

Develop a web application using Node RED to view data separately

Date	4th November 2022
Team ID	PNT2022TMID49483
Project Name	Smart Farmer- IoT Enabled Smart Farming Application

Step 1:

The screenshot shows a web browser window with the URL `node-red-kyyek-2022-11-06.eu-gb.mybluemix.net`. The page features a red header and a light gray background. The main content area includes the following text:

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

More information about Node-RED, including documentation, can be found at nodered.org.

On the right side, there is a red button that says "Go to your Node-RED flow editor" and a link that says "Learn how to customise Node-RED".

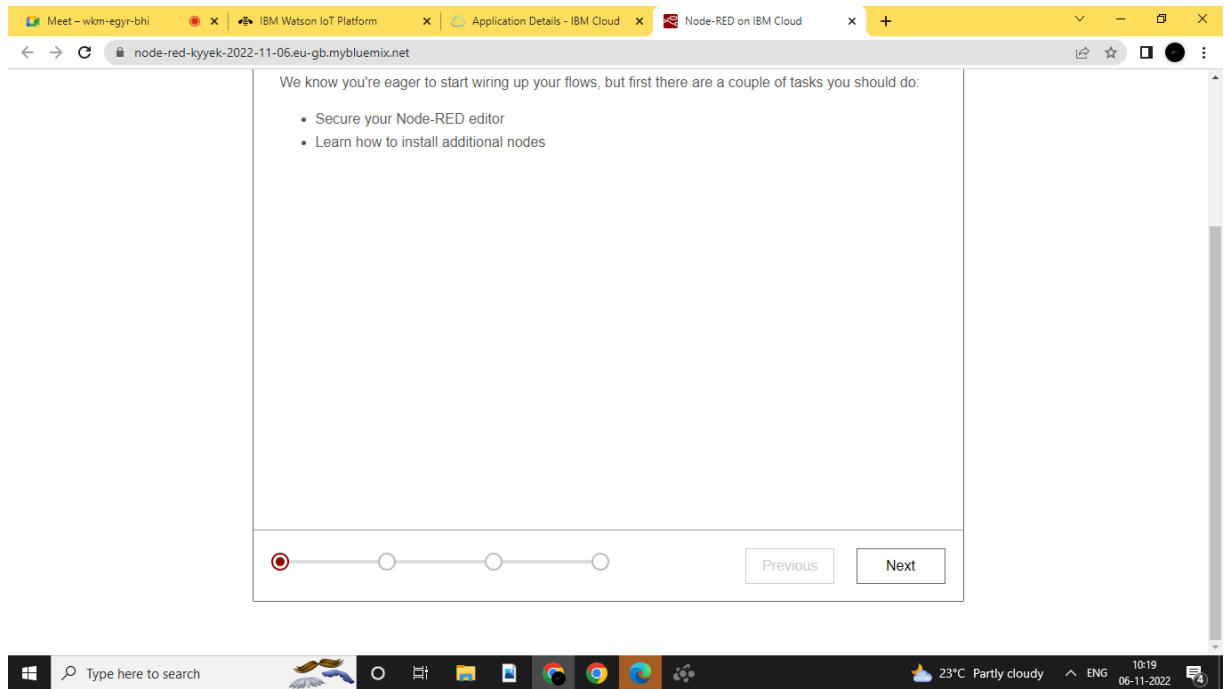
Below the main content area, there is a section titled "Customising your instance of Node-RED" with the following text:

This instance of Node-RED is enough to get you started creating flows.

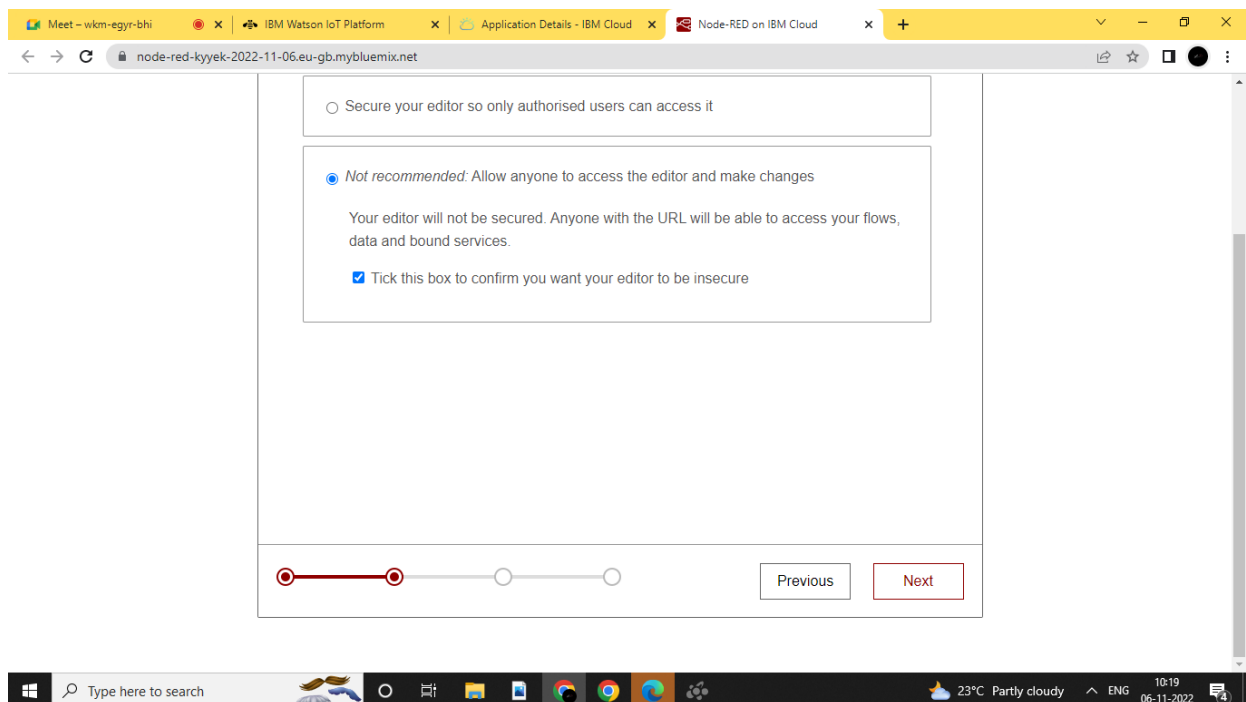
You may want to customise it for your needs, for example replacing this introduction page with your own, adding http authentication to the flow editor or adding new nodes to the palette.

The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system tray on the right indicates a temperature of 23°C, partly cloudy weather, and the date 06-11-2022.

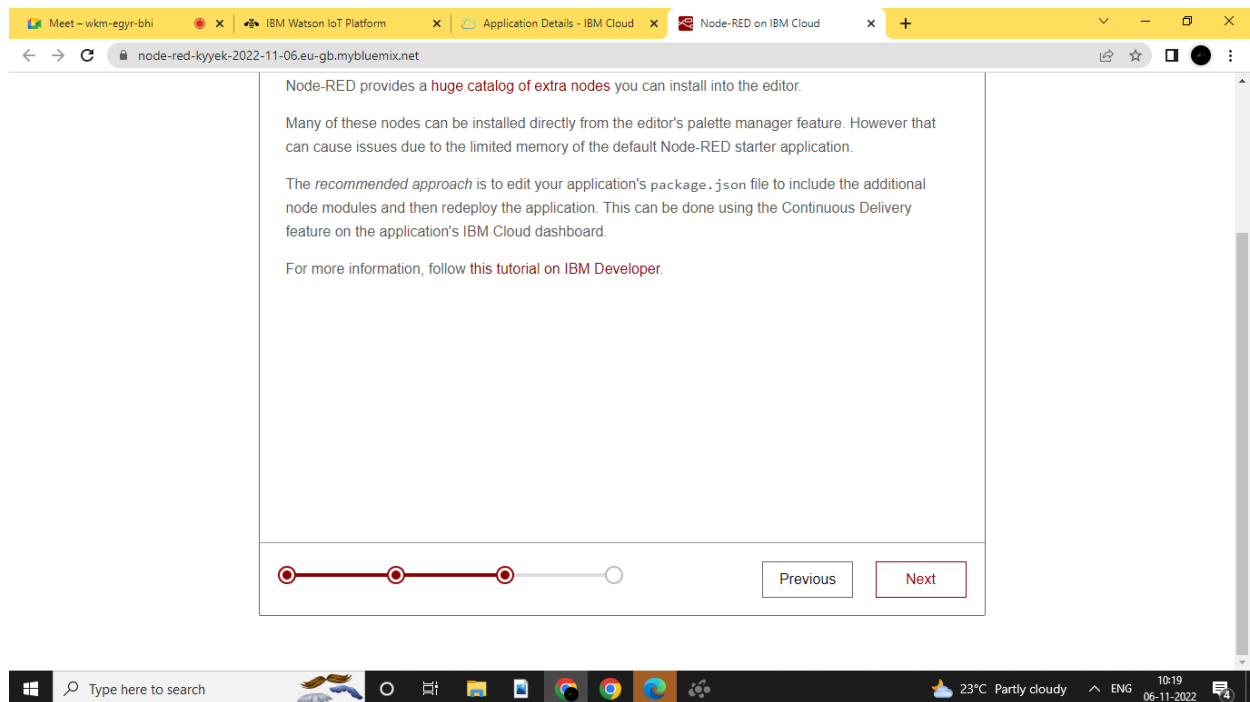
Step 2:



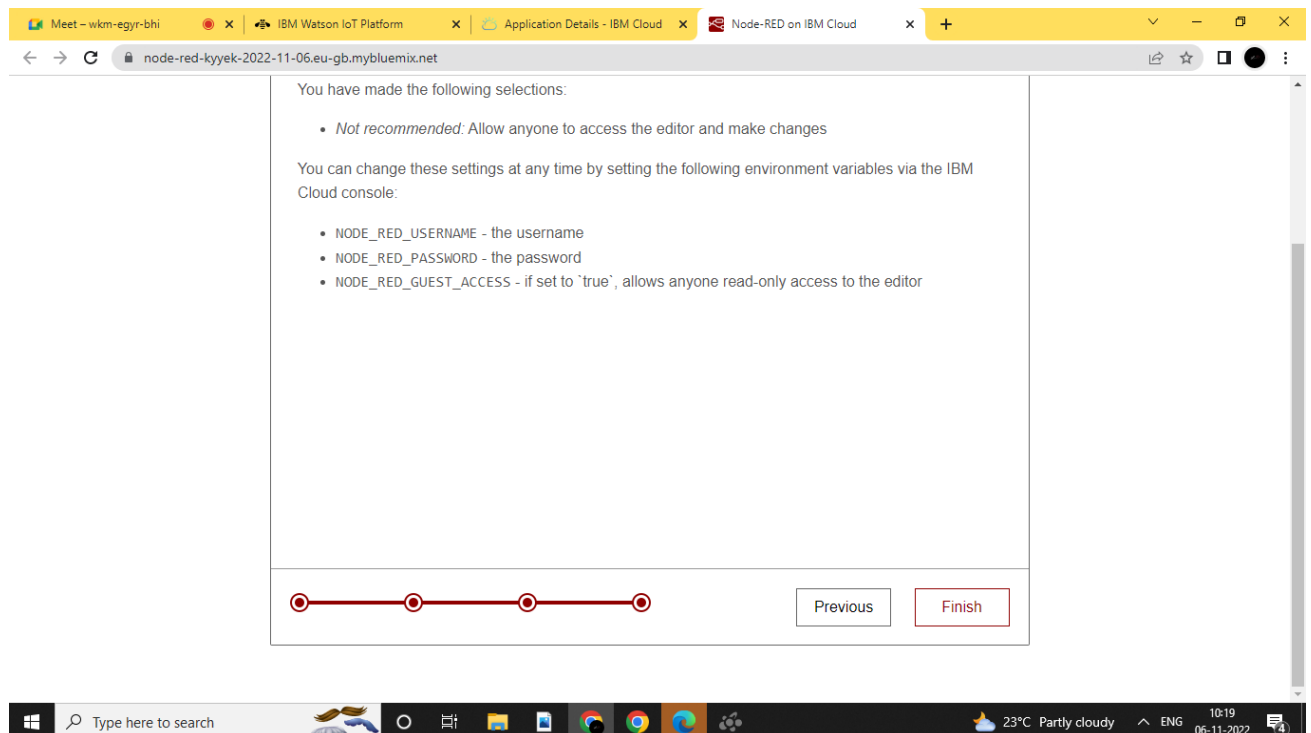
Step 3:



Step 4:



Step 5:



Step 6:

The screenshot shows a web browser window with the URL `node-red-kyyek-2022-11-06.eu-gb.mybluemix.net`. The page has a red header and a light gray body. It contains the following text:

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

More information about Node-RED, including documentation, can be found at nodered.org.

On the right, there is a red button that says "Go to your Node-RED flow editor" and a link below it that says "Learn how to customise Node-RED".

Below the main content, there is a section titled "Customising your instance of Node-RED" with the following text:

This instance of Node-RED is enough to get you started creating flows.

You may want to customise it for your needs, for example replacing this introduction page with your own, adding http authentication to the flow editor or adding new nodes to the palette.

The browser's taskbar at the bottom shows the Windows search bar and several open applications, including IBM Watson IoT Platform and Application Details - IBM Cloud.

Step 7:

The screenshot shows the Node-RED flow editor interface. The main workspace is a grid where two nodes are connected: an "IBM IoT" node (blue) and a "msg.payload" node (orange). The left sidebar contains a "filter nodes" search bar and two categories of nodes: "common" (inject, debug, complete, catch, status, link in, link call, link out, comment) and "function" (function, switch). The right sidebar shows the "info" panel with a search bar and a list of flows, including "Flow 1". Below the list, the "msg.payload" node is selected, showing its details: Node ID "1145e42a92c45d9b" and Type "debug". A tooltip at the bottom of the right sidebar says "ctrl-space will toggle the view of this sidebar". The top of the interface shows a "Deploy" button and a menu icon. The browser's taskbar at the bottom shows the Windows search bar and several open applications, including IBM App Dev, Application, Node-RED, and IBM Watson IoT Platform.

Step 8:

The screenshot shows the Node-RED web interface in a browser. The flow consists of an **IBM IoT** node (labeled 'connected') connected to two **function** nodes. The output of these function nodes is connected to a **msg.payload** node. The **debug** console on the right displays a series of messages received from the IoT node, each containing a timestamp, node ID, and a payload object with temperature and humidity values.

```
msg payload : Object
  { Temperature: 45, Humidity: 28 }
11/6/2022, 9:43:41 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload : Object
  { Temperature: 90, Humidity: 67 }
11/6/2022, 9:43:44 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload : Object
  { Temperature: 78, Humidity: 82 }
11/6/2022, 9:43:47 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload : Object
  { Temperature: 98, Humidity: 43 }
11/6/2022, 9:43:50 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload : Object
  { Temperature: 1, Humidity: 55 }
11/6/2022, 9:43:53 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload : Object
  { Temperature: 32, Humidity: 35 }
```

Step 9:

The screenshot shows the Node-RED web interface with the **Edit function node** dialog box open. The dialog displays the function code: `1 msg.payload=msg.payload.Humidity` and `2 return msg;`. The **Properties** section shows the **Name** as **Humidity**. The **debug** console on the right displays a series of messages received from the IoT node, each containing a timestamp, node ID, and a payload object with temperature and humidity values.

```
11/6/2022, 9:44:16 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload : Object
  { Temperature: 23, Humidity: 47 }
11/6/2022, 9:44:19 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload : Object
  { Temperature: 73, Humidity: 1 }
11/6/2022, 9:44:22 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload : Object
  { Temperature: 47, Humidity: 84 }
11/6/2022, 9:44:25 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload : Object
  { Temperature: 15, Humidity: 90 }
11/6/2022, 9:44:28 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload : Object
  { Temperature: 85, Humidity: 18 }
11/6/2022, 9:44:31 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
```

Step 10:

The screenshot shows the Node-RED web interface. On the left, the 'common' node palette is visible. In the center, the 'Flow 1' workspace contains a single 'IBM IoT' node. To the right, the 'Edit function node' panel is open, showing a JavaScript function:

```
1 msg.payload=msg.payload.Temperature
2 return msg;
```

The 'Properties' section of the function node is set to 'Name' with the value 'Temperature'. Below the function editor, the 'Enabled' checkbox is checked. On the far right, the 'debug' console displays a log of message payloads:

```
msg.payload : Object
  { Temperature: 45, Humidity: 28 }
11/6/2022, 9:43:41 AM node: f2f2649a.0d0d98
iot-2/typeNodeMCU/id/24680/evt/event_1/fmt/json :
msg.payload : Object
  { Temperature: 90, Humidity: 67 }
11/6/2022, 9:43:44 AM node: f2f2649a.0d0d98
iot-2/typeNodeMCU/id/24680/evt/event_1/fmt/json :
msg.payload : Object
  { Temperature: 78, Humidity: 82 }
11/6/2022, 9:43:47 AM node: f2f2649a.0d0d98
iot-2/typeNodeMCU/id/24680/evt/event_1/fmt/json :
msg.payload : Object
  { Temperature: 98, Humidity: 43 }
11/6/2022, 9:43:50 AM node: f2f2649a.0d0d98
iot-2/typeNodeMCU/id/24680/evt/event_1/fmt/json :
msg.payload : Object
  { Temperature: 1, Humidity: 55 }
11/6/2022, 9:43:53 AM node: f2f2649a.0d0d98
iot-2/typeNodeMCU/id/24680/evt/event_1/fmt/json :
msg.payload : Object
  { Temperature: 32, Humidity: 35 }
```

Step 11:

The screenshot shows the Node-RED web interface. On the left, the 'common' node palette is visible. In the center, the 'Flow 1' workspace contains a single 'IBM IoT' node. To the right, the 'User Settings' panel is open, showing a list of installed and available nodes. The 'debug' console on the right displays a log of message payloads:

```
msg.payload : Object
  { Temperature: 34, Humidity: 3 }
11/6/2022, 9:54:49 AM node: f2f2649a.0d0d98
iot-2/typeNodeMCU/id/24680/evt/event_1/fmt/json :
msg.payload : number
  34
11/6/2022, 9:54:49 AM node: f2f2649a.0d0d98
iot-2/typeNodeMCU/id/24680/evt/event_1/fmt/json :
msg.payload : number
  3
11/6/2022, 9:54:52 AM node: f2f2649a.0d0d98
iot-2/typeNodeMCU/id/24680/evt/event_1/fmt/json :
msg.payload : Object
  { Temperature: 96, Humidity: 44 }
11/6/2022, 9:54:52 AM node: f2f2649a.0d0d98
iot-2/typeNodeMCU/id/24680/evt/event_1/fmt/json :
msg.payload : number
  96
11/6/2022, 9:54:52 AM node: f2f2649a.0d0d98
iot-2/typeNodeMCU/id/24680/evt/event_1/fmt/json :
msg.payload : number
  44
```

Step 12:

The screenshot shows the Node-RED web interface in a browser. The left sidebar contains a list of nodes: numeric, dropdown, switch, text input, date picker, colour picker, form, text, gauge, chart, audio out, notification, ui control, and template. The main workspace displays a flow named 'Flow 1'. It starts with an 'IBM IoT' node (labeled 'connected') that branches into two function nodes: 'Temperature' and 'Humidity'. Both function nodes are connected to a 'msg.payload' node. The 'msg.payload' node then branches into two 'gauge' nodes. The right sidebar shows the 'debug' console with a log of messages. The messages are objects containing 'Temperature' and 'Humidity' values. The first message shows Temperature: 96 and Humidity: 44. The second message shows Temperature: 15 and Humidity: 95. The bottom status bar shows the system clock and weather information.

Step 13:

The screenshot shows the Node-RED web interface with the 'Edit gauge node' dialog box open. The dialog box has a 'Delete' button, a 'Cancel' button, and a 'Done' button. The 'Properties' tab is selected, showing the following settings: Group: '[Temperature] Smart Farming', Size: 'auto', Type: 'Gauge', Label: 'Temperature', Value format: '{value}', Units: '°C', Range: 'min 0 max 100', Colour gradient: a gradient from green to yellow to red, Sectors: '0 ... optional ... optional ... 100', and Class: 'Optional CSS class name(s) for widget'. The right sidebar shows the 'debug' console with a log of messages. The messages are objects containing 'Temperature' and 'Humidity' values. The first message shows Temperature: 96 and Humidity: 44. The second message shows Temperature: 15 and Humidity: 95. The bottom status bar shows the system clock and weather information.

Step 14:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1' with an 'IBM IoT' node connected to two function nodes labeled 'Temperature' and 'Humidity'. The 'Edit gauge node' panel is open on the right, showing the configuration for a gauge widget. The properties are as follows:

- Group: [Temperature] Smart Farming
- Size: auto
- Type: Gauge
- Label: Humidity
- Value format: {{value}}
- Units: gal
- Range: min 0, max 100
- Colour gradient: A gradient bar with green, yellow, and red segments.
- Sectors: 0, optional, optional, 100
- Class: Optional CSS class name(s) for widget
- Enabled: ☐

The debug console on the far right shows the following messages:

```
msg payload: Object
{ Temperature: 96, Humidity: 44 }
11/6/2022, 9:54:52 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload: number
96
11/6/2022, 9:54:52 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload: number
44
11/6/2022, 9:56:14 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload: Object
{ Temperature: 15, Humidity: 95 }
11/6/2022, 9:56:14 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload: number
15
11/6/2022, 9:56:14 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCUID/24680/evt/event_1/fmt/json :
msg payload: number
95
```

Step 15:

The screenshot shows a Google search page in a browser. The address bar displays the URL <https://node-red-xjezb-2022-11-05.eu-gb.mybluemix.net/ui>. The Google logo is centered on the page, and the search bar is below it. The search bar contains the text 'Search Google or type a URL'. Below the search bar are three buttons: 'Google', 'Web Store', and 'Add shortcut'. The Windows taskbar is visible at the bottom of the screen.

Step 16:

