

## 

## **Assumptions**

1. You are familiar with Elasticsearch, Kibana, and Logstash
2. You can access the Elastic environment allocated to you.

## **Goal**

* Familiarize with Machine Learning(ml) capabilities in Elastic Stack
* Creating ml jobs to detect anomalies in the **farequote** dataset

**Environment**

You will use the Elastic environment provided to you by facilitators at the beginning of the day.

**Conventions**

An italics colored text in the instructions refers to a button, icon, link that you will click on in Kibana.

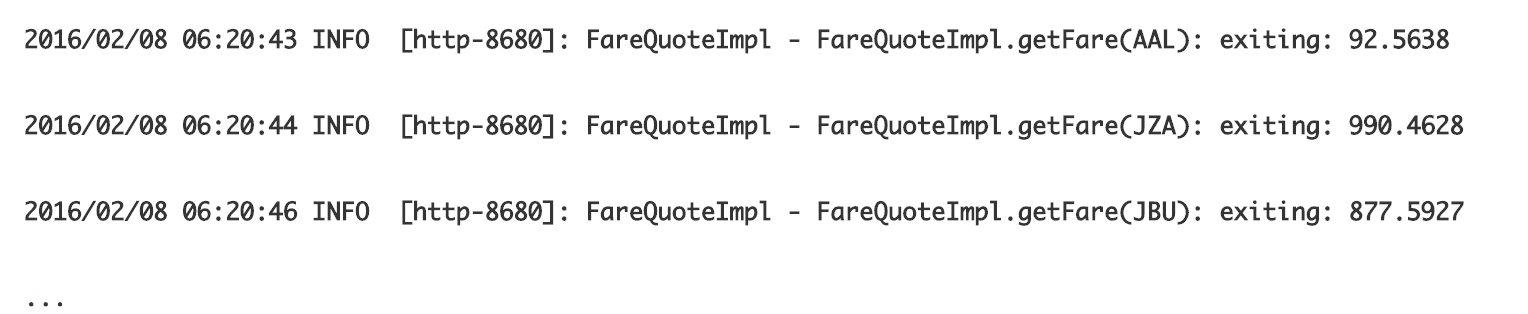
A colored text in the instructions refer to a section/place within Kibana

## **#1**

Familiarize with the dataset

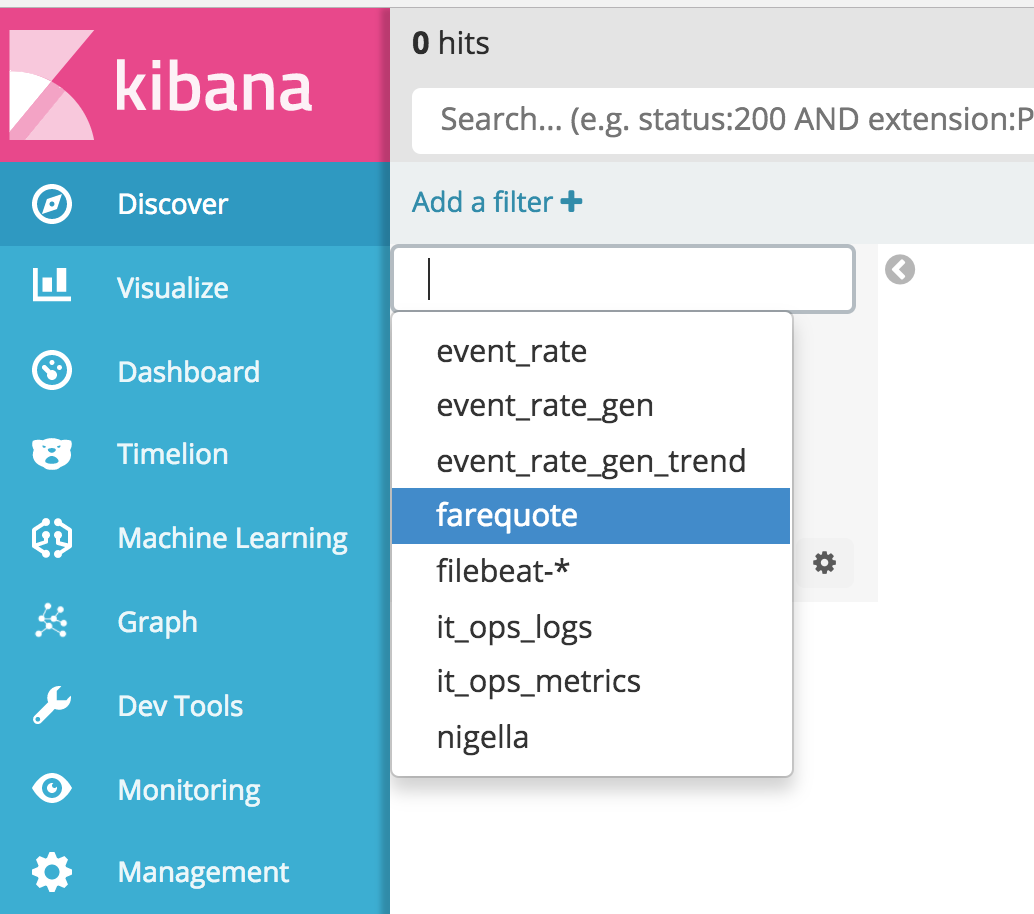
## 

## **farequote**

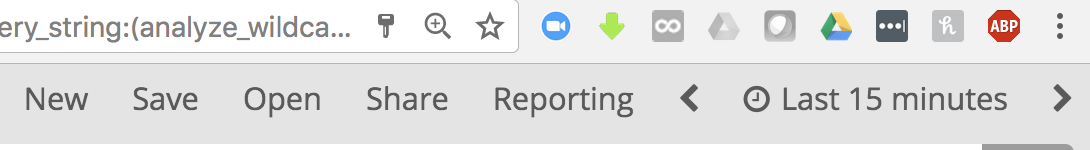


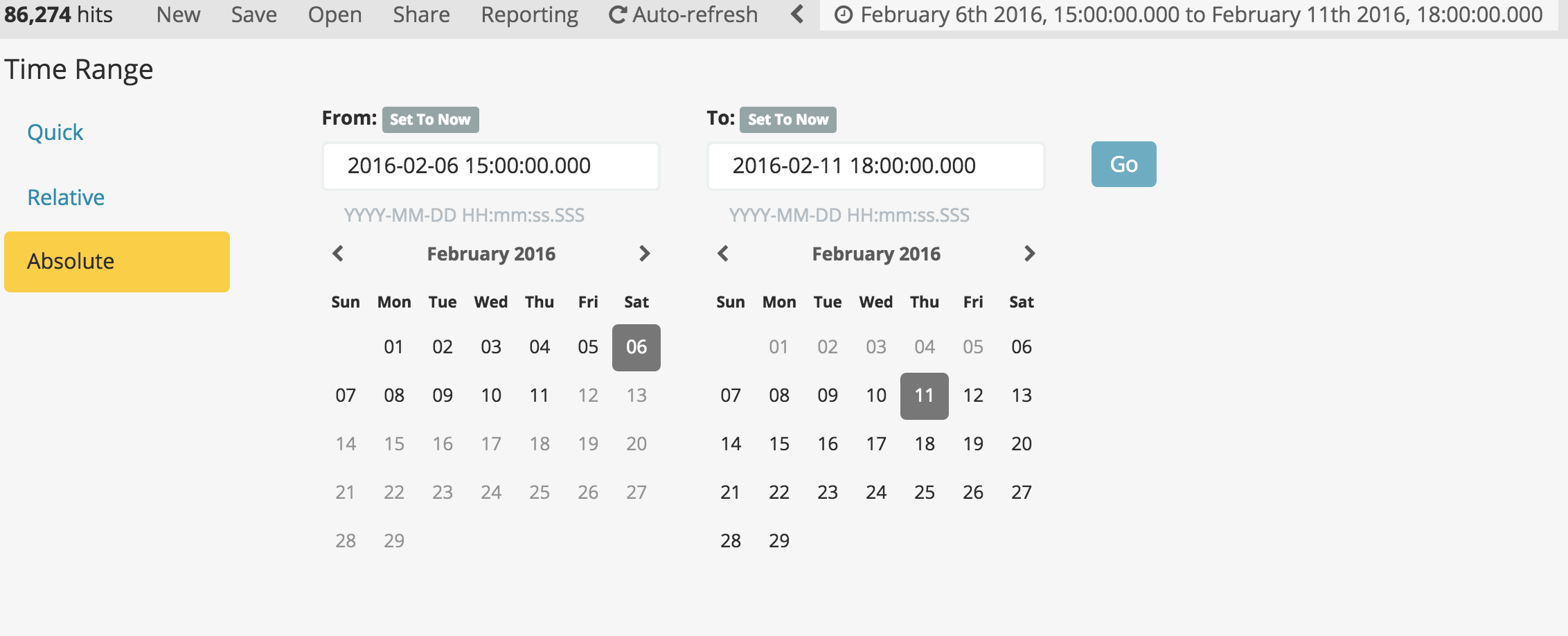
The data set contains application logs from a travel application. The application makes web services call to different airlines to get price information for quoting. A log line (document in farequote index) contain airline name and response time for each call that was made. These are the fields that we will be working with in the rest of the lab

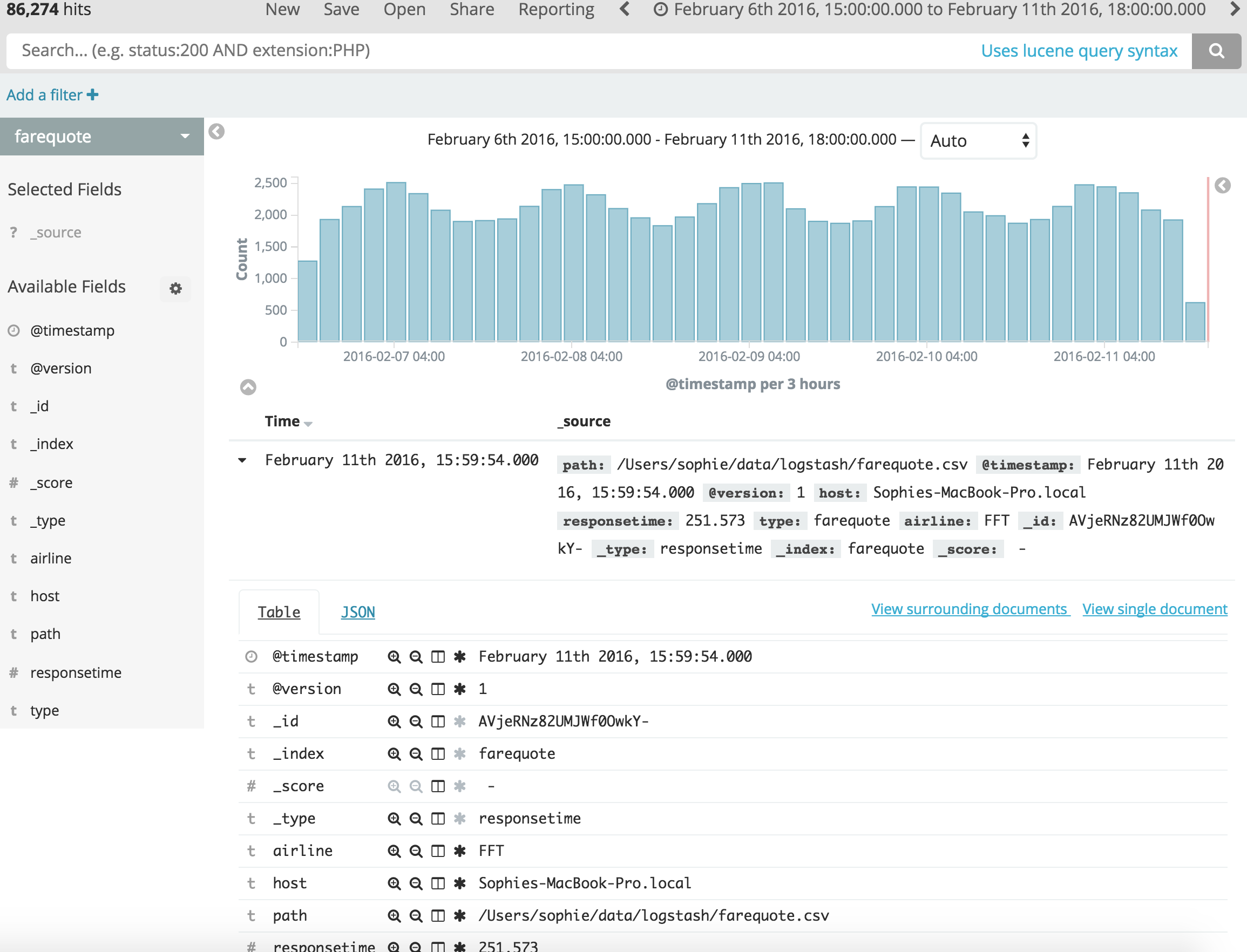
1. In Kibana, go to Discover and select farequote dataset



1. You may not see anything in the dataset since Kibana defaults to Last 15 minutes. This data set has 86,274 documents timestamped over 5 days in Feb of 2016. Click on the Last 15 minutes to change the time interval to be absolute. We are going to look at data between **2016-02-06 15:00:00.000** and **2016-02-11 19:00:00.000**







.

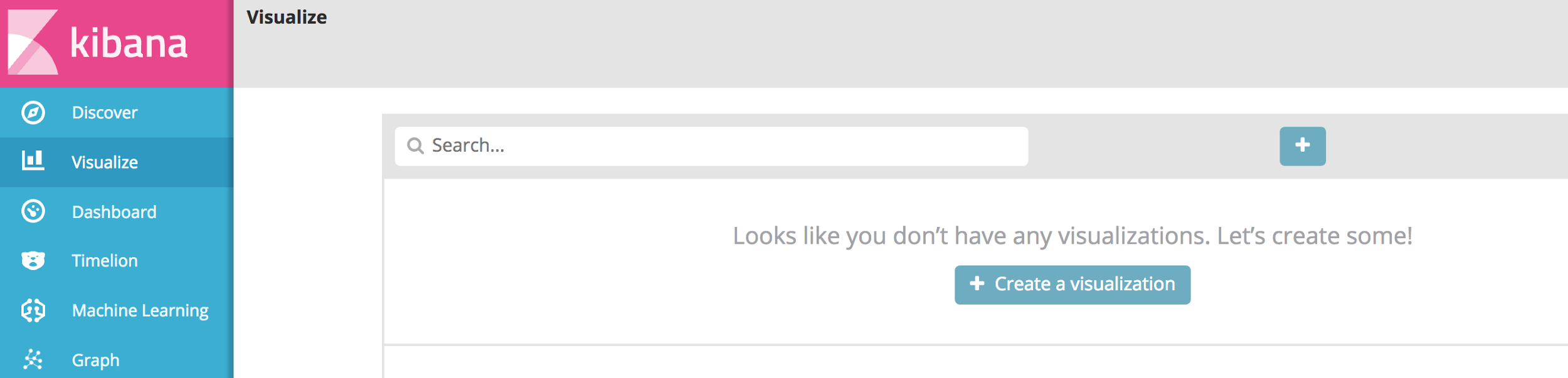
## **#2**

## **Going traditional!**

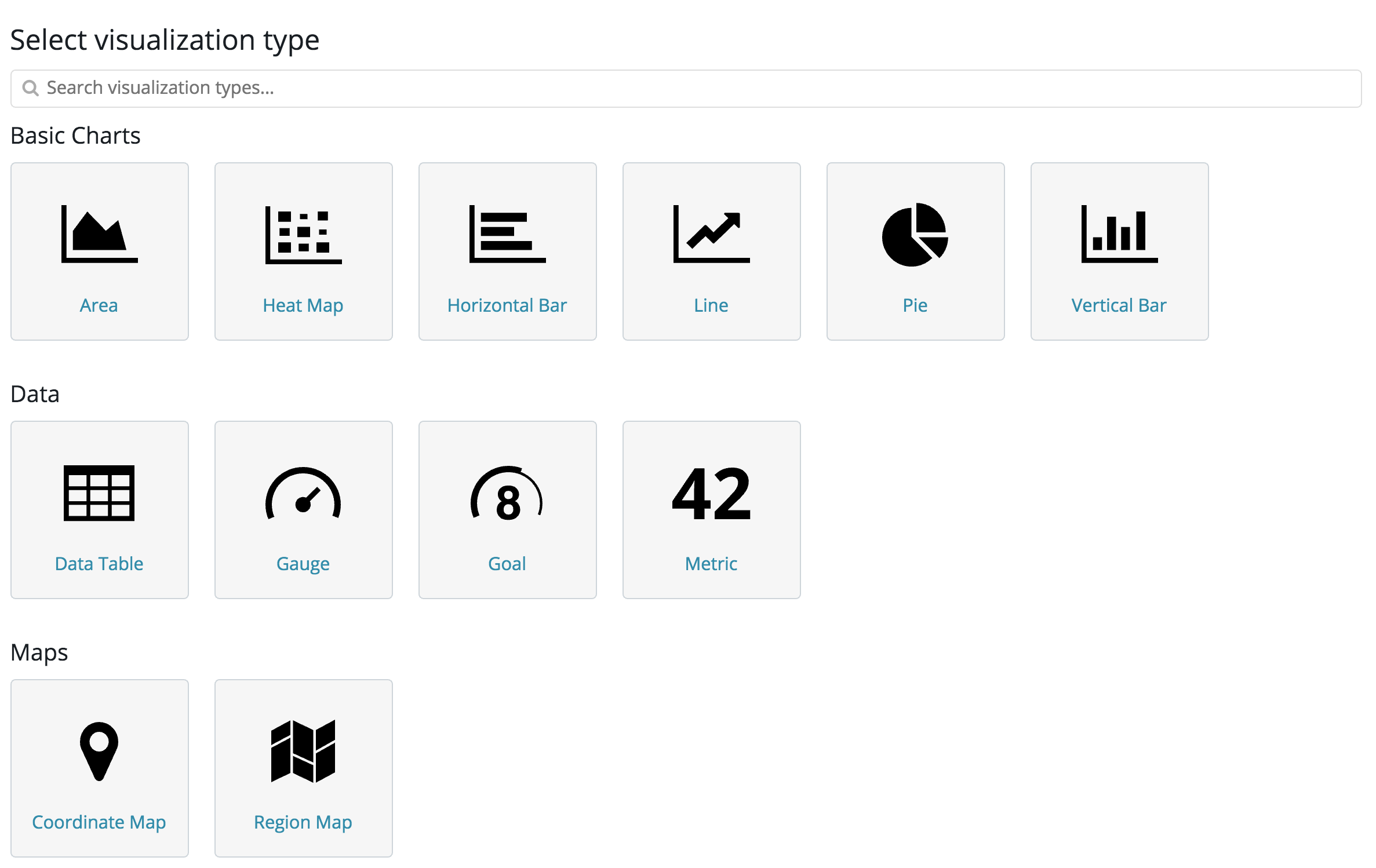
## 

Visualizing the heck out of data to detect anomalies

1. In Kibana, go to Visualize and Create a Visualization



1. Select Line visualization from Basic Charts. Select farequote index.



1. For the visualization:

Y-axis:

Aggregation: Max

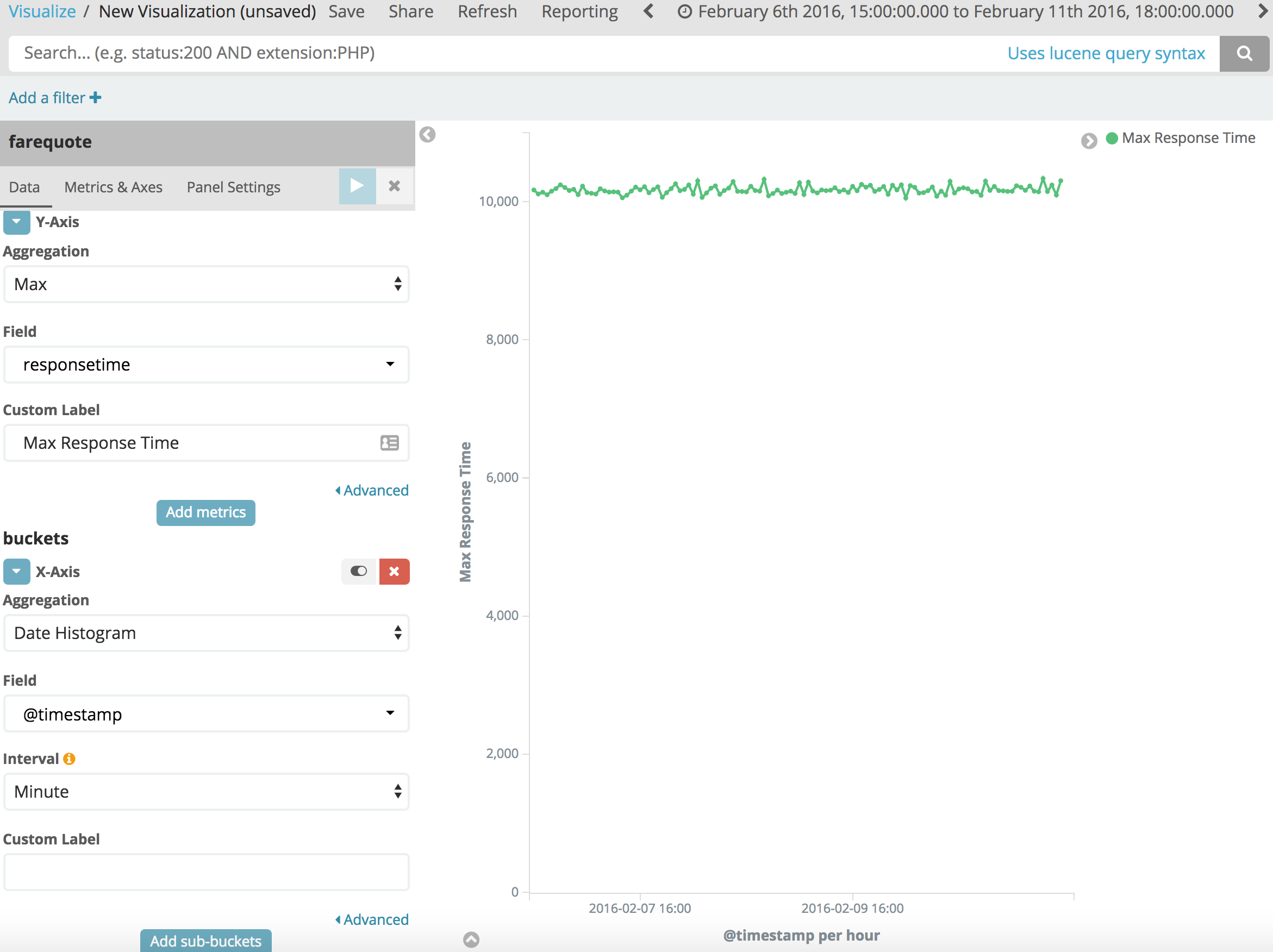
Field: responsetime

X-axis:

Aggregation: Date Histogram

Field: @timestamp

Interval: Minute



1. Add sub-buckets -> Split Series

Sub Aggregation: Terms

Field: airline

Order By: metric: Max Response Time

Order: Descending

Size: 20

Click on Apply Changes icon. The visualization on the right should look like below.



In hope of finding anomalies in our data, we just finished visualizing response time values over time for all the different airlines in the data. Can you see any anomalies in the data that stands out in this visualization? Hard, right? And this is a static data set with only 86,274 records! Imagine of going traditional with anomaly detection in a data set that’s constantly changing.

Save the visualization. Call it response time by airline.

## 

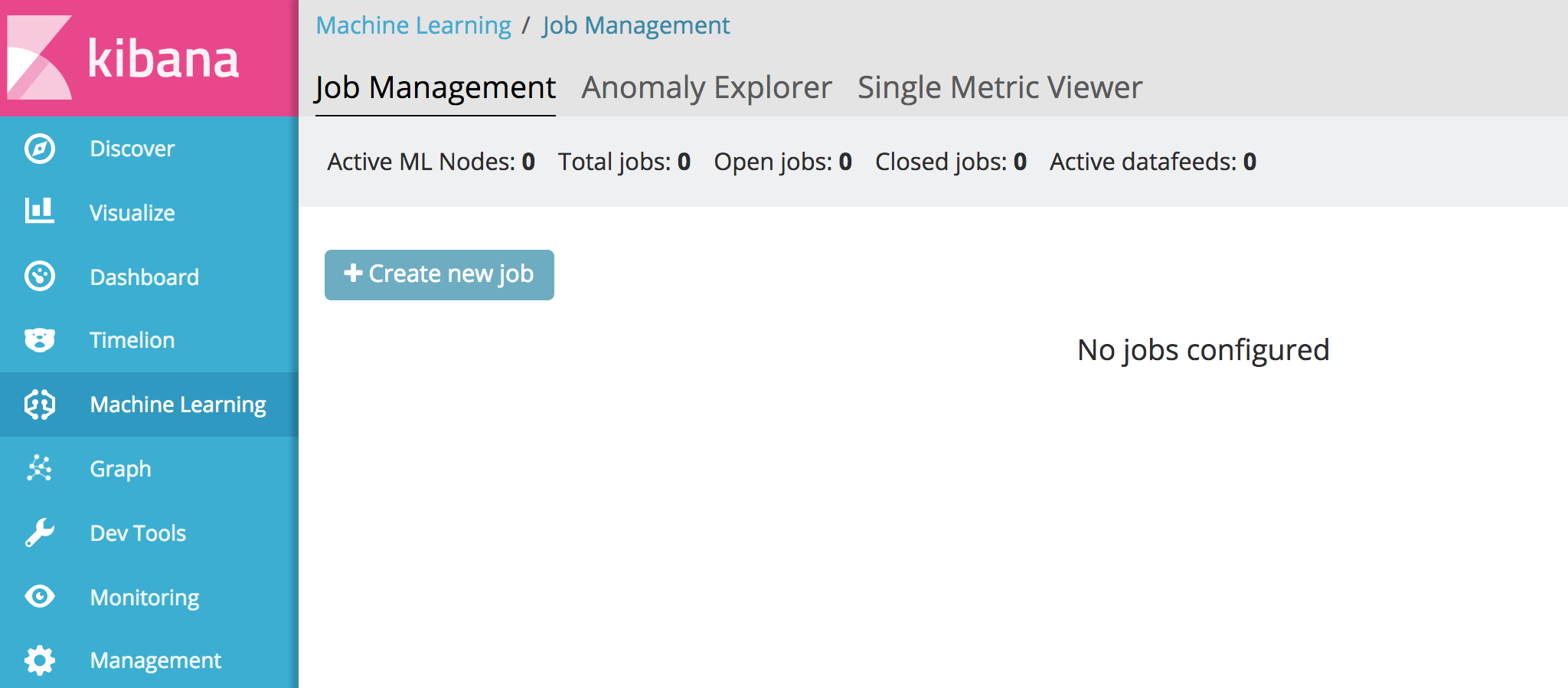
## **#3**

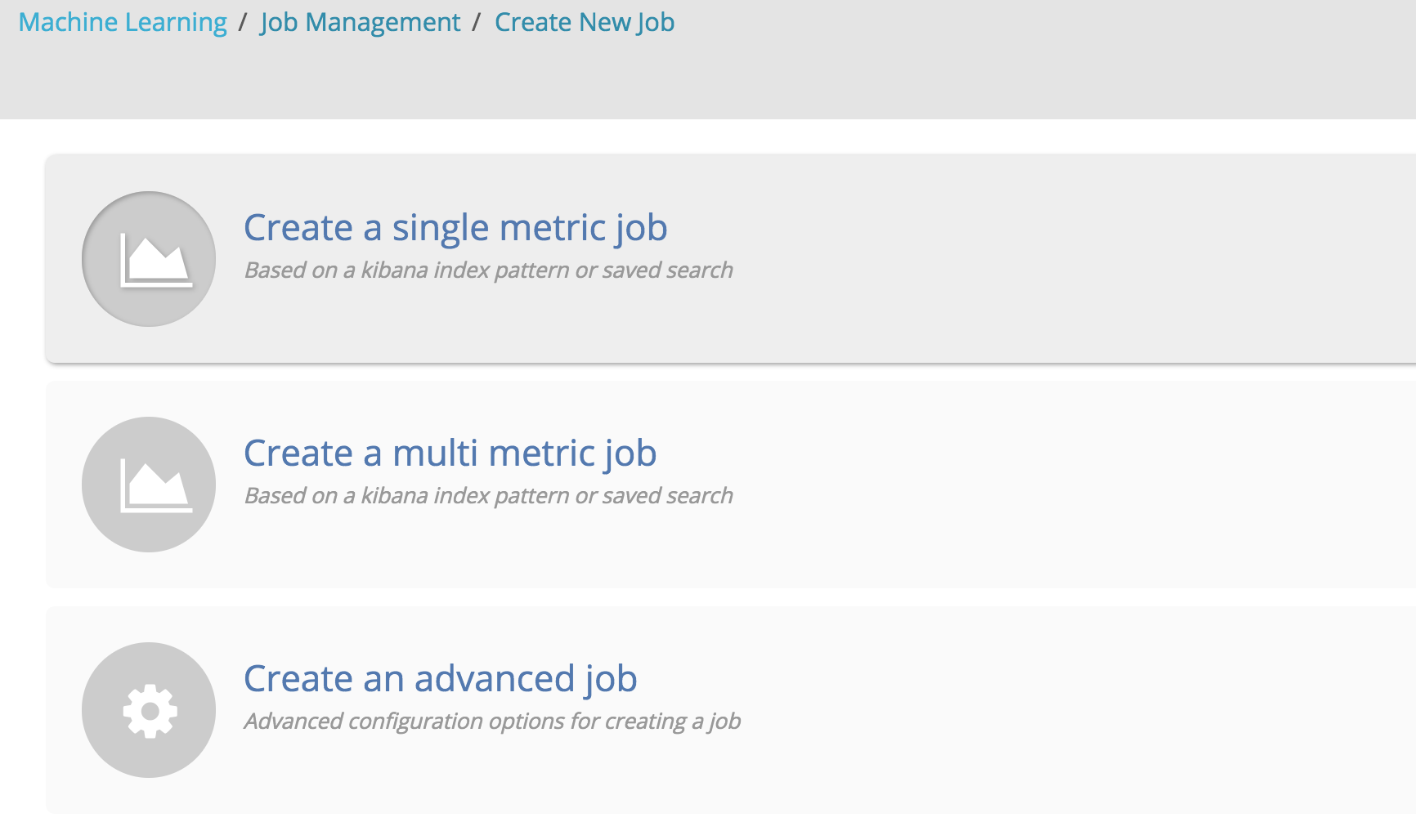
## **Machine Learning(ml) to the rescue!**

## 

## **Automatic anomaly detection**

1. In Kibana, go to Machine Learning tab -> Create a new job -> Create a Single Metric Job





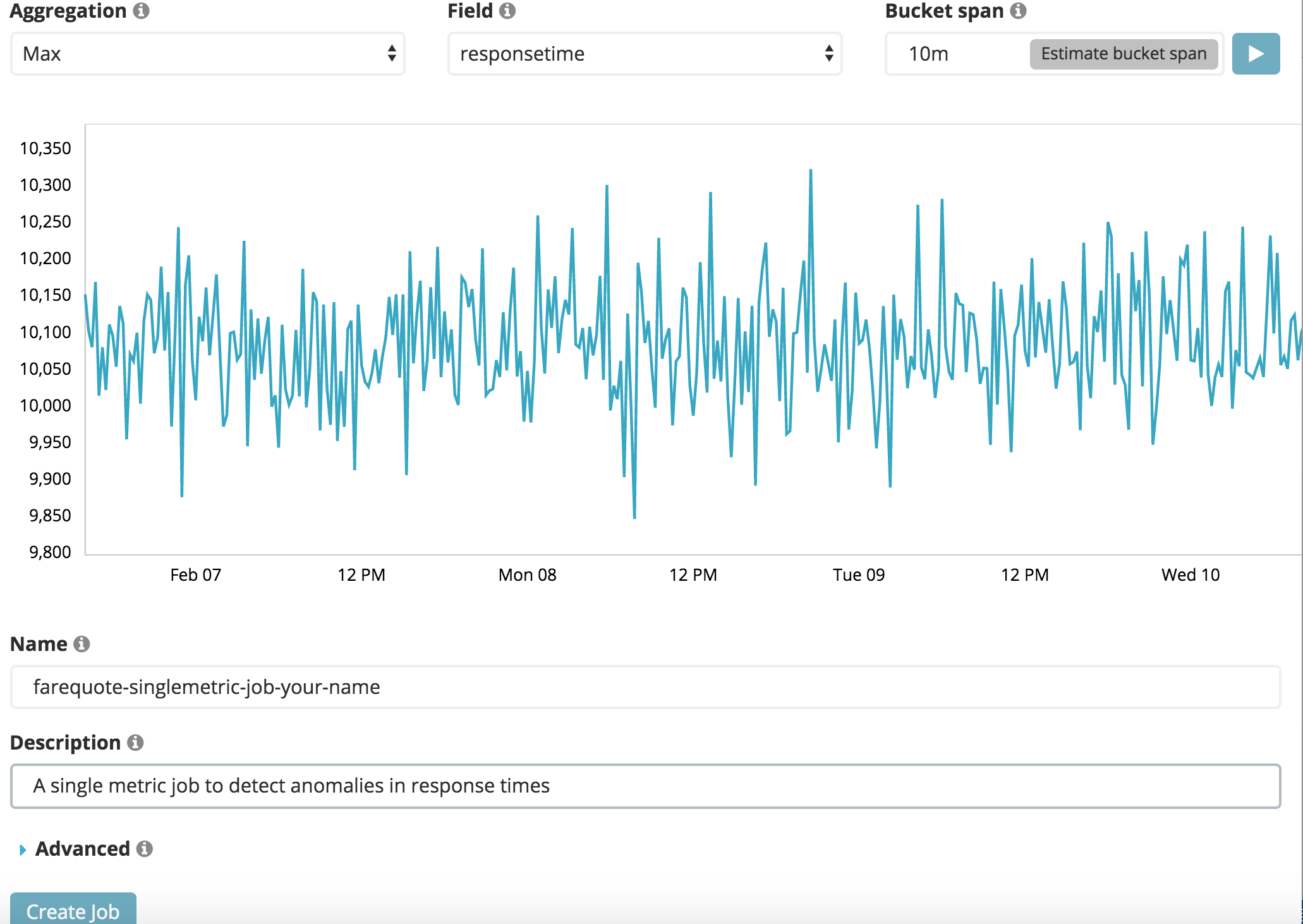
1. Select farequote index. On the Single Metric job creation page, select the same time range or select the button Use full farequote data that is located in the top right.

Aggregation: Max

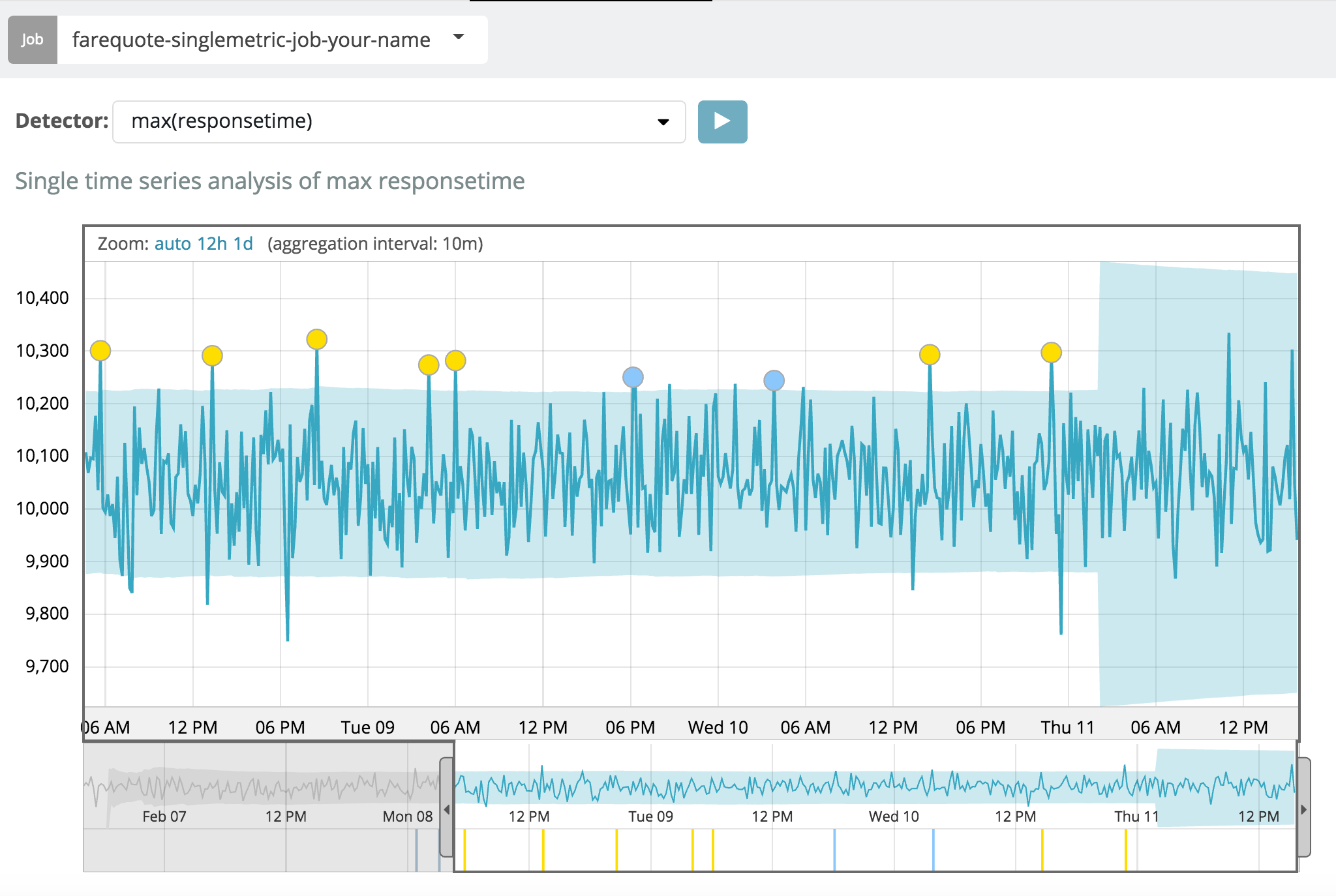
Field: responsetime

Bucket span: 10m

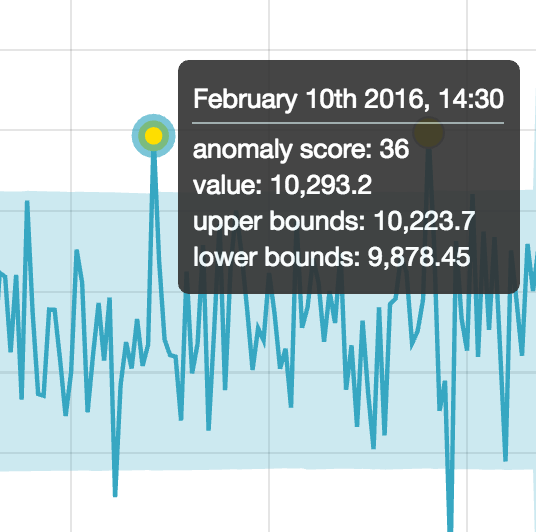
Name: farequote-singlemetric-job-your-name

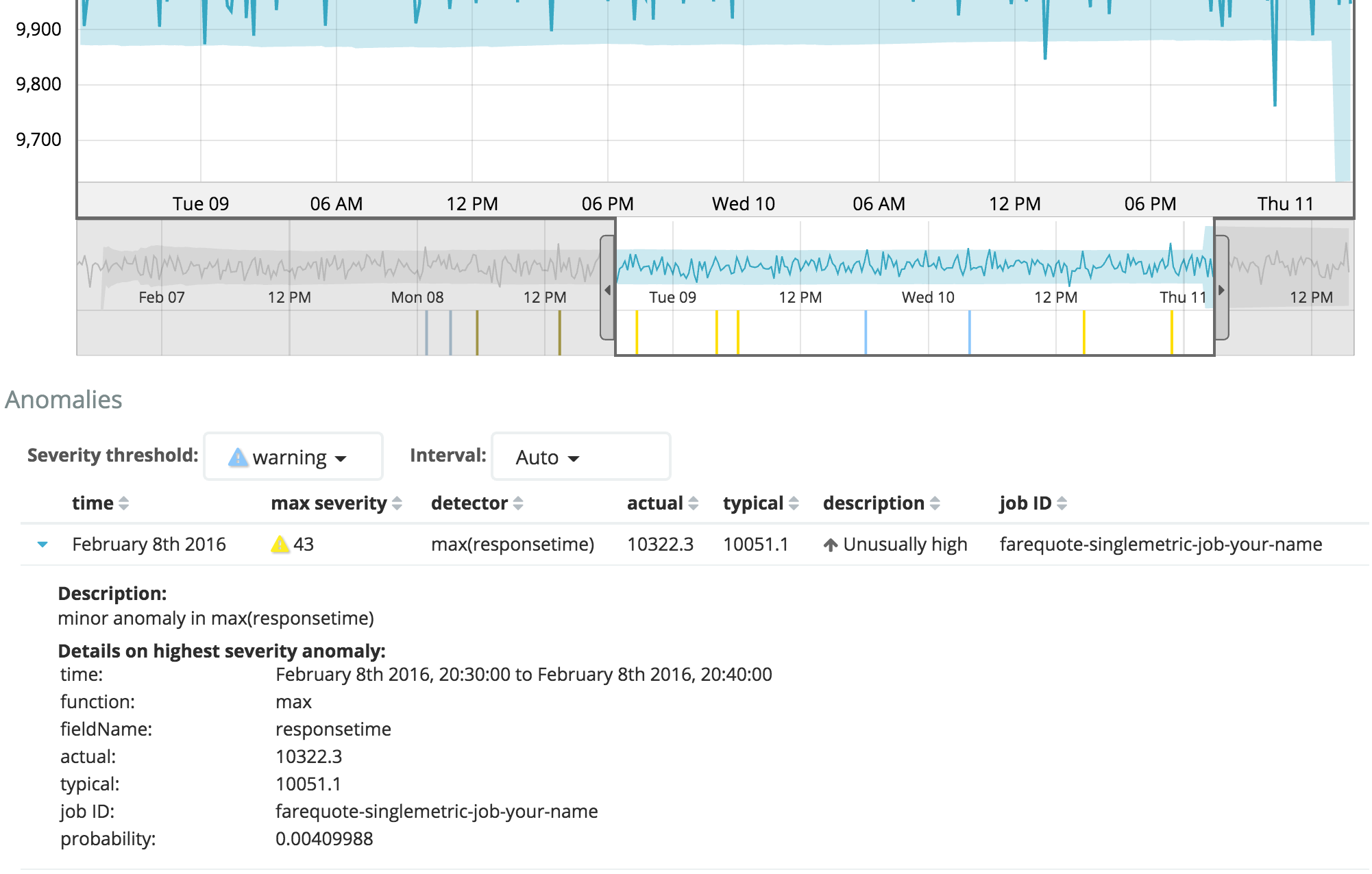


Click on Create job. Once the ML engine has finished the analysis, Click on View Results



Notice that there are few anomalies highlighted by the ML engine. Hover over one of them to get more details. If you click on it, scroll down the page to see further details about the anomaly. Please note: the absolute date / time of the anomaly will be represented based on your browser timezone, so may differ from the below.





We can see more information about the anomaly, but the fact that we created a single metric job, it doesn’t help us get any more information. Like what caused that anomaly, who influenced it. To help with that, let’s go ahead and create a multi metric ml job.

1. Go back to Machine Learning. + Create new job -> Create a Multi metric job -> select farequote index. On the Multi metric job page

Under Job Settings

Select *event rate*: count

Select responsetime: Max

Under Split Data

Select airline

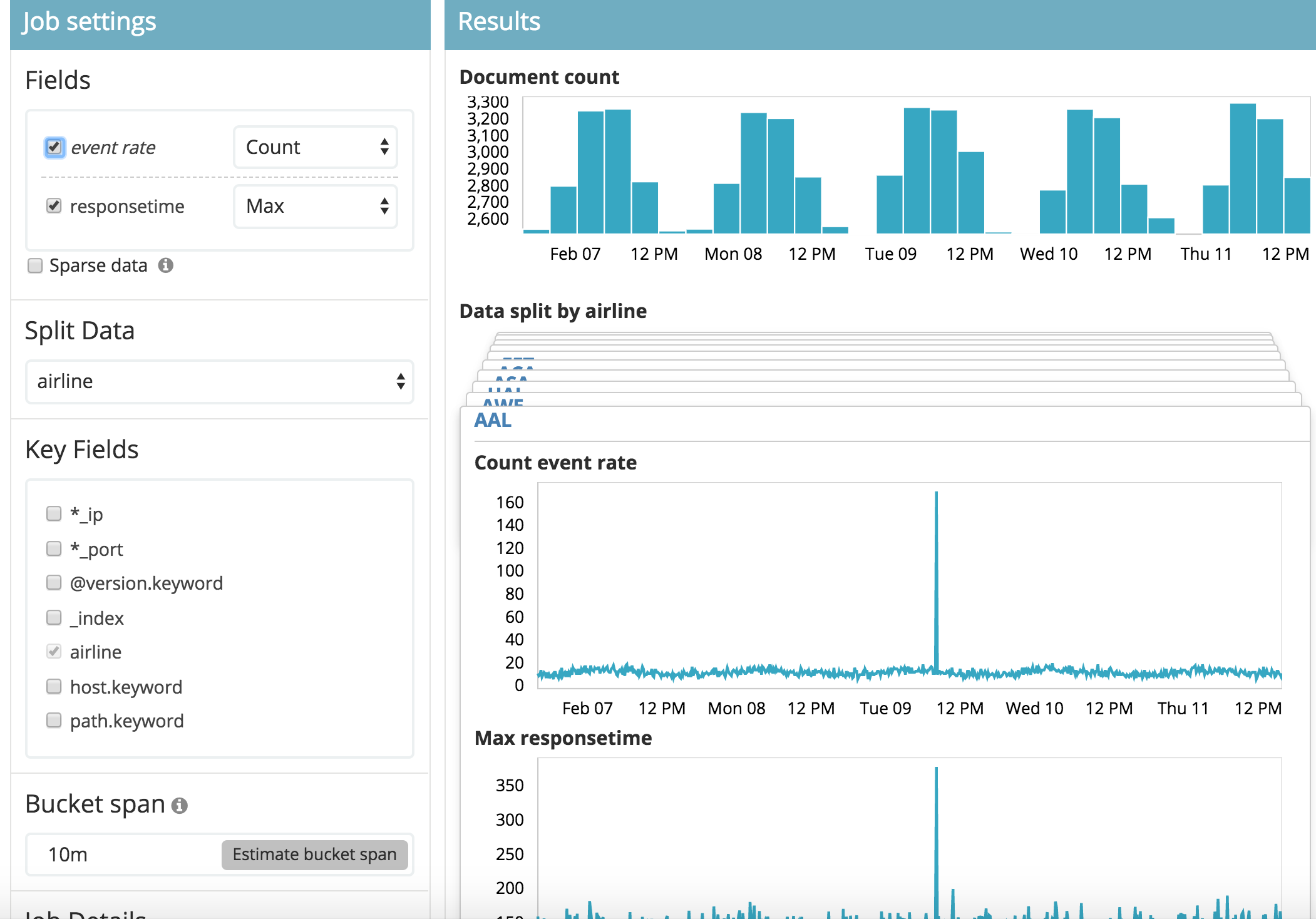
Bucket Span: 10m

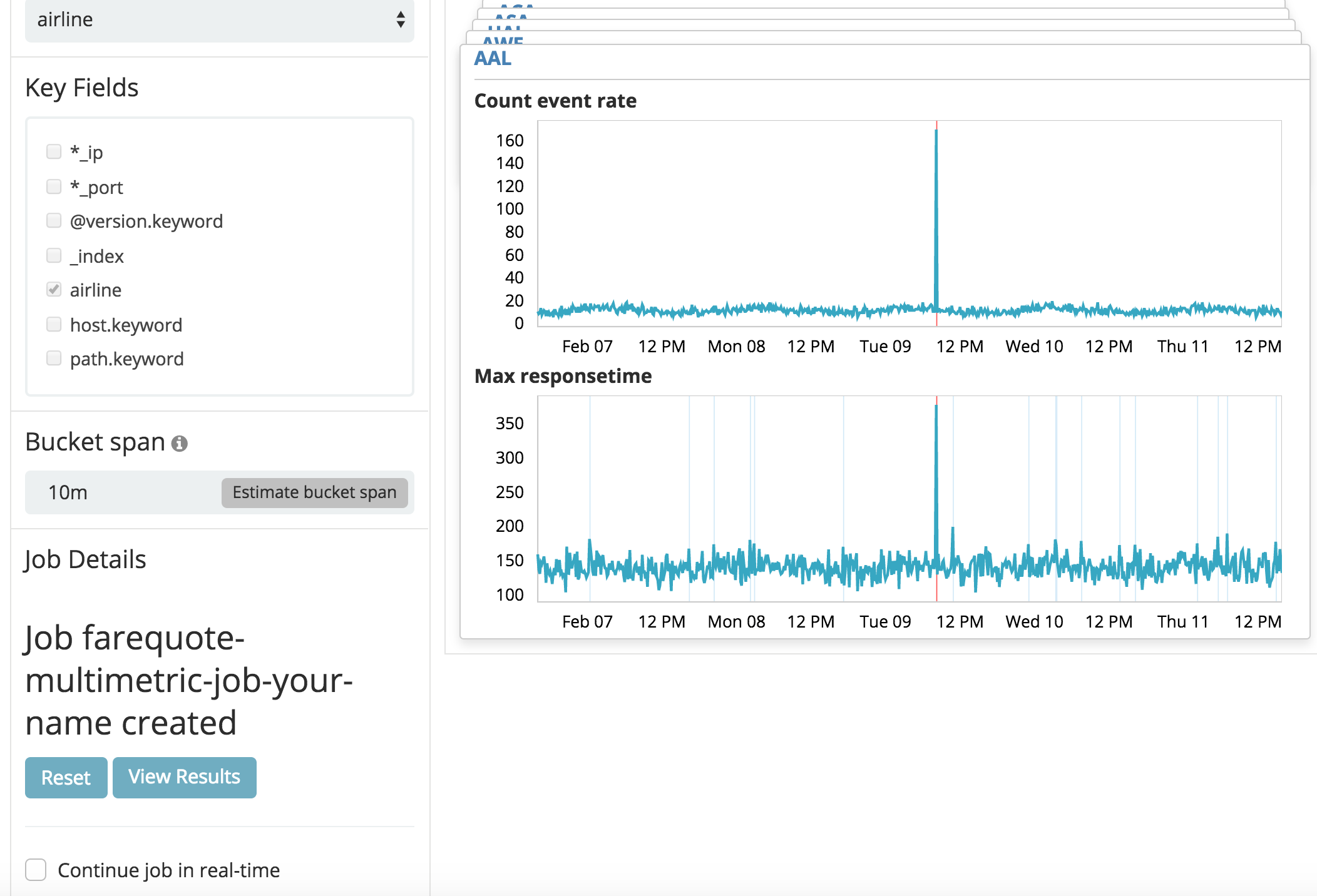
Under Job Details

Name: farequote-multimetric-job-your-name

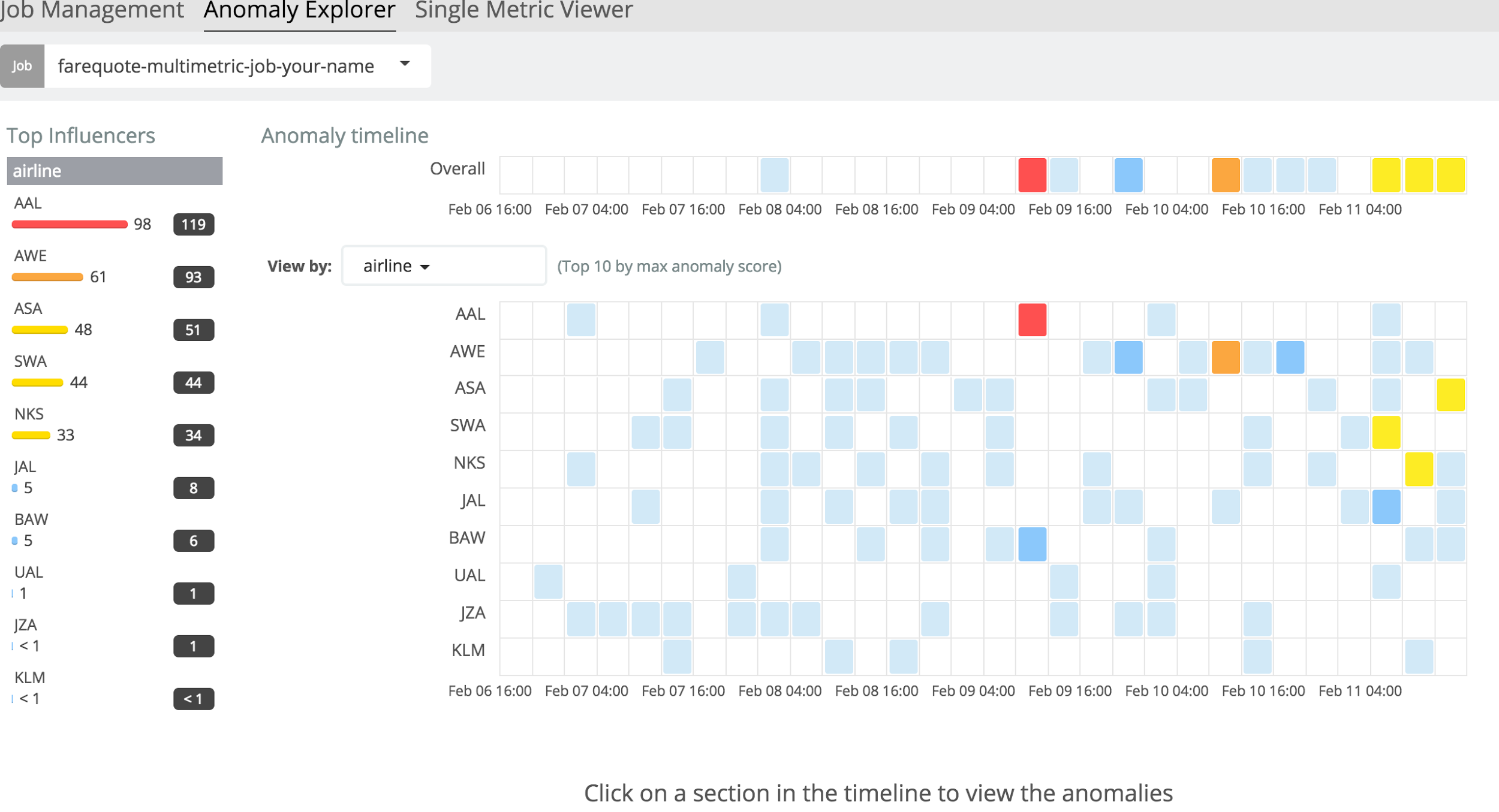
Description: A multi metric job to detect anomalies and the influencers

Click on Create job. Once ML engine is done analyzing the data, click View Results

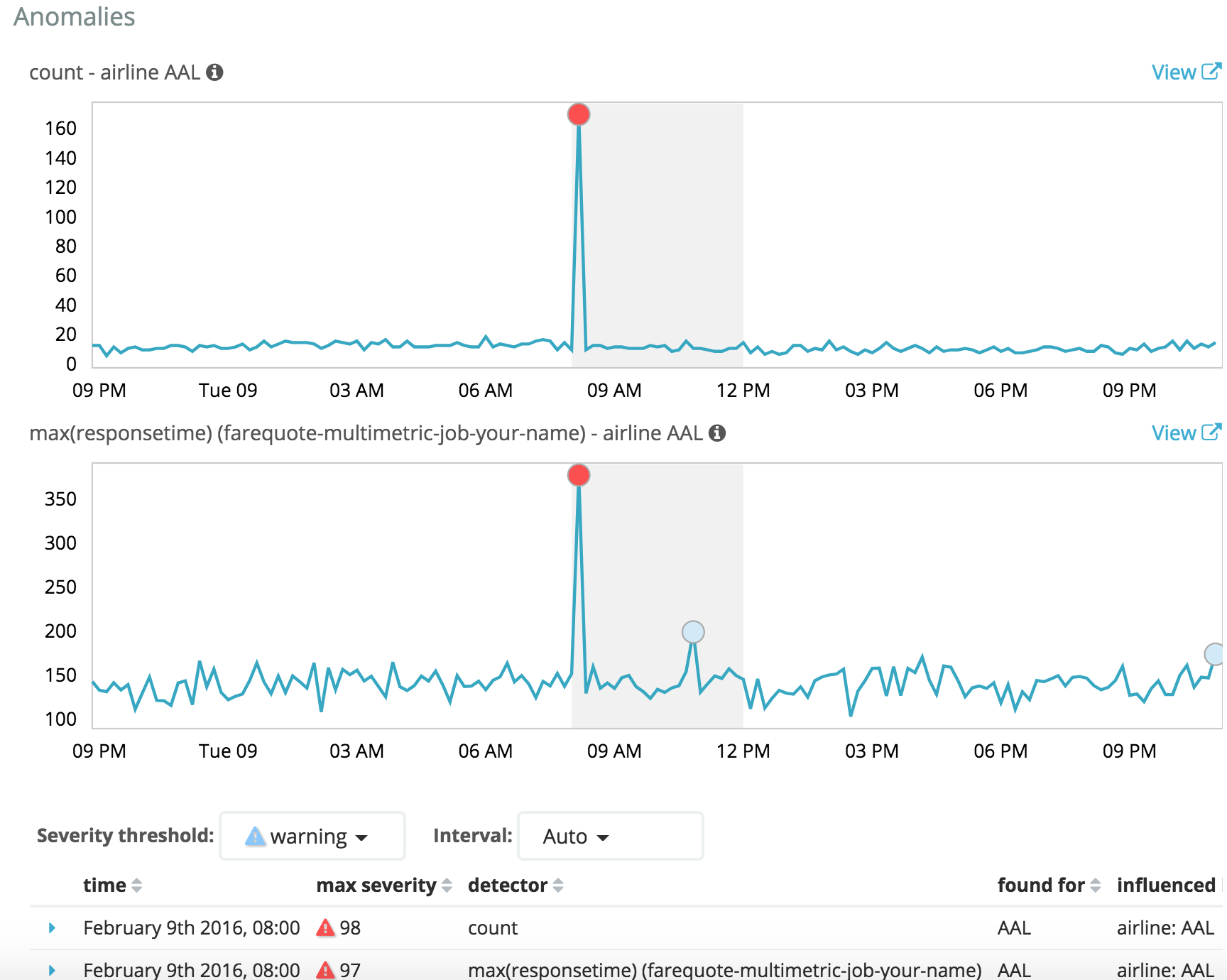


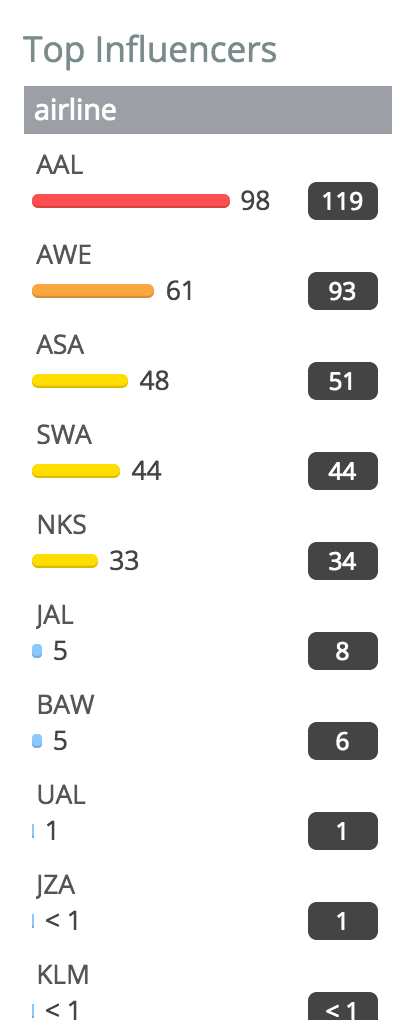


1. In the Anomaly Explorer, we see the anomalies in our dataset that were detected. The anomaly detection is also split across the airlines. Different colors represent the score of an anomaly, an indicator of its potential criticalness.



1. Click on the anomaly for AAL (American Airline). Scrolling down on the page will show further information on the anomaly. Also notice the Top Influencers.





As we look in the details of anomaly, we can see there is a correlation between count rate and response time of the American Airline web service. As the event rate went up – anomalously high number of calls to the web service, the response time of the service also went up to be an anomalous event. Compare the just experienced ease of automatically detecting an anomaly, and understanding what led to that anomalous occurrence, with the traditional approach that we may take in absence of having the ML capability.

Remember this?



## 