

Monitoring d'une Freebox Révolution

Le but de ce Tuto :

Il s'agit ici de présenter un tuto pour monitorer sa Freebox, avec les outils telegraf / influxdb / grafana, en complément de ce Tuto général [TUTO] Monitoring NAS et Réseau qui reste la référence. En conséquence, je suppose déjà opérationnelle une chaîne complète telegraf / influxdb / grafana pour la supervision de votre NAS.

Il faudra bien entendu faire les adaptations spécifiques à votre environnement.

Principe : Le docker telegraf va utiliser un script python pour récupérer les infos de la Freebox. Il y a quelques semaines, je ne connaissait pas python ... donc il y a surement des façons plus élégantes de programmer, merci de votre indulgence.

Limites :

je n'ai pu valider cette méthode que sur ma Freebox, et donc pour les autres modèles et/ou configuration ... ?

Freebox Révolution au dernier Firmware (4.0.7) à ce jour.

Accès réseau fibre ftth (et donc pas testé les compteurs xDSL)

Sources :

<https://www.nas-forum.com/forum/topic/63273-tuto-monitoring-nas-et-réseau/?tab=comments#comment-1319376875>

<https://github.com/tsugliani/freebox-revolution-monitoring>

<https://gikspirit.com/comment-installer-python-3-7-sur-ubuntu-18-04/>

<https://docs.python.org/fr/3/installing/index.html>

https://github.com/influxdata/telegraf/blob/master/docs/DATA_FORMATS_INPUT.md

<https://dev.freebox.fr/sdk/os>

<https://github.com/influxdata/telegraf/issues>

<http://sametmax.com/lencoding-en-python-une-bonne-fois-pour-toute/>

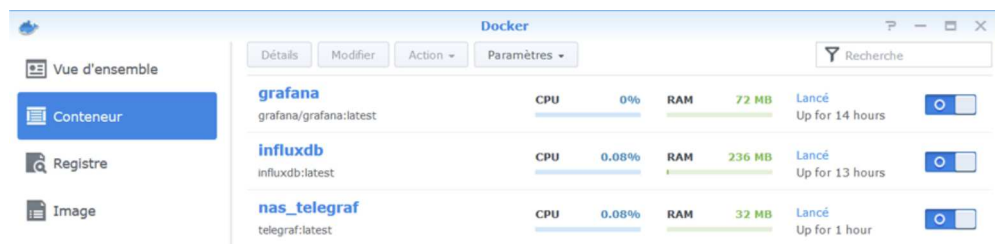
<https://github.com/influxdata/telegraf/blob/master/README.md>

https://docs.influxdata.com/influxdb/v1.7/query_language/spec/

<https://www.elastic.co/guide/en/elasticsearch/reference/6.2/query-dsl-regexp-query.html#regexp-syntax>

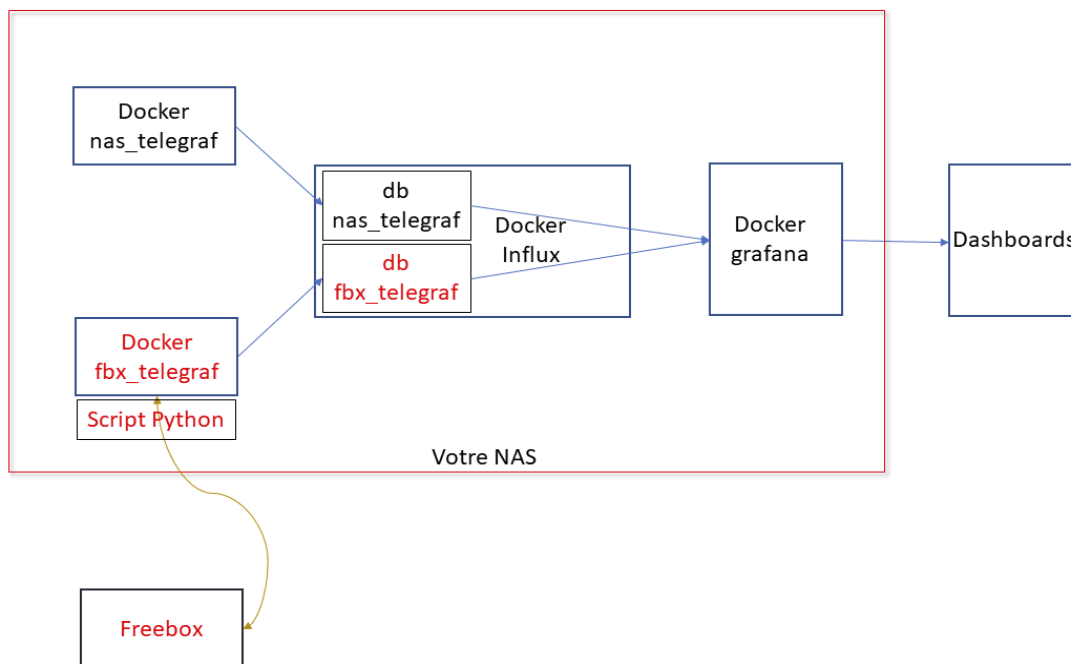
Chaîne existante (hypothèse) :

je suppose donc que vous avez une chaîne telegraf / influxdb / grafana opérationnelle pour votre NAS, en docker mode bridge.



Ajout Freebox :

On va rajouter un second docker telegraf, qui portera le script Pyhton. Ce docker telegraf dédié Freebox va transmettre ses données vers une seconde database que l'on va créer dans le docker Influx. Sur grafana, on créera une nouvelle source de données (la nouvelle database d'influx) qui nous permettra d'afficher les données Freebox



Configuration des dockers existants :

subnet	172.20.0.0/29
gateway	172.20.0.1
ip_range	172.20.0.0/29

Adressage

grafana	d2:ca:ab:cd:00:02	172.20.0.2
influxdb	d2:ca:ab:cd:00:03	172.20.0.3
nas_telegraf	d2:ca:ab:cd:00:04	172.20.0.4

Creation du nouveau docker : fbx_telegraf

Adressage sur le réseau bridge :

fbx_telegraf	d2:ca:ab:cd:00:05	172.20.0.5
--------------	-------------------	------------

Déclaration du service fbx_telegraf dans le docker-compose.yml (extrait):

```
services:
  fbx_telegraf:
    image: telegraf:latest
    container_name: fbx_telegraf
    hostname: fbx_telegraf mac_address: d2:ca:ab:cd:00:05
    networks:
      monitoring:
        ipv4_address: 172.20.0.5
    environment:
      - PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin:/sbin:/usr/local:/usr/src
      - TZ=CET
    mem_limit: 75M
    volumes:
      - "/volume1/docker/monitoring/fbx_telegraf/telegraf.conf:/etc/telegraf/telegraf.conf:ro"

      # pour le fichier de commande python,
      # pour le fichier get-pip.py (install module pip puis requests)
      - "/volume1/docker/monitoring/fbx_telegraf/py:/usr/local/py:ro"
      # pour le fichier log si on le met en place
      - "/volume1/docker/monitoring/fbx_telegraf/log:/usr/local/log"

    ports:
      - 9125:8125/udp
      - 9092:8092/udp
      - 9094:8094
    restart: unless-stopped
```

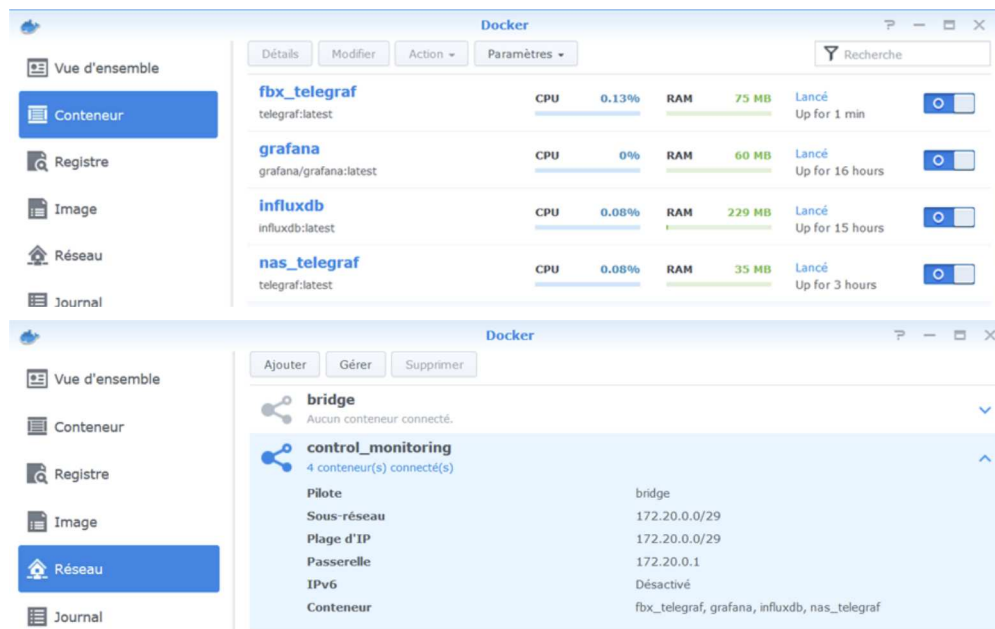
On notera la creation et le mapping de 2 repertoires :

- un repertoire pour le fichier Python et le fichier d'installation de l'utilitaire python "pip" : /usr/local/py
- un repertoire pour d'éventuels logs : /usr/local/log

Création et demarrage du docker fbx_telegraf:

En ssh sur le NAS :

```
docker-compose pull fbx_telegraf
docker-compose up -d fbx_telegraf
```



Via un acces ssh sur le NAS : mise à jour du docker fbx_telegraf

En ssh sur le NAS :

```
docker exec -it fbx_telegraf apt update
docker exec -it fbx_telegraf apt upgrade
docker exec -it fbx_telegraf apt install -y software-properties-common
```

Installation des modules pip et requests

En ssh sur le NAS :

```
docker exec -it fbx_telegraf wget https://bootstrap.pypa.io/get-pip.py
docker exec -it fbx_telegraf python3 get-pip.py --prefix=/usr/local
docker exec -it fbx_telegraf python3 -m pip install requests
```

Script Python :

On place le script Python fbx_telegraf_050.py dans le repertoire /usr/local/py

(ce script a été mis à jour pour mes propres besoins. N'hésitez pas à le reprendre, en particulier en fonction de l'API Freebox si il vous manque des éléments)

Modification du fichier de configuration telegraf :

- Dans la section "Input Plugin" :

```
#####
# INPUT PLUGINS #
#####

#####
# INPUT PLUGINS FREEBOX #
#####

# Read metrics from one or more commands that can output to stdout
[[inputs.exec]]
## Commands array
#
commands = [
"python3 /usr/local/py/freebox_050.py -SPHDIWX"
]

## Timeout for each command to complete.
timeout = "5s"

## Data format to consume.
## Each data format has it's own unique set of configuration options, read
## more about them here:
## https://github.com/influxdata/telegraf/blob/master/docs/DATA_FORMATS_INPUT.md
# data_format = "graphite"
data_format = "influx"
```

- Dans la section "Output Plugins" :

On va envoyer nos données vers une base de données de nom de fbx_database, user fbx_database, password fbx_database
De plus, on va créer nous même cette database => skip_database_creation = true

```
#####
# OUTPUT PLUGINS #
#####

# Configuration for sending metrics to InfluxDB
[[outputs.influxdb]]
## The full HTTP or UDP URL for your InfluxDB instance.
##
## Multiple URLs can be specified for a single cluster, only ONE of the
## urls will be written to each interval.
# urls = ["unix:///var/run/influxdb.sock"]
# urls = ["udp://127.0.0.1:8089"]
urls = ["http://influxdb:8086"]

## The target database for metrics; will be created as needed.
## For UDP url endpoint database needs to be configured on server side.
database = "fbx_telegraf"

## The value of this tag will be used to determine the database. If this
## tag is not set the 'database' option is used as the default.
database_tag = ""

## If true, the database tag will not be added to the metric.
# exclude_database_tag = false

## If true, no CREATE DATABASE queries will be sent. Set to true when using
## Telegraf with a user without permissions to create databases or when the
## database already exists.
skip_database_creation = true

## Name of existing retention policy to write to. Empty string writes to
## the default retention policy. Only takes effect when using HTTP.
retention_policy = ""

## Write consistency (clusters only), can be: "any", "one", "quorum", "all".
## Only takes effect when using HTTP.
write_consistency = "any"

## Timeout for HTTP messages.
timeout = "30s"

## HTTP Basic Auth
username = "fbx_telegraf"
password = "fbx_telegraf"

## HTTP User-Agent
# user_agent = "telegraf"
```

On redemarre le docker fbx_telegraf.

Creation de la database fbx_telegraf sur influxdb

Se connecter en console sur le docker influxdb, puis créer la database :

```
root@influxdb:/# influx -username admin -password admin
Connected to http://localhost:8086 version 1.7.9
InfluxDB shell version: 1.7.9
> show databases
name: databases
----
_internal
nas_telegraf
> create database fbx_telegraf
> use fbx_telegraf
Using database fbx_telegraf
> create user fbx_telegraf with password 'fbx_telegraf'
> grant all on fbx_telegraf to fbx_telegraf
>
> show databases
name: databases
----
_internal
nas_telegraf
fbx_telegraf
> show users
```

```

user admin
-----
admin true
nas_telegraf false
fbx_telegraf false
>

```

On redemarre le docker influxdb pour prise en compte.

Autorisation de l'application sur le Freebox.

Il faut autoriser l'application sur la Freebox.

Pour cela : se placer dans une console fbx_telegraf et aller dans le dossier /usr/local/py

```
root@fbx_telegraf:/usr/local/py# python3 freebox_050.py -r
```

Il faut valider sur l'ecran LCD de la Freebox.

Le nom de l'application codée dans le fichier fbx_telegraf_050.py est : **grafanamonitor**.

Controler sur la Freebox que l'application a bien été acceptée :

Paramètres / Gestion des Accès / Applications



On peut aussi vérifier sur la console fbx_telegraf :

```

root@fbx_telegraf:/usr/local/py# python3 freebox_050.py -s
Status: auth already done
root@fbx_telegraf:/usr/local/py#

root@fbx_telegraf:/usr/local/py# python3 freebox_050.py -r
Already registered, exiting
root@fbx_telegraf:/usr/local/py#

```

Vous trouverez alors un fichiers .credentials dans le répertoire /usr/local/py du docker fbx_telegraf

A partir de là, fbx_telegraf est en mesure de collecter les données sur la Freebox, et de les envoyer sur la base de données fbx_telegraf du docker influxdb.

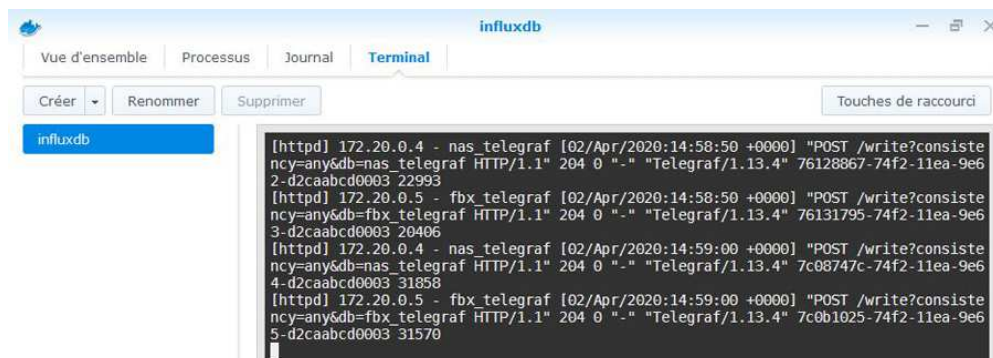
on peut contrôler à la main que les valeurs sont bien accessibles, par exemple :

```

root@fbx_telegraf:/usr/local/py# python3 freebox_050.py -H | grep System
freebox,endpoint=mafreebox.freebox.fr,tag1=System,tag2=NULL,tag3=NULL sys_temp_cpum=64
freebox,endpoint=mafreebox.freebox.fr,tag1=System,tag2=NULL,tag3=NULL sys_temp_sw=54
freebox,endpoint=mafreebox.freebox.fr,tag1=System,tag2=NULL,tag3=NULL sys_uptime_val=2478544
freebox,endpoint=mafreebox.freebox.fr,tag1=System,tag2=NULL,tag3=NULL uptime="28 jours 16 heures 29 minutes 4 secondes"
freebox,endpoint=mafreebox.freebox.fr,tag1=System,tag2=NULL,tag3=NULL firmware_version="4.0.7"
freebox,endpoint=mafreebox.freebox.fr,tag1=System,tag2=NULL,tag3=NULL sys_fan_rpm=2313
freebox,endpoint=mafreebox.freebox.fr,tag1=System,tag2=NULL,tag3=NULL sys_temp_cpub=67
root@fbx_telegraf:/usr/local/py#

```

Dans la console du docker influxdb, on verifie que l'on recoit bien les données de nas_telegraf et fbx_telegraf



Grafana :

Sur grafana, definir une nouvelle source de données :

Name	InfluxDB-1 Freebox	par exemple
URL	http://@NAS:8086	comme pour nas_telegraf, car c'est le même docker influxdb
Access	serveur	
Basic auth	yes	
Basic auth details		
user	fbx_telegraf	

password	fbx_telegraf	
Database	fbx_telegraf	
User	fbx_telegraf	
password	fbx_telegraf	
http method	GET	

On valide (SAVE & TEST) et ce doit être OK.

Il ne reste plus qu'à créer le ou les Dashboard souhaités ! => Ne pas oublier de préciser l'origine des données : InfluxDB-1 Freebox

Pour ma part, j'ai réalisé :

- un Dashboard général (trafic, CPU, status des ports du switch, températures)
- un Dashboard listant les stations actives (globales et Wifi)



Adresse MAC	Équipement / Hostname	Adresse IPv4	Last Activity	Device Type	Active
00:11:00:00:00:00	Freebox	192.168.1.100	Wed Apr 1 11:19:03 2020	Freebox	true
00:26:AB:1B	JetencorePhoto Epson 710	192.168.1.18	Wed Apr 1 11:19:03 2020	printer	true
00:11:32:1F	dy918Blam	192.168.1.171	Wed Apr 1 11:19:32 2020	nas	true
00:01:02:15	Bridge Macvlan	192.168.1.251	Wed Apr 1 11:17:58 2020	other	true
FC:E9:98:26	iPhone	192.168.1.133	Fri Mar 2 19:30:48 2018	smartphone	false
EC:9B:F3:11	Samsung galaxy tab	192.168.1.111	Sun Nov 27 15:09:38 2016	smartphone	false
E4:04:39:F4	tomtom 5200	192.168.1.176	Sat Jan 18 14:02:26 2020	other	false
D8:50:E6:35	grenierw8	192.168.1.116	Wed Apr 1 10:42:19 2020	workstation	false
C8:0A:A9:0D	DESKTOP	192.168.1.161	Fri Sep 29 13:01:59 2017	workstation	false
C0:F4:E6:45	HUAWEI Mate 10	192.168.1.159	Sat Jan 11 01:41:26 2020	workstation	false
BC:85:56:1A	salon pc	192.168.1.120	Sun Apr 23 12:32:13 2017	laptop	false

Adresse MAC	Équipement / Hostname	Adresse IPv4	Last Activity	Device Type	Active
A0:28:ED:5A:1F	Nokia 6.1	192.168.1.172	Wed Apr 1 10:30:37 2020	smartphone	true
44:6D:6C:1D	Galaxy Tab S	192.168.1.125	Wed Apr 1 11:15:19 2020	smartphone	false
10:44:00:14	Honor_9	192.168.1.115	Wed Apr 1 11:18:31 2020	smartphone	true
0C:8F:FF:15	Honor_91	192.168.1.126	Wed Apr 1 11:18:22 2020	smartphone	true

Je peux partager les fichiers de configurations des dashboards (json) si vous le souhaitez.

Resterait à faire : l'API de la Freebox est accessible ici en http. Je n'ai pas réussi par manque de connaissances à passer en https. Si quelqu'un veut s'atteler au sujet ...