Lesson 5 –
Cloud computing basics

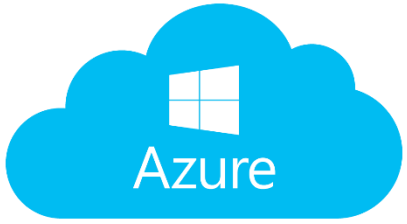
- S.P. Chong

Objectives

- In this lesson, you will learn some basic **cloud computing** concepts.
- You will then learn to **set up** an **account** with a public cloud service provider e.g. **Thingspeak**.
- You will next learn to program an Arduino UNO, to **send data** from a sensor to the cloud platform.
- After that, you will learn to send an event-triggered **notification**, through **Twitter**, for instance.

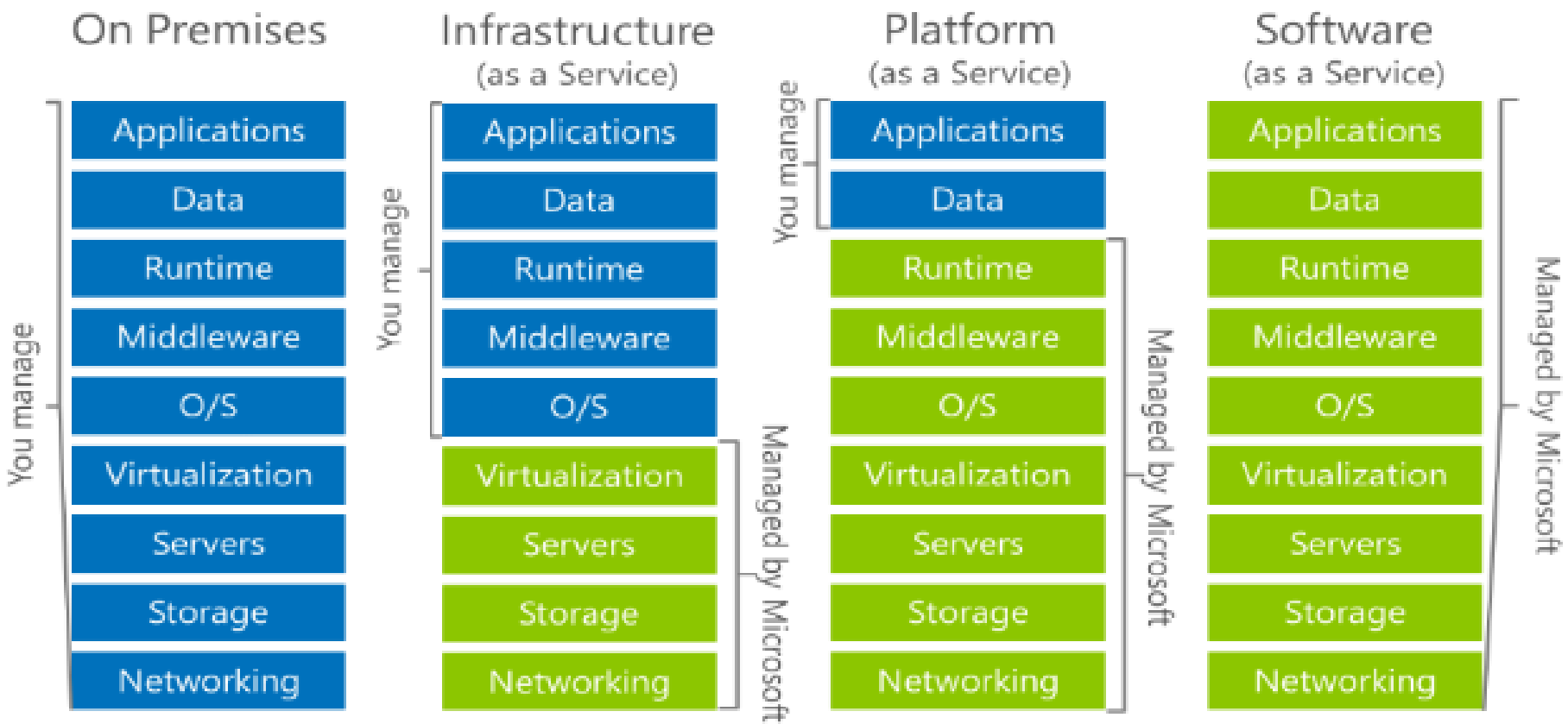
What is cloud computing?

- In the past, companies and organisations maintain their **own servers** e.g. database servers, file servers, web servers, mail servers etc.
- This means a lot of man-hours spent to maintain and update the high-end **computers**, **software packages** & **storage spaces** (hard disks etc.).
- Nowadays, many companies and organisations have started to use cloud computing services provided by **AWS** (Amazon Web Services), Microsoft's **Azure** & **Google Cloud**.
- What this means is, they simply pay for the use of the processors & storage spaces in the cloud.
- Of course high-speed **internet connection** is required for this to be possible.



What is cloud computing? (cont.)

- As an example, Microsoft offers these 4 choices:



What is cloud computing? (cont.)

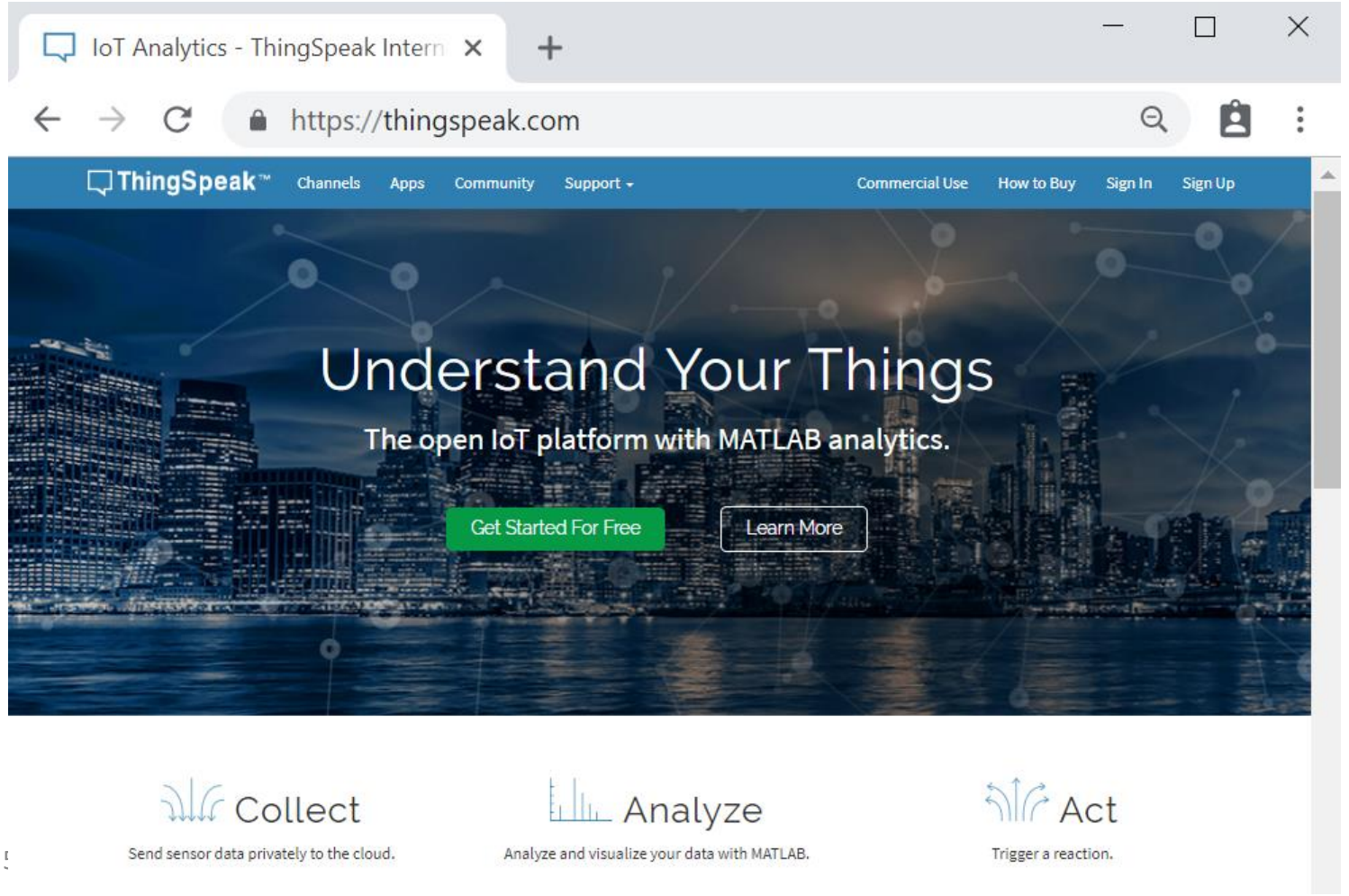
- Cloud computing provides a **cost effective**, **flexible** option for businesses to deploy software solutions.
- When applications run in the cloud, organizations no longer have to make major investments in **infrastructure**. These costs are now a predictable **monthly subscription**, so you only pay for what you actually use.

Some of the benefits the Cloud brings are:

- Reduced hardware and software acquisition costs.
- Ability to focus on the core business vs technology.
- Improved cash flow under Pay for Use payment models.
- Eliminating software upgrade and update processes.
- Improved accessibility to applications remotely.
- Continuous availability to new updates and new features

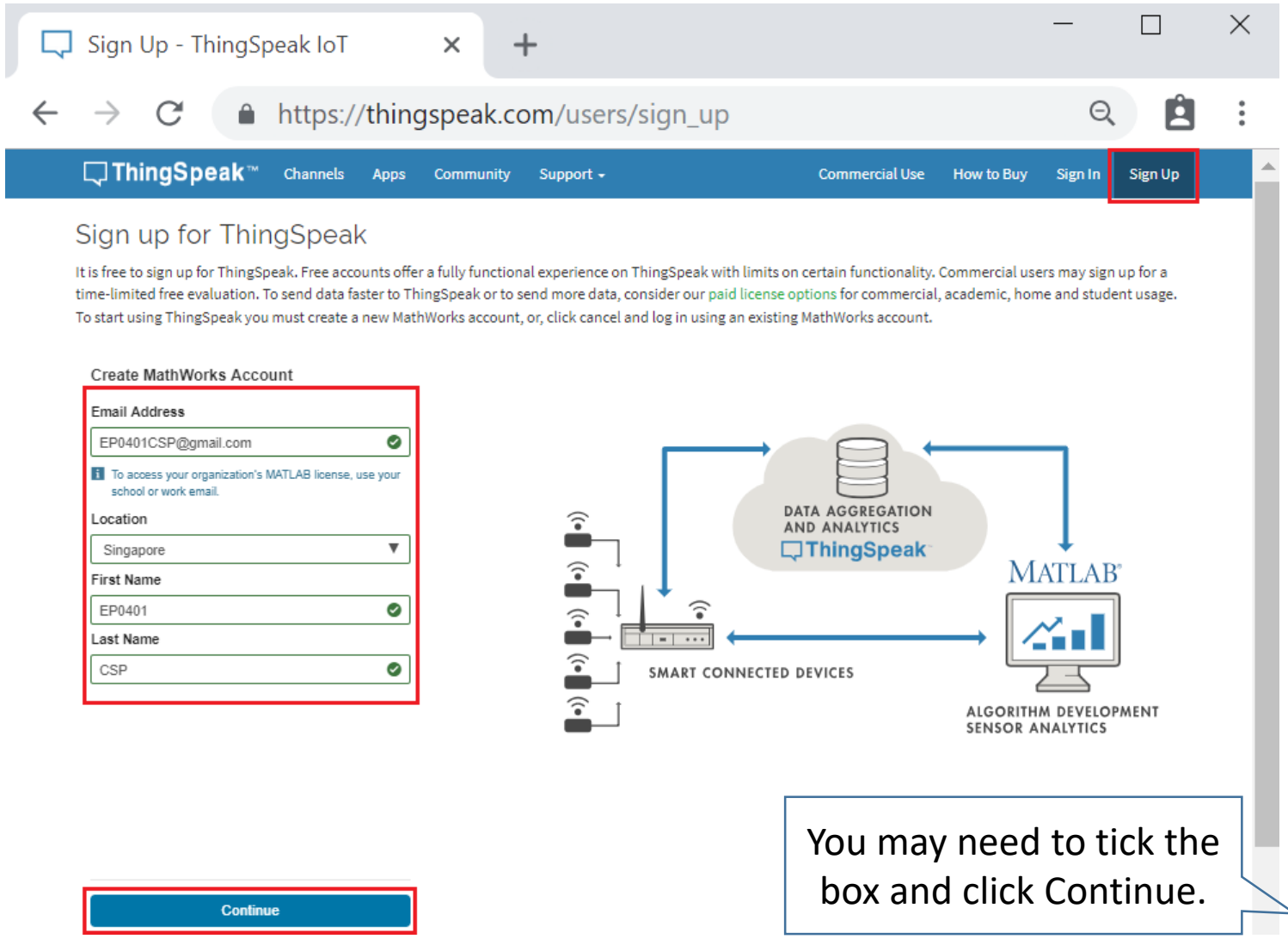
Setting up a Cloud platform (e.g. Thingspeak)

- Let's see how we can set up **Thingspeak**, a cloud platform for uploading sensor data.



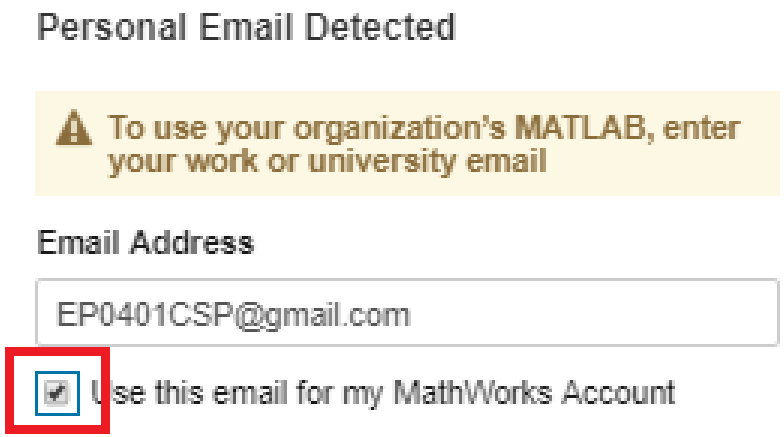
You can create a Thingspeak account at www.thingspeak.com

Setting up a Cloud platform (e.g. Thingspeak) (cont.)



Click Sign Up, fill in the Email Address etc., and click Continue.

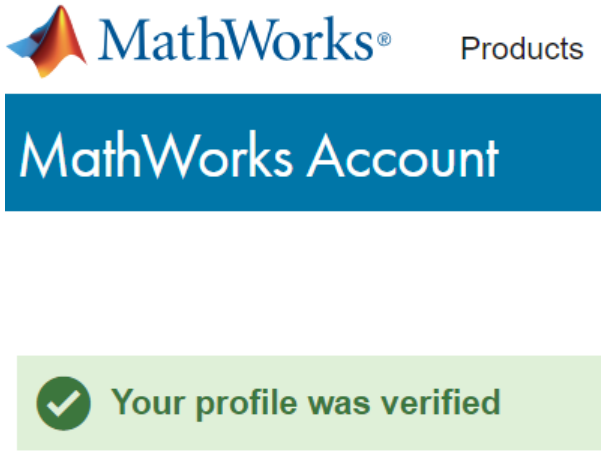
You may need to tick the box and click Continue.



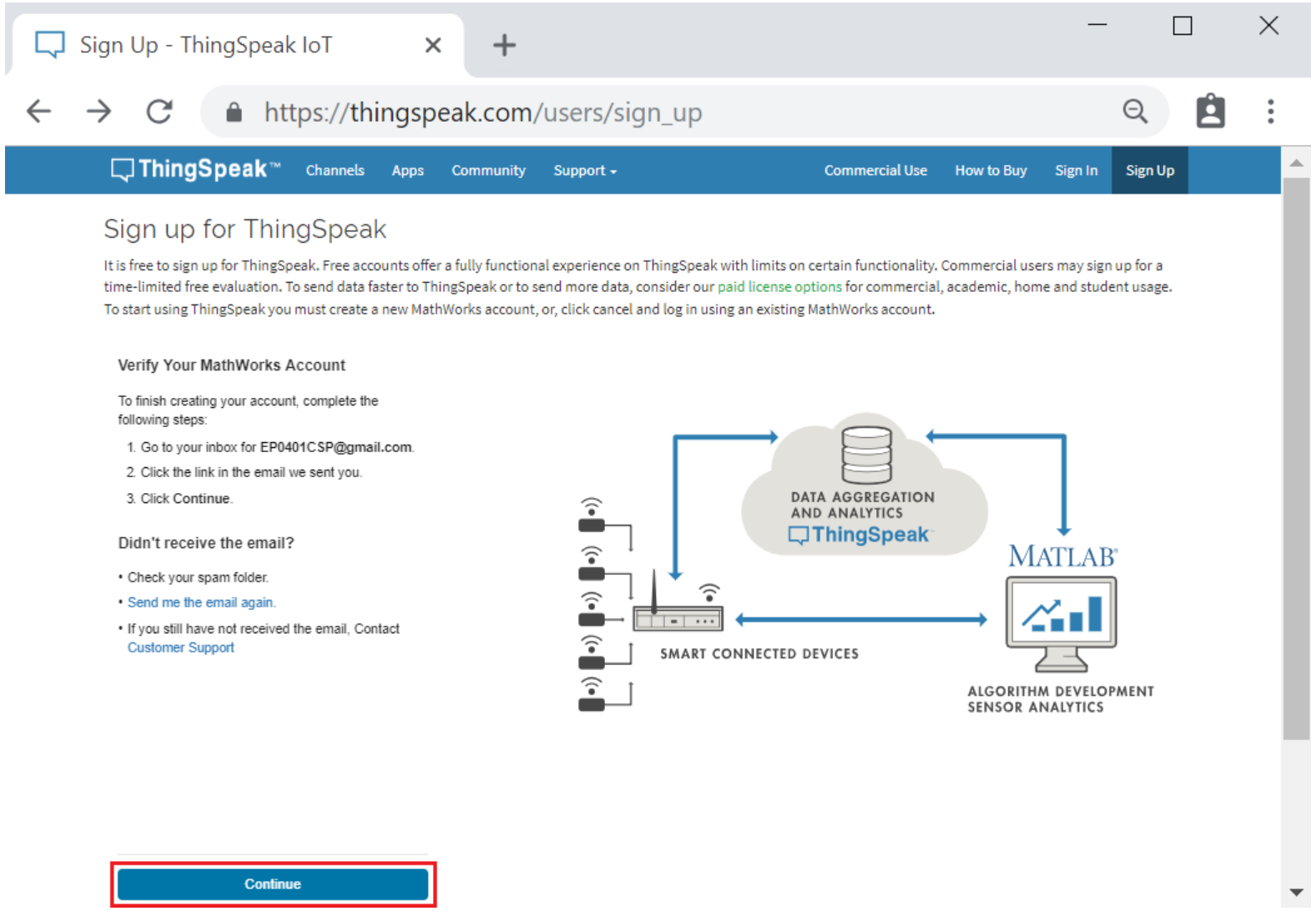
Setting up a Cloud platform (e.g. Thingspeak) (cont.)

An email has been sent to you. Open that email and click Verify your email.

A new tab will open to inform you “Your profile was verified”.

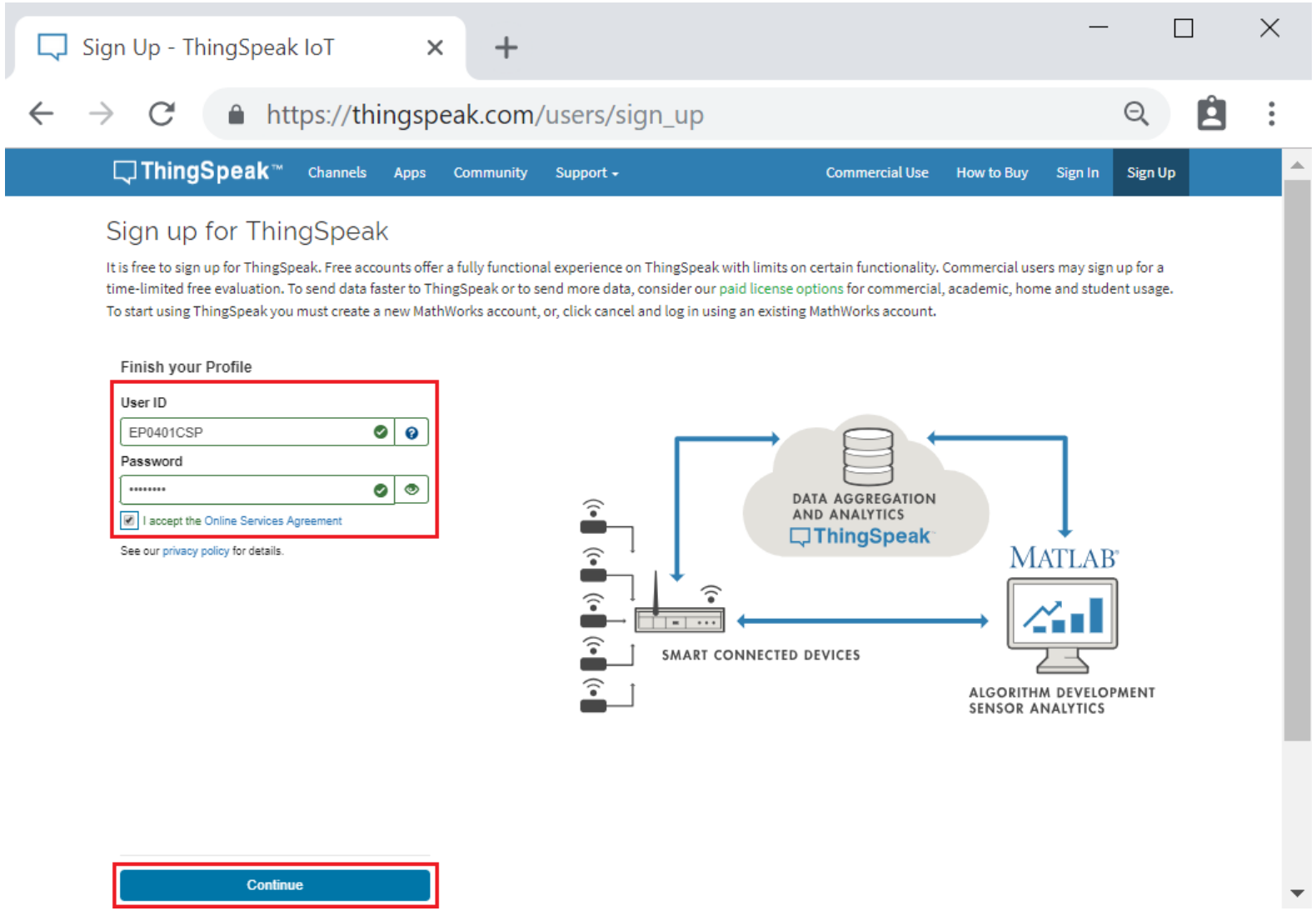


Setting up a Cloud platform (e.g. Thingspeak) (cont.)



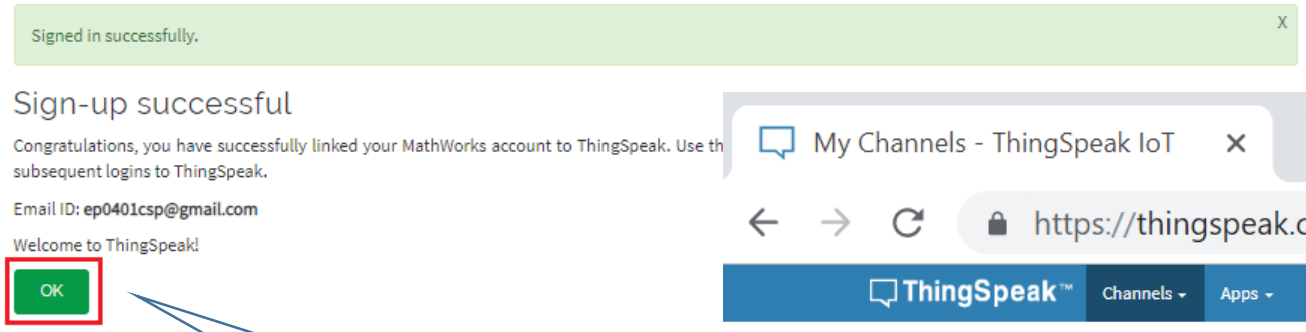
Back to the Thingspeak tab, click Continue.

Setting up a Cloud platform (e.g. Thingspeak) (cont.)

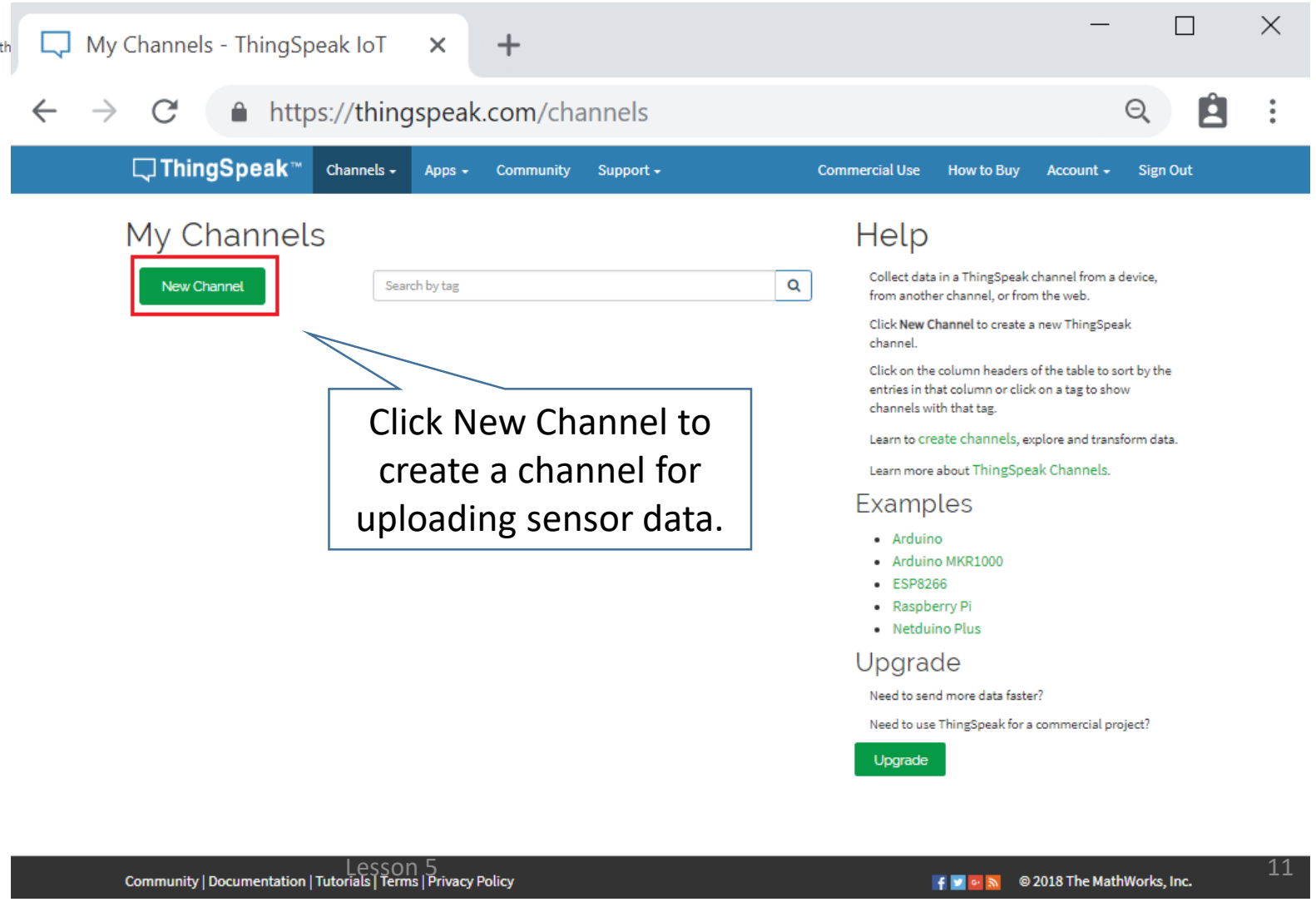


Enter your User ID and Password, tick the box and click Continue.

Setting up a Cloud platform (e.g. Thingspeak) (cont.)

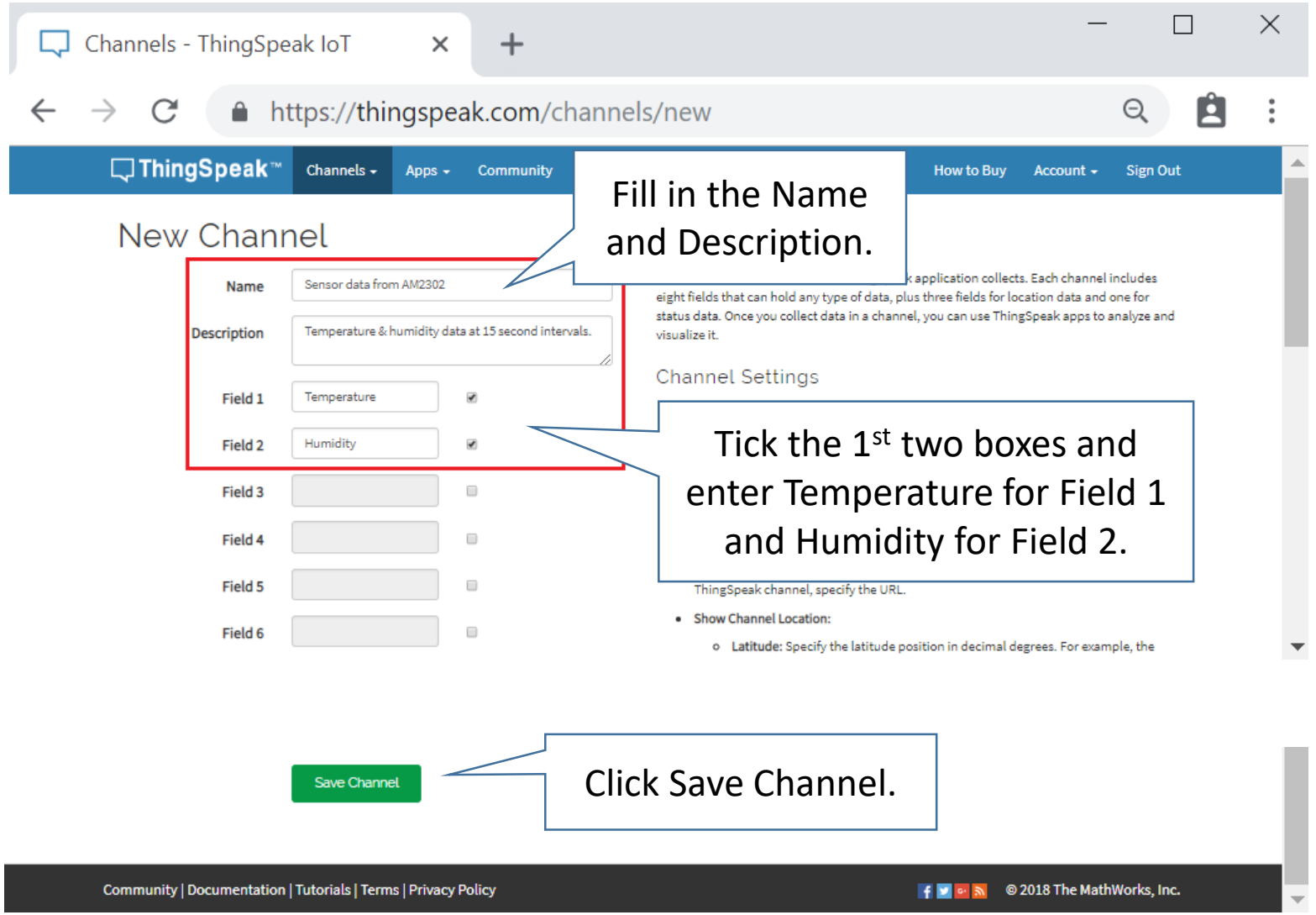


Click OK to start using Thingspeak.

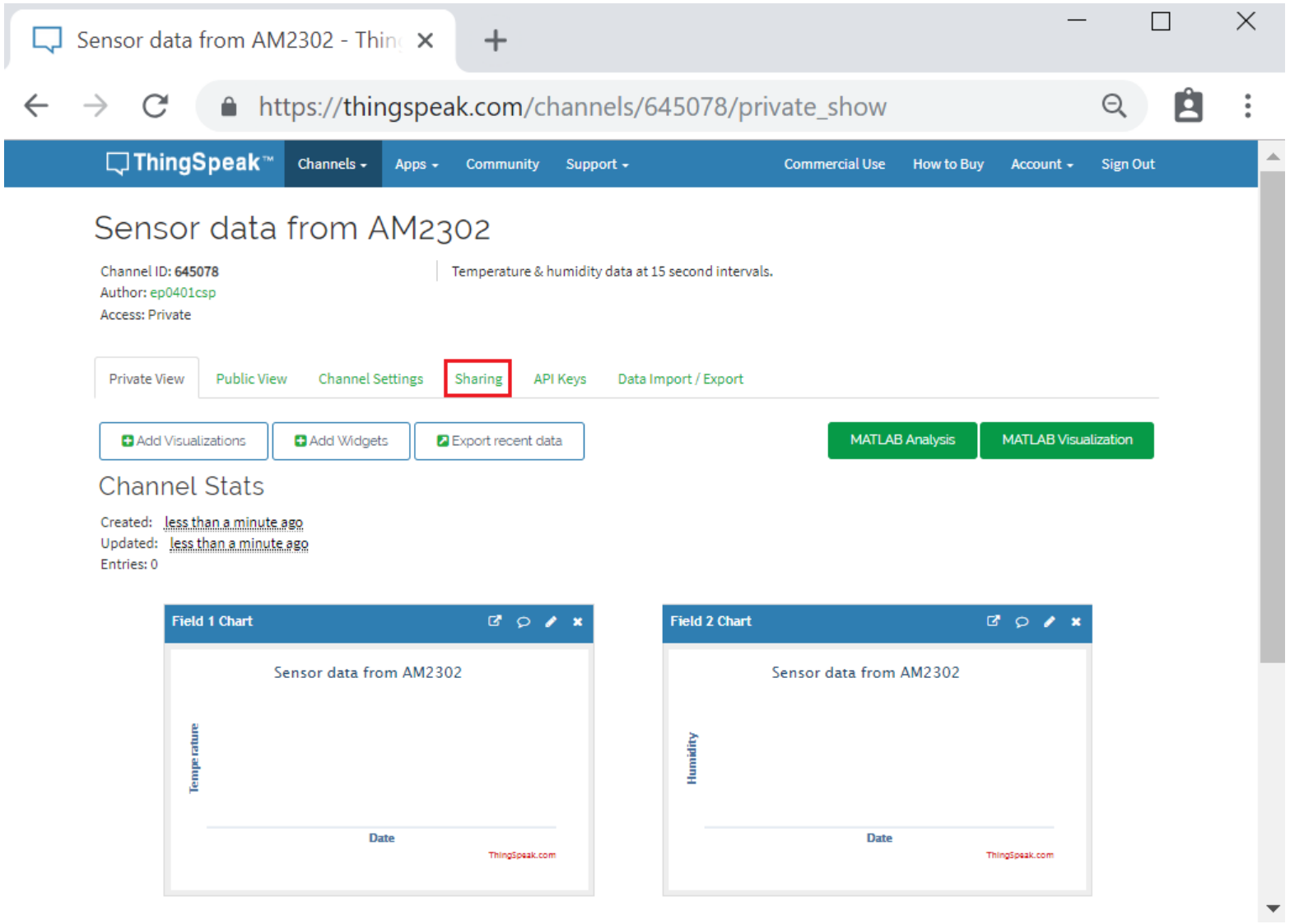


Click New Channel to create a channel for uploading sensor data.

Setting up a Cloud platform (e.g. Thingspeak) (cont.)



Setting up a Cloud platform (e.g. Thingspeak) (cont.)



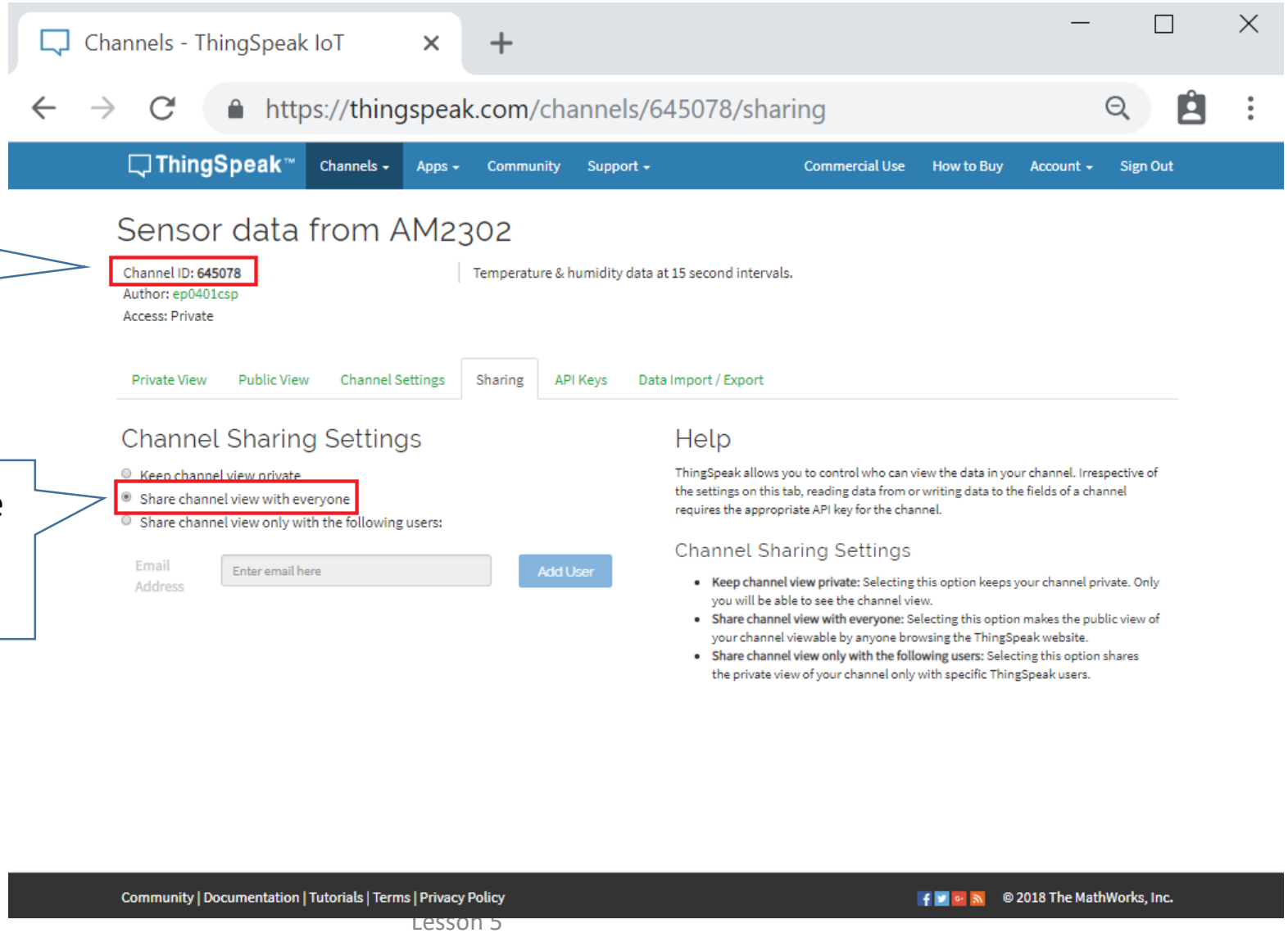
You will see 2 charts (Temperature & Humidity) in the channel just created.

The Access is now Private. To change this, click Sharing.

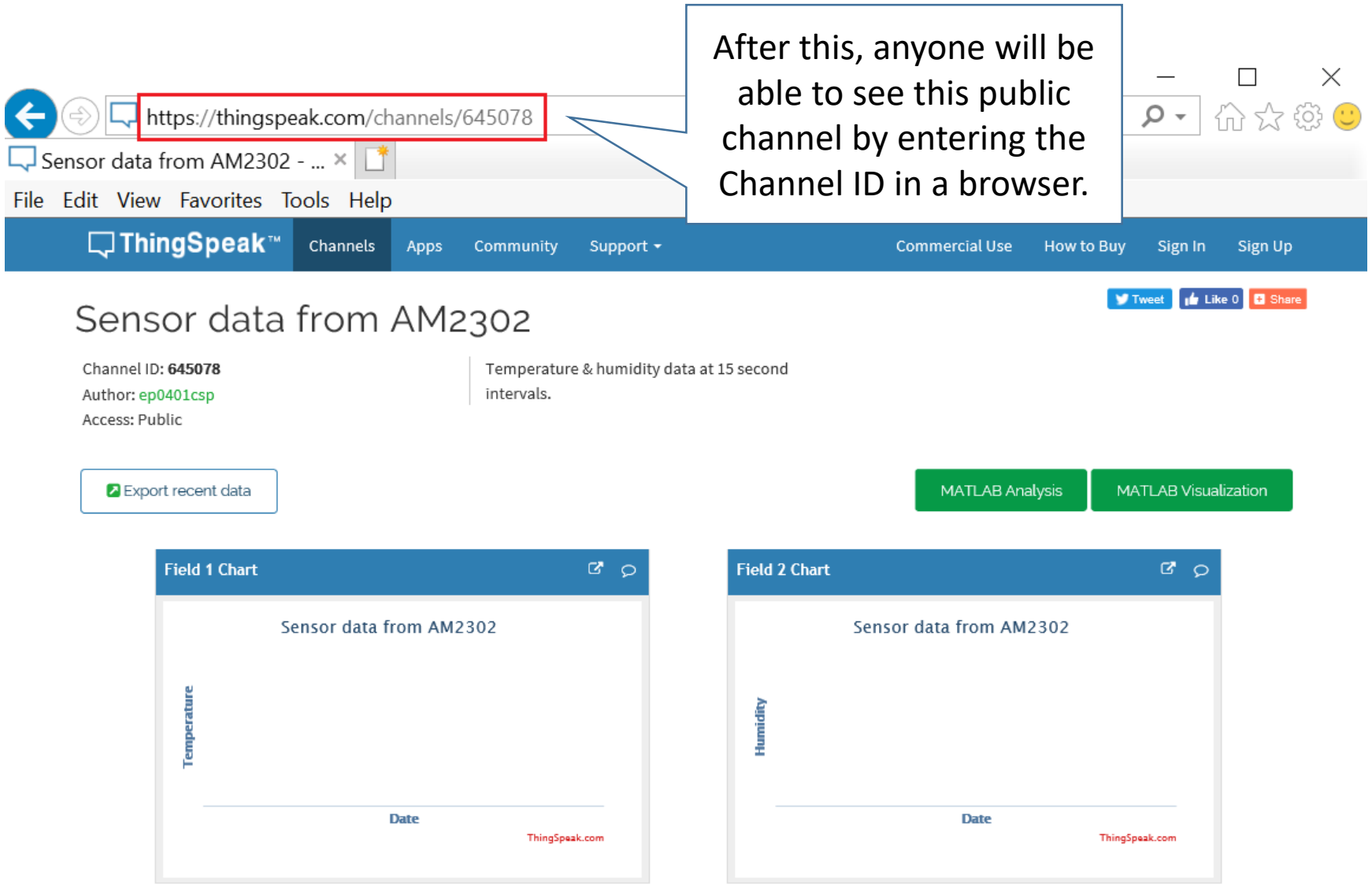
Setting up a Cloud platform (e.g. Thingspeak) (cont.)

Note the Channel ID, in this case 645078.

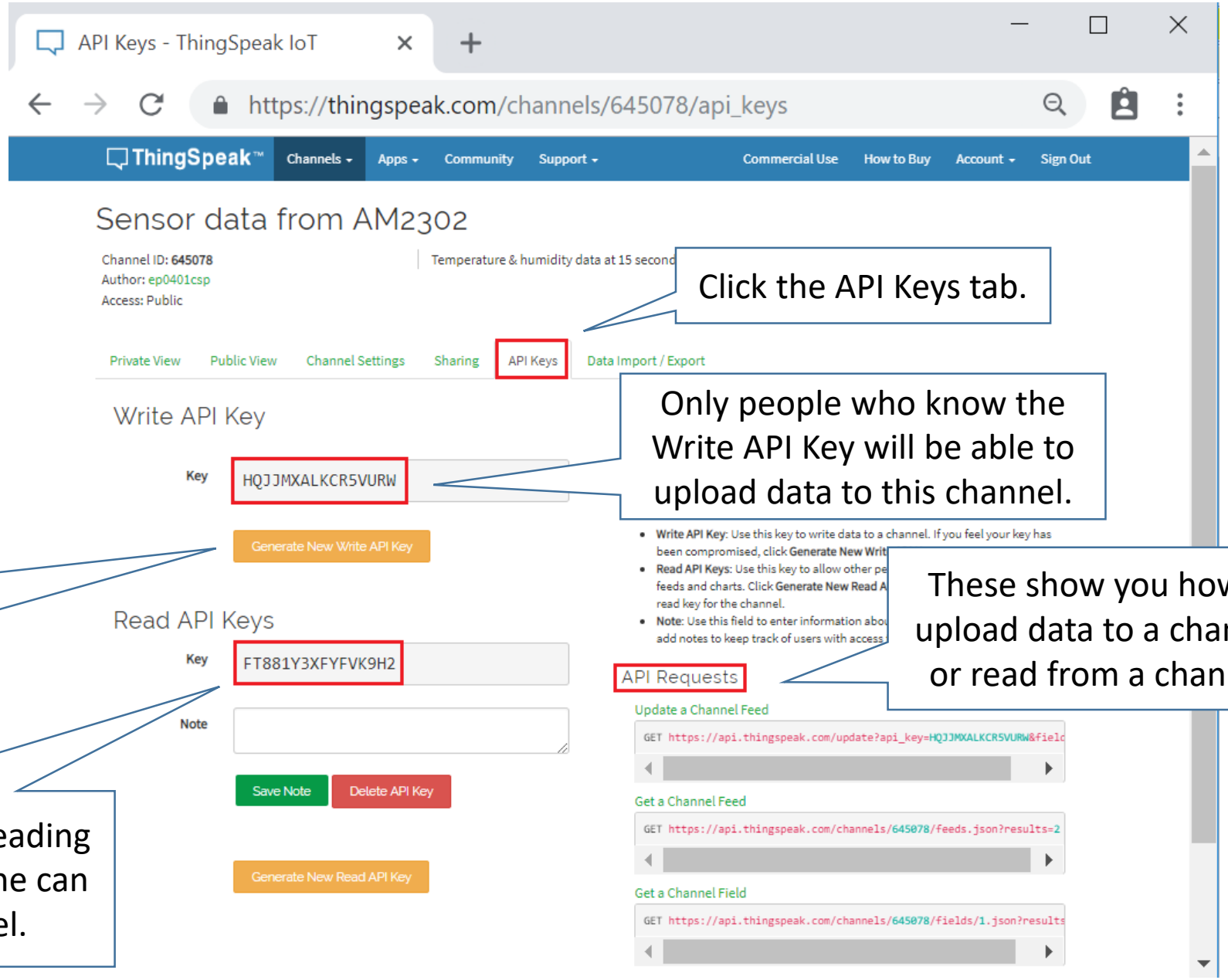
Tick the box “Share channel view with everyone”.



Setting up a Cloud platform (e.g. Thingspeak) (cont.)



Setting up a Cloud platform (e.g. Thingspeak) (cont.)



Click the API Keys tab.

Only people who know the Write API Key will be able to upload data to this channel.

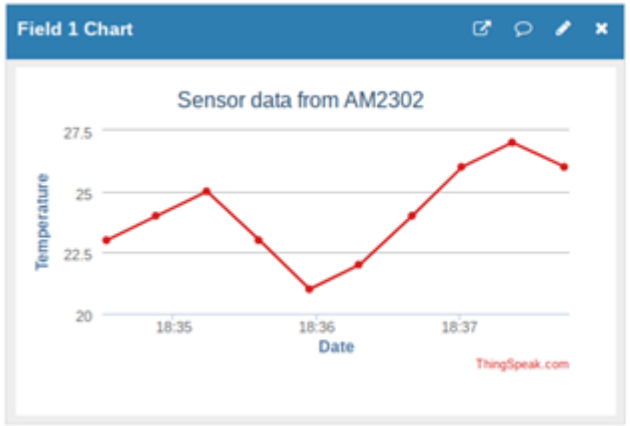
These show you how to upload data to a channel, or read from a channel.

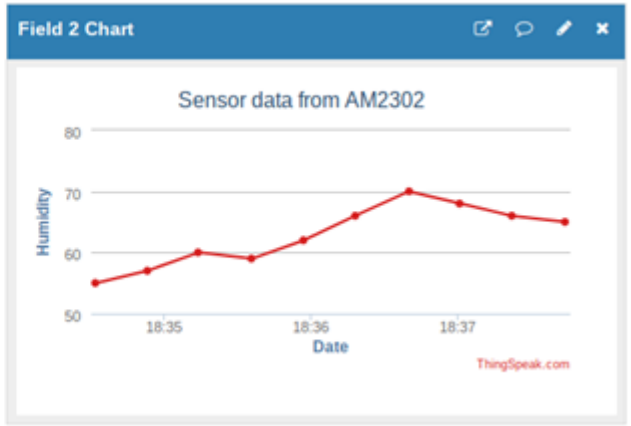
If you think that this "password" has been compromised, you can always generate a new one.

Read API Key is required for reading from a private channel. Anyone can read from a public channel.

Sending data to cloud for storage & visualization

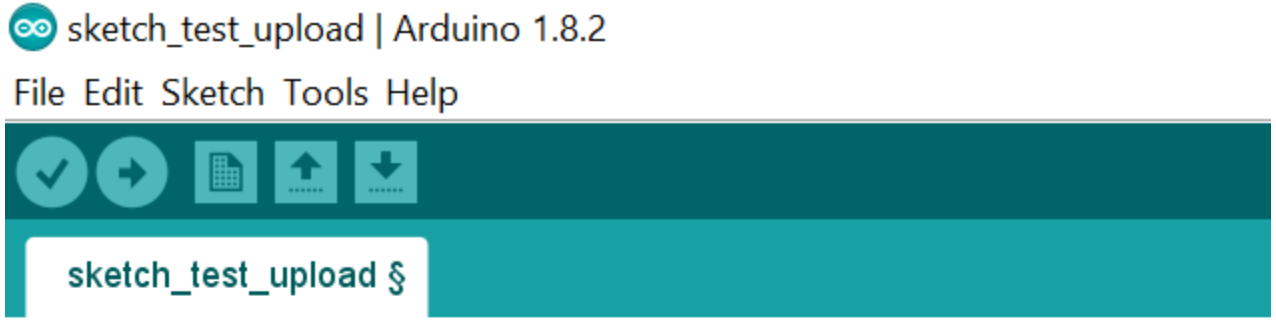
- In Exercise 3.4 (Lesson 3), you have used ESP01 (a WiFi module) and written a program to upload potentiometer readings to a Thingspeak channel.
- In Exercise 2.9 (Lesson 2 part II), you have used DHT11 (a temperature & humidity sensor) and written a program to read from the sensor.
- The code are repeated below for your easy reference.
- You can easily combine these 2 programs, to upload the temperature & humidity readings to a Thingspeak channel. We will not give you the exact program though.





If your project requires uploading sensor data to Thingspeak, you will need to know how to do this.

Sending data to cloud for storage & visualization (cont.)




```
#include <SoftwareSerial.h>

#define DEBUG true

// LED at D13
int ledPin = 13;

// Potentiometer at A0
int potPin = 0;

// replace with your Thingspeak channel's API key!!!
String apiKey = "ABCD1234EFGH"; // *** Change this!

// UNO's D10 connected to ESP's TX
// UNO's D11 connected to ESP's RX via resistor network
SoftwareSerial ESP01(10, 11); // RX, TX
```

uploading

- A program to upload potentiometer readings to a Thingspeak channel.

```
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
  while (!Serial){
  }
  Serial.println("Starting...");
  ESP01.begin(9600);
}
```

Sending data to cloud for storage & visualization (cont.)

```
void loop() {
  // Reset ESP8266, put it into mode 1 i.e. STA only, make it join hotspot / AP,
  // establish single connection
  Serial.println();
  sendData("AT+RST\r\n",2000,DEBUG);
  sendData("AT+CWMODE=1\r\n",2000,DEBUG);
  sendData("AT+CWJAP=\"EEE-IoT\",\"howIknow@07\"\r\n",4000,DEBUG);
   // *** Change these!
  sendData("AT+CIPMUX=0\r\n",2000,DEBUG);

  // Blink LED on board
  digitalWrite(ledPin, HIGH);
  delay(200);
  digitalWrite(ledPin, LOW);
```

uploading

```
// Read potentiometer value
int sensorValue = analogRead(A0); // 10 bit result: 0 - 1023
float voltage = sensorValue * (5.0 / 1023.0); // 0V - 5V
String temp = String(voltage); // convert to string
Serial.println(temp);
```

Replace this part with the code to read from your sensor.

Sending data to cloud for storage & visualization (cont.)

uploading

```
// Make TCP connection
String cmd = "AT+CIPSTART=\"TCP\",\"";
cmd += "184.106.153.149"; // Thingspeak.com's IP address
cmd += "\",80\r\n";
sendData(cmd,2000,DEBUG);
```

```
// Prepare GET string
String getStr = "GET /update?api_key=";
getStr += apiKey;
getStr +="&field1=";
getStr += temp;
getStr += "\r\n";
```

This part need to be changed, if you are sending more than one value at a time.

```
// Send data length & GET string
ESP01.print("AT+CIPSEND=");
ESP01.println (getStr.length());
Serial.print("AT+CIPSEND=");
Serial.println (getStr.length());
delay(500);
if( ESP01.find( ">" ) )
{
  Serial.print(">");
  sendData(getStr,2000,DEBUG);
}

// Close connection, wait a while before repeating...
sendData("AT+CIPCLOSE",16000,DEBUG); // thingspeak needs 15 sec delay between updates
}
```

GET /update?api_key=ABCD…1234&field1=value1&field2=value2\r\n

Sending data to cloud for storage & visualization (cont.)

- A program to read temperature & humidity from DHT11.

DHT11_serial_monitor | Arduino 1.8.2

File Edit Sketch Tools Help

DHT11_serial_monitor §

```
// DHT11 (temperature humidity sensor) sample code

#include "DHT.h"
DHT dht(2, DHT11); // DHT22 for DHT22/AM2302/AM2321, or DHT21 for DHT21/AM2301
    // data pin connected to pin 2

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  delay(2000); // 2 seconds between samples
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print("%  Temperature: ");
  Serial.print(t);
  Serial.println("°C ");
}
```

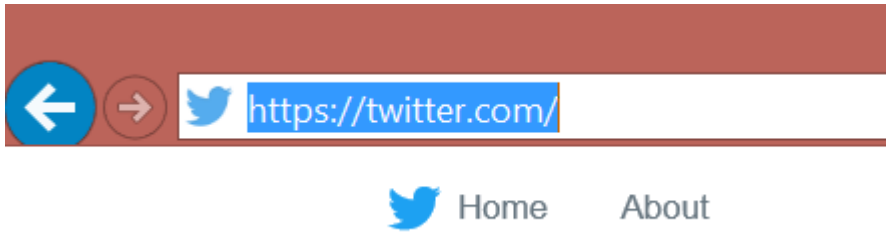
reading sensor

These parts need to be added to the previous program, if you are reading from a DHT11.

Sending an event-triggered notification, through Twitter

- This part shows you how to sent a tweet, when a pre-defined condition is met, using (Thingspeak's) ThingTweet and React apps.

- If you don't have a Twitter account, create one at Twitter.com:



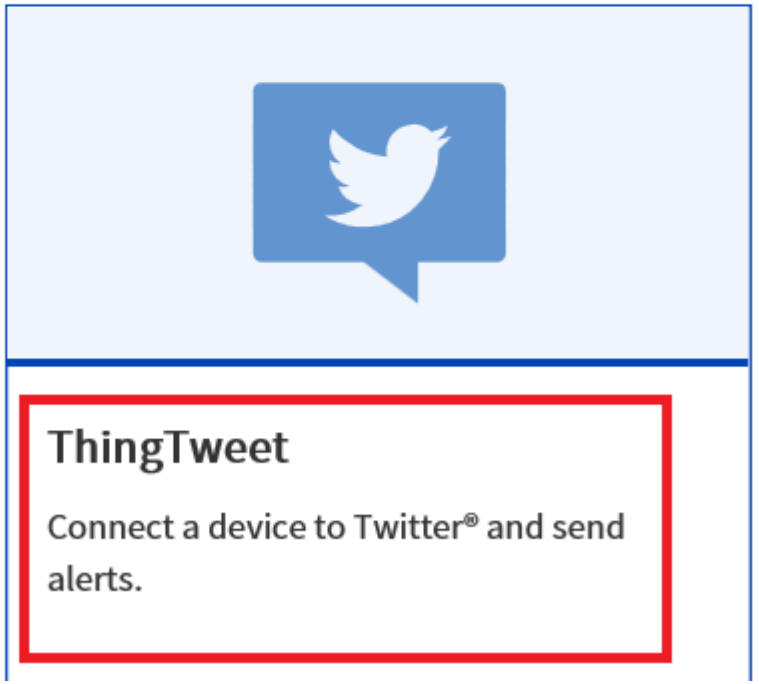
The material presented here is based on the tutorial at:

https://www.mathworks.com/help/thingspeak/act-on-your-data.html

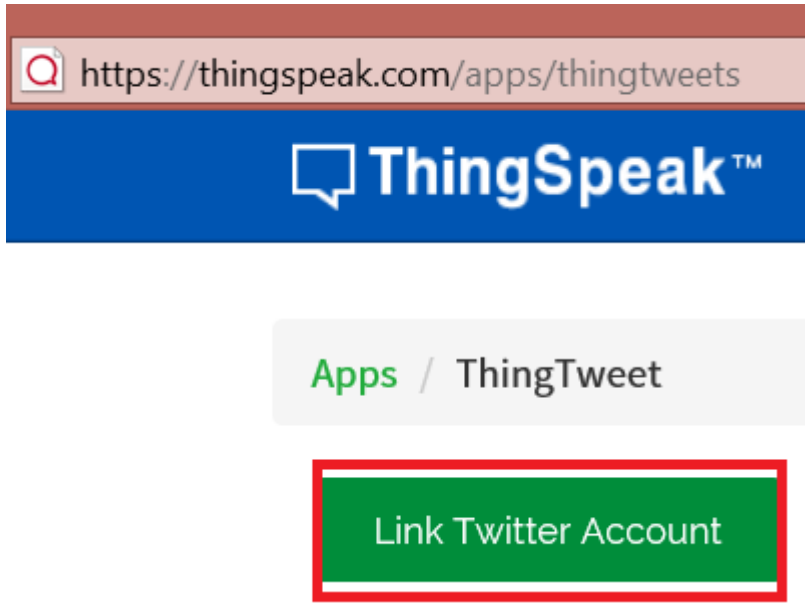
Sending an event-triggered notification, through Twitter (cont.)

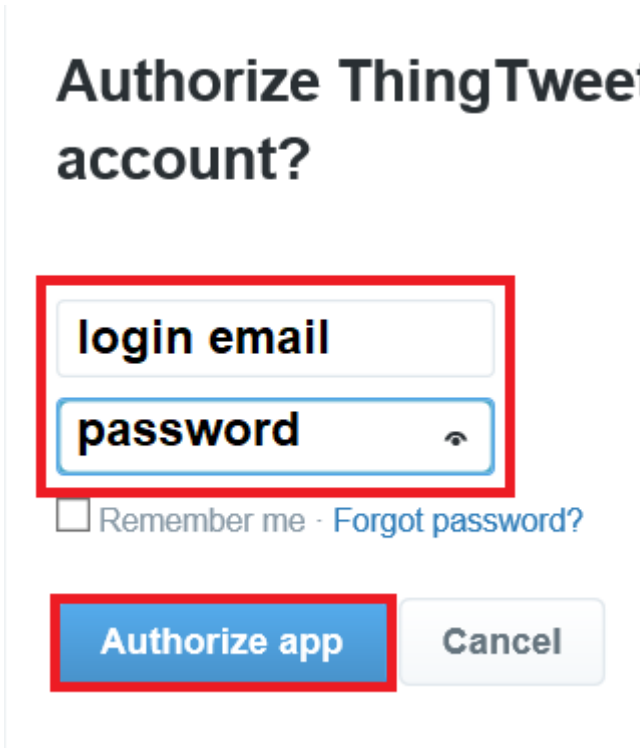
- Link your Twitter account to your Thingspeak account as follows:

Click Apps, followed by ThingTweet…



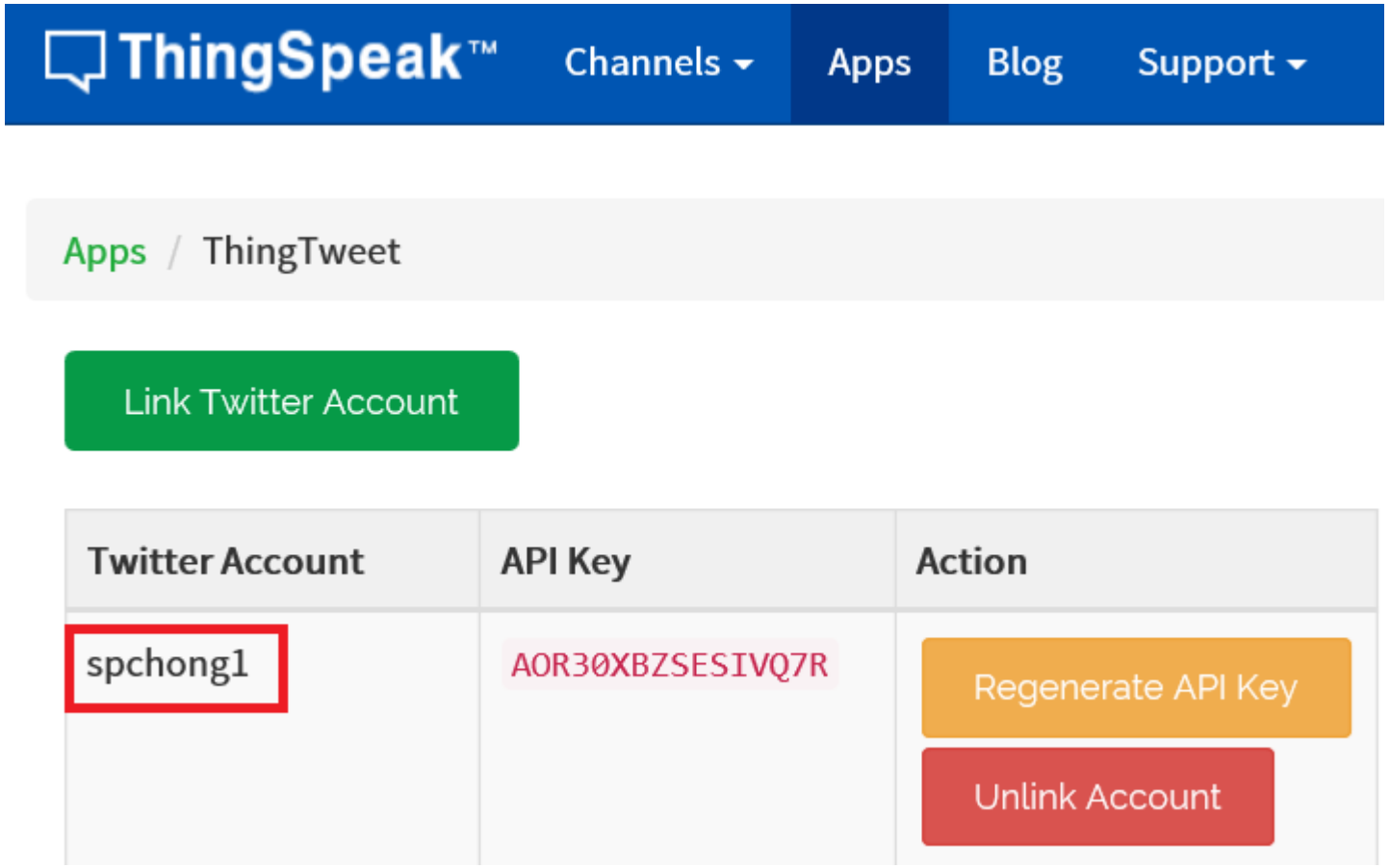
Sending an event-triggered notification, through Twitter (cont.)





Click Link Twitter Account, enter the login details, then click Authorize app.

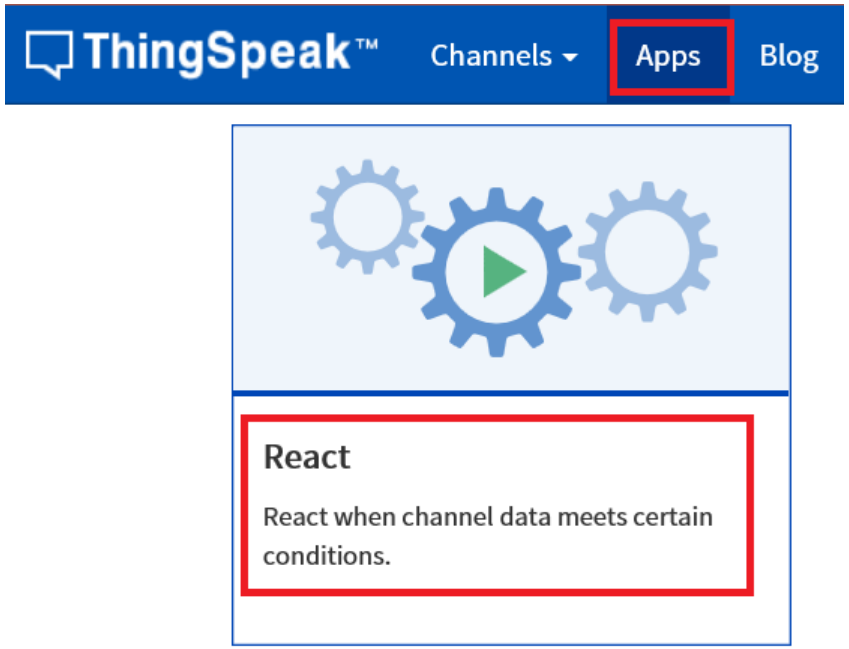
Sending an event-triggered notification, through Twitter (cont.)

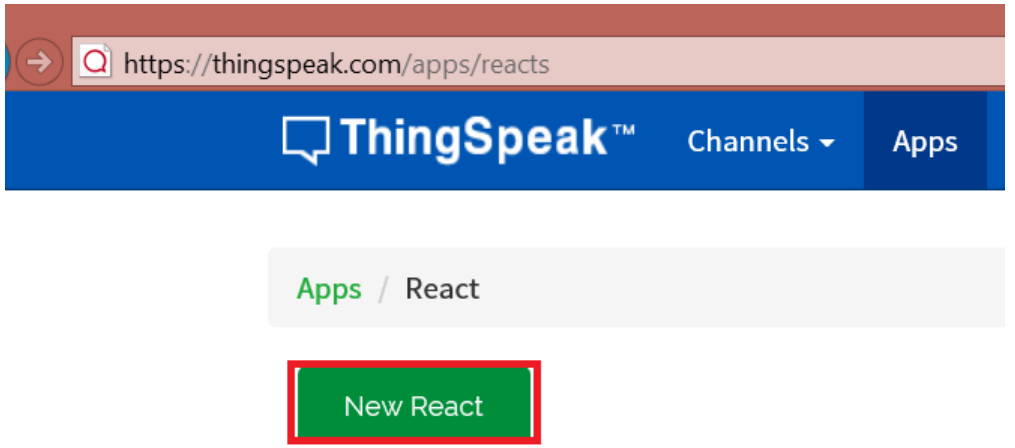


(Back to ThingTweet…) This shows that the Twitter Account has been linked to Thingspeak.

Sending an event-triggered notification, through Twitter (cont.)

- Set up a “React”, a mechanism whereby a tweet will be sent automatically, if pre-defined conditions are met, as follows:



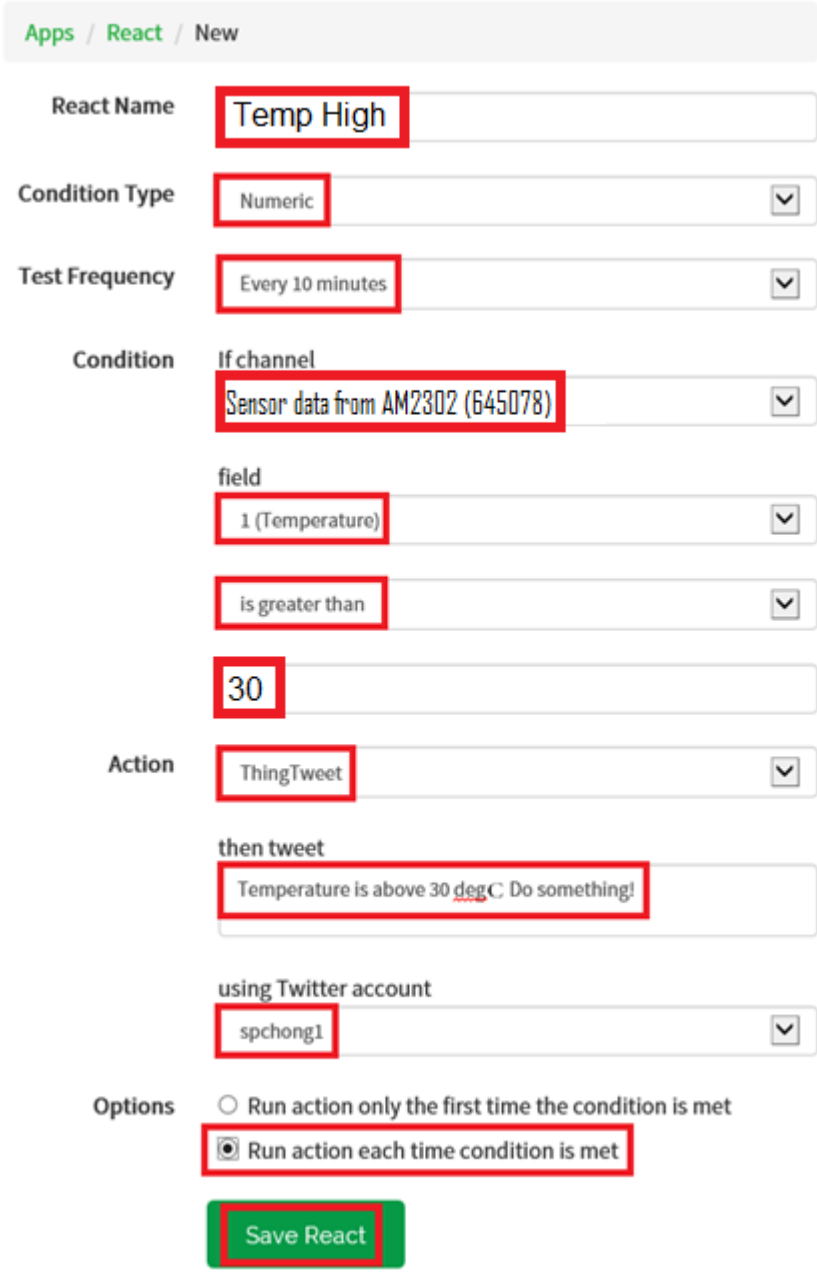


Click Apps, followed by React, then New React…

Sending an event-triggered notification, through Twitter (cont.)

- The temperature data is checked every 10 minutes.
- If it exceeds 30 deg Celsius, ThingTweet will inform us via a tweet.

Fill in the details, then click Save React.



Sending an event-triggered notification, through Twitter (cont.)

- Someone who follows this will be informed, if the pre-defined condition(s) is (are) met.

Lab Exercises

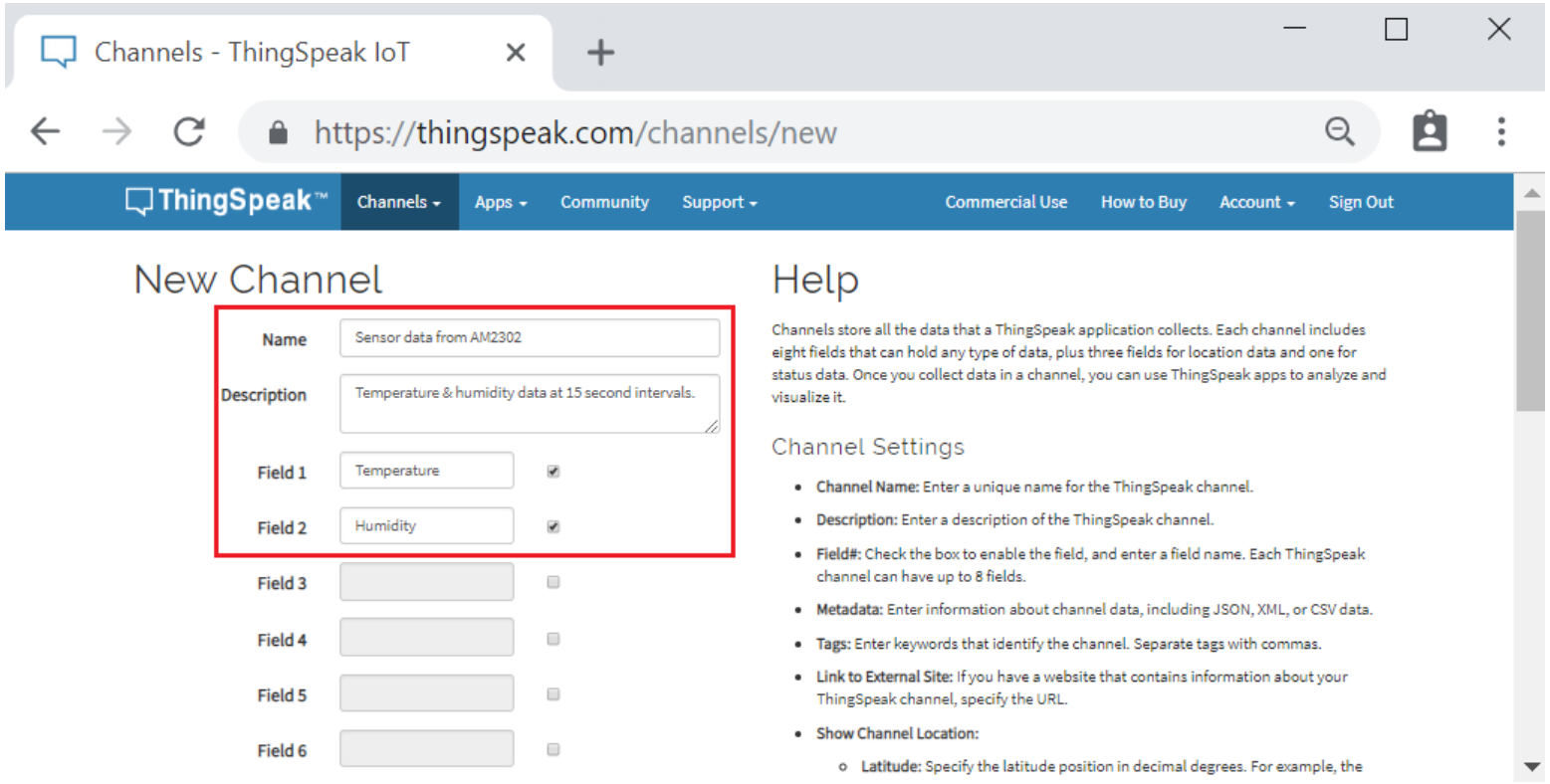
These exercises can be done (as a project team) according to your project requirements.

- Exercise 5.1 – Setting up your Thingspeak channel
- Exercise 5.2 – Uploading sensor readings to Cloud
- Exercise 5.3 – Linking up Thingspeak with your Twitter account
- Exercise 5.4 – Sending a tweet

Exercise 5.1 – Setting up your Thingspeak channel

Ref: slides 6 to 16.

Set up your Thingspeak account. After that, create a channel with one or more fields, for sensor data upload, depending on your project requirements.

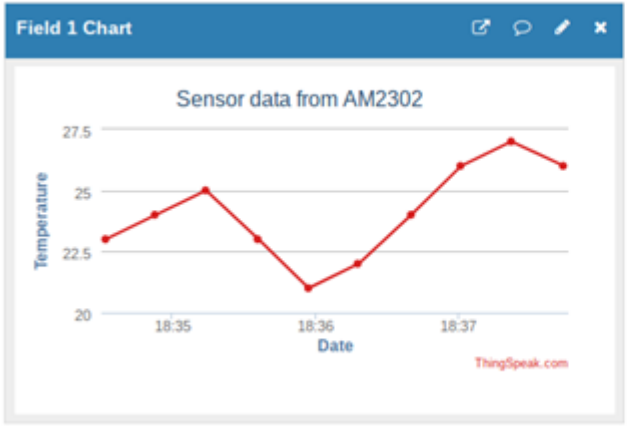


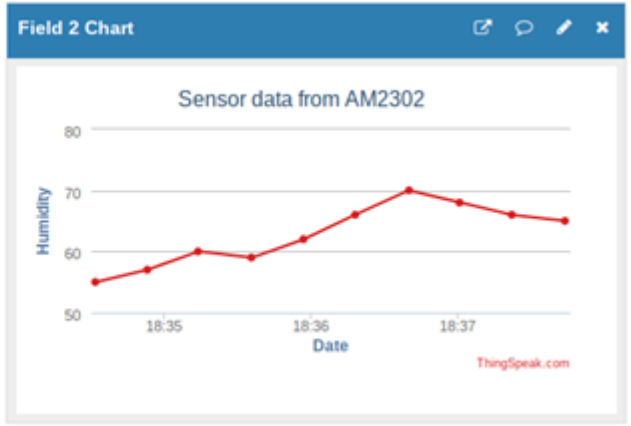
A project with temperature & humidity sensor readings should have 2 fields like this.

Exercise 5.2 – Uploading sensor readings to Cloud

Ref: slides 7 to 21.

Write an Arduino program to send the sensor readings (depending on your project requirements) to your Thingspeak channel, once every 20 seconds.





Two charts from a project that uploads temperature & humidity readings.

Exercise 5.3 – Linking up Thingspeak with your Twitter account

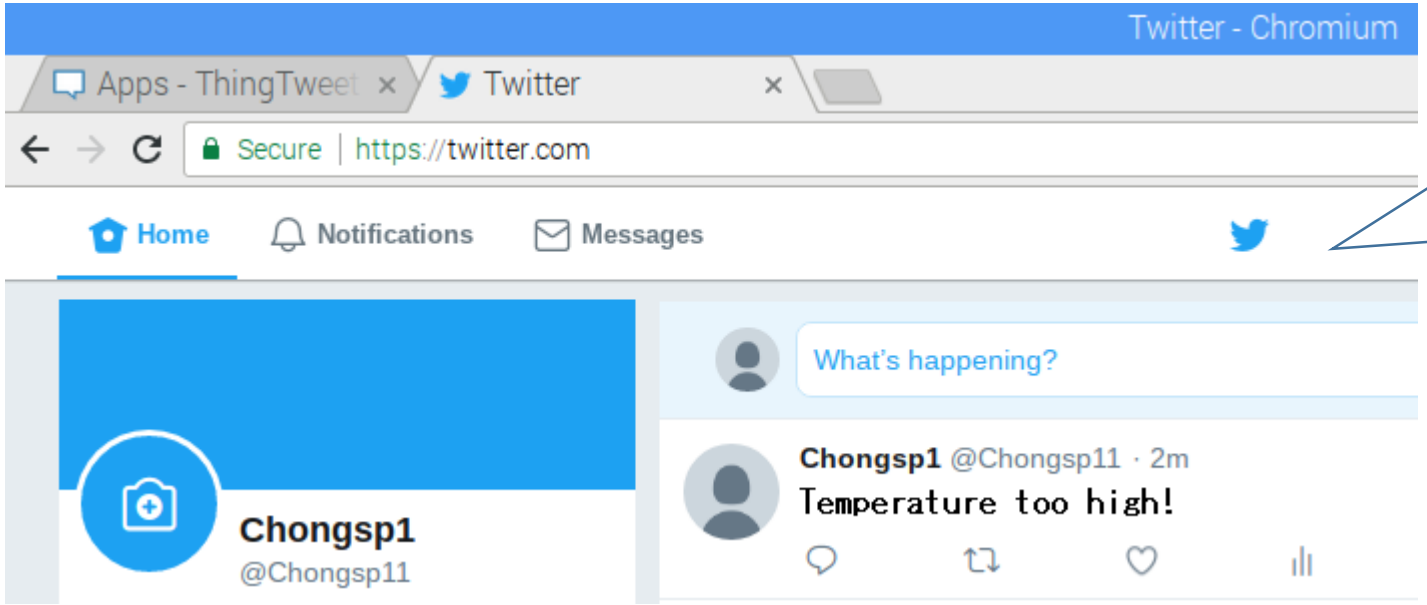
Ref: slides 22 to 25.

Create a Twitter account if you don't have one. Link it to your Thingspeak account.

Exercise 5.4 – Sending a tweet

Ref: slides 26 to 28.

Set up your Thingspeak's React, so that a tweet is sent when a certain condition is met.



A tweet from a project that requires someone to be notified when the temperature is too high.


Thank you