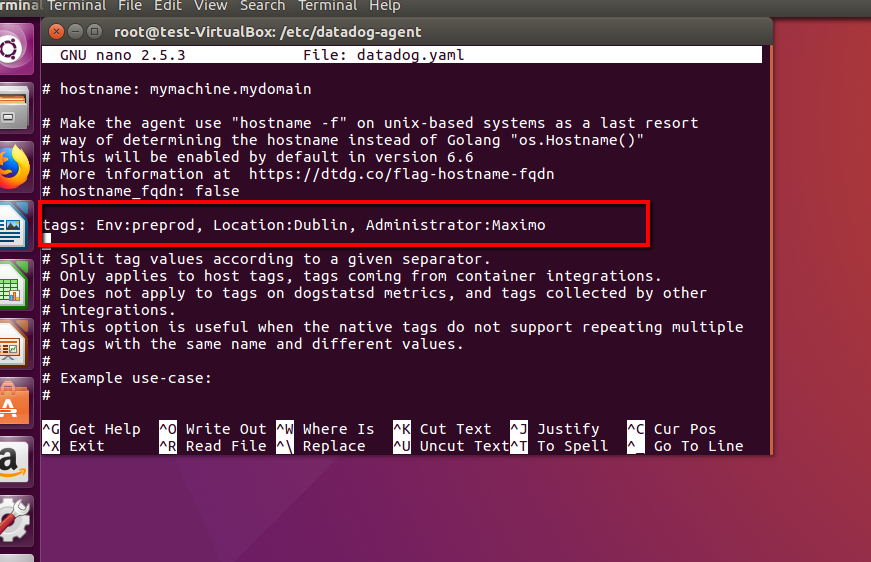
## **Prerequisites - Setup the environment**

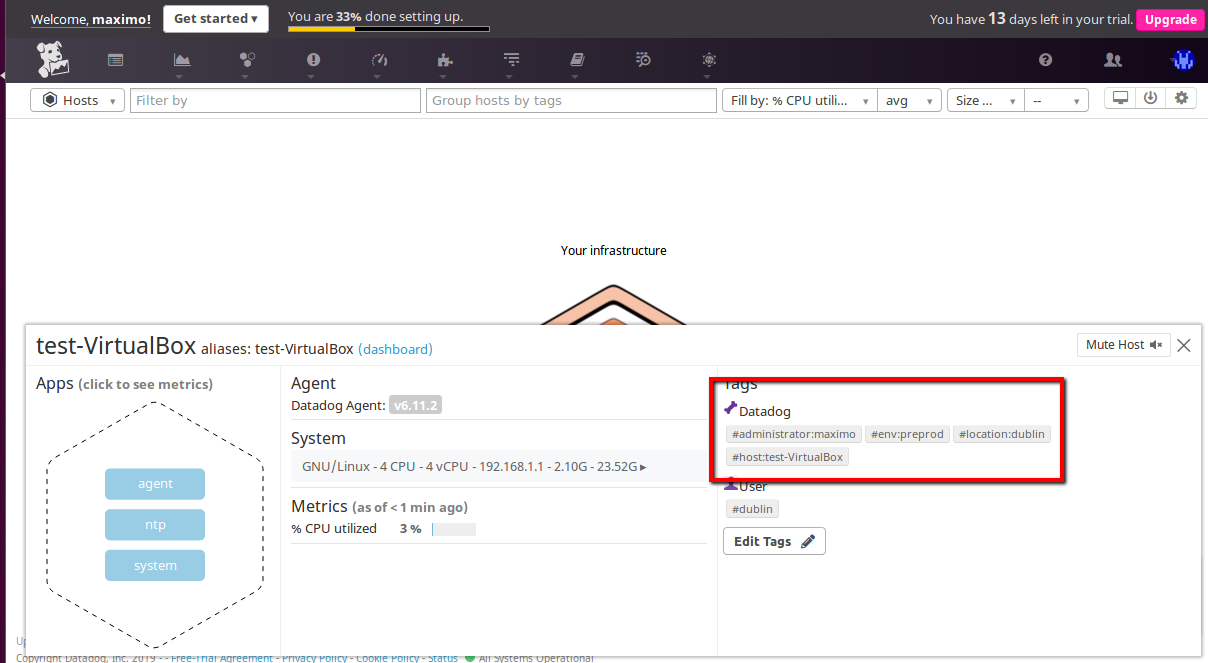
* Vbox with Ubuntu16 x64
* Creation of a test datadog account
* Installation of the Ubuntu Agent

DD\_API\_KEY=ee6c7f446eb6907368b48d2621400e8c bash -c "$(curl -L <https://raw.githubusercontent.com/DataDog/datadog-agent/master/cmd/agent/install_script.sh>)"

## **Collecting Metrics:**

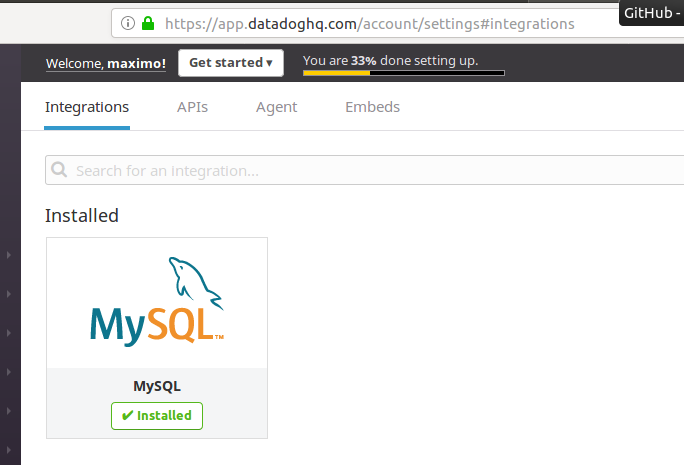
**Add tags in the Agent config**

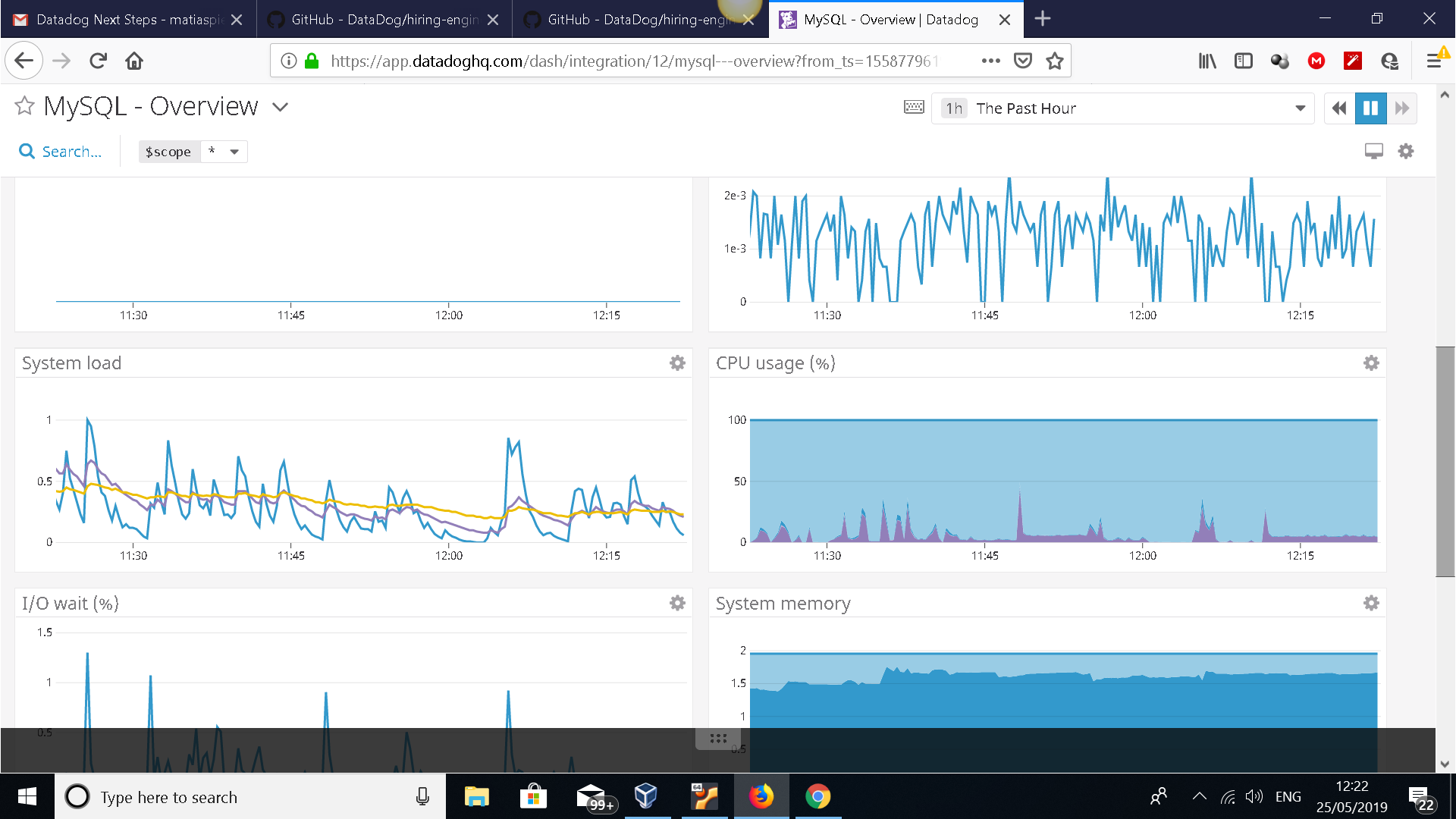




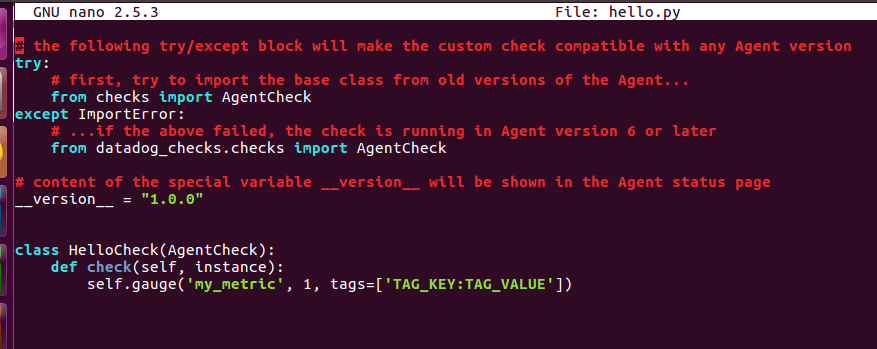
**Install a database on your machine (MongoDB, MySQL, or PostgreSQL) and then install the respective Datadog integration for that database.**

1. Create a datadog2 user in MySQL and grant it permission to run metric queries on your behalf.
2. Copy Datadog’s conf.d/mysql.yaml.example [template](https://github.com/DataDog/integrations-core/blob/master/mysql/datadog_checks/mysql/data/conf.yaml.example) to conf.d/mysql.yaml to create a configuration file for Datadog.
3. Add the login credentials for your newly created datadog user to conf.d/mysql.yaml.
4. Restart the Agent.



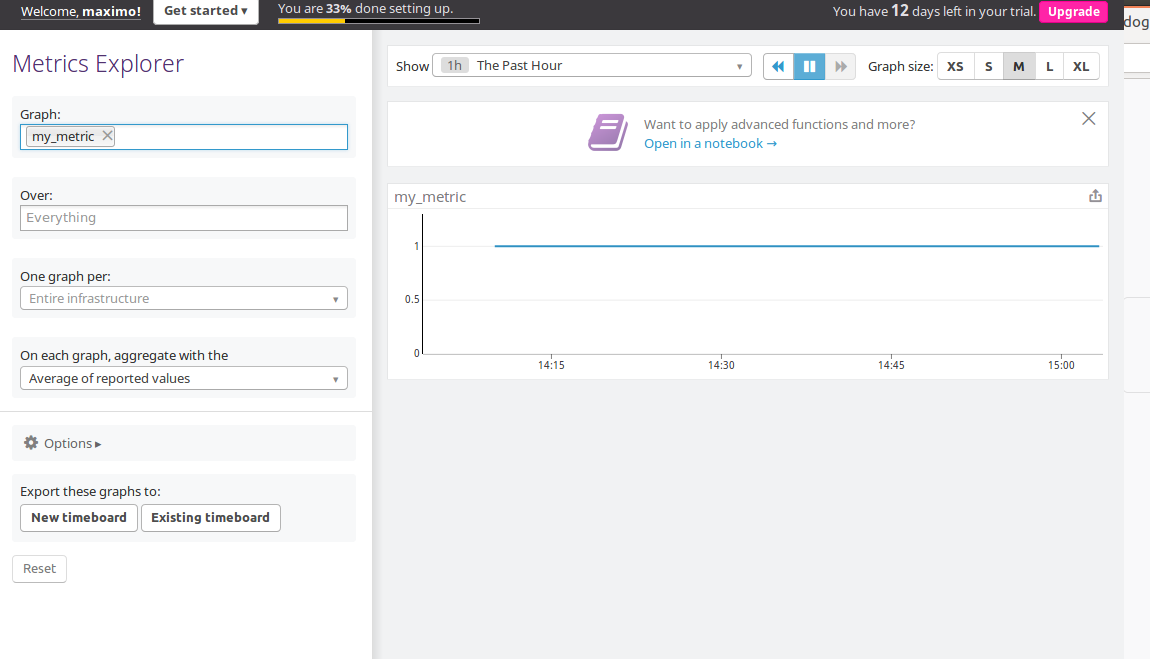


**Create a custom Agent check that submits a metric named my\_metric with a random value between 0 and 1000.**



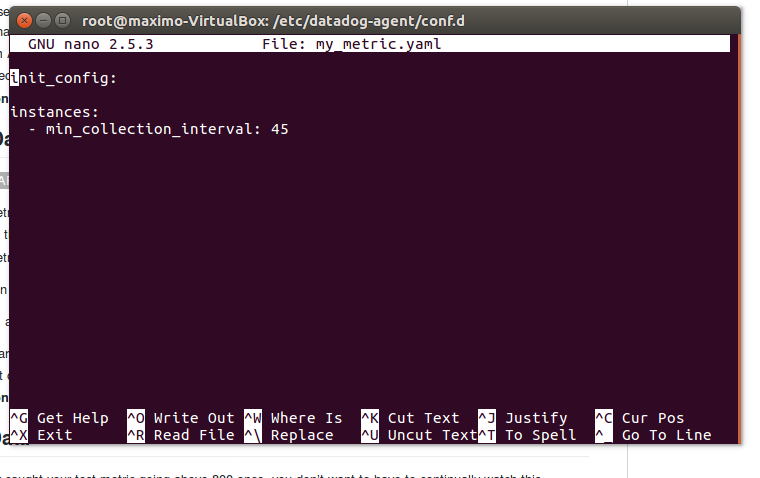


Result



**Change your check's collection interval so that it only submits the metric once every 45 seconds.**

To change the collection interval of your check, use min\_collection\_interval in the configuration file. The default value is 15 which means the check method from your class is invoked with the same interval as the rest of the integrations on the Agent.



**Bonus Question** Can you change the collection interval without modifying the Python check file you created?

Yes ...but

I was not able to find an “official” process for this .

you can achieve this in you add : min\_collection\_interval : 45 in the configuration /etc/datadog-agent/conf.d/my\_metric.d/conf.yaml.

## **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

api\_key=5c80ed4f354b0681429ec597907e3171

app\_key=24b14b2c90191a57605bd01a03b8c19b421a9da6

curl -X POST -H "Content-type: applicaiton/json" \

-d '{

"title": "Maximo Timeboard",

"description": "A New Timeboard with Metric Information",

"graphs" :

[

{

"title": "my\_metric scoped",

"definition": {

"events": [],

"requests": [

{"q": "avg:my\_metric{host:maximo-VirtualBox}"}

]

},

"viz": "timeseries"

},

{

"title": "Mysql Anomallies",

"definition": {

"events": [],

"requests": [

{"q": "anomalies(avg:my\_metric{host:maximo-VirtualBox}, \"basic\", 3)"}

]

},

"viz": "timeseries"

},

{

"title": "my\_metric w/rollup ",

"definition": {

"events": [],

"requests": [

{"q": "avg:my\_metric{host:maximo-VirtualBox}.rollup(\"sum\", 3600)"}

]

}

}

]

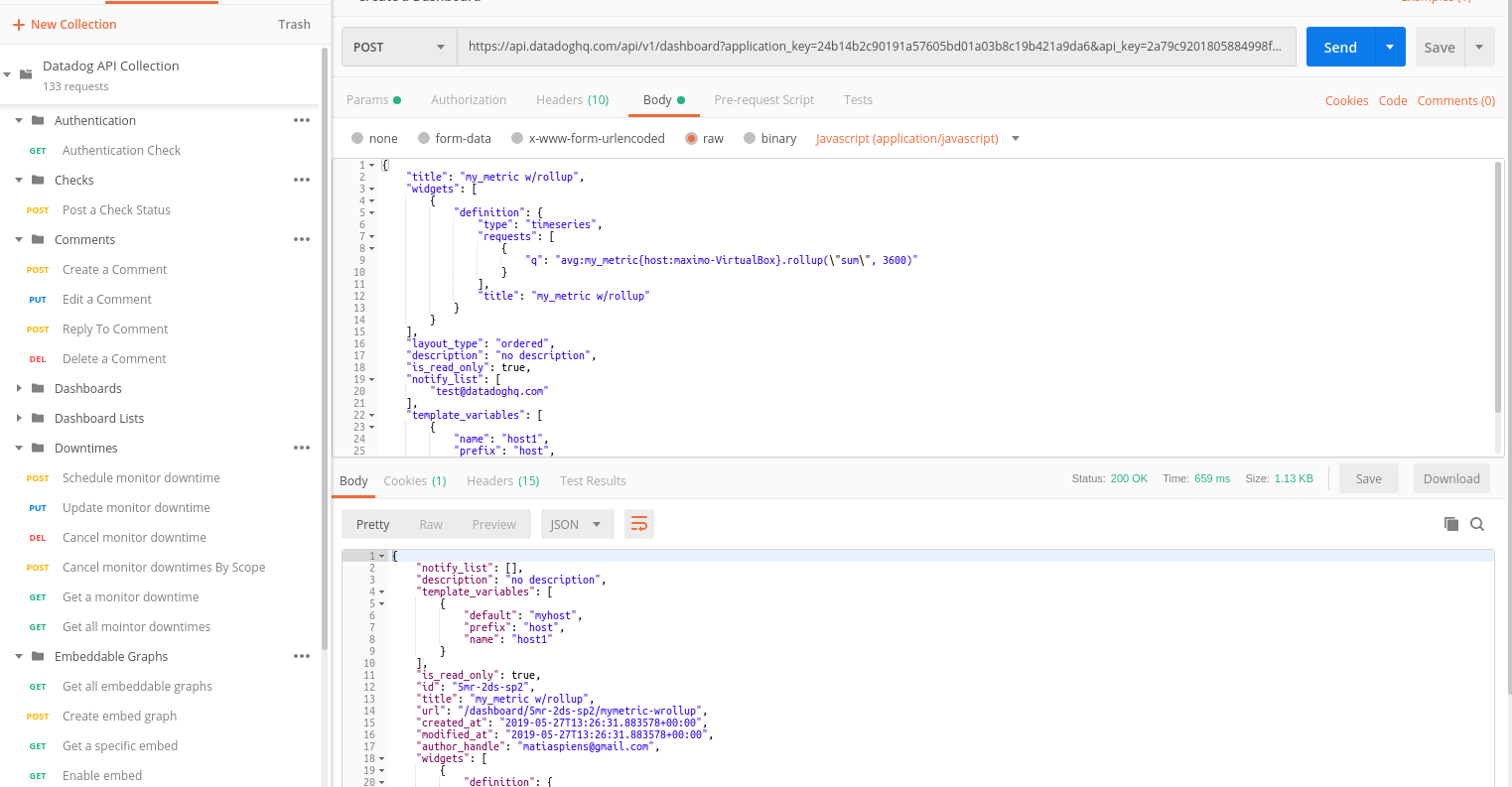
}' \

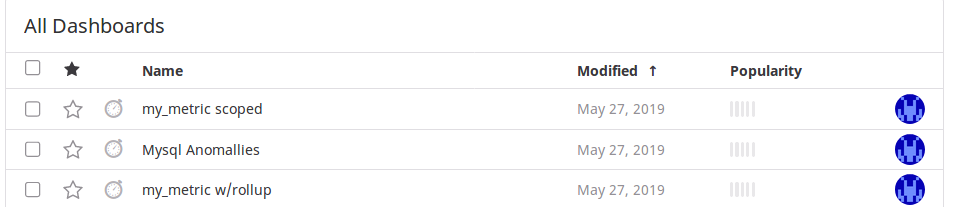
"[https://api.datadoghq.com/api/v1/dash?api\_key=${api\_key}&application\_key=${app\_key}](https://api.datadoghq.com/api/v1/dash?api_key=$%7Bapi_key%7D&application_key=$%7Bapp_key%7D)"

I tried many times but I was not able to send this ( API error message) so I split this one on 3 script and send this one with postman .

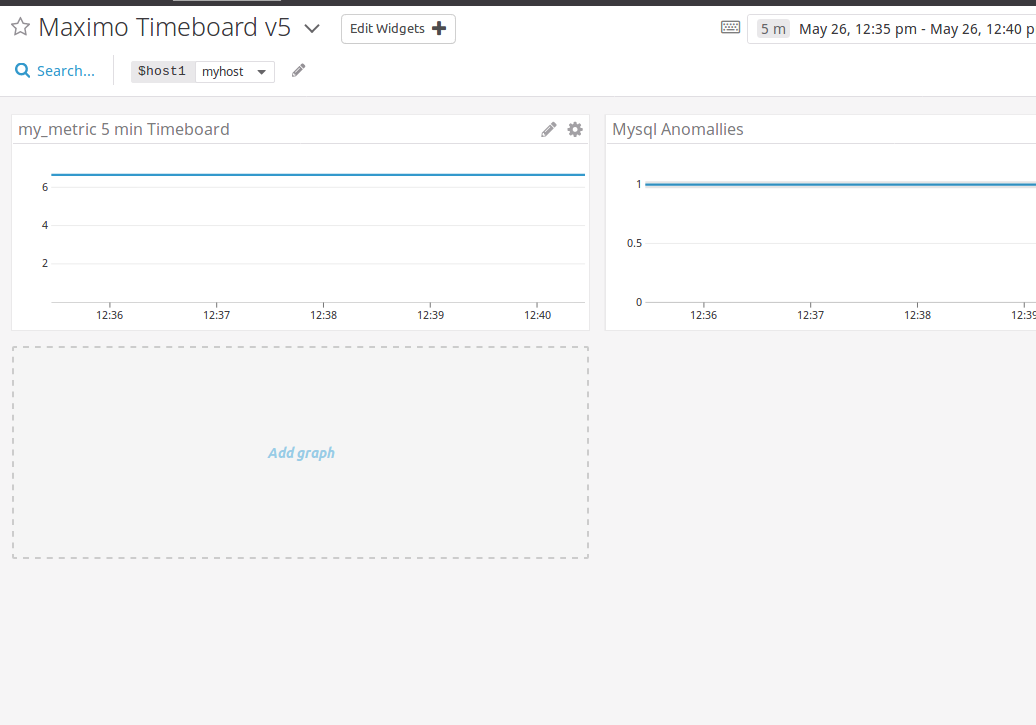
Query present here :

<https://api.datadoghq.com/api/v1/dashboard?application_key=24b14b2c90191a57605bd01a03b8c19b421a9da6&api_key=2a79c9201805884998f434b56aa25d4e>

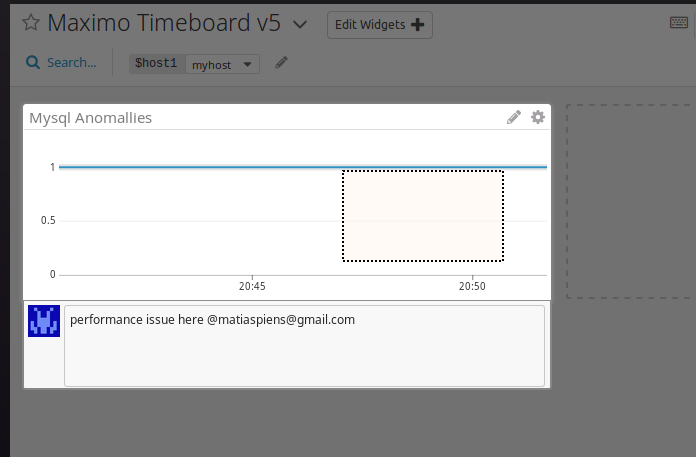




**Set the Timeboard's timeframe to the past 5 minutes**



**Take a snapshot of this graph and use the @ notation to send it to yourself.**



**Bonus Question**: What is the Anomaly graph displaying?

Anomaly detection is an algorithmic feature that allows you to identify when a metric is behaving differently than it has in the past.

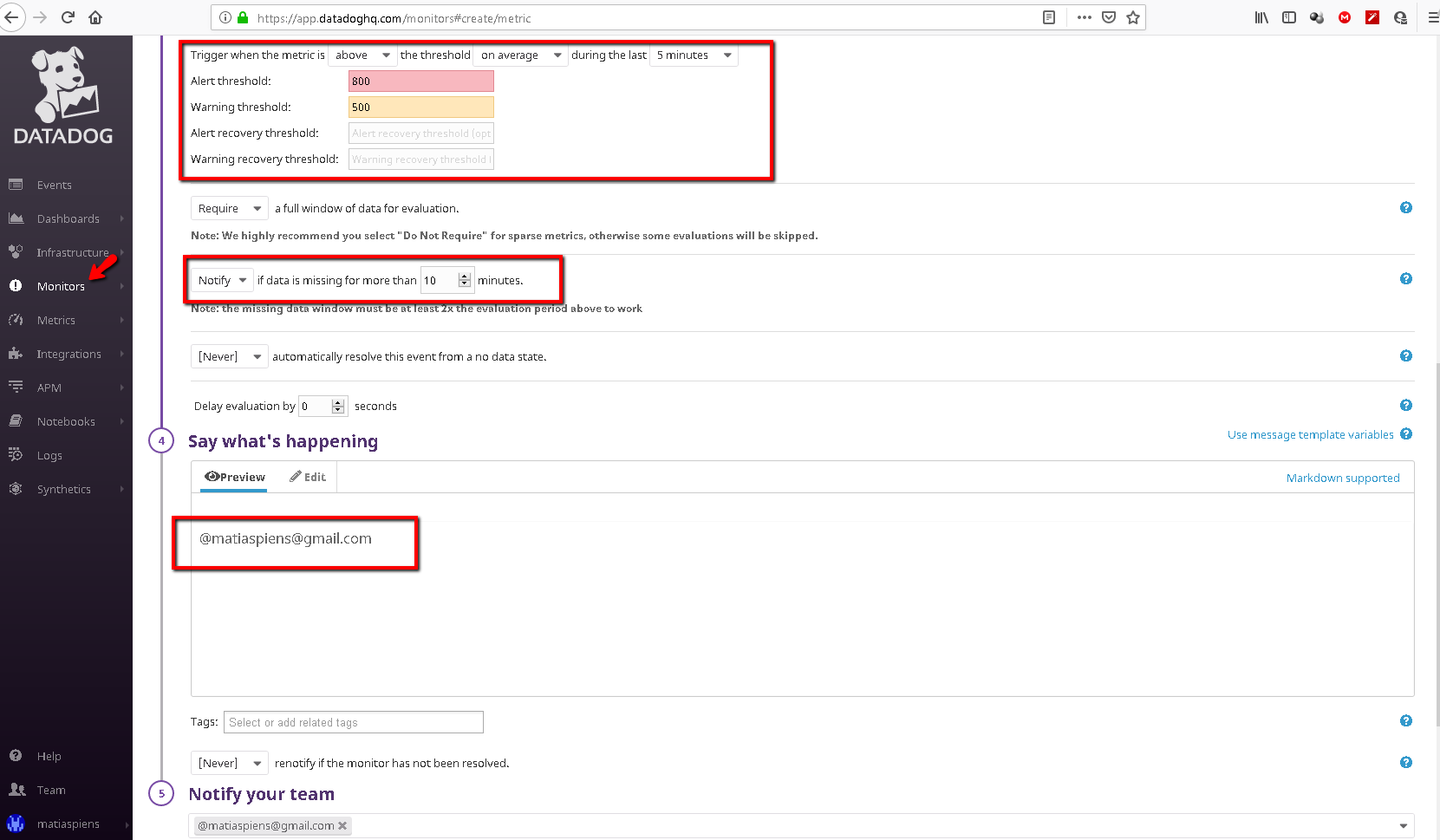
<https://docs.datadoghq.com/monitors/monitor_types/anomaly/>

## **Monitoring Data**

* **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.**

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

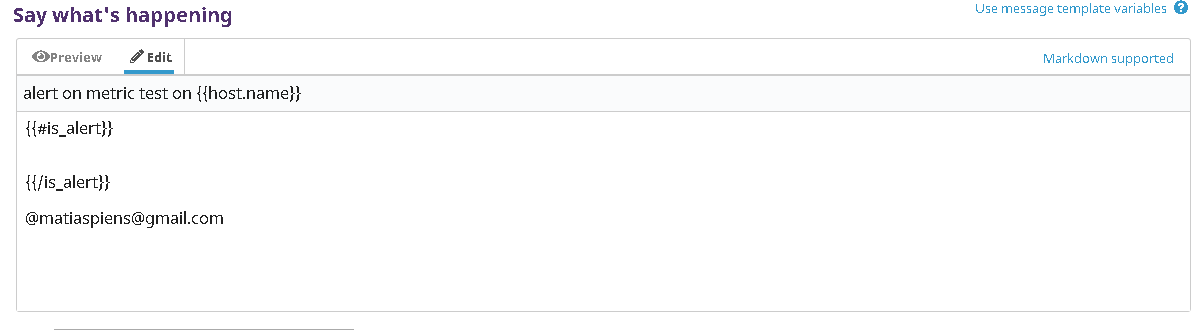
* **Send you an email whenever the monitor triggers.**

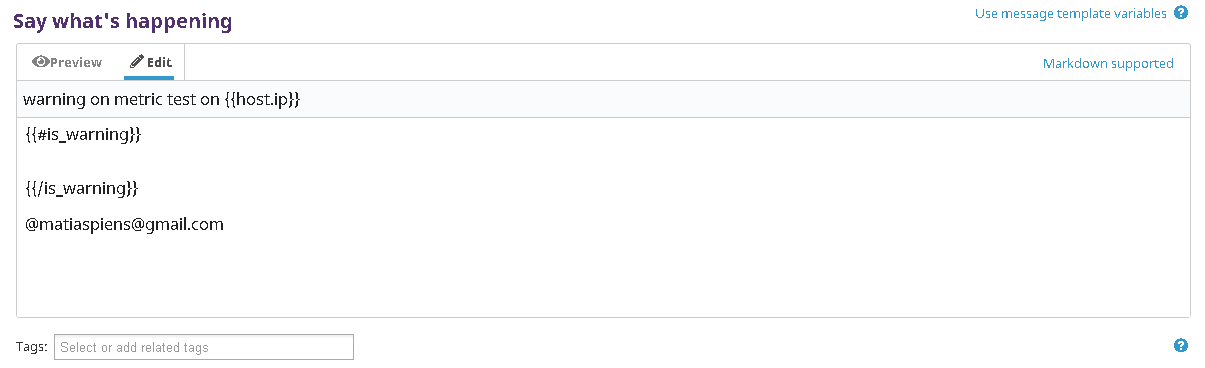


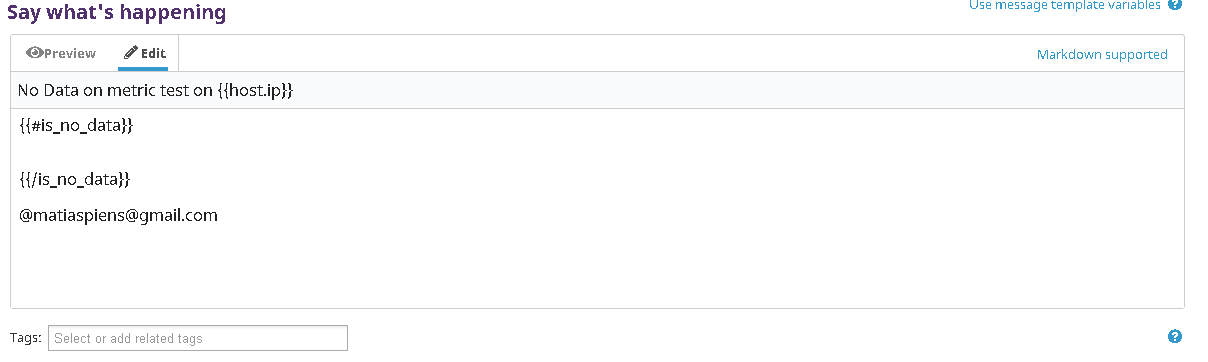
**Create different messages based on whether the monitor is in an Alert, Warning, or No Data state.**

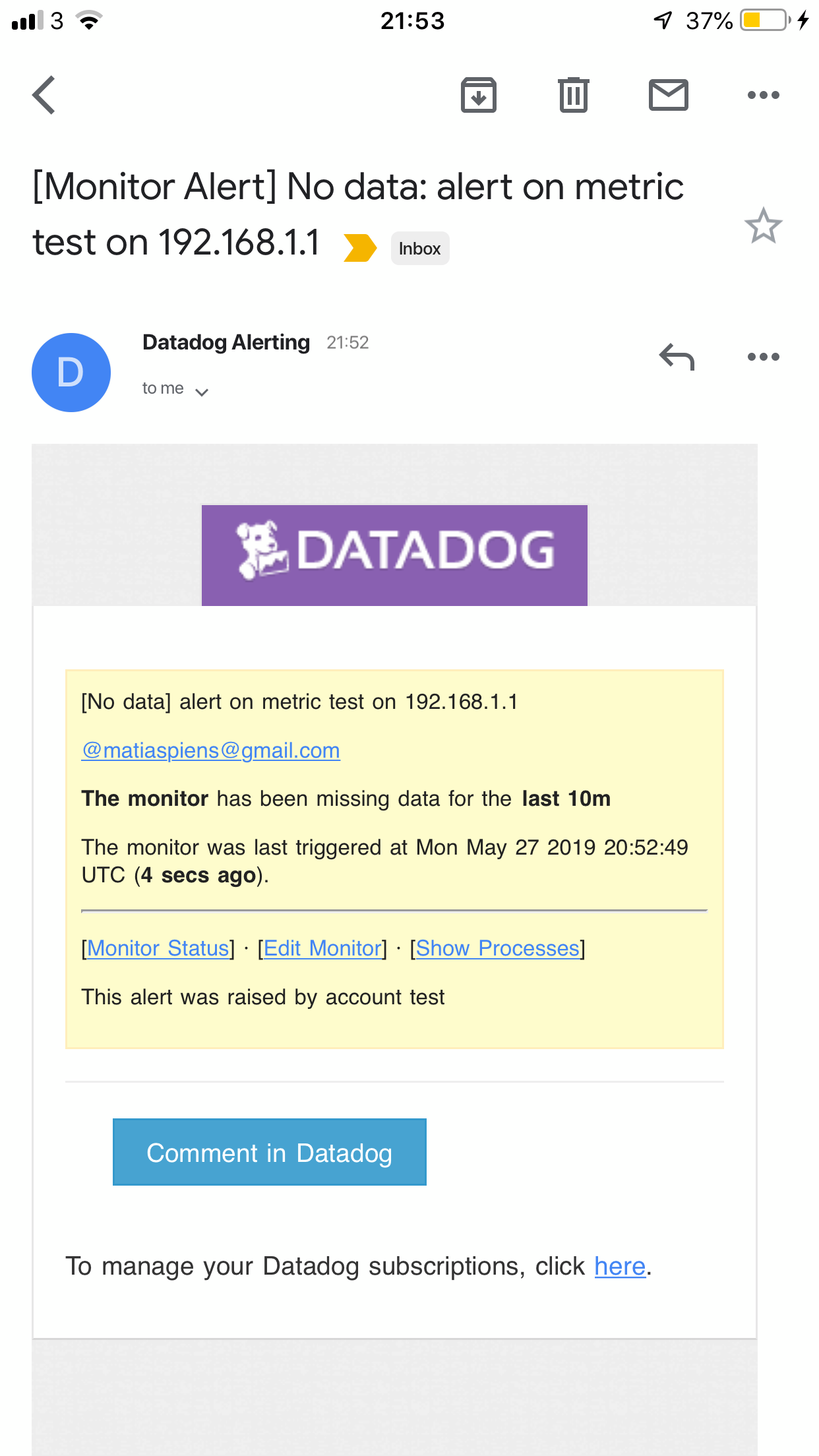
**Include the metric value that caused the monitor to trigger and host ip when the Monitor triggers an Alert state.**

**When this monitor sends you an email notification, take a screenshot of the email that it sends you.**

****

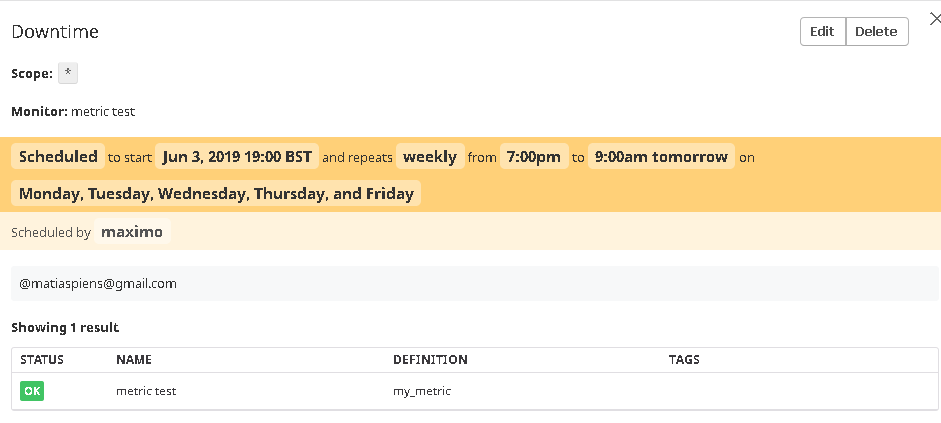
****

****

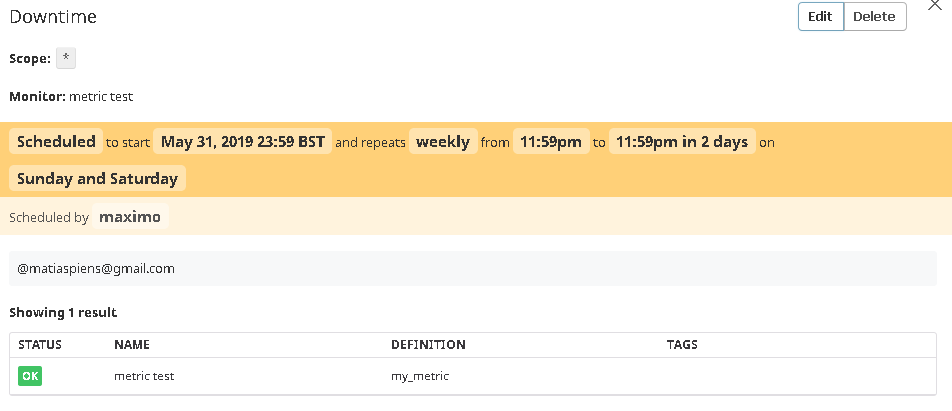
****

**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.



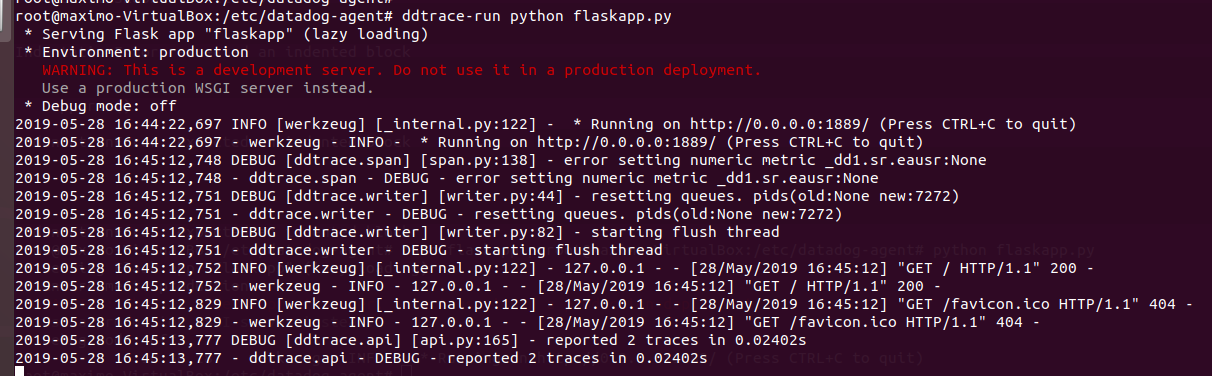
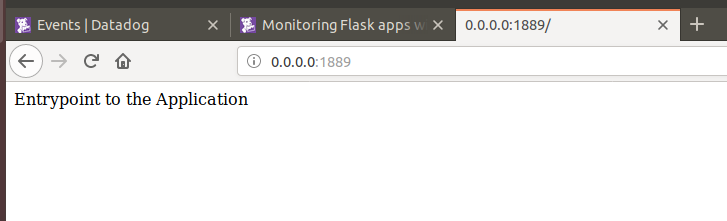
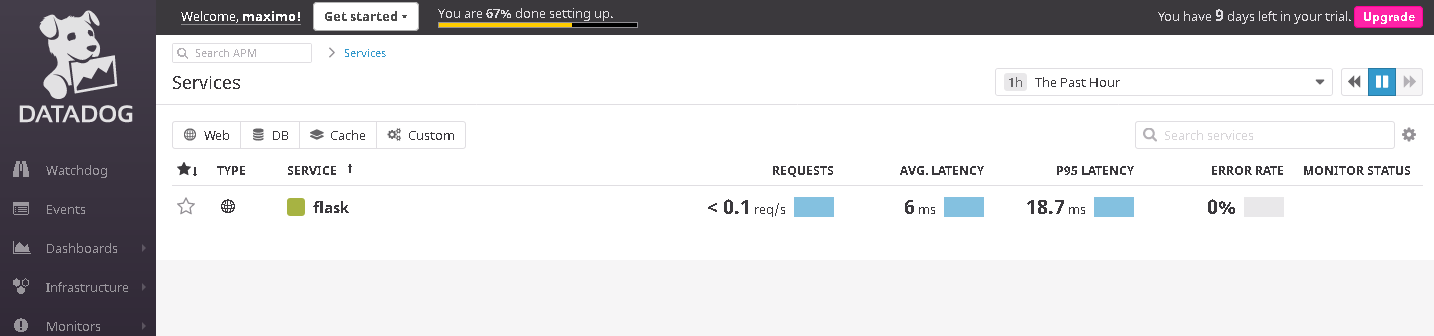
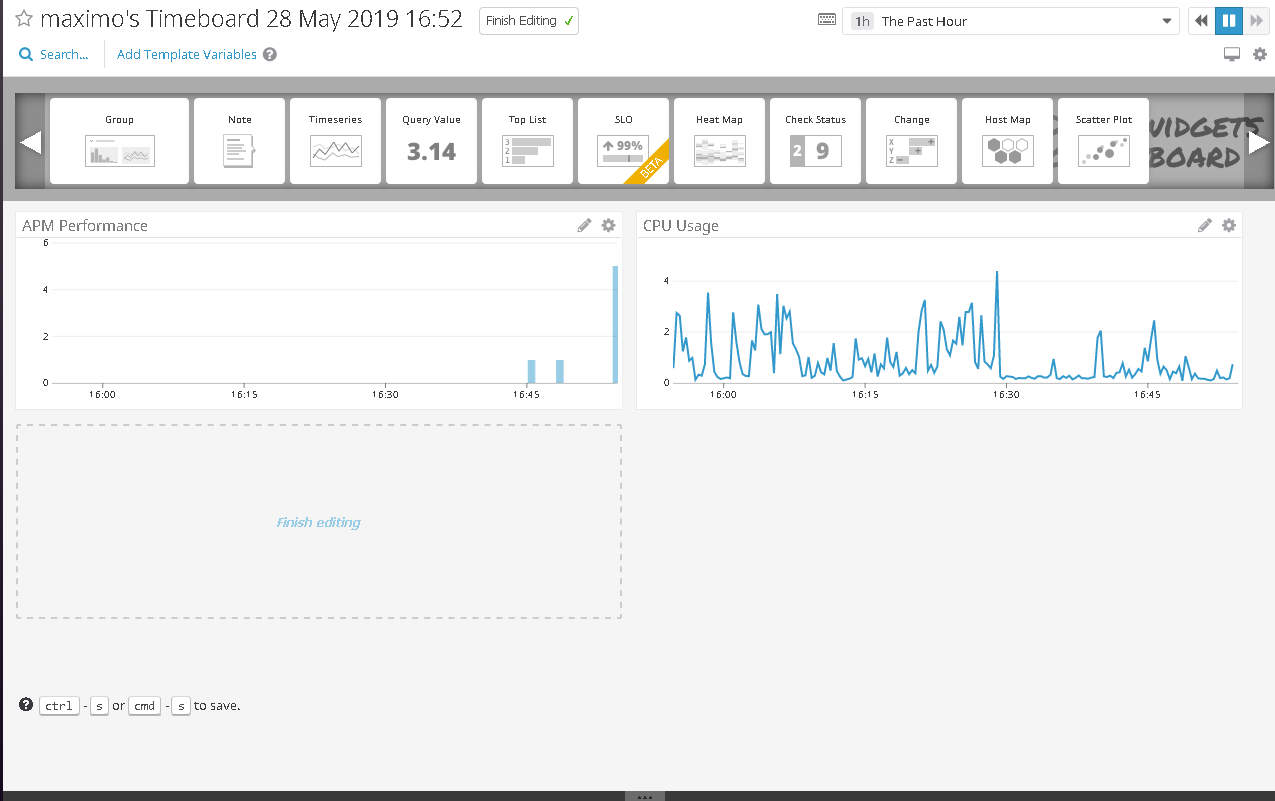






++++ During the test my VM crash and I was not able to recover the image so I rebuild a new one on Ubuntu 18

## **Collecting APM Data:**

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

A services is a set of processes that provide a collection of methods.

A resource is a query to a service like a SQL Query “Select count (\*) Transaction from appname”

## Final Question:

Datadog can have a big place in the New Generation of Straming PlatformMonitoring.

Straming platform like Netflix have alrady a big place in the Market, 2020 will be the Big Year of the Straming platform like Disney Plus( Netflix like) , and Gaming ( Xcloud of Microsoft and Stadia of Google ).

Customers would like to share content like Events, Movies, TVShow and they will need a tool to monitor

Front End-Experiance

-Latency

-Resolution

-Sound Quality

And the back-end experiance

-Publicity revenue

-Publicity Display

-Users connected

Full Stack Monitoring

…

https://app.datadoghq.com/event/stream?tags\_execution=and&show\_private=true&per\_page=30&aggregate\_up=true&use\_date\_happened=false&display\_timeline=true&from\_ts=1559063580000&priority=normal&live=true&is\_zoomed=false&status=all&to\_ts=1559067180000&is\_auto=false&incident=true&only\_discussed=false&no\_user=false&page=0&bucket\_size=60000