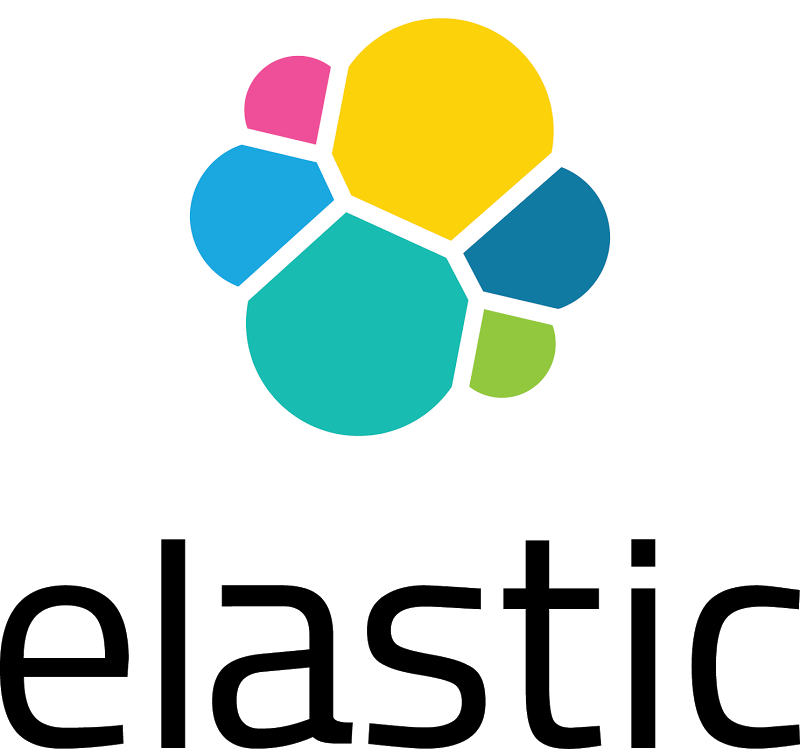
Graph 5.5 Hands On Lab ****

## Goal:

To understand how to install and validate Graph.

To perform an analysis with a relevance graph

## Part 1: Explore Flights Data with Graph

|  |  |
| --- | --- |
| 1. | * Confirm you are logged into Kibana as the elastic user. * Click Graph in Kibana * Choose the flights-\* index pattern * Click the + icon and add the CARRIER field * Choose purple as color, and the airplane icon. |
| 2. | Seed the graph with specific carriers   * Search for each of these one at a time:   + CARRIER:UA   + CARRIER:AA   + CARRIER:DL   + CARRIER:WN |
| 3. | * Click the + icon and add the ORIGIN field, choose green as a color and the flag as an icon * Click the + icon and add the DEST field, choose red as a color and the flag as an icon * Shift click on CARRIER to disable it from returning results. The ORIGIN and DEST remain selected. * Identify routes for each CARRIER node:   + Select an individual CARRIER node   + Click the Expand Selection button:   + Repeat for each additional CARRIER |
| 4. | Notice that graph can tell what the hubs are for each of the carriers just based on the raw data. But now let’s see what airports are actually more shared between carriers:   * Select All * Click the Add Links icon 3 times   Now note that some ORIGIN and DEST are now connected to multiple carriers, also note that some hubs remain only with their controlling airlines.  Remember that graph doesn’t know what we have learned about airports and the dominant carriers, it learned this from the data indexed and can highlight connections in your own data. |
| 5. | Save this workspace for use in Step 2. Call it Airlines Graph. |

## 

## Step 2: Build a visualization for a drill down

|  |  |
| --- | --- |
| 1. | In a new Kibana tab, Click on Visualize and create a new Vertical bar chart as an example   * Choose the flights-\* index * Select an absolute time range of January 16 2015 00:00:00 to December 29 2015 00:00:00 * For the X axis, select   + Aggregation: Histogram   + Field: ARR\_DELAY   + Interval: 5     *Note the shape of the chart to compare to later. If no results appear, check the time frame in the upper right hand corner.* |
| 2. | Save this Visualization as Delay Histogram |
| 3. | Copy the URL from this visualization.   * Click on Graph * Click on Settings in the upper right hand corner   + Drill down   + Paste the URL you copied and answer yes to converting it to a template   + Write a title   + Choose an OR or AND query from the URL parameter type drop-down   + Select a toolbar icon   + Save the Drill-down definition |
| 4. | Create a graph of carriers and/or airports and select some of the nodes with shift clicks, then click on the drill down icon. Choose “Delay Histogram”.    *Note the shape of the ARR\_DELAY histogram changes depending on selection.* This can also be done for an entire dashboard. |
| 5. | Build an external site drill down:   * Open the Graph workspace you saved in Step 1 * Click on Settings   + Click on Drill Down * Enter in the url: https://en.wikipedia.org/wiki/{{gquery}} * URL parameter type: plain text * Title: wikipedia * Icon: whichever you choose, remember it * Save the new drill down |
| 6. | Select a single ORIGIN or DEST node on your workspace. Click on the icon you chose in the last step. You’re being redirected to perform a search on Wikipedia for more information on the airport code if you don’t know which airport it is. |
| 7. | Additional challenges you might want to try:   * Create a new Graph workspace, and seed the graph with some range queries for CARRIER, ORIGIN, and DEST   + TAXI\_OUT:[20 TO \*]   + TAXI\_IN:[20 TO \*] * Note the graph that forms showing you what airports and carriers have a high delay to taxi out from the gate. Choose other fields to range query and experiment with the results. * You can use any of the numeric fields to do this graph seeding query, use discover to experiment with your own theory of delays either delay or taxi related. |

## 