

SUPERVISION PROMETHEUS / GRAFANA — INFRASTRUCTURE RESEAU

Institution : Mediaschool Nice — IRIS

1. CONTEXTE ET OBJECTIF

Dans le cadre du BTS SIO option SISR, ce projet consiste à déployer une chaîne de supervision complète permettant de surveiller en temps réel l'état de serveurs Linux et d'envoyer des alertes automatiques par mail en cas d'incident.

L'objectif est de fournir une visibilité immédiate sur l'état de l'infrastructure (disponibilité, charge CPU, mémoire, disque, réseau) et de déclencher des notifications automatiques dès qu'un seuil critique est atteint.

La solution déployée repose sur quatre composants principaux : Prometheus pour la collecte des métriques, Node Exporter pour l'exposition des données systèmes, Alertmanager pour la gestion et l'envoi des alertes, et Grafana pour la visualisation des tableaux de bord.

2. ARCHITECTURE GLOBALE

La supervision s'appuie sur deux machines virtuelles Debian déployées sous VirtualBox, interconnectées via un réseau Host-Only.

VM	Rôle	IP (NIC2 Host-Only)
monitor	Prometheus + Alertmanager + Grafana + Postfix	192.168.100.10
node1	Node Exporter (serveur supervisé)	192.168.100.11

Configuration réseau des VMs :

- Adaptateur 1 : NAT (accès Internet et SSH depuis l'hôte)
- Adaptateur 2 : Host-Only (réseau interne de supervision)

Redirections de ports NAT (accès depuis l'hôte) :

- SSH monitor : localhost:2223 → VM:22
- Grafana : localhost:2300 → VM:3000
- SSH node1 : localhost:2224 → VM:22

Flux de collecte :

node1 (Node Exporter :9100) → monitor (Prometheus :9090) → Alertmanager (:9093) → Postfix → Gmail SMTP

3. TECHNOLOGIES UTILISEES

Technologie	Version	Rôle
Prometheus	2.54.1	Collecte et stockage des métriques
Node Exporter	1.8.1	Exposition des métriques système Linux
Alertmanager	0.27.0	Gestion et routage des alertes
Grafana	Stable (OSS)	Visualisation des métriques — tableaux de bord

Postfix	Serveur mail — relais Gmail SMTP (TLS)
Debian	Système d'exploitation des VMs
VirtualBox	Hyperviseur de type 2

4. DESCRIPTION DES COMPOSANTS

4.1 Prometheus

Prometheus est le moteur central de collecte. Il scrute (scrape) périodiquement les endpoints exposés par les exporteurs pour récupérer les métriques. L'intervalle de collecte est configuré à 15 secondes.

Fichier de configuration prometheus.yml :

- Cibles surveillées : localhost:9090 (auto-supervision) et 192.168.100.10:9100 / 192.168.100.11:9100 (Node Exporters)
- Référence vers le fichier de règles d'alertes : /etc/prometheus/alerts.yml

4.2 Node Exporter

Node Exporter est déployé sur les deux VMs (monitor et node1). Il expose les métriques système via un endpoint HTTP sur le port 9100. Les métriques collectées incluent : utilisation CPU, mémoire RAM, espace disque, trafic réseau.

4.3 Alertmanager

Alertmanager reçoit les alertes déclenchées par Prometheus lorsque les règles définies dans alerts.yml sont satisfaites. Il gère le routage des alertes et leur envoi par mail via le relais Postfix/Gmail.

Configuration du routage :

- Délai avant premier envoi (group_wait) : 10 secondes
- Intervalle de renvoi (group_interval) : 30 secondes
- Intervalle de répétition (repeat_interval) : 2 minutes

4.4 Grafana

Grafana se connecte à Prometheus comme source de données et permet de créer des tableaux de bord visuels avec des panneaux affichant en temps réel : utilisation CPU (%), RAM (%), espace disque, trafic réseau.

4.5 Postfix (relais Gmail)

Postfix est configuré en tant que relais SMTP authentifié vers les serveurs Gmail (smtp.gmail.com:587). Il reçoit les alertes d'Alertmanager et les transmet par email. La configuration utilise SASL pour l'authentification Gmail et le chiffrement TLS.

5. FONCTIONNEMENT TECHNIQUE

5.1 Règles d'alerte (alerts.yml)

Deux règles d'alerte sont définies :

Alerte InstanceDown :

- Déclenchement : target inaccessible (up == 0) pendant 1 minute
- Sévérité : critical

- Message : Instance {{ \$labels.instance }} down

Alerte HighCPUUsage :

- Déclenchement : utilisation CPU > 80% pendant 2 minutes
- Sévérité : warning
- Calcul : $100 - (\text{avg by(instance)}(\text{rate}(\text{node_cpu_seconds_total}\{\text{mode}=\text{"idle"}\}[2\text{m}])) * 100 > 80$

5.2 Cycle de supervision

1. Node Exporter expose les métriques système sur :9100/metrics

2. Prometheus scrute chaque cible toutes les 15 secondes

3. Si une règle d'alerte est évaluée comme vraie pendant la durée configurée, Prometheus envoie une alerte à Alertmanager

4. Alertmanager agrège l'alerte, respecte les délais de routage et envoie un email via Postfix

5. Grafana affiche en continu les métriques collectées par Prometheus

5.3 Configuration du réseau

Les adresses IP fixes sont configurées sur l'interface enp0s8 (NIC2 Host-Only) des deux VMs. Le fichier /etc/hosts de chaque VM résout les noms monitor.mediaschool.local et node1.mediaschool.local.

6. SECURITE

- Les services Prometheus, Alertmanager et Node Exporter écoutent uniquement sur les interfaces internes (réseau Host-Only 192.168.100.0/24)
- UFW est installé sur les VMs pour filtrer les accès entrants
- L'authentification SASL et le chiffrement TLS sont utilisés pour le relais Gmail
- Les identifiants Gmail sont stockés dans le fichier /etc/postfix/sasl_passwd avec permissions restreintes (chmod 600) et hashés via postmap

7. DEPLOIEMENT ET CONFIGURATION

Pré-requis :

- Deux VMs Debian sous VirtualBox
- Accès Internet depuis les VMs (adaptateur NAT)
- Compte Gmail avec mot de passe d'application activé

Procédure de déploiement (VM monitor) :

1. Configurer les IP fixes sur enp0s8 et mettre à jour /etc/hosts
2. Installer Node Exporter sur les deux VMs et créer le service systemd
3. Télécharger et installer Prometheus, créer les répertoires /etc/prometheus et /var/lib/prometheus
4. Créer prometheus.yml avec les cibles de scrape et la référence aux règles d'alertes
5. Créer le service systemd pour Prometheus
6. Télécharger et installer Alertmanager, créer alertmanager.yml avec la configuration SMTP
7. Créer alerts.yml avec les règles InstanceDown et HighCPUUsage
8. Installer Postfix et configurer le relais Gmail (sasl_passwd, generic, postmap)
9. Installer Grafana, configurer Prometheus comme source de données et créer les tableaux de bord
10. Configurer les redirections de ports NAT dans VirtualBox pour l'accès externe

Accès aux interfaces :

- Prometheus : http://192.168.100.10:9090
- Alertmanager: http://192.168.100.10:9093
- Grafana : http://192.168.56.1:2300 (via redirection NAT)

8. TESTS ET VALIDATION

Tests réalisés :

Test InstanceDown :

- Arrêt du service Node Exporter sur node1 → Alerte InstanceDown firing après 1 minute → Email reçu : succès

Test HighCPUUsage :

- Simulation de charge CPU via stress-ng (4 threads, 180 secondes) → Alerte HighCPUUsage firing après 2 minutes → Email reçu : succès

Test Grafana :

- Vérification de l'affichage en temps réel des métriques CPU, RAM, disque et réseau sur les tableaux de bord : succès

Test relais Postfix :

- Envoi d'un mail de test via la commande mail : succès

Résultats obtenus :

- Chaîne de supervision complète opérationnelle
- Alertes InstanceDown et HighCPUUsage déclenchées et emails reçus
- Tableaux de bord Grafana affichant les métriques en temps réel

Durée de réalisation : 2 jours