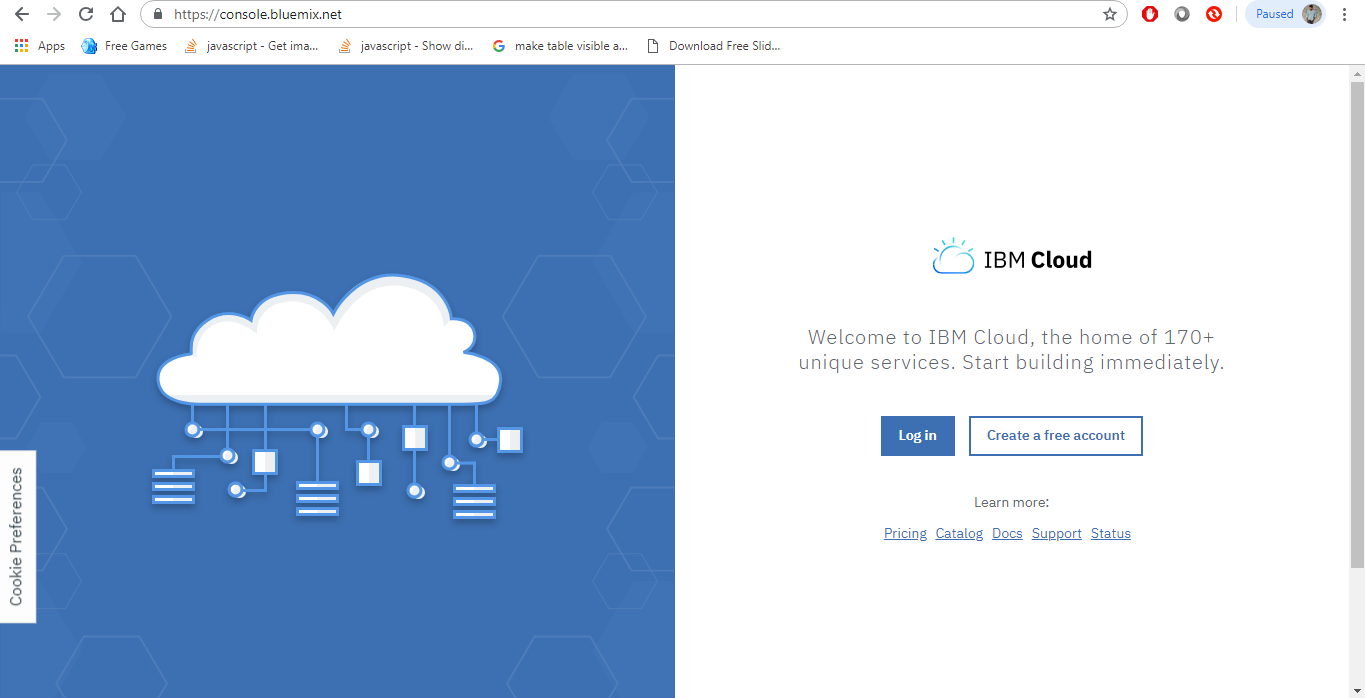
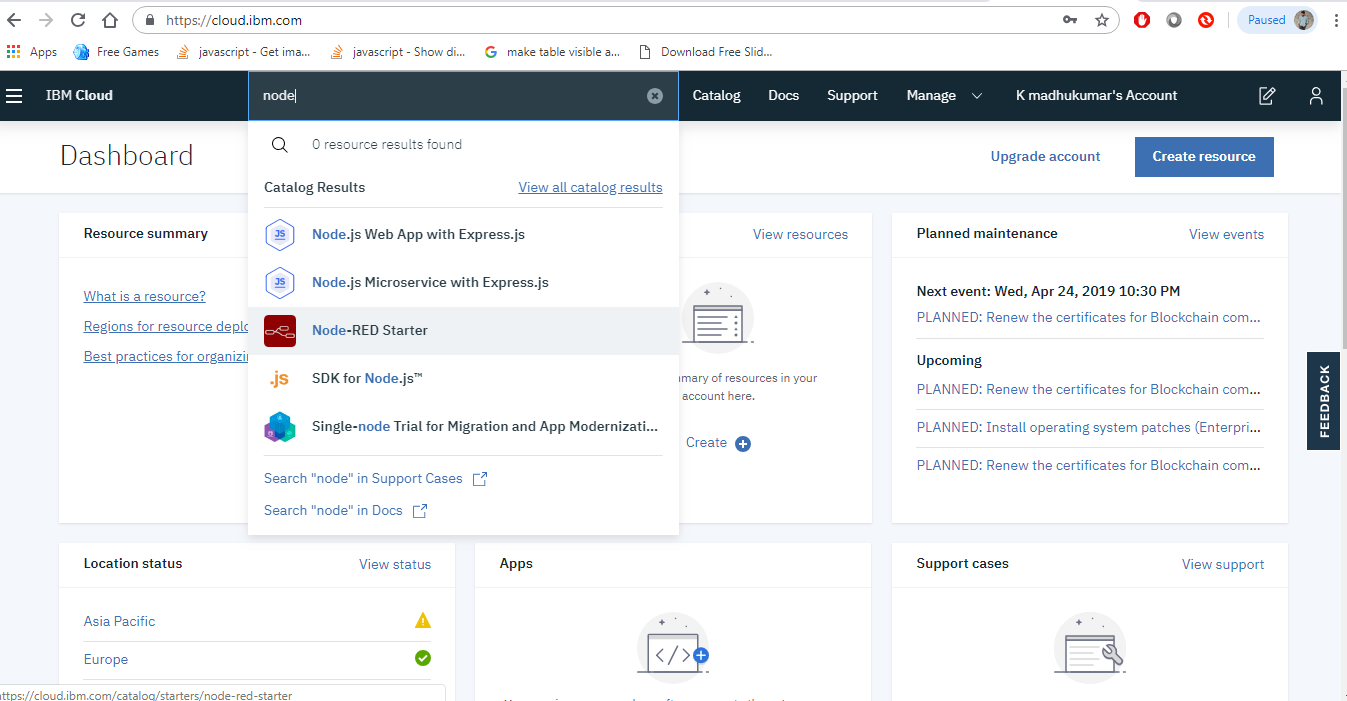
**Sending Moisture data to Nodered using Nodemcu**

**Setting up the Nodered Flow editor using mybluemix.net**

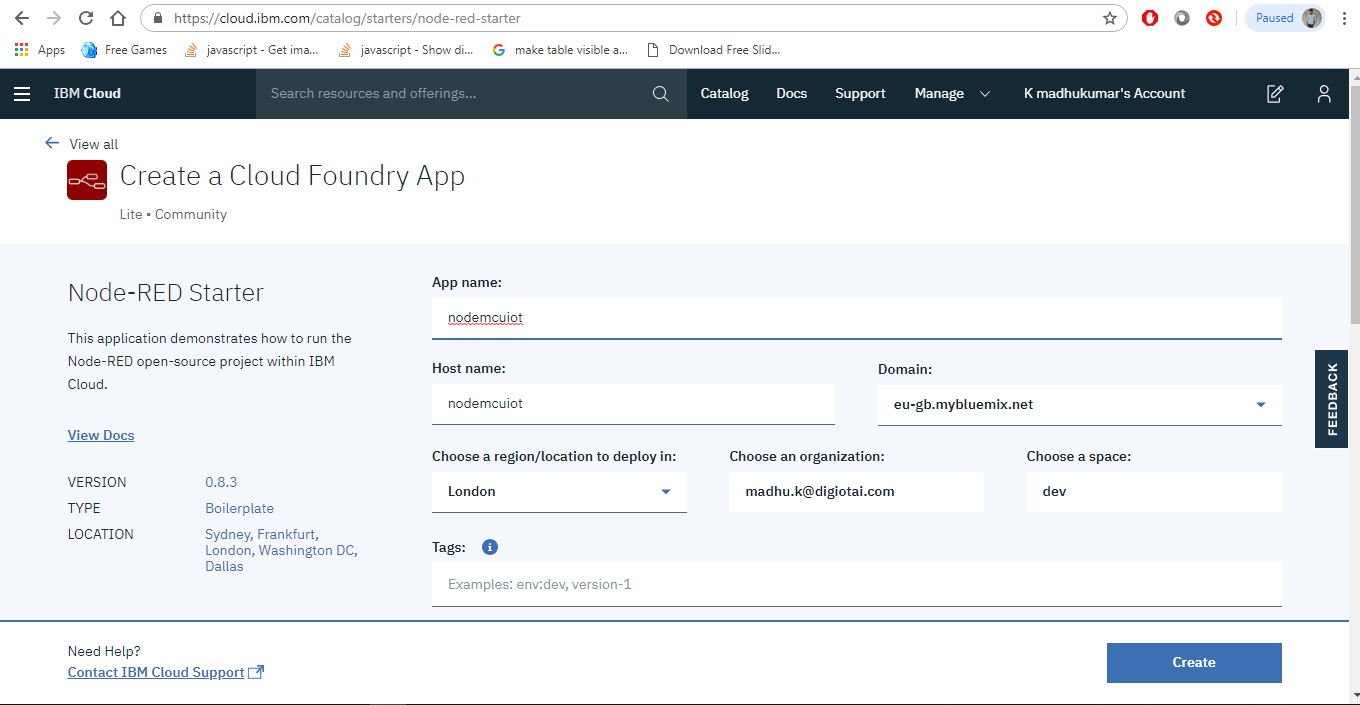
First, create a free account in mybluemix.net



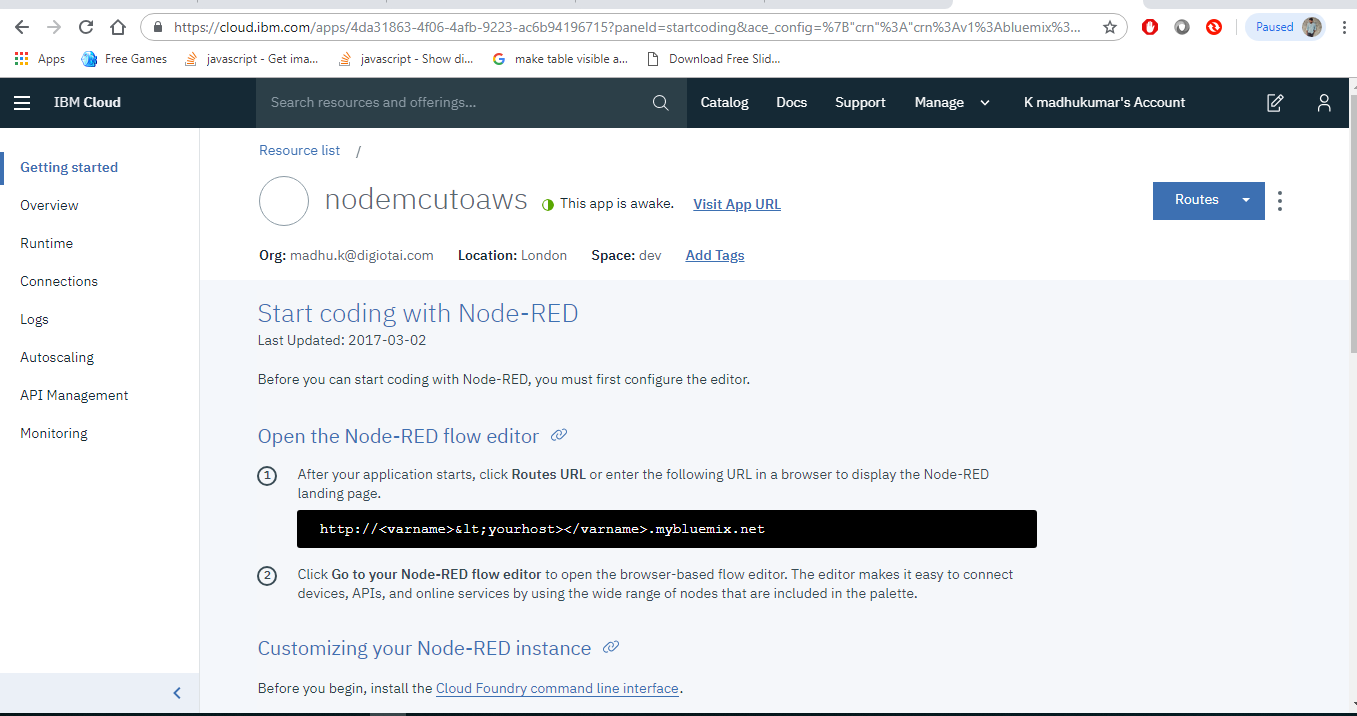
After signing in to the account. In the search box type node-RED



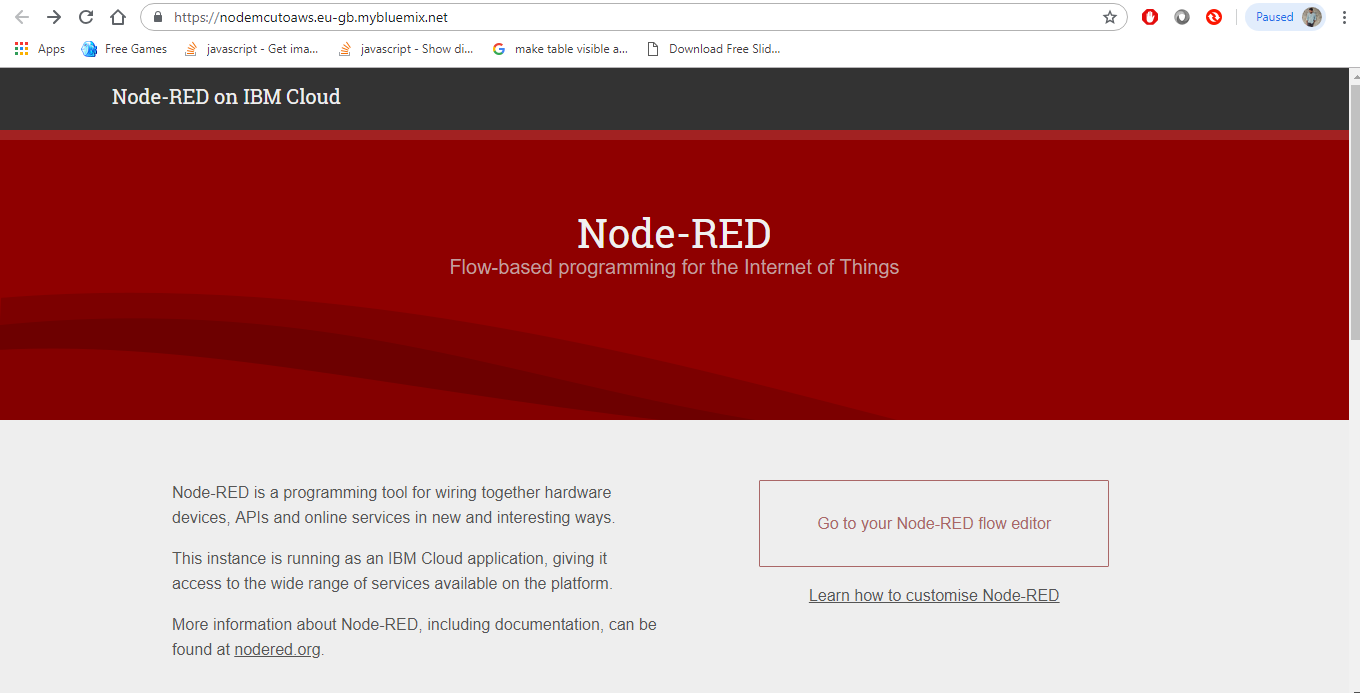
Then select the Node-RED starter from the list and fill the required fields and click on create as shown below



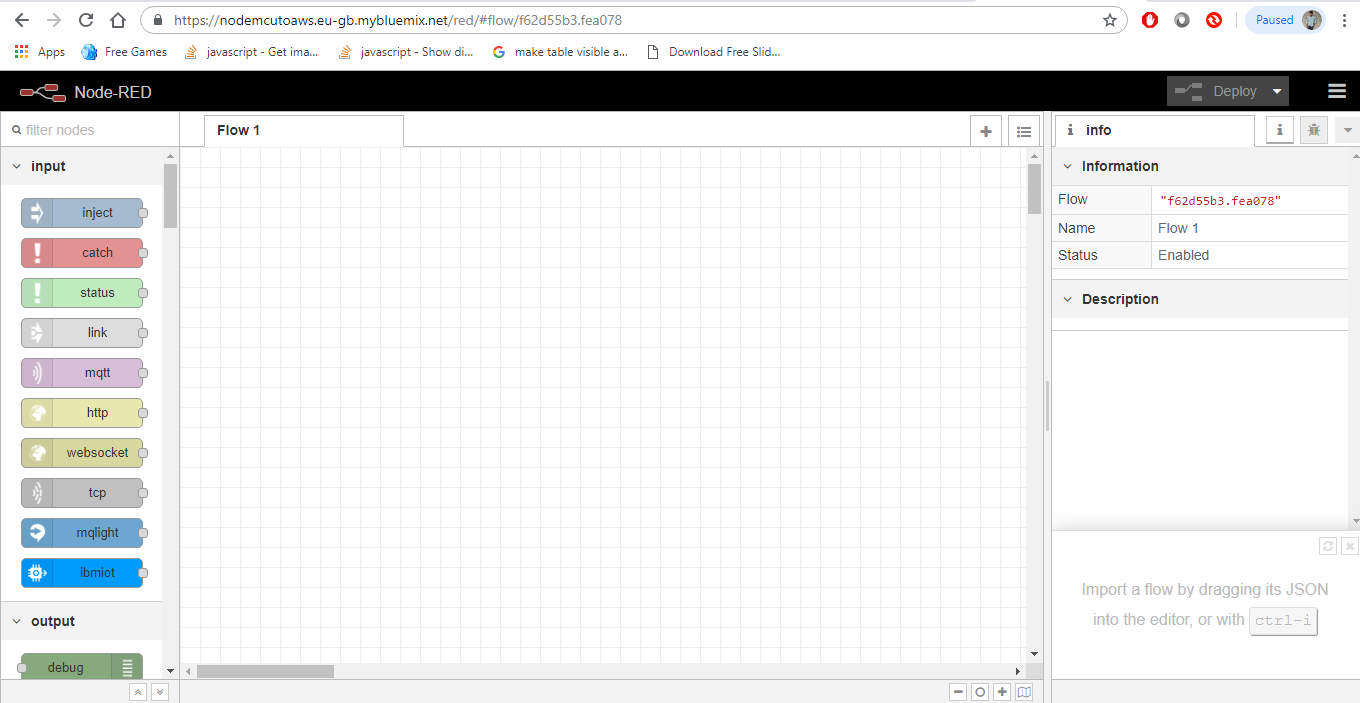
Then click on visit App URL as shown in below



Then follow the steps and finish the process, then the following tab appears

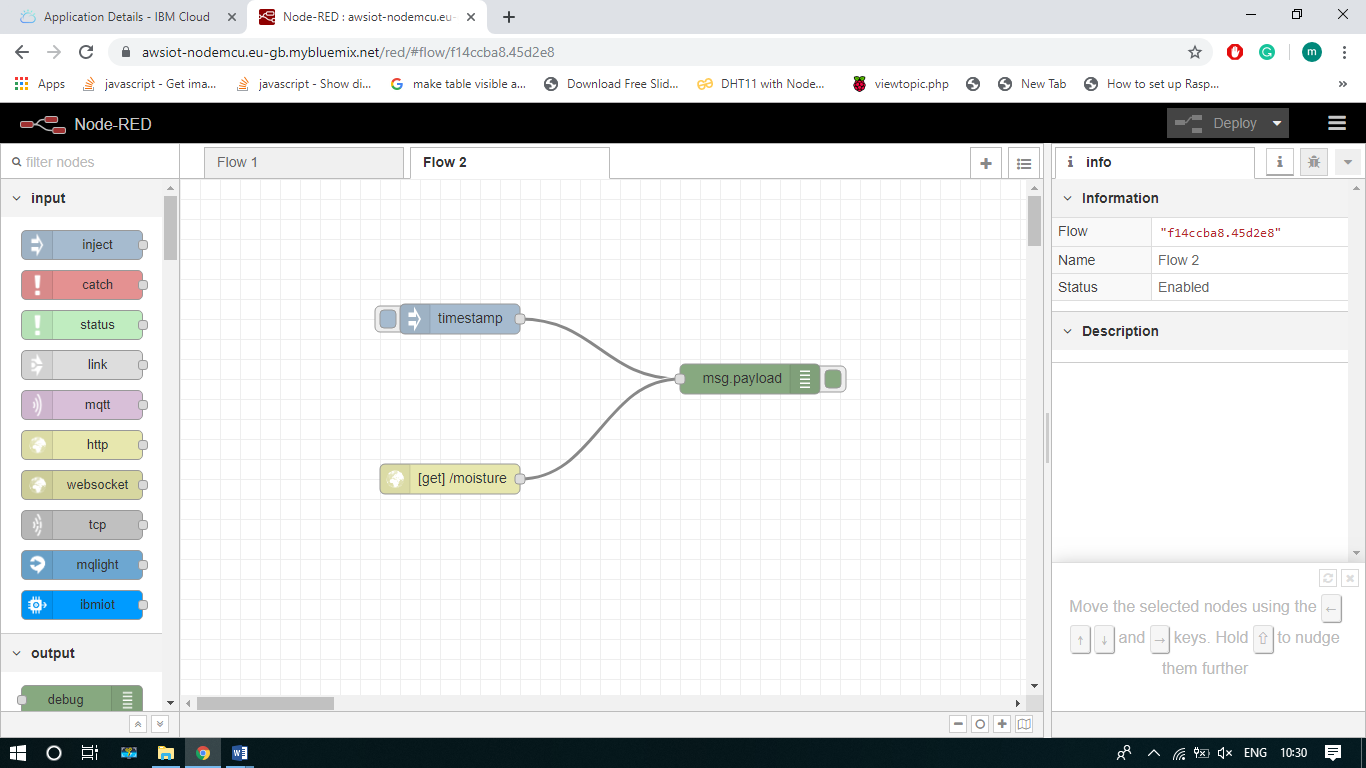


Now click on Go to your Node-RED flow editor



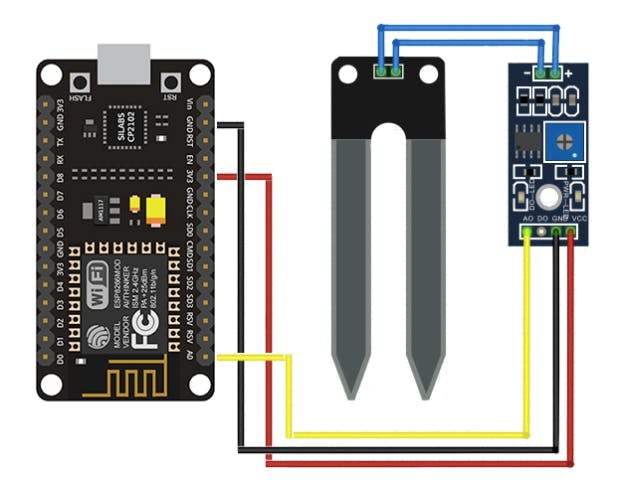
Then configure the connections as follow step by step:

Made the connections as shown in below



After completing the process don’t forget to click on Deploy Button.

Now configure the circuit as shown in below



Now paste the below code in Arduino IDE and compile it.

**#include <ESP8266WiFi.h> //needed for the ESP8266**

**const char WEBSITE[] = "awsiot-nodemcu.eu-gb.mybluemix.net"; //Static IP at Node Red do not use http or https prefix**

**const char\* MY\_SSID = "ACT FIBERNET";**

**const char\* MY\_PWD = "act12345";**

**const int sensor\_pin = A0;**

**void setup()**

**{**

**Serial.begin(115200);**

**// dht.begin();**

**Serial.print("Connecting to "+\*MY\_SSID);**

**WiFi.begin(MY\_SSID, MY\_PWD);**

**Serial.println("going into wl connect");**

**while (WiFi.status() != WL\_CONNECTED) //not connected, ...waiting to connect**

**{**

**delay(1000);**

**Serial.print(".");**

**}**

**Serial.println("wl connected");**

**Serial.println("");**

**Serial.println("Credentials accepted! Connected to wifi\n ");**

**Serial.println("");**

**}**

**void loop()**

**{**

**//Change loop delay as you see fit**

**delay(5000); //5 seconds, adjust as you like relative to sampling rate vs. service call quota**

**float moisture\_percentage;**

**moisture\_percentage = ( 100.00 - ( (analogRead(sensor\_pin)/1023.00) \* 100.00 ) );**

**Serial.print("Soil Moisture(in Percentage) = ");**

**Serial.print(moisture\_percentage);**

**Serial.println("%");**

**WiFiClient client; //Instantiate WiFi object, can scope from here or Globally**

**//Start or API service using our WiFi Client through PushingBox then relayed to Google**

**if (client.connect(WEBSITE, 80))**

**{**

**Serial.print(" Data sent to Node red \n");**

**//http URl format is ---> ?variable1=value&variable2=value2**

**client.print("GET /moisture?moisture=" + (String)moisture\_percentage**

**);**

**// HTTP 1.1 provides a persistent connection, allowing multiple requests to be batched**

**// or pipelined to an output buffer. Careful when altering headers, they arnt forgiving!**

**client.println(" HTTP/1.1");**

**client.print("Host: ");**

**client.println(WEBSITE);**

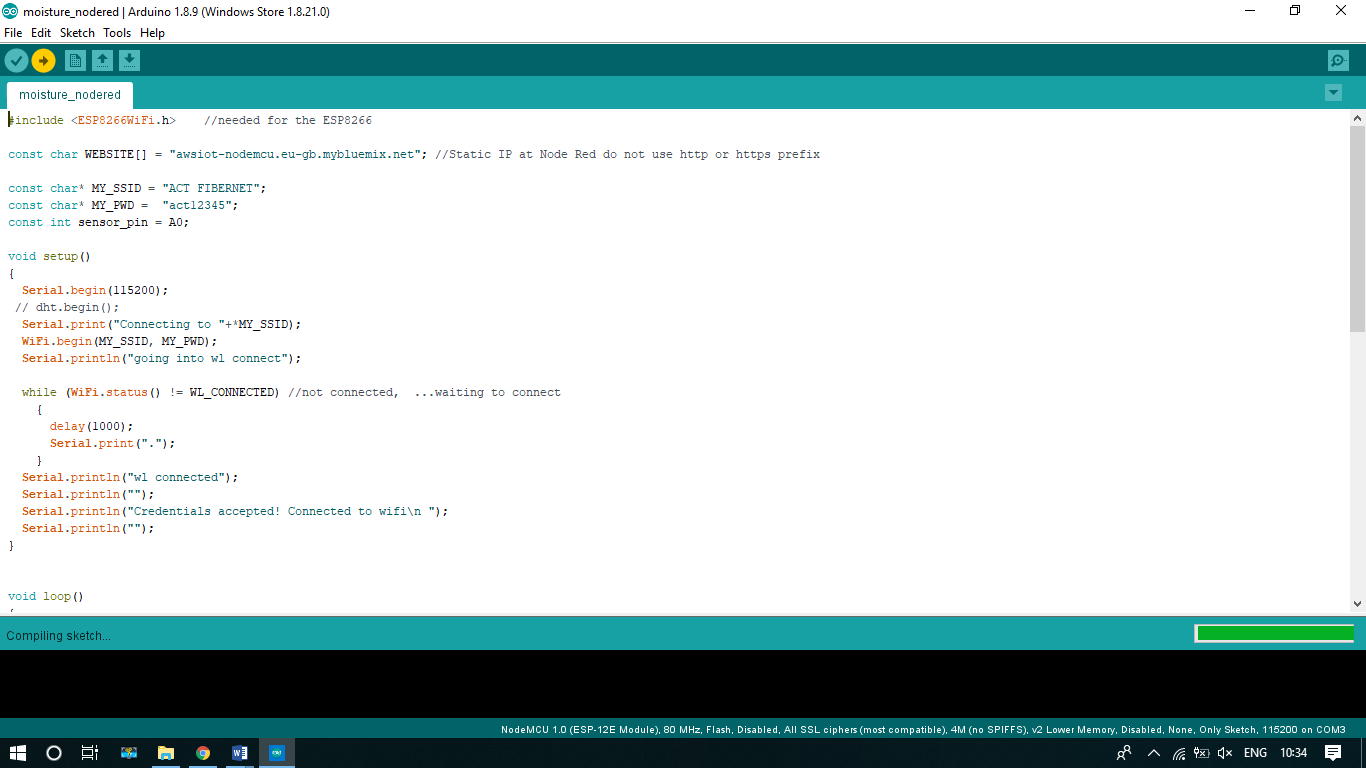
**client.println("User-Agent: ESP32/1.0");**

**client.println("Connection: close");**

**client.println();**

**}**

**}**



After compiling open the serial monitor & Node Red flow editor to check the values are updated as follows

