



# Monitoring avec prometheus et grafana

## Retour d'expérience

Nicolas Gibelin : UMS Gricad  
[Nicolas.Gibelin@univ-grenoble-alpes.fr](mailto:Nicolas.Gibelin@univ-grenoble-alpes.fr)

Version 1.2 (Rev : 109) - 20180424

# Plan

- 1 Introduction
- 2 Prometheus
- 3 Grafana
- 4 Conclusion
- 5 Référence

Mi2S

## Mi2S

- Nagios + Centreon
- Netmet / Netmat
- Nfsen
- Cacti

Mi2S => Gricad : salles du datacentre UGA

- **Cacti** : pour pdus, ups (snmp)... , mais problèmes (graphes, snmp, templates, rrd, ...)

## Mi2S => Gricad : salles du datacentre UGA

- **Cacti** : pour pdus, ups (snmp)... , mais problèmes (graphes, snmp, templates, rrd, ...)
- ajout **Influxdb** + **Grafana**

## Mi2S => Gricad : salles du datacentre UGA

- **Cacti** : pour pdus, ups (snmp)... , mais problèmes (graphes, snmp, templates, rrd, ...)
- ajout **Influxdb** + **Grafana**

## Gricad

## Mi2S => Gricad : salles du datacentre UGA

- **Cacti** : pour pdus, ups (snmp)... , mais problèmes (graphes, snmp, templates, rrd, ...)
- ajout **Influxdb** + **Grafana**

## Gricad

- **Prometheus**



# Pourquoi Prometheus ?

## Test/lecture dernières versions

- Cacti
- Nagios et dérivés
- ...

# Pourquoi Prometheus ?

## Test/lecture dernières versions

- Cacti
- Nagios et dérivés
- ...

## Veille

- Zabbix
- Shinken
- Prometheus
- ...

# Pourquoi Prometheus ?

## Prometheus: à première vue

- Simple
  - Mise en oeuvre
  - utilisation
  - mise à jour
- Efficace
- Plein d'[Exporters](#) (sondes)
- Grosse communauté.
- Intégré / Supporté ([liste complète](#)):
  - [Gitlab](#) : prometheus\_monitoring['enable']
  - [Ceph](#) : ceph mgr module enable prometheus
  - [Openstack](#) : prometheus-openstack-exporter

Mimer la conf snmp de Cacti DC Imag

Mimer la conf snmp de Cacti DC Imag

En combien de temps, sachant:

## Mimer la conf snmp de Cacti DC Imag

En combien de temps, sachant:

- 3 types de pdus
- 150 Pdus
- 1 Ups
- Tout snmp

## Mimer la conf snmp de Cacti DC Imag

En combien de temps, sachant:

- 3 types de pdus
- 150 Pdus
- 1 Ups
- Tout snmp

Et en bonus

- Grapher dans [Grafana](#)
- Grapher plus (versus [Influxdb](#))

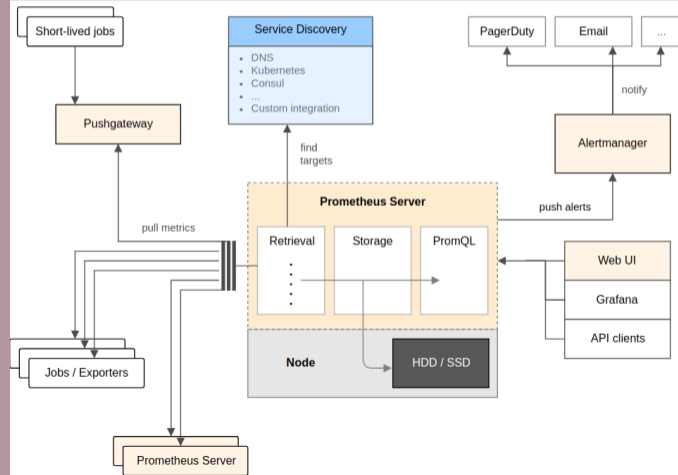
# Plan

- 1 Introduction
- 2 Prometheus**
- 3 Grafana
- 4 Conclusion
- 5 Référence



# Prometheus

## Architecture de Prometheus



## Description

- Code écrit en [Go](#)
- Base de donnée de séries temporelles ([time series](#))
- Collecte à intervalle régulier: **exporters**
- Fonctionne par '**job**'
- Similarité des données = stockage par double delta
- Langage de requête : [PromQL](#)
- Évaluation de définitions d'alerte
- Évaluation de définitions
- API Web, serveur web

## Quelques exporters

- Officiels (sur le gitHub) :
  - snmp\_exporter
  - job\_exporter
  - mysqld\_exporter
  - graphite\_exporter
- Autres (liste ici) :
  - apache\_exporter
  - ceph\_exporter
  - minecraft\_exporter

# Installation du serveur

## Packages

Votre distribution doit probablement les avoir.

## Sinon : Bash

```
adduser --no-create-home --disabled-login prometheus
export PVERSION=2.2.0
mkdir -p /home/softwarewares/prometheus/packages
mkdir -p /home/data/prometheus
cd /home/softwarewares/prometheus/packages
wget "https://github.com/prometheus/prometheus/releases/download/${PVERSION}/
      prometheus-${PVERSION}.linux-amd64.tar.gz"
tar zxvf prometheus-${PVERSION}.linux-amd64.tar.gz
cd ../
ln -fs packages/prometheus-${PVERSION}/ prometheus
```



# Configuration

```
/etc/prometheus/prometheus.yml
```

```
global:
  scrape_interval:     15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: ...

rule_files:
  # recording rules
  # Alerting rules

alerting:
  alertmanagers:
  - static_configs:
    - targets:
      - localhost:9093
```

# Lancement du serveur

```
/etc/systemd/system/prometheus.service
```

```
[Unit]
Description=Prometheus server
After=network.target

[Service]
WorkingDirectory=/home/softwarewares/prometheus/prometheus
ExecStart=/home/softwarewares/prometheus/prometheus/prometheus \
  --config.file=/etc/prometheus/prometheus.yml \
  --storage.tsdb.path=/home/data/prometheus \
  --storage.tsdb.retention=400d
# -alertmanager.url=http://localhost:9093/alert-manager \
User=prometheus

[Install]
WantedBy=multi-user.target
```

- Autant que vous voulez
- Sur différentes machines
- Installation idem serveur
- Utilise des fichiers de MIB
- Fourni un générateur de configuration

# Snmp\_exporter: configuration mibs

```
/etc/prometheus/generator.yml
```

```
modules:  
  if_mib:  
    walk: [sysUpTime, interfaces, ifXTable]  
  
# APC pdus  
# Model Number: AP8853  
ap8853:  
  version: 1  
  walk:  
    - sysName # Pdu Name  
    - 1.3.6.1.4.1.318.1.1.26.6.3.1.5 # Amperage Phase  
    - 1.3.6.1.4.1.318.1.1.26.4.3.1.5 # Power  
    - 1.3.6.1.4.1.318.1.1.26.4.3.1.9 # Energy
```





# Snmp\_exporter: generation de la configuration

```
bash
```

```
generator generate
```

# Snmp\_exporter: generation de la configuration

```
/etc/prometheus/snmp.yml
```

```
ap8853:  
  walk:  
  - 1.3.6.1.2.1.1.5  
  - 1.3.6.1.4.1.318.1.1.26.4.3.1.5  
  metrics:  
  - name: sysName  
    oid: 1.3.6.1.2.1.1.5  
    type: DisplayString  
    help: An administratively-assigned name ...  
  - name: rPDU2DeviceStatusPower  
    oid: 1.3.6.1.4.1.318.1.1.26.4.3.1.5  
    type: gauge  
    help: The power consumption of the Rack PDU load ...  
  indexes:  
  - labelname: rPDU2DeviceStatusIndex  
    type: gauge
```

# Configuration Prometheus

```
/etc/prometheus/prometheus.yml
```

```
- job_name: 'snmp-imag-pdu-ap8853 '  
  file_sd_configs:  
    - files: [ "/etc/prometheus/snmp/imag-pdu-ap8853.yml" ]  
  
  metrics_path: /snmp  
  
  params:  
    module: [ap8853]
```

# Configuration Prometheus

```
/etc/prometheus/snmp/imag-pdu-ap8853.yml
```

```
- targets:  
  - pdu-1.test.fr  
  - pdu-2.test.fr  
  labels:  
    site: 'Grenoble'  
    alley: 'Z'  
    rack: '1'  
  
- targets:  
  - pdu-3.test.fr  
  - pdu-4.test.fr  
  labels:  
    site: 'Grenoble'  
    alley: 'Z'  
    rack: '2'
```

# Attention : bugs

Mauvaise implémentation des MIB sur certains équipements :

## snmp.yml : faux mais qui marche

- name: upsPhaseOutputPercentPower  
oid: 1.3.6.1.4.1.318.1.1.1.9.3.3.1.16.1  
type: gauge  
help: The percentage of the UPS power capacity in Watts ...  
UPS. - 1.3.6.1.4.1.318.1.1.1.9.3.3.1.16.1  
indexes:
  - labelname: upsPhaseOutputPhaseTableIndex  
type: gauge
  - labelname: upsPhaseOutputPhaseIndex  
type: gauge

# Attention : bugs

Mauvaise implémentation des MIB sur certains équipements :

snmp.yml : bon mais que ne marche pas

```
– name: upsPhaseOutputPercentPower
  oid: 1.3.6.1.4.1.318.1.1.1.9.3.3.1.16
  type: gauge
  help: The percentage of the UPS power capacity in Watts ...
  UPS. – 1.3.6.1.4.1.318.1.1.1.9.3.3.1.16
  indexes:
  – labelname: upsPhaseOutputPhaseTableIndex
    type: gauge
  – labelname: upsPhaseOutputPhaseIndex
    type: gauge
```

# Attention : bugs

Mauvaise implémentation des MIB sur certains équipements :

```
> snmpwalk -v 1 -c public ups.test.fr 1.3.6.1.4.1.318.1.1.1.9.3.3.1.16  
  
PowerNet-MIB::upsPhaseOutputPercentPower.1.1.1 = INTEGER: 42  
PowerNet-MIB::upsPhaseOutputPercentPower.1.1.2 = INTEGER: 40  
PowerNet-MIB::upsPhaseOutputPercentPower.1.1.3 = INTEGER: 39
```

- Ici, un ".1" de trop qui n'est pas un indice
- Pour plus de détails : [Issue 273](#)

# Alerts

```
/etc/prometheus/prometheus.yml
```

```
rule_files :  
  - /etc/prometheus/rules/node.rules.yml
```



# Alerts

```
/etc/prometheus/rules/node.rules.yml
```

```
groups:  
- name: Alerts nodes  
  rules:  
- alert: DiskWillFillIn4Hours  
  expr: predict_linear(node_filesystem_free{job="node"}[1h], 4 * 3600) < 0  
  for: 5m  
  labels:  
    severity: critical  
  annotations:  
    title: '{{ $labels.mountpoint }} is almost full on  
           {{ $labels.instance }}: {{ $value | humanize }}%'
```

# Alertmanager

```
/etc/prometheus/prometheus.yml
```

```
alerting:  
  alertmanagers:  
  - static_configs:  
    - targets:  
      - host:9093
```

## gestionnaire d'alerte

```
route:  
  \# On envoie tout vers slack par d'\defaut  
  receiver: slack  
[...]  
receivers:  
- name: 'slack '  
  slack_configs:  
  - api_url: "https://hooks.slack.com/services/xxx/xxx/xxxxxx"  
    channel: '\#prometheus'
```

- Exemple tiré de [1]

Le langage de query du moteur :

## La doc

- Les bases
- Opérateurs
- Fonctions

# Démo

Ou sur le site de [Prometheus Live Demo](#)

# Plan

- 1 Introduction
- 2 Prometheus
- 3 Grafana**
- 4 Conclusion
- 5 Référence

# Démo

Ou sur le site de [Play Grafana](#)

# Plan

- 1 Introduction
- 2 Prometheus
- 3 Grafana
- 4 Conclusion**
- 5 Référence



# Conclusion

## Conclusion : 2 mois après début des tests

- Challenge :
  - 1.0 journée [Prometheus](#) = équivalent [Cacti](#)
  - 0.5 journée = nouveaux dashboards [Grafana](#)
  - 0.5 journée = bug matériel et résolution (patch)
- Les +
  - Conf par fichier
  - Supporté par [Ceph](#), [Openstack](#) ...
  - Bien interfacé avec [Grafana](#) (dashboard existants)
  - M.A.J facile
  - Ajout d'équipement snmp facile
  - Langage puissant
  - Stockage optimisé :  $150\text{pdus} * 30\text{s} * 118\text{metrics} * 2 \text{ mois} = 3 \times 10^9 \text{ data} = 3\text{Go}$

# Plan

- 1 Introduction
- 2 Prometheus
- 3 Grafana
- 4 Conclusion
- 5 Référence**

# Références

- [1] Yannig, "Découverte de l'outil de supervision prometheus," Jan. 2018. [Online]. Available: <http://linuxfr.org/news/decouverte-de-l-outil-de-supervision-prometheus>
- [2] B. B. Rabenstein, "Configuring prometheus for high performance," Mar. 2017. [Online]. Available: [https://sched.ws/hosted\\_files/cloudnativeeu2017/ce/Slides.pdf](https://sched.ws/hosted_files/cloudnativeeu2017/ce/Slides.pdf)
- [3] D. Lee, "Visualize your data with grafana," 2017. [Online]. Available: <https://www.percona.com/live/e17/sites/default/files/slides/Visualize%20Your%20Data%20With%20Grafana%20-%20FileId%20-%20115450.pdf>
- [4] A. Kulkarni, "What the heck is time-series data (and why do i need a time-series database)?" 2017. [Online]. Available: <https://www.percona.com/live/17/sessions/what-heck-time-series-data-and-why-do-i-need-time-series-database>



# Questions ?

?