



Università di Catania

DISTRIBUTED SYSTEMS AND BIG DATA A.A. 2024/2025

HOMEWORK 3

Studenti:
Fabio Gallone 1000001752
Claudio D'Errico 1000004015

Descrizione del sistema e White-box Monitoring

Rispetto l'homework precedente, sono state implementate non solo nuove funzionalità (come ad esempio il meccanismo di monitoraggio tramite white-box monitoring), ma è stata convertita l'intera applicazione distribuita in contenitori Docker orchestrati tramite Kubernetes.

Kubernetes

Kubernetes è un sistema open-source per l'orchestrazione di container, progettate per automatizzare il deployment, lo scaling e la gestione di applicazioni containerizzate. Gestisce il bilanciamento del carico, consentendo di gestire un cluster di nodi in modo efficiente.

Minikube

Permette di eseguire un cluster Kubernetes in locale, su una singola macchina. Sfrutta container Docker per eseguire i componenti del cluster kubernetes. Riduce il consumo di risorse e accelera l'avvio del cluster.

Oggetti Kubernetes Implementati

Per orchestrare e monitorare l'applicazione, abbiamo definito diversi oggetti kubernetes:

1. **Deployments e Services** per ciascun microservizio (nel nostro caso, l'alert_notifier_system, il data collector e il server).
 - Esportano le porte per comunicare con altri servizi e per esporre metriche Prometheus. Inoltre, gestiscono il lifecycle delle applicazioni.

Un "deployment" assicura che il numero specificato di repliche dell'applicazione sia in esecuzione.

Un "service" espone un gruppo di Pod con un endpoint di rete stabile (IPS/DNS)

2. **ConfigMaps** per:
 - configurare l>alertmanager (alertmanager-config) con le impostazioni per l'invio email.
 - Configurare prometheus (prometheus-config, prometheus-alert-rules) con la definizione degli scrape jobs (estrazione automatica dei dati) e delle regole di alertings.

Prometheus e monitoraggio

Prometheus è un sistema open-source per il monitoraggio e l>alerting. Raccoglie e memorizza metriche in una serie temporale, estraendole direttamente dagli endpoint http esposti delle applicazioni (white-box monitoring) tramite il rispetto service. Permette di implementare nuovi target di monitoraggio e facilita l'integrazione con sistemi di alerting come AlertManager per inviare notifiche all'amministratore in caso vengano rilevate anomalie nel sistema.

Diagramma architetturali

Rispetto lo scorso homework, sono stati implementati i nuovi seguenti microservizi:

- Prometheus: sistema di monitoraggio che raccoglie, archivia e analizza metriche da applicazioni basate su microservizi.
- AlertManager. Componente di prometheus che gestisce, raggruppa e instrada gli alerts verso canali di notifica come email o chat basato su regole di configurazioni scritte e gestisce in un file .rules.

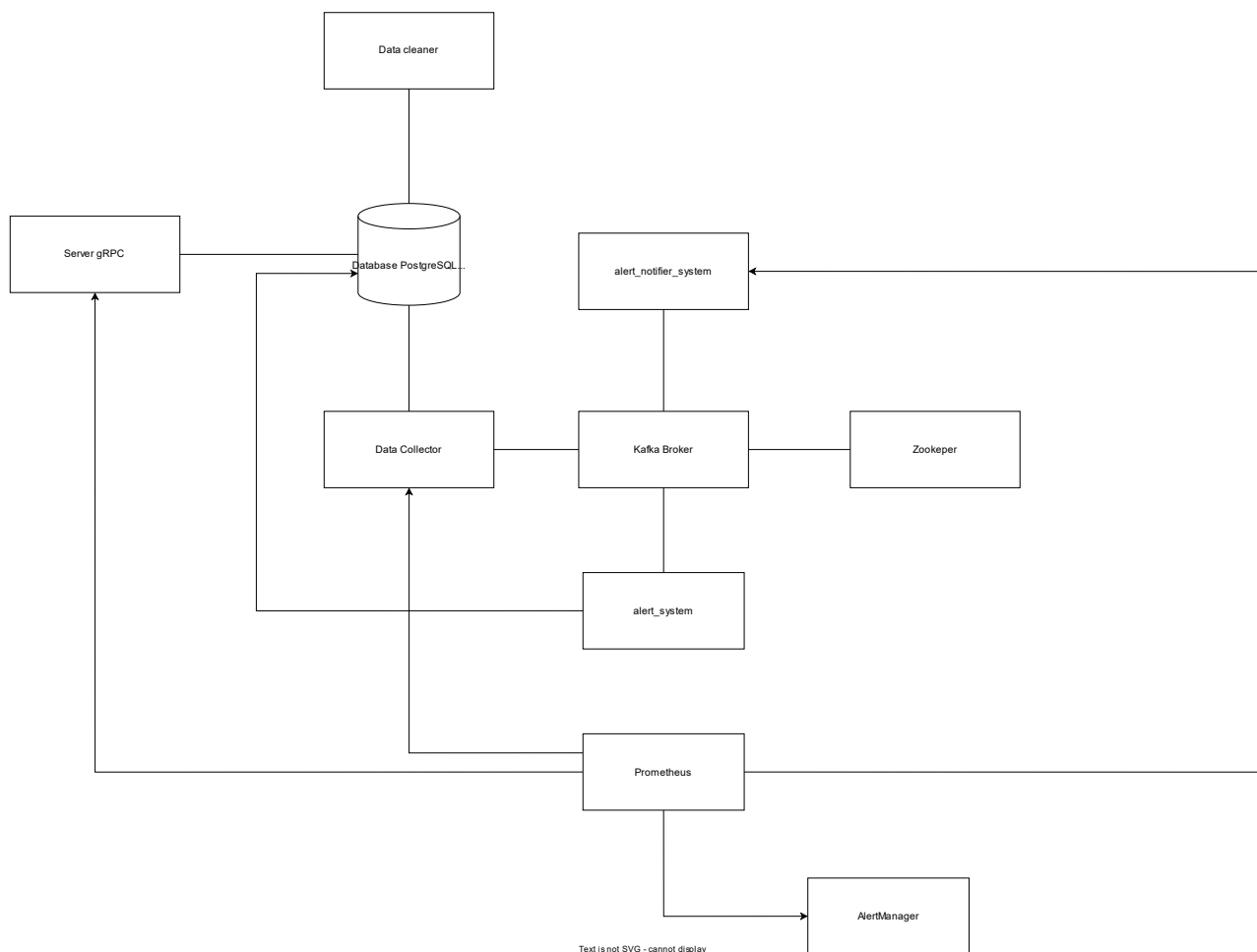


Diagramma delle Interazioni.

Rispetto l'homework precedente, sono state aggiunte nuove interazioni tra i componenti.

- **Prometheus → alert_notifier_system.**

Prometheus raccoglie le metriche da /metrics, esposto su porta 8002, per il conteggio delle email inviate. Per monitorarne lo stato.

- **Prometheus → data_collector**

Esegue lo scraping delle metriche sulla porta 8001 per rilevare il numero di chiamate verso yfinance e la durata degli aggiornamenti del database.

- **Prometheus → server**

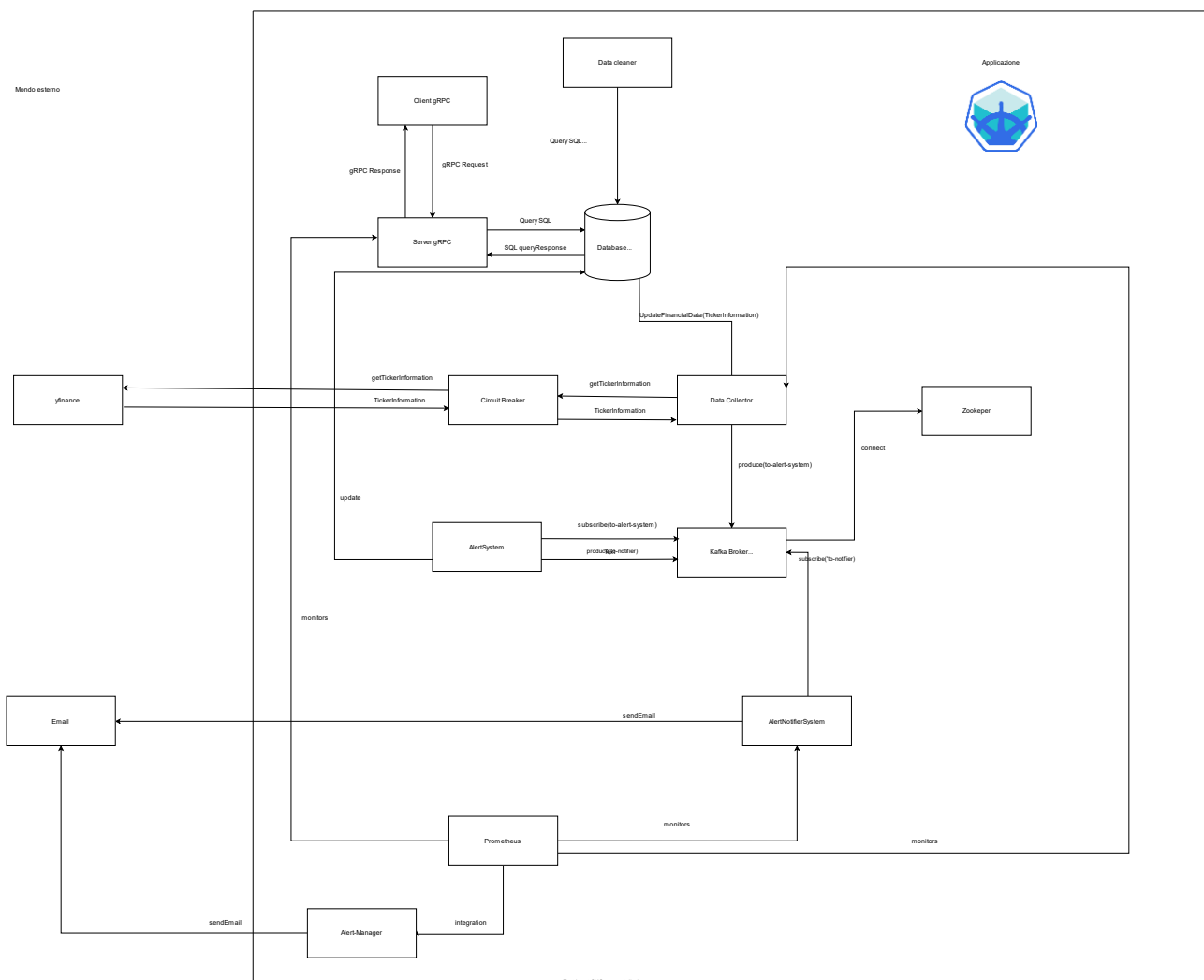
Raccoglie le metriche gRPC dal server sulla porta 8000, monitorando richieste e latenze per i servizi Command e Query.

- **Prometheus → AlertManager**

Quando una regola di alerting si attiva in base alle metriche raccolte, Prometheus invia l'alert ad alertmanager per la gestione delle notifiche.

- **Alertmanager → Email (SMTP)**

Alertmanager processa l'alert ricevuto e, basandosi sulla configurazione, invia una notifica email tramite SMTP all'amministratore di sistema.



Monitoraggio

Alert per Data Collector

- **ExcessiveYFinanceCalls (Counter):** Attiva un alert se il data collector fa più di 2 chiamate/minuto verso yfinance negli ultimi 5 minuti.
- **DatabaseUpdateSlow (Gauge):** Attiva un alert se un aggiornamento del database supera 5 secondi.

Alert per Alert Notifier System

- **HighNumberOfEmails (Counter):** Attiva un alert se in un intervallo di 4 minuti è stato inviata almeno 1 email.

Alert per Server (gRPC Server)

- **HighGrpcCommandLatency (Gauge):** Attiva un alert se la latenza delle richieste gRPC di tipo Command supera una soglia definita.
- **HighGrpcQueryLatency (Gauge):** Attiva un alert se la latenza delle richieste gRPC di tipo Query supera una soglia definita.
- **HighGrpcCommandRequests (Counter):** Attiva un alert se il numero di richieste gRPC di tipo Command supera un valore atteso in un determinato intervallo di tempo.
- **HighGrpcQueryRequests (Counter):** Attiva un alert se il numero di richieste gRPC di tipo Query supera un valore atteso in un determinato intervallo di tempo.