ДОДАТОК А

Технічне завдання

ЗАТВЕРДЖУЮ

Проректор Українського державного

університету науки і технологій

Анатолій РАДКЕВИЧ

ПРОЕКТУВАННЯ ТА РОЗГОРТАННЯ МАСШТАБОВАНОЇ СИСТЕМИ ДЛЯ ЗБОРУ Й АНАЛІЗУ ТЕЛЕМЕТРІЇ В KUBERNETES ЗА ДОПОМОГОЮ ELASTICSEARCH І GRAFANA

Технічне завдання

ЛИСТ ЗАТВЕРДЖЕННЯ

44165850.01516 – 01 – ЛЗ

Завідувач кафедри КІТ

\_\_\_\_\_\_\_\_\_\_Вадим ГОРЯЧКІН

Керівник розробки

\_\_\_\_\_\_\_\_\_\_Ірина ШАПОВАЛ

Виконавець

\_\_\_\_\_\_\_\_\_\_Антон ЛУКАШОВ

Нормоконтролер

\_\_\_\_\_\_\_\_\_\_Світлана ВОЛКОВА

2025

ЗАТВЕРДЖЕНО

44165850.01516 – 01– ЛЗ

Проектування та розгортання маштабованоі системи для збору й аналізу телеметрії в Kubernetes за допомогою Elasticsearch і Grafana

Технічне завдання

44165850.01516 – 01

Листів 14

2025

ЗМІСТ

ДОДАТОК А 2

ВСТУП 3

1 ПІДСТАВА ДЛЯ РОЗРОБКИ 4

2 ПРИЗНАЧЕННЯ РОЗРОБКИ 5

3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ 6

3.1 Вимоги до функціональних характеристик 6

3.2 Вимоги до надійності 6

3.3 Умови експлуатації 7

3.4 Вимоги до складу параметрів та технічних засобів 8

3.5 Вимоги до інформаційної та програмної сумісності 8

3.6 Вимоги до маркування і упаковки 9

3.7 Вимоги до транспортування і зберігання 9

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ 11

5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ 12

6 ПОРЯДОК І КОНТРОЛЬ ПРИЙМАННЯ 13

7 БІБЛІОГРАФІЧНИЙ СПИСОК 14

# ВСТУП

Система моніторингу на базі Kubernetes розроблена для забезпечення комплексного спостереження за станом контейнеризованих додатків та інфраструктури в середовищі Kubernetes. Система інтегрує сучасні технології моніторингу, включаючи збір метрик, аналіз логів, візуалізацію даних та автоматичні оповіщення про інциденти.

Ключові слова: Kubernetes, моніторинг, метрики, логи, візуалізація, Prometheus, Grafana, Elasticsearch, Kibana, Fluent Bit, оповіщення, дашборд, масштабування, висока доступність, контейнери, мікросервіси.

Причини виникнення: сучасні розподілені системи на базі Kubernetes потребують ефективних інструментів моніторингу для забезпечення стабільної роботи, швидкого виявлення та вирішення проблем, а також оптимізації використання ресурсів. Традиційні підходи до моніторингу не забезпечують достатнього рівня спостережності в динамічному середовищі контейнерів.

Область застосування: хмарні платформи, центри обробки даних, DevOps команди, підприємства з мікросервісною архітектурою, організації що використовують Kubernetes для оркестрації контейнерів.

# 1 ПІДСТАВА ДЛЯ РОЗРОБКИ

Основою для розробки є наказ проректора Українського державного університету науки і технології Радкевич А.В. «Про затвердження тем та призначення керівників дипломних проектів» № 328 ст від 03.03.2025 року.

Тема проекту: “Проектування та розгортання маштабованоі системи для збору й аналізу телеметрії в Kubernetes за допомогою Elasticsearch і Grafana”.

Керівник дипломного проекту: Шаповал Ірина Вікторівна.

2 ПРИЗНАЧЕННЯ РОЗРОБКИ

Функціональне призначення системи полягає в забезпеченні комплексного моніторингу Kubernetes-кластерів та додатків, що в них працюють. Система дозволяє:

* збирати та зберігати метрики продуктивності;
* агрегувати та аналізувати логи з усіх компонентів;
* візуалізувати стан системи через інформативні дашборди;
* автоматично виявляти аномалії та генерувати оповіщення;
* забезпечувати історичний аналіз даних для виявлення трендів.

Експлуатаційне призначення: система моніторингу дозволить операційним командам та розробникам ефективно контролювати стан інфраструктури, швидко реагувати на інциденти, оптимізувати використання ресурсів та забезпечувати високу доступність сервісів.

# 3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

### 3.1 Вимоги до функціональних характеристик

Вимоги до функціональних характеристик наступні:

**Підсистема збору метрик:**

* автоматичне виявлення та моніторинг нових сервісів у кластері;
* збір системних метрик (CPU, пам'ять, диск, мережа);
* збір метрик додатків через Prometheus-експортери;
* підтримка custom метрик від додатків;
* агрегація та попередня обробка метрик;
* довготривале зберігання історичних даних.

**Підсистема збору логів:**

* централізований збір логів з усіх контейнерів;
* парсинг структурованих та неструктурованих логів;
* фільтрація та збагачення логів метаданими;
* індексація для швидкого пошуку;
* автоматична ротація та архівація старих логів.

**Підсистема візуалізації:**

* предналаштовані дашборди для типових сценаріїв;
* можливість створення власних дашбордів;
* різні типи візуалізацій (графіки, таблиці, тепловічні карти);
* real-time оновлення даних;
* експорт дашбордів та звітів.

**Підсистема оповіщень:**

* налаштування правил оповіщень на основі метрик та логів;
* багаторівнева ескалація інцидентів;
* інтеграція з зовнішніми системами (email, Slack, PagerDuty);
* групування та дедуплікація оповіщень;
* управління тишею та підтвердженням інцидентів.

Вхідні дані:

* метрики у форматі Prometheus;
* логи у форматах JSON, plaintext, syslog;
* конфігураційні файли у YAML форматі;
* запити користувачів через веб-інтерфейс.

Вихідні дані:

* візуалізовані дашборди з метриками;
* результати пошуку по логах;
* оповіщення про інциденти;
* експортовані звіти та графіки;
* API для інтеграції з зовнішніми системами.

### 3.2 Вимоги до надійності

Вимоги до надійності наступні:

* система повинна забезпечувати доступність на рівні 99.9% (SLA);
* підтримка горизонтального масштабування всіх компонентів;
* автоматичне відновлення після збоїв окремих компонентів;
* реплікація критичних даних для запобігання їх втраті;
* резервне копіювання конфігурацій та історичних даних;
* відмовостійка архітектура без єдиної точки відмови;
* моніторинг самої системи моніторингу (self-monitoring);
* автоматичне видалення застарілих даних для запобігання переповнення сховищ.

### 3.3 Умови експлуатації

Для забезпечення стабільного функціонування системи необхідно дотримуватись наступних умов [2]:

* розгортання в Kubernetes кластері версії 1.24 або вище;
* наявність виділених ресурсів для компонентів моніторингу;
* стабільне мережеве з'єднання між компонентами;
* налаштований RBAC для безпечного доступу до ресурсів;
* регулярне оновлення компонентів системи;
* дотримання рекомендацій щодо розмірів сховищ даних;
* наявність кваліфікованого персоналу для адміністрування.

### 3.4 Вимоги до складу параметрів та технічних засобів

Мінімальні вимоги до технічних засобів:

**Для малих кластерів (до 50 нод):**

* CPU: 8 vCPU для всіх компонентів моніторингу;
* RAM: 32 GB;
* Дискове сховище: 500 GB SSD для метрик та логів [4] ;
* Мережа: 1 Gbps.

**Для середніх кластерів (50-200 нод):**

* CPU: 24 vCPU;
* RAM: 96 GB;
* Дискове сховище: 2 TB SSD;
* Мережа: 10 Gbps.

**Для великих кластерів (200+ нод):**

* CPU: 48+ vCPU;
* RAM: 192+ GB;
* Дискове сховище: 5+ TB SSD;
* Мережа: 10+ Gbps.

**Програмні компоненти:**

* Kubernetes 1.24+;
* Prometheus 2.40+;
* Grafana 9.0+;
* Elasticsearch 8.0+;
* Kibana 8.0+;
* Fluent Bit 2.0+;
* Alertmanager 0.25+.

### 3.5 Вимоги до інформаційної та програмної сумісності

Вимоги до сумісності:

1. **Операційні системи нод кластера [4]:**
   * Ubuntu 20.04 LTS і новіші;
   * CentOS 8/RHEL 8 і новіші;
   * Debian 10 і новіші.
2. **Container Runtime:**
   * containerd 1.6+;
   * Docker 20.10+;
   * CRI-O 1.24+.
3. **Формати даних:**
   * Метрики: Prometheus exposition format;
   * Логи: JSON, plaintext, syslog;
   * Конфігурації: YAML, JSON.
4. **Протоколи:**
   * HTTP/HTTPS для API;
   * gRPC для внутрішніх комунікацій;
   * OpenMetrics для метрик.
5. **Інтеграції:**
   * Prometheus-сумісні експортери;
   * Webhook для оповіщень [5];
   * REST API для зовнішніх систем.

### 3.6 Вимоги до маркування і упаковки

Система постачається у вигляді:

* Helm чартів для розгортання в Kubernetes;
* Docker образів з відповідними тегами версій;
* Конфігураційних файлів та документації.

Приклад маркування представлений на рис. 1.

|  |
| --- |
| ПРОЕКТУВАННЯ ТА РОЗГОРТАННЯ МАСШТАБОВАНОЇ СИСТЕМИ ДЛЯ ЗБОРУ Й АНАЛІЗУ ТЕЛЕМЕТРІЇ В KUBERNETES ЗА ДОПОМОГОЮ ELASTICSEARCH І GRAFANA  розробник: Лукашов Антон  УДУНТ, кафедра КІТ  м. Дніпро, вул. Лазаряна, 2  2025 |

Рис. 1. Приклад маркування

## 3.7 Вимоги до транспортування і маркування

Програмний продукт розповсюджується через:

* Приватний Docker Registry;
* Git репозиторій з Helm чартами;
* Архівні файли на захищених носіях.

Вимоги до зберігання:

* Резервні копії на окремих носіях;
* Шифрування чутливих даних;
* Контроль версій всіх компонентів;

## 4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

До складу програмної документації мають входити:

* Архітектурна документація системи;
* Інструкція з розгортання;
* Керівництво адміністратора;
* Керівництво користувача;
* Опис API та інтеграцій;
* Приклади конфігурацій;
* Troubleshooting guide.

Вся документація повинна відповідати вимогам ДСТУ та стандартам оформлення технічної документації [1].

## 5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

В табл. 1 приведені стадії та етапи розробки програмного продукту.

Таблиця 1. Стадії та етапи розробки

|  |  |  |
| --- | --- | --- |
| Стадія | Зміст робіт | Терміни виконання |
| Технічне завдання | Постановка задачі, збір інформації, вибір та обгрунтування критеріїв розробки. Попередній вибір методів рішення задач. Узгодження і затвердження технічного завдання. | 03.03.25 - 11.04.25 |
| Розробка | Реалізація підсистем збору метрик та логів. | 11.04.25 - 30.04.25 |
|  | Розробка дашбордів та системи оповіщень. | 01.05.25 - 15.05.25 |
| Тестування | Функціональне тестування, навантажувальне тестування, тестування відмовостійкості. | 16.05.25 - 31.05.25 |
| Документування | Підготовка технічної документації, інструкцій, керівництв. | 01.06.25 - 09.06.25 |

# 6 ПОРЯДОК І КОНТРОЛЬ ПРИЙМАННЯ

Контроль за виконанням роботи здійснює керівник дипломного проекту: Шаповал Ірина Вікторівна.

7 БІБЛІОГРАФІЧНИЙ СПИСОК

1. Івченко, Ю.М. Основи стандартизації програмних систем [Текст]: методичні вказівки до дипломного проектування та лабораторних робіт / уклад.: Ю. М. Івченко, В. І. Шинкаренко, В. Г. Івченко; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В. Лазаряна, 2009. - 38 с.
2. Kubernetes Documentation. Cloud Native Computing Foundation. URL: <https://kubernetes.io/docs/>
3. Prometheus: From metrics to insight. URL: <https://prometheus.io/docs/>
4. Elastic Stack Documentation. Elastic. URL: <https://www.elastic.co/guide/>
5. Grafana Documentation. Grafana Labs. URL: <https://grafana.com/docs/>