IST769 Lab J

# Search Model: Elasticsearch and Kibana

### In this lab, we will explore the search index model with the Elasticsearch database and Kibana data visualization tool. We will use Kibana to create maps, dashboards, and canvases from data streamed to Elasticsearch.

### Learning Outcomes

At the end of this lab you should be able to:

* Query data in Elasticsearch using the REST API.
* Import and export data from Elasticserch into spark dataframes.
* Use Kibana to create index patterns and searchable fields.
* Build Dashboards, Maps and canvases in Kibana.

### Pre-Requisites

Before you begin:

* Open a terminal window in the lab environment
* Set the current working directory to **advanced-databases**
* Start the following services required by the lab:   
  **jupyter elasticsearch kibana**

### Tools Used In this Lab

The following tools will be used in this lab:

1. To access Jupyter Lab from your Windows host:  
   <http://localhost:8888>   
   The password is **SU2orange!**
2. To access the Kibana Dashboard Use  
   <http://localhost:5601>
3. To access The Elasticsearch REST Client from your windows computer:  
   <http://localhost:9200>
4. To access The Elasticsearch REST Client from a terminal in Jupyter  
   <http://elasticsearch:9200>

# Lab Problem Set

**QUESTIONS:   
  
For each question, include a copy of the code required to complete the question along with a screenshot of the code and a screenshot of the output.**

1. Turn on tweets from the example **J-Elasticsearch** notebook. Search for tweets in the last 5 minutes from the (windows or linux) with **curl** (do the math yourself – don’t make this calculation automatic).
2. Do the same thing from the Kibana UI for Elasticsearch. What must you do before you can do that and why? Turn off the tweets.
3. Write PySpark to load the 1,600 line weather data set into Elasticsearch under the index `weather` with default index type.
4. Use a **curl** command from the command line to hit the Elasticsearch API and demonstrate that there are 1,600 documents in the **weather** index.
5. Setup a **weather** index pattern in Kibana based on the **weather** index from Elasticsearch. Make sure you have a **geo\_point** based on lat/lon type and have selected a **@timestamp** field using the date field. Provide a screenshot including the fields in question.
6. Demonstrate your Kibana index pattern is functional. For the most recent entry get the weather for a City of your choice.
7. Create a Kibana map displaying the weather locations for the most recent weather data. Use any layer(s) of data you wish. Provide a screenshot of the map with data points on it.
8. Create a Kibana dashboard which when you select a city, will display the average day time and nighttime temperature for that city, in addition to a line chart of the average daily high and lows for all data on that city. Provide a screenshot of the dashboard in action.
9. Create a Kibana Canvas! Display at least 2 metrics and 2 charts. Decide which data you want to display and how you would like to present it. Provide a screenshot of the Canvas. You can do this for a set of cities or a single city.

**IMPORTANT:** When you are finished with the lab, execute:

PS:> docker-compose stop

To turn off all running services, then shut down your Azure Lab instance.