

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@0.19.5
```

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

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

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

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

Modificare il file /lib/systemd/system/nodered.service e cambiare i seguenti parametri:

User = [UTENTE]

Group = [UTENTE]

WorkingDirectory = /home/[UTENTE]

Installazione flussi

```
cd /home/[UTENTE]

curl -O https://git.smartme.io/alessio.cosenza/giomi-flows/raw/docker/settings.js
curl -O https://git.smartme.io/alessio.cosenza/giomi-flows/raw/docker/package.json

curl -o flows_cred.json
https://git.smartme.io/alessio.cosenza/giomiflows/raw/docker/manage\_sensor\_data\_cred.json

curl -o flows.json
https://git.smartme.io/alessio.cosenza/giomiflows/raw/docker/manage_sensor_data.json

npm install


systemctl daemon-reload
systemctl enable nodered.service
```

Configurare endpoint centrale operativa

Creare la cartella /data

```
mkdir /data

cd /data

sudo nano endpoints.json
```

Scrivere quanto segue:

```
{
    "endpoint": "[ENDPOINT: ad es. http://37.148.227.52/semprevicini]"
}
```

Avviare il servizio: systemctl start nodered.service

Installazione grafana e influxdb

```
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)

Installare le dashboards:

```
wget https://raw.githubusercontent.com/igcomsrl/iot\_eCare\_ai/master/dashboard.json -O
/usr/share/grafana/public/dashboards/dashboard.json
```

```
wget https://raw.githubusercontent.com/igcomsrl/iot\_eCare\_ai/master/automatic.yml -O  
/etc/grafana/provisioning/datasources/automatic.yml
```

```
wget https://raw.githubusercontent.com/igcomsrl/iot\_eCare\_ai/master/scripted.js -O  
/usr/share/grafana/public/dashboards/scripted.js
```

```
wget https://raw.githubusercontent.com/igcomsrl/iot\_eCare\_ai/master/dashboards.yml -O  
/etc/grafana/provisioning/dashboards/dashboards.yml
```

Riavviare Grafana:

```
systemctl restart grafana
```