

# MicroX on premise Troubleshooting

MicroX: <https://www.snap4city.org/738>

To get updated version of the slides go to <https://www.snap4city.org/577>

From Snap4City:

- We suggest you read the TECHNICAL OVERVIEW:
  - <https://www.snap4city.org/download/video/Snap4City-PlatformOverview.pdf>
- **Development:** <https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf>
- **See Client-Side Business Logic Widget Manual:** <https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf>
- <https://www.snap4city.org>
- <https://www.snap4solutions.org>
- <https://www.snap4industry.org>
- <https://twitter.com/snap4city>
- <https://www.facebook.com/snap4city>
- <https://www.youtube.com/channel/UC3tAO09EbNba8f2-u4vandg>

**Coordinator:** Paolo Nesi, [Paolo.nesi@unifi.it](mailto:Paolo.nesi@unifi.it)

DISIT Lab, <https://www.disit.org>

DINFO dept of University of Florence,

Via S. Marta 3, 50139, Firenze, Italy

Phone: +39-335-5668674

**Access Level:** public

**Date:** 25-09-2023

**Version:** 1.0



UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB

## Sommario

Installare docker e docker-compose .....	4
Creazione MicroX e setup on premise.....	4
Primi passi.....	4
Deploy su dominio cliente .....	5
Attivare HTTPS.....	5
Errore in keycloak .....	8
IoT App non viene caricata .....	9
Sottoscrizione Orion/Nifi persa .....	9
Errori widget/wsserver .....	9
Cambio password utenti.....	10
Chiudere tutte le porte.....	10
Redirect base URL to URL/dashboardSmartCity .....	11
Adding a nodered app .....	11
Possibile errore a seguito di aggiunta di IoTApp .....	12
Cambiare ownership di IoTApp .....	12
Modifica visualizzazione delle voci del menu.....	12
Modifica nome portale (i.e. Snap4Asymmetrica).....	12
Modifica footer e logo Dashboard .....	13
Modifica limite row/col dei widget .....	13
Dashboard KIBANA .....	13
Set SMTP configuration .....	13
Clonare modelli .....	14
Clonare static attributes .....	14
Mappare cartella locale con docker-compose in un volume dentro ad un container .....	14
Possibili errori e soluzioni.....	16
Tutto quieto... docker-compose ps tutto su ma poi accedi dal browser e schermata bianca senza errori su console.....	17
Delega di Orion al gruppo.....	17
Nessuna IOT app della microx è accessibile dal sottomenu.....	17
Orion non riceve dati da nodeRED (1).....	18
Orion non riceve dati da nodeRED (2).....	18
Orion va in stallo (a seguito di un crash di sistema).....	18
cambi psw di root e nifi va aggiornato .....	19
cambi psw di root e il device non c'è nel data-inspector .....	20
I dati si bloccano su NiFi in enrich-data .....	20

NiFi non salva su opensearch/elasticserach.....	21
Il device non c'è nella service map .....	21
Il device non appare in data-inspector.....	21
Cambiare il timeout della sessione.....	23
Ripulire i token di una iot app .....	23
Impostare il tunnelling per l'accesso all'interfaccia di NIFI.....	24
Lista delle sottoscrizioni del broker.....	24
Abilitare un utente a CSBL.....	24
Aumentare il default refresh time della Micro X.....	24
Aggiornare il dictionary dopo aver aggiunto value type nuovi .....	24
Se api iot-search non funziona .....	24
Autenticazione vuota su flusso Node-RED .....	25
Ripulire il db dai dati di esempio presenti nelle MicroX.....	26
Flusso IOTApp.....	26

## Installare docker e docker-compose

- Docker: <https://docs.docker.com/engine/install/debian/>
- Docker-compose: <https://docs.docker.com/compose/install/>

## Creazione MicroX e setup on premise

Dalla pagina [https://www.snap4city.org/docker-generator/selecting\\_model](https://www.snap4city.org/docker-generator/selecting_model) creare la propria versione richiesta:

- IP deve essere l'IP della macchina (VM) su cui è in esecuzione Docker: su Unix verificare IP con `ifconfig`
- Quindi scaricare i file di installazione, decomprimere e seguire la guida

Installazione (<https://www.snap4city.org/drupal/node/738>):

- scompatta l'archivio
- naviga fino all'archivio, poi esegui con `sudo` (o meglio su `-`)
- quindi esegui:
  - `./setup.sh`
  - `docker-compose up -d`
  - `./post-setup.sh`

Alla pagina <http://dashboard/dashboardSmartCity> troviamo la homepage. Nota: l'IP della VM deve essere aggiunto nel file `/etc/hosts` (o `C:\Windows\System32\drivers\etc\hosts` per Windows) del client.

*In `post-setup.sh`, linea #4, modificare in `./update-ontology.sh localhost`*

## Primi passi

- eseguire l'accesso come `rootadmin`
- **[IMPORTANTE]** prendere possesso del broker IoT cliccando su `edit` e confermare `<<<<`
- delegare al gruppo il broker
- eseguire l'accesso come `areamanager`
- definire un device model indicando il broker delegato,
  - impostare `dateObserved` (timestamp, timestamp in ms, string) per permettere la lettura come time series
  - Consiglio (en) mettere anche un dato tipo io ho messo `lateemperature` a (temperature, Celsius, float)
- costruire un IoT device, usando il modello definito
- da `nodered`, creare un flusso per inviare dati al device, impostando la connessione con il broker, le chiavi e definendo i campi necessari nel messaggio
  - **[IMPORTANTE]** `dateObserved` deve avere
    - `type = string,`
    - `value = new Date().toISOString()`

(en) -> per mandare i dati consiglio di andare su `IoTDevices` e premere il tasto + per poi prendere il payload ngsi v2 cliccarlo e si apre il contenuto da mettere nella function di `nodered` per poi mandare i dati al broker

Ps metto il json del flusso in fondo a questo doc

!!!!!! ricordati di mettere la subnature

- creare una dashboard per leggere i dati del dispositivo IoT
- possibilità di creare anche un KPI

- **orion broker a volte rimane in uno stato di continuo riavvio.**
    - **fermare, rimuovere e riavviare manualmente il container orion**
      - `docker-compose stop orion`
      - `docker-compose rm orion`
      - `docker-compose up -d orion`
- riportando in corretta esecuzione orion.*
- Per arrivare alla corretta creazione di device model e iot device (il broker iotobsf creato non appariva nelle schermate per la creazione dei device) è stato necessario riavviare i singoli container.*
- *A volte il metodo di creazione delle dashboard non permetteva di selezionare gli iot device creati. Il problema può essere legato al container **dashboard-cron**: gli script di controllo per la ricerca di nuovi device erano rimasti bloccati a causa del file **running2.txt** non correttamente rimosso. Rimuovendo manualmente il file gli script sono andati in esecuzione e i device trovati.*

## Deploy su dominio cliente

Operazioni eseguite per Asymmetrica

Il primo passo è verificare il nome del dominio associato al IP del server su cui eseguire l'installazione. In caso di errore (dominio errato, e.g. [esempio.com](http://esempio.com) anziché [www.esempio.com](http://www.esempio.com)) si riscontreranno problemi per accedere alla piattaforma.

*Homepage link: <http://<nome-dominio>/dashboardSmartCity>*

*Durante la creazione della MicroX è utile modificare tutte le password dei servizi.*

*Nota che, modificando la password per rootadmin, questa viene modificata nella configurazione del **dashboardbuilder** che usa per prendere i dati dei sensori privati ma non in LDAP: quindi l'accesso al portale avviene sempre con la password di default, ma non essendo uguale alla password usata per dashboardbuilder, questo può causare errori di comunicazione sui sensori/dispositivi.*

*Soluzioni:*

- *modificare password per rootadmin in LDAP (vedi dopo)*
- *ripristinare la password di default per rootadmin in `dashboard-builder-con/personaldata.ini` e riavviare **dashboard-cron** (ed eventualmente **proxy** per gestire il nuovo IP interno del container)*

## Attivare HTTPS

Nel generatore MicroX, selezionare `https` nel campo `##base-protocol##`

Quindi seguire la guida per installare certbot/letsencrypt e attivare https su proxy nginx

- [Nginx and Let's Encrypt with Docker in Less Than 5 Minutes | by Philipp | Medium](#)
- [How to handle HTTPS using Nginx, Let's encrypt and Docker - Mindsers Blog](#)

*A grandi linee per attivare https, oltre a aggiungere certbot al compose, è necessario modificare `nginx.conf`. La parte meno intuitiva è la necessità di creare dei certificati fittizi per permettere l'avvio di nginx, quindi rimuoverli e fare la richiesta a `letsencrypt` per ottenere i certificati veri.*

Nel *docker-compose.yml* aggiungere certbot e indicare i volumi condivisi tra certbot e nginx per conservare i certificati

```
proxy:
  image: nginx
  logging:
    driver: json-file
    options:
      max-file: '10'
      max-size: 100m
  networks:
    default:
      aliases:
        - www.cityconn.cloud
  ports:
    - "80:80"
    - "443:443"
  restart: unless-stopped
  command: "/bin/sh -c 'while ;; do sleep 6h & wait $$!}; nginx -s reload; done & nginx -g \"daemon off;\""
  volumes:
    - ./nginx-proxy-conf:/etc/nginx/conf.d:rw
    - ./certbot/conf:/etc/nginx/ssl:ro
    - ./certbot/www:/var/www/certbot/:ro
certbot:
  image: certbot/certbot
  entrypoint: "/bin/sh -c 'trap exit TERM; while ;; do certbot renew; sleep 12h & wait $$!}; done;'"
  volumes:
    - ./certbot/conf:/etc/letsencrypt/:rw
    - ./certbot/www:/var/www/certbot/:rw
```

In *nginx.conf* aggiungere la configurazione per https (porta 443) e modificare quella per http (porta 80)

```
server {
    listen 80;
    server_name www.cityconn.cloud;

    location /.well-known/acme-challenge/ {
        root /var/www/certbot;
    }
    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name www.cityconn.cloud;

    ssl_certificate /etc/nginx/ssl/live/www.cityconn.cloud/fullchain.pem;
    ssl_certificate_key /etc/nginx/ssl/live/www.cityconn.cloud/privkey.pem;

    root /var/www/html;

    index index.html index.htm index.nginx-debian.html;

    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_read_timeout 10m;

    location = / {
        rewrite ^ https://www.cityconn.cloud/dashboardSmartCity/ redirect;
    }
    location / {
        proxy_pass "http://dashboard-builder/";
    }
    location /phpldadmin/ {
        proxy_pass "https://myldap:443/";
    }
    location /ServiceMap/api/v1/iot/ {
        proxy_pass "http://servicemap:8080/iot/";
    }
}
```

```

}
location /ServiceMap/ {
    proxy_pass "http://servicemap:8080/ServiceMap/";
}
location /wsserver {
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";

    proxy_pass "http://wsserver/wsserver";
}
location /superservicemap/ {
    proxy_pass "http://servicemap:8080/superservicemap/rest/";
}
location /auth/ {
    proxy_pass "http://keycloak:8088/auth/";
}
location /datamanager/ {
    proxy_pass "http://personaldata:8080/datamanager/";
}
location /kibana/ {
    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_pass "http://opensearch-dashboards:5601/";
}
location /synoptics/ {
    proxy_pass "http://synoptics:3002/";
}
location /synoptics/socket.io/ {
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_pass "https://synoptics:3001/socket.io/";
}

location /iotapp/iotapp-001/ {
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";

    proxy_pass "http://iotapp-001:1880/iotapp/iotapp-001/";
}
location /iotapp/iotapp-002/ {
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";

    proxy_pass "http://iotapp-002:1880/iotapp/iotapp-002/";
}
}

```

*Nota: i proxy\_pass rimangono tutti invariati in http/https*

Quindi

- creare i certificati fittizi

```

docker-compose run --rm --entrypoint "\
openssl req -x509 -nodes -newkey rsa:4096 -days 1\
-keyout '/etc/letsencrypt/live/www.cityconn.cloud/privkey.pem' \
-out '/etc/letsencrypt/live/www.cityconn.cloud/fullchain.pem' \
-subj '/CN=localhost'" certbot

```

- avviare il compose (soprattutto il proxy)

```
docker-compose restart proxy
```

- cancellare i certificati fittizi

```
docker-compose run --rm --entrypoint "\
rm -Rf /etc/letsencrypt/live/www.cityconn.cloud && \
rm -Rf /etc/letsencrypt/archive/www.cityconn.cloud && \
rm -Rf /etc/letsencrypt/renewal/www.cityconn.cloud.conf" certbot
```

- fare la richiesta per i certificati reali

```
docker-compose run --rm certbot certonly --webroot --webroot-path
/var/www/certbot/ -d www.cityconn.cloud
```

## Errore in keycloak

*Riceve errore: Invalid parameter: redirect\_uri*

Aggiungere al compose per keycloak in environment:

- KEYCLOAK\_FRONTEND\_URL: https://<nome\_dominio>/auth/
- PROXY\_ADDRESS\_FORWARDING: "true"

Come nell'esempio qua sotto

```
keycloak:
  command:
    - -Djboss.socket.binding.port-offset=8
  depends_on:
    ldap-server:
      condition: service_started
  environment:
    KEYCLOAK_PASSWORD: sRqdREu20xBTJzdN
    KEYCLOAK_USER: admin
    KEYCLOAK_FRONTEND_URL: https://www.cityconn.cloud/auth/
    PROXY_ADDRESS_FORWARDING: "true"
  image: disitlab/preconf-keycloak:v4
  logging:
    driver: json-file
    options:
      max-file: '10'
      max-size: 100m
  # ports:
  # - published: 8088
  #   target: 8088
  restart: unless-stopped
  volumes:
    - keycloak:/opt/jboss/keycloak/standalone:rw
```

Quindi ricreare il container con `docker-compose up -d keycloak` e eventualmente riavviare il proxy.



### IoT App non viene caricata

È un problema del link nel DB nella tabella dei menu dove per le IoT App è specificato http e non https.

Per correggere, o si accede al DB e si modifica la voce, o da interfaccia di amministrazione si identifica la voce del menu

		nodered2		nodered2					
10230		IoT Application nodered 001	http://www.cityconn.cloud/iotapp/iotapp-001/	IoT Application nodered 001	EDIT DELETE		iotapp-001	MainMenuSubmenu	1035

quindi con EDIT si modifica http in https



### Edit menu element

Link Url:

### Sottoscrizione Orion/Nifi persa

Se su orion i dati arrivano, ma il device non appare in data inspector, va vista la subscription verso nifi provando

```
curl http://localhost:1026/v2/subscriptions
```

In caso di array vuoto in risposta, la registrazione del broker su nifi è andata persa. Per ripristinarla

- login come rootadmin
- vai in edit del broker
- modifica leggermente la url di subscription e poi rimodificala rimettendola come in originale

Eseguendo di nuovo curl si dovrebbe ottenere un array tipo

```
[{"id":"62d5112d75ae1864255a642d","description":"orion-1 nifi","status":"active","subject":{"entities":[{"idPattern":".*","typePattern":".*"}],"condition":{"attrs":[]},"notification":{"attrs":[],"onlyChangedAttrs":false,"attrsFormat":"normalized","http":{"url":"http://nifi:1030/ingestngsi"}}}]}
```

### Errori widget/wsserver

Se in HTTPS, modifica il file `dashboard-builder-conf/webSocketServer.ini` cambiando le opzioni

- porta su 443 e
- protocollo su wss

In `iotapp-00X/settings.js`, modifica

```
wsServerUrl: 'ws://www.cityconn.cloud/wsserver'
```

in

```
wsServerUrl: 'wss://www.cityconn.cloud/wsserver'
```

In *nginx.conf* modifica

```
location /wsserver {
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";

    proxy_pass "http://wsserver/wsserver";
}
```

In

```
location /wsserver {
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";

    proxy_pass "http://wsserver:443/wsserver";
}
```

## Cambio password utenti

La modifica delle password degli utenti si fa da **ldap**, usando **phpldapadmin** raggiungibile all'URL

<https://www.cityconn.cloud:6443/phpldapadmin>

L'accesso a phpldapadmin può essere bloccato a causa di certificato https non validi: è sufficiente accettare il certificato non valido e l'accesso da browser dovrebbe funzionare.

Oltre agli utenti `userrootadmin`, `userareamanager`, ecc., è possibile modificare anche la password dell'admin di keycloak. calma spieghiamo meglio

cambio psw admin keycloak andare su indirizzoinstallazione/auth/admin

per il cambio password utenti andare sul front-end di snap4

menu a sinistra

user management and auditing

Dashboard builder local users

**[NOTA NIC: cambiare la psw di root provoca delle inconsistenze che vanno documentate]**

## Chiudere tutte le porte

Di default, tutti i container del compose hanno un port mapping che mappa le porte della rete interna Docker su porte accessibili dall'esterno. Benché utile in fase di deploy, una volta in produzione tutte le porte devono essere chiuse per modifiche di sicurezza. L'unico container che deve essere accessibile dall'esterno è il proxy per ovvie ragioni.

Per chiudere le porte è sufficiente commentare/rimuovere i port mapping dal file compose e ripristinare i container con `docker-compose up -d`.

A seguito della chiusura delle porte, il sito potrebbe risultare irraggiungibile. In tal caso provare a riavviare prima `dashboard-builder`, poi il proxy per aggiornare gli IP interni associati ai container ricreati.

## Redirect base URL to URL/dashboardSmartCity

In `nginx.conf` aggiungere

```
location = / {
    rewrite ^ https://www.cityconn.cloud/dashboardSmartCity/ redirect;
}
```

## Adding a nodered app

In this configuration to add a new nodered app `nr3 003` you need to:

1. change the `docker-compose.yml` file and copy the rows defining `nodered nr2` `iotapp-001` and change any reference to `nr2 001` into `nr3 002`
2. copy the folder `iotapp-001nr2` into `iotapp-002nr3` and check the Unix permission to files and directories, and files and directories ownership
3. edit the file `iotapp-002nr3/settings.js` and replace any reference to `001nr2` to `002nr3`

```
- line 19: var myappid='iotapp-002'
- line 131: httpRoot: '/iotapp/iotapp-002',
```
4. edit the file `apache-proxy.conf` `nginx-proxy-conf/nginx.conf` and copy the rows related with `001nr2` to `002nr3`
5. connect to the db on port 3306 user/passwordx and
6. add a new row in `profiledb.ownership` with the following SQL command

```
INSERT INTO
profiledb.ownership(username,elementid,elementtype,elementname,elementurl,
elementdetails,created) values('usermanager','nr3','AppID','nodered
3','http://dashboard/iotapp/nr3/', '{"edgeway_type":"linux_Linux_4.9.0-
8-amd64"}',now());
```

```
INSERT INTO
profiledb.ownership(username,elementId,elementType,elementName,elementUrl,
elementDetails,created) values('userareamanager','iotapp-
002','AppID','nodered
2','http://www.cityconn.cloud/iotapp/002/', '{"edgeway_type":"linux_Lin
ux_4.9.0-8-amd64"}',now());
```

7. perform the following SQL command to add a menu option to easily access to the app:

```
INSERT INTO
Dashboard.MainMenuSubmenus(menu,linkUrl,linkid,icon,text,privileges,userTy
pe,externalApp,openMode,iconColor,pageTitle,menuorder,organizations)
VALUES (1035,'http://dashboard/iotapp/nr3/', 'iotappnr3', 'fa fa-file-code-
o', 'IoT Application nodered3', ['\RootAdmin\',
```

```
\'Manager\']','any','yes','iframe','#FFFFFF','IoT Application nodered3',
3, '[\'Organization\','DISIT\','Other\']');
```

```
INSERT INTO
Dashboard.MainMenuSubmenus(menu,linkUrl,linkid,icon,text,privileges,userType,
externalApp,openMode,iconColor,pageTitle,menuorder,organizations)
VALUES (1035,'https://www.cityconn.cloud/iotapp/iotapp-002/','iotapp-
002','fa fa-file-code-o','IoT Application nodered 002','[\'RootAdmin\','
\'AreaManager\']','any','yes','iframe','#FFFFFF','IoT Application nodered
002', 1, '[\'Organization\','DISIT\','Other\']');
```

8. issue command `docker-compose up -d` to bring the new container up
9. restart the dashboard container and the proxy:
10. `docker-compose restart dashboard-builder`
11. `docker-compose restart proxy`

If all it is ok, you should see from the menu of ~~'usermanager'~~ 'userareamanager' the new application and you can connect to it.

**Note:** if you delete the dashboarddb volume you will lose the changes made on the db, to make them more permanent you may change the database/profiledb.sql file and database/dashboard-menu.sql adding the two SQL instructions.

**Note:** Please consider that login on a nodered app is allowed to the owner (as stated on the ownership table) and to any user with RootAdmin role.

#### Possibile errore a seguito di aggiunta di IoTApp

Sia la iotapp 001, sia la nuova 002, non riescono a scrivere su orion riportando il seguente errore:

```
failed to update, reason: {"statusCode":401,"headers":{"x-content-type-options":"nosniff","x-xss-
protection":"1; mode=block","cache-control":"no-cache, no-store, max-age=0, must-
revalidate","pragma":"no-cache","expires":"0","strict-transport-security":"max-age=31536000 ;
includeSubDomains","x-frame-options":"DENY","transfer-encoding":"chunked","date":"Fri, 22 Jul 2022
08:42:35 GMT","connection":"close"},"payload":{"\result\":false,\message\":\\"The passed access
token is not valid\\"}}
```

**Soluzione:** riavviare orionbrokerfilter.

**Nota:** riavviare tutto il compose potrebbe non funzionare, in quanto se orionbrokerfilter si avvia prima che keycloak sia pronto non riesce a recuperare le chiavi per le ownership.

#### Cambiare ownership di IoTApp

Necessario modificare la ownership nel db **profiledb.ownership** per la iotapp di interesse, quindi rieffettuare l'accesso alla IoTApp con il nuovo proprietario per fare il refresh dell'accesstoken

#### Modifica visualizzazione delle voci del menu

È possibile modificare quali voci del menu visualizzare per quali utenti andando a modificare i valori della colonna *privileges* nelle tabelle *Dashboard.MainMenu* e *Dashboard.MainMenuSubmenus*.

#### Modifica nome portale (i.e. Snap4Asimmetrica)

Nel DB modifica nella tabella *Dashboard.Domains* la colonna 'claim'

## Modifica footer e logo Dashboard

In `dashboard-builder`, nella cartella `/var/www/html/dashboardSmartCity/view` modificare il file `index.php`.  
Per garantire persistenza in caso di ricostruzione del container:

- copiare la cartella da container al host  
`docker cp <container_ID>:/var/www/html/dashboardSmartCity/view/* ./dashboard-builder-view/.`
- Aggiungere nel compose il mount della cartella copiata  
`- ./dashboard-builder-view:/var/www/html/dashboardSmartCity/view:rw`
- Fare up del container e quindi modificare il file

## Modifica limite row/col dei widget

Cambia `max_row` e `max_col` nella tabella `Dashboard.Widgets`

## Dashboard KIBANA

In `postsetup.sh` modifica

```
curl -u admin:f89Ux63JkcyN9ofh -XPOST
"http://localhost/kibana/api/saved_objects/_import?overwrite=true" -H "osd-xsrf:
true" -H "securitytenant: global" --form file=@osd-dashboard.ndjson
```

in

```
curl -u admin:f89Ux63JkcyN9ofh -XPOST
"https://www.cityconn.cloud/kibana/api/saved_objects/_import?overwrite=true" -H
"osd-xsrf: true" -H "securitytenant: global" --form file=@osd-dashboard.ndjson
```

## Set SMTP configuration

In `servicemap-conf/servicemap.properties` imposta I seguenti campi,

```
#smtp settings CONFIGURE
mailfrom=me@email.com
smtp=${#smtp-host#}
portSmt=${#smtp-port#}
authSmt=${#smtp-auth#}
#authTypeSmt=TLS
#userSmt=
#passwdSmt=
```

e.g.:

```
#smtp settings CONFIGURE
mailfrom=dashboard@xxxxxcloud
smtp=smtps.aruba.it
portSmt=465
authSmt=${#smtp-auth#}
authTypeSmt=TLS
userSmt=<MANCA>
passwdSmt=xxxx@2022
```

Then restart **servicemap**

## Clonare modelli

- Modelli nella tabella `iotdb.model` del `dashboarddb`
- Ownership nella tabella `profiledb.ownership` del `dashboarddb`

## Clonare static attributes

- Clonare le tabelle `processloader_db.dictionary_table` e `processloader_db.dictionary_relations`
- Aggiornare VIRTUOSO come di seguito

L'importazione del dictionary richiede la ri-sincronizzazione di `servicemap`, utilizzando `update-valuetypes.sh` in `servicemap-conf`, che dovrebbe contenere

```
cd "$(dirname "$0")"
echo -n "$(date -Iseconds) get valuetypes from processloader... "
curl -sS http://localhost:90/processloader/get\_valuetypes.php > valuetypes.vt

STATUS="$(cmp --silent valuetypes.vt valuetypes-last.vt ; echo $?)" # "$?" gives exit status for
each comparison

if [ "$STATUS" -ne "0" ]; then
    echo CHANGED valuetypes upload to virtuoso
    /usr/local/bin/docker-compose exec -T virtuoso-kb isql-v localhost dba dba
/root/servicemap/valuetypes.vt
    cp valuetypes.vt valuetypes-last.vt
else
    echo SAME valuetypes
fi
```

Prima di lanciare lo script, verificare la risposta dell'URL: per l'installazione su Asymmetrica il corretto URL era [https://www.cityconn.cloud/processloader/get\\_valuetypes.php](https://www.cityconn.cloud/processloader/get_valuetypes.php)

È poi necessario modificare la password di virtuoso (e.g. dba) se questa è stata modificata durante la creazione della microX.

Infine, per forzare l'upload, e forzare l'aggiornamento della KB anche se lo script non trova cambiamenti, è necessario cancellare il file `valuetypes-last.vt`

Mappare cartella locale con `docker-compose` in un volume dentro ad un container nel docker compose per esempio nella sezione del `dashboard-builder`

tra i volumi aggiungere

- `developmentfiles:/var/www/html:rw`

poi in fondo al `docker-compose` aggiungere questo

developmentfiles:

driver: local

driver\_opts:

type: none

device: /home/ubuntu/developmentfiles

o: bind

e rifare il docker-compose up -d

e nella cartella troverai tutti i files ;)

## Possibili errori e soluzioni

### Come creare il broker

- 1) come ip mettere l'ip interno della macchina ifconfig dal terminale e porta 1026!
- 2) come access link opzione A) mettere l'ip pubblico oppure B) sempre quello interno con porta 8443

### Come accedere a virtuoso

via ssh fare un tunnelling sulla porta 8890 e accedi all'interfaccia da browser

se metti lo slash sparql puoi fare le query <http://localhost:8890/sparql>

tipo questa

```
select distinct ?c {  
  ?s a ?c.  
}
```

per vedere se c'è qualcosa in kb

### Come gestire il problema No registration in the context broker No registration in kb quando creiamo il device

- 1) rifai la procedura e fai F12 e vai in network - fetch x.. preview e vai in fondo per vedere tutto l'errore per bene
- 2) controllare i logs in servicemap-iot-conf insert se c'è qualcosa
- 3) sudo docker-compose exec servicemap bash
- 4) poi cd logs e vai nell'ultimo access
- 5) cat localhost\_access\_log.2023-08-30.txt | grep iot e vedi cosa succede

### Come aumentare il tempo di timeout prima che la piattaforma ti butti fuori

- 1) spegnere il container dashboard builder
- 2) nella cartella dove risiedono i file di configurazione andare nella cartella indicata nelle virgolette "dashboard-builder-conf"
- 3) aprire il file "general.ini"
- 4) per i valori sessionDuration[dev], sessionDuration[test] e sessionDuration[prod] impostare il valore desiderato (in secondi, quindi un giorno vale 86400)
- 5) salvare il file
- 6) riavviare il container



"docker-compose down dashboard-builder", da eseguire nella console nella cartella dove si trova il docker-compose.yaml

per tirarlo su si sostituisce down con up

volendo si può omettere dashboard-builder; in tal caso tira giù tutto o tira su tutto (tirare su qualcosa già in esecuzione non ha effetto)

per "up" la console sarà occupata dall'output del (dei, se più di un) container, per andare in modalità detached si può usare il flag -d

Tutto quieto... docker-compose ps tutto su ma poi accedi dal browser e schermata bianca senza errori su console

Soluzione: c'è una entry nel database del dominio che è rimasta 'dashboard' e va cambiata con quella specificata... forse nelle future versioni del builder verrà sistemato in automatico comunque gli step per risolverlo sono:

1 trovare nel docker-compose la psw del mysql db

2A accedere o con workbench oppure

2B fare docker bash nel dashboard container e far partire mysql -u root -p

-use dashboard

- select \* from Domains e controllare se è tutto ok

- eventualmente fare l'update del campo

Delega di Orion al gruppo

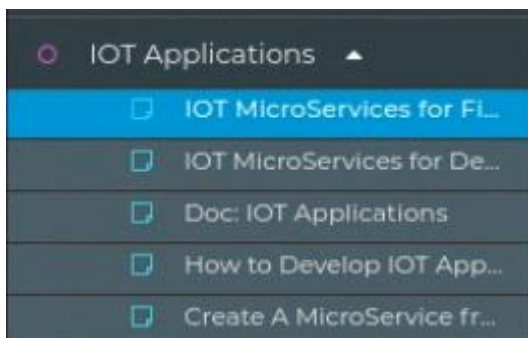
Durante il primo accesso di rootadmin, dopo aver preso possesso di Orion (facendo edit senza modificare), a volte può capitare che non sia possibile delegare il broker al un gruppo.

**Errore:** *Delegated groupname can't be empty*

**Soluzione:** Eseguire restart del container personaldata

Nessuna IOT app della microx è accessibile dal sottomenu

Errore:



**Soluzione:** Eseguito il restart del docker-compose

## Orion non riceve dati da nodeRED (1)

Il nodo che sottoscrive e invia dati su Orion dà il seguente errore

```
"failed to update, reason: {"statusCode":401,"headers":{"x-content-type-options":"nosniff","x-xss-protection":"1; mode=block","cache-control":"no-cache, no-store, max-age=0, must-revalidate","pragma":"no-cache","expires":"0","strict-transport-security":"max-age=31536000 ; includeSubDomains","x-frame-options":"DENY","transfer-encoding":"chunked","date":"Tue, 19 Jul 2022 10:43:08 GMT","connection":"close"},"payload":{"\result\":false,\message\":\"The passed access token is not valid\"}}"
```

**Soluzione:** verificare che la password per rootadmin sia coerente tra quella che è stata imposta nel generatore della microX (e che viene usata da dashboard-builder leggendo il file dashboard-builder-conf/personaldata.ini per accedere ai dispositivi privati) con la password impostata in ldap/keyclock.

## Orion non riceve dati da nodeRED (2)

NodeRED non riesce a comunicare con Orion, dando il seguente errore:

```
"failed to update, reason: {"errno":"ECONNREFUSED","code":"ECONNREFUSED","syscall":"connect","address":"5.189.175.163","port":8443}"
```

**Soluzione:** verifica che i parametri di connessione a Orion utilizzino il nome del container e non l'IP (soprattutto se è stato eliminato il port mapping per chiudere le porte). Poi nella scheda con i parametri della connessione a Orion (vedi figura)

Inserire il nome del container di Orion in IP (e.g. orion-001) e il nome del brokerfilter in AccessLink (e.g. orionbrokerfilter-001). Quindi riavvia i container.

## Orion va in stallo (a seguito di un crash di sistema)

A seguito di crash, Orion può andare in stallo riportando il seguente errore:

```
orion-001_1 | time=2022-11-16T11:26:03.258Z | lvl=ERROR | corr=N/A | trans=N/A | from=N/A | srv=N/A | subsrv=N/A | comp=Orion | op=contextBroker.cpp[432]:pidFile | msg=PID-file '/tmp/contextBroker.pid' found. A broker seems to be running already
```

**Soluzione:** eliminare e ricreato il container con

```
docker-compose stop orion
docker-compose rm orion
docker-compose up -d orion
```

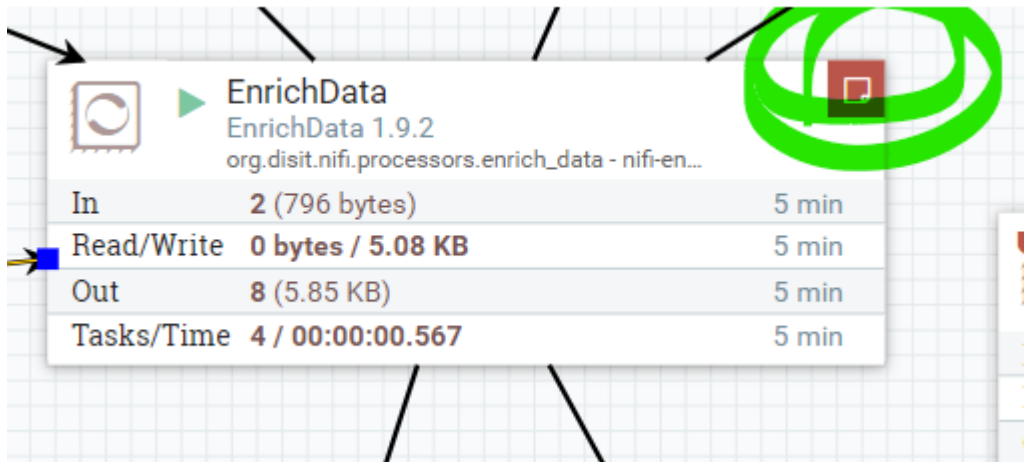
oppure con

```
docker-compose up -d --force-recreate --no-deps orion
```

cambi psw di root e nifi va aggiornato

fai il tunnelling <https://localhost:9090>

poi in enrich data clicchi due volte sul



Property	Value
Enrichment Source Client Service	ServicemapOAuthControllerService2

NIFI Flow Configuration

Name	Type	Bundle	State	Scope
IoTDirectoryOAuthLocatorControllerService	IoTDirectoryOAuthLocatorControllerService 1.9.2	org.disit.nifi.processors.enrich_data - nifi-resource-locator-servi...	Enabled	NIFI Flow
KeycloakTokenProviderControllerService	KeycloakTokenProviderControllerService 1.9.2	org.disit.nifi.processors.enrich_data - nifi-oauth-token-providers...	Enabled	NIFI Flow
OwnershipOAuthControllerService	OwnershipOAuthControllerService 1.9.2	org.disit.nifi.processors.enrich_data - nifi-ownership-client-servi...	Enabled	NIFI Flow
ServicemapOAuthControllerService2	ServicemapOAuthControllerService 1.9.2	org.disit.nifi.processors.enrich_data - nifi-servicemap-client-servi...	Enabled	NIFI Flow
StandardRestrictedSSLContextService	StandardRestrictedSSLContextService 1.16.2	org.apache.nifi - nifi-ssl-context-service-nar	Enabled	NIFI Flow

**Controller Service Details** | KeycloakTokenProviderControllerService 1.9.2

SETTINGS | PROPERTIES | COMMENTS

Required field

Property	Value
Keycloak Uri	https://cuneo.snap4.eu
Client ID	nifi-processor
Client secret	8172c74d-56c0-4b4d-a3a2-44cfac82af96
Realm	master
Username	userrootadmin
Password	Sensitive value

Scope

NiFi Flow
NiFi Flow
NiFi Flow
NiFi Flow
NiFi Flow

OK

riattivi tutti i fulminini

cambiala anche nel compose quella di orionbrokerfilter

fai prima stop poi rm del docker-compose orionbrokerfilter-001

poi up -d

cambi psw di root e il device non c'è nel data-inspector

andare dentro alla cartella dashboard-builder-conf

in personalData.ini modificare a psw di userrootadmin con quella nuova

I dati si bloccano su NiFi in enrich-data

Il passaggio dei dati da Orion attraverso NiFi si blocca in enrich-data, con il seguente errore:

```
EnrichData[id=03fcba02-017c-1000-10b5-1b63ebcd9871]
{"reason":"org.apache.http.client.HttpResponseException while handling the Servicemap response body.
Routing to
FAILURE_RELATIONSHIP", "content":{"id":"testWind", "type":"sensor", "dateObserved":{"type":"string", "va
```

```

"lue":"2022-07-
19T11:07:20.261Z","metadata":{},"latitude":{"type":"float","value":"43.79221","metadata":{}}, "longi
tude":{"type":"float","value":"11.30587","metadata":{}}, "model":{"type":"string","value":"windModel"
,"metadata":{}}, "wind":{"type":"float","value":"8.176945692257064","metadata":{}}, "date_time":"2022-
07-
19T11:07:20.532Z"},"exception":"org.disit.nifi.processors.enrich_data.enrichment_source.EnrichmentSo
urceException: org.apache.http.client.HttpResponseException while handling the Servicemap response
body.", "cause":"org.apache.http.client.HttpResponseException: status code: 400","ff-uuid":"433c8483-
b384-48df-9ba9-ccdd7990a79b","Servicemap_response":{" \n \ "failure\ " : \ "ERROR\ ", \n \ "httpcode\ " :
400, \n \ "message\ " : \ "invalid apikey\ ", \n \ "apiDoc\ " : \ "http://www.disit.org/6991\ "\n }"}

```

**Soluzione:** enrich-data non riesce ad accedere a servicemap, a causa di problemi con APIKEY (che può essere modificata durante la generazione della MicroX). Per risolvere la apikey va rimossa dalle impostazioni di enrich-data.

**Nota:** questa correzione fu fatta direttamente, quindi non è perfettamente chiaro come andare a eliminare la apikey da enrich-data.

### NiFi non salva su opensearch/elasticserach

Come per enrich-data, in questo caso il flusso NiFi si blocca sul nodo che va a scrivere su opensearch. Il problema è che la password per opensearch (\$#opensearch-admin-pwd#\$ in placeholder) non è correttamente settata.

**Soluzione:** modificare la password.

**Note:** anche in questo caso l'operazione fu fatta da noi.

### Il device non c'è nella service map

Errore: appare un alert con " no access to http://...link..al device/Organization/testdevice"

Ma il device è visibile nel data inspector

soluzione: controllare e ricordarsi di mettere il device pubblico sennò in service-map non si vede!

### Il device non appare in data-inspector

Il device creato non appare in data inspector: running2.txt non c'è e i log in **dashboard-cron** riportano

```
# cat health.log
```

```
PHP Warning: syntax error, unexpected '$' in ../conf/ssl_expose.php on line 2
```

```
in /var/www/html/dashboardSmartCity/config.php on line 27
```

```
PHP Warning: Invalid argument supplied for foreach() in /var/www/html/dashboardSmartCity/config.php
on line 29
```

```
Starting HealthinessCheck SCRIPT at: 2022-07-19 15:45:01+02:00
```

```
Udpating : blueCode at: 2022-07-19 15:45:06+02:00 --> healthiness = false
```

```
Udpating : greenCode at: 2022-07-19 15:45:06+02:00 --> healthiness = false
```

```
Udpating : redCode at: 2022-07-19 15:45:06+02:00 --> healthiness = false
```

```
Udpating : state at: 2022-07-19 15:45:06+02:00 --> healthiness = false
```

```
Udpating : whiteCode at: 2022-07-19 15:45:06+02:00 --> healthiness = false
```

```
Udpating : yellowCode at: 2022-07-19 15:45:06+02:00 --> healthiness = false
```

```

1 FINISHED HEALTHINESS CHECK FOR DEVICE: FirstAid
End HealthinessCheck SCRIPT at: 2022-07-19 15:45:06

# cat feed-iot.log
PHP Warning: syntax error, unexpected '$' in ../conf/ssl_expose.php on line 2
in /var/www/html/dashboardSmartCity/config.php on line 27
PHP Warning: Invalid argument supplied for foreach() in /var/www/html/dashboardSmartCity/config.php
on line 29
*** Starting IOT_Sensor_FeedDashboardWizard SCRIPT at: 2022-07-19 15:50:01
1 - IOT DEVICE: , MEASURE:
2 - IOT DEVICE: http://www.disit.org/km4city/resource/iot/orion-1/Organization/testWind, MEASURE:
http://www.disit.org/km4city/resource/iot/orion-1/Organization/testWind/dateObserved
End IOT_Sensor_FeedDashboardWizard SCRIPT at: 2022-07-19 15:50:01

# cat feed-iot-app.log
PHP Warning: syntax error, unexpected '$' in ../conf/ssl_expose.php on line 2
in /var/www/html/dashboardSmartCity/config.php on line 27
PHP Warning: Invalid argument supplied for foreach() in /var/www/html/dashboardSmartCity/config.php
on line 29
*** Starting FeedT IO_App SCRIPT at: 2022-07-19 15:51:02
End Feed IOT_App SCRIPT at: 2022-07-19 15:51:02

```

**Nella tabella dashboard.wizard del DB il device appare ma impostato a old**

```

| 15715 | From IOT Device to KB | Sensor | IoTSensor
| | Organization:orion-1:testWind | single_marker |
http://www.disit.org/km4city/resource/iot/orion-1/Organization/testWind
| NULL | NULL | sensor_map | no | direct
| yes | yes | NULL | NULL |
api/v1/?serviceUri=http://www.disit.org/km4city/resource/iot/orion-
1/Organization/testWind&format=json
| false | NULL | 2022-07-19 16:05:01 | private | Organization
| 43.79221 | 11.30587 | NULL | aGxzc11VUM1V1hUdUkveE50REZIUT09 |
| old |

| 15716 | From IOT Device to KB | Sensor | IoTSensor
| dateObserved | Organization:orion-1:testWind | single_marker |
http://www.disit.org/km4city/resource/iot/orion-1/Organization/testWind
| NULL | NULL | time | no | direct
| yes | yes | NULL | NULL |
api/v1/?serviceUri=http://www.disit.org/km4city/resource/iot/orion-
1/Organization/testWind&format=json
| false | NULL | 2022-07-19 16:05:01 | private | Organization
| 43.79221 | 11.30587 | timestamp | aGxzc11VUM1V1hUdUkveE50REZIUT09 |
| old |

| 15717 | From IOT Device to KB | Sensor | IoTSensor
| wind | Organization:orion-1:testWind | single_marker |
http://www.disit.org/km4city/resource/iot/orion-1/Organization/testWind
| NULL | NULL | float | no | direct
| yes | yes | NULL | NULL |
api/v1/?serviceUri=http://www.disit.org/km4city/resource/iot/orion-
1/Organization/testWind&format=json
| false | NULL | 2022-07-19 16:05:01 | private | Organization
| 43.79221 | 11.30587 | m/s | aGxzc11VUM1V1hUdUkveE50REZIUT09 |
| old |

```

Il campo a old significa che è fallita la healthiness. La healthiness si basa su apikey di servicemap. Il problema è che apikey servicemap ipotizza sia un esadecimale, mentre il generatore genera random lettere numeri.

Soluzione: aggiorna la versione di servicemap alla v4.3, quindi rimetti a null i valori old

## Cambiare il timeout della sessione

\*\* Come aumentare il tempo di timeout prima che la piattaforma ti butti fuori

1. spegnere il container dashboard builder;
2. nella cartella dove risiedono i file di configurazione andare nella cartella indicata nelle virgolette "dashboard-builder-conf";
3. aprire il file "general.ini";
4. per i valori sessionDuration[dev], sessionDuration[test] e sessionDuration[prod] impostare il valore desiderato (in secondi, quindi un giorno vale 86400);
5. salvare il file;
6. riavviare il container;

"docker-compose down dashboard-builder", da eseguire nella console nella cartella dove si trova il docker-compose.yaml

per tirarlo su si sostituisce down con up

volendo si può omettere dashboard-builder; in tal caso tira giù tutto o tira su tutto (tirare su qualcosa già in esecuzione non ha effetto)

per "up" la console sarà occupata dall'output del (dei, se più di un) container, per andare in modalità detached si può usare il flag -d

## Ripulire i token di una iot app

```
docker-compose stop iotapp-xxx
```

```
cd iotapp-xxx
```

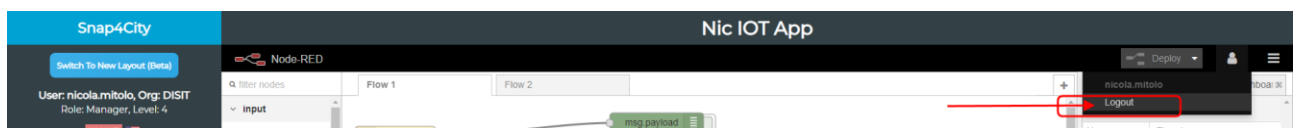
```
ls -la
```

```
rm refresh_token*
```

```
cd ..
```

```
docker-compose start iotapp-xxx
```

Entrare nell'IOTApp e assicurarsi di fare logout dall'icona in alto a destra



Rientrare nell'IOTApp per essere sicuri di accedere con l'utente corretto.

### Impostare il tunnelling per l'accesso all'interfaccia di NIFI

```
ssh -L 9090:localhost:9090 username@IP
```

```
>Yes
```

```
>password
```

accedere all'URL <https://localhost:9090>

Login con password dentro il docker-compre

```
user: admin
```

### Lista delle sottoscrizioni del broker

Broker subscription

Andare su *IP:1026/v2/subscriptions*

### Abilitare un utente a CSBL

Aggiungere il nome utente nella tabella *TrustedUsers* del database *Dashboard*.

### Aumentare il default refresh time della Micro X

Può succedere che nonostante i valori di healthness dei device siano impostati con valori bassi, nel data inspector il pallino di healthness rimane verde anche se il dato non arriva. Questo può essere dovuto a un valore di refresh time impostato di default durante la configurazione della Micro X che va quindi rieditato per renderlo coerente con i valori di healthness desiderati.

## **[BUG IN FASE di RISOLUZIONE]**

### Aggiornare il dictionary dopo aver aggiunto value type nuovi

Se si aggiungono value type al dictionary, ci possono essere errori in fase di creazione dei device con i value type nuovi.

Andare nella cartella *servicemap-conf*

Aggiornare la password di virtuoso dell'utente *dba* nello script `./update-valuetypes.sh`

La password dell'utente *dba* si trova nel file `docker-compose.yml` e viene generata random.

Salvare lo script ed eseguirlo.

Eventualmente può essere utile mettere un cron ogni 10 minuti come in *Snap4City*.

### Se api *iot-search* non funziona

se api *iot-search* non restituisce niente, controllare che *nifi* stia scrivendo i dati su indice giusto, l'indice giusto è *iot-device-state* e non *device-iot-organization*

Collegarsi a *nifi* e controllare il *put-elastic* in alto che salva l'ultimo stato dei device, se è sbagliato:



1. fermare il put-elastic
2. aprire kibana come userrootadmin e andare nei dev tools e mandare il seguente

```
POST _reindex
{
  "source":{
    "index":"device-iot-organization"
  },
  "dest":{
    "index":"iot-device-state"
  }
}
```

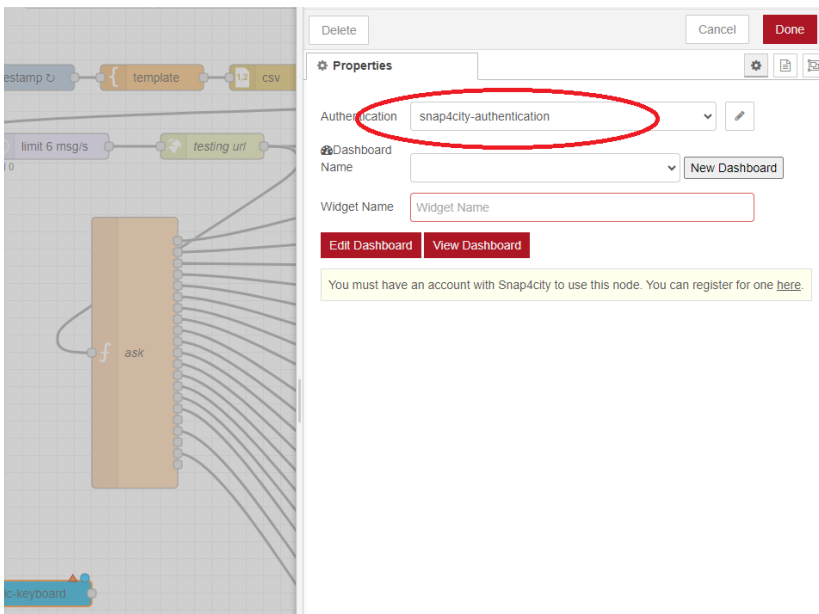
che serve a reindicizzare i dati eventualmente già presenti sull'indice giusto

3. a questo punto ritornare su nifi e modificare nel blocco put-elastic il nome dell'indice su cui salvare e mettere il valore giusto **iot-device-state** e salvare
4. far ripartire il put-elastic

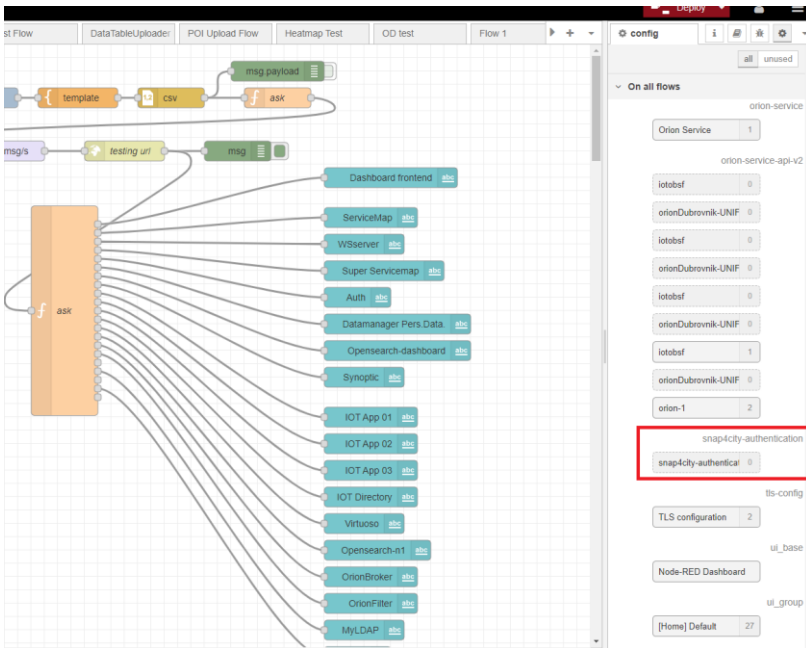
(fare attenzione a non modificare l'altro put-elastic che salva i dati)

### Autenticazione vuota su flusso Node-RED

Nelle iot app delle Micro X c'è un oggetto autenticazione (vuoto) che automaticamente si setta come autenticazione ai blocchetti user.



Per risolvere cancellare dalle impostazioni l'autenticazione come da immagine



(Fabrizio sta risolvendo)

### Ripulire il db dai dati di esempio presenti nelle MicroX

le righe con high\_level\_type = "External Service" nella tabella DashboardWizard nel database Dashboard per cancellarli tutti:

```
DELETE FROM Dashboard.DashboardWizard WHERE high_level_type = "External Service";
```

Ripetere per heatmaps e gli "Special events" se non di interesse

### Flusso IOTApp

```
{{"id":"955e4a58ca1c42e4","type":"tab","label":"Flow
1","disabled":false,"info":"","env":[]},{{"id":"75a8edb4987afe1d","type":"inject","z":"955e4a58ca1c42e
4","name":"","props":{"p":"payload"},{"p":"topic","vt":"str"},"repeat":"","crontab":"","once":false,"once
Delay":0.1,"topic":"","payload":"","payloadType":"date","x":120,"y":140,"wires":[["d62e1d387d5c970
3"]]},{"id":"89ee7a76aa229b75","type":"orion-out-api-
v2","z":"955e4a58ca1c42e4","name":"","tls":"","service":"97a0f29a.0bf4f","entype":"","enid":"","user
k1":"","passk2":"","tenant":"","servicepath":"","apikey":"","basicAuth":"","x":440,"y":160,"wires":[["211
7e58e9449612d"]]},{"id":"83fce0279032f172","type":"debug","z":"955e4a58ca1c42e4","name":"","a
ctive":true,"tosidebar":true,"console":false,"tostatus":false,"complete":false,"statusVal":"","statusT
ype":"auto","x":410,"y":260,"wires":[]},{{"id":"d62e1d387d5c9703","type":"function","z":"955e4a58ca1
c42e4","name":"","func":"adesso = new Date().toISOString()\nmsg.payload =
{\n  \"id\": \"dajedevicel\",\n  \"type\": \"test\",\n  \"dateObserved\": {\n    \"type\": \"string\",\n    \"value\": \"adess
o\",\n  \"latemperature\": {\n    \"type\": \"float\",\n    \"value\": 50.1\n  }\n}\n\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":180,"y":240,"wires":[["83fce0279032f
172","89ee7a76aa229b75"]]},{"id":"2117e58e9449612d","type":"debug","z":"955e4a58ca1c42e4","
name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":false,"statusVal":
":"","statusType":"auto","x":630,"y":160,"wires":[]},{{"id":"97a0f29a.0bf4f","type":"orion-service-api-
```

```
v2","contextbroker":"orion-001","selectedContextbroker":"","name":"orion-001","authentication":"","url":"orionbrokerfilter-001","port":"8443"}]
```