camp
tocamp

INNOVATIVE SOLUTIONS
BY OPEN SOURCE EXPERTS

IA & observabilité : diagnostic des problèmes dans les infrastructures géospatiales

Émilien Devos

Écosystème utilisé

Claude

Model Context Protocol

Grafana loki

Prometheus

Analyse basique des derniers logs de la plateforme (tests)

Based on my analysis of the Loki logs, I found several types of error logs for geoserver. Here's a summary of the key error patterns:

GeoServer Error Log Summary

Main Error Types Found:

1. **Database Schema Errors** (Most Critical):
   - `Schema 'parcellaire_ _personnes_morales_2020_publi' does not exist`
   - `Schema ' _enveloppe_urbaine_2021' does not exist`
   - These appear to be missing database schemas that geoserver is trying to access
2. **WMS Layer Geometry Errors** (High Frequency):
   - `ERROR [geoserver.wms] - An error occurred trying to determine if the layer is geometryless`
   - This error is occurring very frequently (multiple times per minute)
3. **Spring Security/Exception Handling**:
   - Multiple `ExceptionTranslationFilter` stack traces
   - `InvocationTargetException` errors
4. **Load Balancer Warnings**:
   - Nginx upstream duplicate header warnings: `upstream sent duplicate header line: "Expires"`

Affected Components:

- **Namespace**: `-production` and `-test`
- **Containers**: `geoserver-slave`, `geoserver-lb`, `georchestra-geoserver`
- **Pods**: Multiple geoserver slave instances

Recommendations:

1. **Fix Database Schema Issues**: The missing schemas (`parcellaire_ _personnes_morales_2020_publi`, `_enveloppe_urbaine_2021`) need to be restored or the layer configurations updated

Trouver des solutions à une consommation mémoire excessive

I now have a complete picture. Here's my full analysis:

Memory Leak Investigation: GeoServer Slave Restarts

Memory Spike Timeline

Pod 7897b8pckqr spiked from **1.1 GB →15.2 GB in ~2.5 minutes** (09:24:30–09:27:00), with similar patter

---
Root Causes (by severity)

1. Mass concurrent tile flood from [redacted] (PRIMARY)

At ~09:24:40, this IP fired **98 simultaneous tile requests** (all 256×256) across three layers at once:

The VRT layer is the primary pressure point — GDAL's 2 GB cache is filled to capaci
When that's combined with the JVM heap used by all the other concurrent requests, y

Targeted fix

1. Reduce GDAL_CACHEMAX drastically — 2048 MB for a single GeoServer process handli

```
- name: GDAL_CACHEMAX
  value: "256"  # was 2048 — GDAL COG reads are already tile-efficient
```

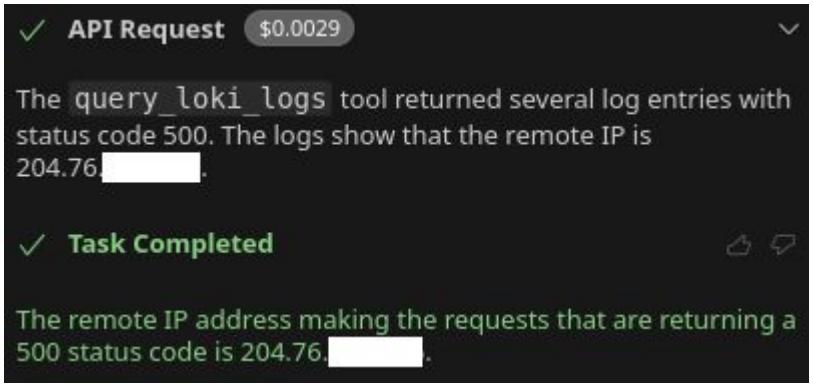
2. Cap the JVM heap to leave room for GDAL's off-heap budget:

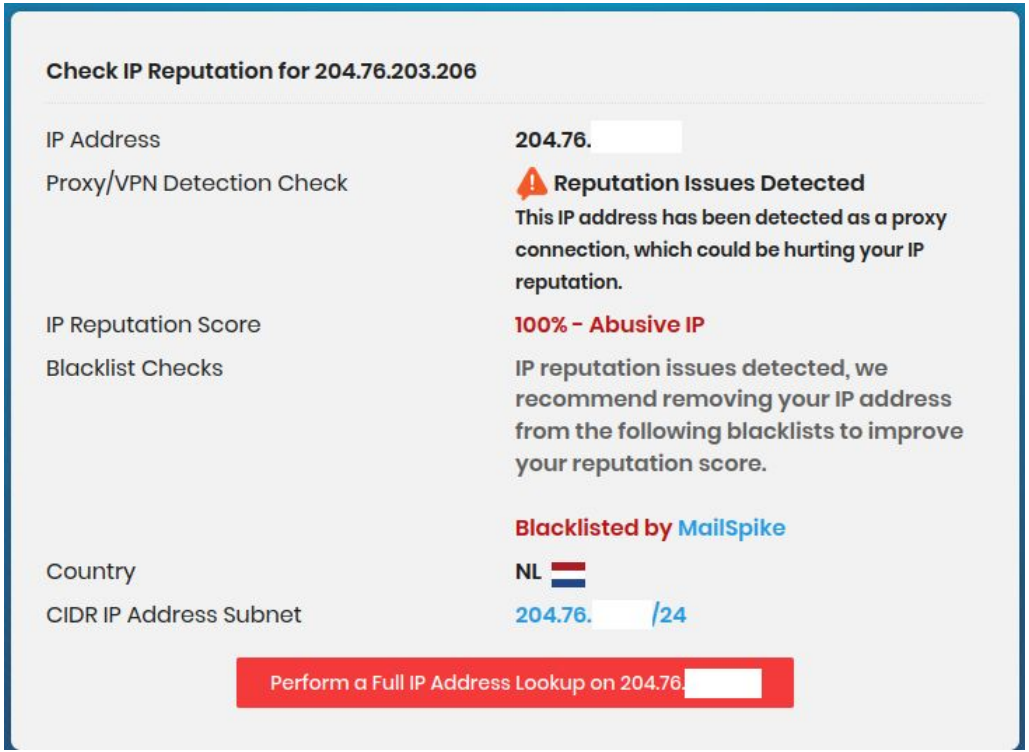
4. Consider rate-limiting concurrent requests to the VRT-backed layer specifically

Actions réalisées par Claude :

1. Lecture des logs GeoServer et Traefik (serveur web) + métriques Prometheus
2. Analyse et compréhension du problème.
3. Génération d'un script pour reproduire le problème en direct.
4. Lancement du script pour reproduction du problème.
5. Génération de solutions pour palier au problème.
6. Application des solutions dans le dépôt Git.

Détecter les visiteurs les plus bruyants





D’autres exemples de cas d’utilisation

- Détection de problèmes de performance en temps réel (analyse de nombreuses sources de données en quelques minutes).
- Création de scripts pour la reproduction de problèmes (performance, erreurs).
- Suggestion de paramètres d’optimisation dans les composants de geOrchestra (GeoServer, GeoWebCache, PostgreSQL, etc).
- Création de scripts pour tester les performances en vue d’une prochaine hausse en charge.

À retenir

- L’IA accompagne la résolution, sans remplacer des outils comme Gaia ou Grafana.
- Capacité d’analyser beaucoup de contenu et de multiples sources en très peu de temps.
- Toujours faire intervenir un humain dans la boucle et vérifier avant d’accepter la solution.

camptocamp
INNOVATIVE SOLUTIONS
BY OPEN SOURCE EXPERTS

Merci pour votre attention.

INNOVATIVE SOLUTIONS
BY OPEN SOURCE EXPERTS