

Rapport automatisé :

Script Bash automatisé nommé rapport_kpi.sh qui :

- Interroge Prometheus en local (http://localhost:9091/metrics)
- Extrait et filtre les **KPI clés** :
 - up (état des services),
 - probe_success (disponibilité Blackbox),
 - scrape_duration_seconds (temps de collecte),
 - node_memory_MemAvailable_bytes (mémoire disponible),
 - node_cpu_seconds_total (utilisation CPU)
- Affiche un **rapport synthétique** (10 lignes max) dans un fichier rapport_kpi.txt.

Nom du fichier : rapport_kpi.sh

Emplacement recommandé : /usr/local/bin/rapport_kpi.sh

ou simplement dans ton home : ~/rapport_kpi.sh

Contenu du script :

KPI Report Script

```
#!/bin/bash

# Fichier de sortie
OUTPUT_FILE="rapport_kpi.txt"

# URL Prometheus local
PROM_URL="http://localhost:9091/metrics"

# Entête
echo "===== RAPPORT KPI PROMETHEUS =====" > "$OUTPUT_FILE"
echo "Date : $(date)" >> "$OUTPUT_FILE"
echo "" >> "$OUTPUT_FILE"

# État des services
echo "[UP STATUS]" >> "$OUTPUT_FILE"
curl -s $PROM_URL | grep '^up{' | head -n 3 >> "$OUTPUT_FILE"
echo "" >> "$OUTPUT_FILE"

# Succès des sondes blackbox
echo "[BLACKBOX PROBE SUCCESS]" >> "$OUTPUT_FILE"
curl -s $PROM_URL | grep 'probe_success' | head -n 3 >> "$OUTPUT_FILE"
echo "" >> "$OUTPUT_FILE"

# Durée des scrapes
echo "[SCRAPE DURATION]" >> "$OUTPUT_FILE"
curl -s $PROM_URL | grep 'scrape_duration_seconds' | grep -v '^#' | head -n 2 >> "$OUTPUT_FILE"
echo "" >> "$OUTPUT_FILE"

# Mémoire disponible
echo "[NODE MEMORY AVAILABLE]" >> "$OUTPUT_FILE"
curl -s $PROM_URL | grep 'node_memory_MemAvailable_bytes' | grep -v '^#' | head -n 1 >> "$OUTPUT_FILE"
echo "" >> "$OUTPUT_FILE"

# Activité CPU
echo "[CPU ACTIVITY]" >> "$OUTPUT_FILE"
curl -s $PROM_URL | grep 'node_cpu_seconds_total' | grep 'node="user"' | head -n 1 >> "$OUTPUT_FILE"

# Affichage du fichier
cat "$OUTPUT_FILE"
```

Instructions d'installation

cd : chmod +x rapport_kpi.sh ./rapport_kpi.sh

Plan opérationnel pour l'affichage des alertes Prometheus en console, via Grafana, et l'activation mail/SMS.

1. Affichage console des alertes en temps réel

Requête Prometheus

cd : curl -s http://localhost:9091/api/v1/alerts | jq

Affiche toutes les alertes **actuellement actives** dans Prometheus.

Pour filtrer les alertes actives uniquement :

cd : curl -s http://localhost:9091/api/v1/alerts | jq '.data.alerts[] | select(.state=="firing") | {alertname: .labels.alertname, severity: .labels.severity, startsAt: .startsAt}'

2. Affichage dans Grafana (via Prometheus datasource)

Étapes :

1. Configurer Prometheus comme Datasource dans Grafana : **ok déjà actifs**

- URL : http://localhost:9091
- Nom : Prometheus

2. Importer ou créer un dashboard d'alerte :

- Utilise un **panel Table ou Stat** avec la requête :
- `ALERTS{alertstate="firing"}`

3. Affiche dynamiquement les alertes actives :

ex : High_CPU_Usage, HTTP_Target_Down).

3. Activer les notifications par mail / SMS via Alertmanager

Prometheus n'envoie pas d'alertes directement. Tu dois passer par Alertmanager.

a) Installation d'Alertmanager

cd : /opt wget

<https://github.com/prometheus/alertmanager/releases/download/v0.27.0/alertmanager-0.27.0.linux-amd64.tar.gz>

cd : tar -xvf alertmanager-0.27.0.linux-amd64.tar.gz

cd : mv alertmanager-0.27.0.linux-amd64 /opt/alertmanager

cd : sudo cp /opt/alertmanager/alertmanager /usr/local/bin/

b) Exemple de configuration : [alertmanager.yml](#)

Chemin : `/etc/alertmanager/alertmanager.yml`

```
yaml
global:
  smtp_smarthost: 'smtp.gmail.com:587'
  smtp_from: 'ton.email@gmail.com'
  smtp_auth_username: 'ton.email@gmail.com'
  smtp_auth_password: 'ton_mot_de_passe_app'

route:
  receiver: 'mail-alerts'

receivers:
- name: 'mail-alerts'
  email_configs:
    - to: 'destinataire@domaine.com'
      send_resolved: true
```

Pour **Gmail**, active les mots de passe d'application (2FA requis).

c) Lier Prometheus à Alertmanager

Dans [prometheus.yml](#) :

```
yaml
alerting:
  alertmanagers:
    - static_configs:
      - targets: ['localhost:9093']
```

d) Lancer Alertmanager

cd : `alertmanager --config.file=/etc/alertmanager/alertmanager.yml`

En option : Notifier par SMS

- Tu peux utiliser des services comme :
 - [Twilio](#)
 - [OVH SMS](#)
 - [Mailgun + SMS Gateway](#)
- Intégration possible via webhook dans [alertmanager.yml](#) :

```
yaml
receivers:
- name: 'sms-alerts'
  webhook_configs:
    - url: 'https://monapi/push_sms'
```