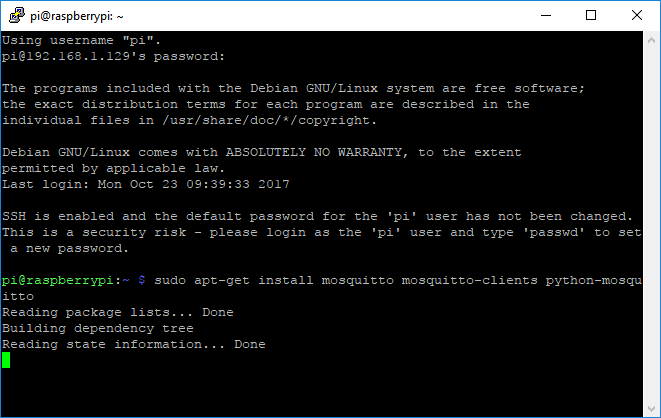
**You can confirm on your phone etc.**

# Install and Configure Mosquitto Mqtt Broker on RPI 3B

Open putty and connect to RPI via Ethernet IP. (same as 1.3)

## 2.1 Installation of mosquito mqtt broker

In putty, enter “sudo apt-get install mosquitto”.



Installation is done, mqtt broker will be performing on RPI.

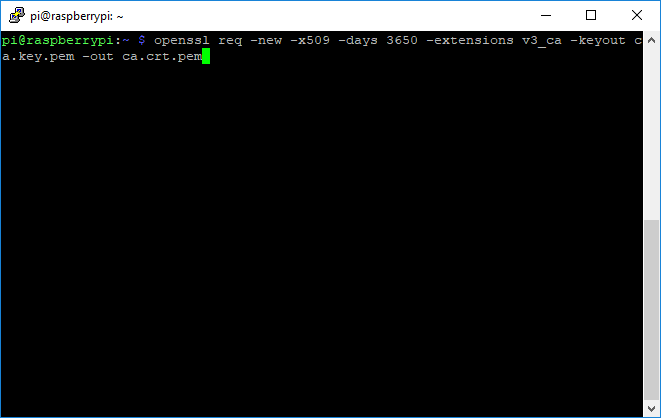
## 2.2 Generating of certification files for TLS between RPI and ESP8266

We have to generate certificate files using openssl on RPI.

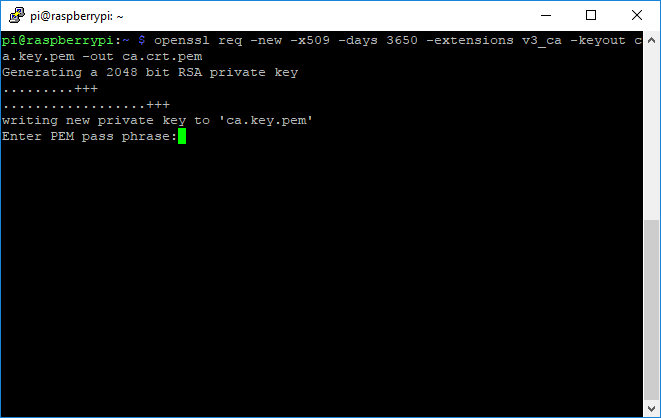
### Certificates for mqtt server

In putty, enter following commands

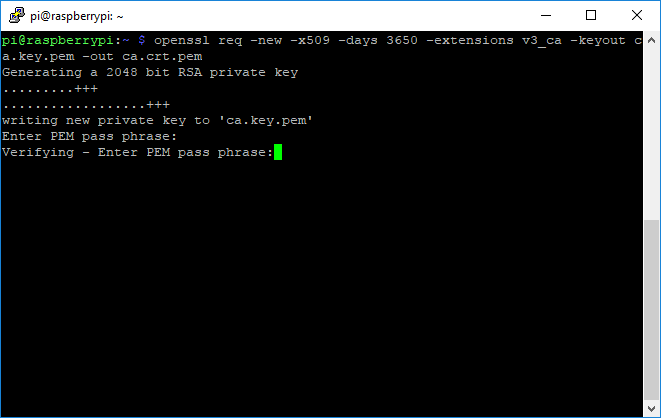
**sudo openssl req -new -x509 -days 3650 -extensions v3\_ca -keyout ca.key.pem -out ca.crt.pem**



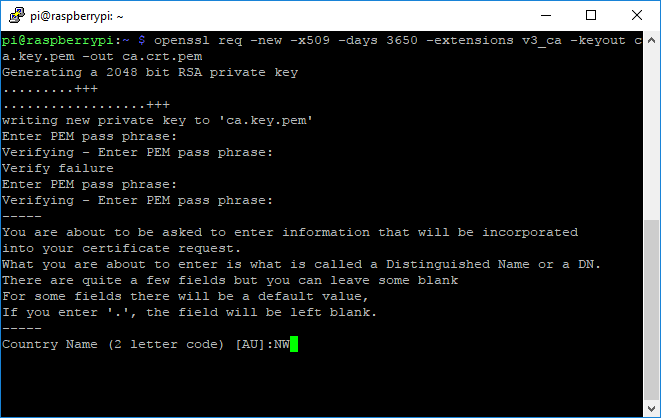
You have to optional enter pass phrase, ex: fradan



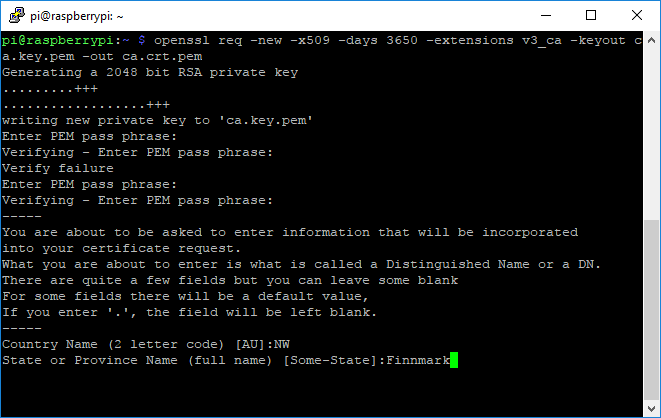
Confirm again



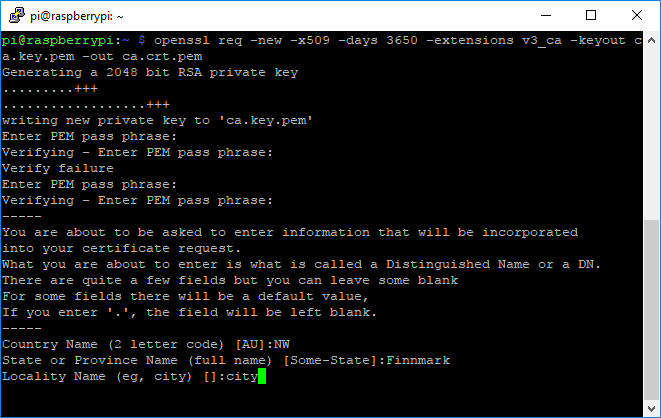
Enter country name(ex:NW)



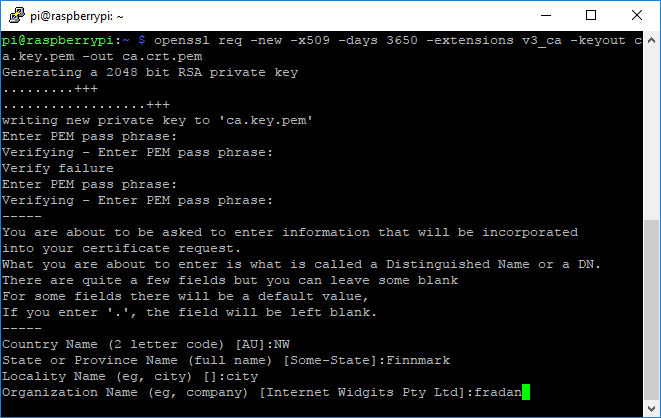
Enter province name (ex:Finnmark)



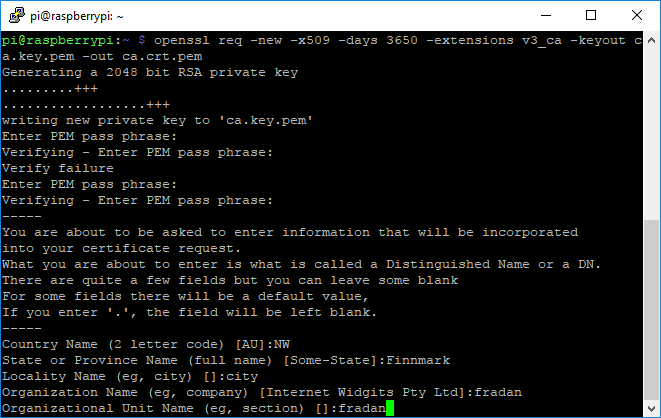
Enter city name(ex:city)



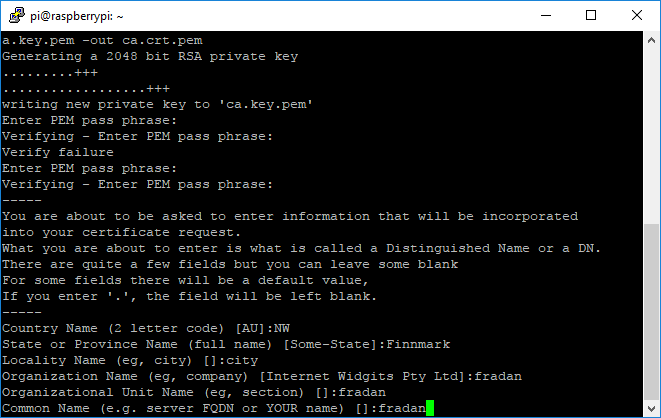
Input … (ex: fradan)



Input … (ex: fradan)



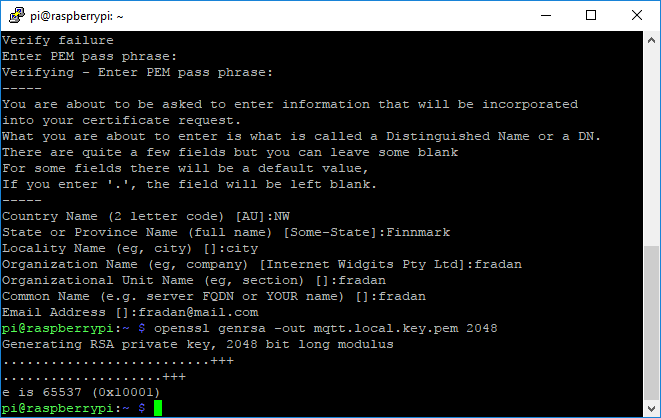
Input …



Input email (ex: [fradan@mail.com](mailto:fradan@mail.com))

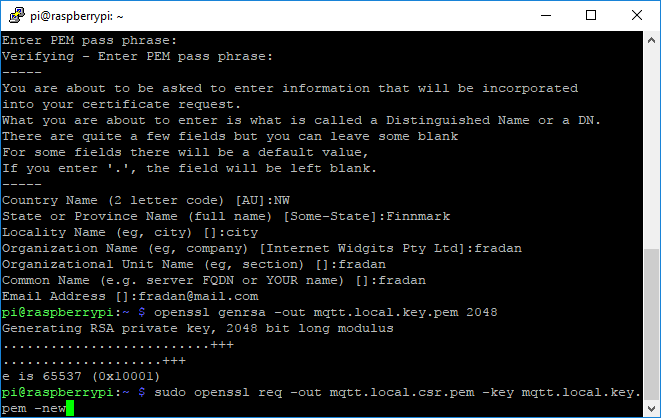
Input command as

**sudo openssl genrsa -out mqtt.local.key.pem 2048**



Input command as

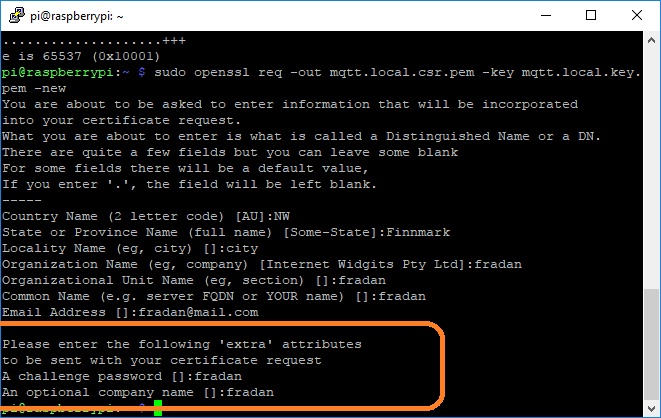
**sudo openssl req -out mqtt.local.csr.pem -key mqtt.local.key.pem -new**



Input same information as we have used above example.

…

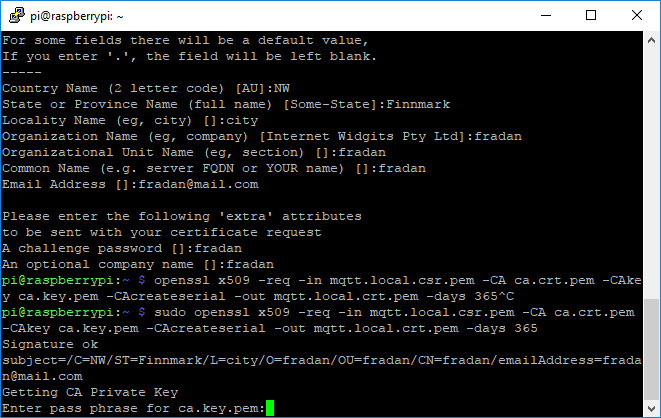
Input challenge password, also same as above. (fradan)



Input command as

**sudo openssl x509 -req -in mqtt.local.csr.pem -CA ca.crt.pem -CAkey ca.key.pem -CAcreateserial -out mqtt.local.crt.pem -days 365**

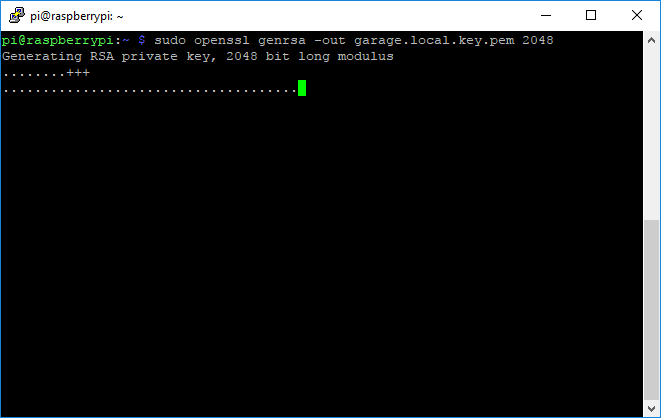
You have to input previous used password



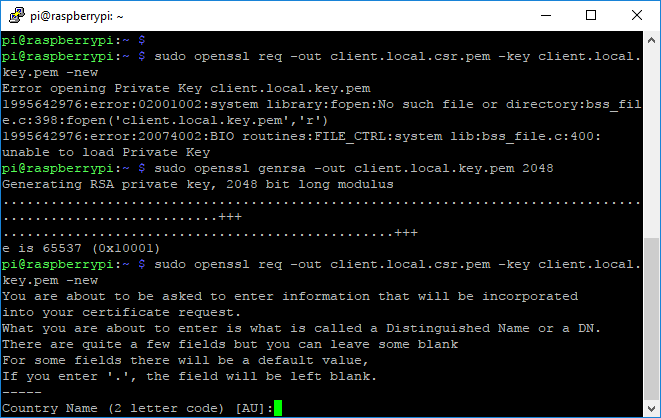
### Generate client certificates

Input following commands.

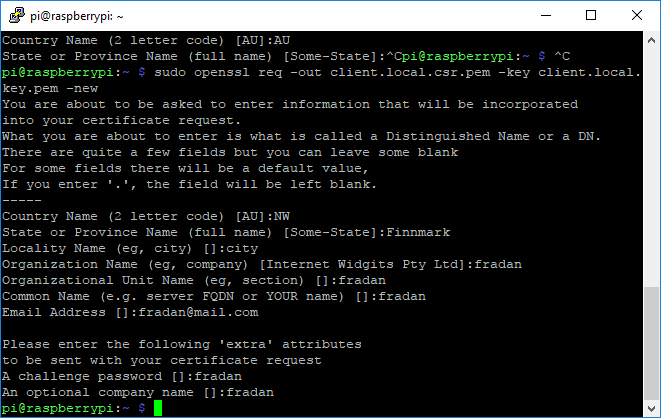
**sudo openssl genrsa -out client.local.key.pem 2048**



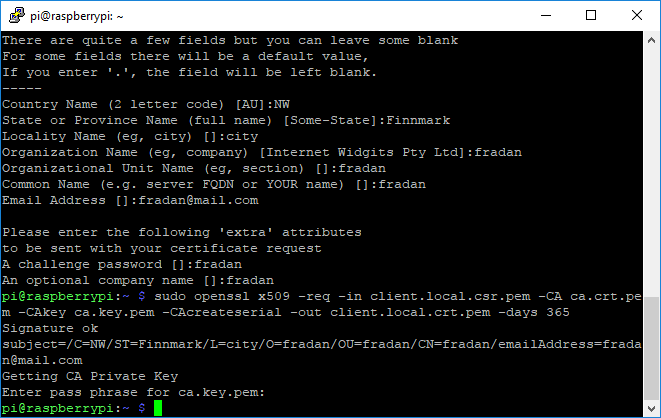
**sudo openssl req -out client.local.csr.pem -key client.local.key.pem -new**



Input same information as previous.



**sudo openssl x509 -req -in client.local.csr.pem -CA ca.crt.pem -CAkey ca.key.pem -CAcreateserial -out client.local.crt.pem -days 365**



Pass phrase is ‘fradan’ as we used.

So, in raspberry pi, /home/pi directory, certificate files are generated.

## Configure TLS on mosquitto mqtt broker

Copy server cert files into mosquito directory

In putty, enter following commands.

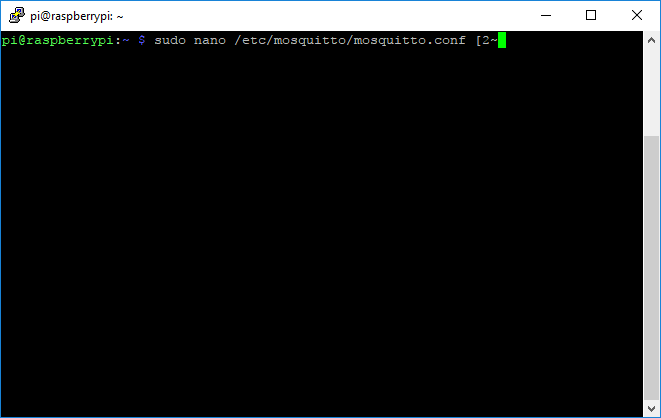
**sudo cp ca.crt.pem /etc/mosquitto/ca\_certificates**

**sudo cp mqtt.local.crt.pem /etc/mosquitto/certs**

**sudo cp mqtt.local.key.pem** **/etc/mosquitto/certs**

Edit config file.

**sudo nano /etc/mosquitto/mosquitto.conf**



Paste following commands at bottom of file.

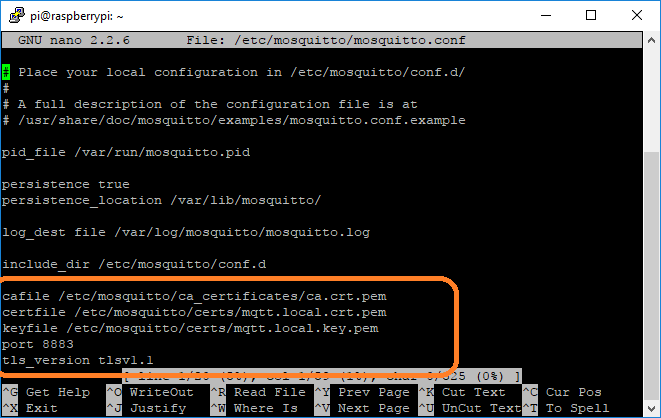
**cafile /etc/mosquitto/ca\_certificates/ca.crt.pem**

**certfile /etc/mosquitto/certs/mqtt.local.crt.pem**

**keyfile /etc/mosquitto/certs/mqtt.local.key.pem**

**port 8883**

**tls\_version tlsv1.1**

****

**Save and exit.(CTRL+O , CTRL+X)**

Reboot rpi(sudo reboot)

# Generate Der format certificate for ESP8266(node MCU)

On putty of RPI,

**sudo openssl x509 -in client.local.crt.pem -out cert.crt -outform DER**

**sudo openssl rsa** **-in client.local.key.pem -out private.crt -outform DER**

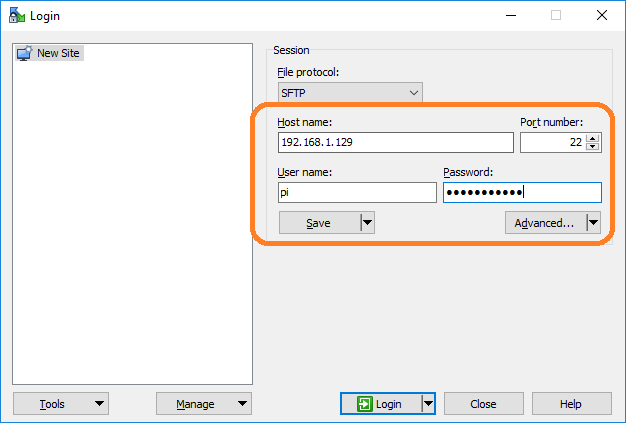
And then you have to copy these files from RPI.

I have been using WINSCP in order to copy and paste into RPI.

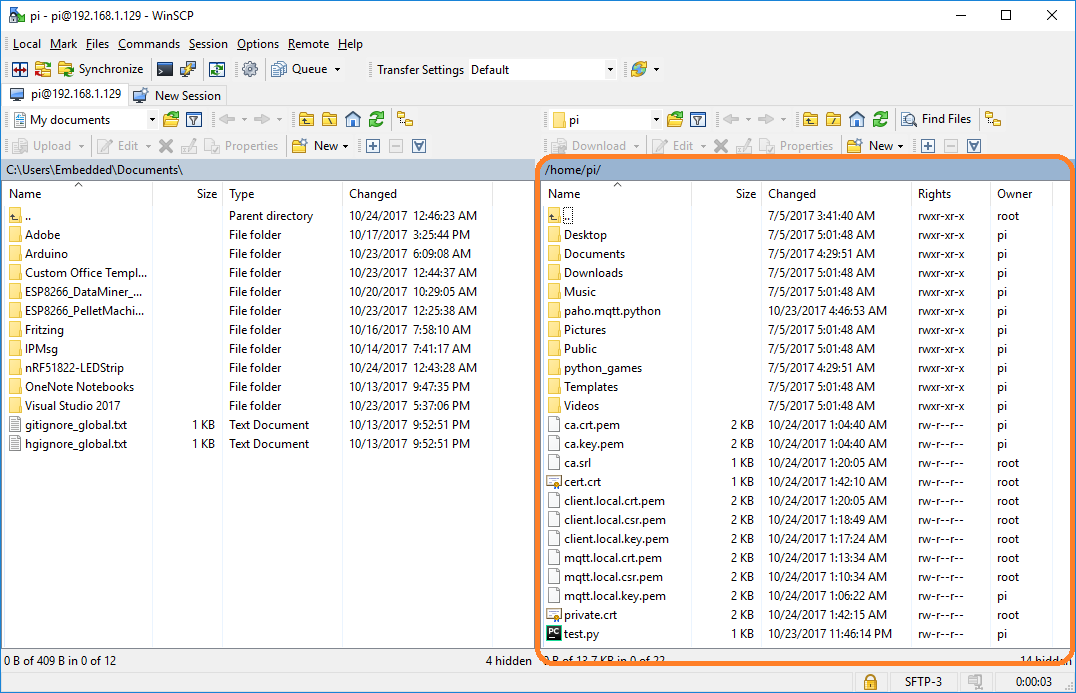
You can download here.

<https://winscp.net/download/WinSCP-5.11.2-Setup.exe>

Open software and input rpi information as figure.

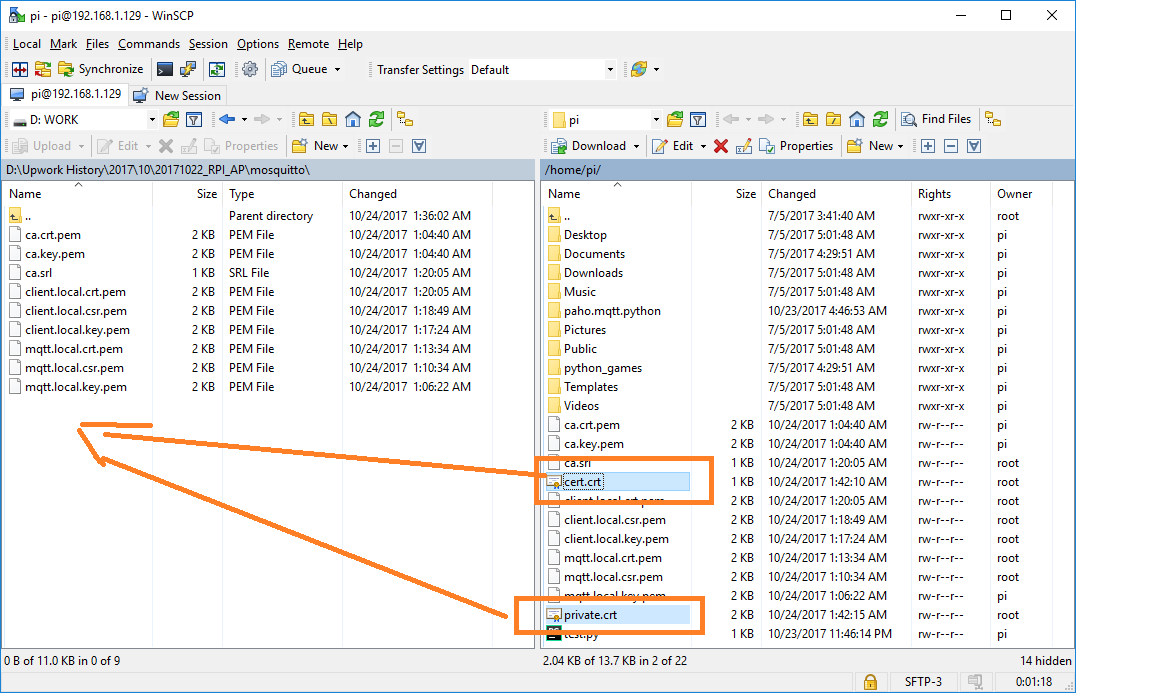


And then press login button.



You can see pi directory.

In this window, you can copy files easily from desktop.(drag and drop files)

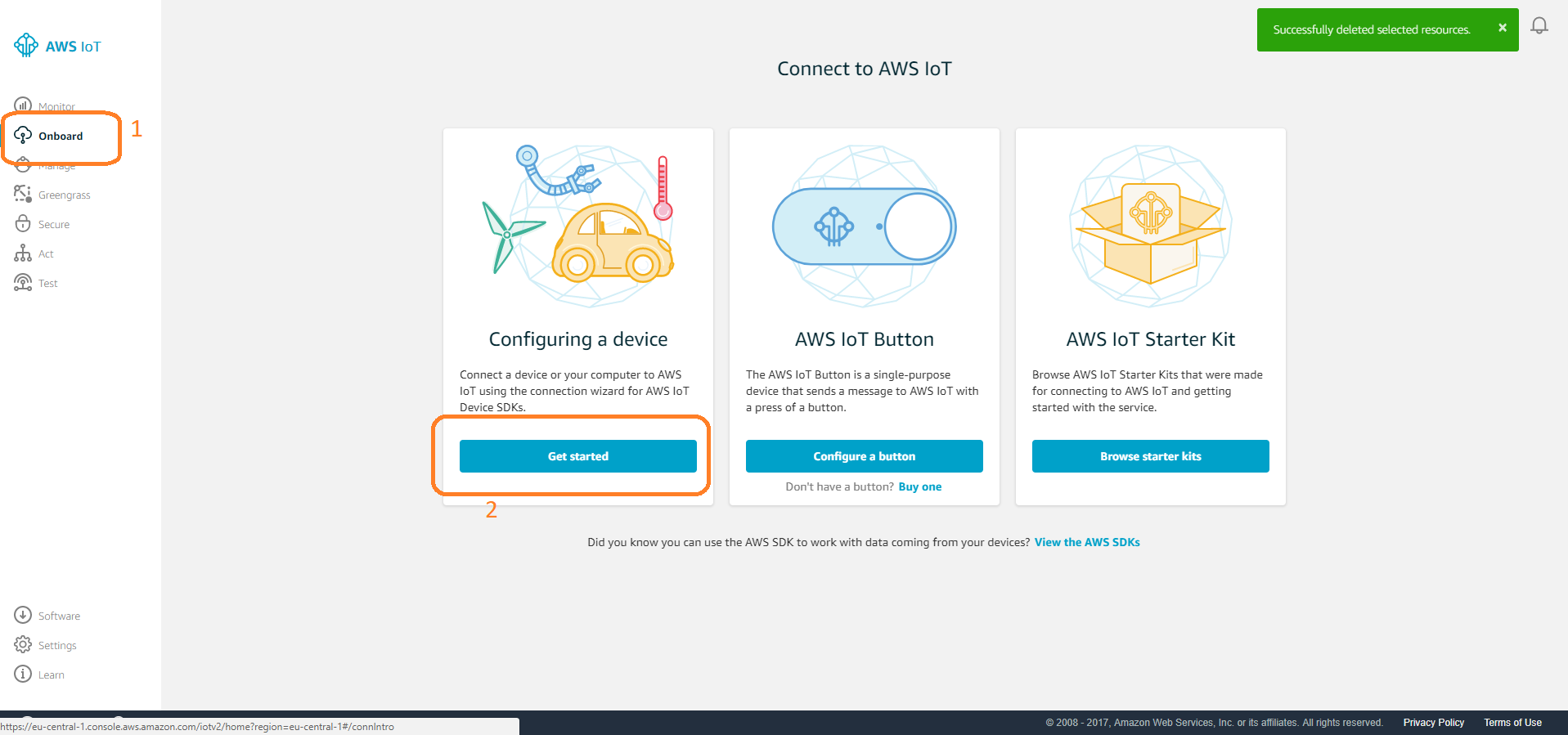


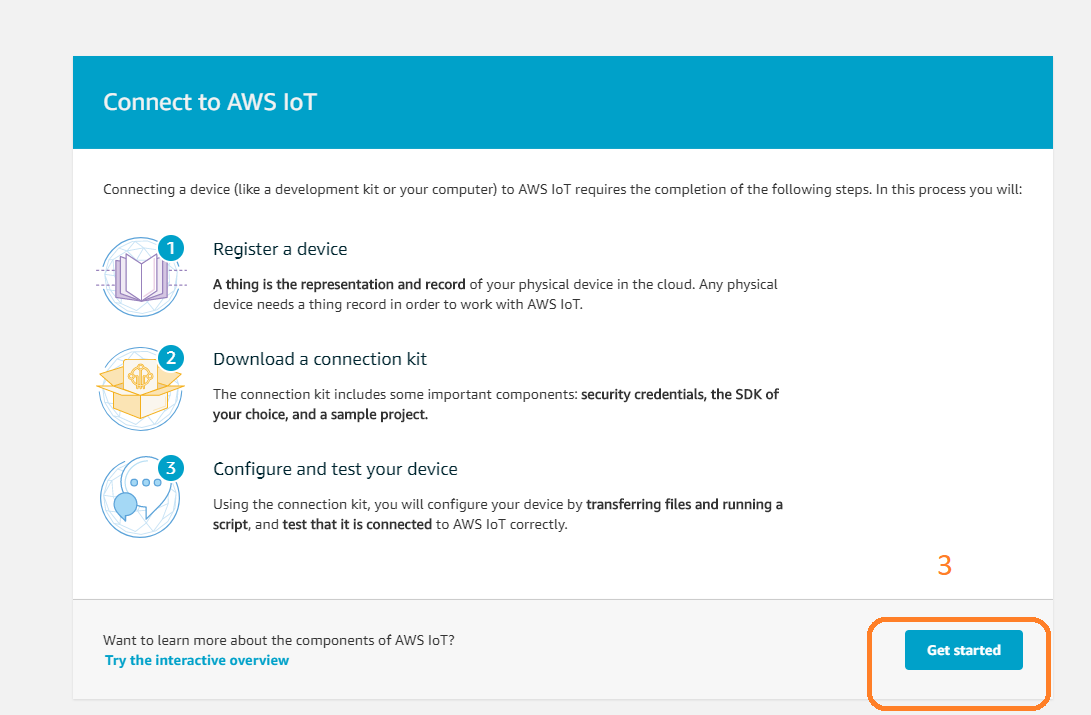
# Generating certificate files for AWS IOT client on Raspberry Pi

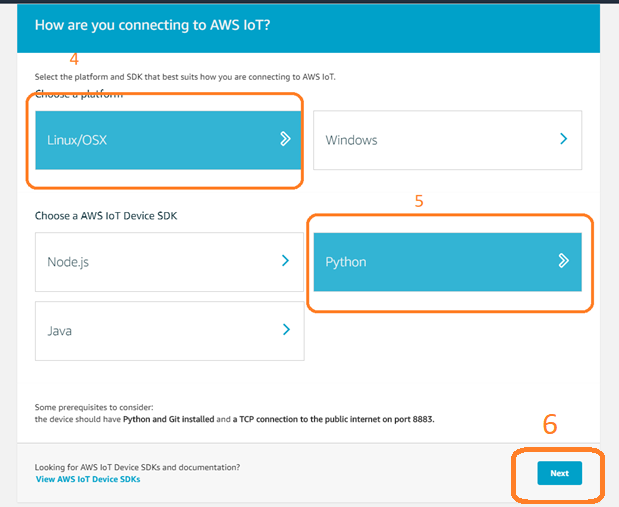
This is simplest way to get cert from AWS IOT.

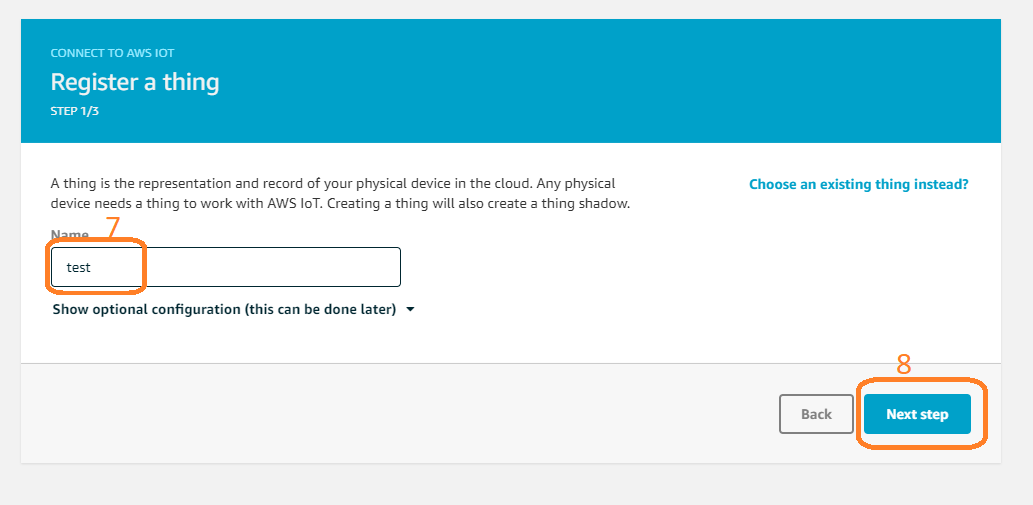
## 4.1 Get connection package from WAS IOT

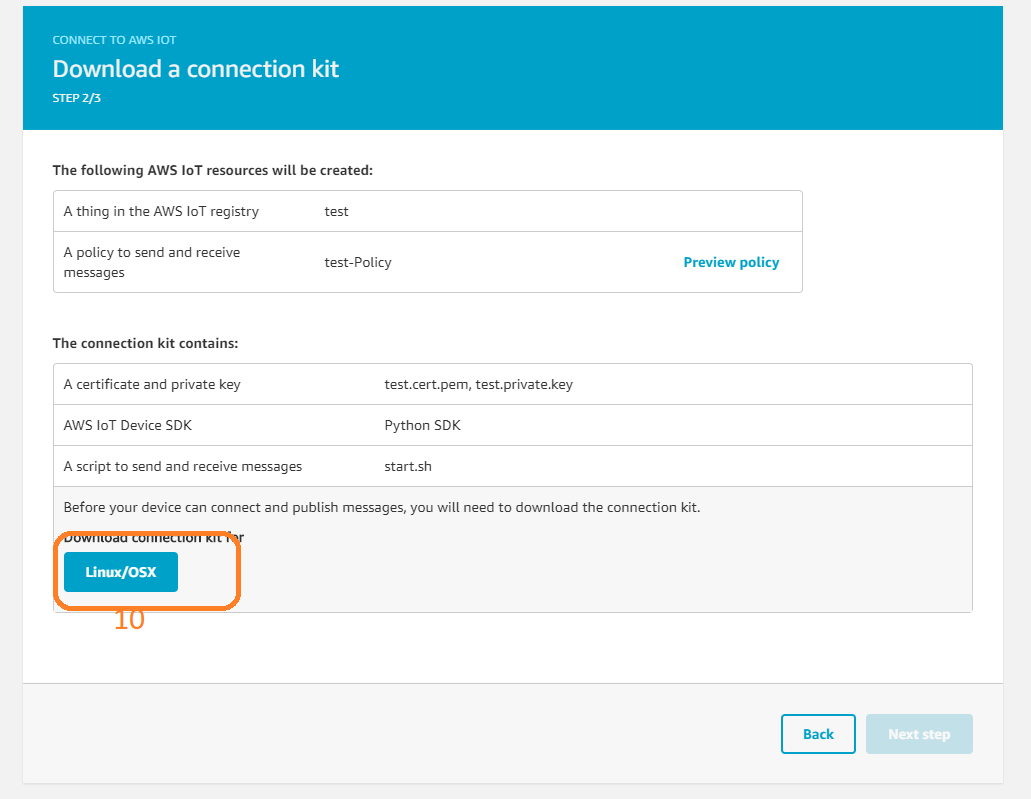
Login into your AWS console page. And then go to AWS IOT dashboard.







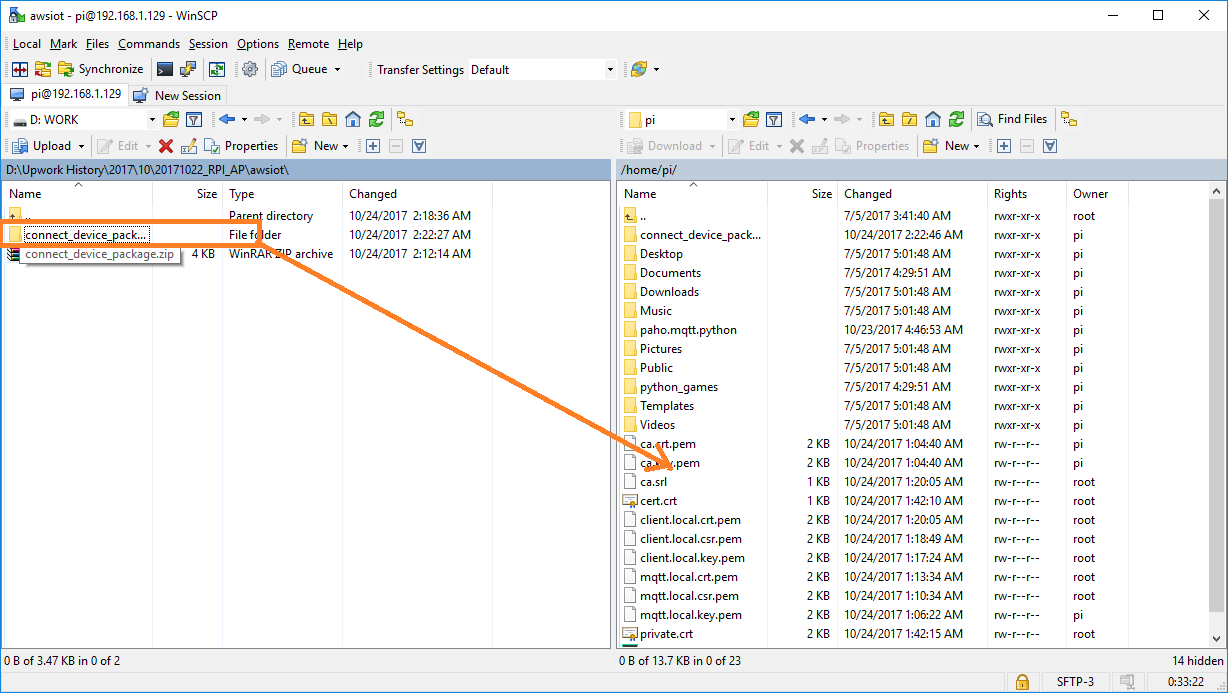




You can download archived connection package file.

## 4.2 Install aws iot python sdk and getting server root file

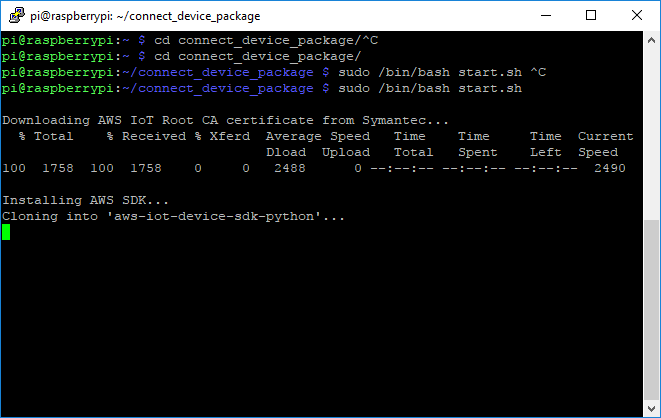
Extract downloaded file and copy it into /home/pi of RPI using WinSCP.



And then, in putty,

cd connect\_device\_package/

sudo /bin/bash start.sh



After 7~8 minutes, close putty window

# Configure DynamoDB and make db tables

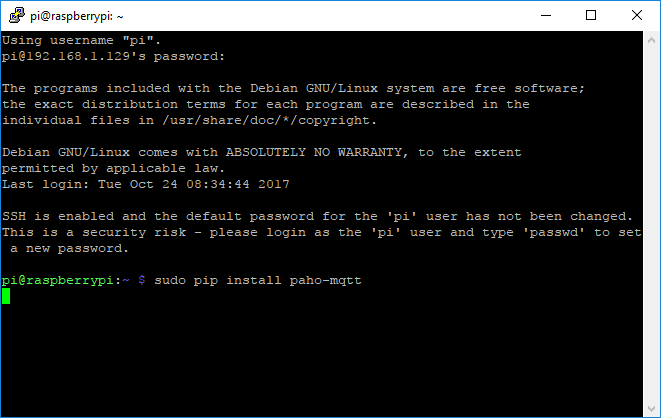
We can do that together easily via teamviewer.

# Install packages for RPI

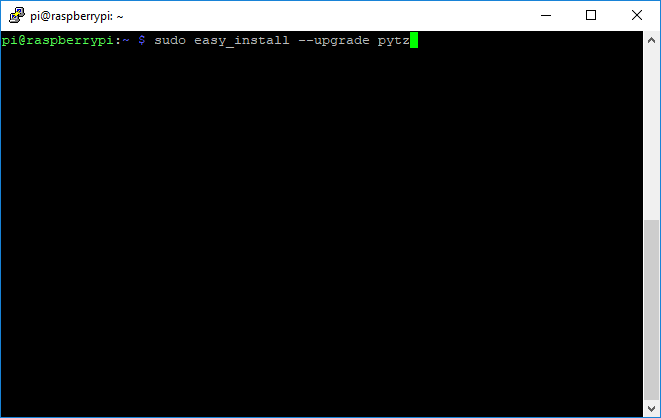
We will use following packages.

* python paho client (mosquitto mqtt client package)
* python pytz (utc and timezone package in db processing)

**sudo pip install paho-mqtt**



**sudo easy\_install --upgrade pytz**



# Deploying gateway program on RPI

In putty, input following commands

**sudo mkdir gateway**

**sudo chmod -R 777 gateway**

**cd gateway/**

**sudo mkdir awsiot**

**sudo cp /home/pi/connect\_device\_package/root-CA.crt /home/pi/gateway/awsiot/**

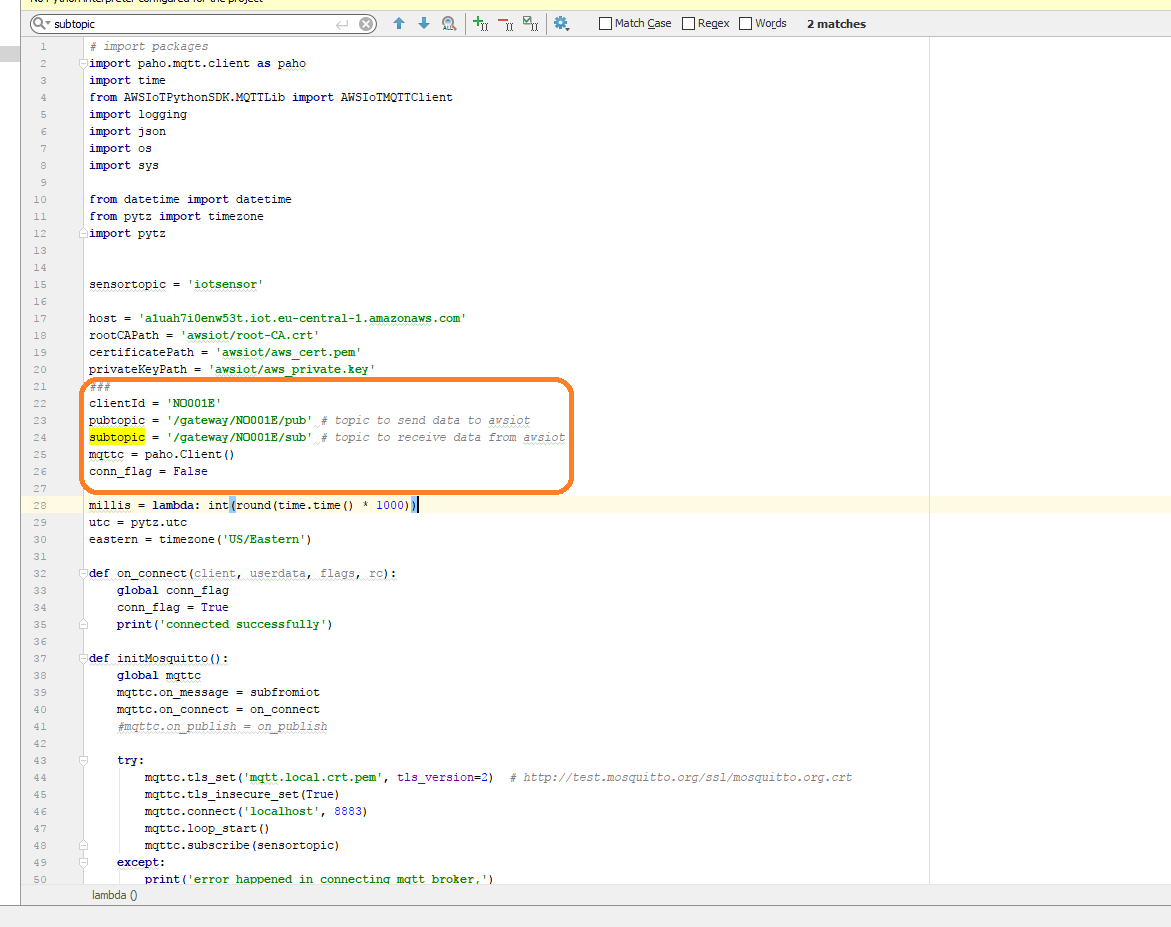
**sudo cp /home/pi/connect\_device\_package/test.cert.pem /home/pi/gateway/awsiot/aws\_cert.pem**

**sudo cp /home/pi/connect\_device\_package/test.private.key /home/pi/gateway/awsiot/aws\_private.key**

**sudo cp /home/pi/connect\_device\_package/root-CA.crt /home/pi/gateway/awsiot/**

**sudo cp /etc/mosquitto/certs/mqtt.local.crt.pem /home/pi/gateway**

Copy attached **gateway.py** file into /home/pi/gateway using WINSCP software.



You can change gateway id. (ex:NO001E)

# Deploy program on iot devices.

In this project, I have used Arduino IDE for ESP8266 node MCU.

Following additional libraries should be installed.

**pubsubclient – mqtt client for ESP8266**

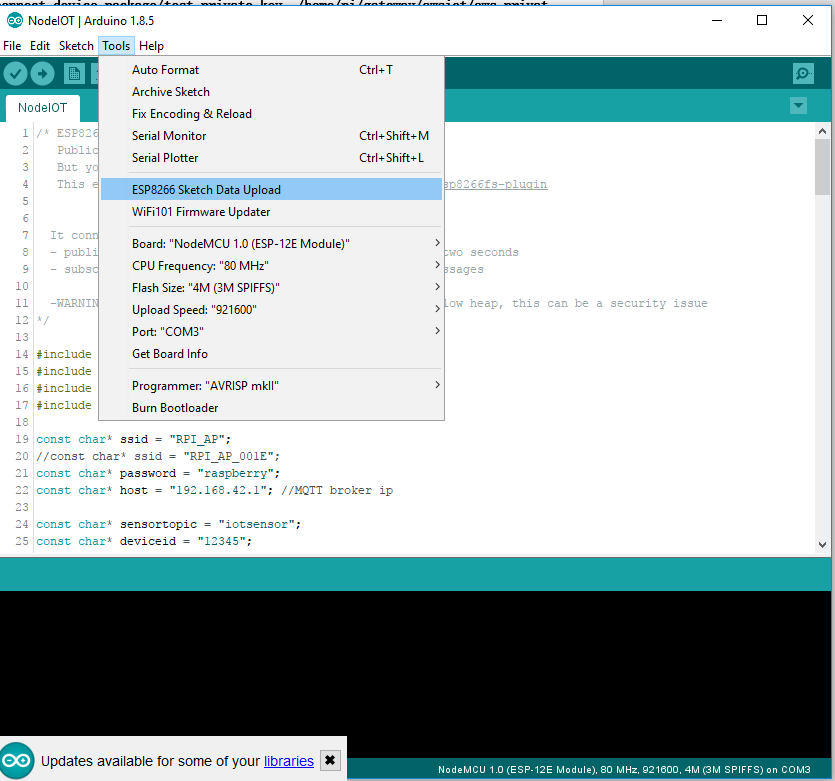
**ArduinoJson – Arduino Json for ESP8266**

And we have to upload cert files for mosquitto on SPIFFS on EPS8266.

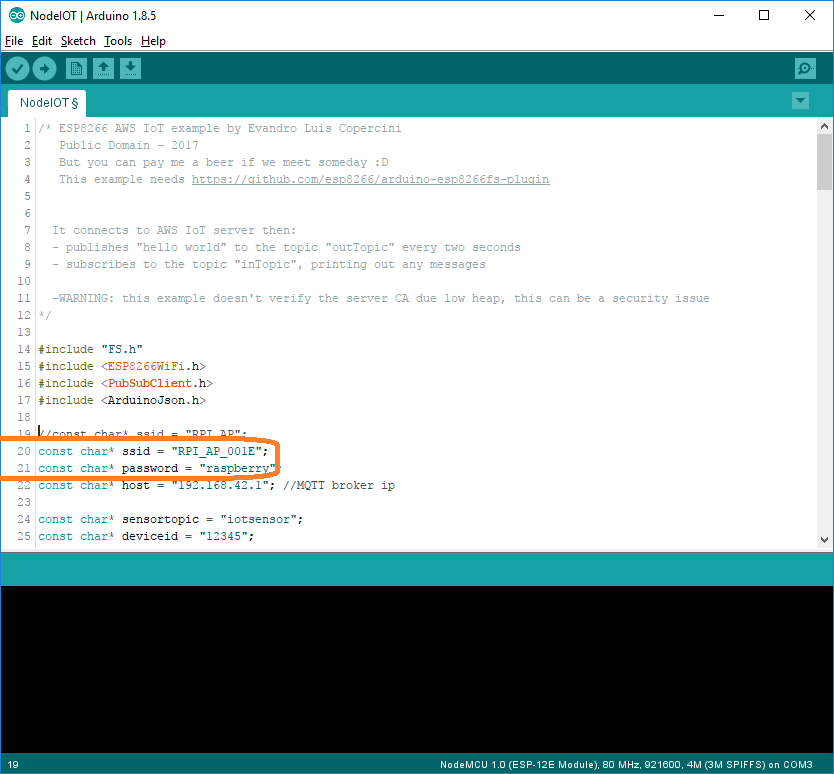
As you know, we have generated cert files for client in step 3.

We have to copy these files(**cert.crt and private.crt)** intodata directory of nodeIOT Arduino sketch directory.

Upload data using Arduino Plugin into SPIFFS on ESP8266.

****

Modify ssid name of RPI wifi AP in sketch source.



Upload sketch.