**IOT INTRODUCTION**

IOT means Internet of Things, the main aim of our project AWS Cloud and Network Security is to retrieve the data of cloud from anywhere at any time, if we have Internet Connection. In this project we are using sensors for getting information about temperature, humidity, rainfall at a particular location.

In this process, we have used different types of sensors they are For temperature and humidity information we are using DHT11 sensor. This is connected with Nodemcu.

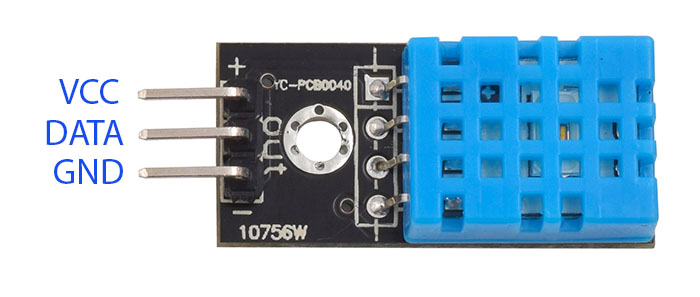
**6.1.1 DHT11 temperature and Humidity sensor**

DHT11 sensor is used for measuring both humidity and temperature values. It can measure relative humidity in percentage (20 to 90% RH) and temperature in degree Celsius in the range of 0 to 50°C.It has 3 pins. They are:

1. VCC

2. DATA OUT

3. GND



**Fig 6.2: DHT 11**

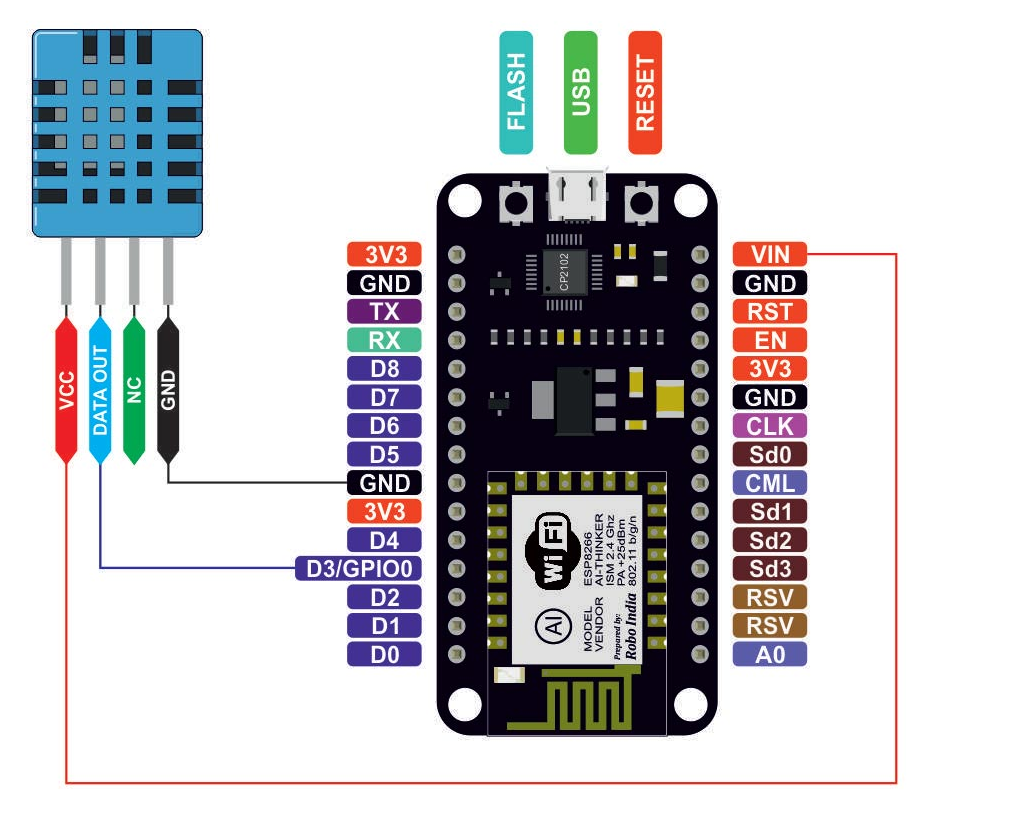
# Nodemcu:



**Fig 6.3: Node MCU**

NodeMCU Dev. Kit/board consist of ESP8266 Wi-Fi enabled chip. The **ESP8266** is a low-cost [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) chip developed by Espressif Systems with TCP/IP protocol.NodeMCU Dev Kit has **Arduino like** Analog (i.e. A0) and Digital (D0-D8) pins on its board. It supports serial communication protocols i.e. UART, SPI, I2C etc. Using such serial protocols we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards etc.

# 6.1.2 DHT11 SENSOR CONNECTION WITH NODEMCU:



**Fig 6.4: Pin Diagram of Node MCU**

**Connection of Nodemcu With Dht11 Sensor:**  
**Pin 1** of the DHT11 goes into **+3v** of the NodeMCU.

**Pin 2** of the DHT11 goes into Digital Pin **D3** of the NodeMCU.

**Pin 3** of the DHT11 goes into Ground Pin (**GND**) of the NodeMCU.

**6.1.3** **Rain Drop Sensor**

The rain sensor module is an easy tool for rain detection. It can be used as a switch when raindrop falls through the raining board and also for measuring rainfall intensity. The module features, a rain board and the control board that is separate for more convenience, power indicator LED and an adjustable sensitivity though a potentiometer

The analog output is used in detection of drops in the amount of rainfall. Connected to 5V power supply, the LED will turn on when induction board has no rain drop, and DO output is high. When dropping a little amount water, DO output is low, the switch indicator will turn on. Brush off the water droplets, and when restored to the initial state, outputs high level.

**Specifications**

• Adopts high quality of RF-04 double sided material.

• Area: 5cm x 4cm nickel plate on side.

• Anti-oxidation, anti-conductivity, with long use time.

• Comparator output signal clean waveform is good, driving ability, over 15mA.

• Potentiometer adjust the sensitivity.

• Working voltage 5V.

• Output format: Digital switching output (0 and 1) and analog voltage output AO.

• With bolt holes for easy installation.

• Small board PCB size: 3.2cm x 1.4cm.

• Uses a wide voltage LM393 comparator.

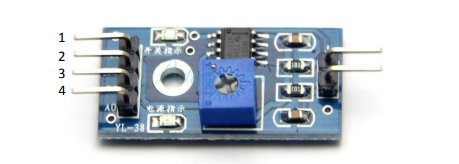
Pin Configuration

1. VCC: 5V DC

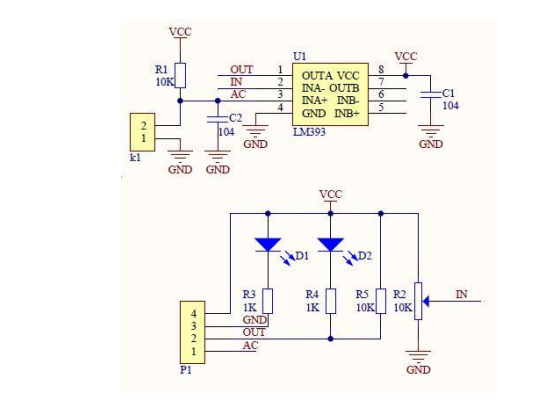
2. GND: ground

3. DO: high/low output

4. AO: analog output 1 2 3 4

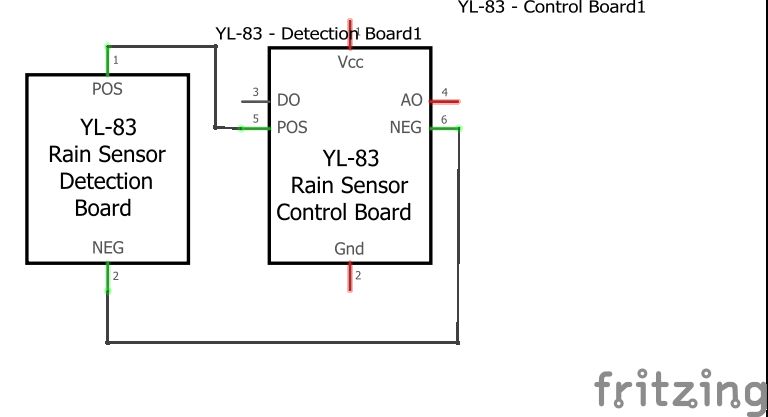


**Schematic Diagram**



**Fig 6.5:** **Schematic Diagram**

**Pin configuration of get spares rain drop sensor:**



**Fig 6.6:** **Pin Configuration of Raining Sensor**

**Connecting sensor on bread board:**

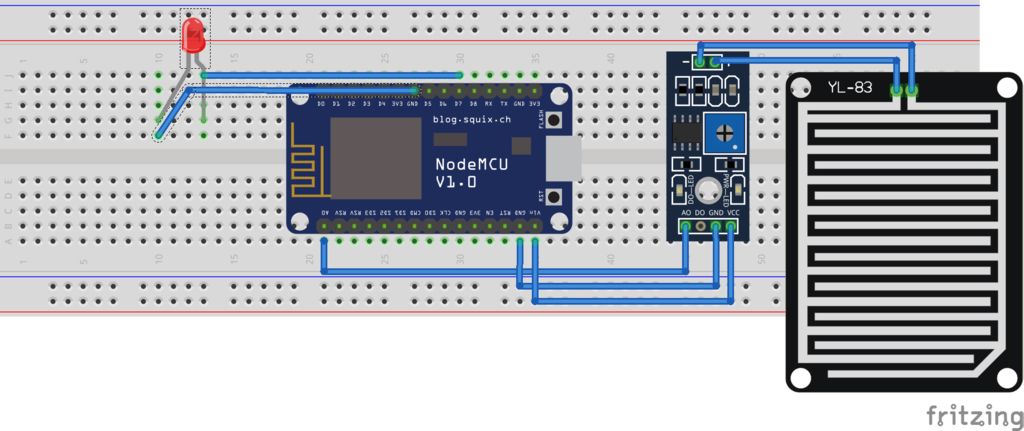


Fig 6.7: Connecting of Rain Sensor with Node MCU



[www.vaisala.com](http://www.vaisala.com/)

# Yl-83 Rain Detector

Heating element for keeping sensor free of snow and condensed moisture, and for quick drying

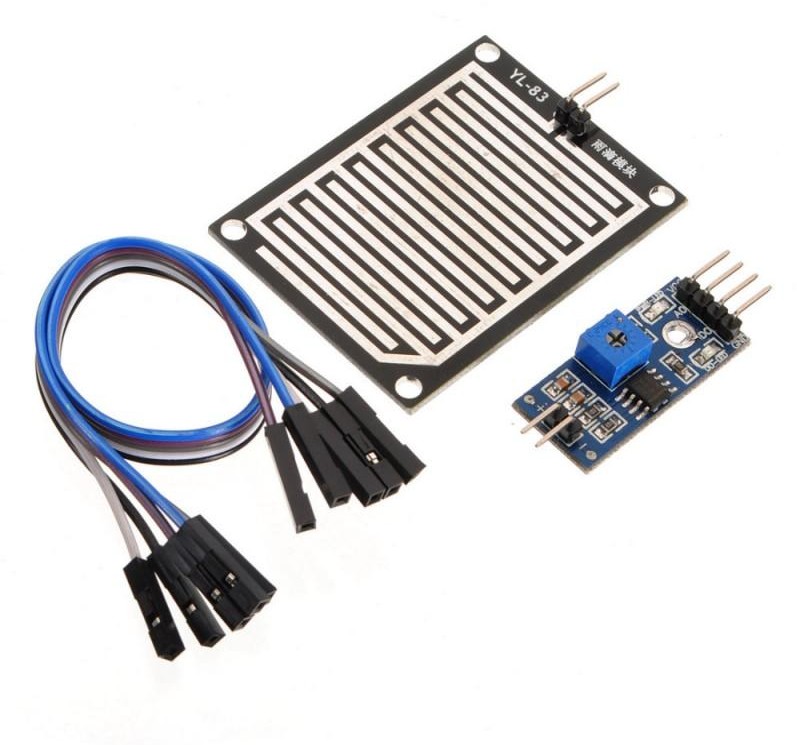
Maintenance free

**▪**

**▪**

* Fast and accurate precipitation detection (ON/OFF)
* Rain intensity measurement with processing unit

Features/Benefits



*Vaisala YL-83 Rain Detector*

Rain and snow are quickly and accurately detected with the YL-83 Rain Detector. The YL-83 operates via droplet detection rather than by signal level threshold.

A special delay circuitry allows about two-minute interval between raindrops before assuming an OFF (no rain) position. This enables the sensor to accurately distinguish between rain cessation and light rain.

The YL-83 also features an analog Rain Signal for estimating rain intensity. Since this signal is proportional to the percentage of moist or wet area on the sensor plate, rain intensity has a direct impact on the amplitude and variation of this analog signal.

The YL-83 sensor is positioned at a 30° angle. This design, together with the internal heating element, ensures that the surface dries quickly, an essential factor in calculating intensity. The same heating element also protects the surface from fog and condensed moisture, and is activated at

low temperatures in order to melt snow, thus allowing snow detection. Sensor performance is not affected by reasonable amounts of dirt and dust due to droplet detection.

It is intended to be used in areas

with only rain or wet/moist snow precipitation.



# Technical Data

Sensor

Capacitive principle, thick layer sensor

RainCap™ with a thin glass shield. Integrated heater element.

Output

Rain ON/OFF

Open collector, active low signal corresponds to rain Maximum voltage

Maximum current

15 V

50 mA

Sensitivity of Rain Detection

Minimum wet area 0.05 cm²

OFF-delay (active) < 5 min

Physical

Sensor plate

Analog output 1...3 V (wet...dry)

Frequency output 1500...6000 Hz,

non-calibrated

Input

Control to switch heater OFF

Sensing area Angle

7.2 cm²

30°

Open circuit input enables the heater. Connection to GND disables the heater.

Housing material Polypropylene

Windshield and support bracket Aluminum Moisture shield Polyurethane

Contact rating min. 15 V, 2 mA

Dimensions

With wind shield Without wind shield

(h × w × l) 110 × 80 × 175 mm

90 × 46 × 157 mm

Ground Wiring

Separate ground wires for signal and heater

Weight 500 g

Cable length 4 m

Electrical

Supply voltage 12 VDC ± 10 %

Supply current

Temperature Range

Operating -15...+55 °C (+5...+131 °F)

Storage -40...+65 °C (-40...+149 °F)

Mounting

Typical less than Maximum Heater OFF

Sensor plate

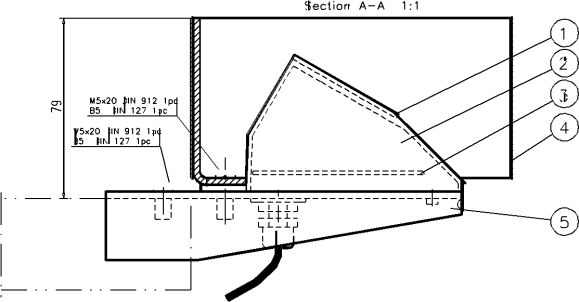
150 mA

260 mA

25 mA

By one screw (M5 x 20 mm) to sensor arm

Heating power 0.5 2.3 W



1. *Wind shield*

Raindrop sensor is basically a board on which nickel is coated in the form of lines. It works on the principal of resistance. When there is no rain drop on board. Resistance is high so we get high voltage according to V=IR. When rain drop present it reduces the resistance because water is conductor of electricity and presence of water connects nickel lines in parallel so reduced resistance and reduced voltage drop across it.

**6.2 Installation of libraries:**

You need to install the **DHTLib** library. It has all the functions needed to get the humidity and temperature readings from the sensor. It’s easy to install.

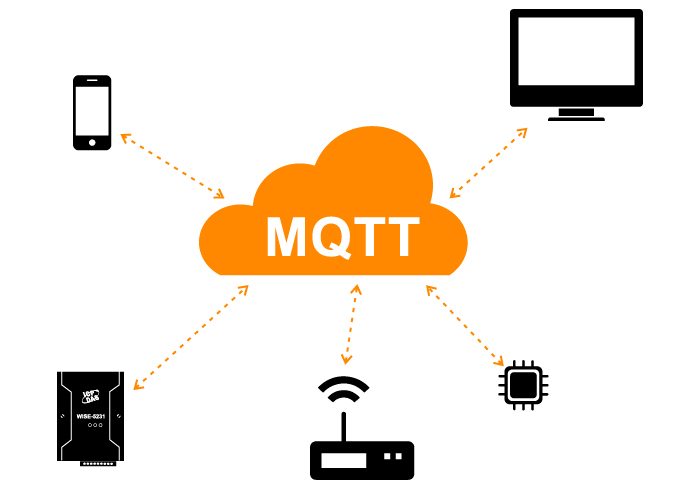
Open up the **Arduino IDE,** then go to **Sketch** > **Include Library** > **Manage Libraries** > **Search DHTLib**

After it’s installed, upload this program to the Node MCU and check output in the serial monitor.

1. After connecting the devices, the code is written and send to the cloud through MQTT protocol.
2. There the code is uploaded into Arduino and it shows live data of temperature and humidity of a particular location for every 5 minutes.

**6.3 MQTT PROTOCOL:**

MQTT means Message Queuing Telemetry Transport. It is small size, light weight, low power usage, minimized data packets and ease of implementation .it is used for “machine-to-machine” connection.



**Fig 6.8: MQTT Protocol**

**Features:**

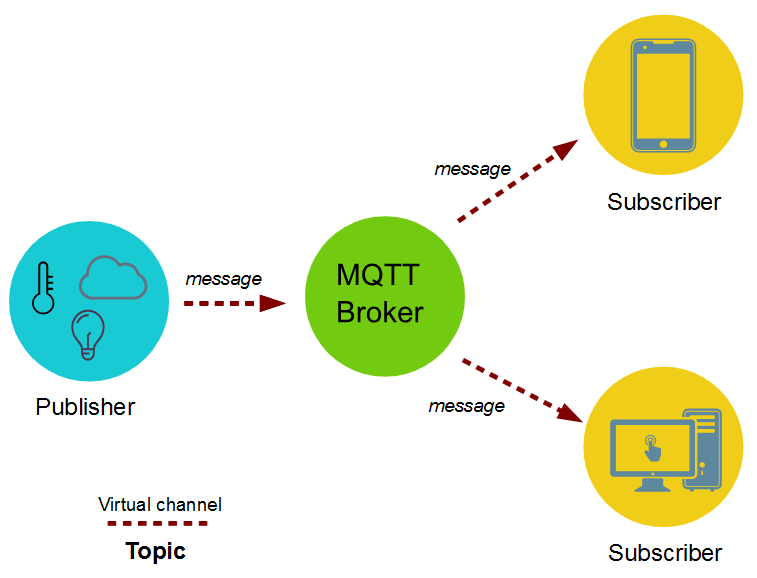
MQTT has unique features you can hardly find in other protocols, like:

* It’s a lightweight protocol. So, it’s easy to implement in software and fast in data transmission.
* It’s based on a messaging technique.
* Minimized data packets. Hence, low network usage.
* Low power usage. As a result, it saves the connected device’s battery.

**Working:**

 MQTT is based on clients and a server. The server is the one who is responsible for handling the client’s requests of receiving or sending data between each other. MQTT server is called a broker and the clients are simply the connected devices.

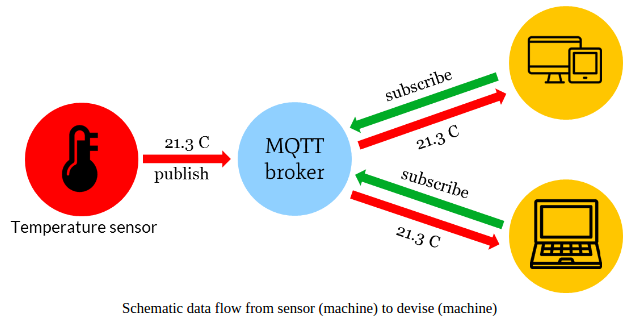
* When a device (a client) wants to send data to the broker, we call this operation a “publish”.
* When a device (a client) wants to receive data from the broker, we call this operation a “subscribe”.
* Publish, is the process a device does to send its message to the broker.
* Subscribe, where a device does to retrieve a message from the broker.

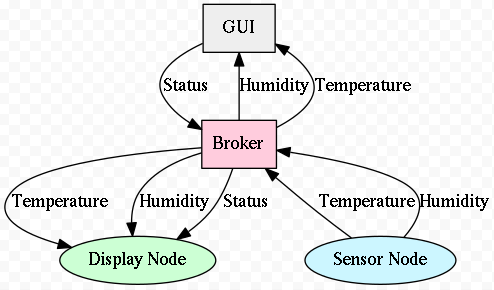


**Fig 6.9: MQTT Broker**

**6.3.1 Sending sensor data to client through MQTT protocol:**

The data such as temperature and humidity values from publisher is send to the **mqtt broker**. The broker role here is to take the message “temperature value” and deliver through a message to the subscriber phone or application.

**Fig 6.10:** **MQTT Schematic Data Flow**

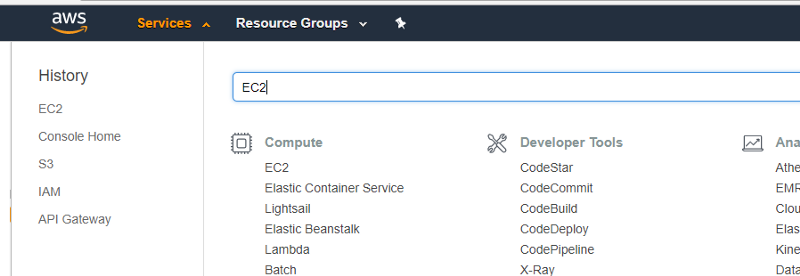


**Fig 6.11:** **MQTT Display INFO**

### 6.3.2 Configure MQTT

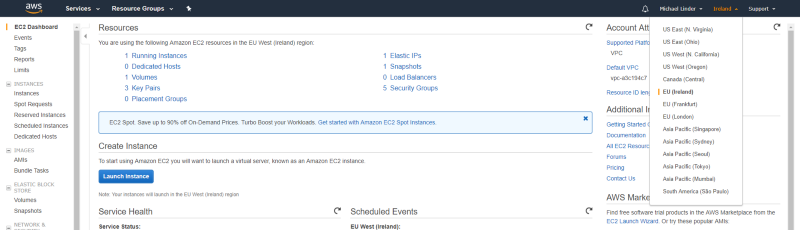
First of all, you need to have an AWS account. If you don’t have one, you can create a new account on AWS ([https://aws.amazon.com](https://aws.amazon.com/)) and get one-year computing, storage and several other services for free using the so called “Free Tier”.

1. Logon to the AWS Console and then select EC2 in services Section



Select EC2

2. Select the preferred AWS region and then launch a new instance



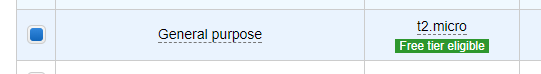
Select AWS region and launch

3. Select Ubuntu Server

https://cdn-images-1.medium.com/max/800/1*njCIH6d4m93BEb5xARdtNg.png

4. Select an instance type

In order to use the free tier contingent, it’s recommended to use t2.micro for testing purposes.



t2.micro selection

5. Configure security group

In this example following ports will be used:

**Step 1: Install Mosquitto**

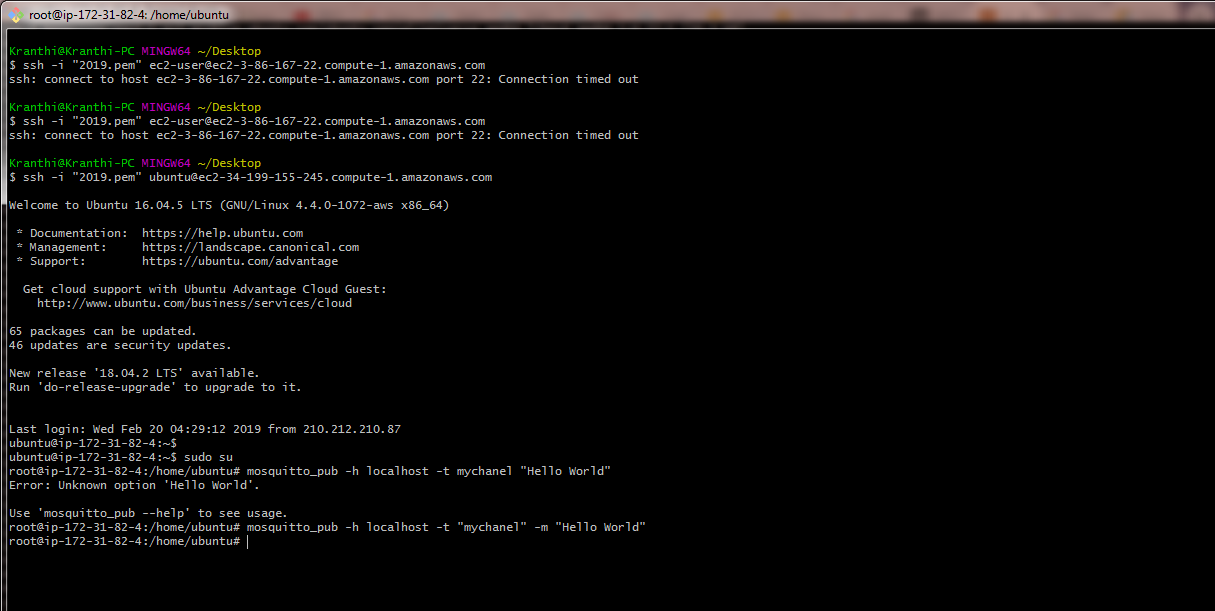
Log into the AWS Ubuntu 16/18 machine.

$ sudo apt-get update

Install

$ sudo apt-get install mosquitto mosquitto-clients

The command above installs both the mosquitto broker and the publish / subscribe clients. The mosquitto broker is now installed and active. You can listen to declare any channel to subscribe and publish to test it



open the duplicate session for this Ubuntu, type the same command for publish

