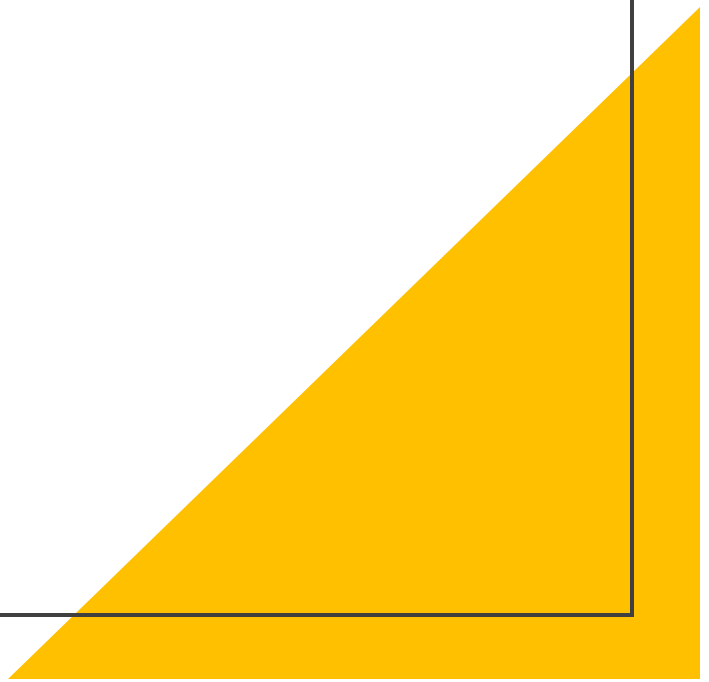
Java logging with Loki and Grafana

Marin Kalapać

TRILIX

Agenda

- Short intro to Grafana and Prometheus
- Loki – what it is and how to use it
- Demo time



Grafana

- Open-source platform dedicated to monitoring and observability
- Designed for visualizing and exploring dana, offers rich visualization tools
- Integration for different dana sources
- Offers powerful alerting tool
- Big community for plugins and extensions

Prometheus

- open-source monitoring and alerting toolkit for metrics
- collects and stores its metrics as time series data
- information is stored with the timestamp at which it was recorded, alongside optional key-value pairs called labels.
- time series collection happens via a pull model over HTTP
- provides a functional query language called PromQL (Prometheus Query Language) that lets the user select and aggregate time series data in real time

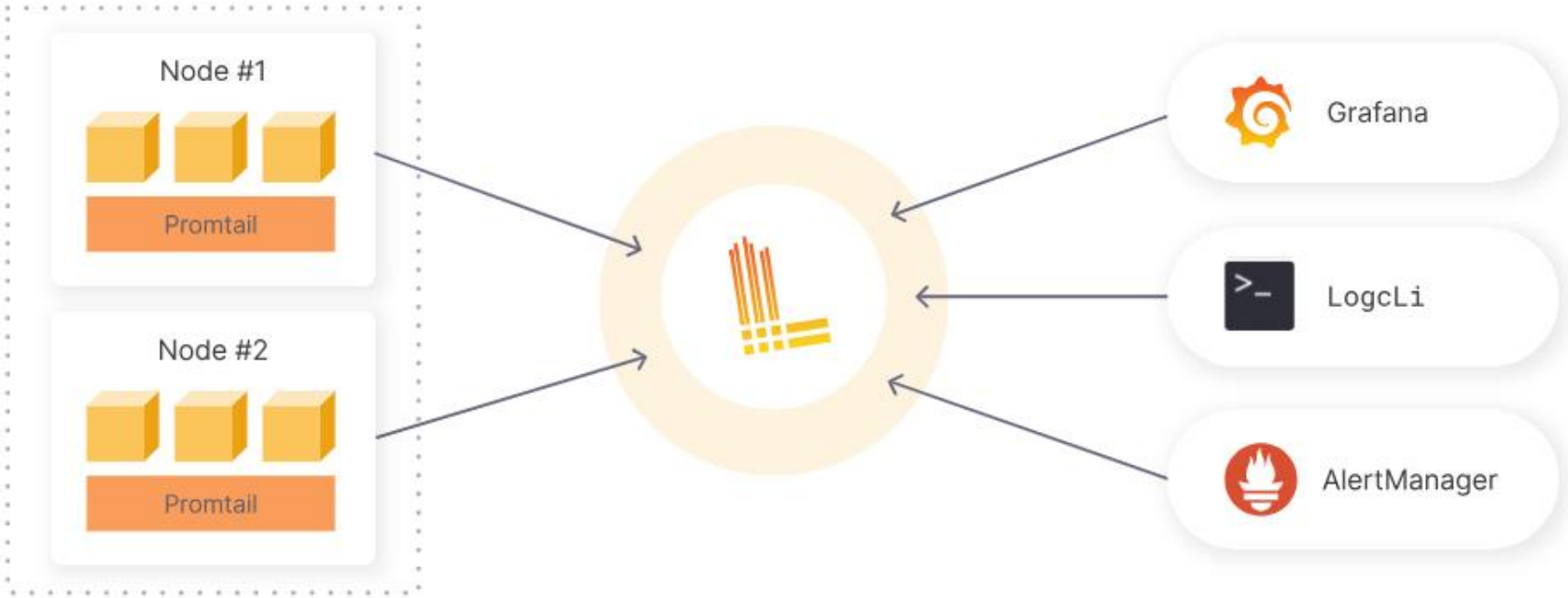
Loki – like Prometheus but for logs!

- log aggregation system designed to store and query logs from all your applications and infrastructure.
- horizontally scalable, highly available, multi-tenant
- It does not index the contents of the logs, but rather a set of labels for each log stream.

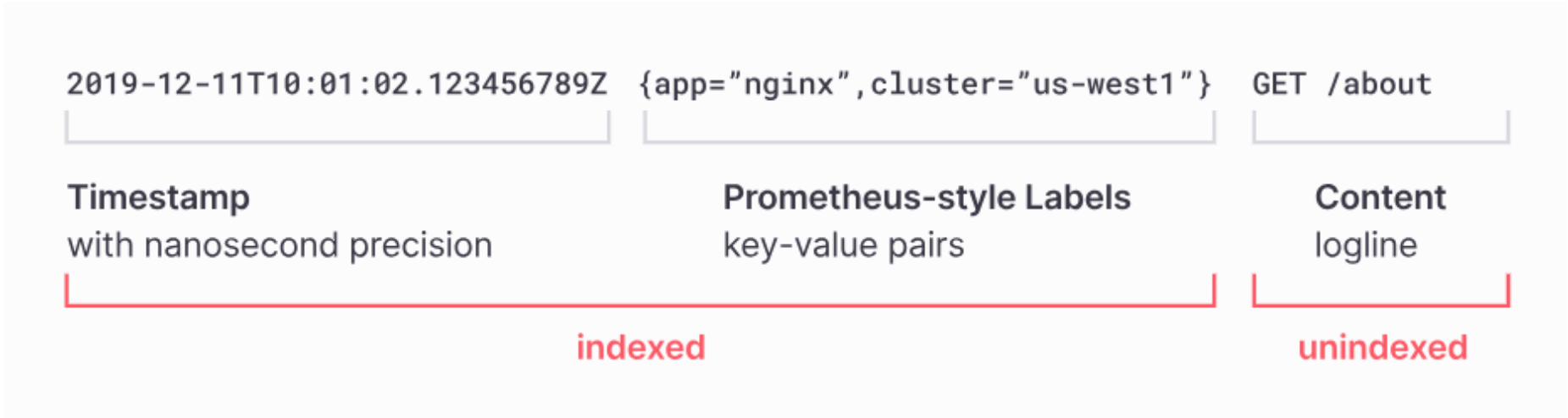
Loki

- It is Open source https://github.com/grafana/loki
- Started in Grafana Labs in 2018.
- AGPL-3.0 license
- Actively maintained, more then 40 releases, and over 860 contributors
- Written mostly using Go

How does it work



Element of log



Components of Loki

- Ingester
  - Ingesting log data, compressing and storing
- Querier
  - Handling user queries, fetching data
- Distributor
  - Accepting incoming logs, organizing them into batches and sending to ingester
- Storage backend
  - Persisting logs permanently

Installation and running

- Single binary
  - Quick and easy setup
  - Smaller systems
  - Testing
  - Dev environment
- Microservices
  - For complex and high volume systems
  - Supports horizontal scalability
  - Replication
  - Separate Read/Write paths
- Using Grafana cloud Logs

Querying - LogQL

- designed for querying logs within Loki – inspired by PromQL
- Two types
  - Log queries – contents of log lines
  - Metric queries – calculate values based on query results
- Log queries consists of
  - Log stream selector
  - Log pipeline / filter expression

Querying - LogQL

- Some examples:
  - Log query:
    - ***{app="my-app"} |= "error„ != „timeout”***
  - Using Regex:
    - ***{app="my-app"} |~ "GET /api/[0-9]+"***
  - Extracting Metrics:
    - ***rate({app="my-app"} |= "error" [5m])***
  - Aggregations:
    - ***sum by (status_code) (count_over_time({app="web-server"}[5m]))"***

Demo time

We have a Spring Boot application with fictive simulation of household energy consumption (sort of)

We are using already Grafana and Prometheus for observing the data inside application

We are going to use Loki to gather more useful information from this application

Documentation

- Loki - https://grafana.com/docs/loki/latest/
- Grafana - https://grafana.com/docs/grafana/latest/

Question?

- Thank you!
- Contact: marin.kalapac@trilix.eu