



Node-REDのノード開発ハンズオン

2019/06/14

日立製作所 中央研究所

横井 一仁

ハンズオン開始前に以下のソフトウェアをインストールしておいてください。

- Node.js v10.x LTS
<https://nodejs.org/ja/download>
- Node generator v0.0.4
コマンドプロンプト上で「npm install -g node-red-nodegen」を実行
- IBM Cloudのライトアカウント登録
<http://www.ibm.biz/ibmcloudfree>

下記リポジトリにあるハンズオン説明スライドvol2も開いておいてください。

<https://github.com/kazuhiyokoi/nodegen-handson>

下記URLにアクセスしNode-REDのソースコードをデプロイ

<https://cloud.ibm.com/catalog/starters/node-red-starter>

The screenshot shows the IBM Cloud console for the 'Node-RED Starter' application. The page is titled 'Cloud Foundry アプリの作成' (Create Cloud Foundry Application). The application description states: 'This application demonstrates how to run the Node-RED open-source project within IBM Cloud.' There is a link for '資料の表示' (View Documentation).

Annotations on the screenshot:

- (1-1) 他の開発者と重複しないアプリ名を入力** (Enter an application name that does not overlap with other developers): Points to the 'アプリ名:' field, which contains 'handson-yokoi'.
- (1-2) リージョンは「ダラス」を選択** (Select 'Dallas' for the region): Points to the 'デプロイする地域/ロケーションの選択:' dropdown menu, which is set to 'ダラス' (Dallas).
- (1-3) 「作成」をクリック** (Click 'Create'): Points to the '作成' (Create) button at the bottom right of the page.

Other visible fields and options include:

- ホスト名:** handson-yokoi
- ドメイン:** mybluemix.net
- 組織の選択:** kazuhiro.yokoi.nx@hitachi.com
- スペースの選択:** dev
- タグ:** env:dev, version-1
- 選択済みプラン:** SDK for Node.js™, デフォルト
- Cloudant:** Lite

At the bottom, there is a link: 'ヘルプが必要ですか? IBM Cloud サポートへのお問い合わせ' (Need help? Contact IBM Cloud support).

(2) デプロイが完了するまで4分程待った後、Node-REDのURLへアクセス

The screenshot shows the IBM Cloud console interface. On the left is a sidebar with navigation links: 開始, 概要, ランタイム, 接続, ログ, 自動スケーリング, モニタリング, and API Management. The main content area displays the details for a resource named 'hands-on-yokoi'. It shows a green dot indicating the status is '実行中' (Running). A callout box points to this status with the text: (2-2)「開始中」が「実行中」に変わればデプロイ完了. To the right of the status is a link 'アプリ URL にアクセス' (Access application URL). Another callout box points to this link with the text: (2-3) Node-REDのURLへアクセス. Below this, there is a section titled 'Before you can start coding with Node-RED, you must first configure the editor.' followed by a link 'Open the Node-RED flow editor'. Below this are two numbered steps: ① After your application starts, click **Routes URL** or enter the following URL in a browser to display the Node-RED landing page. The URL is shown in a black box: `http://<varname><yourhost></varname>.mybluemix.net`. ② Click **Go to your Node-RED flow editor** to open the browser-based flow editor. The editor makes it easy to connect devices, APIs, and online services by using the wide range of nodes that are included in the palette. At the bottom, there is a link 'Customizing your Node-RED instance'.

IBM Cloud

リソース・リスト /

hands-on-yokoi ● 実行中 [アプリ URL にアクセス](#) 経路

組織: kazuhito.yokoi 場所: Dallas

(2-2)「開始中」が「実行中」に変わればデプロイ完了

(2-3) Node-REDのURLへアクセス

Before you can start coding with Node-RED, you must first configure the editor.

[Open the Node-RED flow editor](#)

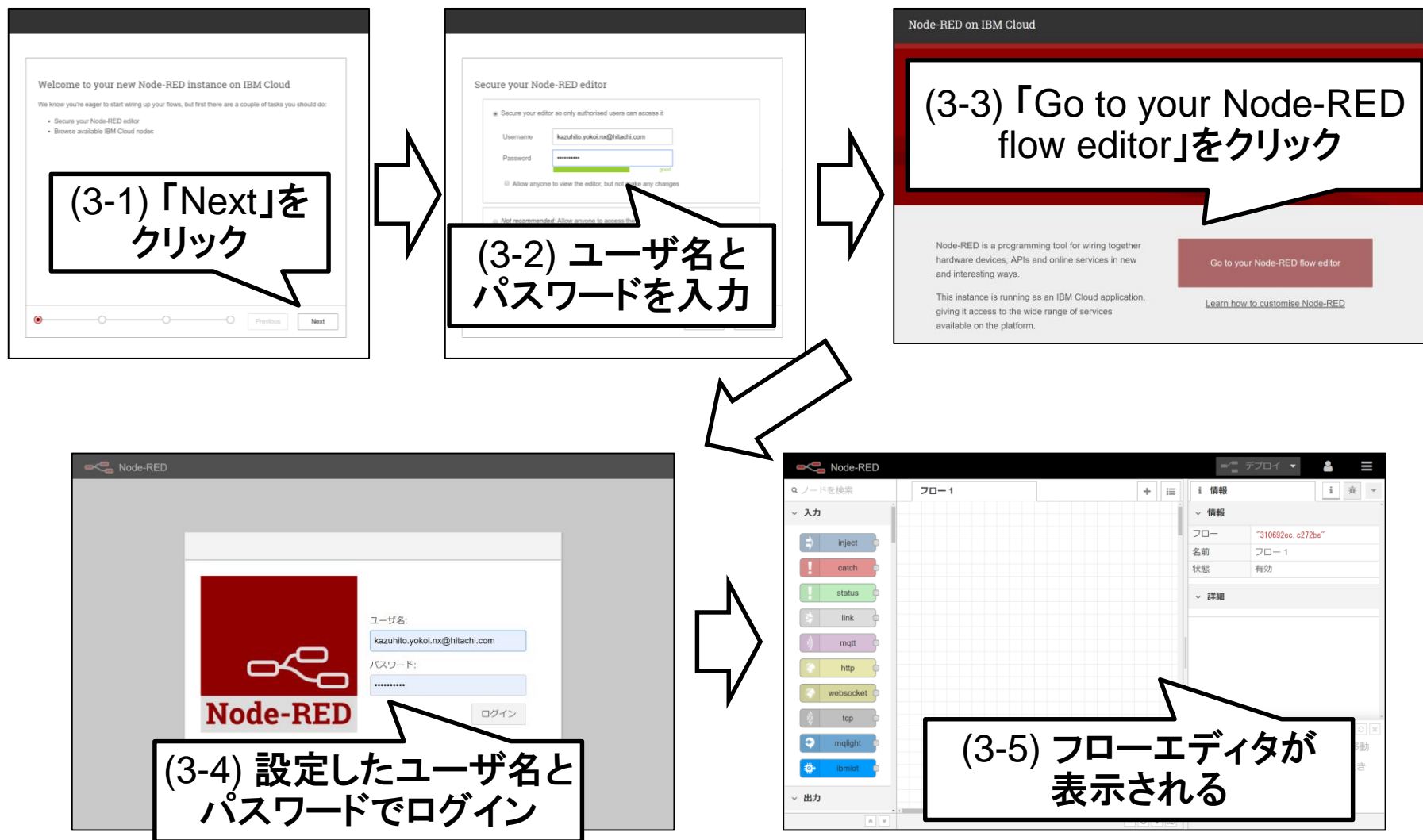
① After your application starts, click **Routes URL** or enter the following URL in a browser to display the Node-RED landing page.

`http://<varname><yourhost></varname>.mybluemix.net`

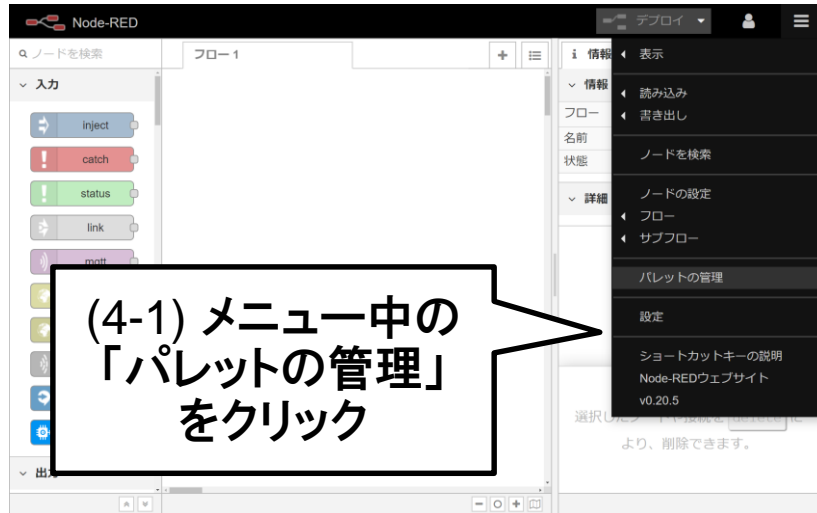
② Click **Go to your Node-RED flow editor** to open the browser-based flow editor. The editor makes it easy to connect devices, APIs, and online services by using the wide range of nodes that are included in the palette.

[Customizing your Node-RED instance](#)

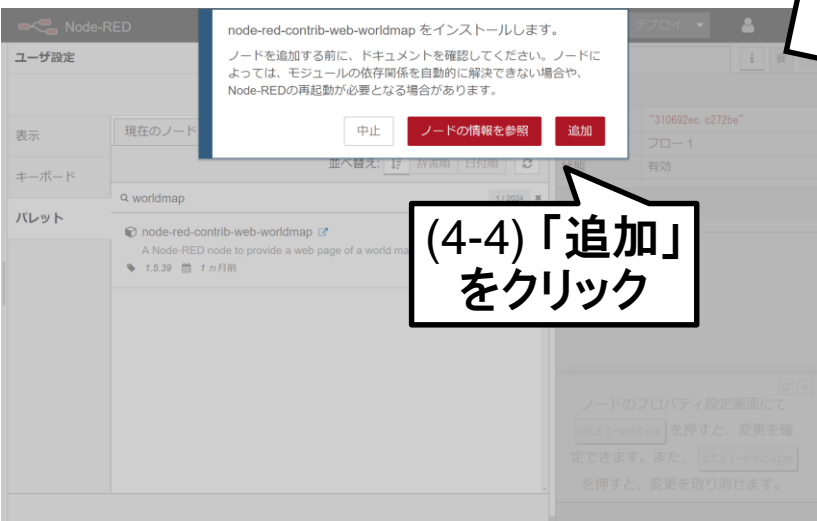
(3) ウィザードに従い、Node-REDの初期セットアップを実施



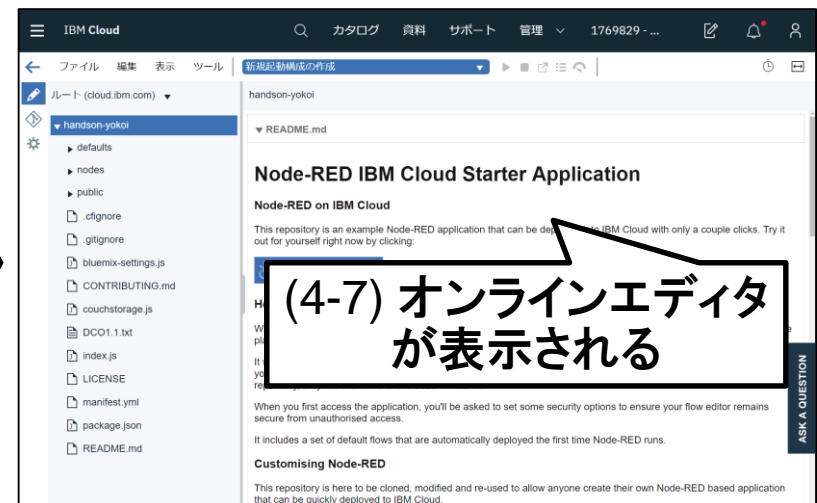
(4) 地図表示を行うworldmapノードをインストール



(4-2) 「パレット」、「ノードを追加」と遷移し、検索キーワードとして「worldmap」を入力



(4) IBM Cloudの管理画面に戻り、Node-REDのカスタマイズ環境を用意



コンテンツ

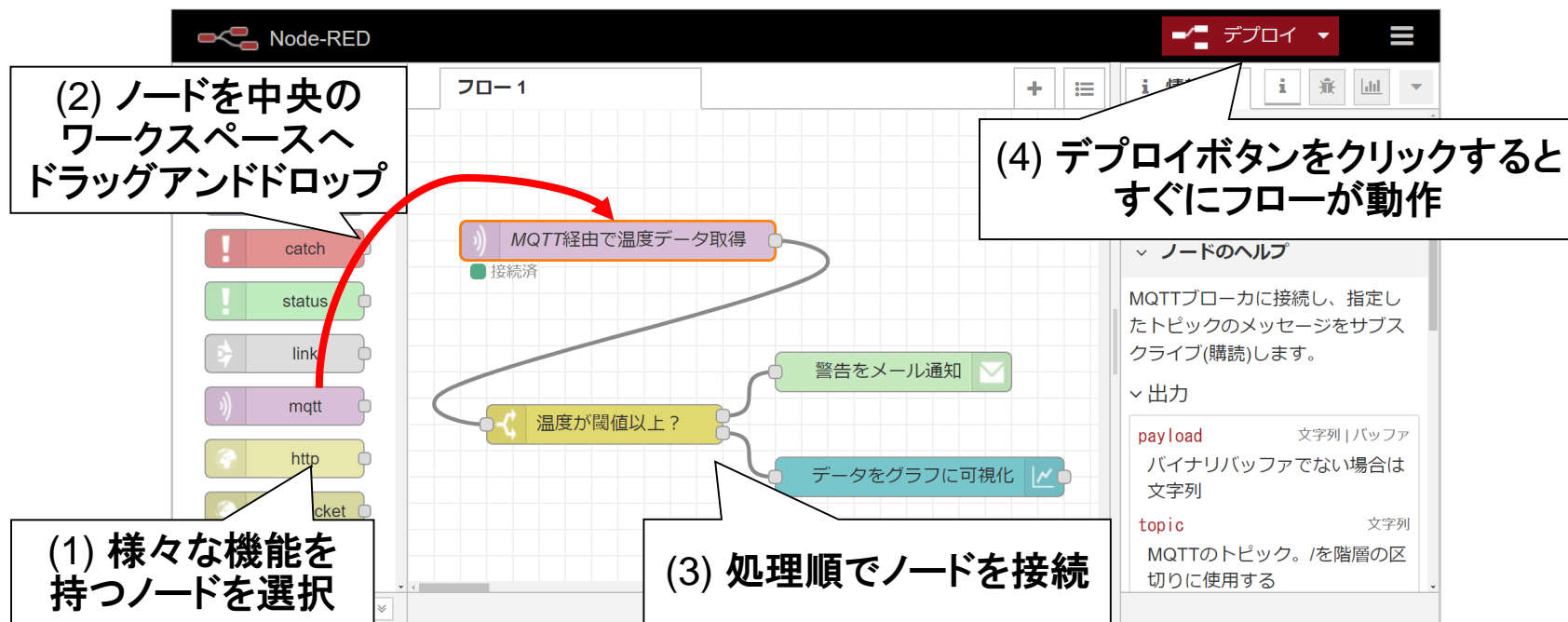
- Node generatorの説明
- ハンズオン: Open APIドキュメントからオリジナルノードを作成



Node generatorの説明

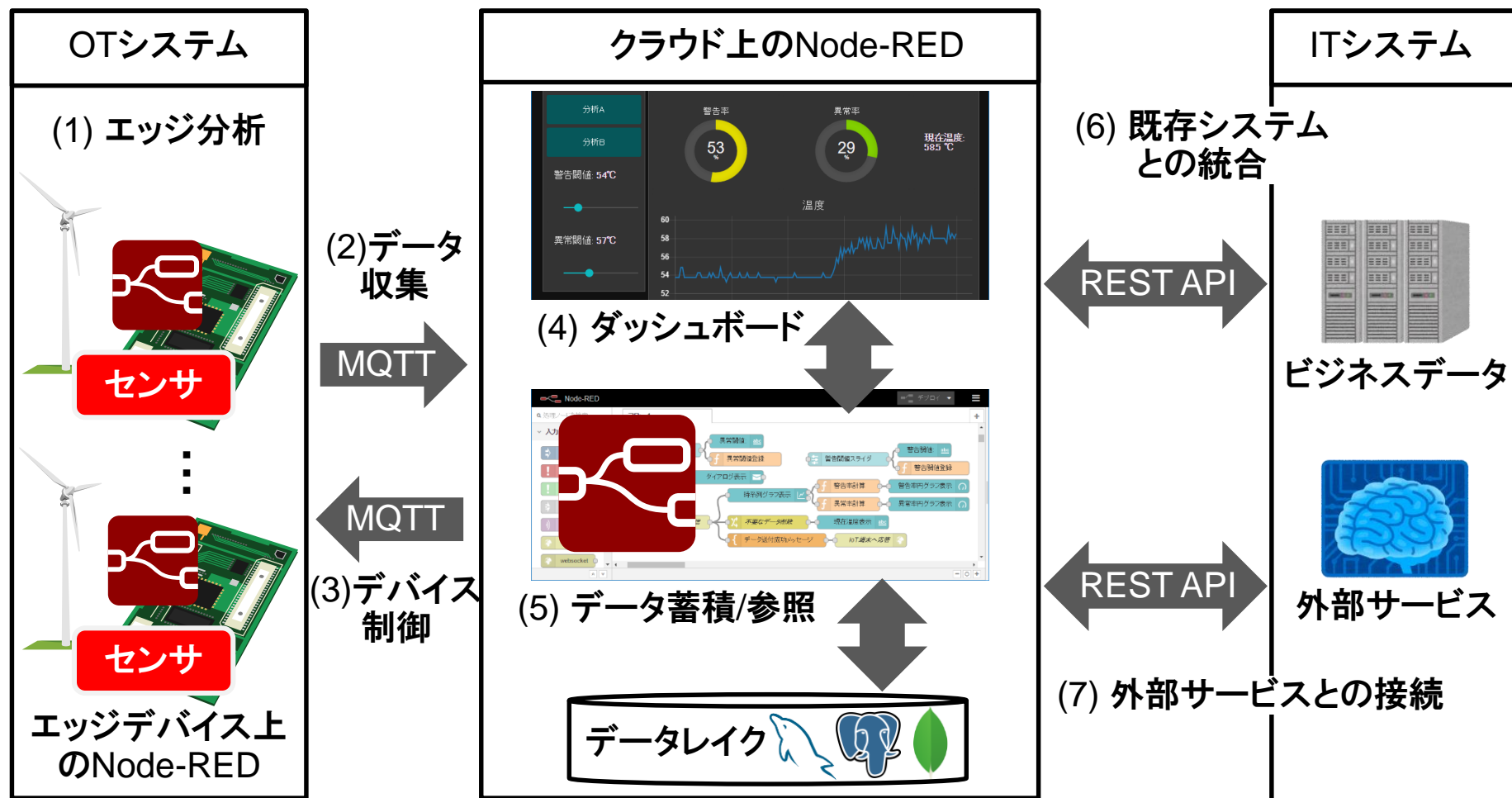
IoTアプリケーション向けのビジュアルプログラミングツール

- 迅速な開発を可能とするフローベースドプログラミング環境
- 新機能を追加できる様々なノード(部品)が存在
- Linux Foundation(JS Foundation)管理下のオープンソースソフトウェア



Node-REDフローエディタ

様々な外部システムやデバイス、ライブラリとつなぐノードを活用することで素早くアプリケーションを開発できる。



- 自社/他社サービスとの接続のため、オリジナルノードの開発が必要となることがある
- 1個のノードは数千行のコード量であるため、ノード開発時間大

数千行のコード

ノード開発時間大

JavaScript
ファイル

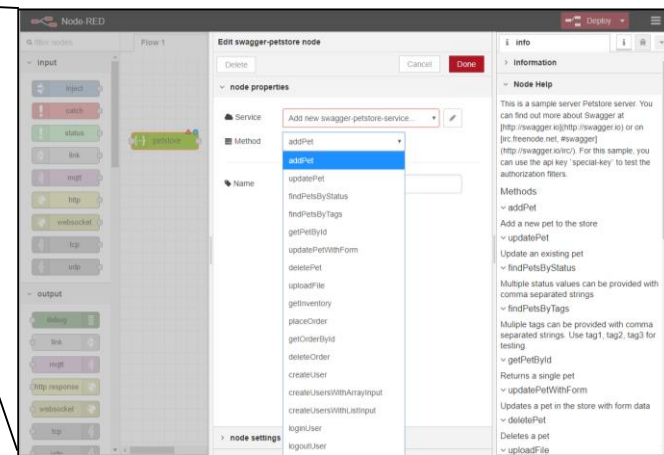
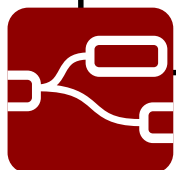
HTML
ファイル



オリジナルノード

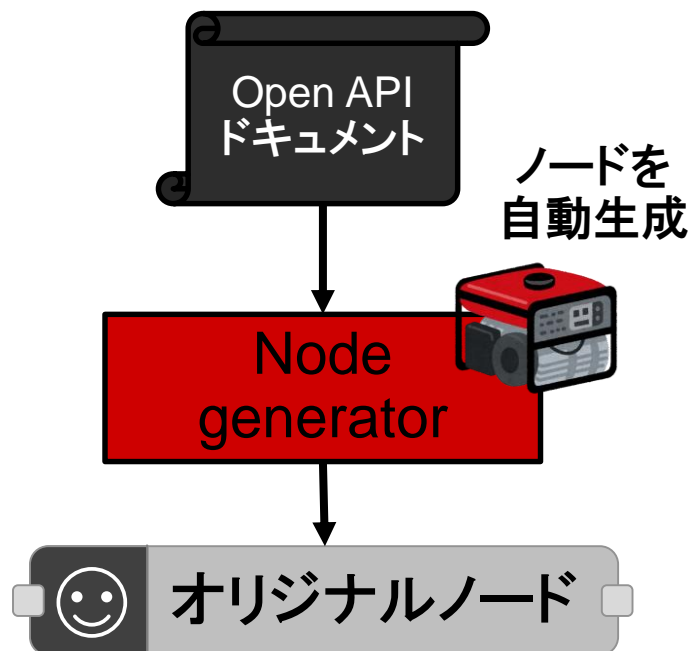
Node.js上
で実行

Node-RED
ランタイム

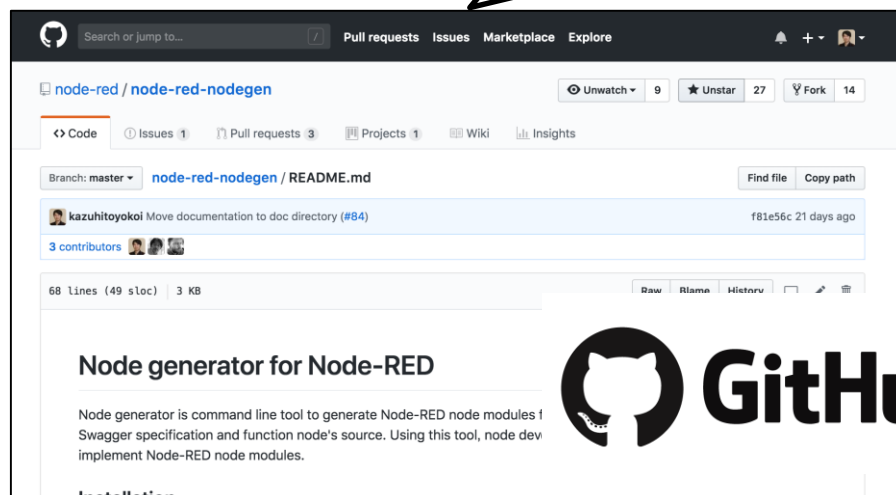


Node-REDフローエディタ
のUI

- Node generatorは、Open APIドキュメントやfunctionノードのソースコードから、ノードを自動生成できるツール
- Linux Foundation (JS Foundation) のオープンソースソフトウェア

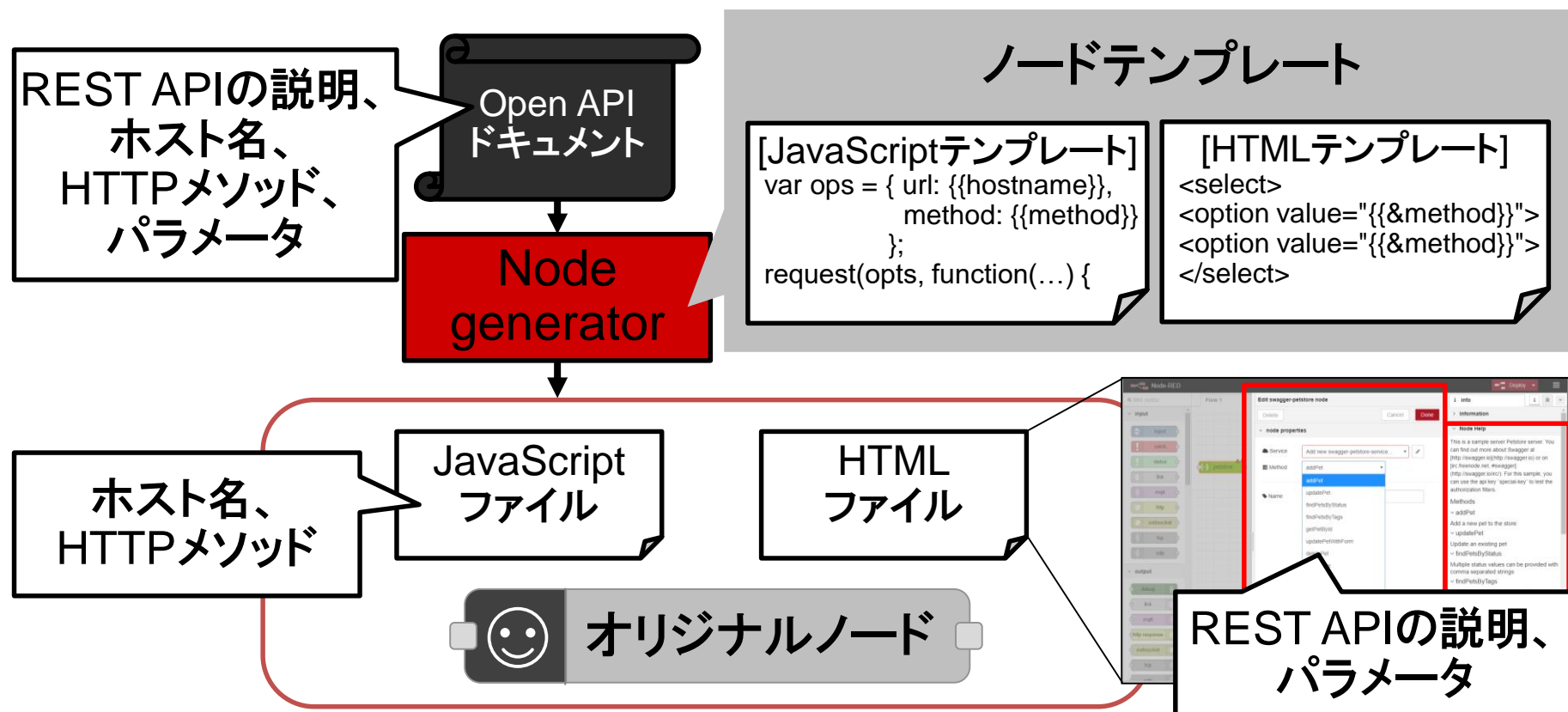


“Node-RED Node generator”
で検索すると見つかります

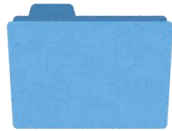


<https://github.com/node-red/node-red-nodegen>

- Node generatorは、内部にてREST API固有の情報を除いた共通のクライアントコードを持つノードテンプレートを保持
- Node generatorは、本ノードテンプレートにREST API固有の情報を挿入することでノードを生成



Node generatorは、以下のディレクトリ、ファイルを出力




node-red-contrib-<ノード名>

- └ node.js (サーバ側のNode.jsで動作するプログラム)
- └ node.html (ノードの見た目、ノードプロパティUI、情報タブを記述)
- └ locales
 - └ en-US — node.json (ノードプロパティUIの英語メッセージ)
 - └ ja — node.json (ノードプロパティUIの日本語メッセージ)
 - └ zh-CN — node.json (ノードプロパティUIの中国語メッセージ)
 - └ de-DE — node.json (ノードプロパティUIのドイツ語メッセージ)
- └ icons
 - └ icon.png (ノードのアイコンファイル)
- └ test
 - └ node_spec.js (ノードのテストケース)
- └ LICENSE (ライセンスファイル)
- └ package.json (ノードのパッケージ情報)
- └ README.md (ノードの使い方等のドキュメント)

IBMが画像認識、音声認識、自然言語処理を行うDockerコンテナにアクセスするノードをNode generatorを用いて開発

IBM Developer Topics ▼ Community ▼ More open source at IBM ▼

 IoT

CODE
Models
Code Patterns
Open Projects

CONTENT
Announcements
Articles
Series
Tutorials
Videos

TUTORIAL

Use Node-RED Node Generator to create new nodes from APIs and services

Access a RESTful deep learning microservice from your Node-RED flows

Nick Kasten | Published March 25, 2019

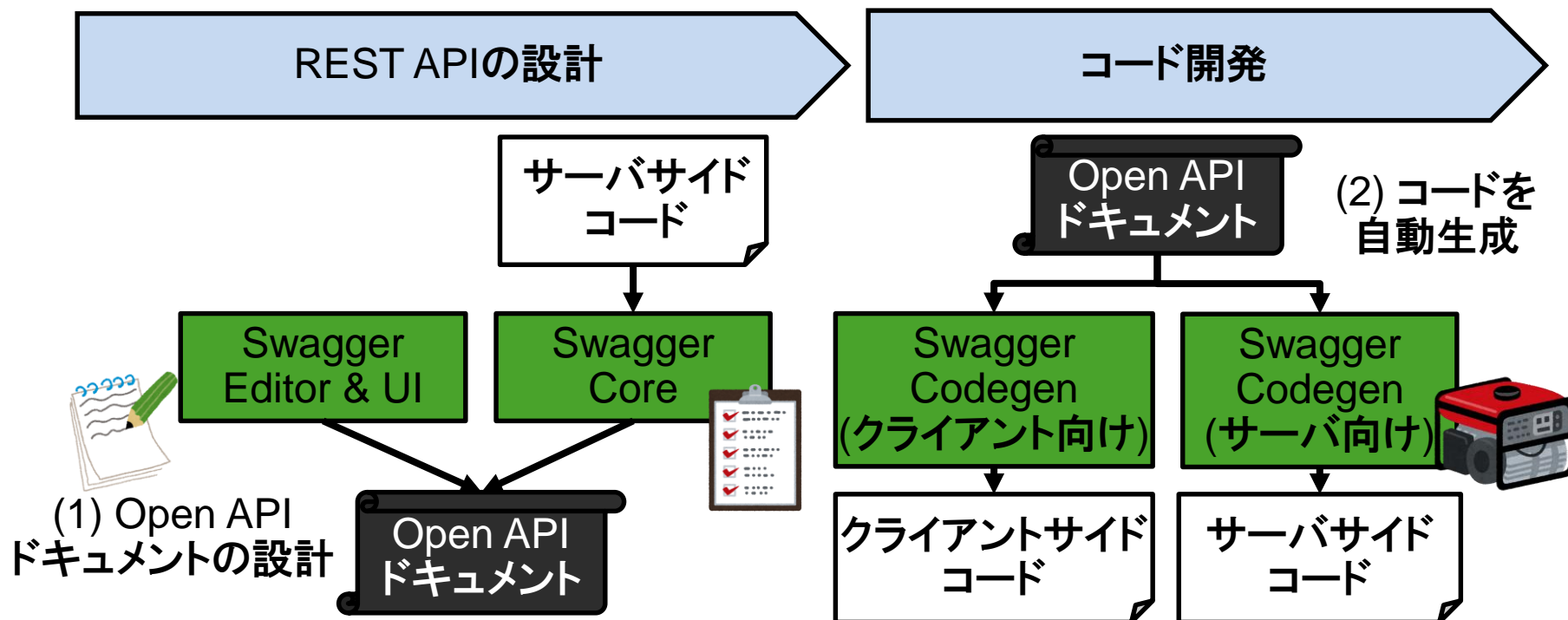
Node generatorの紹介ページ

<https://developer.ibm.com/tutorials/use-node-red-node-generator-to-create-new-nodes-from-apis-and-services/>



ハンズオン: Open APIドキュメントから オリジナルノードを作成

- REST API定義を記述する業界標準のフォーマット
- Open APIプロジェクトにより提供されているツール
 - Swagger Editor & UI: Open APIドキュメントを記述するためのエディタ、REST APIドキュメントの自動生成
 - Swagger Core: サーバサイドのコードからOpen APIドキュメントを自動生成
 - Swagger Codegen: Open APIドキュメントからコードを自動生成



Open APIドキュメントには、REST APIのアクセス方法
についての情報が記載されている。

REST API
の説明

```
swagger: '2.0'
info:
  description: 'This is a sample server Petstore server.'
  version: 1.0.0
  title: Swagger Petstore
  license:
    name: Apache 2.0
    url: 'http://www.apache.org/licenses/LICENSE-2.0.html'
host: petstore.swagger.io
basePath: /v2
paths:
  /store/inventory:
    get:
      summary: Returns pet inventories by status
      description: Returns a map of status codes to quantities
      operationId: getInventory
      produces:
        - application/json
      parameters: []
      responses:
        '200':
          description: successful operation
```

ホスト名

HTTP
メソッド

パラメータ

宇宙ステーションを位置情報を取得するノードを作成

- (1) 位置情報取得APIのOpen APIドキュメントからノードを自動生成
- (2) 世界地図上に宇宙ステーションの位置を表示するフローを作成

宇宙ステーション



REST API

ウェブブラウザ



2. 緯度経度
を取得

3. 位置情報データの
フォーマット変換

4. 世界地図上に
ピンを配置

Open API
ドキュメント

1. ノード生成、
インストール

Open APIドキュメント
からノードを自動生成

- (1-1) ハンズオンのリポジトリ(<https://github.com/kazuhitoyokoi/nodegen-handson>)からOpen APIドキュメント(swagger.yaml)をホームディレクトリ(C:¥Users¥<ユーザ名>¥)へダウンロード

```
swagger: '2.0'
info:
  description: The International Space Station location
  version: 4.5.1
  title: ISS Location
  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
host: api.open-notify.org
basePath: /
schemes:
  - http
paths:
  /iss-now.json:
    get:
      summary: Current ISS location over Earth (latitude/longitude)
      description: This is a simple API to return the current location of the ISS.
      operationId: ISSLocationNow
      produces:
        - application/json
      responses:
        '200':
          description: successful operation
```

ダウンロードしたOpen APIドキュメントの内容

(1-2) コマンドプロンプトを開き、ホームディレクトリに移動

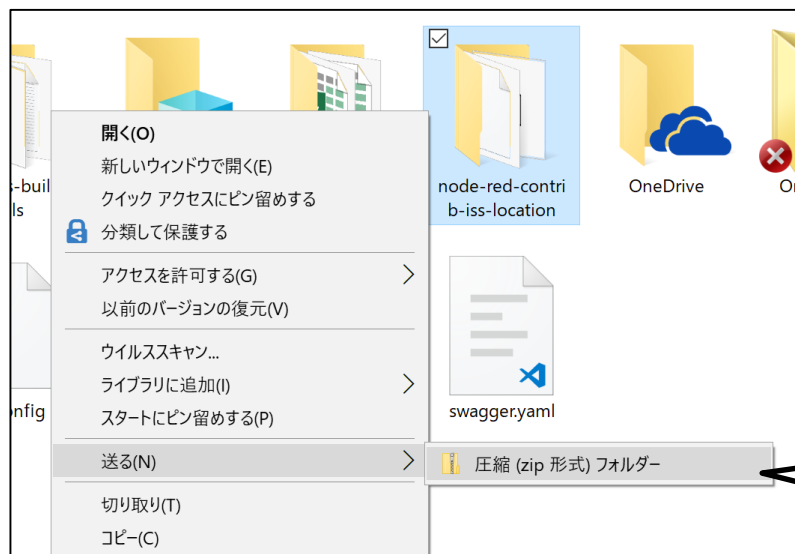
```
> cd C:\Users\<ユーザ名>
```

(1-3) node-red-nodegenコマンドを実行しノードを生成

```
> node-red-nodegen swagger.yaml
```

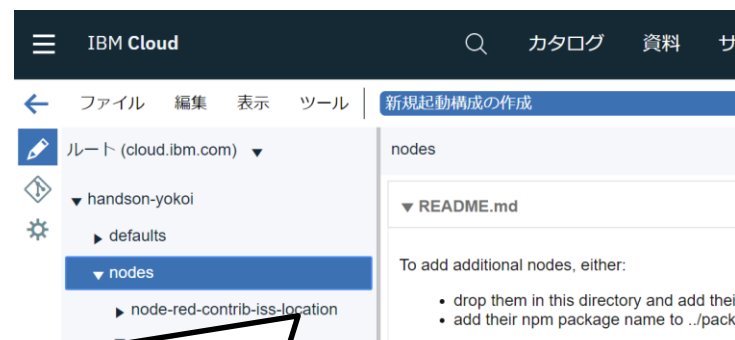
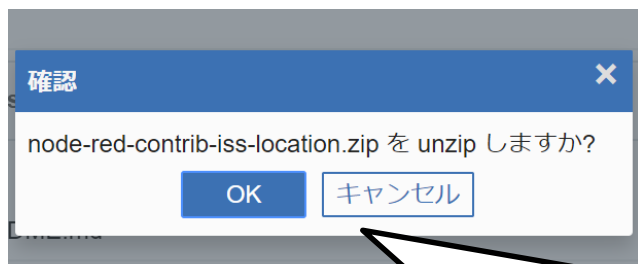
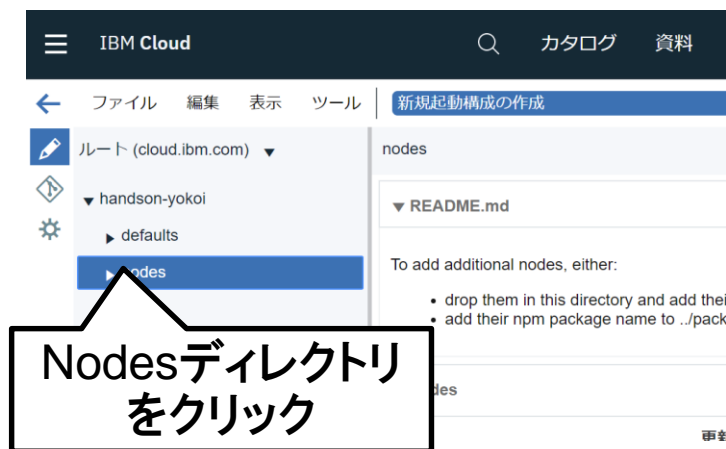
-> ノードを構成するファイル群が入ったディレクトリ
node-red-contrib-iss-locationが生成される

(1-4) エクスプローラ上で生成されたディレクトリをzip圧縮



「送る」の機能を用いてzip圧縮

(1-5) Node-REDのソースコード中のカスタムノードを置くディレクトリnodesに生成したノードが入ったzipファイルをアップロードして展開



(1-5) nodes内に追加したオリジナルノードを依存関係として追加

The screenshot shows the IBM Cloud IDE interface. The top bar includes the IBM Cloud logo, search, catalog, resources, support, and management tabs, along with a user profile icon and a notification bell. The main workspace displays a file explorer on the left and a code editor on the right. The file explorer shows a list of files including .cfignore, .gitignore, bluemix-settings.js, CONTRIBUTING.md, couchstorage.js, package.json, and README.md. The code editor shows the content of package.json, which includes a dependencies section. A callout box (3) points to the 'File' menu, indicating the user should click 'Save' to save the file. Another callout box (4) points to the Git icon in the file explorer, indicating the user should click it. A third callout box (2) points to the 'dependencies' section of the package.json file, indicating the user should add the 'node-red-contrib-iss-location' dependency. A fourth callout box (1) points to the 'package.json' file in the file explorer, indicating the user should click it.

(1) Package.json をクリック

(2) Dependenciesの中に
「node-red-contrib-iss-location」:
「file://./nodes/node-red-contrib-iss-location」,
を追加

(3)「ファイル」->「保存」から
ファイルの変更を保存

(4)Gitアイコン
をクリック

変更内容をGitlab上のソースコードへ反映

The image illustrates the process of committing and pushing code changes to a GitLab repository via the IBM Cloud interface. The steps are as follows:

- (1) コミットメッセージを追加**: Adding a commit message to the commit message field.
- (2) コミットをクリック**: Clicking the **コミット** (Commit) button.
- (3) プッシュをクリック**: Clicking the **プッシュ** (Push) button.
- (4) 矢印ボタンをクリック**: Clicking the arrow button to push the changes to the remote repository.

デプロイの状況を確認

The screenshot shows the IBM Cloud console interface. At the top, there's a navigation bar with 'IBM Cloud', search, and various menu items. Below, the 'handson-yokoi' toolchain is selected, showing its resources: Issues, Git, Delivery Pipeline, and Eclipse Orion Web IDE. A callout box points to the 'Delivery Pipeline' tile with the text '(1) Delivery Pipelineをクリック'.

Below this, the 'handson-yokoi | Delivery Pipeline' page is shown. It displays two stages: 'Build Stage' and 'Deploy Stage'. The 'Build Stage' shows a successful build job. The 'Deploy Stage' shows a job in progress. A callout box points to the 'Deploy Stage' with the text '(2)「実行中」が「成功」に変われば、デプロイ成功'.

Build Stage

スステージは成功

最終入力 Git URL

最終コミット担当者 Kazuhito Yokoi 8 分前
宇宙ステーションの位置情報取得 ノード...

ジョブ ログおよび履歴の表示

Build 成功 現在

Deploy Stage

スステージの実行中...

最終入力 ステージ: Build Stage / ジ...

Build 1

ジョブ ログおよび履歴の表示

Deploy 実行中 現在

(2-2) Node-REDフローエディタをリロード

-> パレットの「機能」カテゴリの中にiss locationノードが登場

(2-3) 生成されたノードをワークスペースにドラッグアンドドロップし、ノードプロパティUI上でメソッドを選択

The screenshot displays the Node-RED web interface. On the left, the 'Node Palette' is visible with a search bar and a list of nodes. The 'iss location' node is highlighted with a red circle. A red arrow points from this node to the workspace. In the workspace, the 'iss-location' node is placed on a grid. To the right, the 'iss-location ノードを編集' (Edit iss-location node) panel is open. It shows the 'メソッド' (Method) dropdown menu with 'ISSLocationNow' selected. A callout box with the text 'メソッドを選択' (Select method) points to the dropdown menu. The '名前' (Name) field is also visible.

(2-4) injectノード、生成したノード、debugノードを順に接続したフローを作成し、デプロイボタンをクリック

(2-5) injectノードの右にあるボタンをクリックして、フローを実行
→ デバッグタブに宇宙ステーションの現在の緯度(latitude)、経度(longitude)が出力される

The screenshot displays the Hitachi Flow Builder interface. On the left, a workflow canvas shows three nodes connected in sequence: a blue 'タイムスタンプ' (Timestamp) node, a green 'iss-location' node, and a green 'msg.payload' node. The 'msg.payload' node is highlighted with a red dashed border. On the right, the 'デバッグ' (Debug) tab is active, showing a log entry for the flow execution. The log entry includes a timestamp, a node ID, and a message object containing the ISS position data.

フロー1

タイムスタンプ

iss-location

msg.payload

デバッグ

2019/4/17 11:50:00 node: bfdb38f7.544468

msg.payload : Object

- object
 - timestamp: 1555469300
 - message: "success"
 - iss_position: object
 - latitude: "-45.9101"
 - longitude: "2.8959"

REST APIから取得した位置情報が表示される

(2-6) injectノードのプロパティUIで以下の繰り返し処理を設定

#	項目	設定値
1	繰り返し	指定した時間間隔
2	時間間隔	1秒

Node-RED

フロー 1

タイムスタンプ

iss-location

inject ノードを編集

削除 中止 完了

プロパティ

ペイロード

トピック

Node-RED起動 0.1 秒後、以下を行う

繰り返し 指定した時間間隔

時間間隔 1 秒

名前

1秒間隔でフローを実行するよう設定

(2-7) 生成したノードの後ろにchange、world mapノード(左側に端子)を接続

(2-8) changeノードに変数の代入を設定し、デプロイボタンをクリック

#	プルダウンメニュー	1つ目の設定値	対象の値
1	値の代入	msg.payload.lat	msg.payload.iss_position.latitude
2	値の代入	msg.payload.lon	msg.payload.iss_position.longitude
3	値の代入	msg.payload.name	msg.payload.timestamp

The screenshot shows the Node-RED interface with a flow named 'フロー 1'. The flow consists of the following nodes in sequence:

- timestamp** (blue node)
- iss-location** (green node)
- set msg.payload** (yellow node)
- worldmap** (orange node)

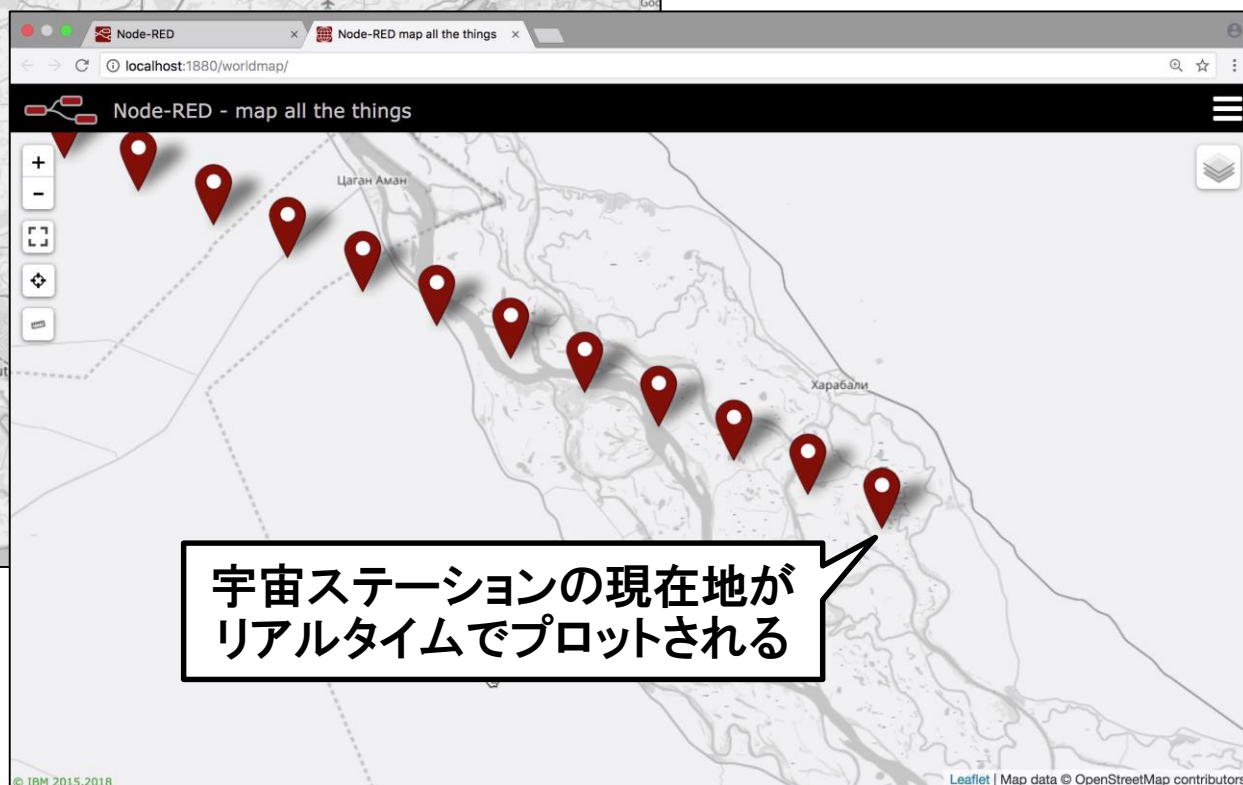
A callout points to the **set msg.payload** node, with the text: **プルダウンメニューから「msg.」を選択** (Select 'msg.' from the dropdown menu).

Another callout points to the **change** node configuration panel, which is titled 'change ノードを編集'. It shows three rules for variable assignment:

- Rule 1: 値の代入 (Value Assignment) for msg.payload.lat, with 対象の値 (Target Value) msg.payload.iss_position.latitude.
- Rule 2: 値の代入 (Value Assignment) for msg.payload.lon, with 対象の値 (Target Value) msg.payload.iss_position.longitude.
- Rule 3: 値の代入 (Value Assignment) for msg.payload.name, with 対象の値 (Target Value) msg.payload.timestamp.

A callout points to the configuration panel with the text: **3つの変数を代入** (Assign 3 variables).

(2-9) <https://<アプリ名>.mybluemix.net/worldmap>を開くと、
世界地図が表示され、宇宙ステーションの位置を確認できる



- ノードが所属するカテゴリを変更

デフォルトの「機能」カテゴリ以外にノードを所属させる場合に指定する

例) 「分析」カテゴリを指定

```
> node-red-nodegen swagger.yaml --category analysis
```



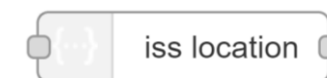
オプションの
ハイフンは2つ

- ノードの色を変更

--colorオプションに16進数で色を指定する

例) ノードを白色に設定

```
> node-red-nodegen swagger.yaml --color FFFFFFFF
```

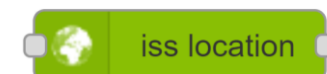


- ノードのアイコンを変更

--iconオプションにPNGファイルやNode-RED標準アイコン名を指定する

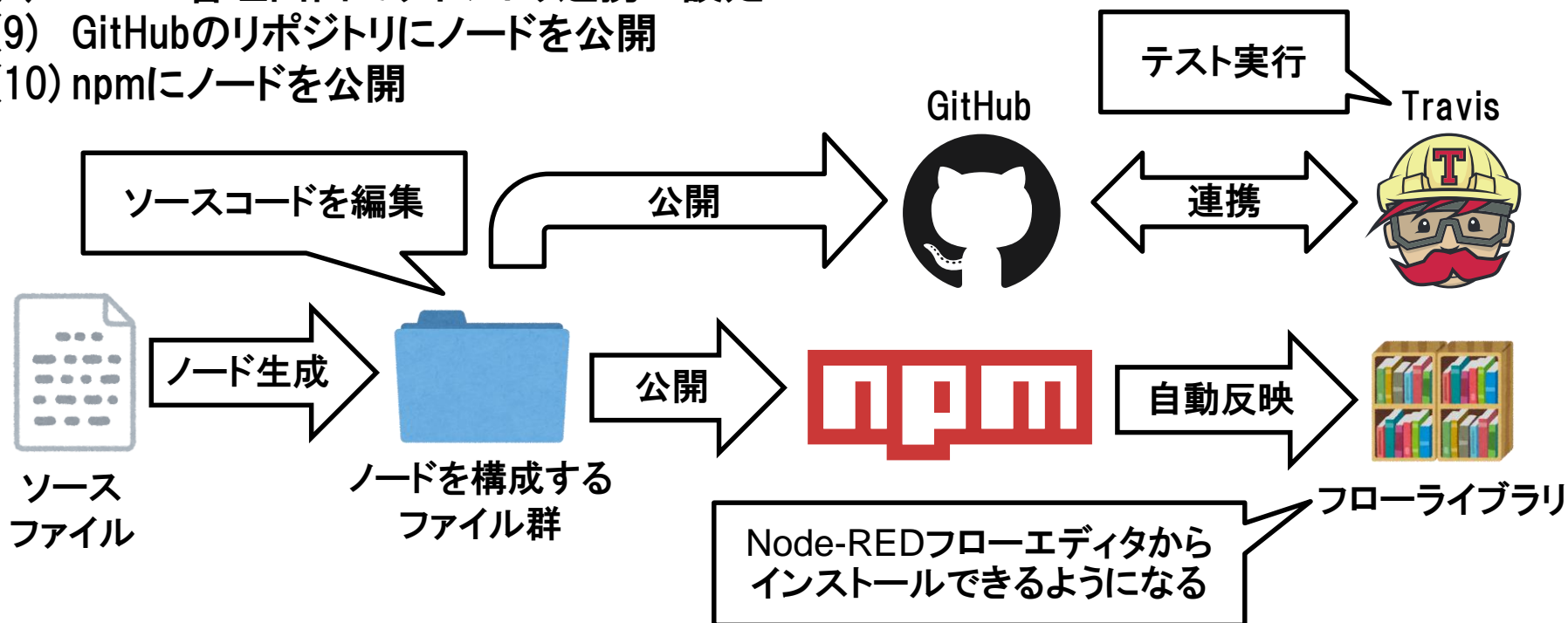
例) 地球のアイコンを設定

```
> node-red-nodegen swagger.yaml --icon white-globe
```



補足: フローライブラリにノードを登録する手順

- (1) Node generatorで「--keywords node-red」オプションでキーワードを指定してノード生成
- (2) 情報タブに表示するノードの使い方をHTMLファイル(node.html)に記載
- (3) ノードの概要やインストール方法のドキュメント(README.md)を記載
- (4) ノードプロパティUIの日英メッセージのファイル(locales/<言語>/node.json)を編集
- (5) ノードのテストケース(test/node_spec.js)を追加
- (6) Travis設定ファイル(.travis.yml)を追加
- (7) packages.jsonにGitHubリポジトリのURLを追加
- (8) Travis管理画面でリポジトリ連携の設定
- (9) GitHubのリポジトリにノードを公開
- (10) npmにノードを公開





最後に

- オリジナルノードを公開し、Node-REDコミュニティを盛り上げましょう
- ノード開発のサポートが必要な場合は、Node-RED公式Slackの#nodegenチャンネルにご質問をお願いします(日本語可)

Node-RED Library

Find new nodes, share your flows and see what's popular

Search library

☐ flows ☒ nodes

2024 of 3277 things

Sort by:
☒ recent
☐ downloads
☐ rating

node-red-contrib-ui-media	@senx/node-red-contrib-warpscript	node-red-contrib-discovery-insert-temp
A simple NODE to allow images and videos on the dashboard	Node-RED node for executing WarpScript™	A node for inserting documents into the watson discovery service.
v1.1.0 106 node	v1.0.2 25 node	v0.1.9 32 node

フローライブラリ (<https://flows.nodered.org>)

END



Node-REDのノード開発ハンズオン

2019/06/14

日立製作所 中央研究所

横井 一仁

HITACHI
Inspire the Next 

- IBMは、International Business Machines Corporationの登録商標です。
- GitHubは、GitHub Incorporatedの登録商標です。
- npmは、npm Incorporatedの登録商標です。
- Slackは、Slack Technologies Inc.の登録商標です。