

Plan

1. Introduction et historique
2. Schéma explicatif général
3. Quelques mots sur les metrics
4. Exemple de configuration de scrapping & envoi sur un serveur victoriaMetrics
5. Un retour d'expérience sur les dashboards & alertes
6. Évolution et alternatives

Grafana

- Projet OpenSource créé en 2014
- Créateur: Torkel Ödegaard
- Licence: `GNU AfferoGeneralPublic V3`
- Permet de visualiser trois types de données:
 - les metrics - les TimeSeries Database (Prometheus, VictoriaMetrics)
 - les logs - agrégateur de log (Loki)
 - les traces (Zipkin ou Jaeger)
- Il peut générer des alertes selon les seuils que vous avez configurés

Dispose d'un catalogue de dashboards déjà
présent

<https://grafana.com/grafana/dashboards/>

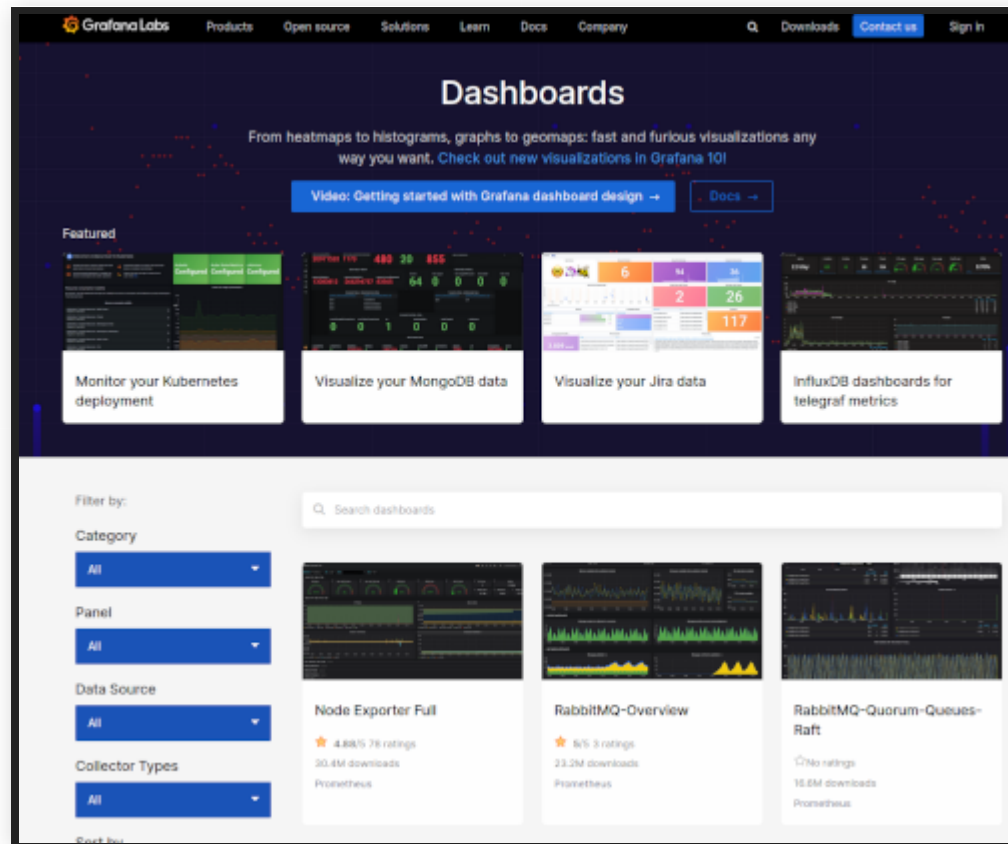


Schéma de flux - exemple Prometheus

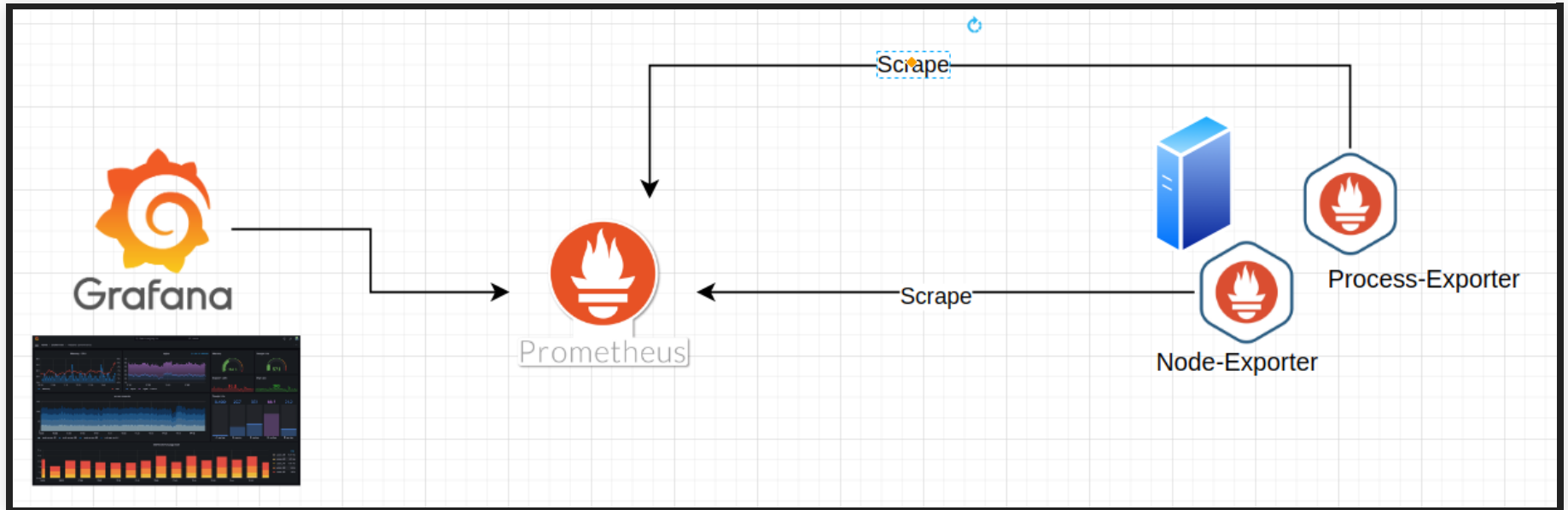


Schéma de flux - exemple VictoriaMetrics

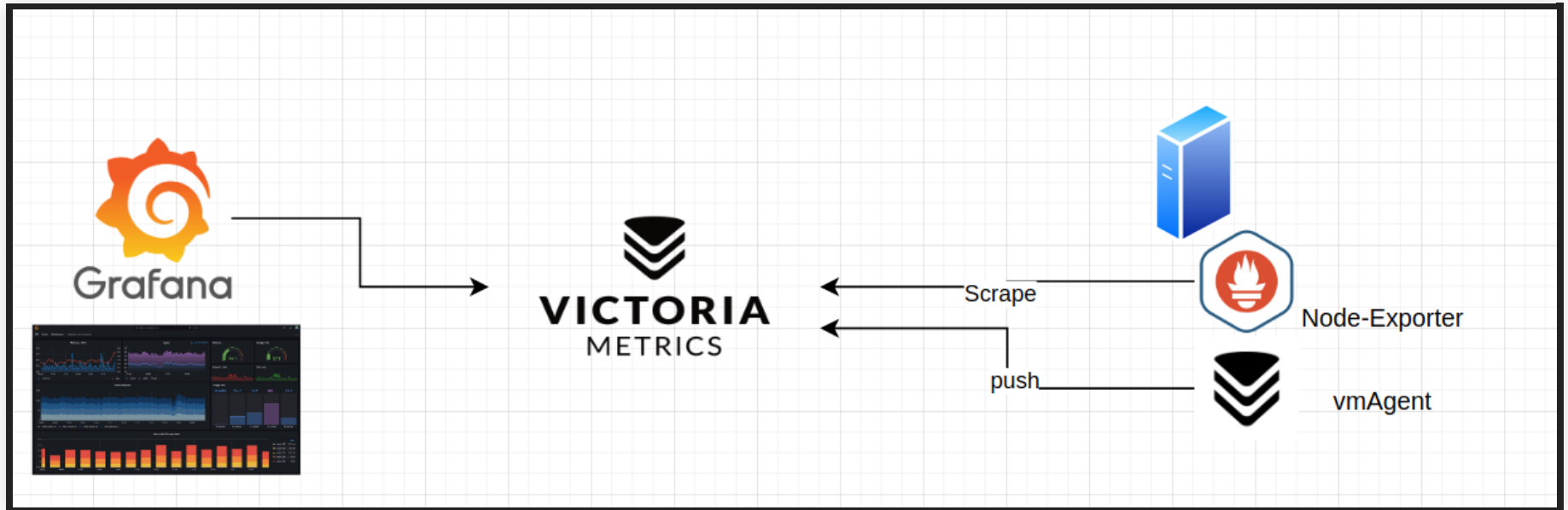
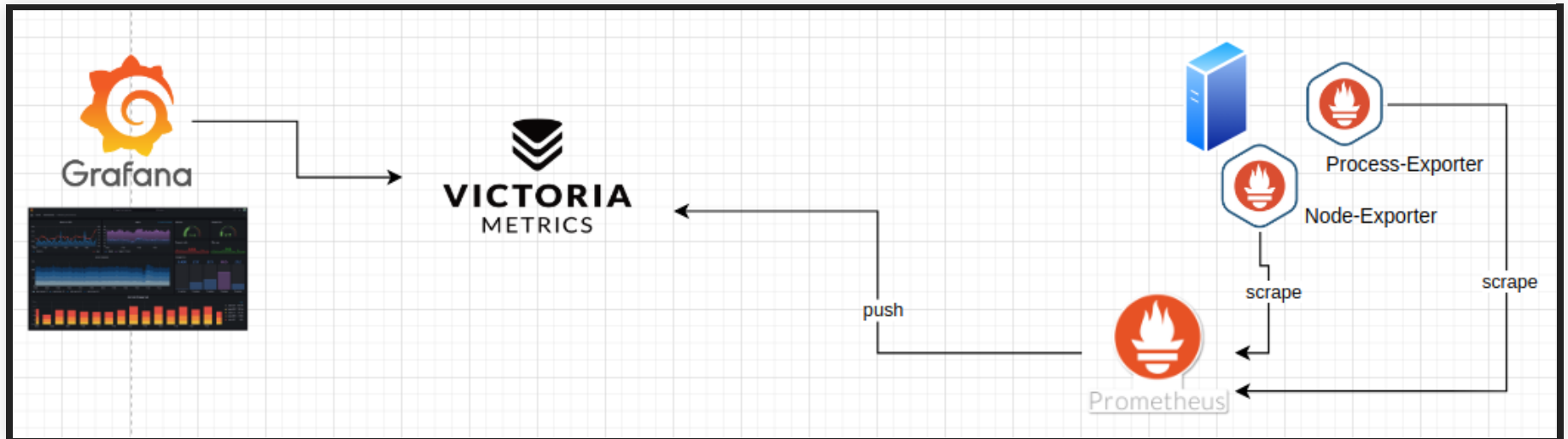


Schéma de flux - exemple PLMlab



Quelques mots sur les metrics

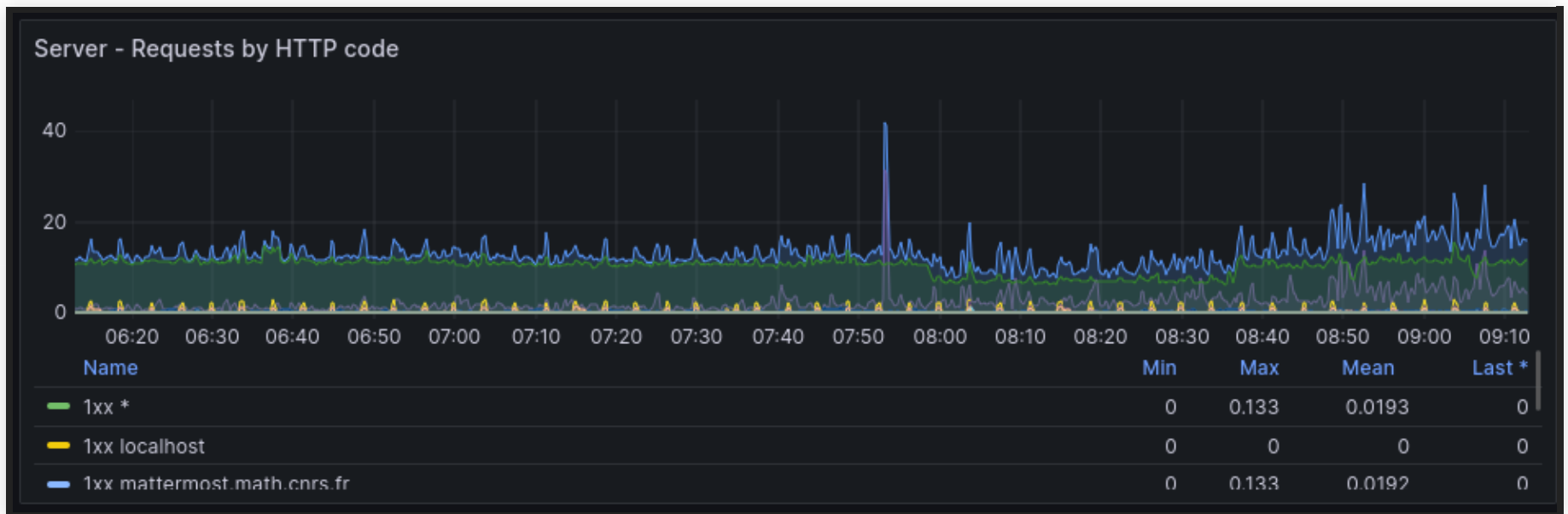
Différence entre counter & gauge

Il est important d'avoir ces deux différences car les fonctions et les types de graphes utilisé seront différents

Counter: Valeur incrémentée

Gauge: valeur instantanée

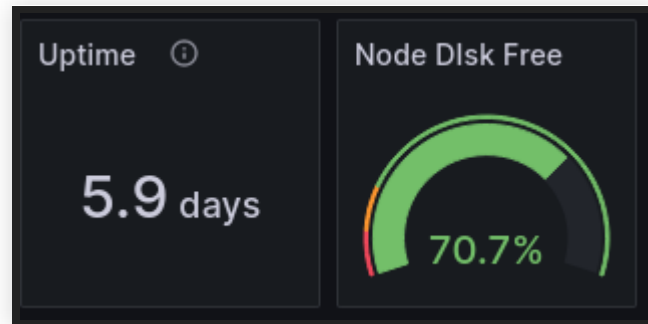
Pour le counter:
fonctions: Rate (mesure une moyenne) ou
Increase (mesure une augmentation)
Graphe: TimeSeries (graphe évolutif)



Pour le Gauge:

fonctions: Delta (différence entre deux timestamps) ou Deriv (augmentation par seconde)

Graphe: Gauge, tableau, texte



Pour plus d'infos, faire une requête CURL sur un
“exporter” :

```
# HELP node_timex_pps_jitter_total Pulse per second count of  
# TYPE node_timex_pps_jitter_total counter  
node_timex_pps_jitter_total 0  
# HELP node_timex_pps_shift_seconds Pulse per second interval  
# TYPE node_timex_pps_shift_seconds gauge  
node_timex_pps_shift_seconds 0
```

Exemple de configuration de scrapping & envoi sur un serveur prometheus

Sur plmlab

Configuration process-exporter

```
1 process_names:
2   - exe:
3     - /opt/gitlab/embedded/bin/gitlab-pages
4     - /opt/gitlab/embedded/bin/mattermost
5     - /opt/gitlab/embedded/bin/postgres
6     - /sbin/iscsid
7   - comm:
8     - redis-server
9   - exe:
10    - ruby
11    cmdline:
12    - '.*mail_room.*'
```

Exemple de configuration d'un prometheus pour scrapper les données:

```
prometheus['scrape_configs'] = [  
  {  
    'job_name': 'process-exporter',  
    'static_configs' => [  
      'targets' => ['localhost:9256'],  
    ]  
  }  
]
```

Exemple d'envois de donnée sur un autre serveur

:

```
prometheus['remote_write'] = [  
  {  
    url: 'https://*****.math.cnrs.fr/api/v1/write',  
    ....  
  }  
]
```

CR - Retour d'expérience

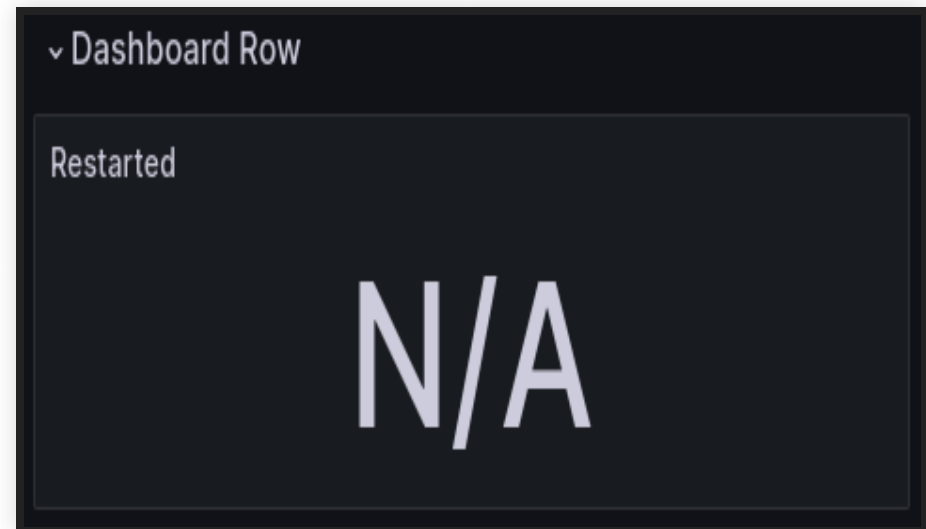
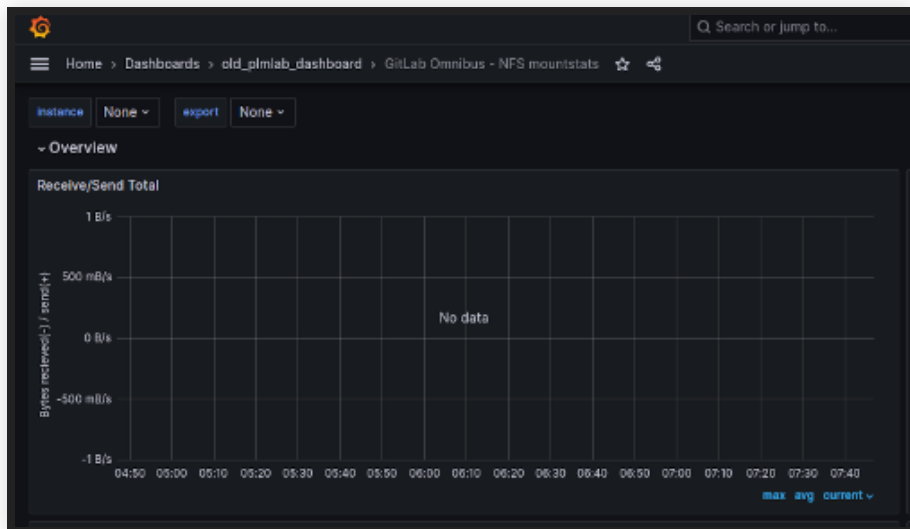
Dashboards

- Il faut savoir ce que vous voulez monitorer
 - Mettre en avant des données qui vous parle
 - que voulons-nous monitorer ?
 - quels services ?
 - Volumes ? (montages ISCSI)
 - quelles activités ? (git push, issues, merge request, login ...)
 - est-ce que l'on a les metrics ? ==> sinon rajouter ou créer des exporters (process-exporter pour gitlab-page)

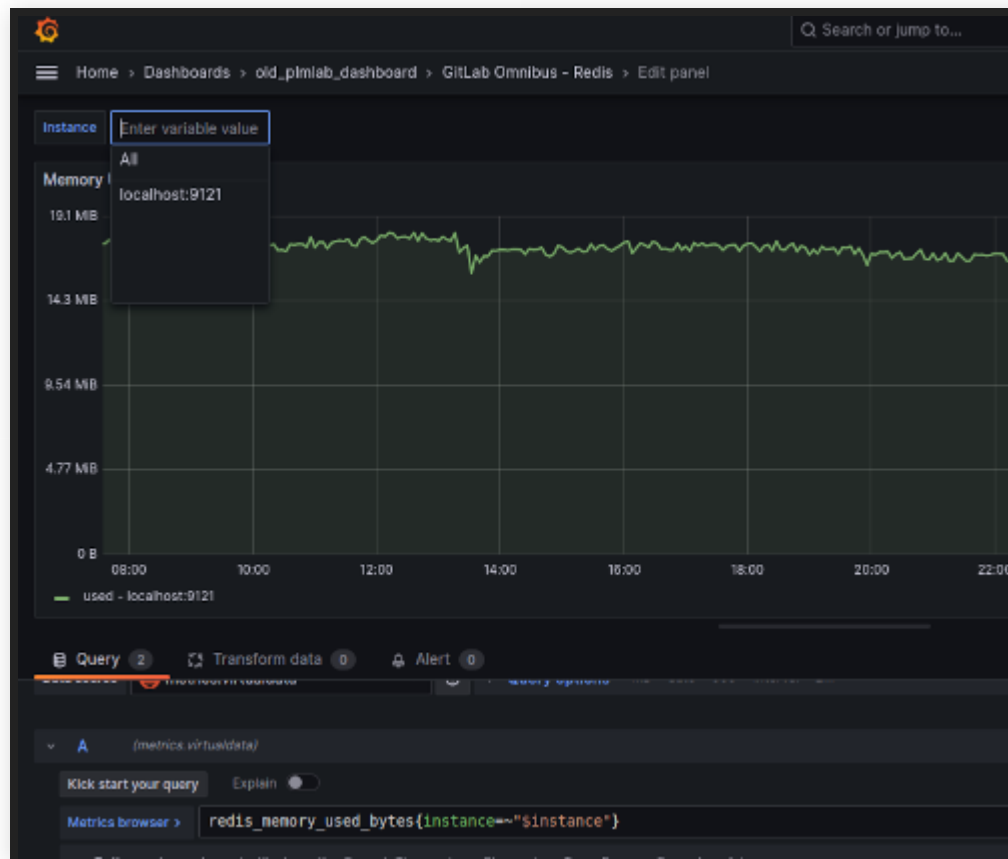
- # Beaucoup de choix

glgo.json	Export of 2024-02-09T16:08:00+00:00	1 month ago
glgo3a-overview.json	Export of 2024-03-09T00:08:04+00:00	2 days ago
global-nfs-stats.json	Export of 2022-02-09T22:49:49+00:00	2 years ago
go-processes.json	Export of 2022-06-15T16:08:19+00:00	1 year ago
google-cloud-storage.json	Export of 2022-03-23T08:05:58+00:00	1 year ago
google-cloud-storage3a-artifact-cdn-detail.json	Export of 2024-03-09T00:08:04+00:00	2 days ago
google-cloud-storage3a-overview.json	Export of 2024-03-09T00:08:04+00:00	2 days ago
haproxy-service.json	Export of 2022-02-09T22:49:49+00:00	2 years ago
haproxy-status-deprecated.json	Export of 2022-02-09T22:49:49+00:00	2 years ago
haproxy.json	Export of 2023-09-28T08:06:03+00:00	5 months ago
host-stats.json	Export of 2023-04-29T00:04:45+00:00	10 months ago
hosts.json	Export of 2022-02-09T22:49:49+00:00	2 years ago
imported-gitlab-dev-web-status.json	Export of 2022-02-09T22:49:49+00:00	2 years ago
imported-sidekiq-jobs.json	Export of 2022-02-09T22:49:49+00:00	2 years ago
imported-staging-deploy.json	Export of 2022-02-09T22:49:49+00:00	2 years ago
imported-time.json	Export of 2022-02-09T22:49:49+00:00	2 years ago
importers.json	Export of 2022-03-23T08:05:58+00:00	1 year ago

- Une étude des dashboards peut aider mais des fois pas suffisant ...
 - Pour Pimlab, souvent les metrics n'existaient plus



- relabelling est important
 - sinon on se perd et ne sait plus faire la différence entre localhost & localhost



exemple de relabelling en ruby :

```
1 prometheus['scrape_configs'] = [  
2   {  
3     'job_name': 'node-exporter',  
4     'static_configs' => [  
5       'targets' => ['plmlab.gricad:9100'],  
6     ],  
7     'relabel_configs' => [  
8       'target_label': 'instance',  
9       'replacement': 'plmlab.gricad'  
10    ],  
11  },  
12  {  
13    'job_name': 'process-exporter',  
14    'static_configs' => [  
15    'targets' => ['plmlab.gricad:9256']
```

Exemple de dashboard de plmlab:

<https://console.supervision.mathrice-plm.virtualdata.cloud.math.cnrs.fr/d/pV1-fisSU/plmlab-overview?orgId=1&refresh=5s>

CR - Retour d'expérience

Alertes

- il existe différents outils pour la génération d'alertes:
 - prometheus
 - grafana


Pourquoi grafana-alerte ?

- gestion en mode gitops (dashboards et alertes versionner au même endroit)
- Je me base sur les metrics pour faire les alertes et définir les seuils.


Exemple d'une alerte:

- 1er exemple: `up{job="gitlab-rails"}`
- 2ème exemple:
`namedprocess_namegroup_oldest_start_time_`
`groupname="mattermost"}`

Critères d'évaluation :

All rules in the selected group are evaluated every 30s. 

Pending period
Period in which an alert rule can be in breach of the condition until the alert rule fires.

Pause evaluation 

▼ **Configure no data and error handling**

Alert state if no data or all values are null

Alert state if execution error or timeout

Canaux d'alertes:

Name	Type	Grafana Alertmanager	Other Alertmanagers
DingDing	dingding	Supported	N/A
Discord	discord	Supported	N/A
Email	email	Supported	Supported
Google Chat	googlechat	Supported	N/A
Kafka	kafka	Supported	N/A
Line	line	Supported	N/A
Microsoft Teams	teams	Supported	Supported
Opsgenie	opsgenie	Supported	Supported
Pagerduty	pagerduty	Supported	Supported
Prometheus Alertmanager	prometheus-alertmanager	Supported	N/A
Pushover	pushover	Supported	Supported
Sensu Go	sensugo	Supported	N/A
Slack	slack	Supported	Supported
Telegram	telegram	Supported	N/A
Threema	threema	Supported	N/A
VictorOps	victorops	Supported	Supported
Webhook	webhook	Supported	Supported (different format)
Cisco Webex Teams	webex	Supported	Supported
WeCom	wecom	Supported	N/A
Zenduty	webhook	Supported	N/A

Exemple avec PLMLAB

<https://console.supervision.mathrice-plm.virtualdata.cloud.math.cnrs.fr/alerting/list>

Alternative et futur ?

- ELK ?
 - trop lourd ?
 - bien qu'il soit similaire, lui se spécialise sur les logs
- openTelemetry
 - se présente comme l'avenir
 - permet de gérer les trois axes (metrics, logs, traces)
 - toujours dépendant de serveurs pour stocker les données (prometheus, loki)

Liens

- [Introduction à openTelemetry](#)
- [Bonnes pratiques Dashboard grafana](#)
- [Liste des dashboards gitlab](#)
- [Prometheus - grafanas - tutoriels](#)