ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Автоматизація моніторингу та логування серверів з використанням ELK Stack»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 Комп'ютерної інженерії
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерна інженерія
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ *Максим БОНДАРЧУК*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. КІД-41
Максим БОНДАРЧУК
Ім'я, ПРІЗВИЩЕ
Керівник: Денис ЖУЖКОВ
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ
Рецензент:
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра 123 Комп'ютерної інженерії

Ступінь вищої освіти бакалавр

Спеціальність 123 Комп'ютерної інженерії

Освітньо-професійна програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
_____ Наталія ЛАЩЕВСЬКА
_____
«_____» _____ 2025р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Бондарчук Максим Олександрович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Автоматизація моніторингу та логування серверів з використанням ELK Stack
керівник кваліфікаційної роботи Денис ЖУЖКОВ,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025р. № 56
2. Строк подання кваліфікаційної роботи «3» червня 2024р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література; ELK Stack, Elasticsearh, Logstash, Kibana
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Поняття системи моніторингу та логування серверів, інструменти, які входять в ELK Stack
2. Опис інструментів, які використовуються в схемах логування та моніторингу сервера
3. Побудова схеми моніторингу та логування серверів з використанням ELK Stack
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання «24» лютого 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	25.02-24.03.2025	Виконано
2	Обробка матеріалу	23.03-05.04.2025	Виконано
3	Розробка теоретичної частини	06.04-10.04.2025	Виконано
5	Висновки за результатами аналізу	21.04-06.05.2025	Виконано
6	Оформлення пояснювальної записки	07.05-08.05.2025	Виконано
7	Розробка презентації	09.05-11.05.2025	Виконано
8	Передзахист	27.05-29.05.2025	Виконано
9	Здача в деканат	18.06-18.06.2025	Виконано

Здобувач вищої освіти _____ Максим БОНДАРЧУК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Денис ЖУЖКОВ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 38 стор., 5 рис., 1 табл., 30 джерел.

Мета роботи – Розробка системи автоматизації та логування серверів на базі технологій ELK Stack

Об'єкт дослідження – Система автоматизації та логування сервера на базі технологій ELK Stack

Предмет дослідження – Система автоматизації та логування сервера

Короткий зміст роботи: У наш час більшість бізнес-процесів тісно пов'язані з безперервною роботою серверів та наявністю актуальної інформації про їхній стан. Будь-який збій у роботі сервера може призвести до втрати даних, зниження продуктивності або фінансових збитків. Саме тому постає питання, як забезпечити ефективне стеження за роботою серверів та оперативне реагування на інциденти. Одним із рішень є використання стеку ELK (Elasticsearch, Logstash, Kibana), який дає змогу автоматизувати процес збору, обробки та візуалізації логів із різних джерел.

Для реалізації цього підходу було проаналізовано особливості логування та моніторингу серверів, обґрунтовано вибір ELK Stack, розроблено архітектуру рішення та налаштовано передачу логів на центральний сервер. На основі проведеного дослідження було доведено актуальність автоматизації моніторингу в сучасних умовах, розглянуто етапи впровадження системи та продемонстровано приклад її практичного використання в умовах IT-інфраструктури.

КЛЮЧОВІ СЛОВА: СТЕК ТЕХНОЛОГІЙ ELK, СЕРВЕР, ELASTIKSEARCH, LOGSTASH, KIBANA, МОНІТОРИНГ, ЛОГУВАННЯ

ЗМІСТ

ВСТУП…… 6
1.1 ПОНЯТТЯ МОНІТОРИНГУ ТА ЛОГУВАННЯ СЕРВЕРІВ…….. 7
1.1 Значення моніторингу в сучасній IT-інфраструктурі……..... 7
1.2 Основні цілі та завдання логування……..……. 9
1.3 Вступ до ELK Stack як комплексного рішення…… 11
2. ІНСТРУМЕНТИ ЯКІ ВИКОРИСТОВУЮТЬСЯ В СХЕМАХ ЛОГУВАННЯ…….…… 15
2.1 Класифікація інструментів за функціональністю…… 15
2.2 Огляд Elasticsearch, Logstash, Kibana…….……..... 17
2.3 Порівняння ELK Stack з альтернативними рішеннями ……. 21
3. РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ ТА ЛОГУВАННЯ СЕРВЕРІВ З ВИКОРИСТАННЯМ ELK STACK…….... 24
3.1 Архітектура рішення та принципи побудови …….……..... 24
3.2 Налаштування та конфігурація компонентів…..……. 27
3.3. Етапи розробки та впровадження проєкту……... 30
3.4 Тестування та оцінка ефективності системи ……... 39
ВИСНОВКИ…… 42
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ…….. 44
Додаток А. Презентація……..……..…….. 47

ВСТУП

У сучасному світі інформаційних технологій стабільна робота серверної інфраструктури є критично важливою для забезпечення безперервного функціонування бізнес-процесів, онлайн-сервісів та інформаційних систем. Зі зростанням масштабів IT-інфраструктури зростає і кількість подій, які необхідно контролювати: системні помилки, збої в роботі додатків, підозріла активність, проблеми з мережею тощо. Ручний моніторинг та аналіз логів у таких умовах стає малоефективним, затратним за часом та схильним до людських помилок. Саме тому виникає потреба в автоматизації процесів моніторингу та логування. Вона дозволяє оперативно виявляти й реагувати на критичні події, знижує ризики простоїв, підвищує безпеку, а також оптимізує ресурси обслуговування IT-систем. Автоматизовані інструменти також дають змогу системно зберігати, аналізувати та візуалізувати великі обсяги логів, що сприяє кращому розумінню роботи інфраструктури та швидшому вирішенню проблем. На допомогу прийде стек інструментів **ELK Stack** (Elasticsearch, Logstash, Kibana). Цей стек є потужним і широко використовуваним рішенням для централізованого збору, обробки та аналізу логів.Завдяки ELK Stack можливо в реальному часі відстежувати стан серверів, виявляти аномалії, знаходити причини збоїв та отримувати аналітичні звіти, що значно підвищує ефективність роботи системного адміністратора чи DevOps-інженера.

1. ПОНЯТТЯ МОНІТОРИНГУ ТА ЛОГУВАННЯ СЕРВЕРІВ

1.1 Значення моніторингу в сучасній інфраструктурі

У сучасну епоху стрімкого розвитку інформаційних технологій моніторинг IT-інфраструктури набув ключового значення для забезпечення стабільної та безперервної роботи інформаційних систем. Компанії приходять до висновку, що лише завдяки ефективному моніторингу можна запобігти значним фінансовим втратам, втраті даних або зниженню репутації через тривалі простої чи серйозні інциденти.

Моніторинг дозволяє у режимі реального часу виявляти проблеми, пов’язані з перевантаженням обладнання, відмовами у роботі сервісів, падінням мережевих з’єднань або помилками у програмному забезпеченні. Завдяки цьому адміністратори можуть оперативно реагувати на зміни в роботі систем, запобігати збоям та усувати загрози ще до того, як вони вплинуть на кінцевих користувачів. У багатьох випадках своєчасне виявлення проблеми дозволяє значно скоротити час відновлення працездатності інфраструктури, що безпосередньо впливає на загальну надійність сервісу.

Окрім цього, моніторинг сприяє глибшому розумінню того, як використовуються ресурси — такі як процесор, оперативна пам’ять, дисковий простір або пропускна здатність мережі. Аналіз цих показників дозволяє приймати обґрунтовані рішення щодо масштабування інфраструктури, оновлення обладнання або оптимізації налаштувань програмного забезпечення. Це, своєю чергою, допомагає уникати надлишкового навантаження, раціонально використовувати ресурси і зменшувати витрати на технічне забезпечення.

Моніторинг також має важливу роль у сфері інформаційної безпеки. Завдяки постійному збору логів, реєстрації спроб входу, відстеженню змін у конфігураціях та іншим параметрам можливо виявити спроби несанкціонованого доступу або

інші аномальні дії. На основі таких даних можна не лише зупинити атаку на ранньому етапі, але й провести ретельний аналіз інцидентів, виявити вразливості та зміцнити захист системи. Моніторингові інструменти, інтегровані з системами управління подіями безпеки (SIEM), дозволяють будувати комплексні рішення для активного реагування на загрози.

Важливо також, що моніторинг є складовою частиною нормативної відповідності у багатьох галузях. У фінансових, медичних, державних та інших критичних сферах існують вимоги щодо фіксації змін у системі, збереження логів, ведення аудиту доступу та реєстрації подій. Виконання цих вимог можливе лише за умови наявності надійної системи моніторингу, яка не лише фіксує події, а й забезпечує їхнє довгострокове зберігання та аналіз.

Окрім технічного боку, моніторинг дозволяє створювати візуальні дашборди, які дають змогу у реальному часі оцінювати стан систем. Це особливо важливо для аналітики, оперативного керування інфраструктурою та інформування керівництва про поточний стан сервісів. Інструменти візуалізації, такі як Kibana, надають доступ до даних у зручній, наочній формі, що полегшує прийняття рішень навіть для осіб, які не мають глибоких технічних знань.

Сучасні моніторингові рішення також підтримують автоматизовані сценарії реагування на інциденти. Це означає, що у разі виявлення критичних подій система може автоматично виконати певні дії: перезапустити службу, масштабувати додаток, повідомити відповідальних осіб або запустити скрипт для виправлення помилки. У хмарних середовищах та в умовах високої динаміки такі можливості є особливо цінними, оскільки дозволяють зменшити час реакції на проблеми до мінімуму.

Таким чином, моніторинг займає одне з найважливіших місць в IT. Він не лише забезпечує технічну стабільність та ефективність, а й сприяє безпеці, відповідності вимогам стандартів, розвитку цифрових продуктів та оптимізації витрат. У світі, де цифрові системи стають критично важливими для функціонування бізнесу, моніторинг перестає бути додатковою опцією і перетворюється на необхідну умову успішної діяльності будь-якої організації.

1.2 Основні цілі та завдання логування

У сучасних інформаційних системах логування відіграє ключову роль у забезпеченні стабільності, безпеки та надійної роботи IT-інфраструктури. *Логи* — це структуровані або напівструктуровані записи, які фіксують події, що відбуваються в операційних системах, програмному забезпеченні, мережевому обладнанні та інших елементах цифрового середовища. Вони слугують джерелом інформації про внутрішні процеси в системах, дозволяючи виявляти проблеми, аналізувати причини збоїв і оптимізувати роботу сервісів. Усі події, що мають значення для стабільності та безпеки системи, повинні бути належним чином зафіксовані для подальшого аналізу.

Однією з головних цілей логування є забезпечення постійного моніторингу систем. Адміністратори мають змогу переглядати журнали подій для того, щоб розуміти, як саме працює система, які помилки виникають, і на якому етапі. Це дозволяє не лише оперативно реагувати на критичні ситуації, але й вчасно виявляти приховані або поступові проблеми, які згодом можуть спричинити серйозні збої. Наприклад, накопичення помилок доступу до бази даних може свідчити про неправильну конфігурацію, яку варто виправити ще до того, як система вийде з ладу.

Не менш важливою функцією логування є забезпечення інформаційної безпеки. Системи логування дають змогу відслідковувати спроби несанкціонованого доступу, підозрілу активність користувачів, зміну критично важливих файлів або конфігурацій. Саме завдяки журналам безпеки можна виявити атаки, злам облікових записів або порушення внутрішніх політик доступу. У випадку інцидентів логи стають основним джерелом інформації для проведення розслідувань та формування доказової бази. У багатьох організаціях лог-файли є частиною комплексної системи захисту, яка охоплює не лише технічні засоби, але й організаційні заходи.

Ще однією суттєвою перевагою ведення логів є можливість проведення аудиту дій користувачів. У середовищі, де доступ до систем мають десятки або сотні працівників, дуже важливо знати, хто, коли і які саме дії виконував. Це дає змогу не тільки контролювати відповідність політикам безпеки, але й виявляти помилки або зловживання, які могли бути спричинені людським фактором. У разі виникнення спірних ситуацій журнал подій виступає джерелом об'єктивної інформації.

Крім того, логування активно використовується для аналізу продуктивності системи. Багато логів містять інформацію про навантаження на сервер, час відповіді, затримки в роботі служб або модулів. Збір таких даних дає змогу аналітикам і адміністраторам будувати моделі навантаження, виявляти вузькі місця в архітектурі та приймати рішення щодо масштабування або оптимізації ресурсів. Це особливо актуально для систем із високим навантаженням, де навіть незначне покращення продуктивності може суттєво вплинути на загальну ефективність.

Важливо також зазначити, що в багатьох сферах діяльності логування є обов’язковим відповідно до вимог законодавства або міжнародних стандартів. Наприклад, у банківському секторі, сфері охорони здоров’я чи в державних установах ведення журналів є необхідною умовою для проходження аудиту безпеки, підтвердження відповідності вимогам регуляторів або захисту від юридичної відповідальності. У цьому контексті лог-файли виступають як інструмент підтвердження прозорості та відповідальності у роботі систем.

Окрему увагу слід приділити ролі логування у процесах відновлення після збоїв. У разі аварії або виходу з ладу компонентів системи, журнали подій дозволяють точно встановити момент збою, його причину та послідовність подій, що призвели до інциденту. Це спрощує процес усунення наслідків та допомагає запобігти подібним ситуаціям у майбутньому. У поєднанні з системами резервного копіювання логування формує основу для побудови надійної стратегії безперервності бізнесу.

Нарешті, сучасні системи логування дедалі частіше інтегруються з автоматизованими засобами обробки даних, такими як ELK Stack, які дозволяють в реальному часі аналізувати великі обсяги логів, виявляти аномалії та генерувати сповіщення про критичні події. Це значно зменшує час реагування, знижує навантаження на персонал та дозволяє системам працювати більш автономно. Завдяки автоматизації логування перетворюється з пасивного інструменту в активний елемент керування IT-інфраструктурою, що має вирішальне значення в умовах зростаючої складності цифрових систем.

1.3 Вступ до ELK Stack як до комплексного рішення

Одним із найпоширеніших рішень у сфері моніторингу та логування є ELK Stack.

ELK Stack — це абревіатура, що об'єднує три ключові компоненти: Elasticsearch, Logstash та Kibana. Разом ці інструменти утворюють комплексну платформу для централізованого логування, що забезпечує зручне управління логами, швидкий пошук інформації та гнучкі можливості візуалізації. Кожен із компонентів виконує свою специфічну функцію, але найкраще вони працюють саме у зв’язці, утворюючи єдине інтегроване середовище.



Рисунок 1.1 – Компоненти ELK Stack

ELK Stack є незамінним інструментом для системних адміністраторів, DevOps-інженерів, розробників і технічної підтримки. Завдяки йому досягається прозорість у роботі системи, скорочується час виявлення та усунення проблем, зменшується ризик простоїв і зростає загальна надійність цифрових сервісів.*Elasticsearch* є основою стека, і відповідає за зберігання та індексацію даних. Це потужна розподілена пошукова та аналітична система, побудована на основі бібліотеки Apache Lucene. Її особливістю є здатність до обробки великих обсягів структурованої та неструктурованої інформації з мінімальними затримками, що забезпечує майже миттєвий пошук навіть у великих масивах логів. Elasticsearch підтримує складні запити, агрегації, фільтрацію даних та виконує обробку запитів у реальному часі, що є важливою перевагою для систем моніторингу.

Другий компонент — *Logstash* — виконує функцію збору, обробки та пересилання логів. Це інструмент із відкритим кодом, який підтримує сотні вхідних і вихідних плагінів, що дозволяє отримувати дані з різноманітних джерел: файлів, баз даних, системних журналів, мережевих потоків тощо. Logstash також дозволяє трансформувати дані — очищати, форматувати, фільтрувати або збагачувати їх перед передачею до Elasticsearch. Це робить систему гнучкою та придатною для роботи у середовищах з різними типами логів та структурою даних.

Kibana, у свою чергу, є веб-інтерфейсом для візуалізації даних, що зберігаються в Elasticsearch. Завдяки Kibana користувачі можуть будувати інтерактивні дашборди, графіки, діаграми та таблиці, що відображають поточний стан системи або історичні дані. Крім того, Kibana має вбудовану підтримку запитів до Elasticsearch, аналітики, а також можливості керування правами доступу, що дозволяє адаптувати інтерфейс під потреби різних категорій користувачів — від системних адміністраторів до бізнес-аналітиків.

У поєднанні ці три компоненти формують потужну платформу для централізованого логування, яка дозволяє здійснювати повноцінний контроль за всіма аспектами функціонування інформаційної системи. Завдяки ELK Stack організації можуть оперативно виявляти відхилення в роботі сервісів, знаходити

вузькі місця, проводити розслідування інцидентів безпеки, а також аналізувати поведінку користувачів або систем у довільному часовому проміжку. Усе це значно підвищує ефективність роботи IT-підрозділів.

Особливістю ELK Stack є його масштабованість та адаптивність. Система легко розгортається як у локальному середовищі, так і в хмарній інфраструктурі. Компоненти можуть бути інстальовані на одному сервері або розподілені між кількома вузлами, що дозволяє обробляти логи від сотень або навіть тисяч джерел одночасно. Крім того, стек може бути доповнений іншими інструментами, такими як Beats (легковагові агенти для збору логів і метрик), або інтегрований із системами сповіщення, автоматичного реагування та захисту, що перетворює його на універсальне рішення для забезпечення стабільної роботи IT-сервісів.

Починаючи з Elasticsearch, масштабування тут реалізовано за допомогою кластерної архітектури, де дані автоматично розподіляються між кількома вузлами (нодами). Кожен вузол виконує певну роль: може зберігати частину індексів виконувати запити або контролювати стан кластера Це дозволяє збільшити обсяг оброблюваних логів без втрати продуктивності. За потреби можна додавати нові вузли в кластер, що автоматично розширює його можливості зберігання й обробки даних. Elasticsearch також підтримує реплікацію, що забезпечує високу доступність і відмовостійкість системи.

У Elasticsearch є три основних варіанти налаштування вузлів:

1. *Головний вузол* (master node) — управляє усім кластером. Він обробляє зміни стану кластеру та передає ці зміни іншим вузлам. Головний вузол відповідає за здійснення важливих операцій на рівні кластеру, таких як створення, оновлення або видалення індексів.

2. *Вузол даних* (data node) — зберігає фактичні дані та зворотні індекси, необхідні для пошукових запитів. Це стандартний тип вузла, який використовується за замовчуванням. Вузли даних здійснюють операції з обробки, пошуку та агрегації даних.

3. *Клієнтський вузол* (client node) — виконує функцію балансувальника навантаження, розподіляючи вхідні запити серед інших вузлів кластеру. Такий

вузол не зберігає даних, а лише пересилає запити до відповідних вузлів для обробки.

Logstash також може бути масштабованим. Його екземпляри можна запускати паралельно для розподілу навантаження по збору та обробці логів із великої кількості джерел. Це особливо важливо для підприємств, де логи надходять з багатьох серверів, контейнерів, мікросервісів тощо. Крім того, в екосистемі Elastic з'являється все більше легких агентів (наприклад, Filebeat та Metricbeat), які виконують частину обробки на стороні джерела і передають попередньо оброблені дані до Logstash або безпосередньо до Elasticsearch. Це дозволяє ще ефективніше масштабувати систему, розвантажуючи центральні вузли.

Kibana, як фронтенд-інтерфейс, добре працює з великими обсягами даних завдяки оптимізації запитів та інтеграції з Elasticsearch. У великих розгортаннях можна використовувати декілька інстансів Kibana для балансування навантаження або забезпечення безперервного доступу до інтерфейсу аналітики для різних відділів і команд. Крім того, за допомогою рольового доступу та розмежування прав можна організувати ефективну роботу кількох команд одночасно в межах однієї системи.

Ще однією особливістю є те, що ELK Stack добре адаптується до хмарних середовищ. Рішення можуть розгортатися в Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure та інших платформах. Elastic навіть пропонує повністю керовану хмарну версію — Elastic Cloud, яка автоматично масштабується в залежності від навантаження та надає всі функції без потреби самостійно керувати інфраструктурою.

Усе це робить ELK Stack надзвичайно гнучким і масштабованим рішенням, яке може рости разом із потребами бізнесу. Від невеликої внутрішньої системи моніторингу до повноцінної корпоративної платформи — ELK Stack однаково добре працює у будь-якому масштабі, забезпечуючи надійність та гнучкість.

2. ІНСТРУМЕНТИ ЯКІ ВИКОРИСТОВУЮТЬСЯ В СХЕМАХ ЛОГУВАННЯ.

2.1 Класифікація інструментів за функціональністю

Компоненти ELK Stack забезпечують повний цикл обробки логів (рис 1.2). Logstash виконує збір даних з різних джерел, здійснює їх попередню обробку (фільтрацію, перетворення) та надсилає у відповідне сховище, після чого Elasticsearch приймає ці дані, індексує їх для ефективного зберігання та проводить аналітику, включно з повнотекстовим пошуком і агрегацією. Надалі зібрані дані можна візуалізувати за допомогою Kibana у вигляді дашбордів або графіків.

Відповідно до цих функцій можна класифікувати інструменти за такими критеріями:

— *Збір і обробка даних;*

— *Індексування та зберігання даних;*

— *Візуалізація та аналітика*

Збір і обробка даних. На цьому етапі відбувається збір логів з різних джерел: серверів, мережевих пристроїв, додатків, баз даних тощо. Основним інструментом для цього виступає Logstash, який може приймати дані з численних джерел, трансформувати їх (за допомогою фільтрів, регулярних виразів, умов тощо) і надсилати до Elasticsearch. Після того, як Logstash отримує дані з різних джерел, вони проходять через серію етапів обробки. Спочатку, за допомогою фільтрів, виконується очищення та нормалізація інформації: видаляються непотрібні або дублюючі записи, коригуються помилки форматування, а також здійснюється перетворення даних у єдиний формат. Далі за допомогою регулярних виразів або спеціальних умов, визначаються ключові параметри для подальшого аналізу. Це може бути, наприклад, визначення рівня важливості події, категорії помилки або часу її виникнення. Після обробки всієї цієї інформації Logstash передає

відфільтровані і трансформовані дані до Elasticsearch, де вони зберігаються для подальшого пошуку і аналізу. Таким чином, процес обробки та очищення даних дозволяє підготувати їх до ефективного використання в системі моніторингу і аналізу.

Індексування та зберігання даних. Після збору та обробки, дані зберігаються в централізованому сховищі для подальшого аналізу. Цю функцію виконує Elasticsearch, пошукова система, яка індексує вхідні дані, що дозволяє здійснювати швидкий повнотекстовий пошук, агрегації, фільтрацію та аналітику. Elasticsearch здатен працювати з великими обсягами даних і підтримує горизонтальне масштабування.

Візуалізація та аналітика. Кінцевий користувач отримує доступ до даних через графічний інтерфейс, який дозволяє зручно переглядати, фільтрувати та аналізувати інформацію за допомогою Kibana — веб-інтерфейсу, який забезпечує інтерактивну візуалізацію даних з Elasticsearch. Дає змогу створювати дашборди, графіки, гістограми, таблиці, а також використовувати механізми попередження та динамічних запитів.

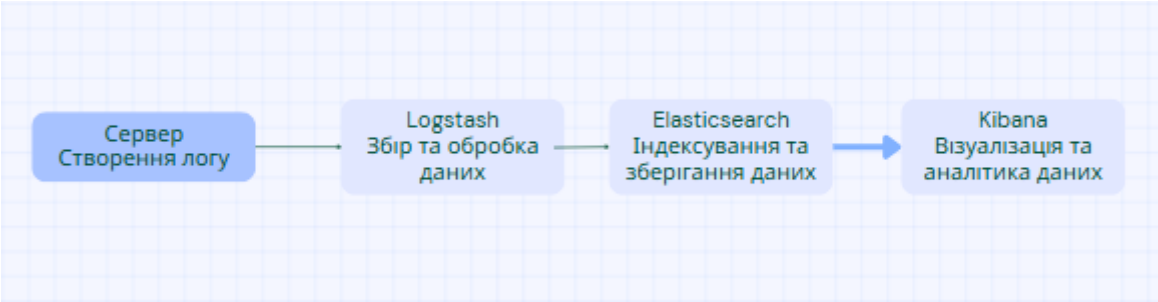


Рисунок 1.2 – Цикл обробки логів за допомогою ELK Stack

2.2 Огляд Elasticsearch, Logstash, Kibana

*ELK Stack (Elasticsearch, Logstash, Kiban*а) — це комплексне рішення для збору, обробки, зберігання та візуалізації даних, зокрема логів. Кожен з його компонентів виконує окрему роль, і їх узгоджена робота дозволяє будувати масштабовану систему моніторингу та аналізу інформації у режимі реального часу. *Elasticsearch* — це розподілена пошукова та аналітична система з відкритим кодом, створена на базі бібліотеки Apache Lucene. Основним завданням Elasticsearch є зберігання, індексація та забезпечення швидкого пошуку великих обсягів даних. На відміну від традиційних реляційних баз даних, Elasticsearch оперує документами у форматі JSON, що робить його зручним для роботи з напівструктурованими та неструктурованими даними.

Ключові характеристики Elasticsearch:

— *Розподіленість.* Elasticsearch може працювати як кластер з декількома вузлами, що забезпечує високу доступність та горизонтальне масштабування.

— *Індексація.* Усі вхідні документи проходять процес індексації, що дозволяє здійснювати повнотекстовий пошук із високою швидкістю.

— *Агрегації.* Система підтримує аналітичні запити, включаючи побудову статистичних звітів, агрегацій, фільтрацій тощо.

— *REST API.* Elasticsearch має зручний API, що дозволяє легко інтегрувати його з іншими системами.

Завдяки своїй продуктивності, Elasticsearch активно використовується в аналітичних системах, SIEM-рішеннях, платформах моніторингу та пошукових сервісах.

Elasticsearch зберігає дані на фізичному носії, використовуючи оптимізовані та стиснуті методи. Дані організовані в індекси, подібні то таблиці в реляційних базах даних, проте не мають схем, що не дозволяє зберігати документи з різною структурою в одному індексі. Організація даних включає наступні компоненти:

1. *шарди*: шарди є основними одиницями зберігання в Elasticsearch. Вони містять актуальні дані і розподілені між вузлами для забезпечення масштабованості та відмовостійкості;

2. *ноди*: кожен вузол у кластері Elasticsearch зберігає частину даних. Вузли відповідають за управління шардами і репліками;

3. *індекси*: індекс - це логічна колекція документів. Кожен індекс поділяється на шарди, які є фізичними екземплярами Apache Lucene.

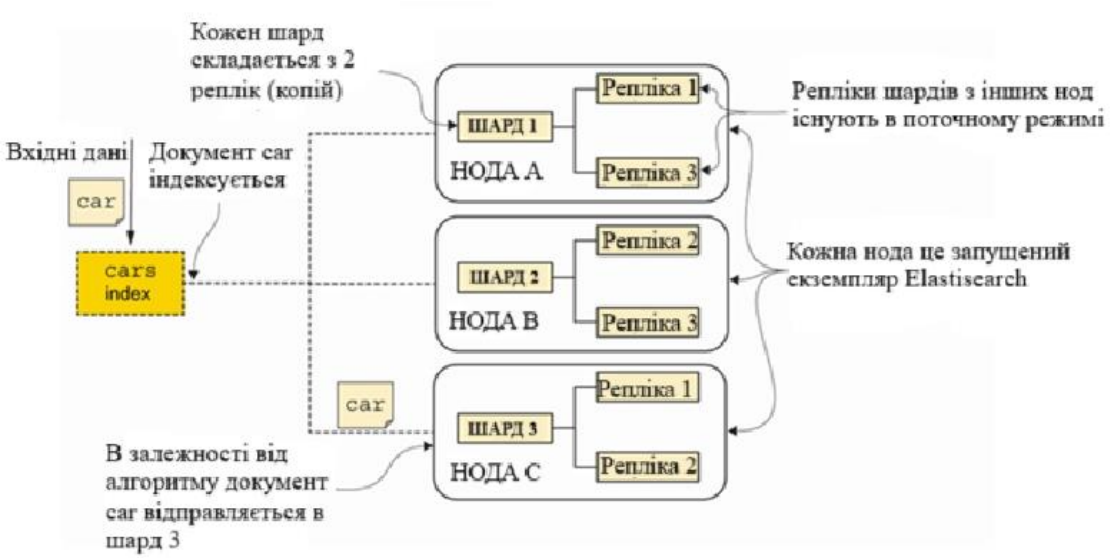


Рисунок 2.1 – Структура Elasticsearch

Elasticsearch отримує дані у форматі JSON і виконує процес, який називається мапуванням, щоб зпівставити отримані дані з типами даних Elasticsearch. Мапування визначає, як проіндексувати поля в документі. При цьому Elasticsearch має наступні типи даних:

Текст. Використовується для повнотекстового пошуку. Текстові поля проходять текстовий аналіз, який включає токенізацію і нормалізацію.

Ключові слова. використовується для точного збігу, наприклад, ідентифікаторів або тегів.

Дата. Використовується для зберігання значень дати і часу.

Числові значення. Використовується для числових даних, таких як цілі числа, числа з плаваючою комою та подвійні числа.

Після індексування тексту, Elasticsearch проводить аналіз тексту, розподіляючийого на окремі токени (токенізація) та додаючи до нього тексту метадані , наприклад синоніми, щоб зробити пошук легшим (рис 2.1)



Рисунок 2.2 – Процес аналізу тексту в Elasticsearch

Logstash — це гнучкий конвеєр для обробки логів і подій, який відповідає за збір даних із різних джерел, їх попередню обробку та передачу у Elasticsearch або інші системи зберігання. Logstash дозволяє працювати з даними у потоковому режимі та має модульну архітектуру, яка базується на трьох основних етапах: *input, filter* та *output.*

— *Input* — визначає джерела даних. Це можуть бути файли, мережеві потоки, бази даних, системні журнали, API-запити тощо.

— *Filter* — виконує обробку даних. За допомогою різноманітних фільтрів (grok, mutate, date, geoip тощо) Logstash може очищати, перетворювати, додавати або змінювати дані перед відправкою.

— *Output* — вказує, куди надсилати оброблені дані: у Elasticsearch, файл, базу даних або сторонній сервіс.

Logstash підтримує понад 200 плагінів, що дозволяє адаптувати його під будь-які джерела даних та сценарії. У поєднанні з легкими агентами Filebeat або Metricbeat, Logstash формує основу надійного конвеєра обробки даних.

Kibana — це веб-інтерфейс для візуалізації даних, які зберігаються в Elasticsearch. Його основне призначення — надання зручного способу для аналізу, перегляду та презентації інформації у вигляді графіків, таблиць, дашбордів та інших візуальних компонентів.

Основні функції Kibana:

— *Дашборди*. Користувачі можуть створювати інтерактивні панелі, які відображають ключові метрики, статуси систем, події тощо.

— *Пошук і фільтрація*. Можливість здійснювати гнучкий пошук по логах із використанням фільтрів, часових діапазонів і складних запитів.

— *Візуалізації*. Підтримка побудови лінійних графіків, гістограм, кругових діаграм, heatmap-карт тощо.

— *Alerts & Actions*. Механізми попередження, що дозволяють налаштовувати тригери для виявлення аномалій або помилок у системі.

— *Security & RBAC*. Контроль доступу до даних, можливість розмежування прав користувачів на рівні ролей.

Kibana також підтримує інтеграцію з іншими продуктами Elastic, такими як Elastic Security, APM (Application Performance Monitoring), Machine Learning тощо.

Взаємодія компонентів. Робота ELK Stack базується на послідовній взаємодії всіх трьох інструментів. Спочатку Logstash або Beats збирають дані з джерел і передають їх до Elasticsearch. Той, у свою чергу, індексує та зберігає ці дані, забезпечуючи можливість ефективного пошуку та аналітики. Нарешті, Kibana надає зручний інтерфейс для візуалізації інформації, що дозволяє користувачам

швидко реагувати на події, аналізувати тенденції та приймати обґрунтовані рішення.

2.3 Порівняння ELK Stack з альтернативними рішеннями

ELK Stack є одним із найпопулярніших рішень для збору, обробки та візуалізації логів, проте на ринку також існує низка альтернативних платформ, які вирішують подібні завдання. Найпопулярнішими з таких рішень є Splunk, Graylog, Fluentd з Grafana/InfluxDB, OpenSearch, а також комерційні хмарні сервіси, зокрема Datadog, Loggly чи New Relic.

Splunk — це платформа для аналізу машинних даних у режимі реального часу. Вона включає в себе модулі для збору логів, індексації, аналітики, побудови звітів і візуалізації.

Порівняння з ELK Stack:

Ціна. Splunk — платне рішення з ліцензією на обсяг індексованих даних, а ELK Stack — безкоштовне з відкритим кодом (Elastic має платні доповнення, але базовий функціонал — безкоштовний).

Функціональність. Splunk «з коробки» на відміну від ELK Stack має аналітичні функції, зокрема кореляцію подій, машинне навчання, SIEM-можливості.

Інтерфейс. Splunk пропонує більш інтуїтивний і зручний GUI для користувача.

Гнучкість. ELK Stack гнучкіший у налаштуванні та розгортанні, оскільки його компоненти можна комбінувати і масштабувати незалежно.

Graylog — ще одне популярне рішення з відкритим кодом, яке використовує Elasticsearch як бекенд для зберігання логів, а також має власний веб-інтерфейс для візуалізації.

Порівняння з ELK Stack:

Архітектура. Graylog інтегрує Elasticsearch, але має власний інтерфейс замість Kibana, а також використовує MongoDB для зберігання метаданих.

Простота розгортання. Graylog легше налаштовується для базових задач, але менш гнучкий у складних сценаріях.

Масштабованість. ELK Stack краще масштабується у великих інфраструктурах.

Підтримка. Graylog також має комерційну версію з розширеними функціями, схоже до Elastic.

Fluentd, InfluxDB та Grafana часто використовується як альтернатива ELK Stack для систем моніторингу, де важливі метрики і часові ряди.

Порівняння з ELK Stack**:**

Тип даних. Fluentd більше орієнтований на обробку структурованих логів і метрик; InfluxDB — база даних для часових рядів; Grafana — інструмент візуалізації.

Призначення. Це рішення більше підходить для DevOps-моніторингу, ніж для повного лог-аналізу.

Гнучкість. ELK Stack пропонує більш широкий спектр джерел і типів даних.

Складність налаштування. ELK Stack має більше плагінів «з коробки», але налаштування Fluentd + Grafana може бути простішим для невеликих систем.

OpenSearch — це форк (відгалуження) Elasticsearch і Kibana, створений Amazon після зміни ліцензії Elastic. Він залишається повністю відкритим і підтримується як спільнотою, так і AWS.

Порівняння з ELK Stack:

Ліцензія. Opensearch має повністю відкритий код (Apache 2.0), без комерційних обмежень.

Сумісність. Більшість API сумісні з Elasticsearch, що дозволяє легко мігрувати.

Підтримка. Менше офіційної документації, але активно підтримується AWS і open source-спільнотою.

Функціональність. OpenSearch включає схожий на Kibana інтерфейс — OpenSearch Dashboards — і має розширення для безпеки, аналізу, ML тощо.

Таблиця 2.1 — порівняння альтернативних рішень з ELK Stack

Платформа	Ліцензія	Тип рішення	Гнучкість	Масштабованість	Особливості
ELK Stack	Відкрита з додатковим платним функціоналом	Налаштовується самостійно	Висока	Висока	Велика спільнота, гнучка архітектура
Splunk	Пропрієтарна	Комерційна	Висока	Висока	Потужна SIEM,
Graylog	Відкрита/платна	Налаштовується самостійно	Середня	Середня	Простий у налаштуванні
Fluentd, Grafana	Відкрита	Комбіноване	Середня	Середня	Підходить для метрик
OpenSearch	Відкрита	Налаштовується самостійно	Висока	Висока	Повна сумісність з Elasticsearch

3. РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ ТА ЛОГУВАННЯ СЕРВЕРІВ З ВИКОРИСТАННЯМ ELK STACK

3.1 Архітектура рішення та принципи побудови

Архітектура ELK Stack побудована за модульним принципом, що дозволяє легко масштабувати систему та адаптувати її до потреб конкретної організації. Основу рішення становлять три ключові компоненти:

Elasticsearch — високопродуктивна пошукова та аналітична система, яка забезпечує зберігання даних у вигляді індексів і дає змогу швидко здійснювати пошук та агрегацію інформації. Завдяки розподіленій природі Elasticsearch легко масштабується та підтримує горизонтальне розширення.

Logstash — потужний конвеєр для збору, обробки та трансформації даних. Він здатний приймати інформацію з десятків джерел одночасно, фільтрувати, очищати та структурувати її за допомогою конфігураційних правил.

Kibana — інструмент для візуалізації даних, що зберігаються в Elasticsearch. Дає змогу створювати інтерактивні дашборди, графіки, таблиці та виконувати глибокий аналіз даних у реальному часі.

У стандартній архітектурі рішення log-файли та події з серверів, додатків і мережевих пристроїв спочатку передаються до Logstash або альтернативного агента, наприклад Filebeat. Filebeat — це легкий агент для передачі логів, який встановлюється на кожному сервері та відправляє логи безпосередньо до Logstash або Elasticsearch. Після обробки дані індексуються в Elasticsearch, звідки їх можна переглядати через Kibana.

Для забезпечення надійності та високої доступності система будується з урахуванням таких принципів:

Розподіленість. дані зберігаються у вигляді шардів (shards), які дублюються (репліки) на інші вузли, що гарантує цілісність і відмовостійкість.

Масштабованість. можна додавати нові вузли до кластеру Elasticsearch, щоб збільшити обсяг зберігання та швидкість обробки.

Безперервність. завдяки потоковій обробці та буферизації Logstash, навіть при пікових навантаженнях система здатна працювати без втрати даних.

Гнучкість інтеграцій. підтримка численних плагінів для Logstash, а також REST API в Elasticsearch роблять систему відкритою для взаємодії з зовнішніми сервісами.

Особливу увагу в архітектурі приділяється безпеці та обмеженню доступу. Elasticsearch підтримує механізми автентифікації, шифрування з'єднань та контроль доступу на основі ролей. Kibana також дозволяє налаштовувати права доступу до окремих візуалізацій, дашбордів або типів даних.

Завдяки своїй гнучкості та відкритості, ELK Stack може використовуватися як в невеликих компаніях, так і в масштабних корпоративних середовищах із великою кількістю серверів та джерел даних. Крім того, його можна розгорнути як на фізичній інфраструктурі, так і в хмарних середовищах, що робить систему універсальною для більшості сценаріїв використання.

Ключовою перевагою архітектури ELK Stack є її гнучкість у побудові відповідно до конкретних вимог інфраструктури. Наприклад, у невеликих проєктах можливо реалізувати однонодовий кластер, у якому всі три компоненти (Elasticsearch, Logstash та Kibana) розміщуються на одному сервері. Такий підхід значно спрощує розгортання та зменшує витрати. Однак для більшої надійності й масштабування у виробничих середовищах зазвичай використовують багатоекземплярні (multi-node) кластери, де ролі розподіляються між окремими вузлами.

В архітектурі ELK Stack важливим є поняття вузлів (нодів) та ролей. Elasticsearch має кілька типів вузлів, які виконують різні функції:

Майстер-вузол (Master node) — відповідає за управління кластером, створення та видалення індексів, балансування навантаження, контроль за доступністю інших нодів.

Дата-вузол (Data node) — зберігає індекси та обробляє запити, пов'язані з даними (запис, читання, агрегація).

Вузол-клієнт (Coordinating node або Ingest node) — приймає запити від клієнтів, передає їх на відповідні дата-вузли, а також може виконувати попередню обробку даних перед індексацією.

Такий підхід до побудови дозволяє масштабувати кожну роль окремо: наприклад, якщо система стикається з великим обсягом даних — можна додати більше дата-вузлів, а якщо зростає кількість користувачів Kibana — варто масштабувати вузли координації.

Ще одним важливим аспектом є обробка даних до моменту їхнього потрапляння до Elasticsearch. Для цього використовується Logstash pipeline (магістраль), яка складається з трьох основних етапів: input, filter та output. На етапі input Logstash приймає дані з джерел, таких як Filebeat, syslog, Kafka, API-запити або файли журналів. Далі, на етапі filter, застосовуються правила для обробки — наприклад, розбиття повідомлень на поля, очищення від зайвого, додавання тегів, геолокації, аналіз IP-адрес, а також парсинг повідомлень за шаблонами. На завершальному етапі output дані передаються до Elasticsearch або інших цільових систем.

Важливо, що Logstash підтримує модульну структуру конфігурації — кожна магістраль може бути незалежною і обслуговувати окремі джерела або типи даних. Це дозволяє будувати складні сценарії обробки, наприклад, окрема магістраль для логів веб-сервера, ще одна — для системних подій, і ще одна — для подій безпеки.

Окрему увагу слід приділити компоненту Filebeat, який часто використовується як агент збору логів безпосередньо на серверах. Filebeat споживає мінімум ресурсів, має вбудований механізм повторної доставки (у разі

збоїв), і дозволяє забезпечити децентралізований, але контрольований збір даних. Filebeat має модулі для різних сервісів, таких як Apache, Nginx, MySQL, Docker тощо, що значно спрощує налаштування.

На рівні візуалізації важливим є правильне планування дашбордів у Kibana, які повинні бути інформативними, адаптивними та відповідати потребам користувачів: системних адміністраторів, DevOps-команд, відділу інформаційної безпеки тощо. Завдяки можливості створення різноманітних типів візуалізацій — гістограм, кругових діаграм, лінійних графіків, таблиць тощо — Kibana дозволяє оперативно аналізувати ситуацію, знаходити аномалії, фільтрувати дані та створювати звіти.

Крім того, у Kibana можливо налаштувати сповіщення на основі певних умов. Наприклад, при виявленні великої кількості помилок певного типу, системі можна доручити автоматично повідомляти відповідальних осіб за допомогою електронної пошти, Slack або іншими каналами. Це значно підвищує оперативність реагування на інциденти.

Ще одним важливим архітектурним аспектом є інтеграція ELK Stack із зовнішніми системами. Через REST API можна взаємодіяти з іншими сервісами: системами резервного копіювання, SIEM-рішеннями, платформами аналітики та штучного інтелекту. Також існують можливості побудови систем на базі контейнерів (Docker, Kubernetes), що відкриває шлях до автоматизованого масштабування та деплойменту.

3.2 Налаштування та конфігурація компонентів

Успішне розгортання системи моніторингу та логування за допомогою ELK Stack передбачає належне налаштування кожного з її основних компонентів: Elasticsearch, Logstash, Kibana та, за необхідності, додаткових інструментів, таких як Filebeat. Цей процес включає встановлення програмного забезпечення,

початкову конфігурацію, а також оптимізацію параметрів відповідно до потреб проєкту.

Встановлення Elasticsearch. Elasticsearch — це основний компонент, відповідальний за зберігання, індексацію та обробку логів. Встановлення можливе як у вигляді окремих бінарних файлів, так і через пакетні менеджери (наприклад, APT або YUM) або за допомогою Docker-контейнерів. Після встановлення необхідно змінити основні параметри у конфігураційному файлі elasticsearch.yml, зокрема:

— cluster.name — назва кластеру;

— node.name — унікальна назва вузла;

— network.host — IP-адреса, до якої прив’язується сервіс;

— http.port — порт, на якому працює Elasticsearch (типово 9200);

— discovery.seed_hosts та cluster.initial_master_nodes — параметри, необхідні для запуску кластеру з кількох вузлів.

Для забезпечення стабільності та захисту даних також налаштовується політика керування пам’яттю JVM, резервне копіювання (snapshot), обмеження доступу через брандмауер або проксі-сервер.

Logstash виконує роль центрального обробника даних. Основна конфігурація здійснюється через .conf файли, які описують логіку обробки в термінах трьох блоків: input, filter та output. Наприклад, типовий конфігураційний файл може виглядати так (Рис. 3.1):

```
input {
  beats {
    port => 5044
  }
}
filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}" }
  }
  date {
    match => ["timestamp", "dd/MMM/yyyy:HH:mm:ss Z"]
  }
}
output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "logs-%{+YYYY.MM.dd}"
  }
}
```

Рисунок 3.1 – Типовий конфігураційний файл налаштування Logstash

У цьому прикладі Logstash приймає дані від Filebeat через порт 5044, обробляє їх за допомогою шаблону grok, нормалізує дату та надсилає до Elasticsearch з прив’язкою до щоденного індексу.

Logstash має велику кількість плагінів, які дозволяють зчитувати логи з файлів, баз даних, системних журналів, мережевих протоколів, API тощо. Важливим етапом налаштування є відлагодження pipeline для уникнення втрат даних, дублювання чи затримок.

Після налаштування Elasticsearch можна переходити до встановлення та конфігурації Kibana. Kibana надає графічний інтерфейс до Elasticsearch, тому у файлі kibana.yml потрібно вказувати адресу Elasticsearch. Після запуску Kibana стає доступною через веббраузер за замовчуванням на порту 5601. На цьому етапі адміністратор створює індексні шаблони (index patterns), які відповідають логам, надісланим Logstash або Filebeat, після чого стає можливим створення дашбордів, візуалізацій і запуск аналітичних запитів.

Kibana також дозволяє налаштовувати сповіщення, ролі та доступи, панель моніторингу кластера та додаткові плагіни для інтеграції з Grafana, Slack, Teams, тощо.

*Налаштування Filebeat***.** Filebeat — це агент, який встановлюється безпосередньо на серверах або пристроях, з яких потрібно отримувати логи. Filebeat має легку вагу, невелике споживання ресурсів та гнучку структуру конфігурації.

Також Filebeat підтримує модулі — наперед налаштовані шаблони для популярних сервісів (наприклад, nginx, apache, systemd), які автоматично додають правила парсингу та візуалізації у Kibana.

Перевірка коректності роботи. Після завершення конфігурації всіх компонентів робимо перевірку з'єднання за допомогою запиту до http://localhost:9200, який має повернути JSON-відповідь від Elasticsearch;

Logstash має приймати та передавати дані, про що свідчать логи;

Kibana має відображати оновлення індексів і логи у реальному часі;

Filebeat повинен повідомляти про успішну доставку даних.

Таким чином, правильне налаштування всіх компонентів забезпечує надійну та масштабовану основу для подальшого впровадження повноцінної системи моніторингу та логування.

3.3 Етапи розробки та впровадження проєкту

Впровадження системи моніторингу та логування з використанням ELK Stack — це складний процес, що потребує системного підходу, чіткої поетапної реалізації, налаштування компонентів і перевірки їх взаємодії. Для досягнення цілей проєкту реалізація проходить за кількома етапами.

Першим і одним із найважливіших етапів у процесі розробки системи моніторингу та логування серверів із використанням ELK Stack є ретельний аналіз потреб майбутньої системи. Цей етап має на меті зрозуміти, які саме проблеми має вирішувати система, якими є очікування замовника або адміністратора, і як забезпечити найвищу ефективність рішення в умовах конкретної IT-інфраструктури. Розглянемо визначення цілей моніторингу.

Насамперед необхідно визначити, що саме повинно моніторитися. В цей пункт можуть входити такі логи, як:

— Системні логи операційних систем (Linux, Windows);

— Мережеві логи (файрволи, маршрутизатори, проксі-сервери);

— Логи додатків (веб-сервери, CRM-системи, API);

— Бази даних (MySQL, PostgreSQL, MongoDB тощо);

— Аномалії та події безпеки (невдалі входи, спроби SQL-ін’єкцій, DDOS-атаки).

Цілі можуть бути різні, від виявлення помилок у роботі сервісів до збору статистики по роботі системи, аналізу продуктивності, безпеки, доступності тощо. Після визначення цілей моніторингу потрібно вибрати джерела даних.

Кожне джерело даних повинно бути задокументоване. Для всіх визначаються:

— Тип логів, які воно генерує;

— Формат записів;

— Частота надходження логів;

— Критичність інформації, що міститься;

— Метод підключення до системи логування.

На цьому етапі також визначається кількість джерел, їх розподіл по серверах, а також — потенційні точки збою.

Наступним пунктом потрібно написати вимоги до зберігання та обробки логів. Важливо оцінити середній об’єм логів на добу, пікове навантаження та максимальний період зберігання. Це впливає на конфігурацію Elasticsearch, кількість шардів та реплік, тип зберігання. Також будуть задіяні параметри

Logstash, такі як черги, буферизація. Відповідно потрібно буде визначати куди саме архівувати старі дані та чи потрібна буде резервна копія.

Також на цьому етапі оцінюються вимоги до продуктивності — з якою швидкістю мають оброблятися та відображатися нові події в Kibana.

Не менш важливо визначити ролі користувачів системи. Виділимо такі ролі як:

— Адміністратори, що налаштовують і обслуговують систему;

— Інженери техпідтримки, які реагують на інциденти та допомагають користувачам;

— Аналітики безпеки, які аналізують спроби атак;

— Розробники, які перевіряють логи додатків.

Для кожної ролі встановлюємо рівні доступу в Kibana, можливості перегляду, редагування або експорту логів.

На основі зібраної інформації формується чітке технічне завдання, яке описує:

— Цілі системи та вимоги;

— Перелік джерел логів;

— Архітектурні вимоги до компонентів ELK;

— Рівні доступу та безпеки;

— Очікувані навантаження та обсяги збереження;

— KPI для оцінки ефективності після впровадження.

Цей документ є основою для подальших етапів — проєктування, конфігурації, тестування та розгортання.

Після визначення вимог і постановки технічного завдання наступним етапом є проєктування архітектури системи моніторингу та логування. Правильно побудована архітектура забезпечує надійність, масштабованість та ефективну роботу всієї системи, а також спрощує подальшу експлуатацію та обслуговування. Система логування та моніторингу на основі ELK Stack будується як розподілена система, що складається з кількох логічних рівнів.

Агенти збору логів (Filebeat). Встановлюються на кожному джерелі логів (сервери, хости, додатки), де відповідають за читання файлів журналів і передачу даних до Logstash або безпосередньо в Elasticsearch.

Logstash. Проміжна ланка, що приймає дані від агентів, виконує трансформацію, нормалізацію, фільтрацію та обробку подій.

Elasticsearch. Основний компонент зберігання, обробки та пошуку логів. У цьому модулі дані індексуються та стають доступними для подальшого аналізу.

Kibana. Інтерфейс користувача для візуалізації логів, побудови дашбордів, пошуку інформації, створення графіків та аналітики.

Кожен із цих компонентів може бути розміщений на окремих фізичних або віртуальних машинах, що дає змогу масштабувати систему відповідно до потреб.

Наступним пунктом розглянемо вибір типу архітектури. Залежно від обсягу логів, кількості джерел і вимог до надійності, можна вибрати один із двох типів архітектур: централізованої та розподіленої (кластерної).

Централізована — усі агенти надсилають логи в єдиний вузол Logstash і далі в Elasticsearch. Така схема проста, але має один вузол відмови.

Розподілена — декілька екземплярів Logstash і Elasticsearch працюють у кластері з балансуванням навантаження та резервуванням. Забезпечує високу доступність і масштабованість.

Логічна схема взаємодії компонентів виглядатиме так:

Filebeat та Logstash (TCP/SSL або HTTP) — передача логів у реальному часі;

Logstash та Elasticsearch (REST API) — обробка й індексація даних;

Elasticsearch та Kibana (REST API) — візуалізація результатів через веб-інтерфейс.

Також можлива схема Filebeat та Elasticsearch (в такому випадку ми обходимо Logstash), але тоді втрачається гнучкість трансформації даних.

На етапі проєктування врахуємо механізми захисту даних:

— Шифрування трафіку між компонентами (TLS/SSL);

— Аутентифікація доступу (API-токени, логін/пароль, інтеграція з LDAP);

— Авторизація (визначення ролей користувачів у Kibana та Elasticsearch);

— Захист API від несанкціонованого використання (обмеження за IP, rate limiting).

Також разом із захистом від несанкціонованого доступу важливо розглянути пункт щоб уникнути втрати даних і забезпечити безперервність роботи. Для цього в кожному компоненті є доступні інструменти.

Elasticsearch налаштовується з кількома shard і replica, щоб зберігати копії індексів;

Logstash може використовувати черги (наприклад, persistent queues або брокери Kafka) для тимчасового зберігання логів при проблемах з Elasticsearch.
Не менш важливим за безпеку пунктом буде планування ресурсів. Розробка плану апаратного забезпечення залежить від обсягу логів в день (зазвчиай вимірюється в гігабайтах), кількості індексів та запитів до них, кількості та середнього розміру подій і відповідно кількості одночасних користувачів веб інтерфейсу Kibana.
Розподіл навантаження забезпечується балансувальниками (наприклад, Nginx, HAProxy) та розгортанням декількох екземплярів кожного компонента.

Після проєктування архітектури наступним важливим етапом є розгортання системи в тестовому середовищі. Цей етап дозволяє перевірити, чи відповідає система вимогам, оцінити її працездатність, протестувати налаштування компонентів, а також виявити потенційні проблеми, перш ніж система буде впроваджена в продуктивне середовище. Перед тим, як приступити до розгортання, потрібно підготувати тестове середовище, яке максимально відповідає умовам реальної експлуатації:

Віртуалізація або фізичне обладнання. Тестове середовище може бути налаштоване як у віртуальних машинах (наприклад, за допомогою VMware або VirtualBox), так і на фізичних серверах. Віртуалізація дозволяє швидко масштабувати середовище, а також зменшує витрати на апаратне забезпечення.

Мережеві налаштування. Важливо налаштувати мережу так, щоб компоненти ELK Stack могли взаємодіяти один з одним через безпечні канали зв'язку. Це включає налаштування портів, фаєрволів, VPN-каналів, якщо необхідно.

Інсталювання та налаштування компонентів. На тестовому середовищі відбувається інсталяція всіх основних компонентів ELK Stack, налаштування кількості шардів і реплік, обсягів пам'яті, параметрів пошукової індексації, налаштування конфігураційних файлів для збору та обробки логів, визначення вхідних і вихідних плагінів, а також налаштування інтерфейсу для візуалізації даних, створення базових дашбордів для моніторингу.

Після підготовки тестового середовища можна розглянути ініціалізацію та запуск компонентів Першим запускається Elasticsearch. Важливо перевірити, чи з'являються індекси та чи можуть надходити запити до сервера. Це дозволяє переконатися, що система індексації працює коректно. Далі потрібно налаштувати Logstash для прийому даних з джерел. У тестовому середовищі можна використовувати кілька типових сценаріїв для передачі логів, таких як з локальних файлових систем або з віддалених серверів через syslog. Після цього перевіряється коректність обробки і трансформації логів за допомогою фільтрів і регулярних виразів. Наступною налаштовується Kibana, створюються базові візуалізації для тестових даних (графіки, таблиці, дашборди), що дають можливість оцінити інтерфейс і зрозуміти, як зручніше працювати з системою моніторингу.

По завершенню ініціалізації можна перейти до тестування процесу збору та обробки даних. Сюди можна включити такі дії як:

Тестування збирання даних. Проводиться тестування механізмів збору логів з різних джерел. Для цього може бути налаштовано кілька серверів, що генерують логи різного типу (наприклад, веб-сервери Apache, бази даних, аплікаційні логи). Перевіряється, чи всі логи успішно передаються до Logstash та чи правильно вони обробляються і відправляються до Elasticsearch.

Перевірка трансформації даних. Один з важливих аспектів — це перевірка трансформації та фільтрації даних. Це включає перевірку того, чи правильно

працюють налаштування фільтрів, чи всі поля коректно індексуються і чи не відбувається втрати даних.

Масштабування навантаження. Для перевірки масштабованості можна згенерувати велику кількість тестових логів або підключити додаткові джерела. Перевіряється, чи система здатна обробляти великі обсяги даних, без збоїв у роботі чи значних затримок.

Після даних тестів можна оцінити стабільність та продуктивність встановленої системи**.** Важливо протестувати, як система працює під навантаженням, як швидко обробляються запити в Kibana і чи немає затримок при виведенні результатів. Тестується доступ до компонентів ELK Stack, перевіряються політики безпеки, шифрування трафіку між компонентами, а також коректність налаштувань аутентифікації та авторизації для різних ролей користувачів. Проводяться тести на відмовостійкість: зупинка одного з компонентів (наприклад, Elasticsearch або Logstash) і перевірка, чи система працює належним чином (збереження працездатності, збереження даних і коректність відновлення після збою).

Після завершення тестів можна скласти звіт по результатах, який допоможе краще зрозуміти, чи все працює. Варто включити

— Оцінку відповідності системи вимогам;

— Виявлені проблеми та рекомендації щодо їх усунення;

— Пропозиції щодо оптимізації продуктивності, масштабування і безпеки.

На етапі впровадження системи моніторингу та логування важливо не лише налаштувати програмне забезпечення та забезпечити його стабільну роботу, а й підготувати персонал до ефективного використання нової системи. Адже лише належне знання можливостей ELK Stack, правильне налаштування та оперативне реагування на інциденти дозволить отримати максимальну вигоду від системи.

Після успішного тестування та система готова до впровадження в продуктивне середовище. Це етап, коли розгортання відбувається на існуючому сервері та починає обробляти справжні дані, які генеруються в процесі роботи компанії.

Впровадження в продуктивне середовище вимагає особливої уваги, адже кожна помилка на цьому етапі може призвести до серйозних збоїв у роботі компанії.

Для того, щоб запезбечити ефективну підготовку до впровадження, варто перевірити наступні пункти:

— *Перевірка всіх налаштувань*. Переконатися, що всі компоненти ELK Stack налаштовані відповідно до вимог продуктивного середовища. Це включає налаштування параметрів продуктивності Elasticsearch, параметрів обробки логів в Logstash та налаштування доступу до Kibana.

— *Створення плану впровадження*. Створення чіткого плану для впровадження системи, який включає етапи запуску, перевірки працездатності, моніторингу та налаштування резервування на всіх етапах. В плані має бути передбачено час на реагування на можливі інциденти, а також механізм відновлення роботи в разі неполадок.

— *Резервування даних*. Важливо налаштувати механізми резервного копіювання для всіх даних, зокрема для індексів Elasticsearch. Це гарантує, що у разі збою або втрати даних можна швидко відновити нормальну роботу системи.

— *Оцінка навантаження та масштабування*. Перед впровадженням необхідно переконатися, що система здатна обробляти необхідні обсяги даних. Для цього проводяться тести під навантаженням для виявлення можливих вузьких місць у продуктивності, а також налаштування механізмів масштабування в Elasticsearch та Logstash.

З метою мінімізації ризиків під час впровадження системи виявлено, що краще застосовувати поетапний підхід. Початкове розгортання здійснюємо на обмеженому наборі серверів або в тестовому середовищі. Такий підхід дає змогу перевірити працездатність системи на реальних даних без створення загрози для всієї IT-інфраструктури.

Після інтеграції системи в продуктивне середовище необхідно здійснити перевірку коректності збору та обробки логів з усіх компонентів інфраструктури. Система має забезпечувати стабільну роботу за умов реального навантаження, демонструючи високу надійність і продуктивність.

Особливу увагу приділяємо налаштуванню механізмів сповіщень та автоматичних оповіщень (алертів) у Kibana. Це дає змогу відповідальним фахівцям оперативно отримувати інформацію про критичні події, такі як перевантаження серверів або невідповідності між індексами, що дозволяє вчасно реагувати на можливі інциденти.

Упродовж усього процесу впровадження необхідно здійснювати постійний моніторинг стану системи. Можемо провести детальний аудит роботи компонентів, аналіз журналів подій, а також контроль за споживанням ресурсів, зокрема процесора, оперативної пам’яті та дискового простору.

Після успішного впровадження система переходить до постпродакшн етапу, основним завданням якого є забезпечення стабільної експлуатації та підтримки.

На цьому етапі організовуємо постійний моніторинг роботи системи та оперативне реагування на можливі інциденти. Важливо, щоб відповідальний персонал мав чітко визначені інструкції та алгоритми дій у разі виникнення аварійних ситуацій або збоїв, що дозволить мінімізувати ризики простоїв та втрати даних.

Також слід передбачити регулярне оновлення компонентів ELK Stack та їх оптимізацію з метою підтримання високої продуктивності. Доцільно періодично коригувати налаштування індексації, фільтрації та візуалізації даних відповідно до отриманих результатів моніторингу та змін у робочому середовищі.

Через кілька місяців експлуатації можемо провести додатковий детальний аналіз ефективності функціонування системи. Зокрема, необхідно оцінити здатність системи до виявлення аномалій, швидкість надання критично важливої інформації, а також вплив на скорочення часу реагування на інциденти.

Важливо також розглянути ризики, тому що під час впровадження системи ми стикаємося з низкою потенційних проблем, які обов’язково враховуємо для забезпечення її стабільної та ефективної роботи.

Одним із ключових викликів є можливе перевантаження системи під час роботи з реальними даними. Існує ймовірність, що система не витримає очікуваного навантаження, що може призвести до уповільнення її роботи або навіть збоїв. Щоб цього уникнути, ми заздалегідь проводимо навантажувальне

тестування та налаштовуємо масштабування Elasticsearch відповідно до прогнозованих обсягів даних. Також звертаємо особливу увагу на правильну конфігурацію всіх компонентів ELK Stack. На етапі впровадження можуть виникнути труднощі, особливо під час інтеграції з іншими системами або за відсутності чіткого плану розподілу навантаження. Тому ретельна перевірка параметрів конфігурації та проведення попереднього тестування є важливим пунктом. Крім того, враховуємо можливість змін в IT-інфраструктурі. У разі додавання нових серверів або сервісів ми своєчасно коригуємо налаштування системи для забезпечення коректної обробки логів з нових джерел та збереження цілісності даних.

3.4 **Тестування та оцінка ефективності системи**

*Підтримка та обслуговування системи після впровадження***.** Після того як система моніторингу та логування ELK Stack була успішно впроваджена в продуктивне середовище, важливо забезпечити її постійну підтримку та обслуговування. Цей етап є критичним для того, щоб система працювала стабільно та ефективно на протязі всього життєвого циклу.

Щоб гарантувати швидке реагування, ми налаштовуємо систему автоматичних сповіщень та алертів, які попереджають відповідальних осіб про критичні події, наприклад, про перевантаження серверів, помилки при обробці логів або нестачу вільного місця. Це дозволяє мінімізувати час реакції на інциденти та забезпечує безперервну роботу системи.

Крім технічних показників, ми також інтегруємо систему з інструментами безпеки для виявлення аномальних активностей, таких як спроби несанкціонованого доступу, підозрілі дії користувачів або збої в автентифікації. Усе це дозволяє нам своєчасно реагувати на загрози інформаційній безпеці та підтримувати високий рівень захищеності інфраструктури.

Не менш важливим є регулярне оновлення всіх компоненти ELK Stack для забезпечення їх актуальності, сумісності та відповідності новим вимогам безпеки. Це стосується як основних елементів системи, так і додаткових плагінів чи бібліотек, які використовуються для розширення функціоналу. З метою зниження ризику експлуатації вразливостей ми своєчасно застосовуємо всі доступні патчі та оновлення безпеки. Особливу увагу ми приділяємо вразливим модулям, які працюють із зовнішніми даними, оскільки саме вони найчастіше є мішенню для атак. Щоб уникнути людських помилок під час оновлення та зменшити час простою системи, впроваджуємо автоматизацію процесів оновлення. Це дозволяє здійснювати оновлення без втрати даних і з мінімальним впливом на працездатність сервісів.

У процесі експлуатації система накопичує велику кількість логів, що призводить до стрімкого зростання кількості індексів у Elasticsearch. Ми впроваджуємо політики управління життєвим циклом індексів, які передбачають автоматичну архівацію старих даних, видалення застарілих або непотрібних індексів, а також оптимізацію активних. Для зниження навантаження на систему ми переносимо неактуальні дані у холодне сховище або архівуємо їх у зовнішні джерела, такі як Amazon S3, HDFS чи інші об'єкти зберігання. Це дозволяє зберігати історичну інформацію без шкоди для продуктивності. Ми також регулярно переглядаємо політики зберігання з урахуванням зростання обсягів даних, змін в інфраструктурі та актуальних нормативних вимог щодо збереження інформації. Це дає змогу адаптувати систему до нових умов і забезпечувати її відповідність вимогам аудиту та аналізу.

Для ефективної роботи користувачів з системою ми забезпечуємо технічну підтримку, яка допомагає при створенні дашбордів, налаштуванні пошукових запитів, роботі з фільтрами та візуалізаціями в Kibana. Наша команда надає оперативні консультації та сприяє ефективному використанню можливостей платформи.

Також не лишнім буде організовувати навчальні сесії та періодичне оновлення знань для користувачів та адміністраторів, зокрема у зв’язку з появою

нових функцій, змін у структурі інтерфейсу або розширенням системи. Це дозволяє зберігати високий рівень компетентності персоналу та уникати помилок у використанні аналітичних інструментів.

Крім цього, ми аналізуємо зворотний зв’язок від користувачів, щоб вдосконалювати дашборди, налаштування візуалізацій та інтерфейс у цілому. Це сприяє підвищенню ефективності роботи з даними й покращує загальний користувацький досвід.

Приділяємо особливу увагу створенню плану відновлення після збоїв, який передбачає наявність регулярних резервних копій усіх критично важливих даних, включно з індексами, конфігураціями та метаданими. Ми тестуємо процес відновлення для впевненості у його надійності й швидкості у разі виникнення надзвичайної ситуації.

Щоб забезпечити високу доступність системи, ми реалізовуємо механізми відмовостійкості, зокрема конфігуруємо Elasticsearch як кластер з кількома вузлами та резервними копіями індексів. Це дозволяє системі продовжувати роботу навіть у разі виходу з ладу одного або кількох елементів інфраструктури.

Також застосовуємо принципи горизонтального масштабування, щоб система могла адаптуватися до зростання навантаження без втрати продуктивності. Усе це дозволяє нам підтримувати безперервну роботу сервісів.

ВИСНОВКИ

У даній роботі було розглянуто процес автоматизації моніторингу та логування серверів за допомогою комплексу технологій ELK Stack, що включає в себе Elasticsearch, Logstash та Kibana. Метою цього проєкту було створення ефективної, надійної та масштабованої системи, здатної забезпечити централізоване зберігання логів, їх обробку та візуалізацію для подальшого аналізу та прийняття рішень щодо оптимізації серверної інфраструктури.

Під час виконання роботи було проведено ретельний аналіз вимог до системи моніторингу та логування, що дозволило вибрати найбільш підходящу технологію для досягнення високої продуктивності та стабільності. Завдяки використанню компонентів ELK Stack, система змогла ефективно обробляти та зберігати великі обсяги даних, забезпечити швидкий доступ до них і надати інтерфейс для зручної візуалізації та аналізу.

Одним з основних досягнень роботи є розробка та налаштування логування для різноманітних джерел даних. Використовуючи Logstash, вдалося реалізувати збирання даних з різних серверів, мережевих пристроїв та додатків, що забезпечило централізовану обробку та зберігання всіх логів в одній системі. Це не лише полегшує процес виявлення помилок і несправностей, а й дозволяє оперативно реагувати на інциденти, знижуючи час на їх усунення.

Завдяки потужним можливостям Kibana, вдалося розробити зручні та інформативні дашборди для моніторингу стану серверів та аналізу роботи інфраструктури в реальному часі. Це дозволяє адміністраторам систем швидко знаходити аномалії, відстежувати критичні події та виявляти потенційні проблеми до того, як вони призведуть до серйозних збоїв у роботі.

Впровадження цієї системи значно полегшує завдання адміністрування серверів і допомагає знизити ризики, пов’язані з людським фактором, завдяки автоматизації збору, обробки та зберігання логів. Зокрема, система забезпечує швидку реакцію на інциденти, покращує безпеку та продуктивність

інфраструктури, а також дозволяє заощадити час і ресурси при виконанні рутинних завдань.

Враховуючи зростаючі вимоги до масштабованості та надійності сучасних інформаційних систем, можна стверджувати, що використання ELK Stack є ефективним і перспективним рішенням для автоматизації моніторингу та логування. В майбутньому система може бути доопрацьована для підтримки ще більш складних сценаріїв, інтеграції з іншими інструментами та адаптації до нових вимог бізнесу та технологічних змін.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is Elasticsearch? - Elasticsearch Explained - AWS. *Amazon Web Services, Inc.* URL: https://aws.amazon.com/what-is/elasticsearch/ (date of access: 22.03.2025).

2. What is ELK stack? - Elasticsearch, Logstash, Kibana Stack Explained - AWS. *Amazon Web Services, Inc.* URL: https://aws.amazon.com/what-is/elk-stack/?nc1=h_ls (date of access: 22.03.2025).

3. Noble D. Monitoring ElasticSearch. Packt Publishing - ebooks Account, 2016. 180 p.

4. Sami W. ElasticSearch Architecture: A Comprehensive Guide. *DEV Community*. URL: https://dev.to/wadee_sami_4562c11ecf8066/elasticsearch-architecture-a-comprehensive-guide-12me (date of access: 20.04.2025).

5. Підручник із стеку ELK: що таке Kibana, Logstash & Elasticsearch?. *Guru99*. URL: https://www.guru99.com/uk/elk-stack-tutorial.html (дата звернення: 12.02.2025).

6. Elasticsearch Architecture: 7 Key Components | NetApp. *Access Denied*. URL: https://www.netapp.com/learn/cvo-blg-elasticsearch-architecture-7-key-components/ (date of access: 12.04.2025).

7. Elastic Stack: (ELK) Elasticsearch, Kibana & Logstash. *Elastic*. URL: https://www.elastic.co/elastic-stack/ (date of access: 13.05.2025).

8. GeeksforGeeks. Introduction to Monitoring using the ELK Stack - GeeksforGeeks. *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/introduction-to-monitoring-using-the-elk-stack/ (date of access: 11.04.2025).

9. An Introduction to ELK Stack | Baeldung on Ops. *Baeldung on Ops*. URL: https://www.baeldung.com/ops/elk (date of access: 08.05.2025).

10. What Is ELK Stack: Tutorial on How to Use It for Log Management. *Sematext*. URL: https://sematext.com/guides/elk-stack/#elk-logging-how-to-use-the-elastic-stack-for-log-management-analysis--analytics (date of access: 08.05.2025).

11. Abrahamsson J. ElasticSearch Quick Start. Ross Publishing, Incorporated, J., 2016.

12. Understanding Elasticsearch Index Mapping & Schema - DevOpsSchool.com. *DevOpsSchool.com*. URL: https://www.devopsschool.com/blog/understanding-elasticsearch-index-mapping-schema/ (date of access: 04.05.2025).

13. Ku? R. Mastering Elasticsearch. Packt Publishing, Limited, 2013.

14. Bennett E. What is Kibana?. *Logit.io*. URL: https://logit.io/blog/post/what-is-kibana/ (date of access: 02.05.2025).

15. Kibana Tutorial - Data Visualization Made Easy. *Tutorials on Technical and Non Technical Subjects*. URL: https://www.tutorialspoint.com/kibana/index.htm (date of access: 02.05.2025).

16. GeeksforGeeks. Introduction to Logstash for Data Ingestion - GeeksforGeeks. *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/introduction-to-logstash-for-data-ingestion/ (date of access: 29.04.2025).

17. Blokdyk G. ELK Stack a Complete Guide - 2020 Edition. Emereo Pty Limited, 2020. 302 p.

18. How to Collect, Process, and Ship Log Data with Logstash | Better Stack Community. *Better Stack - Radically better observability stack*. URL: https://betterstack.com/community/guides/logging/logstash-explained/ (date of access: 21.05.2025).

19. Configure Logstash Elasticsearch Basic Authentication - kifarunix.com. *kifarunix.com*. URL: https://kifarunix.com/configure-logstash-elasticsearch-basic-authentication/ (date of access: 08.05.2025).

20. The Complete Guide to the ELK Stack | Logz.io. *Logz.io*. URL: https://logz.io/learn/complete-guide-elk-stack/#what-elk-stack (date of access: 11.05.2025).

21. Sachdeva G. S. Practical ELK Stack. Berkeley, CA : Apress, 2017. URL: https://doi.org/10.1007/978-1-4842-2626-1 (date of access: 06.04.2025).

22. Spring Boot Microservices + ELK(Elasticsearch, Logstash, and Kibana) Stack Hello World Example| JavaInUse. *Home | JavaInUse*. URL: https://www.javainuse.com/spring/springboot-microservice-elk (date of access: 13.04.2025).

23. Особливості логування даних та вплив типу додатка на вибір методології логування. *ELARKHNU Інституційний репозитарій Хмельницького національного університету :: Головна*. URL: https://elar.khmnu.edu.ua/items/96d3a3f4-bb0f-4fd9-aa68-bcd16624b556 (дата звернення: 22.04.2025).

24. Chhajed S. Learning ELK Stack. Packt Publishing, Limited, 2015.

25. Що таке логи сервера і навіщо потрібні - Блог - HostZealot. *HostZealot*. URL: https://www.hostzealot.com.ua/blog/about-servers/scho-take-logi-servera-i-navischo-potribni (дата звернення: 18.05.2025).

26. Akdogan H. Elasticsearch Indexing. Packt Publishing, Limited, 2015.

27. Що таке логи та як з ними працювати. *Онлайн-курси від компанії QATestLab | Головна сторінка*. URL: https://training.qatestlab.com/blog/technical-articles/logging-in-concepts-requirements-levels/ (дата звернення: 22.05.2025).

28. 8 НАЙКРАЩИХ інструментів і програмного забезпечення для моніторингу серверів (2025). *Guru99*. URL: https://www.guru99.com/uk/best-server-monitoring-software.html (дата звернення: 22.05.2025).

29. Network Olympus. *Програмне забезпечення для керування мережею: програмне забезпечення для інвентаризації, моніторинг серверів, розгортання програмного забезпечення | Softinventive*. URL: https://www.softinventive.com.ua/server-monitoring (date of access: 28.02.2025).

30. Miller D., Srivastava A. Elasticsearch 7 Quick Start Guide: Get up and Running with the Distributed Search and Analytics Capabilities of Elasticsearch. Packt Publishing, Limited, 2019.

Додаток А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ



Навчально-науковий інститут Інформаційних технологій
Кафедра Комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

на тему: «Автоматизація моніторингу та логування серверів з використанням ELK Stack»

Виконав студент групи КІД-41
Бондарчук Максим Олександрович

Керівник кваліфікаційної роботи:
Жужков Денис Ігорович, викладач кафедри КІ

Київ – 2025

Мета роботи – Розробка системи моніторингу та логування сервера

Об`єкт дослідження – Системи моніторингу та логування сервера

Предмет дослідження – Система моніторингу та логування сервера побудована на ELK Stack

Актуальність обраної теми

У сучасному світі інформаційних технологій стабільна робота серверної інфраструктури є критично важливою для забезпечення безперервного функціонування бізнес-процесів, онлайн-сервісів та інформаційних систем. Зі зростанням масштабів IT-інфраструктури зростає і кількість подій, які необхідно контролювати: системні помилки, збої в роботі додатків, підозріла активність, проблеми з мережею тощо. Ручний моніторинг та аналіз логів у таких умовах стає малоефективним, затратним за часом та схильним до людських помилок. Саме тому виникає потреба в автоматизації процесів моніторингу та логування. Вона дозволяє оперативно виявляти й реагувати на критичні події, знижує ризики простоїв, підвищує безпеку, а також оптимізує ресурси обслуговування IT-систем

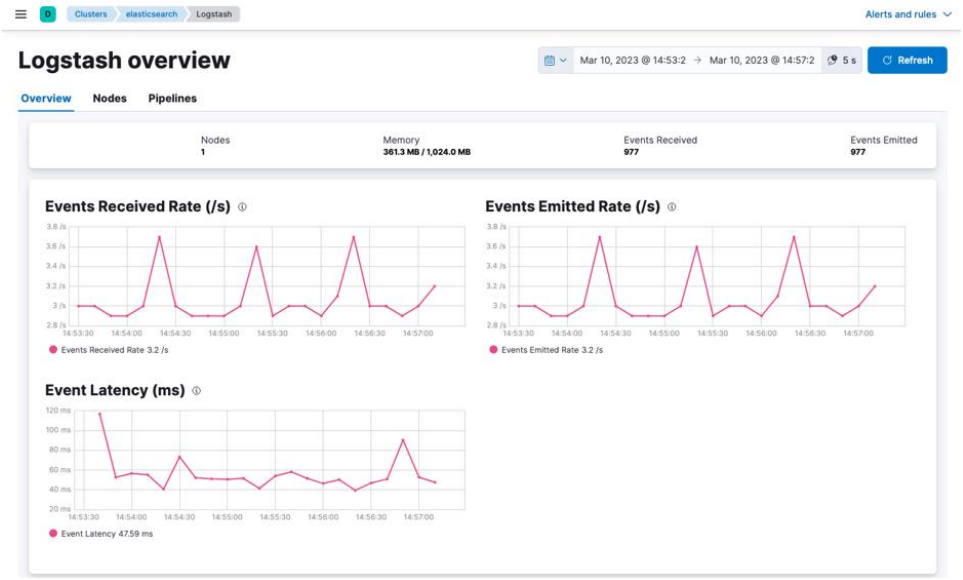
3

Проблеми які вирішує автоматизація:

1. Ручне збирання логів займає багато часу;
2. Складно вчасно виявити проблеми;
3. Ризик втрати важливих подій;
4. Неможливо швидко знайти інформацію;
5. Відсутність масштабування;
6. Навантаження на персонал.

4

Приклад інтерфейсу моніторингу сервера



5

ELK як комплексне рішення

ELK Stack — це абревіатура, що об'єднує три ключові компоненти: Elasticsearch, Logstash та Kibana. Разом ці інструменти утворюють комплексну платформу для централізованого логування, що забезпечує зручне управління логами, швидкий пошук інформації та гнучкі можливості візуалізації. Кожен із компонентів виконує свою специфічну функцію, але найкраще вони працюють саме у зв’язці, утворюючи єдине інтегроване середовище.



Рисунок 1 – Компоненти ELK Stack

6

Elasticsearch

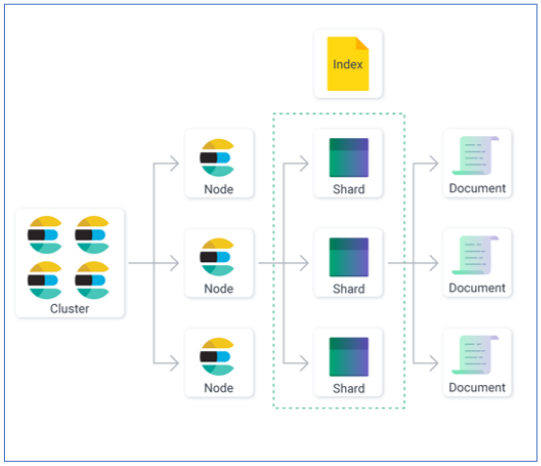


Рисунок 2 – Структура Elasticsearch

Elasticsearch — це розподілена пошукова та аналітична система з відкритим кодом, створена на базі бібліотеки Apache Lucene. Основним завданням Elasticsearch є зберігання, індексація та забезпечення швидкого пошуку великих обсягів даних. На відміну від традиційних реляційних баз даних, Elasticsearch оперує документами у форматі JSON, що робить його зручним для роботи з напівструктурованими та неструктурованими даними.

7

Logstash

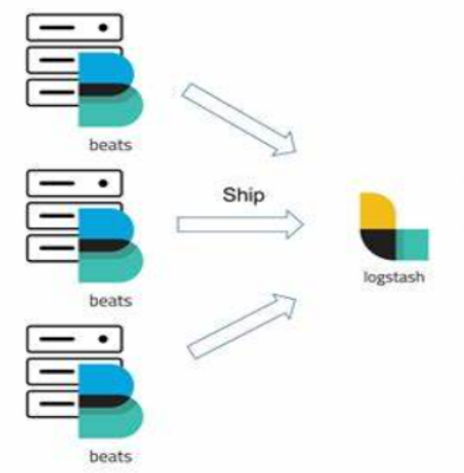


Рисунок 3 – Схема збору логів Logstash

Logstash — це гнучкий конвеєр для обробки логів і подій, який відповідає за збір даних із різних джерел, їх попередню обробку та передачу у Elasticsearch або інші системи зберігання. Logstash дозволяє працювати з даними у потоковому режимі та має модульну архітектуру, яка базується на трьох основних етапах: input, filter та output.

8

Kibana

Kibana — це веб-інтерфейс для візуалізації даних, які зберігаються в Elasticsearch. Його основне призначення — надання зручного способу для аналізу, перегляду та презентації інформації у вигляді графіків, таблиць, дашбордів та інших візуальних компонентів.

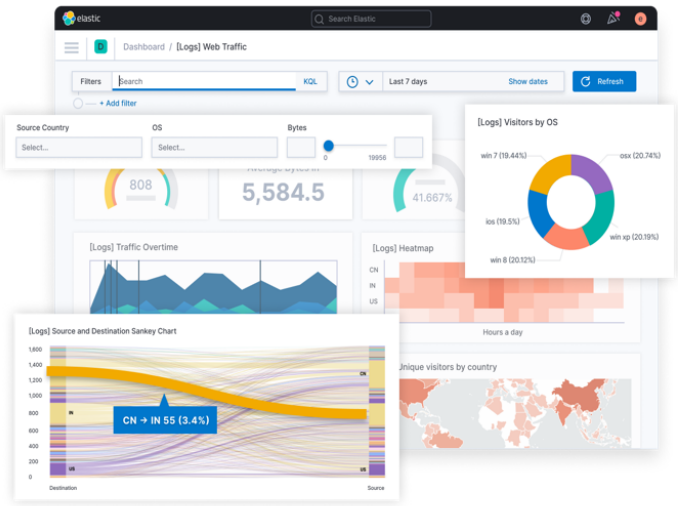


Рисунок 4 – Інтерфейс Kibana

9

Сценарій роботи системи моніторингу ELK Stack

1. Сервер генерує логи в реальному часі;
2. Logstash забирає логи, обробляє й передає їх у Elasticsearch;
3. Elasticsearch індексує та зберігає ці дані;
4. Kibana дозволяє переглядати, фільтрувати та аналізувати інформацію;
5. Система працює постійно й оновлюється в реальному часі.



10

Етапи впровадження системи:

1. **Аналіз вимог до системи моніторингу та логування.** Визначаються основні потреби організації щодо збору, обробки та зберігання логів. Оцінюється тип обладнання і додатків, які потребують моніторингу.
2. **Вибір та проєктування архітектури рішення.** Розробляється структура системи з урахуванням масштабованості та надійності. Визначаються компоненти ELK Stack і способи їх взаємодії.
3. **Налаштування та конфігурація компонентів ELK Stack.** Встановлюються та налаштовуються Elasticsearch, Logstash і Kibana відповідно до потреб проєкту. Виконується налаштування збору, індексації та візуалізації даних.
4. **Розгортання системи на серверному обладнанні.** Встановлення та запуск компонентів системи на фізичних або віртуальних серверах. Забезпечується інтеграція з існуючою інфраструктурою.
5. **Налагодження збору та обробки логів із джерел.** Налаштовується збір логів з різних серверів, мережевих пристроїв і додатків. Використовуються фільтри та трансформації для коректного форматування даних.
6. **Розробка дашбордів для візуалізації даних.** Створюються інформативні панелі моніторингу, що відображають ключові метрики та події. Забезпечується зручний доступ до аналітики для користувачів.
7. **Тестування працездатності та продуктивності системи.** Перевіряється коректність збору, обробки та відображення логів. Оцінюється швидкодія системи під навантаженням.
8. **Впровадження системи в експлуатацію.** Після успішного тестування система запускається в робоче середовище. Забезпечується навчання користувачів і підтримка системи на початковому етапі.

11

ВИСНОВКИ

Розглянуто процес автоматизації моніторингу та логування серверів за допомогою комплексу технологій ELK Stack, що включає в себе Elasticsearch, Logstash та Kibana. Метою цього проєкту було створення ефективної, надійної та масштабованої системи, здатної забезпечити централізоване зберігання логів, їх обробку та візуалізацію для подальшого аналізу та прийняття рішень щодо оптимізації серверної інфраструктури.

Під час виконання роботи було проведено ретельний аналіз вимог до системи моніторингу та логування, що дозволило вибрати найбільш підходящу технологію для досягнення високої продуктивності та стабільності. Завдяки використанню компонентів ELK Stack, система змогла ефективно обробляти та зберігати великі обсяги даних, забезпечити швидкий доступ до них і надати інтерфейс для зручної візуалізації та аналізу.

Одним з основних досягнень роботи є розробка та налаштування логування для різноманітних джерел даних. Використовуючи Logstash, вдалося реалізувати збирання даних з різних серверів, мережевих пристроїв та додатків, що забезпечило централізовану обробку та зберігання всіх логів в одній системі. Це не лише полегшує процес виявлення помилок і несправностей, а й дозволяє оперативно реагувати на інциденти, знижуючи час на їх усунення.

12

ПРОДОВЖЕННЯ ВИСНОВКІВ

Завдяки потужним можливостям Kibana, вдалося розробити зручні та інформативні дашборди для моніторингу стану серверів та аналізу роботи інфраструктури в реальному часі. Це дозволяє адміністраторам систем швидко знаходити аномалії, відстежувати критичні події та виявляти потенційні проблеми до того, як вони призведуть до серйозних збоїв у роботі.

Впровадження цієї системи значно полегшує завдання адміністрування серверів і допомагає знизити ризики, пов’язані з людським фактором, завдяки автоматизації збору, обробки та зберігання логів. Зокрема, система забезпечує швидку реакцію на інциденти, покращує безпеку та продуктивність інфраструктури, а також дозволяє заощадити час і ресурси при виконанні рутинних завдань.

Враховуючи зростаючі вимоги до масштабованості та надійності сучасних інформаційних систем, можна стверджувати, що використання ELK Stack є ефективним і перспективним рішенням для автоматизації моніторингу та логування. В майбутньому система може бути доопрацьована для підтримки ще більш складних сценаріїв, інтеграції з іншими інструментами та адаптації до нових вимог бізнесу та технологічних змін.

13