

# Rapport de Tests de Performance

---

Ce rapport présente les résultats des tests de charge et de performance effectués sur les différentes configurations d'API REST du Système de Gestion de Caisse (SGC). Chaque test inclut des mesures sur la latence, les erreurs, le trafic et un rapport complet généré par K6.

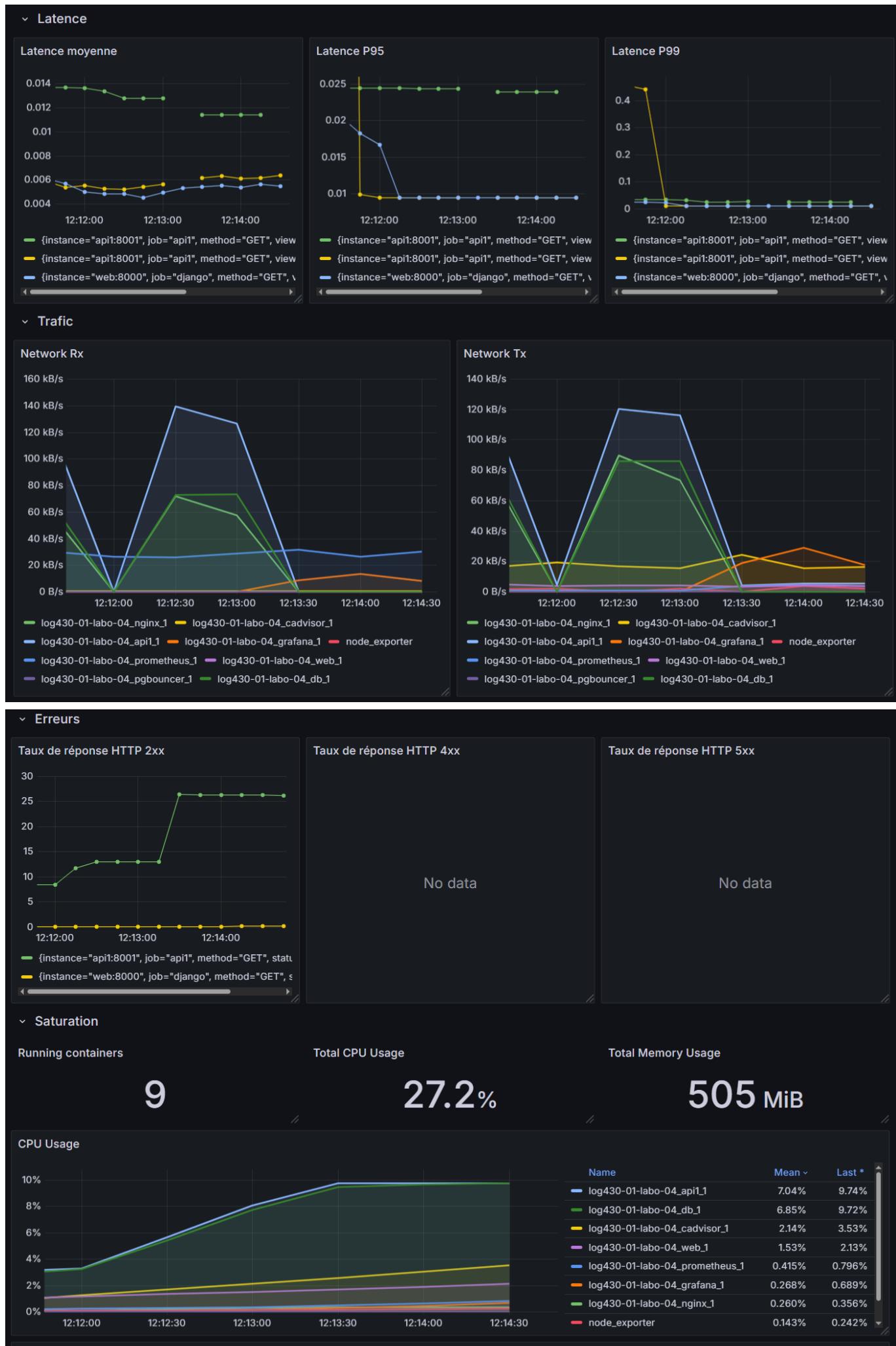
---

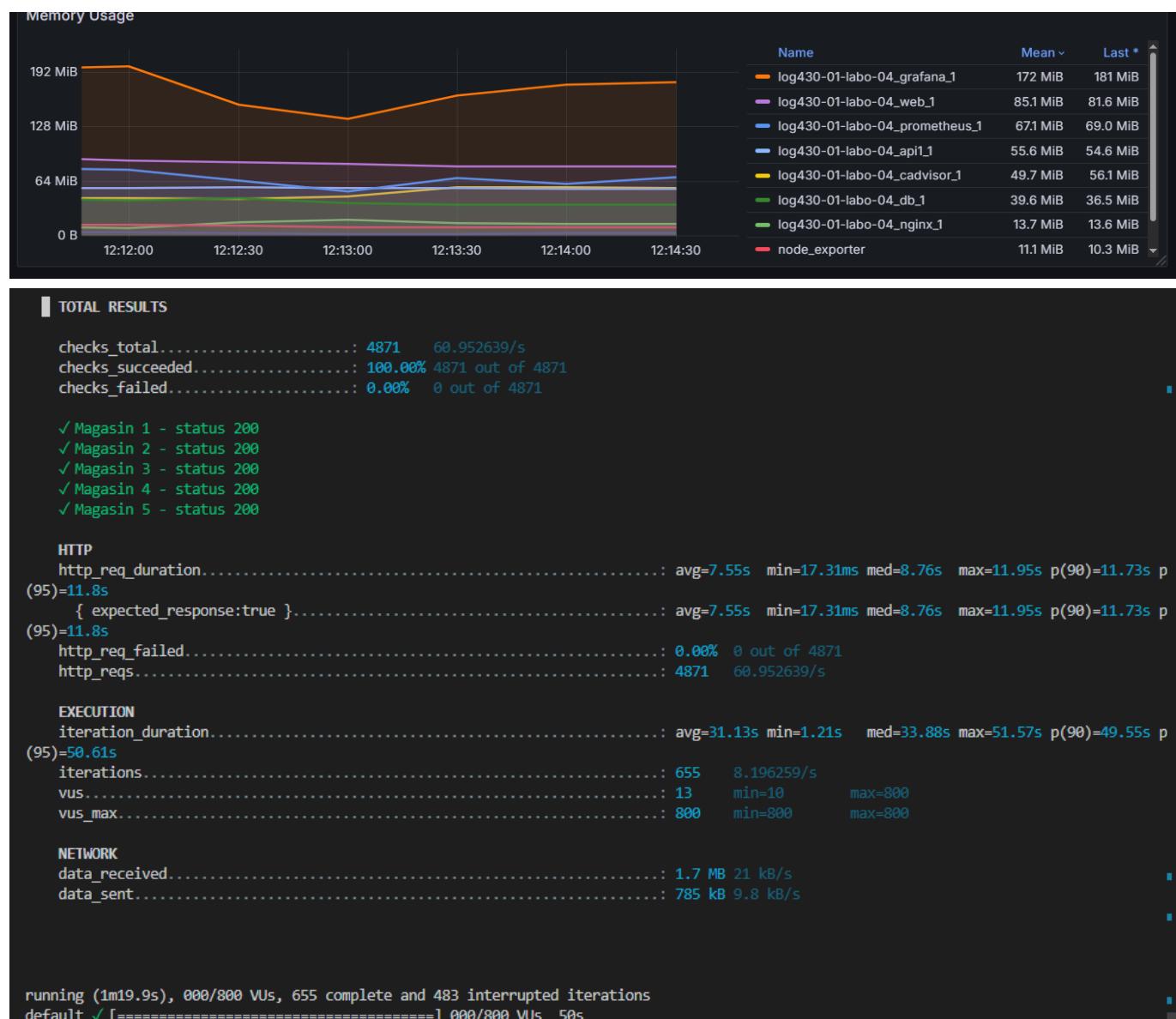
## Test 0 – 1 API Sans Cache

**But :** Évaluer les performances d'une instance unique d'API sans mécanisme de cache.

### testConsultationSimultanée

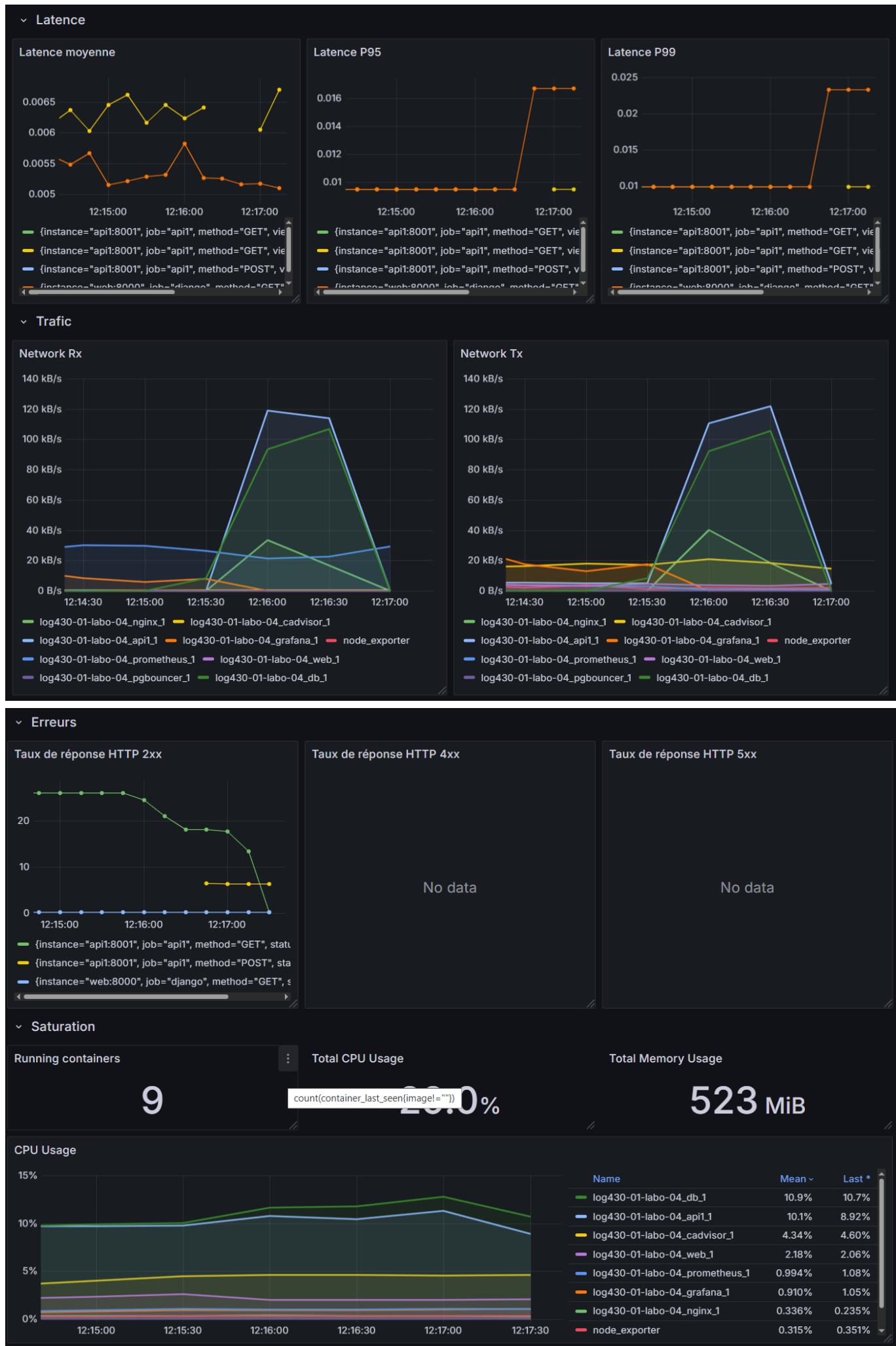
- **Observation :** Très forte latence ( $>7s$ ) et saturation dès peu de requêtes.
- **Conclusion :** Le cache semble essentiel pour les lectures concurrentes.

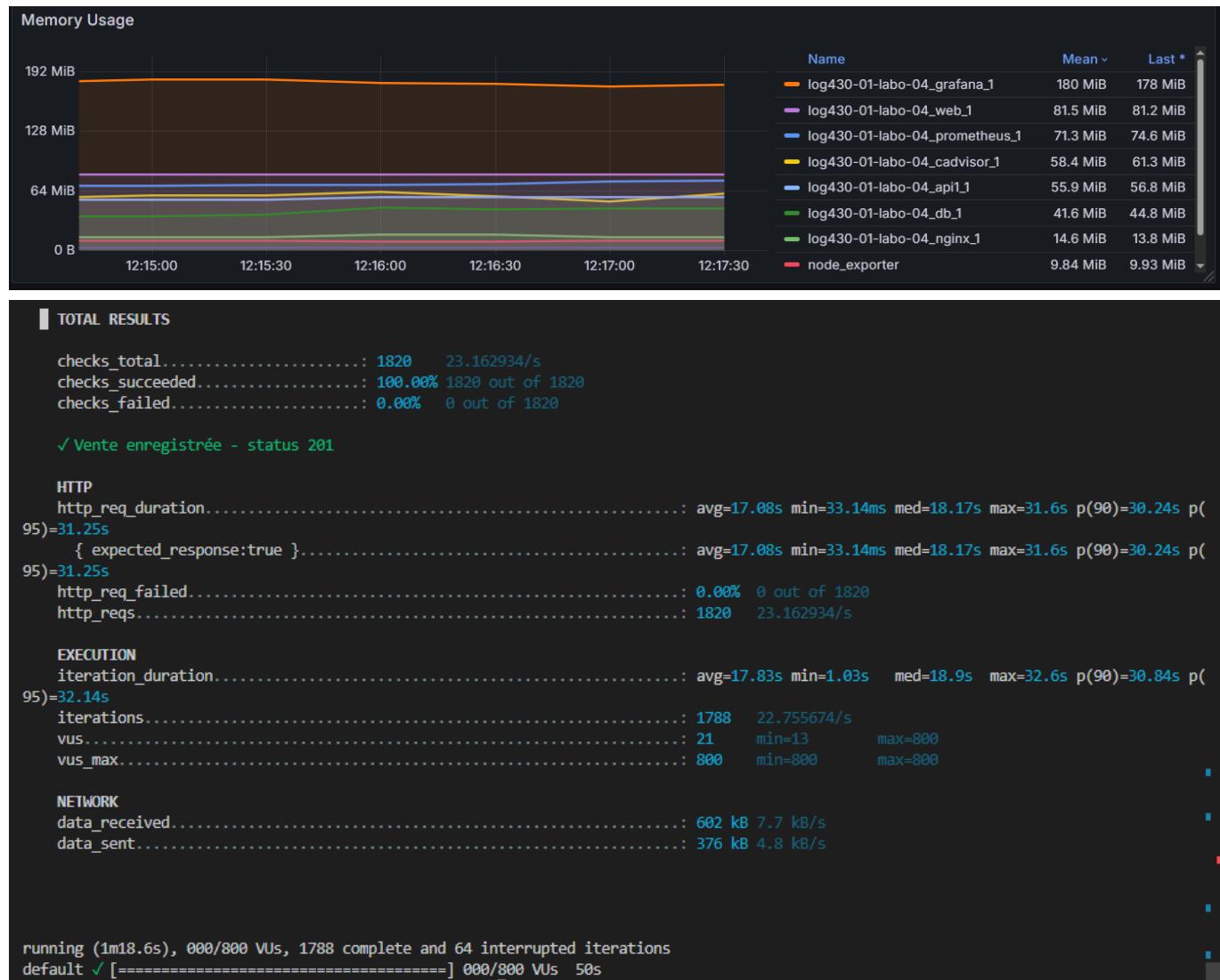




## testEnregistrementVentes

- **Observation** : Latence critique, performances dégradées, erreurs en forte montée.
- **Conclusion** : Inadapté pour un environnement réel sans optimisation.



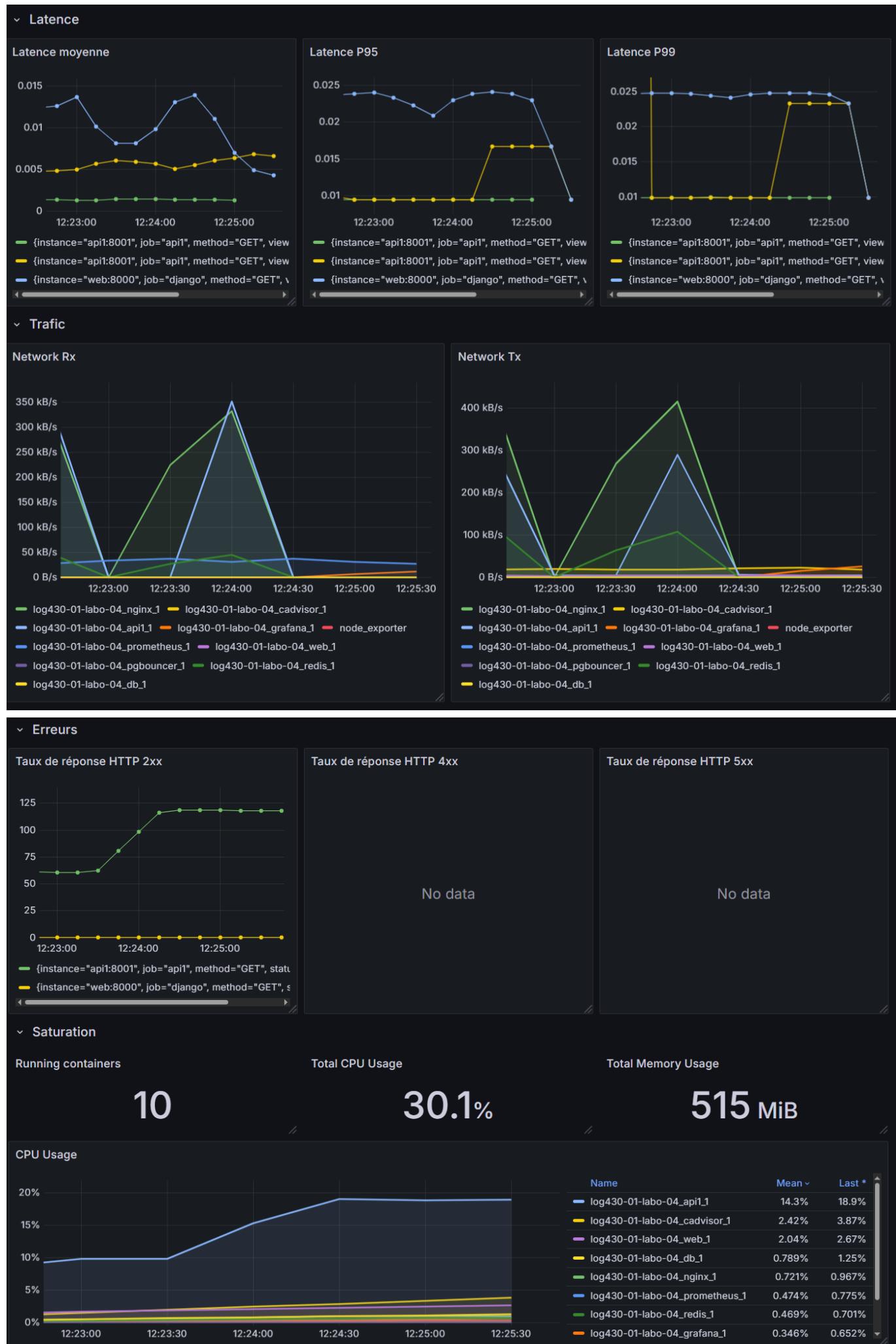


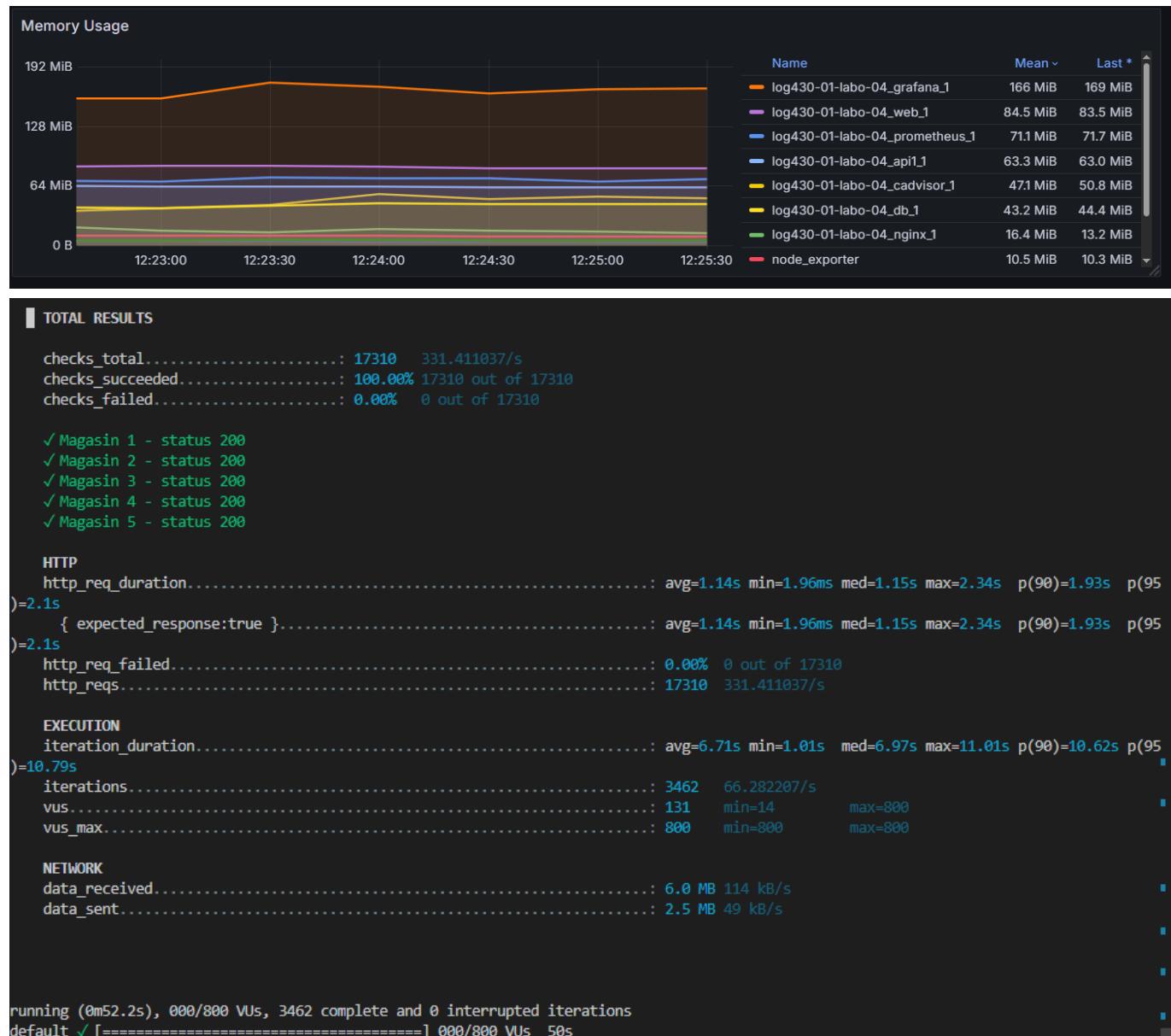
## Test 1 – 1 API

**But :** Tester une instance d'API avec cache activé.

### testConsultationSimultanée

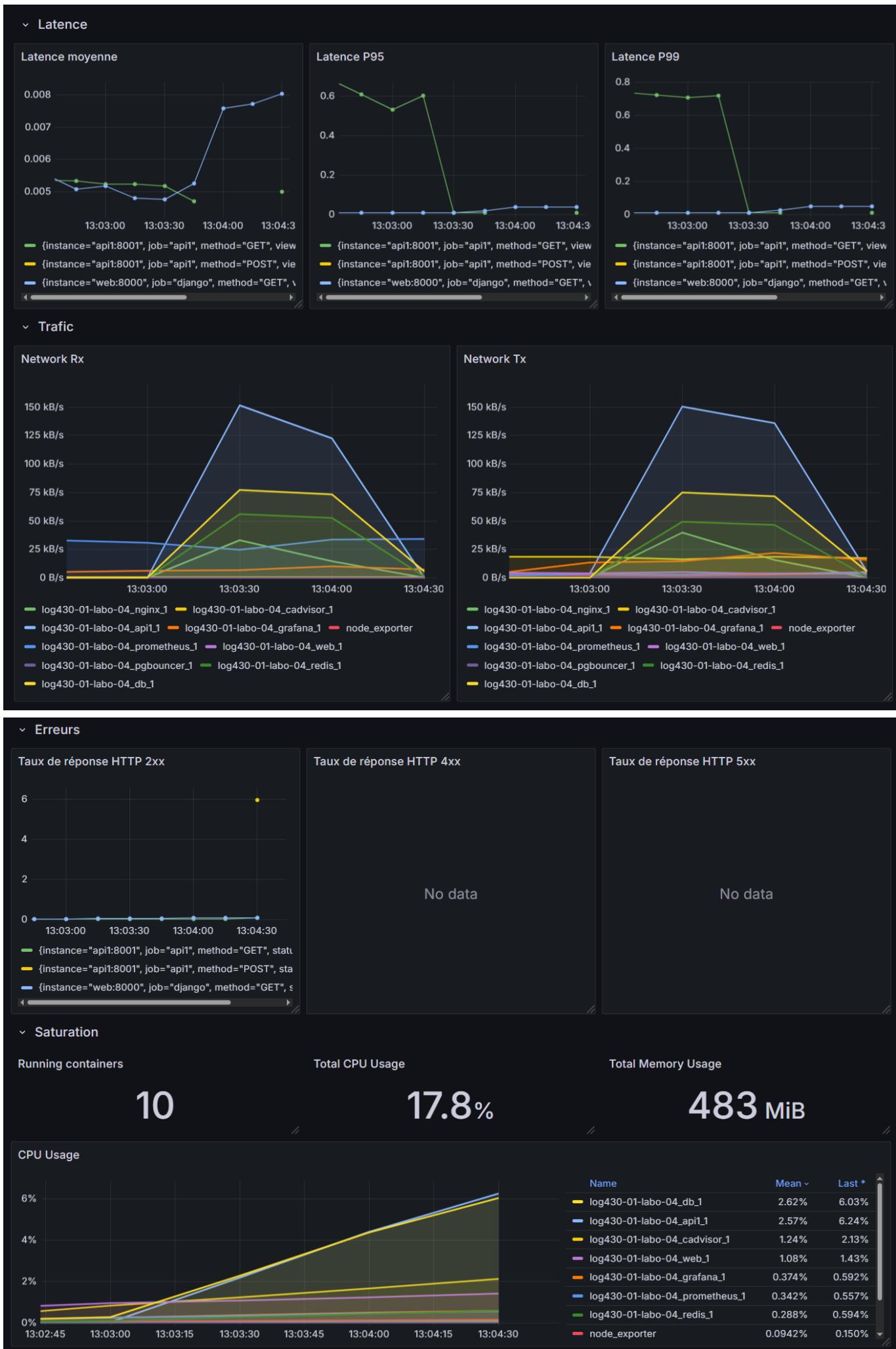
- **Observation** : Gain significatif en latence (~1s), meilleure stabilité.
- **Conclusion** : Le cache améliore clairement la scalabilité lecture seule.

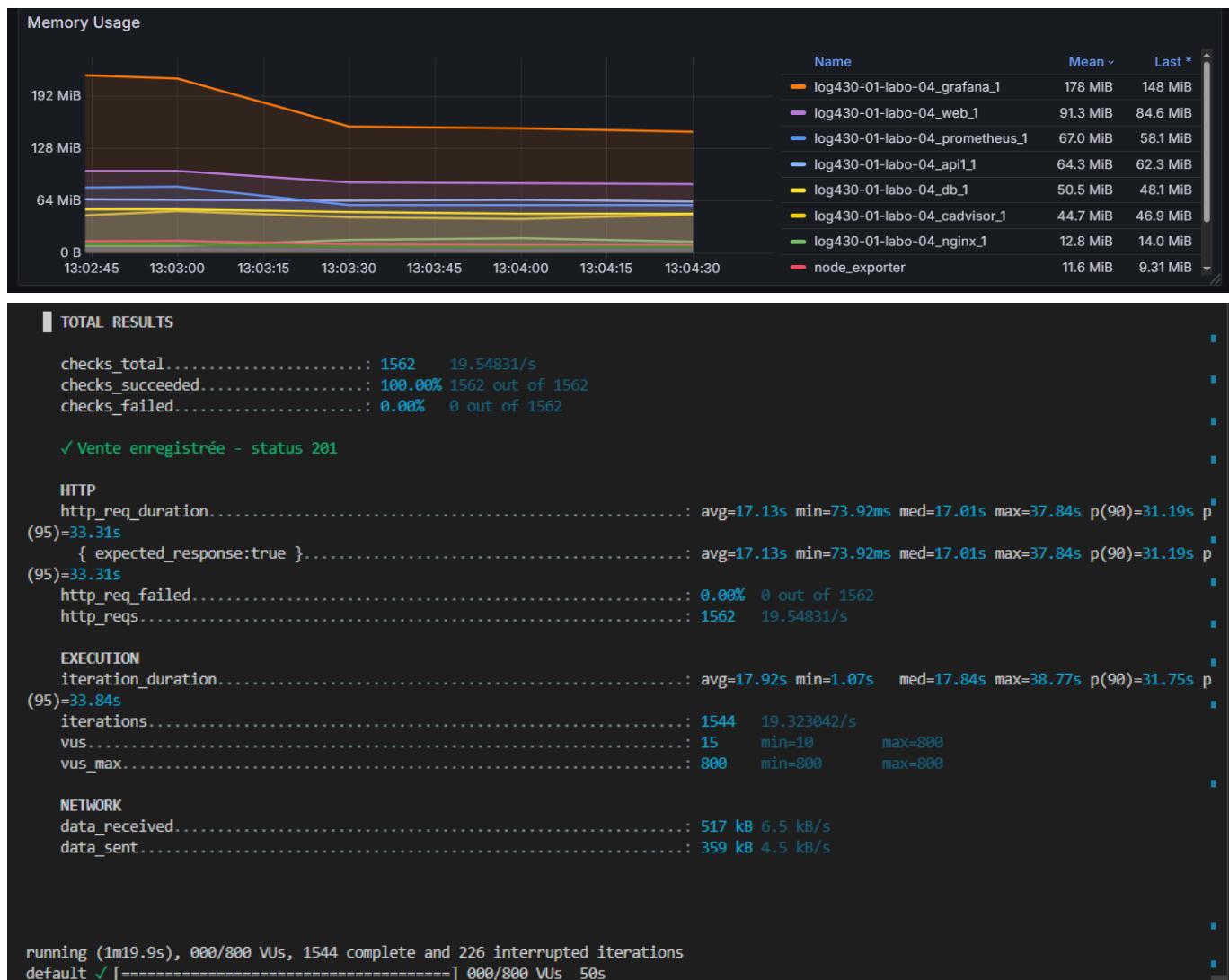




## testEnregistrementVentes

- Observation :** Latence divisée par plus de 10 vs sans cache, trafic acceptable.
- Conclusion :** Système utilisable avec une seule API si le cache est activé.

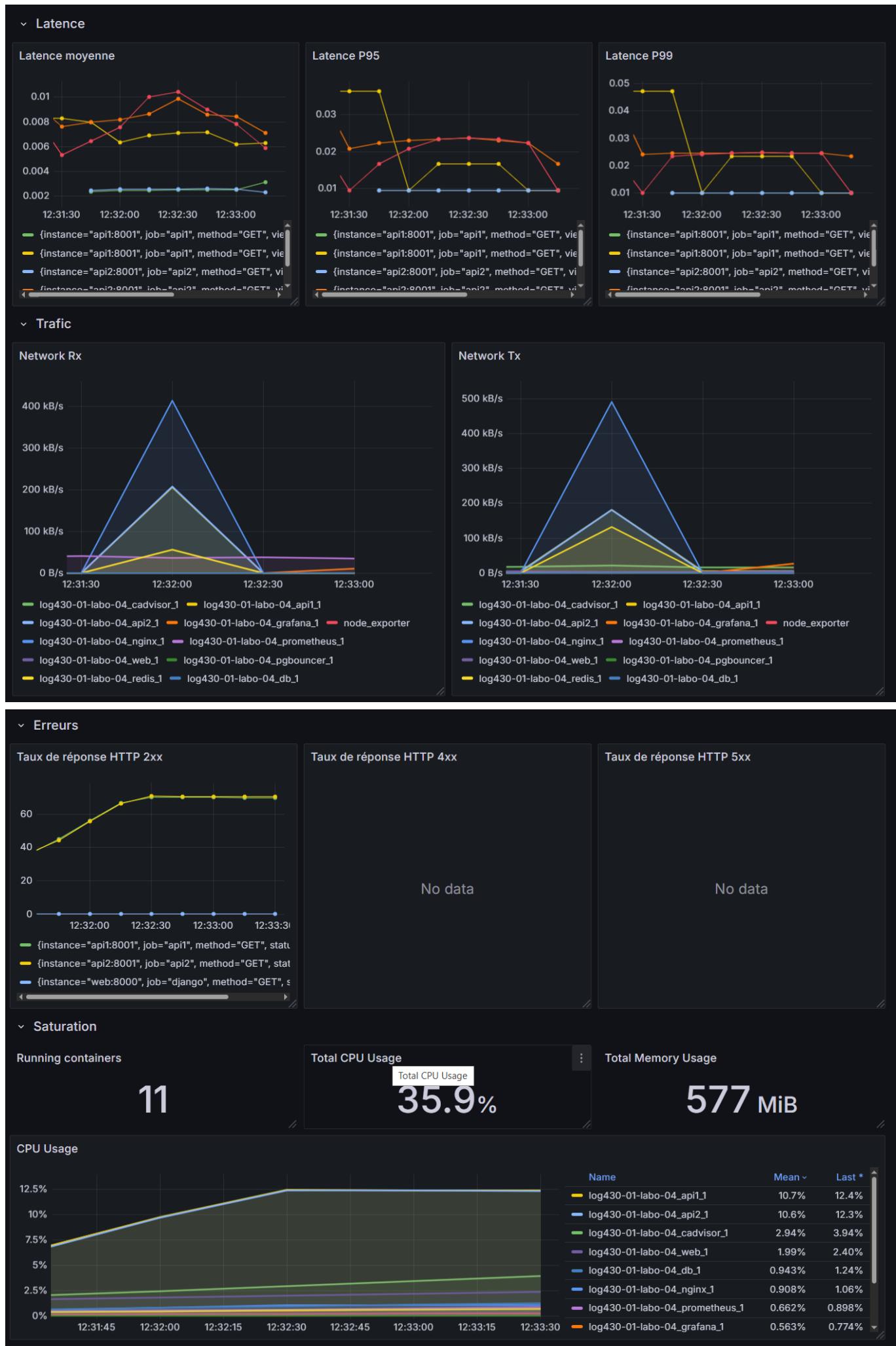


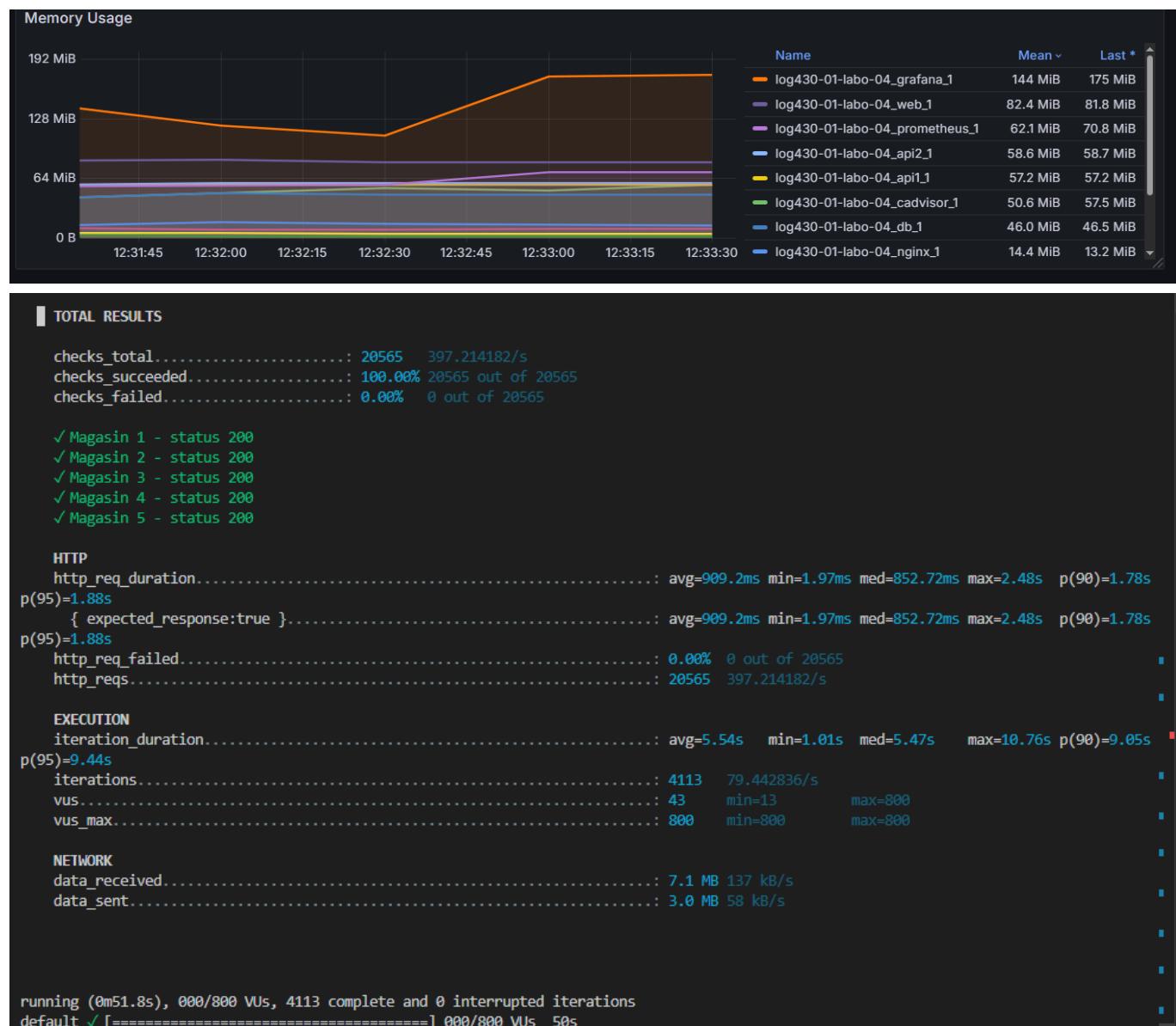


## Test 2 – 2 APIs

### testConsultationSimultanée

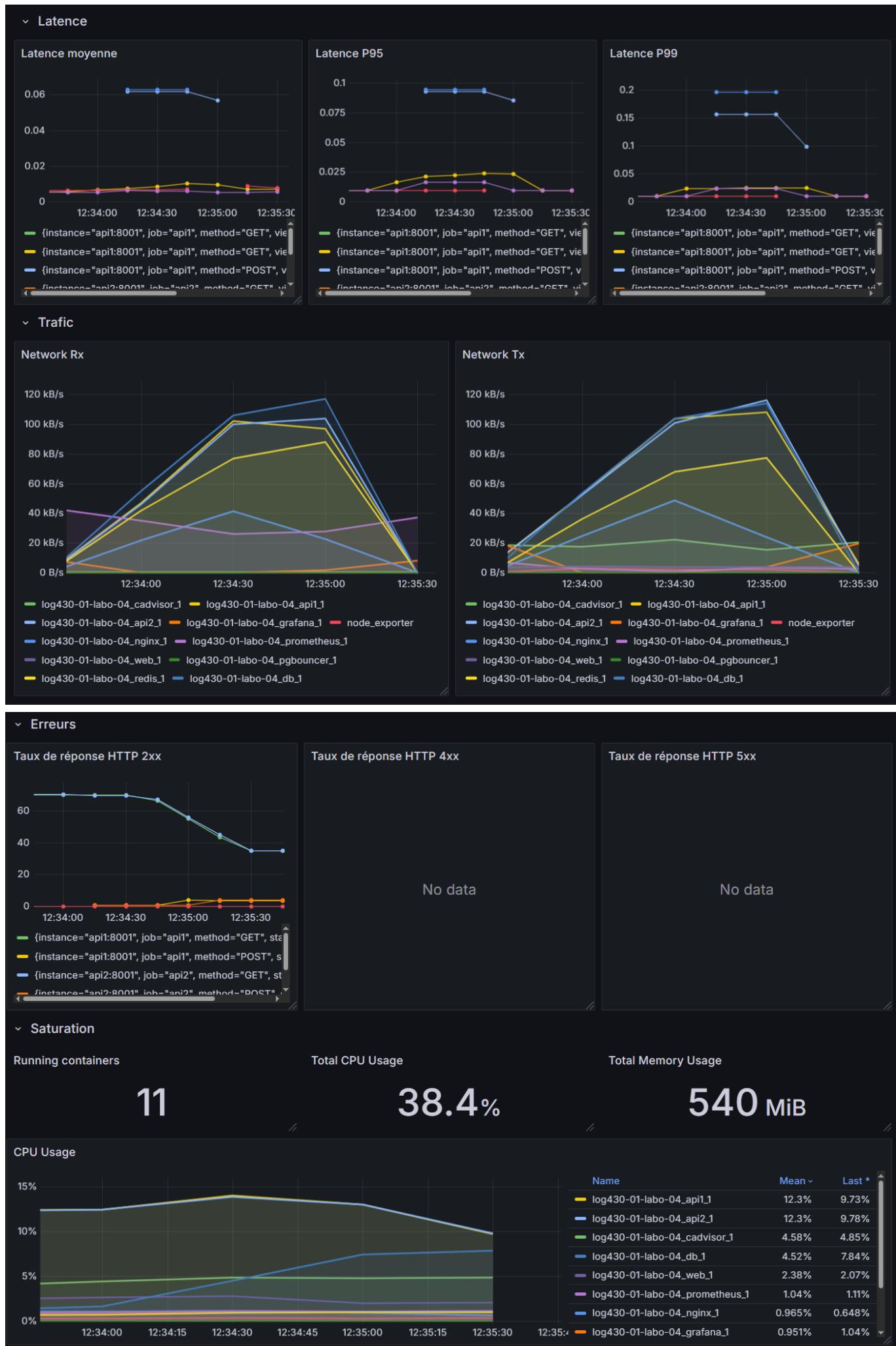
- Observation** : Latence réduite à ~900 ms, bien plus de requêtes servies.
- Conclusion** : Bonne amélioration, mais le CPU monte significativement.

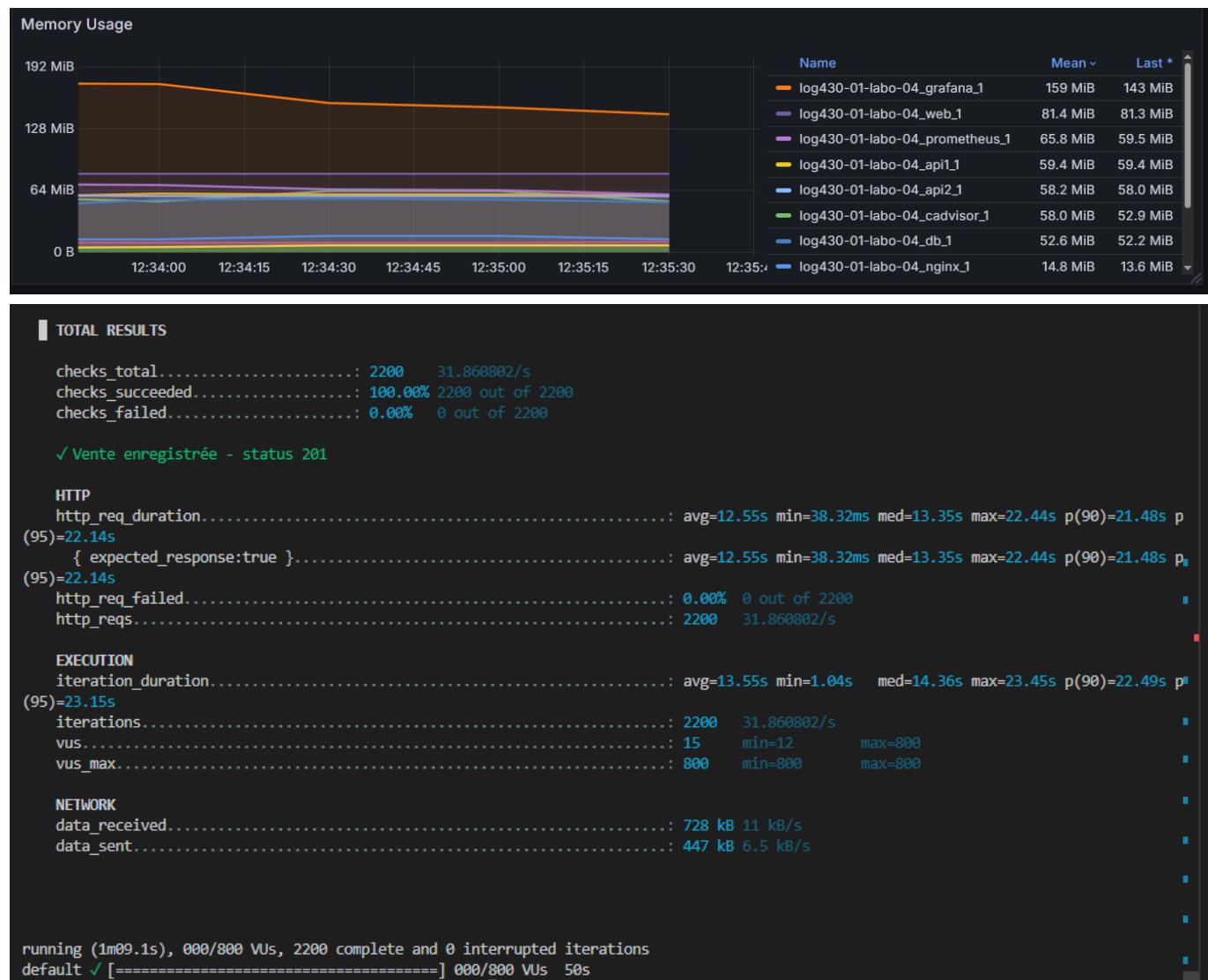




## testEnregistrementVentes

- **Observation** : Comportement stable, quelques erreurs en pointe.
- **Conclusion** : Suffisant pour une charge modérée, mais à surveiller.

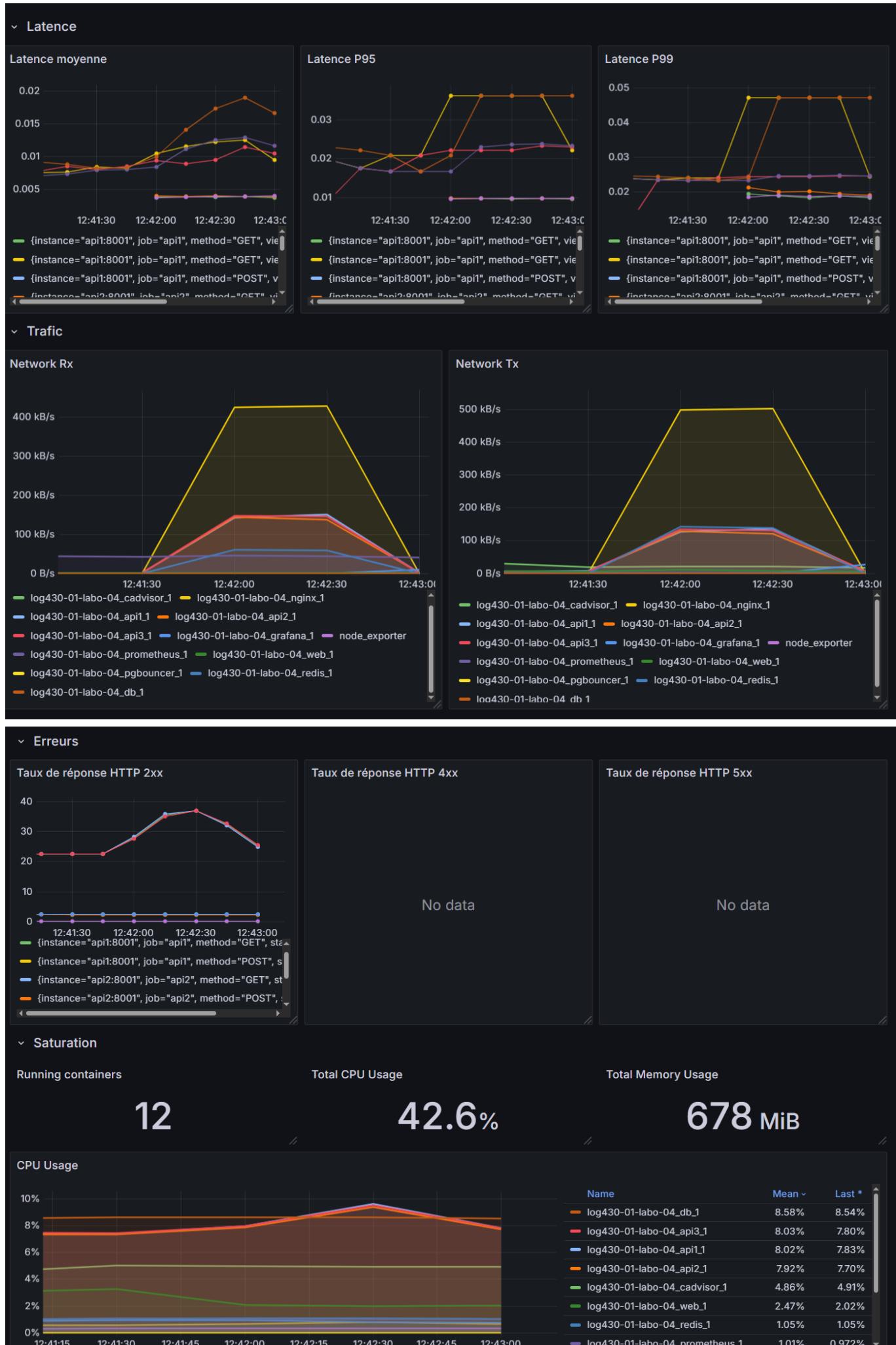


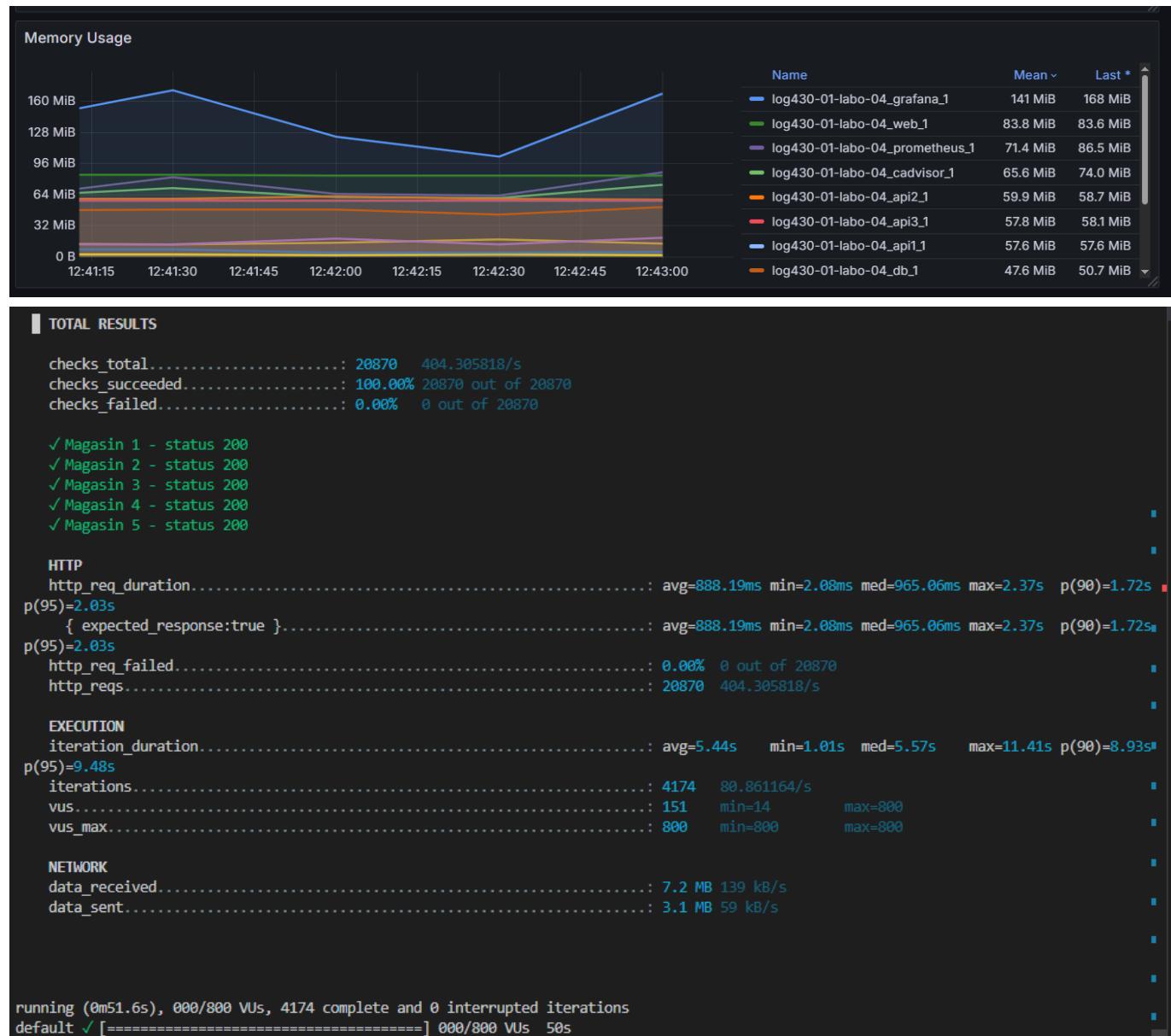


## Test 3 – 3 APIs

testConsultationSimultanée

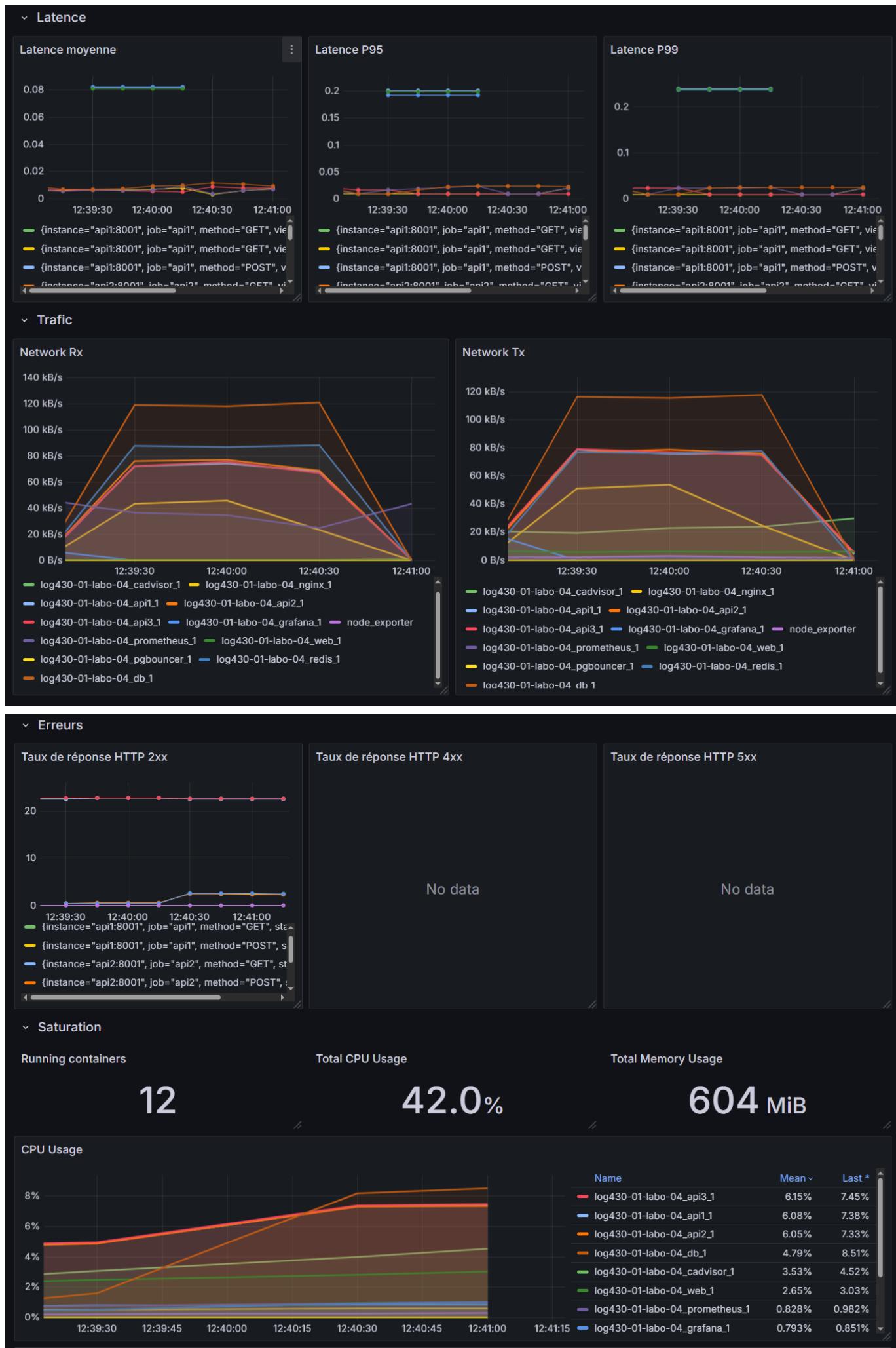
- **Observation** : Amélioration continue, latence stable autour de 880 ms.
- **Conclusion** : Bon équilibre pour des charges moyennes.

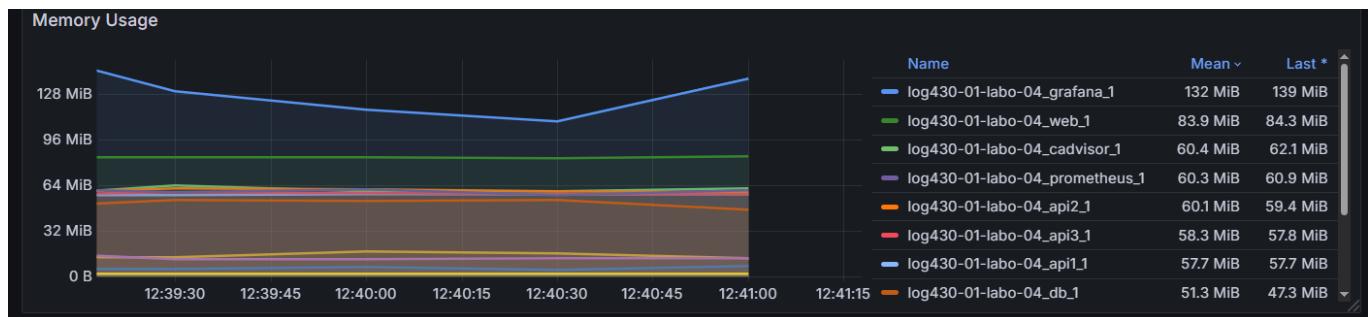




## testEnregistrementVentes

- **Observation** : CPU utilisé plus intensément, mais sans saturation.
- **Conclusion** : Performant sans erreur, un bon seuil de base.





**TOTAL RESULTS**

```

checks_total.....: 2304  33.482588/s
checks_succeeded.: 100.00% 2304 out of 2304
checks_failed....: 0.00%  0 out of 2304

✓ Vente enregistrée - status 201

HTTP
http_req_duration.....: avg=11.88s min=36.86ms med=12.18s max=22.54s p(90)=21.31s p(95)=21.91s
{ expected_response:true }.....: avg=11.88s min=36.86ms med=12.18s max=22.54s p(90)=21.31s p(95)=21.91s
http_req_failed.....: 0.00%  0 out of 2304
http_reqs.....: 2304  33.482588/s

EXECUTION
iteration_duration.....: avg=12.88s min=1.03s  med=13.18s max=23.54s p(90)=22.31s p(95)=22.92s
iterations.....: 2304  33.482588/s
vus.....: 6      min=6   max=800
vus_max.....: 800   min=800 max=800

NETWORK
data_received.....: 763 kB 11 kB/s
data_sent.....: 468 kB 6.8 kB/s

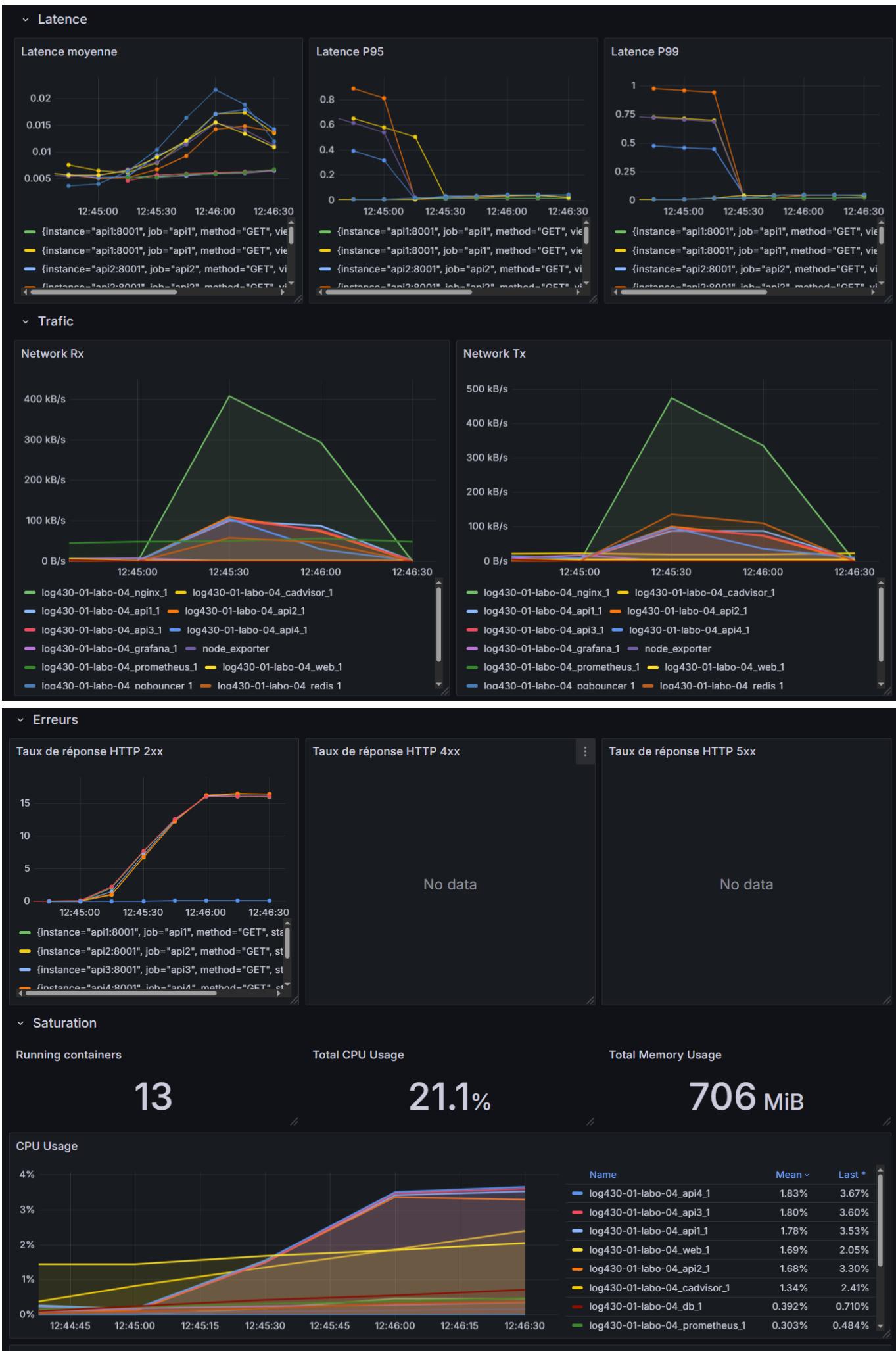
running (1m08.8s), 000/800 VUs, 2304 complete and 0 interrupted iterations
default ✓ [=====] 000/800 VUs 50s

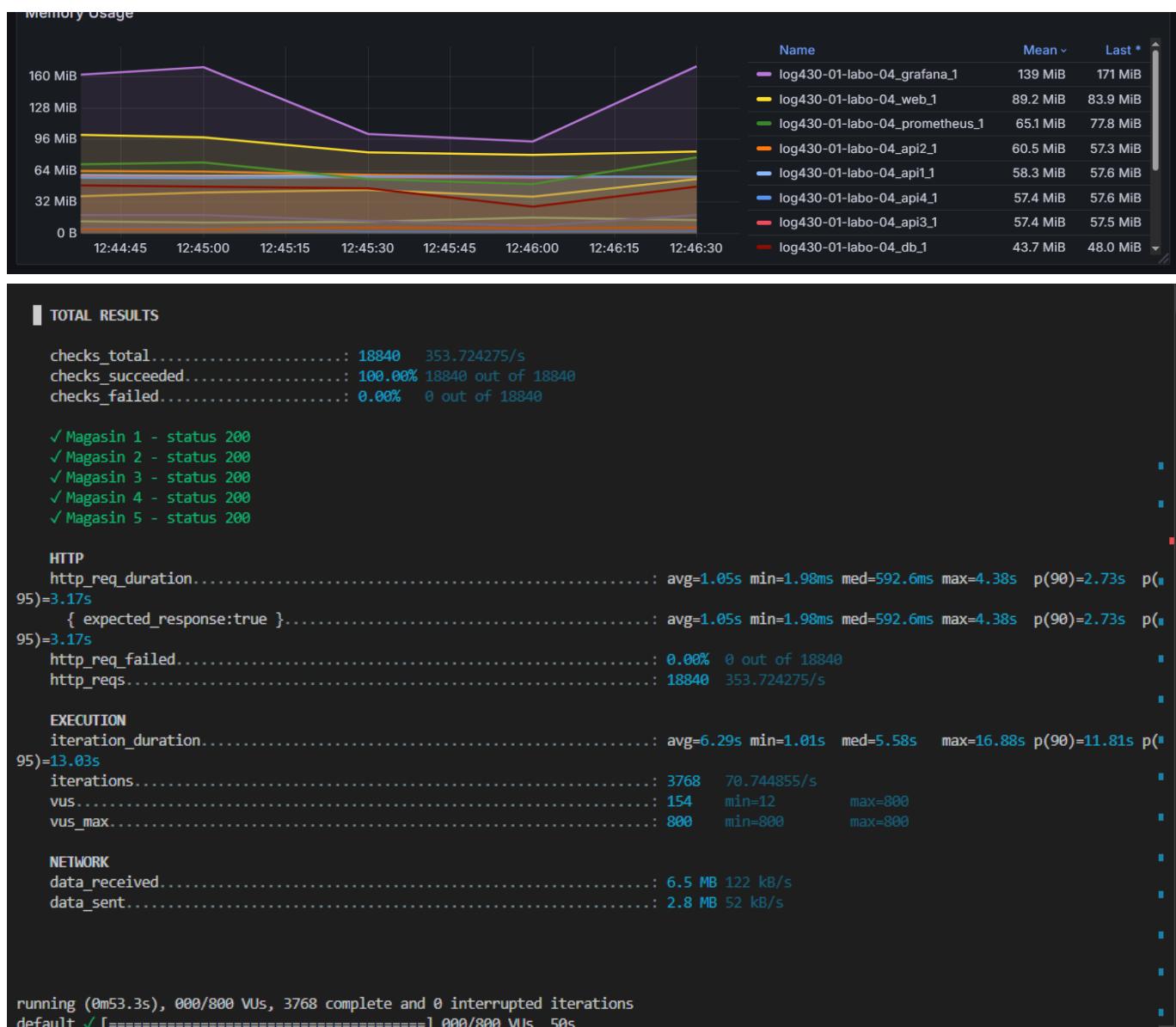
```

## Test 4 – 4 APIs

### testConsultationSimultanée

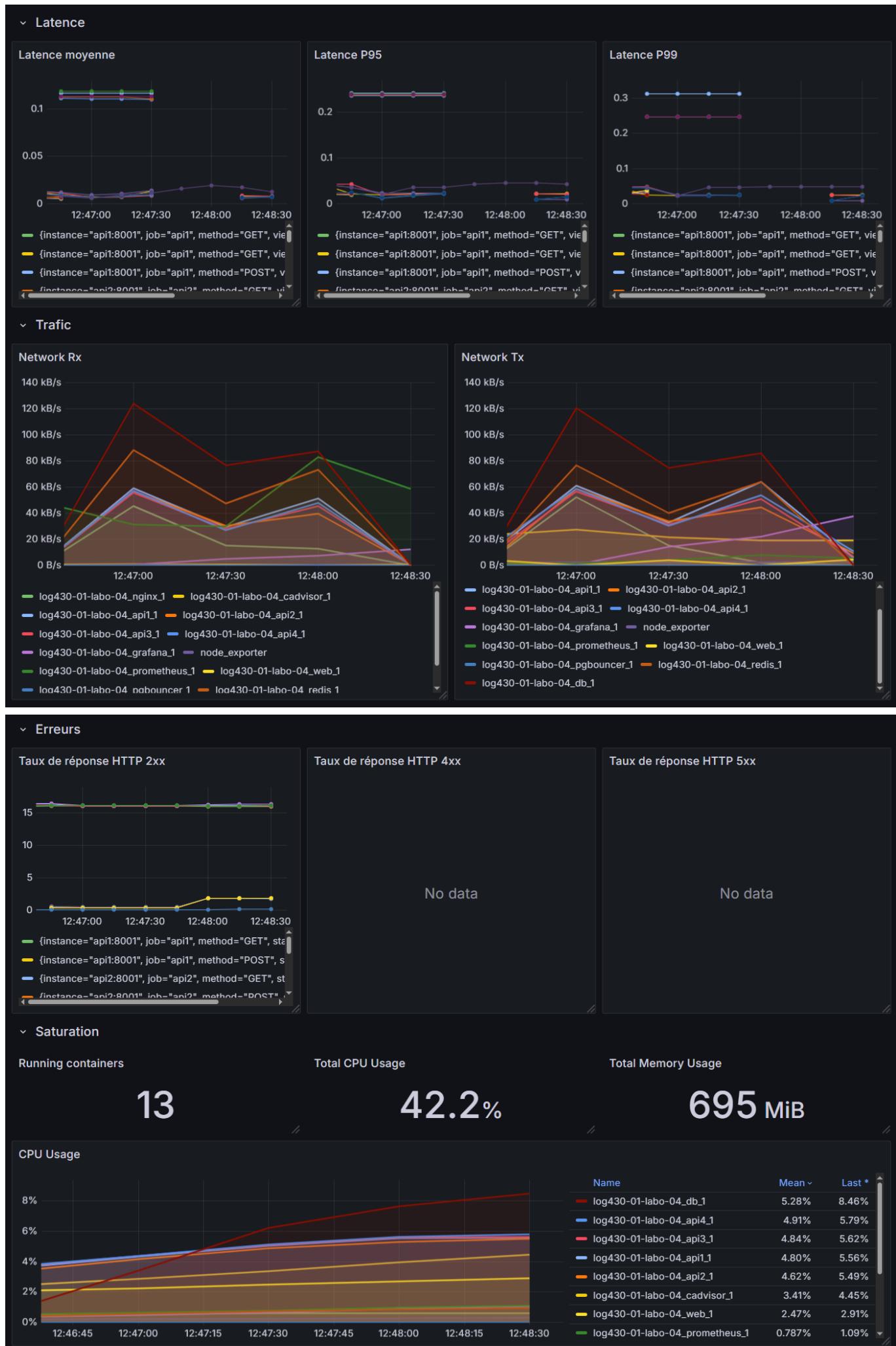
- **Observation :** Usage mémoire élevé, mais latence stable.
- **Conclusion :** Permet un volume élevé sans erreurs.

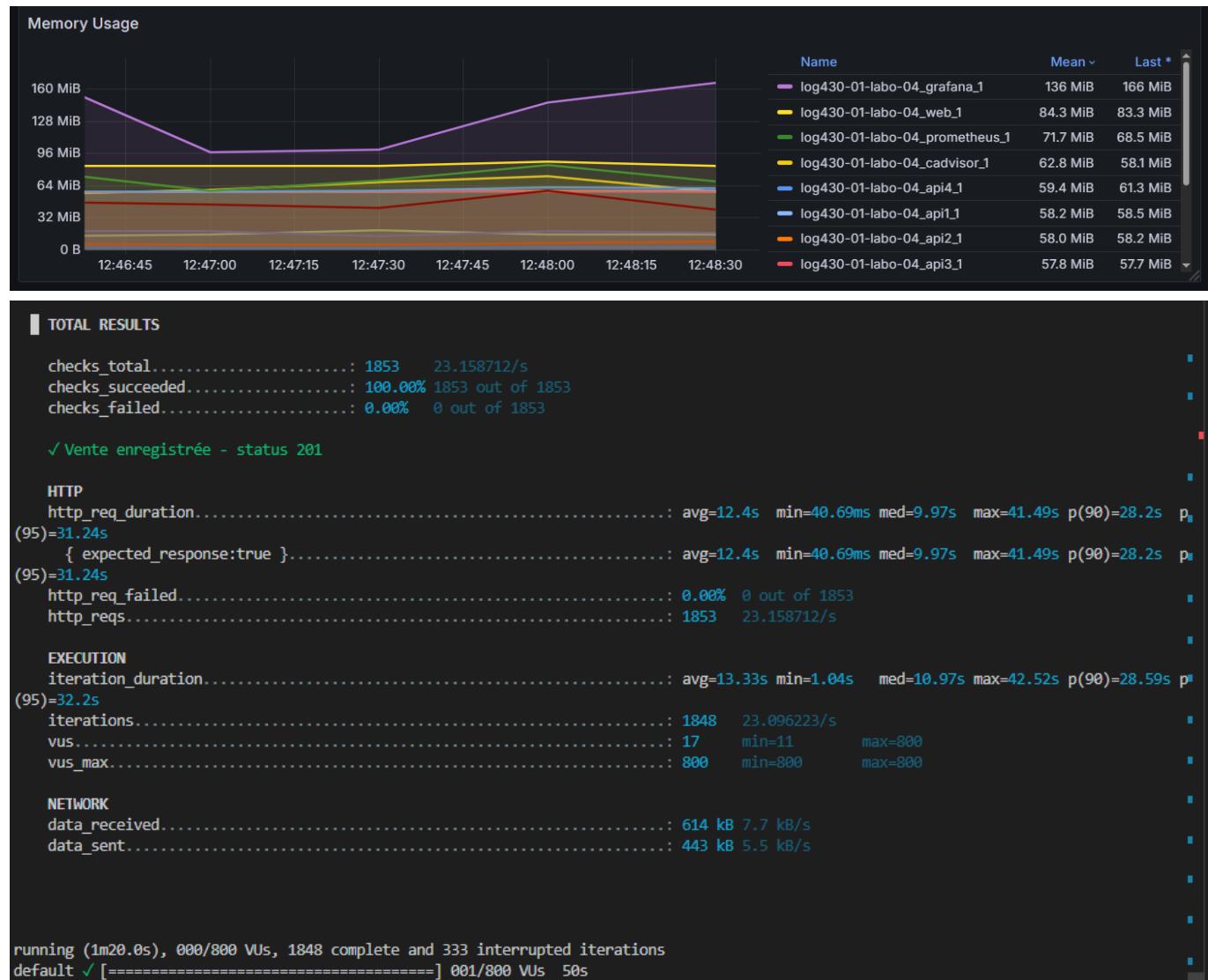




## testEnregistrementVentes

- **Observation** : Résultats comparables à 3 APIs, mais plus stable.
- **Conclusion** : Idéal pour la haute disponibilité.

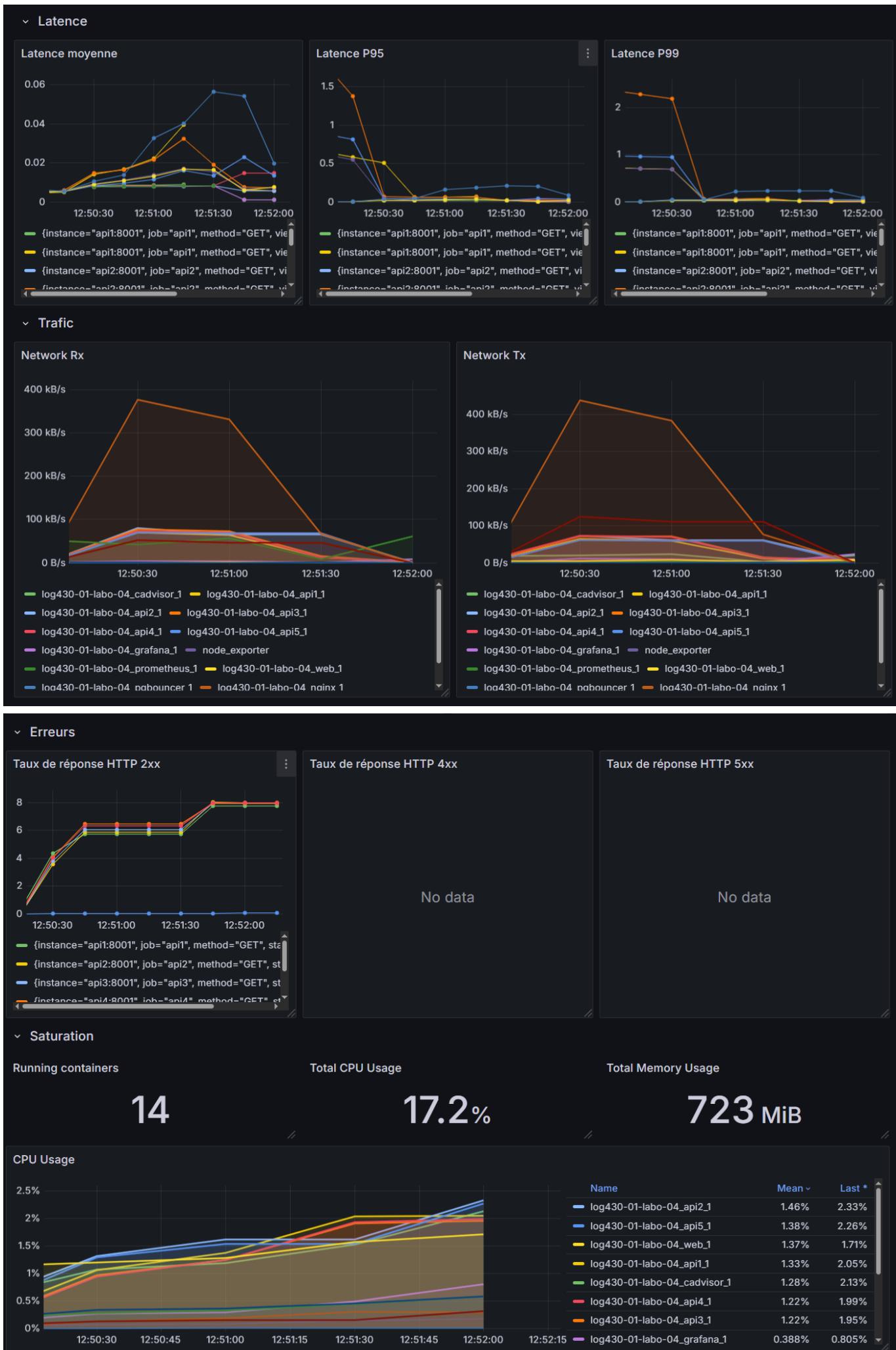


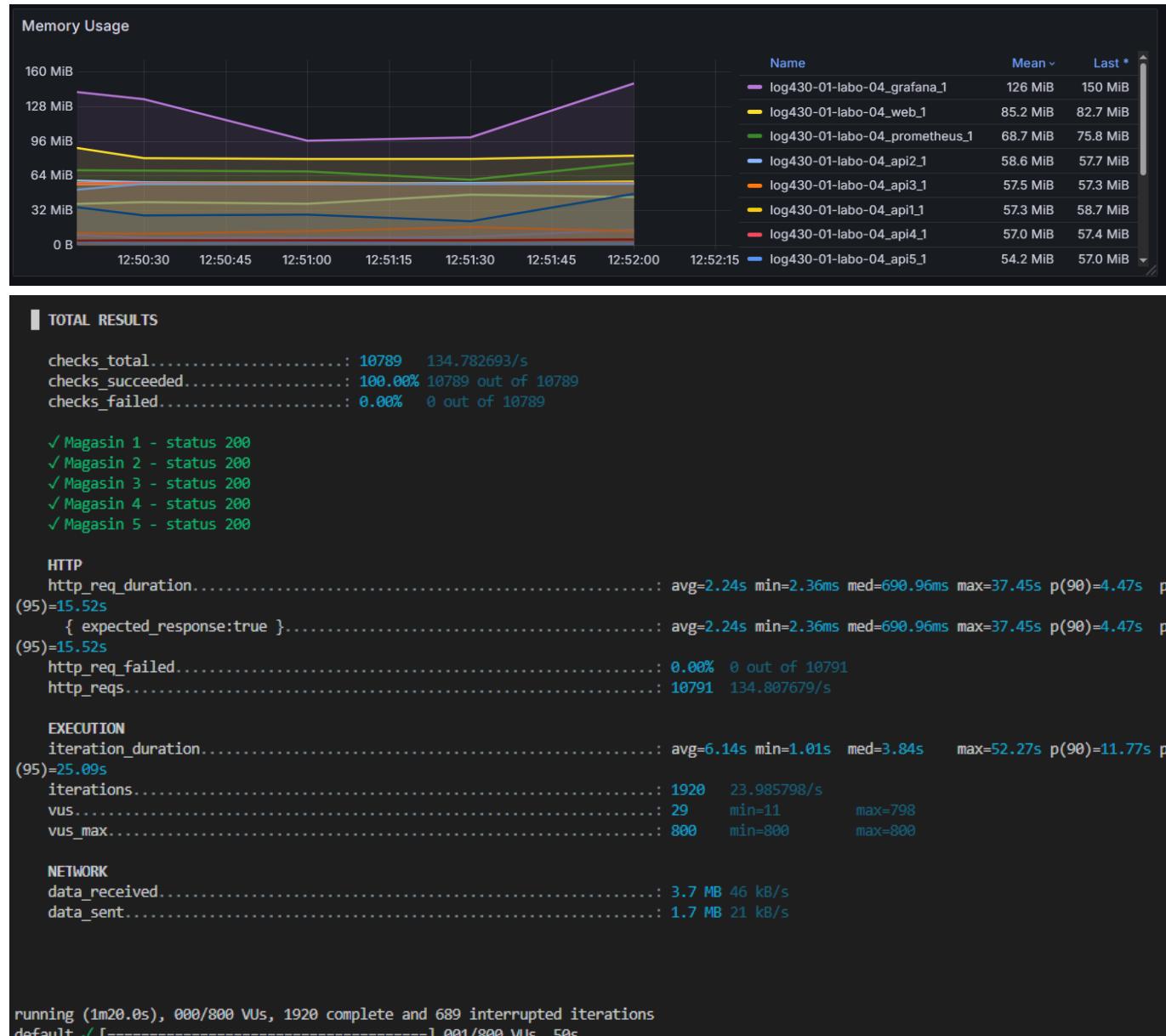


## Test 5 – 5 APIs

### testConsultationSimultanée

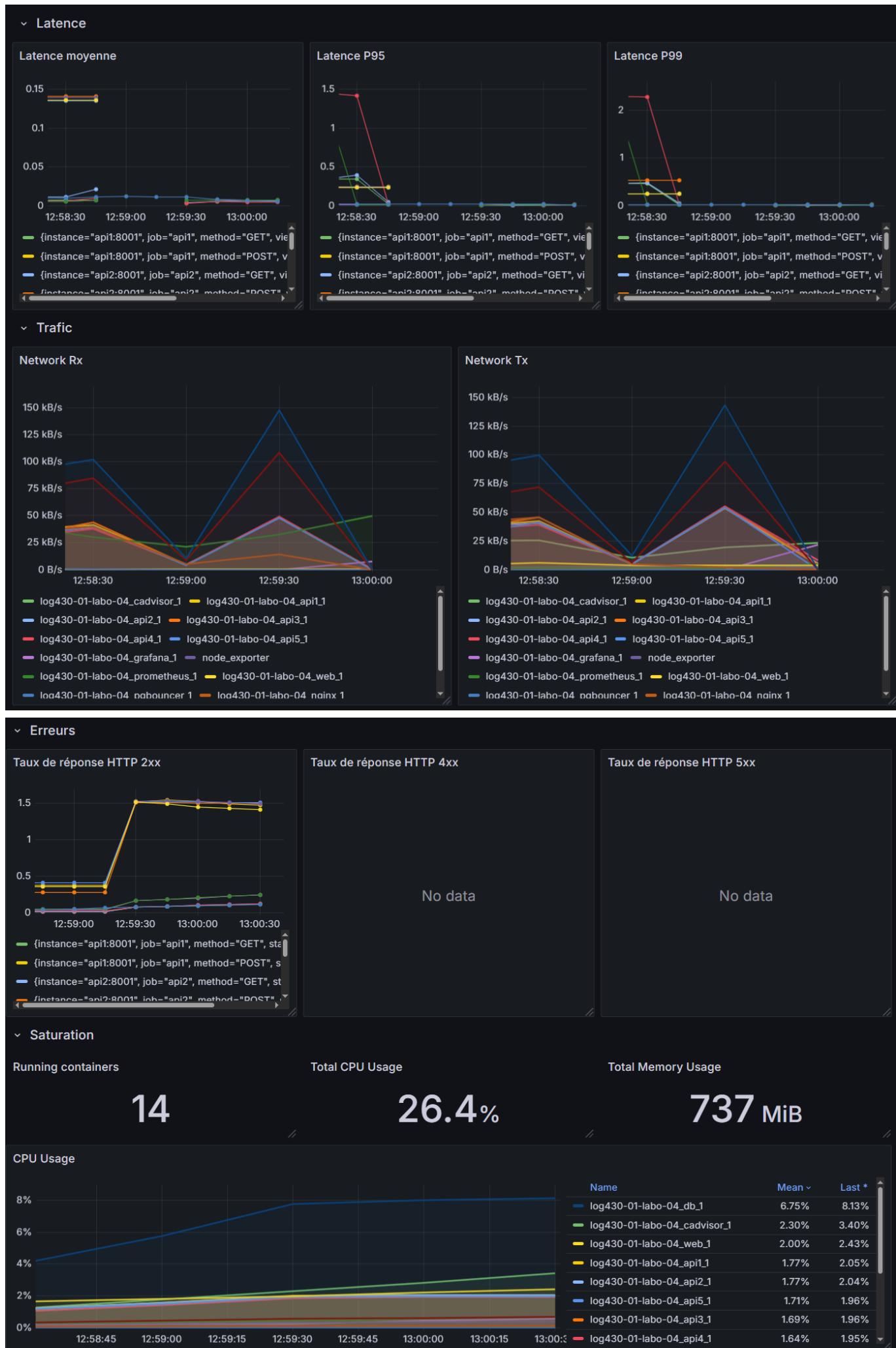
- **Observation :** Diminution du CPU par instance, mais latence peu améliorée.
- **Conclusion :** Rendement marginal décroissant.

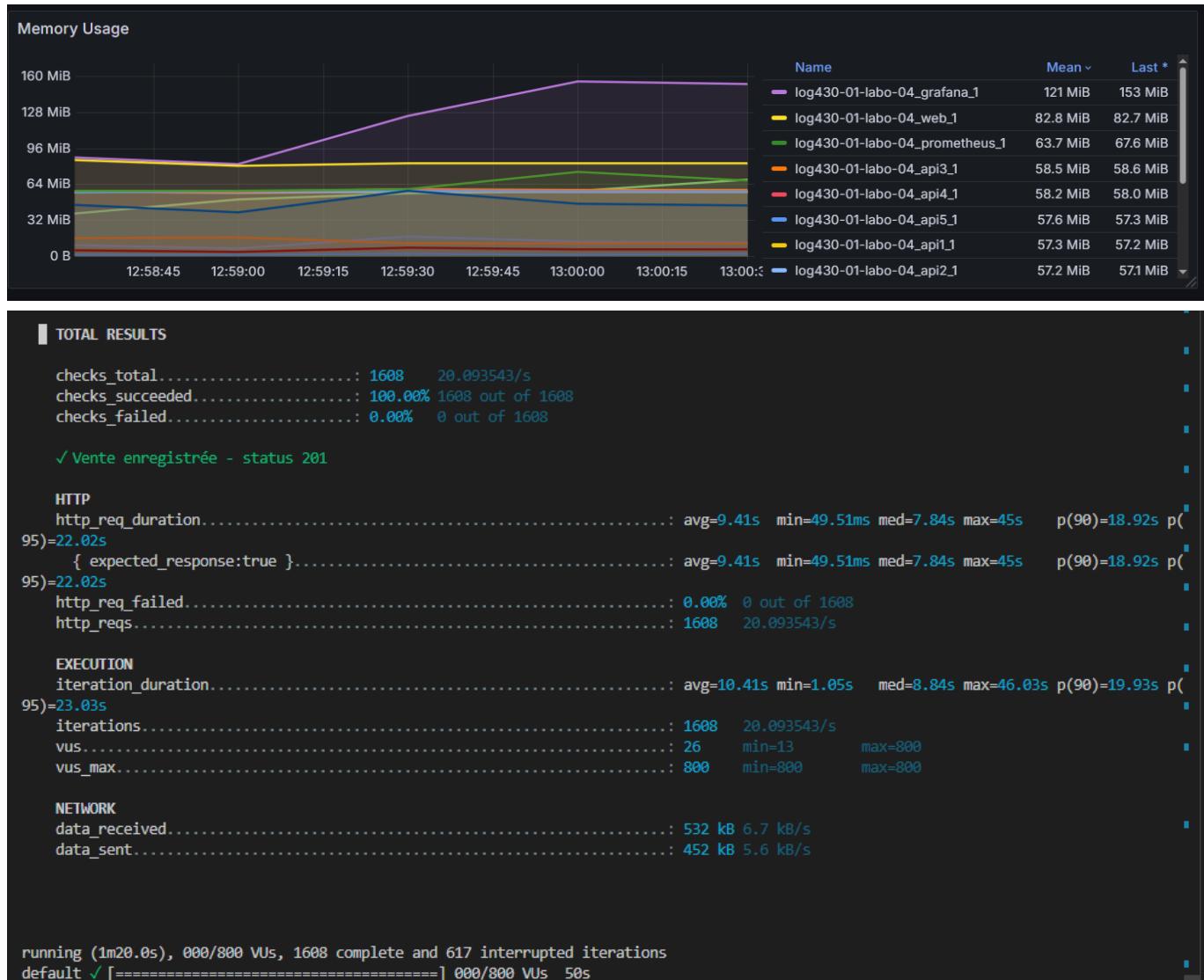




## testEnregistrementVentes

- Observation :** Même comportement que 4 APIs, voire légère régression.
- Conclusion :** Ne vaut pas le coût supplémentaire sans pic extrême.





## Conclusion Générale

- Le cache est **indispensable** dès une seule API.
- La montée en charge avec plusieurs instances **augmente bien le débit**, mais les **gains s'atténuent** à partir de 4.
- Le système reste **fiable jusqu'à 4 APIs** avec une utilisation mémoire stable.
- L'intégration de NGINX et du load balancing a été cruciale pour l'efficacité globale.