Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: **Моніторинг та автоматизація керування серверами в високонавантажених системах**

Виконав(ла): студент(ка) 6 курсу, групи СНм-61
спеціальності
122 «Комп'ютерні науки»
(шифр і назва спеціальності)

(підпис) Волоха А.О. (прізвище та ініціали)

Керівник (підпис) Дмитроца Л.П. (прізвище та ініціали)

Нормоконтроль (підпис) Мацюк О.В. (прізвище та ініціали)

Завідувач кафедри (підпис) Боднарчук І.О. (прізвище та ініціали)

Рецензент (підпис) Петрик М.Р. (прізвище та ініціали)

Тернопіль
2020

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
« » 20___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 122 «Комп'ютерні науки»
(шифр і назва спеціальності)

студенту Волосі Антон Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Моніторинг та автоматизація серверів в високонавантажених системах

Керівник роботи Дмитроца Леся Павлівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «___» _____ 20___ року № _____

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи наукові літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1 Аналіз процесів та існуючих програмних засобів моніторингу та управління подіями. 2 Основні принципи, моделі і методи виявлення причинно-наслідкових відносин між подіями 3. Реалізація системи моніторингу та виявлення причинно-наслідкових зв'язків між подіями. 4. Охорона праці та безпека в надзвичайних ситуаціях. Перелік використаних. джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Дмитроца Л. П., доцент		
Безпека в надзвичайних ситуаціях	Стадник І. Я., професор		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми кваліфікаційної роботи		Виконано
2	Аналіз літературних джерел		Виконано
3	Обґрунтування актуальності дослідження		Виконано
4	Аналіз предмету дослідження та предметної області		Виконано
5	Оформлення розділу «Аналіз процесів та існуючих програмних засобів моніторингу та управління подіями»		Виконано
6	Оформлення розділу «Основні принципи, моделі і методи виявлення причинно-наслідкових відносин між подіями »		Виконано
			Виконано
7	Оформлення розділу «Практична реалізація системи моніторингу та виявлення причинно-наслідкових зв'язків між подіями»		Виконано
8	Оформлення розділу «Охорона праці та безпека в надзвичайних ситуаціях»		Виконано
			Виконано
9	Нормоконтроль		Виконано
10	Перевірка кваліфікаційної роботи на плагіат		Виконано
11	Попередній захист кваліфікаційної роботи		Виконано
12	Захист кваліфікаційної роботи		

Студент _____ Волоха А.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Дмитроца. Л.П
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Моніторинг та автоматизація керування серверами в високонавантаженій мережі // Кваліфікаційна робота освітній рівень «Магістр» // Волоха Антон Олександрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНнм-61 // Тернопіль, 2020 // с.89, рис. – 25, табл. – 10, бібліогр. – 49.

Ключові слова: BIG DATA, ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, МОНІТОРИНГ СЕРВЕРІВ, ELASTICSEARCH, RSYSLOG, МЕТОДОЛОГІЯ, NOSQL DATABASE

У кваліфікаційній роботі проведено дослідження критеріїв та засобів визначення моніторингу та автоматизації керування серверами в високонавантаженій мережі.

Основним завданням кваліфікаційної роботи є аналіз методів моніторингу та автоматизації керування серверами в високонавантаженій мережі.

В першому розділі було розглянуто транзакційні і потокові системи моніторингу, моніторинг інформаційних систем, методи журналювання подій в IT інфраструктурі, розглянуто існуючи засоби аналізу інформації про події такі як Adiscon LogAnalyzer, Splunk та IBM infoShpere.

В другому розділі було розглянуто архітектуру централізованого логування, пошуку та виявлення шаблонів повідомлень в журналах та майбутню модель машинного навчання.

В третьому розділі було здійснено побудову архітектури системи моніторингу, розглянуто способи реалізації та приведено приклади налаштувань які дадуть бажаний результат моніторингу системи.

ANNOTATION

Monitoring and automation of servers control in high-load systems // Thesis of OR "Master" // Volokha Anton Alexandrovich // Ternopil National Technical University named after Ivan Pulyuy, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNNM group -61 // Ternopil, 2020 // p.89, fig. - 25, table. - 19, bibliogr. - 49.

Keywords: BIG DATA, TECHNOLOGY INFORMATION, SERVER MONITORING, ELASTICSEARCH, RSYSLOG, METHODOLOGY, NOSQL DATABASE

In the qualifying work the research of criteria and means of definition of monitoring and automation of management of servers in high-load network.

The main task of the thesis is the analysis of methods of monitoring and automation of server management in a high-load network.

In the first section, transactional and non-transactional monitoring systems, information systems monitoring, methods of event logging in IT infrastructure were discussed, as well as innovative tools for analyzing event information such as Adiscon LogAnalyzer, Splunk and IBM infoShpere.

The second section discusses the architecture of centralized logging, search and detection of message templates in journals and the future model of machine learning.

In the third section, the architecture of the monitoring system was built, the methods of implementation were explained, and examples of settings were given that would give the desired result of system monitoring.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІКТ – інформаційно-комунікаційні технології.

ІТ – інформаційні технології.

ОС – операційна система.

БД – база даних.

СКБД – система керування базою даних.

DNS (англ. Domain Name System) – ієрархічна розподілена система перетворення імені хоста) в IP-адресу.

URI (англ. Uniform Resource Identifier) – компактний рядок літер, який однозначно ідентифікує окремий абстрактний чи фізичний ресурс.

BPMN – Система умовних позначень для моделювання бізнес-процесів

TCP (англ. Transmission Control Protocol) – Протокол призначений для управління передачею даних у комп'ютерних мережах

UDP (англ. User Datagram Protocol) – це один з найпростіших протоколів транспортного рівня

IAC – Інформаційно аналітична система

LCS (англ. longest common subsequence) – Пошук найдовшої спільної підпослідовності – це завдання пошуку послідовності, яка є підпослідовністю кількох послідовностей.

ELK – абревіатура для трьох проєктів з відкритим кодом: Elasticsearch, Logstash та Kibana.

VPS (англ. Virtual private server) – Віртуальний виділений сервер, VDS або – послуга, в рамках якої користувачеві надають віртуальний сервер.

ЗМІСТ

ВСТУП

Актуальність теми роботи. В наш час діяльність більшості підприємств покладається на засоби обчислювальної техніки і автоматизації, тобто на інформаційні системи. Застосування комп'ютерних технологій дозволяє знизити час, що витрачається на виконання рутинних операцій, збільшити продуктивність процесів підприємства, скоротити операційні витрати і ризики. Таким чином, інформаційні технології є компаньйоном і опорою сучасного бізнесу, забезпечуючи його необхідними інформаційними сервісами.

З огляду на те що IT-сервіси дозволяють сучасному бізнесу ефективно функціонувати, вкрай важливо забезпечити передбачуваність їх роботи, захиститися від різного роду несправностей, збоїв і відмов. Дана робота лягає на плечі інженерів технічної підтримки інформаційних систем. У коло їхніх обов'язків входить моніторинг систем і вирішення виникаючих проблем, а саме:

- виявлення факту існування проблеми;
- діагностика, визначення порушених сервісів і систем, а також виникають ефектів;
- пошук причини, що викликала проблему;
- пошук можливих рішень;
- визначення необхідних дій для усунення проблеми;
- виконання операцій, необхідних для усунення проблеми;

При цьому інженери, в разі виникнення проблем, повинні вирішувати їх максимально швидко, щоб не постраждала якість бізнесу, а також ділова репутація та довіра до підприємства в цілому та IT зокрема.

Сучасні інформаційні системи, що надають необхідні бізнесу сервіси, відрізняються великим рівнем складності: вони багатокомпонентні, найчастіше компоненти розподілені в обчислювальної мережі і взаємодіють один з одним за допомогою різних технологій і протоколів передачі даних. Крім того, робота одних систем, як правило, спирається на роботу інших систем, наприклад, доступ до веб-ресурсів по URI спирається на систему DNS, а також на систему,

яка надає ці ресурси – веб-сервер, який в свою чергу спирається на сервер додатків, запитувач інформацію в базах даних сервера баз даних (класична веб-система). Як видно з наведеного прикладу, незважаючи на певну автономність і модульність, інформаційні системи дуже тісно пов'язані між собою, що робить пошук причин несправностей, пошук рішень та інші завдання досить важкими, під силу лише висококваліфікованим інженерам з великим досвідом.

У більшості випадків завдання виявлення самого факту проблеми вирішується системами моніторингу, які досить оперативно доповідають про які – небудь події в спостережуваних інформаційних системах. Іншим джерелом інформації про наявність проблем служать користувачі, які виявляють недоступність або погіршення якості послуг, інформаційних сервісів. Решта перераховані вище завдання інженери підтримки змушені вирішувати лише на основі своїх знань, досвіду, системних журналів подій і доступної інформації з супровідної документації до систем і в мережі Інтернет.

Очевидно, що завдання діагностики, пошуку причини, пошуку вирішення проблеми є найбільш витратними за часом, а необхідне на них час так само, як і успішність виконання, сильно залежить від кваліфікації інженера. На питання "У мене не працює система. Чому?" є лише один правдиву відповідь: "Не знаю. Можливо..." – і цих "можливо" зазвичай необмежено багато.

Як правило, основою для "розслідування" виниклої проблеми є журнали подій інформаційних систем, однак дані журнали містять велику кількість інформації, зазвичай важкі для читання і часто необхідно аналізувати кілька журналів паралельно, щоб виявити причину проблеми. Після того, як причина виявлена, необхідно знайти способи її усунення, що теж часто є нетривіальним завданням – доступної інформації дуже багато, і навіть грамотно складений запит до пошукових систем типу Google, DuckDuckGo і іншим, далеко не відразу дозволяє виявити шуканий відповідь.

Крім оперативного вирішення проблем дуже вітаються превентивні дії, що дозволяють не доводити ситуацію до проблемної, наприклад, своєчасна заміна жорсткого диска сервера максимально близько до моменту його відмови. Як

правило, вартість цих заходів суттєво нижче, ніж економічні та репутаційні втрати бізнесу від зупиненого IT-сервісу. Однак пророкування проблем, так само як і будь-яке передбачення подій взагалі, є не менш складним завданням.

Зазначені вище особливості: велика кількість інформації, що аналізується, неструктуровані тексти журналів, повнотекстовий пошук, необхідність в прогнозі несправностей – роблять проблематичним застосування звичних підходів до автоматизації. З іншого боку, саме через цих особливостей застосування технологій обробки великих даних (Big Data) і машинного навчання (Machine Learning) бачиться доцільним і ефективним.

Метою дослідження є розробка математичного та програмного забезпечення, що дозволяє підвищити ефективність роботи інженерів технічної підтримки за рахунок надання можливих причин виникнення подій, тим самим обґрунтовано направляючи пошук і розслідування подій, інцидентів та проблем, та в разі кричиної ситуації виведе сервер з DNS кластеру та вимкне його.

Для досягнення поставленої мети в кваліфікаційній роботі сформульовані наступні **завдання:**

- Провести аналіз процесів управління подіями.
- Провести огляд і порівняння існуючих засобів обробки і аналізу інформації про події в IT-інфраструктурі.
- Розробити методи збору інформації про події в IT-інфраструктурі, а також структуру одиничного події.
- Розробити методи по вилученню даних з повідомлень про події.
- Розробити модель виявлення причинно-наслідкових зв'язків між подіями.
- Налаштувати систему моніторингу виявлення причинно-наслідкових зв'язків між подіями та забезпечити їх зв'язну роботу.
- Дослідити і верифікувати теоретичні положення в ході дослідної експлуатації розробленої системи моніторингу на даних мережевого і серверного обладнанні S4M Tech Inc.

Об'єктом дослідження в кваліфікаційній роботі є події, зафіксовані в ІТ-інфраструктурі і представлені записами системних журналів і повідомленнями систем моніторингу.

Предметом дослідження є моделі, методи і програмні засоби виявлення причинно-наслідкових зв'язків між подіями в ІТ-інфраструктурі, їх структура і застосування в роботі служби технічної підтримки та автоматичному реагцванню на події.

1 АНАЛІЗ ПРОЦЕСІВ ТА ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ МОНІТОРИНГУ ТА УПРАВЛІННЯ ПОДІЯМИ

1.1 Транзакційні і потокові системи

Для забезпечення своєї діяльності сучасні організації створюють і використовують IT-інфраструктуру. Згідно бібліотеці ITIL [1] (Information Technology Infrastructure Library – бібліотека інфраструктури інформаційних технологій), IT-інфраструктура включає в себе все апаратне і програмне забезпечення, мережі, інженерне забезпечення та т.д.

Необхідні для розробки, тестування, надання, моніторингу, контролю або підтримки IT-послуг, при цьому IT-інфраструктура містить в собі компоненти інформаційних технологій, але не включає пов'язані з ними персонал, процеси і документацію.

Всі інформаційні системи, що становлять IT-інфраструктуру, можна розділити на системи транзакційного характеру і системи потокового характеру.

В основі роботи транзакційних систем лежить поняття транзакції [1] – набір пов'язаних операцій, які переводять систему в новий стан, причому виконатися повинні все операції або не виконатися ні однієї. Схема процесу роботи такої системи в нотації BPMN 2.0 [2] показана на рисунку 1.1

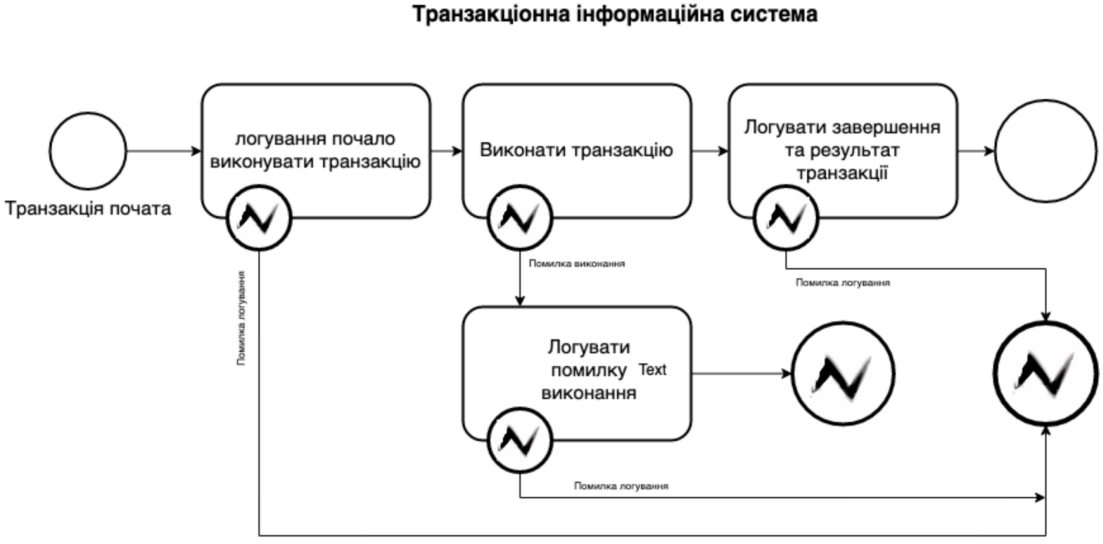


Рисунок 1.1 – Схема процесу роботи транзакційної системи

Підпроцес "Виконати транзакцію" реалізований так само, як і головний процес. Такий підхід дозволяє зберігати систему і дані в ній в цілісному стані. Слід звернути увагу на журнал всіх етапів виконання транзакції, що з однієї сторони створює в разі великого кількості транзакцій призведе до різкого і значного зростання місця, зайнятого журналами, з іншого боку в разі виникнення помилок на будь-якому етапі дозволяє їх відстежити, розслідувати і усунути.

Прикладами транзакційних систем є СКБД і системи, побудовані на їх основі: Інтернет-магазини, системи електронних платежів, і ін.

Потокові системи характеризуються роботою з потоками повідомлень: вхідні повідомлення обробляються системою і передаються далі іншим системам. Схема процесу роботи такої системи в нотації BPMN 2.0 показана на рисунку 1.2.

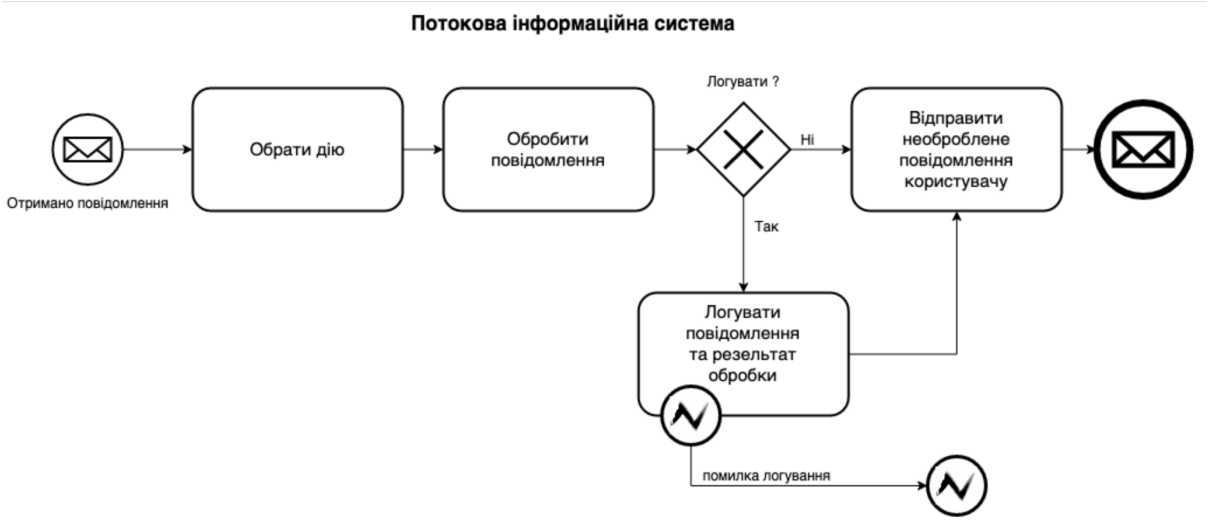


Рисунок 1.2 – Схема процесу роботи потокової системи

Тут завдання " Вибрати дію " зазвичай реалізується в вигляді набору правил, на відповідність яким перевіряється входить повідомлення і вибирається дію по обробці.

Прикладом потокових систем є все мережеве обладнання: комутатори, маршрутизатори, мережеві фільтри.

Слід зазначити, що з огляду на зазвичай великого потоку повідомлень журнал в таких системах є не обов'язковим, так як призведе до миттєвої просідання продуктивності: запис в журнал – порівняно повільна операція, на тлі великого потоку повідомлень система буде в основному займатися записом в журнал, а не корисної роботою. За цією причини журналіруются не все операції, як в транзакційної системі, а лише вибіркові, важливі по думку адміністратора системи.

Управління з потоковими системами проводиться в транзакційному режимі (Див. рисунок 1.1) зі всіма його особливостями.

1.2 Моніторинг інформаційних систем

Стан систем обох типів необхідно постійно відслідковувати, щоб оперативно вирішувати виникаючі інциденти. Для цього організовуються системи моніторингу. Дані системи є клієнт-серверними [3], клієнтська частина називається агент моніторингу, встановлюється на спостережувану систему і займається відстеженням цієї системи і виконанням команд серверної частини вузла управління – для управління спостерігається системою. Системи моніторингу можна розділити на системи пасивного моніторингу і системи активного моніторингу.

Пасивний моніторинг, згідно [1], це моніторинг конфігураційнної одиниці, IT-послуги або процесу, який ґрунтується на оповіщеннях або повідомленнях про поточному стані. Схема процесу пасивного моніторингу в нотації BPMN 2.0 представлена на рисунку 1.3.

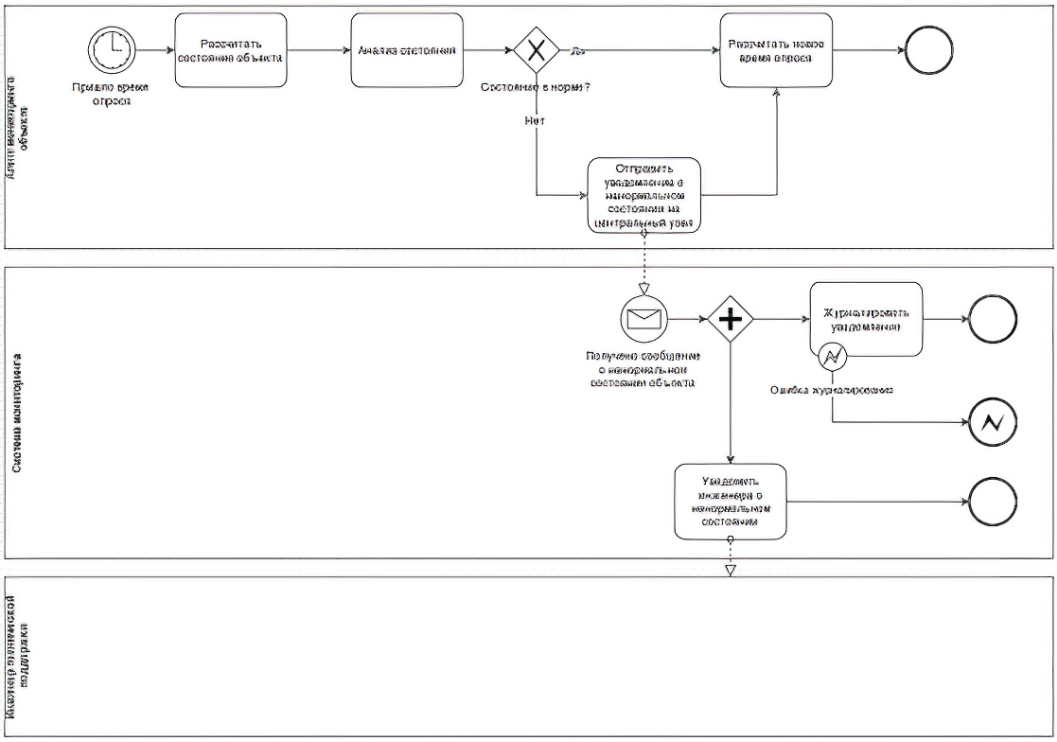


Рисунок 1.3 – Схема процесу пасивного моніторингу

Агент моніторингу періодично заміряє показники спостережуваного об'єкта і в разі відхилення показників від нормального стану сповіщає про це центральний вузол. Центральний вузол журналізує отримане повідомлення в цілях подальшого аналізу, а також повідомляє відповідального інженера, який повинен зробити будь-які дії для повернення спостерігається інформаційної системи в нормальне стан або посилає сигнал до сервісу який автоматично вимикає сервер.

Така схема моніторингу хороша своєю простотою, однак страждає від істотного недоліку: якщо спостерігається система стає недоступною, центральний вузол про це не дізнається і буде думати, що відсутність повідомлень від спостережуваної системи свідчить про її нормальну роботу, хоча проблема – недоступність – присутній. Даний недолік подоланий в системах активного моніторингу.

Активний моніторинг, згідно [1], це моніторинг конфігураційних одиниць або IT-послуг, який використовує автоматизовані регулярні перевірки для

відстеження поточного статусу об'єкта моніторингу. Схема процесу активного моніторингу в нотації BPMN 2.0 представлена на рисунку 1.4.

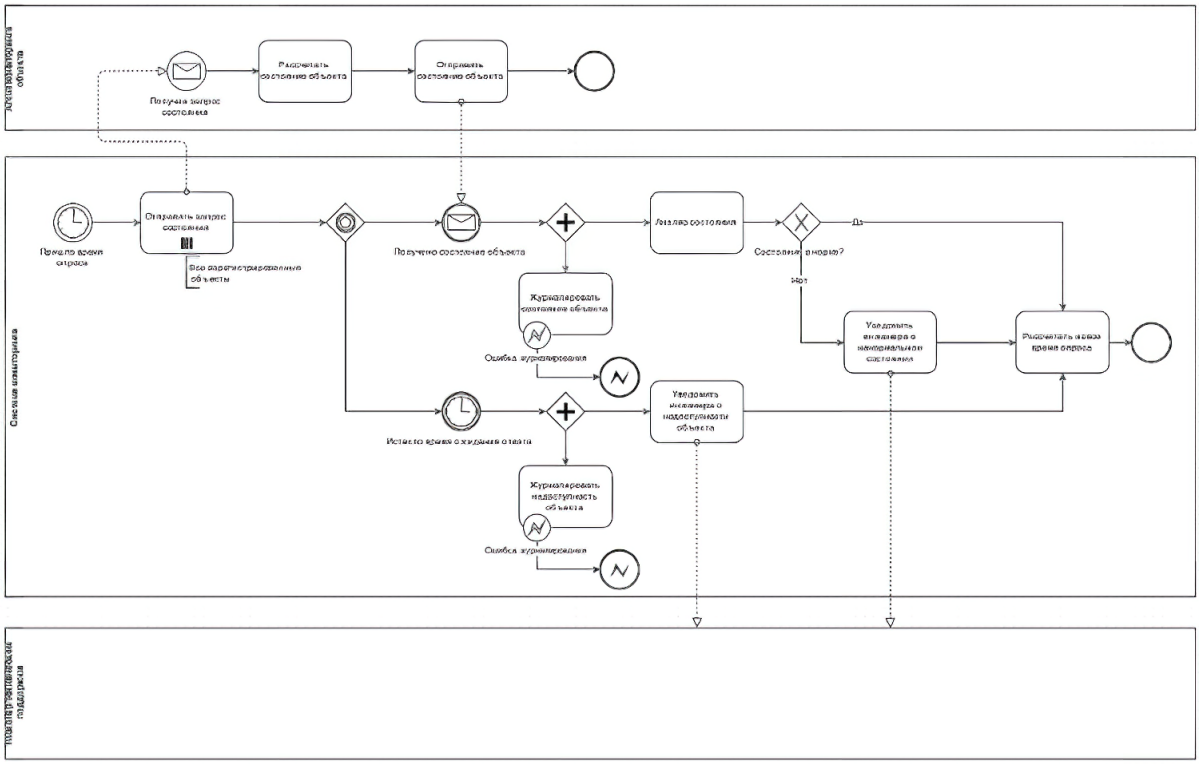


Рисунок 1.4 – Схема процесу активного моніторингу

Центральний вузол періодично опитує агентів моніторингу на спостережуваних системах, оцінює отримані значення показники стану і в разі відхилень сповіщає відповідального інженера. У разі відсутності відповіді від спостережуваної системи в протягом деякого часу система вважається недоступною про чому так же оповіщається відповідальний інженер. Інформація про стан або недоступності систем журналізується.

Така схема кілька складніше в частині роботи центрального вузла, проте дозволяє з похибкою рівною періоду опитування знати поточний стан або факт недоступності спостережуваних систем. Слід звернути увагу, що при великому кількості спостережуваних систем і частих опитуваннях журнали центрального вузла будуть стрімко розростатися.

1.3 Журналювання подій IT-інфраструктури

Найчастіше моніторинг IT-інфраструктури закінчується на розгортанні систем активного або пасивного моніторингу та оповіщення відповідальних.

Однак системи і активного, і пасивного моніторингу навіть при спільній роботі можуть лише відстежувати зовнішні показники спостерігаються інформаційних систем, наприклад, завантаження центрального процесора або оперативної пам'яті, кількість відкритих TCP сесій, статус процесу/сервісу і ін.

Вони не можуть сказати нічого про внутрішньому стані спостерігається системи, так як аспекти її роботи приховані всередині неї. Щоб все ж мати можливість якось відстежувати внутрішню роботу інформаційної системи, розробник вбудовує в неї підсистему журналювання і оповіщення. Тоді, по мірі роботи інформаційної системи, що виникають в ній події заносяться в системний журнал або передаються на центральний сервер журналювання [4]. Схема такого централізованого процесу журналювання в нотації BPMN 2.0 показана на рисунку 1.5.

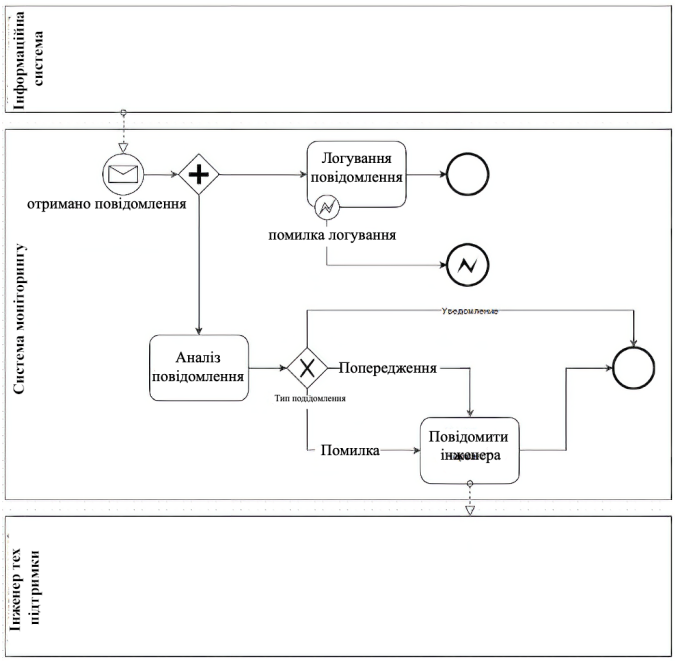


Рисунок 1.5 – Схема процесу централізованого журналюввання

Можна помітити, що даний процес дуже нагадує процес пасивного моніторингу, де відправку повідомлень про події ініціює не агент, а сама

спостережувана система. Крім цього присутній елемент аналізу, як в процесі активного моніторингу. Очевидним наслідком цього є те, що централізоване журнал страждає від проблем як активного (велика кількість подій і записів журналу), так і пасивного (не відстежується доступність) моніторингу.

1.4 Існуючі засоби аналізу інформації про події

Описані вище процеси дозволяють відстежити стан і події в IT-інфраструктурі, проте в разі виникнення інцидентів роботи по аналізу і відновленню нормального стану лежать на плечах відповідального інженера. При цьому його діяльність не формалізована і спирається виключно на знання і досвід, а також відомості від систем моніторингу та централізованого журналювання, з офіційної документації та Інтернет. Загальна схема процесу роботи інженера показана на рисунку 1.6.

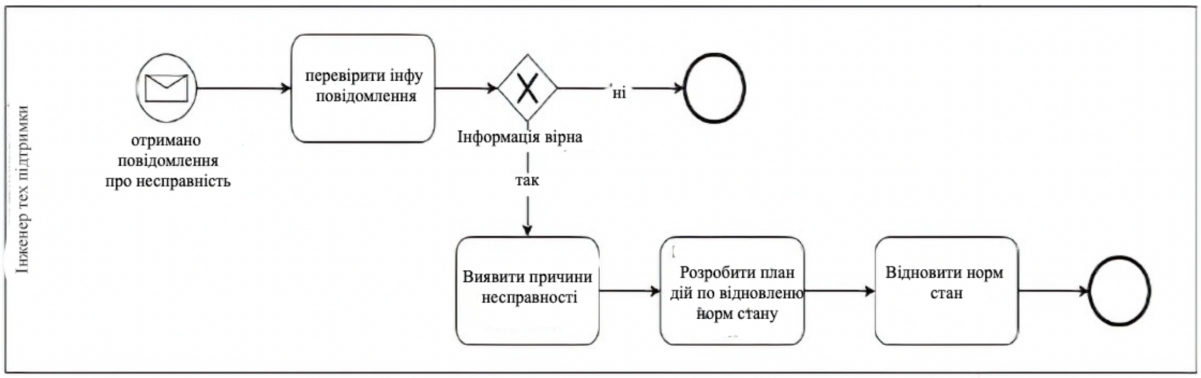


Рисунок 1.6 – Схема процесу роботи інженера по відновленню нормального стану

З огляду на великого кількості типів систем, а також самих систем, інженеру часто досить складно володіти повною інформацією, необхідною для розслідування подій і прийняття адекватних рішень по відновленню і ліквідації наслідків. Крім цього складно передбачити час, необхідне для вирішення інциденту або проблеми.

Щоб полегшити доступ інженера до інформації про події, відображених в системних журналах і повідомленнях систем моніторингу, а також направити його пошуково-дослідницьку діяльність можна задіяти системи аналізу зібраної інформації. Нижче розглянуті широко відомі на поточний момент подібні системи.

1.4.1 Adiscon LogAnalyzer

Adiscon LogAnalyzer [5] – система побудови звітності на основі інформації, отриманої по протоколу Syslog [4]. Інтерфейс системи показаний на рисунку 7.

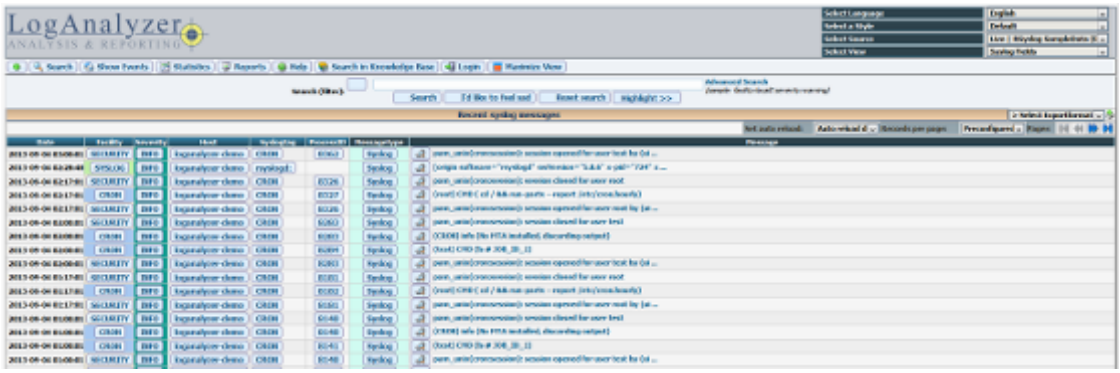


Рисунок 1.7 – Інтерфейс Adiscon LogAnalyzer

Дана система дозволяє переглядати зареєстровані події, виробляти пошук в зібраної інформації і будувати нескладні графіки. Основне перевага системи – наявність бази знань [6] (посилання на http://kb.Monitorware.com/), що представляє собою Інтернет-форум.

Головний недолік цієї системи полягає в тому, що вона не дозволяє виявити навіть статистично зв'язку між відбулися подіями.

1.4.2 Splunk Enterprise

Splunk Enterprise [7] – система моніторингу надходять повідомлень, побудови звітності та аналітики від компанії Splunk.

Дана система є платною, проте має приємний веб-інтерфейс (Див. рисунок 1.8), підтримує велику кількість джерел інформації про події, дозволяє будувати різні графіки, звіти і панелі індикаторів (dashboard).

Крім цього в Splunk вбудований гнучкий повнотекстовий пошук, що дозволяє виконувати гранульований пошук і фільтрацію інформації.

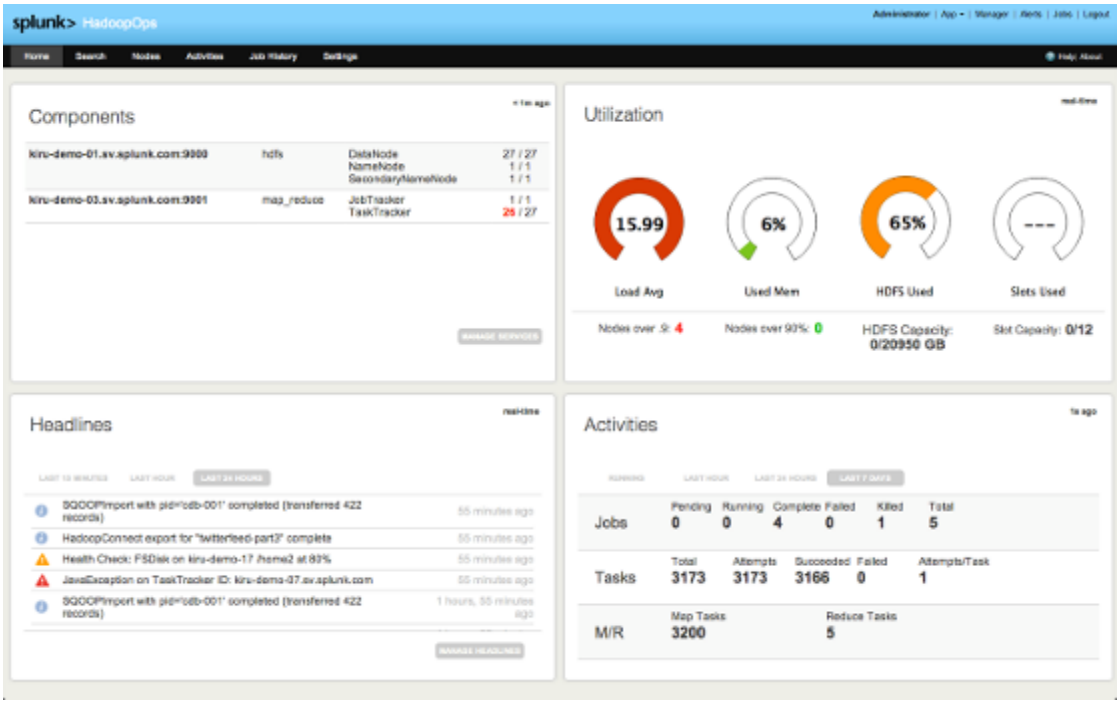


Рисунок 1.8 – Інтерфейс Splunk Enterprise

Так само, як і Adiscon LogAnalyzer, Splunk не має вбудованих засобів для виявлення причинно-наслідкових зв'язків або проведення кореляційного аналізу.

1.4.3 IBM InfoShpere BigInsights

IBM InfoSphere BigInsights [8] є платформою для розподілених обчислень на базі Apache Hadoop [9], обробка інформації на якій виконується згідно парадигмі MapReduce. За порівнянні з стандартною платформою Apache Hadoop IBM InfoSphere BigInsights включає в себе графічний веб-інтерфейс для

адміністрування кластера серверів Hadoop, роботою з розподіленою файлової системою (Hadoop Distributed Fi le System, HDFS), додатками і додатковим інструментом для проведення попередньої аналітики BigSheets. На рисунку 1.9 показаний стартовий екран веб інтерфейсу IBM InfoSphere BigInsights.

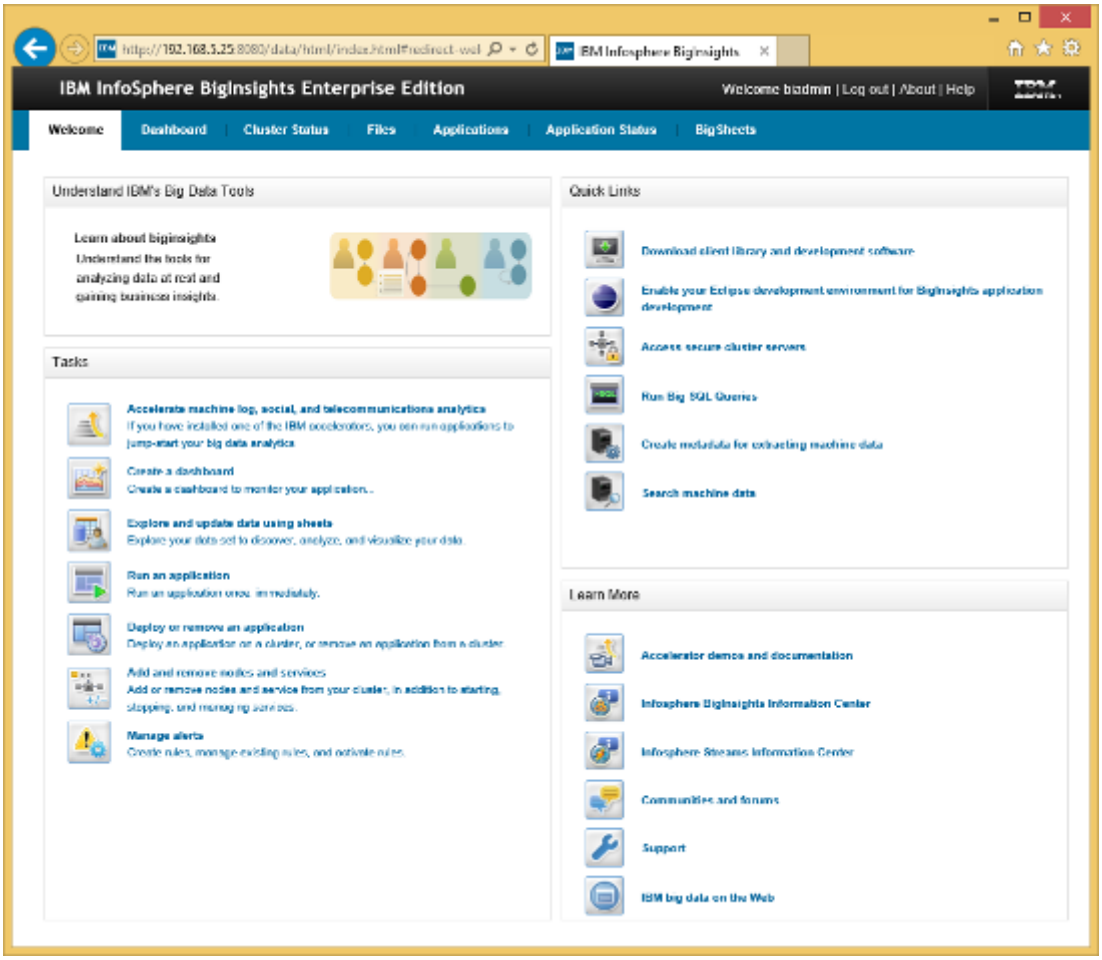


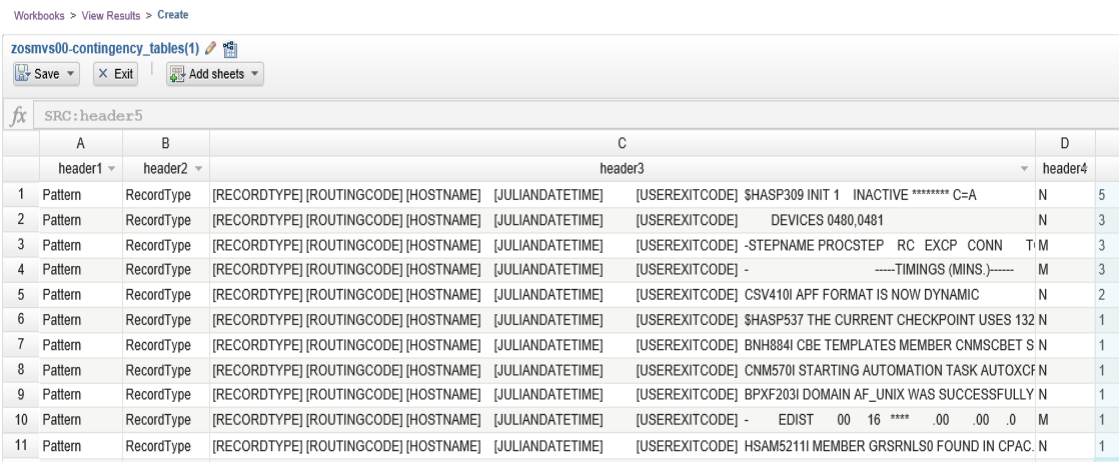
Рисунок 1.9 – Стартовий екран веб-інтерфейсу IBM InfoSphere BigInsights Enterprise Edition

В доповнення до зазначених переваг використовуваний IBM InfoSphere BigInsights Enterprise Edition поставляється з комплектом програм прискорення аналізу машинних даних (Machine Data Accelerators). Дані програми дозволяють організувати розбір завантажених в BigInsights системних журналів певних типів, будувати індекси і виконувати повнотекстовий пошук, виконувати частотний і кореляційний аналіз. У стандартну поставку IBM InfoSphere BigInsights Enterprise Edition входять разборщікі наступних типів журналів:

- Delimiter Separated Value;
- Hadoop Data Node;
- Data Power;
- Generic;
- Hadoop Jobtracker;
- Hadoop Name Node;
- Hadoop Secondary Name Node;
- Syslog;
- Hadoop Task Attempt;
- Hadoop Task Tracker;
- WebSphere Application Server;
- Apache Webaccess;

Виконання повного циклу аналітики (розбір, індексація, частотний аналіз, кореляційний аналіз) з використанням даних інструментів повинно давати певні ідеї (insights – осяяння) про функціонування системи, її справності і причини неполадок, а також в якому напрямку виконувати пошук рішень виникли проблем.

Як можна помітити, список підтримуваних " з коробки " типів журналів вельми обмежений і специфічний. Досліди з різними журналами операційної системи IBM z/OS показали, що застосування IBM BigInsights для аналізу довільних журналів досить важко, а одержувані результати не дозволяють робити будь-яких висновків про причинно-наслідкових зв'язках (Рис 1.10).



	A	B	C	D
	header1	header2	header3	header4
1	Pattern	RecordType	[RECORDTYPE] [ROUTINGCODE] [HOSTNAME] [JULIANDATETIME] [USEREXITCODE] $HASP309 INIT 1 INACTIVE ******* C=A	N 5
2	Pattern	RecordType	[RECORDTYPE] [ROUTINGCODE] [HOSTNAME] [JULIANDATETIME] [USEREXITCODE] DEVICES 0480,0481	N 3
3	Pattern	RecordType	[RECORDTYPE] [ROUTINGCODE] [HOSTNAME] [JULIANDATETIME] [USEREXITCODE] -STEPNAME PROCSTEP RC EXCP CONN T	M 3
4	Pattern	RecordType	[RECORDTYPE] [ROUTINGCODE] [HOSTNAME] [JULIANDATETIME] [USEREXITCODE] - -----TIMINGS (MINS.)-----	M 3
5	Pattern	RecordType	[RECORDTYPE] [ROUTINGCODE] [HOSTNAME] [JULIANDATETIME] [USEREXITCODE] CSV410I APF FORMAT IS NOW DYNAMIC	N 2
6	Pattern	RecordType	[RECORDTYPE] [ROUTINGCODE] [HOSTNAME] [JULIANDATETIME] [USEREXITCODE] $HASP537 THE CURRENT CHECKPOINT USES 132	N 1
7	Pattern	RecordType	[RECORDTYPE] [ROUTINGCODE] [HOSTNAME] [JULIANDATETIME] [USEREXITCODE] BNH884I CBE TEMPLATES MEMBER CNMSCBET S	N 1
8	Pattern	RecordType	[RECORDTYPE] [ROUTINGCODE] [HOSTNAME] [JULIANDATETIME] [USEREXITCODE] CNM570I STARTING AUTOMATION TASK AUTOXCF	N 1
9	Pattern	RecordType	[RECORDTYPE] [ROUTINGCODE] [HOSTNAME] [JULIANDATETIME] [USEREXITCODE] BPXF203I DOMAIN AF_UNIX WAS SUCCESSFULLY	N 1
10	Pattern	RecordType	[RECORDTYPE] [ROUTINGCODE] [HOSTNAME] [JULIANDATETIME] [USEREXITCODE] - EDIST 00 16 **** .00 .00 .0	M 1
11	Pattern	RecordType	[RECORDTYPE] [ROUTINGCODE] [HOSTNAME] [JULIANDATETIME] [USEREXITCODE] HSAM5211I MEMBER GRSRNLS0 FOUND IN CPAC	N 1

Рисунок 1.10 – Таблиця спряженості значень параметрів Pattern і RecordType.

1.4.4 Комплекс Fluentd/Logstash, ElasticSearch, Kibana

Даний комплекс відкритих програмних засобів являє собою альтернативу рішенню Splunk, розглянутому в пункті 1.4.2.

Програмне забезпечення Fluentd [10] – гнучке розширюване за рахунок плагінів рішення по збору журналів з кінцевих інформаційних систем і передачі їх на центральний сервер.

Програмне забезпечення Logstash [11] є аналогом Fluentd і так само забезпечує транспорт повідомлень від кінцевої системи до центрального сервера журналювання.

ElasticSearch [12] – програмне забезпечення індексації та повнотекстового пошуку на базі движка Apache Lucene [13].

Kibana [14] – веб-інтерфейс до ElasticSearch, що дозволяє переглядати інформацію, застосовувати до неї фільтри, виконувати пошукові запити і будувати графіки.

Загальна схема роботи комплексу показана на рисунку 11.

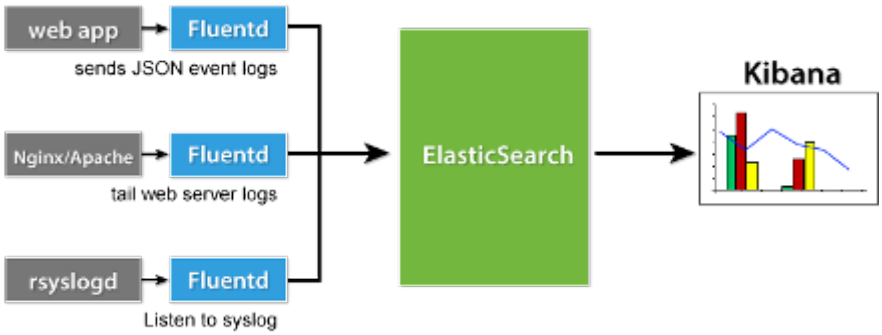


Рисунок 1.11 – Схема потоків даних комплексу Fluend/ElasticSearch/Kibana

Комплекс працює наступним чином: з допомогою агента Fluentd/Logstash забирають повідомлення про події з цільових систем і доставляють на

центральний вузол журналювання, де вони віддаються ElasticSearch для індексації. Доступ до даними проводиться через інтерфейс Kibana (рисунок 12).

Рисунок 1.12 – Інтерфейс Kibana

Так само, як Adiscon LogAnalyzer і Splunk, описаний комплекс не має вбудованих засобів для виявлення причинно-наслідкових зв'язків або проведення кореляційного аналізу.

1.5 Висновки до першого розділу

У першому розділі проведено аналіз існуючих підходів до збору інформації про події в IT-інфраструктурі, розглянуті типи систем і їх особливості. Також проведений огляд існуючих популярних рішень для аналізу зібраної інформації.

Огляд показав, що популярні рішення в основному надають кошти для пошуку інформації та побудови звітів і графіків, що відображають оперативні дані і статистику за короткий проміжок часу, вони не націлені на виявлення причинно-наслідкових зв'язків між подіями. Виняток – IBM BigInsights, однак, інструменти аналізу машинних даних виявилися громіздкі і незручні для роботи з довільними повідомленнями системних журналів.

З огляду на вищесказане, для підвищення ефективності роботи відповідальних інженерів, скорочення часу на виявлення причин виникнення подій в IT-інфраструктурі і вироблення відповідних дій було прийнято рішення власну створити інформаційно-аналітичну систему (ІАС) обробки повідомлень про події, схема роботи якої показана на рисунку 1.13

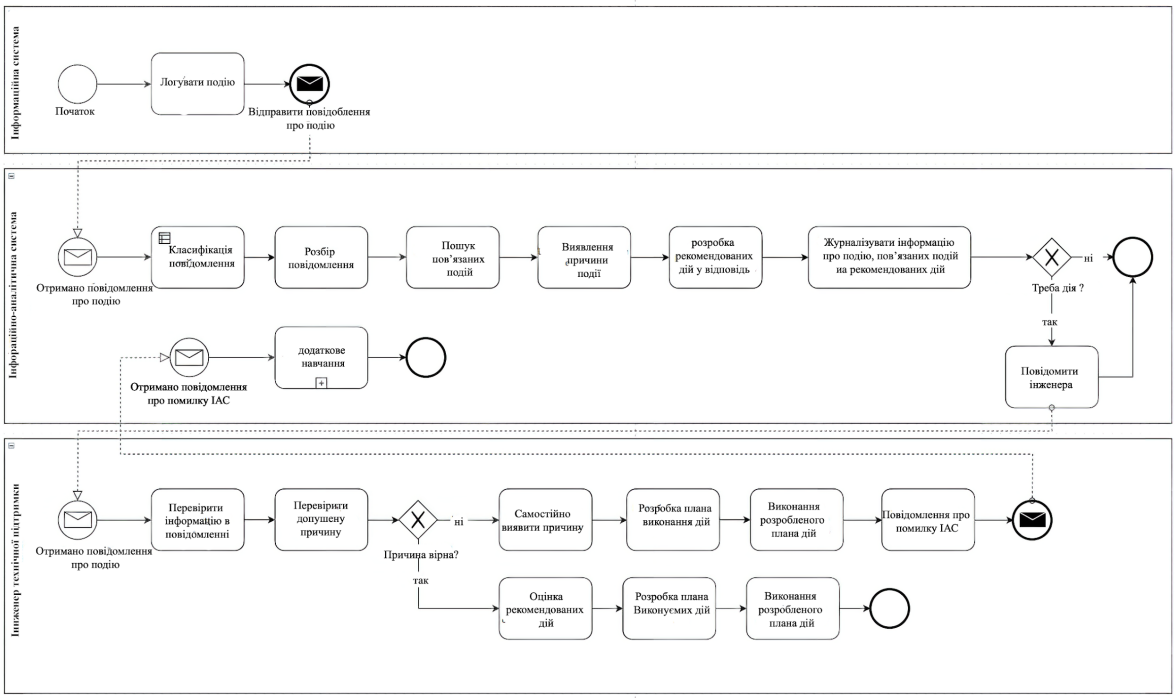


Рисунок 1.13 – Схема процесу роботи інфрмаційно-аналітичної системи обробки повідомлень про події.

2 ОСНОВНІ ПРИНЦИПИ, МОДЕЛІ І МЕТОДИ ВИЯВЛЕННЯ ПРИЧИННО-НАСЛІДКОВИХ ВІДНОСИН МІЖ ПОДІЯМИ В IT-ІНФРАСТРУКТУРІ

2.1 Архітектура централізованого логування

Побудова IT-інфраструктури з централізованим журналюванням означає створення деякого вузла, на який будуть стікатися повідомлення всіх інформаційних систем і сервісів для зберігання, обробки і доступу до них. Різні питання, пов'язані з журналювання добре розглянуті в роботі [15]. Очевидним плюсом централізованого підходу є створення " єдиної точки входу " для системних адміністраторів і аналітичних систем – вся інформація зберігається в одному місці. Так як в одиницю часу в будь-який IT-інфраструктурі відбувається величезна кількість подій, централізація тягне за собою підвищені вимоги до апаратного і програмного забезпечення центрального сховища в відношенні відмовостійкості і продуктивності, вимога постійної доступності та надійної передачі повідомлень журналів.

Також складання всіх повідомлень про події " в одну купу " з огляду на їх великого кількості і відмінностей в форматі уявлення інформації дуже швидко перетворить сервер журналювання в свого роду " смітник ", непридатний для використання людиною. Тому для мінімально прийнятної роботи інженерів з центральним сервером знадобляться кошти індексації та пошуку.

Вже було згадано, що одним з вимог до централізованого журнал є надійний протокол транспорту повідомлень. З іншого боку, цей надійний протокол повинен підтримуватися широким спектром програмного забезпечення, операційних систем і апаратних систем. Історично ж склалося так, що широко поширеним протоколом транспорту повідомлень журналів став ненадійний протокол прикладного рівня Syslog [4], тому що він працює поверх протоколу UDP для забезпечення продуктивності і швидкості передачі. Протоколи надійної передачі повідомлень журналів існують, але широкого

поширення не отримали. Тому, незважаючи на вже перераховані недоліки протоколу Syslog, а також недоліки, описані в наступних пунктах, основою централізованого журналювання в даній роботі обраний саме протокол Syslog.

При централізованому журнал є два підходи до збору інформації з кінцевих систем на центральний сервер:

- Негайна відправка повідомлень кінцевої системою;
- Періодичний збір журналів центральним сервером;

Підхід негайної відправки показаний на рисунку 2.1. Тут кінцева система – веб-сервер – відправляє повідомлення, як тільки вони були згенеровані центральному серверу, захищеному мережевим фільтром (firewall). В цьому випадку в мережевому фільтрі має бути дозвіл на проходження трафіку Syslog. З огляду на те, що систем, які відправляють повідомлення журналів багато, налаштувати вибіркові дозволу для хостів-джерел не вийде і доведеться зробити в мережевому фільтрі " дірку " – дозвіл для всього вхідного Syslog – трафіку. Однак протокол Syslog не має методу аутентифікації відправника повідомлення, що робить центральний сервер вразливим до прийому помилкових, неавторизованих повідомлень. Помилкові Syslog -повідомлення ззовні відкидаються граничним маршрутизатором або мережевим фільтром.

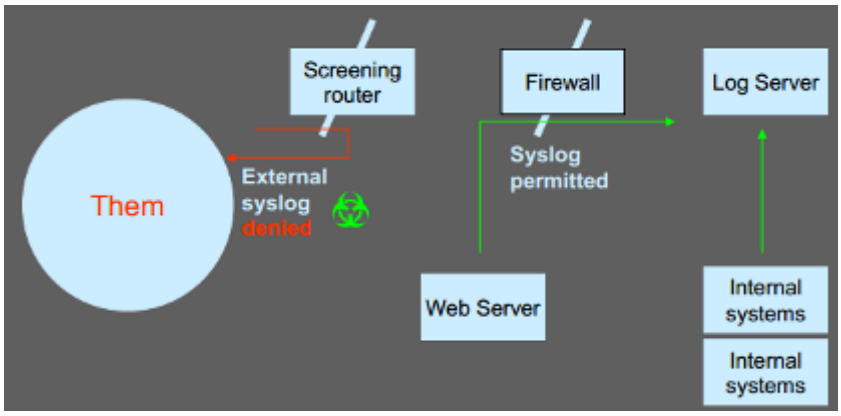


Рисунок 2.1 – Централізація шляхом негайної відправки повідомлень кінцевої системою

Другий підхід – з періодичним збором журналів – показаний на рисунку 2.2. Тут центральний сервер раз в деякий період часу по захищеному протоколу, наприклад FTP/S, SSH, SFTP, підключається до кінцевої системі і забирає накопичилися повідомлення в сховище. Так як з'єднання ініціюються сервером зсередини, то вони безперешкодно проходять через мережевий фільтр, таким чином ліквідується необхідність в потенційно небезпечному дозвільному правилі для повідомлень Syslog. Також з'являється можливість відстежувати доступність кінцевих систем. З іншого боку, даний підхід вимагає постійного оновлення конфигураційної інформації центрального – списку опитуваних кінцевих систем ; перший підхід навпаки передбачає налаштувати центральний сервер один раз, а на кінцевих системах вказувати IP-адреса сервера журналювання.

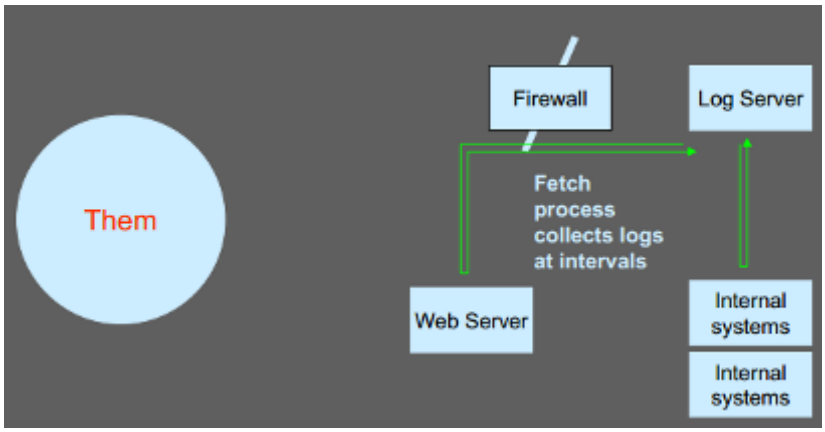


Рисунок 2.2 – Централізація шляхом періодичного збору повідомлень центральним сервером

У разі, коли IT-інфраструктура є розподіленою, а також для захисту центрального вузла від перевантажень можна організувати змішану централізацію: для деякої групи кінцевих систем створюється локальний сервер журналювання по схемі негайної відправки, а центральний сервер журналювання " стягує " накопичення повідомлення з локальних Syslog серверів. Схема централізації для змішаного підходу показана на рисунку 2.3.

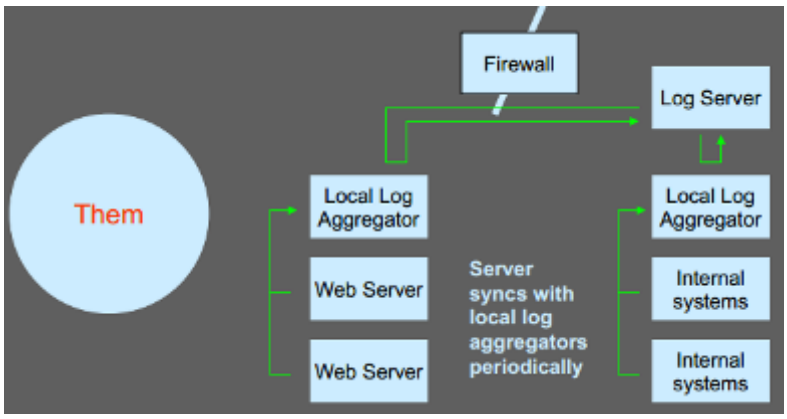


Рисунок 2.3 – Змішана схема централізації журналювання

2.2 Виявлення шаблонів повідомлень

З огляду на те, що Syslog є протоколом передачі повідомлень журналів, він не накладає будь-які обмеження на формат, синтаксис транспортуються повідомлень. Це призводить до того, що кожна система має власний формат повідомлень, і немає ніякого способу передбачити структуру переданого повідомлення, що робить вкрай важкими наступний розбір і аналіз повідомлень на центральному сервері.

Однак завдання розбору повідомлень журналів і виявлення з них важливих значень – парсинг (parsing) – вирішувати потрібно. Для того, щоб з деякою рядки виділити необхідну інформацію, потрібно знати структуру цієї рядки, тобто її шаблон. Шаблоном називається рядок, що складається з незмінних, постійних для однотипних рядків ділянок, і змінних подстрок – містозаповнювачів (placeholder). Для того, щоб розрізняти містозаповнювачі, їх зазвичай називають. В рамках даної роботи шаблони будуть мати наступний вигляд: {SophosInternalId}: client = unknown [{ipAddress}]
де { SophosInternalId }, { ipAddress } – іменовані містозаповнювачі, інша показаного частина шаблону незмінна.

Для того, щоб перевірити відповідність довільній рядки заданому шаблоном, а також обчислити значення містозаповнювачів, застосовуються регулярні вирази [16]. Таким чином для розбору рядків необхідно спочатку побудувати шаблон, а потім трансформувати його в регулярне вираз. Слід

зазначити, що в загальному випадку не можна свідомо сказати, повідомлення якого формату будуть передані на центральний сервер, тобто спочатку шаблони невідомі.

Проста і очевидна ідея описана в [17]. Кожен шаблон задає певний клас рядків, які відповідають шаблоном. При цьому у всіх рядках одного класу присутній незмінна частина відповідного шаблону. У цьому випадку справедливо стверджувати, що рядки одного класу схожі. Тоді, взявши групу схожих рядків і виділивши в них постійну частину, отримаємо шаблон для цієї групи рядків.

Для оцінки схожості рядків існує ряд широко відомих метрик, алгоритмів їх розрахунку і кодів, що реалізують ці алгоритми на різних мовах програмування. Вся ця інформація представлена відповідному розділі відкритої енциклопедії Вікіпедія [18]. З огляду на необхідність порівняння рядків різної довжини відповідними метриками виявилися відстань Левенштейна [19] і коефіцієнт перекриття (міра Шимкевіча-Сімпсона) [20]. З огляду на те, що відстань Левенштейна погано працює на довгих рядках і добре на коротких, а коефіцієнт перекриття навпаки – добре на довгих і погано на коротких, то було прийнято рішення використовувати обидві метрики: відстань Левенштейна на рядках довжиною менше 200 символів (заокруглена середня довжина повідомлення) і коефіцієнт перекриття на рядках довжиною понад 200 символів.

Для виявлення загальної частини для групи схожих рядків пропонується використовувати обчислення найбільшою загальною підпослідовності символів (longest common subsequence, LCS) [21] в рядках групи.

Таким чином, для виявлення шаблону повідомлень маємо наступний алгоритм:

- Серед отриманих на центральний сервер повідомлень формуємо групу схожих, для розрахунку " схожості " повідомлень використовуємо описані вище метрики – відстань Левенштейна і коефіцієнтфіцієнт перекриття;
- Для групи схожих рядків обчислюється LCS;

- Шляхом посимвольного порівняння LCS і рядків групи формуємо шаблон групи повідомлень, в якості містозаповнювачів використовуємо послідовність {i}, де i – порядковий номер (індекс) місцезаповнювача починаючи з 0;

Як правило шаблон, отриманий в відповідно з наведеним алгоритмом, буде мати дефекти зважаючи того, що при його генеруванні неможливо гарантовано на кроці 1 мати повну групу всіх можливих повідомлень. За цією причини згенерований шаблон повинен піддаватися перевірці людини. Крім цього тільки людина може задати імена містозаповнювачів і потрібним чином відкоригувати шаблон, так як тільки він знає семантику повідомлень.

Повторивши описану вище процедуру для різних груп рядків, отримаємо набір шаблонів, який потім використовувати для розбору Syslog -повідомлень. Цей набір шаблонів в рамках даної роботи будемо називати ядром розбору.

2.3 Машинне навчання

Після того, як сформовано ядро розбору, з вхідних повідомлень можна виділити значущі дані: IP -адреси, імена вузлів, назви додатків і інше. Крім цього сам шаблон повідомлення дозволяє судити про семантиці групи повідомлень. Ці дані разом з службовими даними протоколу Syslog (тимчасові мітки, пріоритет/важливість, призначені для користувача мітки і ін.) Формують масив даних для аналізу, метою якого є виявити причинно-наслідкові відносини між зафіксованими подіями.

Питання визначення причинності йде своїми коренями в античну філософію і до сих пір є актуальним. Для даної роботи істотним є його наступні аспекти:

- Причина завжди передує слідству. Це означає, що пошуки причини конкретного події слід шукати лише в попередніх подіях;
- Причиною може бути як наявність, так і відсутність одного або ряду подій;

- Ставлення причинності регулярно і об'єктивно. Це означає, що якщо подія А викликало подія В один раз, то повторення події А в тих же умовах має знову викликати подія В. Таким чином кореляція подій А і В дозволяє припустити, але не гарантує наявність між ними причинно-наслідкового зв'язку;
- Події можуть утворювати ланцюжки, де одне подія стало причиною іншого події. Такі ланцюжки дозволяють будувати моделі процесів, що протікають в IT-інфраструктурі, і можуть послужити основою для модернізації окремих систем або всіх інфраструктури;
- Частина подій не буде мати залогованої події-причини, так як породжені зовнішніми впливами – непідконтрольними системами або людиною;
- У світі інформаційних технологій все будується на основі домовленостей людей і загальних, поділюваних моделях і уявленнях. Це означає, що на основі тільки інформації про події можна буде показати, що " система А працює саме так ", але неможливо відповісти на питання " чому ";

В якості основного тут виступають пункти 1 і 3, тобто в рамках роботи пропонується виявляти регулярні зв'язку між подіями і однотипні регулярні послідовності подій. Відповідний механізм пропонує розділ машинного навчання без вчителя " Пошук асоціативних правил " [22]. Завдання пошуку асоціативних правил (association rules learning) формулюється таким чином: вихідні дані представляються в вигляді ознакових описів. Потрібно знайти такі набори ознак, і такі значення цих ознак, які особливо часто (не випадково часто) зустрічаються в ознакових описах об'єктів. Набір правил, отриманий по доступним аналітичним моделям, буде використаний в побудові ланцюжків подій і виявленні подій-причин.

Очевидно, що одних статистично підтверджених зв'язків недостатньо – потрібна база експертних знань в області побудови інформаційних систем і їх взаємодії. Основи для такої бази знань в вигляді онтології закладені в роботі [23]. Однак створення цієї бази знань – окремий великий проєкт, тому не включений в дану роботу.

2.4 Висновки до другого розділу

У другому розділі описані основні ідеї, концепції і методи, покладені в основу розробленої системи моніторингу виявлення причинно-наслідкових зв'язків між подіями.

Було розглянуто централізоване журнал, його особливості вимоги, переваги і недоліки впровадження. Також проведений аналіз можливих архітектур реалізації централізованого журналювання. В якості транспортного протоколу для повідомлень журналів обраний протокол Syslog зважаючи величезного кількості систем, що підтримують саме цей протокол.

Було дано визначення поняття " шаблон " рядки і описаний алгоритм по його генерування, а також ряд супутніх рекомендацій.

Коротко розглянуті особливості відносини причинності. В якості механізму виявлення причинно-наслідкових зв'язків обраний підхід на основі пошуку асоціативних правил.

Одним з подальших напрямків розвитку даної роботи позначена розробка власної моделі асоціативних правил з урахуванням експертних знань про устрій і взаємодії інформаційних систем та їх компонент.

3 РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ ТА ВИЯВЛЕННЯ ПРИЧИННО-НАСЛІДКОВИХ ЗВ'ЯЗКІВ МІЖ ПОДІЯМИ

3.1 Архітектура системи моніторингу

Як вже раніше було сказано в пункті 1.2 для відстеження існуючих інформаційних систем і сервісів в організаціях виконують розгортання системи моніторингу. Схематично потоки інформації для цього випадку показані на рисунку 3.1.

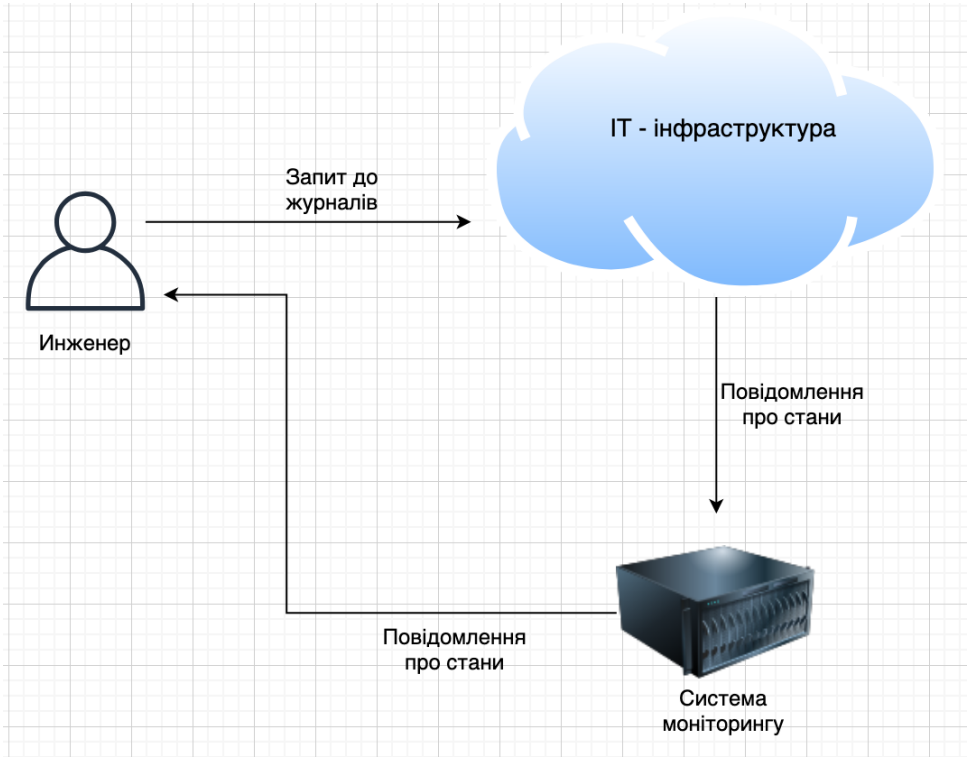


Рисунок 3.1 – Моніторинг і аналіз журналів без IAC

Для підвищення ефективності роботи відповідальних інженерів, скорочення часу на виявлення причин виникнення подій в IT-інфраструктурі і вироблення відповідних дій була створена інформаційно-аналітичну систему (IAC) виявлення причинно-наслідкових відносин між подіями.

Дана система передбачає роботу спільно з існуючими системами моніторингу: моніторинг відслідковує стан, а IAC виявляє ланцюжка подій,

пов'язаних ставленням причинності. Загальна схема роботи системи показана на рисунку 3.2.

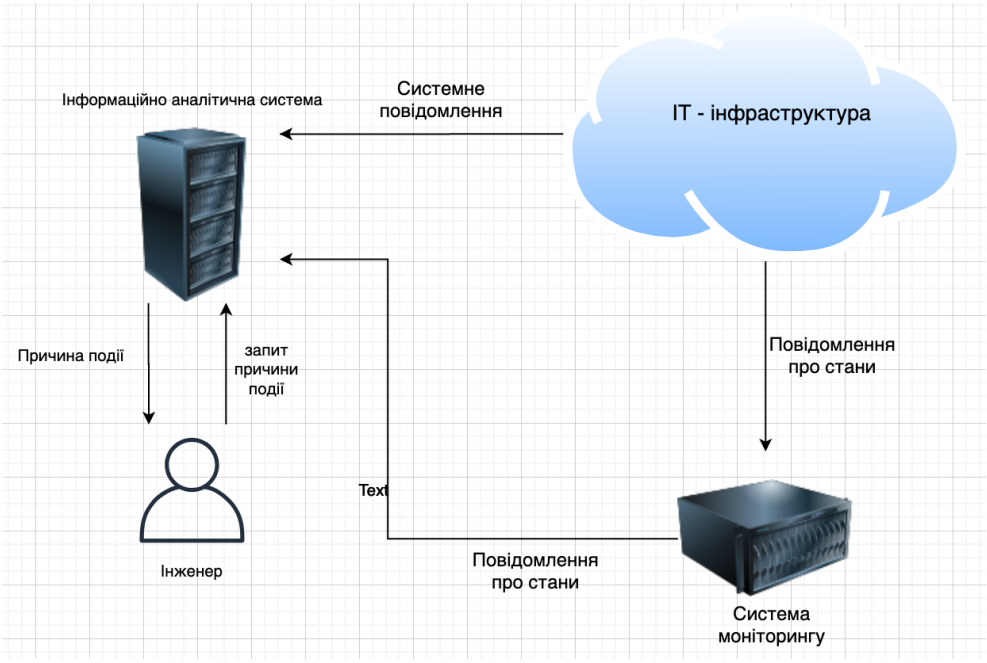


Рисунок 3.2 – Загальна схема роботи системи моніторингу

Як слід з схеми роботи IAC володіє відомостями про події під всієї IT-інфраструктури і навчається по мірі роботи. Навчена система використовується в роботі інженерів технічної підтримки.

У своїй роботі IAC спирається на три основних компонента: централізоване журналювання подій, якого навчають парсер і аналітична модель. Архітектура IAC і використовувані протоколи зображені на рисунку 3.3.

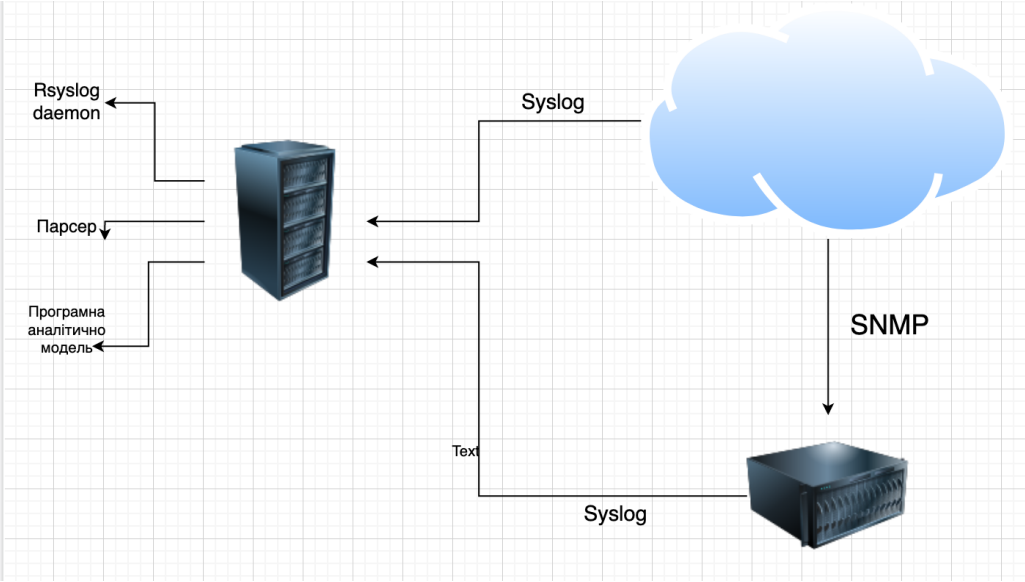


Рисунок 3.3 – Архітектура IAC і використовувані протоколи

Як показано на малюнку 19 IAC виступає в якості сервера журналювання в централізованій архітектурі журналювання, виконаної в відповідно з підходом " негайної відправки " (розділ 2.1). Крім цього стандарт RFC 5675 [24] визначає правила трансляції повідомлень SNMP в повідомлення Syslog дозволяючи транслювати інформацію від систем моніторингу в IAC, ніж доповнює інформацію про події ще й інформацією про стан параметрів спостережуваних систем.

Після того як інформація про події оброблена сервером Rsyslog, вона стає доступна тому, якого навчають парсеру, заснованому на принципах, викладених в розділі 2.2 В результаті розбору повідомлень формується масив даних, який передається на вхід моделям пошуку асоціативних правил, як описано в підрозділі 2.3.

3.2 Технології та засоби реалізації системи моніторингу

3.2.1 Rsyslog

Програмне забезпечення Rsyslog [25] – популярний Syslog сервер, який є стандартним демоном журналювання в операційних системах Unix та Linux. Своє широке поширення отримав за рахунок високої швидкості роботи – по

затвердженню розробників " Rocket – fast syslog server " – великим кількістю модулів отримання повідомлень журналів (input module, im, наприклад imtcp – отримання стандартних syslog -повідомлень по протоколу TCP) і модулів виводу інформації (output module, om, наприклад omelasticsearch – передача оброблених повідомлень сервера ElasticSearch). Крім цього Rsyslog може бути розширено самостійно розробленими фільтрами і парсер заздалегідь відомих повідомлень.

3.2.2 Filebeat

Filebeat призначений для збору даних їх log-файлів, зібрані події відправляються в Logstash або безпосередньо в Elasticsearch для індексації.

Створювався з упором на надійність – якщо збір логів переривається, система запам'ятовує, в якому місці це сталося і продовжує з запомненной позиції, коли знову в мережі.

Простота використання – в комплекті Filebeat поставляються вбудовані модулі (auditd, Apache, NGINX, System, MySQL, і ін.), Які спрощують збір, аналіз і візуалізацію популярних форматів журналів до однієї команди. Досягається це завдяки комбінуванню автоматичного визначення шляхів, грунтуючись на встановленій ОС, визначень Ingest-нод і панелі Kibana.

Можна використовувати всередині контейнера – наприклад, розгорнути Filebeat всередині Docker-контейнера і збирати дані з інших контейнерів, які виконуються на тому ж хості.

Вбудований захист від перевантажень – при відправці даних в Elasticsearch використовується протокол, чутливий до навантаження одержувача: наприклад, якщо Logstash зайнятий обробкою даних, він сповіщає про це Filebeat, в результаті, Filebeat знижує швидкість читання. Як тільки перевантаження усувається, Filebeat продовжує роботу в первісному темпі.

Filebeat є частиною стека Elastic, що означає, що він без проблем працює з Logstash, Elasticsearch і Kibana.

3.2.3 Metricbeat

Metricbeat входить в групу програмного забезпечення Beats, яка допомагає передавати різні типи даних на сервер Elastic Stack. Metricbeat – це легкий відправник даних, який після установки на сервер періодично збирає статистику CPU і пам'яті (глобально і по окремим процесам), а потім передає ці дані в Elasticsearch. Цей відправник може замінити Topbeat в версії 5.0 стека Elastic Stack.

3.3 Компоненти системи моніторингу

3.3.3 Налаштування клієнту Rsyslog

Для того, щоб готувати повідомлення журналів до подальшої обробки стандартна конфігурація сервера Rsyslog була змінена таким чином:

1. Активований прийом повідомлень з файлу syslog. Для цього в розділ " MODULES " внесені/розблоковані наступні рядки.

```
$ModLoad imuxsock
$ModLoad imklog
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
$RepeatedMsgReduction on
$FileOwner syslog
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
$PrivDropToUser syslog
$PrivDropToGroup syslog
$SystemLogUsePIDFromSystem on
$WorkDirectory /var/spool/rsyslog
$IncludeConfig /etc/rsyslog.d/*.conf
```

Рисунок 3.1 – Приклад файлу rsyslog.conf

2. Створений шаблон для перетворення файлів формату syslog в json де кожному елементу в syslog присвоюється відповідне ім’я поля в json докумені.

```
module(load="omelasticsearch") # Elasticsearch output module      You, 3 years ago • Add deployment omelasticsear

set $!es_record_id = $uuid;
template(name="bulkid-template" type="list") { property(name="$!es_record_id") }

template(name="json-template"
  type="list") {
    constant(value="{")
      constant(value="\"@timestamp\":\"")          property(name="timereported" dateFormat="rfc3339")
      constant(value="\",\"host.name\":\"")        property(name="hostname")
      constant(value="\",\"message\":\"")          property(name="msg" format="json")
      constant(value="\",\"process.pid\":\"")      property(name="procid")
      constant(value="\",\"process.name\":\"")     property(name="programname")
      constant(value="\",\"log.syslog.severity.name\":\"") property(name="syslogseverity-text")
      constant(value="\",\"log.syslog.facility.name\":\"") property(name="syslogfacility-text")
    constant(value="\"}\n")
}

```

Рисунок 3.2 – Приклад файлу 05-elastic-json.conf

3. Та для кожного сервісу що пише в syslog строрити файл перенаправлення в rsyslog. Як показано на Рис 3.4 для захисту підключення використовується tls шифрування та пароль. Також вказується розмір черги з повідомлень та шаблон повідомлення який вказано вище.

```
:programname,contains,"dockerd" action(type="omelasticsearch"
  server="elastic.server.com"
  serverport="9200"
  usehttps="on"      You, 4 months ago • Add tls to rsyslog
  tls.cacert="/var/www/rsyslog/certs/company-ca.pem"
  tls.mycert="/var/www/rsyslog/certs/{{ company_hostname }}/{{ company_hostname }}.crt"
  tls.myprivkey="/var/www/rsyslog/certs/{{ company_hostname }}/{{ company_hostname }}.key"
  uid="rsyslog"
  pwd="rsyslog"
  template="json-template"  # use the template defined earlier
  searchIndex="rsyslog"
  dynSearchIndex="off"
  searchType="_doc"
  bulkmode="on"                 # use the Bulk API
  bulkid="bulkid-template"      # use buildid-template
  dynbulkid="on"                # use dynamic id template
  writeoperation="create"       # use create API instead of index
  queue.dequeuebatchsize="5000"   # ES bulk size
  queue.type="LinkedList"       # use in memory linked list queue
  queue.size="100000"           # capacity of the action queue
  queue.workerthreads="5"       # 5 workers for the action
  action.resumeretrycount="10"
  errorfile="/var/log/omelastic.log"
)
& stop
```

Рисунок 3.3 – Приклад файлу 10-forward-docker.conf

3.3.4 Налаштування клієнту Filebeat

Filebeat альтернатива для rsyslog але на відміну від rsyslog його легше налаштовувати і в нашій системі він буде потрібний для логування кастомних

повідомлень які вже надходять з сервісу в json форматі. наприклад з самописного веб проксі чи веб сайту. Приклад конфігурації filebeat можна побачити на Рис 3.4. Підключення як і в rsyslog захищене tsl шифруванням та паролем.

```
filebeat.inputs:

- type: log
  paths: ["/logs/proxy/stats.log"]
  json.add_error_key: true
  json.keys_under_root: true
  json.overwrite_keys: true
  json.add_error_key: true
  fields_under_root: true
  fields:
    type: stats

- type: log
  paths: ["/logs/proxy/acct.log"]
  json.add_error_key: true
  json.keys_under_root: true
  json.overwrite_keys: true
  json.add_error_key: true
  fields_under_root: true
  fields:
    type: acct
    hostname: {{ company_hostname }}

processors:
 - drop_fields:
     fields: [ "log", "input", "host", "agent", "ecs" ]

output.elasticsearch:
  hosts: ["elasticNode1.company.com", "elasticNode2.company.com"]
  ssl.certificate_authorities: [ "/certs/company-ca.pem" ]
  ssl.certificate: "/certs/node/company.crt"
  ssl.key: "/certs/node/company.key"
  protocol: "https"
  username: "filebeat"
  password: "filebeat"
  compression_level: 9
  indices:
    - index: "acct"
      when.contains:
        type: acct
    - index: "stats"
      when.contains:
        type: stats
      Vitaly Repin, 3 years ago • Filebeat was added

logging.level: warning
```

Рисунок 3.4 – Приклад файлу filebeat.yml

Основні місця конфігу:

- filebeat.input – місце де треба вказати з яких файлів буде вичитувати записи;
- preprocessor – дія яка відбувається перед тим як відправити лог в базу даних;
- output.elasticsearch – місце де треба вказати параметри підключення до бази даних та параметри розподілення записів по індексам;

3.3.5 Налаштування клієнту Metricbeat

Сама проста ланка в системі моніторингу це Metricbeat. Він потрібен для зчитування параметрів завантаження серверу такі як CPU, GPU, ROM, RAM та інші. Ці параметри в подальшому будуть брати участь в пошуку аномалій.

Конфігурація дуже проста. в секції metricbeat.modules треба вказати які параметри ми хочемо отримувати та частоту їх оновлення як показано на рисунку 3.5.

```
metricbeat.modules:
        You, 4 months ago • Add metricbeat role
- module: system
  metricsets: [ "cpu", "core", "diskio" , "filesystem", "fsstat", "load", "memory", "network", "uptime" ]
  enabled: true
  period: 60s
  interfaces: [ "br0" ]

  cpu.metrics:  [ "percentages" ]
  core.metrics: [ "percentages" ]

name: server1.company.com

output.elasticsearch:
  hosts: ["elasticNode1.company.com", "elasticNode2.company.com"]
  ssl.certificate_authorities: [ "/certs/company-ca.pem" ]
  ssl.certificate: "/certs/node/company.crt"
  ssl.key: "/certs/node/company.key"
  protocol: "https"
  username: "filebeat"
  password: "filebeat"

setup.dashboards.enabled: false

setup.template:
  name: "metricbeat-%{[agent.version]}"
  pattern: "metricbeat-%{[agent.version]}-*"

setup.ilm:
  enabled: auto
  rollover_alias: "metricbeat-%{[agent.version]}"
  pattern: "{now/d}-000001"
  check_exists: true
  overwrite: false

setupe.template.settings:
  index.number_of_shards: 2
  index.codec: best_compression

logging:
  level: warning
  to_stderr: true
  to_files: false
  to_syslog: false
```

Рисунок 3.5 – Приклад файлу metricbeat.yml

3.3.6 Налаштування кластеру Elasticsearch на Kibana

Налаштування кластеру elasticsearch проходить в 2 етапи.

Перший етап це конфігурація серверу та кластеру де будуть налаштоване шифрування та стандартні параметри доступу до системи також буде вказано з яким серверами треба утворити кластер.

На другому етапі потрібно через curl або графічний інтерфейс кібана або схожі до curl утиліти створити індекси, користувачів, надати їм відповідні права та налаштувати ротацію даних в базі даних. Простими словами ILM policy.

Конфігурація elasticsearch як і будь якого продукту не дуже складна та показана Рис 3.6. Основні моменти на які треба звернути увагу це параметр це discovery.seed_hosts. Він вказує серверу з якими хостами утворювати кластер для розподіленого зберігання даних.

```
---
cluster.name: cluster.company.com

cluster.initial_master_nodes:
  - masterNode.company.com

node.name: node1.company.com

network.host: 0.0.0.0
network.publish_host: 62.80.177.210


http.port: 9200

discovery.seed_hosts: [" node1.company.com", "node2.company.com", " node3.company.com"]

transport.tcp.port: 9300


transport.host: 0.0.0.0
transport.publish_host: 62.80.177.210

action.auto_create_index: .security-*,.monitoring-*,.watches,.triggered_watches,.watcher-history-*,metricbeat-*,rsyslog,acct,stats,


xpack.license.self_generated.type: basic
xpack.security.enabled: true
xpack.security.http.ssl.enabled: true
xpack.security.http.ssl.key: /usr/share/elasticsearch/config/certs/hostname/hostname.key
xpack.security.http.ssl.certificate_authorities: /usr/share/elasticsearch/config/certs/company-ca.pem
xpack.security.http.ssl.certificate: /usr/share/elasticsearch/config/certs/hostname/hostname.crt
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.certificate_authorities: /usr/share/elasticsearch/config/certs/company-ca.pem
xpack.security.transport.ssl.certificate: /usr/share/elasticsearch/config/certs/hostname/hostname.crt
xpack.security.transport.ssl.key: /usr/share/elasticsearch/config/certs/hostname/hostname.key

xpack.monitoring.collection.enabled: true
```

Рисунок 3.6 – Приклад файлу elasticsearch.yml

Для конфігурації подальшої конфігурації через curl потрібно використовувати команди подібні до тих які представлені на Рис 3.7 але це дуже довгий метод бо потребує ввести більше 20 таких команд та вимагає додаткових умінь з використання командного рядку які не притаманні для користувачів Windows/Mac тому розглянемо конфігурацію kibana, яку зображено на рисунку 3.8 для запуску графічного інтерфейсу elasticsearch.

```
curl -X PUT "localhost:9200/test-000001?pretty" -H 'Content-Type: application/json' -d'
{
  "aliases": {
    "test-alias":{
      "is_write_index": true
    }
  }
}
'

curl -X PUT "localhost:9200/my-index-000001/_mapping?pretty" -H 'Content-Type: application/json' -d'
{
  "properties": {
    "employee-id": {
      "type": "keyword",
      "index": false
    }
  }
}
'
```

Рисунок 3.7 – Приклад команди для конфігурації elastic через curl

```
server.port: 5601

server.host: "0"

elasticsearch.hosts: [" node1.company.com", "node2.company.com", " node3.company.com"]

kibana.defaultAppId: "discover"
      You, 6 months ago • Change configs for elastic and kibana
elasticsearch.username: "kibana"
elasticsearch.password: "kibana"

elasticsearch.ssl.certificate: "/certs/hostname/hostname.crt"
elasticsearch.ssl.key: "/certs/hostname/hostname.key"
elasticsearch.ssl.certificateAuthorities: [ "/certs/company-ca.pem" ]
elasticsearch.ssl.alwaysPresentCertificate: true

xpack.reporting.enabled: false
xpack.infra.enabled: true
xpack.infra.sources.default.logAlias: "rsyslog"
xpack.infra.sources.default.metricAlias: "metricbeat-*"
xpack.infra.sources.default.fields.host: "host.name"
xpack.infra.sources.default.fields.message: [ "messages" ]
xpack.ingestManager.enabled: false
xpack.ingestManager.fleet.enabled: false
xpack.security.encryptionKey: "{{ company_kibana_encryption_key }}"
xpack.security.session.idleTimeout: "1h"
xpack.security.session.lifespan: "30d"
```

Рисунок 3.8 – Приклад файлу kibana.yml

3.3.7 ILM: керування життєвим циклом індексу

Ви можете налаштувати політики управління життєвим циклом індексу (ILM) для автоматичного управління індексами відповідно до ваших вимог до продуктивності, стійкості та збереження. Наприклад, ви можете використовувати ILM для:

- Створювання ного індексу, коли індекс досягає певного розміру або кількості документів
- Створення нового індексу кожного дня, тижня чи місяця та архівації попередніх
- Видалення застарілих індексів для забезпечення стандартів збереження даних

Ви можете створювати та керувати політиками життєвого циклу індексу за допомогою Kibana Management або API ILM. Коли ви вмикаєте управління життєвим циклом індексу для Beats або вихідного плагіна Logstash Elasticsearch, політики за замовчуванням налаштовуються автоматично.

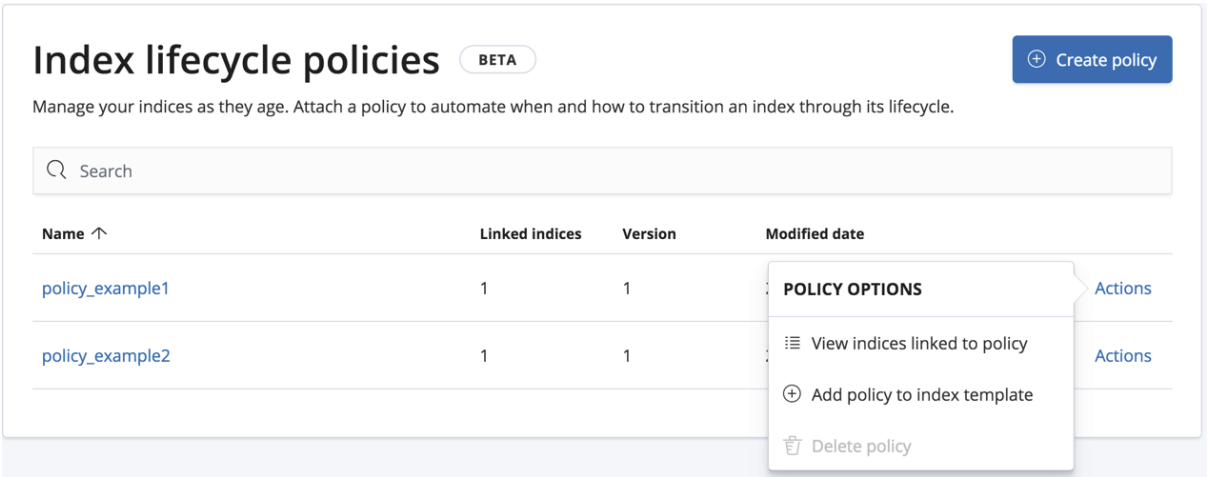


Рисунок 3.9 – Керування політиками життєвим циклом індексу

3.3.8 Управління індексами

Керування індексом дозволяє переглядати налаштування індексу, зіставлення та статистику та виконувати операції на рівні індексу. Сюди входять оновлення, змивання, очищення кеш-пам’яті, примусове злиття сегментів,

індекси заморожування тощо. Практичне управління індексом допомагає забезпечити збереження ваших даних найбільш економічно ефективним способом.

Управління індексами також допомагає створювати шаблони індексів. Шаблон зменшує обсяг бухгалтерії при роботі з індексами. Замість того, щоб вручну налаштовувати свої індекси, ви можете автоматично створювати їх із шаблону, забезпечуючи постійне визначення ваших налаштувань, зіставлення та псевдонімів.

Щоб керувати своїми індексами, відкрийте меню, а потім перейдіть до Stack Management, далі Data, далі Management Index.

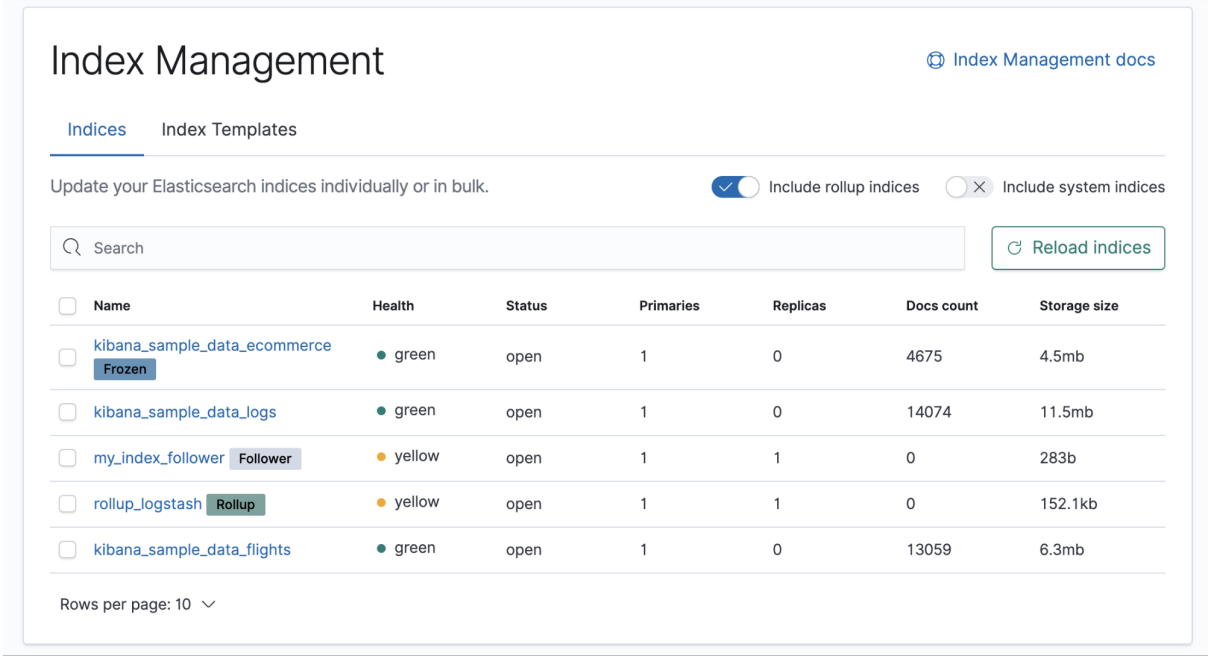


Рисунок 3.10 – Інтерфейс управління індексом

Якщо безпеку увімкнено, ви повинні мати привілеї кластера монітора та view_index_metadata та керувати привілеями індексу для перегляду даних. Для індексних шаблонів ви повинні мати привілеї кластера manage_index_templates. Докладнішу інформацію див. У розділі Привілеї безпеки.

Перш ніж використовувати цю функцію, ви повинні бути ознайомлені з операціями управління індексом. Зверніться до API керування індексом та API шаблону індексу.

Коли ви відкриваєте керування індексами, ви отримуєте огляд налаштованих індексів. Значки вказують, чи індекс заморожений, індекс послідовника чи індекс зведення.

Натискання значка звужує список лише до індексів цього типу. Ви також можете відфільтрувати показники за допомогою рядка пошуку.

Ви можете детально розглянути кожен індекс, щоб дослідити параметри індексу, відображення та статистику. У цьому вікні ви також можете редагувати параметри індексу.

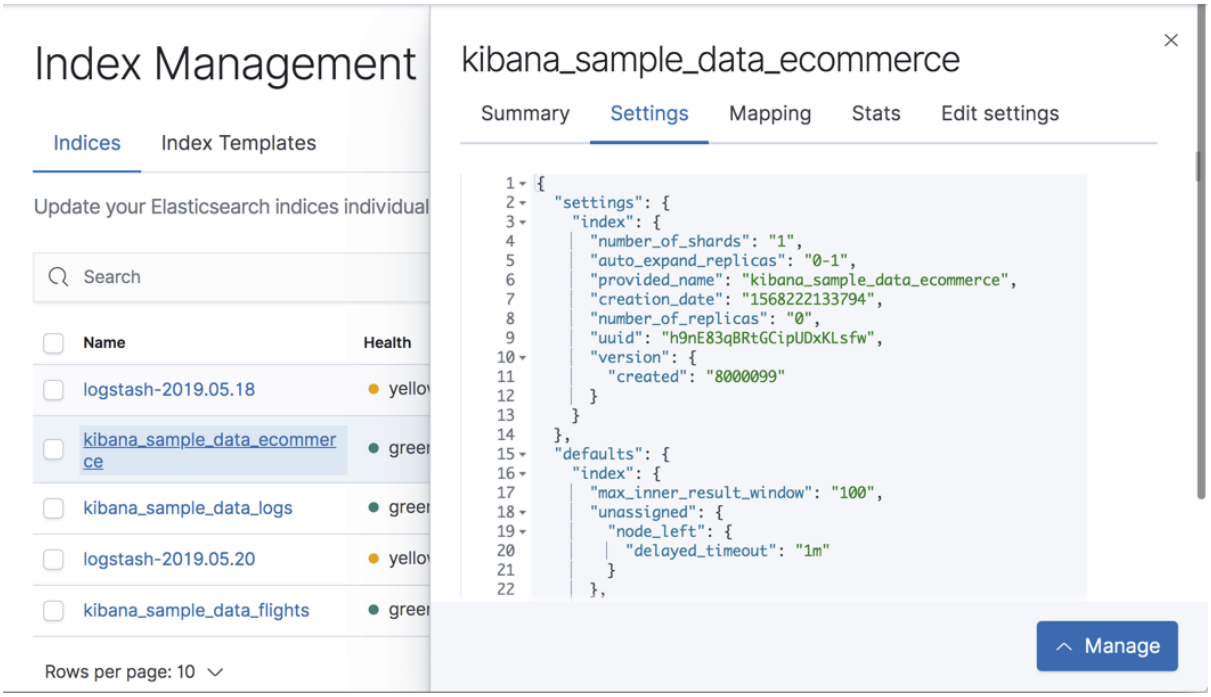


Рисунок 3.11 – Інтерфейс управління індексом

За допомогою меню Керування виконуйте операції на рівні індексу. Це меню доступне у вікні деталізації індексу або коли ви встановите прапорець біля одного або кількох індексів на сторінці огляду. Меню включає такі дії:

- Закрити індекс. Блокує індекс від операцій читання/запису. Закритий індекс існує в кластері, але не споживає інших ресурсів, крім дискового

простору. Якщо ви знову відкриваєте закритий індекс, він проходить звичайний процес відновлення;

- Індекс примусового злиття. Зменшує кількість сегментів у вашому фрагменті шляхом об’єднання менших файлів та очищення видалених. Лише примусове злиття індексу лише для читання;
- Індекс оновлення. Записує операції в буфері індексування в кеш файлової системи. Ця дія виконується автоматично раз на секунду. Примусове ручне оновлення корисно під час тестування, але не повинно регулярно виконуватися у виробництві, оскільки це впливає на продуктивність;
- Очистити кеш-пам’ять. Очищає всі кеші, пов’язані з індексом;
- Індекс змиву. Звільняє пам’ять, синхронізуючи кеш файлової системи на диск і очищаючи кеш. Після завершення синхронізації внутрішній журнал транзакцій скидається;
- Індекс замерзання. Робить індекс лише для читання та зменшує розмір пам'яті, переміщуючи осколки на диск. Заморожені індекси залишаються доступними для пошуку, але запити займають більше часу;
- Видалити індекс. Постійно видаляє покажчик та всі його документи;
- Додайте політику життєвого циклу. Визначає політику управління життєвим циклом індексу;

Шаблон індексу визначає параметри, зіставлення та псевдоніми, які ви можете автоматично застосувати під час створення нового індексу. Elasticsearch застосовує шаблон до нового індексу на основі шаблону індексу, який відповідає імені індексу.

У поданні Шаблони індексів перелічені Ваші шаблони та дозволяють перевіряти, редагувати, клонувати та видаляти їх. Зміни, внесені в шаблон індексу, не впливають на існуючі індекси.

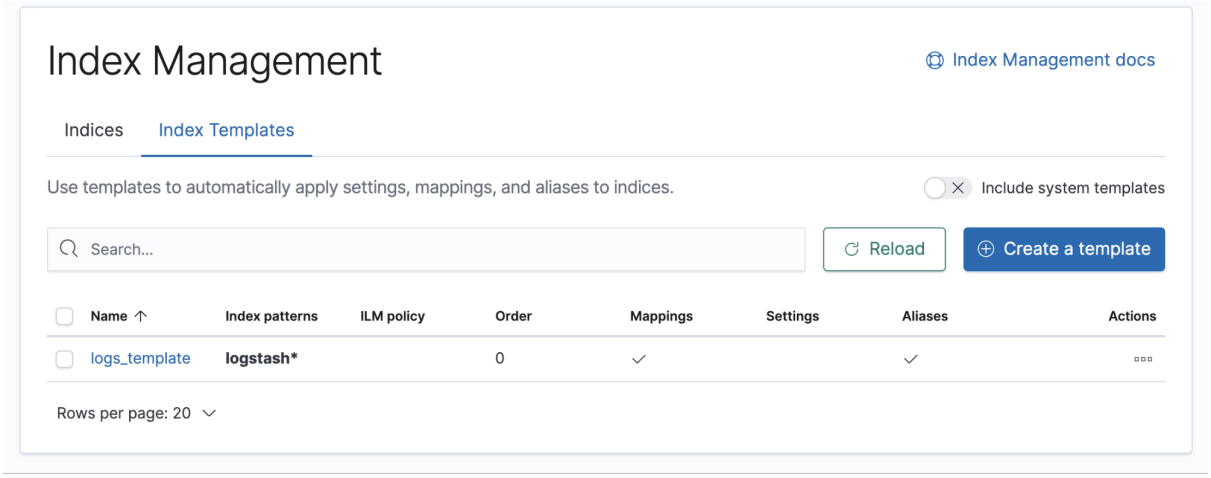


Рисунок 3.12 – Шаблони покажчиків

Якщо у вас немає шаблонів, ви можете створити його за допомогою майстра створення шаблону. Шаблони індексів застосовуються під час створення індексу, тому перед створенням індексів потрібно створити шаблон.

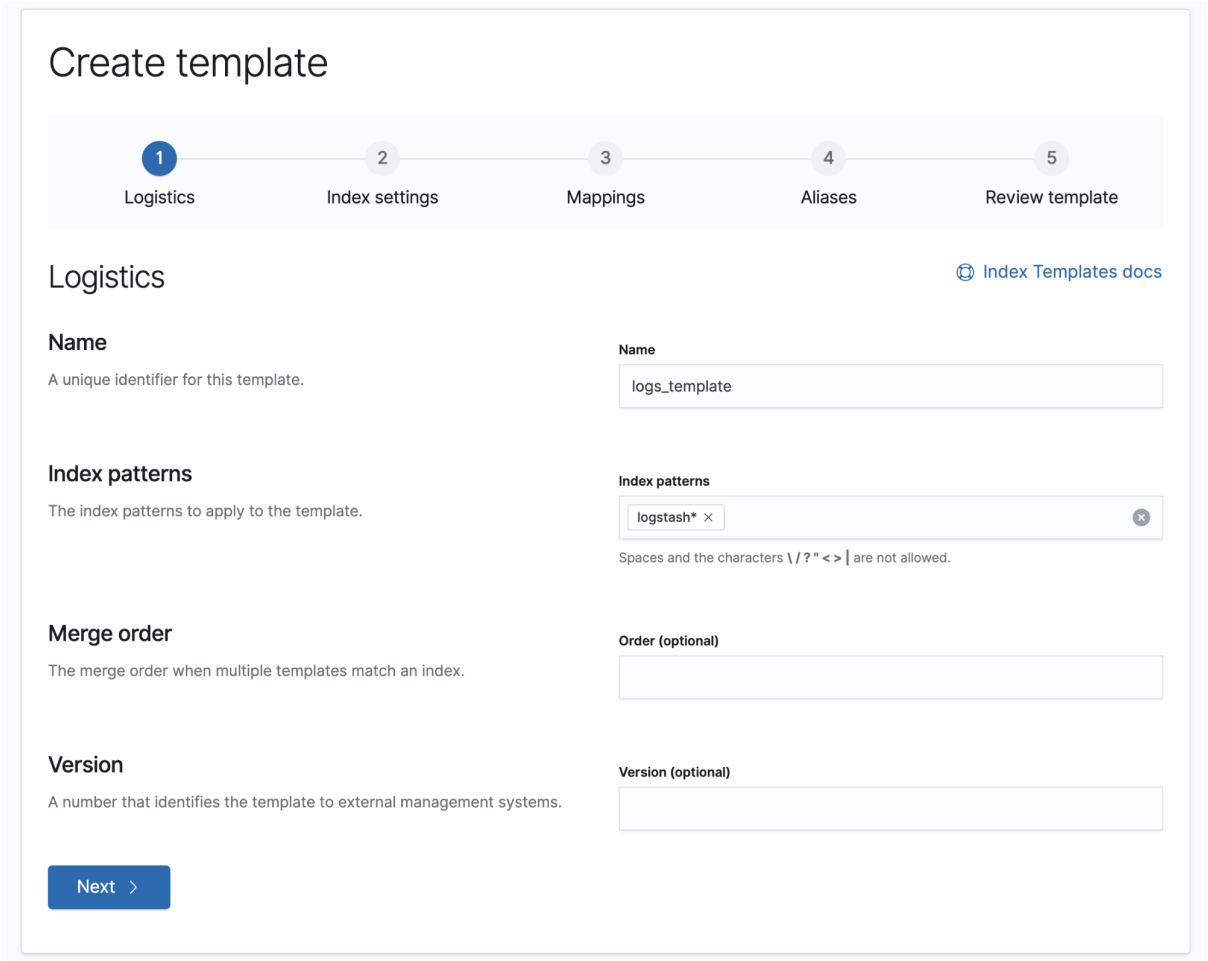


Рисунок 3.13 – Приклад: Створення шаблону індексу

Відкрийте майстер створення шаблону та введіть logs_template у поле Ім'я. Встановіть шаблон індексу як logstash *, щоб шаблон відповідав будь-якому індексу з цим шаблоном індексу. Порядок об’єднання та версія є не обов’язковими, і ви залишите їх пустими в цьому прикладі.

Другий крок у майстрі створення шаблону дозволяє визначити параметри індексу. Ці налаштування є не обов’язковими, і цей приклад пропускає цей крок.

Набір даних журналів вимагає відображення, щоб позначити пари широти та довготи як географічні розташування, застосовуючи тип geo_point. На третьому кроці майстра визначте це відображення на вкладці "Позначені поля" таким чином:

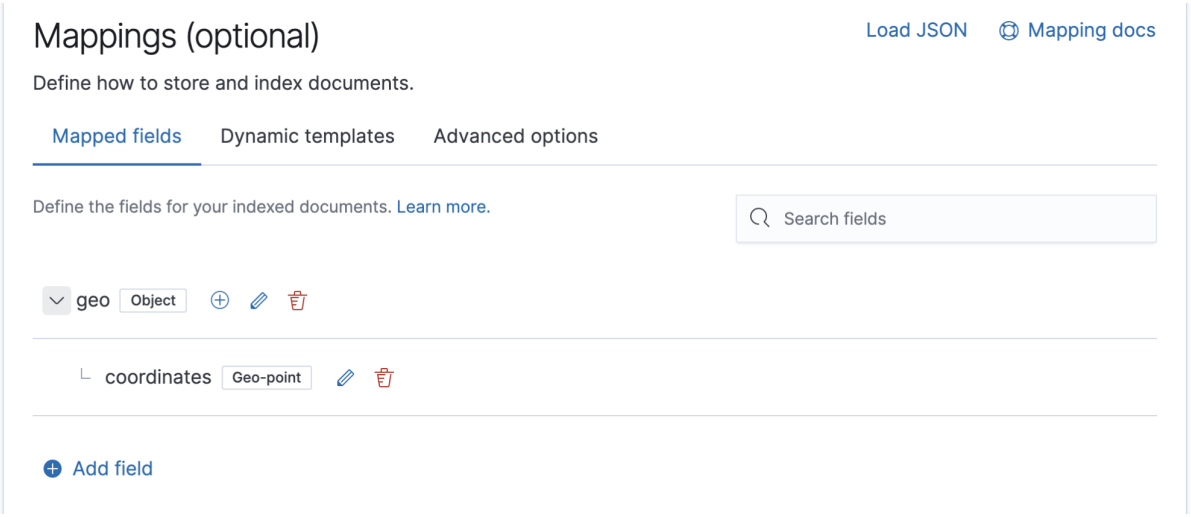


Рисунок 3.13 – Сторінка відображених полів

3.3.9 Створення завдань виявлення аномалій

Завдання виявлення аномалій містять інформацію про конфігурацію та метадані, необхідні для виконання аналітичного завдання.

Ви можете створити завдання виявлення аномалій, використовуючи API створення завдань виявлення аномалій. Також Kibana пропонує наступні майстри для спрощення створення робочих місць:

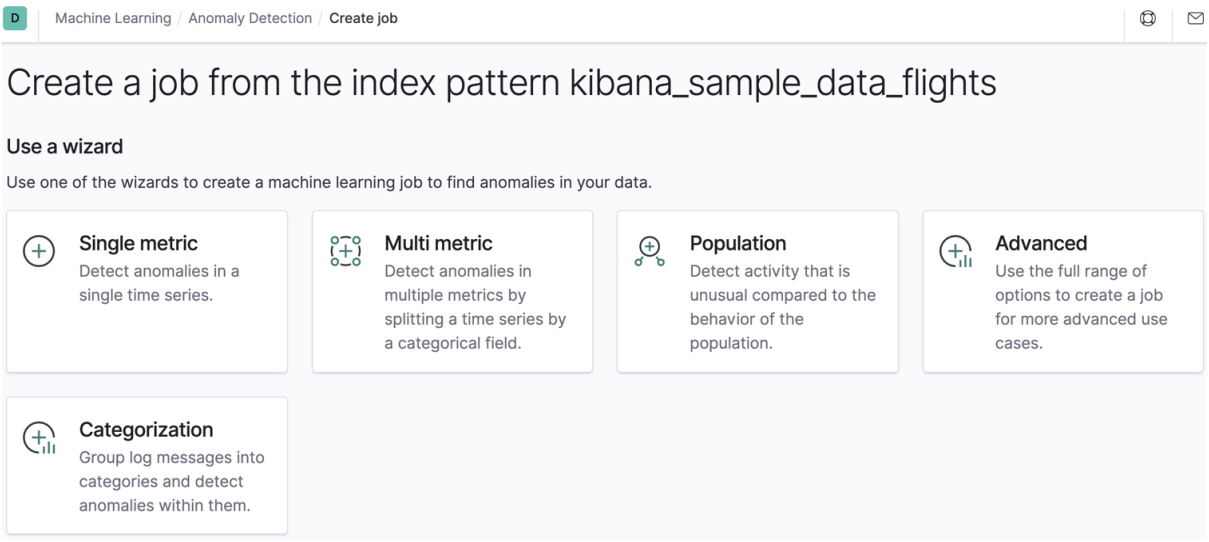


Рисунок 3.14 – Створити нову роботу

Одне метричне завдання – це просте завдання, яке містить один детектор. Детектор визначає тип аналізу, який буде відбуватися, і які поля аналізувати. На додаток до обмеження кількості детекторів, єдиний метричний майстер створення завдань опускає багато більш розширених параметрів конфігурації.

Мультиметричне завдання може містити більше одного детектора, що є більш ефективним, ніж виконання кількох завдань щодо одних і тих самих даних.

Робота населення виявляє незвичну діяльність порівняно з поведінкою населення. Для отримання додаткової інформації див. Виконання аналізу популяції.

Завдання категоризації групує повідомлення журналів за категоріями та використовує функції підрахунку або рідкісні для виявлення аномалій у них. Див. Виявлення аномальних категорій даних.

Розширене завдання може містити кілька детекторів і дозволяє налаштувати всі параметри завдання.

Kibana також може розпізнавати певні типи даних та надавати спеціалізовані майстри для цього контексту. Наприклад, якщо ви додали зразок набору даних веб-журналу, з’явиться такий майстер.

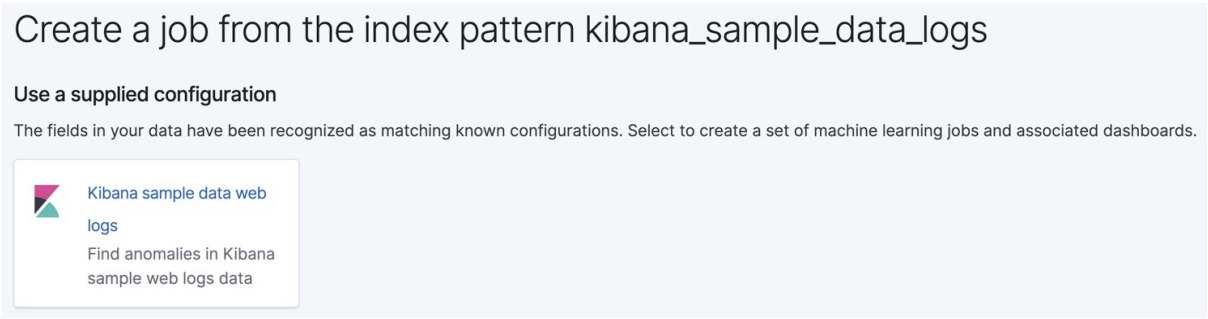


Рисунок 3.15 – Майстер створення веб-журналу зразків даних Kibana

Крім того, після завантаження зразка набору даних на домашню сторінку Kibana ви можете натиснути Перегляд даних> Завдання ML. Існують завдання виявлення аномалій як для зразка набору даних замовлення електронної комерції, так і для зразка набору даних веб-журналів.

Якщо ви використовуєте Elastic APM, Kibana також виявляє ці дані та надає майстри для завдань виявлення аномалій. Наприклад.

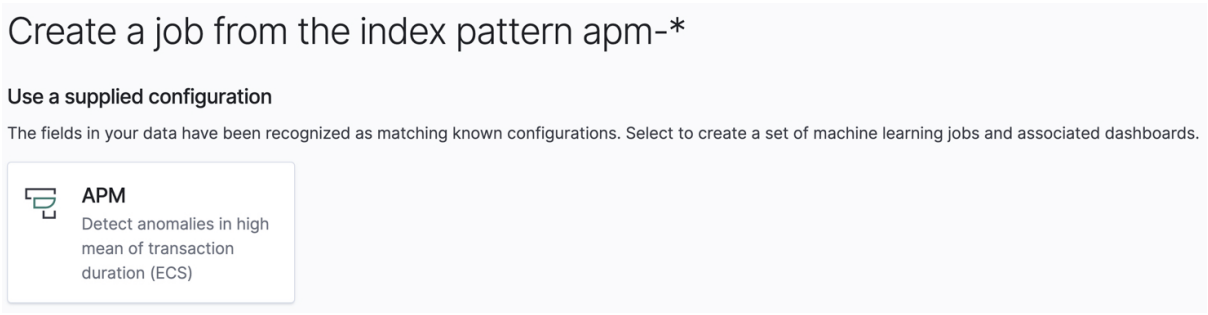


Рисунок 3.16 – Майстер створення робочих місць Kibana APM

Якщо ви використовуєте Filebeat для надсилання журналів доступу з ваших HTTP-серверів Nginx та Apache на Elasticsearch і зберігаєте їх за допомогою полів та типів даних із Спільної схеми пружності (ECS), з’являться такі майстри.

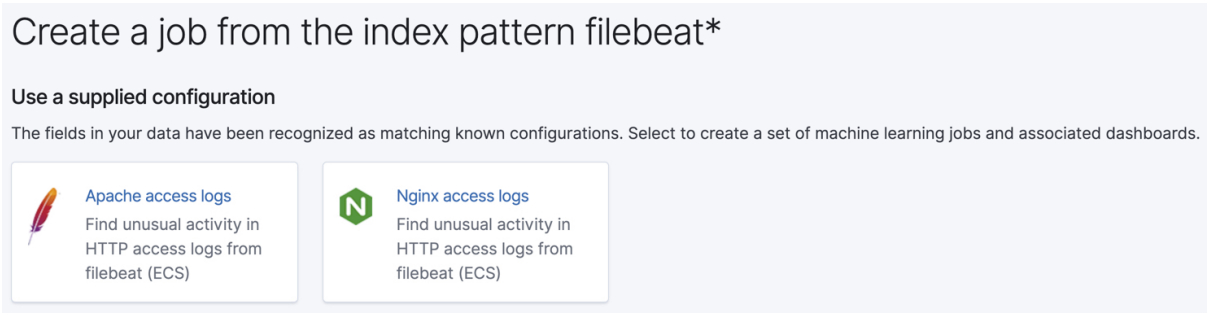


Рисунок 3.17 – Майстер створення завдань Filebeat

Якщо ви використовуєте Auditbeat для аудиту діяльності процесів у ваших системах, з'являються такі майстри.

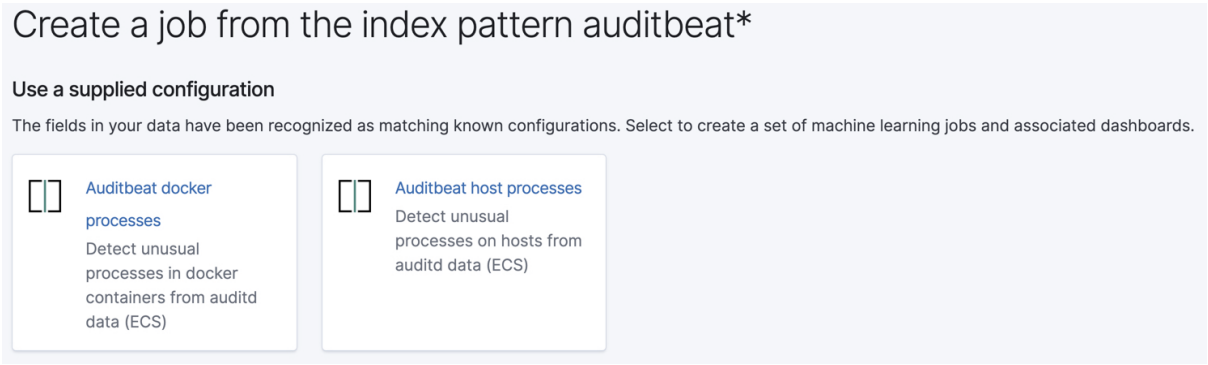


Рисунок 3.18 – Майстер створення робочих місць Auditbeat

Аналогічним чином, якщо ви використовуєте системний модуль Metricbeat для моніторингу своїх серверів, з'являться такі майстри.

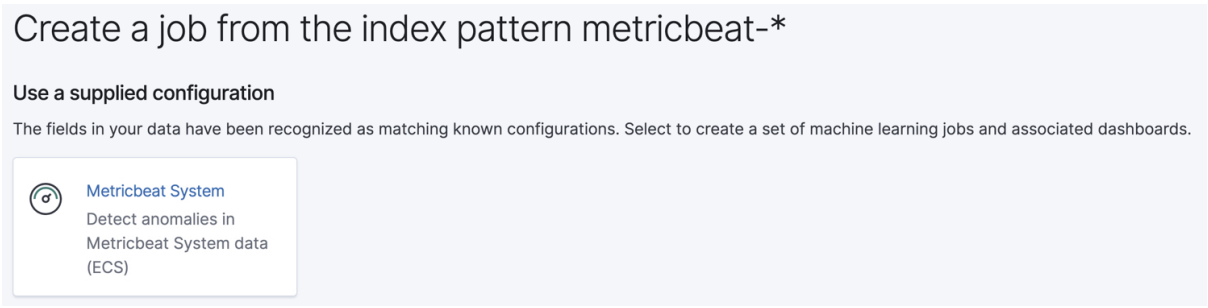


Рисунок 3.19 – Майстер створення завдань Metricbeat

Ці майстри створюють завдання виявлення аномалій, інформаційні панелі, пошуки та візуалізації, які налаштовані для аналізу даних Auditbeat, Filebeat та Metricbeat.

Коли ви створюєте завдання виявлення аномалій у Кібані, майстри створення робочих місць можуть надавати поради на основі характеристик ваших даних. Прислухаючись до цих пропозицій, ви можете створити робочі місця, які, швидше за все, дадуть глибокі результати машинного навчання.

Проміжок часу – це інтервал часу, який аналітика машинного навчання використовує для узагальнення та моделювання даних для вашої роботи. Створюючи завдання виявлення аномалії в Кібані, ви можете оцінити значення діапазону сегмента на основі ваших характеристик даних.

Якщо ви вибрали значення, яке перевищує один день або суттєво відрізняється від розрахункового, ви отримаєте інформаційне повідомлення. Для отримання додаткової інформації про вибір відповідного діапазону ковша див. Розділ Ковші.

Якщо у ваших даних є логічні групування пов’язаних сутностей, аналітика машинного навчання може створювати моделі даних та генерувати результати, які враховують ці групування. Наприклад, ви можете розділити свої дані за ідентифікатором користувача та виявити, коли користувачі отримують доступ до ресурсів не так, як зазвичай.

Якщо поле, яке ви використовуєте для розділення даних, має багато різних значень, завдання використовує більше ресурсів пам'яті. Зокрема, якщо потужність імені by_field, over_field_name або part__name_name перевищує 1000, вам повідомляють, що може використовуватися велика кількість пам'яті.

Подібним чином, якщо ви проводите аналіз популяції, а кількість імен над полем менше 10, вам повідомляють, що це може бути непридатним полем для використання. Для отримання додаткової інформації див. Виконання аналізу популяції.

Кожне завдання виявлення аномалій повинно мати один або кілька детекторів. Детектор застосовує аналітичну функцію до певних полів у ваших даних. Якщо ваше завдання не містить детектора або детектор не містить дійсної функції, ви отримуєте повідомлення про помилку.

Якщо завдання містить повторювані детектори, ви також отримуєте повідомлення про помилку. Детектори є дублікатами, якщо вони мають однакову функцію, ім'я_поля, ім'я_поля_поле, ім'я_поля_поле та ім'я_поля_розділу.

Для кожного завдання виявлення аномалій ви можете додатково вказати model_memory_limit, що є приблизним максимальним обсягом ресурсів пам'яті, необхідних для аналітичної обробки. Значення за замовчуванням – 1 Гб. Після досягнення цієї межі обрізання даних стає більш агресивним. Після перевищення цього обмеження нові сутності не моделюються.

Ви також можете додатково вказати параметр xpack.ml.max_model_memory_limit. За замовчуванням він не встановлений, а це означає, що на ваших робочих місцях немає верхньої межі допустимих значень model_memory_limit.

Якщо ви встановите model_memory_limit занадто високим, відкрити роботу буде неможливо; завдання не можна розподілити по вузлах, у яких недостатньо пам'яті для їх запуску.

Якщо передбачуваний ліміт пам'яті моделі для завдання виявлення аномалій перевищує межу пам'яті моделі для завдання або максимальний ліміт пам'яті моделі для кластера, майстри створення завдань у Кібані генерують попередження. Якщо передбачувана потреба в пам'яті лише трохи перевищує model_memory_limit, завдання, ймовірно, дасть корисні результати. В іншому випадку дії, які ви вживаєте для вирішення цих попереджень, залежать від ресурсів, доступних у вашому кластері:

- Якщо ви використовуєте значення за замовчуванням для model_memory_limit, а вузли машинного навчання в кластері мають багато пам’яті, найкращим способом дій може бути просто збільшення завдання model_memory_limit. Однак перед цим переконайтеся, що вибраний аналіз має сенс. За замовчуванням model_memory_limit є відносно низьким, щоб уникнути випадкового створення завдання, яке використовує величезний обсяг пам'яті.

- Якщо вузли машинного навчання в кластері не мають достатньої пам'яті для розміщення завдання передбачуваного розміру, єдиними варіантами є:
    - Додайте більші вузли машинного навчання до кластера, або
    - Прийміть, що робота досягне межі пам’яті, і не обов’язково знайде всі аномалії, які вона могла б знайти.

Якщо ви використовуєте Elastic Cloud Enterprise або розміщену службу Elasticsearch на Elastic Cloud, xpack.ml.max_model_memory_limit встановлено, щоб заборонити вам створювати робочі місця, які не можуть бути призначені для будь-яких вузлів машинного навчання в кластері. Якщо ви виявите, що не можете збільшити model_memory_limit для своїх завдань машинного навчання, рішення полягає в збільшенні розміру вузлів машинного навчання у вашому кластері.

Для кожного завдання виявлення аномалії ви можете додатково вказати виділений індекс для зберігання результатів виявлення аномалії.

Оскільки завдання виявлення аномалій можуть давати велику кількість результатів (наприклад, завдання з багатьма часовими рядами, малим діапазоном сегмента або з тривалим періодом роботи), рекомендується використовувати спеціальний індекс результатів, вибравши опцію Використовувати виділений індекс у Кібані або вказівка назви_індексу_за допомогою API створення завдань виявлення аномалій.

3.3.10 Сповіщення

Ця функціональність знаходиться в бета-версії та може бути змінена. Дизайн і код менш зрілі, ніж офіційні функції GA, і надаються як є, без гарантій. Бета-функції не підлягають підтримці SLA офіційних функцій GA.

Додаток APM інтегрується з функцією попередження та дій Kibana. Він надає набір вбудованих дій та попередження щодо порогових значень APM, які ви можете використовувати, та забезпечує централізоване управління всіма попередженнями від Kibana Management.

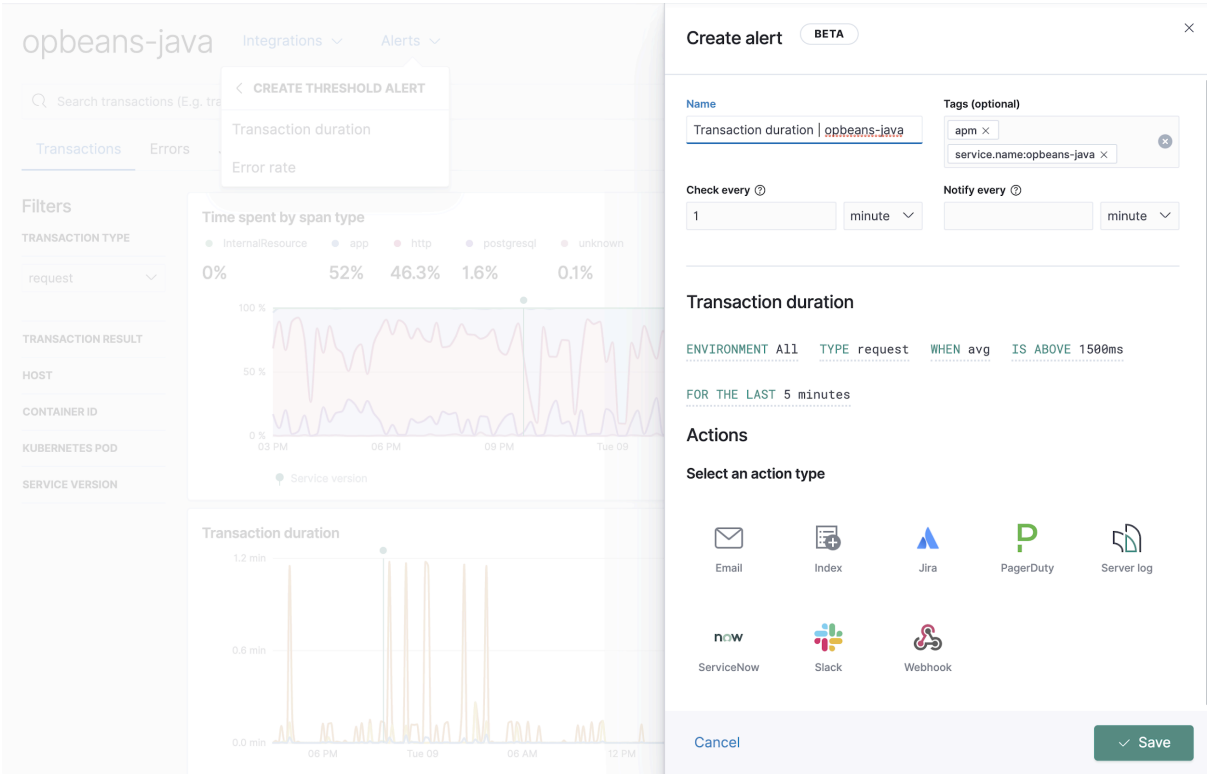


Рисунок 3.20 – Створіть сповіщення в програмі APM

Покрокове керівництво панеллю виведення сповіщень, включаючи детальну інформацію про кожну властивість, що налаштовується, див. У визначених попередженнях Kibana.

Додаток APM підтримує два різні типи порогових сповіщень: тривалість транзакції та частоту помилок. Нижче ми створимо по одному з кожного.

Попередження про тривалість транзакції спрацьовують, коли тривалість певного типу транзакції в службі перевищує визначений поріг. Цей посібник створить попередження для служби opbeans-java на основі таких критеріїв:

- Середовище: Виробництво;
- Тип транзакції: транзакція. Тип: запит;
- Середній запит перевищує 1500 мс за останні 5 хвилин;
- Перевіряйте кожні 10 хвилин і повторюйте попередження кожні 30; хвилин;
- Надішліть сповіщення через Slack;

У додатку APM перейдіть до служби opbeans-java та виберіть Сповіщення> Створити порогове сповіщення> Тривалість транзакції.

Тривалість операції | opbeans-java автоматично встановлюється як ім'я попередження, а apm та service.name:opbeans-java додаються як теги. Добре змінити назву сповіщення, але не редагуйте теги.

На основі критеріїв попередження визначте такі деталі попередження:

- Перевіряйте кожні 10 хвилин;
- Повідомляти кожні 30 хвилин;
- ТИП – запит;
- КОЛИ – сер;
- ВИЩЕ – 1500 мс;
- ОСТАННІШЕ – 5 хвилин;

Виберіть тип дії. Можна вибрати кілька типів дій, але в цьому прикладі ми хочемо розміщувати повідомлення на каналі Slack. Виберіть Slack> Створити з'єднувач. Введіть ім’я роз’єму та вставте URL-адресу веб-хука. Якщо вам потрібно створити його, перегляньте документацію веб-хука Slack.

Додайте тіло повідомлення у форматі розмітки. Ви можете використовувати синтаксис шаблону Mustache, тобто {{змінну}}, щоб передавати значення попередження під час виявлення умови до дії. До списку доступних змінних можна отримати доступ, натиснувши кнопку додати змінну кнопку додати змінну. Виберіть Зберегти. Попередження створено і зараз активне.

Попередження про частоту помилок запускаються, коли кількість помилок у службі перевищує визначений поріг. Цей посібник створює попередження для служби opbeans-python на основі таких критеріїв:

- Середовище: Виробництво;
- Частота помилок вище 25 за останню хвилину;
- Перевіряйте кожну 1 хвилину та повторюйте попередження кожні 10 хвилин;
- Надішліть сповіщення електронною поштою команді opbeans-python;

У програмі APM перейдіть до служби opbeans-python і виберіть Сповіщення далі Створити порогове попередження та Частота помилок.

Частота помилок | opbeans-python автоматично встановлюється як ім'я попередження, а apm та service.name:opbeans-python додаються як теги. Добре змінити назву сповіщення, але не редагуйте теги.

На основі критеріїв попередження визначте такі деталі попередження:

- Перевіряйте кожні – 1 хвилину;
- Повідомляти кожні 10 хвилин;
- ВИЩЕ – 25 помилок;
- ОСТАННІЙ – 1 хвилина;

Виберіть тип дії електронної пошти та натисніть Створити сполучник. Заповніть необхідні дані: відправника, хоста, порту тощо та натисніть «Зберегти».

Додайте тіло повідомлення у форматі розмітки. Ви можете використовувати синтаксис шаблону Mustache, тобто {{змінну}}, щоб передавати значення попередження під час виявлення умови до дії. До списку доступних змінних можна отримати доступ, вибравши кнопку додати змінну кнопку додати змінну.

У програмі APM виберіть Сповіщення> Переглянути активні сповіщення, щоб перейти на сторінку керування сповіщеннями та діями Kibana. На цій сторінці ви можете створювати, редагувати, вимикати, вимикати та видаляти сповіщення, а також створювати, редагувати та вимикати з'єднувачі.

Оповіщення можуть відправлятися на внутрішній мікросервіс та ініціювати дію над сервером в системі, наприклад:

- Виведення серверу з кластеру DNS;
- Вимкнення серверу;

Виведення серверу з ДНС відбувається за допомогою звернення до API методів DNS провайдера, наприклад AWS Route 53 або Constalix.

Вимкнення відбувається за допомогою API методів провайдера VPS провайдера наприклад AWS або Digital Ocean.

3.4 Висновки до третього розділу

У третьому розділі була описана розроблена система моніторингу на збору логів: архітектура, основні компоненти, алгоритми, структури даних та засоби реалізації. Архітектурно IAC-сервер Rsyslog, Filebeat, Metricbeat Elasticsearch, аналітична модель виду "правила пошуку аномалій" та компоненти користувацького інтерфейсу інженера Kibana. Ми показали налаштування, які необхідно виконати на сервері, щоб сформувати записи у форматі JSON. Була продемонстрована робота навчального парсеру APM, показані цикли його навчання. За допомогою графічного інтерфейсу Kibana продемонстрована робота нотифікацій а випадку виявлення аномалій. Продемонстрована робота користувацької компоненти інженера. Усі задіяні засоби реалізуються відкритими та доступними безкоштовно. Отримана система моніторингу системних журналів працює і виконує своє призначення.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці. Вимоги до серверних приміщень

Серверне приміщення слід розміщувати якомога ближче до магістральних кабельних каналів. Або ж проектувати майбутню кабельну інфраструктуру відповідним чином.

В ідеалі, серверне приміщення повинно бути поруч з головним розподільчим пунктом (Main Cross, MC), а якщо є можливість, то слід організувати головний розподільний пункт безпосередньо в самому серверному приміщенні.

Переважно розміщувати апаратну недалеко від вантажних або вантажопасажирських ліфтів, використовуваних для транспортування важкого обладнання, наприклад ДБЖ. Водночас, слід уникати близького розміщення потужних джерел електричних і магнітних полів, а також обладнання, яке може викликати підвищену вібрацію.

Вібрація негативно впливає на роботу активного обладнання, контакти і з'єднання. У діапазоні частот до 25 Гц амплітуда коливань не повинна перевищувати 0.1 мм

Не рекомендується виділяти приміщення для апаратної на верхніх поверхах будівлі, оскільки вони найбільш схильні до пошкоджень у разі пожежі і можуть заливатися при протіканнях даху.

Не рекомендується розміщувати серверне приміщення поруч з сходовими прольотами, ліфтовими шахтами, великими вентиляційними каналами та іншими елементами будівлі, які, згідно існуючого планування, можуть обмежити розширення серверного приміщення в майбутньому.

Не допускається розміщення апаратної під приміщеннями, пов'язаними зі споживанням води: туалети, душові, їдальні, буфети і т. д.)

Серверне приміщення потрібно розмістити осторонь від джерел електромагнітних завад на такій відстані, щоб напруженість електричного поля не перевищувала 3 В/м у всьому спектрі частот.

Серверне приміщення рекомендується розміщувати так, щоб була доступна можливість розширення приміщення за рахунок площі приміщення що знаходиться поруч. При цьому слід враховувати розташування несучих перекриттів, для того, щоб при необхідності була можливість збільшити простір апаратного приміщення шляхом демонтажу суміжних з сусіднім приміщенням стін.

Розмір серверного приміщення вибирається виходячи з розміру обслуговування робочої області і кількості встановленого обладнання. Важливо врахувати не тільки розміри самого обладнання, а й способи монтажу, забезпечення доступу та обслуговування обладнання, можливість установки додаткових пристроїв. Тобто, щоб було місце для будь-якого монтажного маневру.

Розміри апаратної повинні відповідати вимогам до що розташовується в ній обладнанню або (при відсутності даних) складати 0,07 квадратних метра на кожні 10 квадратних метрів площі обслуговування робочих місць.

У будівлях з низькою щільністю робочих місць площа апаратної повинна становити не менше 37 квадратних метрів – на не більше 400 робочих місць, не менше 74 квадратних метрів – на не більше 800 робочих місць і не менше 111 квадратних метрів – на не більше 1200 робочих місць.

Мінімальна висота стелі апаратної повинна становити 2,44 м.

Пол апаратної повинен бути рівний і мати антистатичне покриття з опором 106 Ом, що забезпечує стікання і відведення статичної електрики. Настил підлоги здійснюється на незгораюче підставу. Покриття повинне дозволяти виконувати очитку пилососом і вологе прибирання.

Максимально допустиме навантаження на підлогу в апаратній повинна становити: розподілене навантаження не більше 12 кПа; зосереджене навантаження не більше 4,4 кН.

Уникайте розміщення серверного приміщення нижче рівня поверхні землі, якщо приміщення не буде забезпечено захистом від проникнення води.

Якщо в серверному приміщенні все ж є вікна, то необхідно закласти вікна цеглою. Однак, якщо це нездійсненно і в технічному приміщенні все ж передбачені вікна, то рекомендується розташовувати апаратну на північній або північно-східній стороні будівлі.

Дверний отвір повинен бути в ширину не менше 0.91 м і висотою не менше 2 метрів.

Якщо планується проносити габаритне обладнання в серверне приміщення, то рекомендується встановити подвійні двері з мінімальним отвором в ширину не менше 1.82 метра і висотою не менше 2,28 метра. Навісні двері повинні відкриватися назовні.

Вхідні двері в апаратну повинні виготовлятися з важкогорючих матеріалів, мати протизнімні пристосування і відкриватися назовні з кутом розкриву 180 градусів. Двері повинні мати ущільнюючу прокладку і замикатися на внутрішній замок.

Мінімальна висота стелі апаратної повинна становити 2,44 м.

Поли в апаратній повинні бути рівними і мати антистатичне покриття з опором 106 Ом, що забезпечує стікання і відведення статичної електрики. Настил підлоги здійснюється на незгораючу підставу. Покриття повинне дозволяти виконувати очистку пилососом і вологе прибирання.

Захисного і телекомунікаційного заземлення, причому з апаратної повинна бути забезпечена можливість підключення безпосередньо до головної пластині заземлення.

Система контролю і управління мікрокліматом повинна забезпечити заданий рівень вологості і температури необхідний для нормального функціонування активного обладнання.

Система мікроклімату повинна забезпечити підтримку температурного режиму не тільки влітку, а й взимку і розрахована на цілодобову безперервну роботу.

Рекомендується повна зміна повітря не рідше 1 разу на годину, якщо в приміщенні постійно працює обслуговуючий персонал.

Рекомендується використовувати систему очищення і фільтрації повітря, що поступає в апаратне приміщення.

Якщо в будинку встановлена система резервного електроживлення, то система підтримки мікроклімату повинна бути до неї підключена.

Серверне приміщення має бути захищене від пилу і шкідливих речовин, які можуть негативно впливати на роботу обладнання та на матеріали обладнання.

Необхідно забезпечити освітлення не менше 500 люкс. Світильники необхідно розміщувати на стелі.

Для управління освітленням потрібно використовувати однин або декілька вимикачів і розташовувати їх поряд з дверима на висоті 1.5м від рівня підлоги.

У серверному приміщенні забороняється використовувати пристрої плавного регулювання освітлення (дімери).

Доцільно як мінімум, два окремі блоки подвійних електричних розеток. Блоки електричних розеток рекомендується живити від різних живильних кабелів, електричні розетки повинні бути розраховані на змінний струм до 16А. Додатково потрібно встановити блоки з подвійними електричними розетками з інтервалом 1,8 метра уздовж стіни на висоті не нижче 0,15 метра від рівня підлоги.

Подача електроживлення в серверне приміщення повинна здійснюватися по виділеному силовому кабелю, бажано безпосередньо від головного розподільного щита.

Якщо встановлена система резервного електроживлення, то серверне приміщення повинно бути до неї підключено.

Дозволяється установка джерел безперебійного живлення (ДБЖ) до 100 кВА. ДБЖ потужністю понад 100 кВА повинні бути встановлені в окремому приміщенні.

Засоби розподілу і організації кабельних потоків повинні бути надійно закріплені, витримувати вагу кабелю, повинні забезпечити захист і розподіл кабелів з мінімально допустимим радіусом вигину кабелю.

Після прокладки кабелів необхідно закрити вогнетривким матеріалом всі кабельні вводи в серверне приміщення. Для цих цілей можна використовувати спеціальні заглушки, які встановлюються в кабельному вводі, які у разі виникнення пожежі розширюються, перекривають простір і не дозволяють поширитися вогню і диму.

Стельові перекриття, стіни і перегородки серверного приміщення повинні бути негорючими та забезпечувати вогнестійкість не менше 45 хвилин.

Двері повинні забезпечити вогнестійкість не менше 36 хвилин. Двері може бути виготовлена з важко згораємого матеріалу товщиною не менше 40 мм без внутрішніх порожнеч або можна використовувати дерев'яні двері, але покрити їх шаром азбесту або оббити листовою сталлю товщиною не менше 4 мм з двох сторін.

У приміщенні повинні встановлюватися витяжні шахти з ручним або автоматичним відкриванням. Площа шахт повинна бути не менше 0.2% від площі приміщення і відстань, з будь-якої точки приміщення до шахти повинне бути не більше 20 метрів.

4.2 Оцінка стійкості роботи об’єкту економіки до впливу вражаючих факторів ядерної зброї.

Для оцінки стійкості функціонування підприємства начальником центрального округу об’єкту економіки, штабом центрального округу і головними спеціалістами проводяться спеціальні дослідження. До них

залучаються виконавці від об’єкту економіки, робітники галузевих промислових об’єктів економіко-технологічних і науково-дослідних інститутів.

Робота проводиться в чотири етапи:

- підготовчий;
- оцінка стійкості об’єкту;
- розробка заходів щодо підвищення стійкості функціонування об’єкту економіки і його елементів;
- оформлення документації за результатами дослідження.

На першому етапі дослідження відпрацьовуються необхідні документи:

- наказ начальника центрального округу об’єкту економіки на проведення дослідження;
- календарний план підготовки і проведення дослідження, де визначаються виконавці, терміни виконання робіт, керівники і склад груп, що вирішують специфічні завдання;
- завдання групам на проведення дослідження по конкретним питанням.

Таких груп може бути декілька.

Перша група (від відділу капітального будівництва) визначає фізичну зношеність елементів об’єкту (мінімальний надлишковий тиск, який вони можуть витримати), а також захисних споруд і індивідуальних укриттів для персоналу, що обслуговує агрегати безперервного циклу.

Друга група (від відділу головного механіка) оцінює стійкість технологічного і лабораторного обладнання; можливість виникнення вторинних уражаючих факторів; достатність захисту унікального і цінного обладнання.

Третя група (від відділу головного енергетика) оцінює стійкість функціонування енергооб’єктів, мереж і комунікацій, стійкість функціонування зовнішніх і внутрішніх джерел електроб’єкту економікинергії, а також їх вводів.

Четверта група (від відділу головного технолога) визначає найбільш уразливі ділянки технологічного процесу; можливі руйнування станкового обладнання, місця порушення технологічних процесів внаслідок деформації або обвалу елементів будівель; можливість зміни технологічного процесу при виході

зі строю уразливих ділянок; можливість заміни матеріалів, сировини, комплектуючих виробів, палива з врахуванням місцевих ресурсів.

П’ята група (від відділу постачання і збиту) оцінює: наявність, умови зберігання і забезпечення збереження запасів і резервів матеріальних цінностей (паливо-мастильних матеріалів, сировини, комплектуючих), їх захищеність від впливу уражуючих факторів; стійкість виробничих зв’язків і умов отримання пального, сировини, комплектуючих виробів від постачальників; можливість переходу на підвищення норми запасів; можливість постачання за рахунок дублерів і місцевих ресурсів в умовах надзвичайних ситуацій; доцільність розвитку мережі доріг, під’їзних шляхів; терміни роботи об’єкту без поставок необхідних матеріалів.

Шоста група (від штабу і служб центрального округу) оцінює стійкість систем управління, оповіщення, зв’язку, захисні властивості будівель в радіаційному відношенні; визначає забезпеченість персоналу засобами індивідуального захисту (ЗІЗ), наявність цих засобів і їх готовність до використання за призначенням. Уточнює план центрального округу об’єкту.

Сьома група, під керівництвом головного інженера об’єкта організовує і контролює роботу всіх груп, оформлює усі необхідні документи стосовно дослідження.

Другий етап дослідження (оцінка стійкості) починається з вивчення району розташування об’єкту (місто, рівнинна або болотиста місцевість), дослідження його планування, комунікацій.

При цьому проводиться аналіз уразливості елементів, а також об’єкту в цілому в умовах надзвичайних ситуацій, визначаються інженерно-технічні заходи, виконання яких дозволить забезпечити підвищення стійкості об’єкту.

На даному етапі проводиться аналіз:

- наслідків аварій окремих систем виробництва;
- розповсюдження ударної вибухової хвилі по території об’єкту (місця і характер вибухів, їх потужність і імовірні наслідки);
- розповсюдження вогню при різних видах пожежі;

- надійність комунікацій і промислових комплексів;
- розповсюдження хмари зараженого повітря при викиді шкідливих речовин;
- можливості утворення токсичних і пожеженобезпечних сумішей.

При організації робіт другого етапу можна застосовувати різноманітні методи аналізу пошкоджень і дефектів:

- метод оцінки зростання пошкоджень в системі після аварії з побудовою «древа відмов»;
- метод побудови «древа подій» для визначення імовірності аварії. При цьому використовується інформація про несправності компонентів обладнання і про можливості зниження їх негативного впливу на навколишнє середовище.

Оцінка стійкості елементів об'єкта економіки і об'єкта в цілому до впливу ударної повітряної хвилі. Критерієм оцінки вважають величину надлишкового тиску, яка чинить руйнівний вплив на елементи об'єкта економіки, тобто будівлі, споруди і обладнання об'єкта зберігаються або отримують слабкі чи середні руйнування. Це значення надлишкового тиску прийнято вважати граничною межею стійкості відповідного об'єкту до ударної повітряної хвилі $\Delta P_{ф\ lim}$.

Оцінка стійкості об'єкта до впливу ударної повітряної хвилі полягає у визначенні $\Delta P_{ф\ lim.}$

Для цього потрібні такі вихідні дані:

- місце знаходження точки прицілювання;
- віддалення об'єкта від точки прицілювання R_r, км;
- потужність ядерного вибуху, що очікується q, МТ;
- імовірне максимальне відхилення центру вибуху від точки прицілювання $r_{відх.}$, км;
- характеристика об'єкта і його елементів.
- В якості орієнтовних розрахункових значень для підприємств харчової промисловості $\Delta P_ф$, може дорівнювати 5…60кПа.

Оцінки підлягають всі елементи цеху, в тому числі комунікації (виявляються найбільш уразливі елементи і дільниці, від яких залежить робота всього об’єкту.

Приймаючи різні величини надлишкового тиску, визначають стійкість конкретних елементів цеху і обладнання, а також характер їх руйнувань. Відстані, на яких імовірна поразка елементів об’єкта і ступінь їх ураження визначаються за допомогою довідкового матеріалу центрального округу.

Після аналізу результатів визначається перелік інженерно-технічних заходів центрального округу, які доцільно провести на досліджуваному об’єкті для того, щоб підвищити його стійкість.

При виконанні розрахунків необхідно враховувати, що звичайно обладнання виходить зі строю не від прямого впливу ударної повітряної хвилі, а від вторинних уражаючих факторів (падаючих балок, обломків конструкції будівлі).

Суттєвий вплив на працездатність обладнання чинить місце його розтушування у цеху.

Руйнування будівель звичайно приводить до пошкодження внутрішніх мереж комунікацій, що може бути причиною пожеж, вибухів, затоплень, загазованості.

Оцінка стійкості елементів об’єкту економіки і об’єкту в цілому від впливу світлового випромінювання. Такий вплив приводить до спалахування горючих матеріалів, розвитку пожеж, опікам різної ступені. Критерієм впливу є світловий імпульс, при якому відбувається спалахування або стійке горіння елементів.

Визначення можливості роботи при радіоактивному зараженні території об’єкту економіки. Радіоактивне зараження місцевості звичайно не чинить суттєвого впливу на технологічні процеси, за винятком об’єктів хімічної, електронної і харчової промисловості.

Критерієм оцінки стійкості елементів об’єкту економіки і продукції, яка ним випускається є доза випромінювання. Захист визначається до об’єкту

економіки коефіцієнтом послаблення радіації, який розраховується за формулою:

$K_{посл.} = 2h/a,$

де h – товщина захисного шару, см,

a – товщина шару половинного послаблення, см.

Необхідні для розрахунків дані беруться із довідкових матеріалів центрального округу. Отримані дані зводяться у таблицю і використовуються для визначення режимів радіаційного захисту, які потрібно впроваджувати при реальній радіаційній обстановці.

Оцінка ступеня впливу вторинних уражаючих факторів. Найбільш важливо визначити можливі джерела виникнення вторинних уражаючих факторів.

До внутрішніх джерел вторинних уражаючих факторів відносяться ємкості, резервуари з ЛСР і газами, вибухонебезпечні технологічні установки і комунікації легко спалахуючи споруди, що знаходяться на території досліджуваного об’єкту.

Зовнішні джерела вторинних уражаючих факторів знаходяться зовні об’єкта. Це підприємства нафто-, газо-, хімічної промисловості, холодильники, гідровузли.

Оцінка хімічного і біологічного впливу в районі розташування об’єкту економіки. В результаті збільшення наслідків надзвичайних ситуацій – особливо при температурі повітря порядку 35^0С і забрудненні води, наявності трупів, територія об’єкту може опинитися в осередку біологічного зараження.

Основними заходами по захисту в даному випадку будуть:

- забезпечення персоналу засобами індивідуального і колективного захисту, готовність і вміння використовувати ці засоби;
- забезпеченість знезараженими продуктами і водою;
- оцінка можливості розосередження і евакуації людей в межах карантинної зони.

Аналізується вплив зараження на процес виробництва, продукцію, сировину.

Вивчається можливість герметизації цехів і технологічних ліній, можливість роботи з використанням засобів індивідуального захисту.

Забезпечується можливість проведення спеціальної обробки людей, техніки, обладнання, території, а також проведення протиепідемічних заходів.

Підвищення стійкості управління об’єкту економіки в умовах надзвичайних ситуацій. Управління – це основа діяльності начальника центрального округу об’єкту і його штабу. Сутність управління полягає в здійсненні постійного керівництва персоналом об’єкта, невоєнізованими формуваннями на всіх етапах їх діяльності, доведенні задач до підлеглих і контролі за їх виконанням.

На об’єкту економіки повинна бути розроблена схема оповіщення і зв’язку для всіх варіантів діяльності.

Управління повинно бути безперервним на всіх етапах (при загрозі нападу, при проведенні евакуації і розосередженні, при виконанні Р і НР), твердим і гнучким.

На об’єкті створюються дві групи управління. Одна з них за сигналом «про загрозу нападу» убиває в район розосередження на запасний пункт управління, який повинен бути повністю обладнаним і готовим до роботи.

Для забезпечення надійного управління при надзвичайних ситуаціях в одному зі сховищ створюється пункт управління, якій обладнується необхідною для управління апаратурою. Комунікації до пункту управління підводяться у підземному виконанні, з дублюванням і встановленням захисту від електромагнітного імпульсу.

Між основним і запасним пунктами управління встановлюється надійний зв'язок. Звертається увага на забезпечення зв’язку зі суміжними об’єктами економіки і начальниками центрального округу територій.

Провівши оцінку стійкості окремих елементів об’єкту економіки, можна дати оцінку стійкості його виробничій діяльності в цілому. Таблиці, графіки,

схеми, які відпрацьовані під час проведення дослідження є документами на підставі яких розробляються (оцінюються пропозиції) інженерно-технічні заходи центрального округу.

На третьому етапі дослідження оцінюється реальність і економічна доцільність проведення запропонованих заходів по підвищенню стійкості і проводиться відбір оптимальних рішень. На даному етапі остаточно вирішується питання про готовність об’єкту до відновлення виробництва або зміну його профілю.

План ремонтно-відновлювальних робіт приймає остаточний вигляд.
На четвертому етапі дослідження оформлюються звітні документи, основним з яких є «План – графік нарощування заходів по підвищенню стійкості функціонування об’єкта економіки».

План розроблених заходів представляється по інстанції для його затвердження і виділення необхідних засобів. Остаточно ступінь підвищення стійкості і терміни виконання запланованих заходів визначаються вищою інстанцією або територіальним органом. При цьому проводиться розбивка робіт по термінам, виділяються необхідні сили і засоби, визначаються обсяг і вартість робіт по кожному заходу, джерела фінансування, призначаються відповідальні виконавці.

Оскільки всі роботи не можуть бути виконані в короткий термін, то складається перспективний план з щорічною фіксацією виконання заходів.

Підготовка до безаварійного зупинення виробництва. На кожному промисловому об’єкту економіки на випадок виникнення надзвичайної ситуації розробляється План швидкого і безаварійного зупинення виробництва. Він повинен забезпечити зниження до мінімуму імовірності виникнення вторинних уражаючих факторів. Реальність Плану і готовність персоналу до його виконання визначається на регулярних тренуваннях під час відпрацювання питань центрального округу.

При цьому завчасно розробляється необхідний комплект документації. Планом передбачається навчання персоналу по безаварійному зупиненню виробництва.

Заходи по підготовці до швидкого відновлення виробництва. Аналіз наслідків надзвичайної ситуації показує, що об’єкту економіки отримують пошкодження, які можуть бути відновлені власними силами. Тому на кожному об’єкті відпрацьовуються питання відновлення виробництва після отримання слабких і середніх пошкоджень, при цьому для кожного варіанту складається План першочергових відновлювальних робіт силами персоналу об’єкта з врахуванням запасів матеріальних засобів і обладнання.

До відновлення виробництва персонал об’єкта готується завчасно, при цьому така підготовка повинна включати:

- плани відновлення елементів об’єкта, відповідно до аналізу можливої обстановки при різних варіантах руйнувань;
- розроблені технологічні схеми для продовження виробництва при виході зі строю обладнання, ліній, цехів за рахунок перерозподілу приміщень і людських ресурсів, спрощення технологічного процесу;
- документація для проведення відновлювальних робіт, в тому числі по будівництву тимчасових споруд;
- розрахунки по відновленню споруд при прогнозованому характері руйнувань, перелік і загальний обсяг відновлювальних робіт (вартість, терміни, трудові витрати), необхідні для цього сили і засоби;
- створення матеріальних ресурсів для відновлювальних робіт, забезпечення їх збереження і регулярного освіження;
- розрахунки потреби людських ресурсів;
- визначення імовірної черговості виконання відновлювальних робіт.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи магістра було досягнуто поставленої мети дослідження, а саме було проведено аналіз моделей централізованого розумного моніторингу, аналіз рішень, які дозволяють використовувати різні дані для пошуку аномалій та несправностей та механізми автоматизації управління серверами в мережі.

В ході виконання даного дослідження отримано наступні результати:

- Проведено аналіз літературних джерел щодо актуальності дослідження, розглянуто основні питання;
- Визначено концепції моніторингу серверів;
- Проведено огляд різних аспектів – моніторингу, легування, пошуку аномалій, автоматизації керування серверами в навантаженій мережі;
- Проведено огляд додатків та сервісів розумного міста;
- Проведено огляд деяких аспектів, пов'язаних з технологічним рішенням моніторингу та автоматизації;
- Проведено огляд моделей автоматизацій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Словник термінів ITIL® на українській мові, версія 2.0, 29 липня 2011 г. на основі англійської версії 1.0, 29 липня 2011.

2. Business Process Model and Notation (BPMN). Version 2.0.2 [Електронний ресурс] – Режим доступу до ресурсу: http://www.omg.org/spec/BPMN//2.0.2/PDF

3. RFC 3411 An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks [Електронний ресурс] – Режим доступу до ресурсу: https://www. ietf.org/rfc/rfc3411.txt

4. RFC 5424 The Syslog Protocol [Електронний ресурс] – Режим доступу до ресурсу: https://tools. ietf.org/html/rfc5424

5. Adiscon LogAnalyzer – syslog web viewer, analysis and reporting tool [Електронний ресурс] – Режим доступу до ресурсу: http://loganalyzer. adiscon.com

6. MonitorWare Knowledge Base [Електронний ресурс] – Режим доступу до ресурсу: http://kb.monitorware.com/

7. Операційна аналітика, управління журналами, управління додатками, забезпечення безпеки підприємства та відповідності вимогам | Splunk [Електронний ресурс] – Режим доступу до ресурсу: http://www.splunk.com/ru_ru

8. IBM BigInsights for Apache Hadoop [Електронний ресурс] – Режим доступу до ресурсу: http://www-03.ibm.com/software/products /ru/ibm-biginsights-for-apache-hadoop

9. Welcome to Apache ™ Hadoop® ! [Електронний ресурс] – Режим доступу до ресурсу: http://hadoop.apache.org/

10. Fluentd | Open Source Data Collector [Електронний ресурс] – Режим доступу до ресурсу: http://www.fluentd.org/

11. Logstash | Collect, Enrich & Transport Data [Електронний ресурс] – Режим доступу до ресурсу: https://www.elastic.co/products/logstash

12. Elasticsearch | Search & Analyze Data in Real Time [Електронний ресурс] – https://www.elastic.co/products/elasticsearch

13. Welcome to Apache Lucene [Електронний ресурс] – Режим доступу до ресурсу: https://lucene.apache.org/

14. System Logging and Log Analysis (AKA: Everything we know and hate about system logging) [Електронний ресурс] – Режим доступу до ресурсу: http://www.ranum.com/security/computer_security/archives/logging-notes. pdf

15. Regular-Expressions. info – Regex Tutorial, Examples and Reference – Regexp Patterns [Електронний ресурс] – Режим доступу до ресурсу: http://www.regular-expressions.info/

16. Li, Weixi Automatic Log Analysis using Machine Learning: Awesome Automatic Log Analysis version 2.0 [Електронний ресурс] – Режим доступу до ресурсу: http://uu.diva-portal.org/smash/get/diva2:667650/FULLTEXT01. pdf

17. Category: String similarity measures [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Category:String_similarity_measures

18. Levenshtein distance [Електронний ресурс] – Режим доступу до ресурсу: https://en. wikipedia.org/wiki/Levenshtein_distance

19. Overlap_coefficient [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Overlap_coefficient

20. Longest common subsequence problem [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Longest_common_subsequence_problem

21. Машинне навчання [Електронний ресурс] – Режим доступу до ресурсу: http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5

22. Карасьов А.А. Математичне і програмне забезпечення бази експертних знань для поддерекі прийняття рішення при вирішенні інцидентів в

інформаційних системах: дис... канд тих. наук: 04.1 3.11/Москва, MIPEA 2014 – 243 з

23. RFC 5675 Mapping Simple Network Management Protocol (SNMP)

24. Notifications to SYSLOG Messages [Електронний ресурс] – Режим доступу до ресурсу: https://tools. ietf.org/html/rfc5675

25. Rsyslog. The rocket-fast system for log processing [Електронний ресурс] – Режим доступу до ресурсу: http: // www. rsyslog.com /

26. Java Standard Edition [Електронний ресурс] – Режим доступу до ресурсу: http://www.oracle.com/technetwork/java/javase/overview/index.html

27. SWT: The Standard Widget Toolkit [Електронний ресурс] – Режим доступу до ресурсу: https://www.eclipse.org/swt/

28. JSON. simple – A simple Java toolkit for JSON [Електронний ресурс] – Режим доступу до ресурсу: https://code.google.com/archive/p/json-simple/

29. Порівнюємо Java-бібліотеки для роботи з JSON: JSON. simple, GSON, Jackson і JSONP. [Електронний ресурс] – Режим доступу до ресурсу: https://tproger.ru/translations/java-json-library-comparison/

30. RFC 7159 The JavaScript Object Notation (JSON) Data Interchange Format [Електронний ресурс]] – Режим доступу до ресурсу: https://tools.ietf.org/html/rfc7159

31. Weka 3: Data Mining Software in Java [Електронний ресурс] – Режим доступу до ресурсу: http://www.cs.waikato.ac.nz/~ml/weka/

32. IBM SPSS Statistics Standard [Електронний ресурс] – Режим доступу до ресурсу: http://www-03.ibm.com/software/products/ru/spss-stats-standard

33. R. Agrawal, R. Srikant: Fast Algorithms for Mining Association Rules in Large Databases. In: 20th International Conference on Very Large Data Bases

34. Tobias Scheffer: Finding Association Rules That Trade Support Optimally against Confidence. In: 5th European Conference on Principles of Data Mining and Knowledge Discovery, 424-435, 2001

35. P. A. Flach, N. Lachiche (1 999). Confirmation-Guided Discovery of first-order rules with Tertius. Machine Learning.4 2: 61-95

36. Стандарт TIA/EIA-569. Commercial Building Standard for Telecommunication Pathways and Spaces.

37. Стандарт ANSI/NECA/BICSI 568-2001. Installing Commercial Building Telecommunication Cabling.

38. Стандарт ANSI/TIA/EIA-607. Commercial Building Grounding and Bonding Requirements for Telecommunications.

39. СН 512-78. Інструкція з проектування будинків і приміщень для електронних обчислювальних машин. Будівельні норми.

40. РД 45.120-2000. Норми технологічного проектування.

41. Avarias, J. A., Lpez, J. S., Maureira, C., Sommer, H., and Chiozzi, G., “Introducing high performance distributed logging service for acs,” Proc. SPIE 7740, 77403G–77403G–10 (2010).

42. Reveco, J., Mora, M., Shen, T.-C., Soto, R., Sepulveda, J., and Ibsen, J., “Implementing kanban for agile process management within the alma software operations group,” Proc. SPIE 9152, 91521M–91521M–10 (2014).

43. Gil, J. P., Tejeda, A., Shen, T.-C., and Saez, N., “Unveiling alma software behavior using a decoupled log analysis framework,” Proc. SPIE 9152, 91521G–91521G–7 (2014).

44. Gormley, C. and Tong, Z., [Elasticsearch: The Definitive Guide], ” O’Reilly Media, Inc.” (2015).

45. Bagnasco, S., Berzano, D., Guarise, A., Lusso, S., Masera, M., and Vallero, S., “Towards monitoring-as-aservice for scientific computing cloud applications using the elasticsearch ecosystem,” in [Journal of Physics: Conference Series], 664(2), 022040, IOP Publishing (2015).

46. Andrade, P., Fiorini, B., Murphy, S., Pigueiras, L., and Santos, M., “Monitoring evolution at cern,” Journal of Physics: Conference Series 664(5), 052002 (2015).

47. Vinoski, S., “Advanced message queuing protocol,” IEEE Internet Computing (6), 87–89 (2006).

48. McCandless, M., Hatcher, E., and Gospodnetic, O., [Lucene in Action: Covers Apache Lucene 3.0], Manning Publications Co. (2010).

49. Shen, T.-C., Soto, R., Mora, M., Reveco, J., and Ibsen, J., “Alma operation support software and infrastructure,” Proc.

додатки

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Національна академія наук України
Університет імені П'єра і Марії Кюрі (Франція)
Маріборский університет (Словения)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Шяуляйська державна колегія (Литва)
Жешувський політехнічний університет ім. Лукасевича (Польща)
Білоруський національний технічний університет (Республіка Білорусь)
Міжнародний університет цивільної авіації (Марокко)
Національний університет біоресурсів і природокористування України (Україна)
Наукове товариство ім. Шевченка
ГО «Асоціація випускників Тернопільського національного технічного університету імені Івана Пулюя»

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник

тез доповідей

Том ІІ

ІХ Міжнародної науково-технічної конференції молодих учених та студентів

25-26 листопада 2020 року

УКРАЇНА
ТЕРНОПІЛЬ – 2020

УДК 004.415.5
А.О. Волоха, Л.П. Дмитроца, канд. техн. наук
Тернопільський національний технічний університет імені Івана Пулюя, Україна

МОНІТОРИНГ ТА АВТОМАТИЗАЦІЯ КЕРУВАННЯ СЕРВЕРАМИ В ВИСОКОНАВАНТАЖЕНИХ СИСТЕМАХ

A.O. Volokha, L.P. Dmytrotsa, Ph.D

MONITORING AND AUTOMATION OF SERVER CONTROL IN HIGHLY LOADED SYSTEMS

Під час роботи серверів у компаніях щодня створюється величезна кількість файлів журналів. Оскільки програмне забезпечення для збору інформації все ще перебуває в стадії постійного вдосконалення, журнали не тільки корисні для діагностики несправностей, виявлених під час роботи, але також необхідні для відображення довгострокового аналізу продуктивності та забезпечують швидкий огляд поведінки системи. Файли журналів зберігаються у форматі звичайного рядку й зберігаються у файловій системі серверу. Розробники та інженери служби підтримки - найпоширеніші користувачі цих журналів щодня. Залежно від проблеми діагностики, певні групи файлів завантажуються та перевіряються в текстовому редакторі за бажанням користувачів, що не дозволяє ефективно фільтрувати чи шукати інформацію. Є можливість покращити зручність використання і, крім того, використати цінну інформацію, доступну у файлах журналів. Розмір звичайної бази даних журналів вимірюється в терабайтах, що вимагає гнучкої інфраструктури, здатної обробляти великі дані, щоб відповісти на запитання різних зацікавлених сторін.

Основною задачею системи моніторингу та автоматизації керування є збереження актуальної інформації в кластері Elastic, далі ELK, бази даних та подальша обробка за допомогою машинного навчання на предмет аномалій в журналах.

При виявленні аномалій буде виконана одна з 3-х дій:

1. Автоматичне відправлення повідомлення відповідальній людині у випадку, якщо проблема типу Warning.

2. Виклик мікросервіса, який виведе сервер з кластера, повідомить відповідальну людину та вимкне його у випадку, якщо проблема типу Fatal або Error.

3. Повідомлення відповідальну людину, якщо нетипово багато, але не критичних записів одного типу Info та Debug.

Також в кластері ELK запропоновано будувати візуалізацію з більше ніж 6ТВ даних та отримувати аналітику використання та навантаження серверів, що дозволяє економити на закупках обладнання та покращити взаємодію користувача з сервісом.

Література

1. Gormley, C. and Tong, Z., [Elasticsearch: The Definitive Guide], ” O’Reilly Media, Inc.” (2015).

2. Avarias, J. A., Lpez, J. S., Maureira, C., Sommer, H., and Chiozzi, G., “Introducing high performance distributed logging service for acs,” Proc. SPIE 7740, 77403G–77403G–10 (2010).

3. Bagnasco, S., Berzano, D., Guarise, A., Lusso, S., Masera, M., and Vallero, S., “Towards monitoring-as-a- service for scientific computing cloud applications using the elasticsearch ecosystem,” in [Journal of Physics: Conference Series], 664(2), 022040, IOP Publishing (2015).

Додаток В

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

VIII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

9–10 грудня 2020 року

ТЕРНОПІЛЬ
2020

УДК 004.415.5
А.О. Волоха; Л.П. Дмитроца, к.т.н
(Тернопільський національний технічний університет імені Івана Пулюя)

РЕЗУЛЬТАТИ МОНІТОРИНГУ ТА АВТОМАТИЗАЦІЇ КЕРУВАННЯ СЕРВЕРАМИ В ВИСОКОНАВАНТАЖЕНИХ СИСТЕМАХ

UDC 004.415.5
A. Volokha, L. Dmytrotsa, Ph.D

RESULTS OF MONITORING AND AUTOMATION OF SERVER CONTROL IN HIGHLY LOADED SYSTEMS

Щодня створюється величезна кількість файлів журналів під час роботи серверів в компаніях, об'єм яких може сягати від декількох гігабайтів до сотень гігабайтів в день. Данні файли дуже великі та непридатні для аналізу людиною на наявність проблем чи аномалій. Стек Elastic, далі ELK, покращив чотири області:

1. Повторення розслідувань. Автоматизуючи виявлення нових випадків відомих проблем, було зменшено кількість дубльованих досліджень, тим самим підвищено ефективність діагностики. Це здійснюється шляхом створення запитів Elasticsearch, які визначають конкретну проблему та подання їх у інформаційну панель Kibana. Інформаційна панель, що показує випадки, що відбулися за останні 24 години відображається на великому екрані перед інженером підтримки. Таким чином, призначена особа може одним поглядом знати, що сталася відома проблема, і за допомогою повідомленої мітки часу, пов'яже це з проблемою, що нещодавно повідомлялася.
2. Висока мінливість. Навіть якщо ELK не може безпосередньо допомогти контролювати велику мінливість кількості виготовлених квитків, це ефективно допомагає швидко ідентифікувати квитки у відставанні: як тільки проблема виявляється за допомогою elasitcsearch пошукового запиту, запит виконується з пошуком екземплярів за попередні тижні або місяці. Тоді екземпляри можуть бути пов'язані зі старою, не дослідженою проблемою, про яку повідомлялося, за часом виникнення та опису.
3. Управління журналами. ELK забезпечує швидкий та універсальний спосіб пошуку та фільтрації журналів, завдяки підтримці Lucene API від Elasticsearch та зручному інтерфейсу користувача, наданому Kibana. Поточна архітектура дозволяє нам виконувати швидкі пошуки та фільтри протягом 6 місяців журналів, еквівалентних 6 ТБ, включаючи реплікацію. Однією з переваг цього є отримання за лічені секунди відповіді на запитання типу «Скільки разів ця проблема траплялася за останні місяці?»
4. Відсутність статистики та тенденційної інформації. У Кібані було створено кілька інформаційних панелей, які надають корисну інформацію: чітке уявлення про те, наскільки стабільною була попередній день, скільки спостережень було виконано, яке обладнання було використано; показує кількість журналів, вироблених програмними компонентами. Ненормальна кількість журналів, вироблених компонентом, є гарним показником проблеми. Він також відображає, скільки сигналів тривало в певні години.

Очікується в перспективі підключити стек ELK до Hadoop, Spark та інших рішень для великих даних для повного використання інформації, що міститься в журналах програмного забезпечення, для масштабованості та аналізу тенденцій, а також для характеристики бажаної та небажаної поведінки програмного забезпечення.

Література.

1. Gormley, C. and Tong, Z., [Elasticsearch: The Definitive Guide], " O'Reilly Media, Inc." (2015).
2. Avarias, J. A., Lpez, J. S., Maureira, C., Sommer, H., and Chiozzi, G., "Introducing high performance distributed logging service for acs," Proc. SPIE 7740, 77403G–77403G–10 (2010).