МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

ЛЕКЦИОННЫЕ МАТЕРИАЛЫ

Технологии хранения в системах кибербезопасности

(наименование дисциплины (модуля) в соответствии с учебным планом)

Уровень	бакалавриат
	(бакалавриат, магистратура, специалитет)
Форма обучения	очная
	(очная, очно-заочная, заочная)
Направление(-я) подготовки	10.05.04 Информационно-аналитические системы безопасности
	(код(-ы) и наименование(-я))
Институт	Кибербезопасности и цифровых технологий (ИКБ)
	(полное и краткое наименование)
Кафедра	КБ-2 «Прикладные информационные технологии»
	(полное и краткое наименование кафедры, реализующей дисциплину (модуль))
Лектор	к.т.н., Селин Андрей Александрович
	(сокращенно – ученая степень, ученое звание; полностью – ФИО)

Используются в данной редакции с учебного года 2024/2025

(учебный год цифрами)

Проверено и согласовано «____» _____2024 г. _____ А.А. Бакаев

(подпись директора Института/Филиала с расшифровкой)

Москва 2024 г.

Технологии хранения в системах кибербезопасности

2024 год

Лекция 9.
ELK

Учебные вопросы лекции:

1. **ELK**
2. **Архитектура ELK**

ElasticSearch,

Logstash,

Kibana,

Beats.

Сайзинг кластера Elasticsearch по объёму хранения (долг)

Хранение логов и метрик обычно требует значительного дискового пространства, поэтому стоит использовать количество этих данных для первоначального определения размера нашего кластера Elasticsearch.

Общий объем данных (ГБ) = Количество сырых данных в день (Гб) * Количество дней хранения * (Количество реплик + 1).

Общий объем хранилища (ГБ) = Общий объем данных (ГБ) * (1 + 0.15 запаса дискового пространства + 0.1 дополнительного резерва).

Общее количество нод данных = ОКРВВЕРХ (Общий объем данных (ГБ) / Объём памяти на ноду данных / Соотношение память: данные). В случае крупной инсталляции лучше держать в запасе ещё одну дополнительную ноду.

Elastic рекомендует следующие соотношения память: данные для различных типов нод: «горячие» → 1:30 (30 Гб дискового пространства на каждый гигабайт памяти), «тёплые» → 1:160, «холодные» → 1:500). ОКРВВЕРХ — окружение до ближайшего большего целого числа.

Пример расчёта малого кластера

Давайте предположим, что каждый день прилетает ~1 ГБ данных, которые нужно хранить 9 месяцев.

Общий объем данных (ГБ) = 1 ГБ x (9 месяцев x 30 дней) x 2 = 540 ГБ.

Общий объем хранилища (ГБ) = 540 ГБ x (1+0.15+0.1) = 675 ГБ.

Общее количество нод данных = 675 ГБ / 8 ГБ ОЗУ / 30 = 3 ноды.

Сайзинг кластера Elasticsearch по объёму хранения (долг)

Пример расчета крупного кластера

Вы получаете 100 ГБ в день, будете эти данные 30 дней в горячей зоне и 12 месяцев в теплой зоне. У вас есть 64 ГБ памяти на каждый узел, из которых 30 ГБ выделено для JVM Heap, а оставшаяся часть — для кэш-памяти ОС. Рекомендуемое соотношение память: данные для горячей зоны 1:30, для теплой — 1: 160.

Итого, если вы получаете 100 ГБ в день и должны хранить эти данные в течение 30 дней, получим:

Общий объем данных (ГБ) в горячей зоне = (100 ГБ x 30 дней * 2) = 6000 ГБ

Общий объем хранилища (ГБ) в горячей зоне = 6000 ГБ x (1 + 0,15 + 0,1) = 7500 ГБ

Общее количество нод данных в горячей зоне = ОКРВВЕРХ (7500/64/30) + 1 = 5 узлов

Общий объем данных (ГБ) в теплой зоне = (100 ГБ x 365 дней * 2) = 73 000 ГБ

Общий объем хранилища (ГБ) в теплой зоне = 73 000 ГБ x (1 + 0,15 + 0,1) = 91 250 ГБ

Общее количество узлов данных в теплой зоне = ОКРВВЕРХ (91 250/64/160) + 1 = 10 узлов

Таким образом, получили 5 узлов под горячую зону и 10 узлов плод теплую. Для холодной зоны аналогичные расчеты, но коэффициент память: данные уже будет 1:500.

Введение

Elastic Search – NoSQL БД/поисковая система с открытым исходным кодом, предназначенная для полнотекстового поиска. Она позволяет хранить, анализировать и получать большие объемы данных в режиме реального времени.

Для анализа и поиска Elastic Search использует библиотеку Apache Lucene. Написана она на языке Java и доступна для многих платформ. Все неструктурированные данные хранятся в формате JSON. Для работы с данными у Elastic Search есть специальное REST API.

Основные задачи, решаемые Elastic Search:

Полнотекстовый поиск.

Поиск по параметрам.

Агрегация данных для статистики и их последующая визуализация.

Генерация вариантов для автозаполнения.

ELK

ElasticSearch – центральный элемент экосистемы Elastic: ELK Stack.

ELK – это акроним трех продуктов компании Elastic:

– ElasticSearch (поисковый и аналитический движок);

– Logstash (конвейер обработки данных);

– Kibana (интерфейс для визуализации данных).

На сегодняшний день можно выделить два самых популярных сценария использования ElasticSearch:

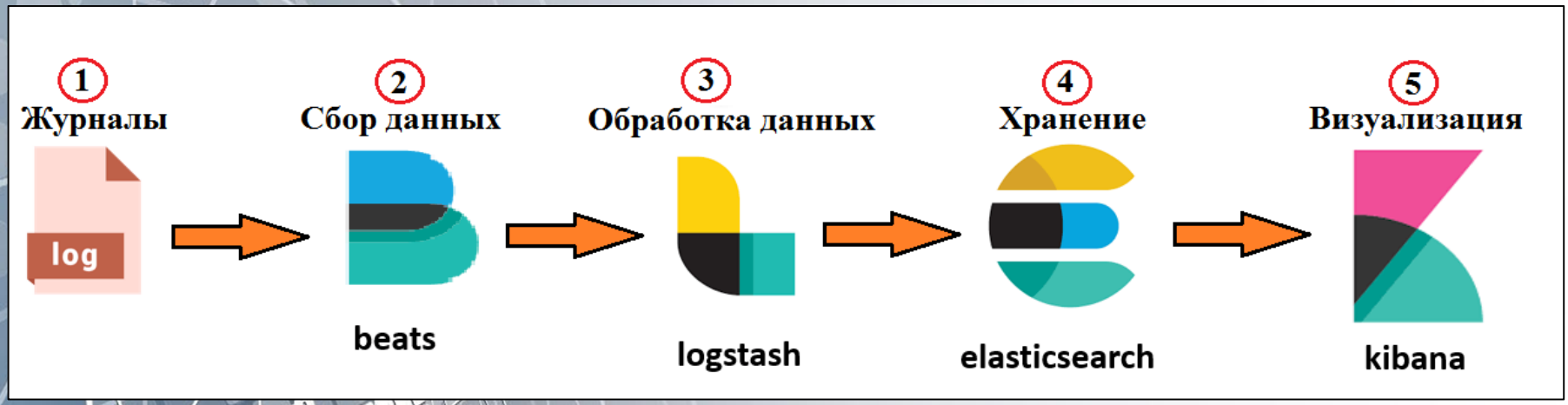
– движок для полнотекстового поиска;

– хранилище логов и метрик в ELK Stack.

Идея создания ElasticSearch состоит в предоставлении возможности библиотеки полнотекстового поиска Apache Lucene для Java пользователям других языков через простой и понятный всем интерфейс: JSON поверх HTTP. Поэтому все запросы представляют собой JSON, а передаются через HTTP.

Для исполнения запросов из примеров можно взять любой HTTP-клиент, но рекомендуется использование Dev Tools в Kibana из-за наличия автозаполнения запросов и подсветки синтаксиса.

Архитектура стека ELK



Журналы(log): идентифицируются журналы сервера, которые необходимо проанализировать.

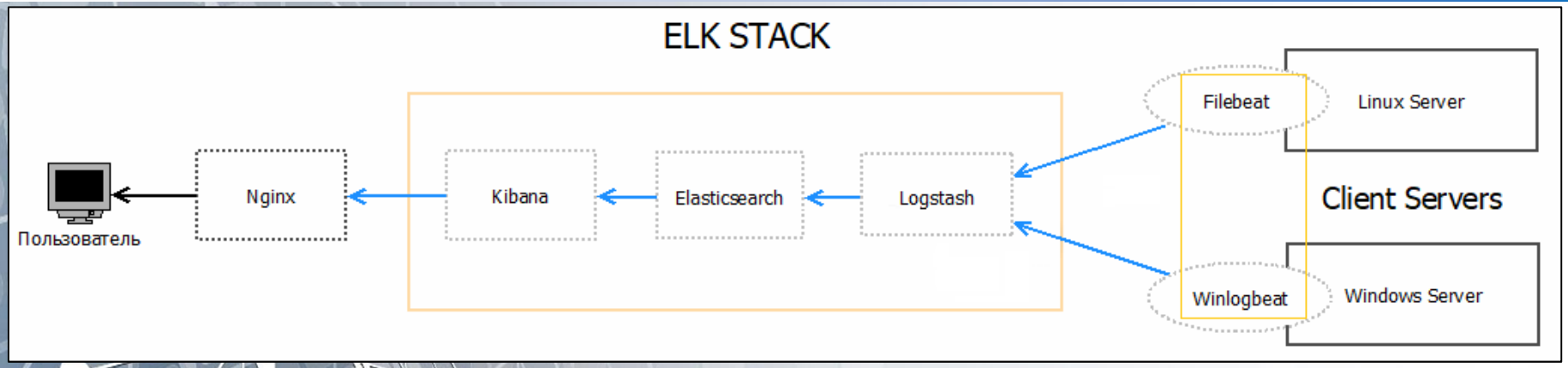
Beats: осуществляется сбор данных.

Logstash: сбор журналов и данных о событиях. Обработка данных.

ElasticSearch: преобразованные данные из Logstash помещаются в хранилище. Элемент хранит данные, осуществляет индексацию данных и поиск .

Kibana использует базу данных Elasticsearch для изучения, визуализации и обмена.

ELK Stack



Стек технологий ELK:

Elasticsearch (масштабируемое хранилище данных с широкими возможностями поиска),

Logstash (инструмент для сбора, обогащения, фильтрации и маршрутизации данных, например журналов приложений),

Kibana (инструмент для исследования и визуализации данных),

Beats (агенты для отправки логов (Filebeat и Winlogbeat).

ELK Stack позволяет пользователям получать данные из любого источника в любом формате, а также выполнять поиск, анализ и визуализацию этих данных в режиме реального времени.

Журналы

Изоляция производительности труднодостижима, особенно когда системы сильно загружены. Каждое соответствующее событие в системе должно регистрироваться.

Целью ведения журнала на уровне приложения является все, что связано с любым взаимодействием пользователя с системой – будь то авторизация пользователя в системе или запрос пользователя на какой-либо URL-адрес, электронное письмо, отправленное пользователю, и т. д. регистрировать разнообразную информацию, этот журнал не структурирован, но он должен содержать некоторую базовую информацию, чтобы было легче получить к нему доступ.

Лучший совет относительно ведения журналов в распределенных системах – «штамповать» любое соответствующее событие в источнике, которое каким-либо образом распространяется через распределенную систему, независимо от того, затрагивает ли оно большее количество частей системы или нет. В случае запроса веб-страницы, например, балансировщик нагрузки или веб-сервер должен быть поставлен на такой «штамп». Это запечатанное событие передается дальше до конца своего срока службы. Эти «штампы» часто реализуются как UUID.

Beats

Filebeat (следит за файловой системой);

Winlogbeat (следит за событиями журнала Windows);

Metricbeat (загрузка процессора, памяти, дисковые IO);

Heartbeat (время работы и доступность системы);

Packetbeat (сетевые пакеты);

Community Beats.

Написаны на языке Go;

Имеется библиотека ElasticSearch;

Документация от сообщества;

Могут передавать данные в:

- Logstash;
- ElasticSearch;
- Kafka;
- …

Beats

1. **FileBeat** - это легкий инструмент для сбора логов. (Интеллектуальная скорость передачи регулировки, предотвращающая перегрузки Logstash, Elasticsearch.
2. **MetricBeat** - это легкий инструмент мониторинга индекса производительности системы. Объединяет индикаторы системы, такие как процессор, память, диск и другие индикаторы различных услуг, таких как Redis, Nginx.
3. **PacketBeat** - это легкий инструмент анализа пакетов сетевых данных. Если вы использовали Wireshark, Fiddler то понимаете концепцию пакетного анализа. Если вы его не использовали, вы можете обратиться к функции инструментов Chrome Dev. PacketBeat может проанализировать сетевое взаимодействие приложения через анализ сетевого трафика и отправить «пойманные данные» в Logstash или Elasticsearch.
4. **Heartbeat** - это инструмент обнаружения «сердцебиения», доступный для мониторинга (проверка правильной работоспособности служб).

Logstash

Файл конфигурации logstash.conf :

```
input {
file {
path => ["/var/log/myapp/*.log"]
}
}
 filter {
// собрать или обработать данные, например 'time‘, в
//качестве временной метки и хранить их в @timestamp.
// эти данные позже будут использоваться Kibana.
 date {
match => [ "time" ]
}
 // добавляет данные геолокации на основе IP-адреса.
geoip {
source => "ip"
}
}
output {
elasticsearch {
hosts => ["localhost:9200"]
}
stdout { codec => rubydebug }
}
```

рнала, в игру вступает Logstash.
дут преобразованы в один из
озже будут отправлены на сервер

атем передает их в виде конвейера

h как базу данных, а LogStash как
яет на него журналы или файлы.

быть системным журналом, Redis
edis часто используется в качестве
инфраструктуре Logstash, где
общения, а один из экземпляров

Журнал приложений :

```
{
"action": "action log",
"user": "Иван",
"time": 11:01:2021,
"ip": "192.168....",
"transaction-id": "r5e1244-32432-1465-q346-6ahsms57081x4"
}
```

Logstash

Функции **LogStash:**

- События проходят через каждую фазу с использованием внутренних очередей.
- Позволяет использовать различные входные данные для журналов.
- Фильтрация/анализ журналов.

Преимущества **LogStash**:

- Централизованная обработка данных.
- Анализирует большое количество структурированных/неструктурированных данных и событий.
- ELK LogStash предлагает плагины для подключения к различным типам источников ввода и платформ.

ElasticSearch

Elasticsearch также позволяет хранить, искать и анализировать большие объемы данных. В основном используется в качестве базового механизма для приложений, отвечающих требованиям поиска. Был принят на платформах поисковых систем для современных веб-приложений и мобильных приложений. Помимо быстрого поиска, инструмент также предлагает комплексную аналитику и множество дополнительных функций.

В тот момент, когда Logstash завершает свою работу и пересылает журналы, ElasticSearch уже может обрабатывать данные.

В результате выполнения команды curl:

curl -XGET 'http://192.168.(host ip):9200/_search'

Можно увидеть «документы» в БД.

192.168.(host ip) в данном случае является адресом виртуальной машины boot2docker , а 9200 - это открытый порт по умолчанию.

Поскольку результат находится в формате JSON, то результаты готовы для дальнейшей обработки.

curl -XGET 'http://192.168.(host ip):9200/_search?hello‘

возвращает более читаемую версию документа для быстрой проверки. В зависимости от разработки, обычная практика заключается в регистрации запроса/ ответа полезной нагрузки, идентификатора корреляции и т. д., чтобы отслеживать ошибку через панель управления пользовательского интерфейса Kibana позже.

ElasticSearch Индексация и поиск документа

Перед индексацией документа происходит его предобработка:

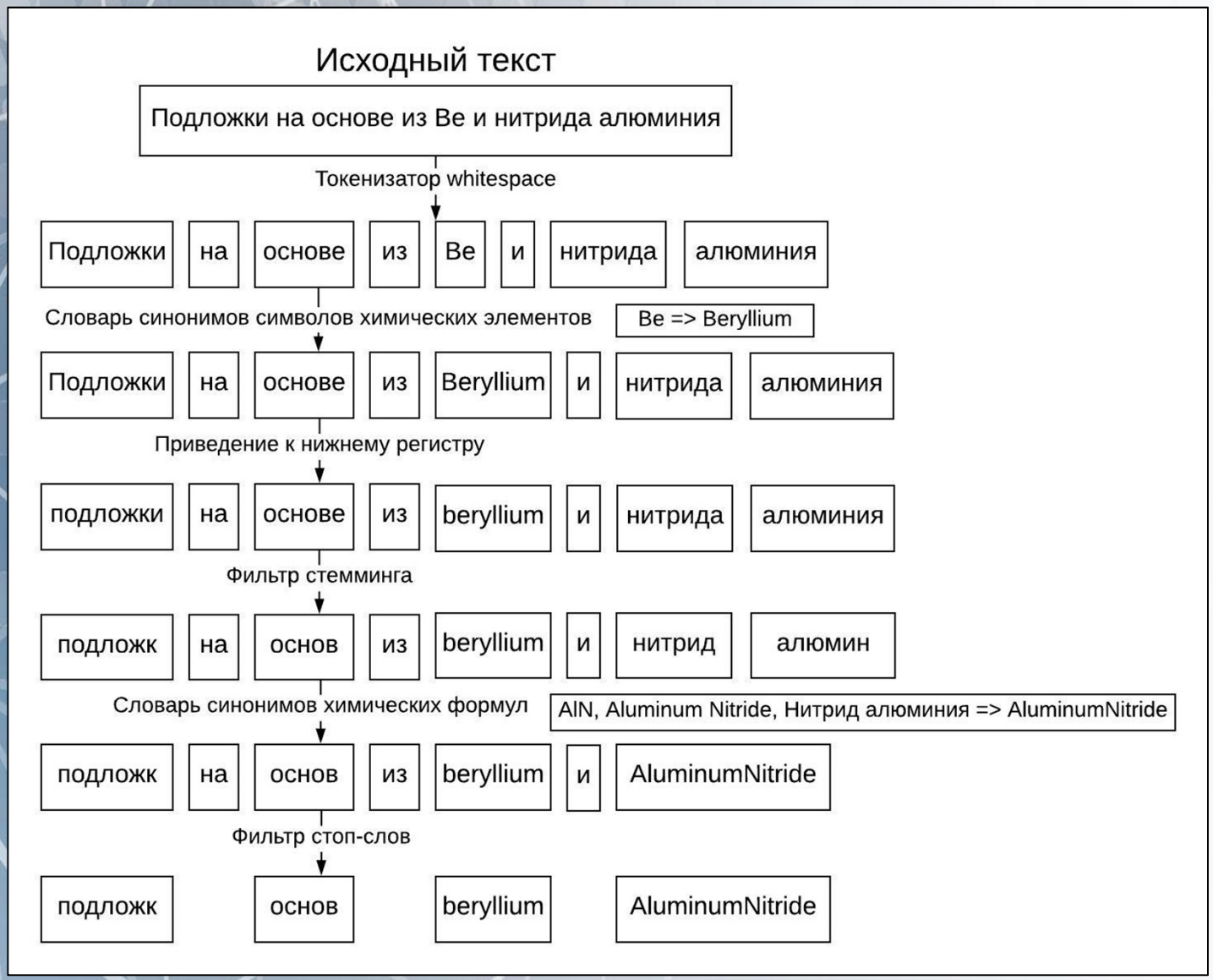
- Разбиение на термы (токены)
- Применение набора фильтров термов
- Сохранение термов в инвертированный индекс вместе с позициями в исходных документах

Аналогичная трансформация происходит с поисковыми запросами:

- Разбиение на термы (токены)
- Применение набора фильтров термов
- Поиск документов в инвертированном индексе по термам
- Сортировка документов по релевантности

ElasticSearch Индексация и поиск документа

Индексация документа, пример:



ElasticSearch Индексация и поиск документа

Инвертированный индекс

Проиндексируем три простых документа:

«Подложки на основе из Be и нитрида алюминия»

«Кристалл с примесью бериллия»

«Beryllium oxide»

Термы	Номера документов
подложк	1
основ	1
beryllium	1, 2, 3
AluminumNitride	1
кристалл	2
примес	2
oxid	3

ElasticSearch

Особенности Elasticsearch:

- Поисковый сервер с открытым исходным кодом написан с использованием Java.
- Используется для индексации любых разнородных данных.
- Имеет веб-интерфейс REST API с выводом JSON.
- Полнотекстовый поиск.
- Поиск почти в реальном времени (NRT).
- Разделенное, реплицированное хранилище документов JSON с возможностью поиска.
- Распределенное хранилище документов без схем, основанное на REST и JSON.
- Поддержка нескольких языков и геолокации.

Преимущества Elasticsearch:

- Хранение данные без схемы, с возможностью создания схемы для данных.
- Управление записью данных с помощью API для работы с несколькими документами.
- Удобная фильтрация и наглядное представление данных
- Основан на Apache Lucene и предоставляет RESTful API.
- Обеспечивает горизонтальную масштабируемость, надежность и возможность работы с несколькими арендаторами для использования индексации в режиме реального времени, чтобы ускорить поиск.
- Легкот масштабировать по вертикали и горизонтали

Kibana

Kibana – это, по сути своей, клиент статического пользовательского интерфейса (HTML + CSS + JS), который отображает данные так, как нужно пользователю, в экземпляре ElasticSearch, где отображаются различные отчеты и аналитика. За исключением того, что с помощью Kibana можно легко выполнять запросы по индексам ElasticSearch, главное преимущество заключается в том, что для этих запросов, какими бы сложными они ни были, возможно «упорядочить».

Kibana поставляется с очень мощным набором инструментов визуализации, поэтому, например, можно видеть, как часто подобное событие повторяется с течением времени, можно агрегировать события по различным критериям (например, сколько запросов пришло с определенного IP-адреса. за последний час, увеличилось ли количество ошибок на конкретном сервере, или даже заметили ли увеличение количества запросов для конкретной страницы). Кроме того, это отличный инструмент для обнаружения аномалий, вызванных изменениями в вашей системе.

В Kibana есть разные методы поиска данных.

Типы поиска данных в Kibana

Тип поиска	Применение
Свободный текстовый поиск	Используется для поиска определенной строки
Поиск на уровне поля	Используется для поиска строки в определенном поле.
Логические утверждения	Он используется для объединения поисков в логический оператор.
Поиск близости	Он используется для поиска терминов в пределах определенной близости символов.

Особенности Kibana:

- Мощная интерфейсная панель мониторинга, визуализирующая проиндексированную информацию из эластичного кластера.
- Обеспечивает поиск индексированной информации в режиме реального времени.
- Поиск, просмотр и взаимодействие с данными, хранящимися в Elasticsearch.
- Выполнение запросов к данным и визуализация результатов в виде диаграмм, таблиц и карт.
- Настраиваемая панель мониторинга журналов logstash в elasticsearch.
- Предоставление исторических данных в виде графиков, диаграмм и т. д.
- Легконастраиваемые панели мониторинга в реальном времени.
- Kibana ElasticSearch обеспечивает поиск индексированной информации в режиме реального времени.

Возможности визуализации Kibana

Преимущества и недостатки стека ELK

Преимущества:

- ELK работает лучше всего, когда журналы из различных приложений предприятия сходятся в одном экземпляре ELK.
- Он обеспечивает потрясающую информацию для этого единственного экземпляра, а также устраняет необходимость входа в сотню различных источников данных журнала.
- Быстрая установка на месте.
- Простота развертывания. Вертикальное и горизонтальное масштабирование.
- Elastic предлагает множество языковых клиентов, включая Ruby. Питон. PHP, Perl, .NET, Java, JavaScript и многое другое.
- Наличие библиотек для разных языков программирования и скриптов.

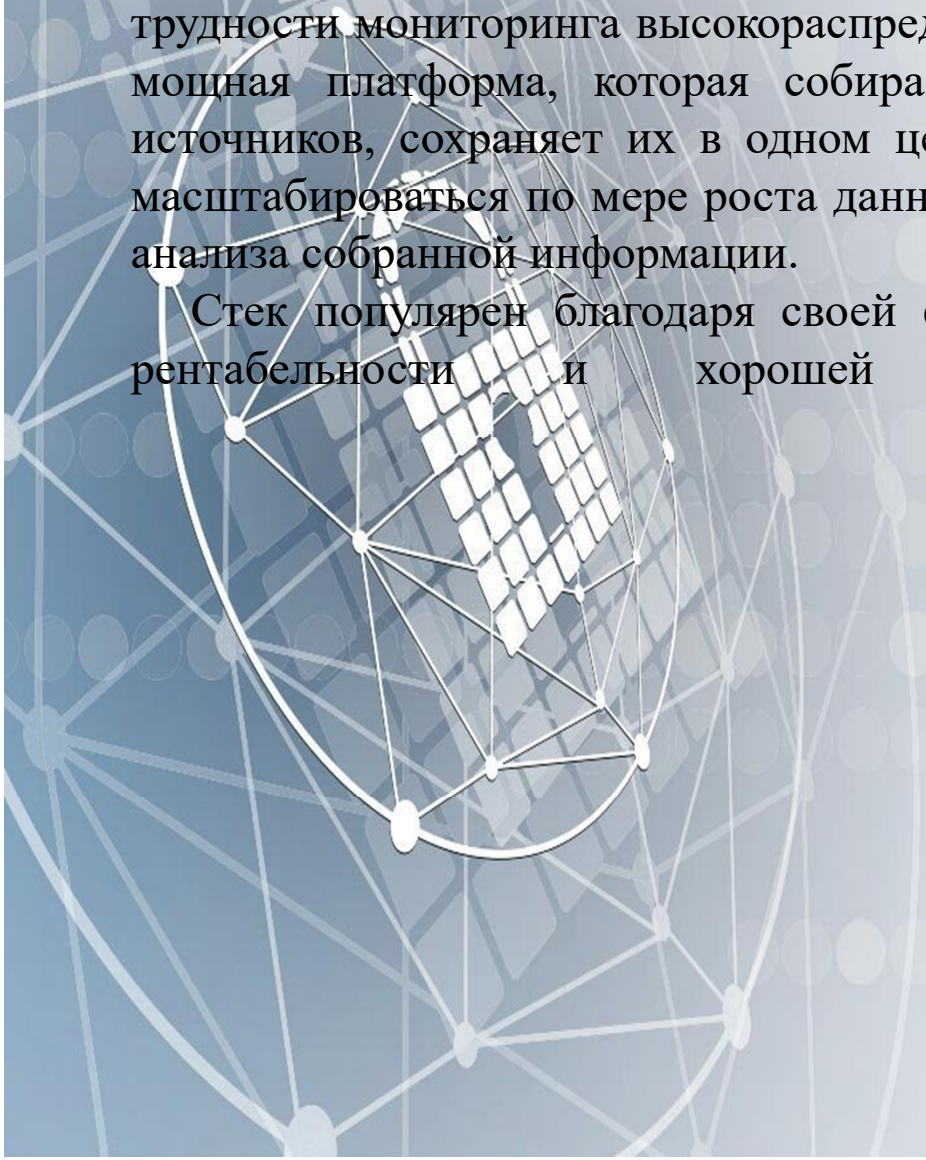
Недостатки:

- Различные компоненты в стеке могут стать трудными для обработки, в случае перехода к сложной настройке

Заключение

Стек ELK (Logstash, Elasticsearch, Kibana) – это современное решение для управления журналами и аналитикой, которое позволяет инженерам преодолевать трудности мониторинга высокораспределенных, динамических и шумных сред. Это мощная платформа, которая собирает и обрабатывает данные из нескольких источников, сохраняет их в одном централизованном хранилище, которое может масштабироваться по мере роста данных, и предоставляет набор инструментов для анализа собранной информации.

Стек популярен благодаря своей функциональности, простоте использования, рентабельности и хорошей поддержке активного сообщества.



СПАСИБО ЗА ВНИМАНИЕ!