

[← Go Back](#)

Cloud

Guides

[Getting Started with
Arduino Cloud](#)[Cloud Editor](#)[Arduino / C++](#)[ESP32 / ESP8266](#)[MicroPython](#)[Python](#)[JavaScript](#)[Node-RED](#)[Alexa](#)[Google Home](#)

Hardware & Devices

Cloud Interface

Cloud Editor

IoT Remote App

Features

API

Arduino Cloud CLI

Cloud Business

Cloud Education

Application Notes

[Home](#) / [Cloud](#) / **Node-RED**

ON THIS PAGE

Node-RED

Learn how to use Node-RED together with the Arduino Cloud.


Author · [Liam Aljundi](#) · Last revision · 07/23/2025

Introduction

Node-RED is a programming tool for connecting hardware devices such as Arduino with other hardware devices, APIs, and online services easily using a web-based flow editor. It allows you to connect those different devices and services by connecting a combination of nodes that create your desired flow.

In this tutorial, we will look into the applications of using Node-RED together with the Arduino Cloud. The integration of the two platforms allows us to facilitate communications between the Arduino Cloud and home automation devices, send and receive data from online services such as Email and SMS, and write JavaScript code to manipulate the data.

Note: this tutorial requires a [subscription to the](#)

 [Arduino Cloud](#), all three subscriptions (Entry, Maker & Maker Plus) are valid.

Introduction

[Goals](#)[Hardware & Software
Requirements](#)[Node-RED Setup](#)[The Node-RED Editor](#)[Setting Up a Thing](#) +[Communicating with Node-RED](#)[Further Applications](#)[Help](#)

Goals

Set up Node-RED with the Arduino Cloud.

Use simple Flows with Node-RED.

Build a Node-RED Flow that pulls data from Arduino Cloud.

Manipulate data received from the Arduino Cloud using JavaScript code.

Send email notifications with data from the Arduino Cloud.

Hardware & Software Requirements

[Arduino Create Agent](#)

[Node.js](#) and [Node-RED](#) installed

An [Arduino account](#).

For this tutorial, you will need a Cloud compatible board. You will find all compatible boards in the link below:

[Arduino Cloud compatible hardware](#).

Node-RED Setup

Setting up Node-RED is simple, we will run it locally following the steps below:

1. Install [Node.js](#). To check if Node is installed, open a

terminal and enter the following command:

```
1 node --version
```

If correctly installed, you will see the version number printed in the terminal.

2. Install Node-RED from the command line using the command

```
sudo npm install -g --  
unsafe-perm node-red
```

, delete `sudo` if you are using a Windows machine.

3. After installation, start it by running the command `node-red` in the terminal, you should see a similar output to this:

```
1 $ node-red  
2  
3   Welcome to Node-RED  
4   =====  
5  
6   30 Jun 23:43:39 -  
7   30 Jun 23:43:39 -  
8   30 Jun 23:43:39 -  
9   30 Jun 23:43:39 -  
10  30 Jun 23:43:44 -  
11  30 Jun 23:43:44 -  
12  30 Jun 23:43:44 -  
13  30 Jun 23:43:44 -  
14  30 Jun 23:43:44 -  
15  30 Jun 23:43:44 -  
16  30 Jun 23:43:44 -  
17  30 Jun 23:43:44 -  
18  30 Jun 23:43:44 -  
19  30 Jun 23:43:44 -
```

4. Open the Node-RED editor by going to your browser and entering

```
http://localhost:1880 .
```



For more details on installing Node-RED, you can check [their installation page](#).

The Node-RED Editor

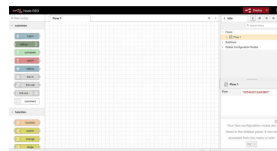
The Node-RED editor consists of four main parts:

A **header on the top** containing the Deploy button, main menu, and the user menu (only visible if user authentication is enabled).

The **palette on the left side**, containing the available nodes.

A **workspace in the middle**, where flows can be created.

The **sidebar on the right**, containing editing tools such as a node configuration tool and a debugger.



The Node-RED editor

You can run the simple flow shown below using Node-RED's default nodes:

• • • • •

side into the workspace in the middle

double-click on the node to edit it

assign a name and topic to it

click on the dropdown menu next to *msg.payload* and choose "**string**", then enter a random message

click **Done**

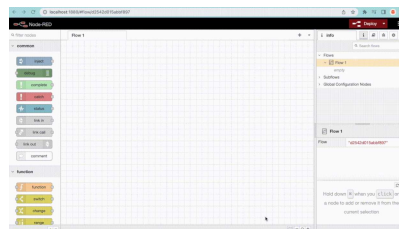
drag the "**debug**" node into the workspace

connect the two nodes by dragging a wire from the message node to the debug node

click on the debug menu from the sidebar on the right

press **Deploy** from the header on the top

finally, press on the checkbox of the message node

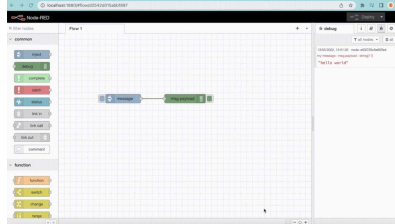


Your message should be printed to the console on the right side.

In addition to the default nodes installed in node-RED, you can use the palette manager to install additional nodes that can be useful to creating more advanced flows. Follow the steps below to install the Arduino Cloud nodes using the palette manager:

1. Click on the menu in the header bar in the top right

2. Select **"Manage palette"**.
3. Go to the **"Install"** tab.
4. Search for **"Arduino"**.
5. From the results, install **"@arduino/node-red-contrib-arduino-iot-cloud"**.



Now, you should be able to use the Arduino Cloud nodes from the palette on the left side of the editor. There are five Arduino Cloud nodes: property in, property out, historic, periodic, and inject. Each of the nodes has a description of its function, which appears when you hover over the node.

Setting Up a Thing

The Thing we will create for this example is quite simple. It is a sketch to read humidity values from a **DHT11 humidity sensor**, any other values can be used alternatively, such as temperature, soil moisture, light etc. Keep in mind that the code and example below are specific to using the DHT11 humidity sensor.

Follow the steps below to set up the Thing:

Go to the Arduino Cloud ->

Add your Device and your Network credentials, which can be changed on the device page

Add an integer Variable, with the name humidity, and set permissions to Read Only

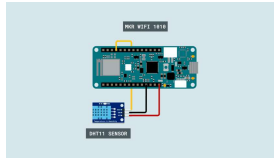
Go to the **"Sketch"** tab and replace the code with the code below:

```
1  #include "thingProperties.h"
2  #include <Arduino_MKRIoTCarrier.h>
3  MKRIoTCarrier carrier;
4
5  #include <dht.h> /
6  dht DHT;
7  #define DHT11_PIN
8
9  void setup() {
10 // Initialize seri
11   Serial.begin(960
12 // This delay gi
13   delay(1500);
14
15 // Defined in th
16   initProperties()
17
18 // Connect to Ar
19   ArduinoCloud.beg
20
21   setDebugMessageL
22   ArduinoCloud.pri
23
24   carrier.begin();
25 }
26
27 void loop() {
28   ArduinoCloud.upd
```

Circuit

The final step is connecting the **DHT11 humidity sensor** to our Arduino MKR Wi-Fi 1010 board, as shown in the image below

Arduino Cloud compatible board and a different sensor.



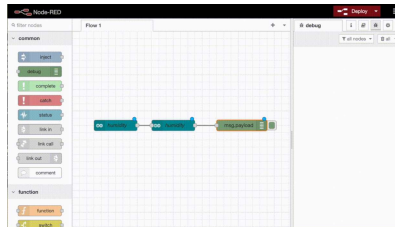
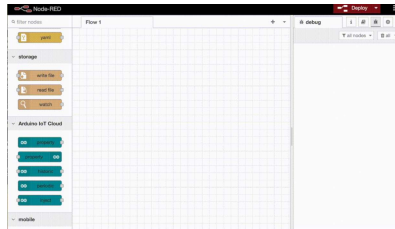
Connecting the DHT11 Sensor.

Communicating with Node-RED

Use the steps below to use Node-RED with the Arduino Cloud:

1. Go to the [API keys section](#), and create a new API key.
2. Save the Client ID and Client Secret in a safe document
3. Go to Node-RED web page at <http://localhost:1880>
4. Add the Arduino Cloud **"property out"** node from the palette menu on the left side
5. Double click on the node -> select "Add new Arduino-connection"
6. Click on the pen icon to enter the API Key details -> enter the Client ID and Client Secret saved in the previous steps -> click on the ADD button
7. You should now be able to select any of your Things and properties (Variables). In our case, we select the **Humidity**

(Variable) to retrieve data values from.



Creating a Flow

Now that you have a node pulling data from the Arduino Cloud, you need to use the *Inject* node in order to manipulate and send the data. All you need to do is dragging the *Inject* node and connect it to the property out node, then double-clicking on the inject node to ensure that the right Thing and property are selected.

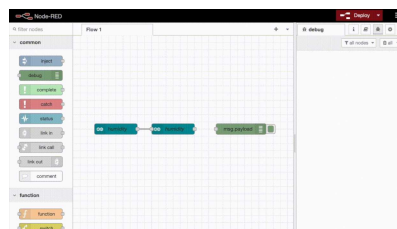
Finally, we will add a debug node from the top of the palette menu, to print the values coming from the Arduino Cloud into the console. You will now be able to see the values from the humidity sensor printed into the Node-RED console.

The next step is adding a **function node** that processes the data coming from the Arduino Cloud. This node is important when working with a constant flow of data, as it allows

the received values. After dragging the function node into the editor, double-click on the node and select **"on message"** tab, then insert the code below:

```
1 let data = msg.payload
2 let trigger = context
3
4 if(typeof trigger ==
5 trigger = false;
6
7 if(data>60 && !trigge
8     trigger = true;
9     context.set("statu
10     return msg;
11 }
12
13 if(data<60 && trigger
14     trigger = false;
15     context.set("statu
16     return msg;
17 }
```

This code will only send values when they are above 60, or when they change back to below 60. You can connect it to the flow, in between the humidity inject node, and the debug node. Try it out by blowing air on the humidity sensors to increase the values.



Sending Email Notifications

Arduino Cloud, it's time to use this data. In this example, we will show how to send alarm notifications to your email, using the Arduino Cloud. To achieve this, we need to add an email nodes following the steps below:

Click on the menu in the header bar in the top right corner.

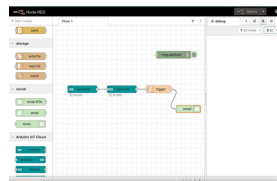
Select **"Manage palette"**.

Go to the "install" tab.

Search for "Email".

From the shown results, install **"node-red-node-email"**.

Next, drag the email node from the palette menu on the left side into the editor. Double-click on the node, then enter an email to send the data to, as well as your email information to send the data from.



Email node

Now, all you need to do is replace the debug node with the email node to receive email notifications with the humidity data received from the Arduino Cloud. You can customize the emails using the function node if you wish.

Further

In this tutorial we demonstrated how you can use the [Arduino Cloud](#) together with [Node-RED](#) to build a more complex automation applications. The email notification is only one of the many different use cases, other applications can be creating communication between the Arduino Cloud to other home automation devices, sending customized SMS and WhatsApp messages, and many more.

Suggest changes Need support?

The content on [docs.arduino.cc](#) is facilitated through a public [GitHub repository](#). If you see anything wrong, you can edit this page [here](#).

[Help Center](#)
[Ask the Arduino Forum](#)
[Discover Arduino](#)
[Discord](#)

License

The Arduino documentation is licensed under the [Creative Commons Attribution-Share Alike 4.0](#) license.

Was this article helpful?

