* 1. *Modifiche Interfaccia Node-RED*

Le modifiche all’interfaccia di Node-RED consentono di importare nelle iotapp i nodi creati all’interno del Resource Manager ( normali e di Data Analytics). Consentono anche di importare ed esportare nel Resource Manager i flussi di un utente in modo da pubblicarlo e renderlo disponibile per altri utenti.

Immagine che contiene schermata, testo, software

Descrizione generata automaticamente

Immagine che contiene testo, schermata, numero, software

Descrizione generata automaticamente

per farlo nella nuova versione di Node-RED viene aggiungo un modulo @Node-RED ed è lì che si devono modificare i file necessari. Attualmente la versione di Node-RED è la 1.1.3 e nel caso in cui si debba cambiare versione forse è meglio creare dei nuovi container senza modificare quelli descritti in precedenza: quelli con v1.1.3.

Le modifiche da effettuare sia nella versione 0.17.5 che nella 1.1.3 sono riportate qui:

<https://192.168.0.252/newrepos/trunk/Applications/sii-mobility/nodered-snap4city-api/importMicroServicesNodered/>

all’interno della cartella 1.1.3 ci sono anche due file readme che spiegano come creare un nuovo container per inserire una nuova versione. Lo riporto qui per comodità:

*A differenza della versione 0.17.5 questa va sostituito alla cartella @Node-RED che si trova dentro node-modules.*

*1. Creare cartella con il nome che si vuole dare al container all'interno della cartella /root/nr-datadirs/ e dargli i permessi corretti*

*mkdir /root/nr-datadirs/nodered-v1.1.3-basic*

*chmod a+w /root/nr-datadirs/nodered-v1.1.3-basic*

*chown -R 1001.1001 /root/nr-datadirs/nodered-v1.1.3-basic*

*2. Copiare nella cartella il file /mnt/data/nr-data/settings.js.tpl e modificarlo in modo tale che al posto di \_\_NRID\_\_ ci sia il nome del container*

*cp /mnt/data/nr-data/settings.js.tpl /root/nr-datadirs/nodered-v1.1.3-basic/settings.js*

*\_\_NRID\_\_ -> nodered-v1.1.3-basic*

*3. Creazione del container della nuova nr-basic a partire dall'immagine snap4city-nodered-basic:v90*

*docker run --publish 1908:1880 --detach --name nodered-v1.1.3-basic -v /root/nr-datadirs/nodered-v1.1.3-basic:/data snap4city-nodered-basic:v90*

*4. Aggiungere nella cartella /root/nr-proxy il file*

*nodered-v1.1.3-basic.conf*

*contenente*

*location /nodered/nodered-v1.1.3-basic {*

*proxy\_set\_header Host $http\_host;*

*proxy\_set\_header X-Real-IP $remote\_addr;*

*proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;*

*proxy\_set\_header X-Forwarded-Proto $scheme;*

*proxy\_http\_version 1.1;*

*proxy\_set\_header Upgrade $http\_upgrade;*

*proxy\_set\_header Connection "upgrade";*

*proxy\_pass "http://192.168.1.82:1908";*

*}*

*5. Rilanciare nginx*

*service nginx reload*

*6. Aggiungere la iotapp alla tabella ownership*

*'20906', 'roottooladmin1', 'nodered-v1.1.3-basic', 'AppID', 'IOTApp Basic based on Nodered v 1.1.3', 'https://iot-app.snap4city.org/nodered/nodered-v1.1.3-basic', '{\"type\":\"basic\",\"image\":\"snap4city-nodered-basic:v90\",\"iotappids\":[]}', NULL, '2020-09-17 12:17:19', NULL, NULL*

*7. Una volta aggiornato NPM*

*docker cp /home/debian/snap4city-user-authentication nodered-v1.1.3-basic:/usr/src/node-red/node\_modules*

*docker cp /home/debian/importMicroServicesNodered/1.1.3/\@node-red nodered-v1.1.3-basic:/usr/src/node-red/node\_modules*

*8.Per salvare e fare una nuova immagine*

*docker commit nodered-v1.1.3-basic snap4city-nodered-v1.1.3-basic:v1*

*docker save snap4city-nodered-v1.1.3-basic:v1 | gzip > /mnt/data/imgs/snap4city-nodered-v1.1.3-basic-v1.tgz*

Nel caso in cui di debba modificare il codice che è stato inserito dentro la cartella @Node-RED conviene importare tale cartella all’interno di un IDE e cercare la stringa //S4C in quanto il codice modificato o integrato viene racchiuso fra commenti che iniziano con questa stringa.

Inoltre i file che devono essere modificati dovrebbero essere questi:

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

* 1. *Aggiornamento Node-RED 3.0.2*

Le operazioni necessarie per aggiornare Snap4City alla versione Node-RED 3.0.2 si limitano a trasferire tutte le sezioni di codice comprese tra i commenti //S4C nella versione ufficale di Node-RED 3.0.2, di fatti, questo si limita a coinvolgere esclusivamente la cartella **packages/node-modules/@node-red**.

* 1. *Integrazione Progetto CAULDRON*

Tutte le modifiche apportate dal progetto CAULDRON, anche loro limitate alla cartella @node-red, sono comprese tra commenti // CAULDRON. La logica principale del progetto è contenuta nei file all’interno della cartella **packages\node\_modules\@node-red\editor-client\src\js\monitoring**. Per una documentazione più completa del progetto, leggere la documentazione ufficiale al seguente link <https://repositorio-aberto.up.pt/bitstream/10216/128964/2/414763.pdf>.

Tale documentazione non include però indicazioni sulle modalità di utilizzo delle funzioni di debug. In particolare l’attuale versione di Cauldron basata su node-red 3.0.2 permette di selezionare attraverso una apposita textbox all’interno di ogni nodo l’attributo dell’oggetto che attraverso il nodo. La sintasi ammessa è quella usuale del Javascript quindi per un oggetto del tipo:

{

A:[

{

B:’0b’

},

{

B:”1b”

}

]

}

Digitare nella textbox l’espessione A[0].B mostra il valore ‘0b’ mentre l’espressione

A[1].B mostra il valore ‘1b’. Se l’espressione non corrisponde a nessun attributo dell’oggetto non vien mostrato alcun risultato.

Tale funzionalità può essere usata in modalità statica(‘static’) o dinamica(‘dynamic’): per modalità statica si intende che il cambio della textbox ha effetto dal prossimo messaggio che attraversa il nodo, mentre per modalità dinamica si intende che l’aggiornamento della textbox comporta un aggiornamento istantaneo del valore visualizzato a schermo.