

Applied Observability

Prognozy i trendy rozwojowe w informatyce

Mateusz Bączek

Politechnika Wrocławska

2023



Politechnika Wrocławska

Definicja problemu

1. Tworzymy produkt,
2. uruchamiamy produkt (na sprzęcie dedykowanym/w chmurze/na sprzęcie klienta),
3. **czy produkt działa?**



Uszczegółowienie problemu

Co to znaczy, że nasz produkt **działa**?

1. Czy produkt działa **w tej chwili**?
2. Czy wystąpiły awarie **w przeszłości**?
3. Czy sprawność działania produktu jest **stała w czasie**?
4. Jakie metryki oceniają **sprawność działania produktu**?
5. Czy wdrożono procedury, które należy podjąć w przypadku **wystąpienia awarii**?

A gdy już zdaży się awaria?

- ▶ Powiadomienie o wystąpieniu awarii,
- ▶ automatyczne restartowanie usług,
- ▶ automatyczne przywracanie backupów,
- ▶ automatyczne cofanie aktualizacji (*rollback*),
- ▶ **analiza przyczyn wystąpienia awarii.**

Co monitorujemy? [Perri, 2022]

1. Infrastrukturę:

- ▶ serwery,
- ▶ chmurę,
- ▶ sieć.

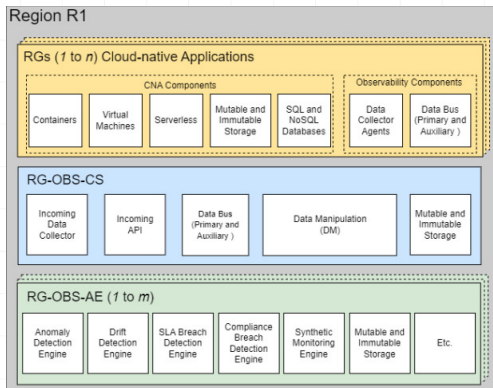
2. Aplikacje:

- ▶ interfejsy użytkownika.
- ▶ bazy danych,
- ▶ interfejsy API.

3. Funkcje biznesowe:

- ▶ zdarzenia domenowe w systemie.

Observability jako standardowy element infrastruktury projektowej



Rysunek: Rekomendowana struktura aplikacji chmurowej opracowana przez specjalistów z uniwersytetów w Toronto oraz Maryland, we współpracy z IBM [Pourmajidi, 2023].

Jakie są źródła danych dla systemu monitorującego?

1. Logi generowane przez usługi,
2. metryki pobierane z usług,
3. zdarzenia (*events*) emitowane przez system.

Od monitoringu do *applied observability*

By 2021, 60% of IT monitoring investments will include a focus on business-relevant metrics, up from less than 20% in 2017. [Shetty, 2017]

- ▶ **zdecentralizowana** infrastruktura,
- ▶ coraz większy **koszt** *downtime'u*,
- ▶ użytkownicy przyzwyczajeni do **ciągłej** dostępności usługi,
- ▶ większa moc urządzeń mobilnych - możliwość **monitorowania urządzeń klienckich**,
- ▶ wykorzystywanie sztucznej inteligencji do **wykrywania anomalii**,
- ▶ metryki jako obiektywne źródło danych przy **podejmowaniu decyzji** technicznych/biznesowych.

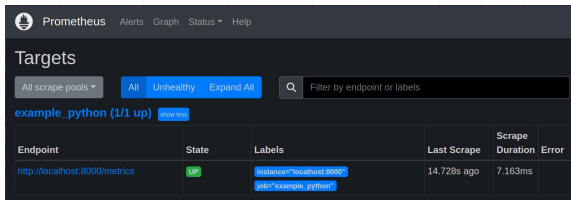
Bardzo niski koszt wprowadzenia *observability*

```
from prometheus_client import start_http_server, Summary

app = Flask(__name__)
REQUEST_TIME = Summary(
    "request_processing_seconds",
    "Time spent processing request"
)

@app.route("/endpoint")
@REQUEST_TIME.time()
def example_endpoint():
    print("Processing a request..")
    time.sleep(random.randrange(1, 10))
    print("Processing finished!")
    return "Hello, world!"
```

Po konfiguracji w systemie *Prometheus*



The screenshot shows the Prometheus web interface. At the top, there's a navigation bar with 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. Below this, the 'Targets' section is active. It includes a dropdown for 'All scrape pools', buttons for 'All', 'Unhealthy', and 'Expand All', and a search bar labeled 'Filter by endpoint or labels'. A link for 'example_python (1/1 up)' is visible. Below this is a table with the following data:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:8000/metrics	UP	instance="localhost:8000" job="example_python"	14.728s ago	7.163ms	

Rysunek: Status usługi w systemie *Prometheus*.



Rysunek: Wykres metryki `increase(process_cpu_seconds_total)`

Popularne rozwiązania do monitorowania infrastruktury IT

Your **observability** stack

Operational dashboards for your data here, there, or anywhere

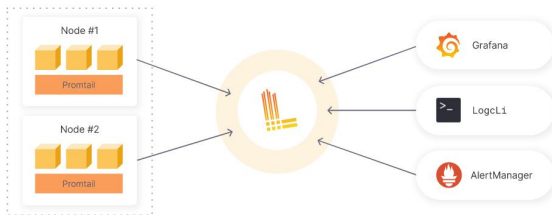
✨🛒☁️✨
The (actually useful) free forever plan
Grafana, of course + 10K series
Prometheus metrics + 50GB logs +
50GB traces
[Create free account](#)
(No credit card required)



Rysunek: Grafana – wizualizacja danych z wielu źródeł, system alertowania.

Popularne rozwiązania do monitorowania infrastruktury IT

How does Grafana Loki work?



Pull in any logs with Promtail

Promtail is a logs collector built specifically for Loki. It uses the same service discovery as Prometheus and includes analogous features for labeling, transforming, and filtering logs before ingestion into Loki.

Store the logs in Loki

Loki does not index the text of logs. Instead, entries are grouped into streams and indexed with labels. Not only does this reduce costs, it also means log lines are available to query within milliseconds of being received by Loki.

Use LogQL to explore

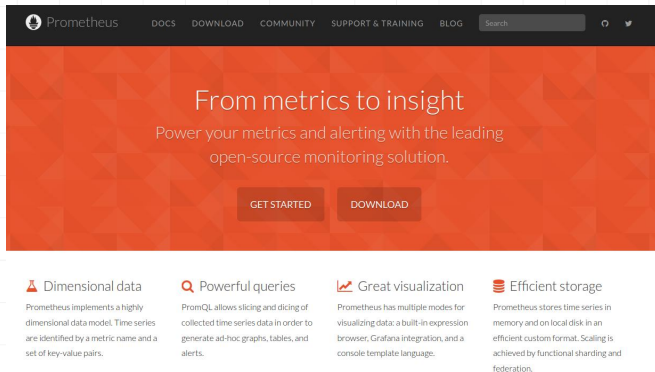
Use Loki's powerful query language, LogQL, to explore your logs. Run LogQL queries directly within Grafana to visualize your logs alongside other data sources, or with LogCLI, for those who prefer a command line experience.

Alert on your logs

Set up alerting rules for Loki to evaluate on your incoming log data. Configure Loki to send the resulting alerts to a Prometheus Alertmanager so they can then get routed to the right team.

Rysunek: Loki – centralne repozytorium logów.

Popularne rozwiązania do monitorowania infrastruktury IT



The screenshot shows the Prometheus website homepage. The header is dark with the Prometheus logo and navigation links: DOCS, DOWNLOAD, COMMUNITY, SUPPORT & TRAINING, and BLOG. A search bar is on the right. The main content area has an orange background with the text "From metrics to insight" and "Power your metrics and alerting with the leading open-source monitoring solution." Below this are two buttons: "GET STARTED" and "DOWNLOAD". At the bottom, there are four feature sections, each with an icon and a title: "Dimensional data" (triangle icon), "Powerful queries" (magnifying glass icon), "Great visualization" (line graph icon), and "Efficient storage" (stack of disks icon). Each section has a brief description of the feature.

Dimensional data
Prometheus implements a highly dimensional data model. Time series are identified by a metric name and a set of key-value pairs.

Powerful queries
PromQL allows slicing and dicing of collected time series data in order to generate ad-hoc graphs, tables, and alerts.

Great visualization
Prometheus has multiple modes for visualizing data: a built-in expression browser, Grafana integration, and a console template language.

Efficient storage
Prometheus stores time series in memory and on local disk in an efficient custom format. Scaling is achieved by functional sharding and federation.

Rysunek: Prometheus – narzędzie do zbierania i monitorowania metryk systemów informatycznych.

Popularne rozwiązania do monitorowania infrastruktury IT



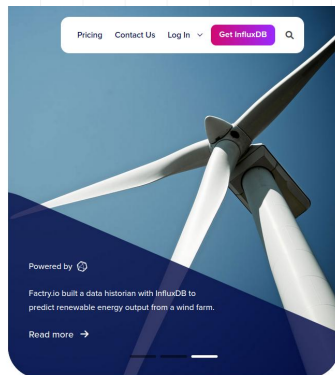
Products ▾ Developers ▾ Customers ▾ Resources ▾

It's About Time. Build on InfluxDB.

The Time Series Data Platform where developers build IoT, analytics, and cloud applications.

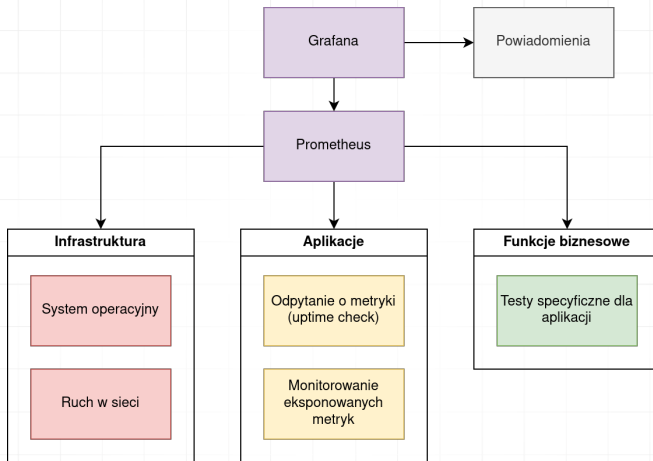
Get InfluxDB

Find the right product →



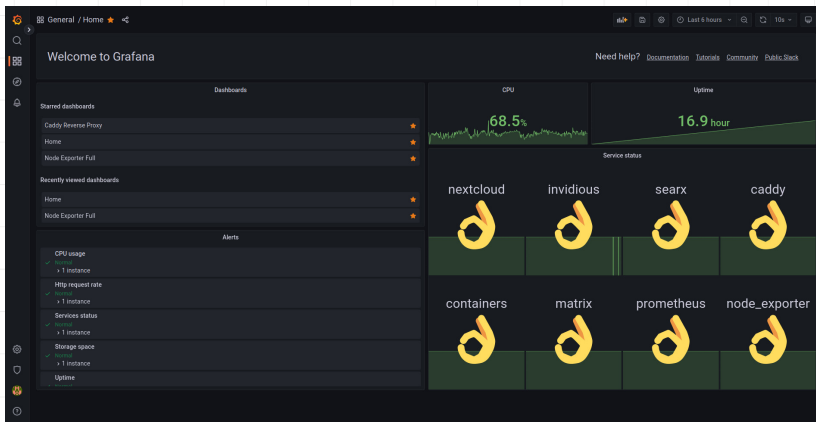
Rysunek: InfluxDB – baza danych dedykowana do przechowywania danych pomiarowych.

Przykład – monitorowanie własnej infrastruktury



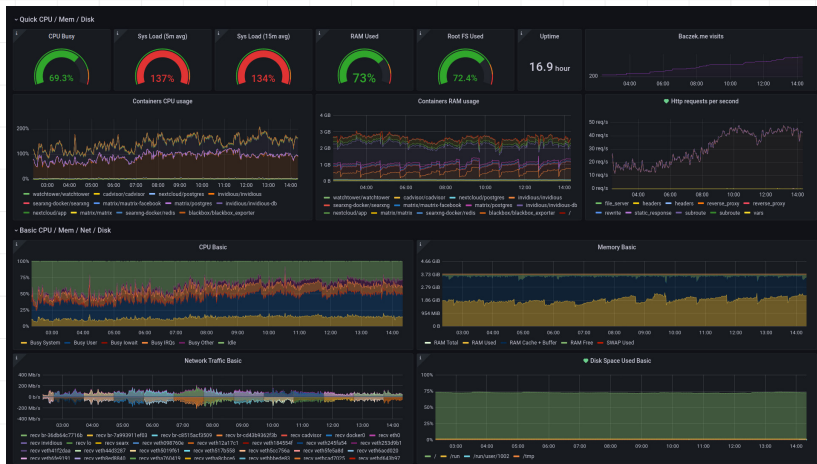
Rysunek: Monitoring infrastruktury na serwerze prywatnym.

Przykład – monitorowanie własnej infrastruktury



Rysunek: Główny panel Grafany – ogólny podgląd stanu infrastruktury.

Przykład – monitorowanie własnej infrastruktury



Rysunek: Prometheus Node Exporter - podgląd metryk systemu operacyjnego.

Przykład – monitorowanie własnej infrastruktury



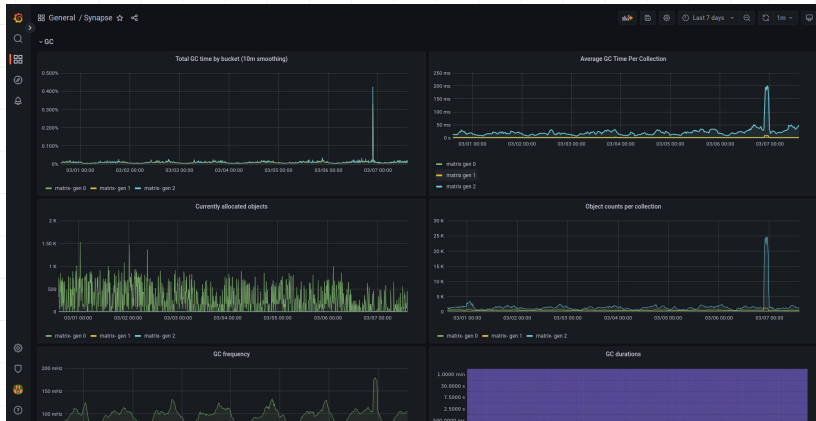
Rysunek: Metryki eksponowane przez serwer webowy Caddy (odpowiednik Apache/NGINX).

Przykład – monitorowanie własnej infrastruktury



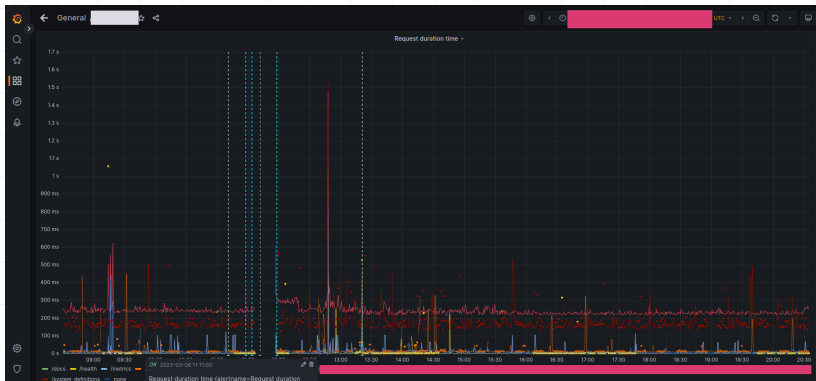
Rysunek: Przykładowy incydent w infrastrukturze - chwilowe przedłużenie czasu przetwarzania zapytań. Przykład korelacji metryk.

Przykład – monitorowanie własnej infrastruktury



Rysunek: Metryki dotyczące *garbage collector* w aplikacji Synapse – serwer protokołu *Matrix* napisany w języku *Python*.

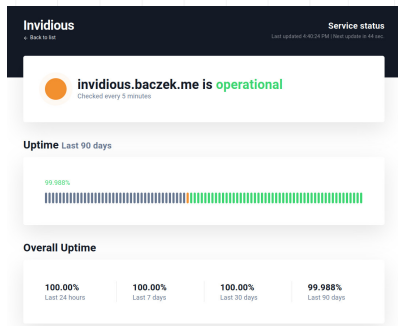
Monitorowanie już na poważnie



Rysunek: Zapis występowania alertów na panelu monitorującym czas przetwarzania zapytania w API.

Applied Observability – podsumowanie

1. **Niski koszt** wprowadzenia,
2. szeroki wybór *open-source*'owych narzędzi,
3. **duży wzrost zrozumienia** budowanego/hostowanego produktu,
4. powiadomienia o awariach,
5. automatyczne reagowanie na incydenty.
6. dane pozwalające na **wykonywanie decyzji** technicznych i biznesowych.



Rysunek: Statystyki dostępności usługi *Invidious*, hostowanej i monitorowanej w ramach własnej infrastruktury.



Pourmajidi, William and Zhang, Lei and Steinbacher, John and Erwin, Tony and Miransky, Andriy (2023)

A Reference Architecture for Observability and Compliance of Cloud Native Applications

Toronto Metropolitan University, University of Maryland, Baltimore County, USA, IBM Canada Lab, Toronto, Canada, IBM Cloud Platform, Austin, USA

<https://www.gartner.com/smarterwithgartner/how-to-start-an-it-monitoring-initiative>



Lori perri (2022)

Monetizing Observable Data Will Separate the Winners and Losers

Gartner Insights

<https://gartner.com/en/articles/monetizing-observable-data-will-separate-the-winners-and>



Sony Shetty (2017)

How to Start an IT Monitoring Initiative

Gartner Insights

[https://www.gartner.com/smarterwithgartner/
how-to-start-an-it-monitoring-initiative](https://www.gartner.com/smarterwithgartner/how-to-start-an-it-monitoring-initiative)