Seminario 4: Node-RED

Sistemas Distribuidos

Gabriel Guerrero Contreras

Departamento de Ingeniería Informática
Universidad de Cádiz

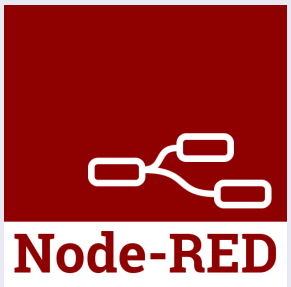
Indice

Sección 1 | Introducción

Introducción

Node-RED

- Entorno gráfico de desarrollo para construir aplicaciones Javascript
- Funciona sobre Node.js
- Open Source y diseñado por IBM
- Node-RED 1.0 fue publicado en 2019
- Su principal objetivo es simplificar el desarrollo de aplicaciones orientadas al manejo de `eventos asíncronos`



Instalación

Instalación en Linux:

1. Descargamos la información de todas las fuentes configuradas:

   ```
   $ sudo apt-get update
   ```

2. Instalamos el sistema de gestión de paquetes por defecto para Node.js:

   ```
   $ sudo apt-get install npm
   ```

3. Instalamos Node-RED:

   ```
   $ sudo npm install -g -unsafe-perm node-red
   ```

   La opción `-g` añade el comando node-red al PATH

4. Comprobamos Node-RED se ha instalado correctamente:

   ```
   $ node-red -help
   ```

   Nos debe aparecer la ayuda del comando `node-red` junto a la versión

---

Instalación en Windows: `https://nodered.org/docs/getting-started/windows`

Lanzando Node-RED Localmente

1. Desde la terminal ejecutamos el comando:

   ```
   $ node-red
   ```

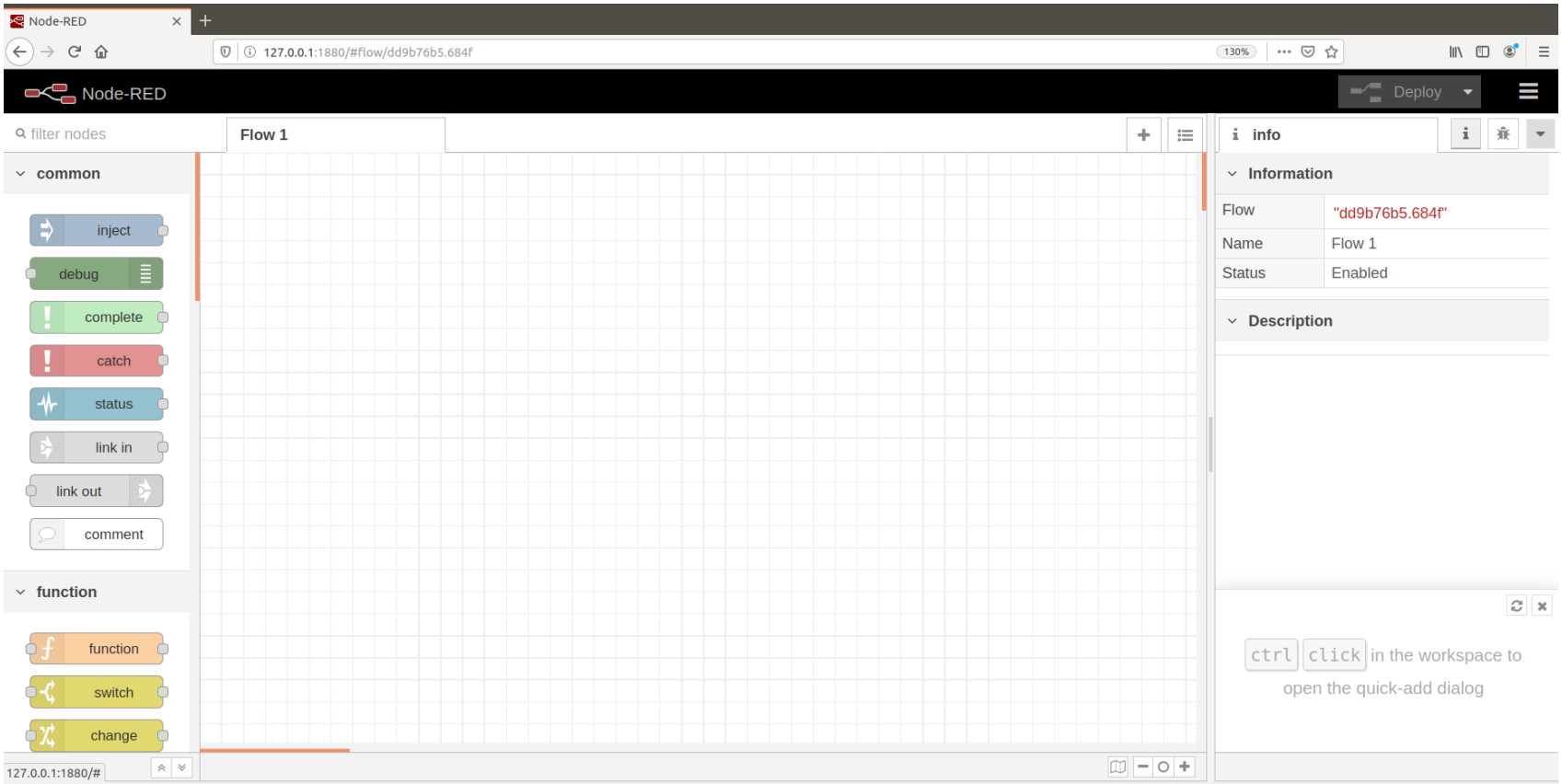
2. Entre la información mostrada debe aparecer la dirección en la que está corriendo:

   ```
   [info] Server now running at http://127.0.0.1:1880/
   ```

3. Accedemos a la dirección indicada a través del navegador web.

Lanzando Node-RED Localmente

Nos aparecerá la Interfaz de Usuario de Administración de Node-RED



Sección 2 | Conceptos Básicos

Conceptos Básicos

Node

- Unidad básico de construcción (nodo)
- Bloque de software que procesa `mensajes`
- Puede tener `entradas` y `salidas` que permiten el paso de mensajes entre nodos
- Una entrada puede aceptar conexiones de múltiples nodos y una salida puede, también, dar salida a múltiples nodos

Conceptos Básicos

Inject Node

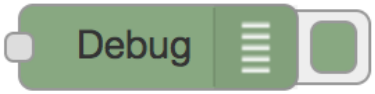
nodo de inyeccion

- Nodo de salida
- Producen mensajes sin necesidad de entrada

Debug Node

nodo de depuracion

- Nodo de entrada
- Se utiliza para mostrar mensajes de depuración en el editor



Conceptos Básicos

`Function Node` nodo de funcion

- Nodo de entrada/salida
- Permite ejecutar código `JavaScript` sobre los mensajes que pasan por él

`Change Node`

- Nodo de entrada/salida
- Puede utilizarse para modificar las propiedades de un mensaje y establecer propiedades de contexto sin tener que recurrir a un `Function Node`

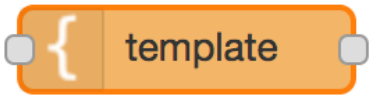
Conceptos Básicos

`Switch Node` 1:1-n

- Nodo de entrada/salida
- Permite enrutar los mensajes a diferentes ramas evaluando un conjunto de reglas en cada mensaje

`Template Node` nodo de plantilla

- Nodo de entrada/salida
- Usado para generar texto mediante una plantilla usando las propiedades de un mensaje
- Utiliza el lenguaje de plantillas Mustache

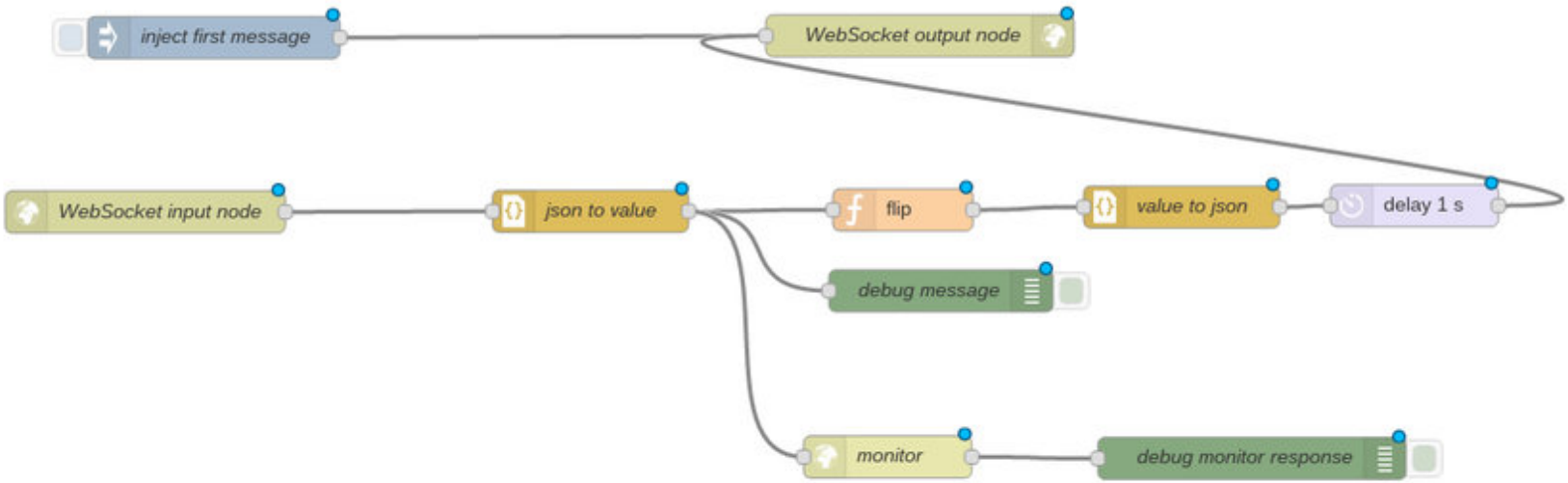


Mustache: `https://mustache.github.io/mustache.5.html`

Conceptos Básicos

Flow

- Una colección de nodos `conectados` entre sí y funcionando como una aplicación o programa
- Cada flujo puede tener un nombre y una descripción que se muestra en la barra lateral de información
- Todos los nodos de un flujo pueden acceder al mismo `contexto` de flujo



Conceptos Básicos

Message

- Es la información que se pasa entre los nodos de un flujo
- Son simples `JavaScript Objects` que pueden tener cualquier conjunto de propiedades (`payload`)
- En el editor aparecerán como `msg`
- El valor de una propiedad puede ser cualquier tipo de JavaScript válido, como:
  - Boolean (`true`, `false`), Number (`5`, `23.56`), String (`"Hello"`), Array (`[1,2,3,4]`), Object (`{"a":1, "b":2}`) o Null
- Cuidado, se usa el formato JSON, JSON $\neq$ Javascritp Object

Wire

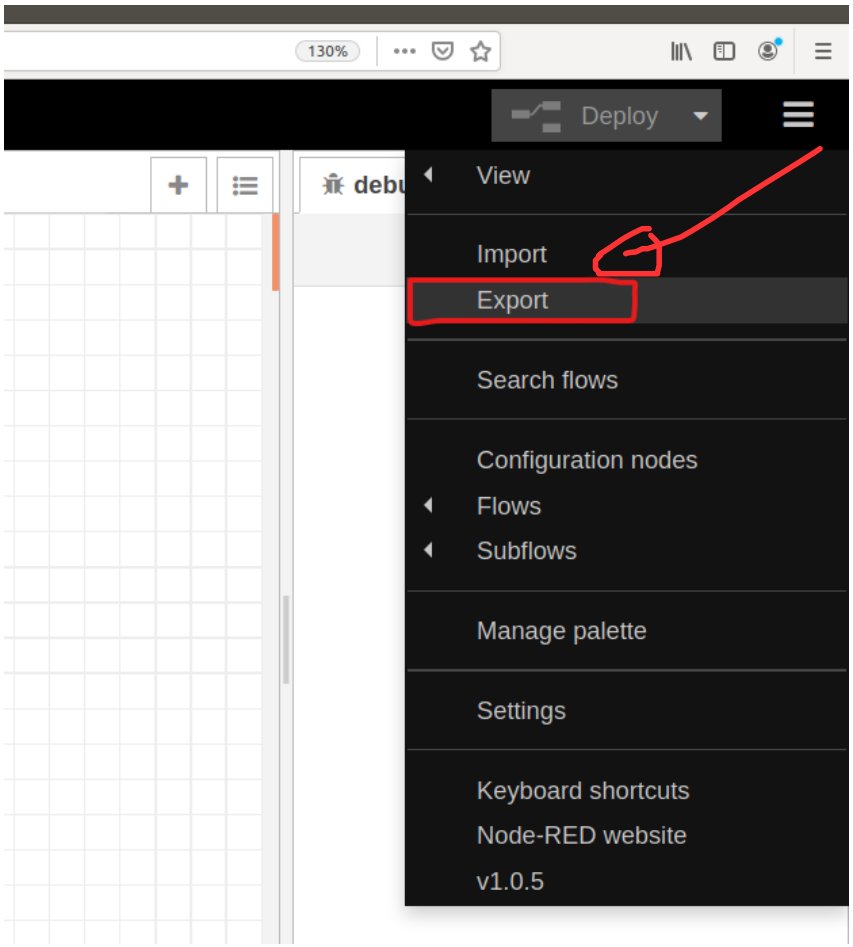
- Conectan los nodos y representan cómo los mensajes pasan a través del flujo (sin propiedad)

Conceptos Básicos

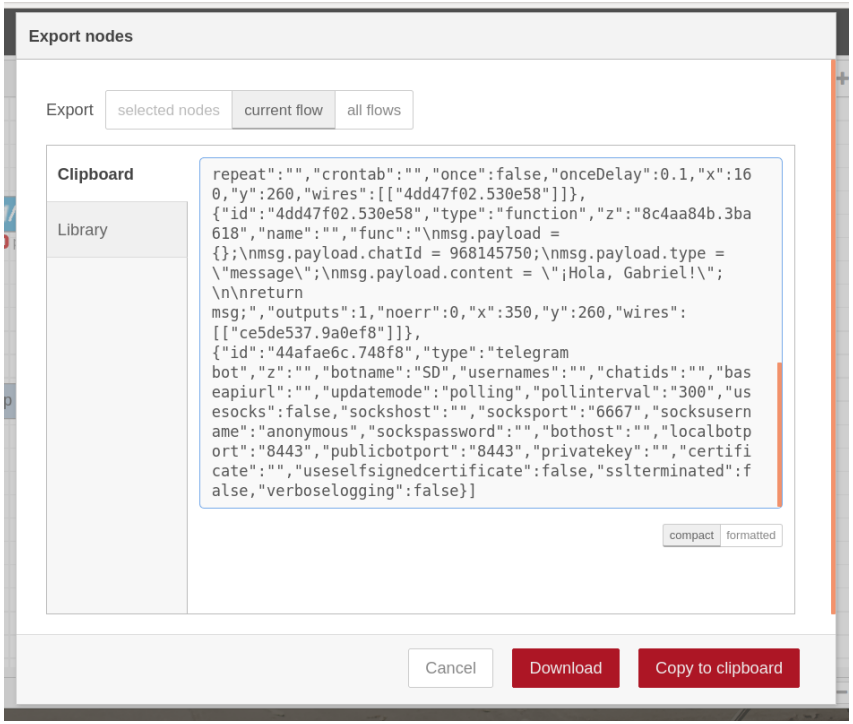
Context

- Utilizado para almacenar información que puede ser compartida entre diferentes nodos sin usar los mensajes que pasan a través de un flujo
- Alcance:
  - `Node`: solo visible para el nodo que almacena el valor
  - `Flow`: visible para todos los nodos que comparten el mismo flujo
  - `Global`: visible a todos los nodos
- Por defecto, el contexto se almacena solo en memoria, es decir, su contenido se borra cada vez que Node-RED se reinicia

Exportar Proyecto

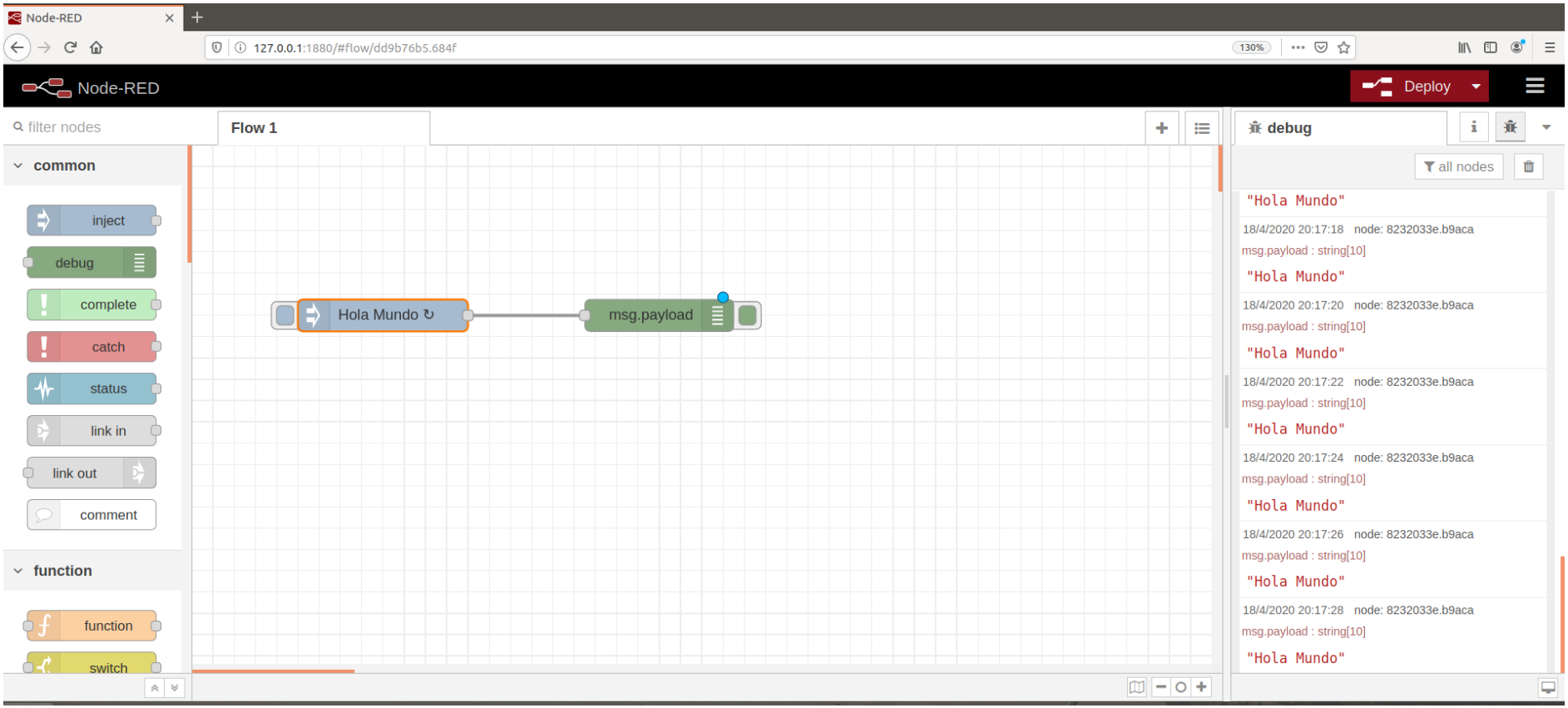


JSON

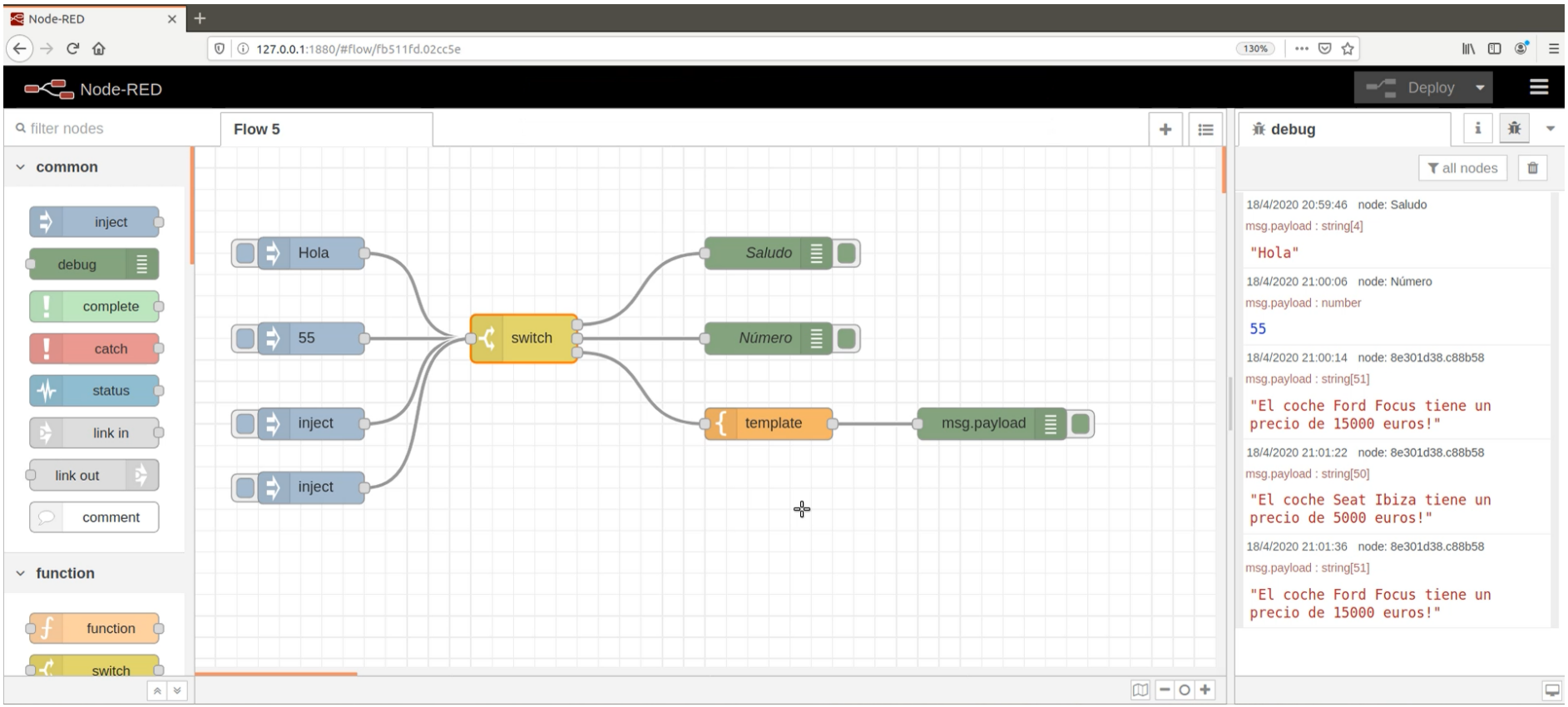


Sección 3 | Creando Flujos

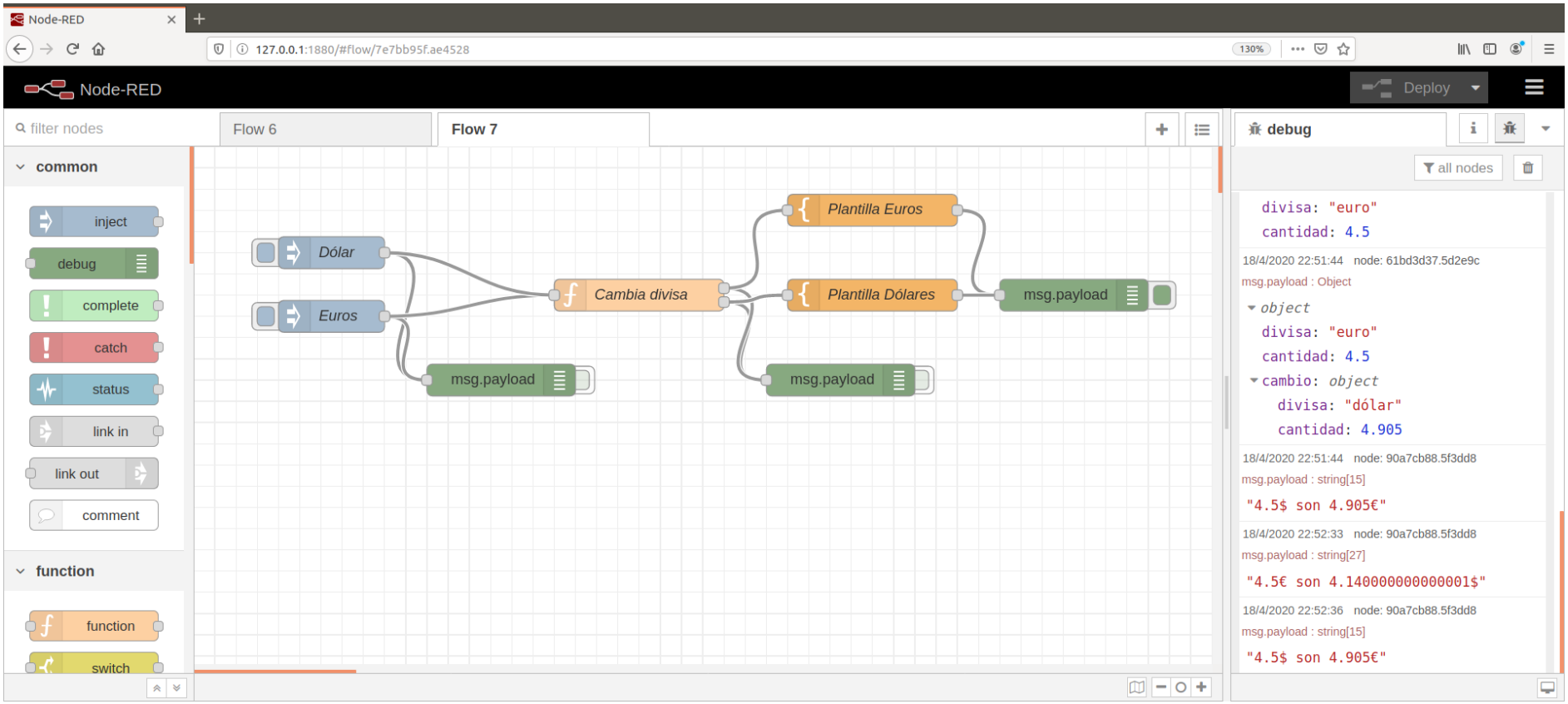
Hola Mundo


Node-RED
127.0.0.1:1880/#flow/dd9b76b5.684f
Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
Hola Mundo
msg.payload
debug
all nodes
"Hola Mundo"
18/4/2020 20:17:18 node: 8232033e.b9aca
msg.payload : string[10]
"Hola Mundo"
18/4/2020 20:17:20 node: 8232033e.b9aca
msg.payload : string[10]
"Hola Mundo"
18/4/2020 20:17:22 node: 8232033e.b9aca
msg.payload : string[10]
"Hola Mundo"
18/4/2020 20:17:24 node: 8232033e.b9aca
msg.payload : string[10]
"Hola Mundo"
18/4/2020 20:17:26 node: 8232033e.b9aca
msg.payload : string[10]
"Hola Mundo"
18/4/2020 20:17:28 node: 8232033e.b9aca
msg.payload : string[10]
"Hola Mundo"

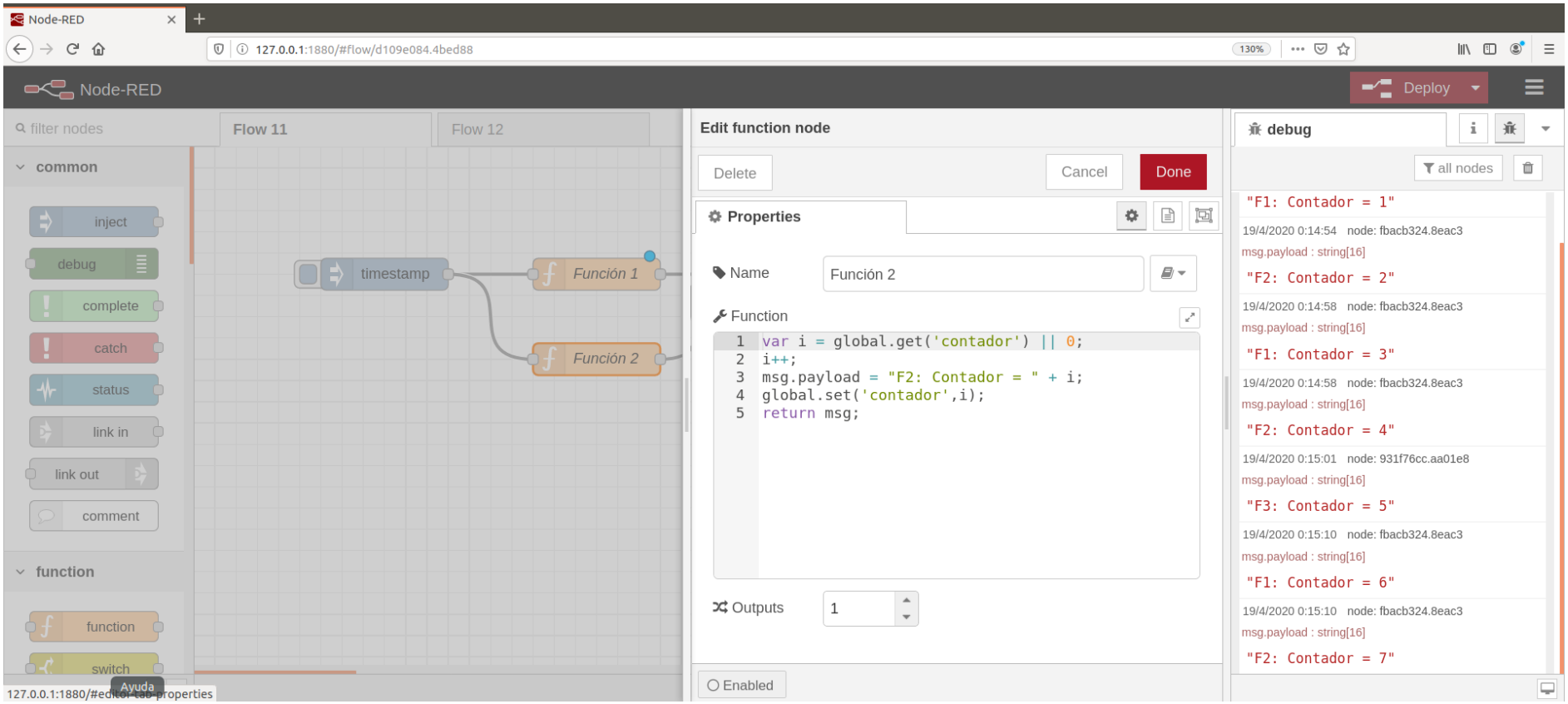
Switch & Template Nodes



Function Node

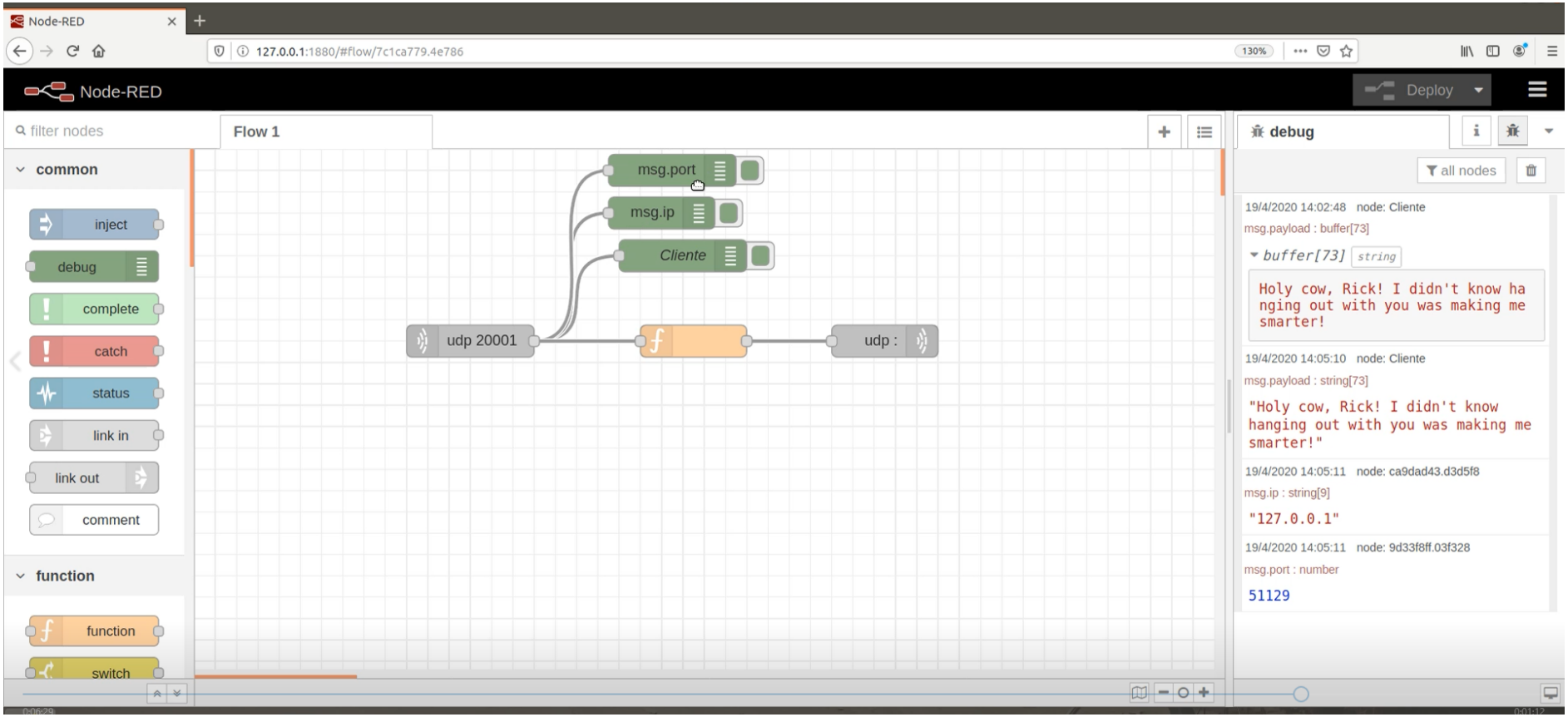


Context

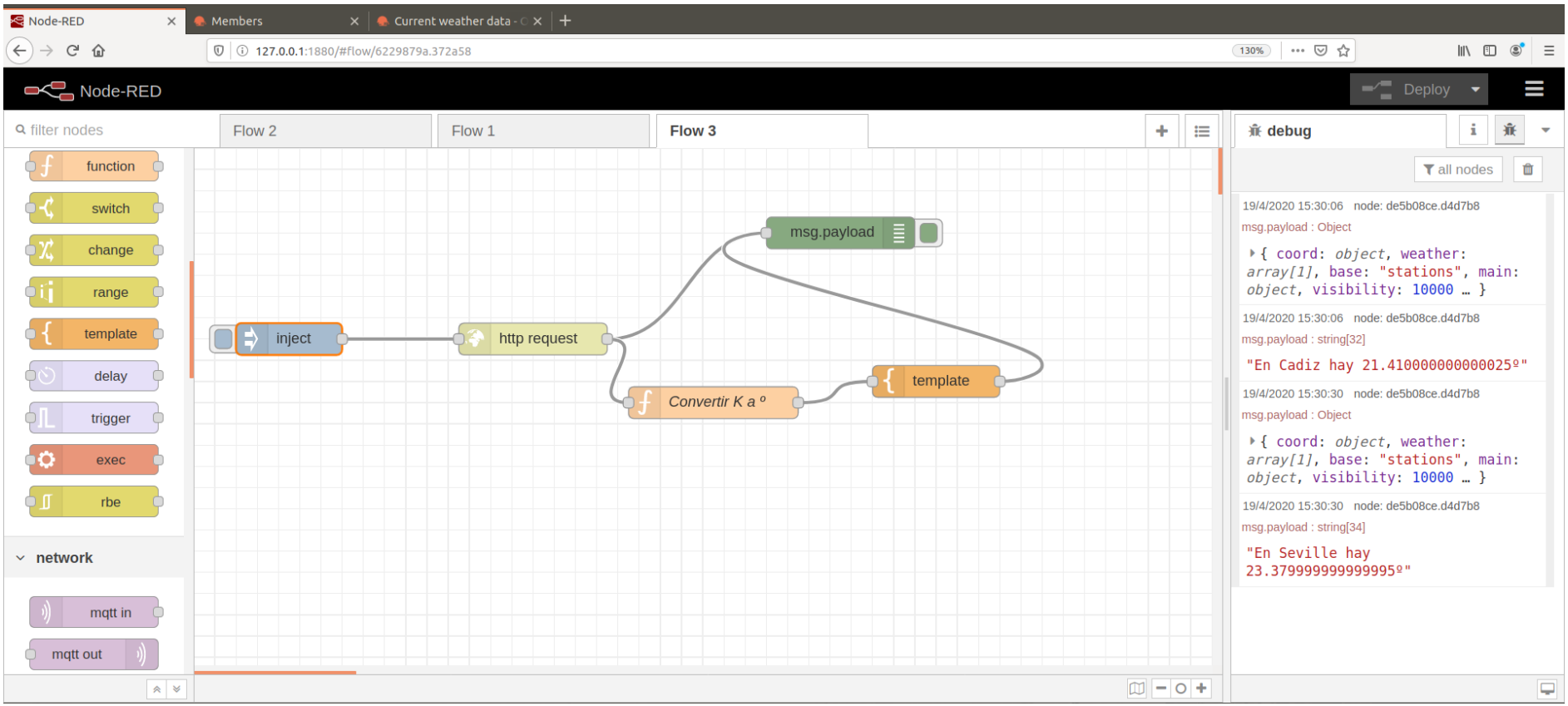

Node-RED
127.0.0.1:1880/#flow/d109e084.4bed88
Node-RED
Deploy
filter nodes
Flow 11
Flow 12
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
Ayuda
timestamp
Función 1
Función 2
Edit function node
Delete
Cancel
Done
Properties
Name
Función 2
Function
1 var i = global.get('contador') || 0;
2 i++;
3 msg.payload = "F2: Contador = " + i;
4 global.set('contador',i);
5 return msg;
Outputs
1
Enabled
debug
all nodes
"F1: Contador = 1"
19/4/2020 0:14:54 node: fbacb324.8eac3
msg.payload : string[16]
"F2: Contador = 2"
19/4/2020 0:14:58 node: fbacb324.8eac3
msg.payload : string[16]
"F1: Contador = 3"
19/4/2020 0:14:58 node: fbacb324.8eac3
msg.payload : string[16]
"F2: Contador = 4"
19/4/2020 0:15:01 node: 931f76cc.aa01e8
msg.payload : string[16]
"F3: Contador = 5"
19/4/2020 0:15:10 node: fbacb324.8eac3
msg.payload : string[16]
"F1: Contador = 6"
19/4/2020 0:15:10 node: fbacb324.8eac3
msg.payload : string[16]
"F2: Contador = 7"

Sección 4 | Sockets y HTTP

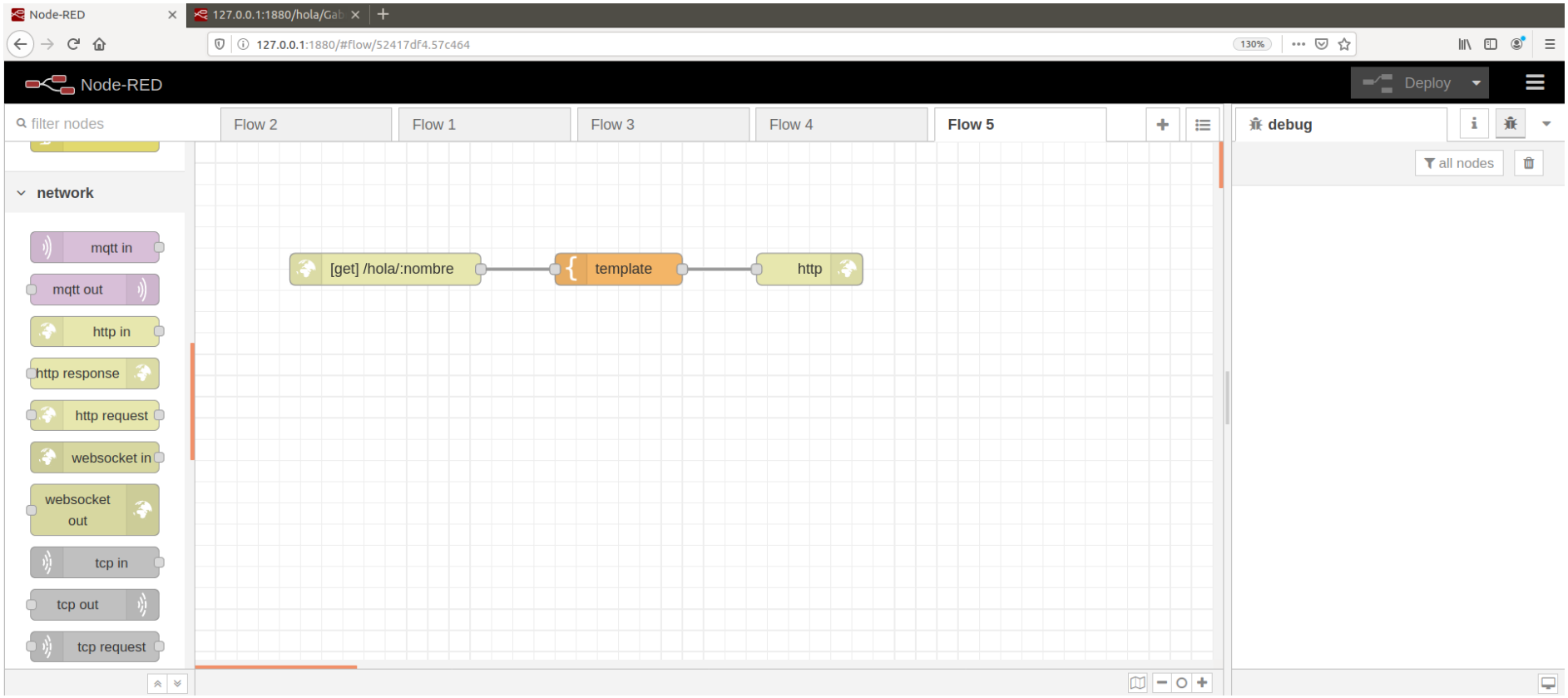
UDP


Node-RED
127.0.0.1:1880/#flow/7c1ca779.4e786
Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
msg.port
msg.ip
Cliente
udp 20001
udp :
debug
all nodes
19/4/2020 14:02:48 node: Cliente
msg.payload : buffer[73]
buffer[73] string
Holy cow, Rick! I didn't know ha nging out with you was making me smarter!
19/4/2020 14:05:10 node: Cliente
msg.payload : string[73]
"Holy cow, Rick! I didn't know hanging out with you was making me smarter!"
19/4/2020 14:05:11 node: ca9dad43.d3d5f8
msg.ip : string[9]
"127.0.0.1"
19/4/2020 14:05:11 node: 9d33f8ff.03f328
msg.port : number
51129

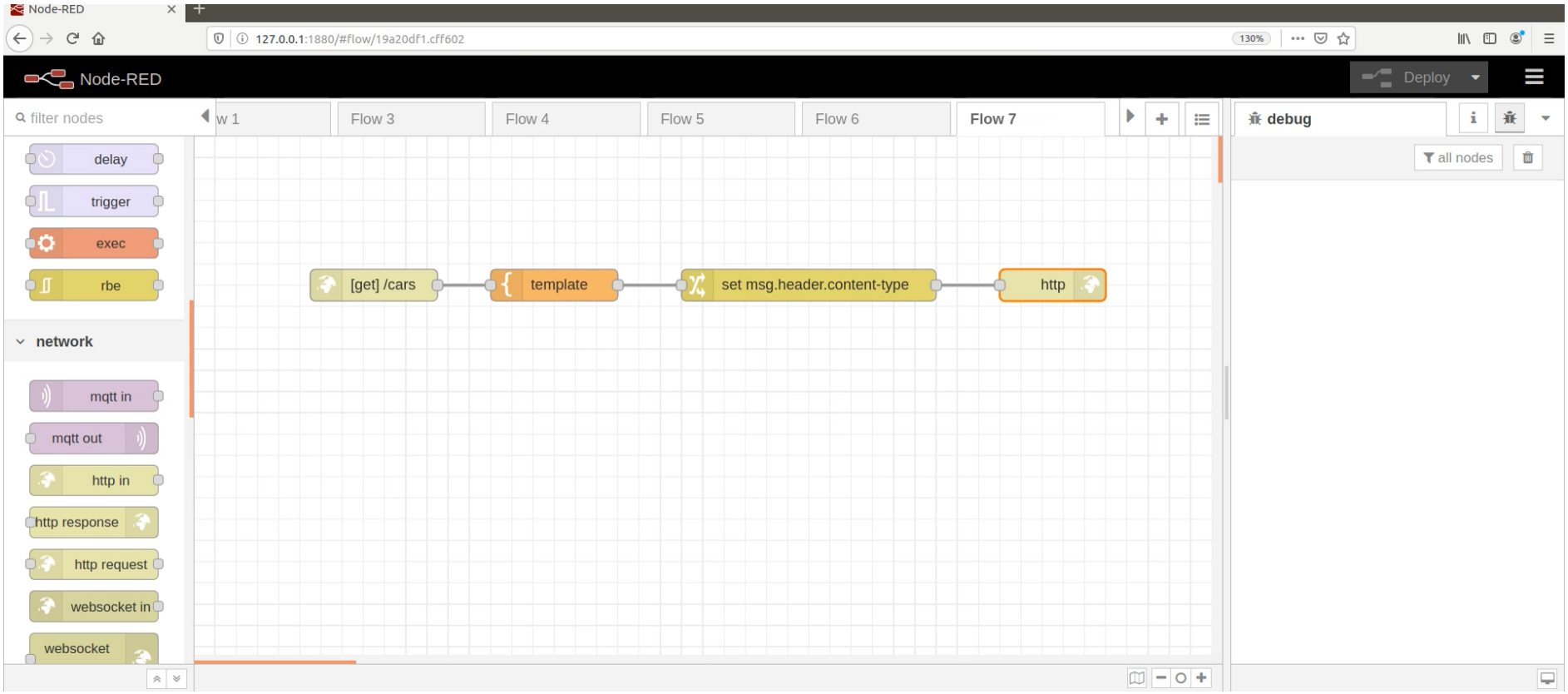
GET Request


Node-RED
Members
Current weather data -
127.0.0.1:1880/#flow/6229879a.372a58
130%
Node-RED
Deploy
filter nodes
Flow 2
Flow 1
Flow 3
debug
all nodes
function
switch
change
range
template
delay
trigger
exec
rbe
network
mqtt in
mqtt out
inject
http request
msg.payload
Convertir K a º
template
19/4/2020 15:30:06 node: de5b08ce.d4d7b8
msg.payload : Object
{ coord: object, weather: array[1], base: "stations", main: object, visibility: 10000 … }
19/4/2020 15:30:06 node: de5b08ce.d4d7b8
msg.payload : string[32]
"En Cadiz hay 21.4100000000000025º"
19/4/2020 15:30:30 node: de5b08ce.d4d7b8
msg.payload : Object
{ coord: object, weather: array[1], base: "stations", main: object, visibility: 10000 … }
19/4/2020 15:30:30 node: de5b08ce.d4d7b8
msg.payload : string[34]
"En Seville hay 23.3799999999999995º"

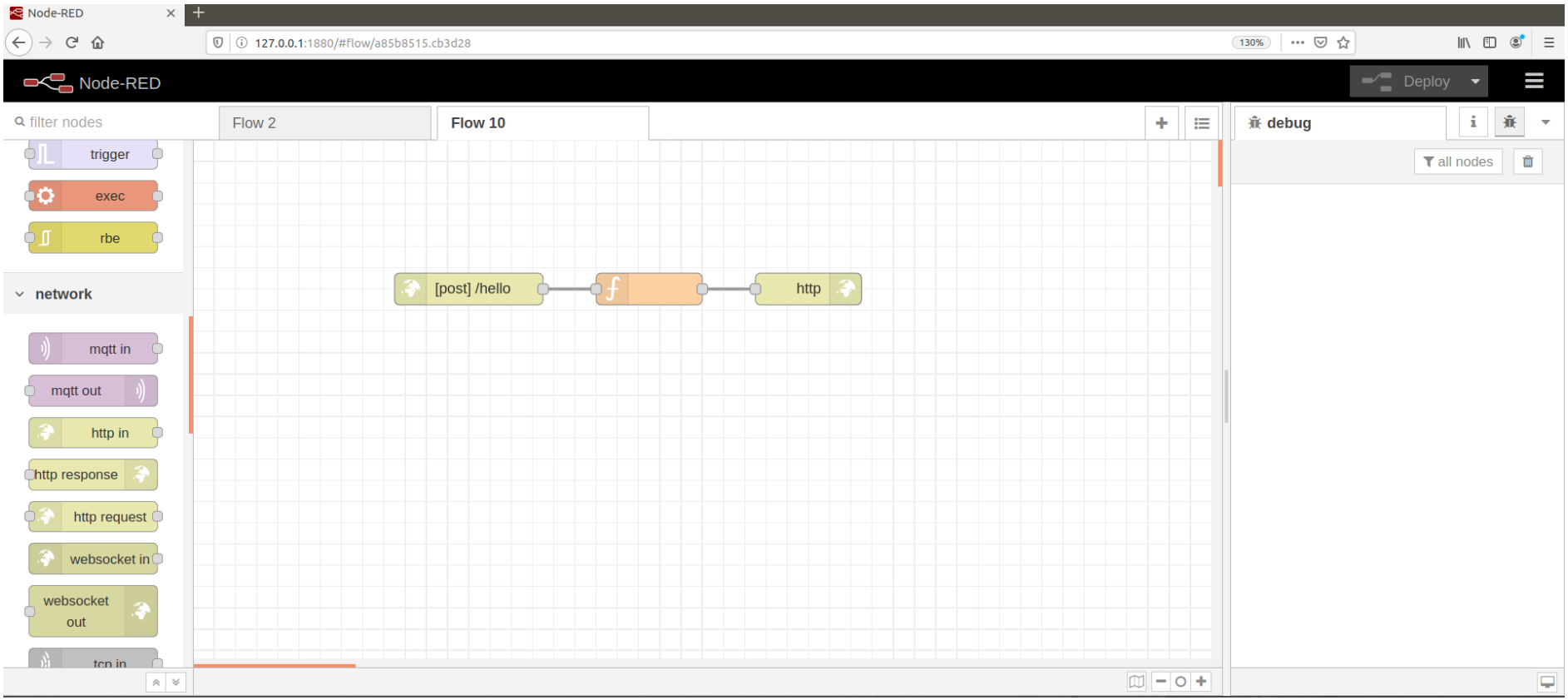
Endpoint GET + HTML Response



Endpoint GET + JSON Response

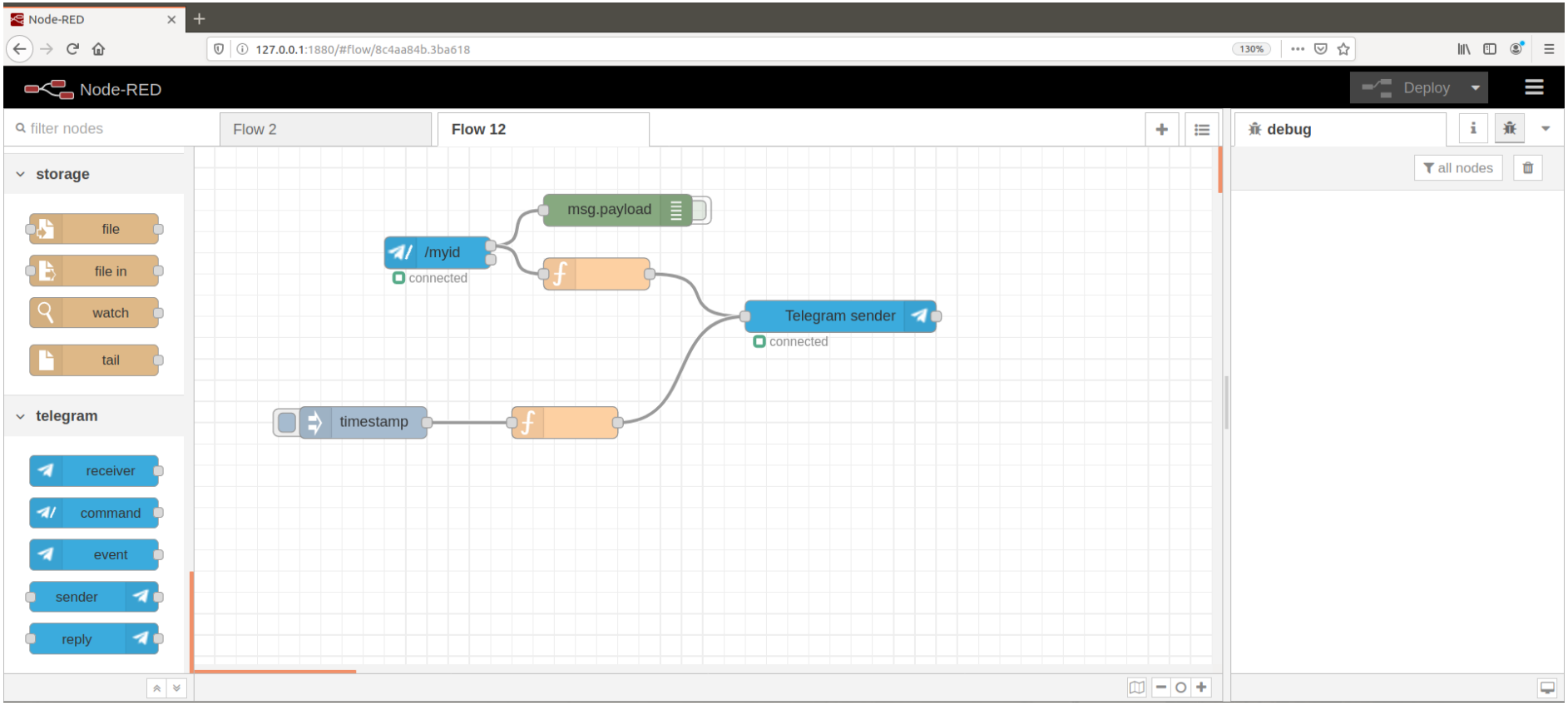


Endpoint POST + JSON Response



Sección 5 | Extensiones

Telegram


Node-RED
127.0.0.1:1880/#flow/8c4aa84b.3ba618
Node-RED
Deploy
filter nodes
Flow 2
Flow 12
debug
all nodes
storage
file
file in
watch
tail
telegram
receiver
command
event
sender
reply
msg.payload
/myid
connected
Telegram sender
connected
timestamp

Sección 6 | Bibliografía

Bibliografía

- https://nodered.org/docs/
- https://flows.nodered.org/node/node-red-contrib-telegrambot