

Module

# Kibana for Time Series Data



# Topics

- Kibana Fundamentals
- Visualizations and Dashboards
- Time-Series Visualization Builder (TSVB)



Lesson 1

# Kibana Fundamentals



# Event Data

# Configuring Index Patterns

- Typically, Kibana data does contain "time-based" events, which contain a timestamp and some data
- Kibana requires *index patterns* in order to find this data as it lives in Elasticsearch
- Commonly, *time-series* indices will include some sort of timestamp in the index name
  - (e.g. "**apachelogs-2018.12.15**")
- Other times, the data may live in one *static* index
  - (e.g. "**products**")
- Kibana prefers a timestamp field, so that it can render visualizations over time (for example date histograms)
  - (e.g. "**@timestamp**")

# First look at Kibana

- Click on Management in the left nav to begin

The screenshot shows the Kibana home page. On the left, there is a vertical navigation bar with various icons. The icon for 'Management' (a gear) is circled in red. The main content area has three main sections: 'Add Data to Kibana' (with APM, Logging, and Metrics options), 'Visualize and Explore Data' (with Add sample data and Upload data from log file options), and 'Manage and Administer the' (with Index Patterns, Saved Objects, Spaces, Reporting, and Advanced Settings options). The 'Index Patterns' option is also circled in red.

Add Data to Kibana  
Use these solutions to quickly turn your data into pre-built dashboards and monitoring systems.

**APM**  
APM automatically collects in-depth performance metrics and errors from inside your applications.  
[Add APM](#)

**Logging**  
Ingest logs from popular data sources and easily visualize in preconfigured dashboards.  
[Add log data](#)

**Metrics**  
Collect metrics from the operating system and services running on your servers.  
[Add metric data](#)

**Add sample data**  
Load a data set and a Kibana dashboard

**Upload data from log file**  
Import a CSV, NDJSON, or log file

**Manage and Administer the**

- Kibana
- Index Patterns**
- Saved Objects
- Spaces
- Reporting
- Advanced Settings

# Index Patterns

# Configuring an index pattern

- Type "apachelogs-\*" in the Index pattern box then click on the "Next step" button

The screenshot shows the 'Create index pattern' interface in the Elasticsearch Management section. On the left, there's a sidebar with various management options like Index Management, Index Lifecycle Policies, and Watcher. The main area is titled 'Create index pattern' and includes a sub-section 'Step 1 of 2: Define index pattern'. In the 'Index pattern' input field, the value 'apachelogs-\*' is entered and highlighted with a red circle. Below the input field, there's a note: 'You can use a \* as a wildcard in your index pattern. You can't use spaces or the characters \, /, ?, ", <, >, |.' A success message indicates 'Success! Your index pattern matches 31 indices.' A list of matching indices is shown, starting with 'apachelogs-2019-10-17'. To the right of the input field, there's a checkbox for 'Include system indices' and a 'Next step' button, which is also highlighted with a red circle.

# Picking the time filter field

- Select the field which Kibana will use to render time-based visualizations: **@timestamp** works well for us here. Then click the Create Index pattern button.

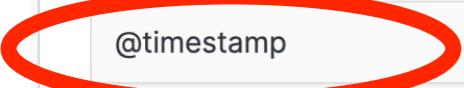
## Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.   Include system indices

### Step 2 of 2: Configure settings

You've defined **apachelogs-\*** as your index pattern. Now you can specify some settings before we create it.

Time Filter field name [Refresh](#)

**@timestamp** 

The Time Filter will use this field to filter your data by time.  
You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

[Show advanced options](#)

[Back](#) **Create index pattern** 

# Index Pattern successfully created

- After successful creation, you will be able to view the various fields which the documents in this pattern contain

★ apachelogs-\* ★ ⌂ ⚡

Time Filter field name: @timestamp Default

This page lists every field in the **apachelogs-\*** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch [Mapping API](#) ⚡

| Fields (73)  | Scripted fields (0) | Source filters (0) |            |              |          |
|--|---------------------|--------------------|------------|--------------|----------|
| <input type="text"/> Filter <span style="float: right;">All field types ▾</span> |                     |                    |            |              |          |
| Name   | Type                | Format             | Searchable | Aggregatable | Excluded |
| @message   | string              |                    | ●          |              | ✎        |
| @message.keyword   | string              |                    | ●          | ●            | ✎        |
| @timestamp ⓘ   | date                |                    | ●          | ●            | ✎        |
| @version   | string              |                    | ●          | ●            | ✎        |
| _id  | string              |                    | ●          | ●            | ✎        |
| _index   | string              |                    | ●          | ●            | ✎        |
| _score   | number              |                    |            |              | ✎        |
| _source  | _source             |                    |            |              | ✎        |
| _type  | string              |                    | ●          | ●            | ✎        |
| agent  | string              |                    | ●          |              | ✎        |

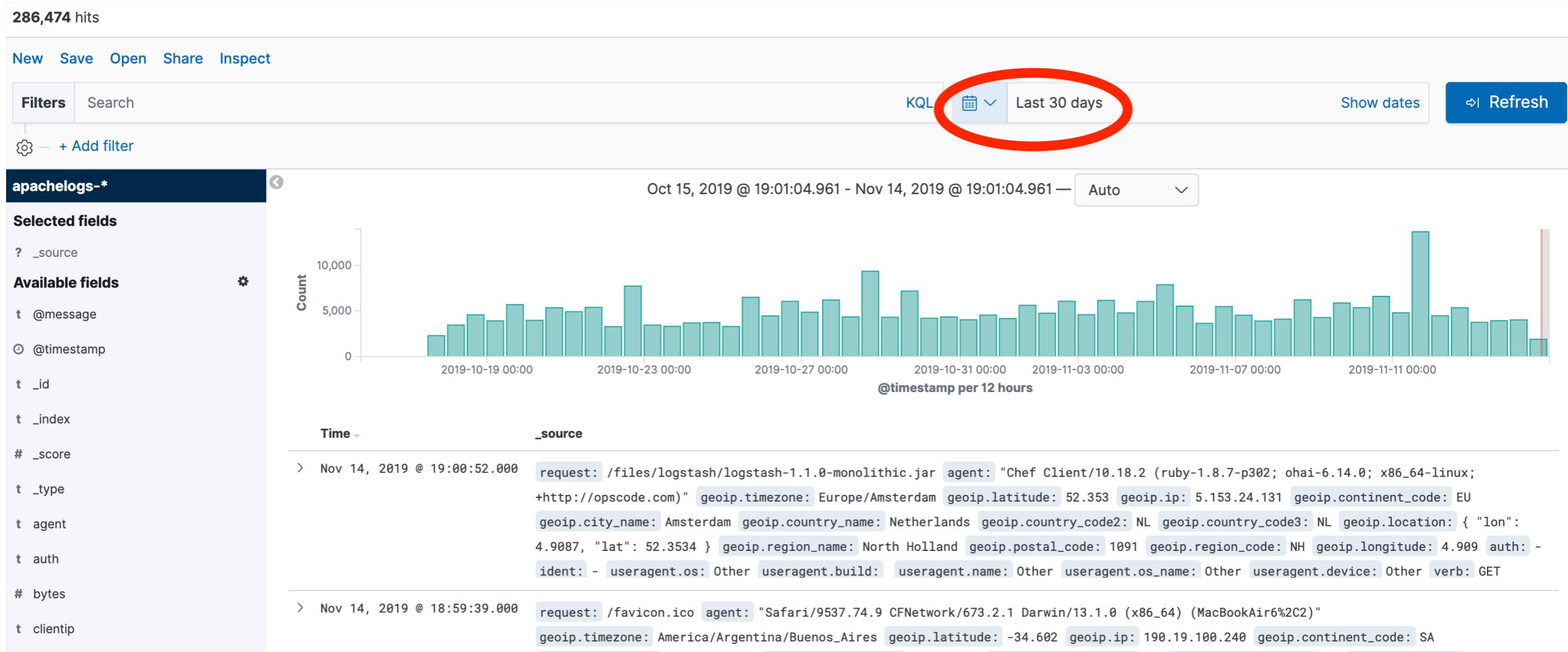
Rows per page: 10 ▾ < 1 2 3 4 5 ... 8 >

# Discover



# Discover Main Page

- Clicking on the Discover tab on the left nav brings you here
- Click to ensure the time picker reflects "Last 30 days" for your lab's dataset



# Time Picker

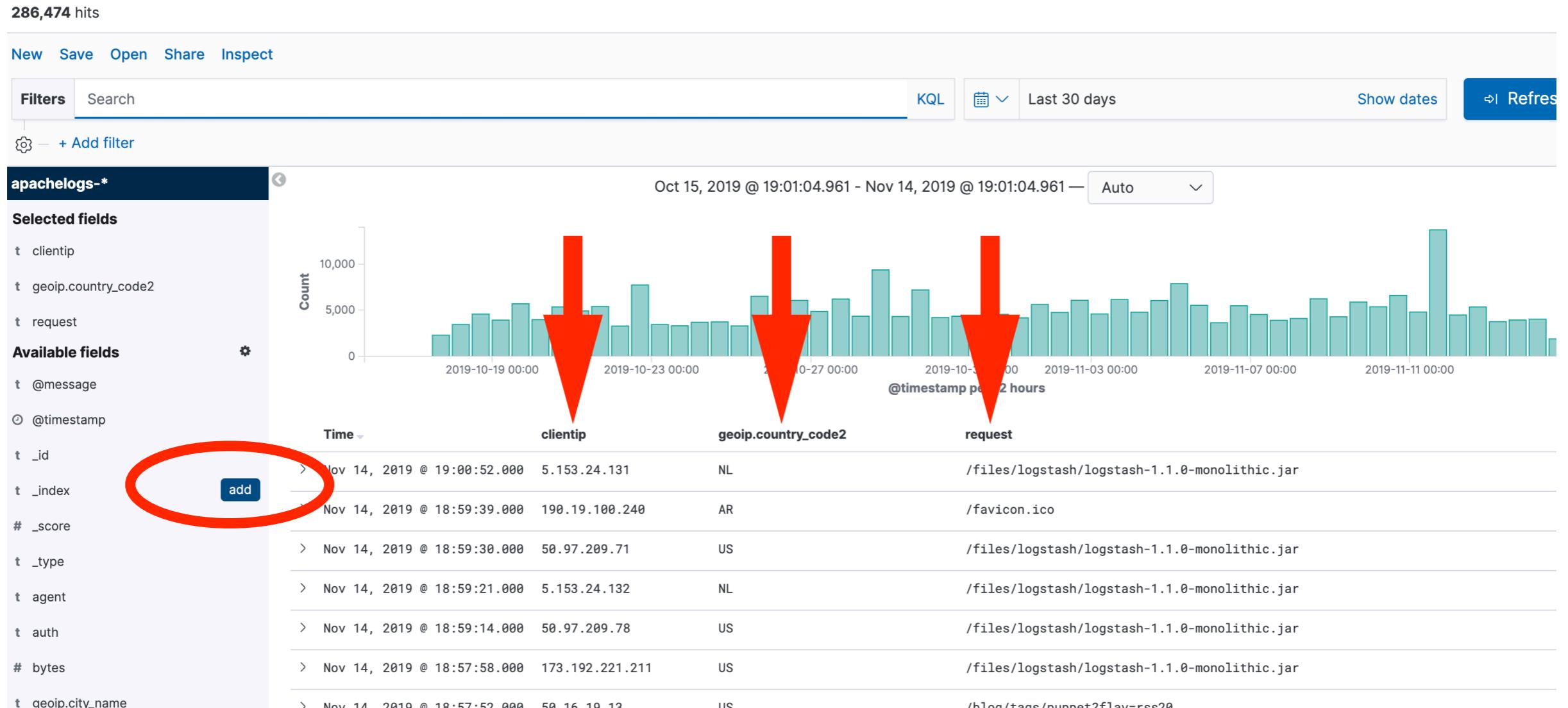
- Quick: convenient shortcuts
- Relative: "last 7 days", "a week ago until a day ago" etc
- Absolute: fine-grained selection using calendar
- Recent: your history

The screenshot shows a time picker interface with the following sections:

- Quick select:** A row of input fields and buttons. The first field contains "Last" with a dropdown arrow, the second contains "30", the third contains "days" with a dropdown arrow, and the fourth is a blue "Apply" button. Navigation arrows < and > are positioned above and to the right of the "Apply" button.
- Commonly used:** A list of pre-defined time intervals:
  - Today
  - Last 15 minutes
  - Last 1 hour
  - Last 7 days
  - Last 90 days
  - This week
  - Last 30 minutes
  - Last 24 hours
  - Last 30 days
  - Last 1 year
- Recently used date ranges:** A list of recently used time intervals:
  - Last 30 days
- Refresh every:** A row of input fields and buttons. The first field contains "0", the second contains "seconds" with a dropdown arrow, and the third is a blue "Start" button.

# Customized Columnar View

- Add columns to view from the left nav:



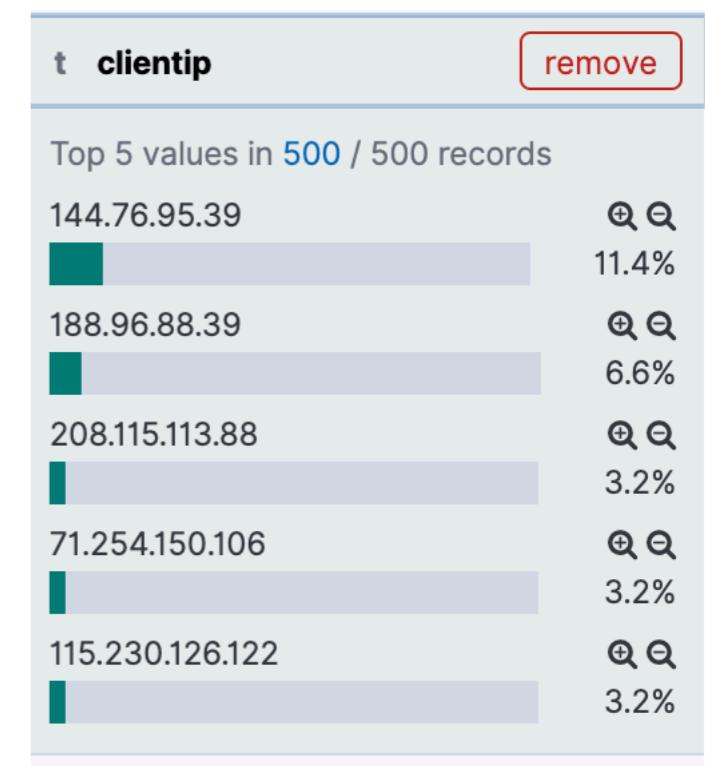
# Field-level controls

- Click "remove" to delete column from custom view

Selected fields

- t clientip
- t geoip.country\_code2
- t request

remove



- Click fieldname to review sampled stats:

# Document view

- Table View (shown here) or original JSON
- Can create a filter easily here!
- Link to view document or documents "surrounding" this one

| Time                          | clientip       | geoip.country_code2 | request                                       |
|-------------------------------|----------------|---------------------|---|
| > Nov 14, 2019 @ 19:00:52.000 | 5.153.24.131   | NL                  | /files/logstash/logstash-1.1.0-monolithic.jar |
| Nov 14, 2019 @ 18:59:39.000   | 190.19.100.240 | AR                  | /favicon.ico                                  |

[Expanded document](#) [View surrounding documents](#) [View single document](#)

[Table](#) [JSON](#)

|       |                      |   |
|-------|----------------------|---|
| t     | @message             | 190.19.100.240 GET /favicon.ico   |
| ⌚     | @timestamp           | Nov 14, 2019 @ 18:59:39.000   |
| t     | _id                  | y540am4B0HbrE43Q5tBM  |
| t     | _index               | apachelogs-2019-11-14   |
| #     | _score               | -   |
| t     | _type                | _doc  |
| 🔍     | 🔍                    | *   |
| agent |                      | "Safari/9537.74.9 CFNetwork/673.2.1 Darwin/13.1.0 (x86_64) (MacBookAir6%2C2)" |
| t     | auth                 | -   |
| #     | bytes                | 3,638   |
| t     | clientip             | 190.19.100.240  |
| t     | geoip.city_name      | Buenos Aires  |
| t     | geoip.continent_code | SA  |
| t     | geoip.country_code2  | AR  |

# Surrounding Document view

- Five before/after by default, but you can load more
- Before/after determined by sorting on timestamp field

⚙️ — + Add filter

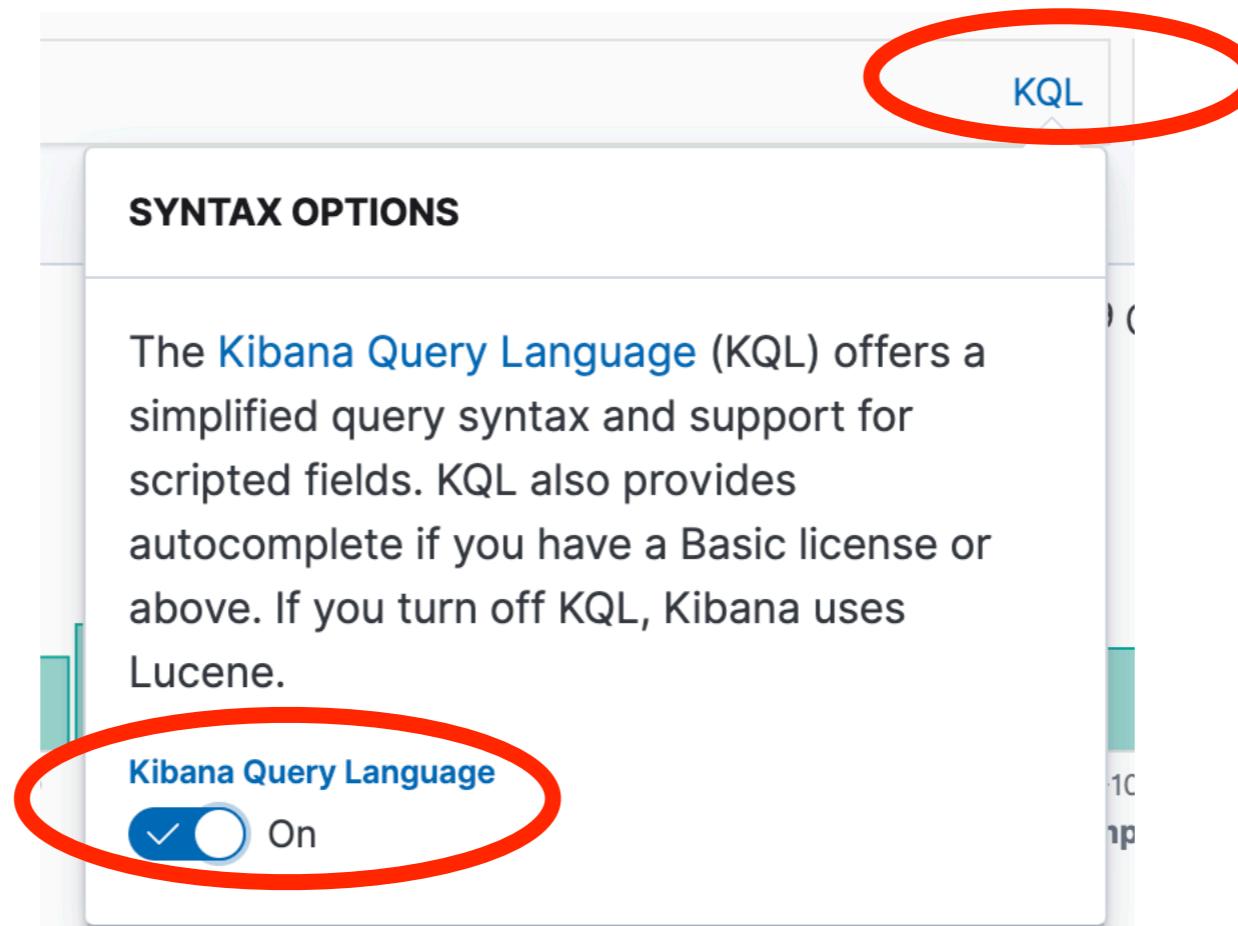
▲ Load 5 more 5 newer documents

| Time                          | clientip        | geoip.country_code2 | request                                       |
|-------------------------------|-----------------|---------------------|---|
| > Nov 14, 2019 @ 18:53:33.000 | 87.98.155.12    | FR                  | /   |
| > Nov 14, 2019 @ 18:53:30.000 | 74.125.40.22    | US                  | /?flav=rss20                                  |
| > Nov 14, 2019 @ 18:53:21.000 | 195.248.32.227  | AT                  | /favicon.ico                                  |
| > Nov 14, 2019 @ 18:52:45.000 | 199.217.118.68  | US                  | /blog/tags/tv                                 |
| > Nov 14, 2019 @ 18:52:34.000 | 199.217.118.68  | US                  | /blog/tags/nethack                            |
| > Nov 14, 2019 @ 18:52:26.000 | 199.217.118.68  | US                  | /blog/geekery?page=8                          |
| > Nov 14, 2019 @ 18:52:24.000 | 178.137.165.99  | UA                  | /blog/tags/rants                              |
| > Nov 14, 2019 @ 18:52:20.000 | 173.193.192.137 | US                  | /files/logstash/logstash-1.1.0-monolithic.jar |
| > Nov 14, 2019 @ 18:52:10.000 | 199.217.118.68  | US                  | /blog/geekery/nethack-bingesoft-testing.html  |
| > Nov 14, 2019 @ 18:52:02.000 | 199.217.118.68  | US                  | /blog/?page=35                                |
| > Nov 14, 2019 @ 18:51:53.000 | 46.105.14.53    | FR                  | /blog/tags/puppet?flav=rss20                  |
| ▼ Load 5 more                 | 5               | older documents     |   |

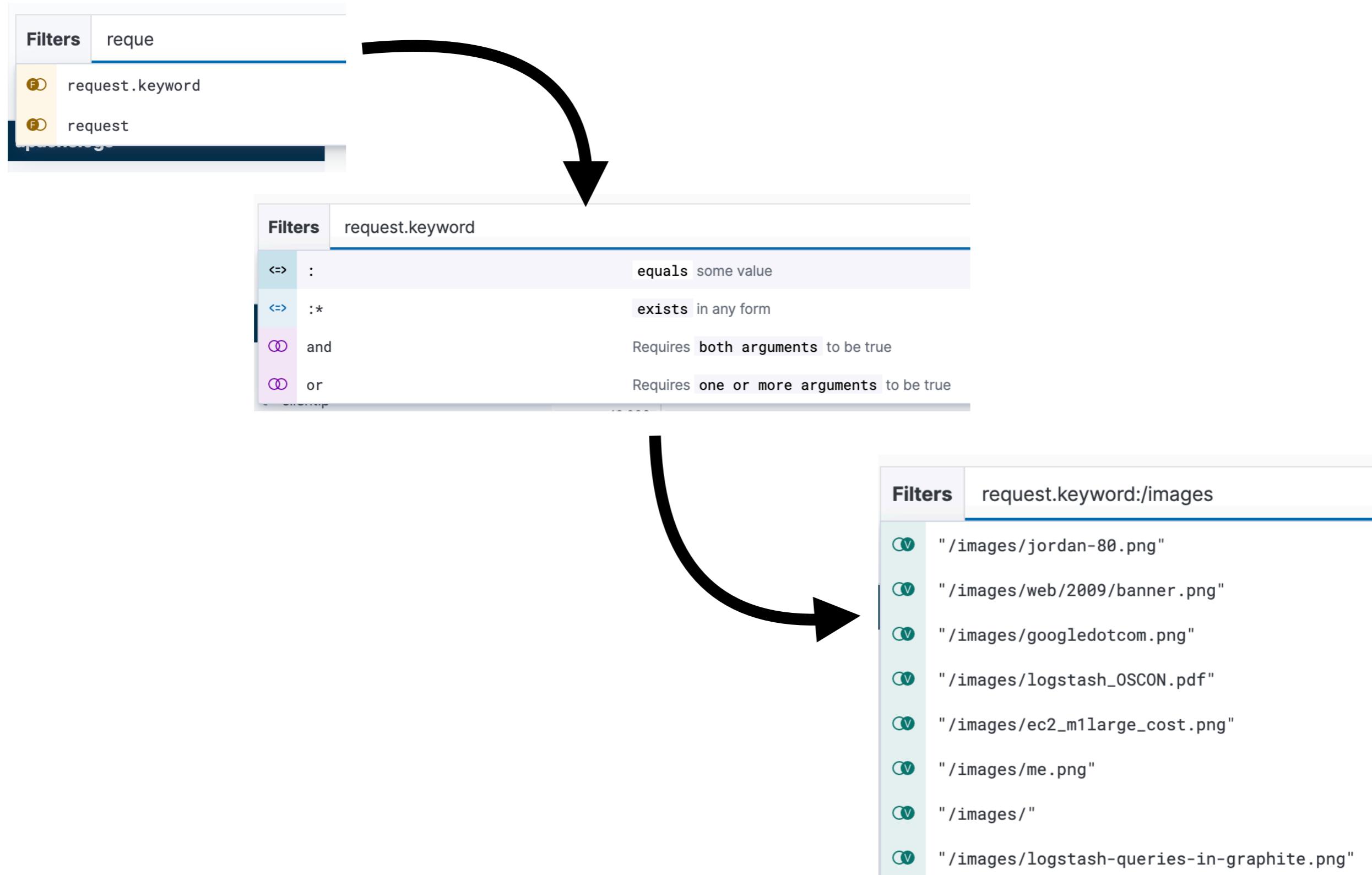
# Text Search

# Text search

- The default search is Kibana Query Language (KQL)
- Click the “KQL” link and turn off KQL to enable Lucene Query Syntax

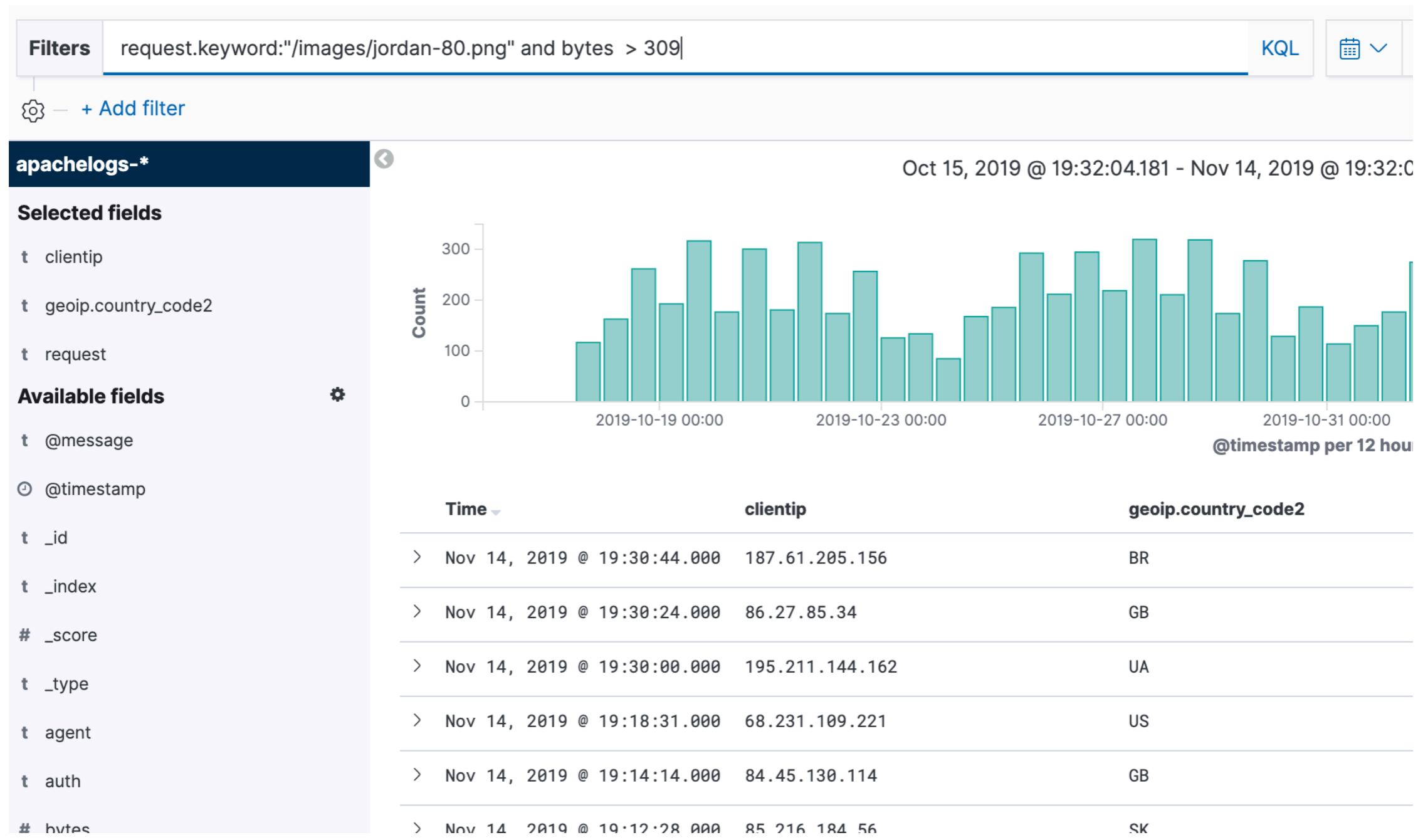


# Auto-complete in Kibana query language



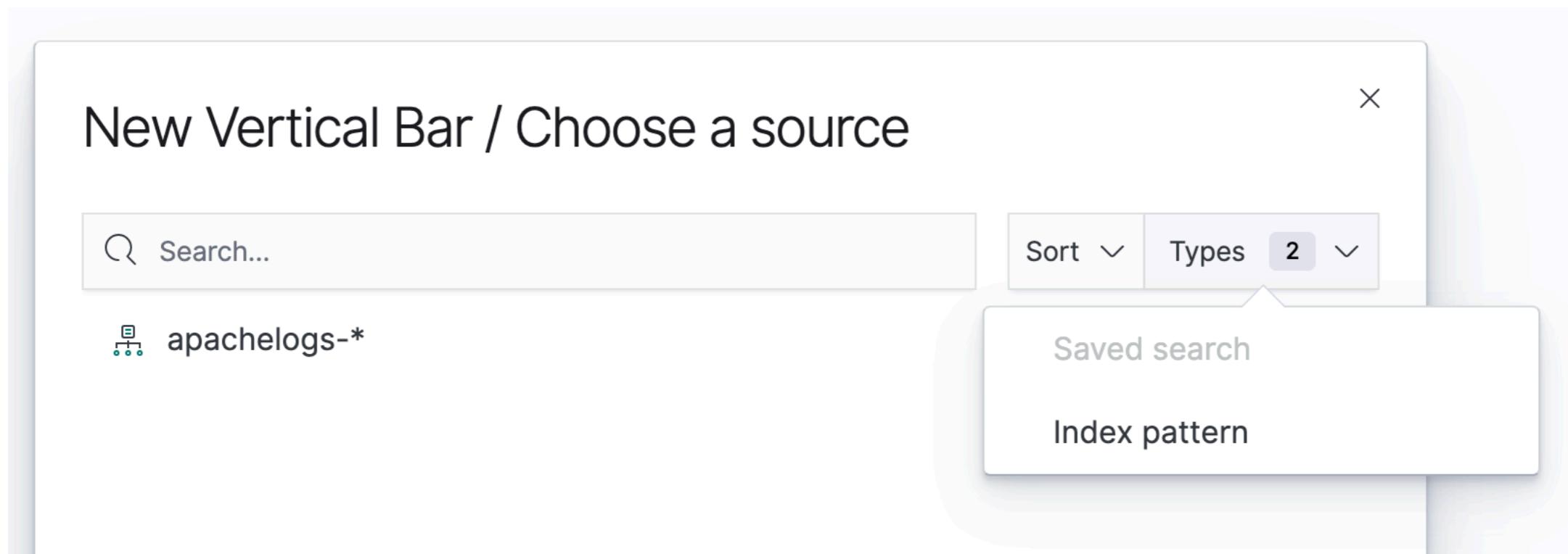
# Compound Boolean Queries

- Use operators such as **and**, **or**, **not** in Kibana queries



# Saved searches

- Any search can be saved
- Any time you find an interesting search, go ahead and save it for later use (just like creating filters)
- Saved searches can be attached to visualizations in the Create Visualization screen



# Manipulating Filters

# Using and Manipulating Filters

- Filters are an important tool in your discovery process
- They can help you include or exclude documents in searches as well as visualizations
- Look for the "magnifying glass" icons, or click on any element of a visualization representing a value
- It's easy to toggle a filter on and off for comparison
- "Pin" filters and they will follow you throughout Kibana
- Filters can be associated with a visualization, also



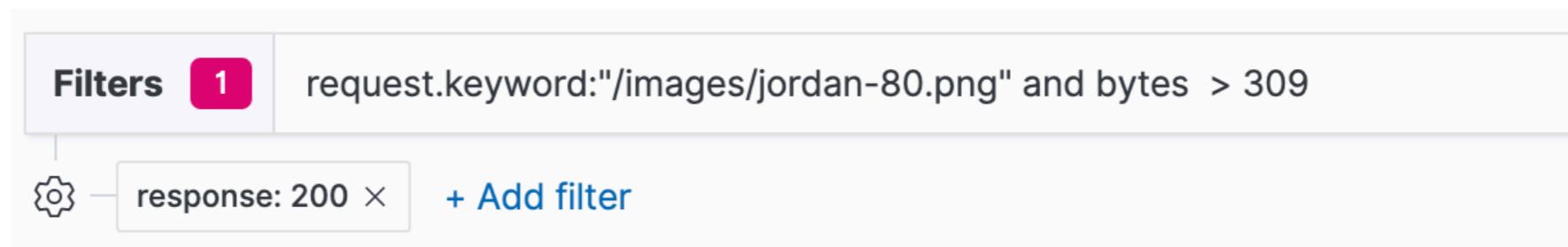
# Creating a filter by clicking on a value

- Anywhere Kibana shows you a particular value, you can create a filter by clicking on it. Two common examples:



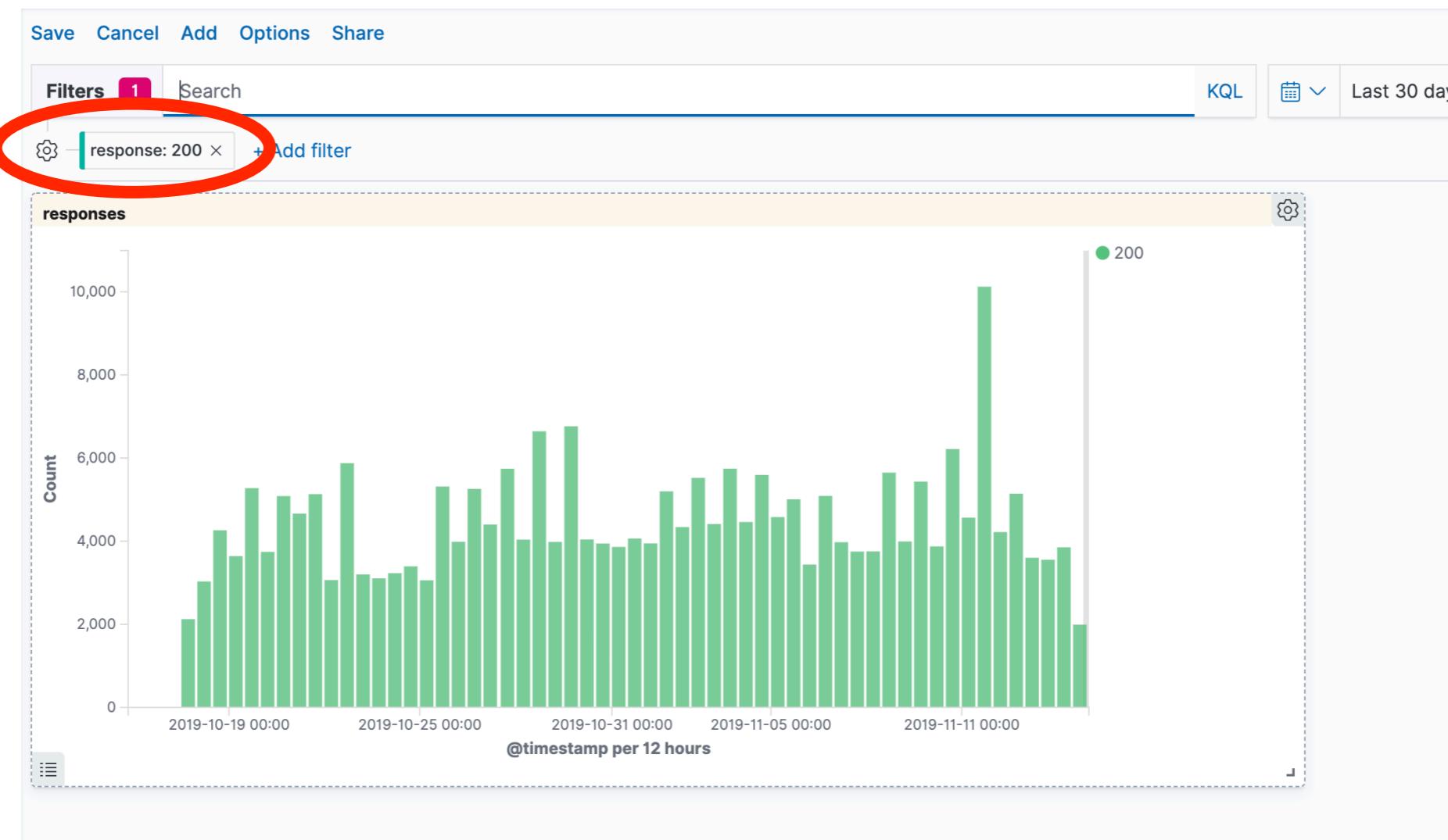
# Applying Filters

- Kibana will prompt you to apply the filters
- In the example, we can apply a filter such as **response:404**



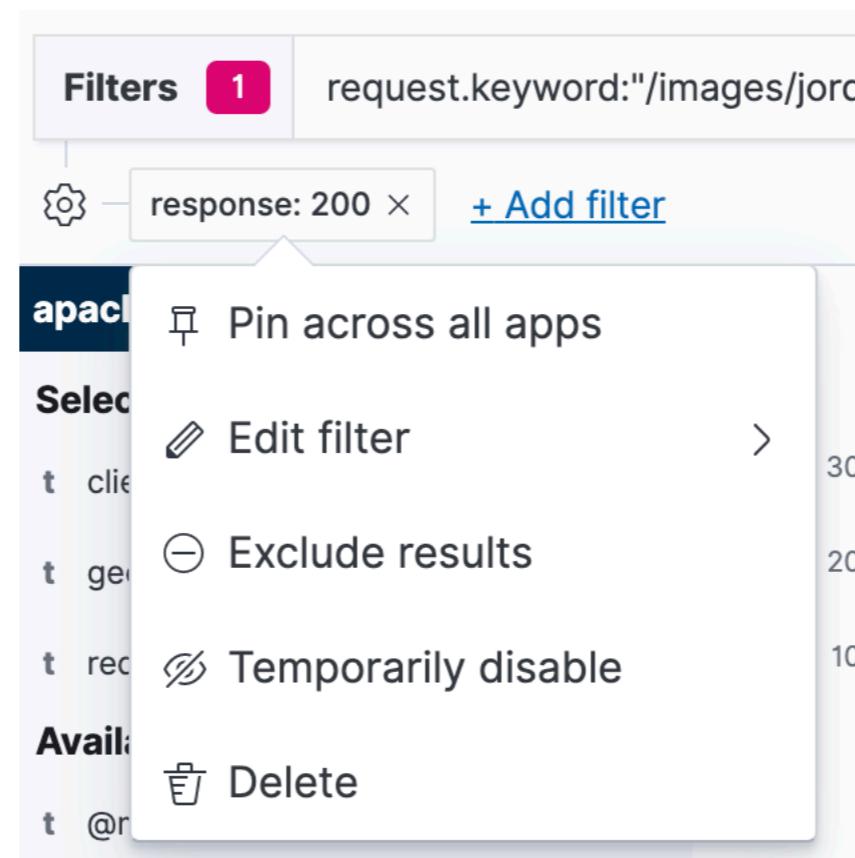
# Effect of Filters

- You can pin a filter to be used across different applications
- Example: when browsing back to a dashboard, a pinned filter will apply, and you will only see matching documents visualized



# Filter Actions (descriptions on next slide)

- You can manipulate each filter in several ways



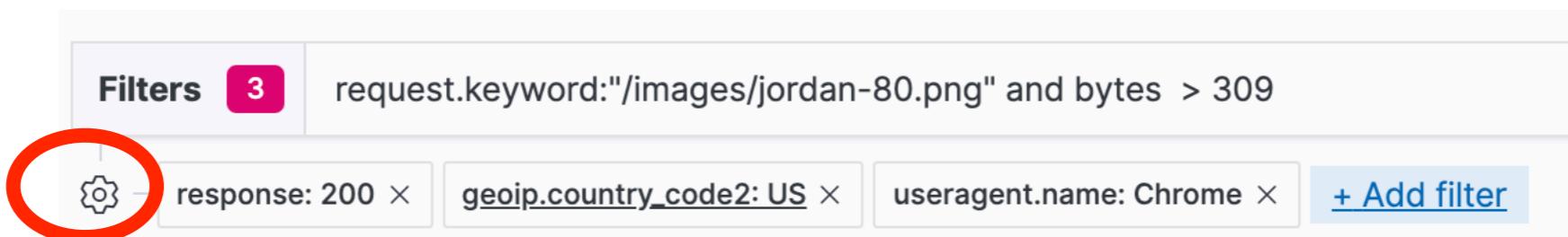
# Filter Actions

- Toggle: Turn the effect of the filter on / off. Useful to see the influence of a filter on a visualization when toggled rapidly.
- Pin: When a filter is pinned, it will follow you around the various areas of Kibana. Whenever you spot an "interesting" data point, you should consider creating and pinning a filter, for later inspection.
- Invert: Change the filter's effect from "matches this value" to "does not match this value"
- Delete: Remove a filter



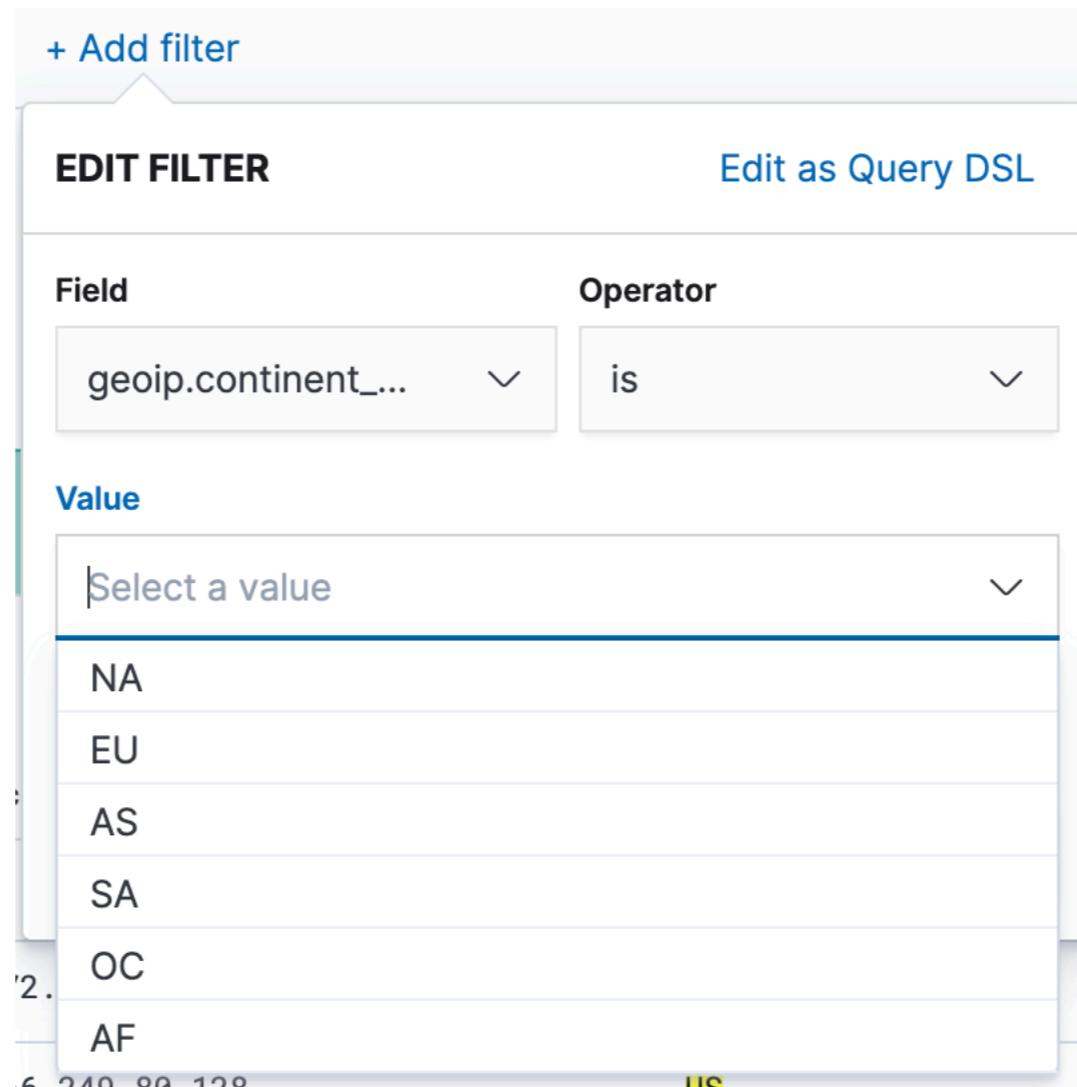
# Actions on multiple filters

- You can add as many filters as you like
- Pin them! They will follow you in Kibana
- The "Gear" icon on the left will allow you to perform a particular action on all of the filters simultaneously



# Adding a filter manually

- You can specify exactly the criteria which the filter matches





Kibana for Time Series Data

Lesson 1

# Review - Kibana Fundamentals



# Summary

- Kibana requires the use of Index Patterns for time-series datasets
- Sometimes the index pattern is created for you, but other times you must configure it yourself
- Kibana has a multitude of visualization types which give you a rich set of tools to build custom panels
- Dashboards are constructed by adding existing visualizations and re-arranging as necessary
- Filters are a useful tool in exploring your data
- Pinning filters is an easy strategy to help you visualize data.
- Filters can be toggled on and off to perform comparisons



# Quiz

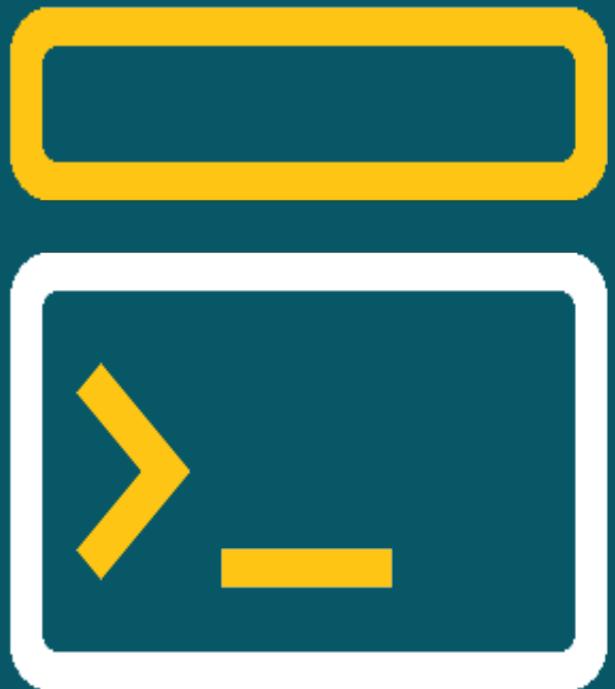
1. **True or False:** Kibana will automatically discover your data and pre-configure index patterns 100% for you.
2. **True or False:** Kibana Discover allows you to inspect individual documents
3. Which strategy allows you to make a filter "permanent", so it will follow you around as you browse Kibana?
4. **True or False:** Kibana Discover allows you to view only documents matching a particular query and/or filter
5. Which tool allows you to zoom in on the last 7 days?
6. What are the two query languages available in Kibana?
7. **True or False:** searches can be saved and attached to visualizations



Kibana for Time Series Data

Lesson 1

# Lab - Kibana Fundamentals





Kibana for Time Series Data

Lesson 2

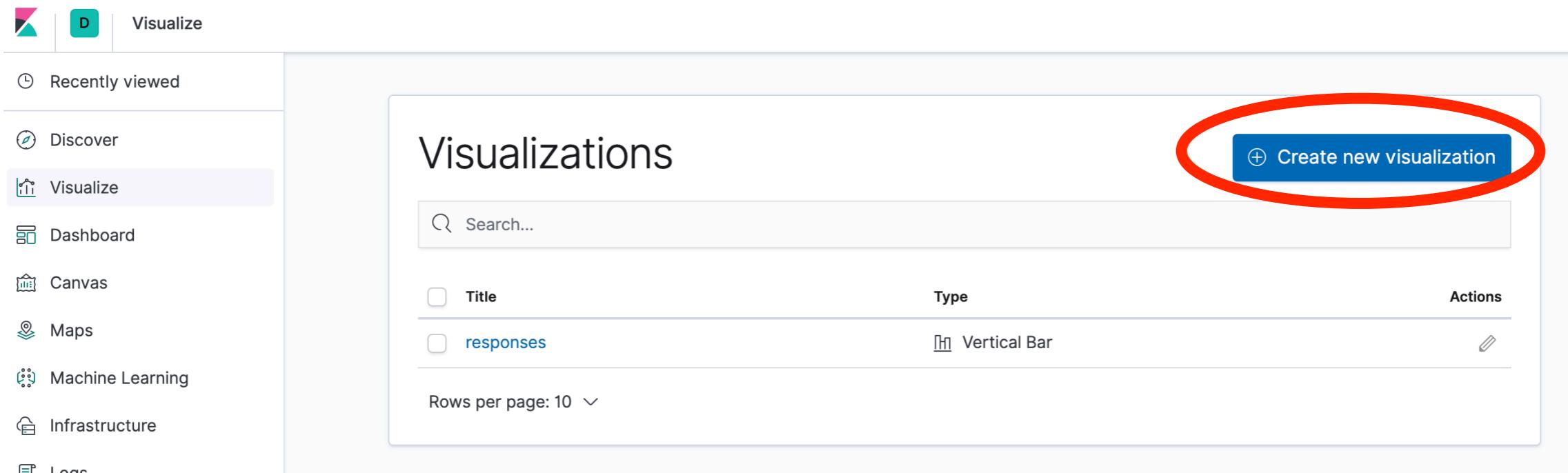
# Visualizations and Dashboards



# Visualizations

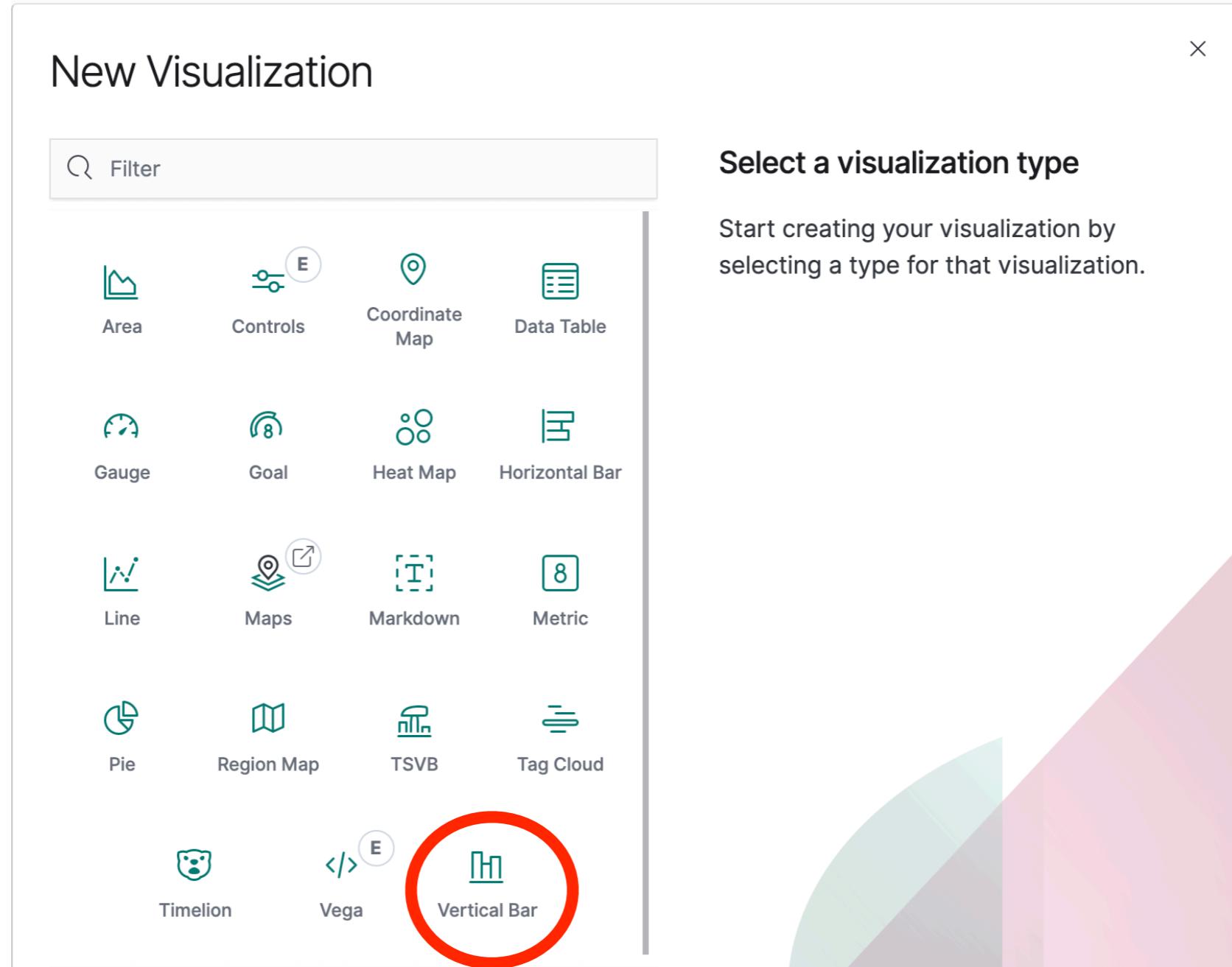
# Creating Visualizations

- After clicking on the “Visualize” tab in the left nav, you will see the interface for creating visualizations
- Click on the “Create new visualization” button to begin configuring the visualization



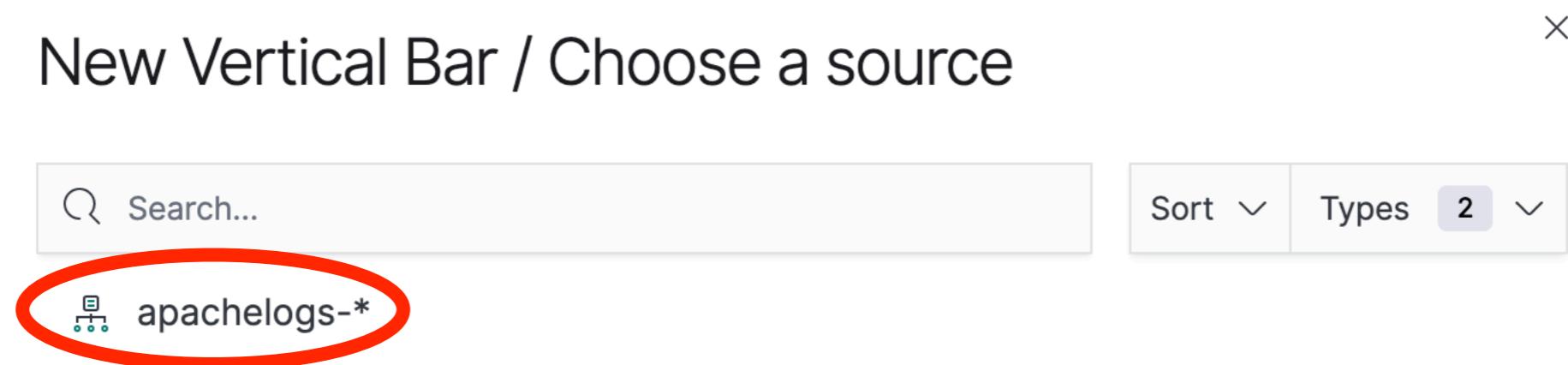
# Visualization types

- As you can see, there are many types to choose from



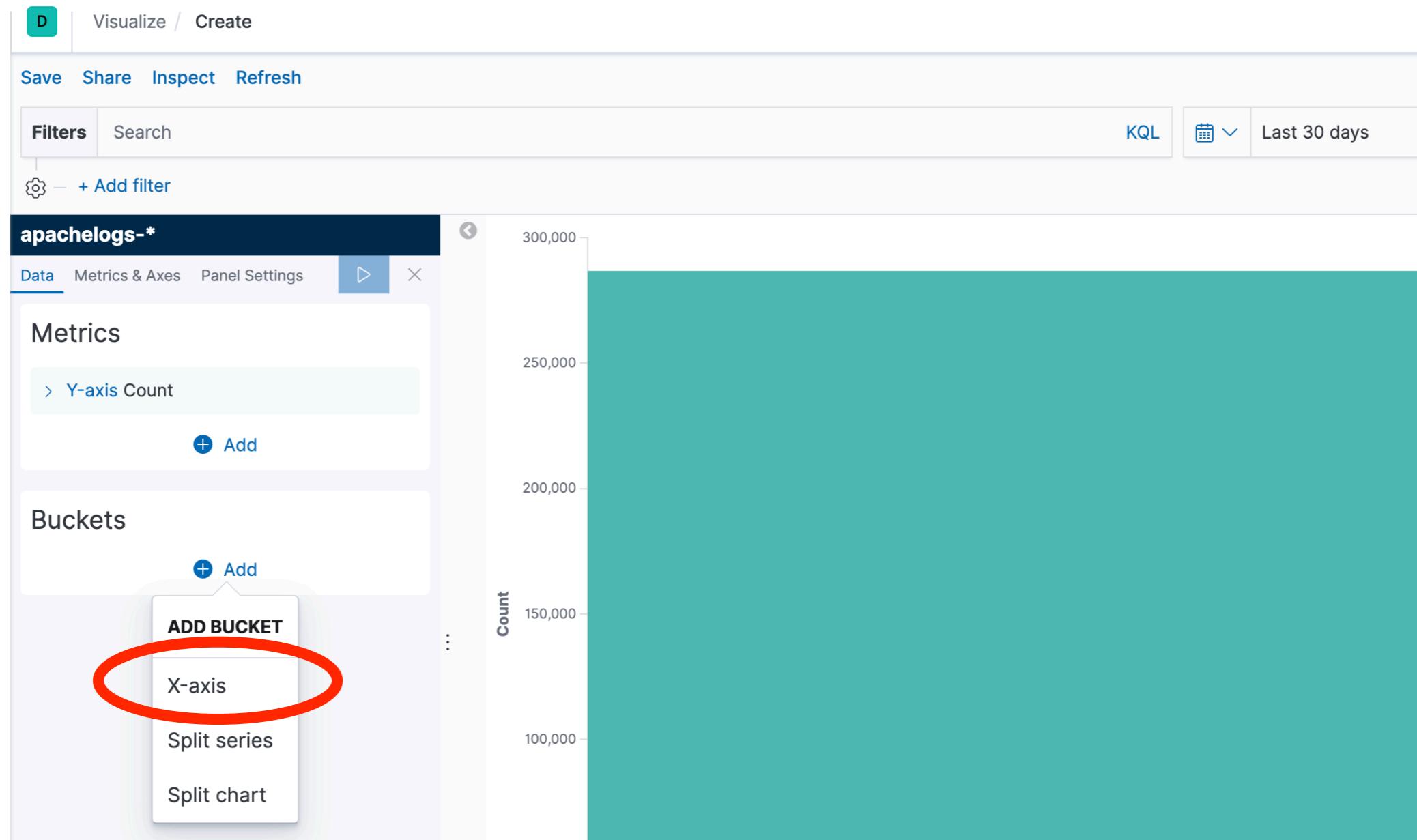
# Select the index pattern

- We only have one index pattern, so let's use it



# Configure the vertical bar chart

- We will use a *date histogram* to build this chart
- Click on "X-axis" to continue



# Configure buckets

- Use the Date Histogram and select the **@timestamp** field
- Then click "Add sub-bucket"

The screenshot shows the 'Buckets' configuration panel in Kibana. At the top, there's a header bar with tabs for 'Data', 'Metrics & Axes', and 'Panel Settings'. Below the header, the title 'apachelogs-\*' is displayed. The main area is titled 'Buckets' and contains settings for the X-axis. Under 'Aggregation', 'Date Histogram' is selected. Under 'Field', '@timestamp' is chosen. A red circle highlights the 'Date Histogram' dropdown. A modal window titled 'ADD SUB-BUCKET' is open, listing options: 'X-axis', 'Split series', and 'Split chart'. The 'Add' button at the bottom of the modal is also highlighted with a red circle.

# Configure buckets

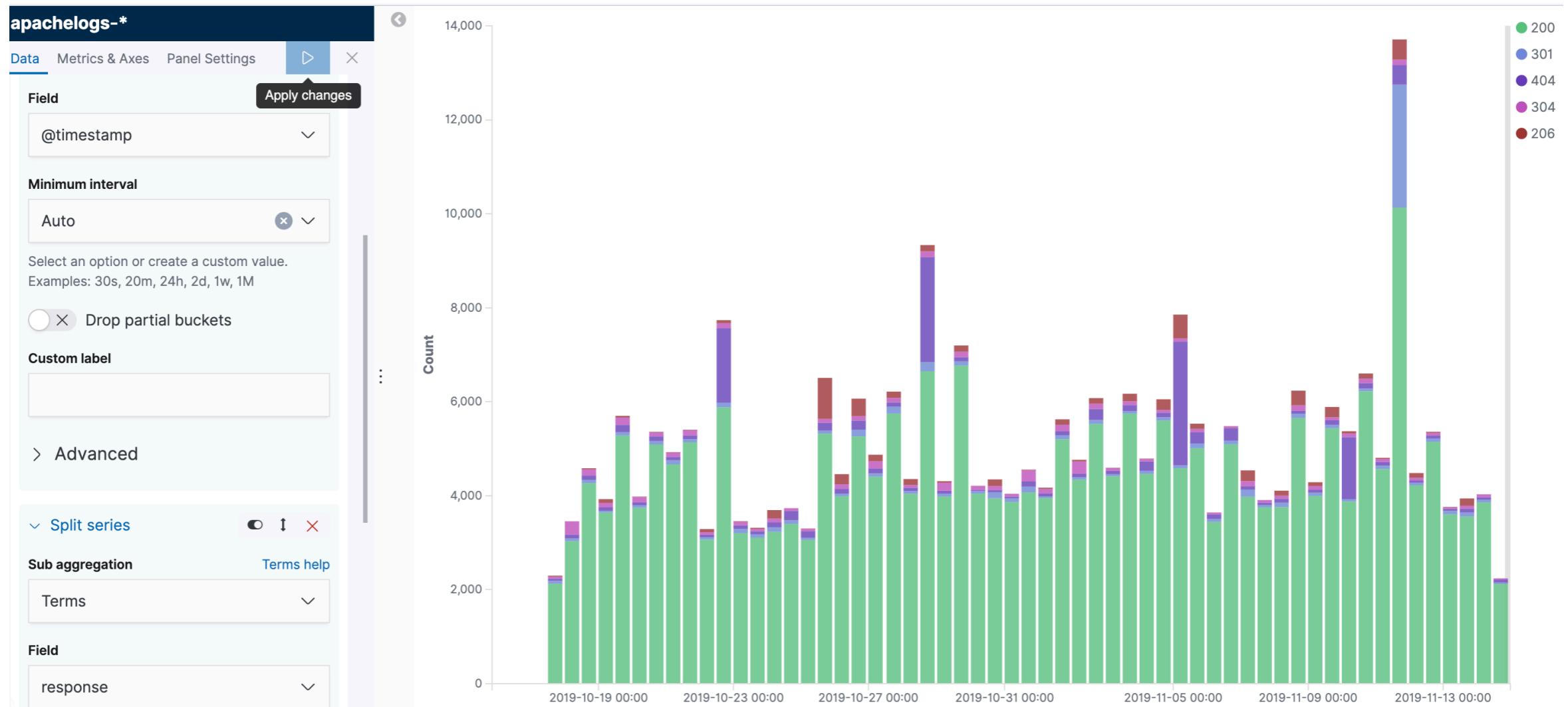
- Use "Split Series"
- Sub-Aggregation: *Terms*
- Field: **response**
- Click blue "play" arrow at top right

The screenshot shows the Kibana interface for the 'apachelogs-\*' panel. At the top, there are tabs for 'Data', 'Metrics & Axes', and 'Panel Settings'. A red circle highlights the blue 'play' arrow icon in the top right corner of the panel header. Below the header, there are several configuration sections:

- Split series:** Set to 'Sub aggregation Terms'. A red circle highlights this section.
- Field:** Set to 'response'. A red circle highlights this section.
- Order by:** Set to 'Metric: Count'.
- Order:** Set to 'Descending'.  
**Size:** Set to '5'. A red circle highlights this section.
- Group other values in separate bucket:** An unchecked checkbox.
- Show missing values:** A checked checkbox.
- Custom label:** An empty field.

# Multi-bucket visualization

- Now you can see the result: each color-coded bar shows the ratio of response codes per time bucket



# Saving visualizations

- Don't forget to save any visualizations you create; you can add them to dashboards later

The screenshot shows the Elasticsearch visualization interface. At the top, there are navigation links: Save, Share, Inspect, and Refresh. A red circle highlights the 'Save' link. Below the navigation, there are tabs for Filters, Search, KQL, and Last. A gear icon with '+ Add filter' is also present.

The main area displays a histogram titled 'apachelogs-\*'. The y-axis is labeled 'Count' and ranges from 0 to 14,000. The x-axis shows time intervals. The bars are green with some purple at the top, representing different response codes. On the left, there are settings for the histogram: 'Field' set to '@timestamp', 'Minimum interval' set to 'Auto', and an option to 'Drop partial buckets'. There is also a 'Custom label' input field and an 'Advanced' section with a 'Split series' button.

A modal dialog box titled 'Save visualization' is open in the center. It has a 'Title' input field containing 'Response codes over time'. At the bottom right of the dialog are 'Cancel' and 'Confirm Save' buttons. The background of the interface is blurred, showing the histogram data.

# Saving visualizations

- You can be able to view saved visualizations in the next screen. As you build more, they will appear. Later, we will see how to add them to dashboards.

## Visualizations

[Create new visualization](#)

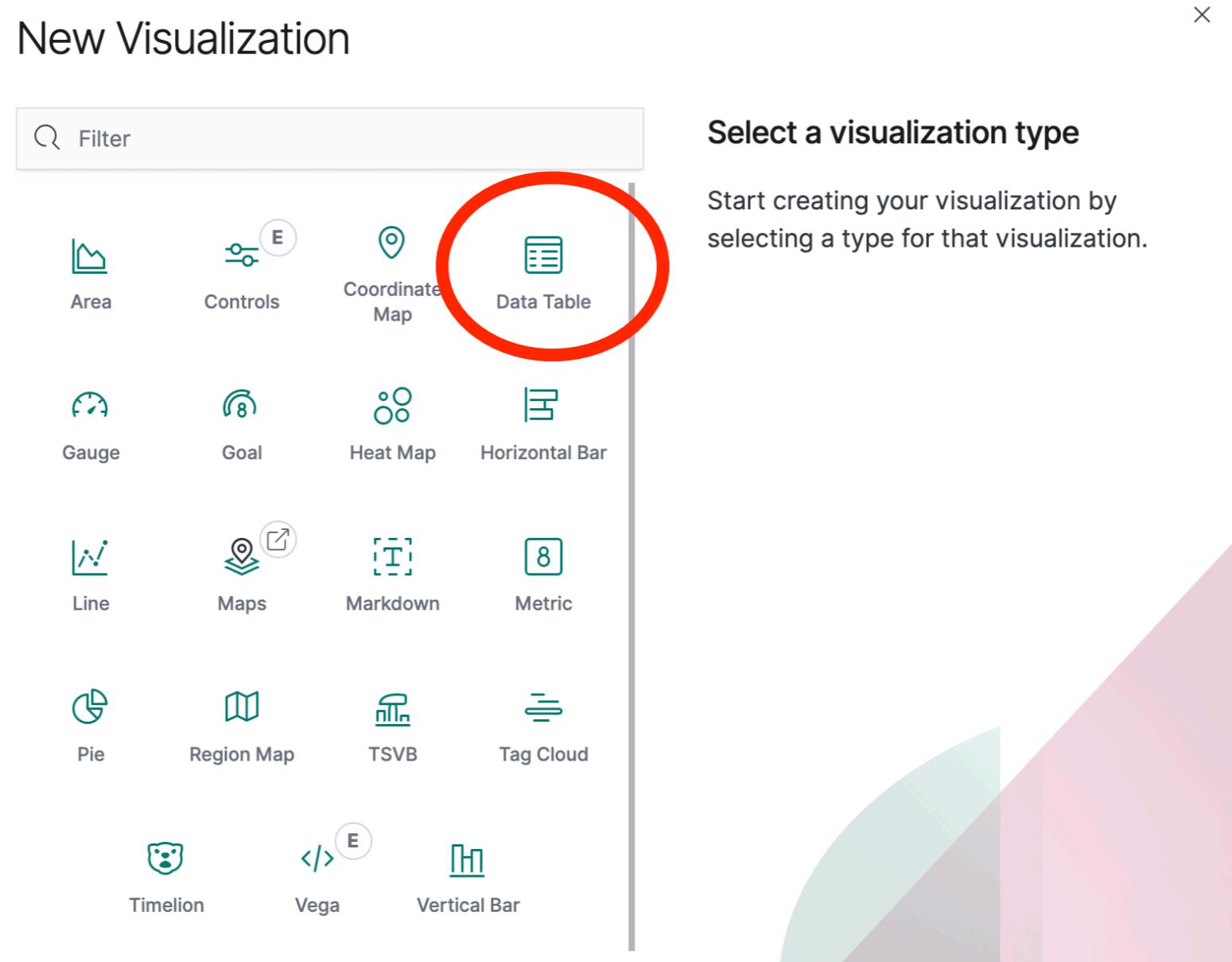
Search...

| Title                                    | Type         | Actions |
|--|--------------|---------|
| <a href="#">Response codes over time</a> | Vertical Bar |         |
| <a href="#">responses</a>                | Vertical Bar |         |

Rows per page: 10 ▾

# Data Tables

- Browse to the Visualization page again
- Click on the data table icon to build a "Top" terms viz



# Configure data table

- Use the Count metric
- Choose to Split Rows
- Terms Aggregation
- Field: request.keyword
- Size: 10
- Click the "play" button

The screenshot shows the configuration interface for a visualization titled "apachelogs-\*". The top navigation bar has tabs for "Data" (which is selected) and "Options". On the far right are a blue "Play" button and a red "X" button. The main area is divided into sections: "Metrics" (with a "Metric Count" link and an "Add" button), "Buckets" (with a "Split rows" section containing "Aggregation: Terms" and "Field: request.keyword", both with dropdown menus), "Order by" (set to "Metric: Count"), and "Order" and "Size" settings (set to "Descending" and "10").

# Top Requests

- You can now see the top request URLs
- Don't forget to save it!

| request.keyword: Descending                   | Count  |
|---|--------|
| /favicon.ico                                  | 17,942 |
| /files/logstash/logstash-1.1.0-monolithic.jar | 13,290 |
| /style2.css                                   | 12,338 |
| /reset.css                                    | 12,237 |
| /images/jordan-80.png                         | 11,941 |
| /images/web/2009/banner.png                   | 11,676 |
| /blog/tags/puppet?flav=rss20                  | 10,829 |
| /   | 5,912  |
| /presentations/fpm-scale12x.pdf               | 4,933  |
| ?flav=rss20                                   | 4,856  |

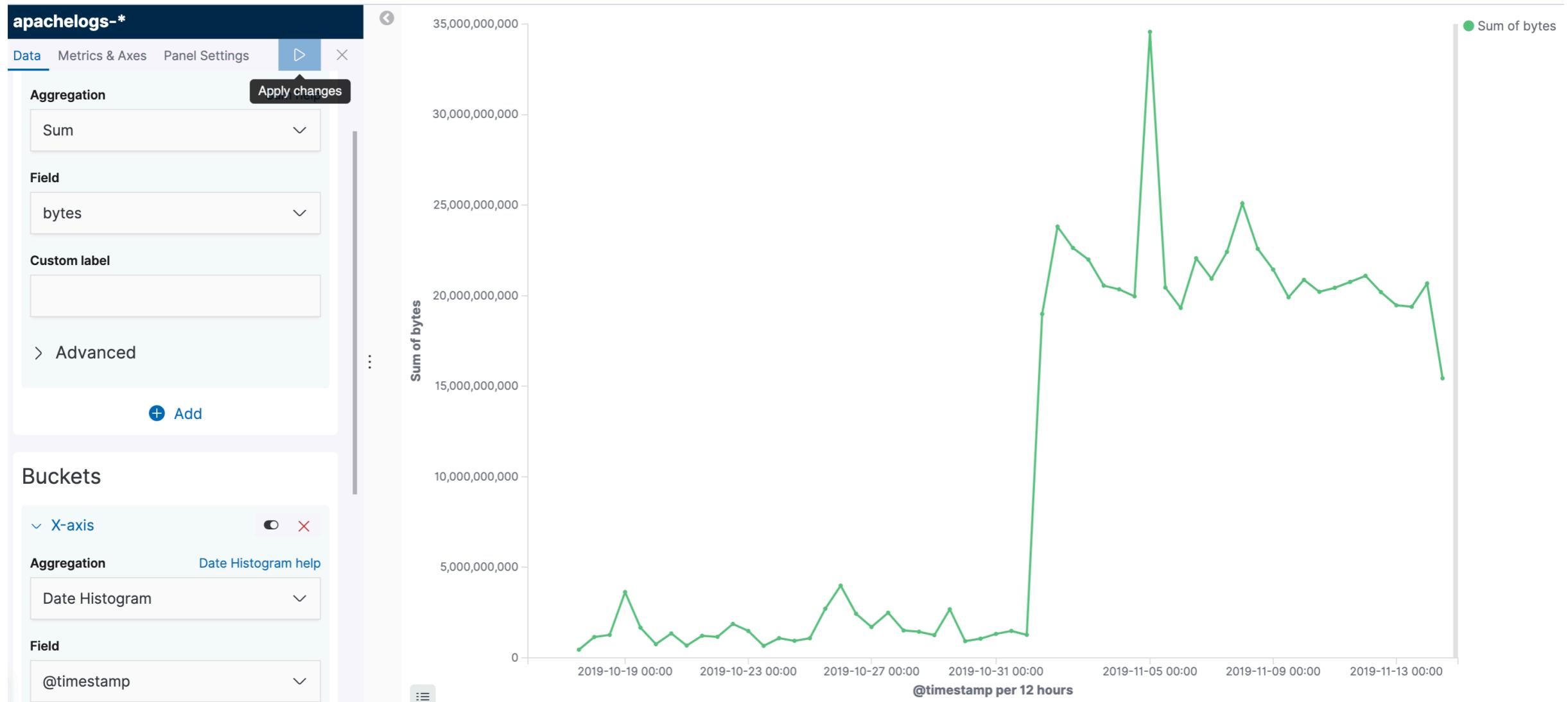
Export: [Raw](#) [Formatted](#)

# Line Graph

- Let's see how the line graph visualization works
- Create a new visualization
- Select "Line Graph"
- This time, Use "Sum" Metric on the "Bytes" field
- Split the buckets on the X-Axis
- Use the Date Histogram
- Select the "@timestamp" field
- Click the Play button to render the visualization

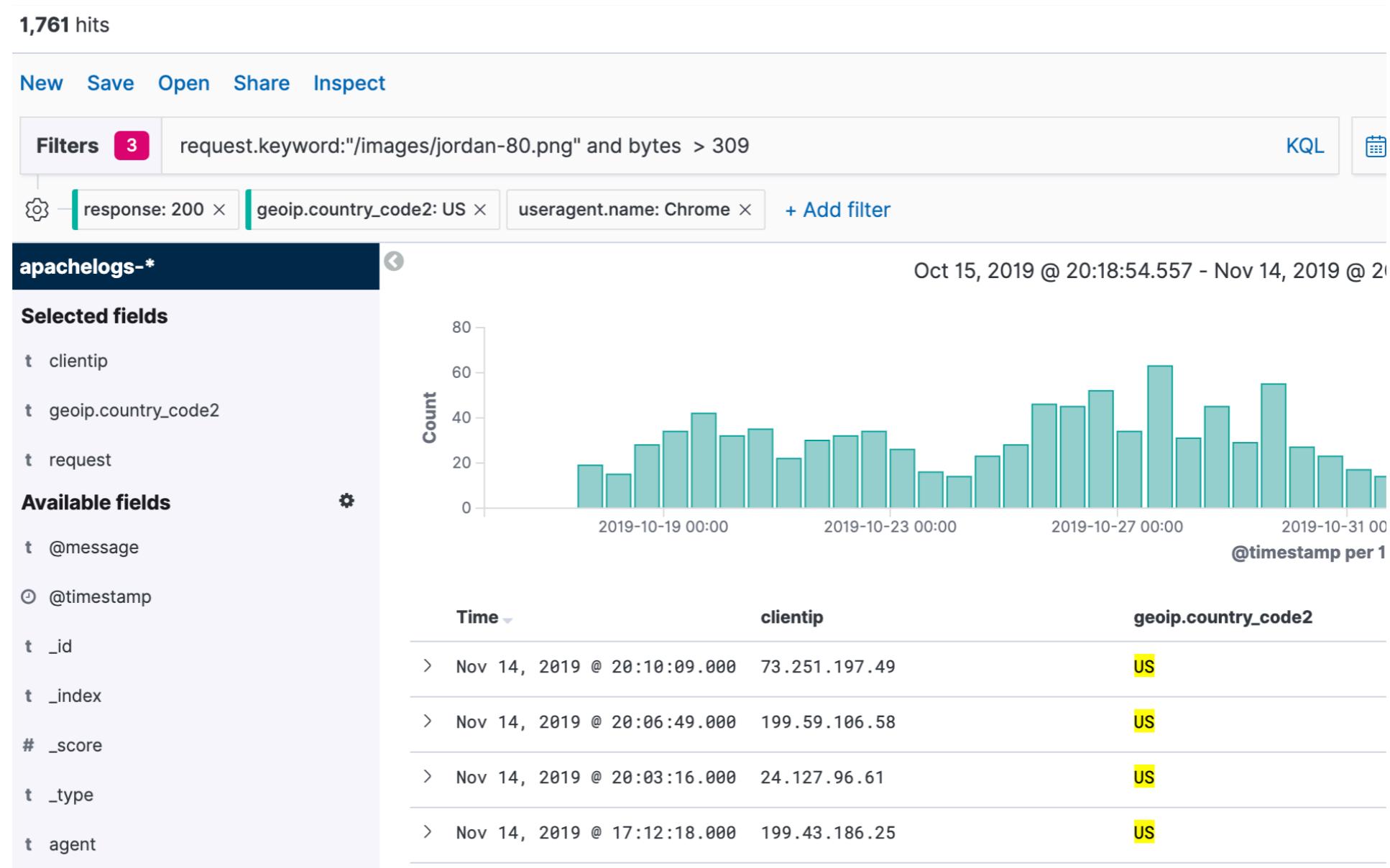
# Completed Line Graph

- Should look something like this. Don't forget to save it!



# Putting it all together

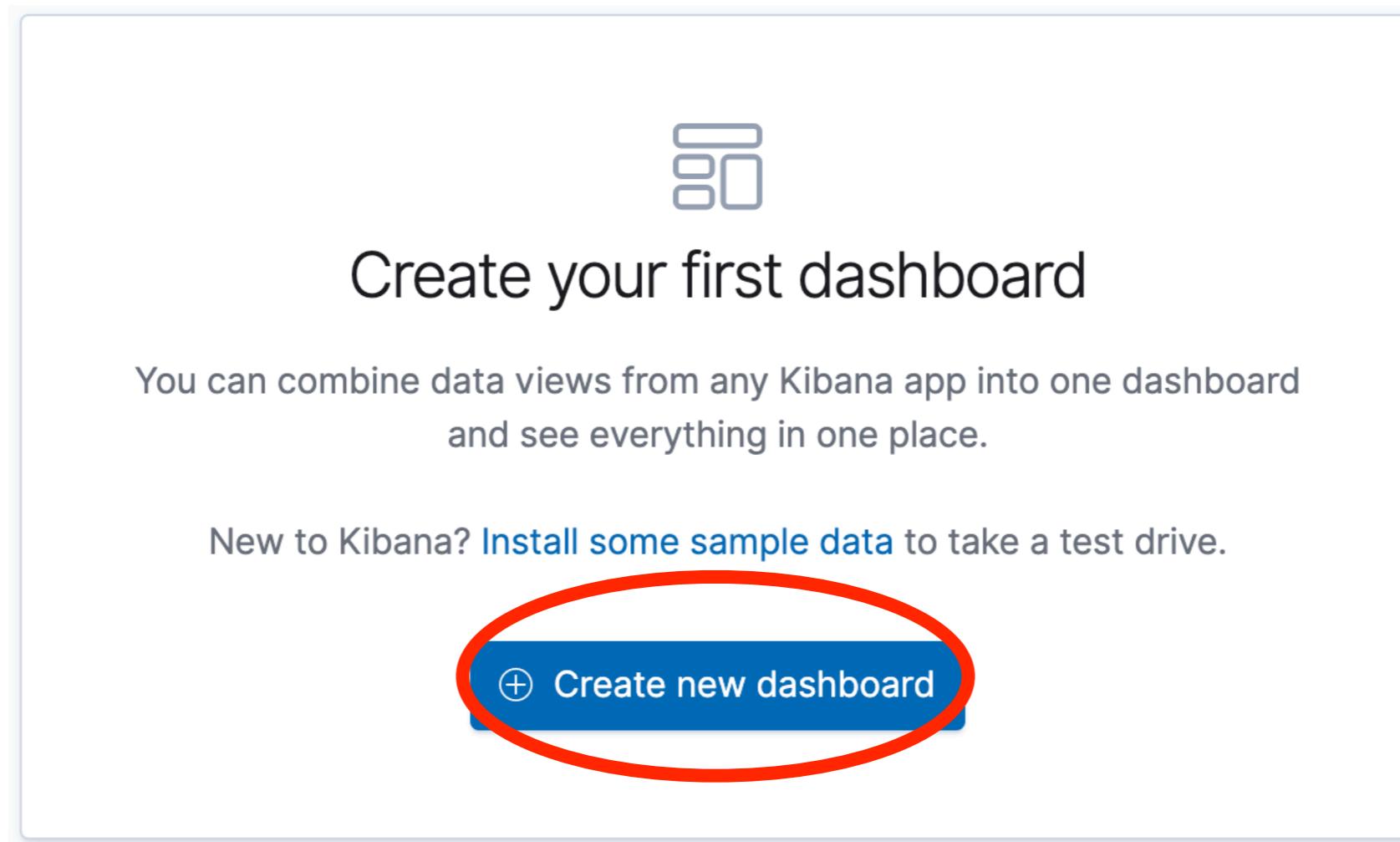
- Create a visualization based on a search and/or filter
- Combine with pinned filters for maximum power!



# Dashboards

# Creating Dashboards

- Select the dashboard icon in the left nav:
- Click on the "Create new dashboard" button



# Add existing panels to a dashboard

- Simply click on the visualization name and it will be added!

Add panels ×

---

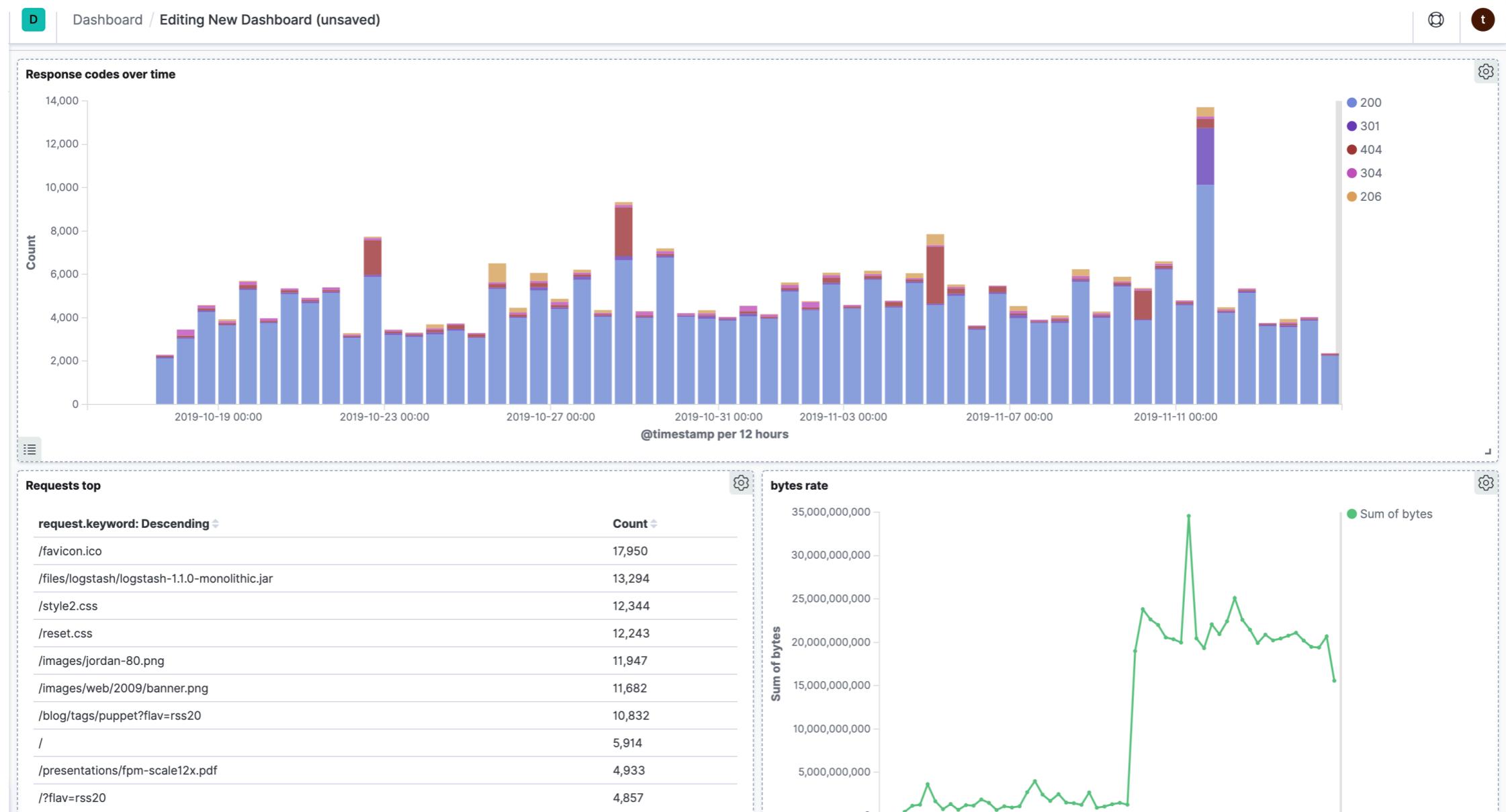
Sort ▾ Types 3 ▾

- bytes rate
- Requests top
- Response codes over time
- responses

[Create new visualization](#)

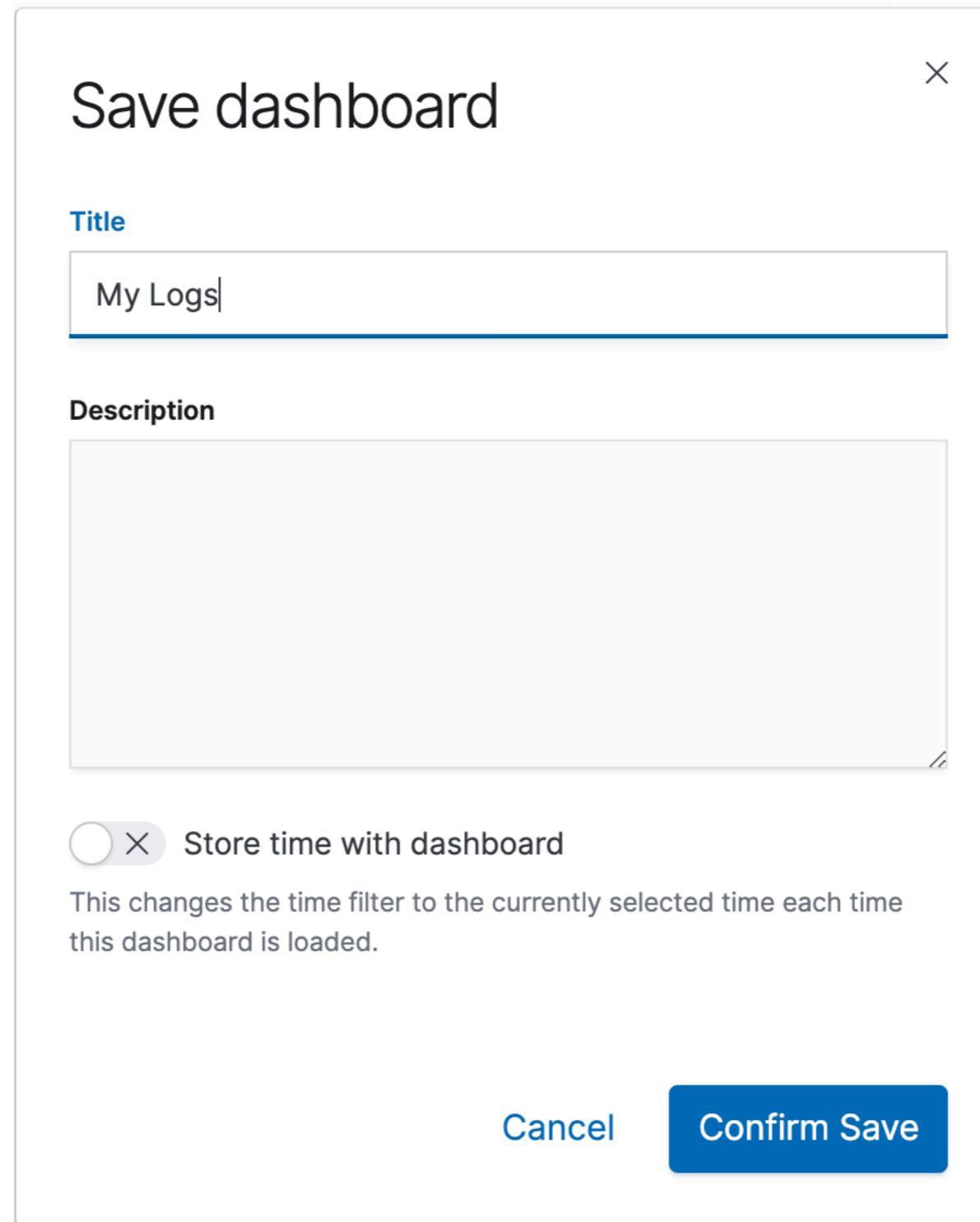
# Arrange visualizations in the Dashboard

- Clicking on the title bar allows you to drag a visualization
- Clicking in the bottom right corner allows you to resize it



# Save Dashboard

- As with all your Kibana creations, you can save dashboards



# Field Formatters

# Field Formatters

- You might have noticed that the **bytes** values in the previous visualization looked awkward. Very large byte values are difficult for humans to read! Let's fix that.
- Browse to the Index Pattern, as shown below, and search or scroll to the **bytes** field in the mapping. Click Edit icon:

★ apachelogs-\* ★ ⌂ ⚡

Time Filter field name: @timestamp Default

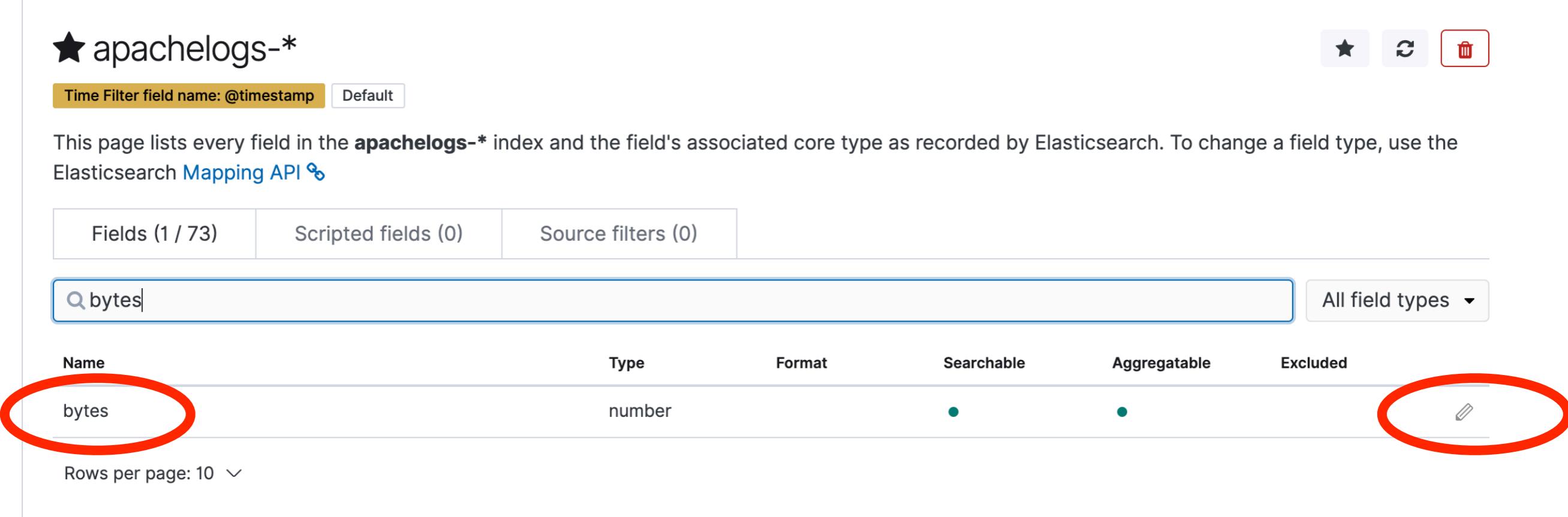
This page lists every field in the **apachelogs-\*** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch [Mapping API](#) ↗

Fields (1 / 73) Scripted fields (0) Source filters (0)

All field types ▾

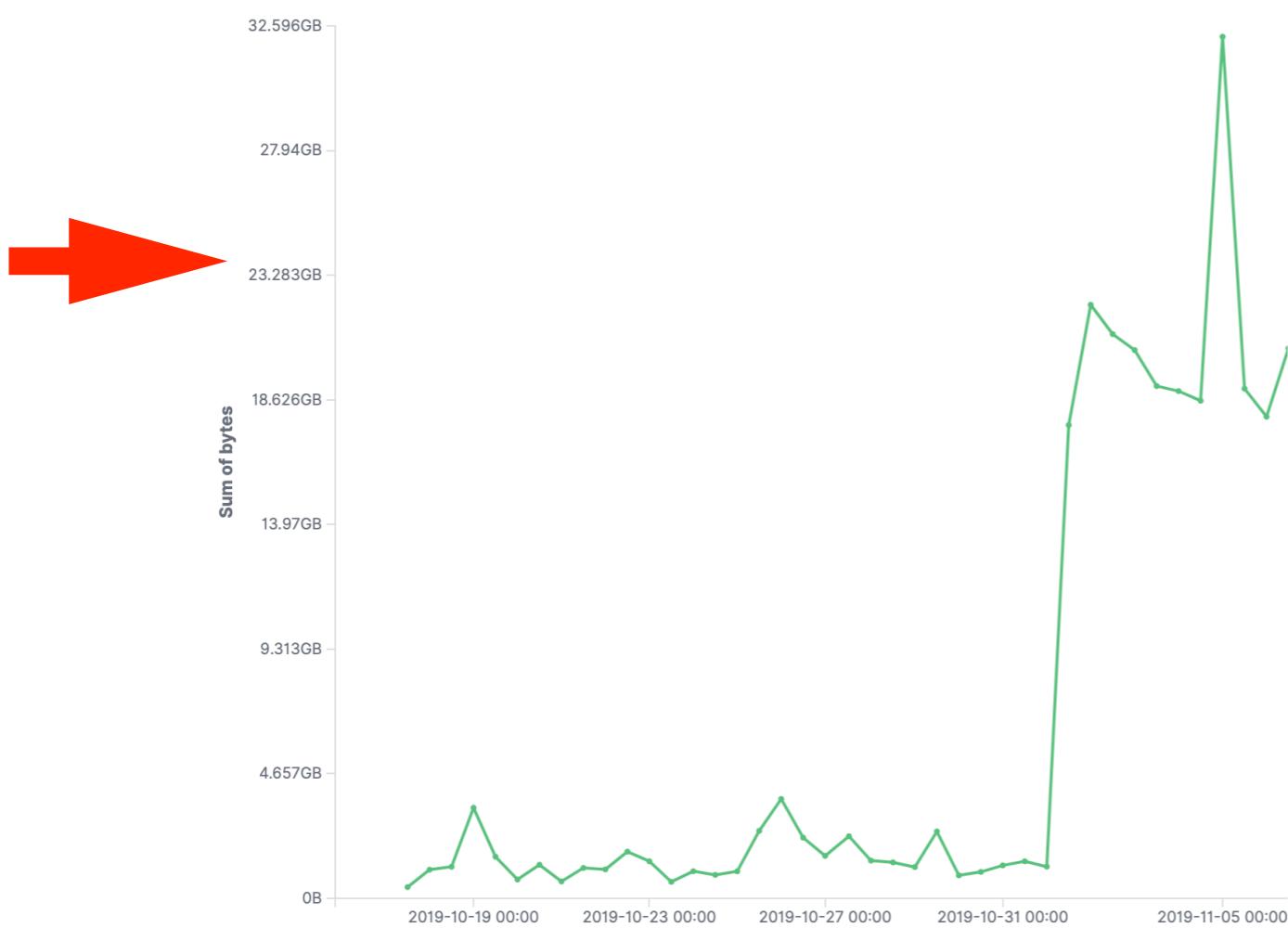
| Name  | Type   | Format | Searchable | Aggregatable | Excluded  |
|-------|--------|--------|------------|--------------|---|
| bytes | number |        | •          | •            |  |

Rows per page: 10 ▾



# Field Formatters

- Select "Bytes" format
- Click "Save field"
- Browse back to visualization



Edit bytes

Type: number

Format (Default: Number )

Bytes (circled in red)

Formatting allows you to control the way that specific values are displayed. It can also cause values to be completely changed and prevent highlighting in Discover from working.

Numerical.js format pattern (Default: 0,0.[000]b )

0,0.[000]b

Documentation ↗

Samples

| Input      | Output  |
|------------|---------|
| 256        | 256B    |
| 1024       | 1KB     |
| 5150000    | 4.911MB |
| 1990000000 | 1.853GB |

Popularity: 0

Save field (circled in red)

Cancel



Kibana for Time Series Data

Lesson 2

# Review - Visualizations and Dashboards



# Summary

- Discover is Kibana's built-in search tool
- Use the Time Picker to select time ranges for viewing
- Customize Columnar views by adding fields
- View surrounding documents
- Lucene query syntax is the default text search
- Enabling Kibana query language allows auto-complete of field names, operators, and field values
- Searches can be saved, and visualizations can be created with searches attached

# Quiz

- 1. True or False:** You can create individual panels in the "Visualization" tab.
- 2. True or False:** Dashboards are assembled by choosing pre-configured visualizations and simply adding them to the dashboard one-by-one.
- 3. Which aggregation allows you to see "Top 10" things?**
- 4. Which aggregation allows you to bucket by time period?**
- 5. Which aggregation allows you calculate totals for a field?**



Kibana for Time Series Data

Lesson 2

# Lab - Visualizations and Dashboards





Kibana for Time Series Data

Lesson 3

# Time-Series Visualization Builder (TSVB)



# Time-Series Visualization Builder

# Time Series Visualization Builder

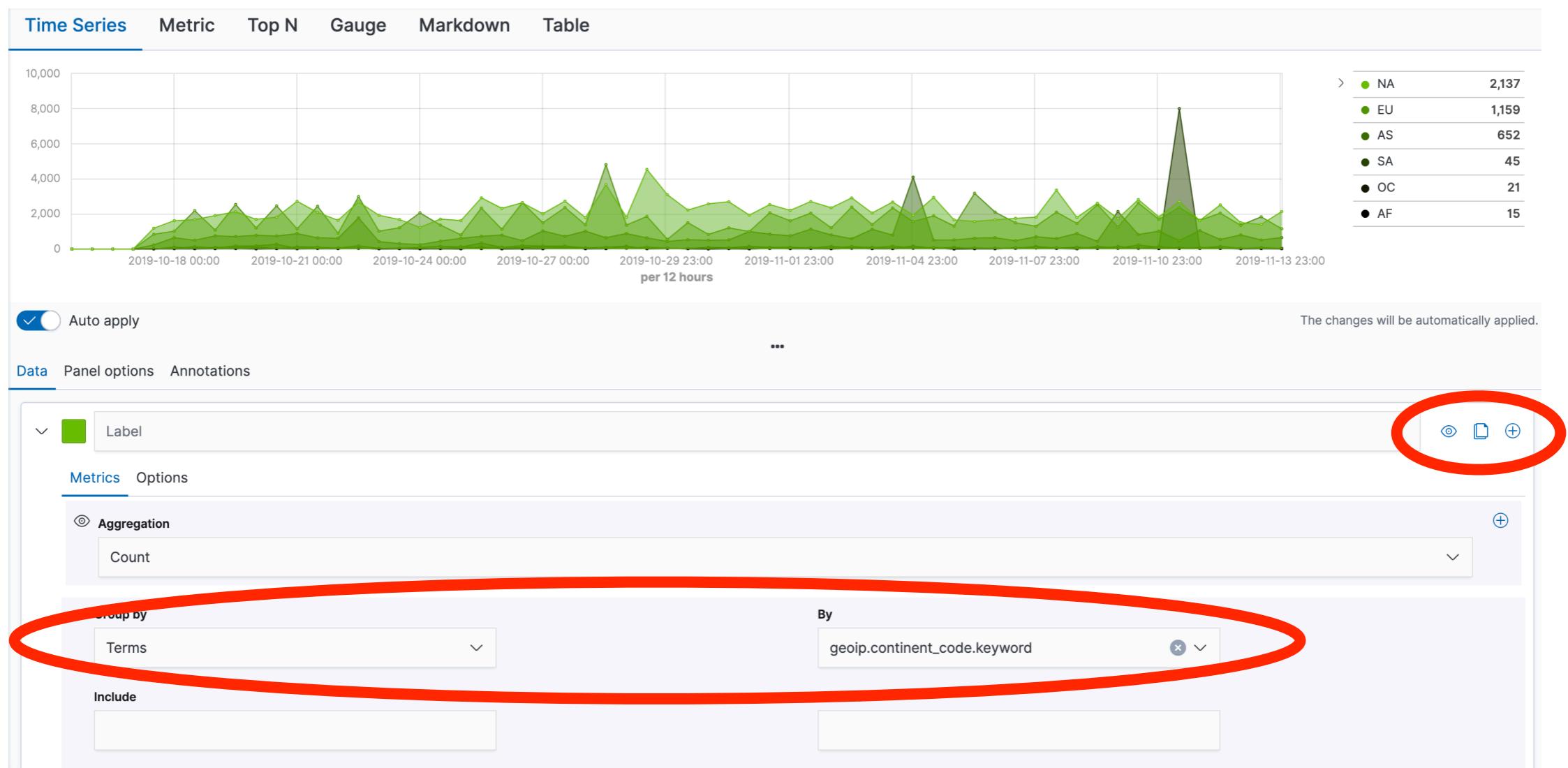
- TSVB was created especially for time series data; it's great for viewing log data
- Makes it possible to overlay multiple time series in one visualization!
- May be embedded into a Kibana dashboard just like standard visualizations
- Rich colors and customization
- Features visual annotations
- To begin, click this icon on the Create Visualization page:



TSVB

# Creating a Time Series chart

- Start by selecting "Terms" to Group By. Use the geoip.continent\_code.keyword field
- Then, click the plus sign on the right to add another chart



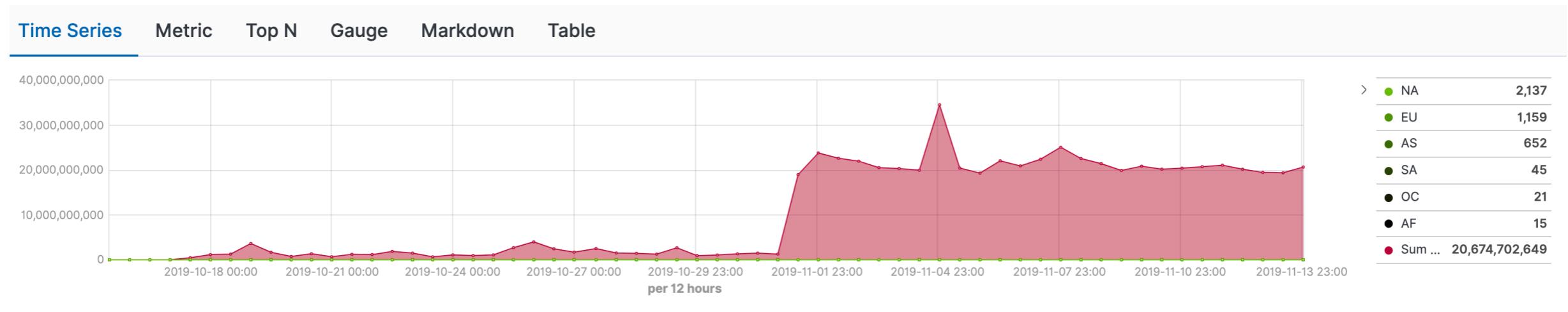
# Adding the second time series

- Pick a color which contrasts or works well with the first time series you just created
- Use the Sum metric aggregation instead of Count
- Choose the *bytes* field
- Group by Everything
- Give it a meaningful label

The screenshot shows the Metrics configuration interface. At the top, there is a color swatch (red) and a label input field. Below that, tabs for 'Metrics' (selected) and 'Options' are visible. Under the 'Metrics' tab, there are sections for 'Aggregation' and 'Field'. The 'Aggregation' section has a radio button selected for 'Sum'. The 'Field' section contains the text 'bytes |'. In the bottom left, there is a 'Group by' section with a dropdown menu set to 'Everything'. On the right side of the interface, there are several small icons for managing the configuration.

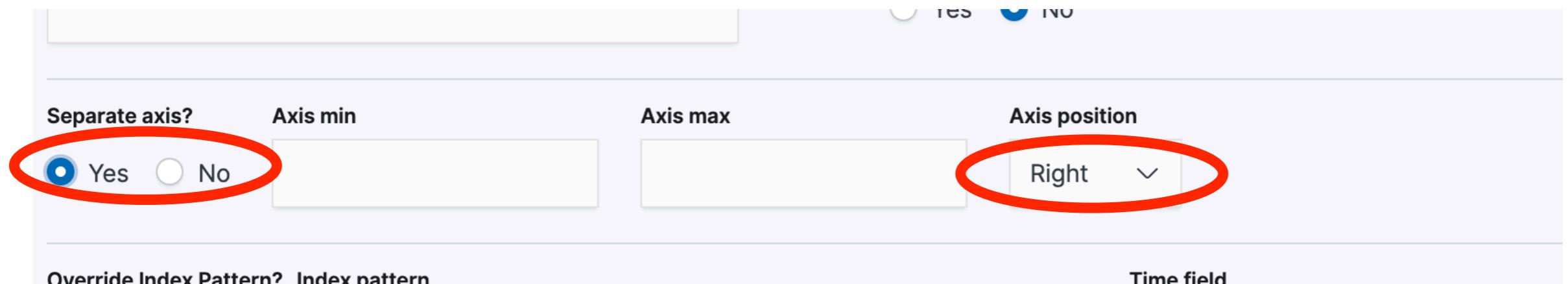
# Incompatible y-axis?

- In this case, the values for the new (Sum of bytes) time series are on a completely different scale, so we can no longer view the original data



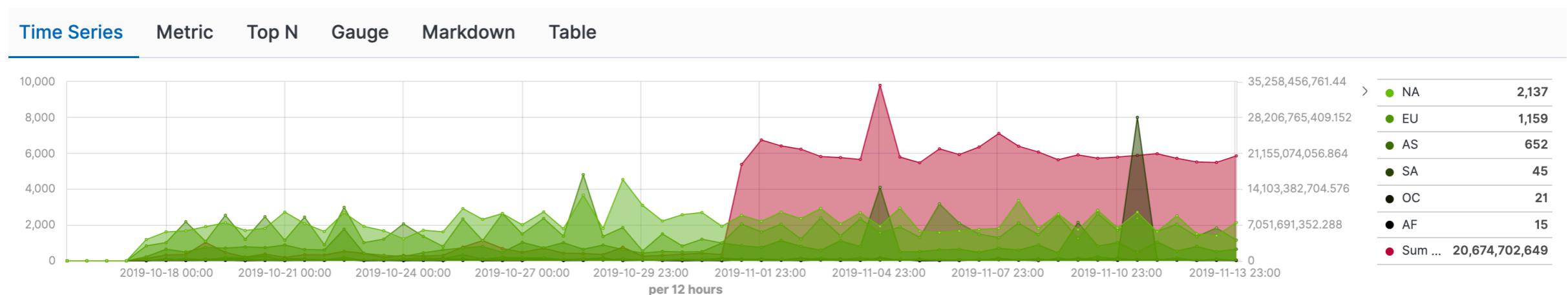
# Separate Axis to the rescue!

- On the second (or any subsequent) time series, simply select the "Separate Axis" option, and a location (here we will display it to the right)



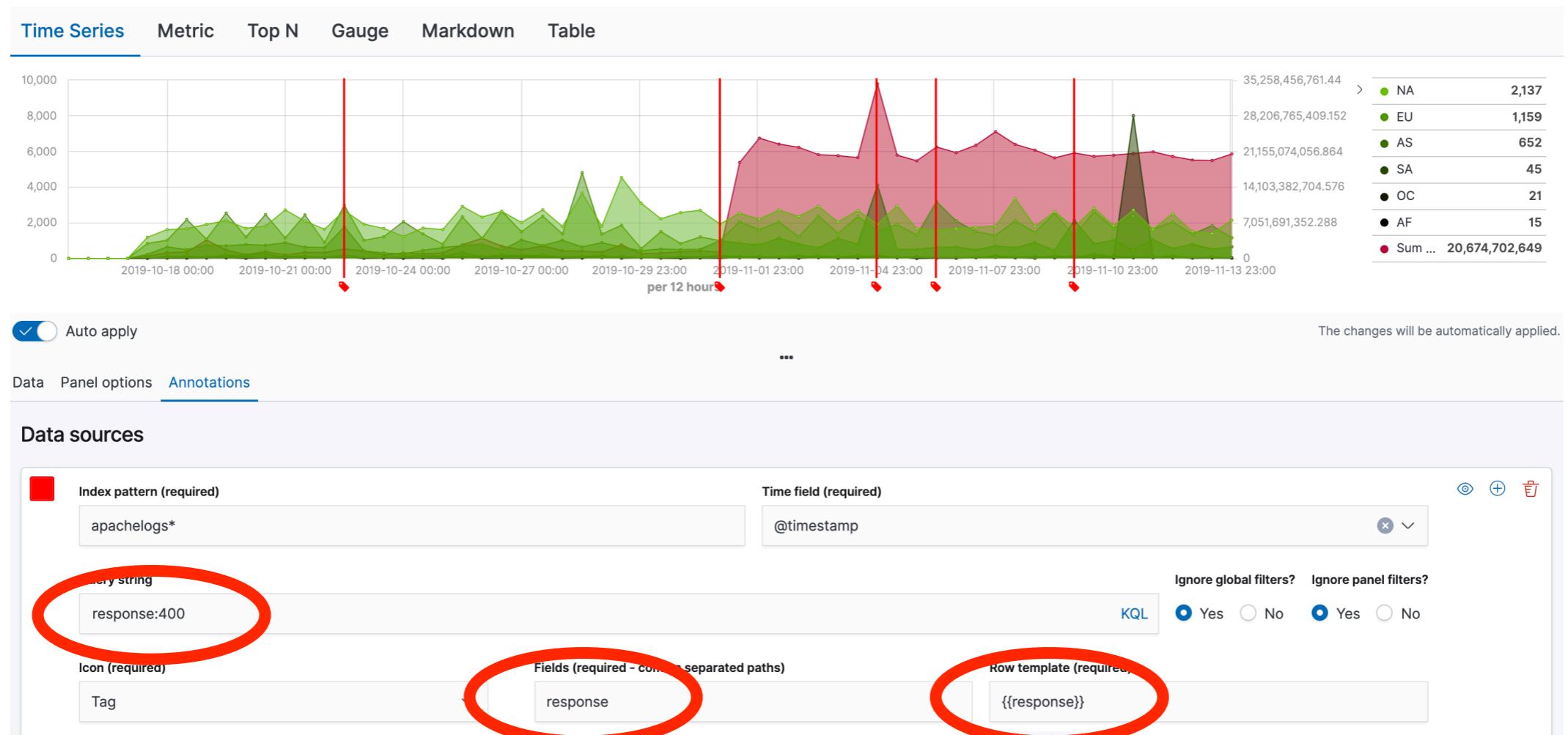
# Separate Axis results

- Now we can see the first (green) time series, overlaid with the second (red) time series



# Annotations in TSVB

- Annotations can be created by specifying a Query String which represents the events in Elasticsearch
- We will use "response:400"; don't forget to fill out the Fields and Row Template as shown below. Save your work!



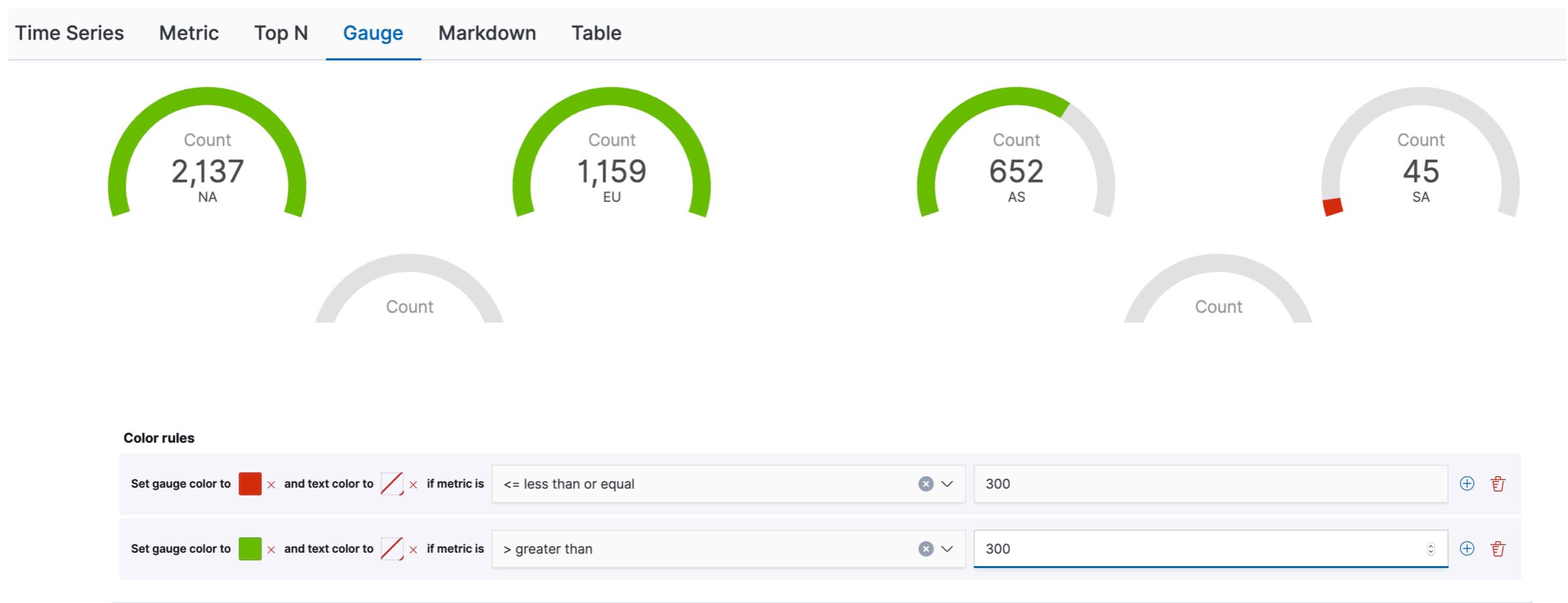
# Metrics

- Simple to create
- Reflects the value in the last time bucket. Therefore, if you do not have recent data, the value could be 0!

Count  
**356**

# Gauges

- Gauges measure against a defined goal
- Set the "Gauge Max" value if desired
- Set gauge colors based on data values and conditionals



# Tables

- Similar to Data Tables in Kibana
- Just pick the Group By field and number of Rows
- You may also specify different Metric aggs to display

The screenshot shows the Kibana interface for creating a Table visualization. At the top, there are tabs for Time Series, Metric, Top N, Gauge, Markdown, and Table, with Table selected. Below the tabs is a table with the following data:

|          | Count |
|----------|-------|
| Other    | 2,167 |
| Windows  | 918   |
| Mac OS X | 308   |
| Linux    | 278   |
| Ubuntu   | 136   |

Below the table is a section with a checked "Auto apply" toggle and a "Columns" button. A note says: "For the table visualization you need to define a field to group by using a terms aggregation." A dropdown menu labeled "Category field" has "useragent.os.keyword" selected. A red circle highlights this dropdown.



Kibana for Time Series Data

Lesson 3

# Review - Time-Series Visualization Builder



# Summary

- Time-Series Visualization Builder (TSVB) was designed to work best with time series data
- Multiple time series can be overlaid in one visualization
- Options include Time Series, Metric, Gauge, Table

# Quiz

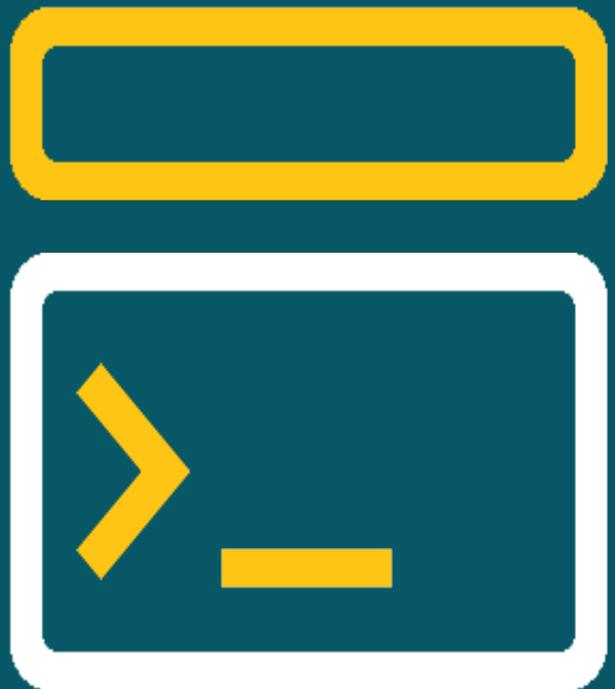
- 1. True or False:** Time Series Visual Builder is excellent for displaying log data
- 2. True or False:** You should create one panel per time series, and display them one above another on the dashboard page
- 3. True or False:** TSVB allows you to choose multiple color palettes in one visualization
4. What should you do if you overlay two visualizations with y-axis values on very different scales?



Kibana for Time Series Data

Lesson 3

# Lab - Time-Series Visualization Builder



# Quiz Answers



# Kibana Fundamentals

1. False (usually). You must configure index patterns. (ML does have some advanced log detection algos)
2. True
3. Pinning filters
4. True
5. Time Picker
6. Lucene Query Syntax and Kibana Query Language
7. True

# Visualizations and Dashboards

1. True
2. True
3. Terms bucket aggregation
4. Date Histogram bucket aggregation
5. Sum metric aggregation

# Time-Series Visualization Builder (TSVB)

1. True
2. False, you can overlay them
3. True
4. Use "Separate Axis"