

Visualisation de données temporelles sous Grafana : Exemples d'utilisation dans des projets de recherche ou d'administration système

Guillaume BAULIEU - Sylvain FERRIOL - Denis PUGNÈRE
CNRS / IN2P3 / IP2I

Besoins en visualisation de données temporelles et monitoring

- Pour les ASR : Visualisation synthétique d'indicateurs des composantes d'un réseau informatique
 - Comment monitorer les serveurs : la charge (cpu, I/O, nb processus), l'état des serveurs, espace disque... ?
 - Comment monitorer les climats (la température, l'état, la charge...), les onduleurs (charge, l'autonomie..) ?
 - Comment monitorer le réseau (météorologie, les flux ...) ?
- Pour les développeurs :
 - Comment monitorer un système complet d'acquisition (les sous-systèmes) d'une ou de plusieurs expériences ?
 - Comment représenter les données ?
- Pour les expériences :
 - Comment monitorer les systèmes d'acquisition ?
 - Comment monitorer l'acquisition des données d'expérience (qualité des données, flux...) ?
 - Comment offrir différentes vues selon les profils des utilisateurs ?

La visualisation de données temporelles et monitoring ont considérablement évolués ces dernières années

Visualisation de données temporelles et monitoring

- Avant, il y avait (il y a toujours) :
 - **MRTG** (Multi Router Traffic Grapher, Tobias Oetiker), toujours présent : Outil de visualisation en temps réel de trafic de liens réseaux
 - **RRDtool** : outil de gestion de base de données RRD (Round-Robin database) et de représentation graphique de données RRD
 - plein d'autres (gnuplot...), mais c'était généralement un peu fastidieux
- Monitoring (avec ou sans alarme)
 - Il y a Zabbix, nagios, centreon, cacti...

Exemple création / visualisation graphe de données temporelles avec RRDtool

Création du fichier RRD (database round-robin)

```
$ rrdtool create t3-stat.rrd --step=60 \  
DS:R_Alice:GAUGE:3600:U:U \  
DS:R_CMS:GAUGE:3600:U:U \  
DS:R_IPNL:GAUGE:3600:U:U \  
DS:W_Alice:GAUGE:3600:U:U \  
DS:W_CMS:GAUGE:3600:U:U \  
DS:W_IPNL:GAUGE:3600:U:U \  
RRA:LAST:0.5:1:288 \  
RRA:AVERAGE:0.5:1:1440 \  
RRA:AVERAGE:0.5:5:2016 \  
RRA:AVERAGE:0.5:60:720 \  
RRA:AVERAGE:0.5:360:1460
```

```
DS (types) = GAUGE | COUNTER | DERIVE | ABSOLUTE  
RRA (round robin archive) = AVERAGE | MIN | MAX | LAST
```

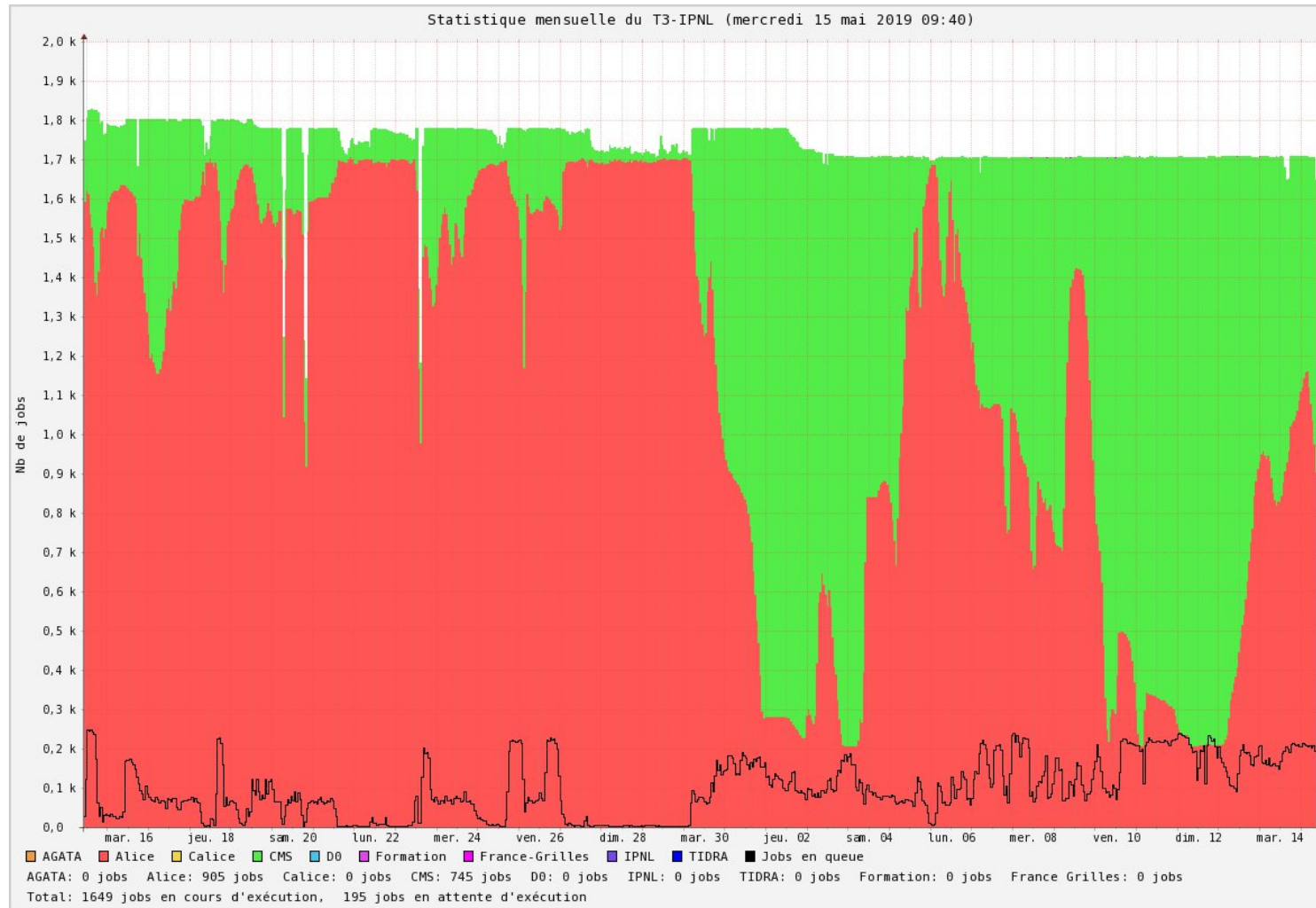
Insertion des données dans le fichier RRD

```
$ rrdtool update /tmp/t3-stat.rrd \  
N:10:31:35:60:52:29:56:70:44:7:6:4
```

Création du graphe à partir des données du fichier RRD

```
rrdtool graph $webpath/t3-stat-month.png --end now --start end-1m -a PNG \  
--title="Statistique mensuelle du T3-IPNL (`date '+%A %d %B %Y %R'`)" \  
--vertical-label "Nb de jobs" --width 1100 --height 700 \  
DEF:R_Alice=t3-stat.rrd:R_Alice:AVERAGE \  
DEF:R_CMS=t3-stat.rrd:R_CMS:AVERAGE \  
DEF:R_IPNL=t3-stat.rrd:R_IPNL:AVERAGE \  
DEF:W_Alice=t3-stat.rrd:W_Alice:AVERAGE \  
DEF:W_CMS=t3-stat.rrd:W_CMS:AVERAGE \  
DEF:W_IPNL=t3-stat.rrd:W_IPNL:AVERAGE \  
CDEF:R_total=R_Alice,R_CMS,R_IPNL,+,+,+,+,+ \  
CDEF:W_total=W_Alice,W_CMS,W_IPNL,+,+,+,+,+ \  
AREA:R_Alice#FF5555:Alice:STACK \  
AREA:R_CMS#54EC48:CMS:STACK \  
AREA:R_IPNL#7648EC:IPNL:STACK \  
LINE1:W_total#000000:"Jobs en queue" \  
COMMENT:"\s" \  
COMMENT:"\s" \  
COMMENT:"\s" \  
GPRINT:R_Alice:LAST:"Alice\ : %.01f jobs" \  
GPRINT:R_CMS:LAST:"CMS\ : %.01f jobs" \  
GPRINT:R_IPNL:LAST:"IPNL\ : %.01f jobs" \  
COMMENT:"\s" \  
COMMENT:"\s" \  
COMMENT:"\s" \  
GPRINT:R_total:LAST:"Total\ : %.01f jobs en cours d'exécution," \  
GPRINT:W_total:LAST:"%.01f jobs en attente d'exécution \1"
```

Ça donne un graphique dans ce genre...

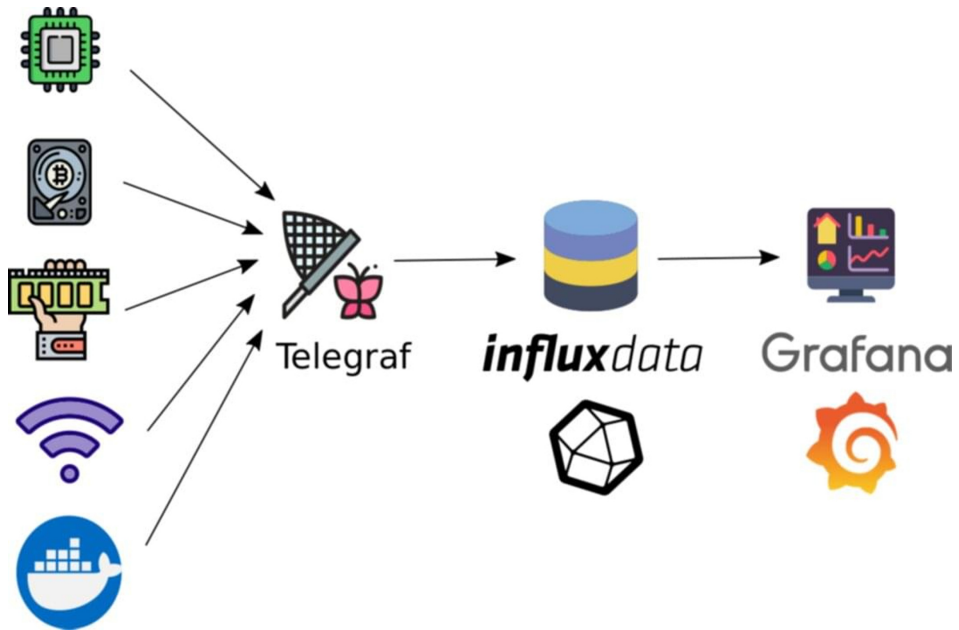


Librairies et piles logicielles orientées « séries temporelles »

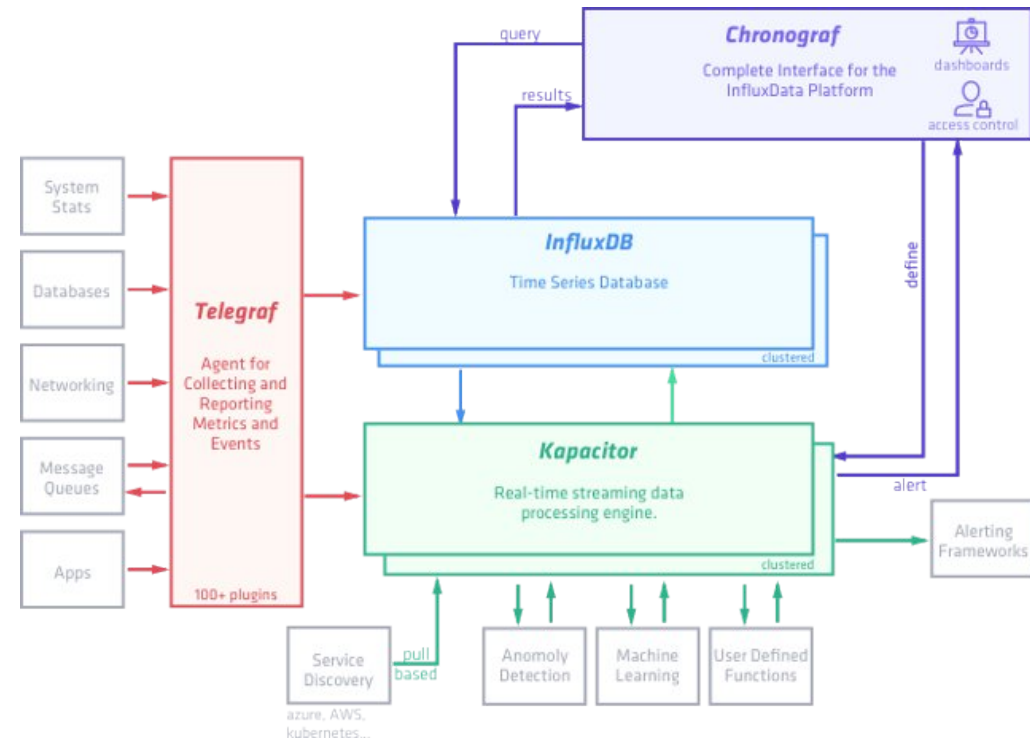
- Maintenant il y a aussi toutes les librairies [1] (javascript, python...) D3js, matplotlib, bokeh...
- Mais aussi des logiciels orientés « séries temporelles » :
 - La **collecte** des métriques: Telegraf, netdata, collectd...
 - Le **stockage** des informations temporelles (bases de données « timeseries ») : RRDTOol, InfluxDB, Prometheus, TimeScaleTB, OpenTSDB, QuestDB, Graphite...
 - La **visualisation** des données : Grafana, Prometheus, Graphite, Chronograf ...
 - Les **alertes** : Prometheus, Kapacitor ...
- Les piles logicielles orientés « séries temporelles » rassemblent généralement : collecte + stockage + visualisation (+ alerte).
- Exemple :
 - (Telegraf | netdata) => InfluxDB => Grafana
 - Telegraf => InfluxDB => Chronograf => Kapacitor

[1] <https://pyviz.org>

Pile **TIG** : Telegraf -> InfluxDB -> Grafana



Pile **TICK** : Telegraf -> InfluxDB -> Chronograf -> Kapacitor



Avantages de la pile TIG :

- configuration plus simple : ajout métriques (dans le collecteur, pas de déclaration à faire dans InfluxDB)
- dashboards dans Grafana sont conviviaux (drag, drop, copie json), auto-complétion à la création de la requête

Installation et configuration **Telegraf** :

- collecte (mode push ou pull) et envoi des métriques
- métriques = systèmes et applicatives

```
# yum|apt install telegraf
```

```
# visualisation de la liste des sources de données reconnues :
```

```
$ telegraf --input-list
```

```
quelques exemples : cpu, disk, net, kernel, hddtemp, nginx, ceph, haproxy, kubernetes, postfix, mysql, postgresql, proxmox ...
```

```
# visualisation de la liste des formats de données exportés en destination :
```

```
$ telegraf --output-list
```

```
quelques exemples : elasticsearch, influxdb, http, graphite, kafka, syslog, opentsdb ...
```

```
# Génération du fichier de configuration
```

```
$ telegraf --input-filter cpu:disk:diskio:kernel:mem:processes:swap:system:kernel_vmstat:net:net_response:netstat:sensors:temp:nstat \  
--output-filter influxdb config > telegraf.conf.temp
```

```
éditer /etc/telegraf/telegraf.conf.temp
```

```
[[outputs.influxdb]]
```

```
urls = ["http://influx_server:8086"]
```

```
database = "telegraf"
```

```
username = "my_experience"
```

```
password = "ldZmXVBix1FiEocO2xJFKQXWzYefMENJWKtigqARAMIS"
```


Installation et configuration **InfluxDB**:

- écoute par défaut sur le port 8086
- est accessible en local ou à distance
- une instance InfluxDB peut contenir plusieurs « databases »

```
# yum|apt install influxdb
```

```
$ influx
```

```
> create database telegraf
```

```
> use telegraf
```

```
> CREATE USER my_experience WITH PASSWORD "ldZmXVBix1FiEocO2xJFKQXWzYEfMENJWKtigqARAMIS"
```

```
> GRANT ALL ON telegraf TO my_experience
```

```
# toutes les données exportées depuis telegraf
```

```
> SHOW MEASUREMENTS
```

```
# tous les champs de chaque donnée
```

```
> SHOW FIELD KEYS
```

```
# ajuster la politique de rétention des données (par défaut infinie = remplit votre disque)
```

```
> SHOW RETENTION POLICIES
```

```
RETENTION POLICY autogen = INFINITE
```

```
# on modifie la politique de rétention pour la réduire à 3 mois
```

```
> CREATE RETENTION POLICY "3m" ON telegraf DURATION 12w REPLICATION 1 DEFAULT
```

```
> DROP RETENTION POLICY "autogen" ON telegraf
```

```
> exit
```

Installation et configuration **Grafana** :

- Ajouter la data source
- concevoir ou importer (et éventuellement modifier) les dashboards
- protéger l'accès aux dashboards (admin, users, droits..)

```
# yum|apt install grafana
```

Configuration **Grafana** :

- écoute par défaut sur le port 3000
- => Add data source : source type influxdb sur **influx_server**

Il existe une grande quantité de dashboards déjà développés

<https://grafana.com/grafana/dashboards/?dataSource=influxdb>

Les Dashboards :

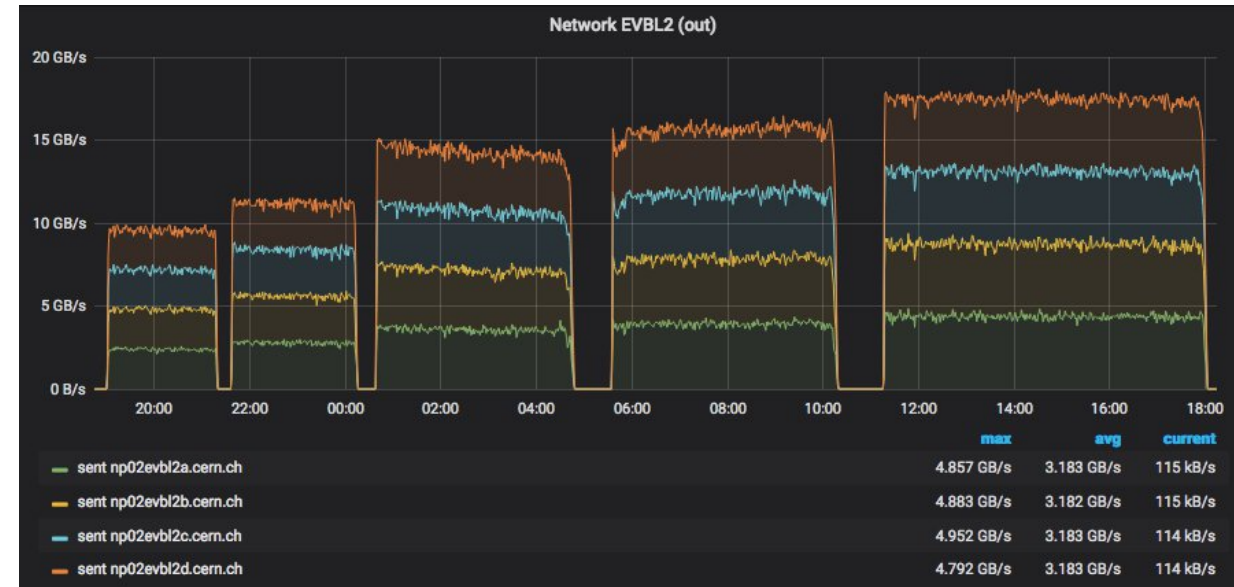
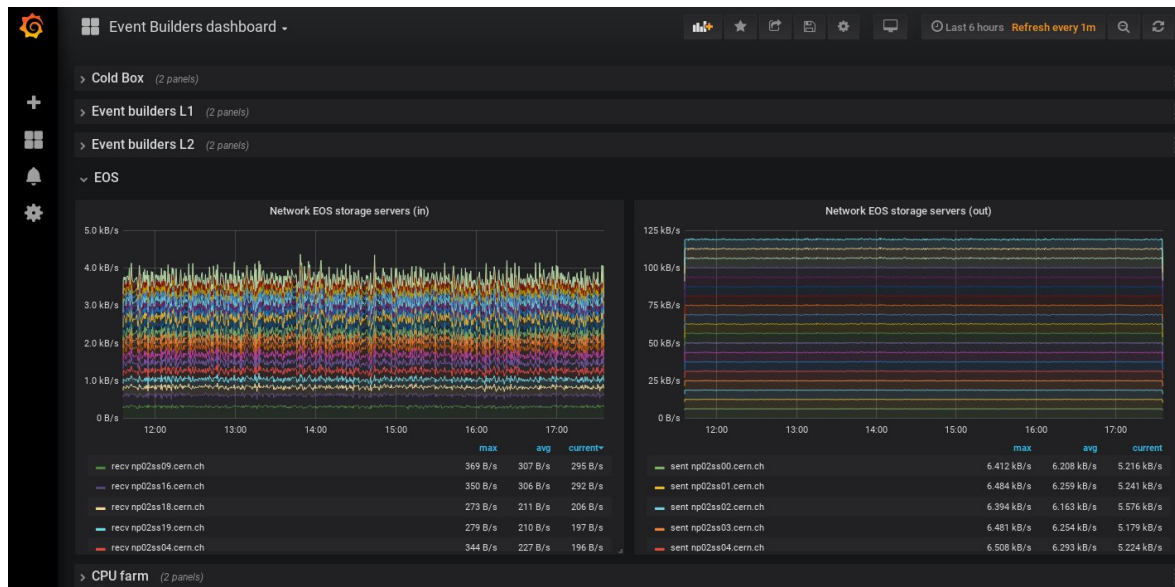
- sont en .json
- Peuvent être créés, importés, modifiés, exportés..
- Contiennent des Panels, qui regroupent 1 ou plusieurs graphes
- Les graphes peuvent être déplacés, dimensionnés à la souris
- pour créer les graphes, la création des requêtes est assistée (auto-completion...)

Construction de dashboards
pour les différents types d'utilisateurs :

ASR
Sysadmin



Pour les chercheurs de l'expérience / shifters



Grafana : installation par Docker

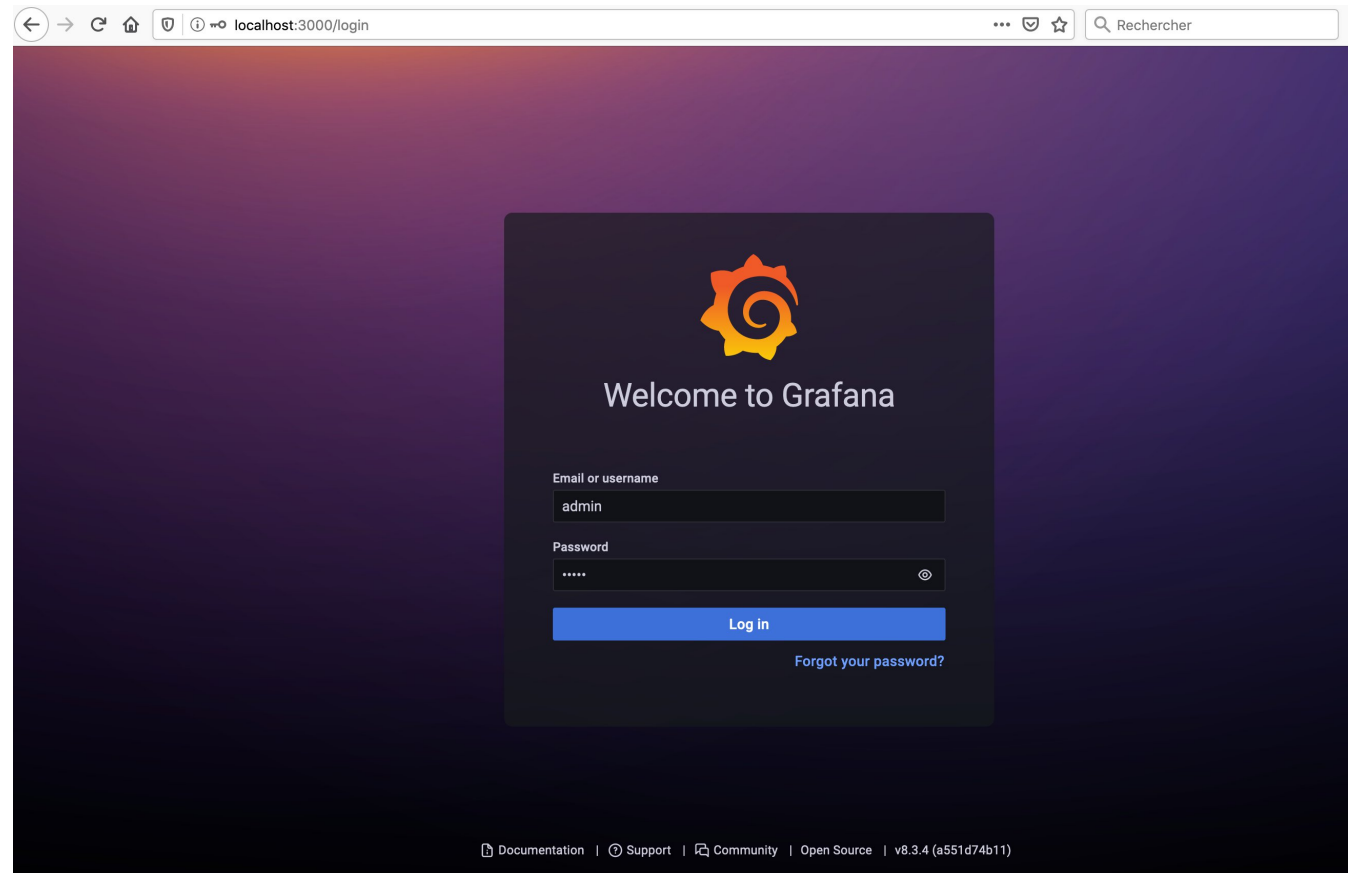
Lancement en une seule commande !

```
docker run -d --name=grafana --rm -p 3000:3000 grafana/grafana
```

—> Téléchargement d'une image de la dernière version de Grafana et lancement en locale

Grafana est maintenant accessible
depuis un navigateur à l'adresse
http://localhost:3000

Arrêt du service avec
docker stop grafana



Grafana : installation par Docker

Possibilité de modifier la configuration par défaut avec un fichier docker-compose.yml :

```
version : '3.7'
services :
  grafana-web :
    image : grafana/grafana:8.1.6
    container_name : grafana
    environment :
      - « GF_SERVER_DOMAIN=grafana.mydomain.fr »
      - « GF_SERVER_ROOT_URL=http://grafana.mydomain.fr »
      - « GF_SMTP_ENABLED=true »
      - « GF_SMTP_HOST=smtp.mydomain.fr:25 »
      - « GF_SMTP_FROM_ADDRESS=no-reply@mydomain.fr »
      - « GF_INSTALL_PLUGINS=grafana-piechart-panel,natel-plotly-panel »
    volumes :
      - grafana-vol : /var/lib/grafana
    ports :
      - 3000:3000
    restart : always

volumes :
  grafana-vol :
    driver : local
```

Grafana : installation par Docker

Possibilité de modifier la configuration par défaut avec un fichier docker-compose.yml :

```
version : '3.7'
services :
  grafana-web :
    image : grafana/grafana:8.1.6
    container_name : grafana
    environment :
      - « GF_SERVER_DOMAIN=grafana.mydomain.fr »
      - « GF_SERVER_ROOT_URL=http://grafana.mydomain.fr »
      - « GF_SMTP_ENABLED=true »
      - « GF_SMTP_HOST=smtp.mydomain.fr:25 »
      - « GF_SMTP_FROM_ADDRESS=no-reply@mydomain.fr »
      - « GF_INSTALL_PLUGINS=grafana-piechart-panel,natel-plotly-panel »
    volumes :
      - grafana-vol : /var/lib/grafana
    ports :
      - 3000:3000
    restart : always

volumes :
  grafana-vol :
    driver : local
```

image à utiliser

Grafana : installation par Docker

Possibilité de modifier la configuration par défaut avec un fichier docker-compose.yml :

```
version : '3.7'
services :
  grafana-web :
    image : grafana/grafana:8.1.6
    container_name : grafana
    environment :
      - « GF_SERVER_DOMAIN=grafana.mydomain.fr »
      - « GF_SERVER_ROOT_URL=http://grafana.mydomain.fr »
      - « GF_SMTP_ENABLED=true »
      - « GF_SMTP_HOST=smtp.mydomain.fr:25 »
      - « GF_SMTP_FROM_ADDRESS=no-reply@mydomain.fr »
      - « GF_INSTALL_PLUGINS=grafana-piechart-panel,natel-plotly-panel »
    volumes :
      - grafana-vol : /var/lib/grafana
    ports :
      - 3000:3000
    restart : always

volumes :
  grafana-vol :
    driver : local
```

Variables d'environnement qui vont sur-écrire la configuration par défaut



Grafana : installation par Docker

Possibilité de modifier la configuration par défaut avec un fichier docker-compose.yml :

```
version : '3.7'
services :
  grafana-web :
    image : grafana/grafana:8.1.6
    container_name : grafana
    environment :
      - « GF_SERVER_DOMAIN=grafana.mydomain.fr »
      - « GF_SERVER_ROOT_URL=http://grafana.mydomain.fr »
      - « GF_SMTP_ENABLED=true »
      - « GF_SMTP_HOST=smtp.mydomain.fr:25 »
      - « GF_SMTP_FROM_ADDRESS=no-reply@mydomain.fr »
      - « GF_INSTALL_PLUGINS=grafana-piechart-panel,natel-plotly-panel »
    volumes :
      - grafana-vol : /var/lib/grafana
    ports :
      - 3000:3000
    restart : always

volumes :
  grafana-vol :
    driver : local
```

Enregistrement de la base de données interne sur la machine hôte (possibilité de sauvegarde)

Grafana : installation par Docker

Possibilité de modifier la configuration par défaut avec un fichier docker-compose.yml :

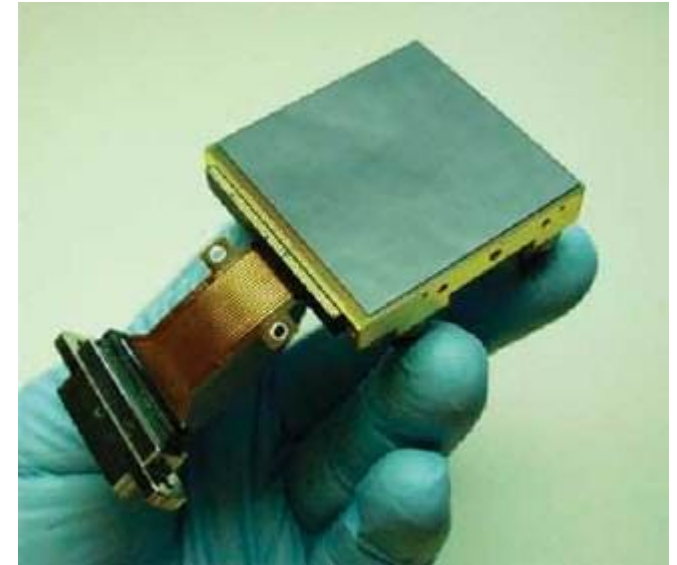
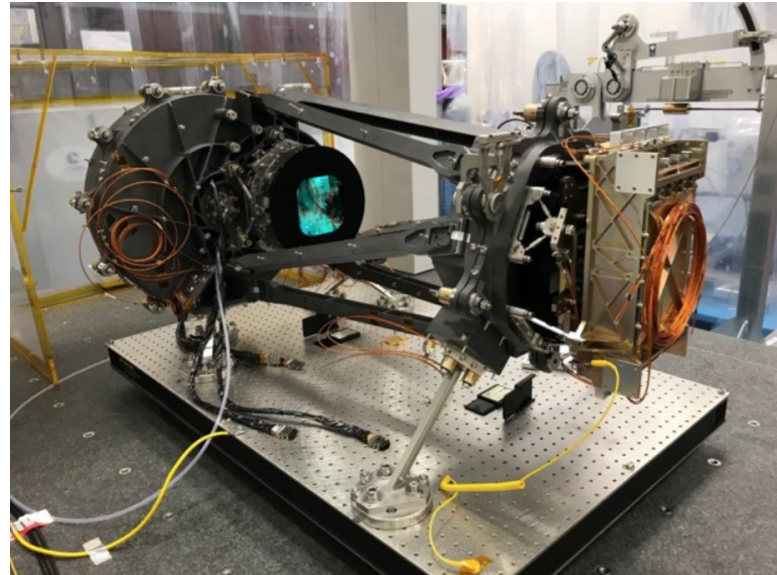
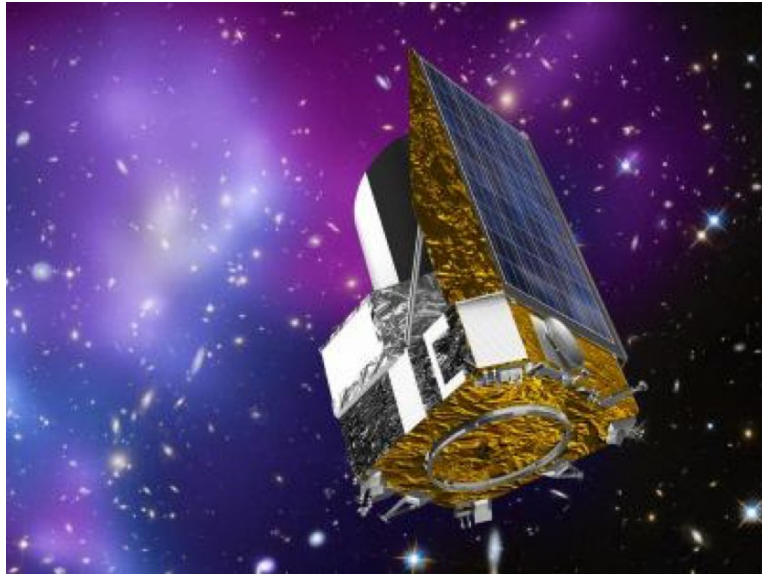
```
version : '3.7'
services :
  grafana-web :
    image : grafana/grafana:8.1.6
    container_name : grafana
    environment :
      - « GF_SERVER_DOMAIN=grafana.mydomain.fr »
      - « GF_SERVER_ROOT_URL=http://grafana.mydomain.fr »
      - « GF_SMTP_ENABLED=true »
      - « GF_SMTP_HOST=smtp.mydomain.fr:25 »
      - « GF_SMTP_FROM_ADDRESS=no-reply@mydomain.fr »
      - « GF_INSTALL_PLUGINS=grafana-piechart-panel,natel-plotly-panel »
    volumes :
      - grafana-vol : /var/lib/grafana
    ports :
      - 3000:3000
    restart : always

volumes :
  grafana-vol :
    driver : local
```

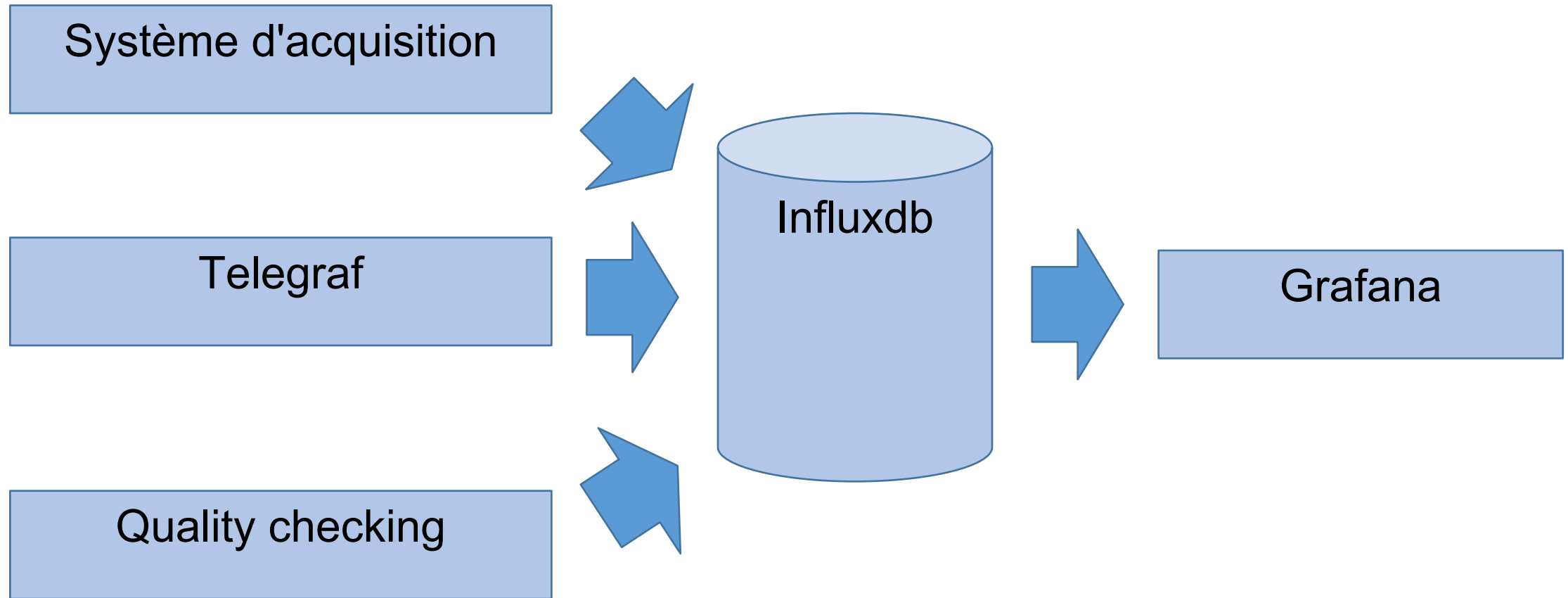
Lancement du service avec la commande :

docker-compose up -d

Projet EUCLID : caractérisation des capteurs infrarouges



Chaîne de monitoring



Grafana : dashboard_1 pour le shifteur

Begin End



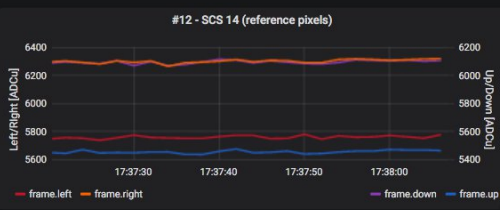
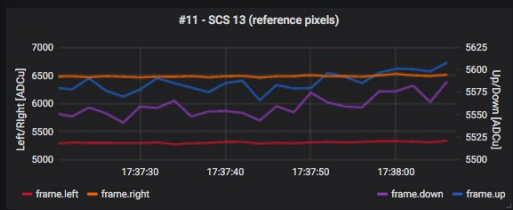
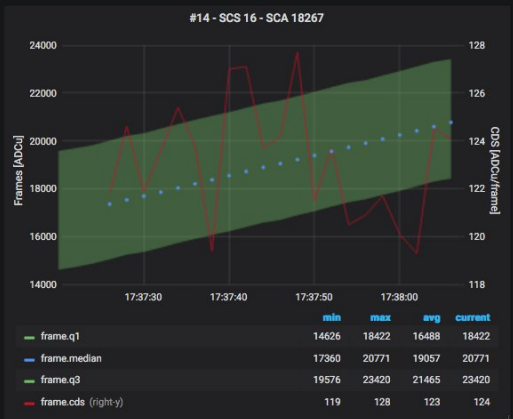
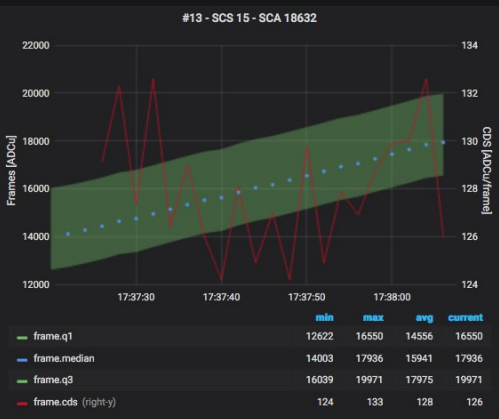
Run status

20191123_16h38_05: run.frame

Run epoch
1574526903.460991

Run Name
ff:ff_led1_ir2_photoj_r1

| | |
|------------------------|----------------------------------|
| Run number in the list | Number of runs in the current... |
| 1 | 17 |
| Cycle | MACC mode |
| 1 | [4L, 16L, 4L] |
| Iteration | Sequence id |
| 1 | seq_1 |



Sequence
1

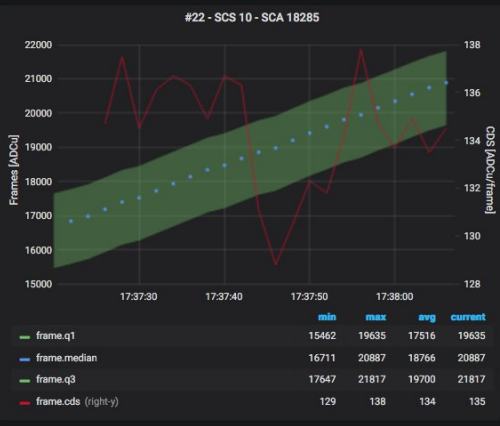
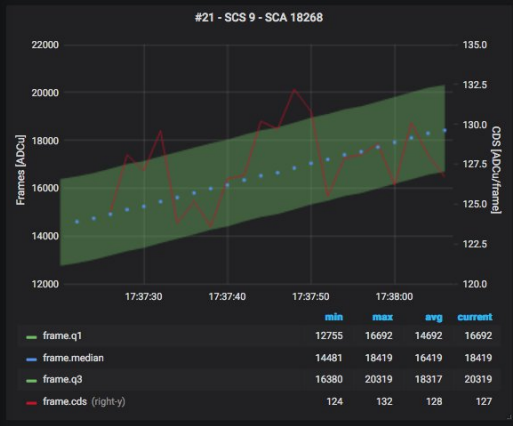
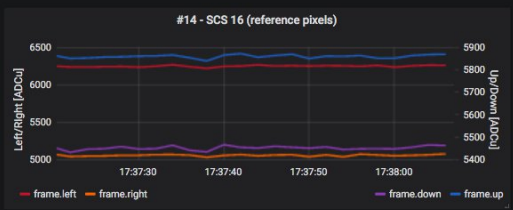
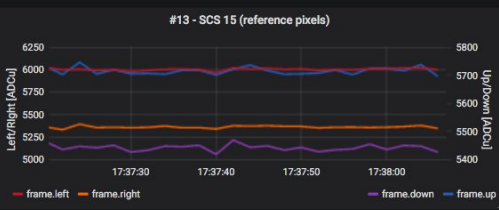
Ramp
1

Progress
71%

Acquisition live buffer queue
0

Total number of frames in the...
76

Number of frames acquired i...
54

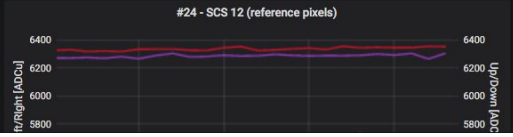
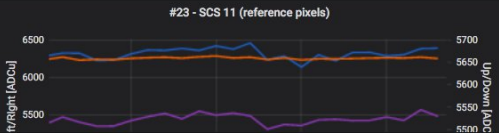
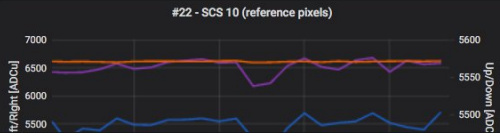
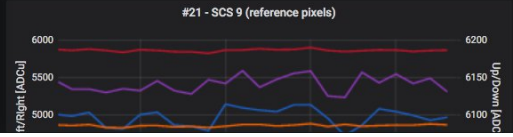


led
1

flux [ADC_]
151

flux [e-/s]
200

| | | |
|-----------|-----------|---------------|
| led 1 CUR | led 1 PWM | led 1 status |
| N/A | N/A | 970 nm |
| led 2 CUR | led 2 PWM | led 2 status |
| N/A | N/A | off |
| led 3 CUR | led 3 PWM | led 3 status |
| N/A | N/A | off |
| led 4 CUR | led 4 PWM | led 4 status |
| N/A | N/A | off |



Grafana : dashboard_2 pour le shifteur



Grafana : monitoring de la sauvegarde

Number of copi... **11406**

Total number of... **11406**

Status **copying**

Progress **100%**

Number of copi... **4872**

Total number of... **14624**

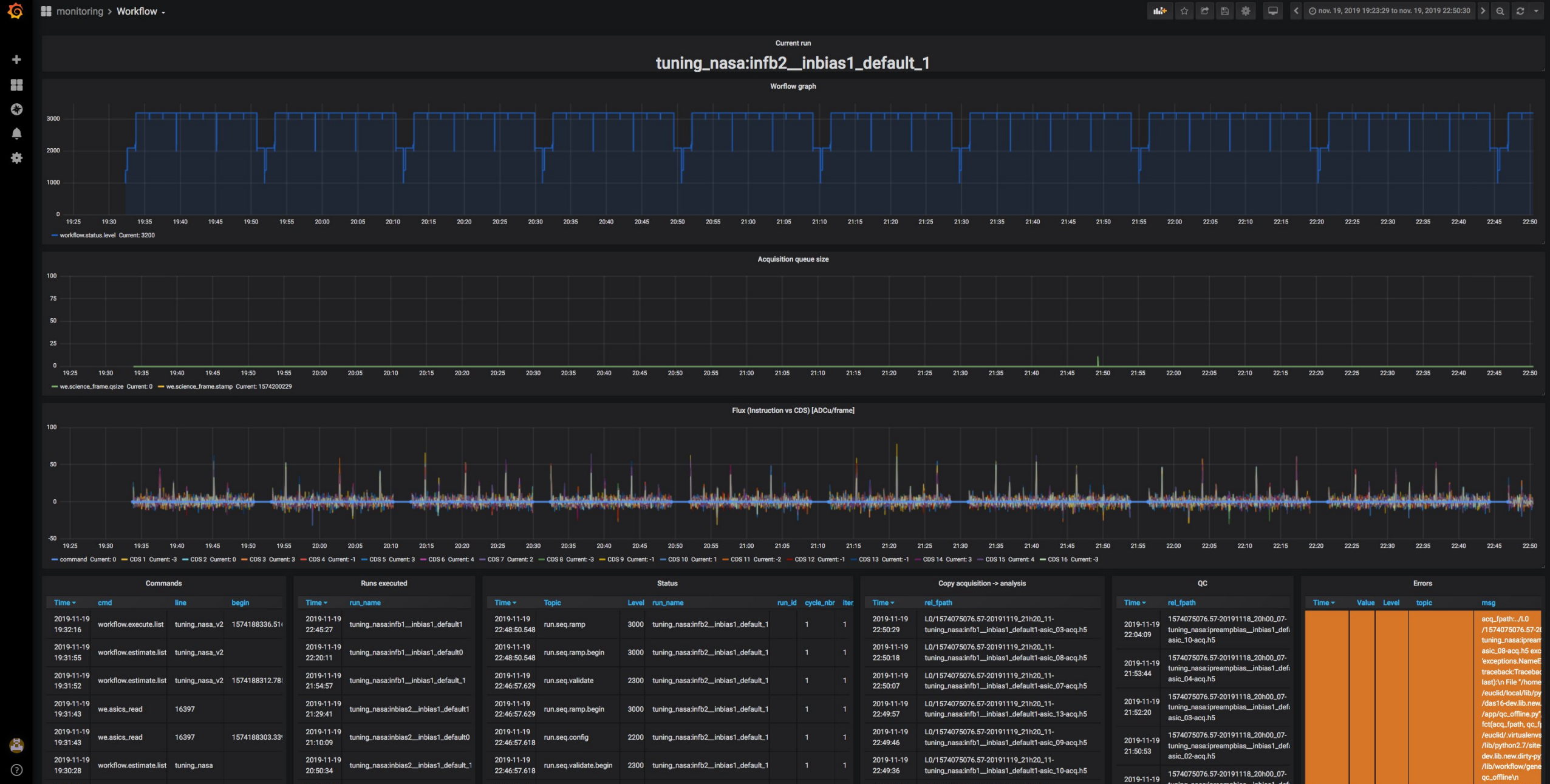
Status **copying**

Progress **33%**

| Copied files | | | | | | | |
|---------------------|---|---------------------|------------|----------------|------------------|------------|-----|
| Time | rel_fpath | copy_end_estimation | copy_speed | n_copied_files | n_copied_seconds | n_synchros | n_1 |
| 2019-11-30 22:45:04 | TV1/OT /log/1574847853.63-1575140309.98 | 0:00:00 | 654261864 | 11406 | 25037 | 755 | 1 |
| 2019-11-30 22:45:04 | TV1/OT /L0/1574847853.63-20191130_18h58_29-la:la_led3_irl46_photoy_r15-asic_14-acq.h5 | 0:00:00 | 654240811 | 11405 | 25037 | 755 | 1 |
| 2019-11-30 22:43:52 | TV1/OT /L0/1574847853.63-20191130_18h58_29-la:la_led3_irl46_photoy_r15-asic_12-acq.h5 | 0:01:14 | 654181917 | 11404 | 24966 | 755 | 1 |
| 2019-11-30 22:42:39 | TV1/OT /L0/1574847853.63-20191130_18h58_29-la:la_led3_irl46_photoy_r15-asic_01-acq.h5 | 0:02:28 | 654120941 | 11403 | 24894 | 755 | 1 |
| 2019-11-30 22:41:29 | TV1/OT /L0/1574847853.63-20191130_18h58_29-la:la_led3_irl46_photoy_r15-asic_15-acq.h5 | 0:03:42 | 654016271 | 11402 | 24824 | 755 | 1 |
| 2019-11-30 22:40:18 | TV1/OT /L0/1574847853.63-20191130_18h58_29-la:la_led3_irl46_photoy_r15-asic_07-acq.h5 | 0:04:56 | 653922991 | 11401 | 24753 | 755 | 1 |
| 2019-11-30 22:39:05 | TV1/OT /L0/1574847853.63-20191130_18h58_29-la:la_led3_irl46_photoy_r15-asic_16-acq.h5 | 0:06:10 | 653868439 | 11400 | 24681 | 755 | 1 |
| 2019-11-30 22:37:54 | TV1/OT /L0/1574847853.63-20191130_18h58_29-la:la_led3_irl46_photoy_r15-asic_06-acq.h5 | 0:07:24 | 653754351 | 11399 | 24611 | 755 | 1 |
| | TV1/OT | | | | | | |

| Backup CC | | | | | | |
|---------------------|--|---------------------|------------|----------------|------------------|------------|
| Time | rel_fpath | copy_end_estimation | copy_speed | n_copied_files | n_copied_seconds | n_synchros |
| 2019-11-30 23:57:22 | PT/L1/1574075076.57-20191119_19h11_26-tuning_nasa:ipreampbias_inbias1_default1-asic_16-qc.h5 | 5 days, 0:13:14 | 96098074 | 4872 | 196950 | 1 |
| 2019-11-30 23:56:59 | PT/setup/markury/Activate_AckMode.glf | 5 days, 0:13:38 | 96097915 | 4871 | 196927 | 1 |
| 2019-11-30 23:56:57 | OT/L1/1574847853.63-20191127_12h23_32-nl:nl_led3_irl1_spectro_r1-asic_10-qc.h5 | 5 days, 0:13:36 | 96098471 | 4870 | 196926 | 1 |
| 2019-11-30 23:56:52 | PT/L1/1574075076.57-20191118_20h19_40-tuning_nasa:ipreampbias_inbias1_default1-asic_15-qc.h5 | 5 days, 0:13:37 | 96098773 | 4869 | 196922 | 1 |
| 2019-11-30 23:56:30 | PT/L0/1574075076.57-20191121_20h41_21-dark:dark_spectro_r30-asic_04-acq.h5 | 5 days, 0:14:04 | 96097956 | 4868 | 196901 | 1 |
| 2019-11-30 23:40:24 | PT/setup/mcd/268-045-060.mcd | 5 days, 0:35:41 | 96047358 | 4867 | 195936 | 1 |
| 2019-11-30 23:40:22 | RT/setup/.git/objects /9d/ad59fa5c15d2a6fb2a8fdda9ee37dca7051ba2 | 5 days, 0:35:38 | 96047874 | 4866 | 195935 | 1 |
| 2019-11-30 23:40:21 | PT/setup/.git/objects /65/d6dd3c838ddefc68dc7913a507737fef8ffbf | 5 days, 0:35:37 | 96048266 | 4865 | 195934 | 1 |
| 2019-11-30 23:40:19 | PT/L1/1574075076.57-20191118_20h58_49-tuning_nasa:inbias2_inbias1_default0-asic_05-qc.h5 | 5 days, 0:35:35 | 96048639 | 4864 | 195933 | 1 |
| 2019-11-30 23:39:57 | OT/qc/maps/CPPM/269-072-056.h5 | 5 days, 0:36:02 | 96047808 | 4863 | 195912 | 1 |
| 2019-11-30 23:39:47 | OT/L0/1574720220.67-20191127_10h17_31-dark:dark_spectro_r30-asic_12-acq.h5 | 5 days, 0:36:12 | 96047592 | 4862 | 195903 | 1 |
| 2019-11-30 23:35:12 | OT/setup/.git/objects /43/d277fbda329d28253aba19493925344462f4f4 | 5 days, 0:43:25 | 96023380 | 4861 | 195629 | 1 |
| 2019-11-30 | PT/L0/1574075076.57-20191119_22h36_00- | | | | | |

Grafana : monitoring de l'acquisition



Grafana : dashboard en cours de fabrication



Query NISP monitoring Add Query Query Inspector ?

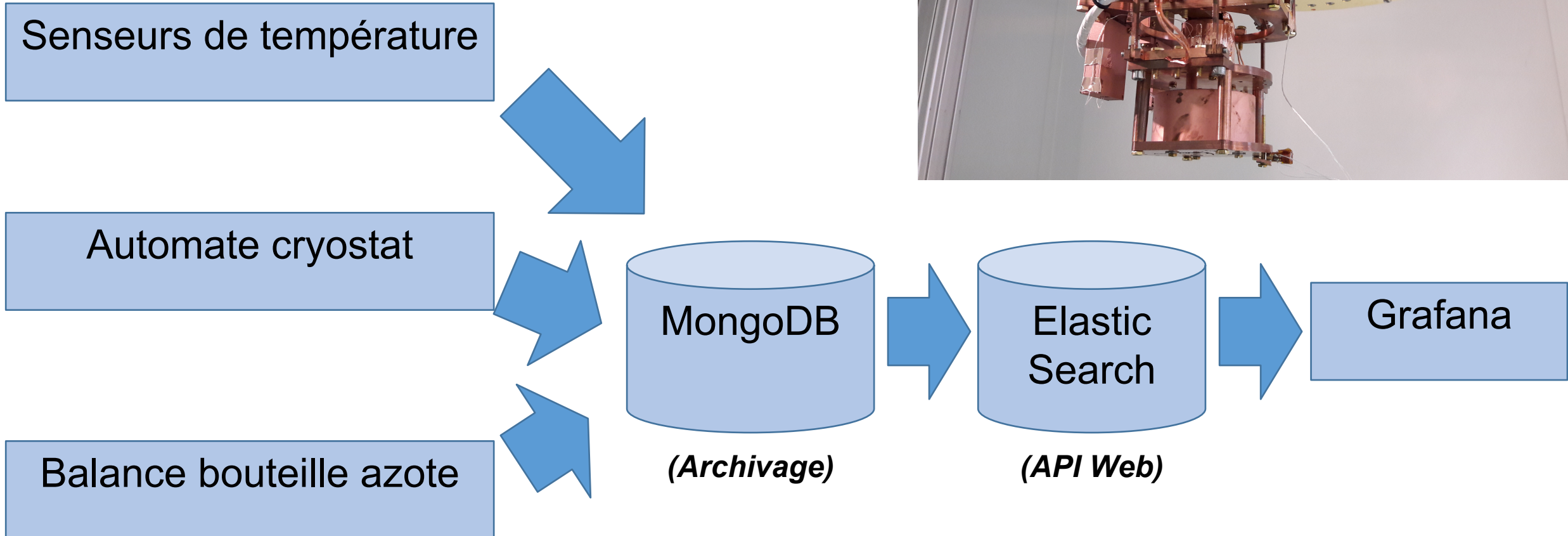
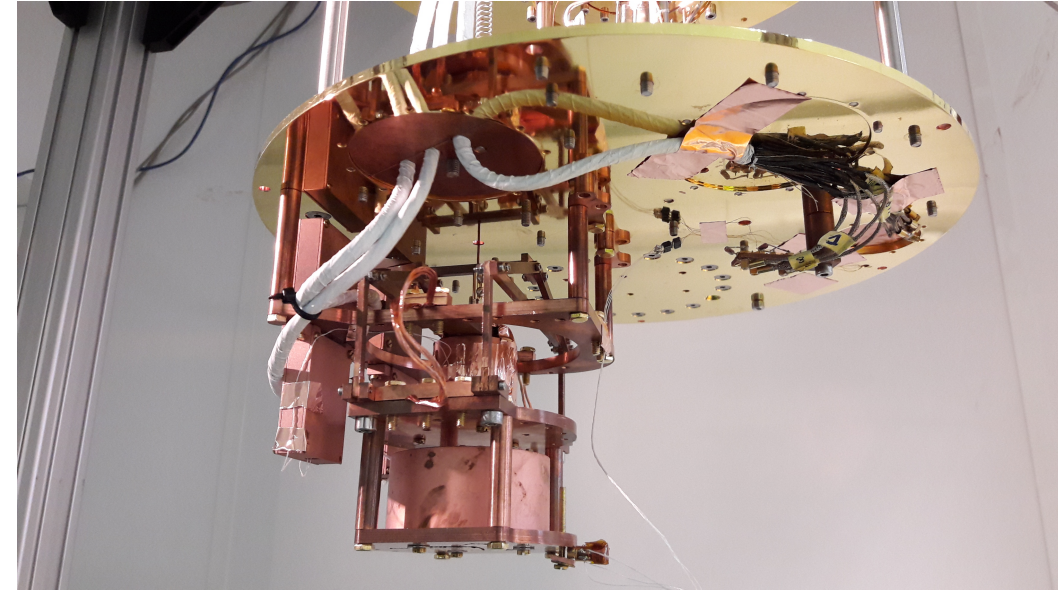
```
SELECT mean("q1") AS "q1", mean("median") AS "median", mean("q3") AS "q3", mean("cds") AS "cds" FROM "frame" WHERE ("scs_nbr" = '$scs_nbr' AND "run_name" = '$run_name' AND "run_time" = $run_time) AND $timeFilter GROUP BY time($__interval) fill(null)
```

FORMAT AS Time series ALIAS BY Naming pattern

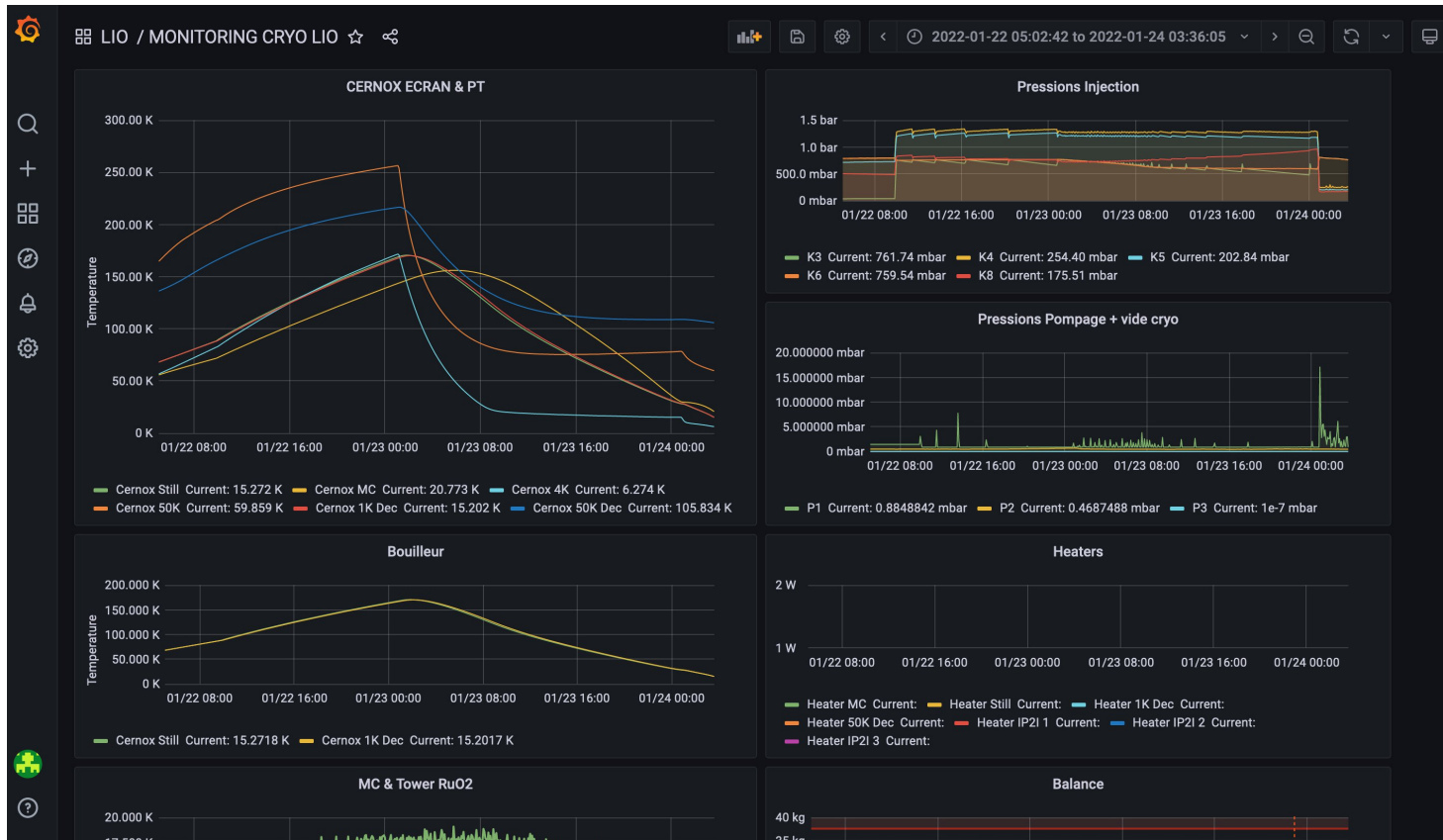
Min time interval 1.4548s Relative time 1h Time shift 1h



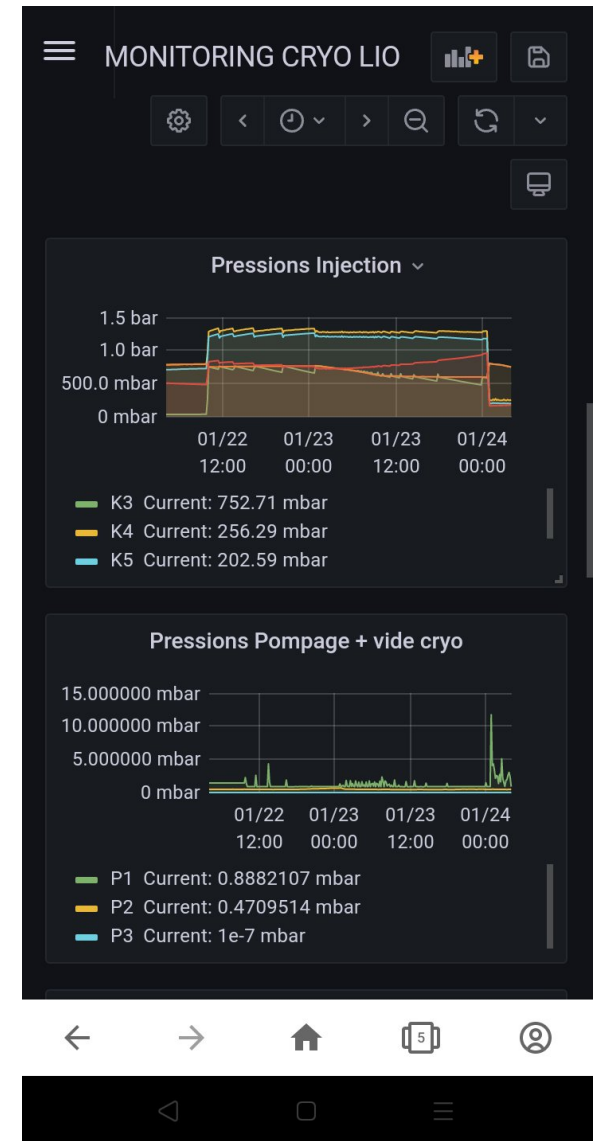
Projet Ricochet : surveillance du fonctionnement d'un cryostat



Grafana : exemple de dashboard de surveillance



Affichage sur machine de bureau



Affichage sur téléphone

Bilan Grafana : Avantages

- Facilité d'installation
- Multitude de sources de données
- Multi-user
- Fabrication des dashboards en moins de deux jours !!!
- Adaptation automatique de l'affichage
- Grande qualité d'affichage des graphs
- Puissance de la sélection temporelle commune sur toutes les graphs d'un dashboard
- Beaucoup de réglages disponibles
- La sauvegarde de dashboard se résume à un fichier json (utilisation facile de dashboards existants)
- puissance des annotations pour chaque point

Bilan Grafana : Inconvénients

- pas de zoom en Y
- pas de graphs 2D (ondelettes)
- Que pour des données temporelles