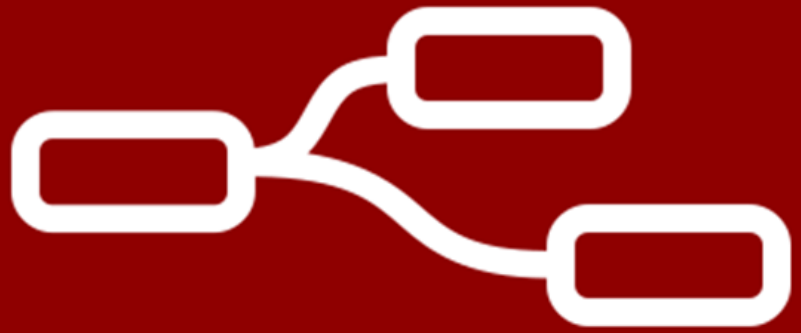
UNIVERSITÀ DEGLI STUDI FIRENZE
DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB DISTRIBUTED DATA INTELLIGENCE AND TECHNOLOGIES LAB

SNAP4CITY

Snap4City IoT App



Corso di: Big Data Architectures

Prof. Paolo Nesi

Dep DINFO, University of Florence
Via S. Marta 3, 50139, Firenze, Italy
DISIT Lab, Sistemi Distribuiti e Tecnologie Internet
http://www.disit.dinfo.unifi.it/ , Https://www.disit.org
paolo.nesi@unifi.it http://www.disit.dinfo.unifi.it/nesi

Node-RED

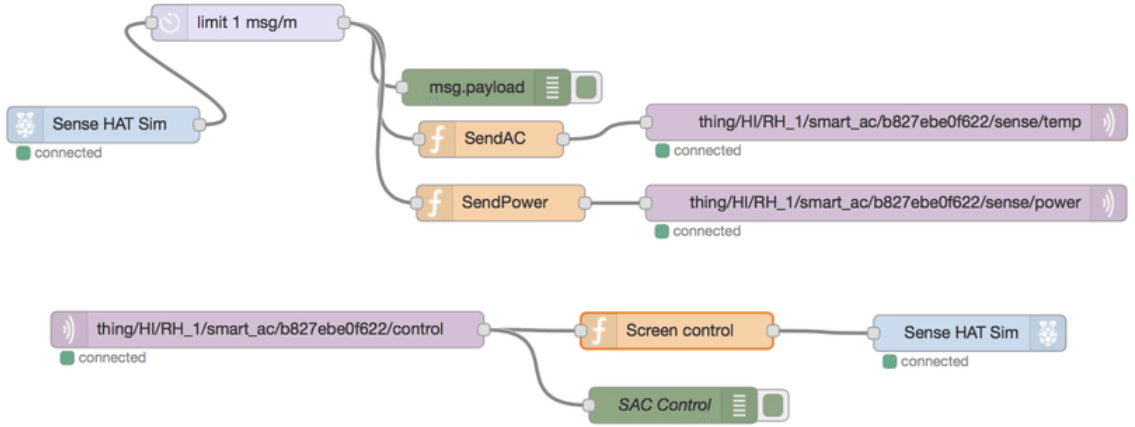
QUICK INTRO

BIG DATA ARCHITECTURES

a.a. 2023/2024

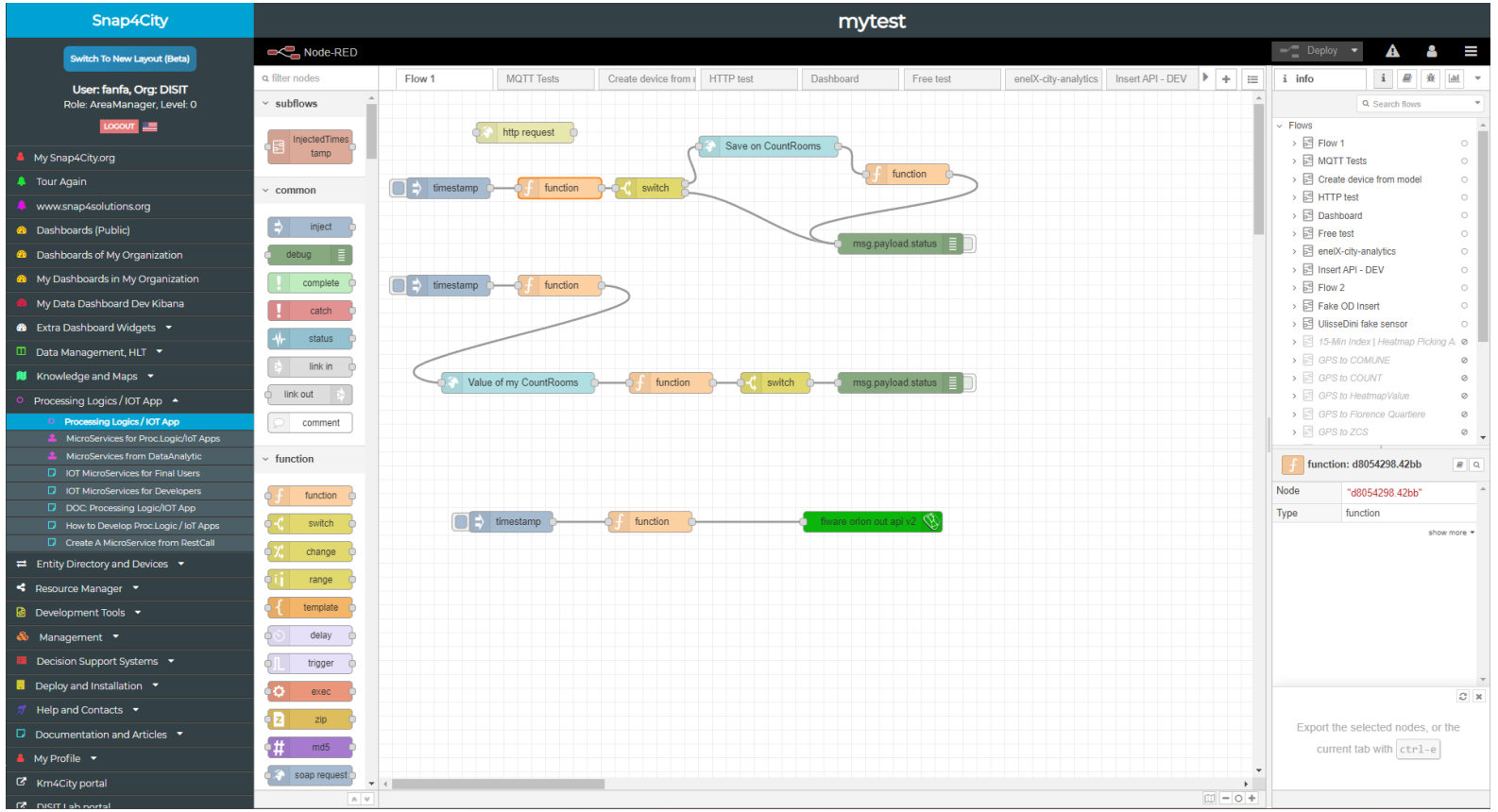
Node-RED

- Node-RED (https://nodered.org/) is a programming tool for **wiring together** hardware devices, APIs and online services to create apps
- It provides a **browser-based editor** that makes it easy to wire together flows using the wide range of nodes…
- …and then **deploy** and **run** the app in a single-click



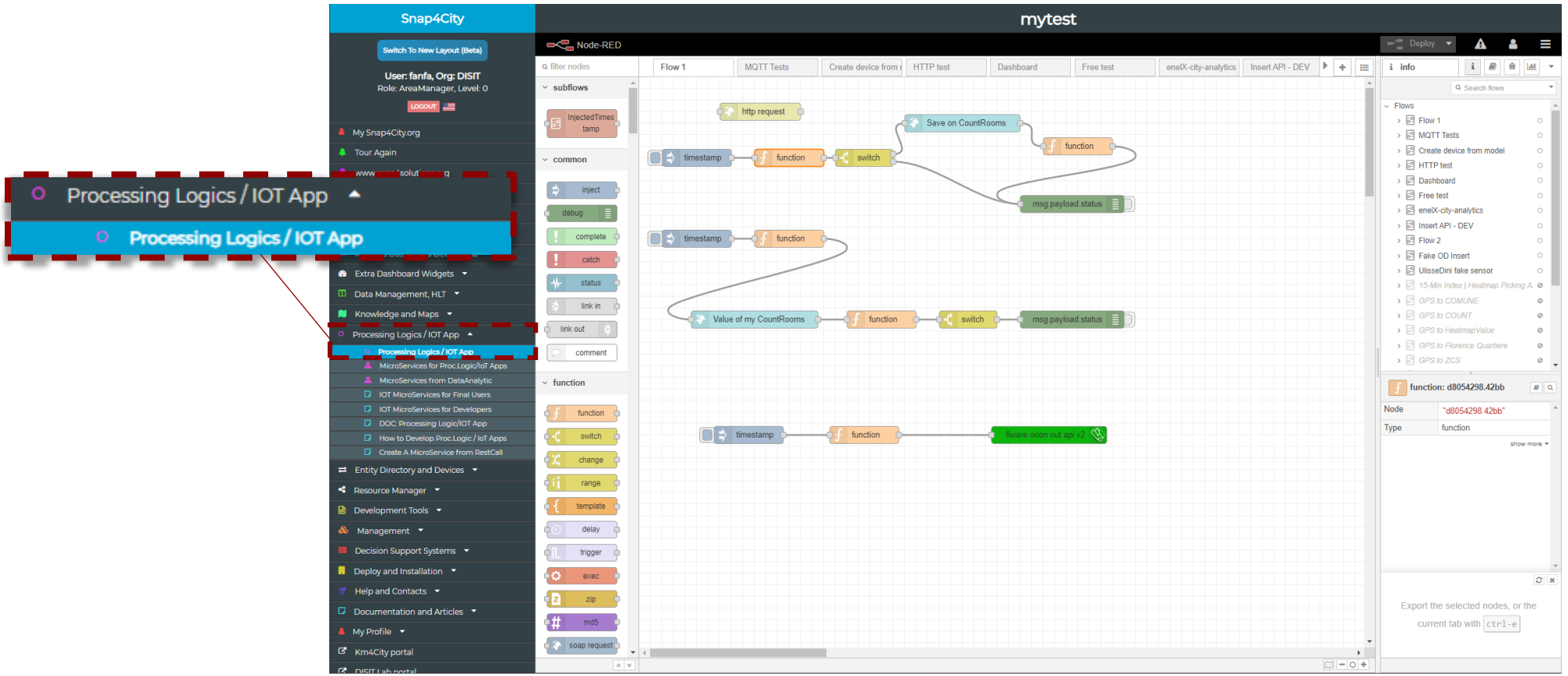
Node-RED in Snap4City

- Node-RED is integrated into the Snap4City platform



Node-RED in Snap4City

- Node-RED is integrated into the Snap4City platform



Node-RED in Snap4City

- Node-RED is integrated into the Snap4City platform
- It can be used to
  - Real-time data ingestion trough IoT Orion broker
  - Data manipulation and conversion
  - Execute analytic processes (even with specific scheduling or after receiving events)
  - Perform business-logic operation on data
  - Create and control dashboard widgets
  - …

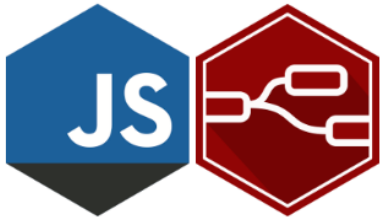
Nodes

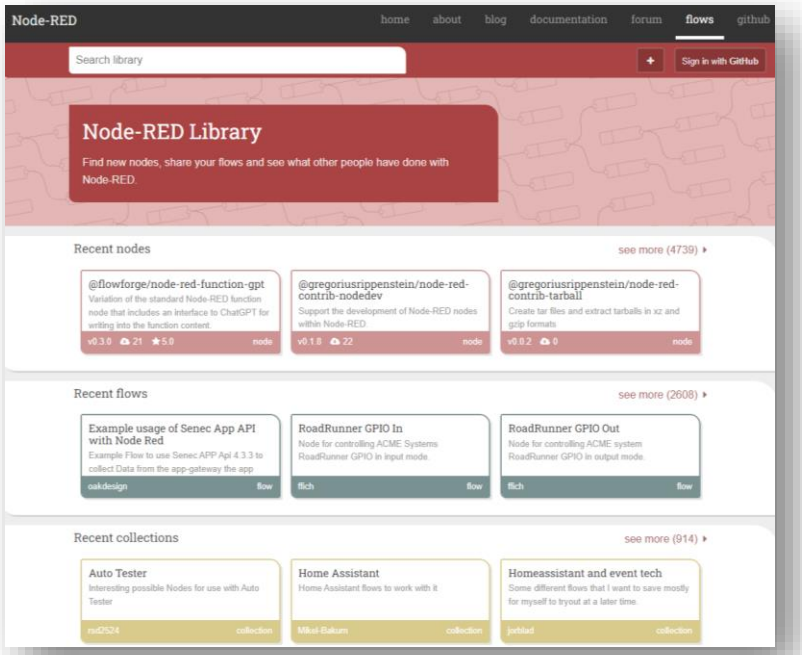
- Nodes are the building-blocks used to create flows
- Nodes can be of three main types
  - Input nodes (e.g., the **inject**) 

  - Output nodes (e.g., the **debug**) 

  - Input/output nodes (e.g., the **function**) 

- The basic installation is provided with a minimum set of nodes
- Many other blocks can be easily added loading them from a large library made available by the JS Foundation



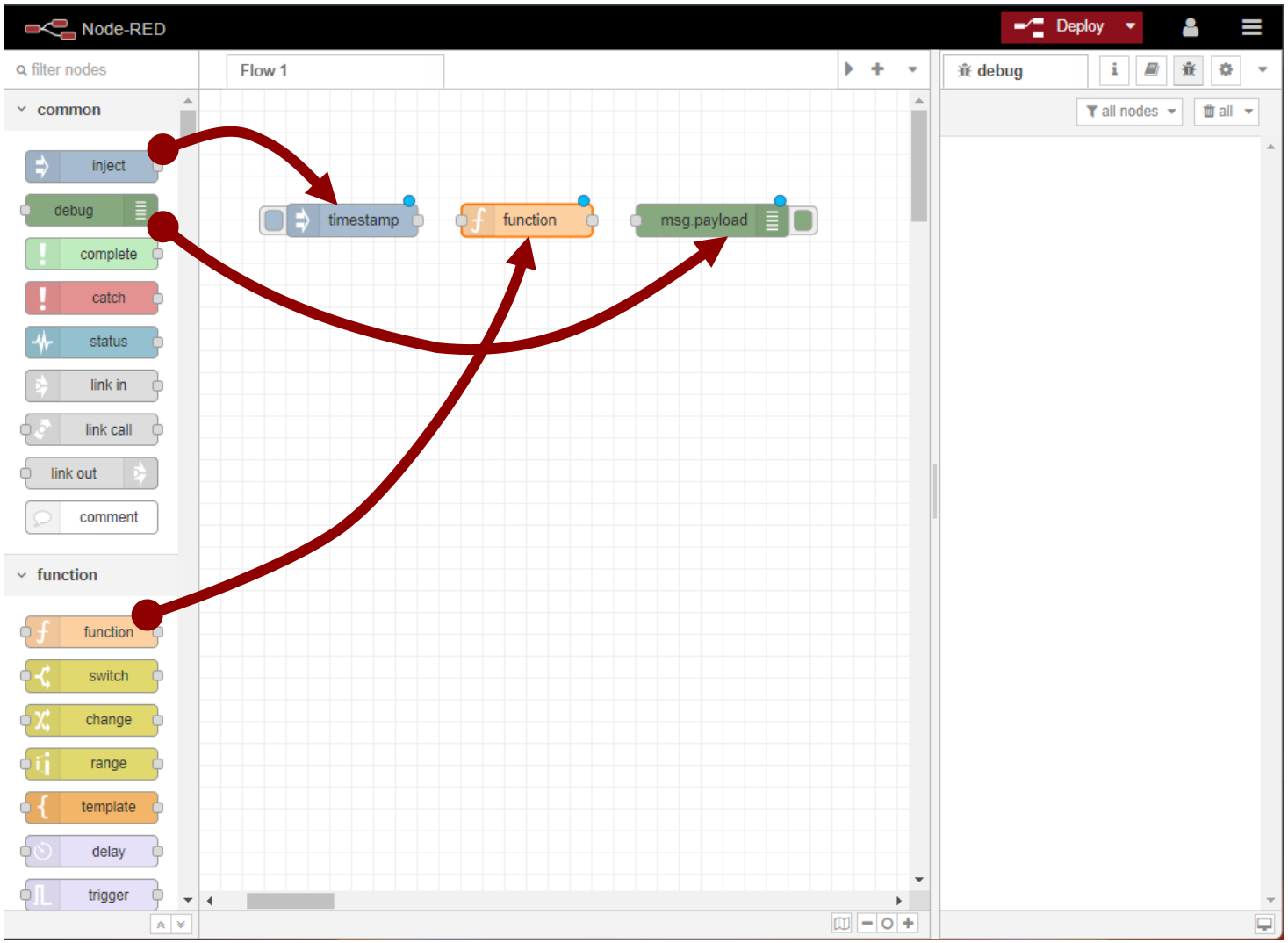


https://flows.nodered.org/

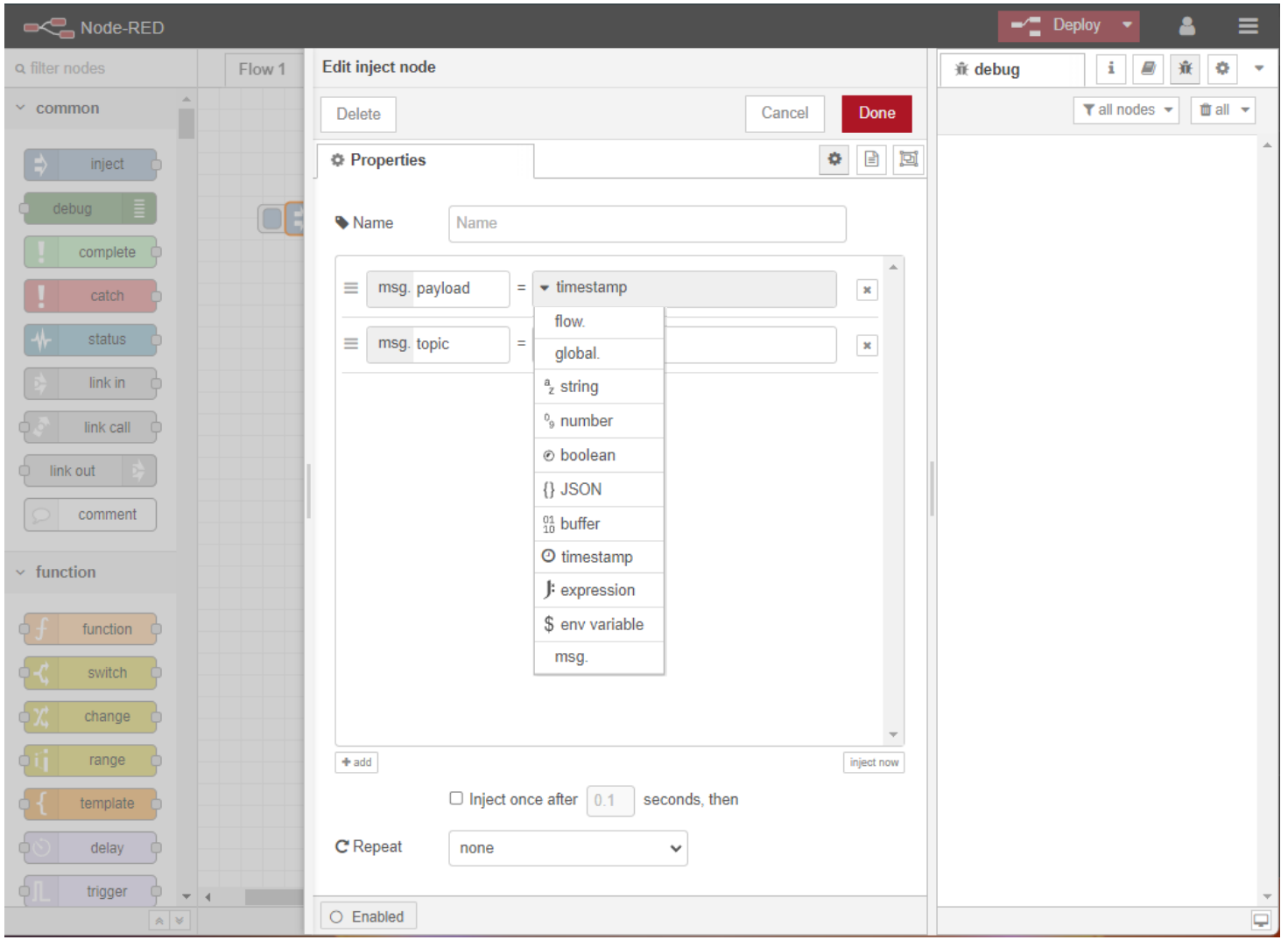
Nodes and flows

- In the Node-RED editor each node can be dragged into the main window
- Different nodes can then be linked to build flows
- Then each node in the flow can communicate with its neighbors by passing messages that are simple JavaScript objects that can have any set of properties

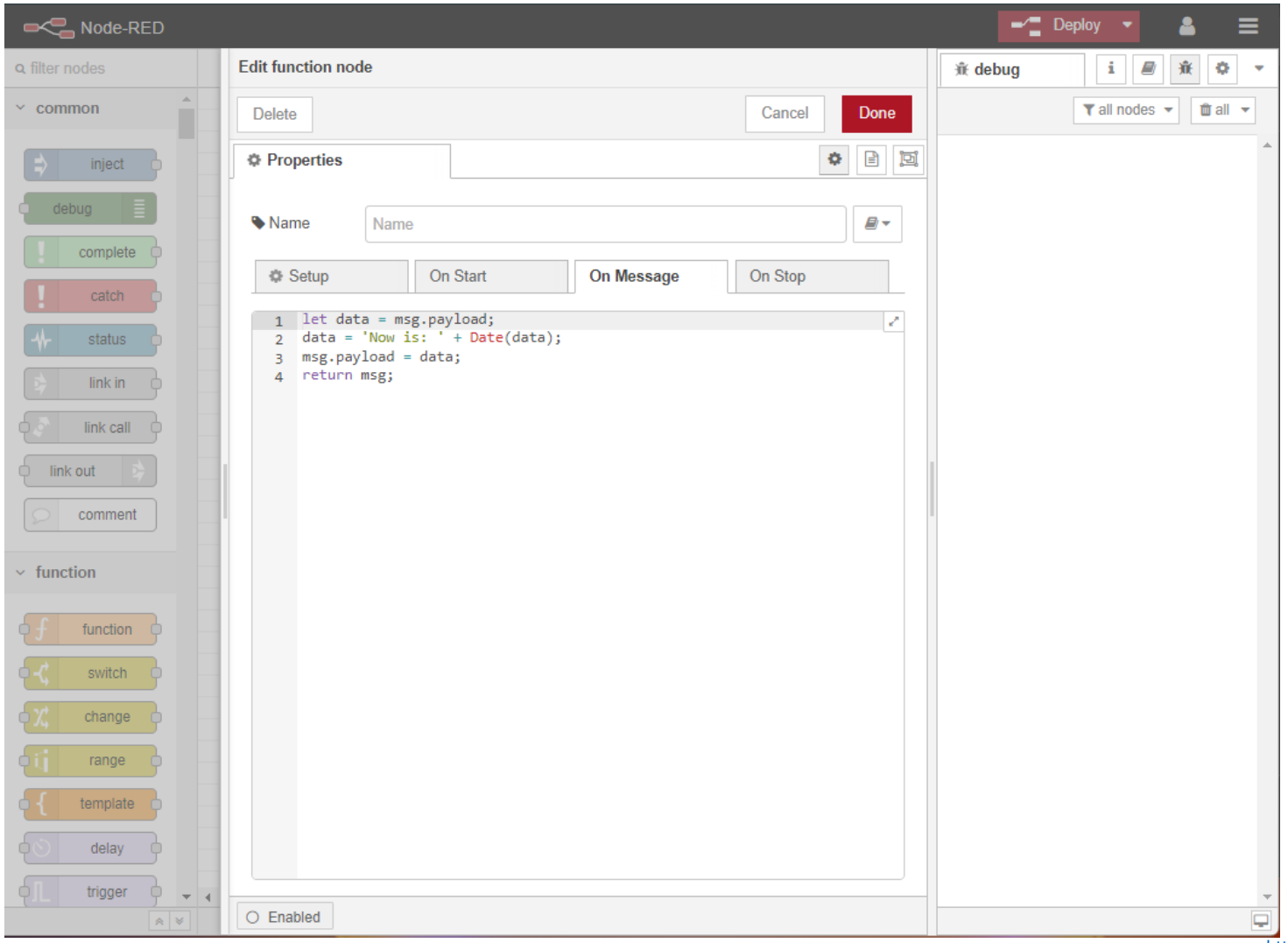
Nodes and flows



Nodes and flows

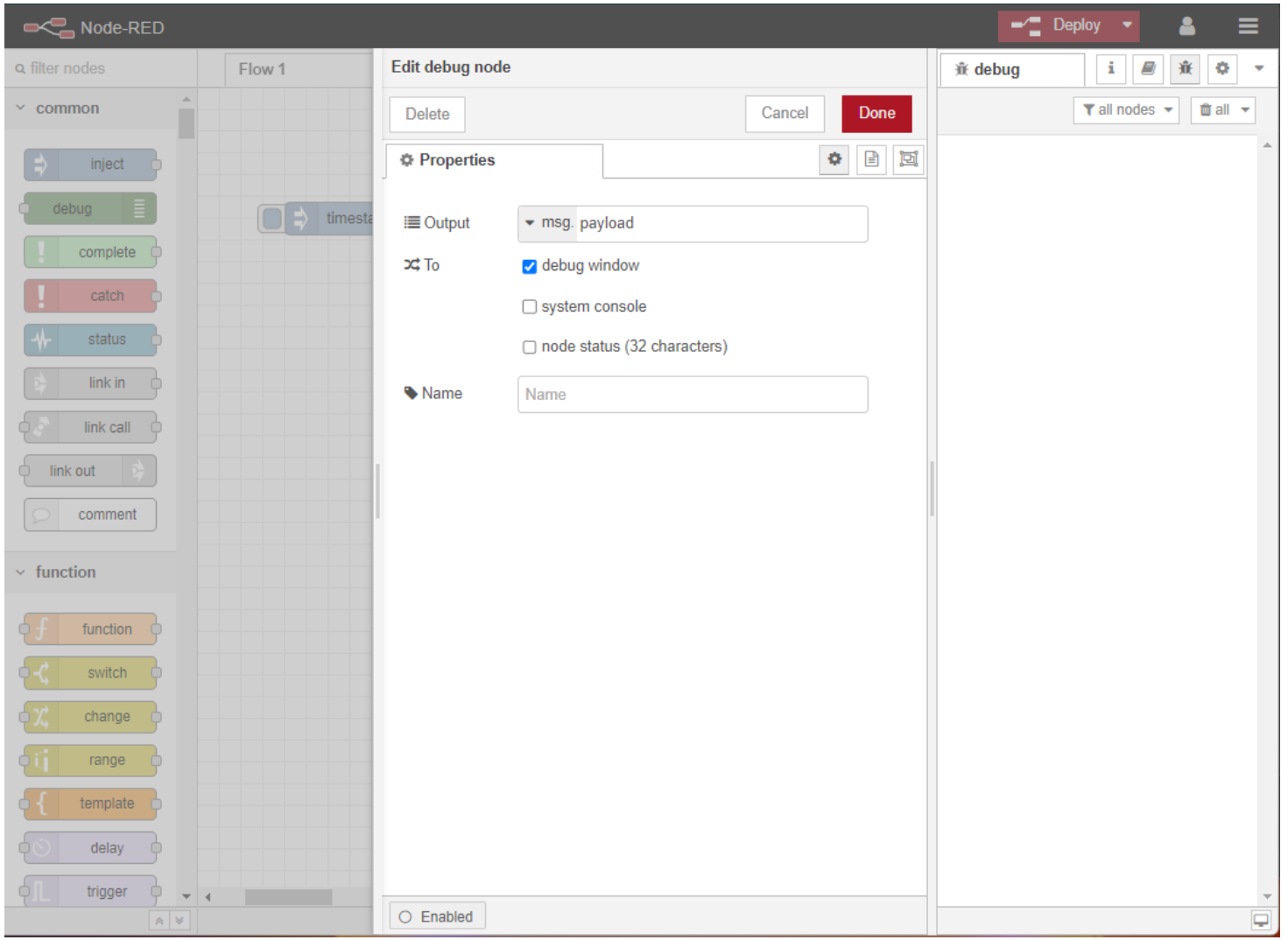


Nodes and flows

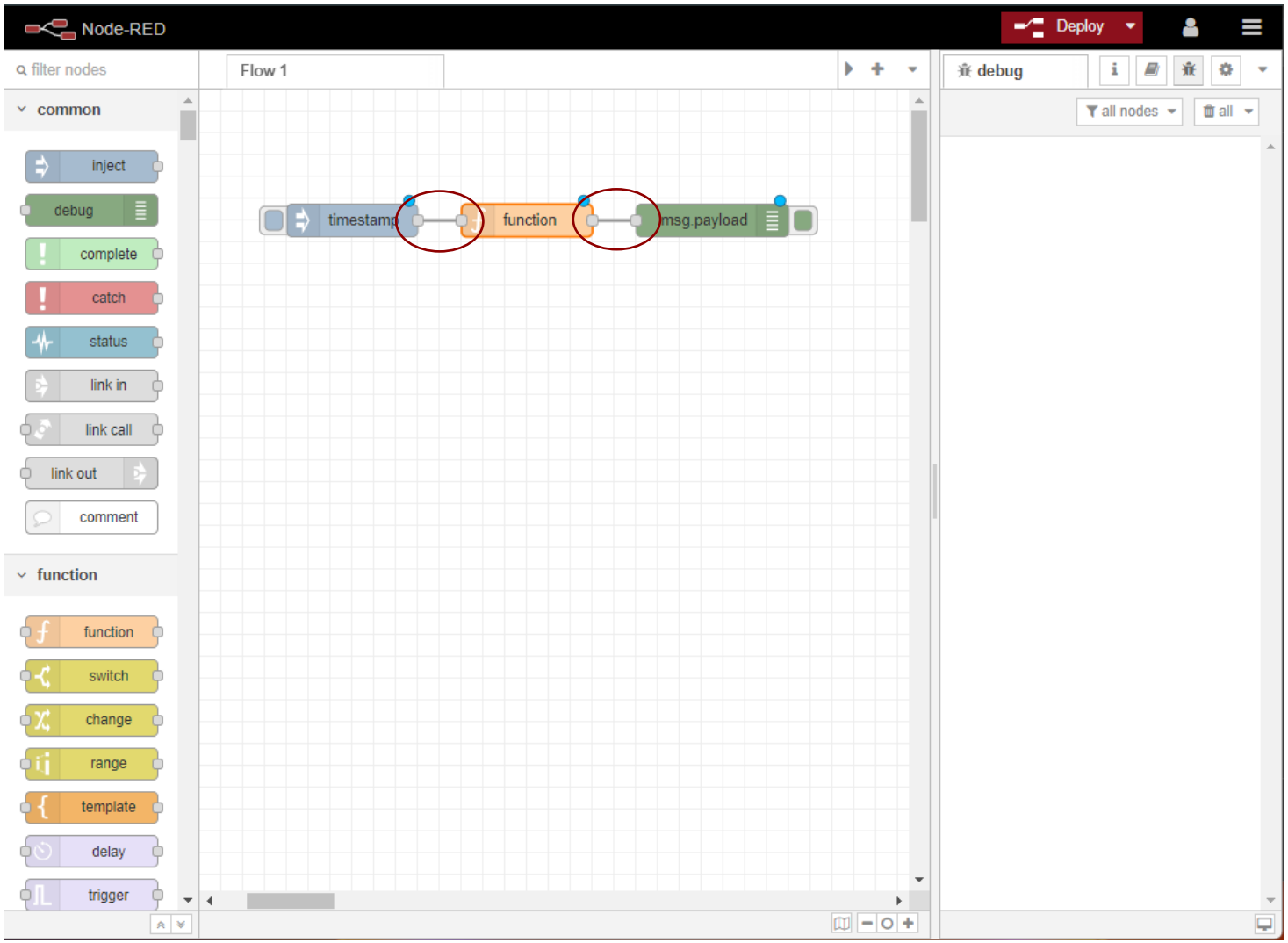


https://nodered.org/docs/user-guide/writing-functions

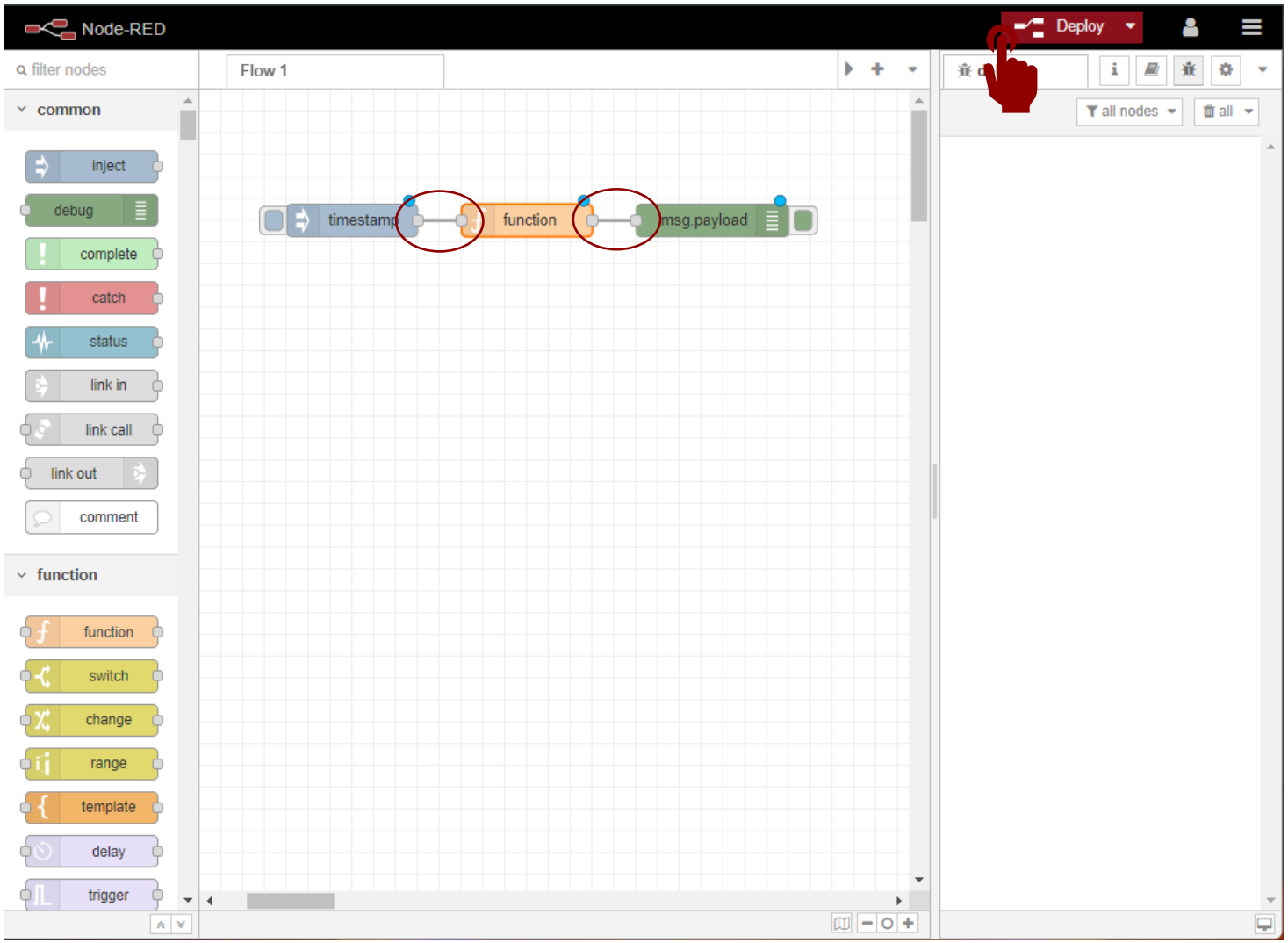
Nodes and flows



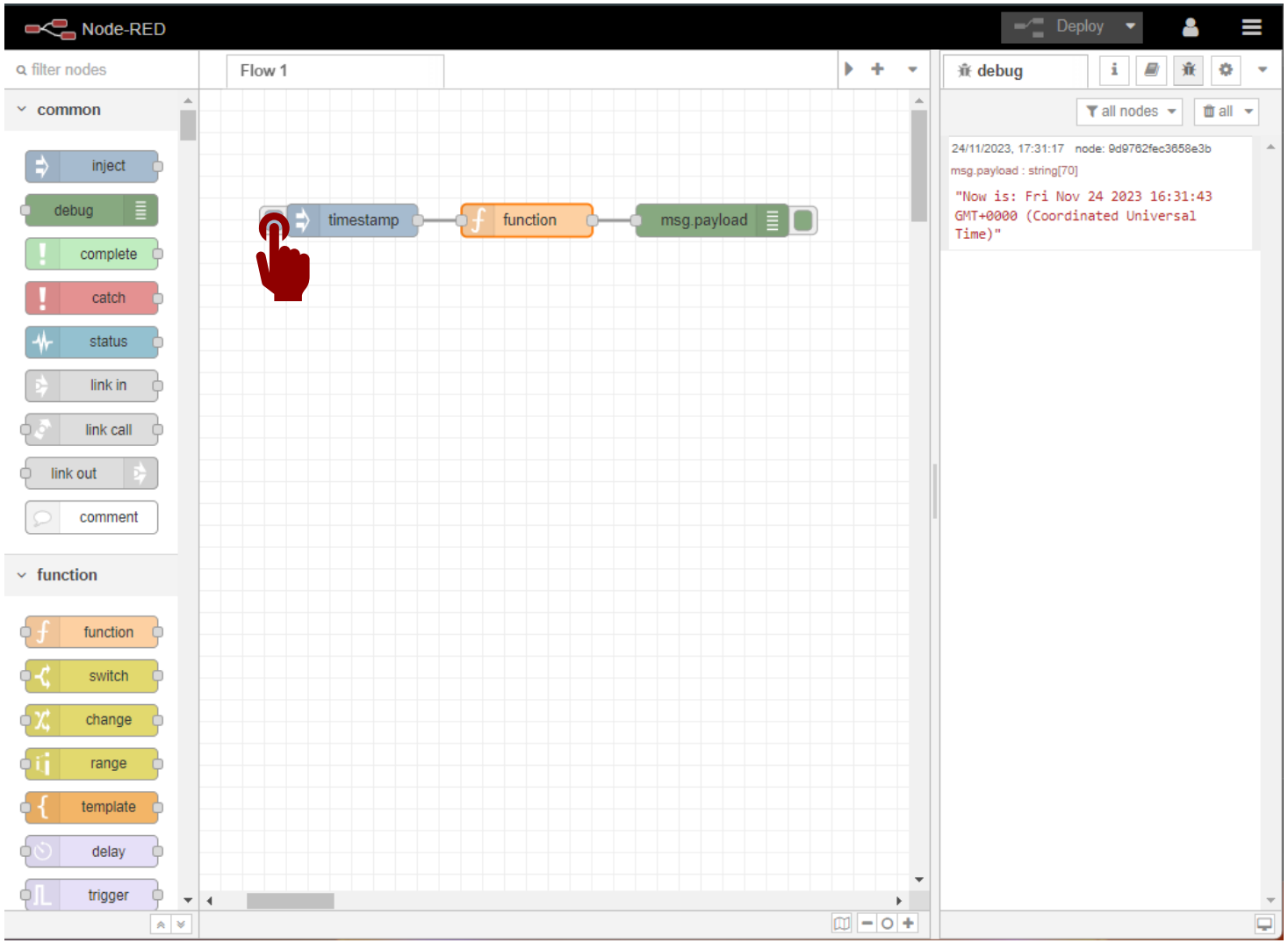
Nodes and flows



Nodes and flows



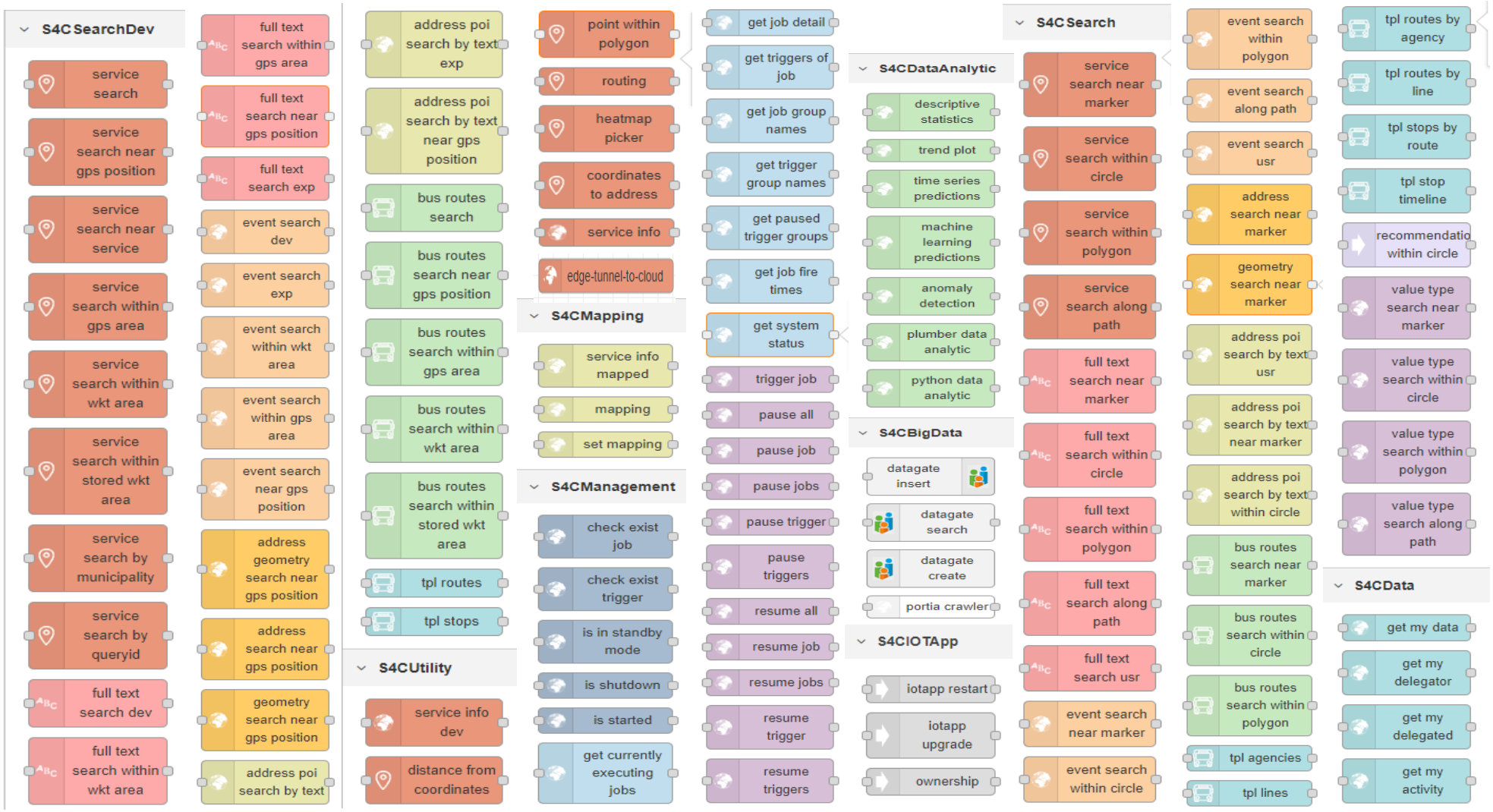
Nodes and flows



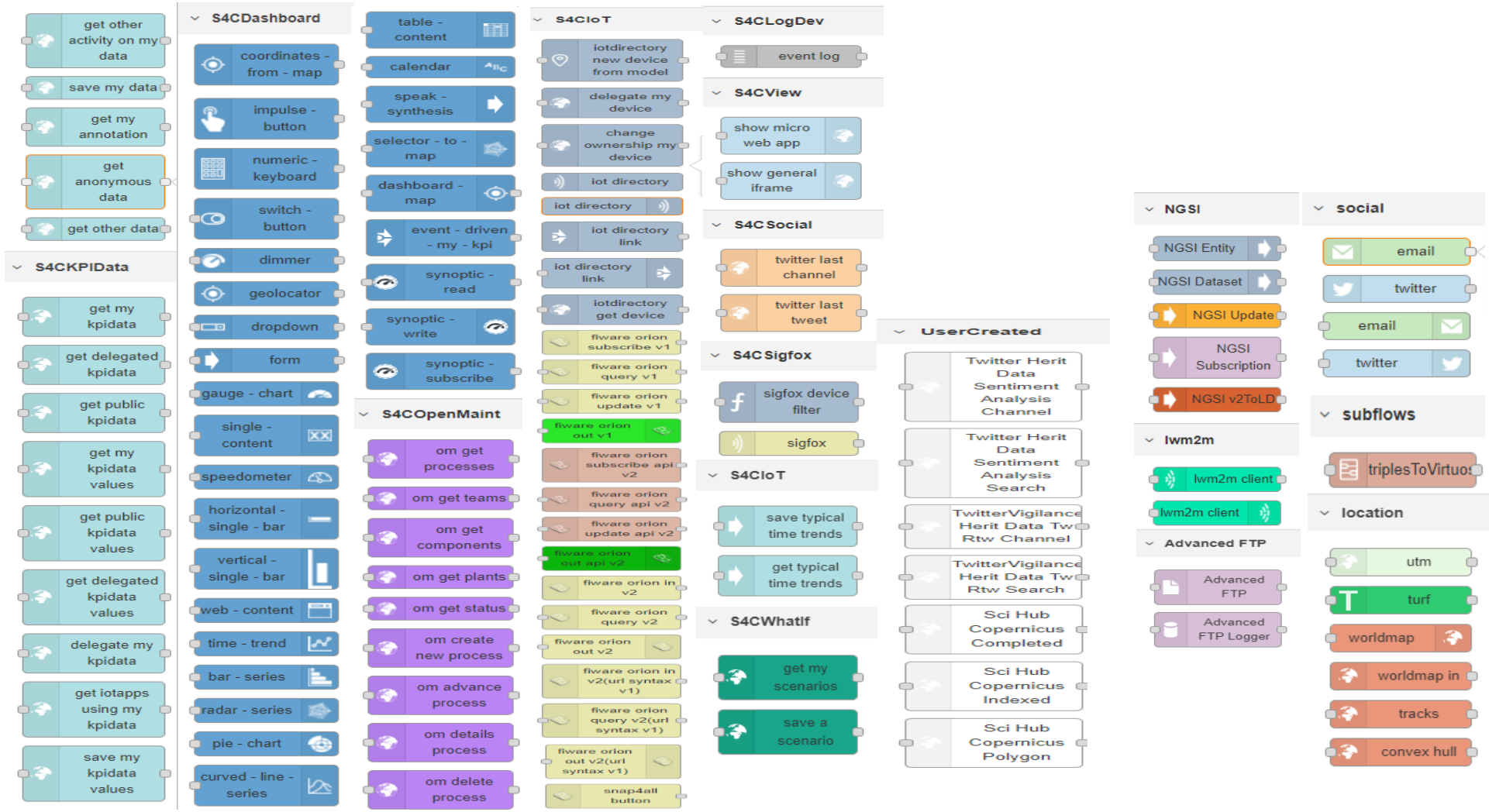
Nodes


common
inject
debug
complete
catch
status
link in
link out
comment
location
turf
worldmap
worldmap in
tracks
convex hull
input
amqp in
amqp2 in
stomp in
output
amqp out
amqp2 out
stomp out
function
function
switch
change
range
template
delay
trigger
exec
zip
md5
soap request
string
xml converter
random
rbe
network
mqtt in
mqtt out
http in
http response
http request
websocket in
websocket out
tcp in
tcp out
tcp request
udp in
udp out
social
email
twitter in
email
twitter out
advanced
feedparser
NGSI
NGSI Entity
NGSI v2ToLD
lwm2m
lwm2m client in
lwm2m client out
dashboard
button
dropdown
switch
slider
numeric
text input
date picker
colour picker
form
text
gauge
chart
audio out
notification
ui control
template
sequence
split
join
sort
batch
parser
csv
html
json
xml
yaml
base64
msgpack
storage
file
file in
watch
ftp in
mysql
tail
time
sunrise

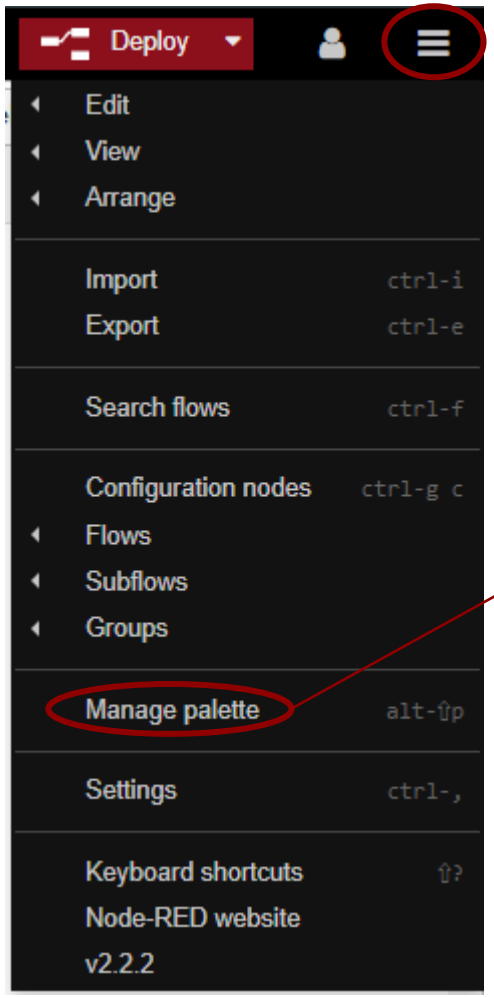
Nodes – Snap4City libraries

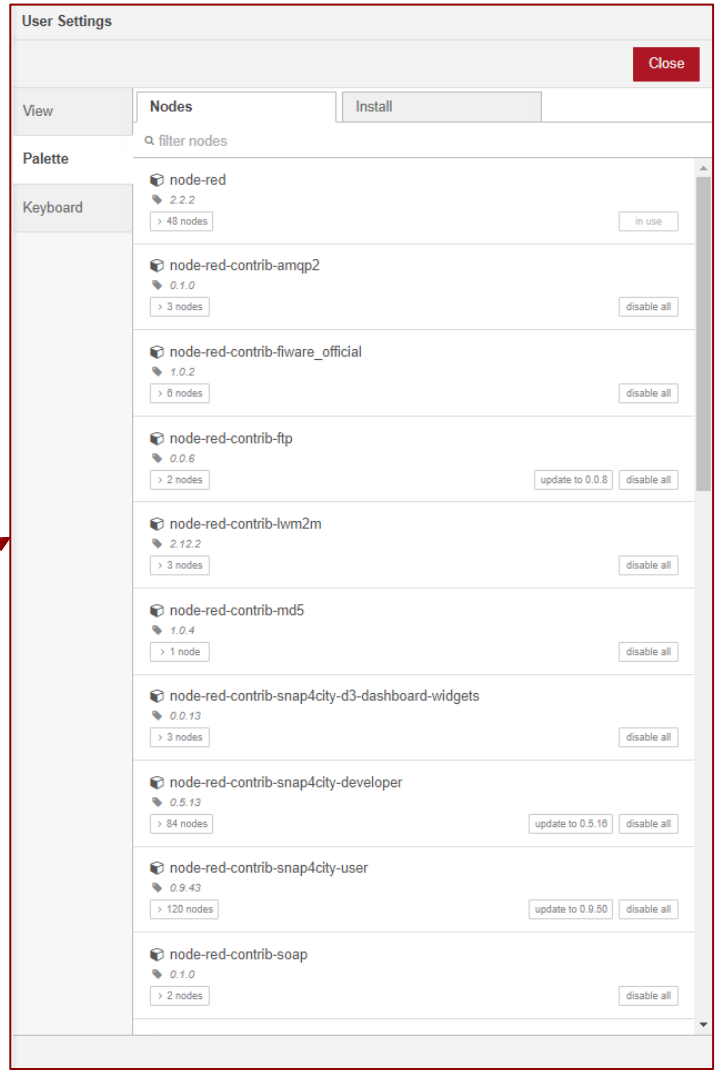


Nodes – Snap4City libraries

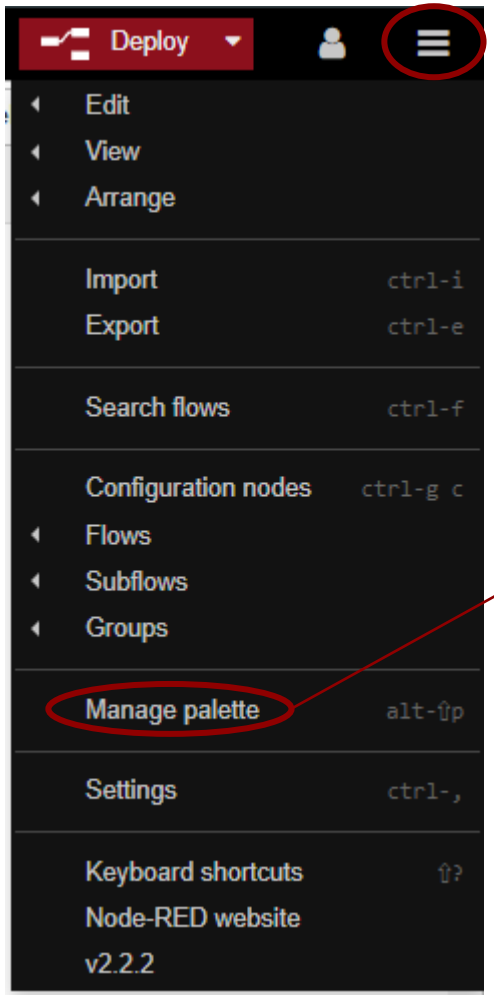


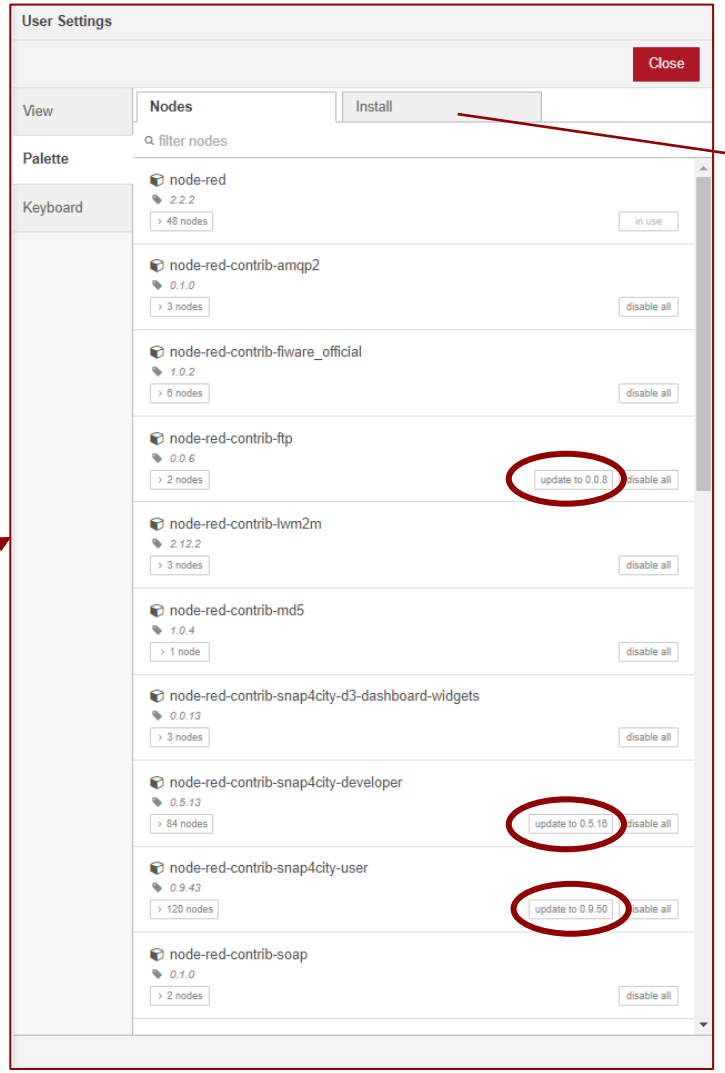
Manage palette

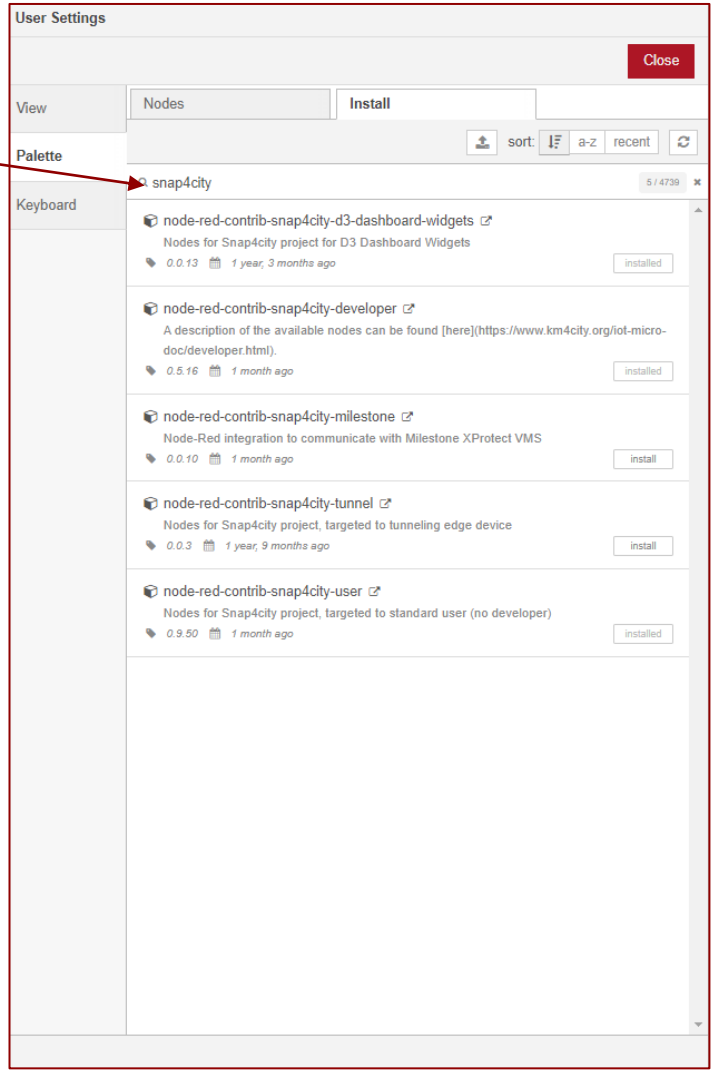




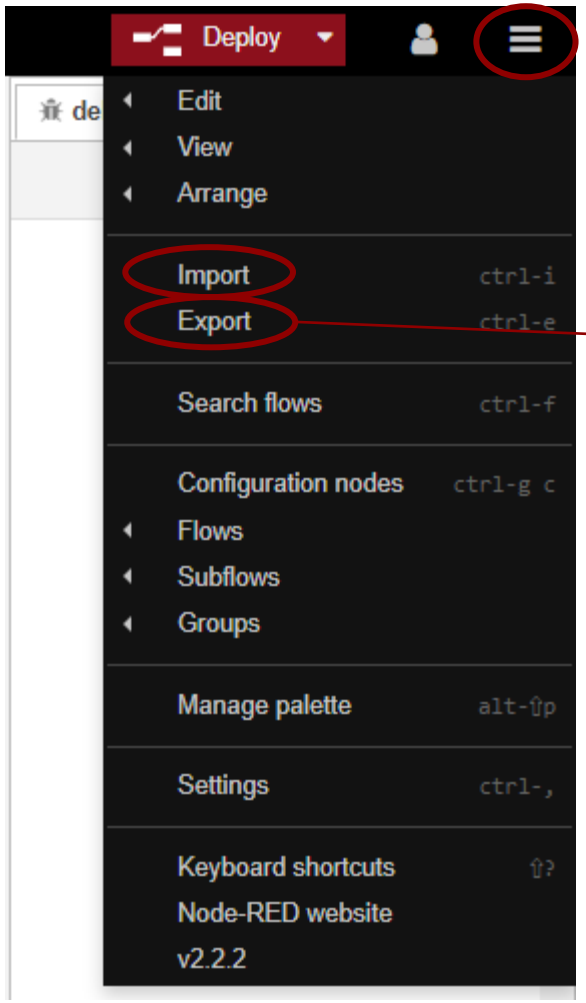
Manage palette – update & install

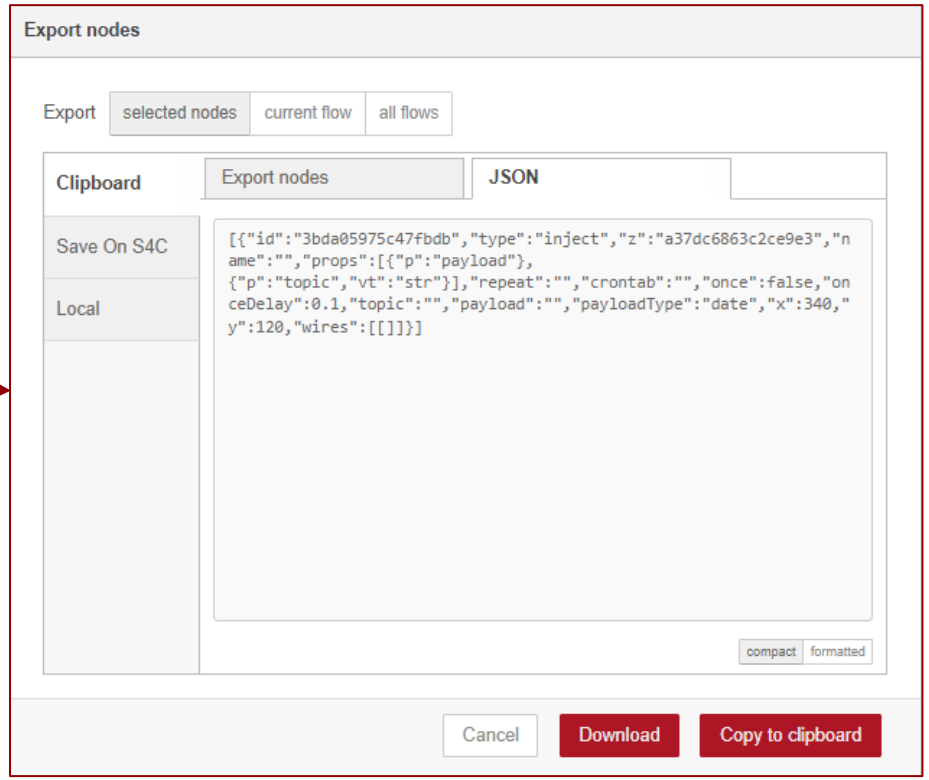




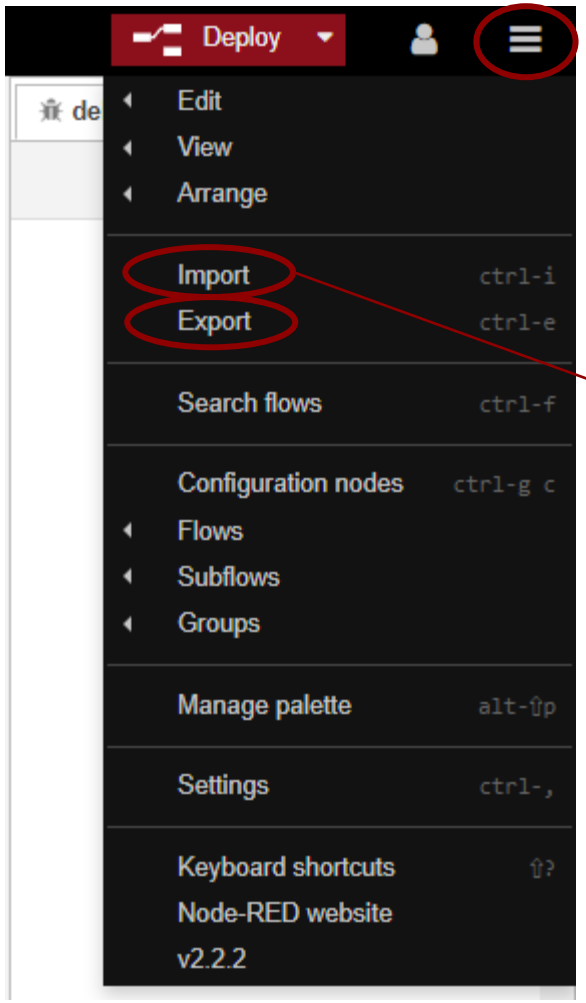


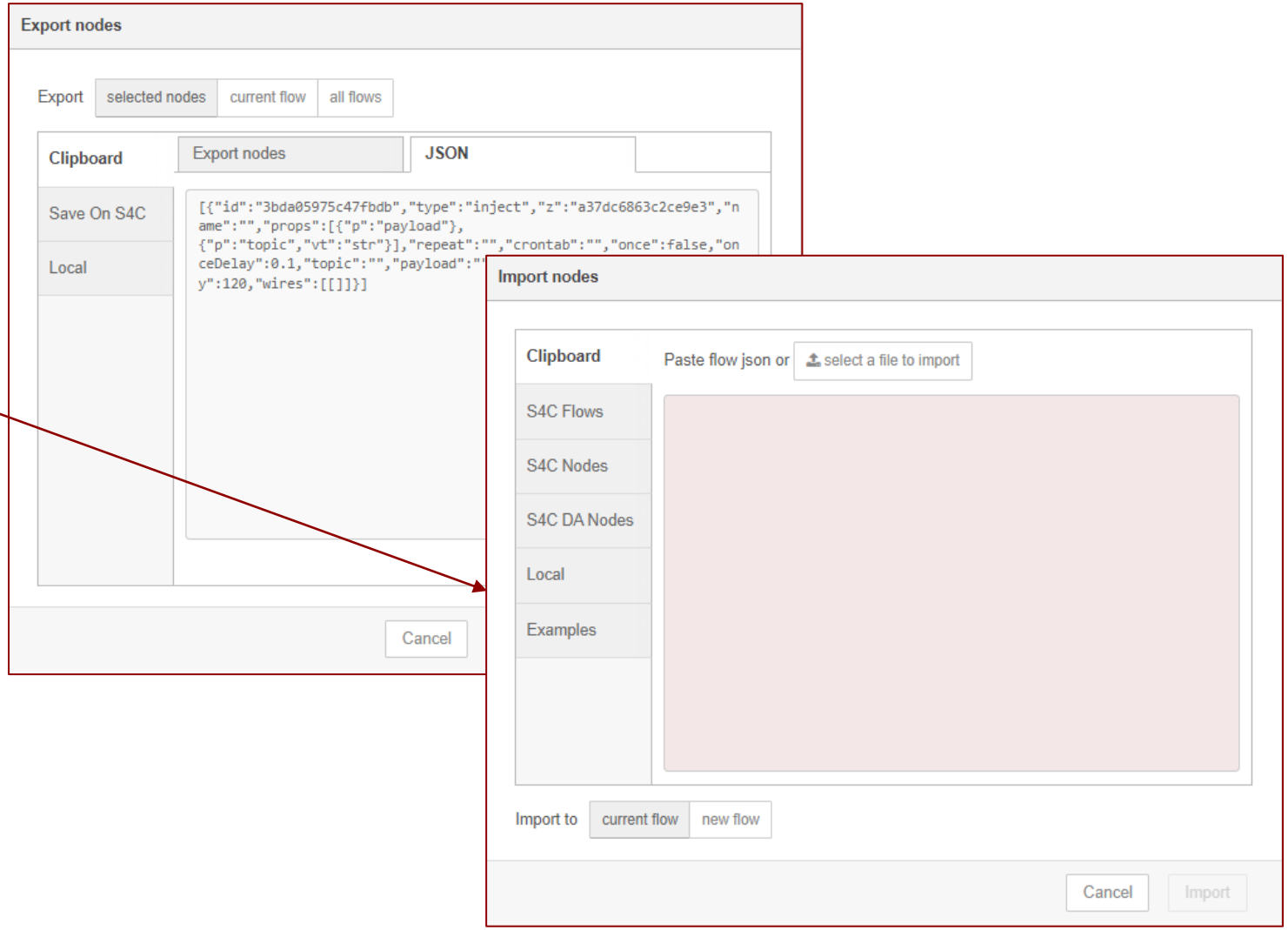
Import & export flows





Import & export flows





Working with context

- Node-RED provides a way to store information that can be **shared between different nodes without using the messages** that pass through a flow. This is called **context**
- There are three different context scope:
  - Node - only visible to the node that set the value
  - Flow - visible to all nodes on the same flow (or tab in the editor)
  - Global - visible to all nodes
- For example, in the function node, we can use
  - `flow.set("count", 123);` - to set the context name “count”
  - `var myCount = flow.get("count");` - to read the context