Setup

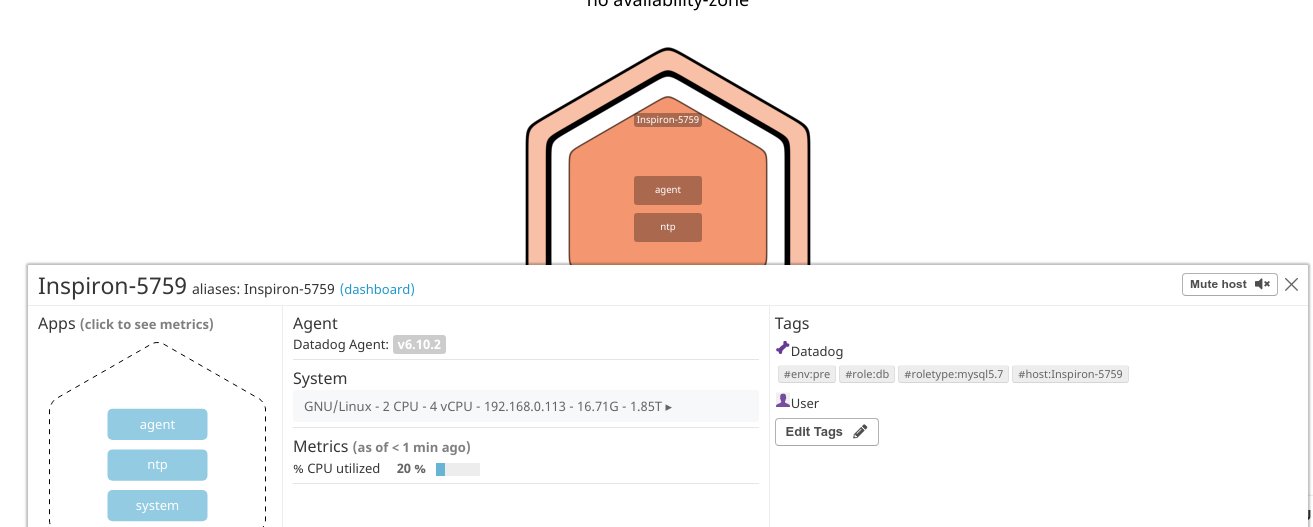
Agent installed on an Ubuntu server.

**I Collecting Metrics:**

1.- Tagging

Speciying several tags for fine grained filtering capabilities

tags: role:db, env:pre, roletype:mysql5.7



2.- Installed Mysql sw

mysql-client-5.7 install

mysql-client-core-5.7 install

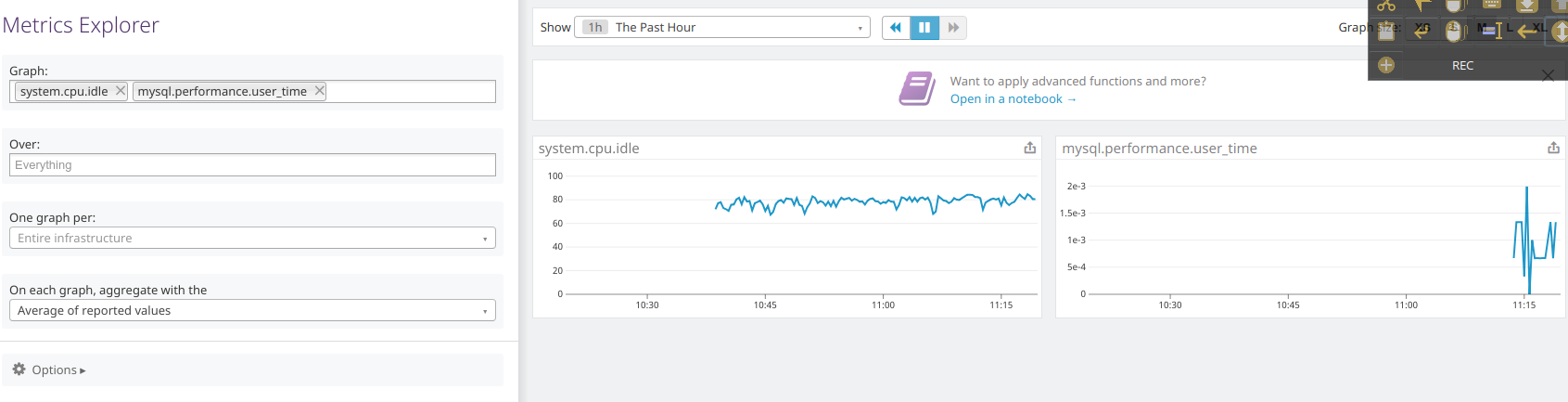
mysql-common install

mysql-server install

mysql-server-5.7 install

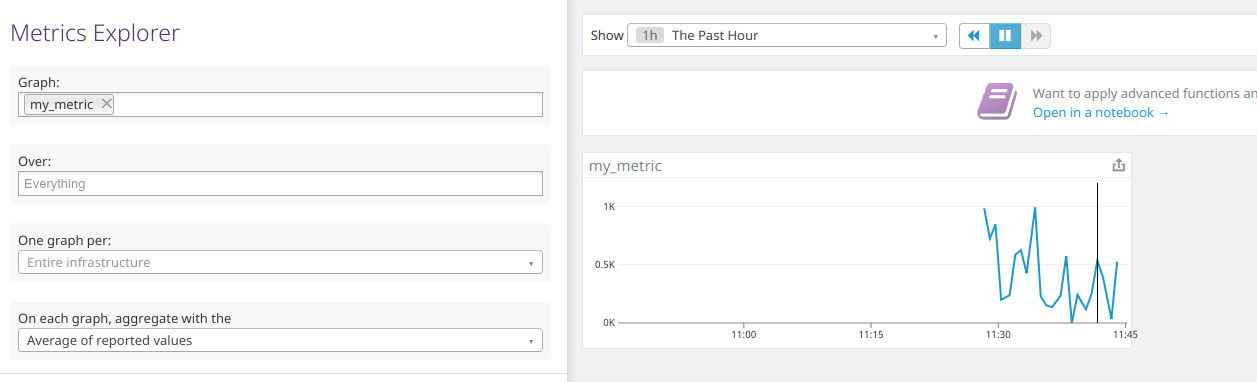
mysql-server-core-5.7 install

Then, installing Mysql integration. Checking with the explorer that we are getting data from server and mysql



3.- Create Agent check

Files copied into the corresponding folder.

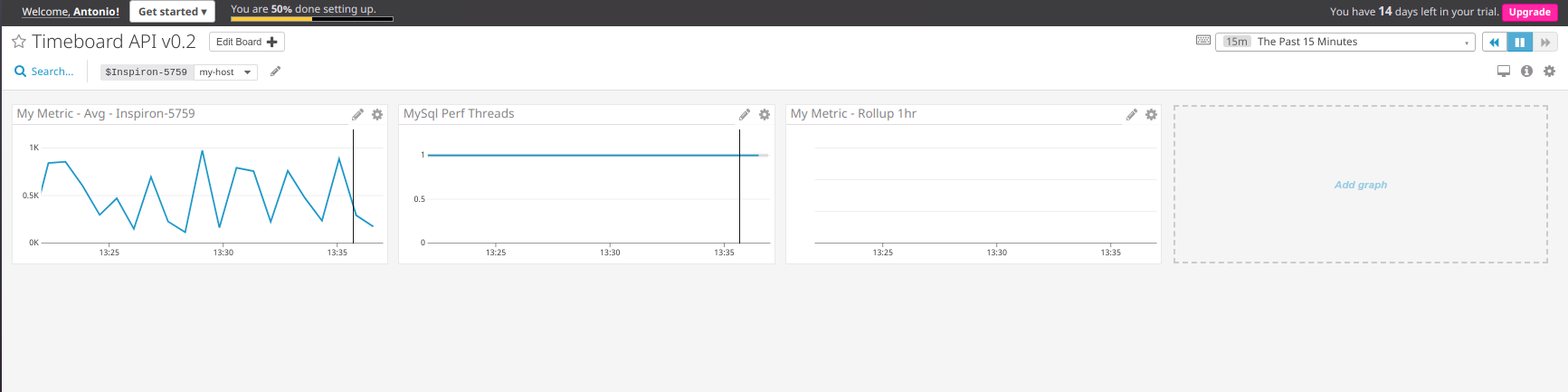


**II Visualizing Data:**

1.- Create a Dashboard via API

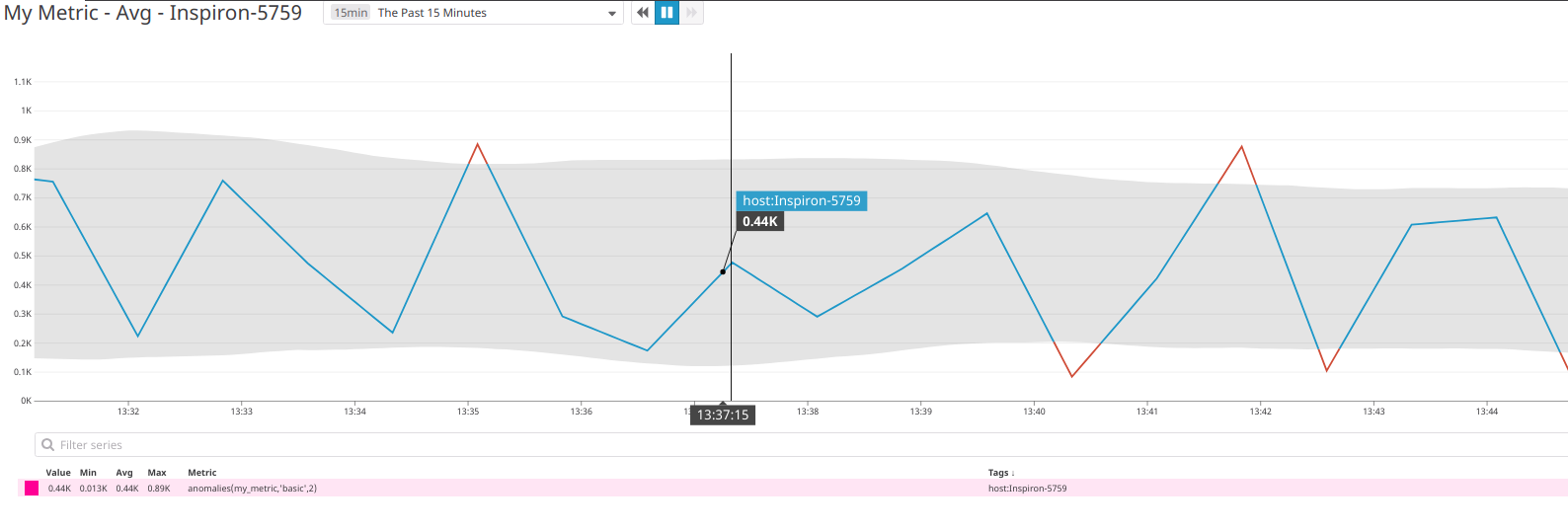
2.- Taken an screenshot of the my\_metric graph

Could not find last 5 min option in time range, using last 15 minutes instead



Bonus question

I had to add the anomaly function to the first timeseries showing the my\_metric, because the mysql metrics were mostly flat.

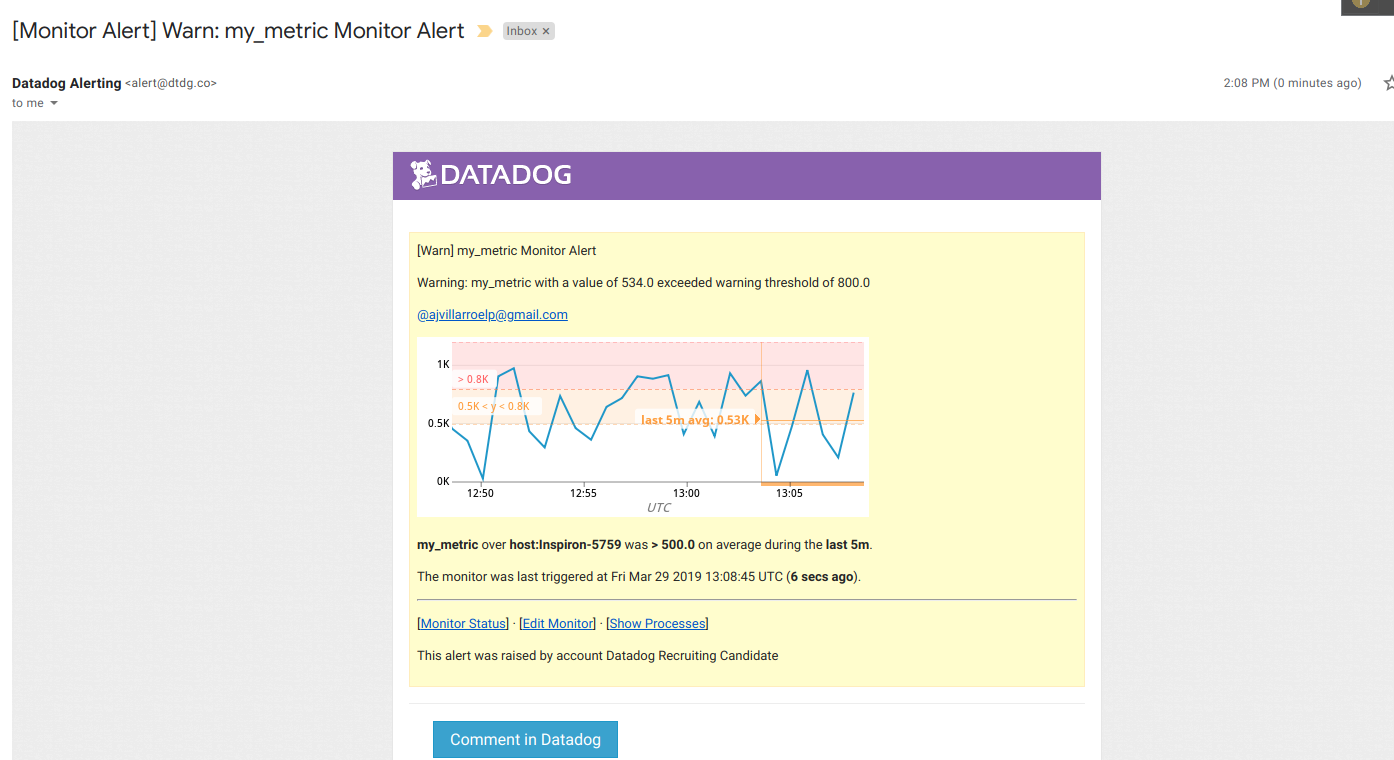


So, the graph with the anomaly function highlights in red the values that are not expected (too low/high) based on the values seen in the past. In grey the graph shows the predicted values based on the algorithm used for the anomaly function.

**III Monitoring Data:**

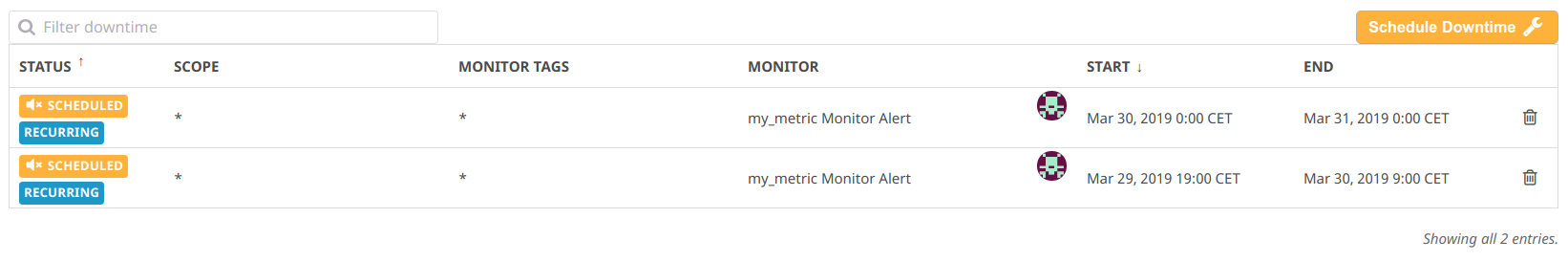
1.- Create a monitor for the my\_metric metric

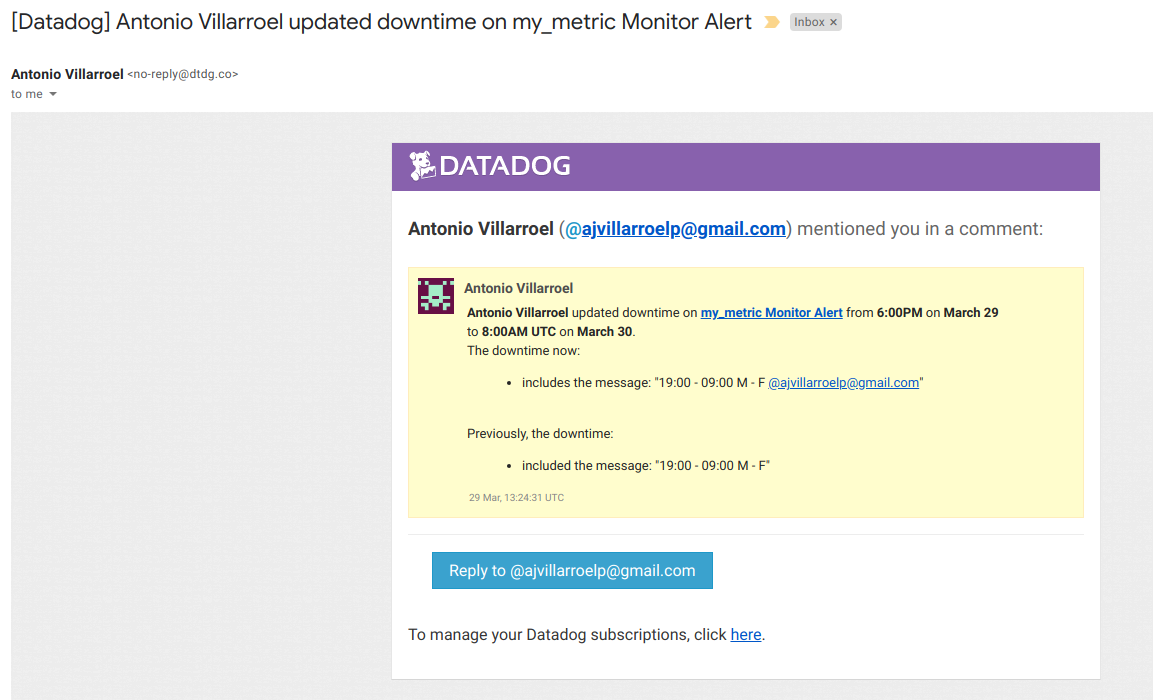
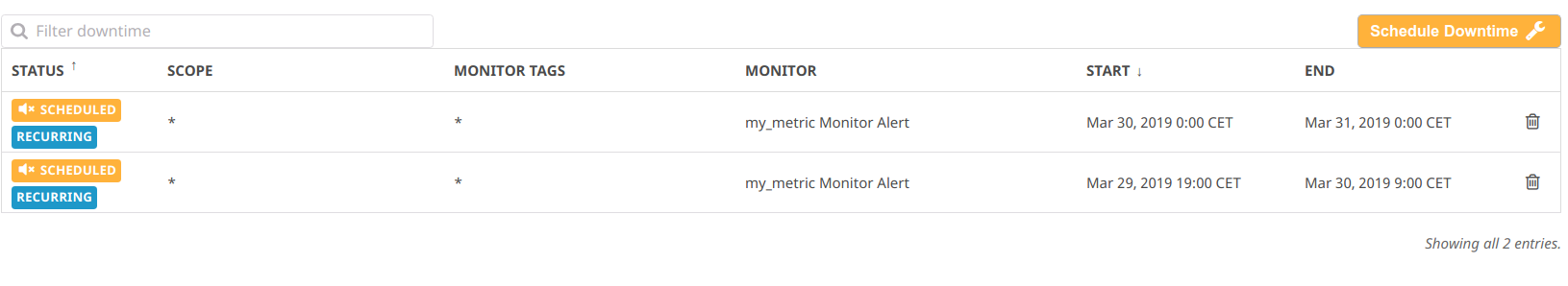
Warning alert

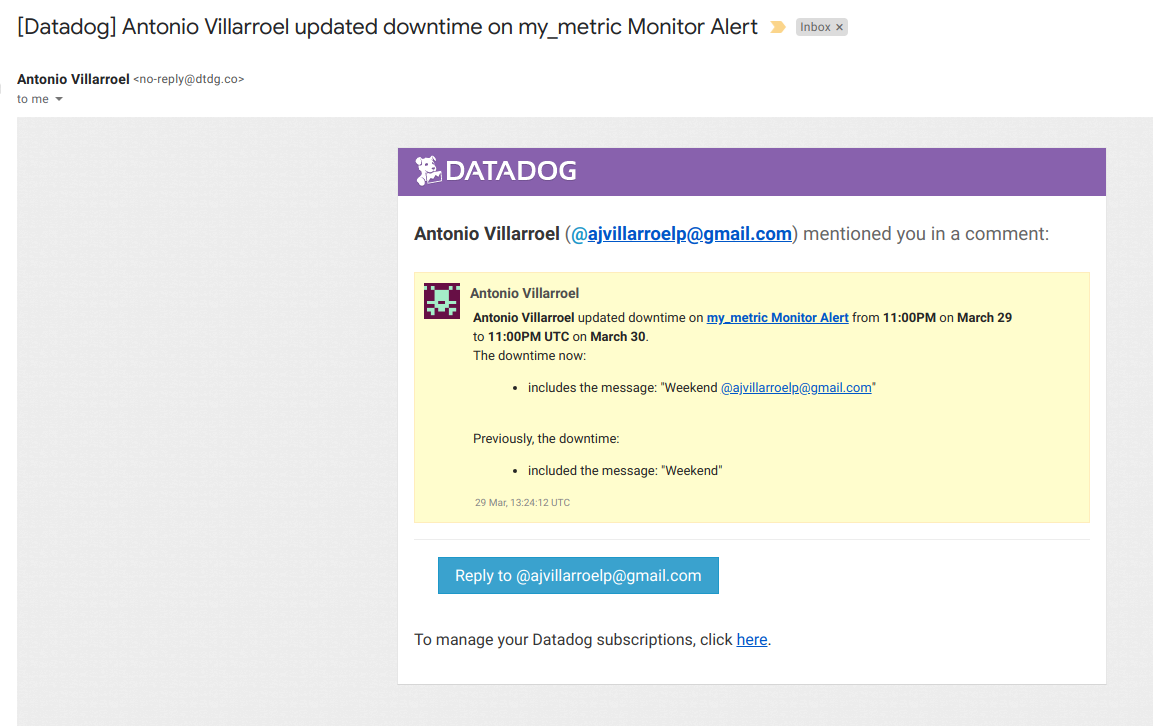


Bonus:

Downtimes:







**IV Collecting APM Data:**

1.- Instrument an application (python based, flask\_app.py)

Steps taken

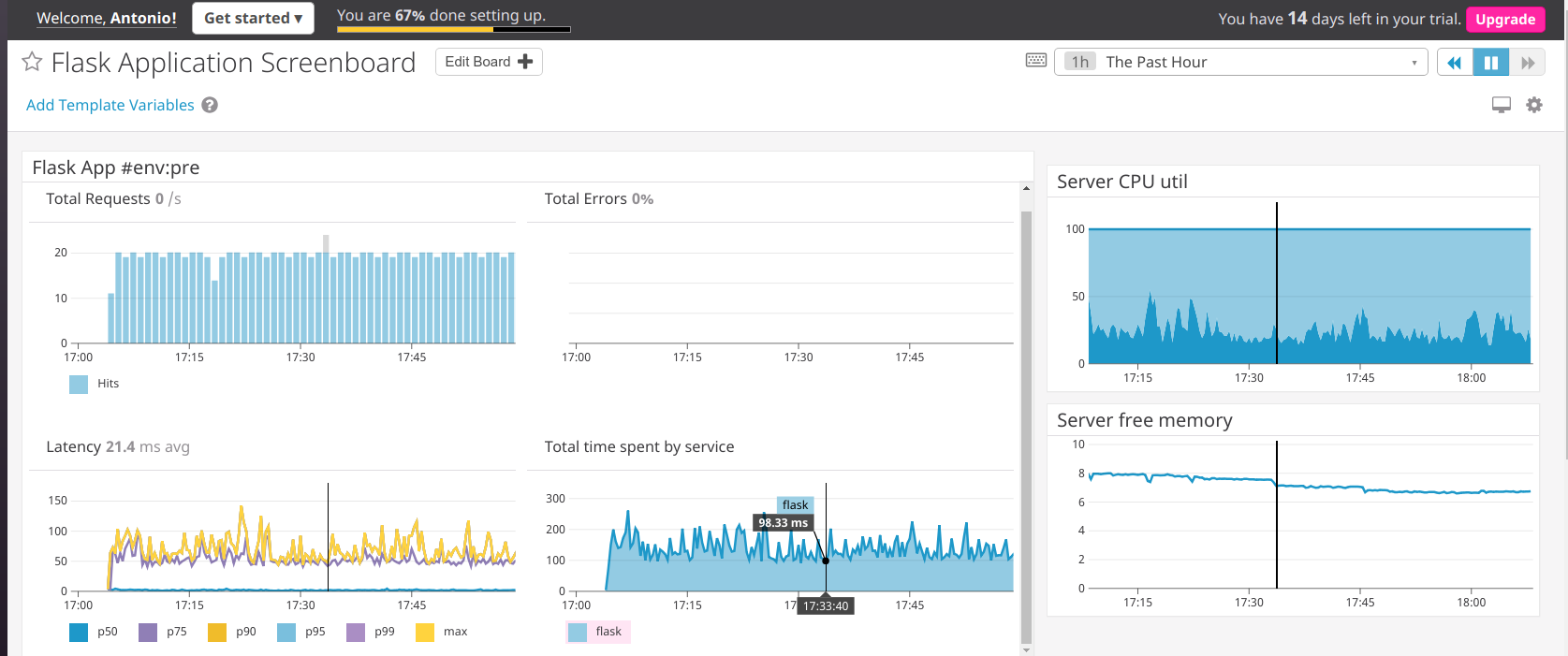
- Activated tracing in datadog agent config file

- install library sudo pip3 install ddtrace

- run ddtrace-run python3 flask\_app.py

Also, provided a simulator of activity to generate traffic to the app

Screenboard for Flask app



https://p.datadoghq.com/sb/9d4bnv9ca849v6du-f5d7594f12c27f8853f72dc5d4ecccb1

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

A service is process that accepts requests, and do some function. They are develop in languages like java, php, python, go, etc. Resources are the different requests of a service, in a web app is an endpoint. In our instrumented application, flask is the service (a process that accepts web requests from a browser), and the web requests /, /api/apm, etc are the resources.

**Final Question**

I would suggest this idea:

Monitor IOT devices, in particular smart trucks, for fleet management. Datadog agent could collect data from the truck itself (like temperature, location, fuel level, tires pressure, hours driving, etc) and complement services like the ones envisioned in the following article:

http://www.supplychain247.com/article/how\_the\_internet\_of\_things\_transforms\_trucking

Datadog dashboards could provide all this information with sexy reports, tons of options to combine different indicators, and more importantly, alerts could enforce logistics and regulation in place (like drivers cannot drive more than x hours, or the need to check the tires pressure because is too low). With the inherent multy-tenancy of datadog Saas environment, this scheme could be replicated for different customers very easily.