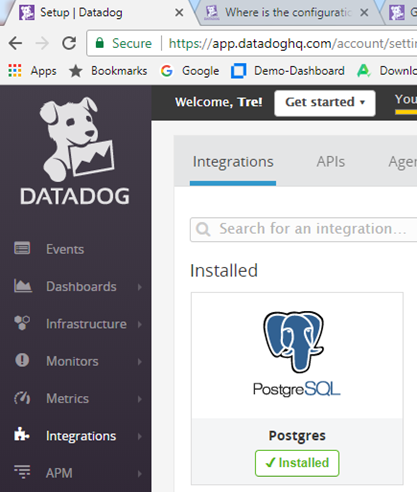
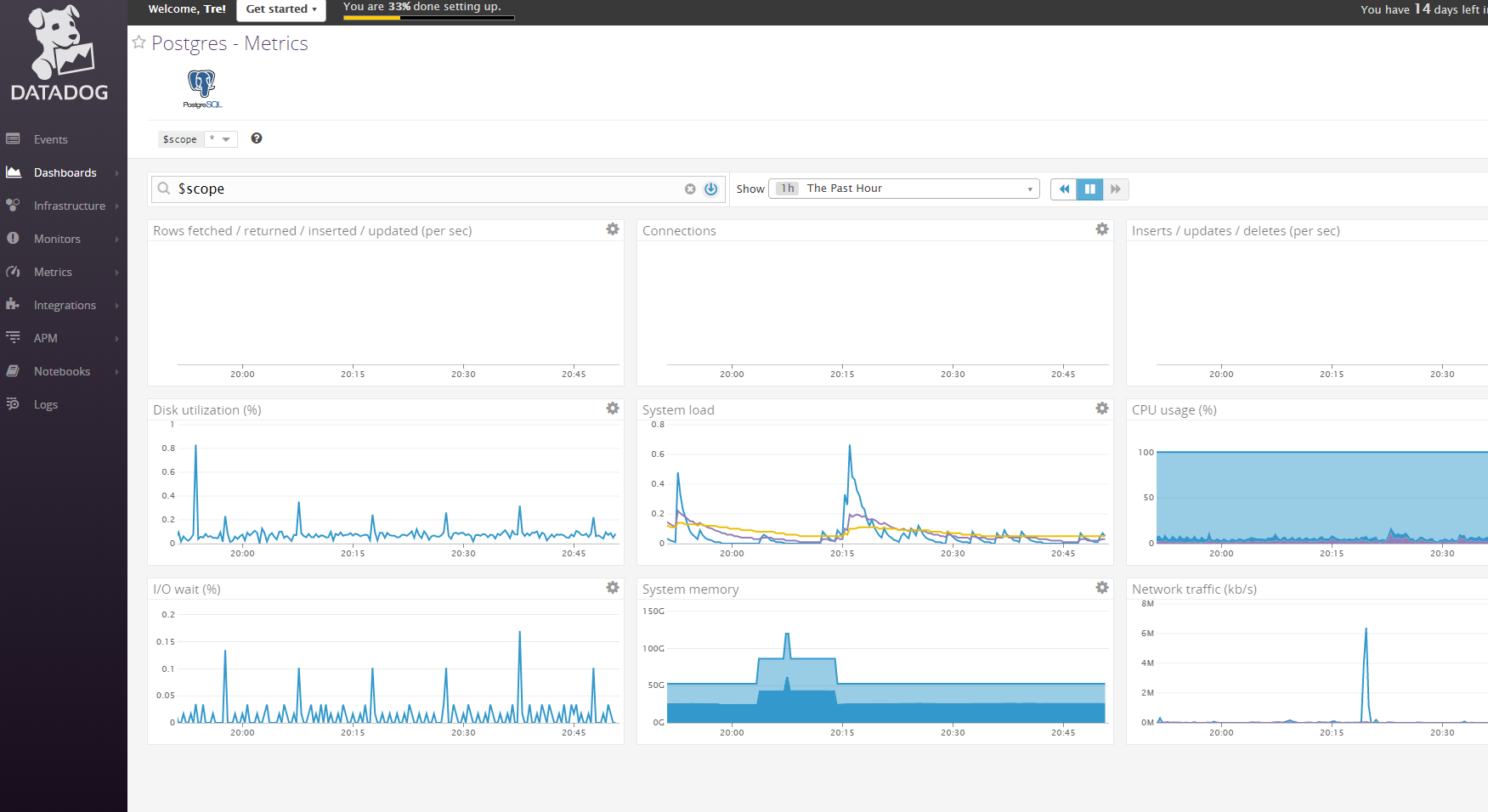
**Collection Metrics:**

**PostgresSql Integration**





**Custom Agent Check:**

Added a .yaml file in the conf.d directory called checkvalue.yaml

Added a .py file in the checks.d directory called checkvalue.py

\*Note both files must be named the same

Added the script to my .py file  
“from checks import AgentCheck

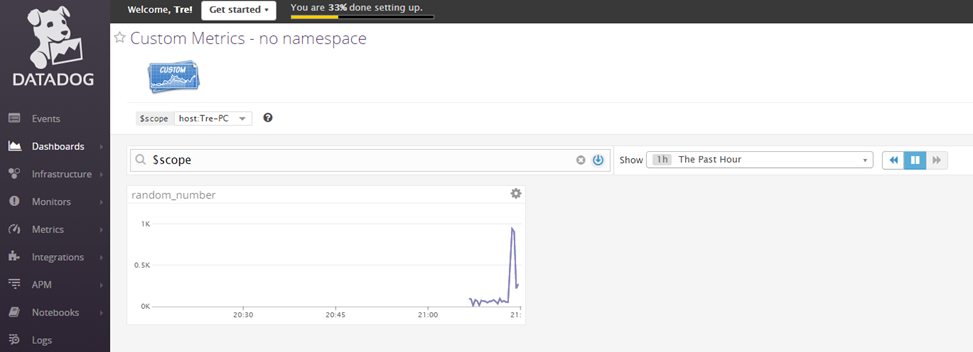
from random import \*

class RandomNumber(AgentCheck):

  def check(self, instance):

    self.gauge('random number', randint(0, 1000))”

Stopped/Started Agent

Can now see Custom Metrics data

**Changing collection interval and Bonus Question:**

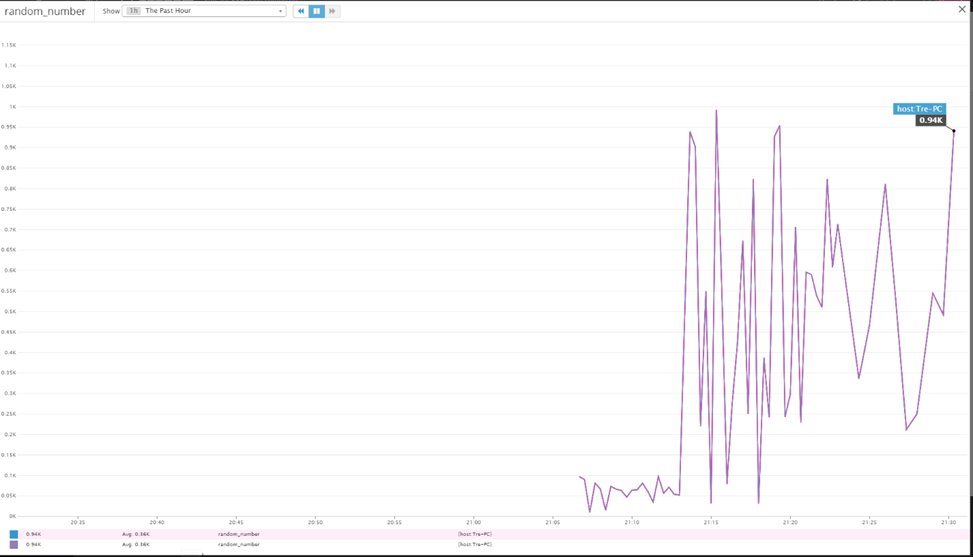
The bonus question asked if there was a way to change the collection interval without modifying the Python check file. Which according to the latest documentation, <https://docs.datadoghq.com/agent/agent_checks/>, you are supposed to change the yaml file by adding in the min\_collection\_interval under the instance.

If I was to modify the python file I would add this line:

time.sleep(30)

before my check call in the script, to add 30 seconds to the defaulted 15 seconds check time

You can see there is now ~45 seconds between data point of my customer metric



**Visualizing Data using API:**

See attached file “TimeboardAPI.txt”

TimeBoard with Snapshot and @ notation

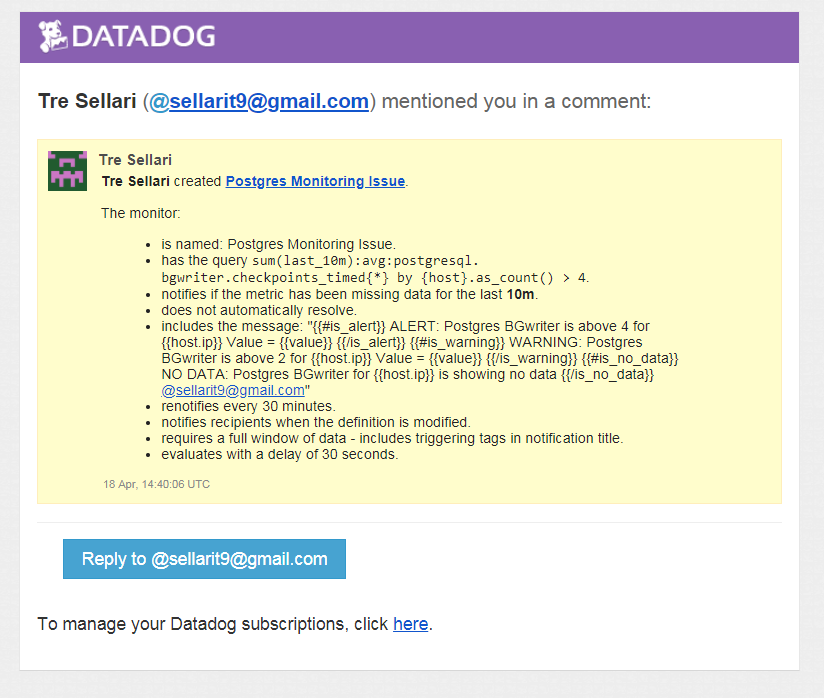


**Bonus Question: What is the Anomaly graphing displaying?**“If it ain’t broke, don’t fix it!” If you are only looking for what is broken, you will miss all other potential issues of your application. How do you know what broken is? Is it just when the application is down? What if it is running s….l…o…w… and causing issues for your customers? Anomaly detection will measure where values are in a normal range, and will alert you if they are suddenly outside the range.

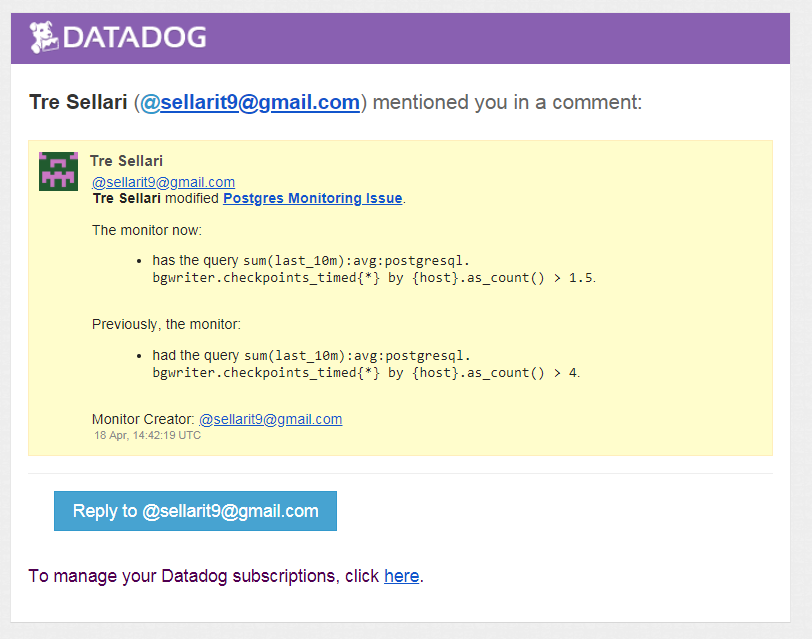
It is a lot like a golf swing, if you are always hitting your draw and then suddenly you hit a terrible slice, that is an anomaly. Your swing isn’t broken, because you still hit the golf bad, but it certainly wasn’t good. Now this is where analytics and metrics will come into play, maybe it was because your ball was below your feet, or maybe out of the sand, but you can determine the potential cause. This is the same thing that DataDog can do with the Anomaly detection and all the other metrics gathered, show you the reason behind something out of the ordinary.

**Monitoring Data**

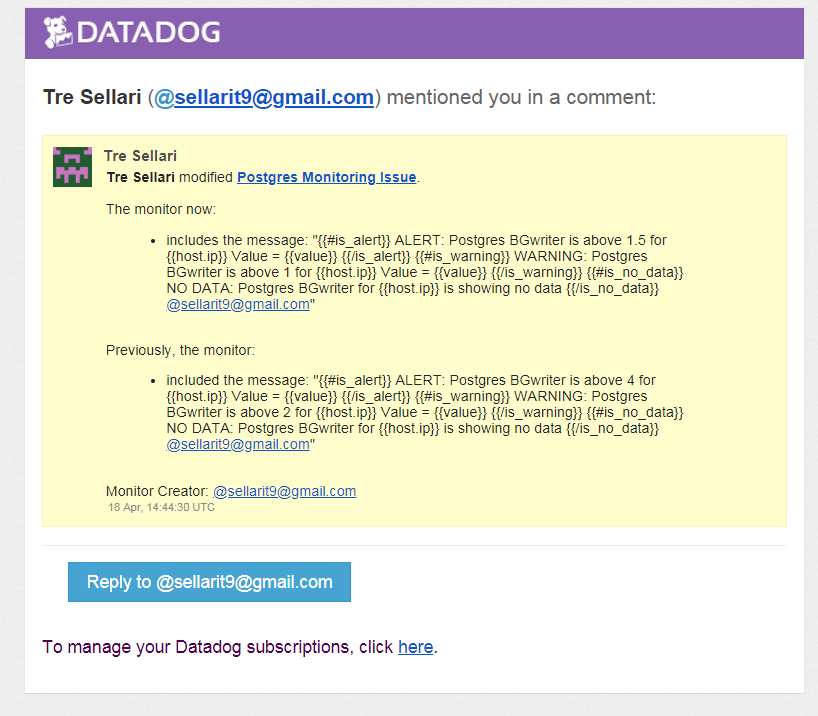
Created Monitor

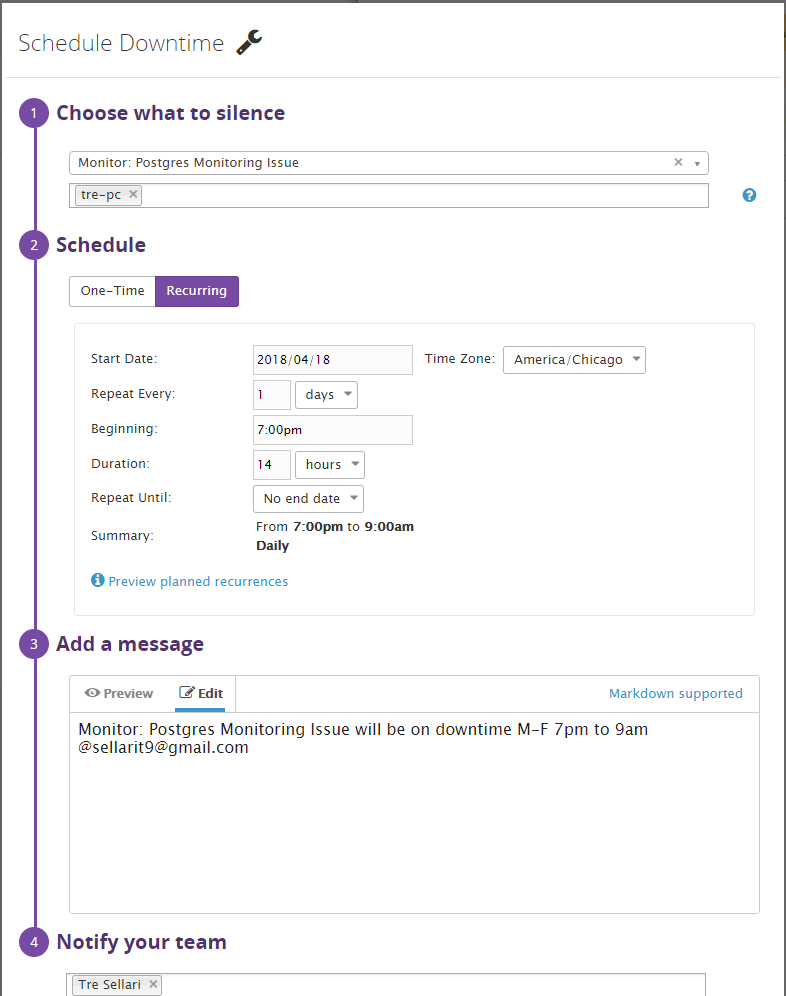


Modified Monitor

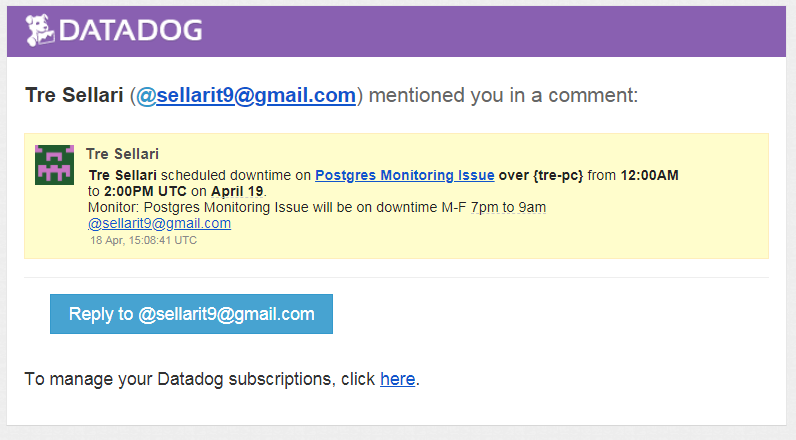


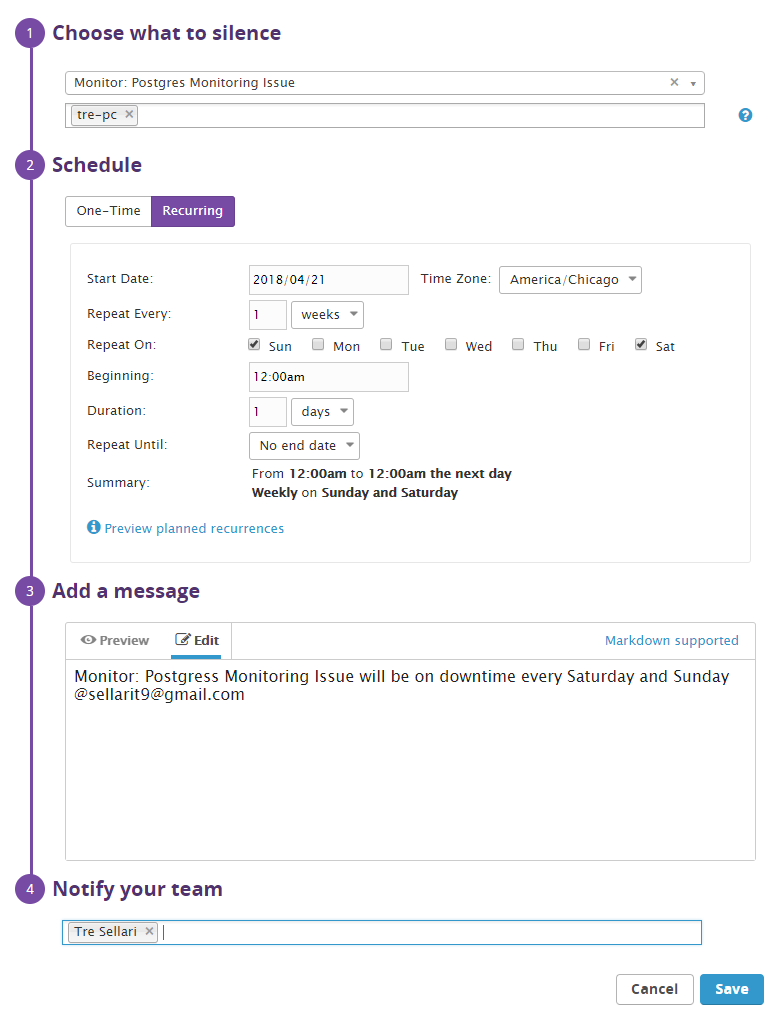
Email from Monitor:

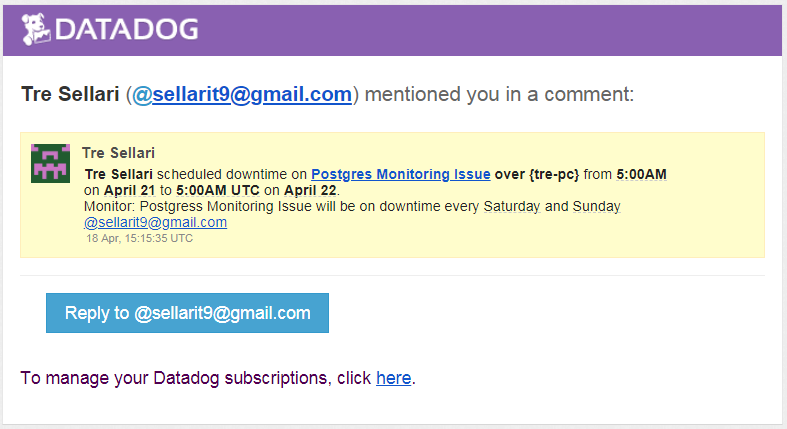


Schedule downtime for M-F 7pm – 9am:

Email for Downtime:



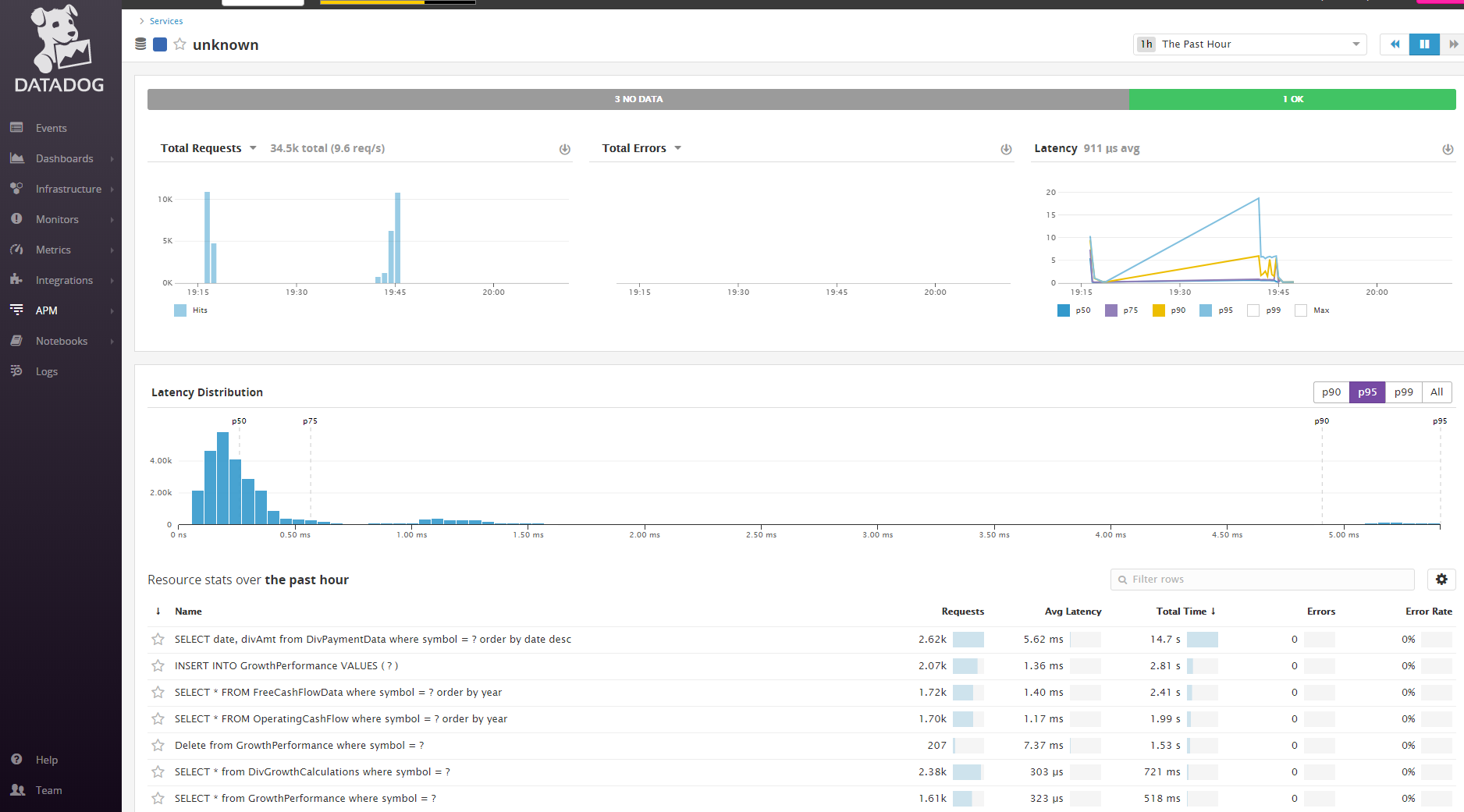
Downtime for Sat-Sun:

Email for downtime:

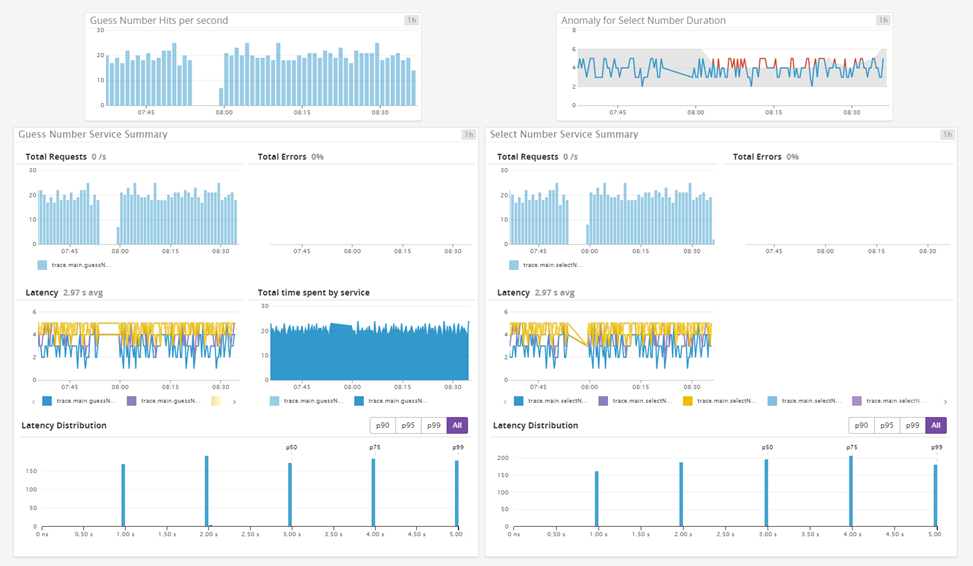
**Collecting APM Data:**

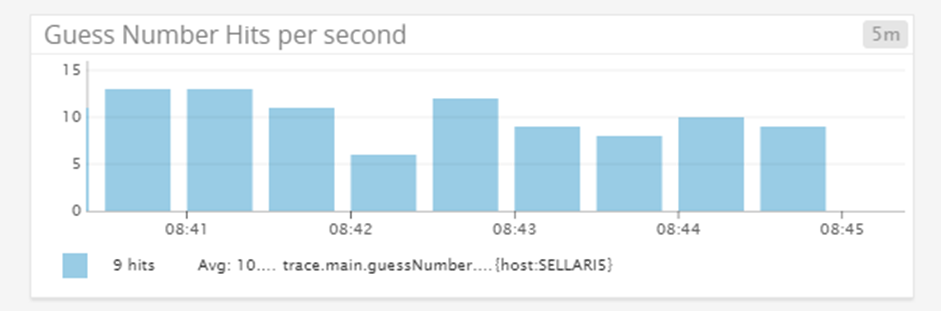
Link - <https://p.datadoghq.com/sb/bd421a7ea-80a6e2bb6282b24c204cb9b24394dda4>

I instrumented two applications. One was a Java and Sql application I personally wrote for dividend stock information gathering and analyzing. I did not upload this application to share, but here is a screen show of the APM collection data.



I also wrote a quick Python application where the computer is guessing numbers. I wrote a lot of random “guessing” times to give it some metrics of different response times. See file “SellariRandomGame.py”





**Bonus Question: What is the difference between a Service and a Resource?**

* Resource is a single query or action to a service. For example a SQL query or a url call.
* Service is a set of processes that are working together to provide a feature set. For example a webapp, a master-db, and a replica-db. In DataDog if we define a “Service” for these three processes it will help us quickly analyze issues between different processes.

**Final Question: What would I monitor with Datadog?**

Chick-fil-a close to my house with data related to when fresh fries are made and the length of the line. I would also monitor traffic data from my house to the said Chick-fil-a. Combining and analyzing this data I would find the optimal time to leave my house, be in line, order, and get the freshest fries possible!