FR

Monitoring OPNsense avec Grafana + InfluxDB

Tableaux de bord temps réel — Série Stack OPNsense #9

Mars 2026 | botum.ca

Monitoring OPNsense avec Grafana et InfluxDB

Tableaux de bord temps réel — Billet 9/9 de la Série Stack OPNsense

---

Sommaire

Article complet : blog.botum.ca/opnsense-monitoring-grafana-influxdb

FR

Monitoring OPNsense avec Grafana + InfluxDB

Tableaux de bord temps réel — Série Stack OPNsense #9

Mars 2026 | botum.ca

Hub de la série : blog.botum.ca/opnsense-stack-securite-enterprise-proxmox

1. Pourquoi monitorer son firewall ?

Un firewall sans monitoring, c'est comme conduire les yeux bandés. Sur mon infra BOTUM, j'ai appris à la dure que les problèmes réseau les plus coûteux sont ceux qu'on ne voit pas venir : une saturation WAN silencieuse à 3h du matin, une tentative d'intrusion qui passe entre les mailles, un VLAN IoT qui génère 10x plus de trafic que la normale.

Le monitoring temps réel avec Grafana + InfluxDB m'apporte trois choses essentielles :

• **Visibilité complète :** voir en temps réel ce qui passe sur chaque interface, chaque VLAN, chaque tunnel VPN.

• **Détection d'anomalies :** identifier les comportements anormaux — spike de trafic, flood DNS, connexions vers des IPs suspectes.

• **Conformité SLA :** prouver la disponibilité (uptime), documenter les incidents réseau, mesurer la latence WAN.

■ *Ce billet est le 9e et dernier de la Série Stack OPNsense — vous avez déjà le firewall, les VLANs, le VPN, le WiFi, la protection IDS/IPS, le NAC et le DNS. Il ne manquait que les yeux.*

2. Architecture de monitoring

La stack que j'utilise est simple et éprouvée :

```
OPNsense (données brutes)
     ↓  plugin os-telegraf (collecte)
     ↓
Telegraf (agent de métriques)
     ↓  HTTP/UDP vers port 8086
     ↓
InfluxDB 2.x (base de données time-series)
     ↓  source de données Grafana
     ↓
Grafana (visualisation + alerting)
     ↓  notifications
     ↓
Email / Telegram / Slack
```

Chaque composant tourne dans un container Docker sur une VM Proxmox dédiée (ou sur le même host Proxmox qu'OPNsense). J'alloue 2 vCPU + 2 Go RAM — largement suffisant pour une infra jusqu'à 1 Gbps de trafic agrégé.

■ *Telegraf est l'agent léger installé dans OPNsense qui collecte les métriques système et réseau, puis les envoie à InfluxDB via l'API HTTP.*

3. Activer l'export InfluxDB dans OPNsense

3.1 Installer le plugin os-telegraf

```
# Dans OPNsense : System > Firmware > Plugins
# Chercher "telegraf" et installer : os-telegraf

# Ou via SSH dans OPNsense :
pkg install os-telegraf
```

3.2 Configurer Telegraf dans OPNsense

```
# Services > Telegraf > General
# Cocher : Enable Telegraf
# InfluxDB URL : http://PROXMOX-IP:8086
# Token : (généré dans InfluxDB — voir section 4)
# Bucket : opnsense
# Organization : botum

# Activer les collecteurs :
# ✓ Interfaces (trafic par interface)
# ✓ Firewall (règles, états)
# ✓ CPU / Memory / Disk
# ✓ Netflow (si activé)
# ✓ Gateway (latence WAN)
```

3.3 Configuration manuelle telegraf.conf (avancé)

```
# /usr/local/etc/telegraf.conf

[agent]
  interval = "10s"
  flush_interval = "10s"

[[inputs.net]]
  interfaces = ["em0", "vtnet0", "igc0"]

[[inputs.cpu]]
  percpu = false
  totalcpu = true

[[inputs.mem]]

[[inputs.system]]

[[inputs.netstat]]

# Métriques firewall OPNsense spécifiques
[[inputs.exec]]
  commands = ["/usr/local/opnsense/scripts/OPNsense/Diagnostics/interface.php"]
  data_format = "json"
  timeout = "5s"

[[outputs.influxdb_v2]]
  urls = ["http://192.168.1.x:8086"]
  token = "VOTRE-TOKEN-INFLUXDB"
  organization = "botum"
  bucket = "opnsense"
  timeout = "5s"
```

4. Installer InfluxDB 2.x en Docker sur Proxmox

Je déploie InfluxDB 2.x sur la même VM Proxmox que Grafana pour minimiser la latence et simplifier la maintenance.

4.1 Docker Compose — Stack monitoring complète

```
mkdir -p /opt/monitoring/{influxdb,grafana}
cd /opt/monitoring

cat > docker-compose.yml << 'EOF'
version: '3.8'

services:
  influxdb:
    image: influxdb:2.7
    container_name: influxdb
    restart: unless-stopped
    ports:
      - "8086:8086"
    volumes:
      - ./influxdb/data:/var/lib/influxdb2
      - ./influxdb/config:/etc/influxdb2
    environment:
      DOCKER_INFLUXDB_INIT_MODE: setup
      DOCKER_INFLUXDB_INIT_USERNAME: admin
      DOCKER_INFLUXDB_INIT_PASSWORD: BotumSecure2026!
      DOCKER_INFLUXDB_INIT_ORG: botum
      DOCKER_INFLUXDB_INIT_BUCKET: opnsense
      DOCKER_INFLUXDB_INIT_RETENTION: 90d

  grafana:
    image: grafana/grafana:latest
    container_name: grafana
    restart: unless-stopped
    ports:
      - "3000:3000"
    volumes:
      - ./grafana:/var/lib/grafana
    environment:
      GF_SECURITY_ADMIN_PASSWORD: BotumSecure2026!
      GF_INSTALL_PLUGINS: grafana-clock-panel,grafana-worldmap-panel
    depends_on:
      - influxdb

EOF

docker compose up -d
echo "Stack monitoring démarrée !"
echo "InfluxDB : http://PROXMOX-IP:8086"
echo "Grafana  : http://PROXMOX-IP:3000"
```

4.2 Générer le token InfluxDB

```
# Via l'interface web InfluxDB (http://IP:8086)
# 1. Connectez-vous avec admin / BotumSecure2026!
# 2. Load Data > API Tokens > Generate API Token
# 3. Sélectionner : All Access Token
# 4. Copier le token (format : long string 88 chars)
# 5. Coller dans la config Telegraf d'OPNsense

# Ou via CLI :
docker exec influxdb influx auth create   --org botum   --all-access   --description "Telegraf OP

# Le token s'affiche UNE SEULE FOIS — notez-le dans votre vault !
```

■ *Stockez le token InfluxDB dans votre gestionnaire de secrets (Vaultwarden / 1Password). Une fois généré, il ne sera plus affiché en clair.*

5. Configurer Grafana avec InfluxDB

5.1 Ajouter la source de données InfluxDB

```
# Grafana → Configuration > Data Sources > Add Data Source
# Sélectionner : InfluxDB

# Query Language : Flux (recommandé pour InfluxDB 2.x)
# URL : http://influxdb:8086  (ou http://PROXMOX-IP:8086)

# InfluxDB Details :
# Organization : botum
# Token : VOTRE-TOKEN
# Default Bucket : opnsense

# Cliquer : Save & Test
# ✓ "datasource is working. 1 buckets found"
```

5.2 Requête Flux de base — trafic WAN

```
// Flux query — trafic WAN en temps réel
from(bucket: "opnsense")
  |> range(start: -1h)
  |> filter(fn: (r) => r._measurement == "net")
  |> filter(fn: (r) => r.interface == "em0")
  |> filter(fn: (r) => r._field == "bytes_recv" or r._field == "bytes_sent")
  |> derivative(unit: 1s, nonNegative: true)
  |> map(fn: (r) => ({ r with _value: r._value * 8.0 }))
  |> aggregateWindow(every: 30s, fn: mean, createEmpty: false)
```

6. Importer les dashboards OPNsense

La communauté Grafana propose d'excellents dashboards préconçus pour OPNsense. Voici ceux que j'utilise :

• **OPNsense Overview (ID: 13713)** : Trafic WAN, CPU, mémoire, états firewall, gateways

• **OPNsense Interfaces (ID: 14966)** : Débit par interface, erreurs, paquets/sec

• **OPNsense VPN WireGuard (personnalisé)** : Pairs connectés, trafic par tunnel, latence

• **VLAN Traffic (personnalisé)** : Trafic par VLAN, top clients, anomalies

• **OPNsense Firewall Rules (ID: 15413)** : Hits par règle, connexions bloquées, géolocalisation IPs

```
# Importer via Grafana UI :
# Dashboards > Import > Dashboard ID ou JSON

# Import rapide depuis grafana.com :
# Dashboard ID 13713 → OPNsense Overview
# Sélectionner : Data Source = InfluxDB (opnsense)
# Importer

# Personnaliser les variables du dashboard :
# Bucket : opnsense
# Interfaces WAN : em0, vtnet0 (selon votre config)
# Interfaces LAN/VLAN : vtnet1, vtnet2, vtnet3...
```

■ *Les IDs de dashboards communautaires évoluent. Si un ID ne fonctionne plus, cherchez "OPNsense" sur grafana.com/grafana/dashboards.*

7. Alerting Grafana : bande passante et connexions suspectes

7.1 Configurer un contact point (notification)

```
# Grafana → Alerting > Contact Points > Add Contact Point
# Type : Telegram (recommandé pour alertes mobile)

# Bot Token : votre Telegram Bot Token
# Chat ID   : votre Chat ID (ou groupe)

# Tester : "Test" — vous devriez recevoir un message Telegram
# Sauvegarder le contact point

# Types disponibles : Email, Slack, PagerDuty, OpsGenie, Webhook...
```

7.2 Créer une alerte — saturation bande passante WAN

```
# Grafana → Alerting > Alert Rules > New Alert Rule

# 1. Query :
from(bucket: "opnsense")
  |> range(start: -5m)
  |> filter(fn: (r) => r._measurement == "net" and r.interface == "em0")
  |> filter(fn: (r) => r._field == "bytes_sent" or r._field == "bytes_recv")
  |> derivative(unit: 1s, nonNegative: true)
  |> map(fn: (r) => ({ r with _value: r._value * 8.0 }))
  |> sum()

# 2. Condition :
# WHEN avg() IS ABOVE 900000000  (900 Mbps)
# FOR 2 minutes

# 3. Annotations :
# Summary  : Saturation WAN détectée
# Description : Trafic WAN > 900 Mbps depuis {{ $values.A }} secondes

# 4. Notification policy → Telegram
```

7.3 Alerte — connexions suspectes (spike)

```
# Flux query — connexions firewall par minute
from(bucket: "opnsense")
  |> range(start: -10m)
  |> filter(fn: (r) => r._measurement == "pf_state")
  |> filter(fn: (r) => r._field == "count")
  |> derivative(unit: 1m, nonNegative: true)
  |> mean()

# Condition : WHEN mean() IS ABOVE 5000 (connexions/min)
# Ce seuil doit être calibré sur votre trafic normal
# Commencez par observer 1 semaine avant de fixer des seuils
```

8. Intégrer Suricata et AdGuard Home dans Grafana

8.1 Logs Suricata via Loki (recommandé)

```
# Installer Grafana Loki pour les logs (complément à InfluxDB)
# Ajouter dans docker-compose.yml :

  loki:
    image: grafana/loki:latest
    container_name: loki
    restart: unless-stopped
    ports:
      - "3100:3100"
    volumes:
      - ./loki:/loki

  promtail:
    image: grafana/promtail:latest
    container_name: promtail
    restart: unless-stopped
    volumes:
      - /var/log:/var/log:ro
      - ./promtail-config.yml:/etc/promtail/config.yml

# Sur OPNsense : configurer Suricata pour écrire les EVE logs en JSON
# Intrusion Detection > Administration > Log to syslog : ON
# Format : EVE JSON

# Dans Proxmox : rediriger syslog OPNsense vers Promtail
# /etc/promtail-config.yml :
scrape_configs:
  - job_name: suricata
    static_configs:
      - targets: [localhost]
        labels:
          job: suricata
          __path__: /var/log/suricata/*.json
```

8.2 Dashboard Suricata dans Grafana

```
# Grafana → Dashboards → Import
# Dashboard communautaire : "Suricata EVE Logs" (ID: 14972)
# Data Source : Loki

# Panels utiles :
# - Top 10 des règles déclenchées
# - Alertes par sévérité (HIGH/MEDIUM/LOW)
# - Timeline des attaques
# - Carte géo des IPs sources (WorldMap plugin)
# - Flow: src_ip → dst_ip → proto
```

8.3 Métriques AdGuard Home via API REST

```
# AdGuard Home expose ses stats via API REST
# Intégration avec Telegraf :

[[inputs.http]]
  urls = ["http://192.168.1.x:3000/control/stats"]
  method = "GET"
  username = "admin"
  password = "VOTRE-MOT-DE-PASSE"
  data_format = "json"
  name_suffix = "_adguard"
  [inputs.http.tags]
    source = "adguard-home"

# Métriques collectées :
# dns_queries, blocked_filters, blocked_safebrowsing
# replaced_safesearch, blocked_parental
# avg_processing_time

# Dans Grafana → Panel → Query InfluxDB :
from(bucket: "opnsense")
  |> range(start: -24h)
  |> filter(fn: (r) => r._measurement == "http_adguard")
  |> filter(fn: (r) => r._field == "dns_queries" or r._field == "blocked_filters")
```

Avec Suricata (billet 7) et AdGuard Home (billet 8) intégrés dans Grafana, vous avez une vue de sécurité unifiée : détection d'intrusion + filtrage DNS, tout dans un seul dashboard.

9. Conclusion : récapitulatif de la série complète

Nous voici au bout de 9 billets qui construisent une infrastructure réseau enterprise complète à partir d'OPNsense sur Proxmox. Voici ce que vous avez maintenant :

- **Billet 1 — OPNsense sur Proxmox :** VM firewall haute performance, interfaces réseaux virtuelles
- **Billet 2 — VLANs Zero-Trust :** Segmentation réseau : LAN, IoT, DMZ, Guest — isolation complète
- **Billet 3 — WireGuard VPN + SD-WAN LTE :** VPN site-à-site, accès remote workers, failover 4G/LTE
- **Billet 4 — WiFi + UniFi AP :** Gestion des APs WiFi, isolation VLAN sans fil, portail captif
- **Billet 5 — CrowdSec + Fail2ban :** Protection collaborative, blocklists dynamiques, bannissement IP
- **Billet 6 — NAC FreeRADIUS 802.1X :** Authentification réseau, certificats, politique d'accès par identité
- **Billet 7 — Suricata IDS/IPS + DPI :** Détection d'intrusion, prévention en ligne, inspection deep packet
- **Billet 8 — AdGuard Home + DoH :** Filtrage DNS, blocklists, DNS over HTTPS — vie privée et sécurité
- **Billet 9 — Grafana + InfluxDB :** Monitoring temps réel, dashboards, alerting — visibilité totale

Cette stack vous donne une infrastructure réseau que la plupart des PME à $50k/an de budget réseau ne possèdent pas — pour moins de $500 en matériel et $0 en licences logicielles.

Les prochaines étapes : la Série OpenClaw — automatisation de l'infra, agents intelligents, self-healing network. Restez connectés.

Article complet : blog.botum.ca/opnsense-monitoring-grafana-influxdb

Hub de la série : blog.botum.ca/opnsense-stack-securite-enterprise-proxmox

Site : www.botum.ca • contact@botum.ca