**Системная и программная инженерия**

Лабораторная работа №7 «Знакомство с Elasticsearch»

**Введение**

**Elasticsearch** — это высокомасштабируемая распределенная поисковая система полнотекстового поиска и анализа данных с веб-интерфейсом, REST API и неформализованными JSON-документами, которая разработана на базе полнотекстового поиска Lucene и работает в режиме реального времени. Кроме того, Elasticsearch — это нереляционное хранилище документов в формате JSON, которое разработано на Java и выпущено как проект с открытым исходным кодом в соответствии с условиями лицензии Apache.

**Преимущества Elasticsearch**

* Выгодное соотношение цены и времени. Elasticsearch предлагает простые API на основе REST и легкий HTTP-интерфейс, а также использует документы JSON без схем, благодаря чему проще приступить к работе и быстро создавать приложения для различным примеров использования.
* Высокая производительность. Распределенная система Elasticsearch позволяет параллельно обрабатывать большие объемы данных, мгновенно подбирая наилучшее соответствие к запросу.
* Бесплатные инструменты и модули. Elasticsearch встроен в Kibana, популярный инструмент визуализации и составления отчетов. Доступна также интеграция с Beats и Logstash, при этом исходные данные легко преобразовывать и загрузить в кластер Elasticsearch. Можно использовать ряд подключаемых модулей Elasticsearch с открытым исходным кодом, таких как языковые анализаторы и механизмы рекомендаций, для более широкой функциональности ваших приложений.
* Операции в режиме, близком к реальному времени. Выполнение операций в Elasticsearch, таких как чтение или запись данных, обычно занимает менее секунды. Это позволяет использовать его в таких примерах, где необходимо реагировать почти в режиме реального времени, например для мониторинга приложений и обнаружения аномалий.
* Простая разработка приложений. Elasticsearch обеспечивает поддержку различных языков, включая Java, Python, PHP, JavaScript, Node.js, Ruby и многие другие.

**Как работает Elasticsearch?**

Вы можете отправлять данные в Elasticsearch в виде документов JSON с помощью API или инструментов приема, таких как Logstash и Amazon Kinesis Firehose. Elasticsearch автоматически сохраняет исходный документ и добавляет ссылку на него в индекс кластера, включая возможность поиска. Следом можно найти и извлечь документ, используя API Elasticsearch. Также для визуализации данных и создания интерактивных панелей управления можно задействовать Kibana – инструмент визуализации с Elasticsearch.

Данные в Elasticsearch организованы в виде индексов. Каждый индекс состоит из одного или нескольких шардов. Каждый сегмент представляет собой экземпляр индекса Lucene, который можно рассматривать как автономную поисковую систему, которая индексирует и обрабатывает запросы для подмножества данных в кластере Elasticsearch.

**Индекс** в Elasticsearch – это логическое пространство имен, которое сопоставляется с одним или несколькими первичными шардами и может иметь ноль или более шардов-реплик.

**Шард** — это единица, в которой Elasticsearch распределяет данные по кластеру. Скорость, с которой Elasticsearch может перемещать сегменты при перебалансировке данных, например, после сбоя, будет зависеть от размера и количества сегментов, а также производительности сети и диска.

**Практическая часть**

**Установка**

В данной лабораторной работе для удобства развертывания Elasticsearch рекомендуется использовать Docker. Предварительно необходимо скачать образ Elasticsearch следующей командой:

docker pull elasticsearch:7.8.0

Создадим Docker-сеть следующей командой:

docker network create elastic

Далее запустим контейнер с образом Elasticsearch следующей командой:

docker run -d --name elasticsearch --net elastic -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" elasticsearch:7.8.0

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Запуск контейнера

После старта контейнера необходимо немного подождать пока Elasticsearch запустится и будет готов принимать Http-запросы:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 – GET запрос на localhost:9200

Для взаимодействия с Elasticsearch в данной лабораторной работе мы будем использовать его HTTP API. Для этого рекомендуется скачать [Postman](https://www.postman.com/downloads/).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 – GET запрос на localhost:9200 из postman

Для добавления новых данных в Elasticsearch необходимо выполнить PUT-запрос по следующему пути: {index\_name}/\_doc/{id}. Где index\_name ­– название индекса (если такого имени нет, то создается автоматически), id – уникальный идентификатор записи. В тело нашего запроса следует поместить JSON-объект необходимый для индексации, например:

{

"fullName": "Иванов Иван",

"homeTown": "Саратов",

"formed": 2016,

"active": **true**

}

В результате мы получим следующий ответ от сервера с кодом 201:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 – создание новой записи в индексе

При повторном PUT-запросе по этому id запись будет перезаписана:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 – перезапись записи в индексе

Если необходимо удалить какую либо запись, то следует отправить DELETE-запрос по следующему пути: {index\_name}/\_doc/{id}. Где index\_name ­– название индекса, id – уникальный идентификатор записи. В результате мы получим следующий ответ с сервера:   
  
Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 – удаление записи из индекса

Для получения записи необходимо отправить GET-запрос по следующему пути: {index\_name}/\_doc/{id}. Где index\_name ­– название индекса, id – уникальный идентификатор записи. В результате мы получим следующий ответ с сервера:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7 – получение записи

Для вывода всех записей необходимо отправить GET-запрос по следующему пути: {index\_name}/\_search. Где index\_name ­– название индекса.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8 – получение всех записей

В Elasticsearch для поиска и фильтрации записей существует механизм составных поисковых запросов, например, чтобы вывести все записи в которых поле “active” = true

Необходимо отправить GET-запрос по следующему пути: {index\_name}/\_search. Где index\_name ­– название индекса. А в теле запроса JSON–объект:

{

"query":{

"match":{

"active":**true**

}

}

}

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 9 – поисковый запрос в Elasticsearch

Тема поиска в Elasticsearch достаточно обширна, более подробно о ней можно почитать в официальной [документации](https://www.elastic.co/guide/en/elasticsearch/reference/current/search-your-data.html).

**Практическое задание**

1. Необходимо добавить 5 произвольных записей в индекс, которые описывают студента (ФИО, пол, группа), и вывести их.
2. На основании 1-го практического задания продемонстрировать поиск по имени в Elasticsearch