

SPRINT-3

Date	13 November 2022
Project Name	Smart Farmer - IoT Enabled Smart Farming Application
TEAM ID	PNT2022TMID15139

PROCEDURE:

- Open IBM Cloud and search Node-RED in catalog.
- Create Node-RED configuration and visit the app URL.
- Construct all nodes for Smart farm and enter all configuration in nodes.
- After configuring click deploy and see the output in python IDLE and debug
- Visit the api web and monitor the temperature and humidity reading.

Creation of Node-RED

The screenshot displays the IBM Cloud console for a Node-RED application named 'Node RED WIKHX 2022-11-12'. The top navigation bar includes the IBM Cloud logo, a search bar, and user account information. The main content area is divided into several sections:

- Details:** Contains fields for App URL, Source (with a 'Download code' button), Resource group (set to 'Default'), Deployment target, and Created date (11/12/2022).
- Services:** Features a 'Cloudant' service card with links to 'Open dashboard', 'Documentation', and 'API reference', along with a 'Credentials' dropdown and buttons for 'Connect existing services' and 'Create service'.
- Deployment Automation:** Includes a 'Configure Continuous Delivery' section with a 'Deploy your app' button.
- Getting started quickly:** A sidebar with a 4-step guide for configuring the app, connecting services, and deploying.

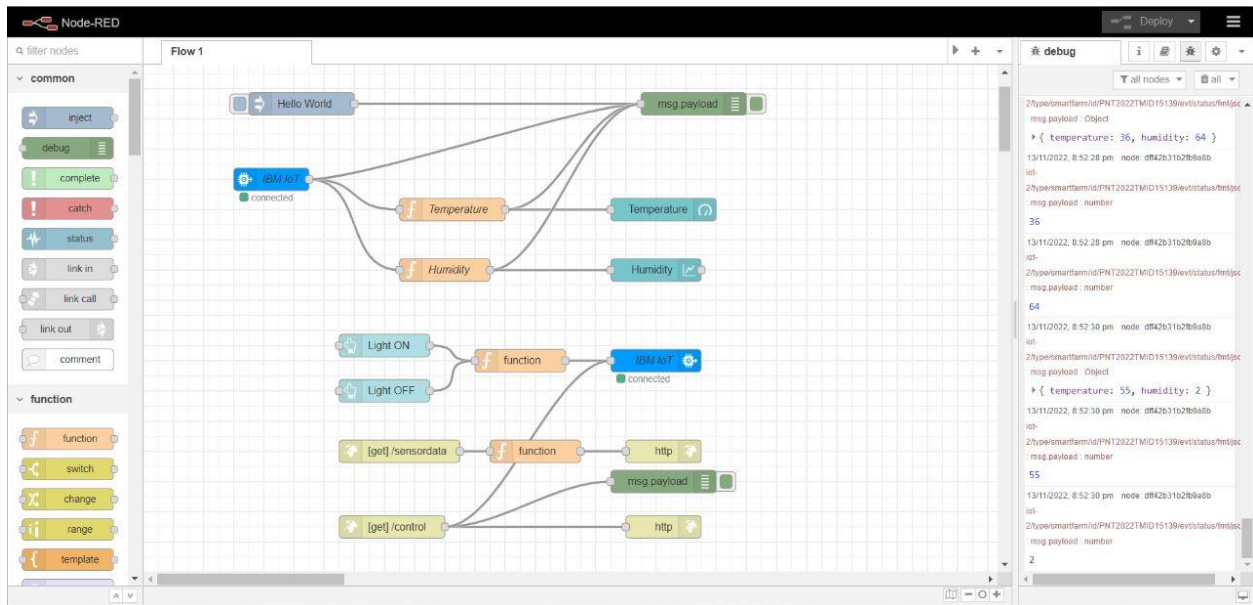
The 'Getting started quickly' sidebar contains the following steps:

1. Use the **Services** card to connect a service to your app. Select an existing service instance, or create a new one. [Learn more.](#)
2. If you want to view the code before your app is deployed, click **Download code** to obtain the .zip file.
3. Click **Deploy your app** in the **Deployment Automation** card to select the deployment target and configure the Continuous Delivery service. The deployment begins automatically.
4. After the deployment begins, you can view the status of the deployment, modify your app, view your repo, or...

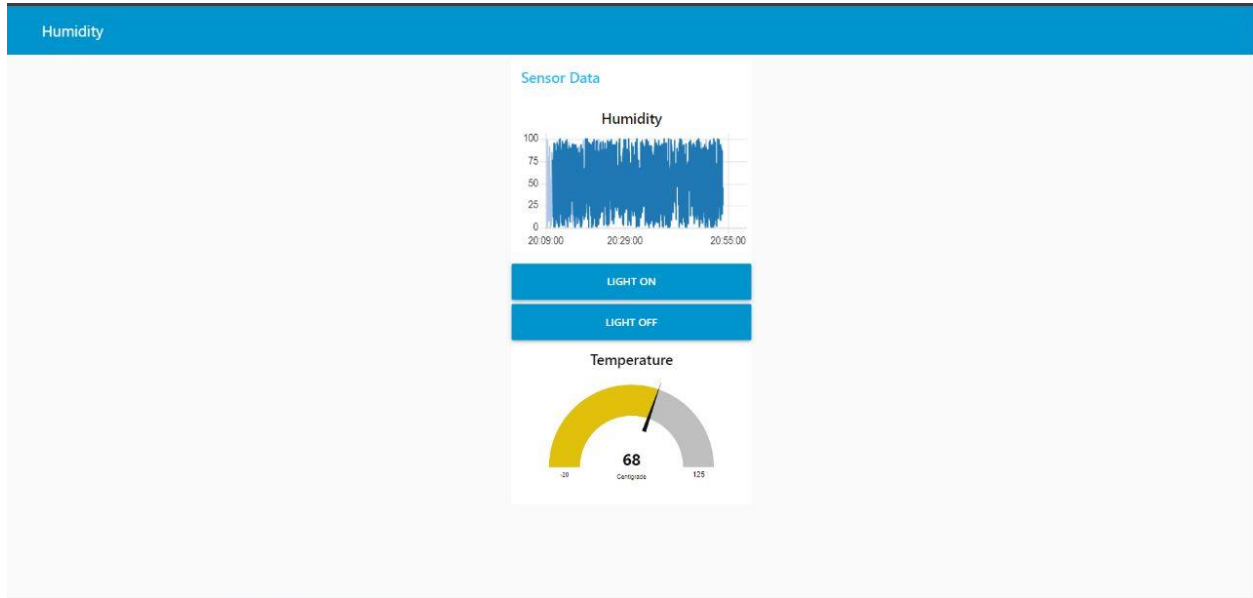
Run the Node-RED and launch the URL from IBM Cloud

The screenshot displays the IBM Cloud console interface for a resource named "Node RED WIKHX 2022-11-12". The resource is in a "Running" state. The left sidebar shows navigation options: Getting started, Overview (selected), Runtime, Connections, Logs, API Management, and Autoscaling. The main content area includes a warning banner about IBM Cloud Foundry Public being deprecated. Below this, the "Instances" section shows a health status of 100% (1/1 instance(s) are running) and a memory allocation of 256 MB. The "Runtime" section shows a Node.js runtime with a total MB allocation of 256. The "Runtime cost" section shows a current and estimated cost of \$0.00. The "Connections (1)" section shows a single connection to "node-red-wikhx-2022--cloudant-1668231376847-2026".

Construction of Nodes



Monitoring the sensor nodes



Configuring the IBM Nodes

The image shows the Node-RED interface with a flow titled "Flow 1". The flow starts with a "Hello World" node, followed by an "IBM IoT" node (labeled "connected"). This node connects to "Temperature" and "Humidity" nodes, which then connect to "Light ON" and "Light OFF" nodes. Below these, there are "function" nodes connected to "[get] /sensordata" and "[get] /control" nodes. The "Edit IBM IoT node" dialog is open, showing the following configuration:

- Authentication: API Key
- API Key: d533fad203f76f3a
- Output Type: Device Command
- Device Type: smartfarm
- Device Id: PNT2022TMID15139
- Command Type: cmd
- Format: json
- Data: data
- QoS: 0
- Name: IBM IoT
- Service: registered

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

```
2022/11/11 8:55:25 pm node: dff42b31e27b9a8b  
msg.payload: Object  
{ temperature: 79, humidity: 13 }  
2022/11/11 8:55:25 pm node: dff42b31e27b9a8b  
msg.payload: number  
79  
2022/11/11 8:55:25 pm node: dff42b31e27b9a8b  
msg.payload: number  
13  
2022/11/11 8:55:27 pm node: dff42b31e27b9a8b  
msg.payload: Object  
{ temperature: 24, humidity: 12 }  
2022/11/11 8:55:27 pm node: dff42b31e27b9a8b  
msg.payload: number  
24  
2022/11/11 8:55:27 pm node: dff42b31e27b9a8b  
msg.payload: number  
12
```

Interfacing Node-RED and MIT App Inventor

The screenshot displays the Node-RED web interface. On the left, the 'common' and 'function' node palettes are visible. The main workspace shows a flow named 'Flow 1' starting with a 'Hello World' node, followed by a 'msg to 1' node. This node is connected to two function nodes labeled 'Temperature' and 'Humidity'. Below these, there are 'Light ON' and 'Light OFF' nodes, and two input nodes labeled '[get] /sensordata' and '[get] /control'. The 'Edit function node' panel is open for the selected function node, showing the following JavaScript code:

```
1= msg.payload({
2  "temperature":global.get('t'),
3  "humidity":global.get('h'),
4= }
5  return msg;
```

The 'debug' console on the right shows the execution of the function. It displays the message payload: `{ temperature: 107, humidity: 45 }`. The console also shows the global variables being updated: `global.get('t')` and `global.get('h')`.