Installazione lato Server

OS: Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-141-generic x86_64)

HW: 8GB Ram, 4vCPU, 50GB storage

NodeJS

In questa sezione andremo ad installare l'ultima versione stabile di nodeJS ed il pacchetto npm che servirà per l'installazione dei moduli necessari a Nodered.

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -

apt-get install -y nodejs node-gyp

npm install -g npm

echo "NODE_PATH=/usr/lib/node_modules" | tee -a /etc/environment
```

Nodered

Installiamo nodered e ne personalizziamo il funzionamento in modo da memorizzare i flussi nella home dell'utente di default ovvero **ubuntu**.

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

wget
https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/reso
urces/nodered.service -O /lib/systemd/system/nodered.service

wget
https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/reso
urces/node-red-start -O /usr/bin/node-red-start

wget
https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/reso
urces/node-red-stop -O /usr/bin/node-red-stop

chmod +x /usr/bin/node-red-st*

sed -i 's/User=pi/User=ubuntu/g' /lib/systemd/system/nodered.service

sed -i 's/Group=pi/Group=ubuntu/g' /lib/systemd/system/nodered.service

sed -i 's/WorkingDirectory=\/home\/pi/WorkingDirectory=\/home\/ubuntu/g'
/lib/systemd/system/nodered.service


systemctl daemon-reload

systemctl enable nodered.service
```

Se fosse necessario aggiungere un utente in sola lettura lanciare il seguente comando e verrà richiesto l'inserimento di una password che verrà criptata per poterla copiare successivamente nell'apposito file di configurazione.

```
node-red-admin hash-pw
```

Password: `<PASSWORD>`

Copiare la stringa che viene restituita ed incollarla all'interno del file di `/home/ubuntu/.node-red/settings.js` di nodered specificando l'username e prestando attenzione al campo permissions. E' necessario quindi aggiungere una nuova sezione come quella evidenziata in rosso.

```
adminAuth: {
    type: "credentials",
    users: [{
            username: "ubuntu",
            password: "<HASH_UBUNTU_PSW>",
            permissions: "*"
        },
        {
            username: "user",
            password: "<HASH_USER_PSW>",
            permissions: "read"
        }
    ]
},
```

Per proteggere con password l'accesso alla dashboard è necessario decommentare le seguenti righe (sempre nello stesso file `settings.js`):

```
    httpNodeAuth: {user:"ubuntu",pass:"<HASH_UBUNTU_PSW>"},
    httpStaticAuth: {user:"ubuntu",pass:"<HASH_UBUNTU_PSW>"},
```

Avviare il servizio:

```
systemctl start nodered.service
```

e collegarsi alla sua interfaccia:

```
http://<SERVER_IP>:1880
```

All'interno dei flussi è stato inserito un blocco che richiede l'installazione di un modulo di nodered. Andare in Manage palette e cercare `node-red-contrib-influxdb` nella tab di Install e procedere all'installazione

Influxdb & grafana

Per memorizzare le misurazioni dei pazienti si è concordato di utilizzare un database timeseries based come InfluxDB che verrà poi interrogato da Grafana per esportare i grafici di monitoraggio al frontend IGCOM tramite iframe.

```
cd /tmp
wget https://dl.influxdata.com/influxdb/releases/influxdb_1.7.2_amd64.deb
sudo dpkg -i influxdb_1.7.2_amd64.deb
wget https://dl.grafana.com/oss/release/grafana_5.4.2_amd64.deb
sudo dpkg -i grafana_5.4.2_amd64.deb
systemctl enable grafana-server.service influxdb.service
systemctl start influxdb grafana-server
sudo update-rc.d grafana-server defaults
```

Per blindare l'accesso al database e renderlo accessibile solo internamente al server stesso è necessario specificare il bind-address come segue.

```
vim /etc/influxdb/influxdb.conf
      bind-address = "127.0.0.1:8086"
systemctl restart influxdb
```

Configurazione Grafana → Influxdb
Creazione database
Entrare nella bash del server:

```
influx
      create database giomi
```

Collegarsi alla pagina di grafana esposta dal server `http://<SERVER_IP>:3000`

1. Andare su configuration -> Data Sources -> Add data source -> selezionare Influxdb
2. specificare l'url come segue: http://127.0.0.1:8086
3. specificare il nome del database specificato in precedenza: **giomi** con le relative credenziali (se specificate)

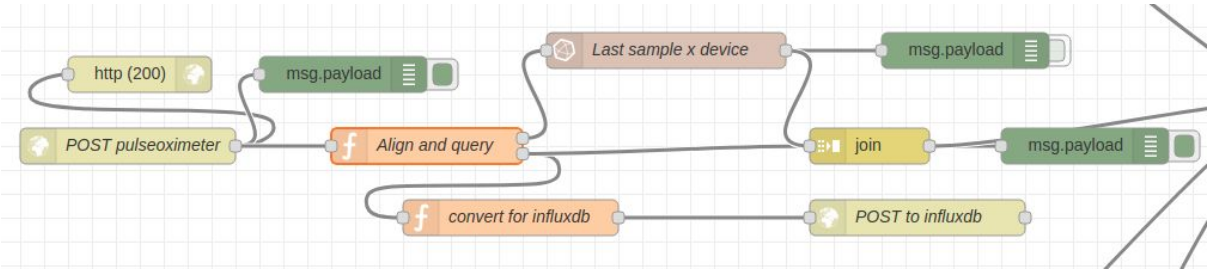
Nodered Dashboard
Una volta loggati all'interno dell'interfaccia di Nodered è necessario andare nell'apposita sezione ed importare i flussi necessari all'elaborazione dei dati provenienti dai sensori per la generazione degli allarmi nonché per la memorizzazione sul database.

Sul server esistono i seguenti flussi:

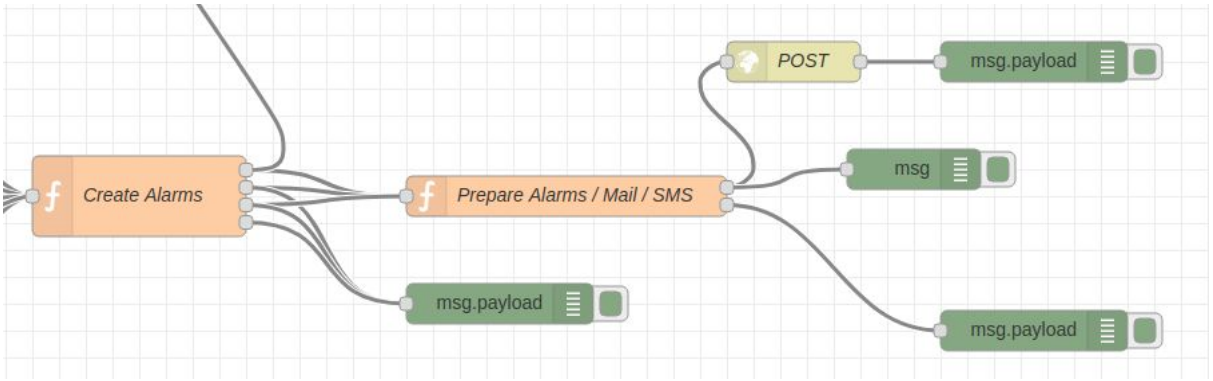
1. Manage Sensor Data
2. Movement
3. SmartWatch Sensor Data
4. GET Token

Manage Sensor Data
Questo flusso si compone di due parti di cui la prima (immagine sotto) espone l'endpoint per le metriche provenienti dai device (a cui sono agganciati gli allarmi provenienti dall'API

IGCOM richiamata precedentemente dalla Raspberry), si preoccupa di convertirli ed inviarli ad influxdb e di recuperare gli ultimi campioni degli altri devices facendo delle query al database in modo da poterli agganciare in un unico oggetto che verrà elaborato nella seconda parte.



Per semplicità di gestione, nonché per una più rapida fase di debug, è stato creato un endpoint per ogni tipologia di device. Questo risulta comodo qualora fosse necessario agganciare nuove tipologie di device.



La seconda parte prevede una fase di analisi degli allarmi in base alle soglie configurate lato frontend ed in funzione dei valori recuperati dal database. Nel caso in cui si rientri in uno dei casi per i quali è necessario inviare un allarme, il sistema provvede ad inviare una POST ad IGCOM con tutti i dati necessari per inviare una mail, un sms e per notificare tramite il portale la presenza di un allarme.

Movement
Questo flusso permette di memorizzare sul database gli eventi di movimento registrati dalla Raspberry. Ovviamente non è prevista alcuna elaborazione in quanto viene inviato solo un valore 1 nel caso di rilevamento.

SmartWatch Sensor Data
A dispetto degli altri device che sono agganciati al Raspberry, il quale si incarica di contattare l'API IGCOM per ottenere l'oggetto con tutti gli allarmi, nel caso dello SmartWatch questo invierà solamente l'oggetto contenente la rilevazione e sarà il Cloud che si occuperà di contattare l'API IGCOM per ottenere tutti gli allarmi. Questo oggetto (allarmi + rilevazione) verrà inviato al corrispettivo endpoint del Manage Sensor Data descritto in precedenza per poter effettuare i controlli relativi alla metrica in questione.

GET Token
In questo flusso viene banalmente effettuata una chiamata all'endpoint di IGCOM in modo da ottenere il token che permetterà di autenticare le chiamate POST fatte all'API, esposta

anch’essa da IGCOM, per l’invio di tutte le informazioni necessarie al fine di poter poi scatenere gli allarmi, SMS ed email.