



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>

(1) 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 are three callout boxes with instructions:

- (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): The 'アプリ名:' (Application Name) field is filled with 'handson-yokoi'.
- (1-2) リージョンは「ダラス」を選択** (Select 'Dallas' for the region): The 'デプロイする地域/ロケーションの選択:' (Select deployment region/location) dropdown is set to 'ダラス' (Dallas).
- (1-3) 「作成」をクリック** (Click 'Create'): The '作成' (Create) button at the bottom right is highlighted.

Other visible fields include:

- ホスト名:** (Host Name) 'handson-yokoi'
- ドメイン:** (Domain) 'mybluemix.net' (dropdown)
- 組織の選択:** (Select organization) 'kazuhiro.yokoi.nx@hitachi.com'
- スペースの選択:** (Select space) 'dev' (dropdown)
- タグ:** (Tags) 'env:dev, version-1' (example: 'env:dev, version-1')
- 選択済みプラン:** (Selected plan) 'SDK for Node.js™' with 'デフォルト' (Default) selected.
- Cloudant:** 'Lite' (dropdown)

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、アプリのURLへアクセス

The screenshot shows the IBM Cloud dashboard for a service named 'hands-on-yokoi'. The status is '実行中' (Running), indicated by a green dot. A callout box (2-1) points to this status, stating: 「(2-1)「開始中」が「実行中」に変わればデプロイ完了」 (When the status changes from 'Starting' to 'Running', deployment is complete). Another callout box (2-2) points to the 'アプリ URL にアクセス' (Access application URL) link, stating: 「(2-2) アプリのURLへアクセス」 (Access the application URL). The main content area provides instructions on how to access the Node-RED flow editor, including a URL template: `http://<varname><yourhost></varname>.mybluemix.net`.

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

(2-1)「開始中」が「実行中」に変わればデプロイ完了

(2-2) アプリのURLへアクセス

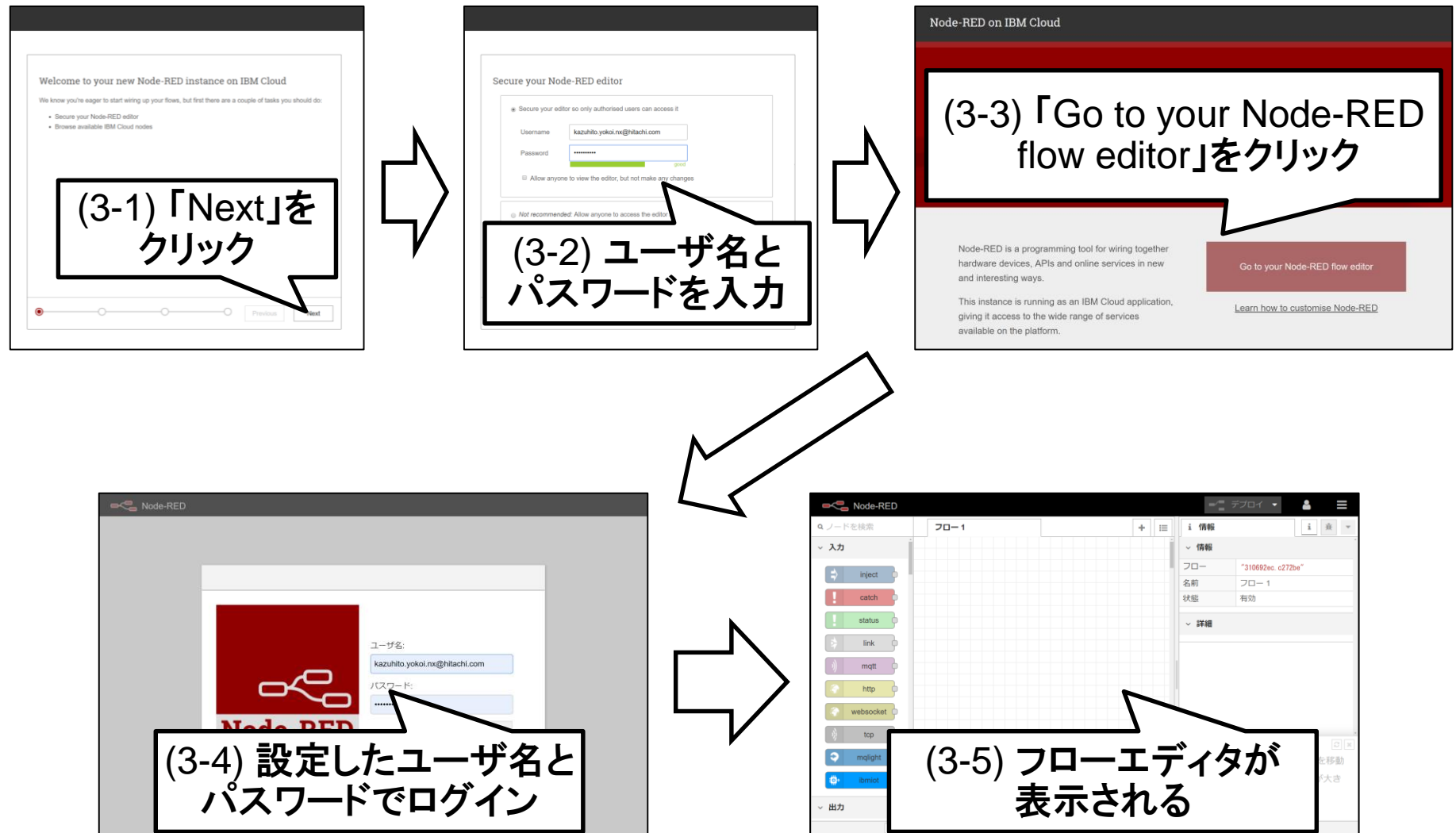
Before you can start coding with Node-RED, you must first configure the editor.

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

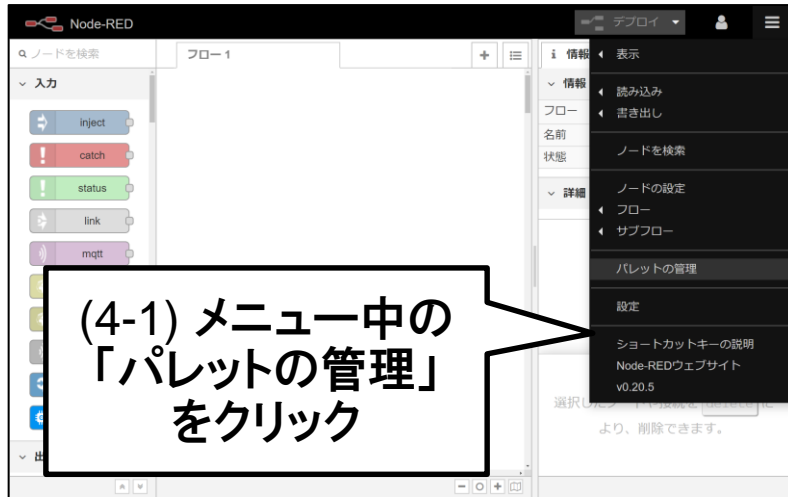
- ① After your application starts, click **Routes URL** or enter the following URL in a browser to display the Node-RED landing page.
`http://<varname><yourhost></varname>.mybluemix.net`
- ② Click **Go to your Node-RED flow editor** to open the browser-based flow editor. The editor makes it easy to connect 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



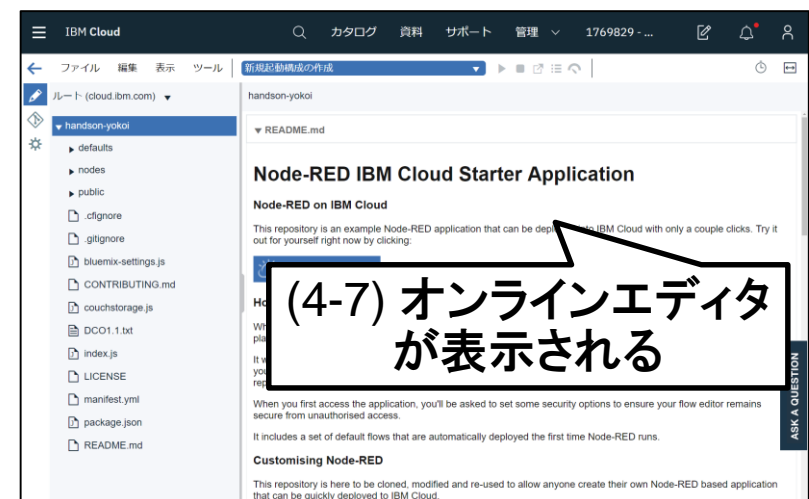
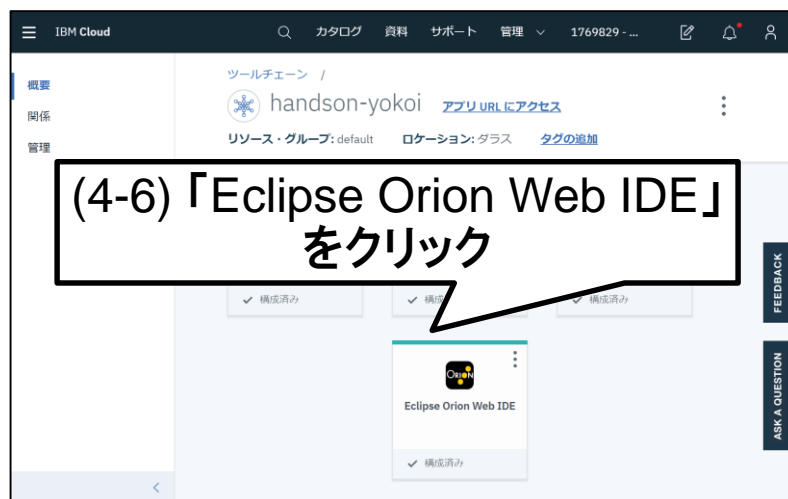
(4) 地図表示を行うworldmapノードをインストール



(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

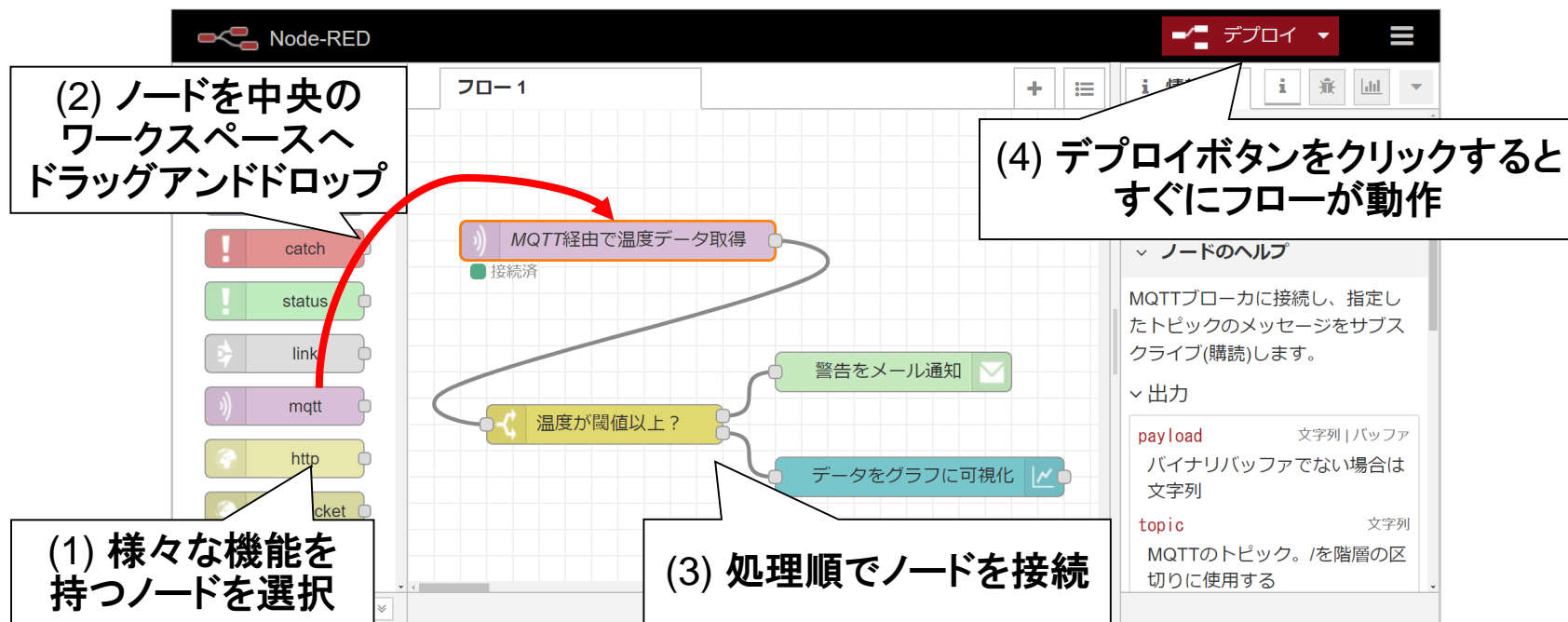
- Node generatorの説明
- ハンズオン: Open APIドキュメントからオリジナルノードを作成



Node generatorの説明

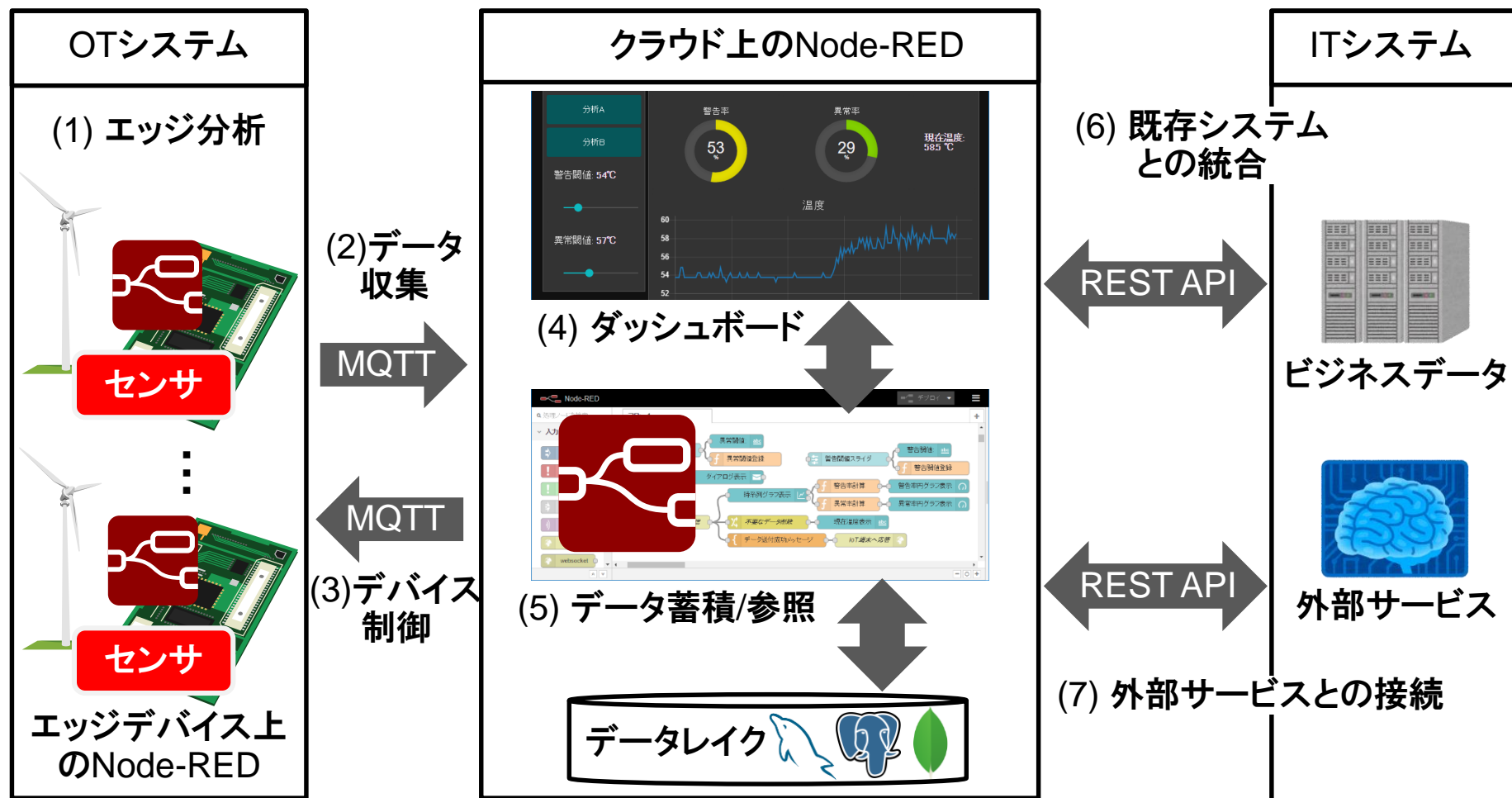
IoTアプリケーション向けのビジュアルプログラミングツール

- 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(OpenJS Foundation)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

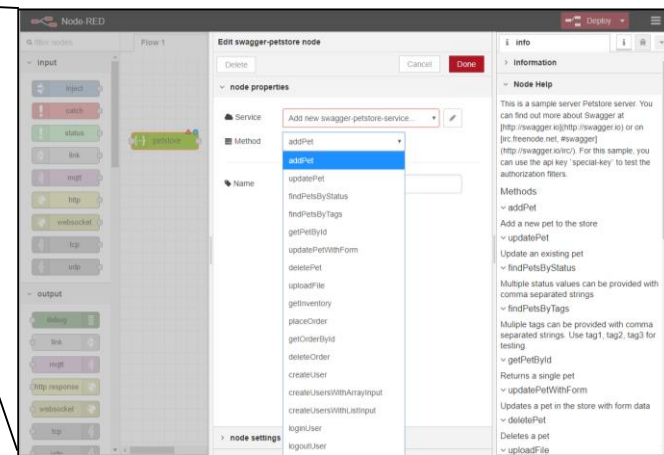
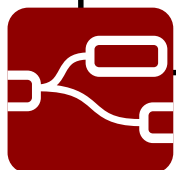
HTML
ファイル



オリジナルノード

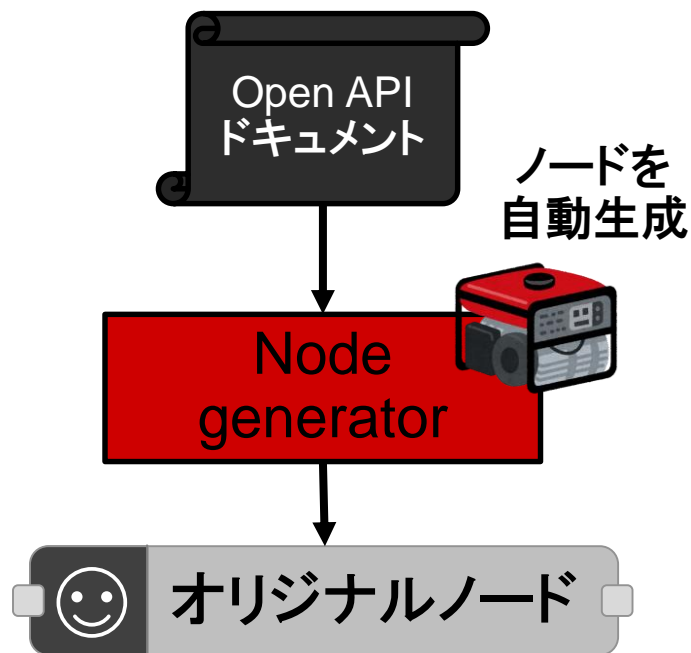
Node.js上
で実行

Node-RED
ランタイム

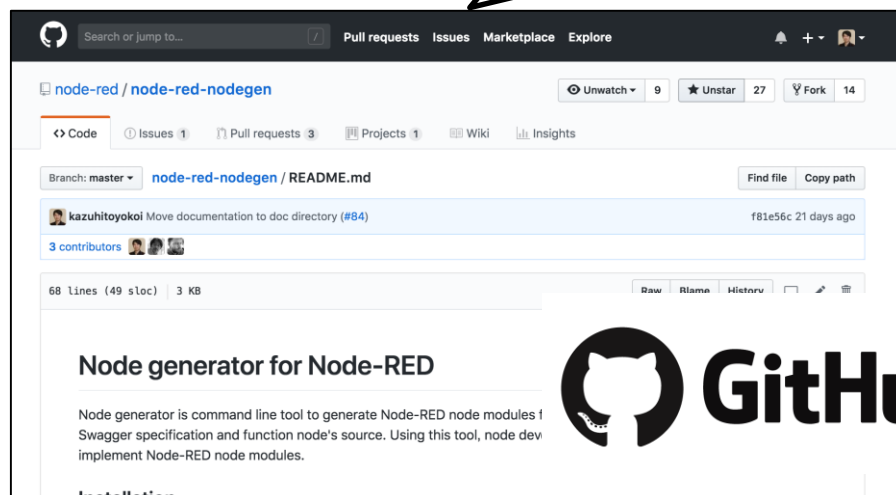


Node-REDフローエディタ
のUI

- 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 (OpenJS Foundation) のオープンソースソフトウェア

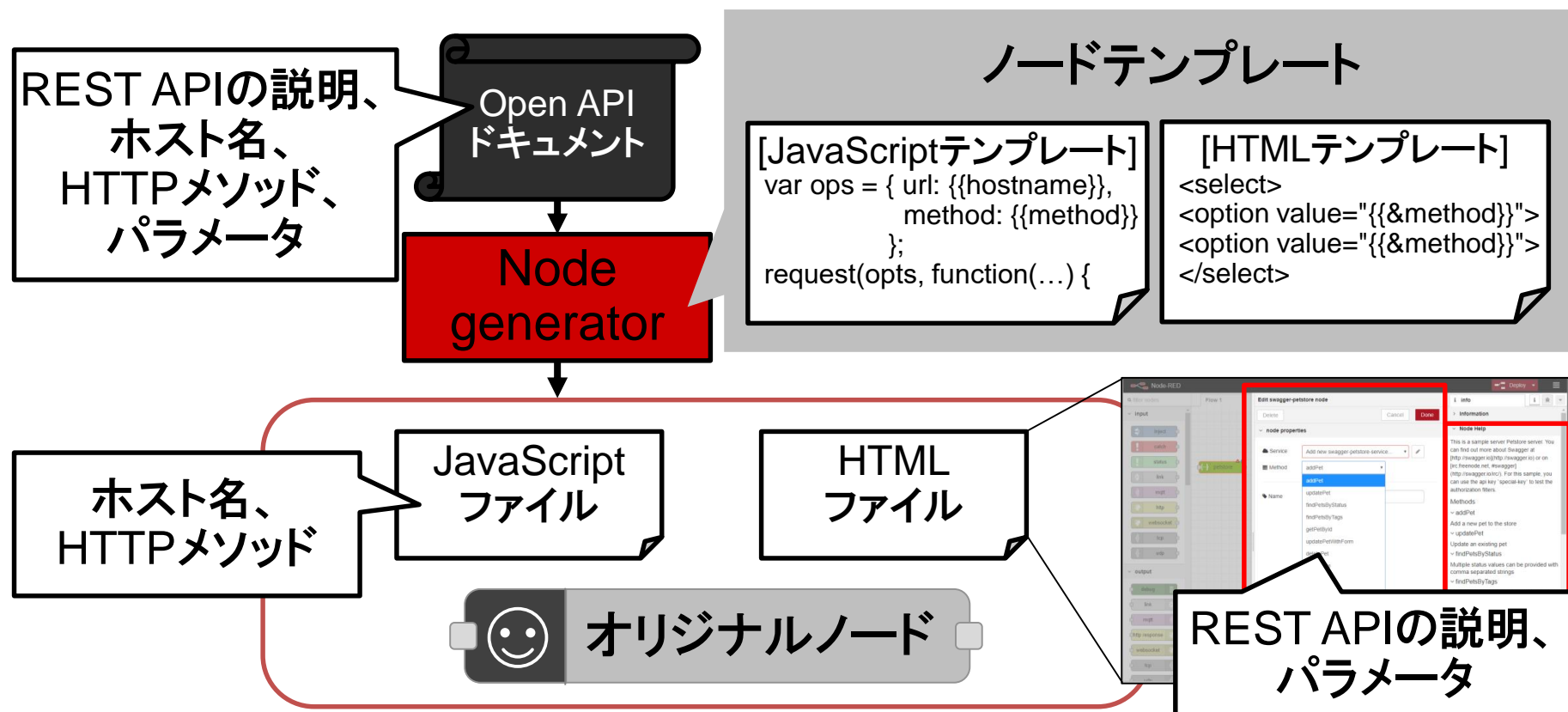


“Node-RED Node generator”
で検索すると見つかります

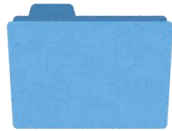


<https://github.com/node-red/node-red-nodegen>

- 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

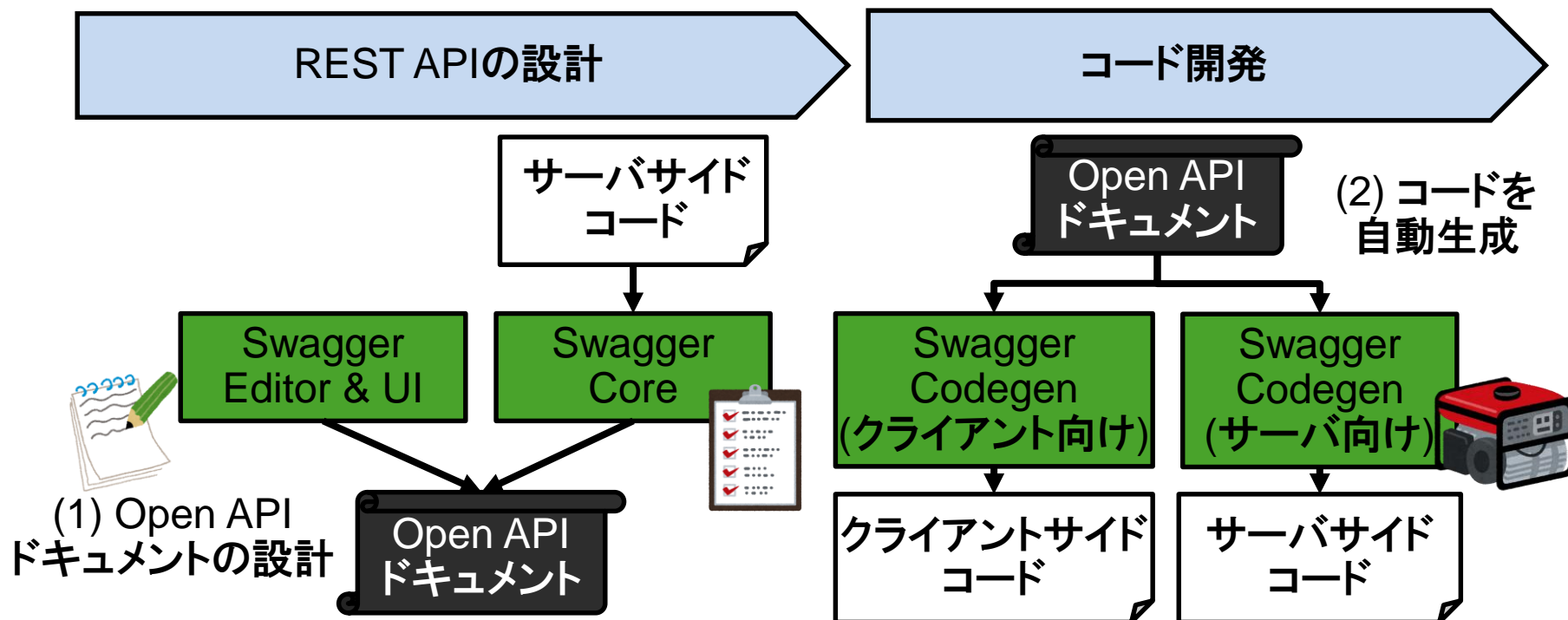
Node generatorの紹介ページ

<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

ハンズオンのリポジトリ (<https://github.com/kazuhiyokoi/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) コマンドプロンプトを開き、ホームディレクトリに移動

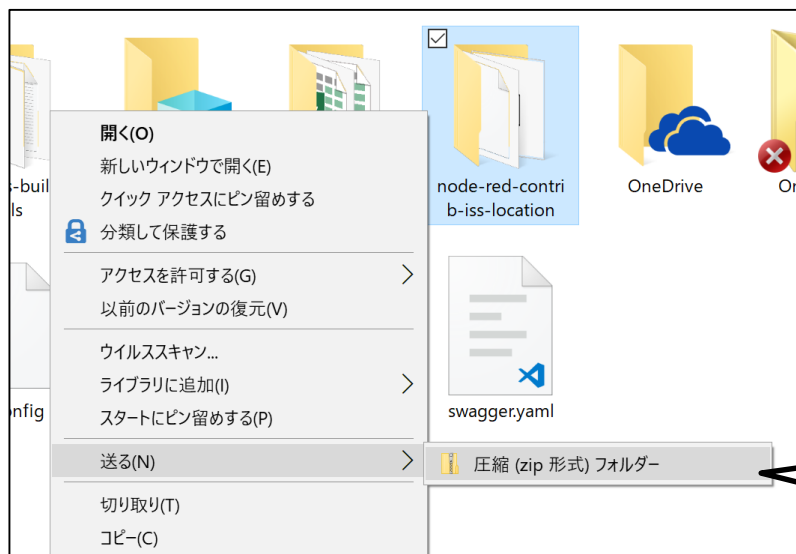
```
> cd C:\Users\<ユーザ名>
```

(2) node-red-nodegenコマンドを実行しノードを生成

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

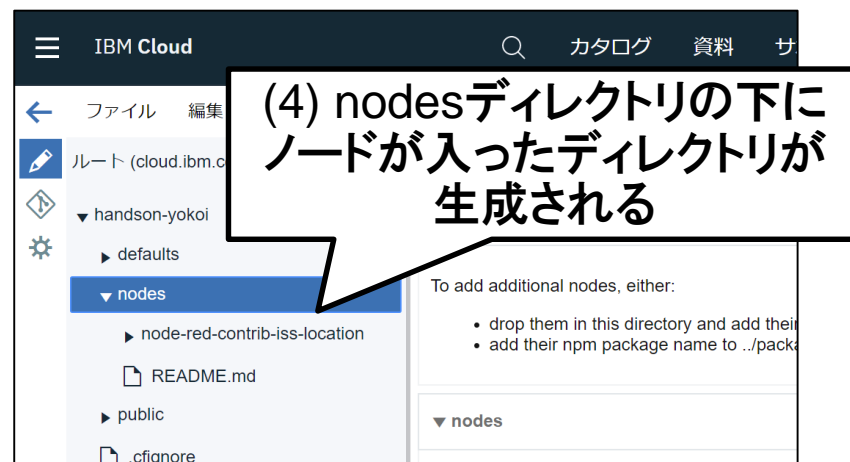
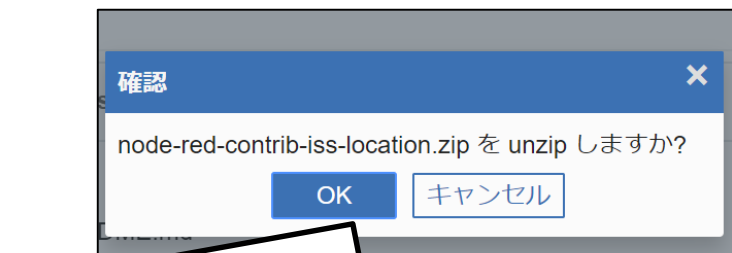
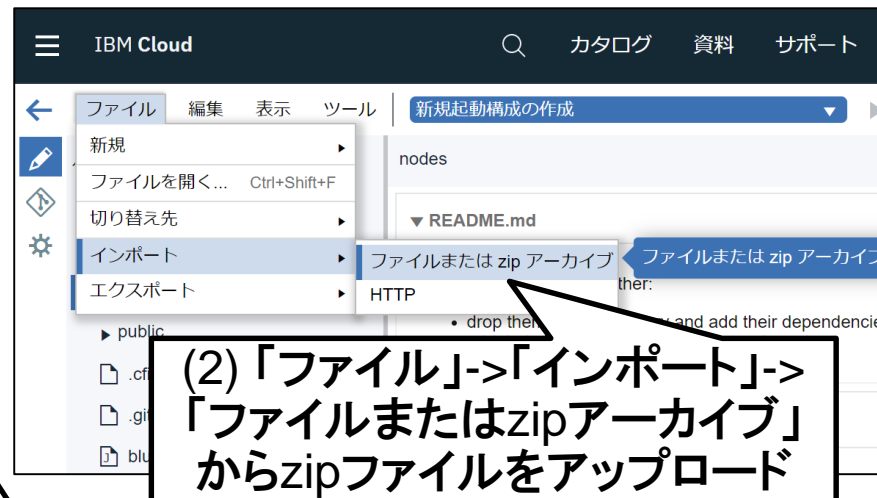
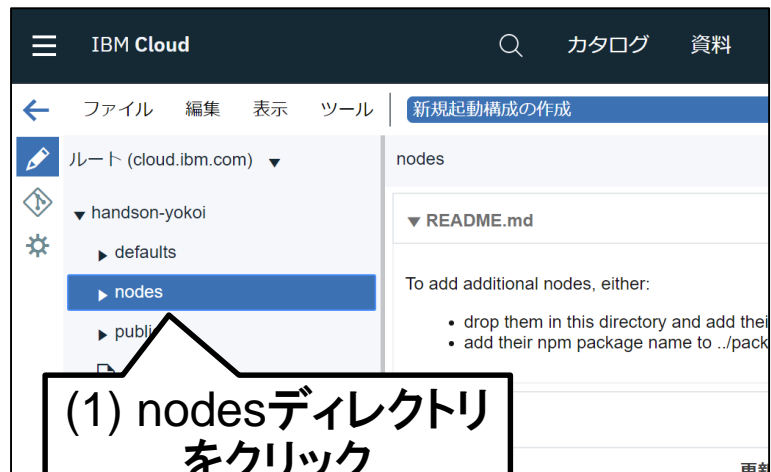
-> ノードを構成するファイル群が入ったディレクトリ
node-red-contrib-iss-locationが生成される

(3) 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

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



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

IBM Cloud

カタログ 資料 サポート 管理 1769829 - ...

新規起動構成の作成

行 13: 列 1

(4) Gitアイコンをクリック

(3) 「ファイル」->「保存」からファイルの変更を保存

(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

```
4  "dependencies": {
5    "when": "~3.x",
6    "nano": "6.2.x",
7    "bcrypt": "1.0.2",
8    "cfenv": "~1.0.0",
9    "express": "4.x",
10   "body-parser": "1.x",
11   "http-shutdown": "1.2.0",
12   "node-red": "0.x",
13   "node-red-contrib-iss-location": "file:///./nodes/node-red-contrib-iss-location",
14   "node-red-blue-mix-nodes": "1.x",
15   "node-red-node-watson": "0.x",
16   "node-red-node-openwhisk": "0.x",
17   "node-red-node-cf-cloudant": "0.x",
18   "node-red-contrib-scx-ibmiotapp": "0.x",
19   "node-red-nodes-cf-sqlldb-dashdb": "0.x"
20 }
21
22
23
24
25
26
27
```


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 changes to a repository using the IBM Cloud interface. It consists of two screenshots with numbered callouts indicating the steps:

- (1) コミットメッセージを追加**: In the left screenshot, the user is in the 'Commit' stage. The 'Commit Message' field contains '宇宙ステーションの位置情報取得ノードを追加' (Add location information acquisition node for space station). The 'Push' button is visible.
- (2) コミットをクリック**: In the left screenshot, the 'Commit' button is highlighted.
- (3) プッシュをクリック**: In the right screenshot, the 'Push' button is highlighted.
- (4) 矢印ボタンをクリック**: In the right screenshot, the 'Fetch' button is highlighted.

A large arrow points from the left screenshot to the right screenshot, indicating the flow of the process.

ソースコードのデプロイの状況を確認

IBM Cloud

ツールチェーン / handson-yokoi [アプリ URL にアクセス](#)

リソース・グループ: default ロケーション: ダラス [タグの追加](#)

Issues handson-yokoi ✓ 構成済み

Git handson-yokoi ✓ 構成済み

Delivery Pipeline handson-yokoi

BACK

(1) Delivery Pipelineをクリック

IBM Cloud

ツールチェーン / handson-yokoi / handson-yokoi

handson-yokoi | Delivery Pipeline

Build Stage [🔍](#) [⚙️](#) >

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)「実行中」が「成功」に変われば、デプロイ成功

(1) 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) 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 image shows the Node-RED interface. On the left, the 'Node-RED' header is visible. Below it, a search bar says 'q ノードを検索'. A list of nodes is shown, including 'html', 'json', 'xml', 'yaml', 'rbe', and 'iss location'. The 'iss location' node is circled in red. A red arrow points from this node to the 'iss-location' node in the workspace. The workspace is titled 'フロー 1' and contains the 'iss-location' node. To the right, the 'iss-location ノードを編集' panel is open. It has buttons for '削除', '中止', and '完了'. Below these are tabs for 'プロパティ', 'メソッド', and '名前'. The 'メソッド' tab is selected, showing a dropdown menu with 'ISSLocationNow' selected. A callout box with the text 'メソッドを選択' points to the dropdown menu.

- (3) 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
- (4) 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 Designer 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.

REST APIから取得した位置情報が表示される

```
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"
```

(5) 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

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

(6) 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

(7) 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)

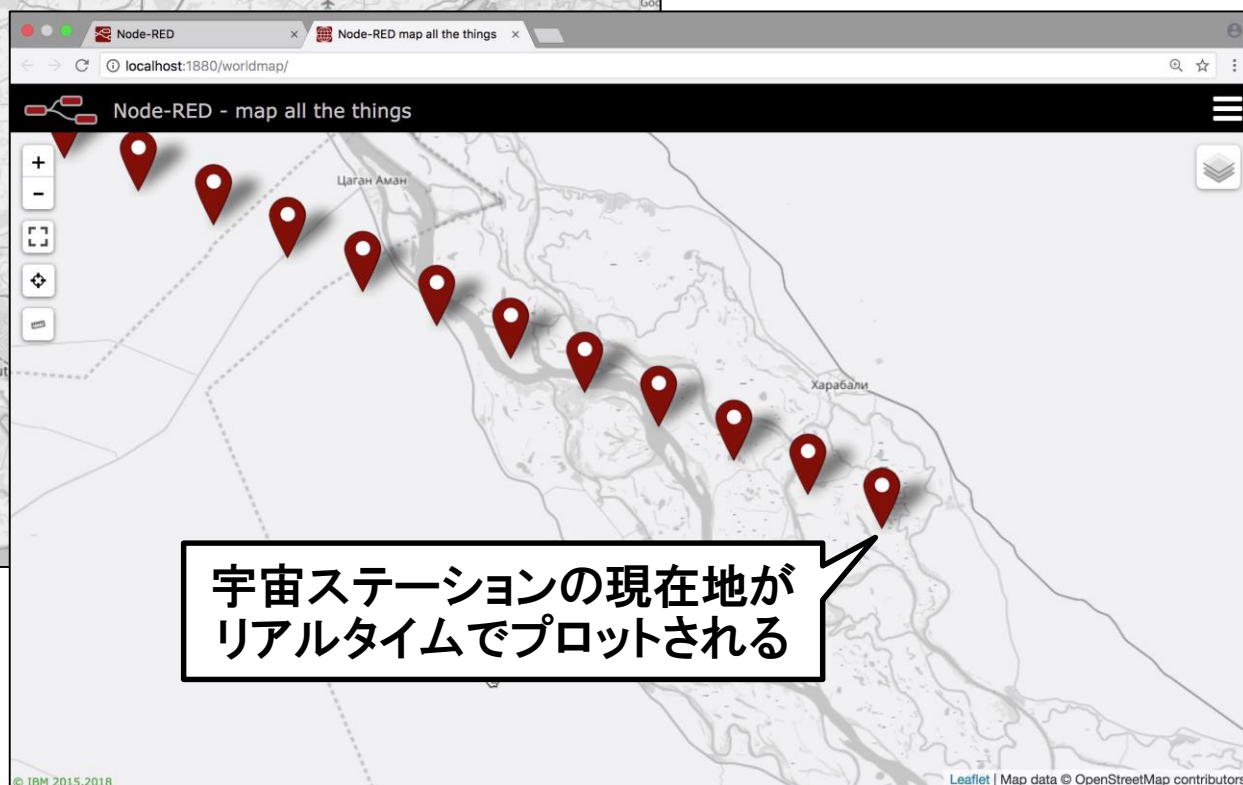
A callout points to the **set msg.payload** node, with the text: **プルダウンメニューから「msg.」を選択** (Select 'msg.' from the dropdown menu).

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 substitution:

- Rule 1: 値の代入 (Value Substitution) for `msg. payload.lat` to `msg. payload.iss_position.latitude`.
- Rule 2: 値の代入 (Value Substitution) for `msg. payload.lon` to `msg. payload.iss_position.longitude`.
- Rule 3: 値の代入 (Value Substitution) for `msg. payload.name` to `msg. payload.timestamp`.

A callout points to these three rules with the text: **3つの変数を代入** (Substitute 3 variables).

(8) <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
```



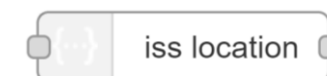
オプションの
ハイフンは2つ

- 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

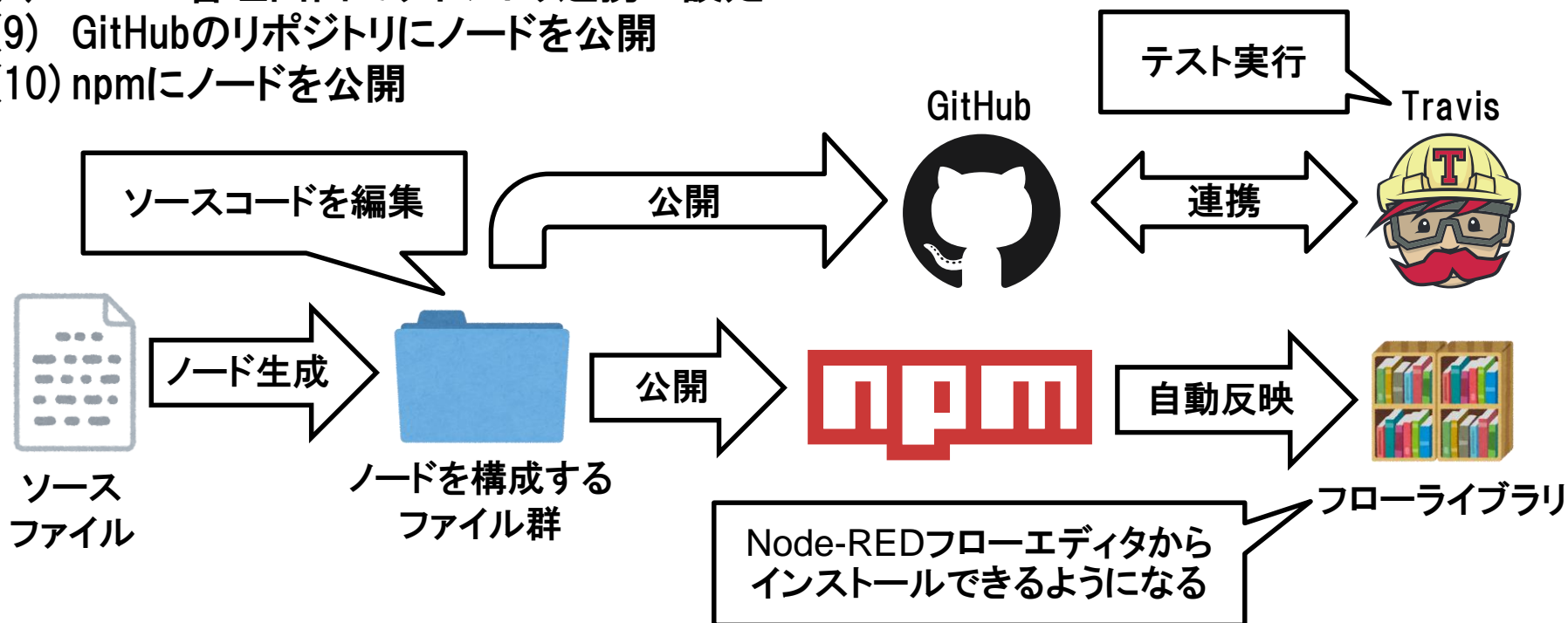
例) 地球のアイコンを設定

```
> 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。