

Plan

Transition

Données des capteurs



Mise en place de services

Collecte, traitement, visualisation, alerte...

Plan

Couches d'abstraction

Permet de réduire la difficulté afin de faciliter les interactions.

1. Réseau et Sécurité
2. Traitement des données
3. Visualisation

Recettes et outils

Node-red



Meilleurs nœuds

Exercices

MQTT

Architecture
logicielle

Grafana



Configuration

Visualisations

– Canevas

– Cours
Plugins

Plateforme de développement



VPN

SBC

Debian

CasaOS

Docker

1. Réseau et sécurité

Objectif :

Je veux accéder à mon environnement de développement facilement de façon sécurisée

Problématique Sécurité :

Politique DSI, ouverture de ports, exposition des données, certificats https

Solution :

Gestion des données en local

Mise en place d'un serveur VPN → encapsulation sur IP

2. Traitement des données

Objectif : Je veux traiter mon flux de données, ajouter des meta-données, les transformer en événement et être alerté lors de dysfonctionnement.

Problématiques : écriture et maintenance du code en python, gestion difficile des dépendances et des couplages.

Solution :

Node-red permet de programmer graphiquement et simplement

3. Visualisation

Objectif : Je veux faire ma page web de visualisation

Problématiques : Peu de connaissance en dev web (html, css, java), besoin d'un système modulaire et évolutif.

Solution :

Node-red permet de gérer le Front/Back d'une page Web

L'interface de Grafana permet d'afficher et de récupérer les données

Réseau et sécurité

1. Réseau et sécurité

Objectif :

Je veux accéder à mon environnement de développement facilement de façon sécurisée.

Problématique Sécurité :

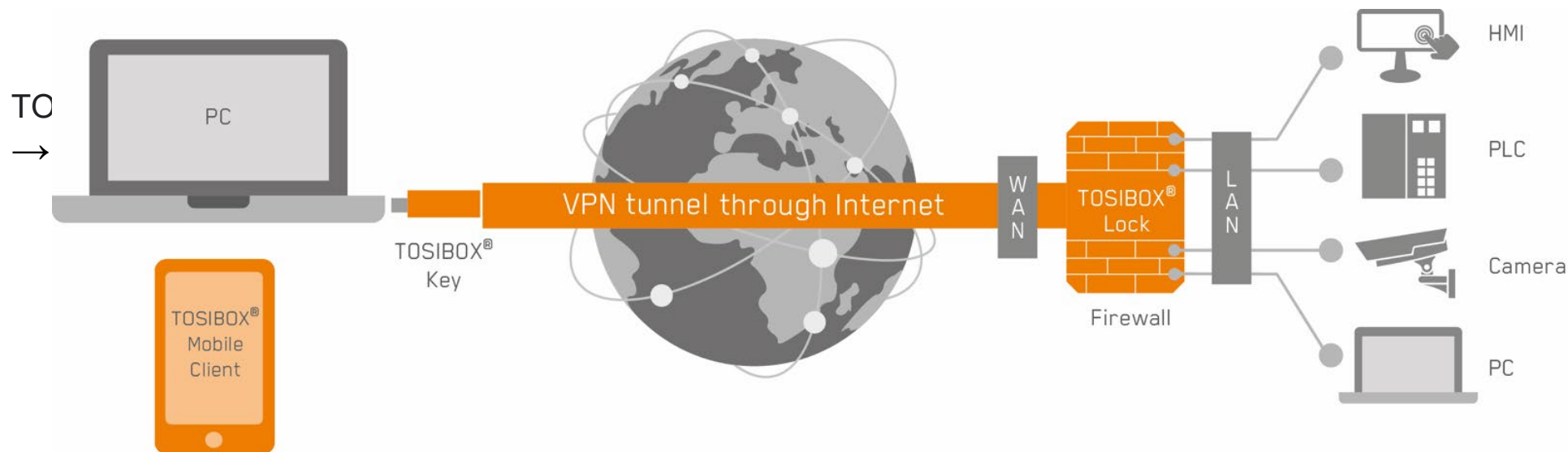
Politique DSI, ouverture de ports, exposition des données, certificats https...

Solution :

Gestion des données en local

Mise en place d'un serveur VPN → encapsulation sur IP

Solution Commerciale VPN TOSIBOX



ISO 27001-
certified

Modèle Tosibox 175



Accès internet :

- Port WAN
- Wifi (client)
- 4g

Accès aux ressources :

- Local : Port WAN + Wifi
- Clé USB
- Clé Soft



- # 400 €

1



Fonctionnement de la clé
→ Test avec un utilisateur

2

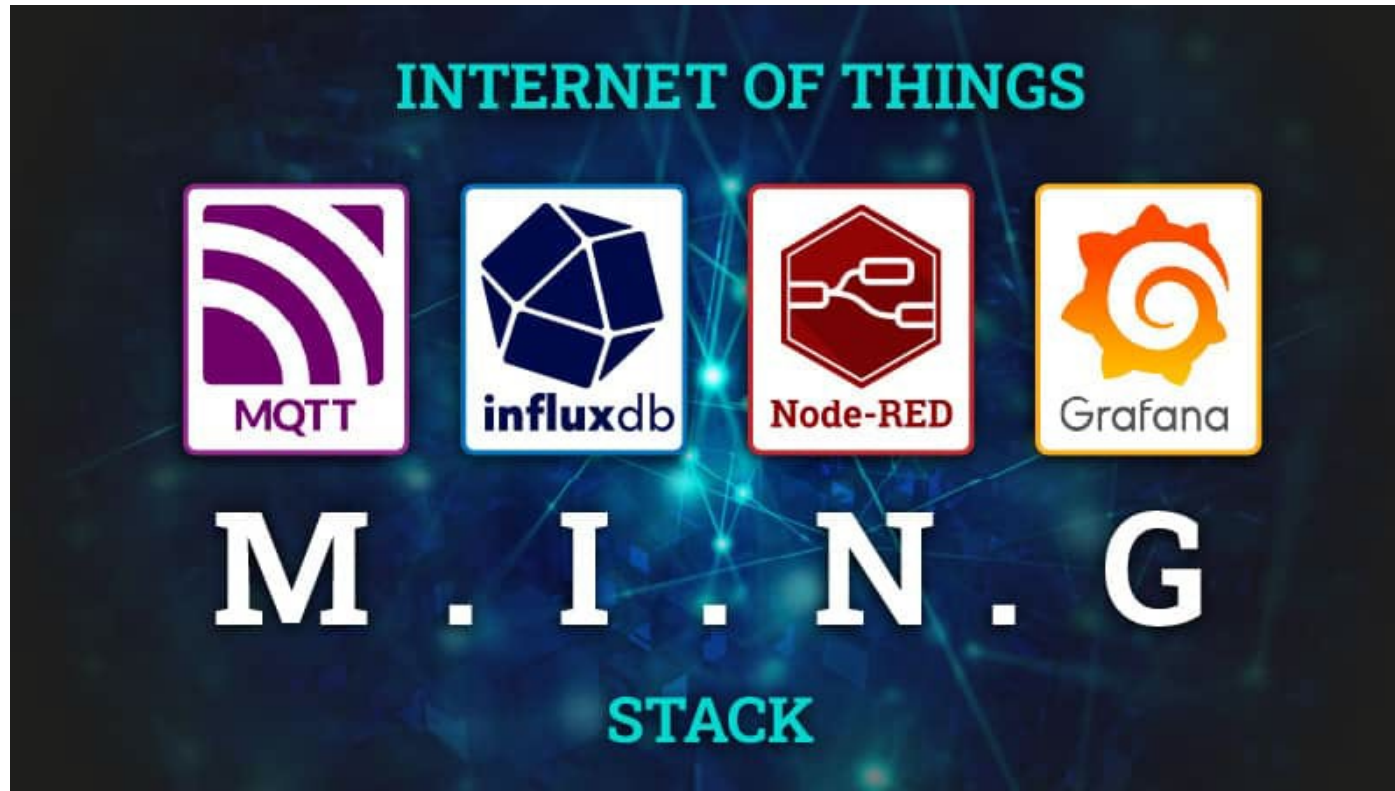


Interface Tosi control

- Connexion aux routeurs
- Accès aux ressources
- Gestion des droits

Services

IOT





Monitoring (s)

Objectif : Alerte, maintenance, aide à la décision, Proto -> Prod

Watchdog(s)

API opérateur

Flux data

Edge computing

Node-red

Test base de donnée

API Influxdb

Service Grafana

Sensors Data



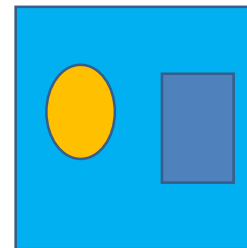
Passerelle



Stockage DB

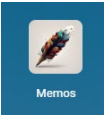


Visualisation





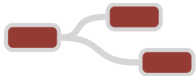
Documentation (s)



Memos : Documentation (Notes rapide)



Wikis : Documentation (Site Web)



Node-red : Project, Flow, Nodes

Format Commun



Markdown (texte structuré)



Mermaid (diagramme)

Service pour les groupes



Bookmarks

groupe_02 → 15

<http://10.10.10.110:20402/>

Node-red

<http://10.10.10.110:20002/>

Grafana

<http://10.10.10.110:20102/>

Service pour les groupes

Influxdb

<http://10.10.10.110:20202/>

groupe_02 → 15

Pass / Login : admin/admin

database : anf

Version : 1.8.10

Service pour les groupes

NextCloud

http://10.10.10.110:10081

groupe_02 → 15

Login NextCloud

login : anf-groupe_xx

pass : anf-groupe_xx

SBC + OS + Logiciels

Single Board Computer



Objectifs

- Ressources en **local**
- Tous les outils sous la main



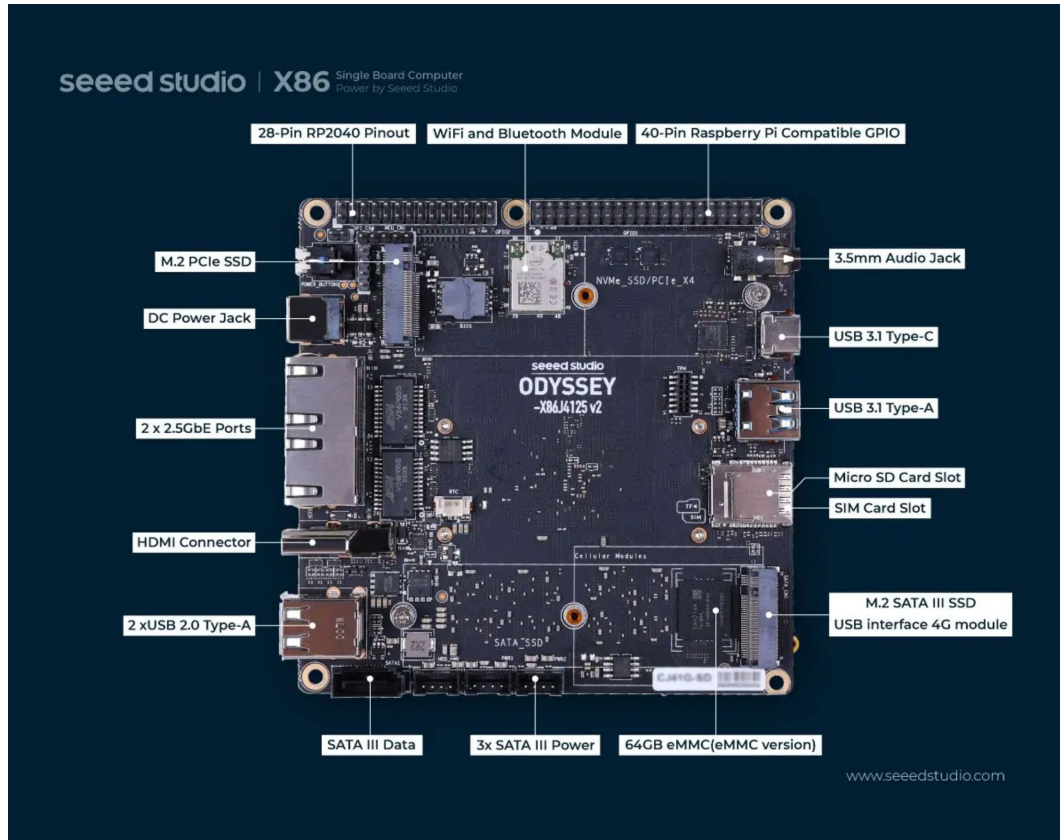
SBC ?

Mini ordinateur incluant :

- Processeur
- Ram vive : 4 Go
- « Disque » : EMMC ou SSD
- I/O

Requis OS 64 bit

Matériel : Recomputer Odyssey



M.2 PCIe :
SSD



M.2
LoRaWAN



M.2
4G

1) Debian 12 (via clé une USB)

2)



Script : `curl -fsSL
https://get.casaos.io | sudo
bash`
<https://www.casaos.io/>



Parcourir le site
Web de CasaOS



<https://www.casaos.io/>

Parcourir le site Web de CasaOS



Configuration réseau

Routeur Tosibox

Objectifs

1) Accès internet

Port WAN (Ethernet) : Internet du Centre Jean-Bosco

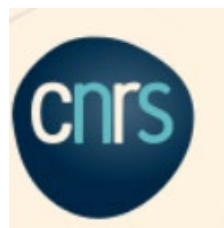
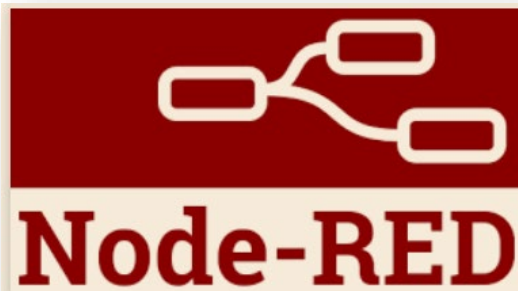
2) Ressources locales

- Wifi point d'accès
- Port LAN

Wifi du routeur

Nom (SSID) : ANF-IOT

Password : ANF-IOT-2023



Sondage :

Qui connaît Node-red ?

Qui a déjà utilisé Node-red ?

Qui utilise en production Node-red ?



Emmanuel Landrивon



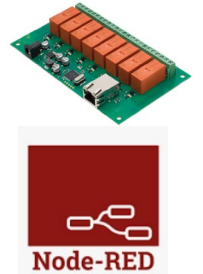
My env. → Engineer // Research team // Chemical process

My job → Build test bench : 50 I/O, 1S/s.



Strategy : decoupled jobs

- Remote control → ~~LabVIEW~~ + **Node-red**
- Security → autonomous Web relay card
- Monitoring and alerting → **Node-red** + Gateway



Benefices : Set it and forget it

- Quick to deploy and easy to maintain
- Autonomous end-user oriented

Node-RED

Low-code programming for event-driven applications

Latest version: v3.0.2 (npm)

<https://nodered.org/>



Emmanuel Landrison

Aujourd'hui vous aller découvrir

- Environnement Node-red
- Installation de node-red
- Bonnes pratiques
- Les recettes

Why Node-red ?

- **Graphical programming :**

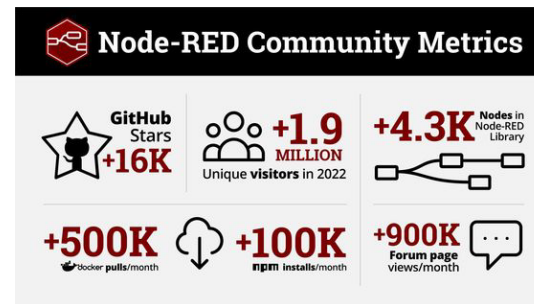
- Quick to deploy (no compilation)
- Increase code maintainability
- Easy to debug
- Local documentation

- **Open source :**

- Community and **easy sharing**
- Free

- **Multi-usage / stack:**

- Com with devices (modbus, lora..)
- IOT
- Process control
- Home Automation
- Web pages...



Browser-based flow editing, on a server

data = {Json object}

Deploy button (to push
before executing a flow)

Menu button

<http://localhost:1880>

Palette
of node
types

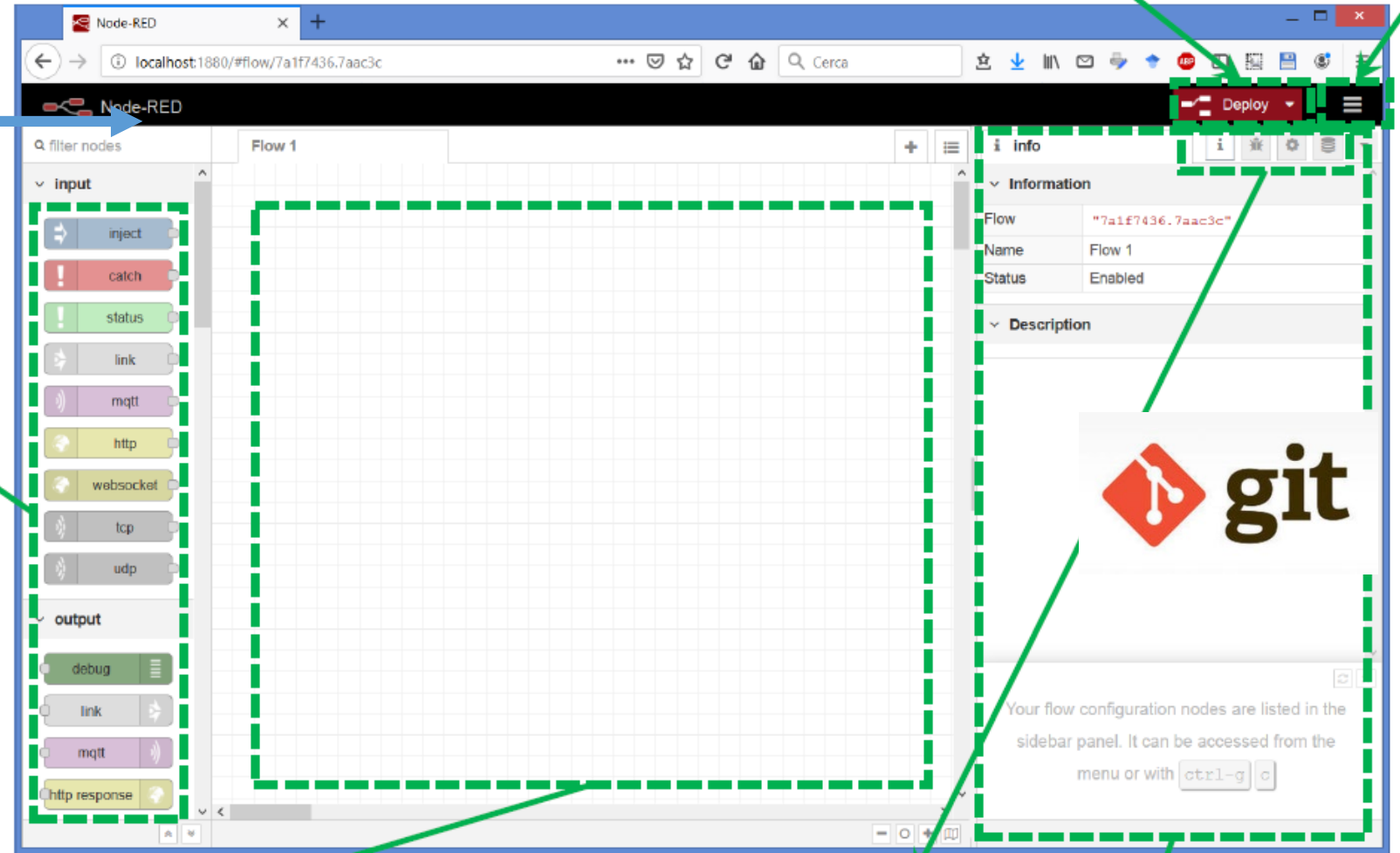
Layout to draw flows

Panel showing contents:

- Info of selected node
- Debug
- Configuration Nodes



Designed and built by IBM



Runs everywhere with everything (almost)

Inbuild
product



ADAM-6717
8AI/5DI/4DO Intelligent I/O Gate

New
I/O



All Images

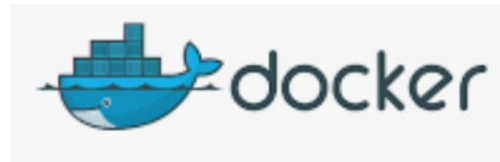
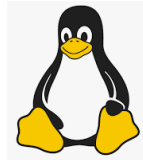
4G router



Industrial PLC



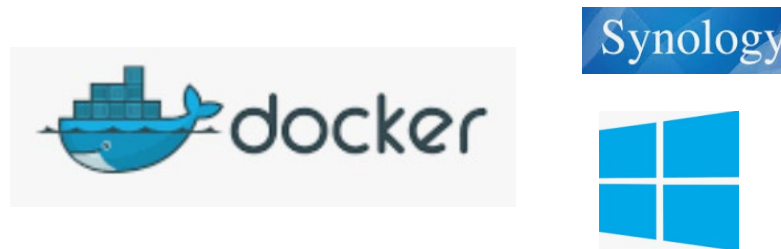
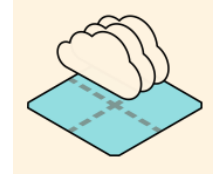
Local environnement



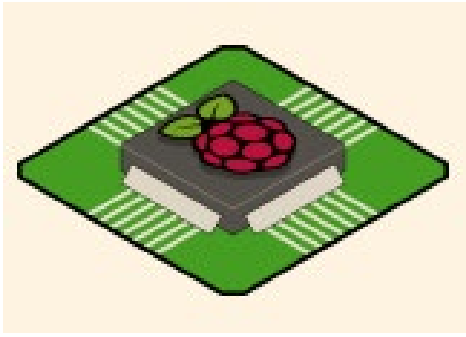
Synology



Cloud



FlowFuse

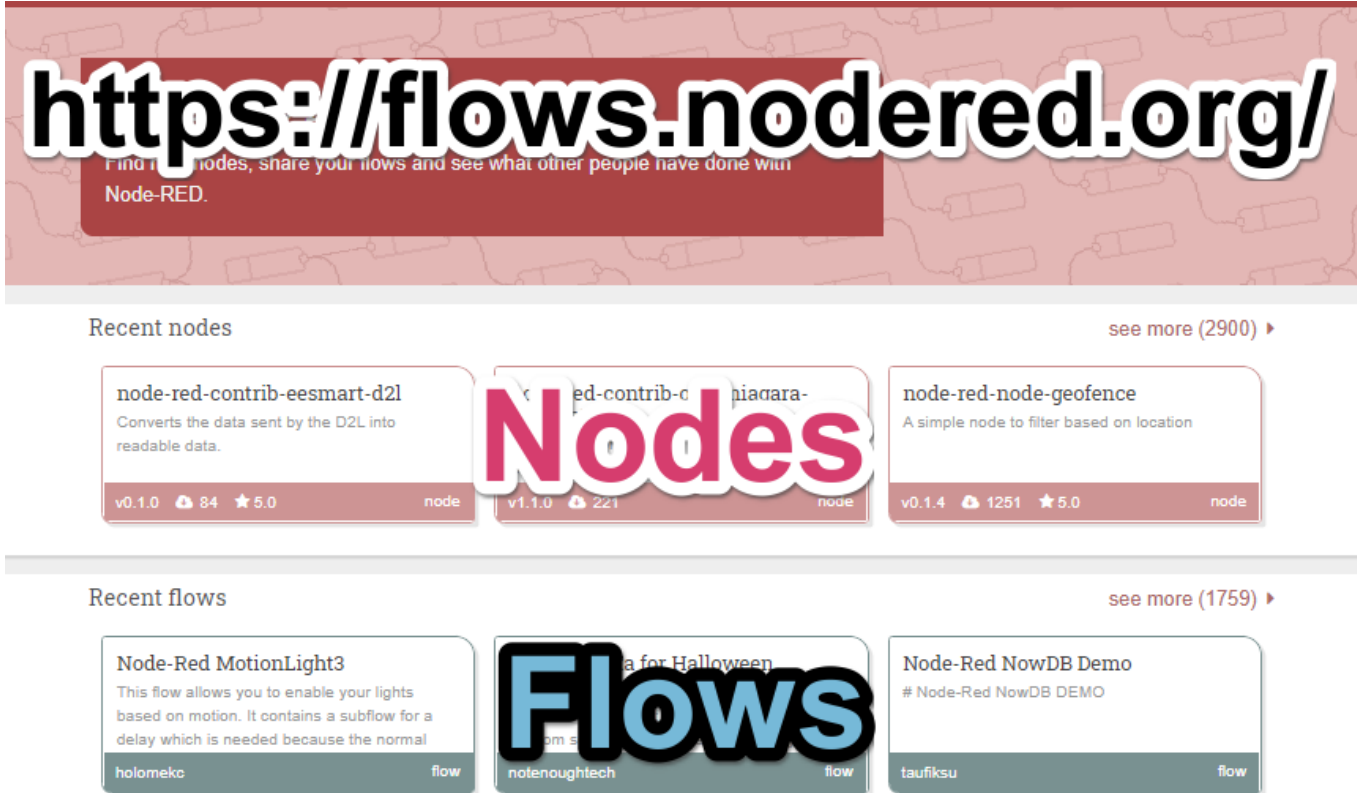


Installation sur Linux

- 1) Run the Bash Script : `bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)`
- 2) Run as a service : `sudo systemctl enable nodered.service`
- 3) Access with browser : `http://<hostname>:1880`

Social Development

→ Easy sharing, flow community, clone Git repo...



The screenshot shows the Node-RED website interface. At the top, a large red banner displays the URL **https://flows.nodered.org/** in white text. Below the banner, the page is divided into two main sections: "Recent nodes" and "Recent flows".

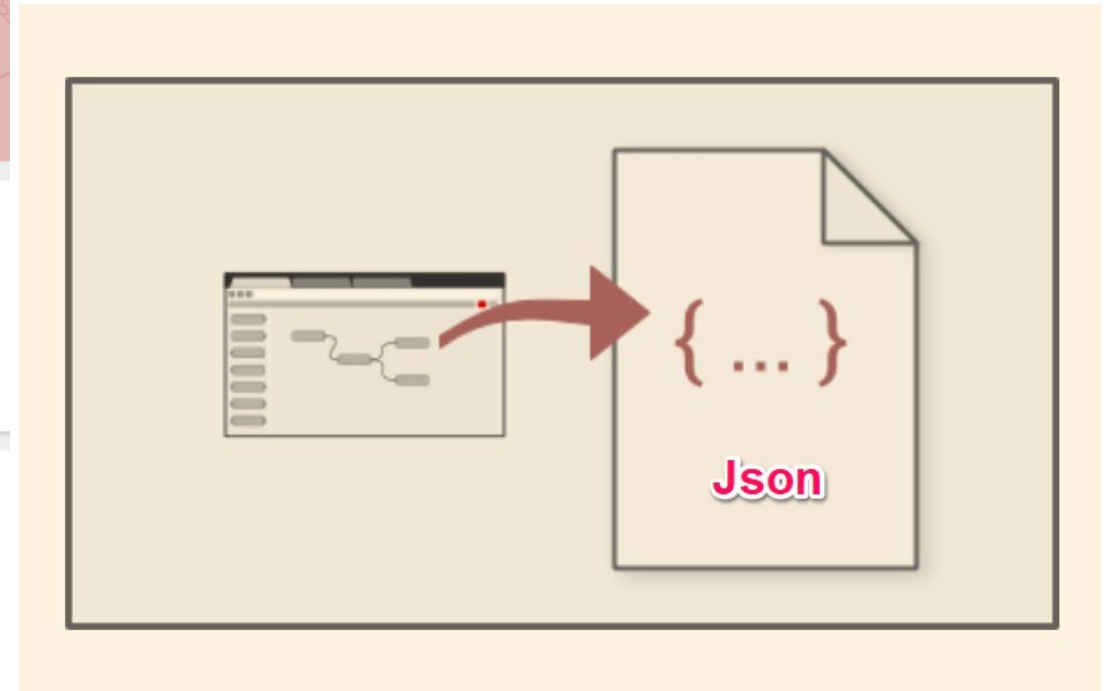
Recent nodes section:

- node-red-contrib-eesmart-d2l**: Converts the data sent by the D2L into readable data. Version v0.1.0, 84 likes, 5.0 stars.
- node-red-contrib-c**: A simple node to filter based on location. Version v1.1.0, 221 likes, 5.0 stars.
- node-red-node-geofence**: A simple node to filter based on location. Version v0.1.4, 1251 likes, 5.0 stars.

Recent flows section:

- Node-Red MotionLight3**: This flow allows you to enable your lights based on motion. It contains a subflow for a delay which is needed because the normal... Author: holomeko.
- Node-Red NowDB Demo**: # Node-Red NowDB DEMO. Author: taufiksu.

Large stylized text "Nodes" and "Flows" are overlaid on the respective sections.

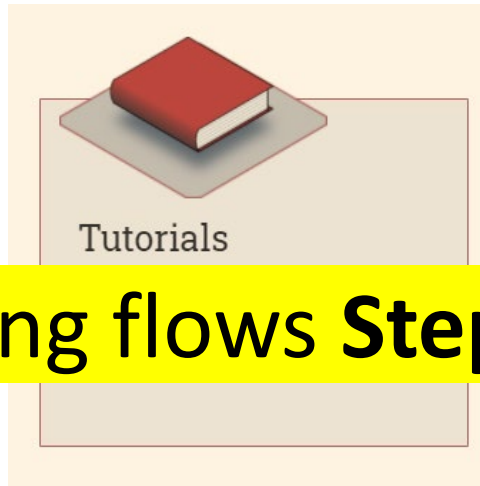


Check code compliance (coder reputation, git statistics, user comments) of the nodes.

Gérer sa courbe d'apprentissage

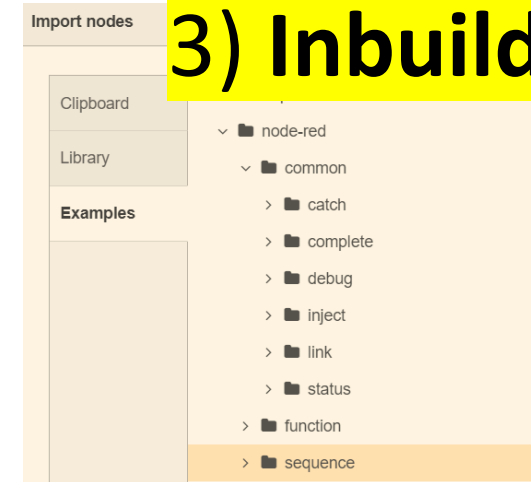


1) Creating flows Step by step



Tutorials

3) Inbuild Examples



2) Cookbook watch,copy and paste



Cookbook

Recipes to help you get things
done with Node-RED

4)



Why web server interface ?



Clients needs :

Watch test bench indicators
Retreive the data
Alerts



: DevOps for Node-RED

Intégration dans les produits industriels

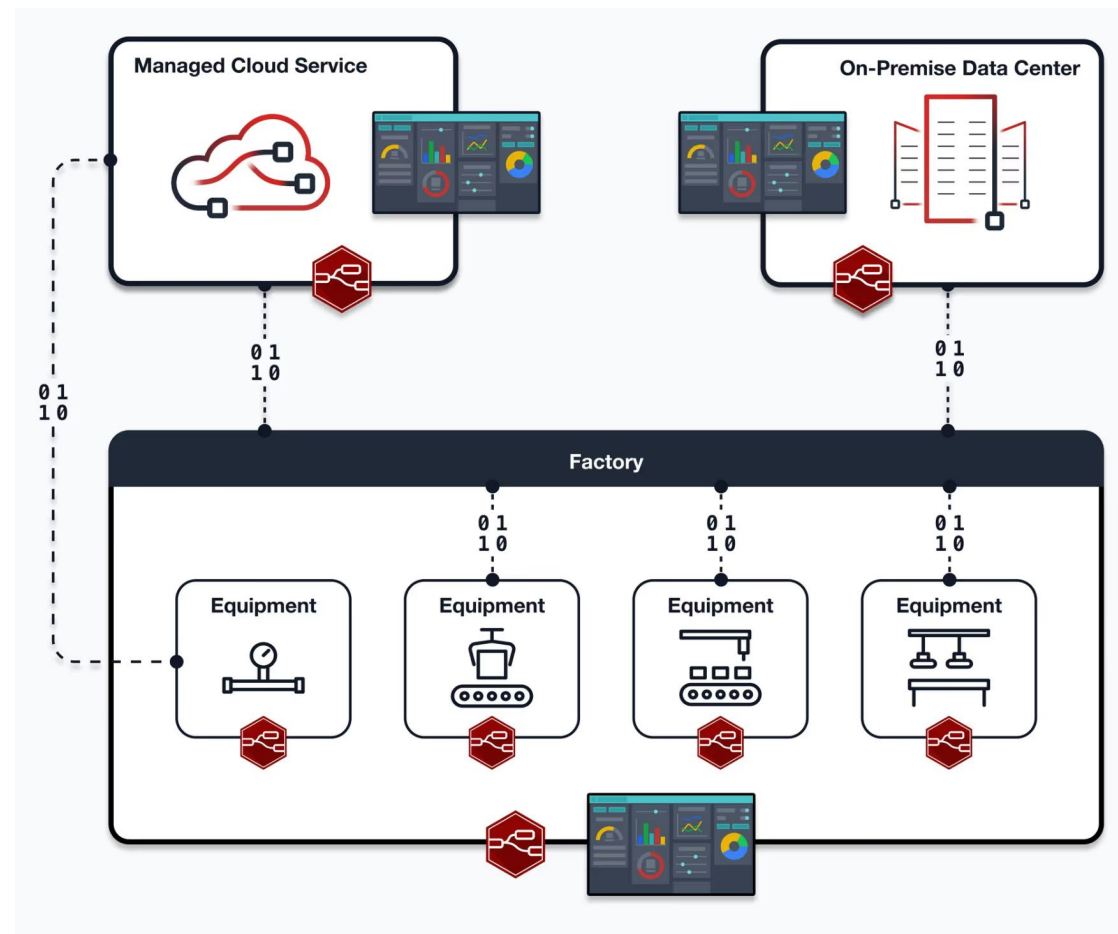
1. [Advantech](#) Node-RED Field Gateway
2. [Bechhoff](#) TwinCAT
3. [Bivocom](#) TG452 IoT Edge Gateway
4. [BLIOT Edge Computing Gateway](#) EdgeCom BL302
5. [Bosch CtrlX Core](#)
6. [Broadsens](#) GU200 & GU 200S
7. [Emerson](#) PACEdge
8. [Hilscher Automation](#)
9. [Opto22](#) groov RIO & EPIC
10. [Parallax AV](#) Control System
11. [Particle.io](#) Particle
12. [Pepperl+Fuchs](#) AS-Interface gateway
13. [Raspberry Pi](#)
14. [Renesas](#) FT Click
15. [RevPi](#) Connect
16. [Schneider Electric](#) ExoStructure Plant Data Expert
17. [Siemens](#) S7 PLC
18. [ST-One](#)
19. [Tulip](#) Edge MC & Edge IO
20. [Wago](#) Edge Controller & Computer
21. [Weidmueller](#) control web

Version locale gratuite
Community édition



Déploiement

Node-red en **local**
sur les équipements
Via un Device Agent




Questions ?

- Pourquoi apprendre Node-red ?

Le format

{JSON}

Définition

-  JavaScript Object Notation
- JSON est un format de données textuelles dérivé de la notation des objets du langage JavaScript.
- Il permet de représenter une **information structurée**

Les bases de JSON

- Structure de base : paire **clef-valeur** (key-value)
- "titre": "hello ANF iot"

<https://jsoneditoronline.org/>

The image displays the JSON Editor Online interface, showing two documents side-by-side. The left document, titled "New document 2", contains the following JSON structure:

```
{
  "array": [
    1,
    2,
    3
  ],
  "boolean": true,
  "color": "gold",
  "null": null,
  "number": 123,
  "object": {
    "a": "b",
    "c": "d"
  },
  "string": "Hello World"
}
```

The right document, titled "New document 1", shows the same JSON structure in a collapsed, interactive view. The "array" is expanded, showing 3 items. The "boolean" is checked, "color" is set to "gold", "number" is 123, "object" has 2 props, and "string" is "Hello World".

Between the two documents is a central panel with the following options:

- Copy
- Transform
- Differences
- ☐ Enable

Editeur Json intégré à



Edit inject node

Delete Cancel Done

Properties

Name

msg.payload = { "array": [1, 2, 3], }

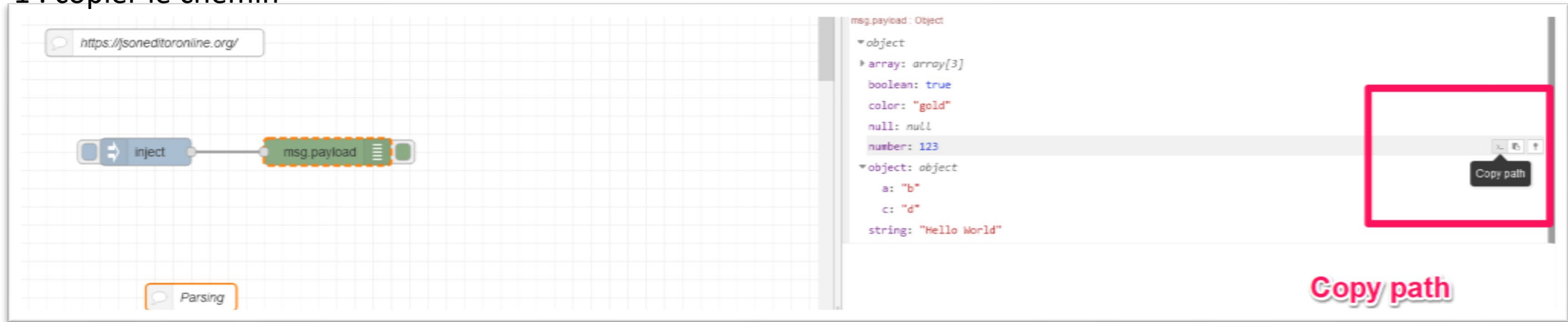
Edit JSON Visual editor

```

{
  "array": [
    1,
    2,
    3
  ],
  "boolean": true,
  "color": "gold",
  "null": null,
  "number": 123,
  "object": {
    "a": "b",
    "c": "d"
  },
  "string": "Hello World"
}
```

Parsing Json : récupérer une valeur

1 : copier le chemin

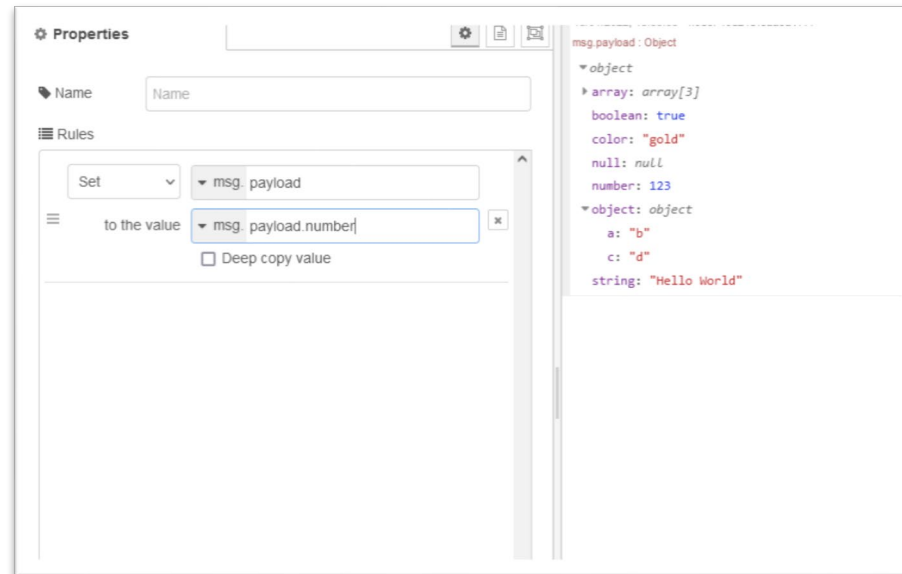
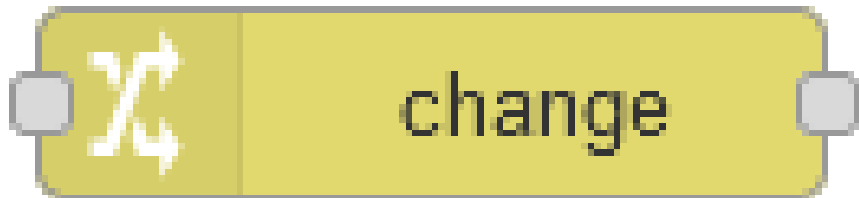


The screenshot shows a workflow editor interface. On the left, there is a grid with an 'inject' node and a 'msg.payload' node. The 'msg.payload' node is highlighted with a red dashed box. Below the grid, there is a 'Parsing' button. On the right, there is a JSON viewer showing the structure of the payload. The JSON is as follows:

```
msg.payload: Object
  object
    array: array[3]
    boolean: true
    color: "gold"
    null: null
    number: 123
    object: object
      a: "b"
      c: "d"
      string: "Hello World"
```

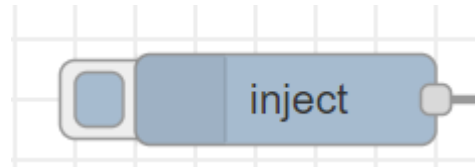
A red box highlights the 'Copy path' button in the right-hand pane. The text 'Copy path' is written in red below the box.

2: Coller dans un change node / Fonction set

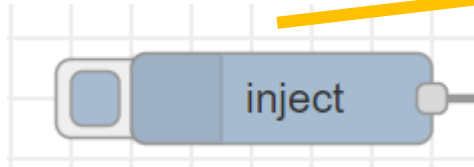
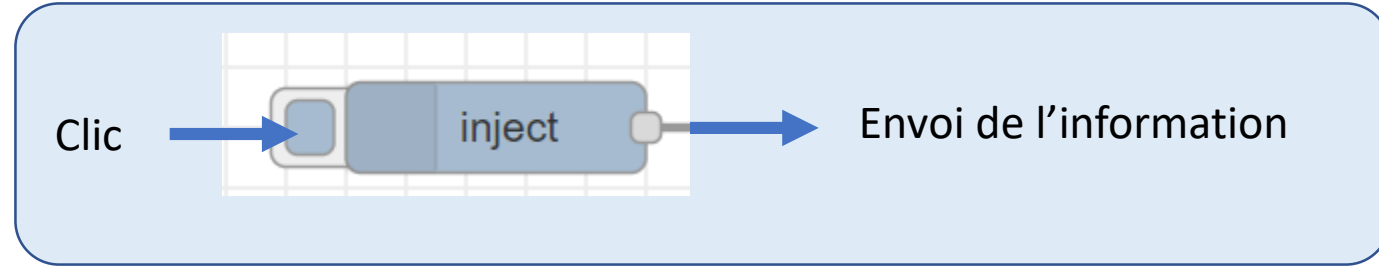
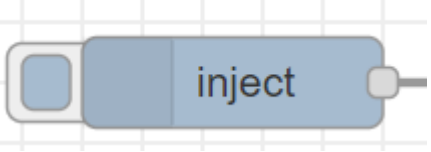


The screenshot shows the 'Properties' panel of a workflow editor. The 'Rules' section is expanded, showing a 'Set' rule. The 'to the value' field is set to 'msg.payload.number'. The 'Deep copy value' checkbox is unchecked. On the right, there is a JSON viewer showing the structure of the payload, which is the same as in the first screenshot.

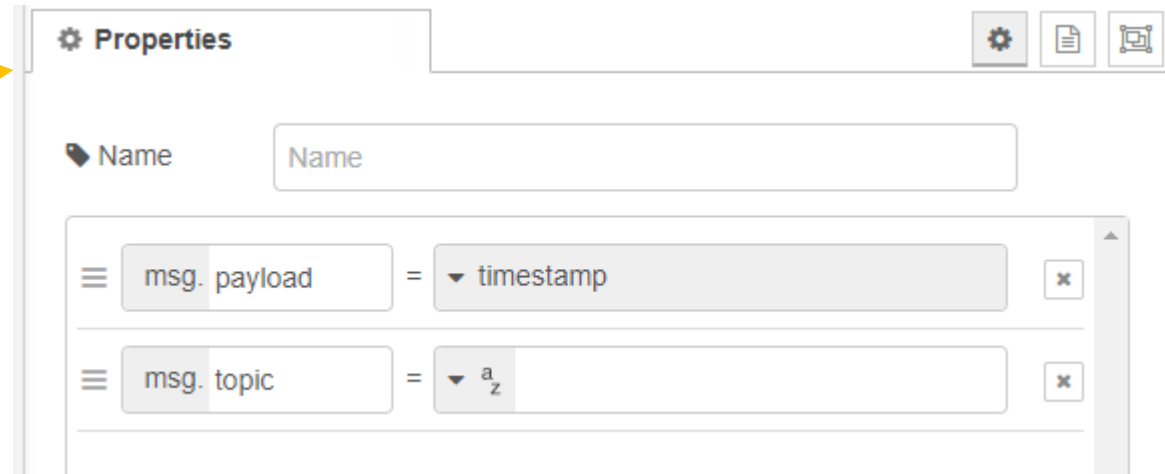
Basics Nodes



Inject node

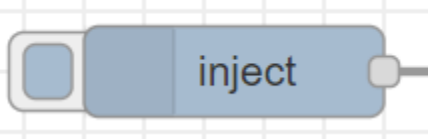


Double Clic

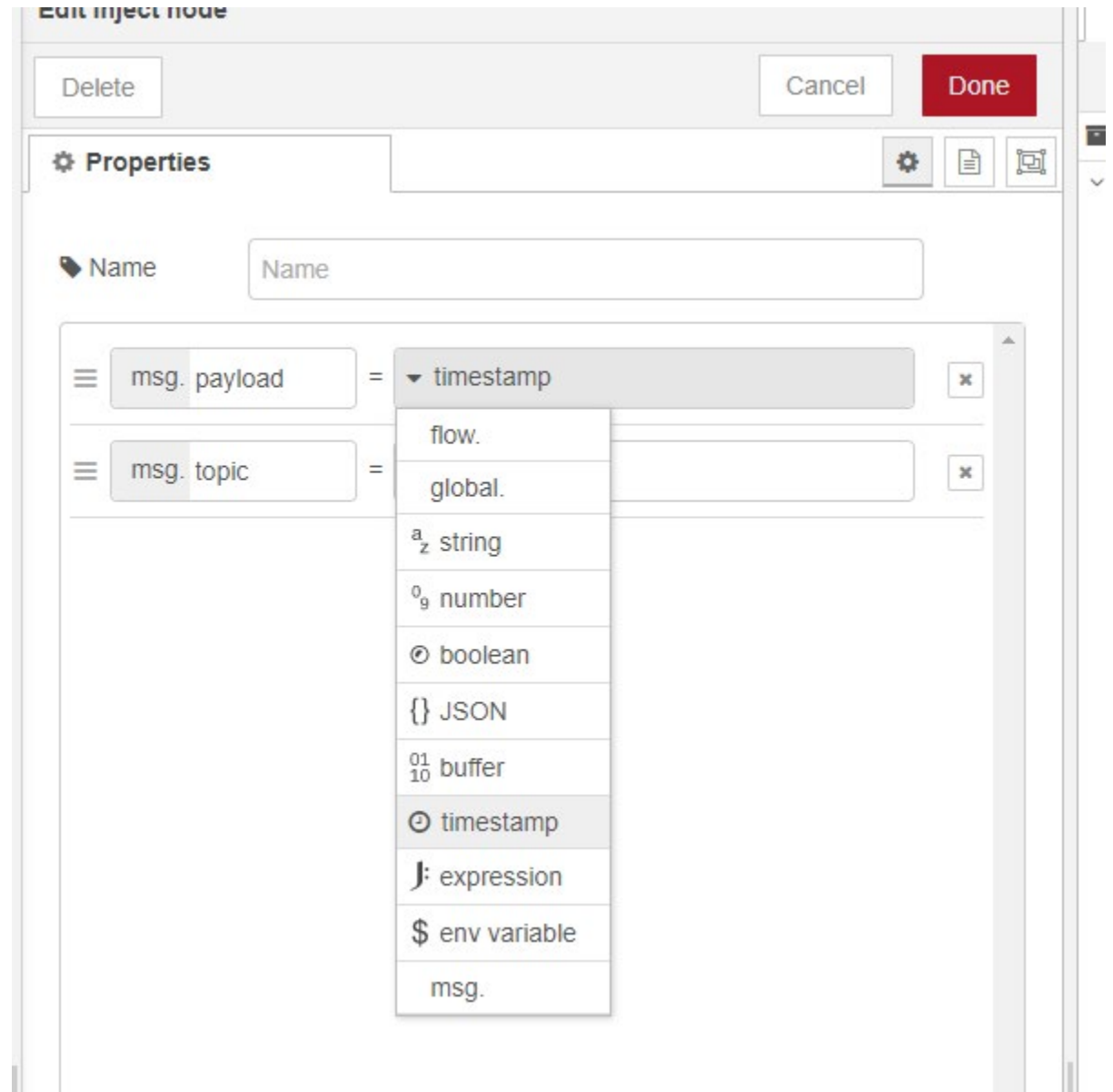


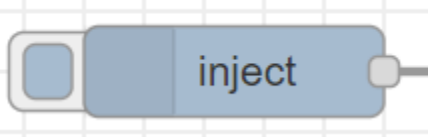
Payload = message principal

Topic = sujet utilisé par les nodes en aval



Type de données
string, number...
Variables : flow / global





Inject at start : permet de définir les variables par exemple

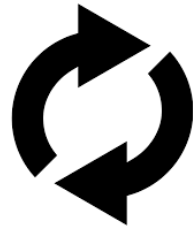
 **Inject démarrage** 


☐ Inject once after seconds, then



 Repeat

☐ Enabled

Repeat = boucle

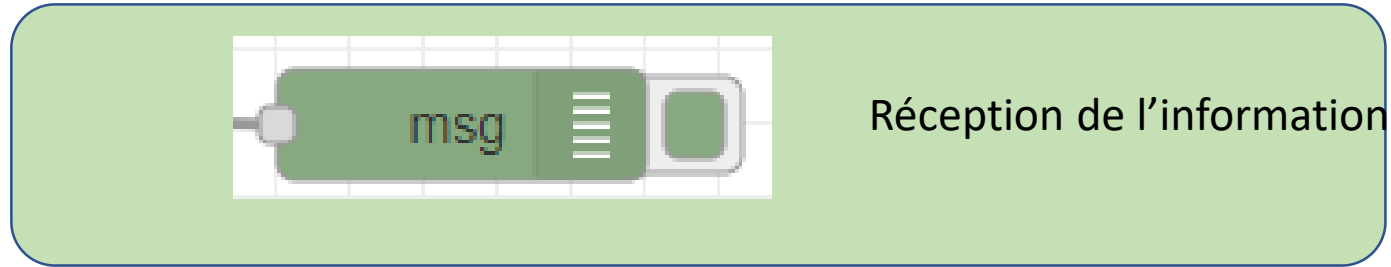
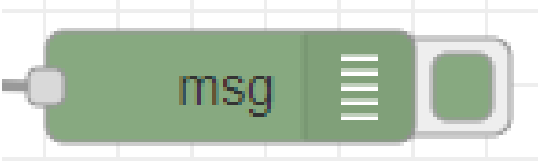


 Repeat

every  

☐ Enabled

Debug node



Payload = message principal
Msg complete = full message
Node status = UI indication

A screenshot of the Debug node configuration and output. The left pane shows the 'Properties' tab with the following settings:

- Output: complete msg object
- To: ☒ debug window, ☐ system console, ☐ node status (32 characters)
- Name: (empty text field)

The right pane shows a list of messages received by the node, each with a timestamp, node ID, and a JSON payload. The messages are as follows:

Timestamp	Node ID	Message
14/01/2022, 16:48:17	node: c16db2db81f5b9dd	{ "_msgid": "c10de15f4569afb6" }
14/01/2022, 16:48:19	node: c16db2db81f5b9dd	{ "_msgid": "23bfed25666fd3a" }
14/01/2022, 16:48:19	node: c16db2db81f5b9dd	{ "_msgid": "078e47840386c3b3" }
14/01/2022, 16:48:20	node: c16db2db81f5b9dd	{ "_msgid": "88746dbd463a58b5" }
14/01/2022, 16:48:20	node: c16db2db81f5b9dd	{ "_msgid": "bd774c7f7f892c49" }
14/01/2022, 16:48:22	node: c16db2db81f5b9dd	{ "_msgid": "a8d3d7af2735c3b8" }
14/01/2022, 16:48:22	node: c16db2db81f5b9dd	{ "_msgid": "d79612f7f35f768f" }

NR Interface matériel

USB / Série



IPX800



10.10.10.120

root/root