What is Node-RED?

Node-RED is a programming tool for wiring together hardware devices, APIs and online services. Primarily, it is a visual tool designed for the Internet of Things, but it can also be used for other applications to very quickly assemble flows of various services.

It is open source and was originally created by the IBM Emerging Technology organisation. It is included in IBM's Bluemix (a Platform-as-a-Service or PaaS) IoT starter application package. Node-RED can also be deployed separately using the Node.js application. At present, Node-RED is a JS Foundation project.

Node-RED enables users to stitch together Web services and hardware by replacing common low-level coding tasks (like a simple service talking to a serial port), and this can be done with a visual drag-drop interface. Various components in Node-RED are connected together to create a flow. Most of the code needed is created automatically.

Features of Node-RED

The major features of Node-RED are listed below.

- It supports browser-based flow editing.
- As it is built on Node.js, it supports a lightweight runtime environment along with the event driven and non-blocking model.
- The various flows created in Node-RED are stored using JSON, which can be easily imported and exported for sharing with others.
- You can run it locally (Docker support, etc).
- It can easily fit on most widely used devices like Raspberry Pi, BeagleBone Black, Arduino, Android based devices, etc.
- It can run in the cloud environment like Bluemix, AWS, MS-Azure, etc.

Node-RED architecture

Some of the key aspects of Node-RED architecture are listed below:

- Node-RED (latest version 0.16) is fast because it is driven by the latest supported version of Node.js (LTS 6.x, 6.x).
- Asynchronous io and event-driven architecture.
- Single-threaded event queue, which supports simplicity.
- Single language support for JavaScript (both from front-end and back-end).
- Complete architecture has been built using express, d3, jquery and ws.

Different types of nodes

Inject node

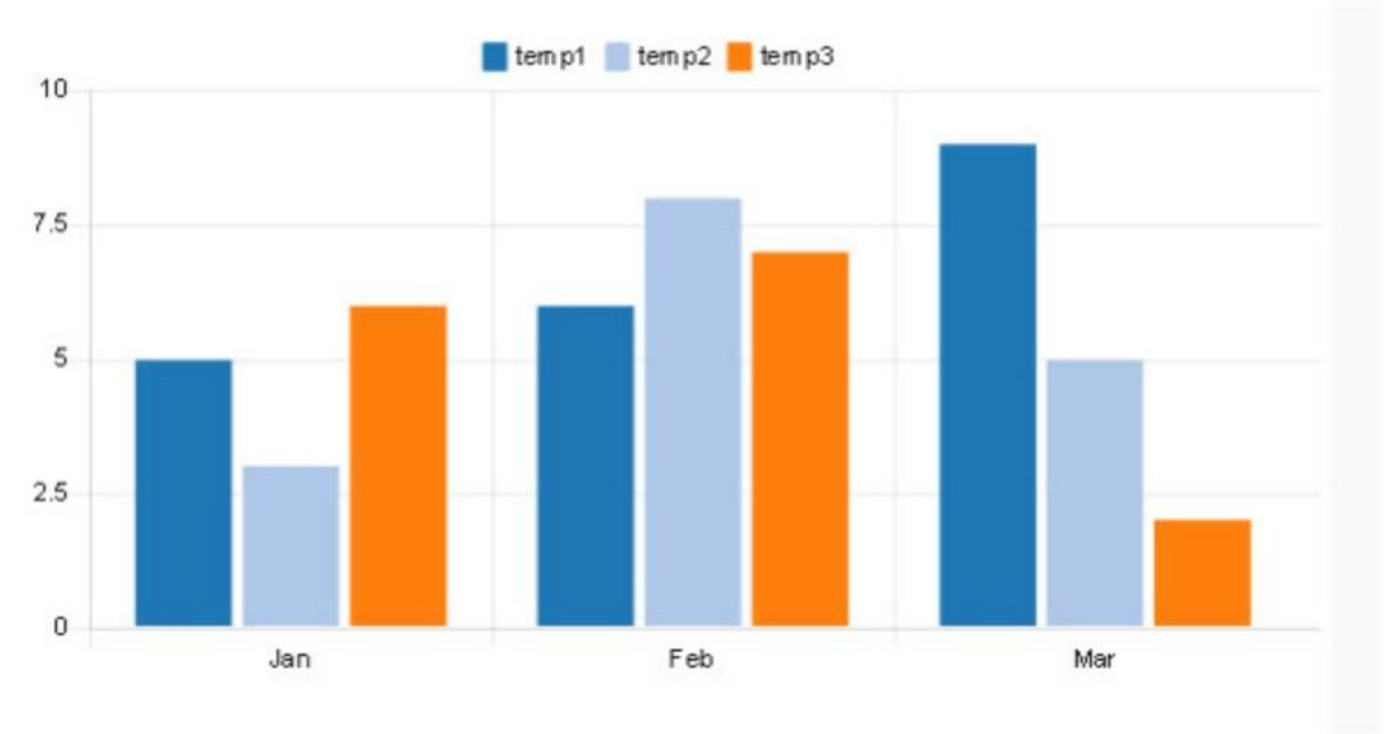
- This node allows manual triggering of flows.
- It helps us to inject events at scheduled intervals.

Debug node

- The debug node helps in displaying the contents of a message—either the payload or the entire object.

Displaying Bar Charts

Besides the line char bar charts are another common format



You can see in the example above that the x axis contains the month names passed in the **msg.label** property.

The actual data is shown below:

```
msg.payload=[{
    "series": ["temp1", "temp2", "temp3" ],
    "data": [ [5,6,9], [3,8,5], [6,7,2] ],
    "labels": [ "Jan", "Feb", "Mar" ]
}];
```