

---

# Boosting Productivity of Node-RED with Large Language Model

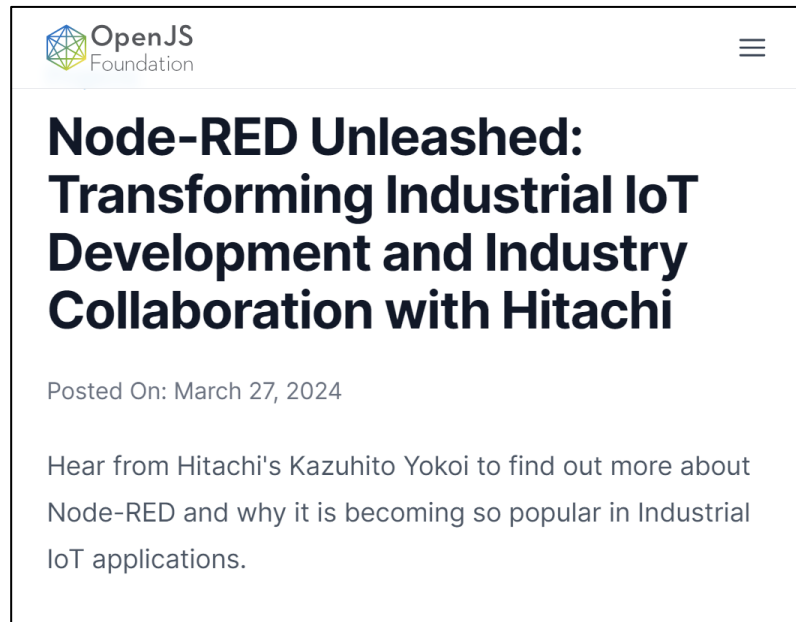
October 21, 2024

Kazuhito Yokoi  
Hitachi Academy

## Kazuhito Yokoi (横井 一仁)



- Software Engineer, Hitachi Academy
- No.3 contributor in Node-RED project
- Organizer of Node-RED User Group



My interview on  
OpenJS Foundation blog

<https://openjsf.org/blog/node-red-unleashed>

# Contents

---

1. What is Node-RED?
2. Large Language Model (LLM)
3. Code Completion Plugin for Node-RED
4. Demonstrations
5. Conclusions

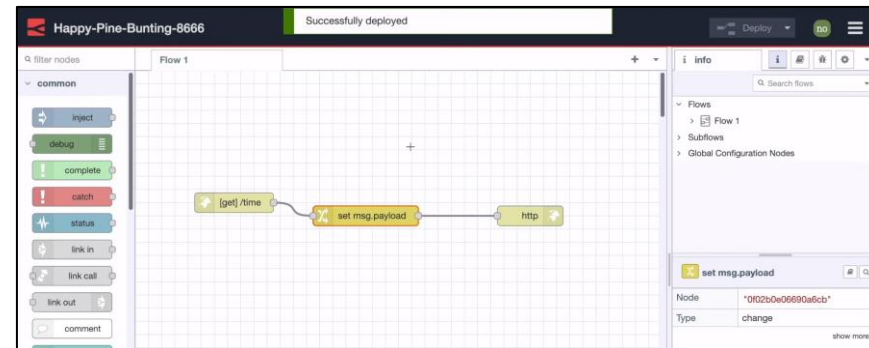
---

## What is Node-RED?

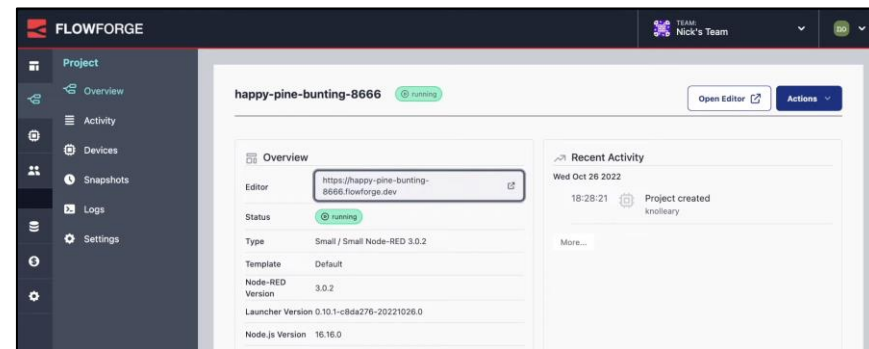
# What is Node-RED?

Visual programming tool for IoT and web applications

- Developed by IBM in 2013
- OSS hosted in OpenJS Foundation
- Browser based development environment
- Management OSS available



Node-RED flow editor



Management software

# The Number of Downloads

In 10 years, the number of downloads from the npm has gradually increased.



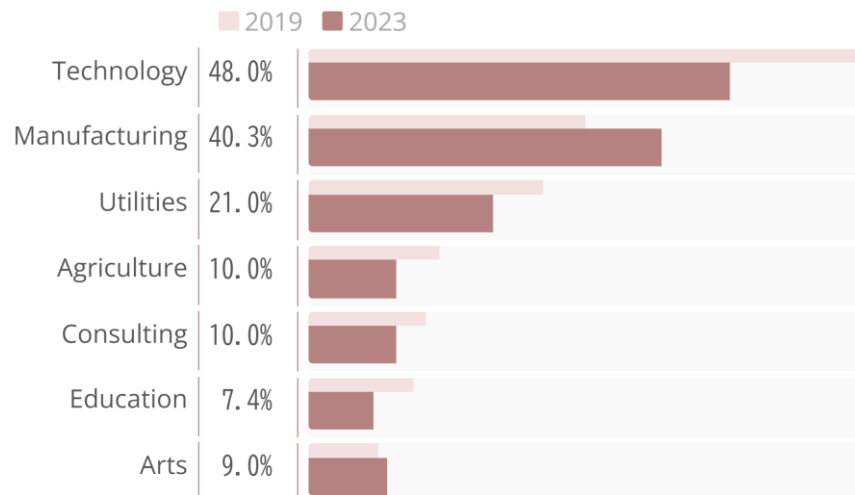
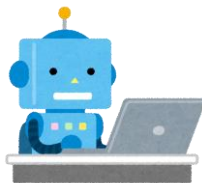
<https://npm-stat.com/charts.html?package=node-red&from=2015-01-01&until=2024-12-31>

- 229 contributors to Node-RED project
- 3rd party 5081 nodes available
- Pre-installed edge devices
  - reTerminal DM, Seed Studio
  - EdgePi
- Node-RED events
  - Node-RED conference
  - Regular meetups
- Node-RED user companies  
Siemens, Schneider, NEC, Fujitsu, HPE, Hitachi and others



Node-RED is widely used in the following industries and use cases:

- **Technology**
  - Chatbot
  - API Gateway
- **Manufacturing Industry**
  - Factory Automation
  - Data Visualization
- **Utilities**
  - Energy Management
  - Predictive Maintenance
- **Agriculture**
  - Smart Agriculture



Community survey  
about major industries

<https://nodered.org/about/community/survey/2023/>



Visual programming environment by connecting functional blocks in the processing order

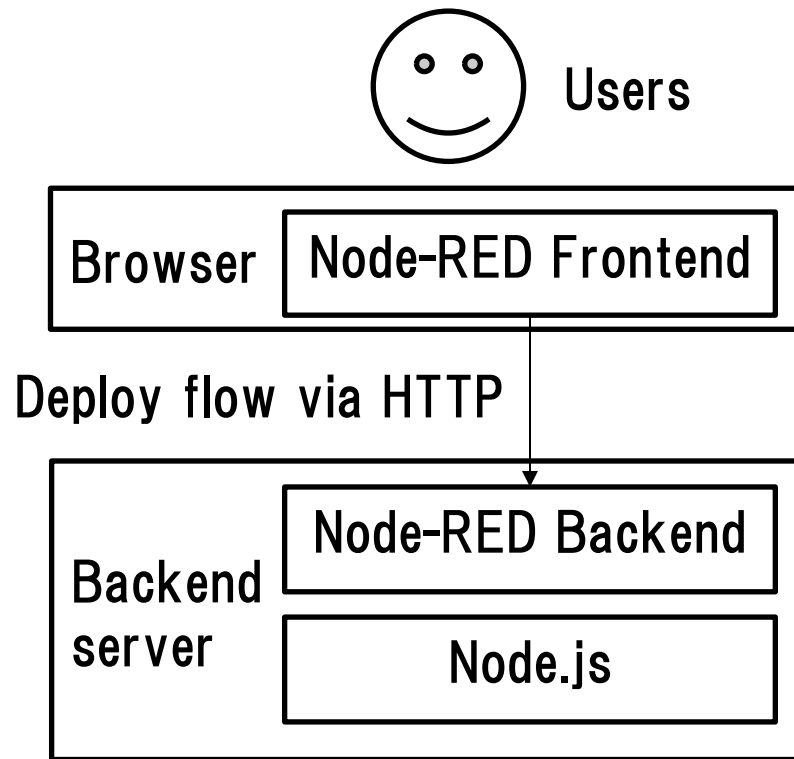
- No code development environment for beginners
- Extensibility through partial code writing

The screenshot displays the Node-RED Flow Editor interface. On the left, a sidebar shows a search bar 'filter nodes' and a 'common' category with nodes like 'inject', 'debug', and 'complete'. The main workspace shows a flow: an 'mqtt' node (purple) connects to a 'set msg.payload' node (yellow), which then connects to a 'function' node (orange). The 'function' node connects to a 'sendgrid' node (blue). A callout box points to the 'mqtt' node with the text 'Development by connecting blocks'. Another callout box points to the 'function' node, which contains a JavaScript code snippet, with the text 'Partial coding to extend flow'. The code in the function node is as follows:

```
1 const values = context.get("values") || []; // 直近の値を保存する配列を取得
2 const threshold = 0.5; // 50%の閾値
3
4 values.push(msg.payload); // 受け取った値を配列に追加
5
6 if (values.length > 7) {
7   values.shift(); // 配列の先頭から要素を削除して7つに保つ
8 }
9
10 const average = values.reduce((total, value) => total + value, 0) / values.length;
11
12 if (msg.payload > average * (1 + threshold)) {
```

Node-RED consists of a front-end UI and a back-end server.

- In flow development, users use the Node-RED front-end UI in their browser.
- The Node-RED back-end runs on Node.js as an HTTP server.



Type a few commands in terminal on the local PC.

(1) Install Node.js with Node Version Manager

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.0/install.sh | bash  
nvm install 22
```

(2) Restart the terminal

(3) Install Node-RED using npm command

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

(4) Start the Node-RED server

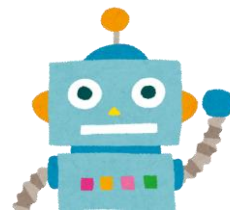
```
node-red
```

-> You can access the Node-RED from "http://localhost:1880".

To increase the productivity of Node-RED, we have recently adopted technologies that support coding by AI.

I will introduce the following background technologies with demonstrations:

- Large Language Model
- Granite Code
- Code completion
- Plugin for Node-RED



Coding support  
by AI



```
1 const values = context.get("values") || []; // 直近の値を保存する配列を取得
2 const threshold = 0.5; // 50%の閾値
3
4 values.push(msg.payload); // 受け取った値を配列に追加
5
6 if (values.length > 7) {
7   values.shift(); // 配列の先頭から要素を削除して7つに保つ
8 }
9
10 const average = values.reduce((total, value) => total + value, 0) / values.length;
11
12 if (msg.payload > average * (1 + threshold)) {
```

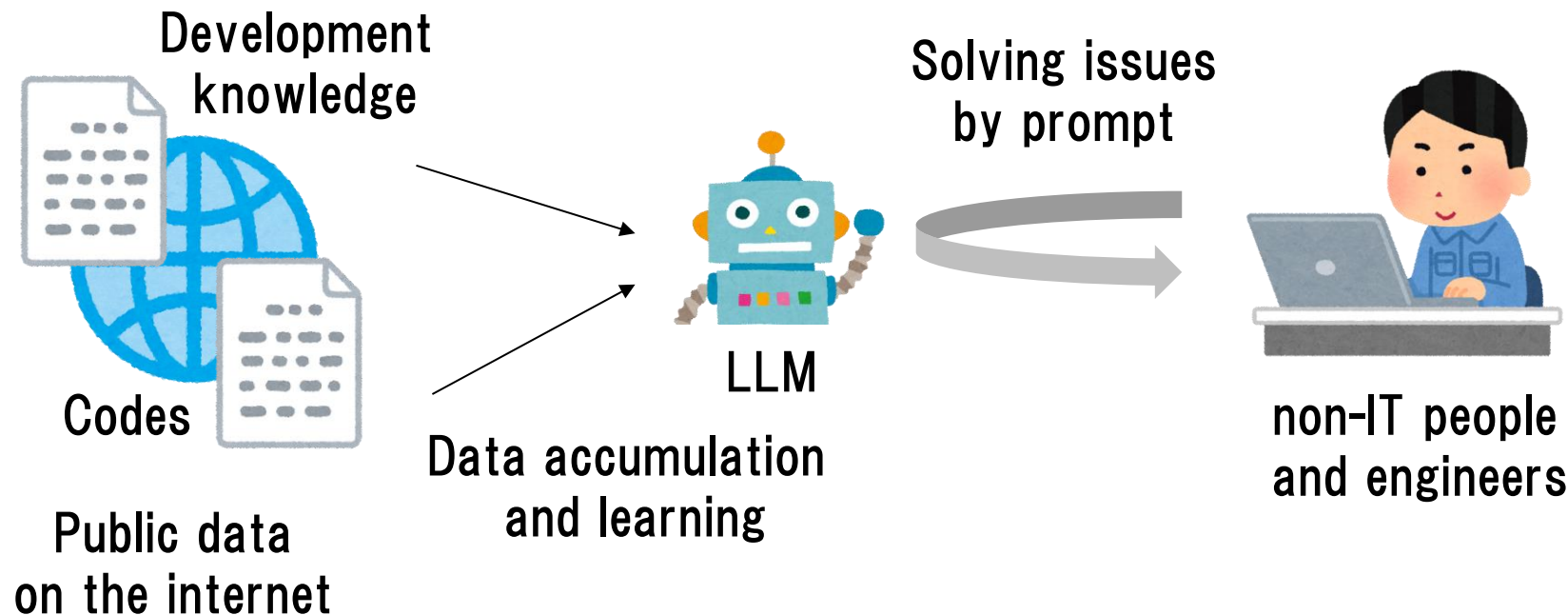
Coding to extend flow



---

## Large Language Model (LLM)

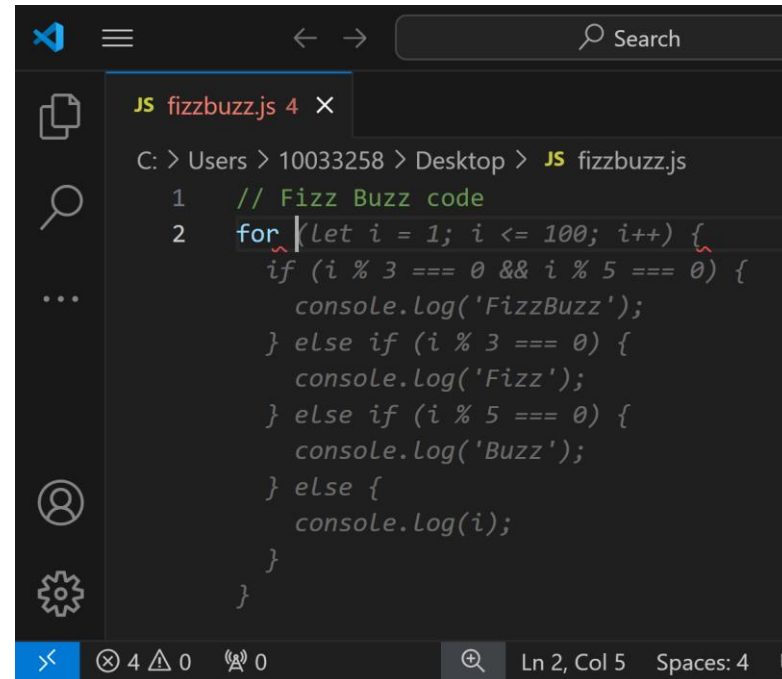
Both non-IT people and engineers have used LLM like ChatGPT to solve the programming and system problems by themselves.



## LLM for software coding tasks to increase software development productivity

- Key functionalities
  - Code generation from comment
  - Code explanation from code
  - Code fixing
- Major LLMs
  - Public Service LLMs (GitHub Copilot)
  - Local LLMs (Granite Code)

<https://github.com/features/copilot>



The screenshot shows a code editor window with a file named 'fizzbuzz.js'. The code is a JavaScript implementation of the Fizz Buzz problem. The editor interface includes a search bar at the top right, a sidebar on the left with icons for Explorer, Search, and Source Control, and a status bar at the bottom showing 'Ln 2, Col 5' and 'Spaces: 4'. The code is as follows:

```
JS fizzbuzz.js 4 X
C: > Users > 10033258 > Desktop > JS fizzbuzz.js
1 // Fizz Buzz code
2 for (let i = 1; i <= 100; i++) {
    if (i % 3 === 0 && i % 5 === 0) {
        console.log('FizzBuzz');
    } else if (i % 3 === 0) {
        console.log('Fizz');
    } else if (i % 5 === 0) {
        console.log('Buzz');
    } else {
        console.log(i);
    }
}
```

## Code generation on GitHub Copilot

# Demonstration: Code Generation Using GitHub Copilot

JS fizzbuzz.js 4 ×

C: > Users > 10033258 > Desktop > JS fizzbuzz.js

```
1 // Fizz Buzz code
2 for (let i = 1; i <= 100; i++) {
    if (i % 3 === 0 && i % 5 === 0) {
        console.log('FizzBuzz');
    } else if (i % 3 === 0) {
        console.log('Fizz');
    } else if (i % 5 === 0) {
        console.log('Buzz');
    } else {
        console.log(i);
    }
}
```



As of 2024, we evaluated two types of LLMs as follows.

#	Item	Public Service LLMs	Local LLMs
1	Cost	Monthly or PAYG	Free
2	Speed	Good	Depends on PC spec
3	Offline use	Impossible	Possible
4	Quality	Good	Depends on PC spec

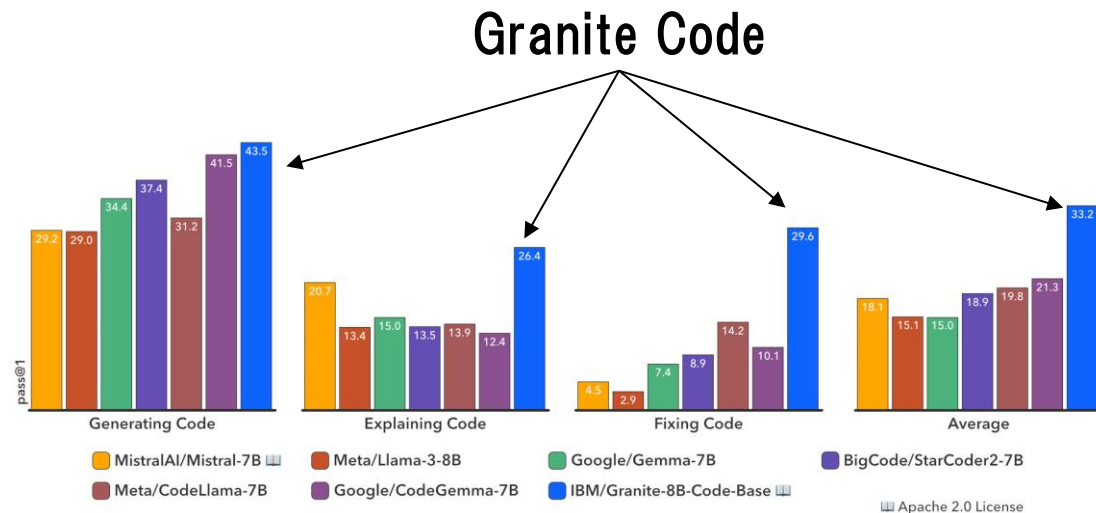
(Green color means positive)

Local LLMs are suitable for the following cases:

- Development like testing and documentation, which uses LLM many times
- Projects under the compliance policies such as export control

## Local LLM for programming tasks released by IBM Research in May 2024

- High quality compared to other models
- Apache-2.0 license
- 116 programming languages supported
- Open data source
- Model size:
  - 3B parameters: 2.0 GB
  - 8B parameters: 4.6 GB
  - 20B parameters: 12 GB
  - 34B parameters: 19 GB



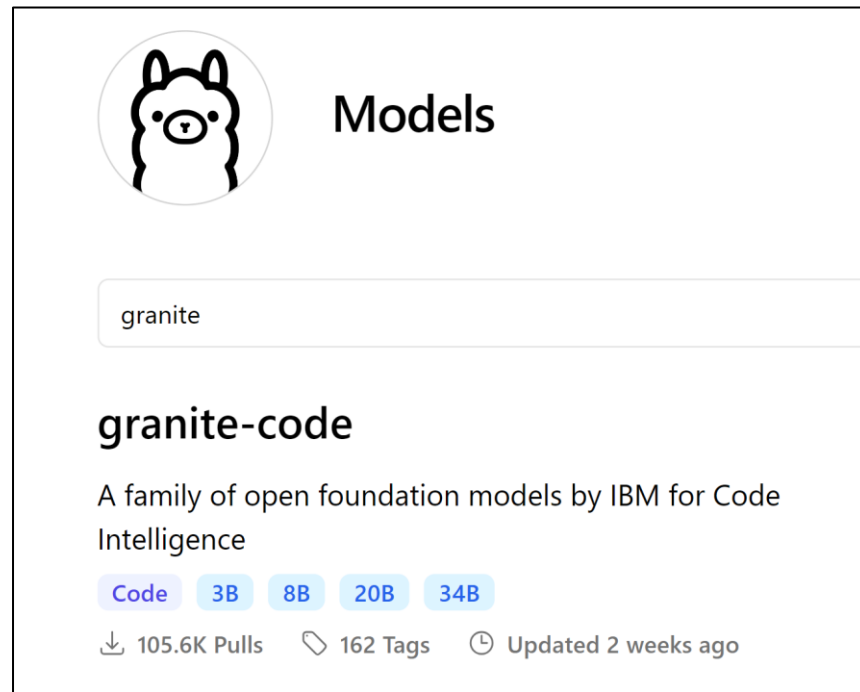
## Quality Comparison with Local LLMs

<https://github.com/ibm-granite/granite-code-models>

## Backend server software for running LLMs on the local PC

- Available models
  - Mistral
  - CodeLlama
  - Gemma
  - StarCode2
  - Granite Code
- PC specs used in the Granite Code
  - Lenovo ThinkPad T14s Gen 2 (32 GB memory)
  - M1 MacBook Air (8 GB memory)

<https://ollama.com/library/granite-code>



## Granite Code model on Ollama page

After setting up Ollama, you can execute model through CLI or REST API.

- Install Ollama for Linux environment

```
curl -fsSL https://ollama.com/install.sh | sh  
ollama serve
```

- Execute model
  - Command line interface

```
ollama run granite-code:3b-instruct
```

- REST API

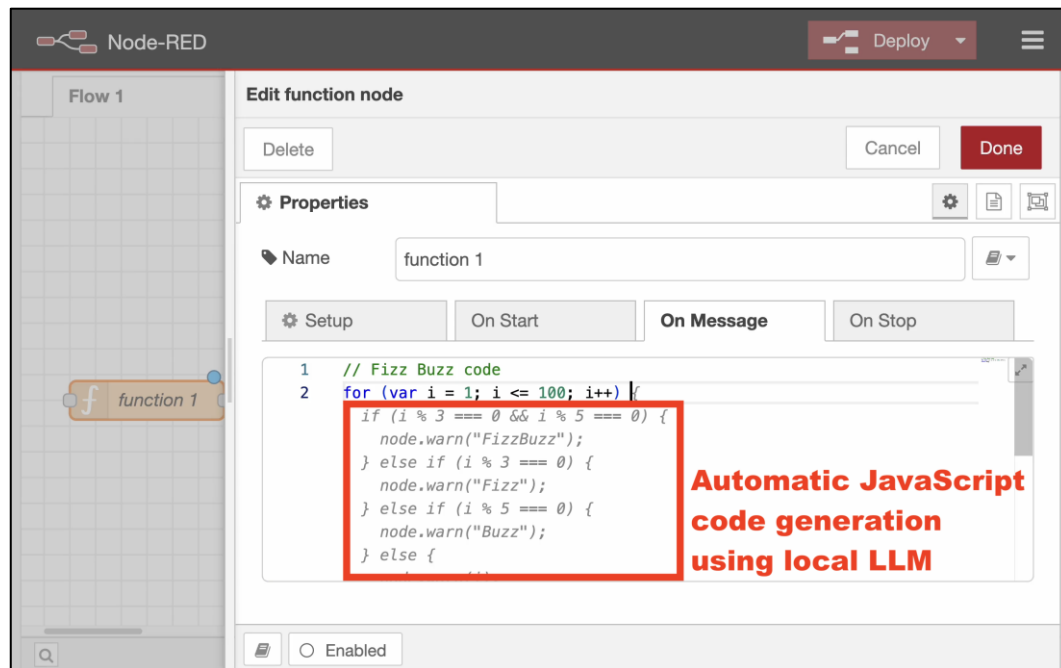
```
curl http://localhost:11434/api/generate -d '{ "model": "granite-code:3b-instruct", "prompt": "Hello world in Node.js", "stream": false }'
```

---

## Code Completion Plugin for Node-RED

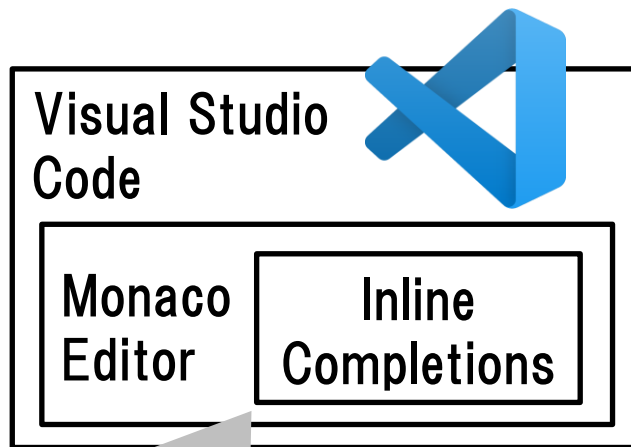
## Plugin to enable code completion in code editor of function node

- Provides the same UX as code completion in GitHub Copilot-enabled VS Code
- Generates JavaScript code from comments using Granite Code Model
- Available as an OSS on Node-RED library



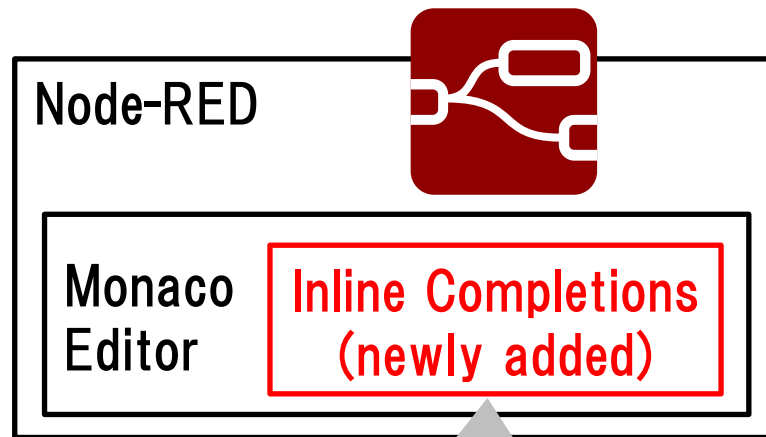
## Code editor of function node

Node-RED can be applied for the same UX as Visual Studio Code, because they both have the same Monaco Editor.



[ UX in GitHub Copilot ]

- Code Suggestion
- Accepting code by tab key

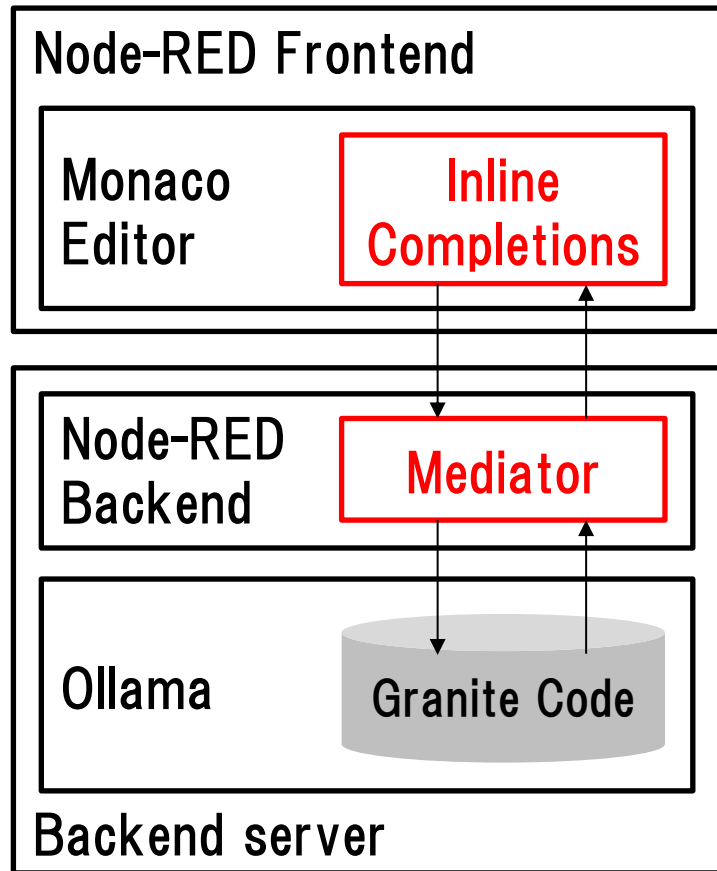


Apply for the same  
UX for Node-RED

# Architecture of Code Completion Plugin

Using the Node-RED plugin feature, the UI and backend components are installed.

- Inline Completions UI
  - Provides code suggestions UX
  - Accesses Mediator endpoint which has the same port as Node-RED backend
- Mediator
  - Provides REST API with authentication for Inline Completions UI
  - Accesses to the Ollama endpoints within the same backend server

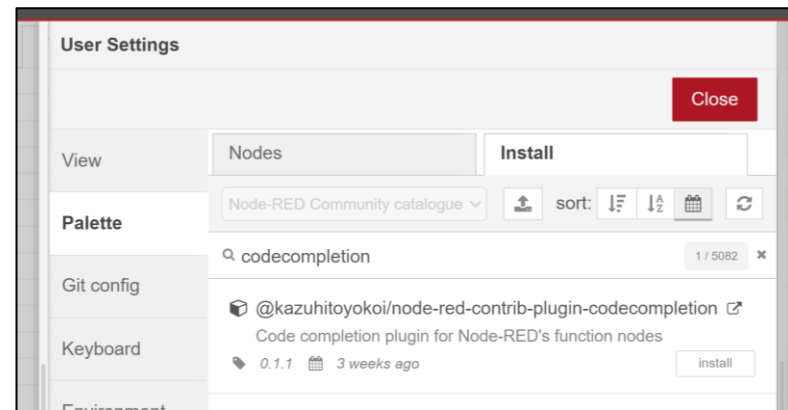
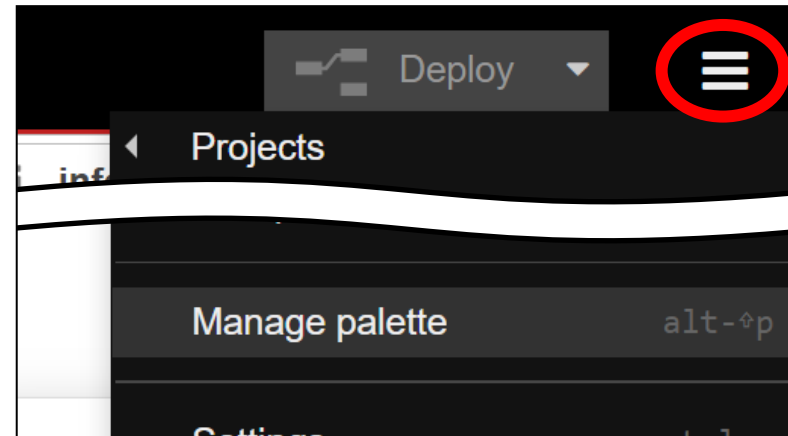




# How to Install Code Completion Plugin

Node-RED has a UI for installing plugins in the browser.

- (1) Select "Manage Palette" from the menu
- (2) Open the "Install" tab in the user settings UI
- (3) Type "codecompletion" in the search field
- (4) Click the install button of the "node-red-contrib-plugin-codecompletion"



# Downloading Granite Code Model

The image displays two side-by-side windows. The left window, titled 'node-red', shows a terminal console with the following text:

```
-----  
Your flow credentials file is encrypted using a sy  
stem-generated key.  
  
If the system-generated key is lost for any reason  
, your credentials  
file will not be recoverable, you will have to del  
ete it and re-enter  
your credentials.  
  
You should set your own key using the 'credentials  
ecret' option in  
your settings file. Node-RED will then re-encrypt  
your credentials  
file using your chosen key the next time you deplo  
y a change.  
-----  
  
7 Oct 10:03:59 - [info] Server now running at http  
://127.0.0.1:1880/  
7 Oct 10:03:59 - [warn] Encrypted credentials not  
found  
7 Oct 10:03:59 - [info] Starting flows  
7 Oct 10:03:59 - [info] Started flows
```

The right window, titled 'Node-RED : Flow 1', shows the Node-RED flow editor interface. It includes a 'filter nodes' search bar, a 'Deploy' button, and a list of nodes under 'common' (inject, debug, complete, catch, status, link in, link call, link out, comment) and 'function' (function). The main workspace is a grid for building flows.

**Console in backend server**

**Node-RED flow editor**

---

## Demonstrations

# Demonstration 1: Generating Fizz Buzz Code

(1) Typing the comments  
and loop block

```
1 // Fizz Buzz code
2 for (var i = 1; i <= 100; i++)
3 return msg;
```

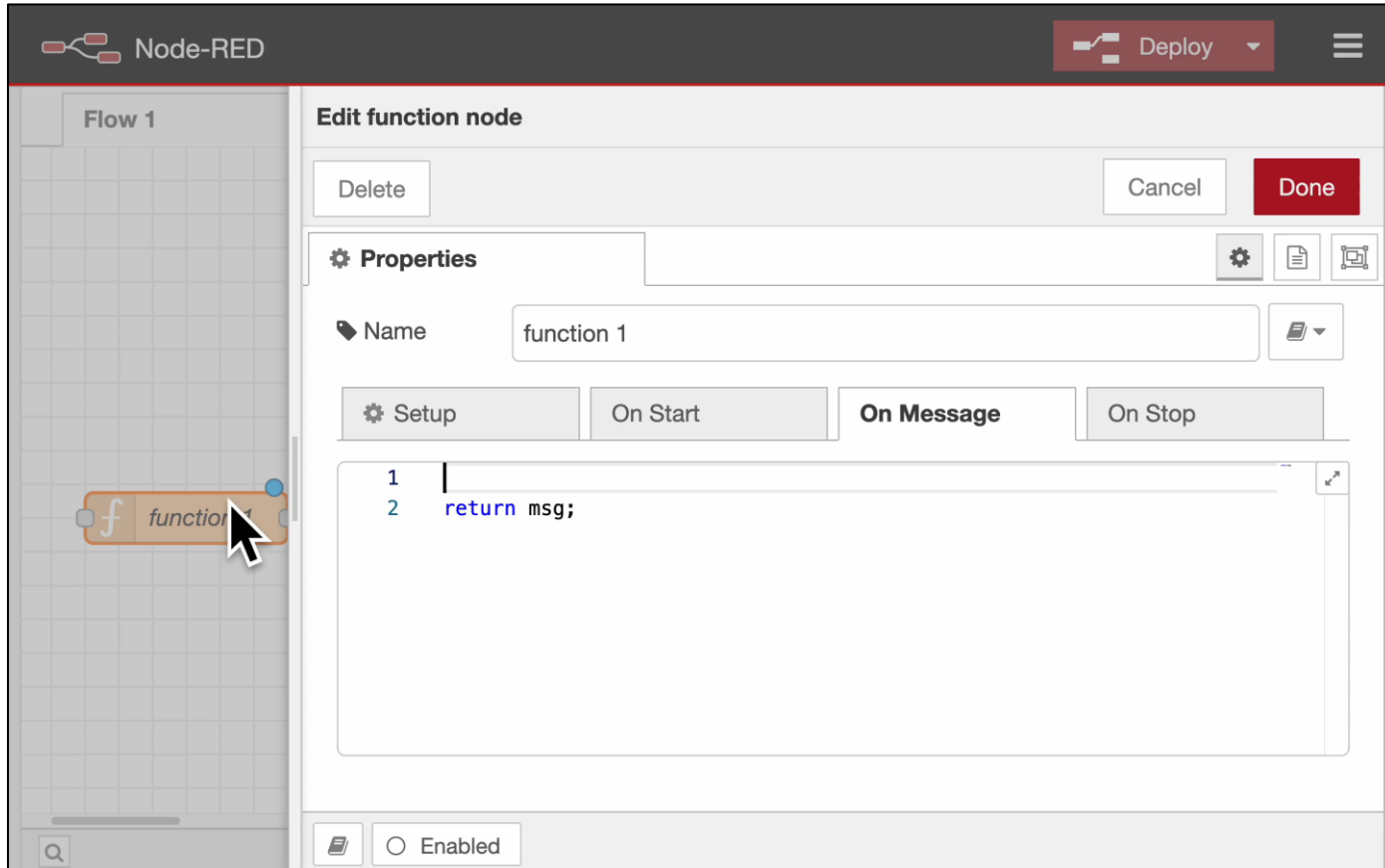
(2) Fizz Buzz Code is  
suggested in gray text

```
1 // Fizz Buzz code
2 for (var i = 1; i <= 100; i++) {
    if (i % 3 === 0 && i % 5 === 0) {
        node.warn("FizzBuzz");
    } else if (i % 3 === 0) {
        node.warn("Fizz");
    } else if (i % 5 === 0) {
        node.warn("Buzz");
    } else {
        node.warn(i);
    }
}
```

(3) Applying suggested code  
by tab key

```
1 // Fizz Buzz code
2 for (var i = 1; i <= 100; i++) {
3     if (i % 3 === 0 && i % 5 === 0) {
4         node.warn("FizzBuzz");
5     } else if (i % 3 === 0) {
6         node.warn("Fizz");
7     } else if (i % 5 === 0) {
8         node.warn("Buzz");
9     } else {
10        node.warn(i);
    }
}
```

# Demonstration 1: Generating Fizz Buzz Code



The screenshot shows the Node-RED web interface. On the left, a flow canvas labeled 'Flow 1' contains a function node icon labeled 'function 1'. A mouse cursor is clicking on this node. On the right, the 'Edit function node' panel is open. It features a 'Delete' button, 'Cancel' and 'Done' buttons, and a 'Properties' section with icons for settings, file, and help. The 'Name' field is set to 'function 1'. Below this are four tabs: 'Setup', 'On Start', 'On Message', and 'On Stop'. The 'On Message' tab is selected, showing a code editor with the following code:

```
1 |  
2 return msg;
```

At the bottom of the panel, there is a file icon and a radio button labeled 'Enabled'.

## Demonstration 2: Calculating Distance between Two Locations

Generating code to calculate the distance between two locations consist of latitude and longitude values

[ Haversine Formula ]

$$d = 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

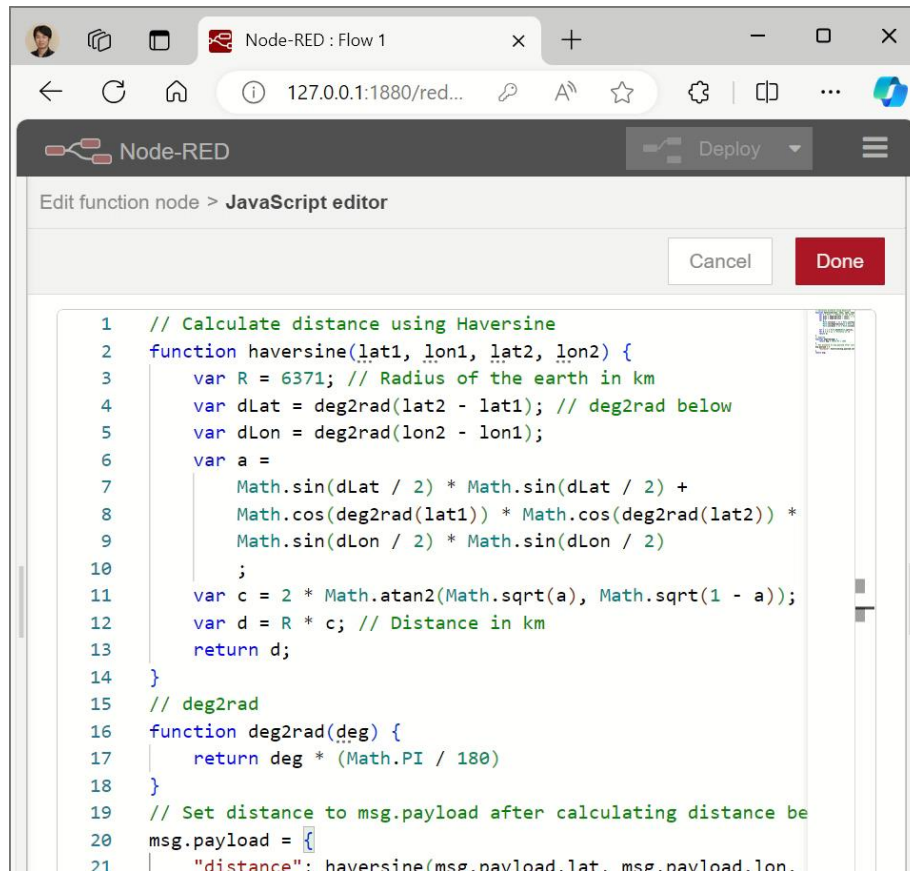
It takes a time  
to write the code :(



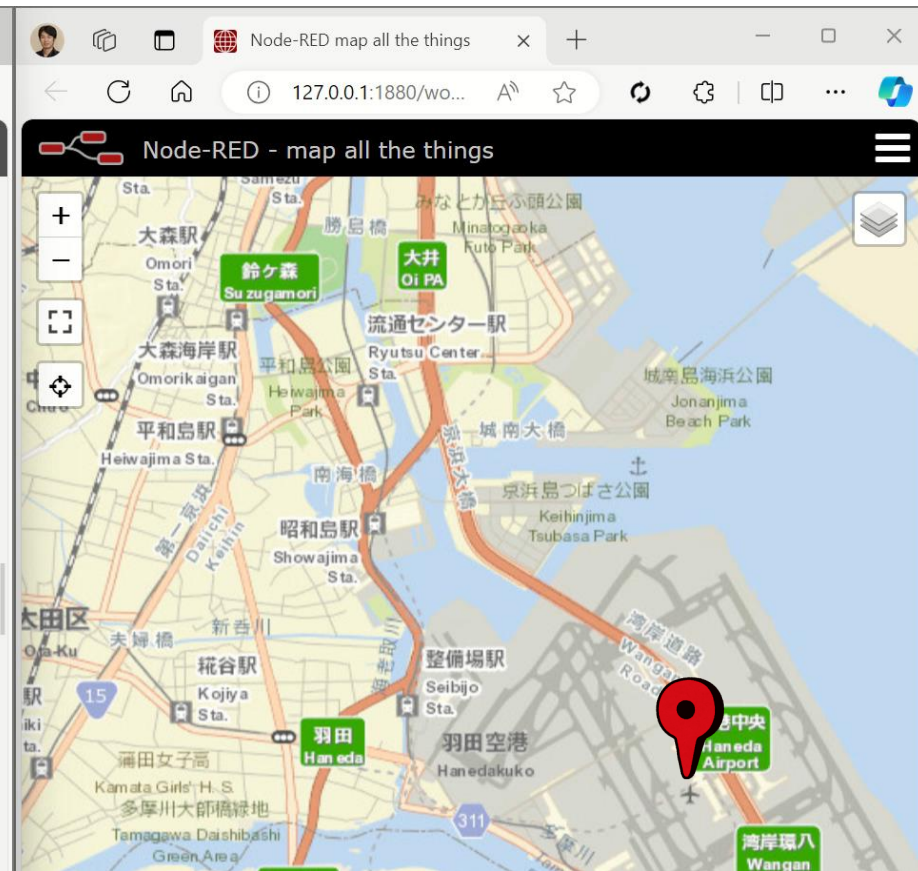
## Demonstration 2: Calculating Distance between Two Locations

```
1 |  
2 return msg;
```

# Demonstration 2: Calculating Distance between Two Locations



```
1 // Calculate distance using Haversine
2 function haversine(lat1, lon1, lat2, lon2) {
3   var R = 6371; // Radius of the earth in km
4   var dLat = deg2rad(lat2 - lat1); // deg2rad below
5   var dLon = deg2rad(lon2 - lon1);
6   var a =
7     Math.sin(dLat / 2) * Math.sin(dLat / 2) +
8     Math.cos(deg2rad(lat1)) * Math.cos(deg2rad(lat2)) *
9     Math.sin(dLon / 2) * Math.sin(dLon / 2)
10   ;
11   var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
12   var d = R * c; // Distance in km
13   return d;
14 }
15 // deg2rad
16 function deg2rad(deg) {
17   return deg * (Math.PI / 180)
18 }
19 // Set distance to msg.payload after calculating distance be
20 msg.payload = {
21   "distance": haversine(msg.payload.lat, msg.payload.lon,
```

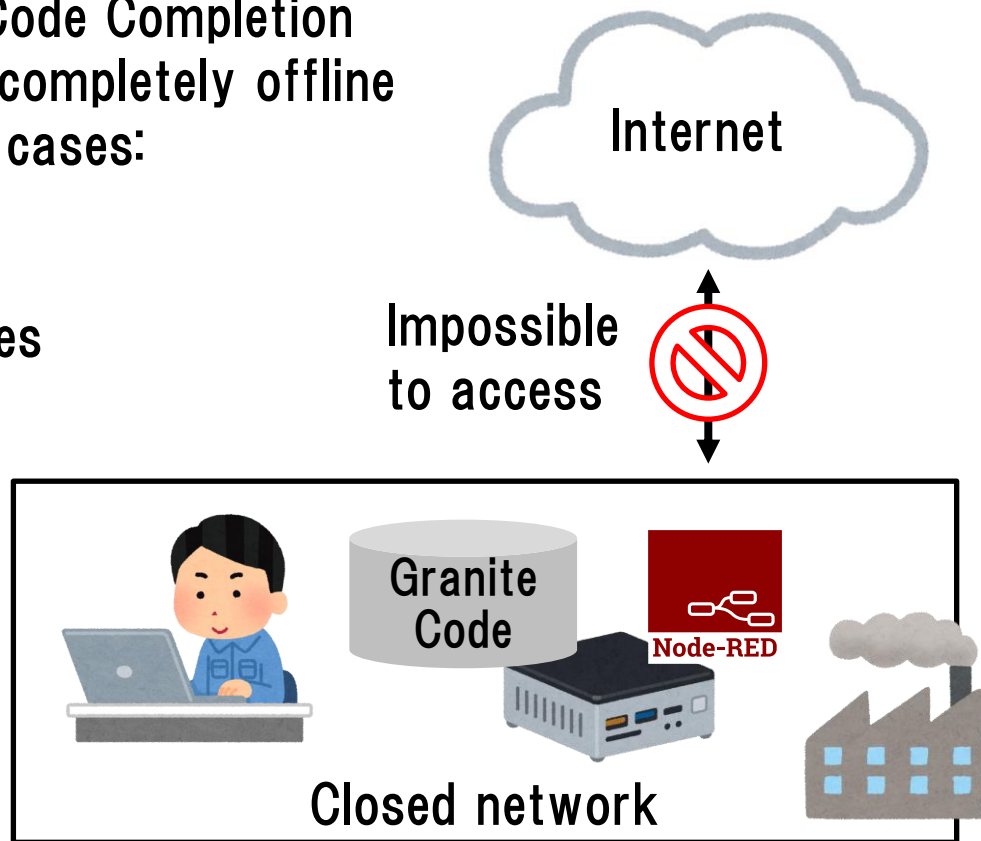




# Benefits of Offline Use

The combination of Node-RED with Code Completion and the Granite Code Model enables completely offline development that is suitable for use cases:

- Mission-critical systems under the strict compliance policies
- Factory automation systems inside a closed network



---

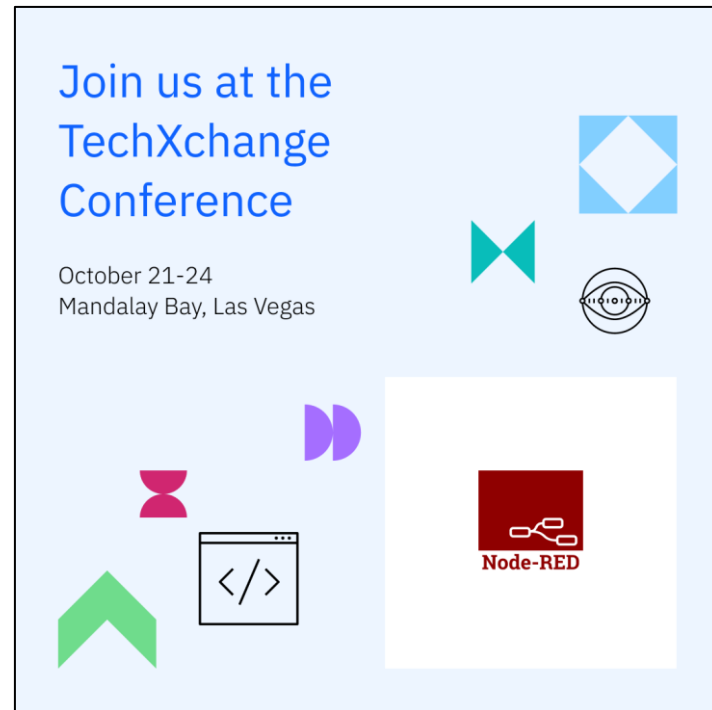
## Conclusion

**Coding with LLM increases productivity in Node-RED.**

- **Code Completion plugin provides the latest developer experience for Node-RED.**
- **In addition, the Granite code model enables offline use and reduces compliance risks.**

**Please look forward to the future developments.**

- **Boosting Productivity of Node-RED with Large Language Model**
  - October 21 12:00PM - 12:30PM
  - South, Reef D, Level 2
- **Node-RED hands-on with Open Technologies**
  - October 22 3:00PM - 4:00PM
  - South, Banyan F, Level 3
- **Tips for controlling HVAC and lighting with a voice from xR using Code Engine and watsonx.ai!**
  - Thursday, October 24 11:30AM - 12:30PM
  - South, Surf C, Level 2



<https://reg.tools.ibm.com/flow/ibm/techxchange24/sessioncatalog/page/sessioncatalog?search=Node-RED>

END



## **Boosting Productivity of Node-RED with Large Language Model**

October 21, 2024  
Kazuhito Yokoi  
Hitachi Academy

- Node-RED, Node.js, and Node Version Manager are trademarks of OpenJS Foundation.
- reTerminal DM is a trademark of Seeed Studio.
- npm is a trademark of npm, Inc.
- GitHub Copilot is a trademark of GitHub, Inc.
- Granite code, code engine and watsonx.ai are trademarks of International Business Machines Corporation.
- ThinkPad is a trademark of Lenovo Corporation.
- macbook is a trademark of Apple Incorporated.
- Visual Studio code and monaco editor are trademarks of Microsoft Corporation.
- ChatGPT is a trademark of OpenAI, Inc.
- Claude is a trademark of Anthropic PBC.
- Other company, product or service names may be trademark or registered trademark of others.