ДОДАТОК С

Керівництво користувача

ЗАТВЕРДЖЕНО   
44165850. 01516-01 I3 01-ЛЗ

ДОДАТОК ВЛАСНИЙ ОРГАНАЙЗЕР

Керівництво користувача

44165850.01516 -01 I3 01

Листів 20

2025

АНОТАЦІЯ

Документ 121.01.01-01 І3 01 "Система моніторингу та аналізу телеметрії в Kubernetes. Керівництво користувача" входить до складу програмної документації на систему, яка розроблена з метою забезпечення повного циклу збору, зберігання, аналізу та візуалізації телеметричних даних у середовищі Kubernetes.

У даному документі представлено область призначення і умови застосування, підготовка до роботи, опис операцій, аварійні ситуації, рекомендації по застосуванню. Система реалізована з використанням Prometheus для збору метрик, Elasticsearch для зберігання логів, Fluent Bit для збору логів та Grafana для візуалізації даних.

Об'єм пам'яті, який займають компоненти системи, складає приблизно 8-12 ГБ залежно від конфігурації. Конфігурація кластера повинна відповідати мінімальним вимогам для Kubernetes. Система функціонує у будь-якому середовищі Kubernetes (локальному, хмарному), а розгортання здійснюється за допомогою Helm charts та Kubernetes операторів.

ЗМІСТ

ДОДАТОК В 1

ВСТУП 4

1 ПРИЗНАЧЕННЯ І УМОВИ ЗАСТОСУВАННЯ 5

1.1 Функціональне і експлуатаційне призначення 5

1.2 Умови застосування 5

2 ПІДГОТОВКА ДО РОБОТИ 6

2.1 Склад і зміст дистрибутивного носія даних 6

2.2 Порядок завантаження даних і програми 6

2.3 Порядок перевірки працездатності 7

3 ВИКОНАННЯ ПРОГРАМИ 8

3.1 Робота з системою моніторингу метрик 8

3.1.1 Доступ до Prometheus 8

3.1.2 Перегляд та аналіз метрик 9

3.1.3 Створення власних запитів 10

3.2 Робота з системою аналізу логів 11

3.2.1 Доступ до Kibana 11

3.2.2 Пошук та фільтрація логів 11

3.2.3 Створення візуалізацій логів 12

3.3 Робота з дашбордами Grafana 13

3.3.1 Доступ до Grafana 13

3.3.2 Перегляд існуючих дашбордів 14

3.3.3 Створення власних дашбордів

3.4 Система оповіщень 15

3.4.1 Перегляд активних оповіщень 16

3.4.2 Налаштування правил оповіщень 16

3.5 Адміністрування системи 17

3.5.1 Моніторинг стану компонентів

3.5.2 Масштабування системи

4 РЕКОМЕНДАЦІЇ ПО ЗАСТОСУВАННЮ 19

5 ПОВІДОМЛЕННЯ 20

ВСТУП

Область застосування системи "Моніторинг та аналіз телеметрії в Kubernetes" — DevOps команди, системні адміністратори, розробники, SRE інженери та інші фахівці, які працюють з контейнеризованими додатками у Kubernetes.

Система призначена для автоматизації процесів моніторингу та аналізу телеметрії в Kubernetes кластерах. Основні можливості включають:

* **збір метрик продуктивності**: автоматичний збір метрик з усіх компонентів кластера та додатків;
* **централізований збір логів**: збирання логів з усіх контейнерів та системних компонентів;
* **візуалізація даних**: створення інформативних дашбордів для моніторингу стану системи;
* **система оповіщень**: автоматичне виявлення аномалій та відправка сповіщень;
* **аналіз трендів**: історичний аналіз використання ресурсів та продуктивності;
* **масштабованість**: автоматичне адаптування до змін розміру кластера.

Система спрощує діагностику проблем, підвищує надійність інфраструктури та дозволяє проактивно управляти ресурсами кластера.

1. ПРИЗНАЧЕННЯ І УМОВИ ЗАСТОСУВАННЯ
   1. Функціональне і експлуатаційне призначення

Функціональне призначення системи полягає в забезпеченні повного циклу збору, зберігання, аналізу та візуалізації телеметричних даних у середовищі Kubernetes. Система дозволяє ефективно моніторити стан інфраструктури, відстежувати продуктивність додатків, аналізувати логи та швидко виявляти проблеми.

Експлуатаційне призначення: система "Моніторинг та аналіз телеметрії в Kubernetes" дозволить організувати централізований моніторинг кластера, зекономити час на діагностиці проблем, підвищити надійність сервісів та ефективність використання ресурсів.

* 1. Умови застосування

Для забезпечення стабільного та надійного функціонування системи моніторингу, необхідно дотримуватись наступних умов експлуатації:

* **Kubernetes кластер** повинен бути розгорнутий та функціонувати стабільно;
* **користувачі** повинні мати базові навички роботи з Kubernetes та веб-інтерфейсами;
* **доступ до мережі** для взаємодії з компонентами системи через веб-браузер.

**Мінімальні технічні вимоги:**

* Kubernetes версії 1.20 або вище;
* мінімум 3 робочі вузли кластера;
* загальний обсяг ОЗУ кластера: не менше 16 ГБ;
* загальна кількість CPU cores: не менше 8;
* постійне сховище: не менше 100 ГБ для зберігання метрик та логів;
* мережеве з'єднання між вузлами кластера;
* веб-браузер (Google Chrome, Firefox, Edge тощо) для доступу до інтерфейсів.

1. ПІДГОТОВКА ДО РОБОТИ

2.1 Склад і зміст дистрибутивного носія даних

Для роботи з системою моніторингу необхідне наступне програмне забезпечення:

**Обов'язкові компоненти:**

* Kubernetes кластер версії 1.20+ (розгорнутий та налаштований);
* Helm версії 3.0+ для встановлення компонентів;
* kubectl для взаємодії з кластером;
* Prometheus Operator для управління компонентами Prometheus;
* Elasticsearch Operator (ECK) для управління Elasticsearch;

**Компоненти системи:**

* Prometheus — збір та зберігання метрик;
* Grafana — візуалізація метрик та створення дашбордів;
* Elasticsearch — зберігання та індексація логів;
* Kibana — аналіз та візуалізація логів;
* Fluent Bit — збір логів з контейнерів;
* AlertManager — управління оповіщеннями;

**Додаткові інструменти:**

* Веб-браузер для доступу до інтерфейсів;
* Git для отримання конфігураційних файлів (опціонально).

2.2 Порядок завантаження даних і програми

Для розгортання системи моніторингу необхідно:

1. **Підготувати Kubernetes кластер:**

kubectl cluster-info

kubectl get nodes

1. **Встановити Helm (якщо не встановлений):**

curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash

1. **Додати необхідні Helm репозиторії:**

helm repo add prometheus-community https://prometheus-community.github.io/helm-charts

helm repo add elastic https://helm.elastic.co

helm repo update

1. **Створити namespace для моніторингу:**

kubectl create namespace monitoring

1. **Встановити Prometheus Stack:**

helm install prometheus prometheus-community/kube-prometheus-stack \

--namespace monitoring \

--set grafana.adminPassword=admin123

1. **Встановити Elasticsearch Operator:**

kubectl create -f https://download.elastic.co/downloads/eck/2.10.0/crds.yaml

kubectl apply -f https://download.elastic.co/downloads/eck/2.10.0/operator.yaml

1. **Розгорнути Elasticsearch та Kibana:**

kubectl apply -f elasticsearch.yaml

kubectl apply -f kibana.yaml

1. **Встановити Fluent Bit:**

helm install fluent-bit fluent/fluent-bit \

--namespace monitoring \

--set-file config=fluent-bit-config.yaml

2.3 Порядок перевірки працездатності

Для перевірки успішного розгортання системи:

1. **Перевірити статус всіх подів:**

kubectl get pods -n monitoring

Всі поди повинні мати статус "Running".

1. **Перевірити доступність сервісів:**

kubectl get svc -n monitoring

1. **Налаштувати port-forwarding для доступу до інтерфейсів:**

*# Prometheus*

kubectl port-forward -n monitoring svc/prometheus-kube-prometheus-prometheus 9090:9090

*# Grafana*

kubectl port-forward -n monitoring svc/prometheus-grafana 3000:80

*# Kibana*

kubectl port-forward -n monitoring svc/kibana-kb-http 5601:5601

1. **Відкрити веб-інтерфейси в браузері:**
   * Prometheus: [http://localhost:9090](http://localhost:9090/)
   * Grafana: [http://localhost:3000](http://localhost:3000/)
   * Kibana: [http://localhost:5601](http://localhost:5601/)

Якщо всі інтерфейси доступні та відображають дані, система працездатна.

1. ВИКОНАННЯ ПРОГРАМИ

### 3.1 Робота з системою моніторингу метрик

#### 3.1.1 Доступ до Prometheus

Для доступу до Prometheus:

1. Налаштувати port-forwarding:

kubectl port-forward -n monitoring svc/prometheus-kube-prometheus-prometheus 9090:9090

1. Відкрити браузер та перейти за адресою: [http://localhost:9090](http://localhost:9090/)
2. На головній сторінці з'явиться інтерфейс Prometheus з можливістю виконання запитів (рис 3.1).

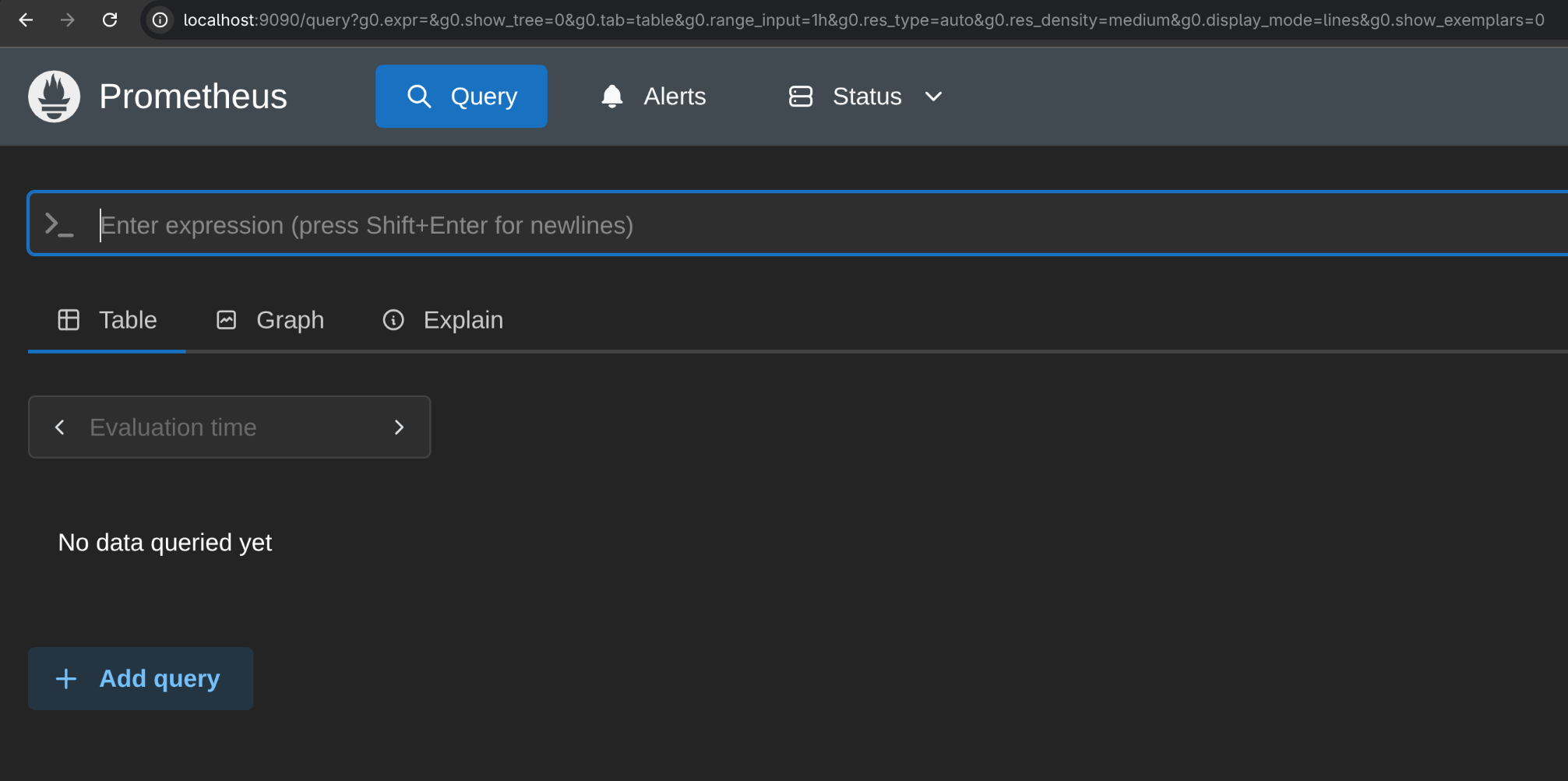


Рисунок 3.1 - Інтерфейс Prometheus з можливістю виконання запитів

#### 3.1.2 Перегляд та аналіз метрик

У Prometheus можна переглядати різні метрики:

1. **Метрики використання CPU:**

100 - (avg by (instance) (rate(node\_cpu\_seconds\_total{mode="idle"}[5m])) \* 100)

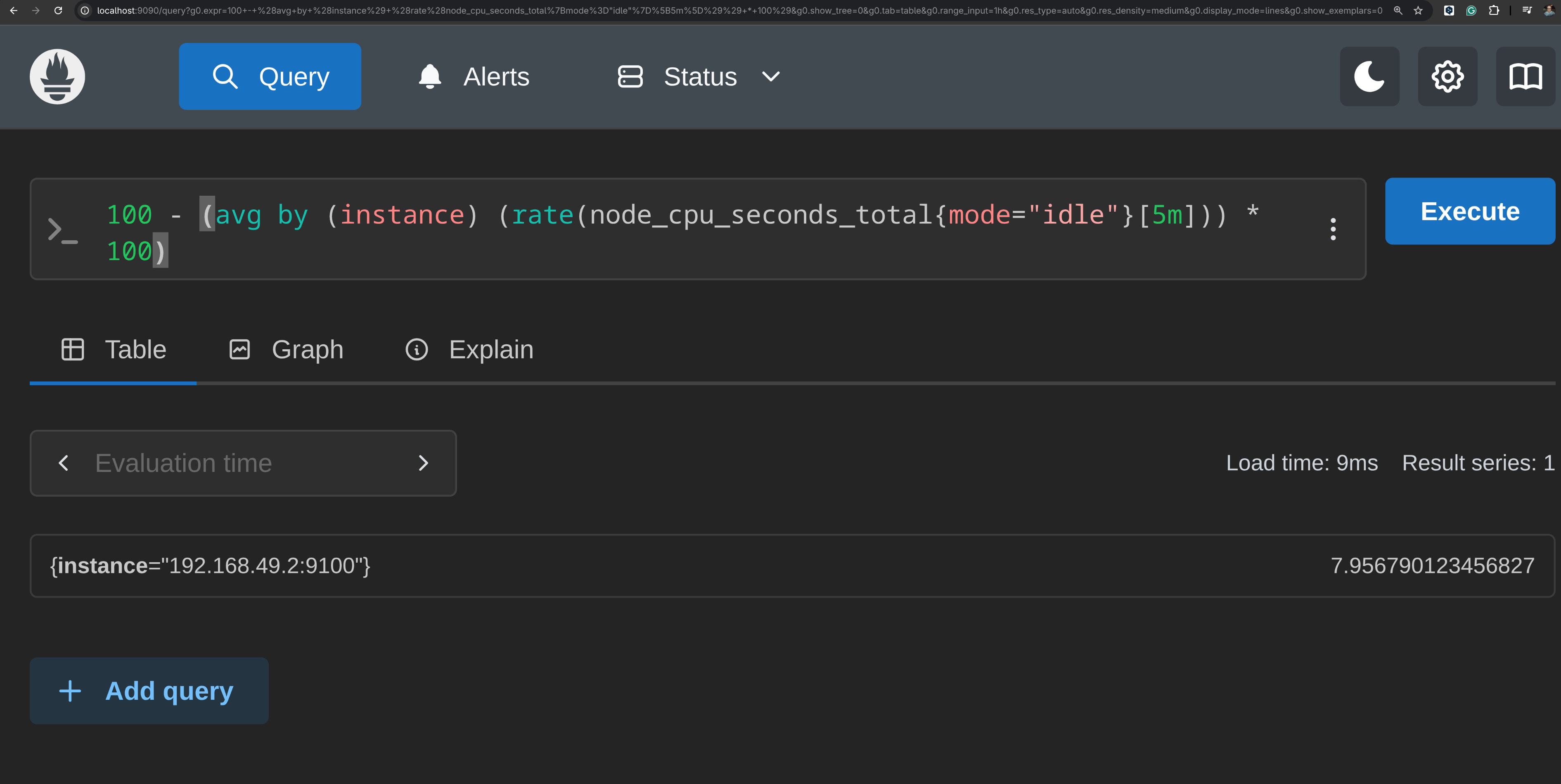
1. **Метрики використання пам'яті:**

(1 - (node\_memory\_MemAvailable\_bytes / node\_memory\_MemTotal\_bytes)) \* 100

1. **Метрики стану подів:**

kube\_pod\_status\_phase{phase="Running"}

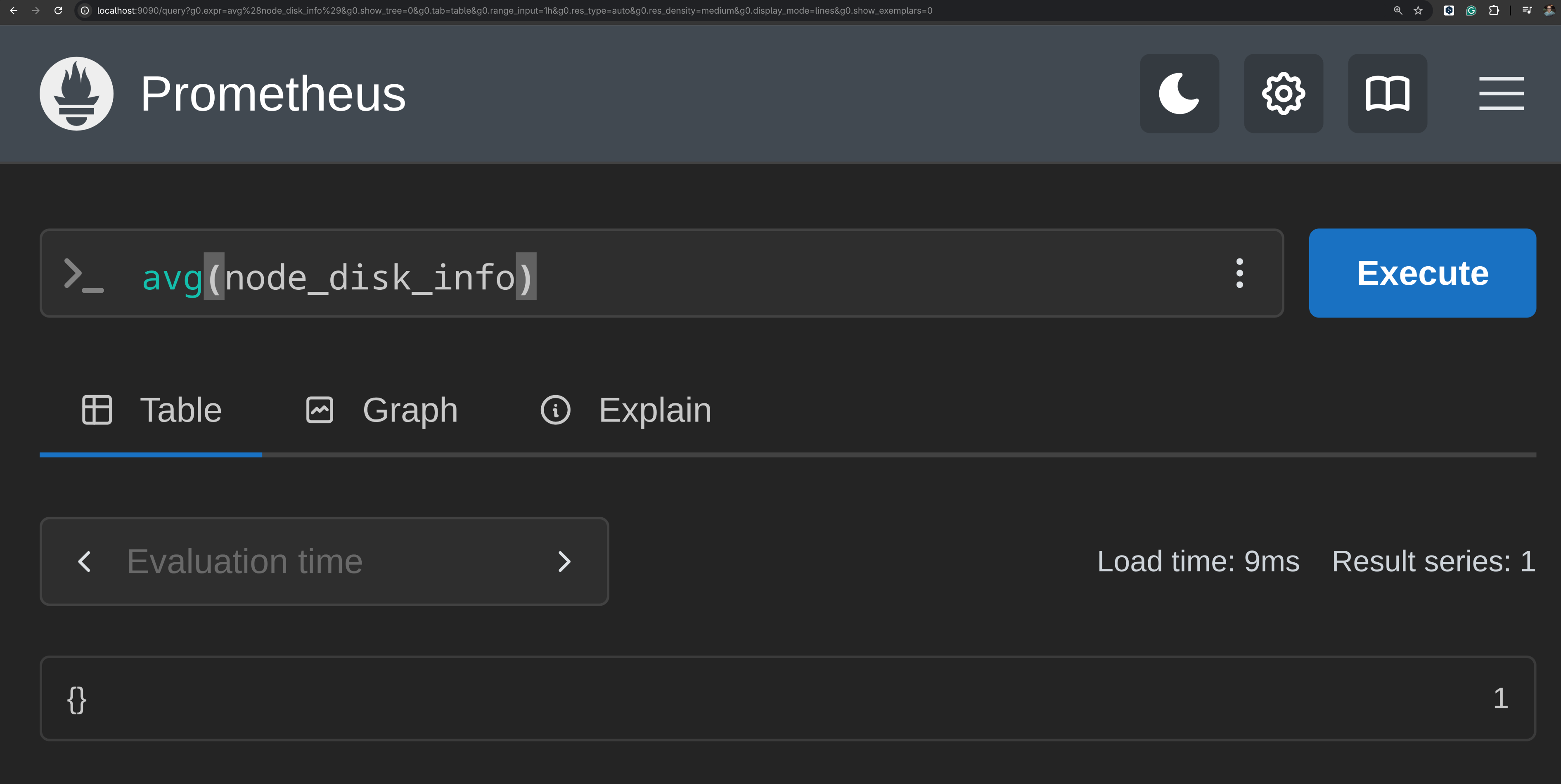
Для виконання запиту введіть його у поле "Expression" та натисніть "Execute".

Рисунок 3.2 — Результат виконання запиту

#### 3.1.3 Створення власних запитів

Для створення власних запитів використовуйте мову запитів PromQL:

1. Оберіть метрику з випадаючого списку або введіть вручну
2. Додайте фільтри за допомогою фігурних дужок: {label="value"}
3. Використовуйте функції агрегації: sum(), avg(), max(), min()
4. Застосовуйте часові функції: rate(), increase(), delta()

Рисунок 3.3 — Запит мовою PromQL

### 3.2 Робота з системою аналізу логів

#### 3.2.1 Доступ до Kibana

Для доступу до Kibana:

1. Налаштувати port-forwarding:

kubectl port-forward -n monitoring svc/kibana-kb-http 5601:5601

1. Відкрити браузер та перейти за адресою: [http://localhost:5601](http://localhost:5601/)
2. При першому входженні необхідно створити index для app-logs-\*



Рисунок 3.4 — Створення індекса в Kibana

#### 3.2.2 Пошук та фільтрація логів

У Kibana для пошуку логів:

1. Перейти до розділу "Discover"
2. Обрати часовий діапазон у правому верхньому кутку
3. Використовувати поле пошуку для фільтрації:
   * За namespace: kubernetes.namespace\_name:"default"
   * За рівнем логування: level:"error"
   * За назвою поду: kubernetes.pod\_name:"app-logs-\*"

#### 3.2.3 Створення візуалізацій логів

Для створення візуалізацій:

1. Перейти до розділу "Visualize"
2. Натиснути "Create visualization"
3. Обрати тип візуалізації (графік, таблиця, pie chart)
4. Налаштувати метрики та buckets
5. Зберегти візуалізацію

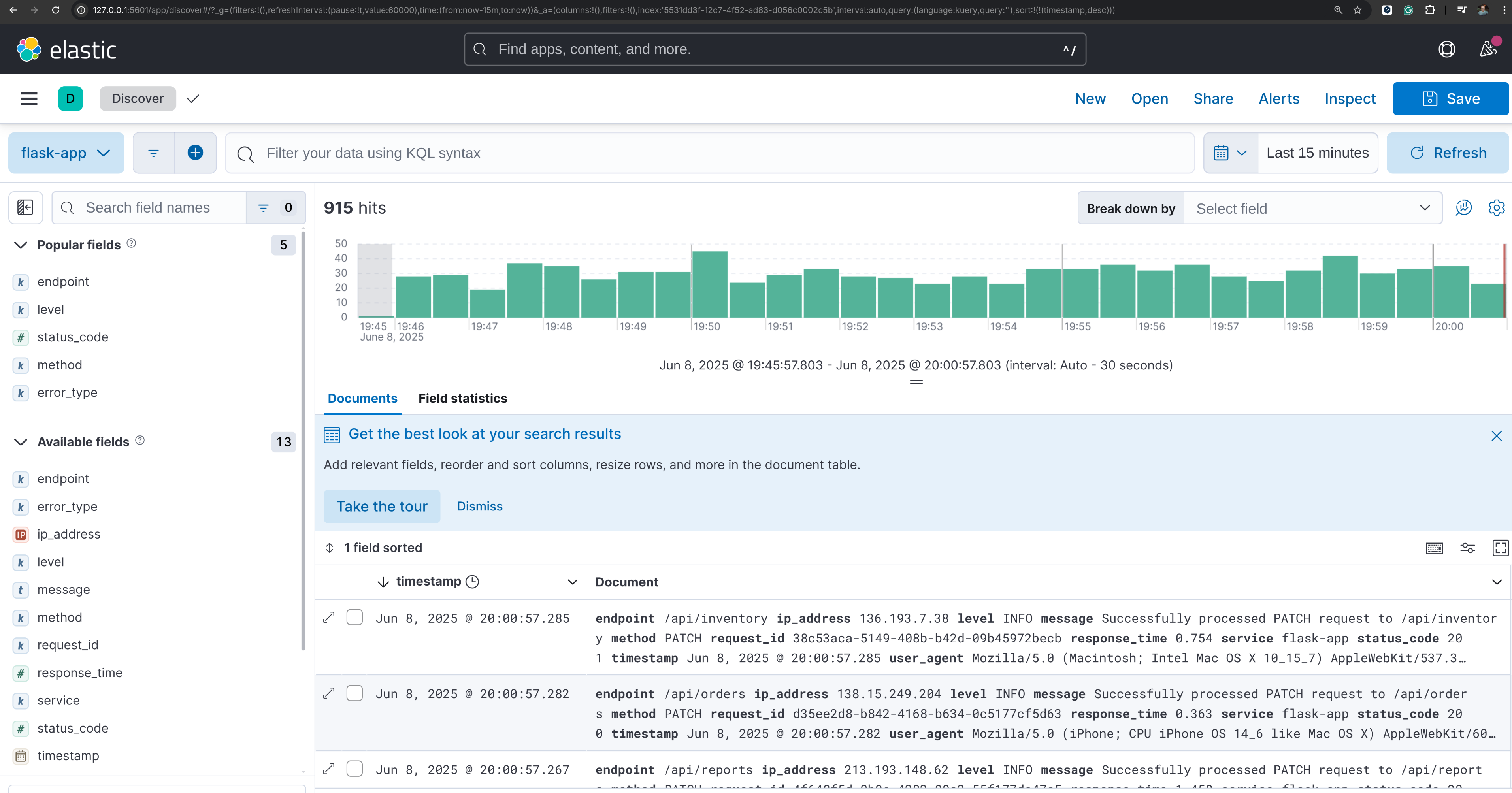


Рисунок 3.5 — Візуалізація логів

### 3.3 Робота з дашбордами Grafana

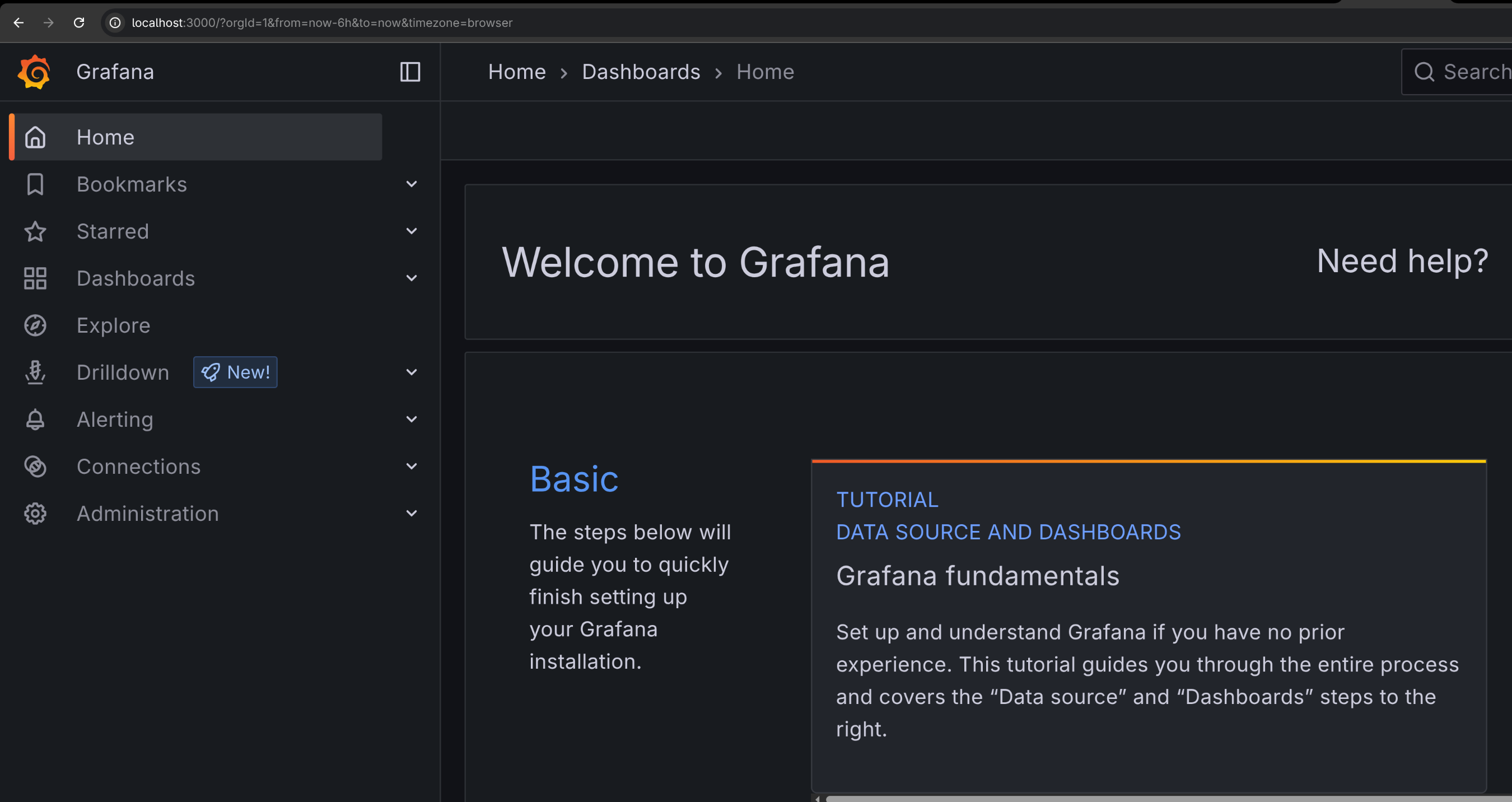
#### 3.3.1 Доступ до Grafana

Для доступу до Grafana:

1. Налаштувати port-forwarding:

kubectl port-forward -n monitoring svc/prometheus-grafana 3000:80

1. Відкрити браузер та перейти за адресою: [http://localhost:3000](http://localhost:3000/)

Рисунок 3.6 Головна сторінка Grafana

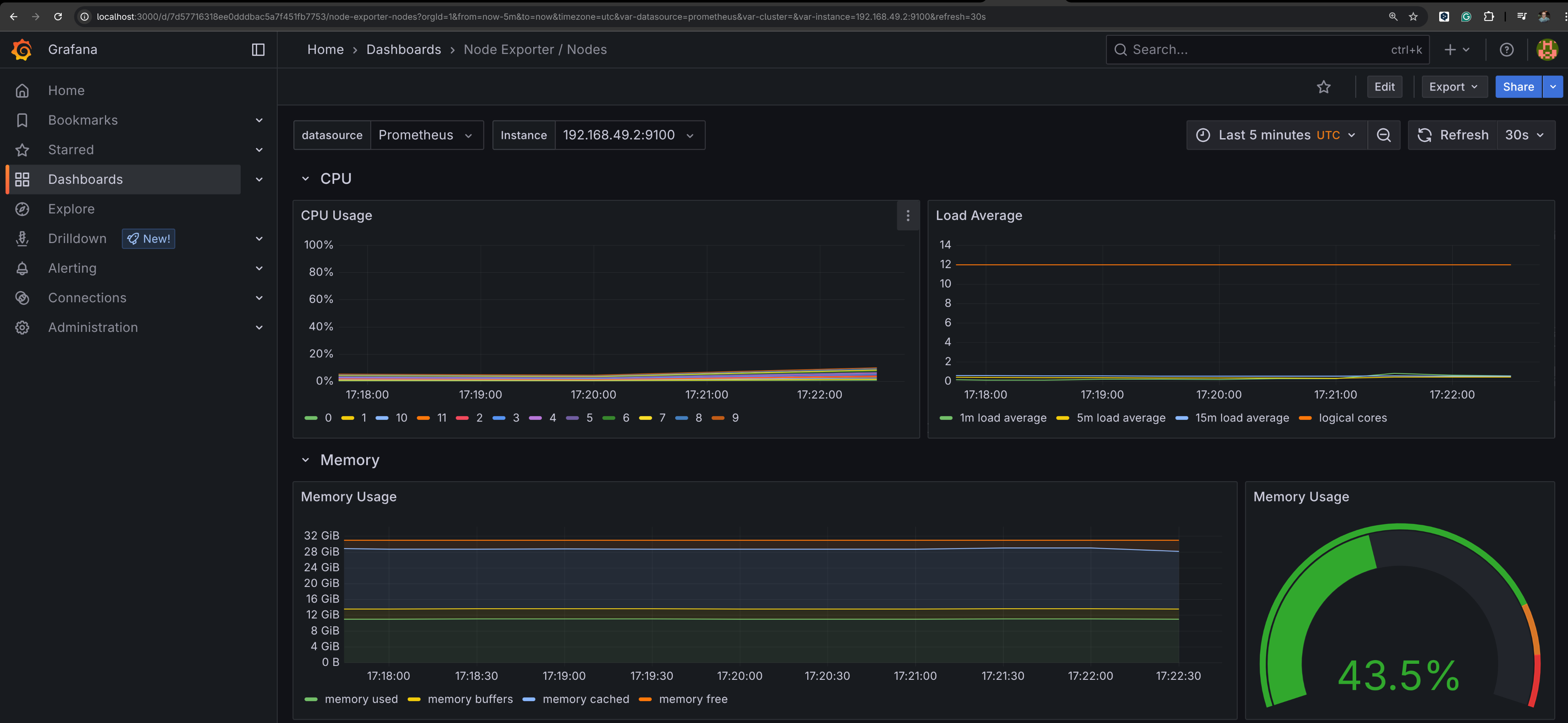
#### 3.3.2 Перегляд існуючих дашбордів

Grafana містить попередньо налаштовані дашборди:

1. **Kubernetes Cluster Overview** — загальний огляд кластера
2. **Kubernetes Pods** — моніторинг подів
3. **Node Exporter Full** — детальний моніторинг вузлів
4. **Kubernetes Deployments** — моніторинг deployments

Для перегляду дашборду:

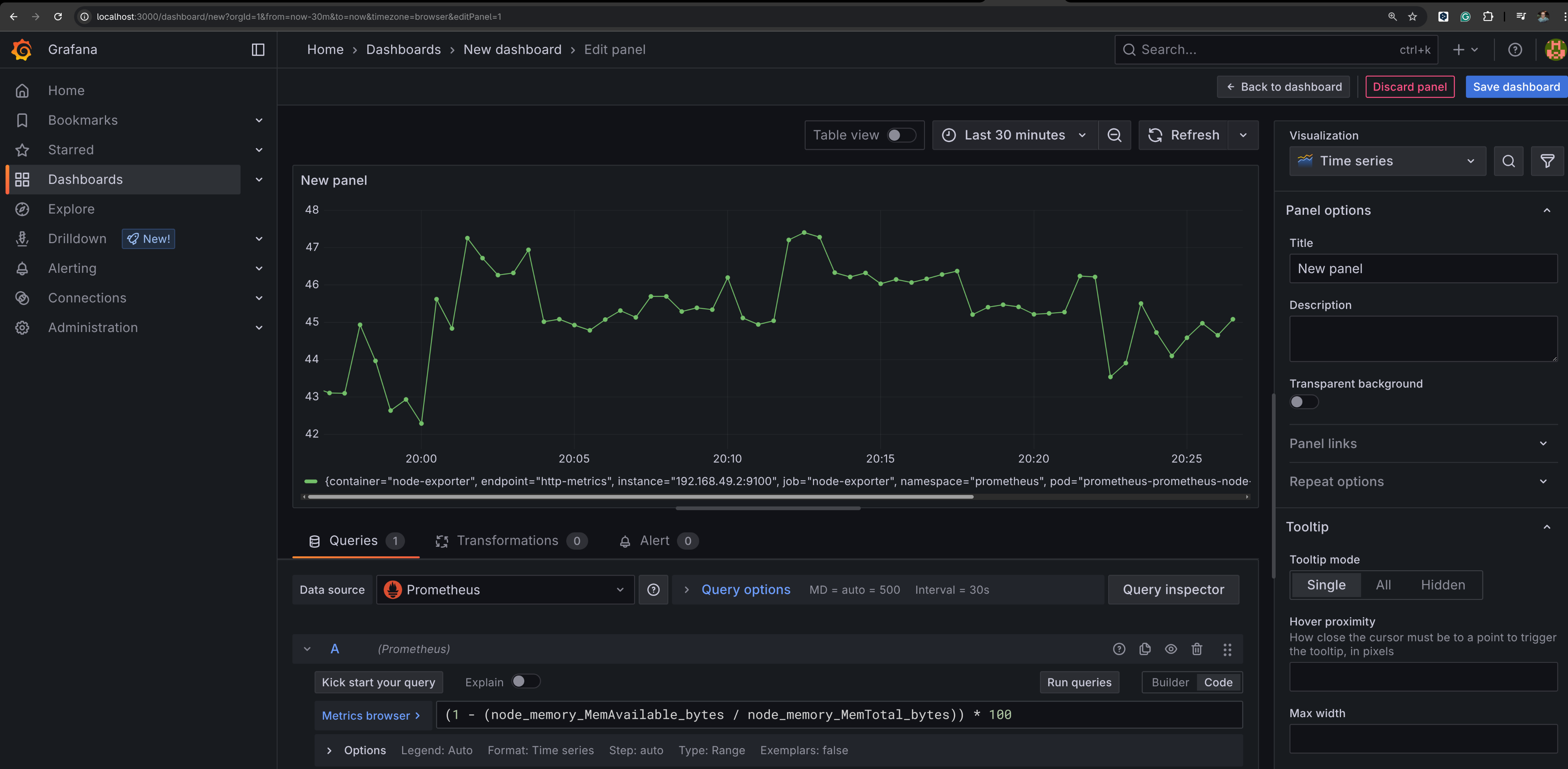
1. Натиснути іконку "+" у лівому меню
2. Обрати "Import"
3. Або перейти до "Dashboards" -> "Browse"

Рисунок 3.7 Дашборд Node Exporter

#### 3.3.3 Створення власних дашбордів

Для створення власного дашборду:

1. Натиснути "+" -> "Dashboard"
2. Натиснути "Add new panel"
3. У полі "Metrics" ввести PromQL запит
4. Налаштувати відображення (тип графіка, кольори, легенду)
5. Натиснути "Apply"
6. Зберегти дашборд натиснувши іконку дискети

Рисунок 3.8 — Створення влясних дашбордів

### 3.4 Система оповіщень

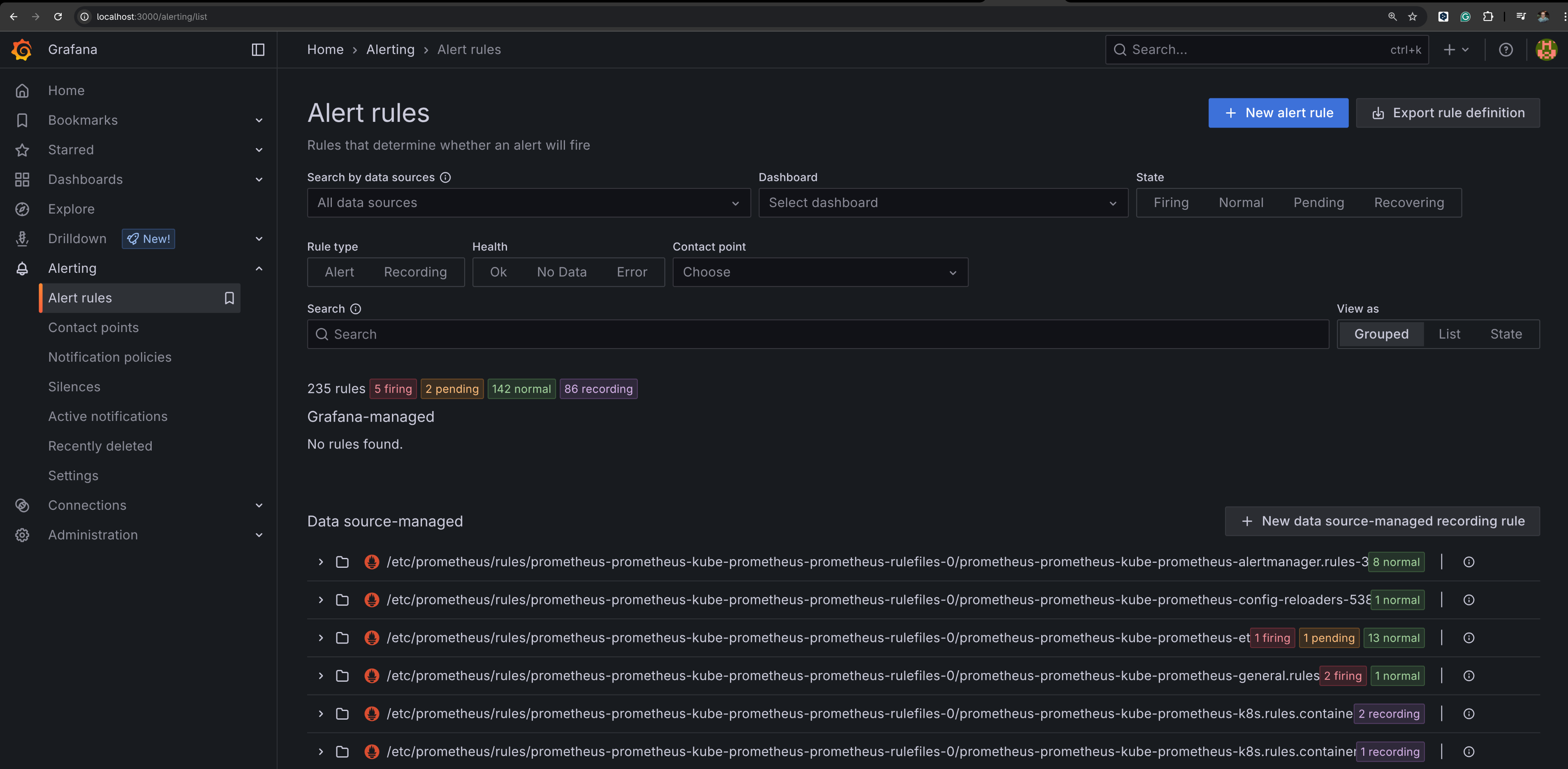
#### 3.4.1 Перегляд активних оповіщень

Для перегляду оповіщень:

1. В Grafana перейти до "Alerting" -> "Alert Rules"
2. Або відкрити AlertManager за адресою (після port-forward):

kubectl port-forward -n monitoring svc/prometheus-kube-prometheus-alertmanager 9093:9093

[http://localhost:9093](http://localhost:9093/)

Рисунок 3.9 — Створення Alerti Rules

#### 3.4.2 Налаштування правил оповіщень

Для створення нового правила оповіщення:

1. В Grafana перейти до "Alerting" -> "Alert Rules"
2. Натиснути "New rule"
3. Налаштувати умови спрацювання
4. Вказати канали сповіщення (email, Slack, тощо)
5. Зберегти правило

### 3.5 Адміністрування системи

#### 3.5.1 Моніторинг стану компонентів

Для перевірки стану компонентів системи:

1. **Перевірити статус подів:**

kubectl get pods -n monitoring

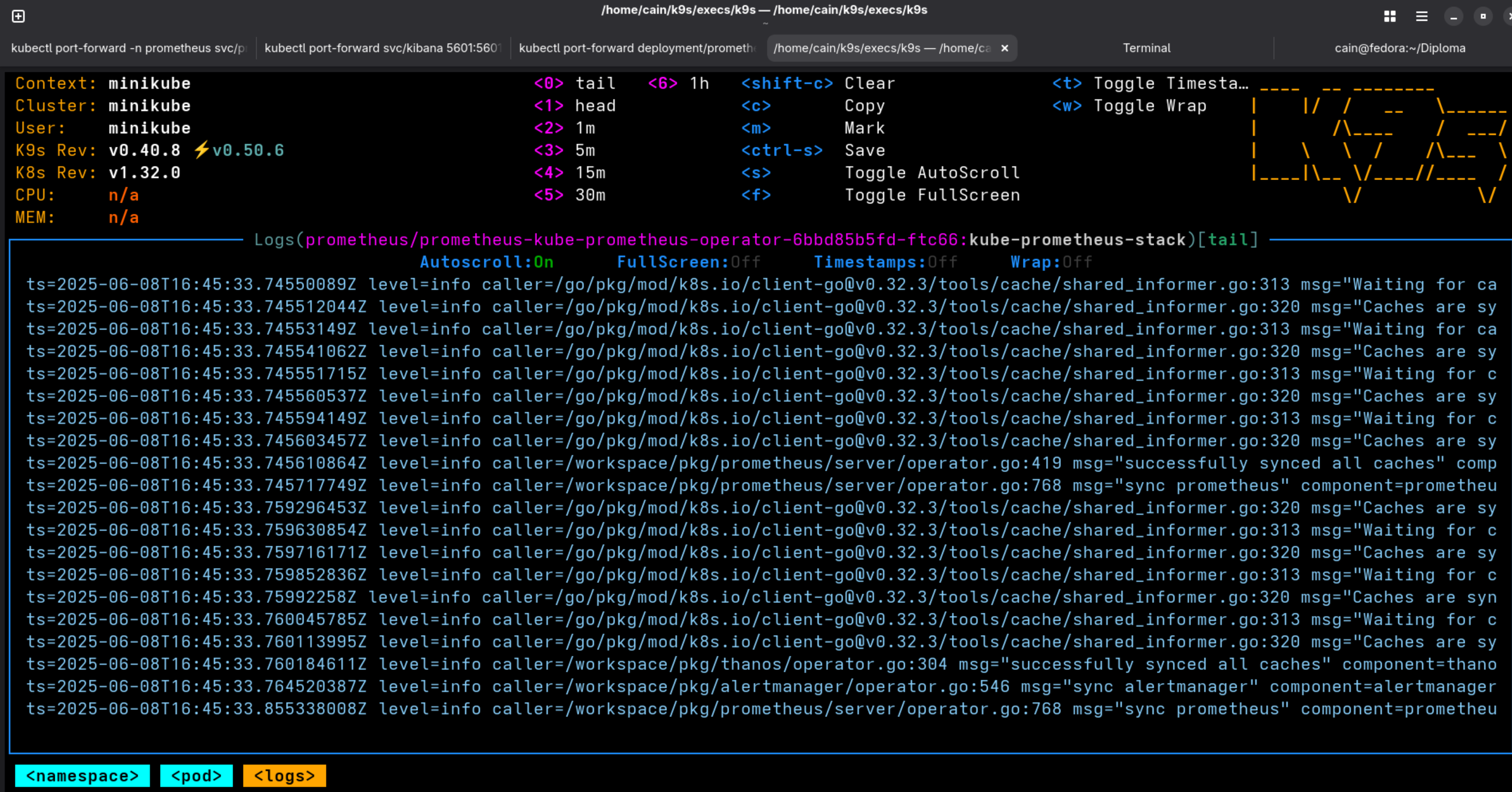
1. **Переглянути логи компонентів:**

kubectl logs -n monitoring deployment/prometheus-operator

1. **Перевірити використання ресурсів:**

kubectl top pods -n monitoring

kubectl top nodes

Рисунок 3.10 — Логи з поди prometheus-operator

#### 3.5.2 Масштабування системи

Для масштабування компонентів:

1. **Масштабування Grafana:**

kubectl scale deployment prometheus-grafana -n monitoring --replicas=3

1. **Додавання вузлів Elasticsearch:**

spec:

nodeSets:

- name: default

count: 3

1. **Налаштування retention policy для Prometheus:**

prometheus:

prometheusSpec:

retention: "30d"

retentionSize: "50GiB"

1. РЕКОМЕНДАЦІЇ ПО ЗАСТОСУВАННЮ

Для успішної роботи з системою моніторингу необхідно:

* **отримати навички роботи з Kubernetes** та основними командами kubectl;
* **мати базове розуміння PromQL** для створення ефективних запитів до метрик;
* **освоїти інтерфейси Grafana та Kibana** для ефективної візуалізації даних;
* **регулярно перевіряти стан системи** та оновлювати компоненти;
* **налаштувати backup** для критично важливих даних та конфігурацій;
* **моніторити використання ресурсів** та масштабувати систему при необхідності.

**Рекомендації по безпеці:**

* Змінити стандартні паролі для всіх компонентів;
* Налаштувати RBAC для обмеження доступу;
* Використовувати TLS для шифрування трафіку;
* Регулярно оновлювати компоненти системи.

5 ПОВІДОМЛЕННЯ

В табл. 5.1 приведені повідомлення користувачу.

Таблиця 5.1 Повідомлення користувачу

|  |  |  |
| --- | --- | --- |
| Текст повідомлення | Опис ситуації | Рекомендовані дії |
| "Pod not found" | Под не знайдений у кластері | Перевірити назву поду та namespace |
| "No data points" | Відсутні дані для візуалізації | Перевірити правильність PromQL запиту |
| "Connection refused" | Неможливо підключитися до сервісу | Перевірити статус сервісу та мережеві налаштування |
| "Authentication failed" | Помилка автентифікації | Перевірити облікові дані |
| "Insufficient resources" | Недостатньо ресурсів для роботи | Збільшити ресурси кластера або оптимізувати навантаження |
| "Index not found" | Індекс не знайдений в Elasticsearch | Створити index pattern або перевірити налаштування Fluent Bit |
| "Alert manager unreachable" | AlertManager недоступний | Перевірити статус AlertManager та мережеві з'єднання |