

Create The IBM Watson IoT Platform And A Device

Project Title	SmartFarmer – IoT Enabled Smart Farming Application
Team ID	PNT2022TMID22142
Content	Node-RED

STEP 1:

Login into IBM Cloud and Open your node-red app.

The screenshot displays the Node-RED web interface within a browser. The browser's tab bar includes several open tabs, with the active one being 'Node-RED : node-red'. The address bar shows the URL 'https://node-red-zelxz-2022-11-08.au-syd.mybluemix.net/red/#flow/60daaa1e194009d0'. The Node-RED interface is visible, featuring a left sidebar with a 'filter nodes' search bar and a 'dashboard' category expanded. The main workspace shows a flow with an 'IBM IoT' node connected to 'temp', 'humidity', and 'soil moisture' nodes, which then connect to 'temperature', 'humidity', and 'soil moisture' output nodes. Below this, there are 'motor on' and 'motor off' nodes connected to a 'text input' node, which is connected to an 'IBM IoT OUT' node. The right sidebar contains 'User Settings' and a 'debug' console. The 'User Settings' panel shows a list of installed and available nodes, including 'dashboard', 'cn-dashboard-nodes', 'node-red-dashboard', 'feezal', 'node-red-contrib-dashboard-average-bars', 'node-red-contrib-dashboard-bar-chart-data', 'node-red-contrib-dashboard-sum-bars', 'node-red-contrib-mdashboard', and 'node-red-contrib-meraki-dashboard-api'. The 'debug' console shows a log of a message: 'msg: string[21] \"JSON Message expected\"'.

STEP 2:

For IBM Cloud connection you need certain nodes which can be installed by going to Manage Palette and then install required nodes.

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes tabs for 'IBM', 'IBM-EPBL/IBM-Project', 'Node-RED.pdf', 'IBM Watson IoT Platform', 'Node-RED: node-red-...', 'Node-RED Dashboard', 'IBM Cloud and Node-RED', 'Download file | iLoveP...', and 'IBM App Development'. The address bar shows the URL: <https://node-red-zelxz-2022-11-08.au-syd.mybluemix.net/red/#flow/60daaa1e194009d0>.

The main workspace shows a flow named 'Flow 1'. It starts with an 'IBM IoT' node (blue) which is connected to three function nodes: 'temp', 'humidity', and 'soil moisture'. Each function node is connected to a corresponding output node: 'temperature', 'humidity', and 'soil moisture'. Below this, there is a 'motor on' and 'motor off' node connected to a 'text input' node, which is then connected to an 'IBM IoT OUT' node (blue).

The left sidebar contains a 'filter nodes' search bar and a 'dashboard' section with various widgets like 'button', 'dropdown', 'switch', 'slider', 'numeric', 'text input', 'date picker', 'colour picker', 'form', 'text', 'gauge', 'chart', 'audio out', 'notification', 'ui control', and 'template'.

The right sidebar shows a 'debug' console with a list of selected nodes. The console displays a series of messages, each containing a timestamp, a node ID, and a payload. The payload is a JSON object with the following structure:

```
{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 ... }
```

STEP 3:

Now you can connect your cloud by entering API Credentials and enter device details.

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes tabs for IBM, IBM-EPBL/IBM-Project, Node-RED.pdf, IBM Watson IoT Platform, Node-RED: node-red..., Node-RED Dashboard, IBM Cloud and Node-RED, and Download file | iLovePi. The address bar shows the URL: <https://node-red-zelxz-2022-11-08.au-syd.mybluemix.net/red/#flow/60daaa1e194009d0>.

The main workspace shows a flow diagram. On the left, a sidebar lists various nodes under 'filter nodes' and 'dashboard'. The flow includes an 'IBM IoT' node (connected) which branches into three functions: 'temp', 'humidity', and 'soil moisture'. Each function is connected to a corresponding output node: 'temperature', 'humidity', and 'soil moisture'. Below this, there is a 'motor on' and 'motor off' node connected to a 'text input' node, which is also connected to an 'IBM IoT OUT' node (connected).

The right sidebar shows the 'Edit ibmiot in node' configuration panel. The 'Properties' tab is active, displaying the following settings:

- Authentication: API Key
- API Key: 77dd251f59c85ca3
- Input Type: Device Event
- Device Type: ☐ All or ☒ testing
- Device Id: ☐ All or ☐ 0909
- Event: ☐ All or ☐ INPUT
- Format: ☐ All or ☒ json
- QoS: 0
- Name: IBM IoT
- Service: registered

A yellow tooltip message states: "Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to IoT Applications. Check the info tab, to get more information about each of the fields".

The bottom right corner shows a 'debug' console with a log entry: "JSON Message expected". The console also displays a series of messages from the 'node' and 'msg.payload' fields, showing a JSON object with weather data: `{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 ... }`.

STEP 4:

Create your flow by drag and drop the elements.

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes tabs for 'IBM', 'IBM-EPBL/IBM-Project', 'Node-RED.pdf', 'IBM Watson IoT Platform', 'Node-RED : node-red', 'Node-RED Dashboard', 'IBM Cloud and Node-RED', 'Download file | iLoveP', 'IBM App Development', and a search icon. The address bar shows the URL: <https://node-red-zelz-2022-11-08.au-syd.mybluemix.net/red/#flow/60daaa1e194009d0>. The main workspace shows a flow diagram on a grid. On the left, a sidebar lists available nodes under 'filter nodes' and 'dashboard'. The flow starts with an 'IBM IoT' node (connected) which branches into three function nodes: 'temp', 'humidity', and 'soil moisture'. Each function node is connected to a corresponding output node: 'temperature', 'humidity', and 'soil moisture'. Below this, there is a 'motor on' and 'motor off' node connected to a 'text input' node, which is then connected to an 'IBM IoT OUT' node (connected). The right sidebar shows a debug console with a log of messages. The messages are JSON objects with the following structure:

```
{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 }
```

You can see web ui by adding ui after your url.

