SOLUTIONS ENGINEER TECHNICAL EXERCISE

Chris Reites

October 2018

BEFORE WE GET STARTED…

Here's a little bit about me. My name is Chris Reites, and I'm currently in presales at CA Technologies helping to sell their DevOps/Continuous Testing software solutions. Prior to joining CA, I spent 16 years at FedEx as a software developer, QA Engineer, and Enterprise Architect. Having spent a lot of time as a customer of enterprise software, I believe I have great perspective and appreciation for the challenges that my customers face. The reason why I made the move into presales, and why I enjoy it so much, is that I love being able to combine the excitement of sales with my passion for helping customers solve their problems and bring more value into their own organizations.

Now I'm looking for my next adventure. That's where you come in! I've been talking to people I know at Datadog, and it sounds like an amazing place to be. Even before I knew about this technical exercise, I signed up for a free trial at datadoghq.com and started playing around with the technology. I was highly impressed and that's what ultimately convinced me that I wanted to join your team. My hope is that after reviewing my submission, you'll feel the same about me.

When I'm not busy helping customers with their digital transformations, I enjoy living in the Tampa area with my wife and two children. On the weekends you can usually find us boating or at the beach. I also like running, but to ensure I don't become *too* healthy, I also enjoy brewing my own beer.

https://www.linkedin.com/in/chris-reites/

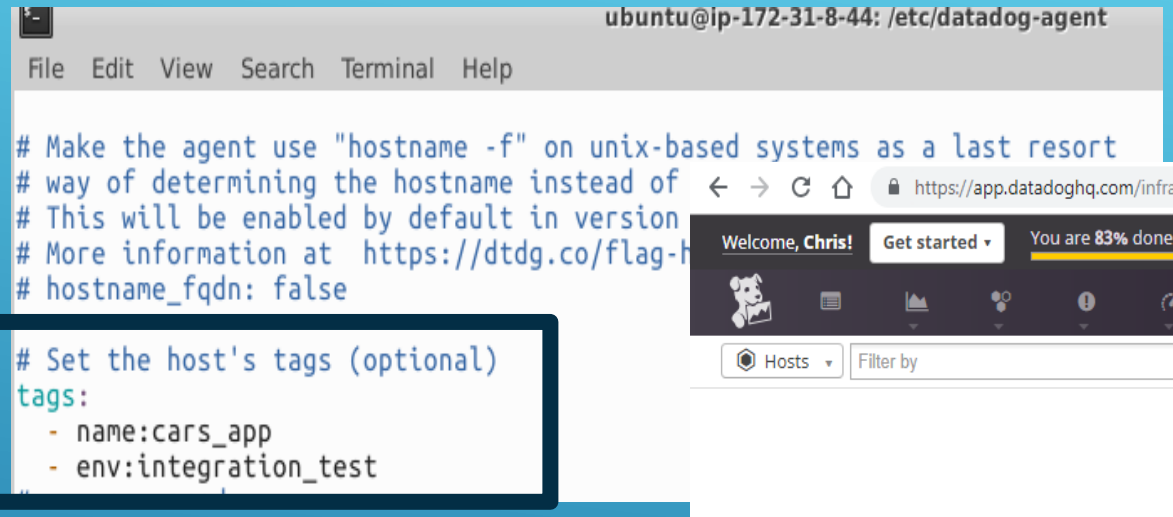
MY ENVIRONMENT

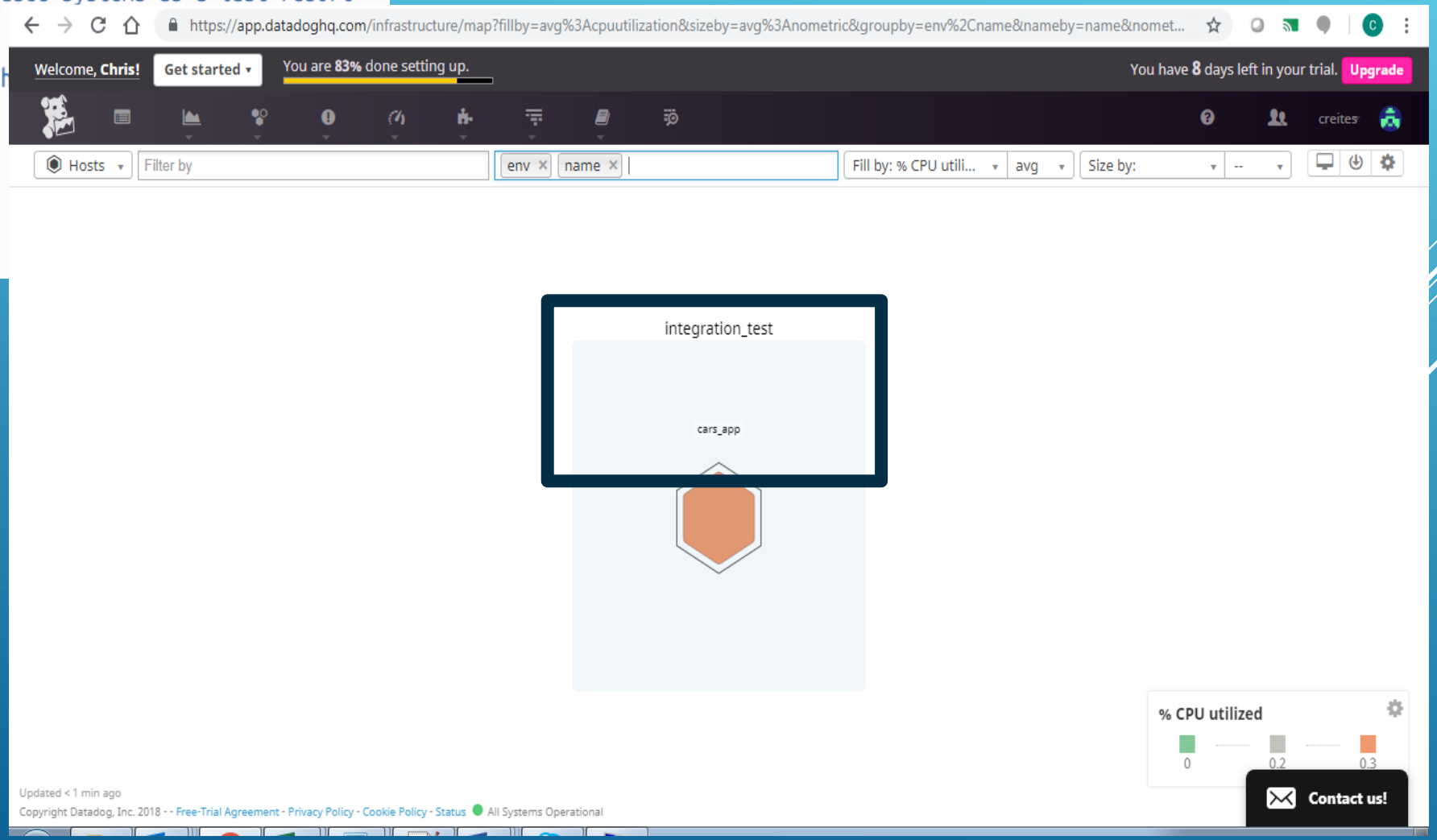
For this exercise, I chose to use an AWS instance I already had setup. It's running Ubuntu 16.04 and I have a few java apps running there that I was able to play around with to check out more Datadog functionality. I previously had configured my applications to use SQL Server as the backend database, but because the current version of that integration doesn't work on Linux, I installed MySQL and changed my apps to use that for the backend. Finally, in order to create some interesting moments that I could look at metrics for, I used BlazeMeter, which is a cloud-based load generation solution, to create a lot of traffic to my applications.

Section 1

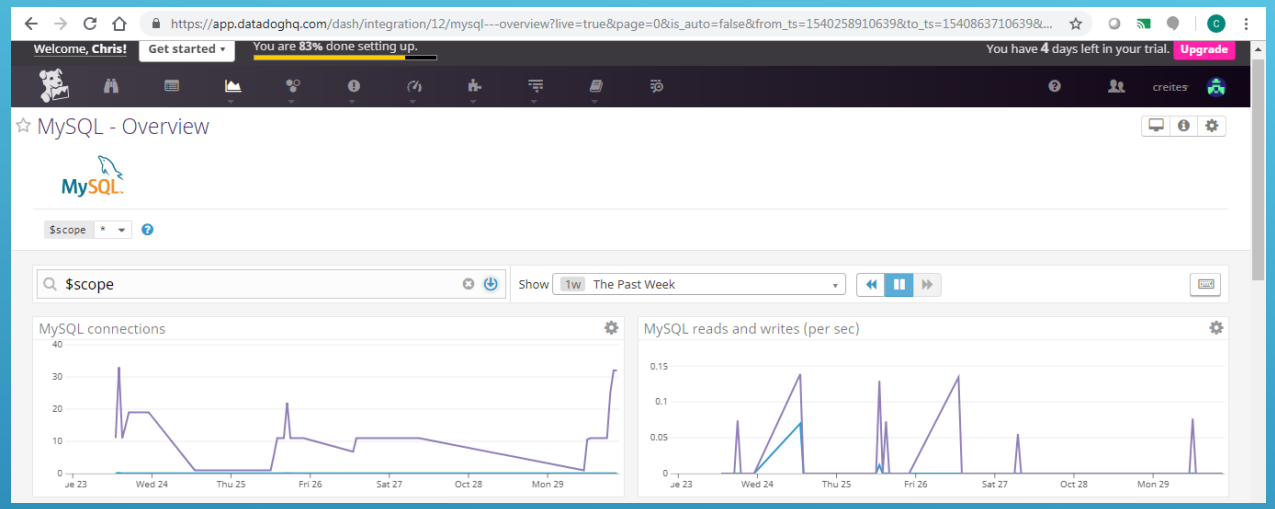
COLLECTING METRICS

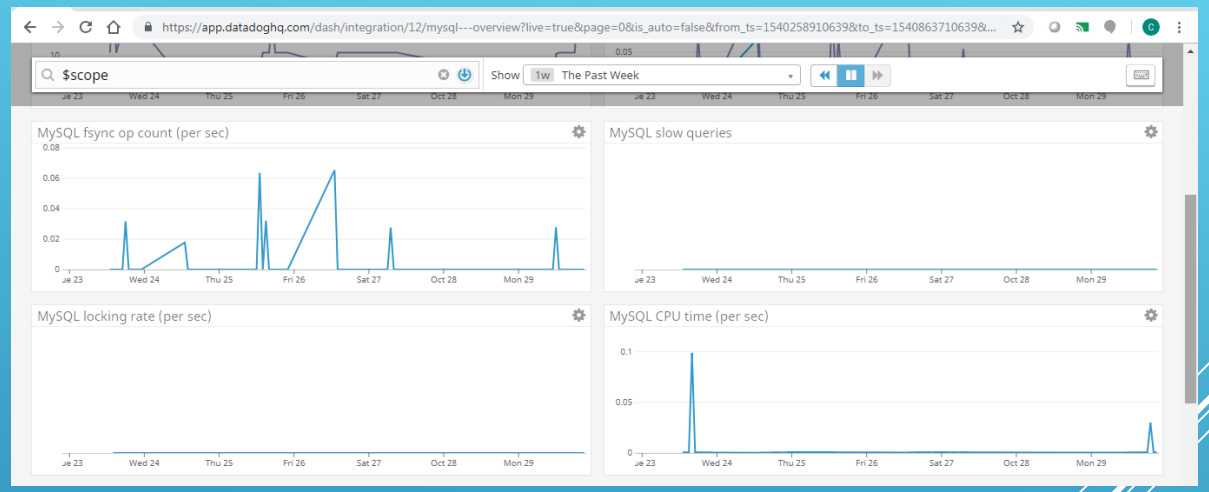
The first thing I needed to do was get the Datadog Agent stood up on my server, and configure it with a meaningful name and some helpful tags. To do this, I simply had to edit the datadog.yaml file found in /etc/datadog-agent

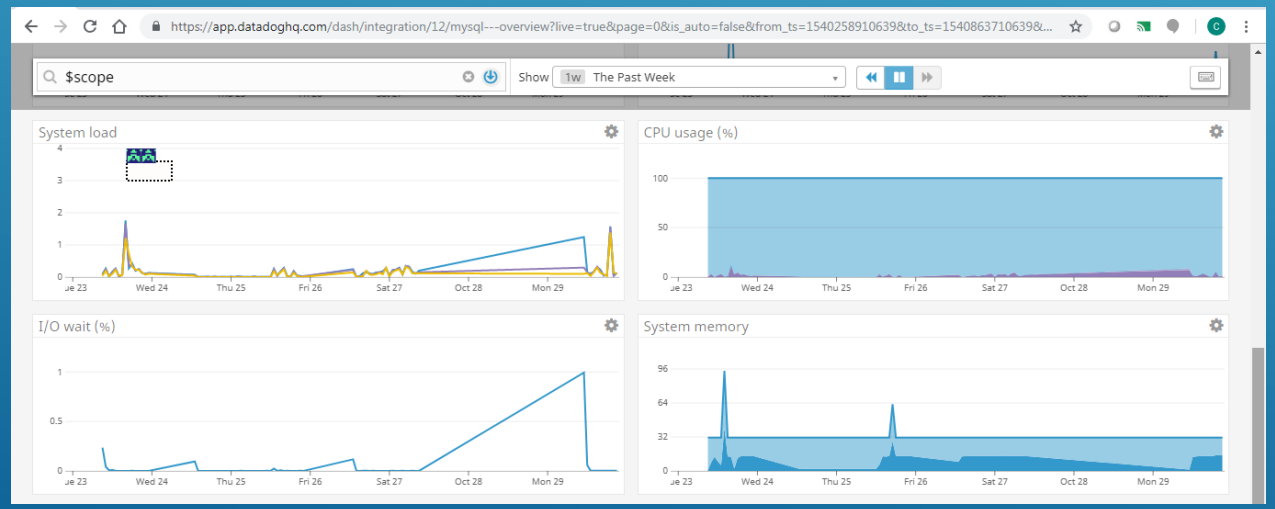


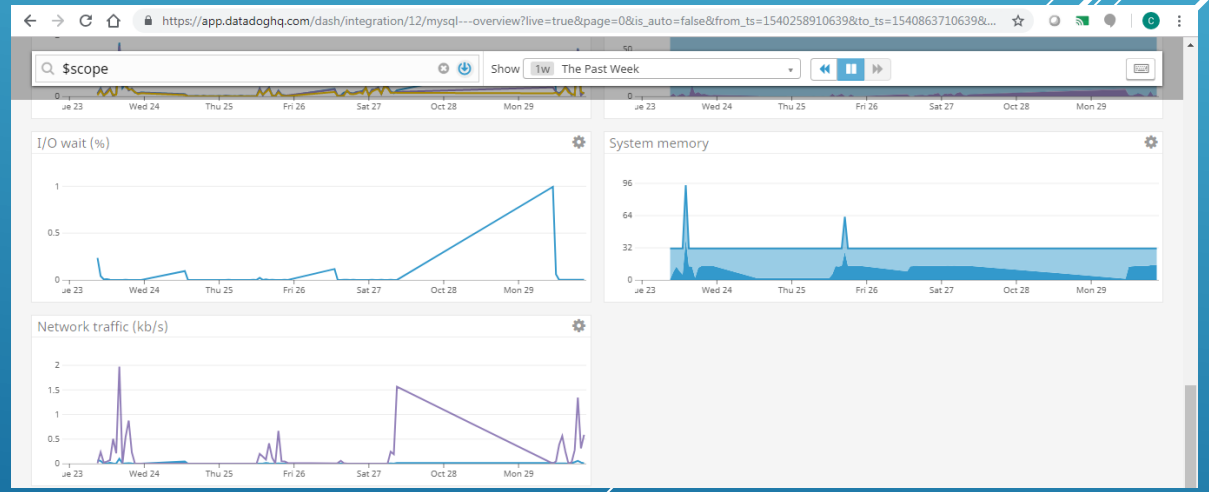


Next I added in the MySQL integration so that I could keep an eye on my database metrics. One of the nice features of Datadog is that when you install integrations, you get default Dashboards which include popular metrics. Here are some shots of my default MySQL Dashboard. It's pretty easy to spot when I was running my load tests!

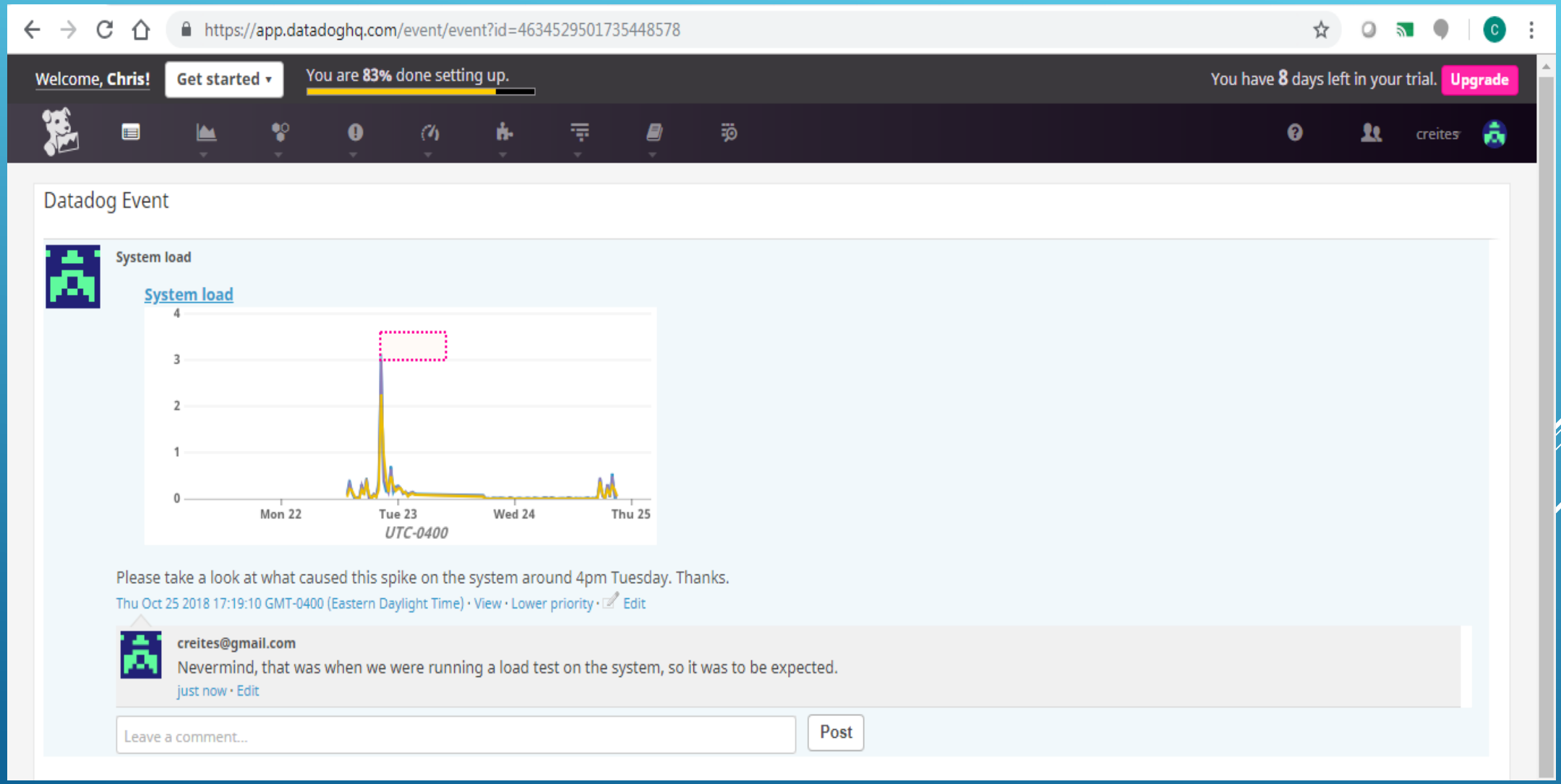




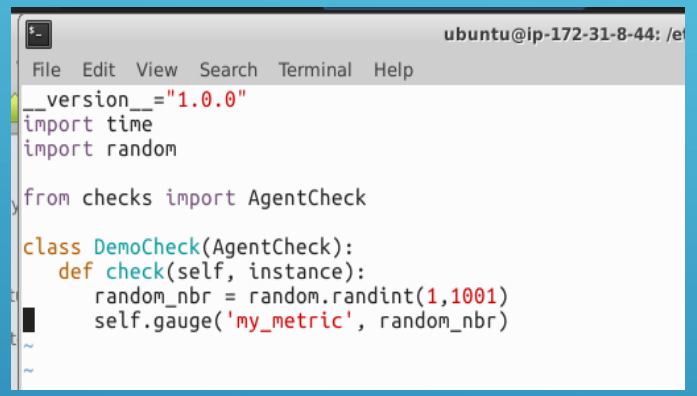


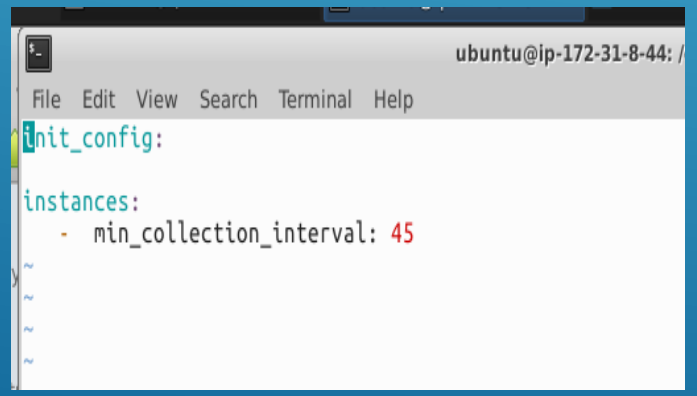


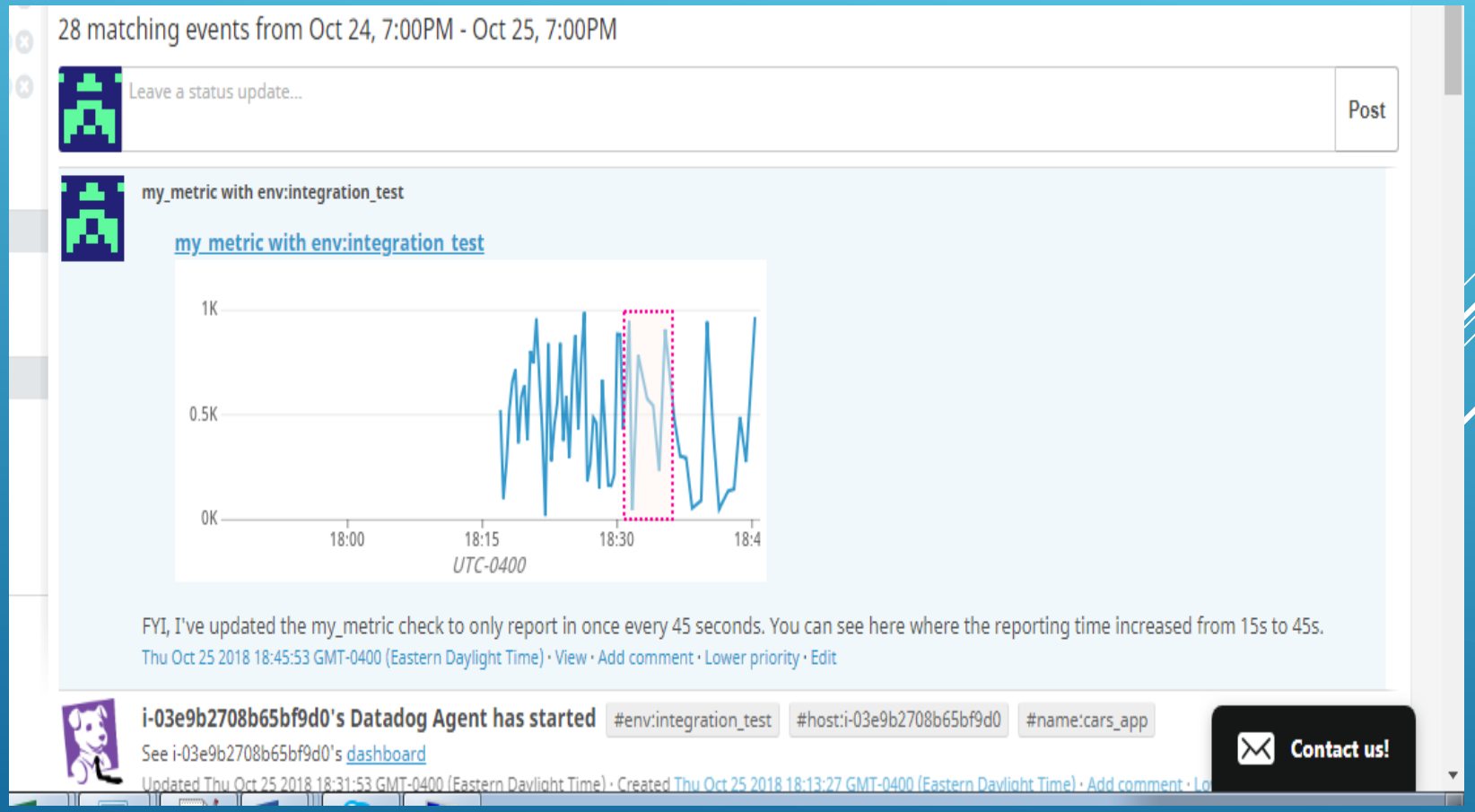
Speaking of being easy to spot… When you do see something that catches your eye, it's very easy to collaborate with your team. Here, I created a snapshot of one of the spikes, annotated it, and then sent if off for the team to look into. Unfortunately, I'm the only one on my team, so it was up to me to figure out what was going on and reply. Sometimes it feels like I have to do everything!



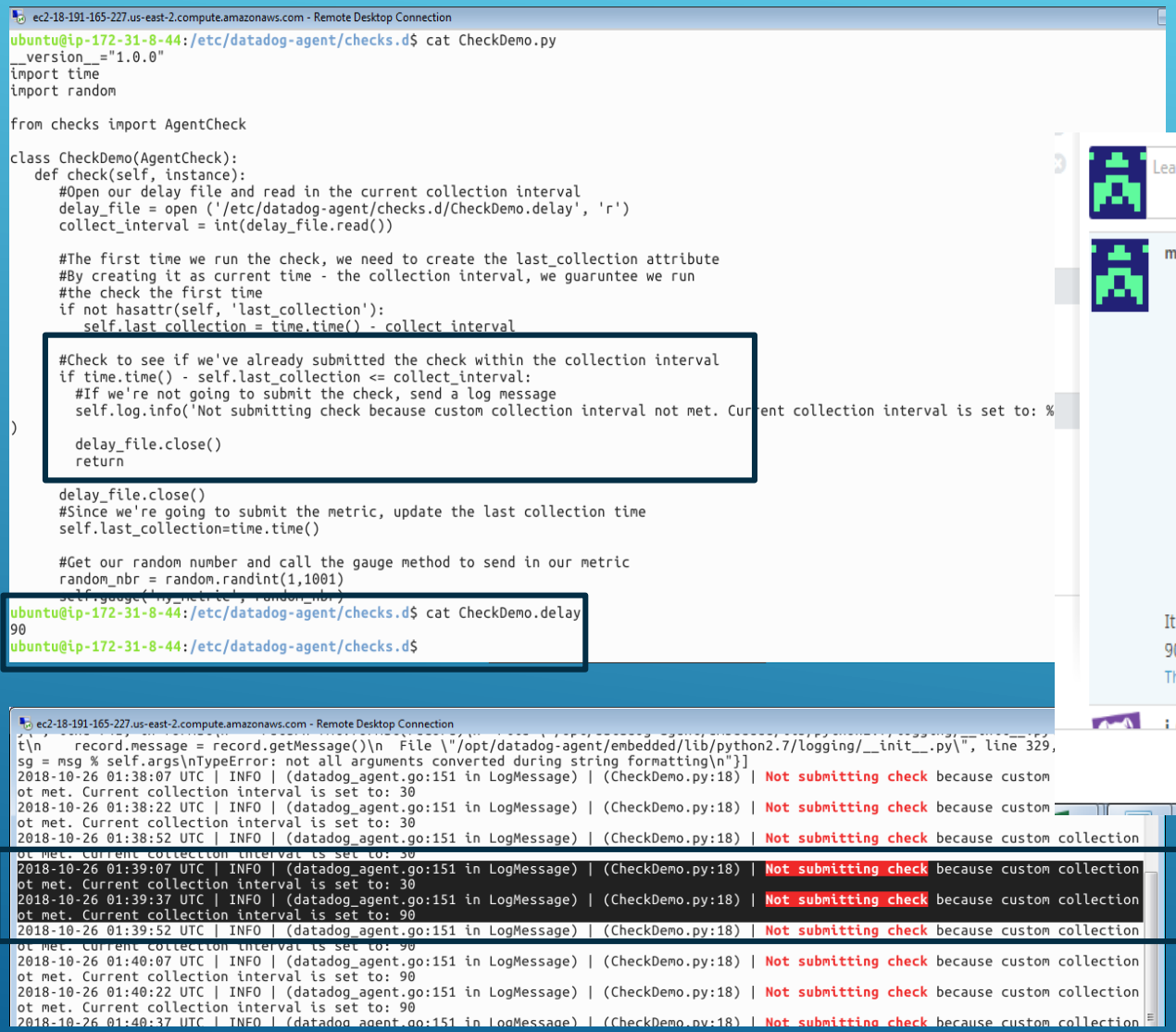
In order to further test out the metric gathering capabilities of Datadog, I wrote a simple python script that generates a random number between 1 and 1000 and then submits the value to the Agent with the name my_metric. After setting up the script, I then changed the collection interval to be every 45 seconds instead of the default 15 seconds.

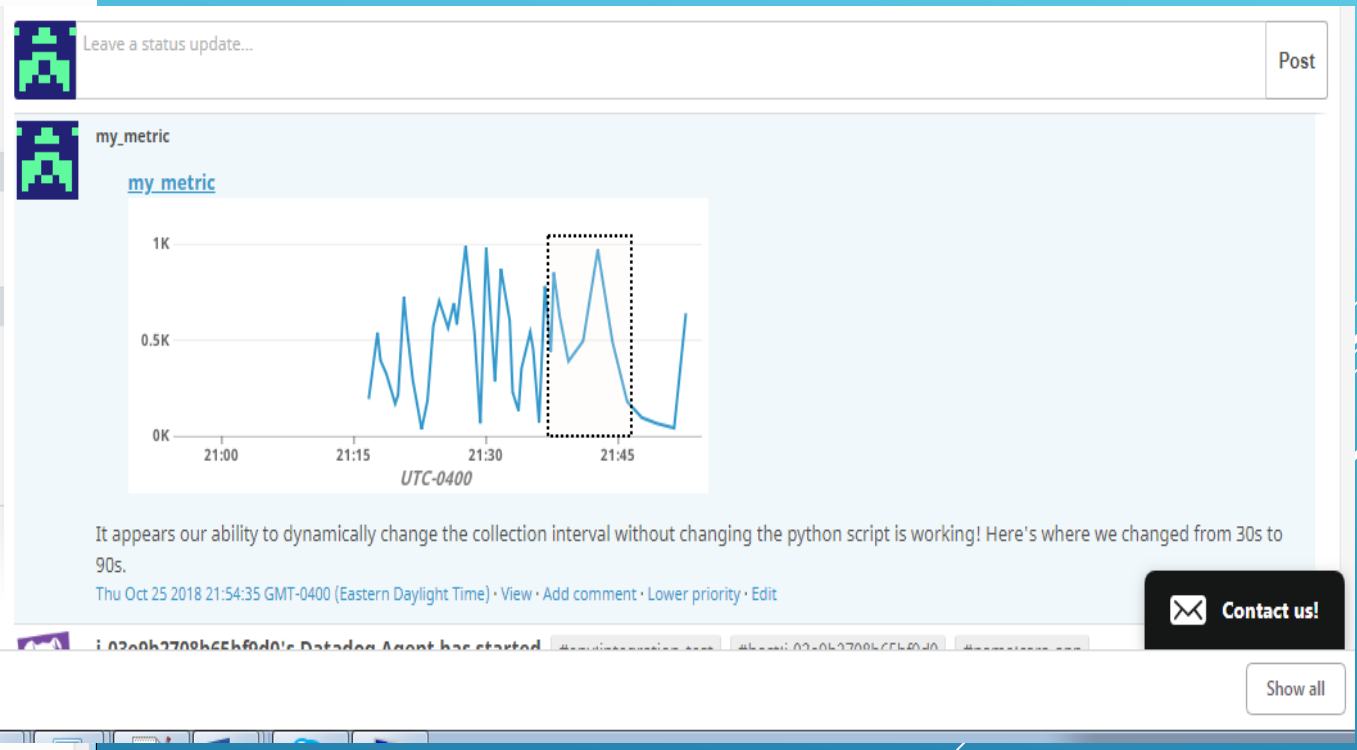






For fun (and bonus points!) I added in the ability to dynamically change the collection interval by simply updating a delay file on the filesystem. Whenever the check is fired, it will re-read the delay file and skip reporting the metric if the collection interval hasn't passed.

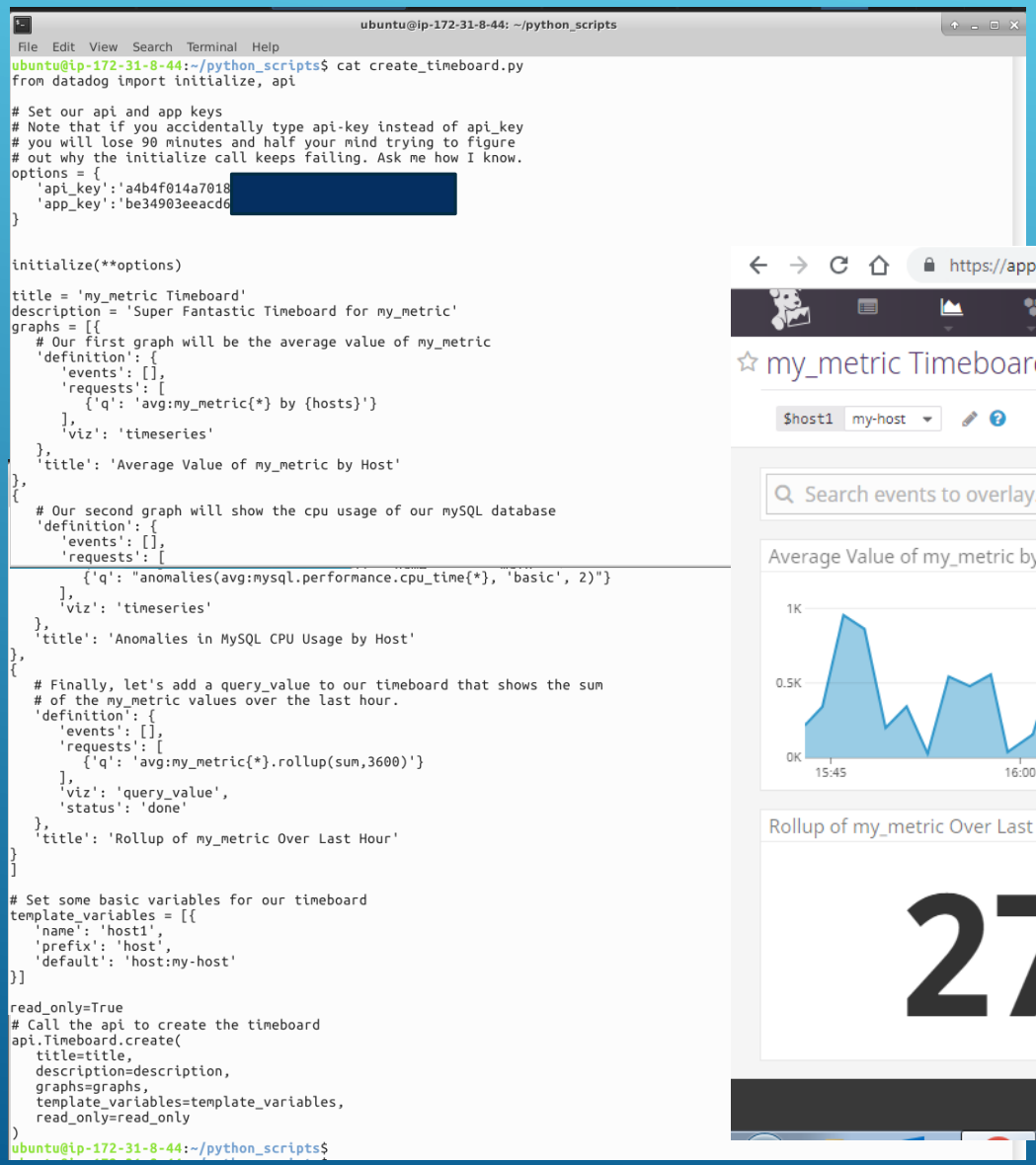


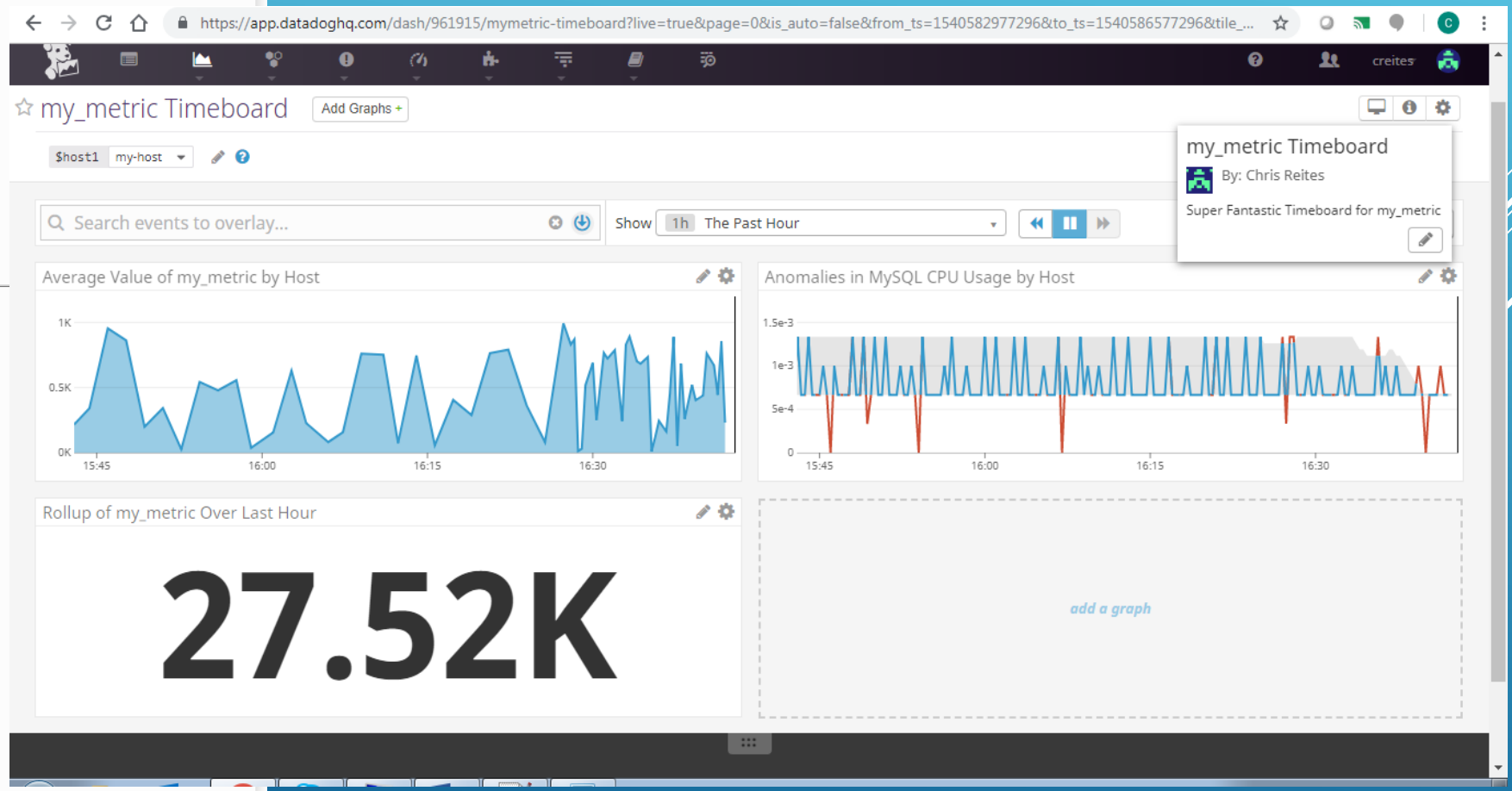


Section 2

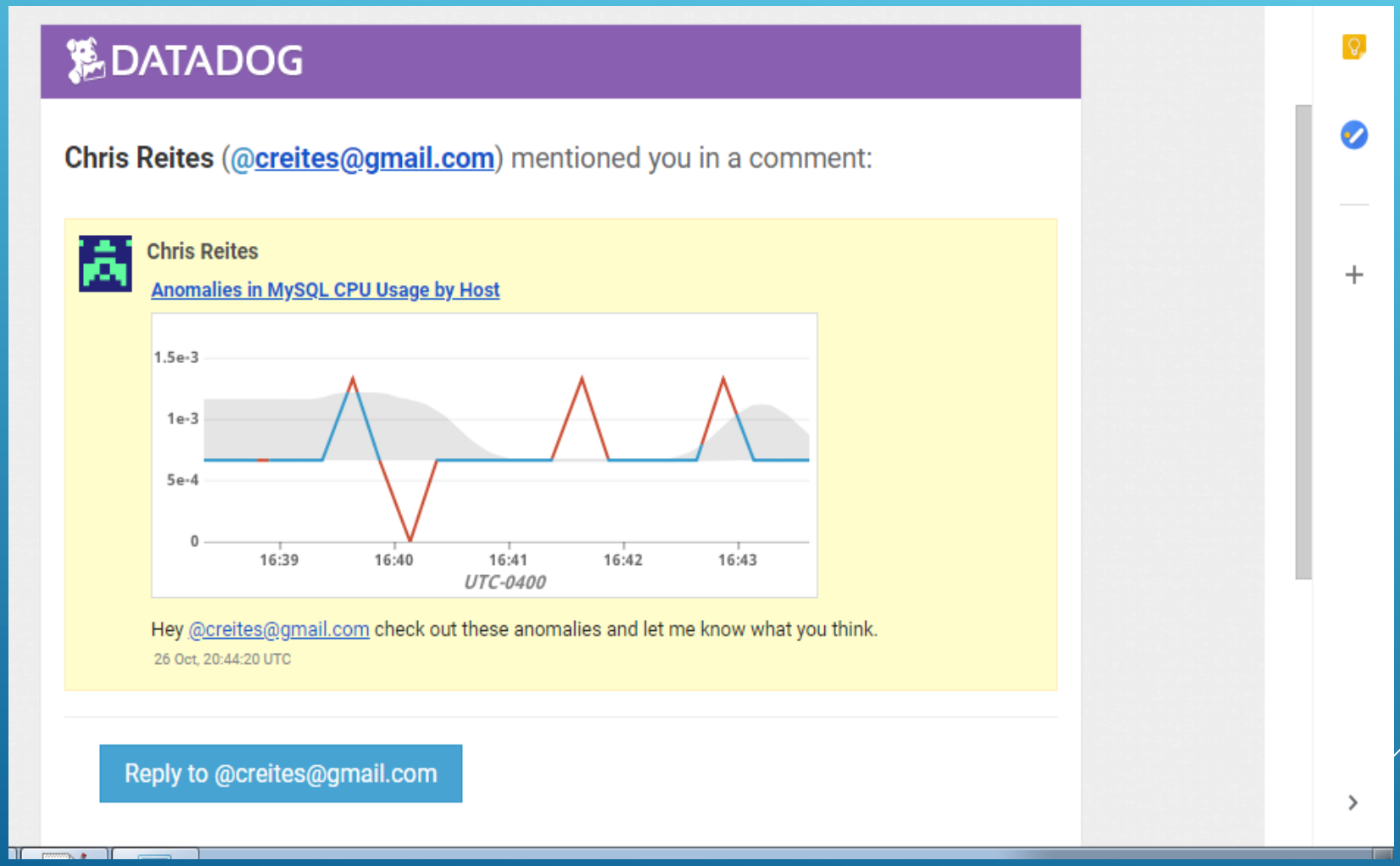
VISUALIZING DATA

The Datadog platform includes very powerful APIs that can be used to programmatically create many of the same assets you can build manually inside the web frontend. One example is Timeboards. Here is a python script I wrote to create a custom Timeboard.





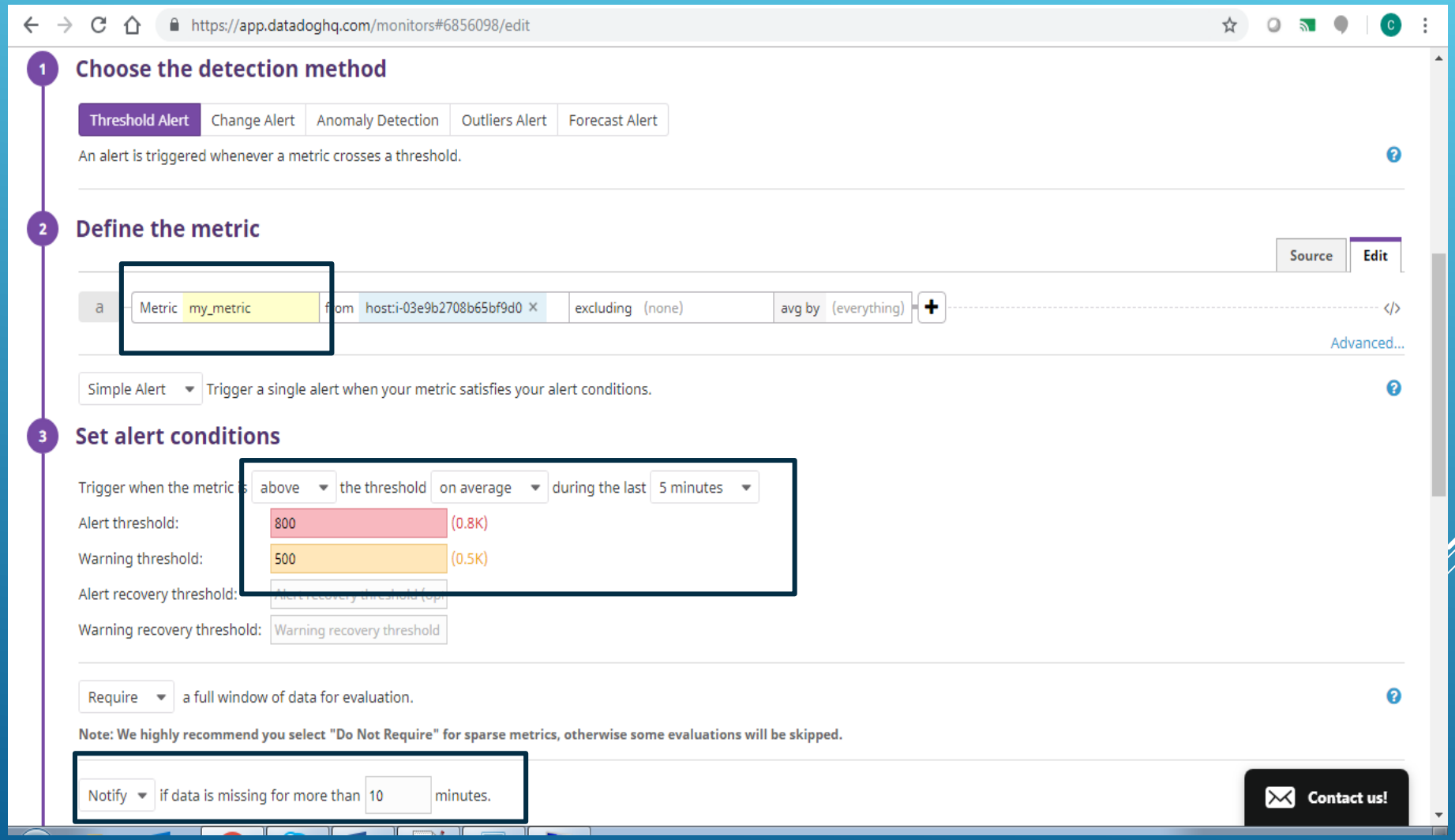
Here's an email containing a snapshot that I generated from my custom Timeboard. In this case, I'm looking at a 5 minute window of my graph that shows Anomalies in the CPU usage of MySQL. The Anomalies function is powerful because it compares actual metrics to expected values derived from historical trend analysis.



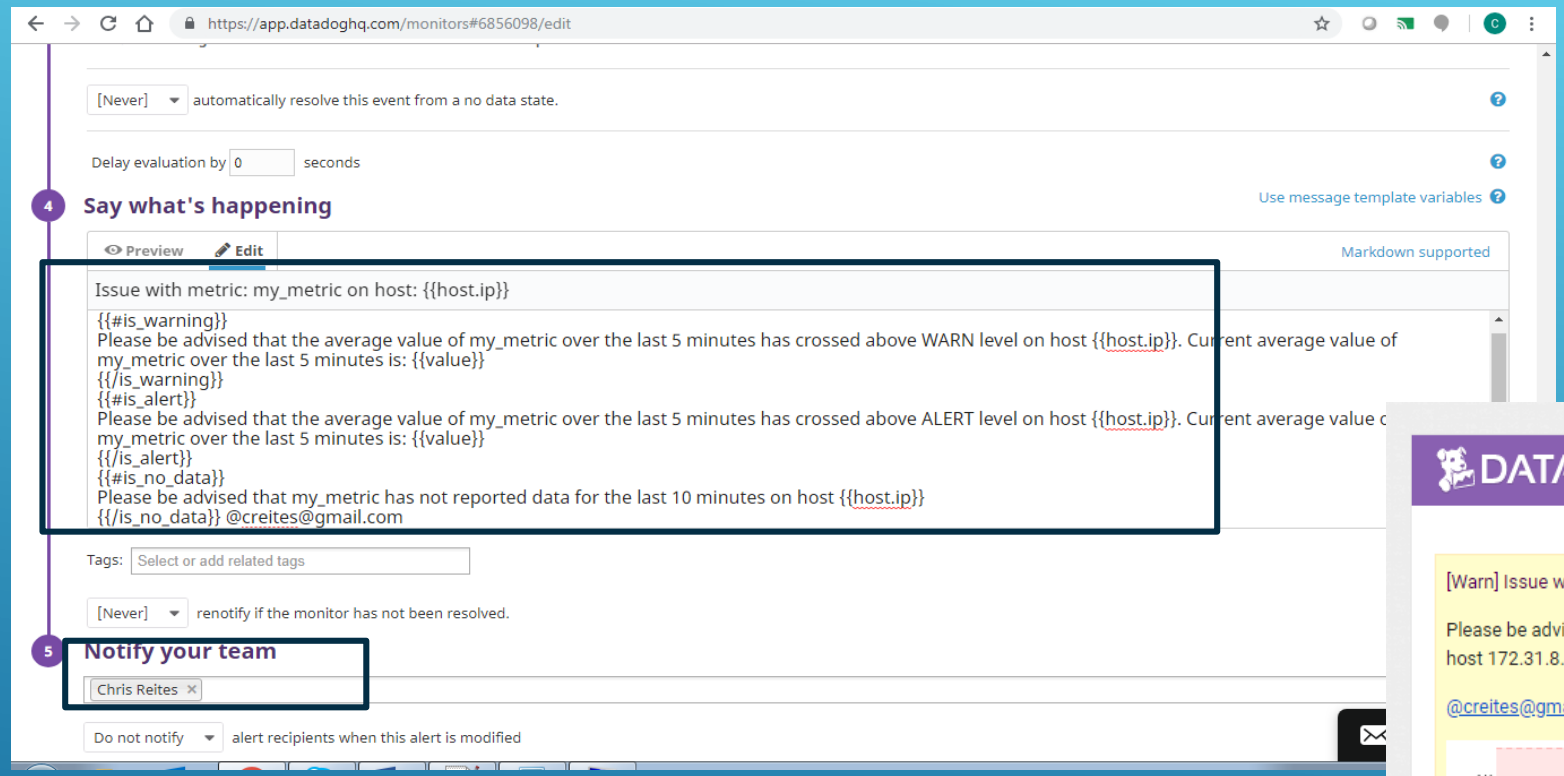
Section 3

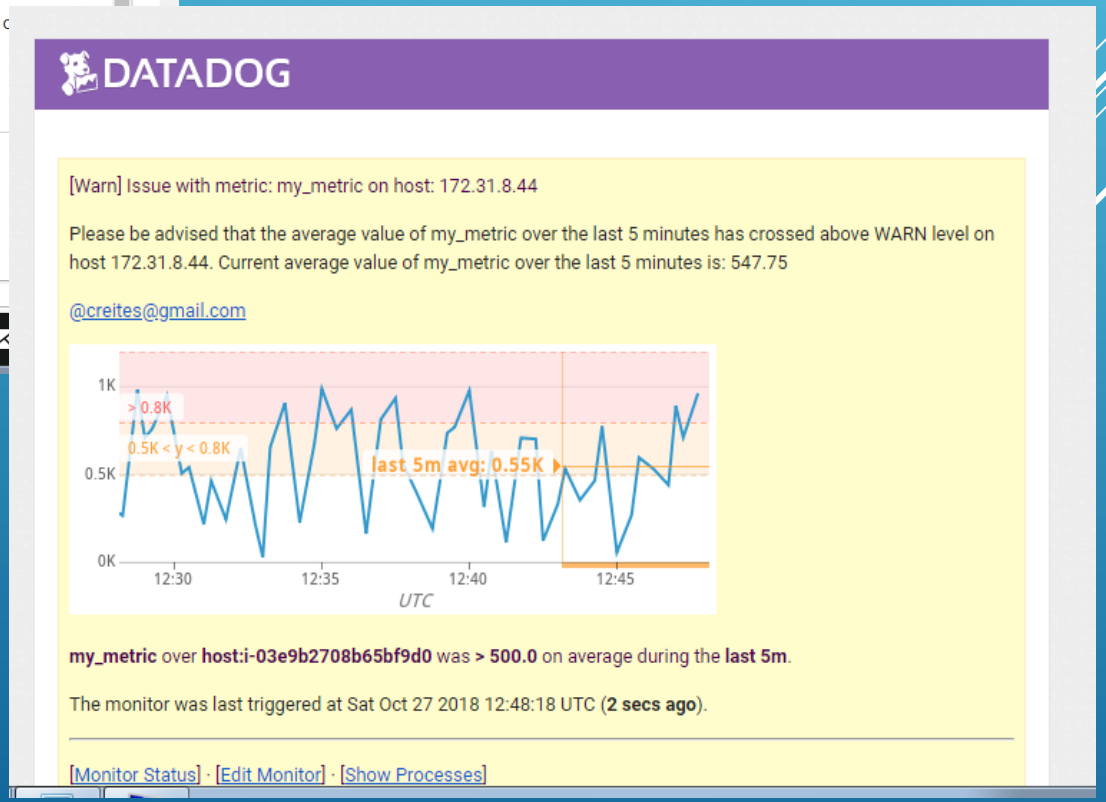
MONITORING DATA

One of the most useful aspects of the Datadog platform is the ability to create monitors for anything you want to be notified about. In this example, I created a monitor to alert me with either a warning or an alert if my custom random number metric rose above an average of either 500 or 800 over a 5 minute period. I'll also be notified if my metric doesn't send in any data over the last 10 minutes.

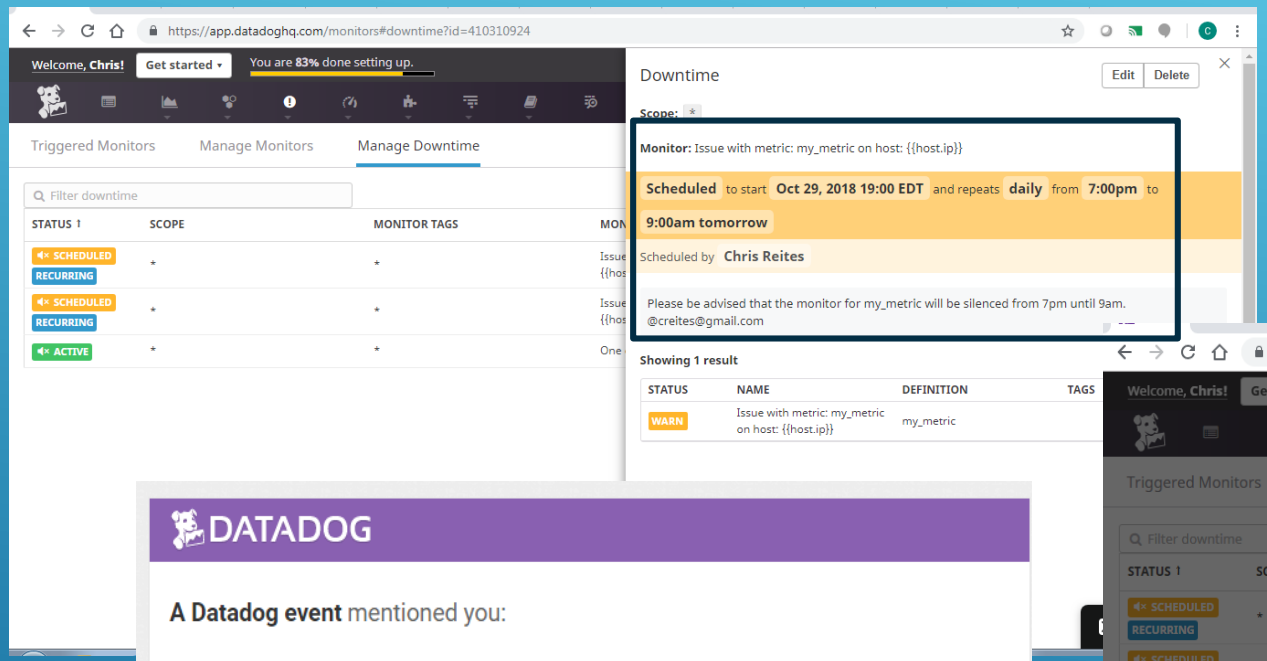


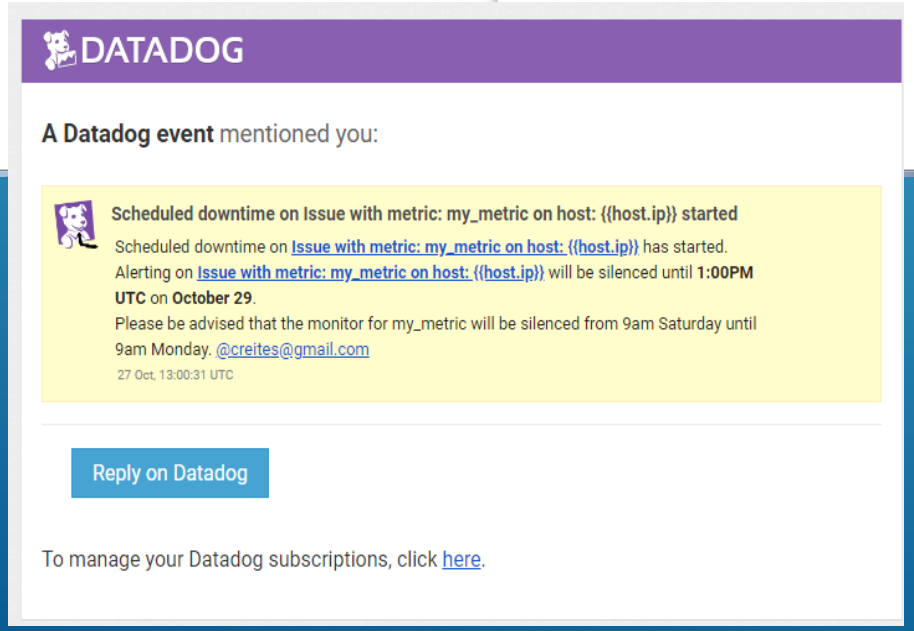
When setting up monitors, you can create highly customized messages that vary depending on things such as warnings vs alerts. You can also include variables such as the host IP address and metric values to make the monitor emails more meaningful.

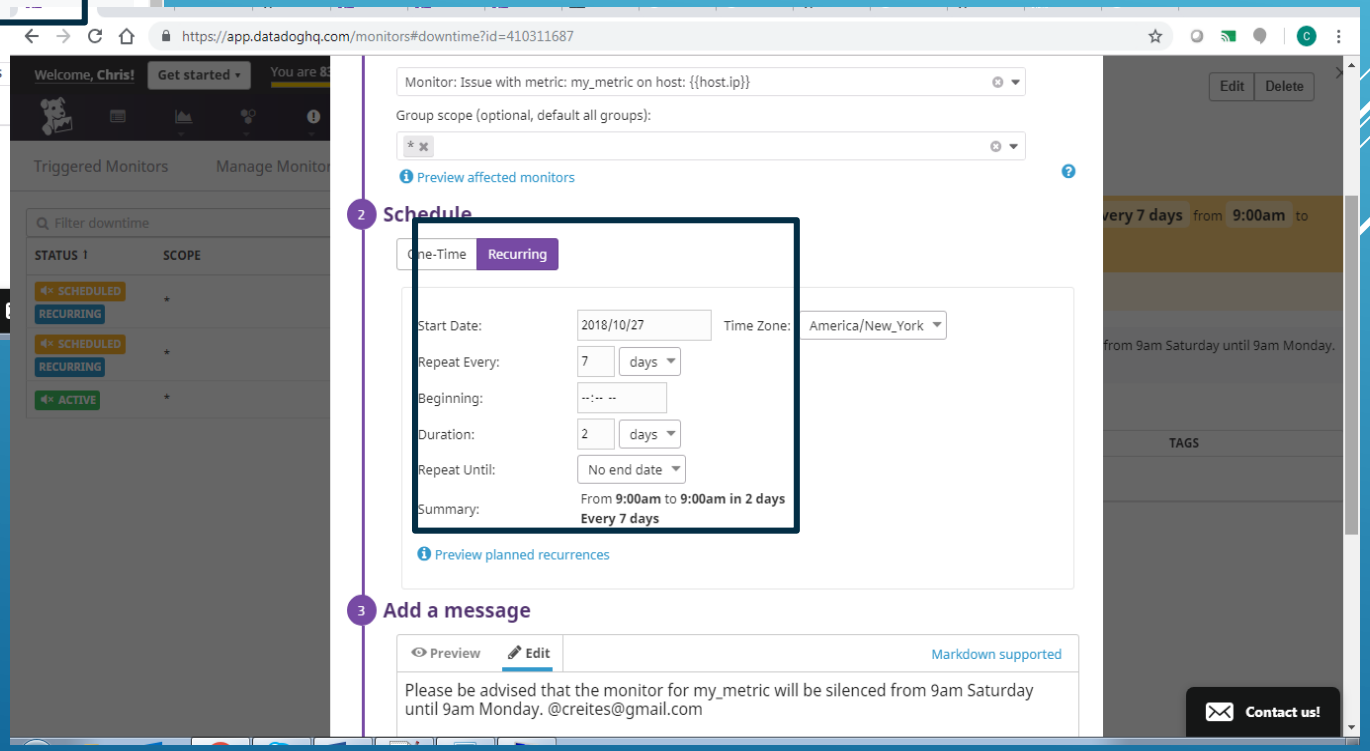




And because I don't want to be bothered while watching football on Saturdays or Sundays, it's always great to be able to schedule downtime windows when the monitors will automagically mute themselves. Here are two that when combined will mute the monitor on weekday evenings and also Saturday starting at 9am through Monday at 9am. We can also make sure the system sends us an email when it mutes a monitor.



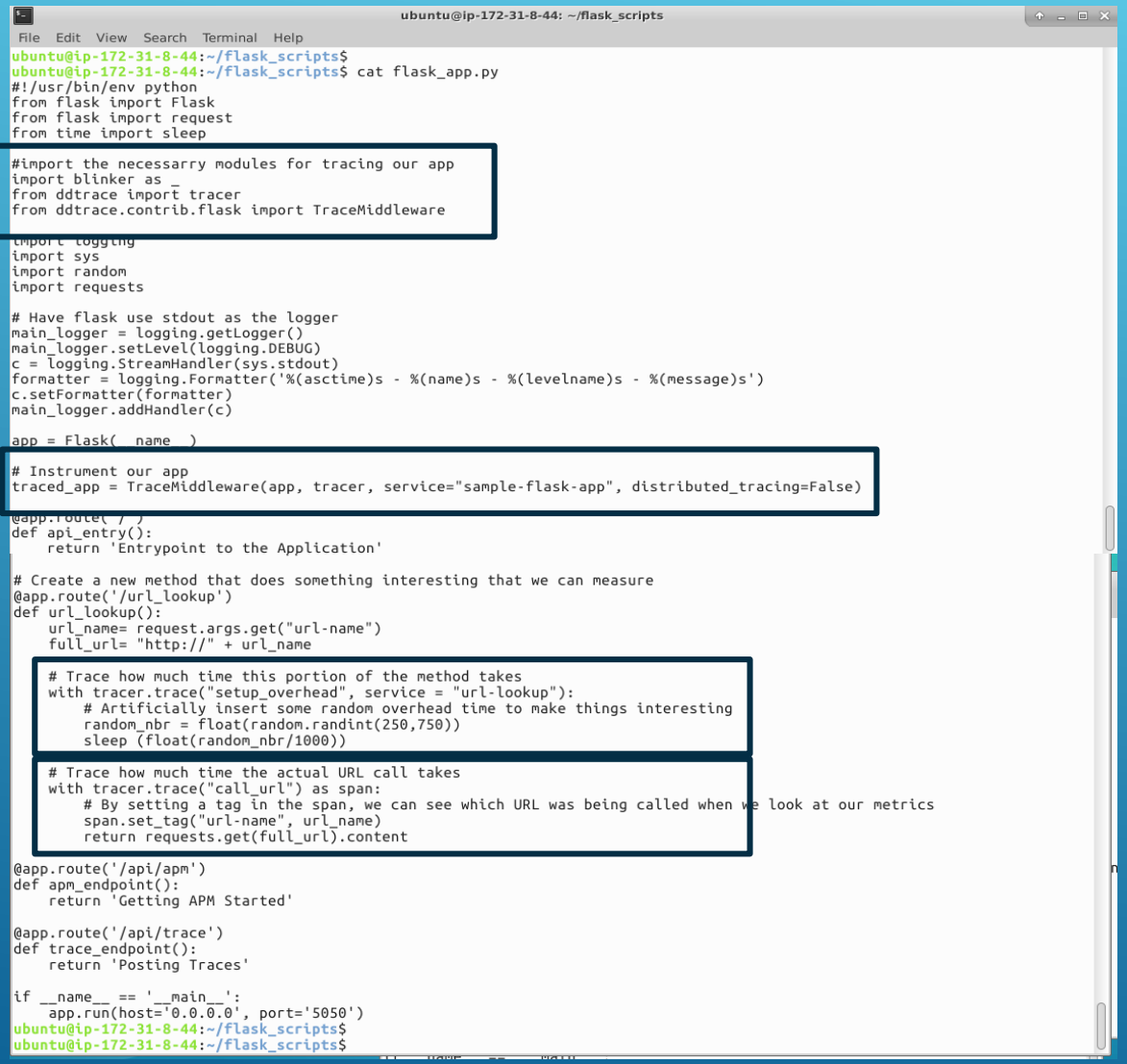




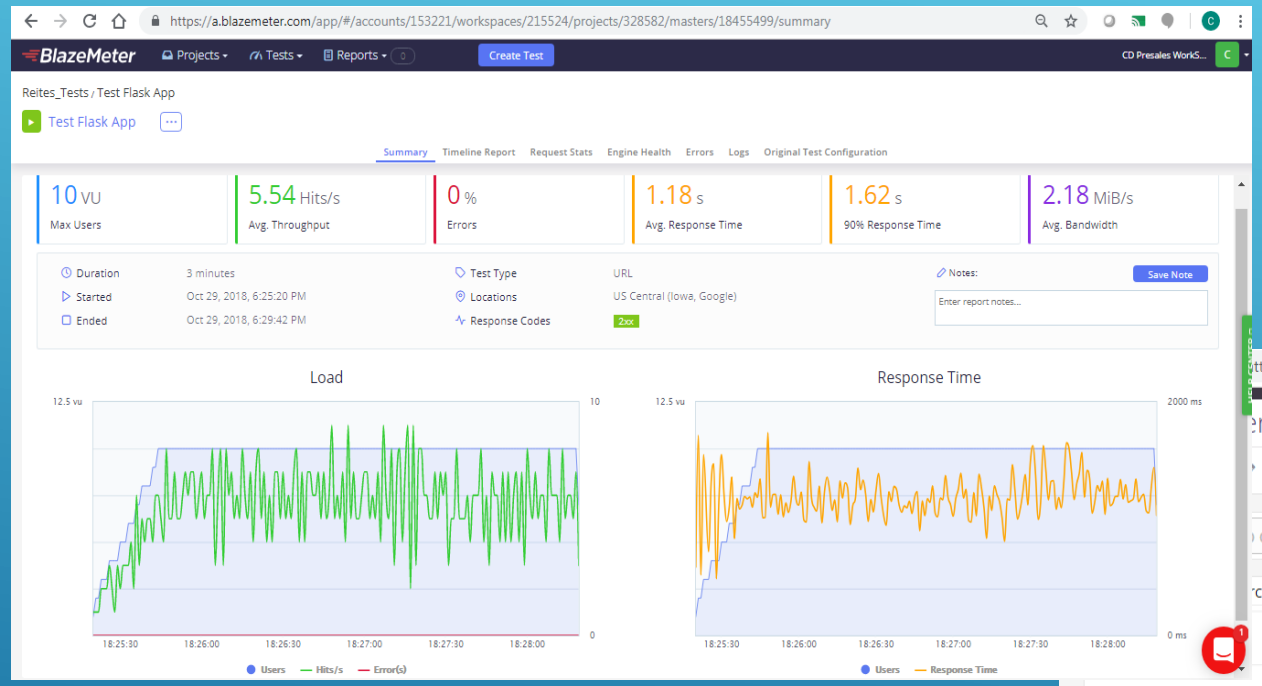
Section 4

COLLECTING APM DATA

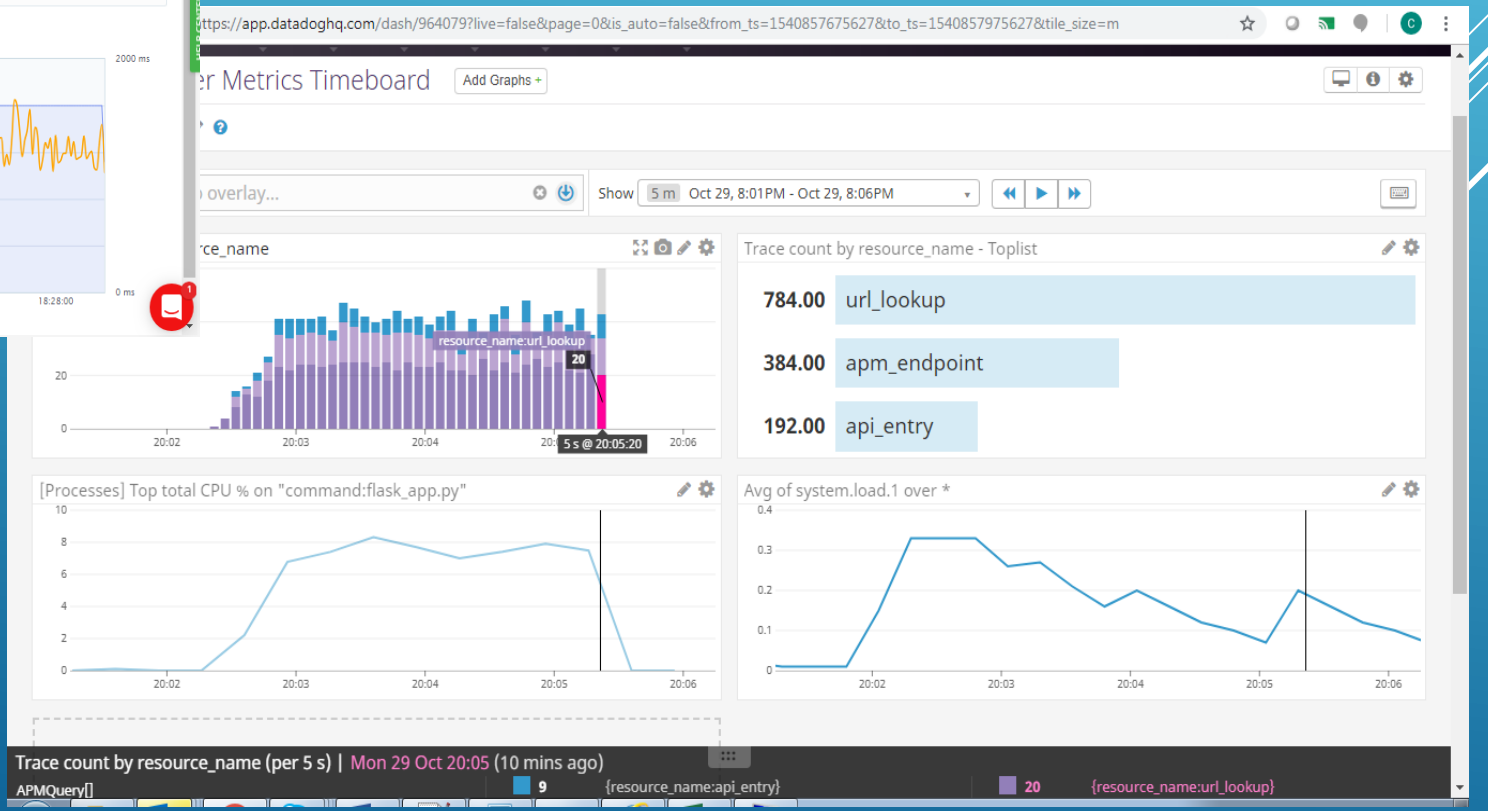
Using the Datadog platform, it's very easy to instrument applications for performance monitoring, either by running the process with ddtrace-run or by explicitly instrumenting your application. In the below example, I've instrumented a Flask application to feed metrics related to the performance of several routes.



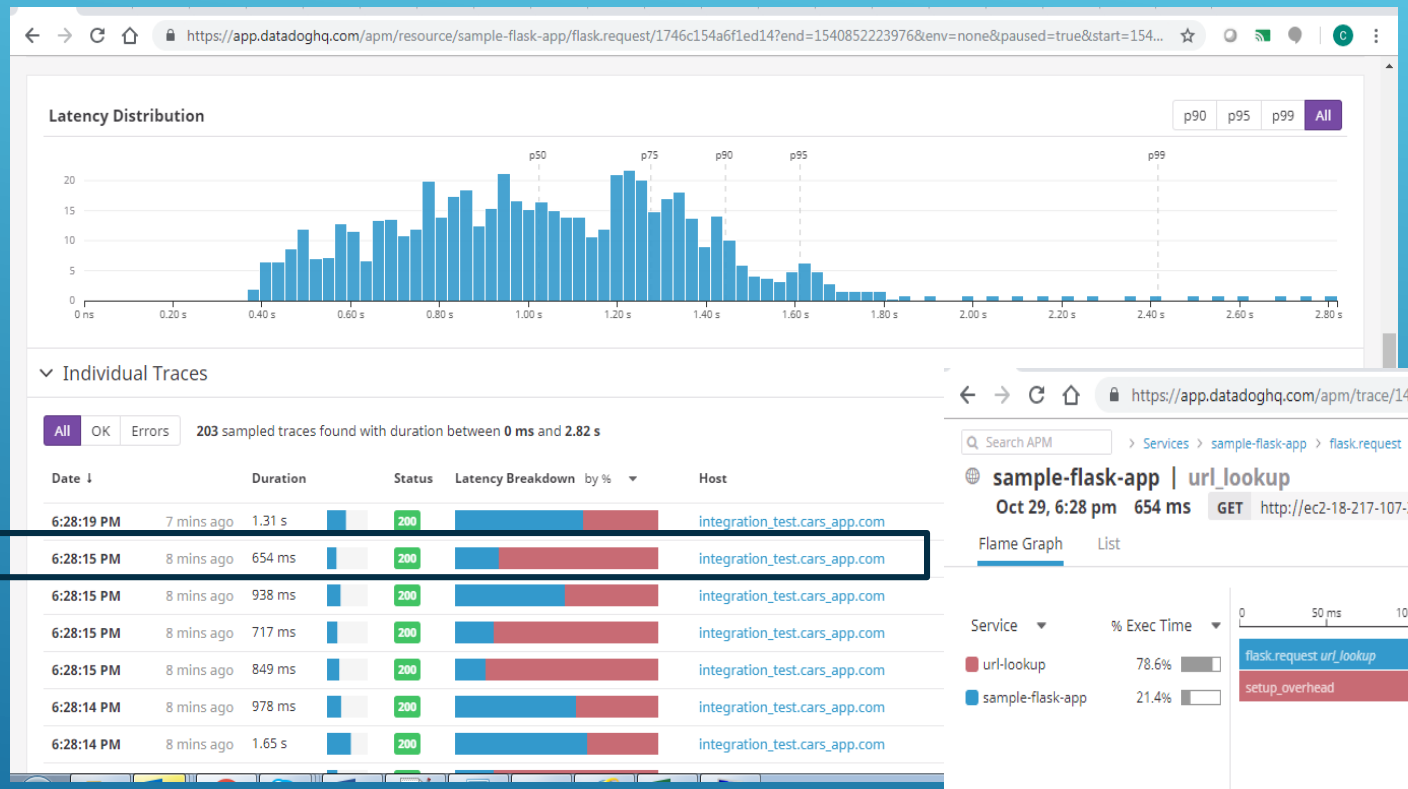
After I instrumented my application, I then used BlazeMeter, a cloud based load generation tool, in order to send thousands of requests to my application. The Timeboard below shows the counts by resource name, along with metrics about the CPU usage of the python app, and overall CPU utilization.



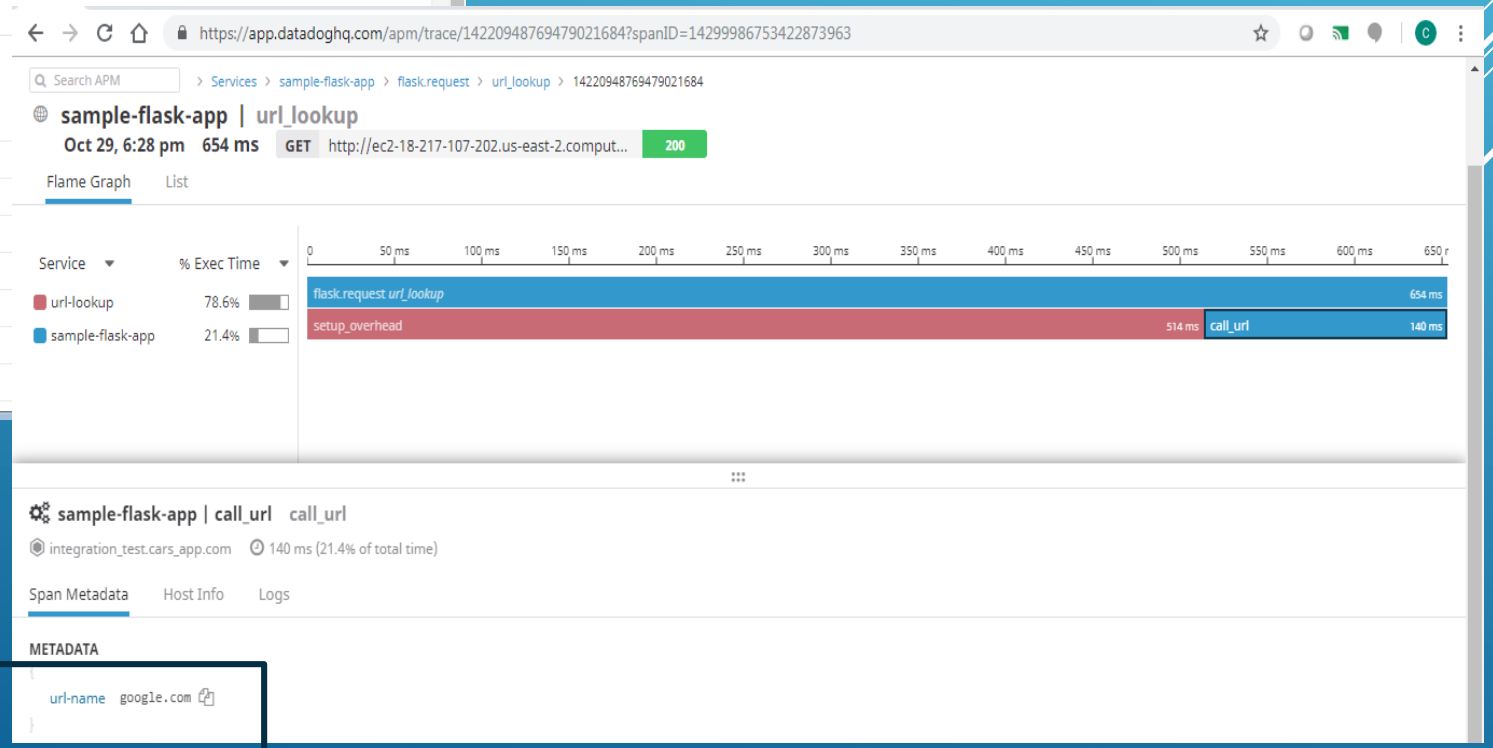
https://app.datadoghq.com/dash/964079?live=false&page=0&is_auto=false&from_ts=1540857675627&to_ts=1540857975627&tile_size=m&fullscreen=false



If we dive in a little deeper into our APM metrics, we can analyze latency as well. Because we setup a span with a tag name that matches the URL being loaded, we can also drill into individual traces and see the breakdown of time spent, along with the span data which shows the URL our app was trying to load on this particular call.




```
# Trace how much time the actual URL call takes
with tracer.trace("call_url") as span:
    # By setting a tag in the span, we can see which URL
    span.set_tag("url-name", url_name)
    return requests.get(full_url).content
```



Helpful Tip: What's the difference between a Service and a Resource?
A **resource** is one particular function which is typically part of a larger collection of other resources. Examples would be route functions of a web application.
A **service** is a collection of processes which together provides a set of functionality.

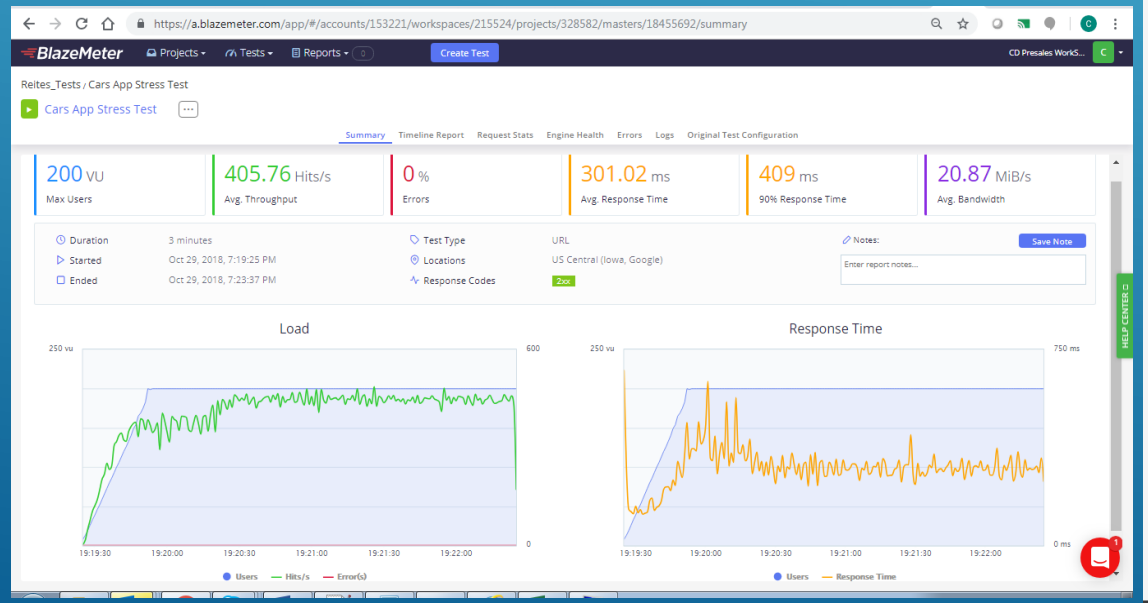
Section 5

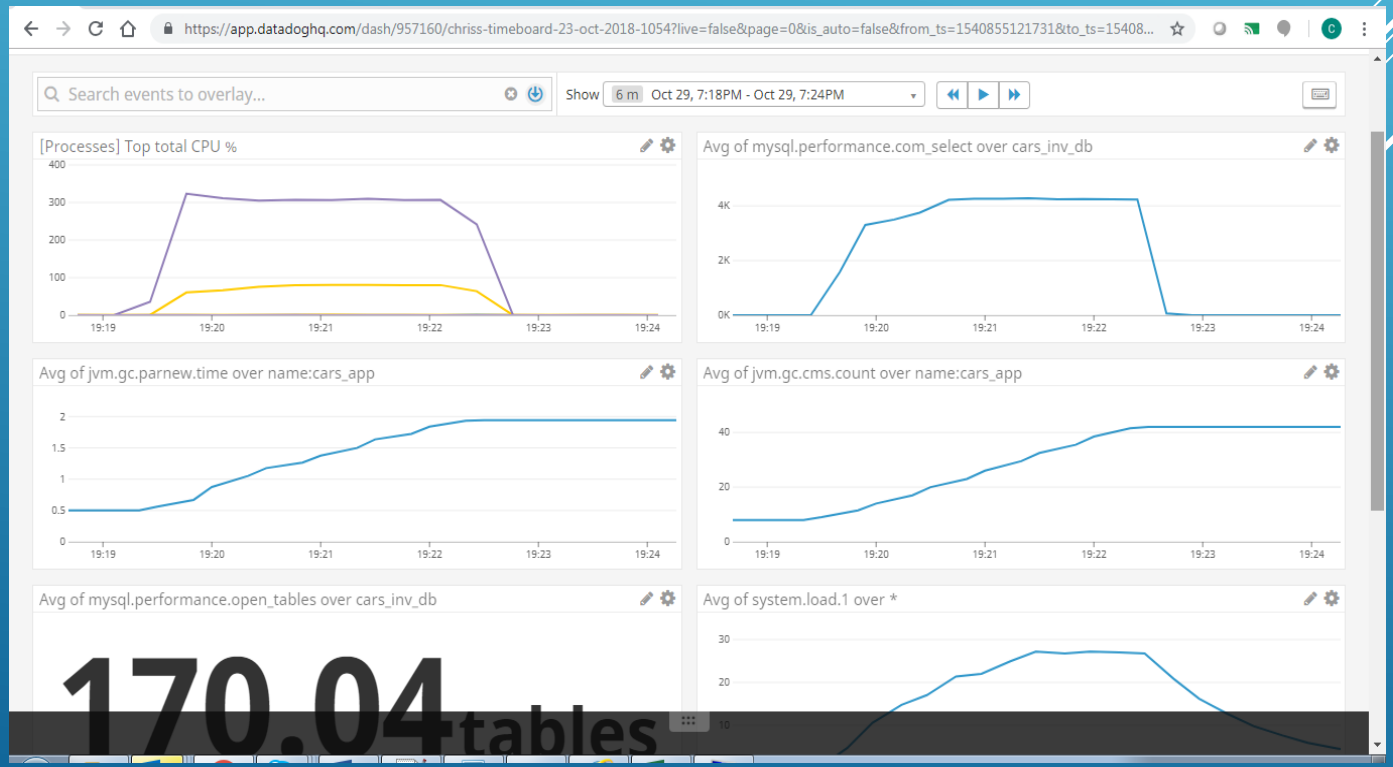
BUT WAIT… THERE'S MORE!!!

Prior to beginning this exercise, I had signed up for a free account on Datadog and was exploring functionality on my own. Because I had java applications, I installed the Java integration and ran some tests. The first thing I had to do was setup my application to accept JMX connections. I then had to configure the conf.yaml for the jmx integration to know how to connect. After that it was time to run some tests using BlazeMeter. I was most interested in two aspects of my application: The Garbage Collection metrics as well as the backend database performance.

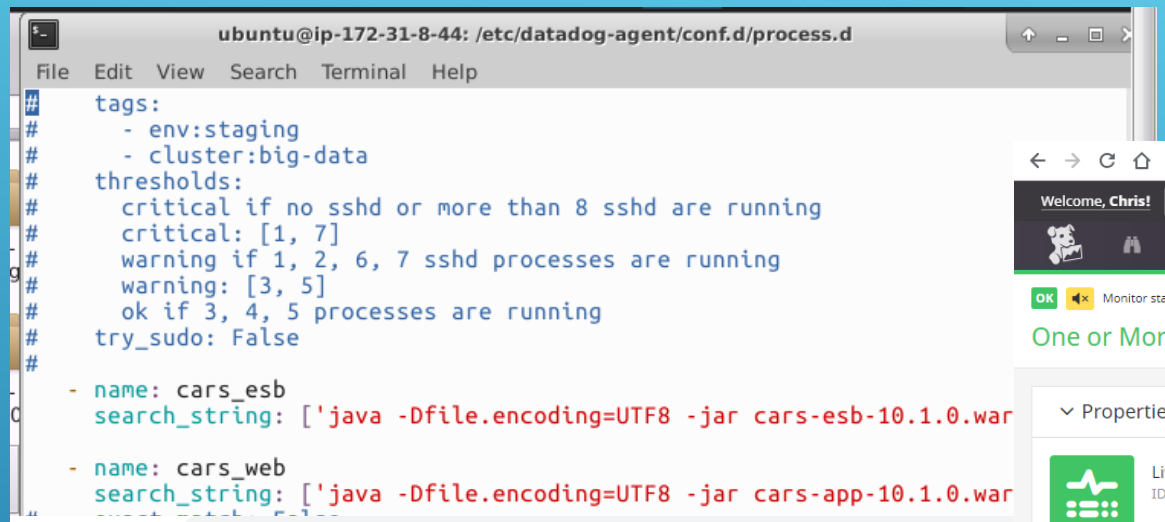
```
    if [ "$WEB" = "true" ]; then
        echo "Starting WEB"
        java -Dfile.encoding=UTF8 -Dcom.sun.management.jmxremote.port=7199 -Dcom.sun.management.jmxremote.ssl=false $WEB_AG
ENT -jar cars-app-10.1.0.war &
```

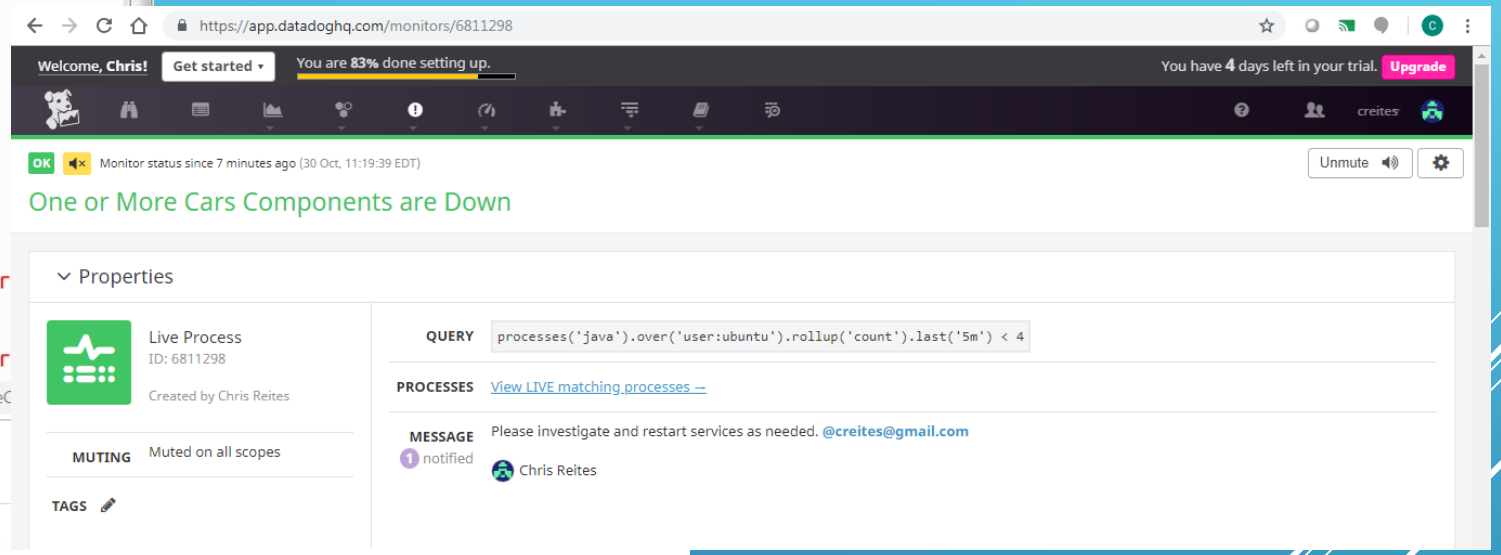
```
instances:
- host: localhost
  port: 7199
  user: monitorRole
  password: DD_READONLY_JMX
```

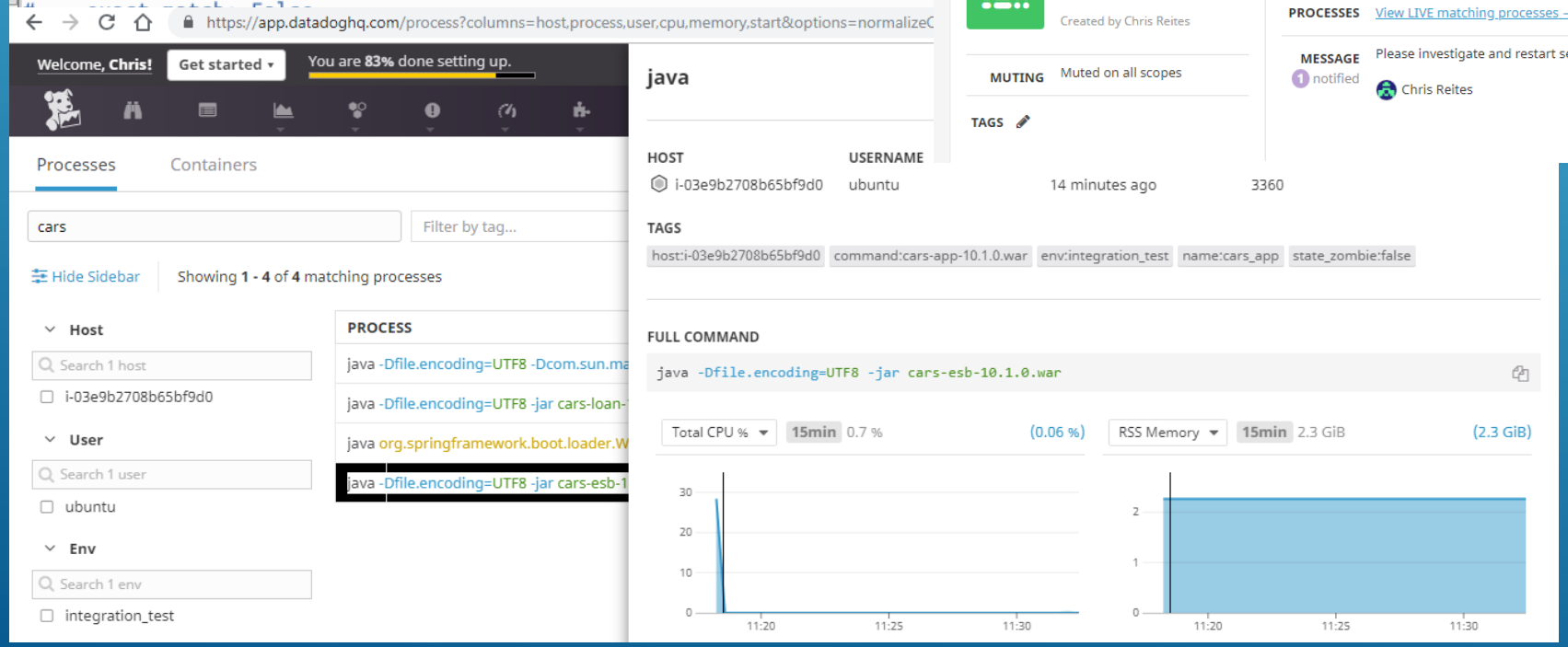




For this same application, I set up a monitor to make sure all 4 components of the application were always up and running. This also allowed me to play around with enabling the process check functionality.







Section 6

FINAL QUESTION

IMPROVING THE EFFICIENCY OF A BIKE SHARING BUSINESS WITH DATADOG

After having used several of the features that make up the Datadog platform, I began thinking of a fun use case that Datadog could bring value to. Having lived a big portion of my life in Central Florida (Mickey Mouse Town), a lot of the things I thought about were theme park related (ride wait times, parking, trend analysis of concession sales, etc…) but then while riding my bike to lunch today, I thought of a great use case having to do with bike shares.

Bike share programs are a great solution to commuting woes in large cities, however they're often plagued by a common problem. As users rent bikes and transport them to wherever they're headed, you end up with too many bikes in some places, and not enough bikes where you need them.

By utilizing the Datadog platform, you could keep track of how many bikes were currently in each location, and based on monitoring criteria, you could notify the appropriate "bike relocation" staff to transport bikes from areas with low demand to areas with high demand. You could also use historical trend analysis to proactively move bikes into areas where you are forecasting upcoming higher demand.

THANK YOU

I really enjoyed getting to explore the Datadog platform over the past week. Datadog has a fantastic product from what I've seen. So many customers I talk with today are using several different tools from multiple vendors to accomplish what Datadog can do on a single cloud based platform. I would greatly value the chance to join the team there and help drive the Datadog business forward while enabling customers to solve the many challenges they have around infrastructure, application, and log monitoring. If you have any questions for me about this exercise, please feel free to reach out to me at creites@gmail.com.

Thank You,

Chris