

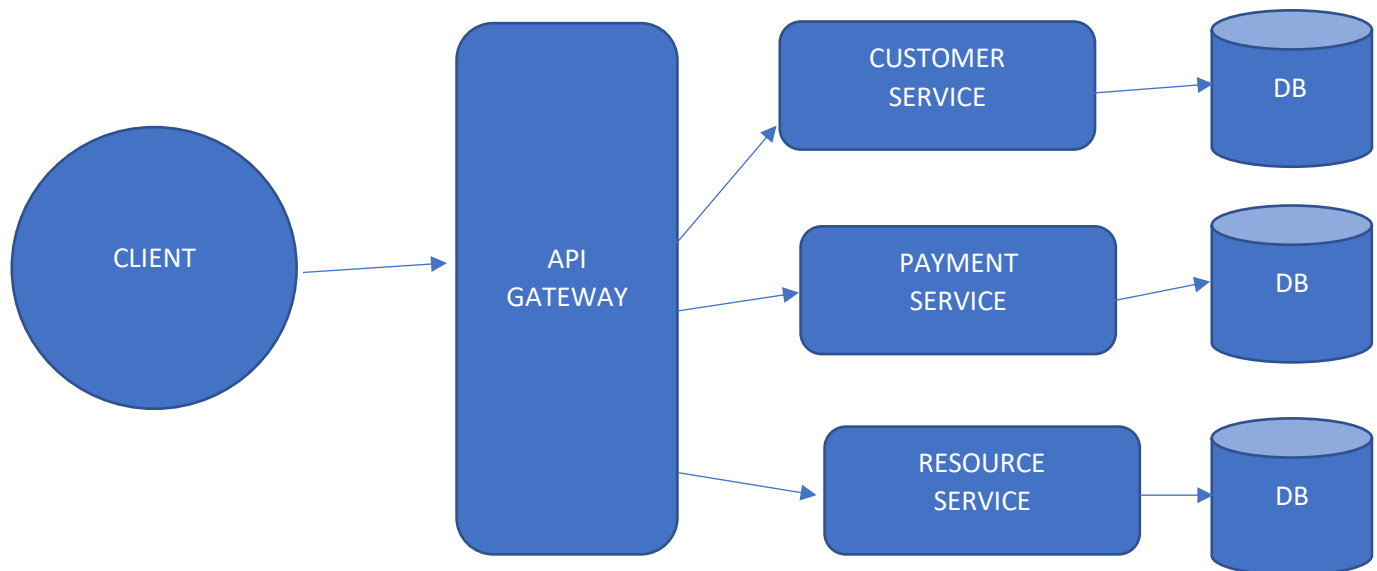
Distributed System And Big Data

. 2021/2022

Nel seguente documento viene preso in esame un progetto che ha come obiettivo quello di realizzare un sistema di gestione di contenuti multimediali. Le tecnologie utilizzate per realizzare il tutto sono state:

- **MYSQL** per realizzare dei databases necessari alla persistenza dei dati, in particolare ogni micro-servizio va ad utilizzare un database separato;
- **Spring Boot** per la realizzazione dei micro-servizi e dell'API-Gateway;
- **Prometheus** e **Grafana** sono stati utilizzati per la raccolta e la corretta visualizzazione delle metriche;
- **Docker** e **Kubernetes** sono stati utilizzati per la distribuzione.

Al progetto è stato assegnato il nome *YouMedia*. L'architettura è descritta a seguire.



Micro-servizi

I micro-servizi che implementano la business logic realizzati sono 3. All'interno dei moduli troviamo tre tipologie di classi: Entity, Controller e Services. Le classi che fanno parte di Entity sono state realizzate per definire gli oggetti che vengono mappati sui database. La classe services contiene dei metodi che permettono di disaccoppiare le istruzioni di interazione coi databases dalla diretta gestione della chiamata REST. Tali istruzioni vengono poi richiamate all'interno dei controller. Per ognuno dei quali sono state implementate diverse API basate su HTTP/REST.

- ***Customer:*** permette la corretta gestione degli utenti. Offre l'opportunità di registrare un utente, recuperare informazioni su di esso o su tutti gli utenti registrati alla piattaforma e ne permette l'eliminazione.
- ***MediaResource:*** permette la corretta gestione delle risorse multimediali. Offre l'opportunità di caricare una risorsa sul database, recuperare informazioni su di essa o su tutte le risorse presenti sulla piattaforma e ne permette l'eliminazione.
- ***Payment:*** permette la corretta gestione degli abbonamenti degli utenti. Offre l'opportunità di sottoscrivere un abbonamento recuperare informazioni su di esso o su tutti gli abbonamenti alla piattaforma e ne permette l'eliminazione.

API-Gateway

Permette l'interfacciamento tra i client e i micro-servizi, comportandosi come un proxy per accettare tutte le chiamate API, indirizzarle verso il servizio richiesto e restituire la risposta corretta. È stato realizzato come applicativo Spring Boot, andando a definire le regole di routing all'interno del file *GatewayRoutes*. Il servizio è esposto alla porta 58081 ed è stato scelto come target del server Prometheus.

Prometheus

Come sistema di monitoring è stato scelto il monitoraggio White-box basato su Prometheus, dato che esso permette di monitorare parti interne del sistema come log ed endpoint. In particolare, sono state analizzate per un determinato lasso di tempo le richieste HTTP ricevute dall'API-Gateway ed inoltrate ai diversi micro-servizi. Come si evince dai grafici a seguire, sono stati analizzati sia i metodi delle richieste (GET, POST con i relativi http status 200,404,500) sia l'utilizzo della cpu di sistema. Le metriche sono state analizzate grazie alla dashboard creata su Grafana (localhost:3000).

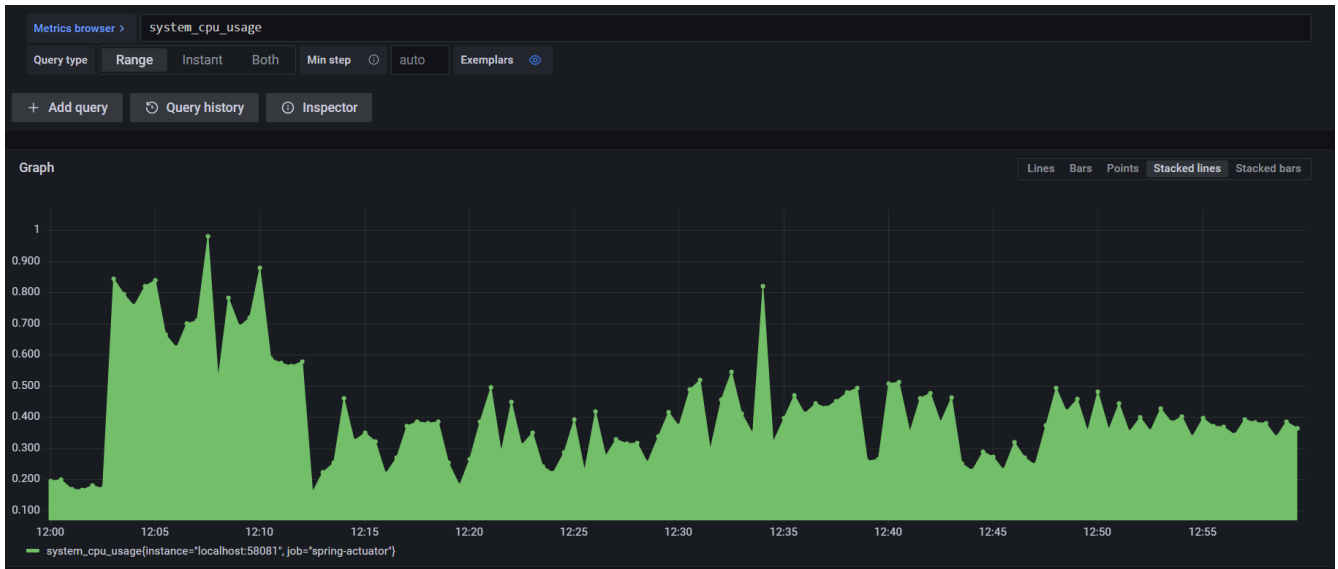


Figure 1-utilizzo CPU

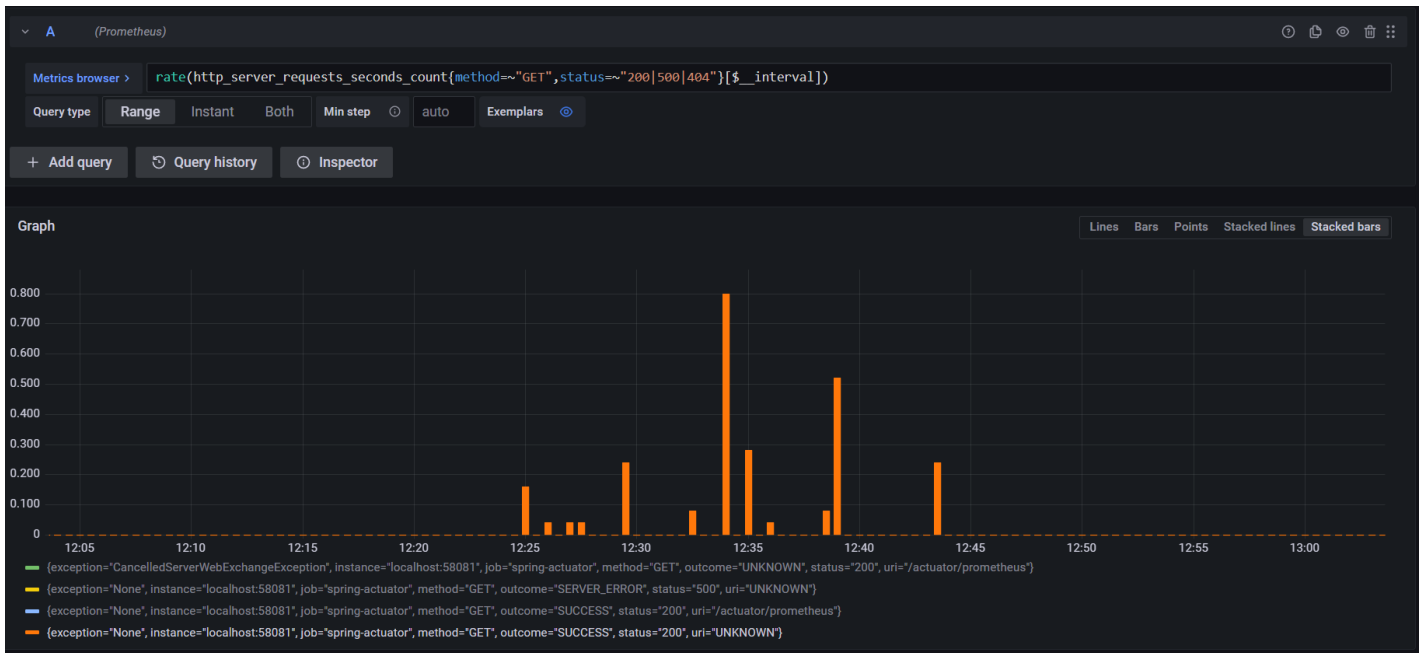


Figure 2-richieste GET con status 200

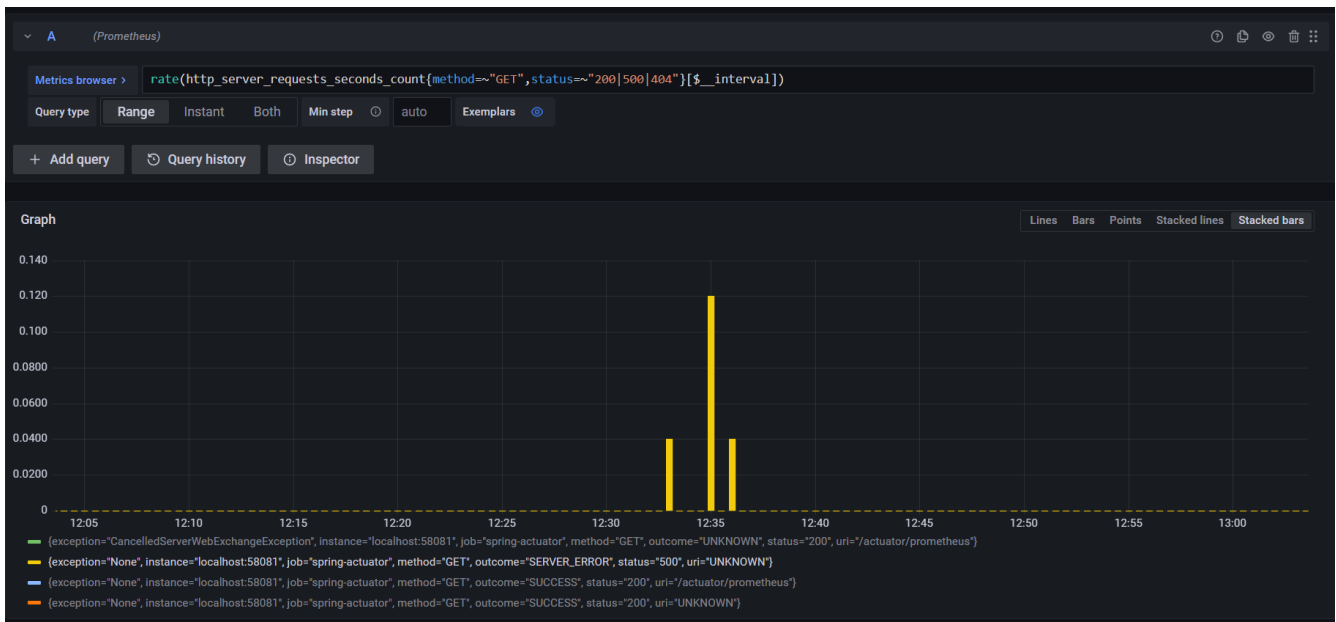


Figure 3-richieste GET con status 500



Figure 4-richieste POST con status 200