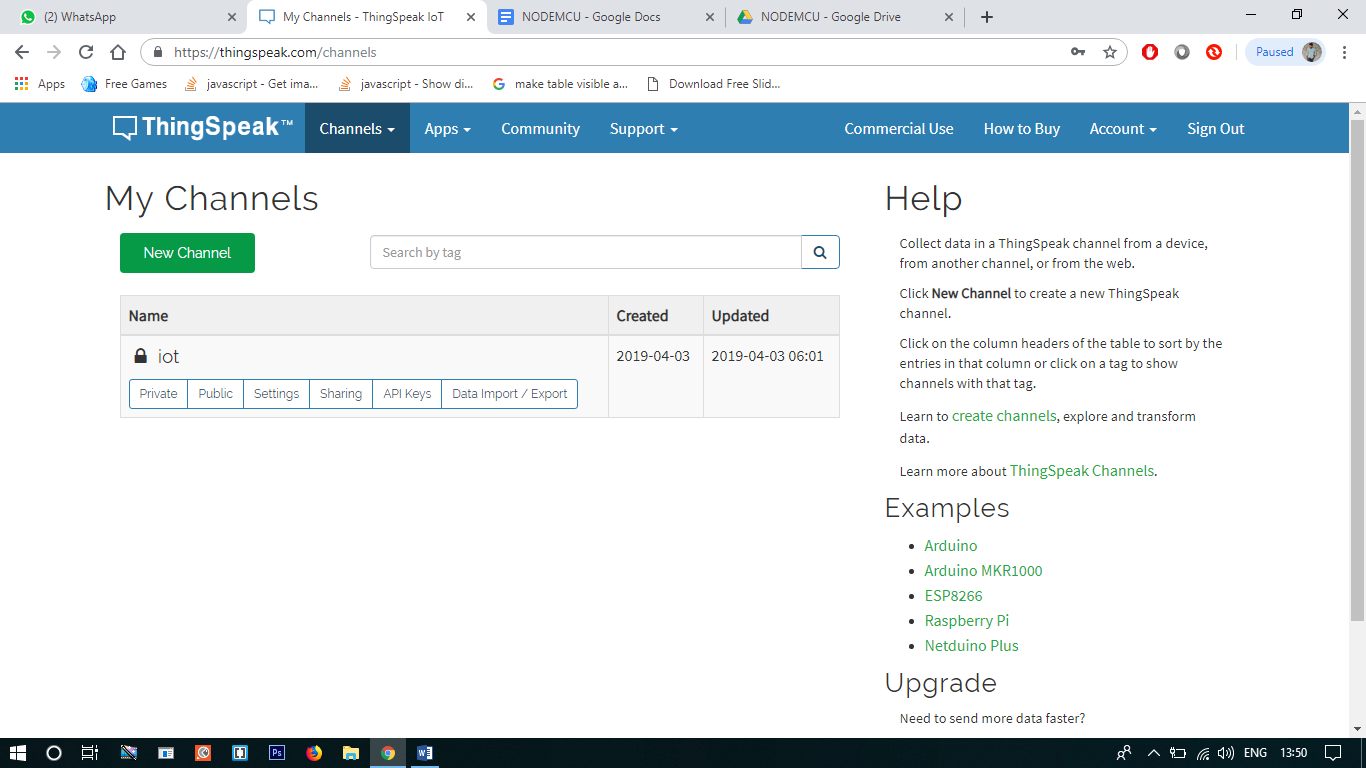
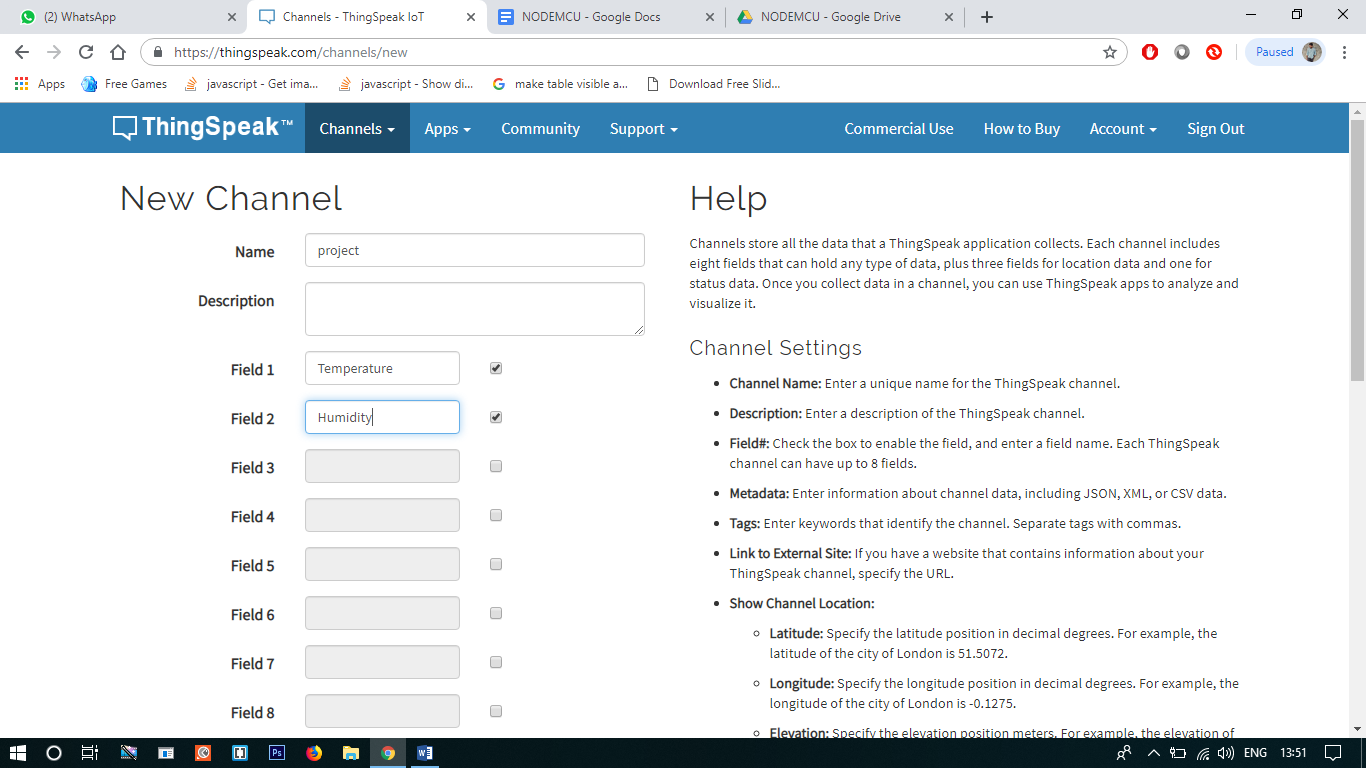
DIGIOTAI’S Step by step guide to Monitor the cold Storage temperature and humidity from DHT11 sensor with ARDUINO UNO in a logistics environment in a supply chain management scenario by using Thingspeak

Step 1: Setting Up the ThingSpeak

Go to <https://thingspeak.com/> and Sign up a free account

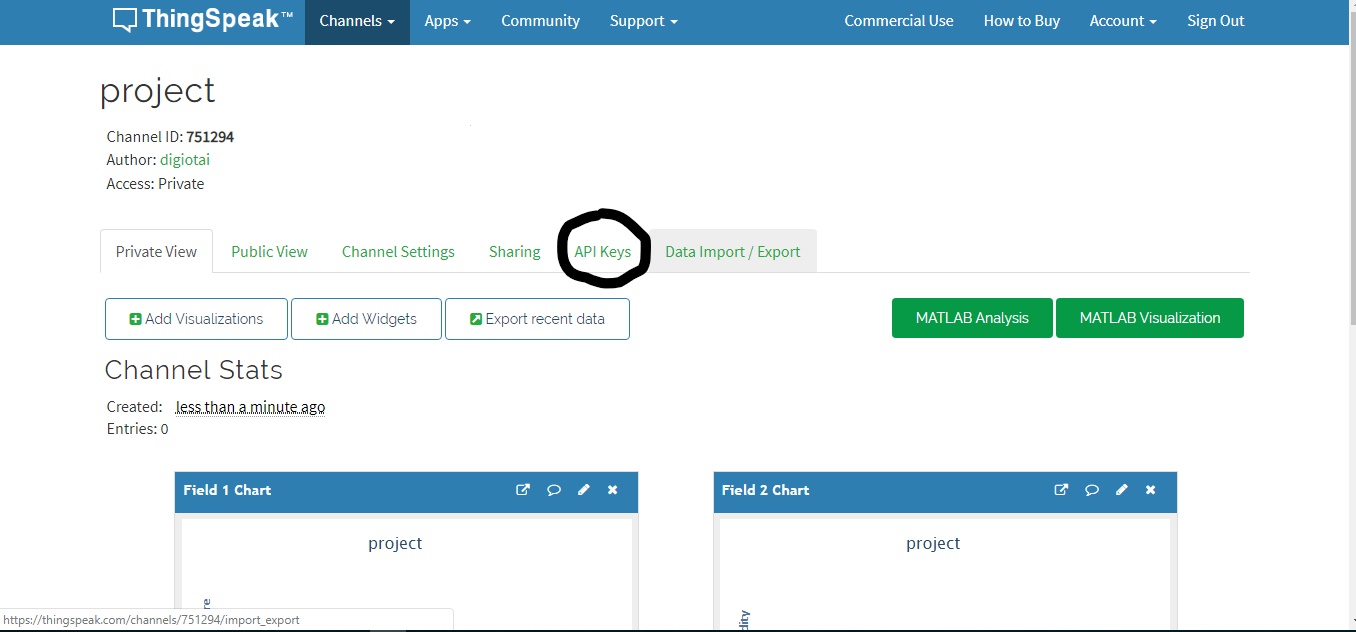


Then Click on New Channel

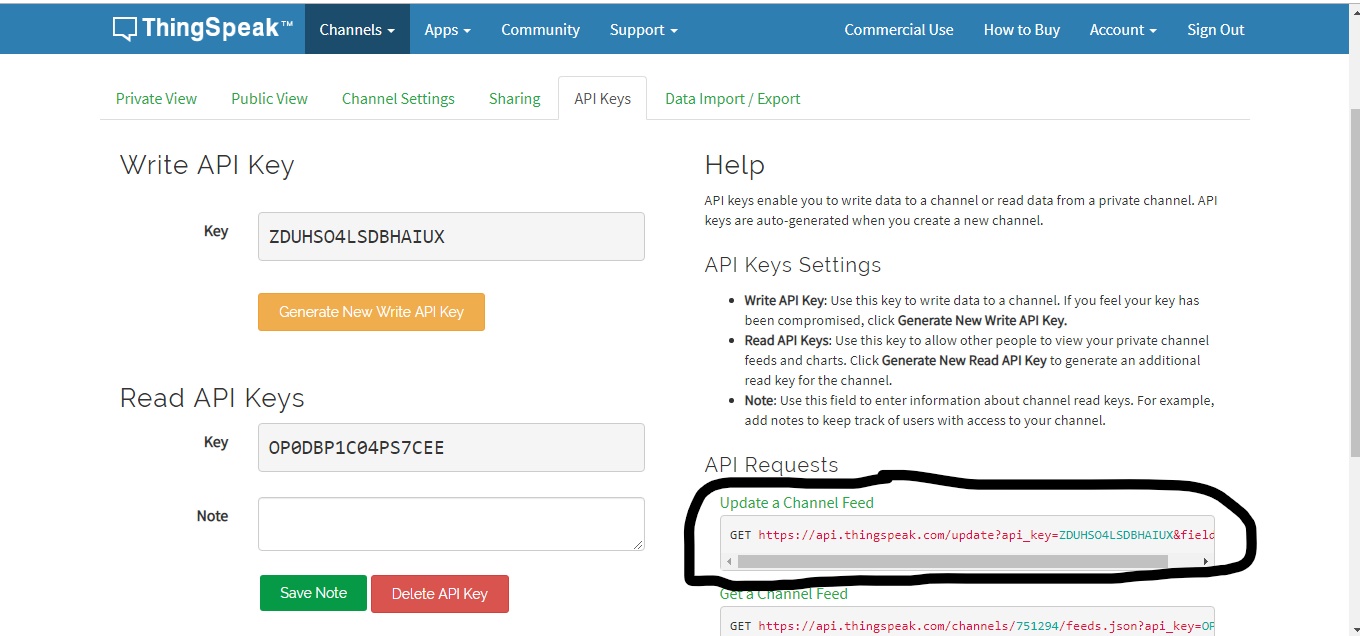


Then fill the required fields and click on Save Channel

Then navigate through API Keys



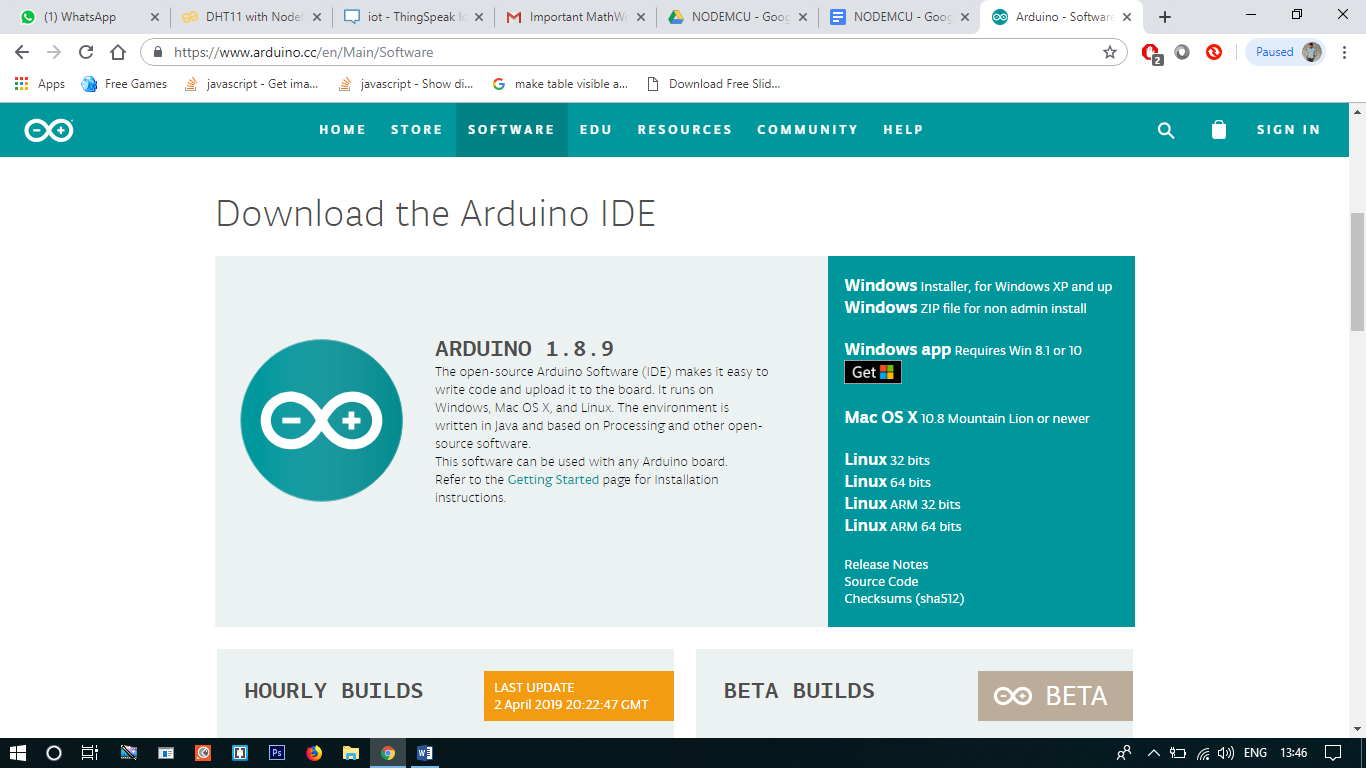
Here Copy and paste the url in Source code which is shown below



Step 2: Setting up Arduino IDE and Install Libraries

The Arduino IDE is a cross platform application that is written in the programming language java. It is used to write and upload programs to Arduino Board.

1. Go to <https://www.arduino.cc/en/Main/Software>
2. Select the platform and download

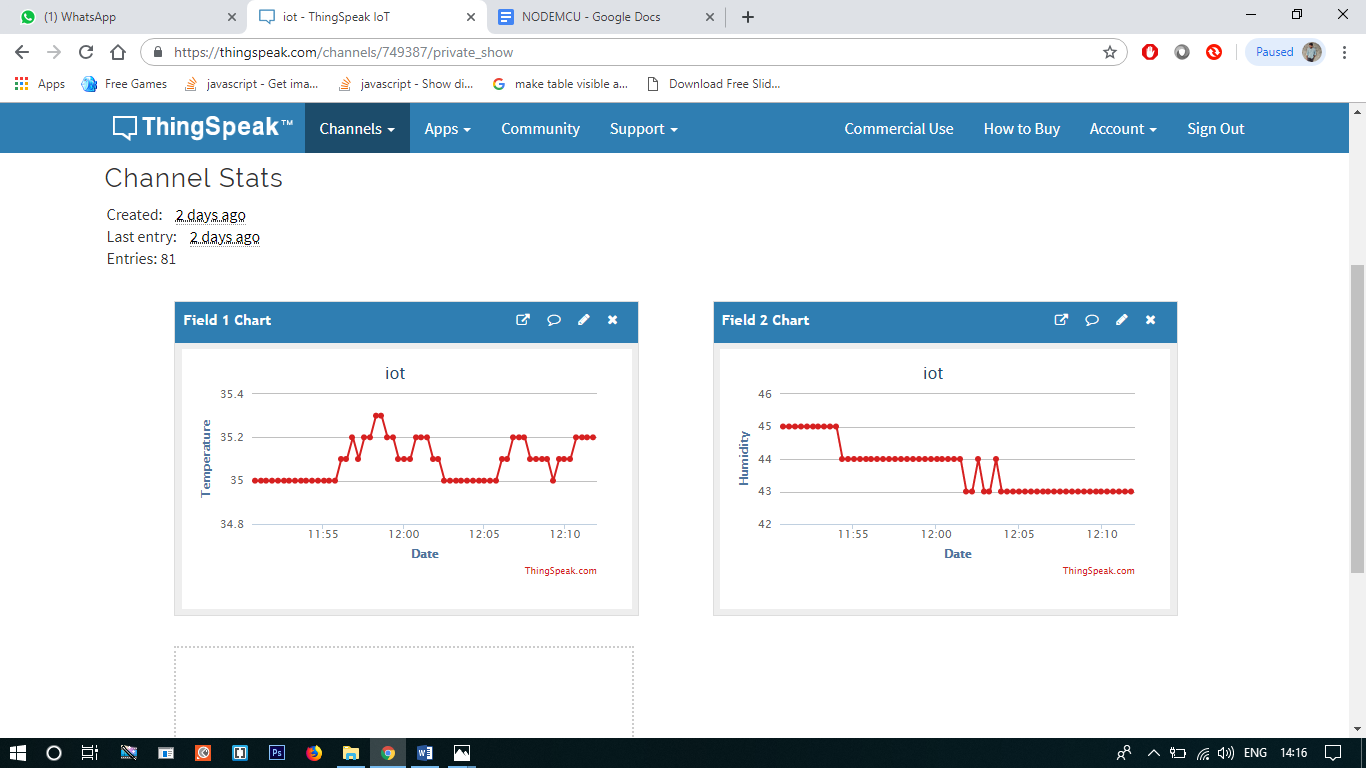


1. Install and open Arduino IDE
2. Download Libraries from <https://github.com/adafruit/DHT-sensor-library>
3. Unzip the downloaded libraries and paste them in c:\users\.....\documents\Arduino\Libraries

Step 3: Executing source code in Arduino IDE

1. Paste the temperature and humidity source code in Arduino IDE and upload to  NodeMCU board.
2. Click on the serial monitor present at the right corner of the Arduino IDE and we can see the temperature and humidity reporting to Amazon AWS.

After running the code go to things speak -> my channel



The source code is given below :

#include <ESP8266WiFi.h>

#include "DHT.h"

#define DHTPIN D3 // what pin we're connected to

#define DHTTYPE DHT11 // define type of sensor DHT 11

DHT dht (DHTPIN, DHTTYPE);

const char\* ssid = "ACT FIBERNET";//Enter the ssid of your router

const char\* password = "act12345";//Enter the password of your router

const char\* host = "api.thingspeak.com";

const char\* privateKey = "H24DFUQFPGIZ1MU3";

float h,t;

void setup() {

Serial.begin(115200);

delay(10);

dht.begin();

// We start by connecting to a WiFi network

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

void loop()

{

delay(15000);

h = dht.readHumidity();

t = dht.readTemperature();

if (isnan(t) || isnan(h))

{

return ;

}

Upload\_to\_Cloud();

}

/\*......................................Sensor Data.....................................

void Sensor\_Data()

{

h = dht.readHumidity();

t = dht.readTemperature();

}

.....................................Uploading to Cloud...............................\*/

void Upload\_to\_Cloud()

{

Serial.print("connecting to ");

Serial.println(host);

// Use WiFiClient class to create TCP connections

WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {

Serial.println("connection failed");

return;

}

// https://api.thingspeak.com/update?api\_key=ZKHCKOU5Q3D8I9IM&field1=0

// We now create a URI for the request

String url = "/update";

url += "?api\_key=";

url += privateKey;

url += "&field1=";

url += t;

url += "&field2=";

url += h;

Serial.print("Requesting URL: ");

Serial.println(url);

// This will send the request to the server

client.print(String("GET ") + url + " HTTP/1.1\r\n" +

"Host: " + host + "\r\n" +

"Connection: close\r\n\r\n");

delay(1000);

// Read all the lines of the reply from server and print them to Serial

while(client.available()){

String line = client.readStringUntil('\r');

Serial.print(line);

}

Serial.println();

Serial.println("closing connection");

}