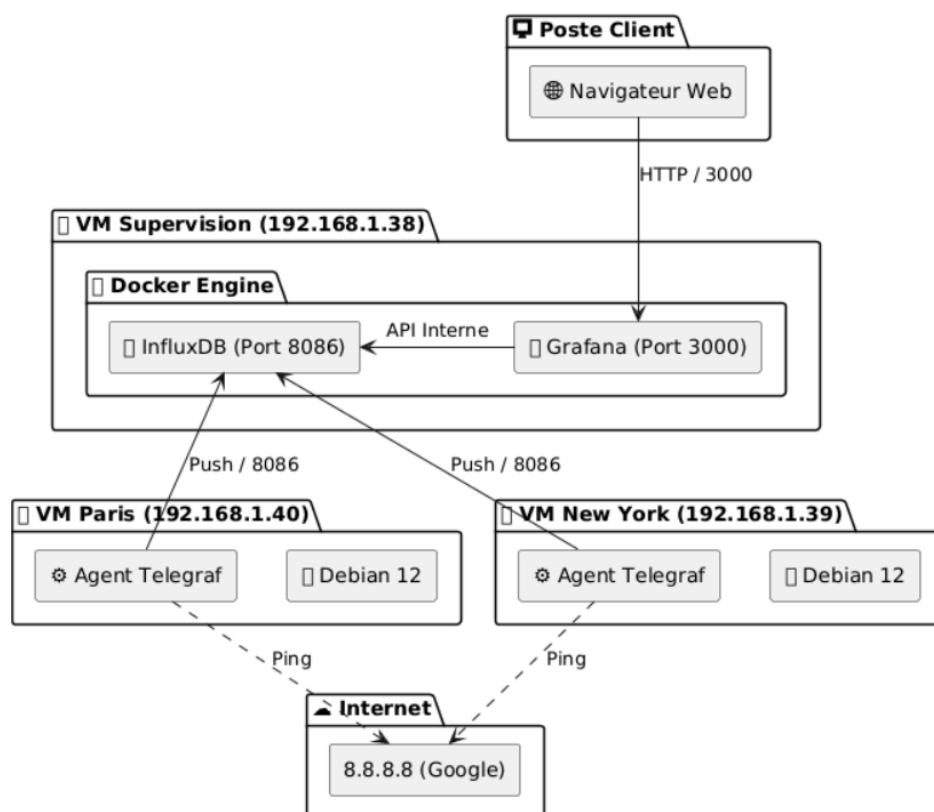


SOMMAIRE

- L'Objectif..... 1
- 1. Architecture & Déploiement.....3
- 2. Fonctionnalités Clés du Dashboard.....3
- 3. Challenges Techniques & Troubleshooting (La valeur ajoutée)..... 4
 - A. Installation sur Debian 12 (Bookworm)..... 4
 - B. Typage des Données GPS..... 5
 - C. Permissions & Configuration du Ping..... 5
- Bilan..... 5

L'Objectif

Concevoir et déployer une stack de monitoring complète pour surveiller en temps réel un parc de serveurs Linux répartis géographiquement (Simulés : Paris, New York, Londres). L'objectif était de visualiser la santé des systèmes (CPU) et la qualité du réseau (Latence/Ping) sur un tableau de bord centralisé et interactif.



Stack Technique

- ❖ OS : 3 VMs Linux (Debian 12 Bookworm).
- ❖ Conteneurisation : Docker & Docker Compose.
- ❖ Agent de collecte : Telegraf.
- ❖ Base de données : InfluxDB (Time Series Database).
- ❖ Visualisation : Grafana.

1. Architecture & Déploiement

J'ai opté pour une architecture distribuée :

1. Le Serveur Central (Master) : Héberge la base de données (InfluxDB) et le Dashboard (Grafana) dans des conteneurs Docker isolés via un réseau interne (docker-compose).
2. Les Agents (Nodes) : Chaque VM exécute un service Telegraf qui collecte les métriques locales et les pousse vers le Master.

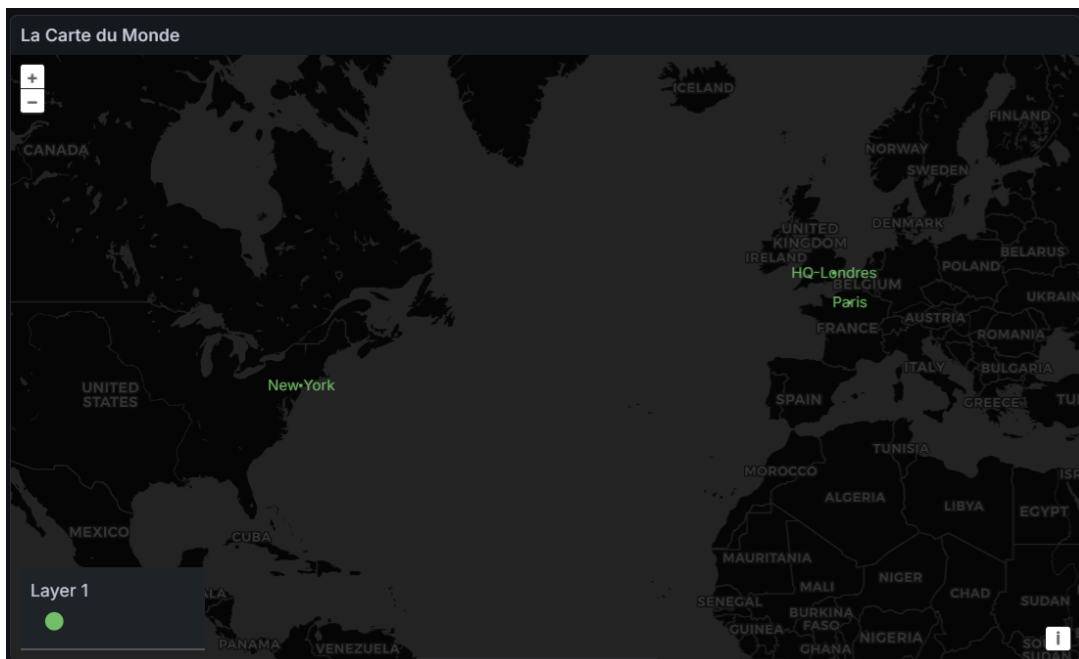
```
root@vm-supervision:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
US            PORTS
85c8bde236a1   grafana/grafana  "/run.sh"               5 hours ago   Up 5
hours         0.0.0.0:3000->3000/tcp, :::3000->3000/tcp   grafana
3ec41b47122c   influxdb:1.8    "/entrypoint.sh infl..." 5 hours ago   Up 5
hours         0.0.0.0:8086->8086/tcp, :::8086->8086/tcp   influxdb
```

Vue du terminal confirmant l'exécution des conteneurs InfluxDB et Grafana sur le serveur maître, orchestrés via Docker Compose avec les ports exposés.

2. Fonctionnalités Clés du Dashboard

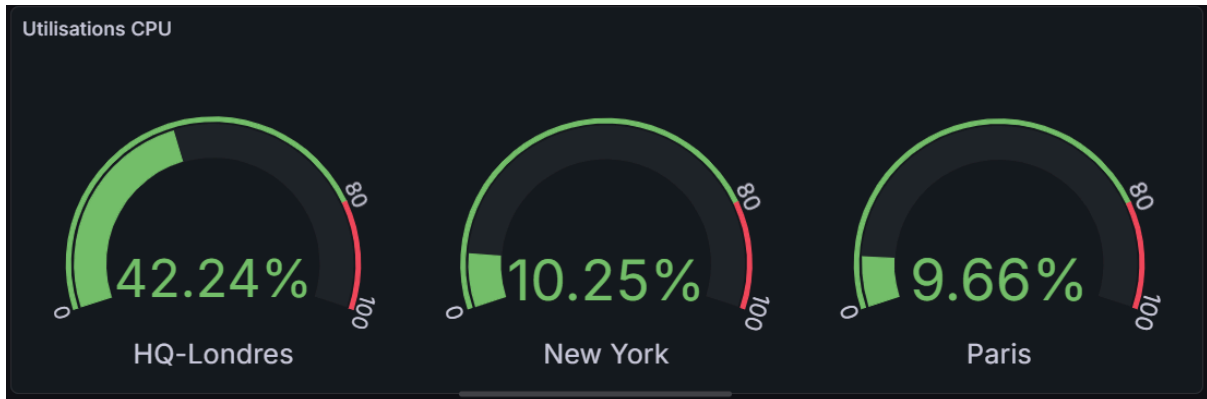
J'ai configuré Grafana pour offrir une vue "Météo" instantanée de l'infrastructure :

- ❖ Géolocalisation (Worldmap) : Injection de tags GPS (lat, lon) dans la configuration Telegraf de chaque serveur pour les placer automatiquement sur une carte dynamique.



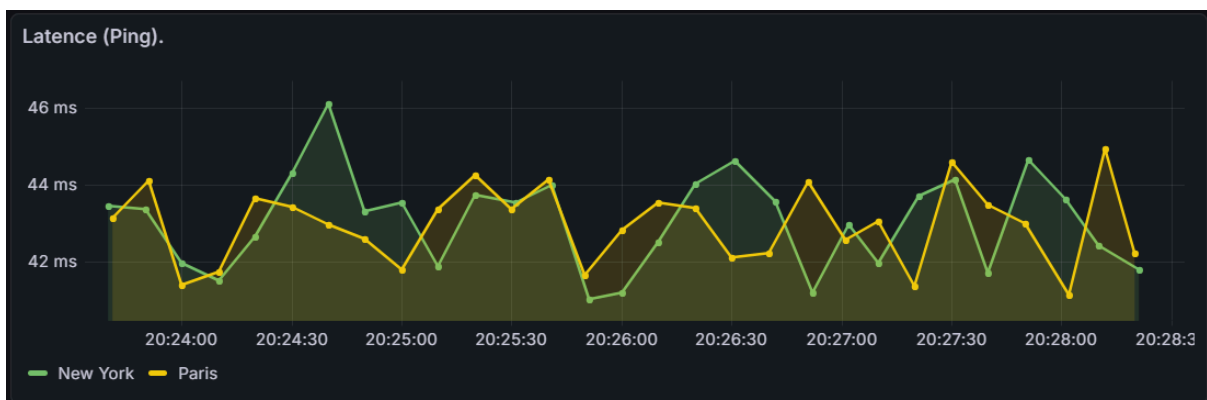
Restitution cartographique temps réel des nœuds d'infrastructure basés à Paris, New York et Londres grâce à l'injection de métadonnées GPS par Telegraf.

- ❖ Alerting Visuel (Jauges) : Configuration de seuils critiques. Les jauges passent du vert au rouge si le CPU dépasse 80%.



Monitoring de la charge CPU par datacenter avec des seuils d'alerte visuels configurés pour passer au rouge au-delà de 80% d'utilisation

- ❖ Analyse Réseau (Latence) : Surveillance de la connectivité Internet via des sondes Ping actives depuis chaque agence.



Analyse comparative de la latence réseau (en ms) entre les agences, validant la résolution des problèmes de permissions sur le plugin Ping.

3. Challenges Techniques & Troubleshooting (La valeur ajoutée)

Ce projet m'a confronté à plusieurs obstacles techniques que j'ai dû résoudre :

A. Installation sur Debian 12 (Bookworm)

- Problème : Les nouvelles politiques de sécurité de Debian bloquaient l'ajout des clés GPG du dépôt InfluxData.
- Solution : J'ai contourné le blocage en téléchargeant manuellement le paquet .deb officiel et en forçant l'installation via dpkg, garantissant une installation propre.

B. Typage des Données GPS

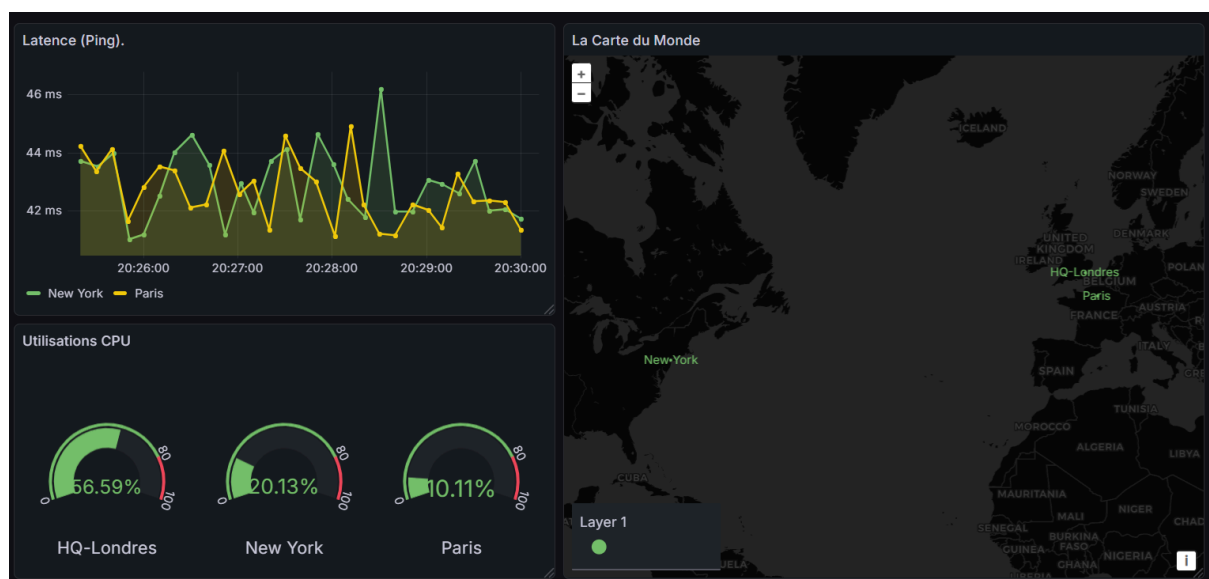
- Problème : Grafana ne pouvait pas afficher la carte car les coordonnées GPS étaient transmises sous forme de "Texte" (String) par la base de données.
- Solution : Utilisation des Transformations Grafana ("Convert field type") pour caster les données en "Number" à la volée directement dans le dashboard.

C. Permissions & Configuration du Ping

- Problème : Sur la VM New York, le service Telegraf plantait (failed to start) lors de l'activation du plugin Ping.
- Diagnostic : Analyse des logs via journalctl -xeu telegraf et test de config (telegraf --test) révélant une erreur de syntaxe sur les unités de temps (1.0s vs 1.0) et un manque de droits.
- Solution : Correction de la syntaxe TOML et ajout de la directive method = "exec" pour permettre à l'agent d'exécuter les commandes système.

Bilan

Ce projet valide ma capacité à maîtriser la chaîne complète de la donnée (**Collecte > Stockage > Visualisation**) et à maintenir une infrastructure linux saine. Il démontre également des compétences solides en troubleshooting sur des environnements conteneurisés et des systèmes récents.



Vue d'ensemble du dashboard final consolidant la géolocalisation temps réel, le monitoring CPU critique et l'analyse de la latence réseau multi-sites.