Rapport du laboratoire 4

Github : https://github.com/SunnyKentz/labo-4-log430.git

Préambule

Dans le cadre du laboratoire 4, j'ai ajouté deux nouvelles dépendances à mon architecture : Grafana et Prometheus. Ces deux dépendances augmentent l'observabilité de mon application. Sur Grafana, je surveille les 4 Golden Signals.

Test de charge initial et observabilité de base

Outils choisis :

J'ai créé mon propre outil sous `tests/loadtests/test.js`. Cet outil me permet d'envoyer des requêtes fetch selon le nombre de requêtes par seconde mis en argument. Je fais ces trois requêtes :

- Consultation simultanée des stocks de plusieurs magasins.
- Génération de rapports consolidés.
- Mise à jour de produits à forte fréquence.

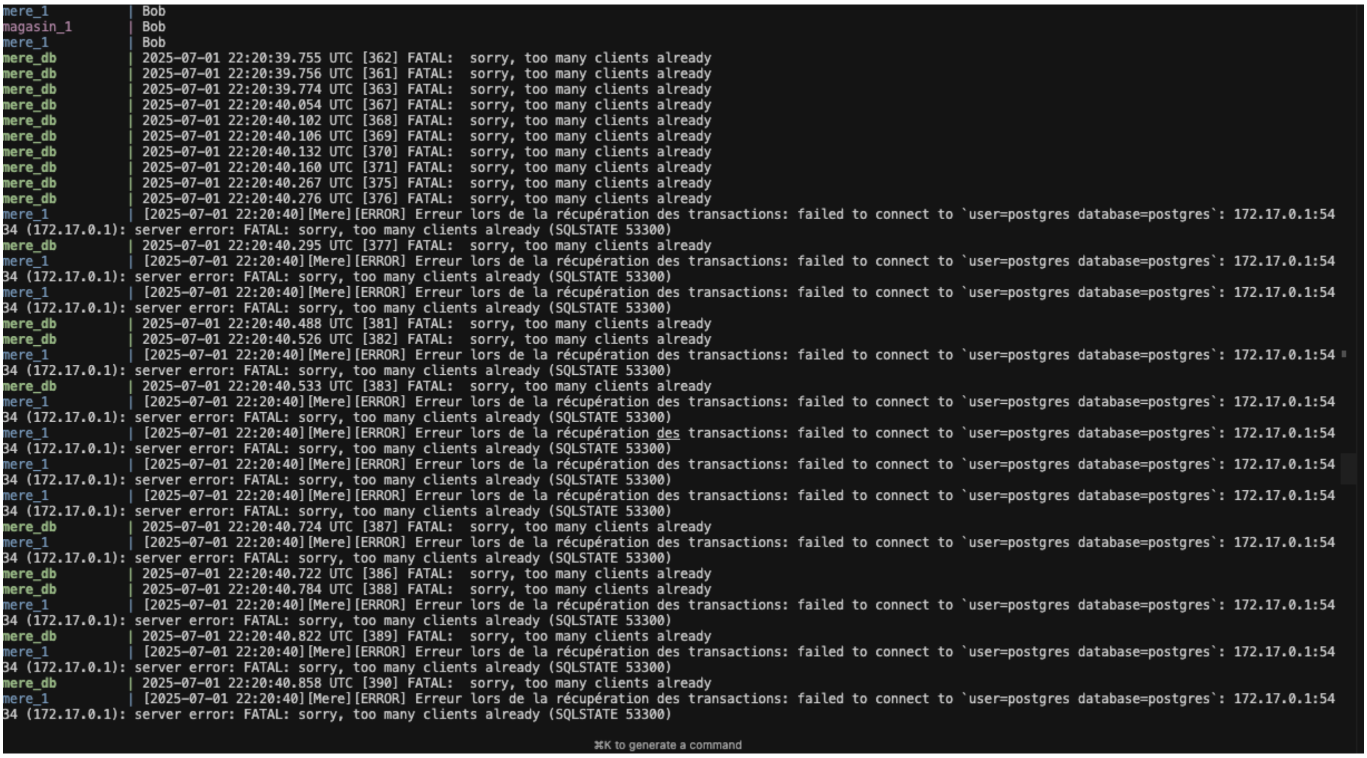
Résultats :

Mon application fonctionne bien à 1 RPS (requête par seconde) mais tombe en défaillance à 30 RPS. Le nombre de requêtes sur le service logistique est considérablement plus élevé que les autres à cause des rapports consolidés.

ici on peut voir que le système est stable à 1rps

le service logistique tombe en premier en retournans des erreurs de la base donnée à 30rps

ensuite à 100rps je perd la connection ssh et la base de donner mere tombe



```
mere_1          | Bob
magasin_1       | Bob
mere_1          | Bob
mere_db         | 2025-07-01 22:20:39.755 UTC [362] FATAL:  sorry, too many clients already
mere_db         | 2025-07-01 22:20:39.756 UTC [361] FATAL:  sorry, too many clients already
mere_db         | 2025-07-01 22:20:39.774 UTC [363] FATAL:  sorry, too many clients already
mere_db         | 2025-07-01 22:20:40.054 UTC [367] FATAL:  sorry, too many clients already
mere_db         | 2025-07-01 22:20:40.102 UTC [368] FATAL:  sorry, too many clients already
mere_db         | 2025-07-01 22:20:40.106 UTC [369] FATAL:  sorry, too many clients already
mere_db         | 2025-07-01 22:20:40.132 UTC [370] FATAL:  sorry, too many clients already
mere_db         | 2025-07-01 22:20:40.160 UTC [371] FATAL:  sorry, too many clients already
mere_db         | 2025-07-01 22:20:40.267 UTC [375] FATAL:  sorry, too many clients already
mere_db         | 2025-07-01 22:20:40.276 UTC [376] FATAL:  sorry, too many clients already
mere_1          | [2025-07-01 22:20:40][Mere][ERROR] Erreur lors de la récupération des transactions: failed to connect to `user=postgres database=postgres`: 172.17.0.1:54
34 (172.17.0.1): server error: FATAL: sorry, too many clients already (SQLSTATE 53300)
mere_db         | 2025-07-01 22:20:40.295 UTC [377] FATAL:  sorry, too many clients already
mere_1          | [2025-07-01 22:20:40][Mere][ERROR] Erreur lors de la récupération des transactions: failed to connect to `user=postgres database=postgres`: 172.17.0.1:54
34 (172.17.0.1): server error: FATAL: sorry, too many clients already (SQLSTATE 53300)
mere_1          | [2025-07-01 22:20:40][Mere][ERROR] Erreur lors de la récupération des transactions: failed to connect to `user=postgres database=postgres`: 172.17.0.1:54
34 (172.17.0.1): server error: FATAL: sorry, too many clients already (SQLSTATE 53300)
mere_db         | 2025-07-01 22:20:40.488 UTC [381] FATAL:  sorry, too many clients already
mere_db         | 2025-07-01 22:20:40.526 UTC [382] FATAL:  sorry, too many clients already
mere_1          | [2025-07-01 22:20:40][Mere][ERROR] Erreur lors de la récupération des transactions: failed to connect to `user=postgres database=postgres`: 172.17.0.1:54
34 (172.17.0.1): server error: FATAL: sorry, too many clients already (SQLSTATE 53300)
mere_db         | 2025-07-01 22:20:40.533 UTC [383] FATAL:  sorry, too many clients already
mere_1          | [2025-07-01 22:20:40][Mere][ERROR] Erreur lors de la récupération des transactions: failed to connect to `user=postgres database=postgres`: 172.17.0.1:54
34 (172.17.0.1): server error: FATAL: sorry, too many clients already (SQLSTATE 53300)
mere_1          | [2025-07-01 22:20:40][Mere][ERROR] Erreur lors de la récupération des transactions: failed to connect to `user=postgres database=postgres`: 172.17.0.1:54
34 (172.17.0.1): server error: FATAL: sorry, too many clients already (SQLSTATE 53300)
mere_1          | [2025-07-01 22:20:40][Mere][ERROR] Erreur lors de la récupération des transactions: failed to connect to `user=postgres database=postgres`: 172.17.0.1:54
34 (172.17.0.1): server error: FATAL: sorry, too many clients already (SQLSTATE 53300)
mere_1          | [2025-07-01 22:20:40][Mere][ERROR] Erreur lors de la récupération des transactions: failed to connect to `user=postgres database=postgres`: 172.17.0.1:54
34 (172.17.0.1): server error: FATAL: sorry, too many clients already (SQLSTATE 53300)
mere_db         | 2025-07-01 22:20:40.724 UTC [387] FATAL:  sorry, too many clients already
mere_1          | [2025-07-01 22:20:40][Mere][ERROR] Erreur lors de la récupération des transactions: failed to connect to `user=postgres database=postgres`: 172.17.0.1:54
34 (172.17.0.1): server error: FATAL: sorry, too many clients already (SQLSTATE 53300)
mere_db         | 2025-07-01 22:20:40.722 UTC [386] FATAL:  sorry, too many clients already
mere_db         | 2025-07-01 22:20:40.784 UTC [388] FATAL:  sorry, too many clients already
mere_1          | [2025-07-01 22:20:40][Mere][ERROR] Erreur lors de la récupération des transactions: failed to connect to `user=postgres database=postgres`: 172.17.0.1:54
34 (172.17.0.1): server error: FATAL: sorry, too many clients already (SQLSTATE 53300)
mere_db         | 2025-07-01 22:20:40.822 UTC [389] FATAL:  sorry, too many clients already
mere_1          | [2025-07-01 22:20:40][Mere][ERROR] Erreur lors de la récupération des transactions: failed to connect to `user=postgres database=postgres`: 172.17.0.1:54
34 (172.17.0.1): server error: FATAL: sorry, too many clients already (SQLSTATE 53300)
mere_db         | 2025-07-01 22:20:40.858 UTC [390] FATAL:  sorry, too many clients already
mere_1          | [2025-07-01 22:20:40][Mere][ERROR] Erreur lors de la récupération des transactions: failed to connect to `user=postgres database=postgres`: 172.17.0.1:54
34 (172.17.0.1): server error: FATAL: sorry, too many clients already (SQLSTATE 53300)

⌘K to generate a command
```

Analysez les points faibles de l'architecture:

1. il y a trops d'appel ver le service logistique, il faut donc cache les réponses du service
2. il faut aussi cache les résultats de la base de donné lors du listing des produits dans les magasins.

3. il fau aussi cache les résultat de la db quand on liste les transactions pour la base de donne mere

Ajout d’un Load Balancer et résilience

J'ai décidé d'utiliser Traefik pour mon load balancing car, traefik est plus récent et est construit avec go and se configure sur docker-compose facilement. et l'orchestration via Docker Compose.

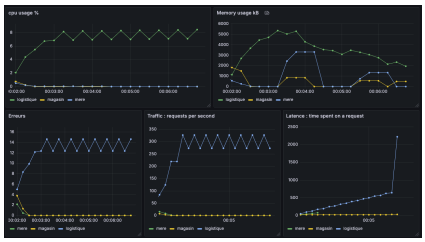
Analyse par nombre d:instance :

Je decide de scale avec un nombre N le service de logistique car ce service est celui qui reçois le plus de requête Dans ces testes je perd la connection ssh à 30 rps

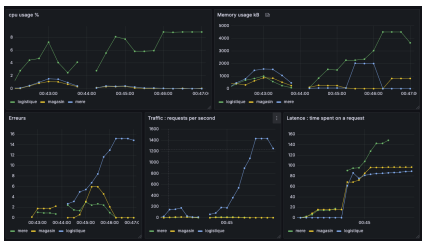
N = 1 instance

RPS = 1	RPS = 15	RPS = 30
		

N = 2 instances

RPS = 1	RPS = 15	RPS = 30
		

N = 3 instances

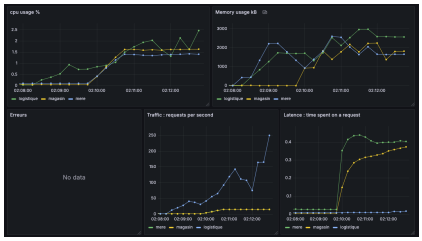
RPS = 1	RPS = 15	RPS = 30
		

N = 4 instances

RPS = 1	RPS = 15	RPS = 30

RPS = 1

RPS = 15



RPS = 30