

# Platformă pentru monitorizarea unor noduri de tip container/VM

## Descriere

Proiectul pe care îl voi dezvolta este o platformă distribuită pentru monitorizarea în timp real a unor noduri de tip container sau mașini virtuale simulate. Ideea de bază este să construiesc un sistem care să funcționeze similar cu mecanismele interne ale unui cloud modern: fiecare nod din platformă rulează un agent care colectează metrii despre starea sistemului (consum de CPU, memoria utilizată, numărul de procese active etc.) și trimit aceste informații către un serviciu central de colectare. Această colectare este făcută în mod continuu, astfel încât platforma poate oferi o imagine actualizată permanent asupra sănătății și încărcării nodurilor monitorizate.

Pentru accesarea datelor, utilizatorii interacționează cu platforma printr-un API Gateway replicate pentru a asigura că există mai multe instanțe care pot răspunde cererilor în paralel. Această replicare ajută la scalarea aplicației și îi oferă reziliență: dacă o instanță cade, celelalte preiau traficul fără întreruperi. De asemenea, gateway-ul implementează un mecanism de rate-limiting distribuit prin intermediul unui Redis Cluster. Platforma limitează numărul de cereri pe care un utilizator le poate face într-o anumită perioadă, chiar dacă cererile ajung la replici diferite ale gateway-ului, prevenind supraîncărcarea sistemului și protejează platforma de abuzuri sau atacuri de tip DoS.

Datele colectate de agent sunt preluate de un serviciu propriu numit Metrics Collector. Acest serviciu agregă și normalizează informațiile, le salvează într-o bază de date TimescaleDB pentru analiză și, în același timp, le expune către Prometheus, un sistem foarte folosit în monitorizarea serviciilor. Ulterior, aceste metrii sunt afișate vizual și intuitiv în Grafana, unde se pot crea dashboard-uri în timp real ce arată starea fiecărui nod, evoluția consumului de resurse sau eventualele probleme detectate.

Platforma include și un serviciu de alertare care poate notifica administratorii atunci când unul dintre noduri depășește anumite praguri critice, cum ar fi un consum de CPU neobișnuit de mare sau absența raportelor de la agent (ceea ce poate indica faptul că nodul a căzut). În ceea ce privește autentificarea, utilizatorii se conectează printr-un serviciu Keycloak, care implementează autentificare Single Sign-On, iar accesul la date este controlat prin roluri precum „admin” sau „viewer”.

# Arhitectură

Platforma va fi implementată sub forma unui stack Docker Swarm format din următoarele containere:

- Keycloak: autentificare și autorizare (SSO)
- API Gateway (replicat): rutare, validare token, rate-limiting
- Redis Cluster: stocare pentru rate limiting distribuit
- Node Agent: colectare metriki de pe noduri
- Metrics Collector (replicat): agregare și procesare metriki
- TimescaleDB: stocare metriki
- Prometheus: colectare metriki expuse de sistem
- Grafana: dashboard-uri și vizualizare
- Alerting Service: generare notificări pe baza pragurilor



