

Prerequisites - Setting Up The Environment

I began by trying to install VirtualBox & Vagrant. It took me a while to figure out exactly why I wasn't able to get it working - I'm using a Mac with an M1 chip. A curse of the early adopter.

The screenshot shows a Quora post with the following details:

- Title:** Can you run VirtualBox on Macs with Apple's M1 chip?
- Author:** Asheq Reza, Software Developer
- Answered:** November 28
- Content:** You unfortunately cannot run VirtualBox on Macs with Apple's M1 chip. If this will change in the future is currently unknown but it doesn't appear very likely as the Oracle has not said anything about it whereas others like VMWare Fusion and Parallels have both had support pledged for them. It definitely can be possible, it is a matter of if Oracle will update the application.
If Oracle does pledge support for VirtualBox, it is unclear what it will run exactly, x86/64 like Windows 10 or ARM only OSes (for which Microsoft could choose to support M1 – Apple has said it really is up to them).
- Comments:** 2 comments from Gary Dauphin and more
- Related Questions:** A sidebar listing related questions such as "Can you run VMware on Macs with Apple's M1 chip?", "Can you run VirtualBox on Macs with Apple's M1 chip? Is it legal to run macOS on...", etc.

Image(1) - Mac M1 chip issue

The screenshot shows a desktop environment with a browser displaying a Datadog exercise guide and a terminal window running a command:

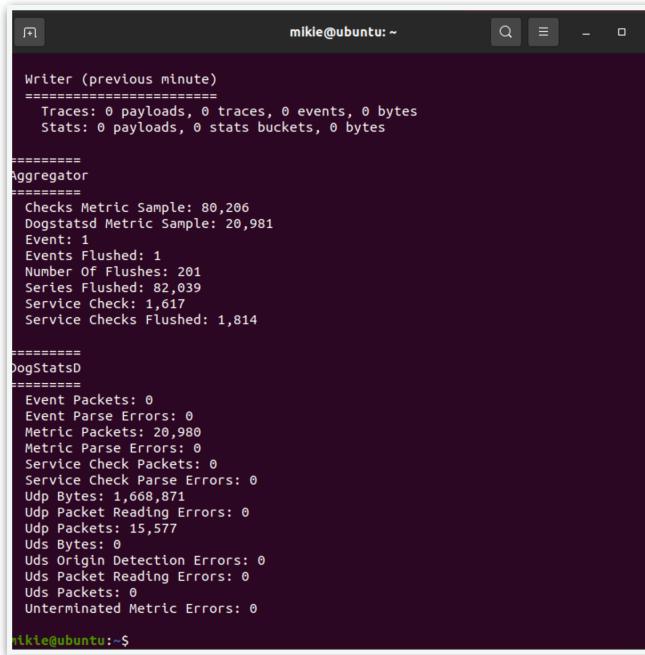
```
mikie@ubuntu:~$ sudo systemctl start datadog-agent
[sudo] password for mikie:
[  OK  ] Started Datadog Agent.
```

The exercise guide includes sections such as:

- The Exercise:** Don't forget to read the [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.
 - You can spin up a fresh linux VM via Vagrant or other tools so the [instructions](#) for setting up a Vagrant Ubuntu VM. We strongly recommend this approach.
 - You can utilize a Containerized approach with Docker for Linux and Mac OS X.
- Once this is ready, sign up for a trial Datadog at <https://www.datadoghq.com/trial>.**
- Please make sure to use "Datadog Recruiting Candidate" in the "Organization" dropdown when signing up.**
- Then, get the Agent reporting metrics from your local machine and monitor them in the Datadog interface.**
- Collecting Metrics:**
 - Add tags in the Agent config file and show us a screenshot of your configuration file.
 - Install a database on your machine (MongoDB, MySQL, or PostgreSQL).
 - Create a custom Agent check that submits a metric named my_new_metric every 10 seconds.
 - Change your check's collection interval so that it only submits the metric every 10 seconds.

Image(2) - Installation of the agent in Ubuntu VM

I tried to use Docker, but I ran into similar issues. I found articles and videos online with tutorials using Parallels Desktop to set up Ubuntu, so I decided to go with that. It was pretty easy to see up and I went with the Ubuntu 20.04 version, and installed the Datadog Agent.



```
mikle@ubuntu: ~
=====
Writer (previous minute)
=====
Traces: 0 payloads, 0 traces, 0 events, 0 bytes
Stats: 0 payloads, 0 stats buckets, 0 bytes

=====
Aggregator
=====
Checks Metric Sample: 80,206
Dogstatsd Metric Sample: 20,981
Event: 1
Events Flushed: 1
Number Of Flushes: 201
Series Flushed: 82,039
Service Check: 1,617
Service Checks Flushed: 1,814

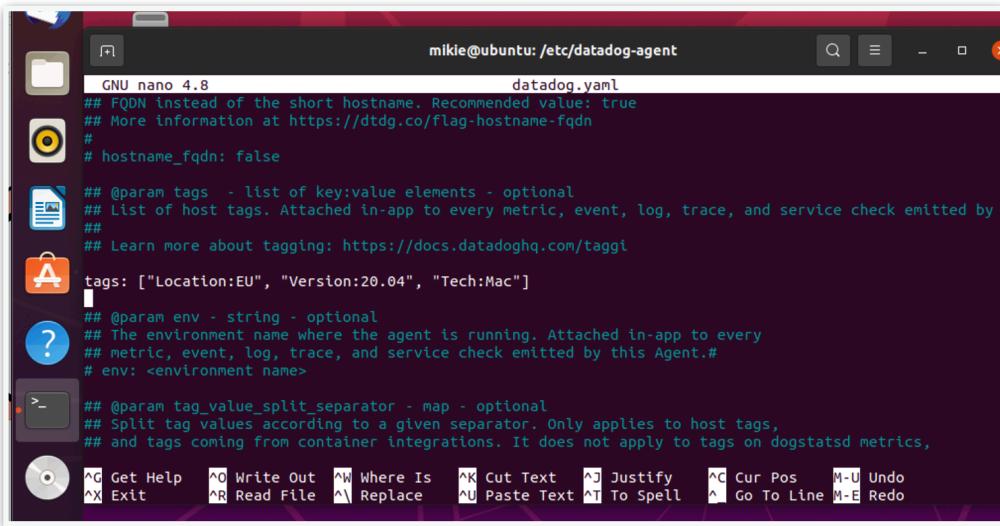
=====
DogStatsD
=====
Event Packets: 0
Event Parse Errors: 0
Metric Packets: 20,980
Metric Parse Errors: 0
Service Check Packets: 0
Service Check Parse Errors: 0
Udp Bytes: 1,668,871
Udp Packet Reading Errors: 0
Udp Packets: 15,577
Uds Bytes: 0
Uds Origin Detection Errors: 0
Uds Packet Reading Errors: 0
Uds Packets: 0
Unterminated Metric Errors: 0

mikle@ubuntu:~$
```

Image(3) - Agent Reporting Metrics, Local Machine

I was unsure of the ‘Agent Reporting Metrics’ request, so I ran ‘sudo datadog-agent status’ in the Ubuntu terminal to get the current status of the agent on my machine

Collecting Metrics:



```
GNU nano 4.8 datadog.yaml
## FQDN Instead of the short hostname. Recommended value: true
## More information at https://dtdg.co/flag-hostname-fqn
#
# hostname_fqdn: false

## @param tags - list of key:value elements - optional
## List of host tags. Attached in-app to every metric, event, log, trace, and service check emitted by
## Learn more about tagging: https://docs.datadoghq.com/taggi
tags: ["Location:EU", "Version:20.04", "Tech:Mac"]

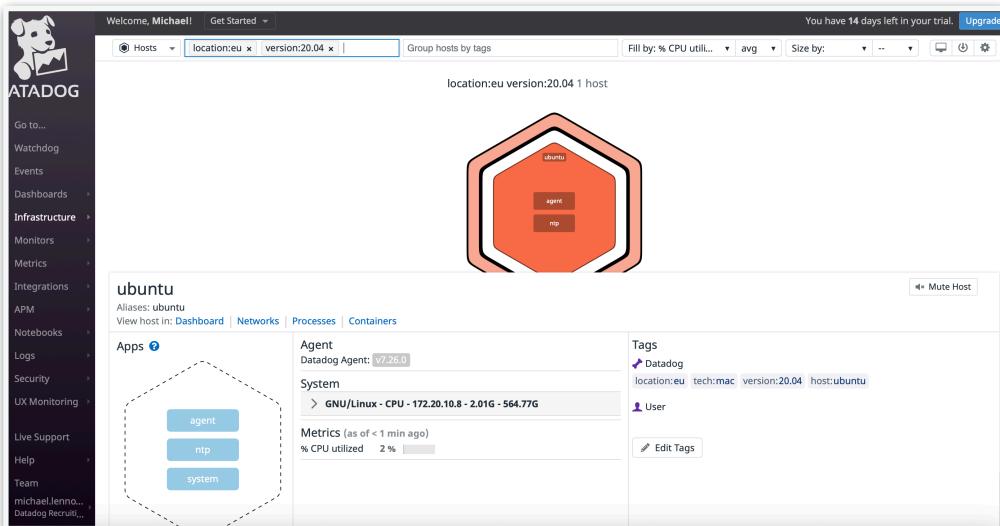
## @param env - string - optional
## The environment name where the agent is running. Attached in-app to every
## metric, event, log, trace, and service check emitted by this Agent.
## env: <environment name>

## @param tag_value_split_separator - map - optional
## Split tag values according to a given separator. Only applies to host tags,
## and tags coming from container integrations. It does not apply to tags on dogstatsd metrics,
```

Image(4) - Adding Tags to the Agent config file

I decided on the below tags that were of some relevance to where I was at the time:

- General location
- Version of Ubuntu
- Type of machine used



Welcome, Michael! Get Started | You have 14 days left in your trial! Upgrade

Hosts | location:eu | version:20.04 | Group hosts by tags | Fill by: % CPU util... avg Size by: | Mute Host

location:eu version:20.04 1 host

ubuntu

Aliases: ubuntu View host in: Dashboard | Networks | Processes | Containers

Apps (1)

Agent Datadog Agent: v7.26.0

System > GNU/Linux - CPU - 172.20.10.8 - 2.016 - 564.77G

Metrics (as of < 1 min ago)
% CPU utilized 2%

Tags

Datadog
location:eu tech:mac version:20.04 host:ubuntu

User

Edit Tags

Image(5) - View from Host Map

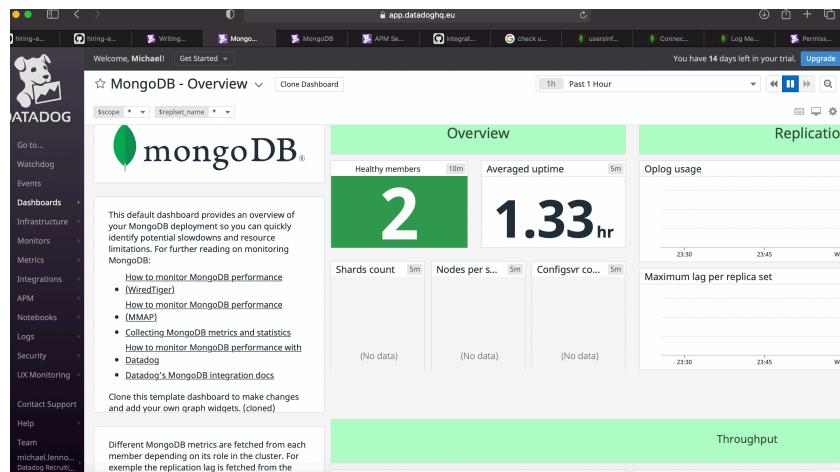
```

Activities Terminal Mar 17 00:25
mikie@ubuntu: /etc/datadog-agent/conf.d/mongo.d
=====
mongo
-----
  - Type: file
    Path: /var/log/mongodb/mongod.log
    Status: Error: open /var/log/mongodb/mongod.log: permission denied
    BytesRead: 0
    EncodedBytesSent: 28
    LogProcessed: 0
    LogsSent: 0
  mongo
  -----
  mongo
  -----
APM Agent
-----
  Status: Running
  Pid: 208646
  Uptime: 1 minute
  Avg Alloc: 9,794 bytes
  Hostname: ubuntu
  Service: localhost:8126
  Endpoints: https://trace.agent.datadoghq.eu
  Receiver (previous minute)
  -----
  Aggregator
  -----
  DogStatsD
  -----
  Event Packets: 0
  Event Parse Errors: 0
  Metric Packets: 0
  Metric Parse Errors: 0
  Service Check Packets: 0
  Service Check Parse Errors: 0
=====

```

Image(6) - MongoDB setup

Initially I started with a MySQL database, but I kept running into errors with the ‘localhost’ vs 127.0.0.1. I tried to change all of the details and permissions for the users, but the problem persisted. Instead of digging down into all the issues and wasting time, obsessing over why it wouldn’t work, set up a MongoDB interface instead. I found it much simpler and quicker to connect to.



Image(7) - Successful MongoDB integration from Datadog UI side

```

mikie@ubuntu:/etc/datadog-agent/checks.d$ nano my_metric.yaml
instances:
  - collection_interval: 45

mikie@ubuntu:/etc/datadog-agent/checks.d$ nano my_metric.py
import random

from datadog_checks.base import AgentCheck

__version__ = "1.0.0"

class MyClass(AgentCheck):
    def check(self, instance):
        self.gauge(
            'my_metric.gauge',
            random.randint(0, 1000),
            tags=["env:dev", "metric_submission_type:gauge"])

```

Image(8) - Custom Agent Check for 'my_metric' with python class and interval at 45s backend

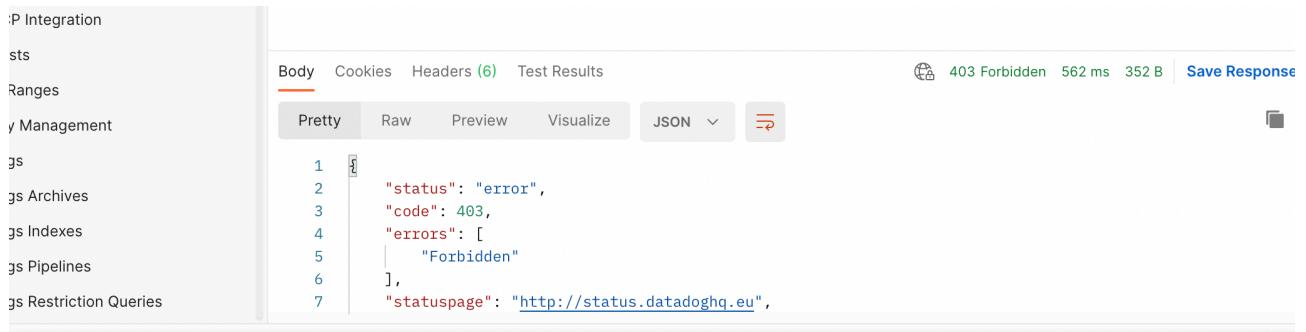
I had difficulty getting the metric generated at the start, but after combing through the documentation I found an example of 'randint' and was able to use it for the custom agent check.

TAG KEY	COUNT	TAG VALUES
env	1	env:dev
location	1	location:eu
metric_submission_t...	1	metric_submission_type:gauge
tech	1	tech:mac

Image(9) -Bonus Q - Updated collection interval via Datadog UI

Visualizing Data

Ran into authorisation issues with the API and I was unable to use Postman to use Get or Push requests.



A screenshot of the Postman application interface. The left sidebar shows various collections and environments. The main area has tabs for Body, Cookies, Headers (6), and Test Results. The Headers tab is selected, showing a status of 403 Forbidden. The Body tab displays a JSON response:

```
1 "status": "error",
2 "code": 403,
3 "errors": [
4     "Forbidden"
5 ],
6 "statuspage": "http://status.datadoghq.eu",
```

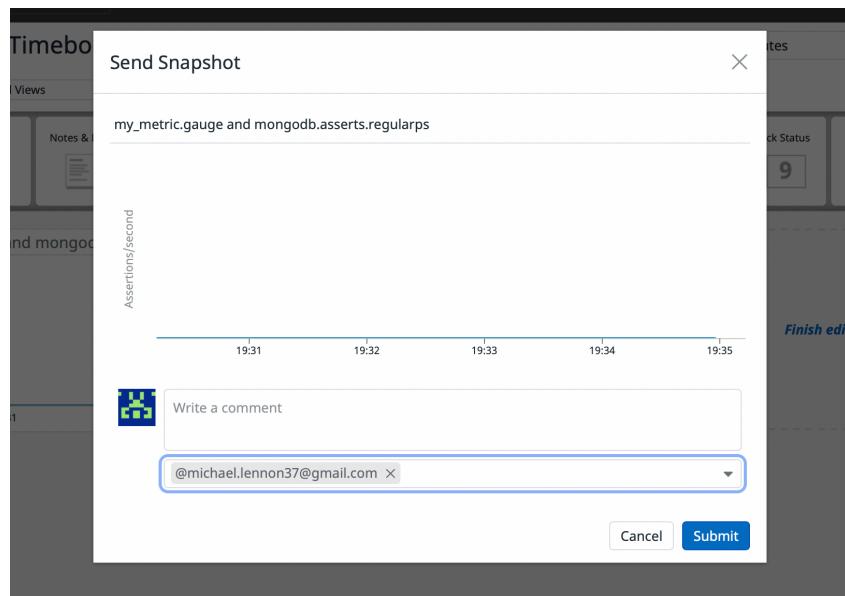
Image(10) - Forbidden Error

I followed the process to update the API key into the Datadog Authorisation environment and it did not seem to work.

I tried to validate the authorisation, but every time I received the 403 error. I tried to input the API key into the authorisation for the parent collection as well as child collections, but every time I was met with the 403 error.

I also made sure that the headers were updated to include the [datadoghq.eu](http://status.datadoghq.eu) path, but this also didn't work. I am unsure why this issue is persisting and would need to link in with somebody from the Datadog team to resolve.

Ultimately, I made a dashboard using the Datadog UI and was able to create the following snapshot and send it to myself. This is how I generated the JSON file for submission and not through the API



Image(11) - 5 minute interval snapshot sent to my email

Bonus Question: Anomaly graph is displaying a straight line, as I chose the mongodb.uptime metric to report on, which is a constant and wouldn't anomaly wouldn't be expected.

Dashboard link: <https://p.datadoghq.eu/sb/803nv4zj2tjisvyo-fed156a3ee880b1c4e82d79cecabafbf>

Monitoring Data

Created a metric monitor for 'my_metric' using the Datadog UI.

The screenshot shows the Datadog Metric Monitor configuration interface. Step 2, 'Define the metric', shows a search bar with 'Metric my_metric.gauge' and filters 'from (everywhere)' and 'avg by (everything)'. Step 3, 'Set alert conditions', shows trigger settings: 'Trigger when the metric is above the threshold on average during the last 5 minutes'. Alert threshold is set to > 800, and Warning threshold is set to > 500. A note states: 'Conditions are checked every minute regardless of the timeframe you choose'. Below this, there are sections for 'Require a full window of data for evaluation' and 'Notify if data is missing for more than 10 minutes'. A note at the bottom says: 'Note: We highly recommend you select "Do Not Require" for sparse metrics, otherwise some evaluations will be skipped.'

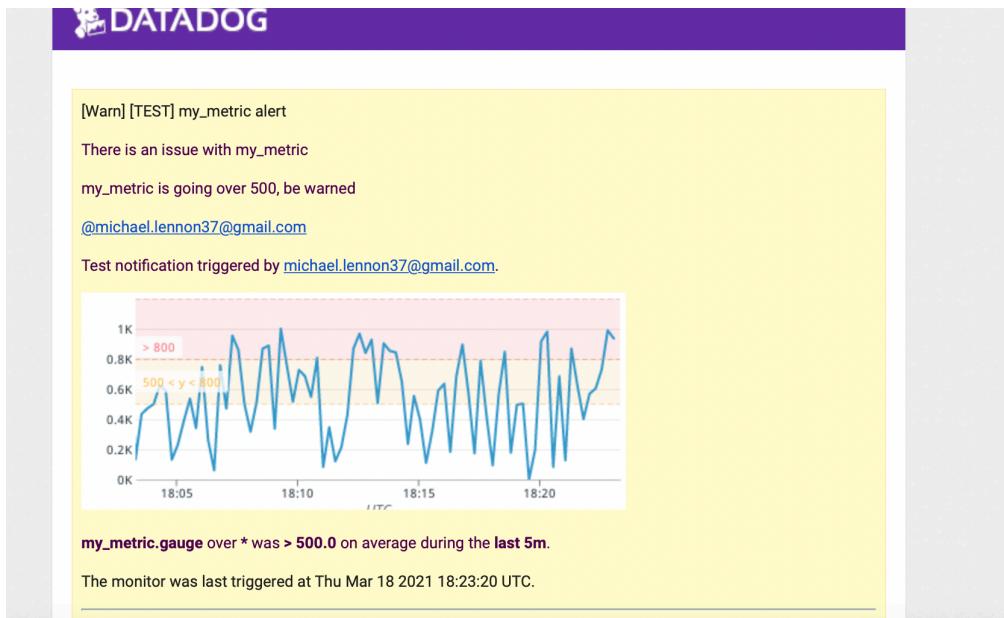
Image (12) - Metric Monitor

I updated the message so that I was notified by email based on the above criteria.

The screenshot shows the Datadog notification configuration interface for the 'my_metric.alert' monitor. It contains a preview of the notification message: 'my_metric alert {{#is_alert}} my_metric is going over 800: value {{value}} with host IP {{host.ip}} {{/is_alert}} {{#is_warning}} my_metric is going over 500, be warned{{/is_warning}} {{#is_no_data}} No data is coming through at the minute{{/is_no_data}} {{#is_recovery}} my_metric is going back to normal{{/is_recovery}}'. The message is directed to '@michael.lennon37@gmail.com'. Below the message, there are fields for 'Tags' (Select or add related tags), 'Renotify' (set to 'Never'), and 'Priority' (set to 'Not Defined').

Image(13) - Customised descriptions for notifications

The messaging was updated so that a customised message was sent, depending on the warning
Image showing the recovered metric



Image(14) - Warning message

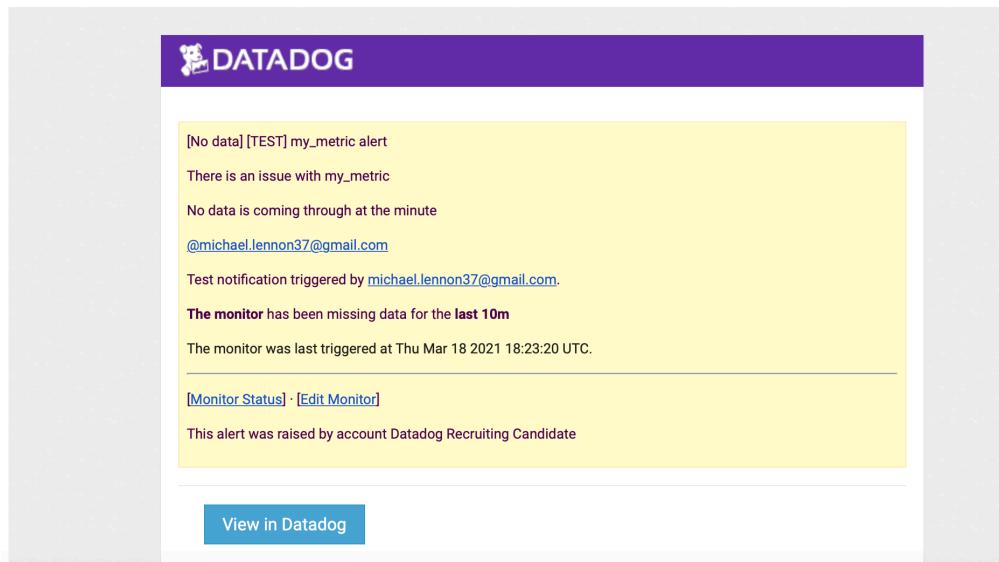


Image (15) No Data message

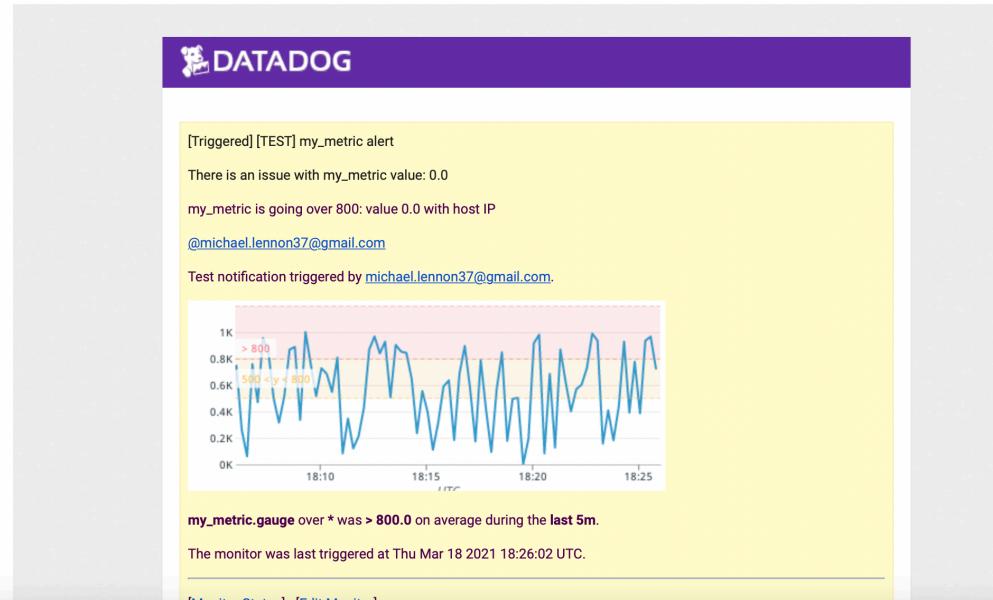


Image (16) Alert, along with the value and host IP

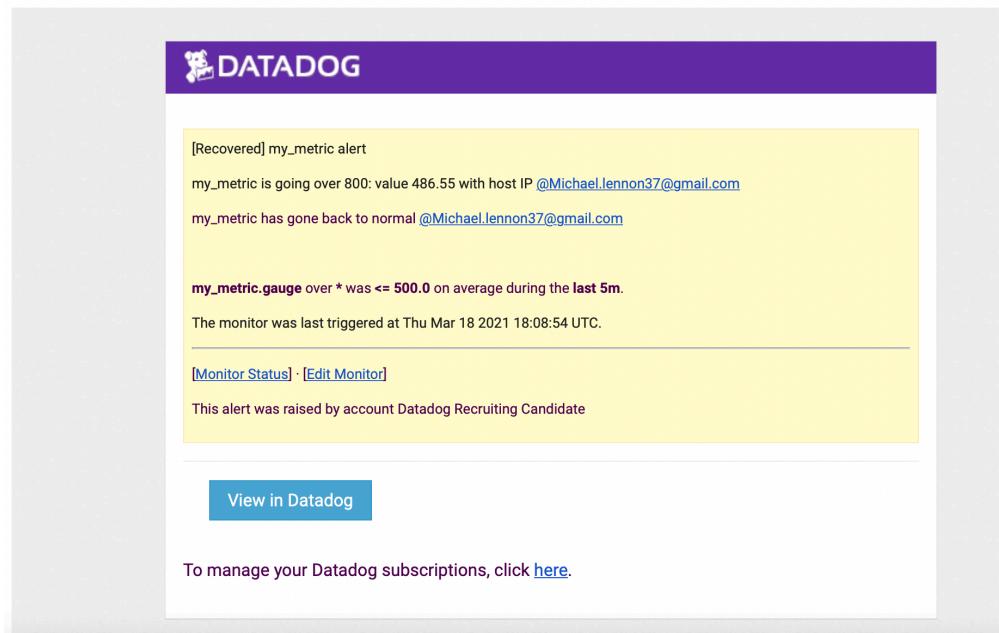


Image (17) The Recovered Notification

Bonus Question

The screenshot shows a table of scheduled downtimes. There are two entries:

STATUS	SCOPE	MONITOR TAGS	MONITOR	↓ START	END	
SCHEDULED	RECURRING *	*	my_metric alert	Mar 20, 2021 0:00 GMT	Mar 22, 2021 0:00 GMT	
ACTIVE	RECURRING *	*	my_metric alert	Mar 18, 2021 19:00 GMT	Mar 19, 2021 9:00 GMT	

Image (18) - Datadog UI with downtime scheduled

Michael Lennon (@michael.lennon37@gmail.com) mentioned you in a comment:

Michael Lennon
Michael Lennon scheduled downtime on [my_metric_alert](#) from 7:00PM on March 18 to 9:00AM UTC on March 19.
Silences from 7pm to 9am daily on M-F
[@michael.lennon37@gmail.com](#)
18 Mar, 18:38:00 UTC

[View or reply in Datadog](#)

To manage your Datadog subscriptions, click [here](#).

Image (19) - Daily downtime from 7PM - 9AM

Michael Lennon (@michael.lennon37@gmail.com) mentioned you in a comment:

Michael Lennon
Michael Lennon scheduled downtime on [my_metric_alert](#) from 12:00AM on March 20 to 12:00AM UTC on March 22.
Silences all day on Sat-Sun
[@michael.lennon37@gmail.com](#)
18 Mar, 18:39:29 UTC

[View or reply in Datadog](#)

To manage your Datadog subscriptions, click [here](#).

Image (20) - Weekend Downtime

Collecting APM Data

Link to dashboard - <https://p.datadoghq.eu/sb/803nv4zj2tjisvvo-7c9f371357eea1e9f41000a2d2c6c4d9>

Bonus Question - A service groups together endpoints, queries, or jobs for the purposes of building an application whereas a resource represent a specific domain of an application, such as a database query or a background job.

Final Question: Is there anything creative you would use Datadog for? Maybe not the most creative use, but I believe it would be of great use in a digital hospital, such as the new children's hospital in Ireland. A building with as many sensors and data inputs as is expected there would be a solid user I believe.