Prácticas de laboratorio del curso

«Sistemas de análisis de grandes datos»

ПОЛИТЕХ

Semestre de primavera
Stack de tecnologías ELK (Elasticsearch, Logstash, Kibana)

Uno de los procesos clave en el desarrollo de software es el **registro de eventos (logging)**. Con el tiempo, los programas se vuelven más complejos, los archivos de registro (logs) más voluminosos y, en consecuencia, la navegación por ellos más difícil. A lo largo del tiempo surgió la necesidad de herramientas especializadas que permitan trabajar de manera rápida y cómoda con los registros.
Una de estas soluciones es el **conjunto de programas ELK Stack**, que consta de tres productos de software principales de código abierto: **Elasticsearch**, **Logstash** y **Kibana**.
A veces este conjunto se complementa con programas de terceros, pero estos "tres pilares" siguen siendo las herramientas fundamentales.

- **Elasticsearch** es un sistema escalable y distribuido de almacenamiento y búsqueda de datos con funcionalidad flexible y amplias posibilidades de configuración. Está construido sobre **Apache Lucene**, añadiendo innovaciones que permiten realizar búsquedas rápidas y eficientes en proyectos con grandes volúmenes de datos.
- **Logstash** es una aplicación para recopilar información desde diversas fuentes, transformarla a un formato conveniente para su análisis y enviarla a un almacenamiento para su posterior uso. Su facilidad de uso y capacidad para manejar grandes volúmenes de datos le otorgan ventajas frente a proyectos similares.
- **Kibana** es un complemento diseñado específicamente para Elasticsearch. Se encarga de la **visualización de datos, análisis y presentación de información** en un formato comprensible y fácil de interpretar. Esta herramienta permite analizar rápidamente los resultados de búsqueda e identificar patrones, además de ofrecer amplias posibilidades de configuración.

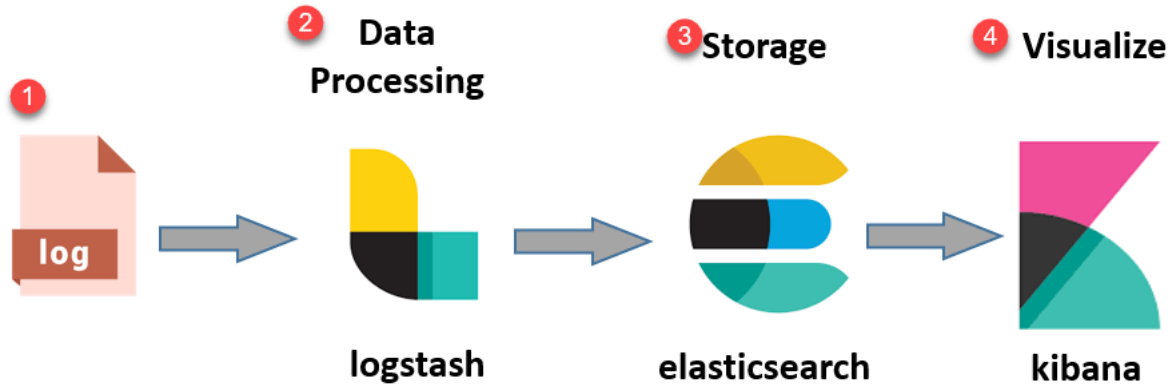
Así, el **mecanismo de recopilación de logs** funciona de la siguiente manera:

- **Logstash** recopila los registros y los coloca en el almacenamiento.
- **Elasticsearch** se utiliza para buscar las líneas necesarias dentro de esos registros.
- **Kibana** permite analizar y visualizar los resultados.

El conjunto de estos productos es una excelente solución para la búsqueda y resolución rápida de errores en el código, y una herramienta muy conveniente para los desarrolladores, especialmente aquellos que trabajan en grandes proyectos. Además, la funcionalidad del stack

ELK permite usarlo como **almacenamiento centralizado de registros**, **agregador de eventos con navegación avanzada**, **sistema analítico con algoritmos de aprendizaje automático**, y mucho más.

Arquitectura modular de la tecnología:



Objetivos del curso (semestre de primavera):

- Aprender a **configurar un entorno funcional** con uno de los stacks tecnológicos más populares para el análisis y visualización rápida de datos.
- Adquirir **habilidades prácticas en el uso de las tecnologías del stack ELK** para resolver tareas orientadas a la práctica.
- Familiarizarse con el **REST API (de Elasticsearch)** y con la **interfaz gráfica de Kibana**.

---

1. Instalación del stack ELK

Es necesario **instalar y configurar** Elasticsearch, Kibana y Logstash.
Los distribuibles en forma de archivo comprimido pueden descargarse desde los siguientes enlaces:

- Kibana 7.15.2
- Elasticsearch 7.15.2
- Logstash 7.15.2

Las instrucciones necesarias para la instalación pueden encontrarse en Internet.
En el sitio oficial están disponibles las guías:

- Instalación de Elasticsearch
- Instalación de Kibana
- Instalación de Logstash

También existen otras fuentes, por ejemplo, cómo instalar ELK en Ubuntu:

- Versión en inglés (Ubuntu 14.04)
- Versión en ruso (Ubuntu 18.04)

Quienes dominen **Docker** pueden utilizar el siguiente enlace para ejecutar Elasticsearch y Kibana en contenedores:

- https://opendistro.github.io/for-elasticsearch/#

---

2. Descarga del conjunto de datos para el análisis

Debe accederse al sitio **Kaggle.com**, escribir en la barra de búsqueda **«headhunter»** y descargar el conjunto de datos **«HeadHunter Employer Review Competition»**.
Dentro del archivo descargado se encuentran tres archivos CSV. Nos interesa el archivo **HeadHunter_train.csv**, que contiene un conjunto anonimizado de reseñas sobre empleadores del sitio hh.ru.

---

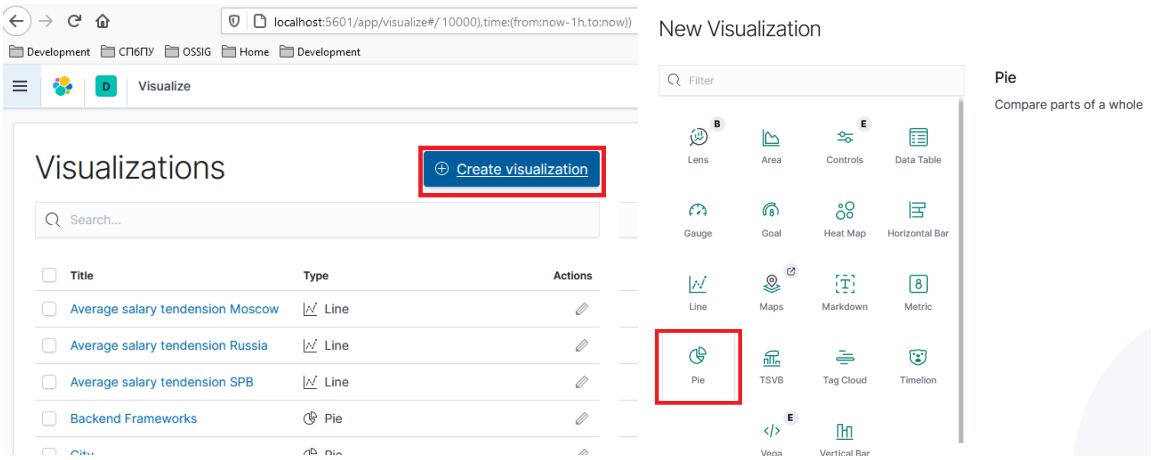
3. Importación de datos en Elasticsearch

Es necesario configurar **Logstash** para que el archivo CSV sea **convertido a formato JSON** y **cargado en Elasticsearch**.
Ejemplo de configuración:
https://www.zenitk.com/import-from-csv-to-elasticsearch-with-logstash

---

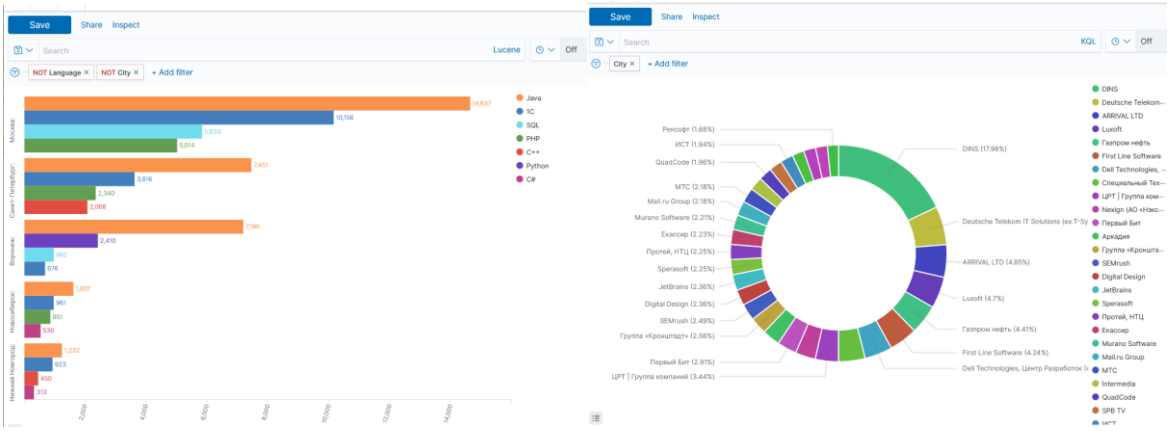
4. Creación de gráficos y visualizaciones

Debe añadir **dos nuevas visualizaciones** de datos desde la interfaz de **Kibana**
(Menú principal → Kibana → Visualize):



- **Gráfico de barras horizontal (Horizontal bar chart)** – distribución de reseñas según la **valoración del empleador** (campo workplace_rating).
- **Gráfico circular (Pie chart)** – distribución de reseñas según la **ciudad** (campo city).

Ejemplos de visualización «Horizontal bar chart» y «Pie chart»:



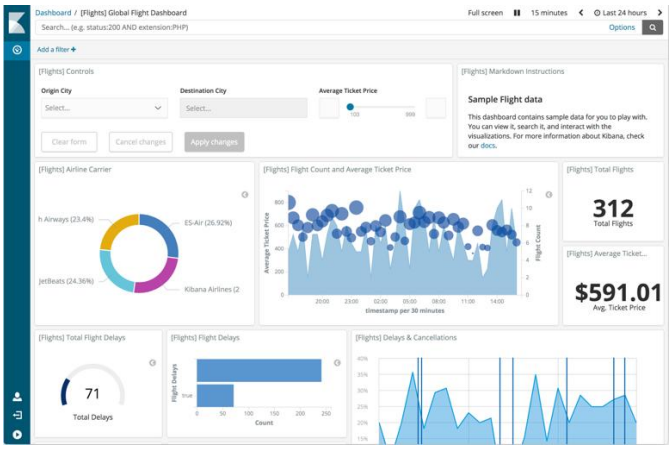
---

5. Creación de paneles (dashboards)

Un **panel (dashboard)** permite mostrar varias visualizaciones en una sola página, logrando una **presentación eficiente e intuitiva** de la información.

Debe **crear un nuevo dashboard** y añadirle las dos visualizaciones creadas anteriormente.

Ejemplo de Dashboard:



---

6. Búsqueda mediante REST API

Además del uso del **interfaz web (pestaña Discover)** para buscar datos, también puede emplearse el **REST API** que proporciona Elasticsearch.

Los **consultas REST** pueden ejecutarse mediante herramientas como **Postman**: https://www.postman.com/

Ejemplo de búsqueda de un documento original por su **ID**:
Formato de consulta:

<URL-de-Elasticsearch>/<nombre_del_índice>/_search?q=<campo>:<valor>

Ejemplo:

http://localhost:9200/vacancyidx/_search?q=employer.name:Dell

Más detalles en la documentación oficial:
https://www.elastic.co/guide/en/elasticsearch/reference/current/search-search.html

---

En el marco de esta práctica, se debe desarrollar un programa propio

La aplicación debe recibir una **consulta de búsqueda** como entrada y mostrar los **resultados obtenidos de Elasticsearch** como salida.
El **lenguaje de programación y las bibliotecas** a utilizar son de **libre elección**, aunque se recomienda que la aplicación tenga **interfaz gráfica**, no solo de consola.

Durante la defensa de las prácticas de laboratorio, se debe demostrar:

1. El **stack ELK instalado y funcionando**
2. La **configuración de Logstash**
3. Los **datos cargados en Elasticsearch** (pestaña *Discover* en Kibana)
4. El **dashboard** creado con las dos visualizaciones
5. La **aplicación desarrollada** para búsqueda en Elasticsearch mediante **REST API**