**he Exercise**

Don’t forget to read the [References](https://github.com/DataDog/hiring-engineers/blob/solutions-engineer/README.md#references)

**Questions**

Please provide screenshots and code snippets for all steps.

**Prerequisites - Setup the environment**

You can utilize any OS/host that you would like to complete this exercise. However, we recommend one of the following approaches:

* You can spin up a fresh linux VM via Vagrant or other tools so that you don’t run into any OS or dependency issues. [Here are instructions](https://github.com/DataDog/hiring-engineers/blob/solutions-engineer/README.md#vagrant) for setting up a Vagrant Ubuntu VM. We strongly recommend using minimum v. 16.04 to avoid dependency issues.
* You can utilize a Containerized approach with Docker for Linux and our dockerized Datadog Agent image.

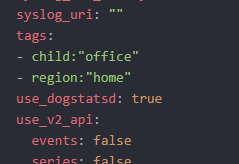
Then, sign up for Datadog (use “Datadog Recruiting Candidate” in the “Company” field), get the Agent reporting metrics from your local machine.

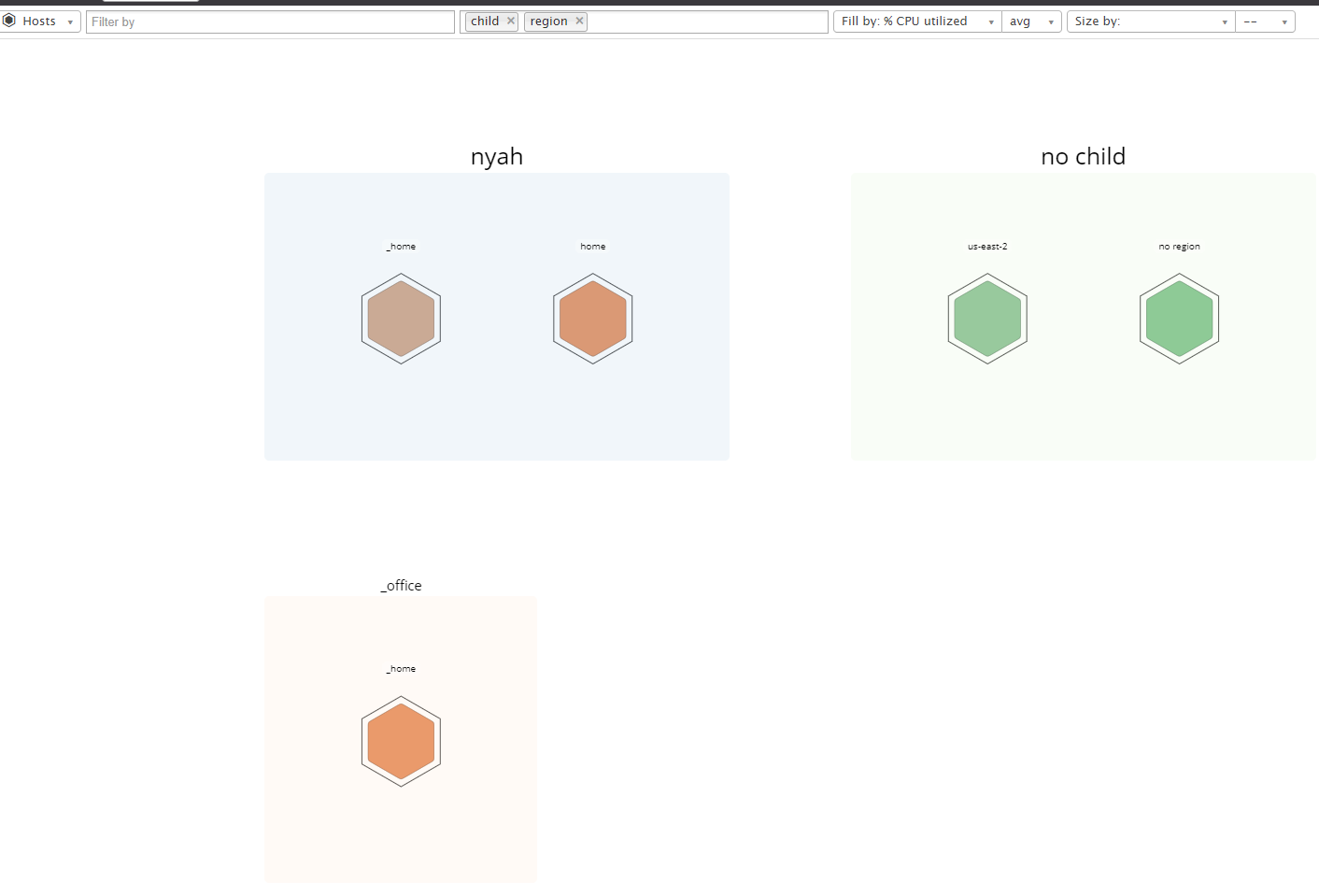
***I set this up on Windows 7, 2 MAC laptops, Linux RedHat and also AWS. All reporting metrics.***

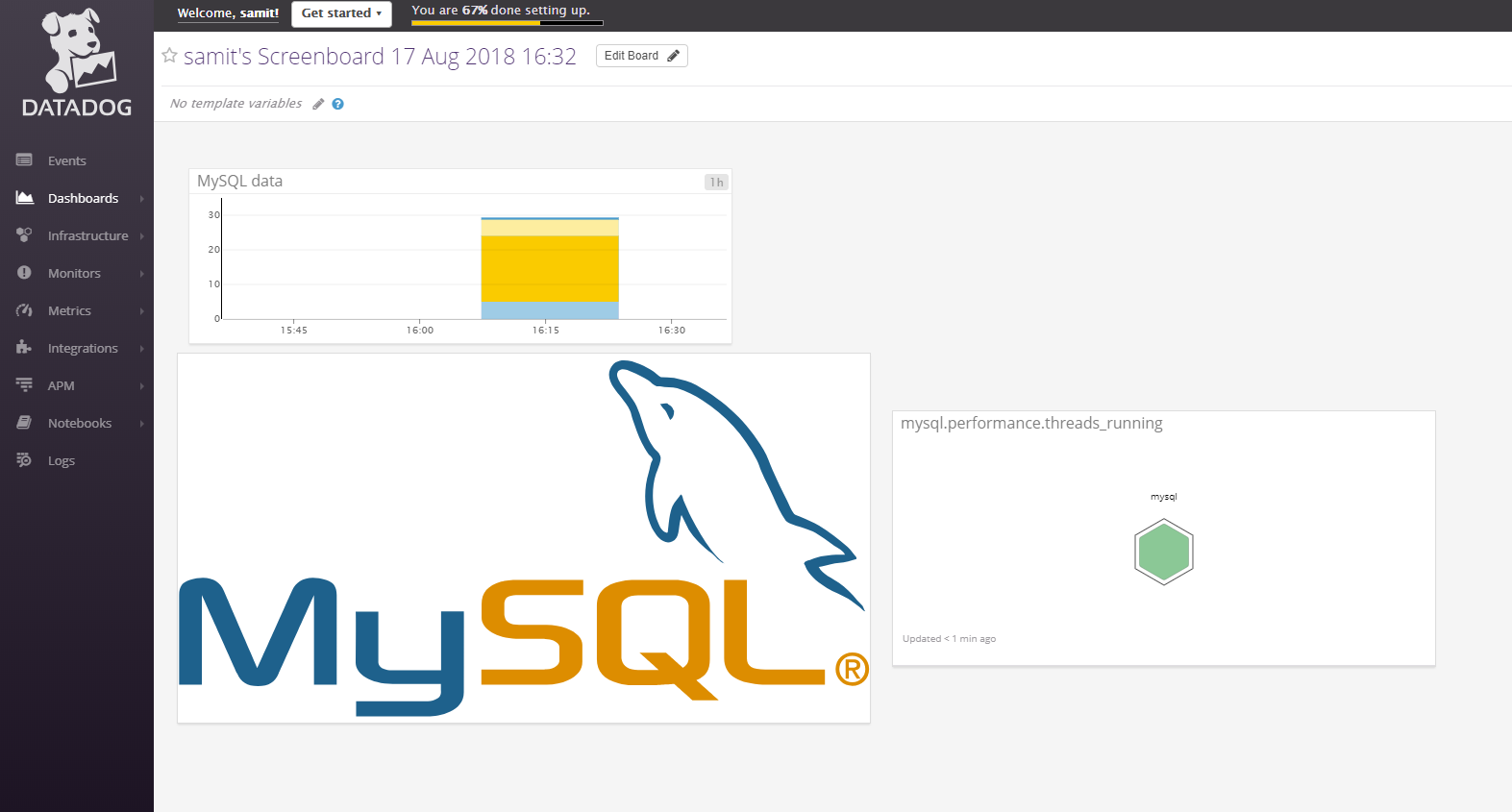
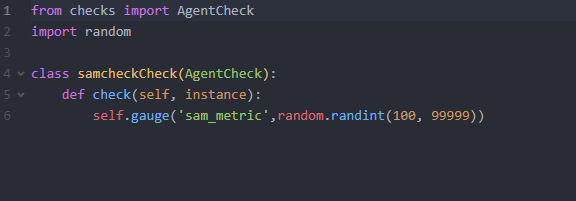
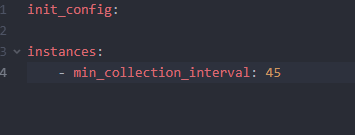
**Collecting Metrics:**

* Add tags in the Agent config file and show us a screenshot of your host and its tags on the Host Map page in Datadog.
  + ***I added various tags to the different systems.***

***For example, the Child tag shows the device/computer used by that child, if any. (Nyah is my eldest daughter)***

* + 

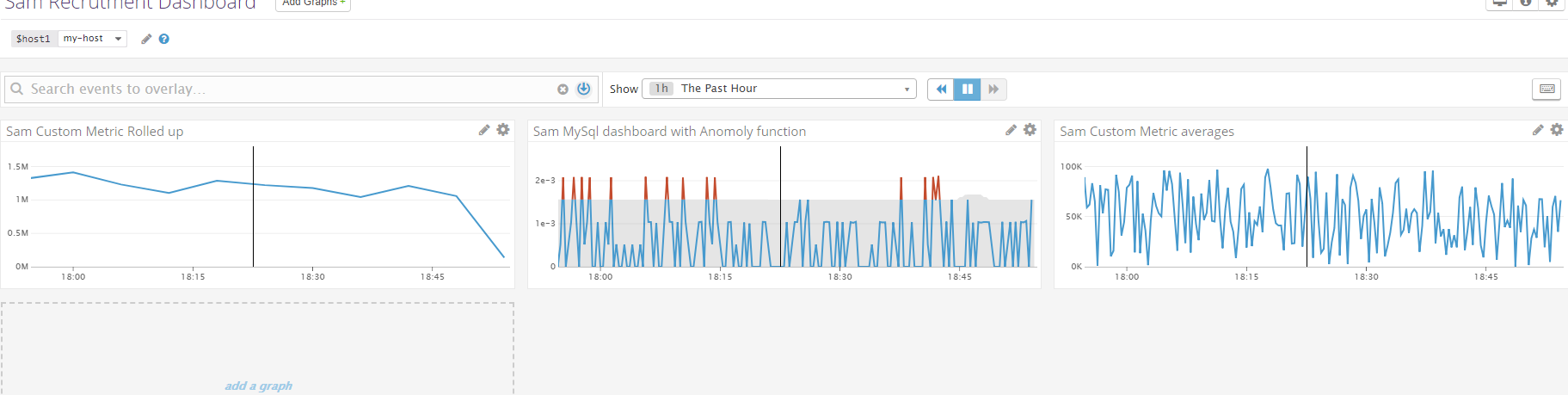


* Install a database on your machine (MongoDB, MySQL, or PostgreSQL) and then install the respective Datadog integration for that database.
  + ***Running on MySql***
  + 
* Create a custom Agent check that submits a metric named my\_metric with a random value between 0 and 1000.
  + ***Used a larger Random range than default***
  + 
* Change your check's collection interval so that it only submits the metric once every 45 seconds.
  + 
* **Bonus Question** Can you change the collection interval without modifying the Python check file you created?
  + ***This is done in the check’s yaml file, nowhere else as far as I can see***

**Visualizing Data:**

Utilize the Datadog API to create a Timeboard that contains:

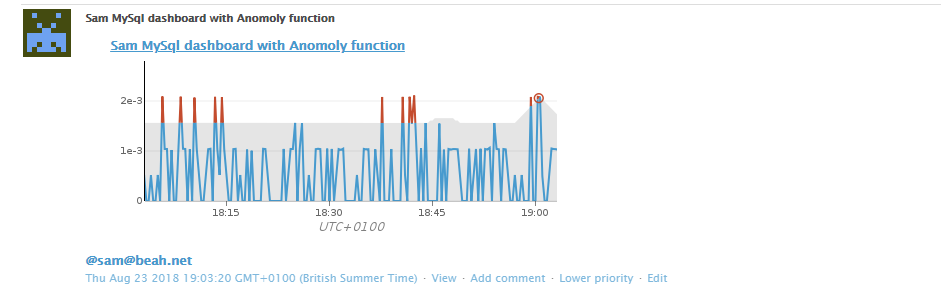
* Your custom metric scoped over your host.
* Any metric from the Integration on your Database with the anomaly function applied.
* Your custom metric with the rollup function applied to sum up all the points for the past hour into one bucket



Please be sure, when submitting your hiring challenge, to include the script that you've used to create this Timeboard.



Once this is created, access the Dashboard from your Dashboard List in the UI:

* Set the Timeboard's timeframe to the past 5 minutes
* Take a snapshot of this graph and use the @ notation to send it to yourself.
* 
* **Bonus Question**: What is the Anomaly graph displaying?
* ***I tried to set it as Agile but it says there was not enough data, so kept at Bsaic. It basically shows when data is outside of “Trend”. It points out those occasions when things are worse than expected.***

**Monitoring Data**

Since you’ve already caught your test metric going above 800 once, you don’t want to have to continually watch this dashboard to be alerted when it goes above 800 again. So let’s make life easier by creating a monitor.

Create a new Metric Monitor that watches the average of your custom metric (my\_metric) and will alert if it’s above the following values over the past 5 minutes:

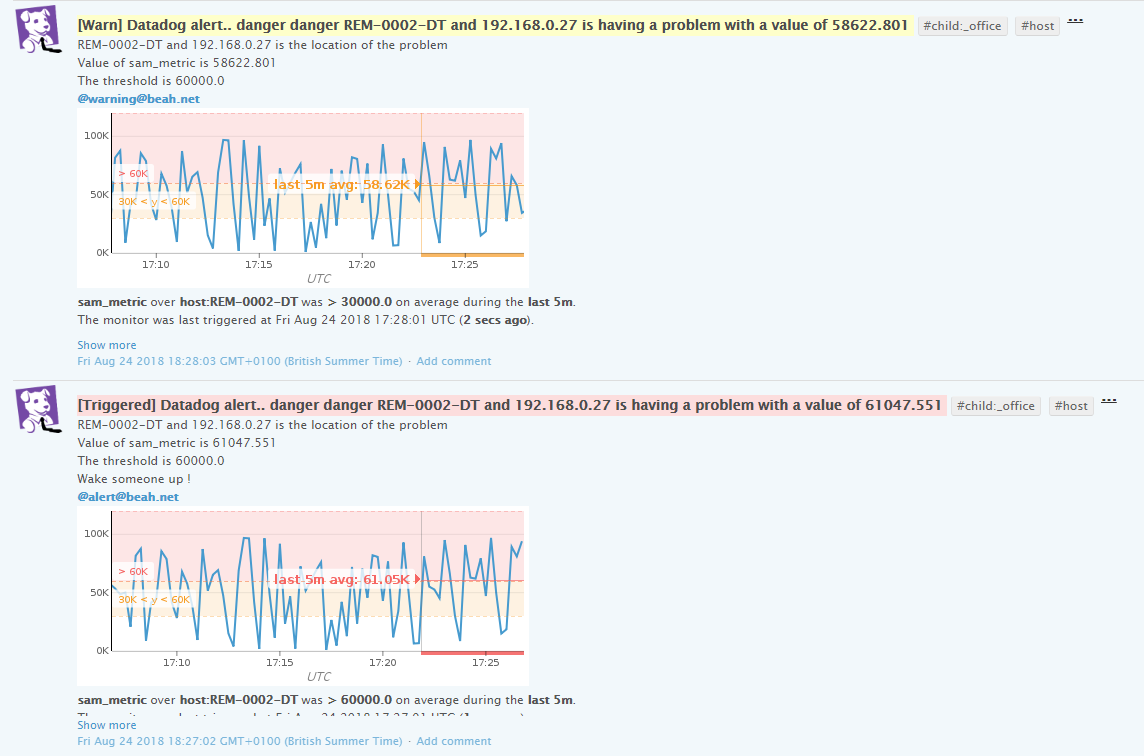
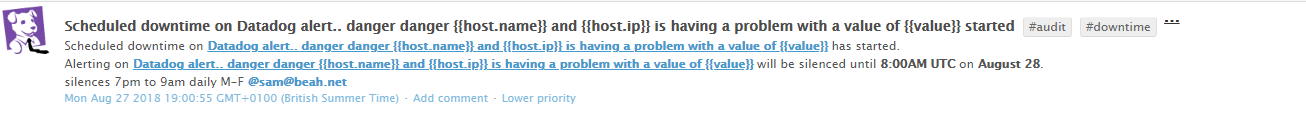
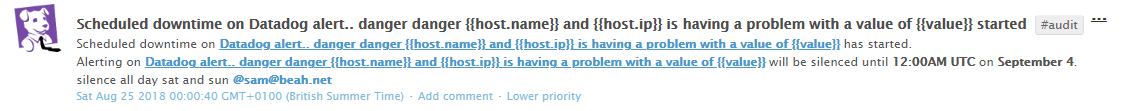
* Warning threshold of 500
* Alerting threshold of 800
* And also ensure that it will notify you if there is No Data for this query over the past 10m.

***As I was using larger variables, I altered the thresholds accordingly.***

Please configure the monitor’s message so that it will:

* Send you an email whenever the monitor triggers.
* Create different messages based on whether the monitor is in an Alert, Warning, or No Data state.
* Include the metric value that caused the monitor to trigger and host ip when the Monitor triggers an Alert state.
* When this monitor sends you an email notification, take a screenshot of the email that it sends you.

***I created the alert so it sends it to separate email addresses for Warning and Alert, and also added “wake someone up!” to the message only when it is an Alert. See below for both examples.***

* + 
* **Bonus Question**: Since this monitor is going to alert pretty often, you don’t want to be alerted when you are out of the office. Set up two scheduled downtimes for this monitor:
  + One that silences it from 7pm to 9am daily on M-F,
  + 
  + And one that silences it all day on Sat-Sun.
  + 
  + Make sure that your email is notified when you schedule the downtime and take a screenshot of that notification.

**Collecting APM Data:**

Given the following Flask app (or any Python/Ruby/Go app of your choice) instrument this using Datadog’s APM solution:

from flask import Flask

import logging

import sys

# Have flask use stdout as the logger

main\_logger = logging.getLogger()

main\_logger.setLevel(logging.DEBUG)

c = logging.StreamHandler(sys.stdout)

formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')

c.setFormatter(formatter)

main\_logger.addHandler(c)

app = Flask(\_\_name\_\_)

@app.route('/')

def api\_entry():

return 'Entrypoint to the Application'

@app.route('/api/apm')

def apm\_endpoint():

return 'Getting APM Started'

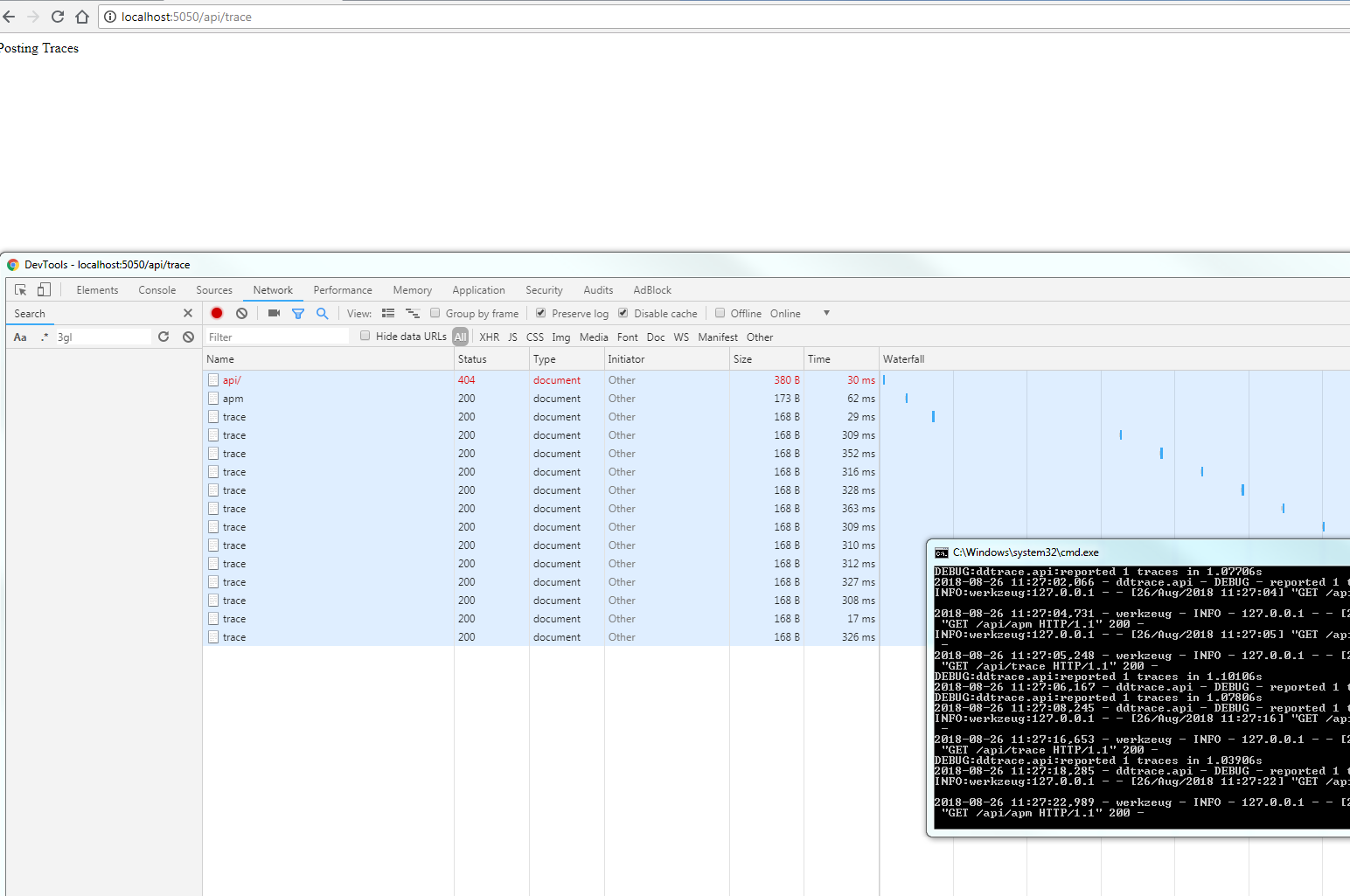
@app.route('/api/trace')

def trace\_endpoint():

return 'Posting Traces'

if \_\_name\_\_ == '\_\_main\_\_':

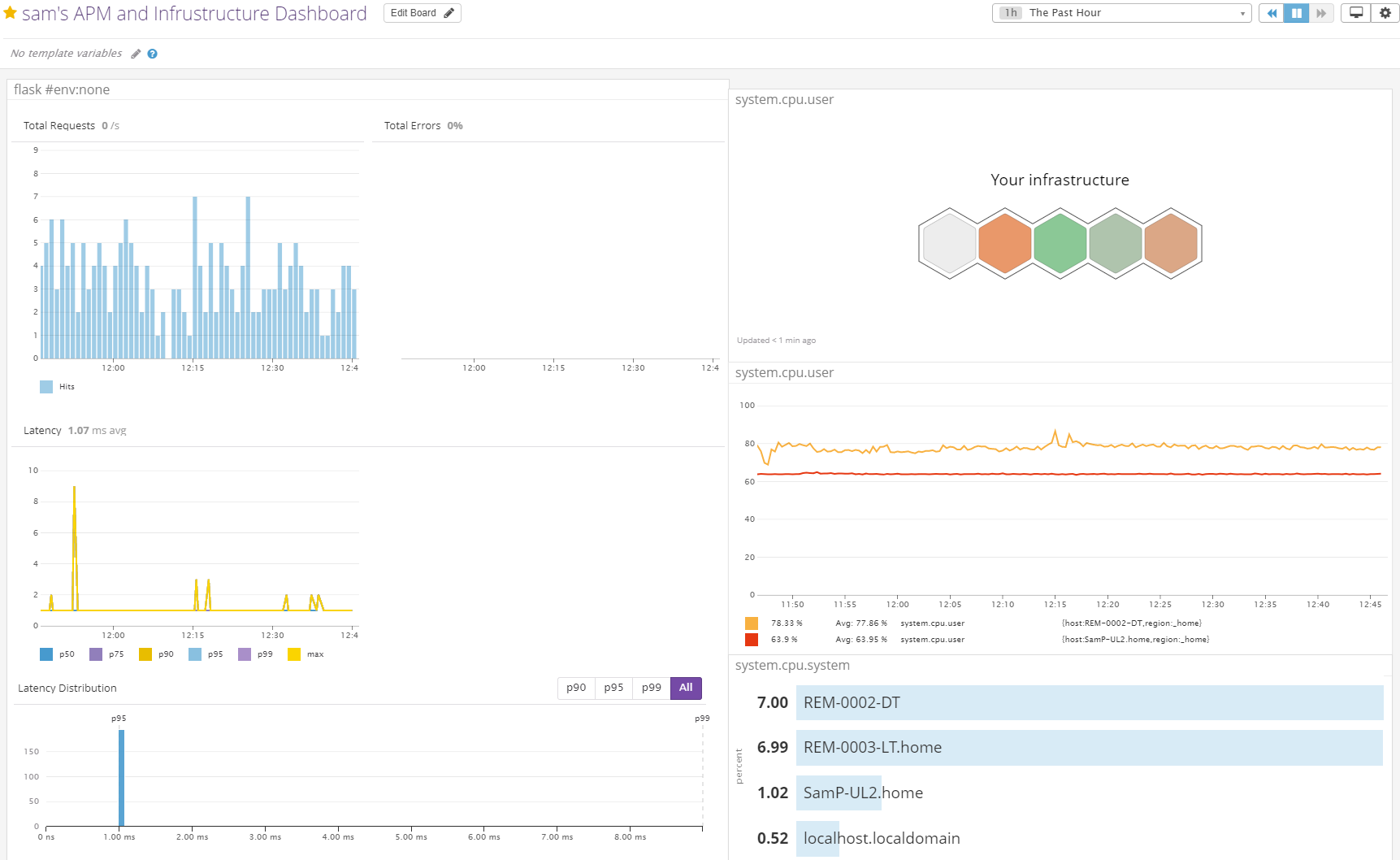
app.run(host='0.0.0.0', port='5050')

* **Note**: Using both ddtrace-run and manually inserting the Middleware has been known to cause issues. Please only use one or the other.
  + ***I had to change the host to 127.0.01 and then I used a script to refresh the browser to keep loading the flask app apm pages to get more data.***
  + 
* **Bonus Question**: What is the difference between a Service and a Resource?

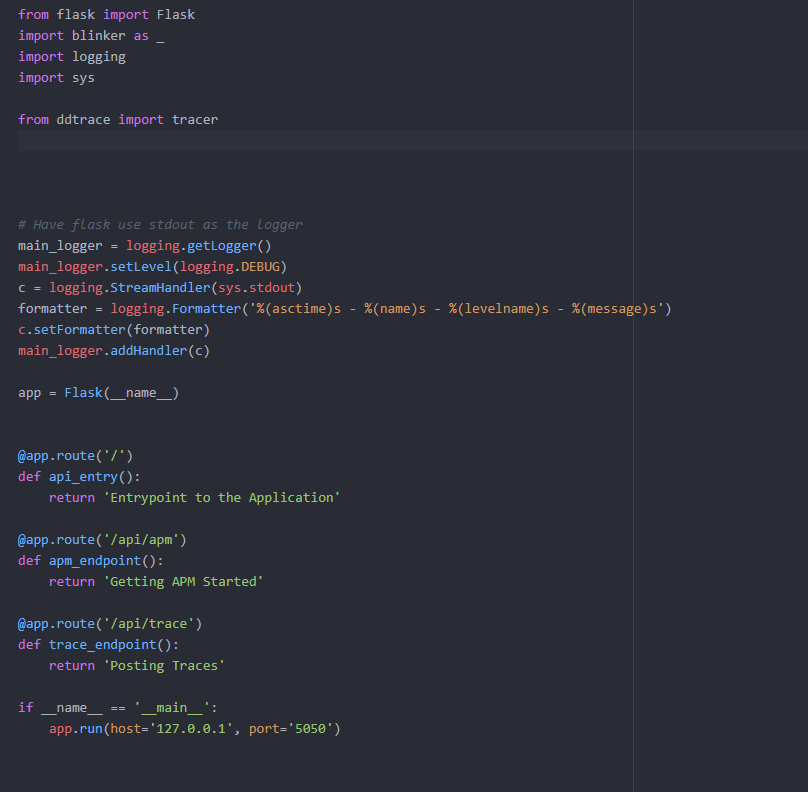
***A service refers to a set of processes which combined will provide a feature set. Whereas a Resource is a specific query to a service, like an SQL request, or URL request.***

Provide a link and a screenshot of a Dashboard with both APM and Infrastructure Metrics.

Public link is <https://p.datadoghq.com/sb/63023bf39-1fd69d2b7c0cd398a3ba6196531034a6>



Please include your fully instrumented app in your submission, as well.



**Final Question:**

Datadog has been used in a lot of creative ways in the past. We’ve written some blog posts about using Datadog to monitor the NYC Subway System, Pokemon Go, and even office restroom availability!

Is there anything creative you would use Datadog for?

***There are so many use cases as there Is data all around us. But one vision of the future comes to Driverless cars. Imagine a company like Uber having a huge fleet of driver less cars. They will all need to be monitored.***

***So I’m picturing the wall at Uber HQ. A huge screen showing maps, in each map you can see each car moving around. Each car comes with all these metrics, all delivered to Datadog from the cars themselves. Speed, energy levels, time on road, wheel pressure, passengers, revenue, addresses and so much more.***

***This data can not only be used to track the cars, make sure they are running efficiently, but also provide exec reports and dashboards showing usage and revenue.***