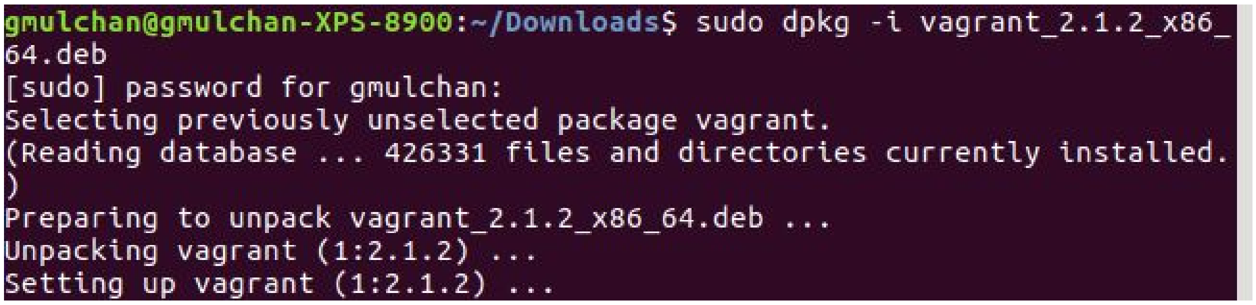
Prerequisites: Setup the environment

I have a ubuntu box and already had virtualbox installed. So decided to use vagrant to setup the environment. Instructions:

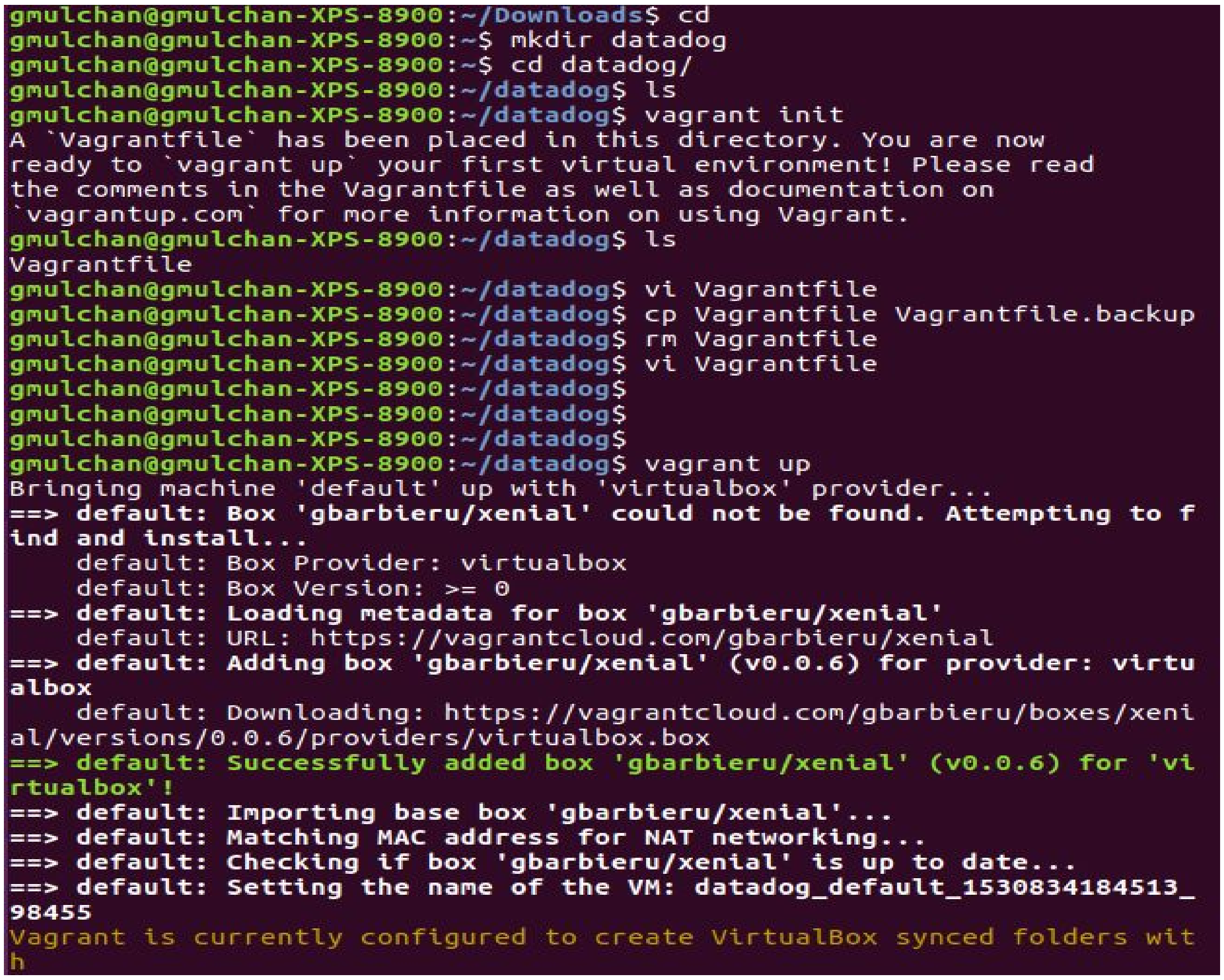
https://www.vagrantup.com/downloads.html

Screenshot:



```
gmulchan@gmulchan-XPS-8900:~/Downloads$ sudo dpkg -i vagrant_2.1.2_x86_
64.deb
[sudo] password for gmulchan:
Selecting previously unselected package vagrant.
(Reading database ... 426331 files and directories currently installed.
)
Preparing to unpack vagrant_2.1.2_x86_64.deb ...
Unpacking vagrant (1:2.1.2) ...
Setting up vagrant (1:2.1.2) ...
```

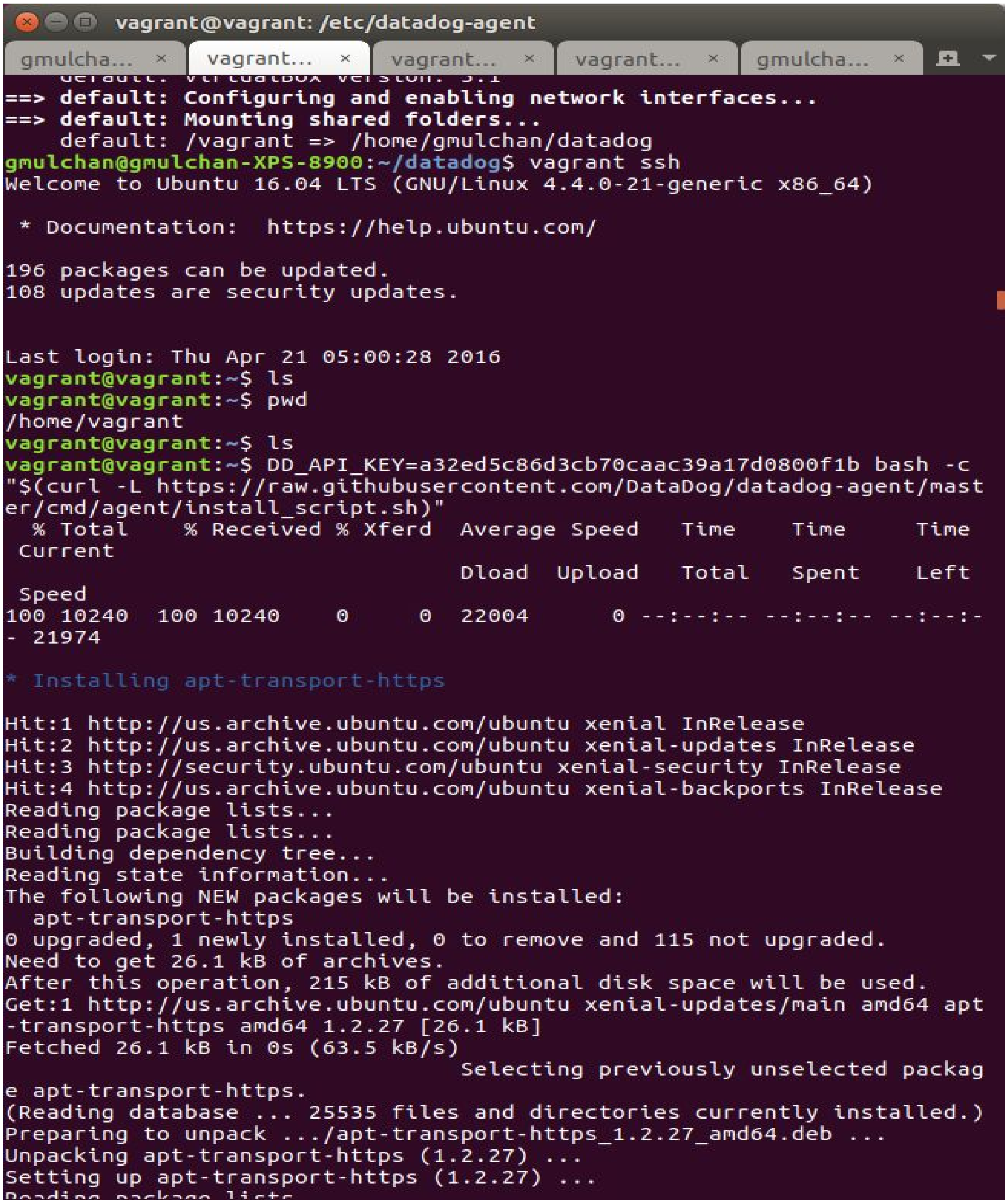
Start vagrant:



```
gmulchan@gmulchan-XPS-8900:~/Downloads$ cd
gmulchan@gmulchan-XPS-8900:~$ mkdir datadog
gmulchan@gmulchan-XPS-8900:~$ cd datadog/
gmulchan@gmulchan-XPS-8900:~/datadog$ ls
gmulchan@gmulchan-XPS-8900:~/datadog$ vagrant init
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
gmulchan@gmulchan-XPS-8900:~/datadog$ ls
Vagrantfile
gmulchan@gmulchan-XPS-8900:~/datadog$ vi Vagrantfile
gmulchan@gmulchan-XPS-8900:~/datadog$ cp Vagrantfile Vagrantfile.backup
gmulchan@gmulchan-XPS-8900:~/datadog$ rm Vagrantfile
gmulchan@gmulchan-XPS-8900:~/datadog$ vi Vagrantfile
gmulchan@gmulchan-XPS-8900:~/datadog$
gmulchan@gmulchan-XPS-8900:~/datadog$
gmulchan@gmulchan-XPS-8900:~/datadog$
gmulchan@gmulchan-XPS-8900:~/datadog$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'gbarbieru/xenial' could not be found. Attempting to f
ind and install...
    default: Box Provider: virtualbox
    default: Box Version: >= 0
==> default: Loading metadata for box 'gbarbieru/xenial'
    default: URL: https://vagrantcloud.com/gbarbieru/xenial
==> default: Adding box 'gbarbieru/xenial' (v0.0.6) for provider: virtu
albox
    default: Downloading: https://vagrantcloud.com/gbarbieru/boxes/xeni
al/versions/0.0.6/providers/virtualbox.box
==> default: Successfully added box 'gbarbieru/xenial' (v0.0.6) for 'vi
rtualbox'!
==> default: Importing base box 'gbarbieru/xenial'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'gbarbieru/xenial' is up to date...
==> default: Setting the name of the VM: datadog_default_1530834184513_
98455
Vagrant is currently configured to create VirtualBox synced folders wit
h
```

Login to Vagrant, create account at datadog and Install Datadog agent. Datadog agent already has the right keys so that it can report to the correct account, the default hostname looks good.

Screenshot of installing agent:



vagrant@vagrant: /etc/datadog-agent

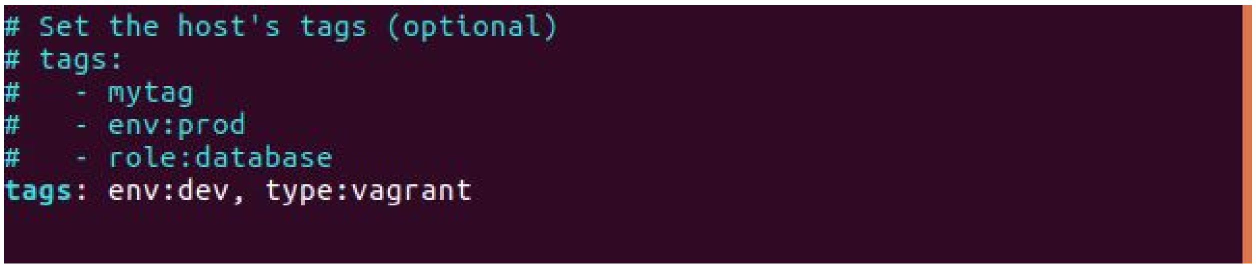
```
==> default: Configuring and enabling network interfaces...
==> default: Mounting shared folders...
    default: /vagrant => /home/gmulchan/datadog
gmulchan@gmulchan-XPS-8900:~/datadog$ vagrant ssh
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-21-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

196 packages can be updated.
108 updates are security updates.


Last login: Thu Apr 21 05:00:28 2016
vagrant@vagrant:~$ ls
vagrant@vagrant:~$ pwd
/home/vagrant
vagrant@vagrant:~$ ls
vagrant@vagrant:~$ DD_API_KEY=a32ed5c86d3cb70caac39a17d0800f1b bash -c
"$(curl -L https://raw.githubusercontent.com/DataDog/datadog-agent/mast
er/cmd/agent/install_script.sh)"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
 Current
                                 Dload  Upload   Total   Spent    Left
 Speed
100 10240  100 10240    0     0  22004      0 --:--:-- --:--:-- --:--:-
- 21974

* Installing apt-transport-https

Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 115 not upgraded.
Need to get 26.1 kB of archives.
After this operation, 215 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apt
-transport-https amd64 1.2.27 [26.1 kB]
Fetched 26.1 kB in 0s (63.5 kB/s)
                                 Selecting previously unselected packag
e apt-transport-https.
(Reading database ... 25535 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.2.27_amd64.deb ...
Unpacking apt-transport-https (1.2.27) ...
Setting up apt-transport-https (1.2.27) ...
```

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.

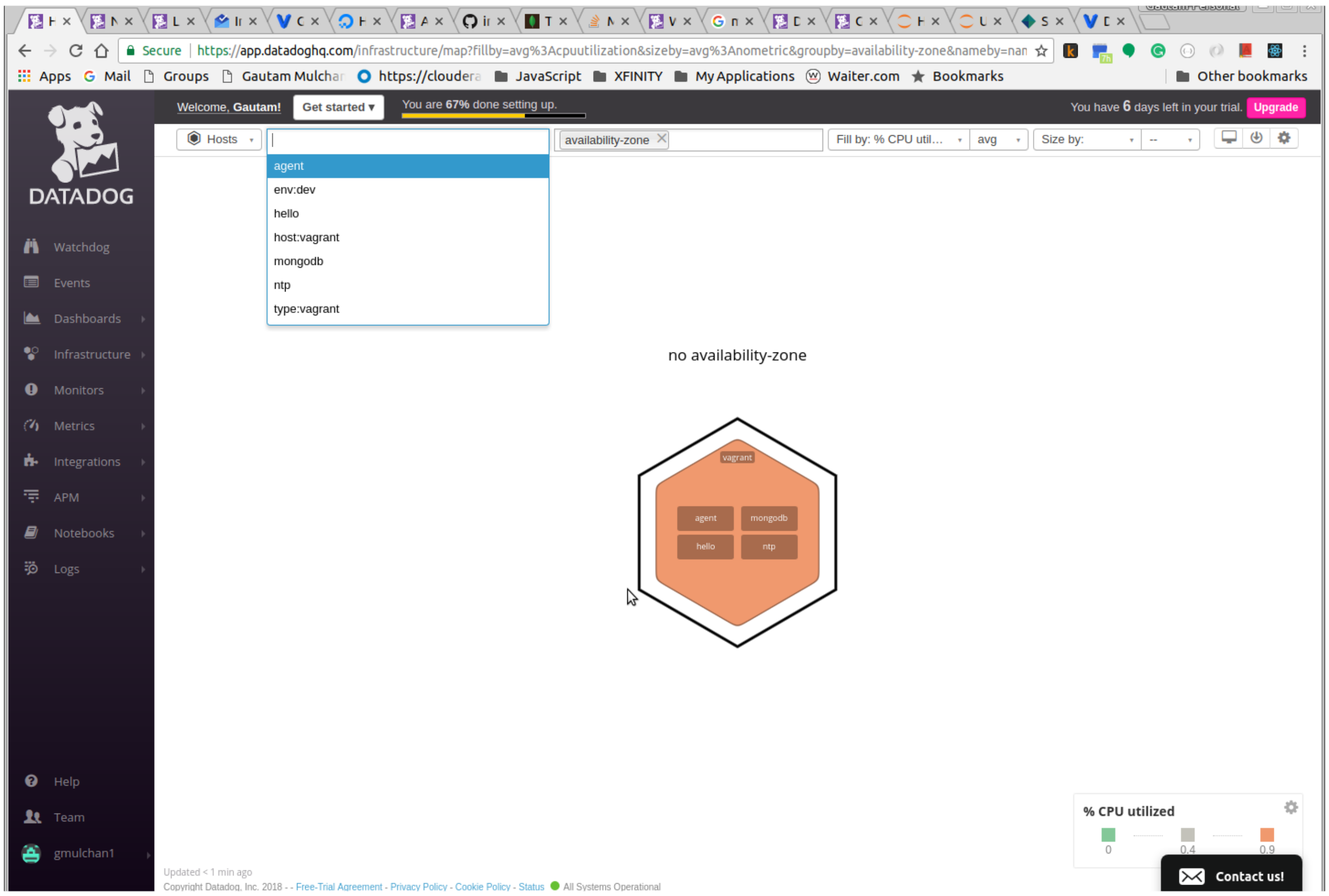
Assigning tags to components allows analytics on the reporting. One can slice and dice the reports, aggregate and group metrics based on tags (for example, dev/stage/prod envs). This is a concept from business analytics increasingly applied to operations and service management. Objective is to better understand metrics as a group of entities where the entities can be dynamically grouped using tags as a mechanism. Used mechanism of adding tags to the datadog.yaml file. Screenshot below:

https://docs.datadoghq.com/getting_started/tagging/assigning_tags/



```
# Set the host's tags (optional)
# tags:
#   - mytag
#   - env:prod
#   - role:database
tags: env:dev, type:vagrant
```

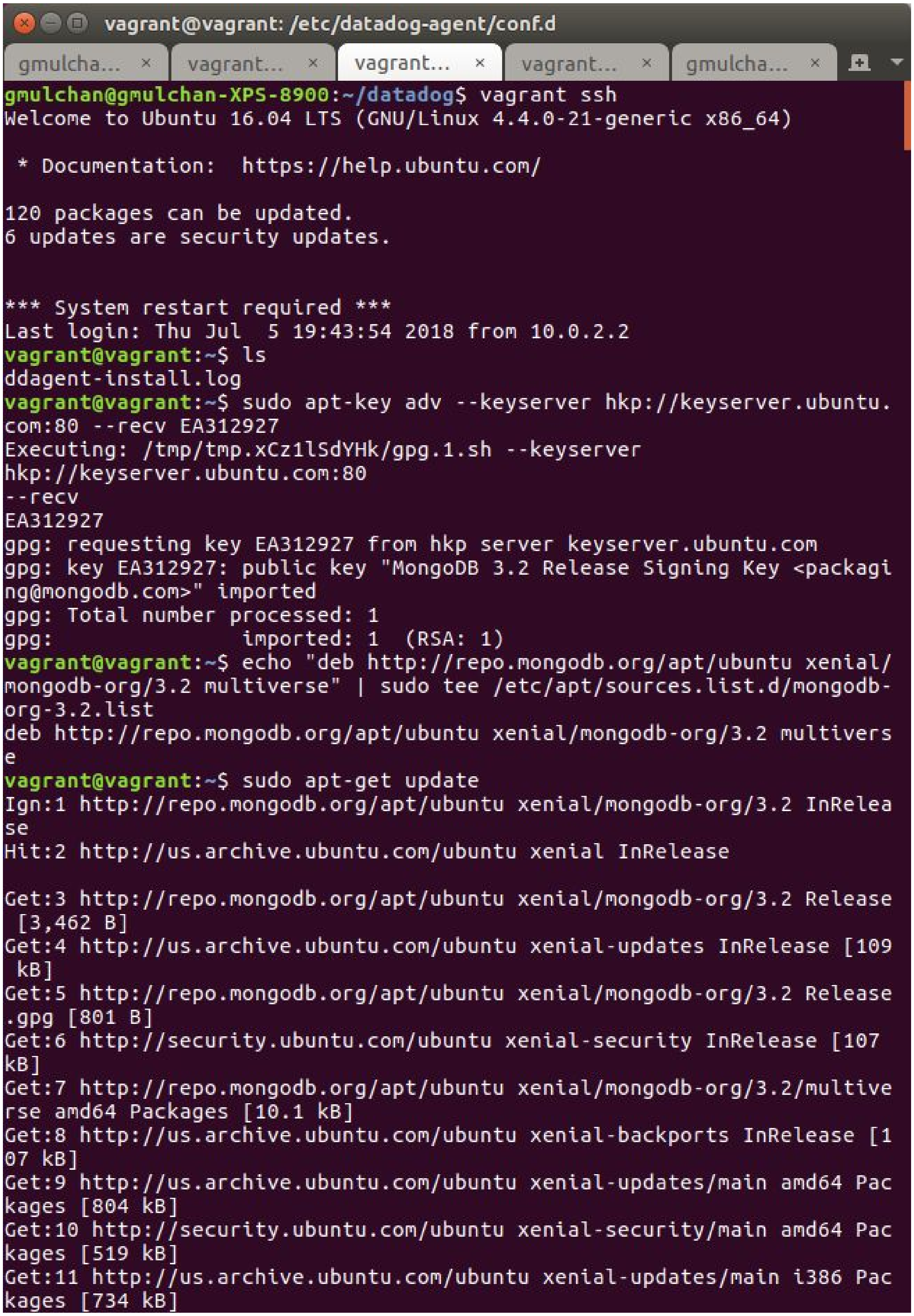
And the tag shows up in datadog UI making it easier to search:



In this case there is only one host so benefits of tagging is only for searching. With larger amounts of data, tagging helps in reporting and analytics.

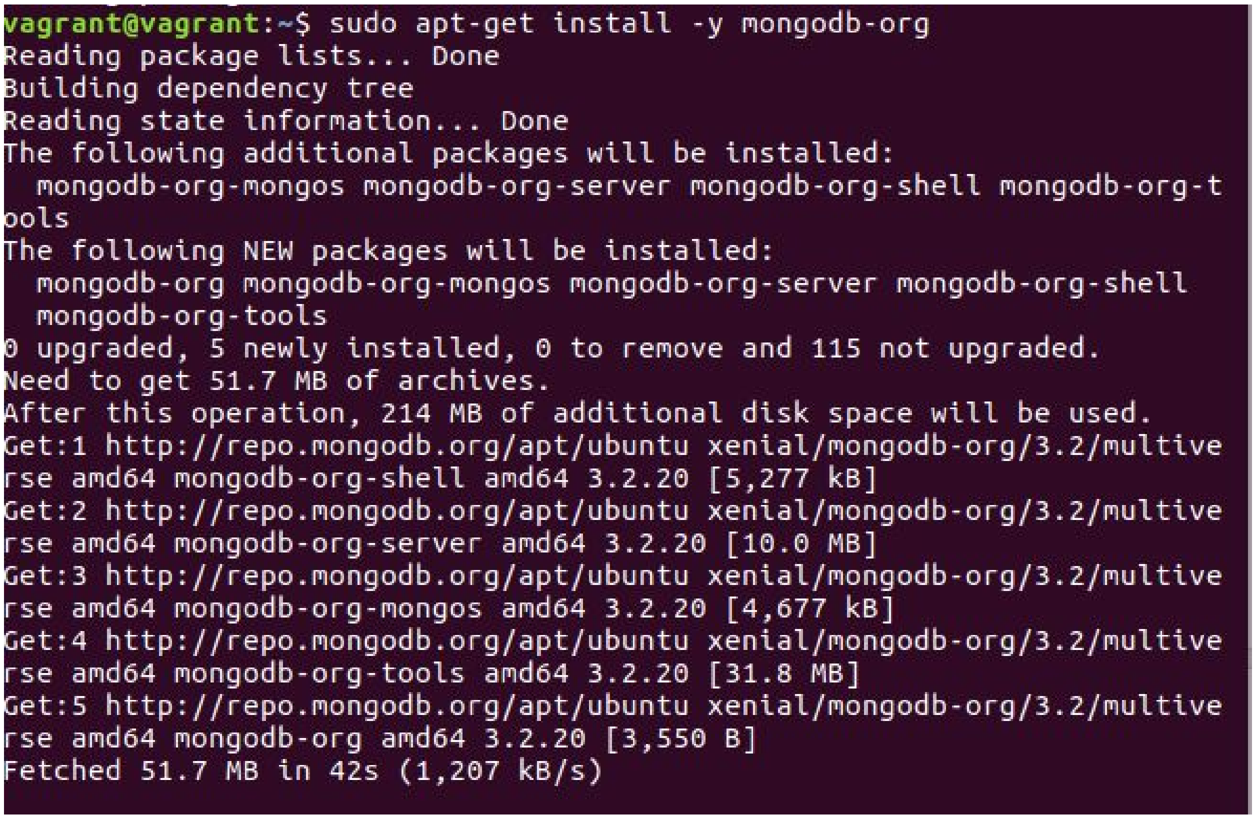
- Install a database on your machine (MongoDB, MySQL, or PostgreSQL) and then install the respective Datadog integration for that database.

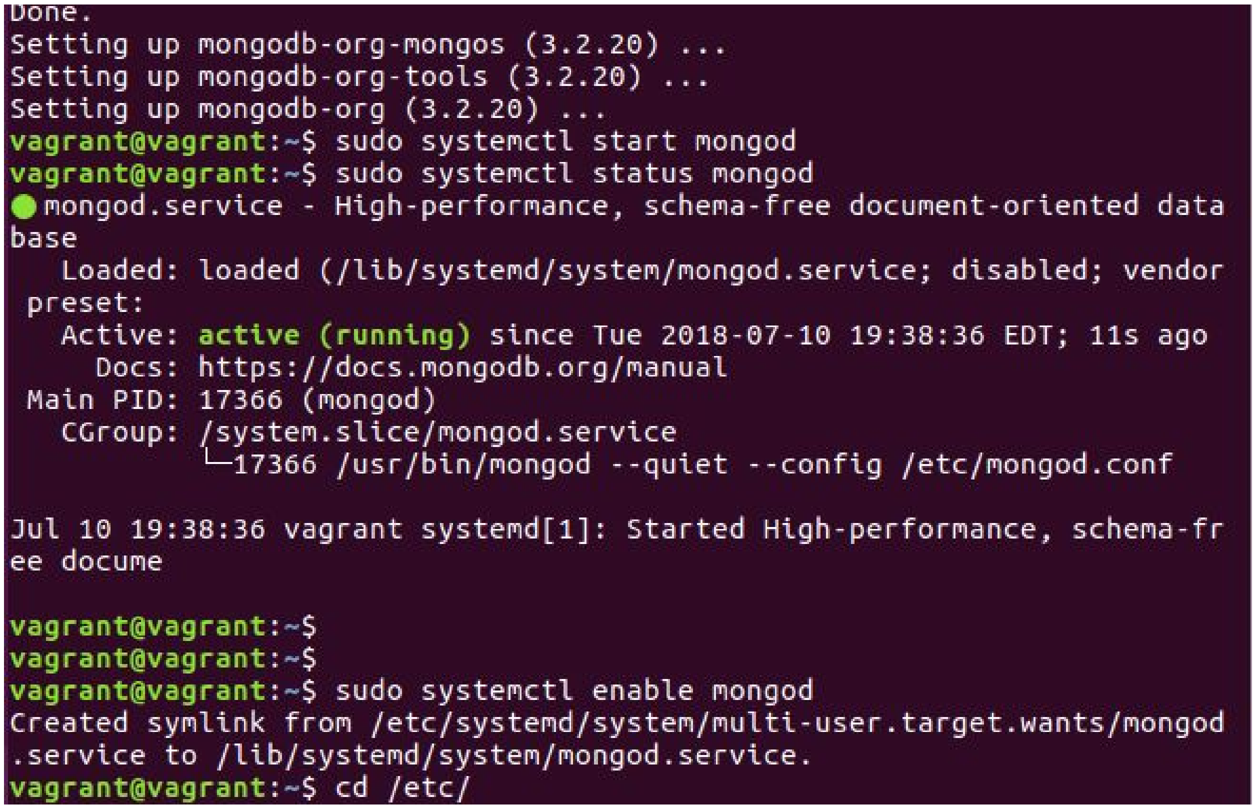
Installed mongodb on the vagrant host.



vagrant@vagrant: /etc/datadog-agent/conf.d

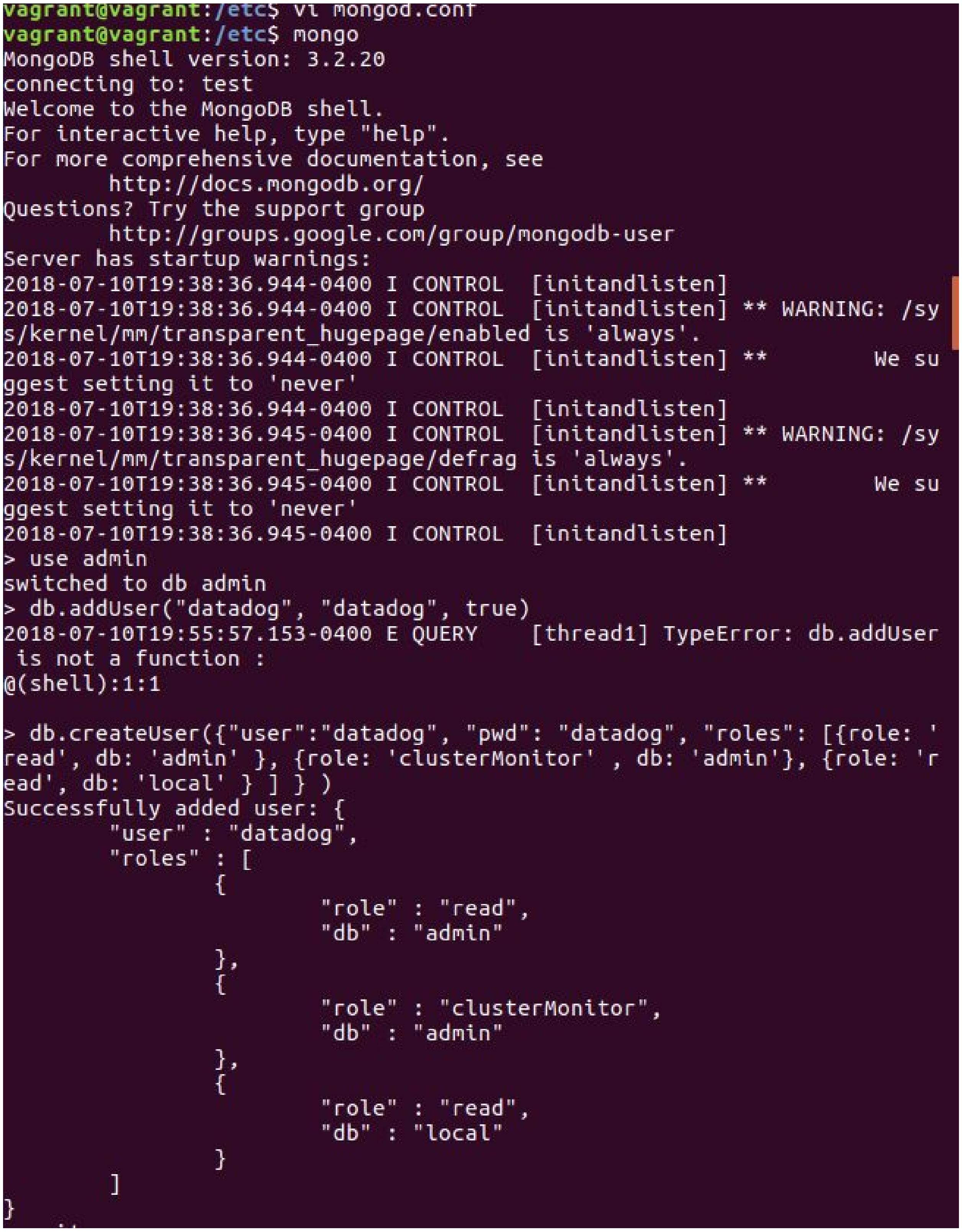
```
gmulchan@gmulchan-XPS-8900:~/datadog$ vagrant ssh
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-21-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

120 packages can be updated.
6 updates are security updates.


*** System restart required ***
Last login: Thu Jul  5 19:43:54 2018 from 10.0.2.2
vagrant@vagrant:~$ ls
ddagent-install.log
vagrant@vagrant:~$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.
com:80 --recv EA312927
Executing: /tmp/tmp.xCz1lSdYHk/gpg.1.sh --keyserver
hkp://keyserver.ubuntu.com:80
--recv
EA312927
gpg: requesting key EA312927 from hkp server keyserver.ubuntu.com
gpg: key EA312927: public key "MongoDB 3.2 Release Signing Key <packagi
ng@mongodb.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
vagrant@vagrant:~$ echo "deb http://repo.mongodb.org/apt/ubuntu xenial/
mongodb-org/3.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-
org-3.2.list
deb http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2 multivers
e
vagrant@vagrant:~$ sudo apt-get update
Ign:1 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2 InRelea
se
Hit:2 http://us.archive.ubuntu.com/ubuntu xenial InRelease

Get:3 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2 Release
 [3,462 B]
Get:4 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [109
 kB]
Get:5 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2 Release
.gpg [801 B]
Get:6 http://security.ubuntu.com/ubuntu xenial-security InRelease [107
kB]
Get:7 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2/multive
rse amd64 Packages [10.1 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [1
07 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 Pac
kages [804 kB]
Get:10 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Pac
kages [519 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu xenial-updates/main i386 Pac
kages [734 kB]
```


```
vagrant@vagrant:~$ sudo apt-get install -y mongodb-org
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  mongodb-org-mongos mongodb-org-server mongodb-org-shell mongodb-org-t
ools
The following NEW packages will be installed:
  mongodb-org mongodb-org-mongos mongodb-org-server mongodb-org-shell
  mongodb-org-tools
0 upgraded, 5 newly installed, 0 to remove and 115 not upgraded.
Need to get 51.7 MB of archives.
After this operation, 214 MB of additional disk space will be used.
Get:1 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2/multive
rse amd64 mongodb-org-shell amd64 3.2.20 [5,277 kB]
Get:2 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2/multive
rse amd64 mongodb-org-server amd64 3.2.20 [10.0 MB]
Get:3 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2/multive
rse amd64 mongodb-org-mongos amd64 3.2.20 [4,677 kB]
Get:4 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2/multive
rse amd64 mongodb-org-tools amd64 3.2.20 [31.8 MB]
Get:5 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2/multive
rse amd64 mongodb-org amd64 3.2.20 [3,550 B]
Fetched 51.7 MB in 42s (1,207 kB/s)
```

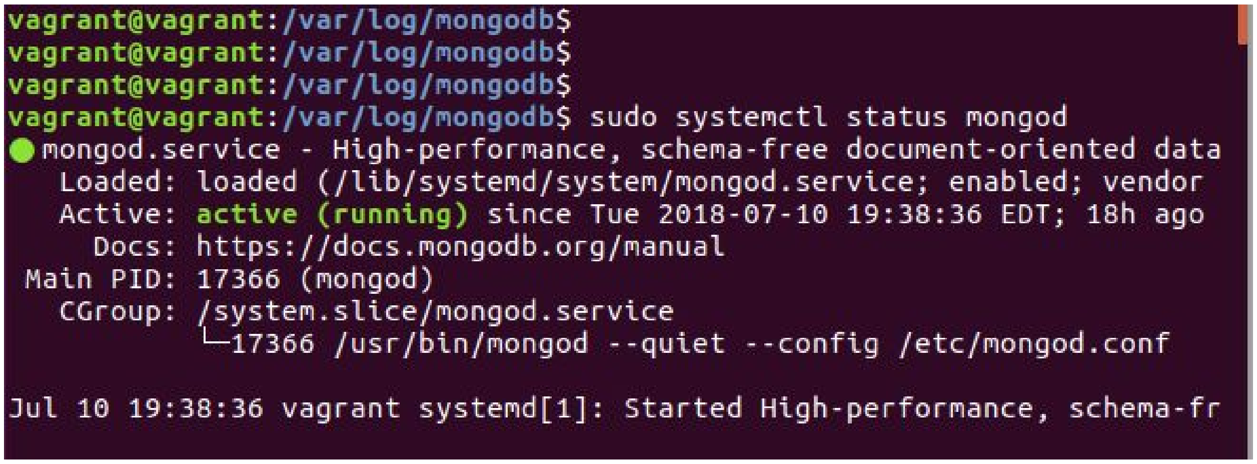


```
Done.
Setting up mongodb-org-mongos (3.2.20) ...
Setting up mongodb-org-tools (3.2.20) ...
Setting up mongodb-org (3.2.20) ...
vagrant@vagrant:~$ sudo systemctl start mongod
vagrant@vagrant:~$ sudo systemctl status mongod
● mongod.service - High-performance, schema-free document-oriented data
base
   Loaded: loaded (/lib/systemd/system/mongod.service; disabled; vendor
 preset:
   Active: active (running) since Tue 2018-07-10 19:38:36 EDT; 11s ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 17366 (mongod)
   CGroup: /system.slice/mongod.service
           └─17366 /usr/bin/mongod --quiet --config /etc/mongod.conf

Jul 10 19:38:36 vagrant systemd[1]: Started High-performance, schema-fr
ee docume

vagrant@vagrant:~$
vagrant@vagrant:~$
vagrant@vagrant:~$ sudo systemctl enable mongod
Created symlink from /etc/systemd/system/multi-user.target.wants/mongod
.service to /lib/systemd/system/mongod.service.
vagrant@vagrant:~$ cd /etc/
```

Enabled access from datadog agent to query mongodb to collect metrics:

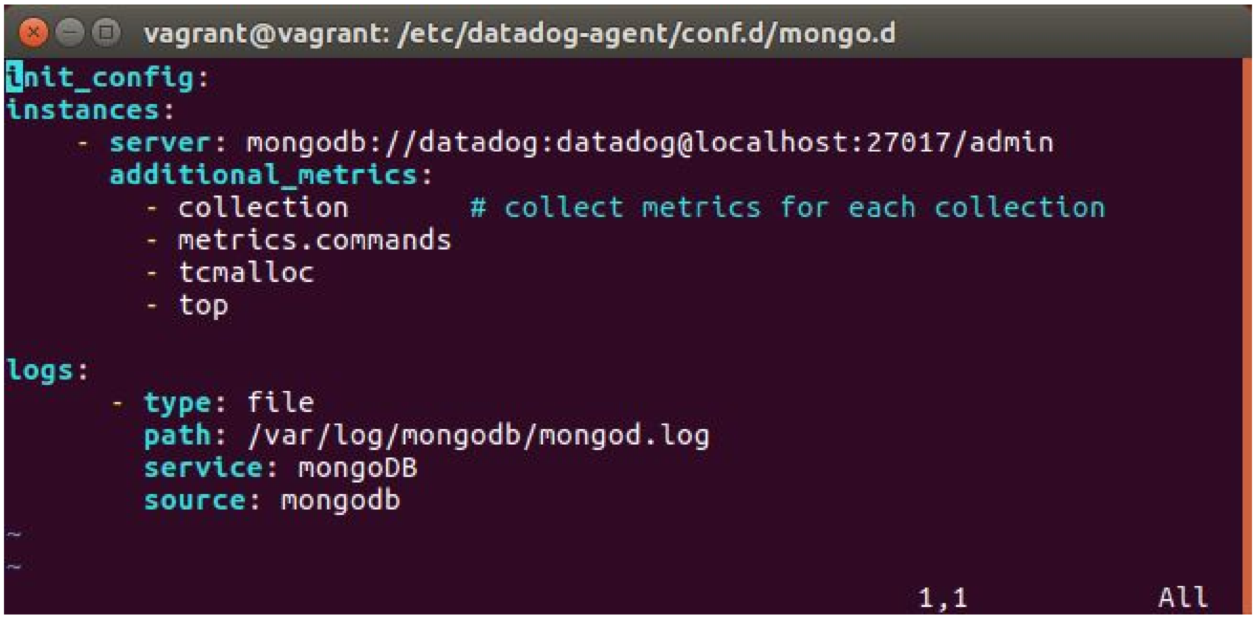


```
vagrant@vagrant:/etc$ vi mongod.conf
vagrant@vagrant:/etc$ mongo
MongoDB shell version: 3.2.20
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
Questions? Try the support group
        http://groups.google.com/group/mongodb-user
Server has startup warnings:
2018-07-10T19:38:36.944-0400 I CONTROL  [initandlisten]
2018-07-10T19:38:36.944-0400 I CONTROL  [initandlisten] ** WARNING: /sy
s/kernel/mm/transparent_hugepage/enabled is 'always'.
2018-07-10T19:38:36.944-0400 I CONTROL  [initandlisten] **        We su
ggest setting it to 'never'
2018-07-10T19:38:36.944-0400 I CONTROL  [initandlisten]
2018-07-10T19:38:36.945-0400 I CONTROL  [initandlisten] ** WARNING: /sy
s/kernel/mm/transparent_hugepage/defrag is 'always'.
2018-07-10T19:38:36.945-0400 I CONTROL  [initandlisten] **        We su
ggest setting it to 'never'
2018-07-10T19:38:36.945-0400 I CONTROL  [initandlisten]
> use admin
switched to db admin
> db.addUser("datadog", "datadog", true)
2018-07-10T19:55:57.153-0400 E QUERY    [thread1] TypeError: db.addUser
 is not a function :
@(shell):1:1

> db.createUser({"user":"datadog", "pwd": "datadog", "roles": [{role: '
read', db: 'admin' }, {role: 'clusterMonitor' , db: 'admin'}, {role: 'r
ead', db: 'local' } ] } )
Successfully added user: {
        "user" : "datadog",
        "roles" : [
                {
                        "role" : "read",
                        "db" : "admin"
                },
                {
                        "role" : "clusterMonitor",
                        "db" : "admin"
                },
                {
                        "role" : "read",
                        "db" : "local"
                }
        ]
}
```



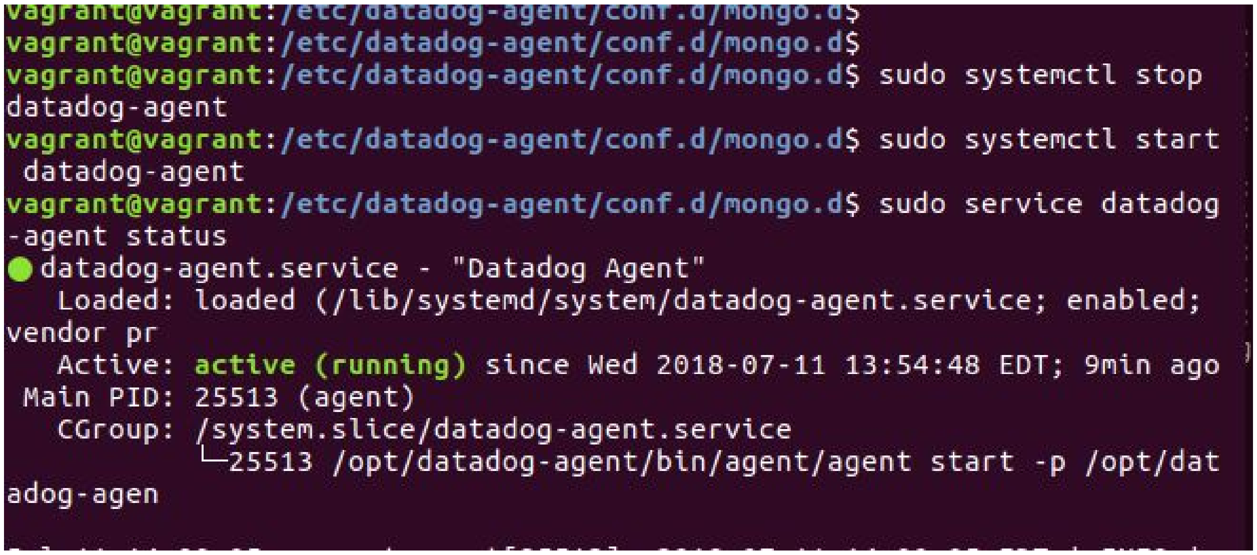
```
vagrant@vagrant:/var/log/mongodb$
vagrant@vagrant:/var/log/mongodb$
vagrant@vagrant:/var/log/mongodb$
vagrant@vagrant:/var/log/mongodb$ sudo systemctl status mongod
● mongod.service - High-performance, schema-free document-oriented data
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor
   Active: active (running) since Tue 2018-07-10 19:38:36 EDT; 18h ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 17366 (mongod)
   CGroup: /system.slice/mongod.service
           └─17366 /usr/bin/mongod --quiet --config /etc/mongod.conf

Jul 10 19:38:36 vagrant systemd[1]: Started High-performance, schema-fr
```

Change the config in the mongodb config file to monitor the local mongodb instance. Also enabled logs monitoring.



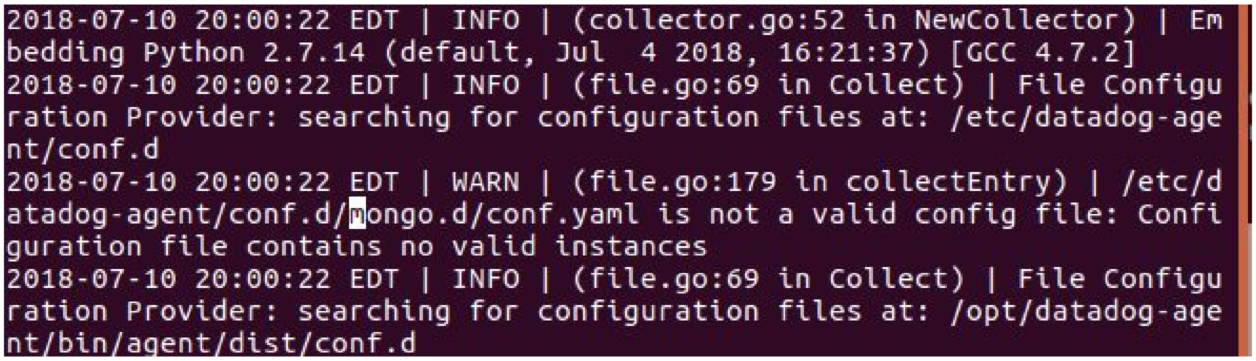
```
vagrant@vagrant: /etc/datadog-agent/conf.d/mongo.d
init_config:
instances:
    - server: mongodb://datadog:datadog@localhost:27017/admin
      additional_metrics:
        - collection       # collect metrics for each collection
        - metrics.commands
        - tcmalloc
        - top

logs:
      - type: file
        path: /var/log/mongodb/mongod.log
        service: mongoDB
        source: mongodb
~
~
                                                     1,1           All
```

Restarted agent after enabling mongodb monitoring. Hopefully should not need to restart agent after enabling/disabling any component. Agent should pick up the changes in config automatically. This needs to be done for every step.



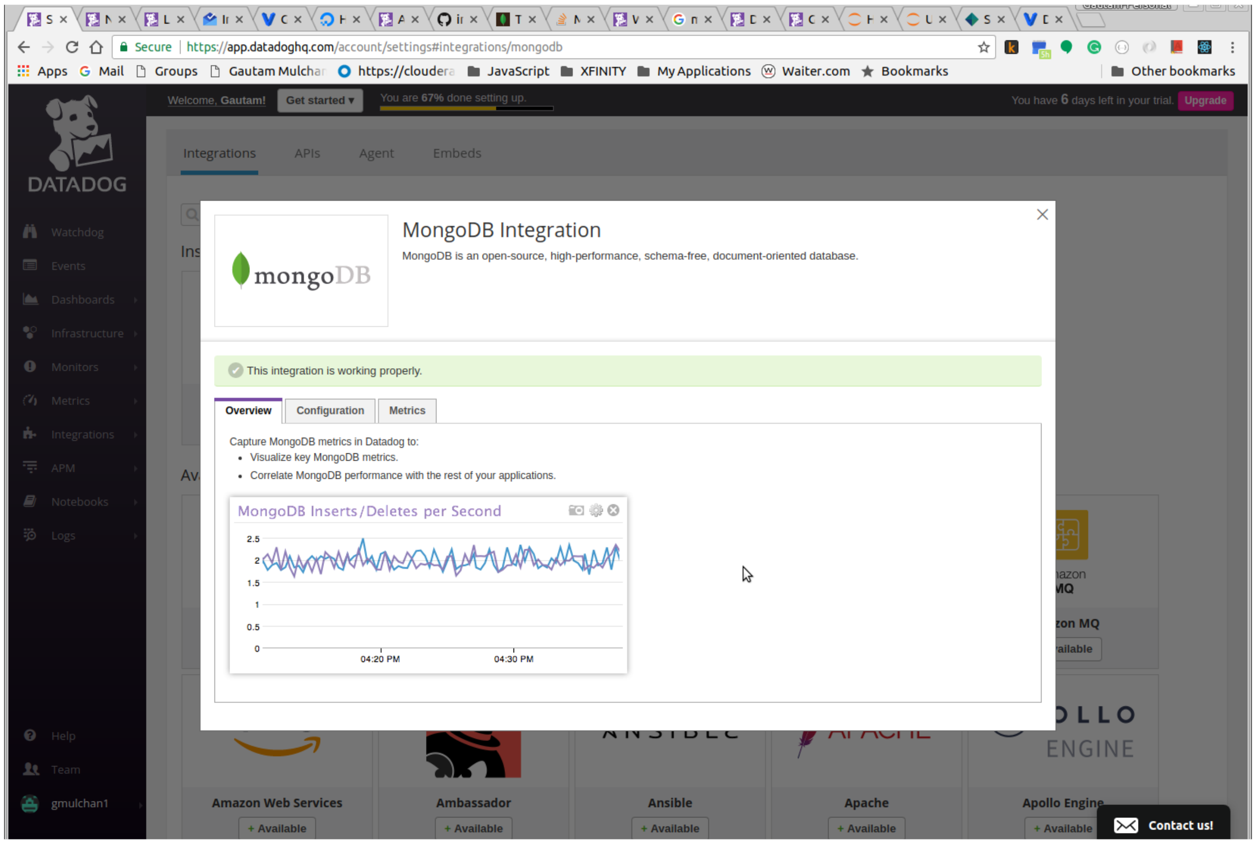
```
vagrant@vagrant:/etc/datadog-agent/conf.d/Mongo.d$
vagrant@vagrant:/etc/datadog-agent/conf.d/mongo.d$
vagrant@vagrant:/etc/datadog-agent/conf.d/mongo.d$ sudo systemctl stop
datadog-agent
vagrant@vagrant:/etc/datadog-agent/conf.d/mongo.d$ sudo systemctl start
 datadog-agent
vagrant@vagrant:/etc/datadog-agent/conf.d/mongo.d$ sudo service datadog
-agent status
● datadog-agent.service - "Datadog Agent"
   Loaded: loaded (/lib/systemd/system/datadog-agent.service; enabled;
vendor pr
   Active: active (running) since Wed 2018-07-11 13:54:48 EDT; 9min ago
 Main PID: 25513 (agent)
   CGroup: /system.slice/datadog-agent.service
           └─25513 /opt/datadog-agent/bin/agent/agent start -p /opt/dat
adog-agen
```

Mongodb metrics not showing up in datadog UI, looked at datadog agent log and saw an error:



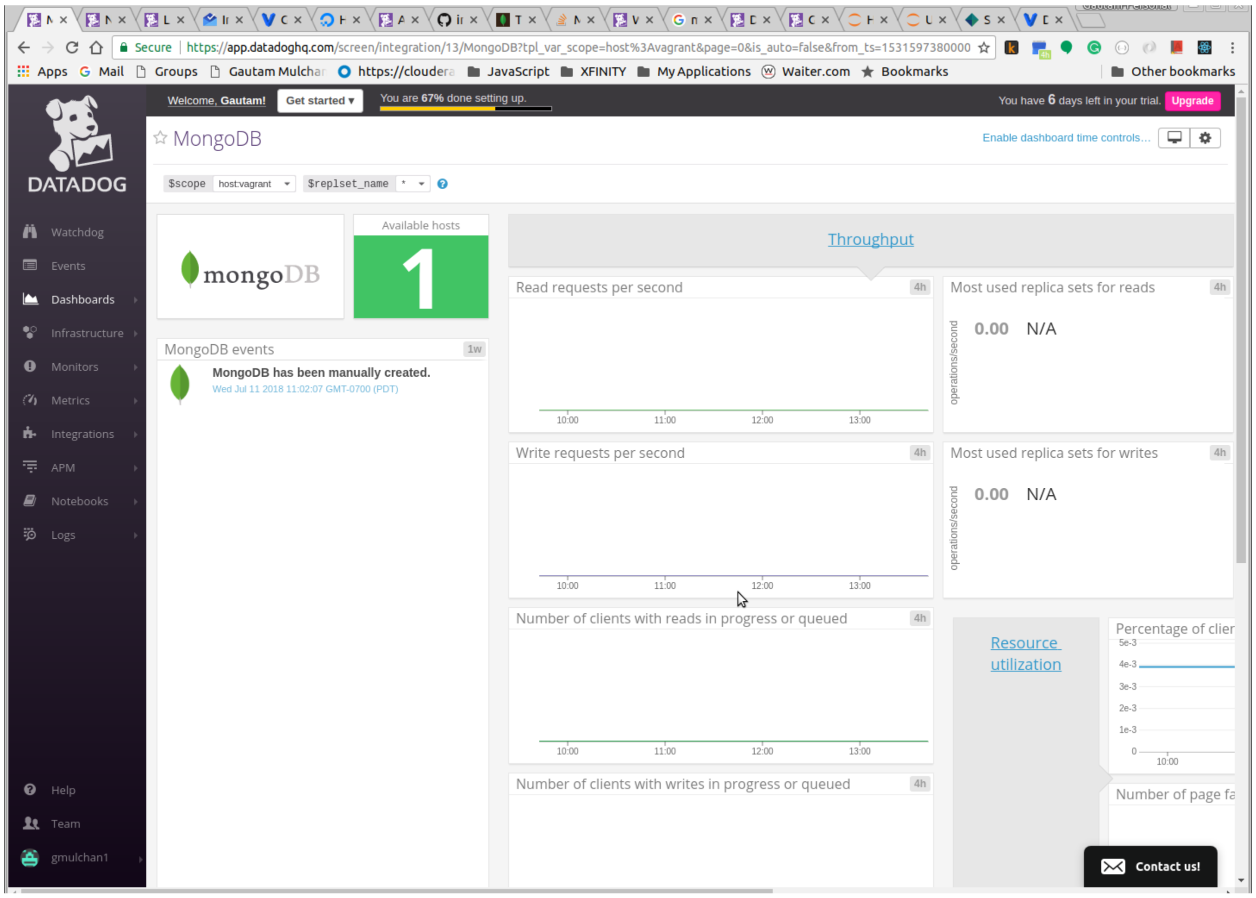
```
2018-07-10 20:00:22 EDT | INFO | (collector.go:52 in NewCollector) | Em
bedding Python 2.7.14 (default, Jul  4 2018, 16:21:37) [GCC 4.7.2]
2018-07-10 20:00:22 EDT | INFO | (file.go:69 in Collect) | File Configu
ration Provider: searching for configuration files at: /etc/datadog-age
nt/conf.d
2018-07-10 20:00:22 EDT | WARN | (file.go:179 in collectEntry) | /etc/d
atadog-agent/conf.d/mongo.d/conf.yaml is not a valid config file: Confi
guration file contains no valid instances
2018-07-10 20:00:22 EDT | INFO | (file.go:69 in Collect) | File Configu
ration Provider: searching for configuration files at: /opt/datadog-age
nt/bin/agent/dist/conf.d
```

Turned out a typo in the yaml file. In meantime also enabled the mongodb integration.


MongoDB Integration
MongoDB is an open-source, high-performance, schema-free, document-oriented database.
This integration is working properly.
Overview
Configuration
Metrics
Capture MongoDB metrics in Datadog to:
Visualize key MongoDB metrics.
Correlate MongoDB performance with the rest of your applications.
MongoDB Inserts/Deletes per Second

It is unclear what integrations (in the UI) do. The documentation lists the over 200 integrations but does not explain what specifically is the purpose of integrations and how exactly the integration helps from a use case perspective.
https://docs.datadoghq.com/integrations/

Here is mongodb dashboard:

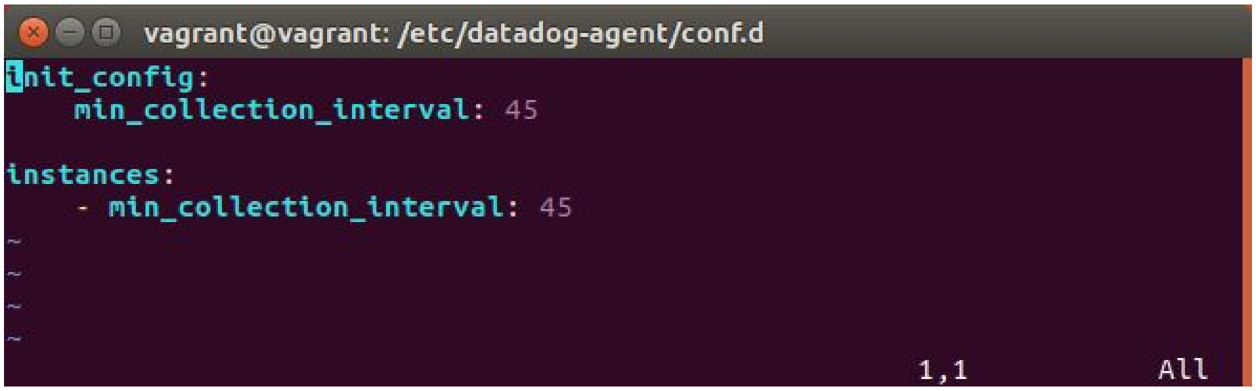

MongoDB
$scope host:vagrant
$replset_name *
Available hosts
1
Throughput
Read requests per second
Most used replica sets for reads
0.00 N/A
Write requests per second
Most used replica sets for writes
0.00 N/A
Number of clients with reads in progress or queued
Number of clients with writes in progress or queued
Resource utilization
MongoDB events
Contact us!

MongoDB has been manually created.
Wed Jul 11 2018 11:02:07 GMT-0700 (PDT)

Overall, very easy to setup monitoring for various infrastructure components with just a small config change in the agent and everything else in UI.

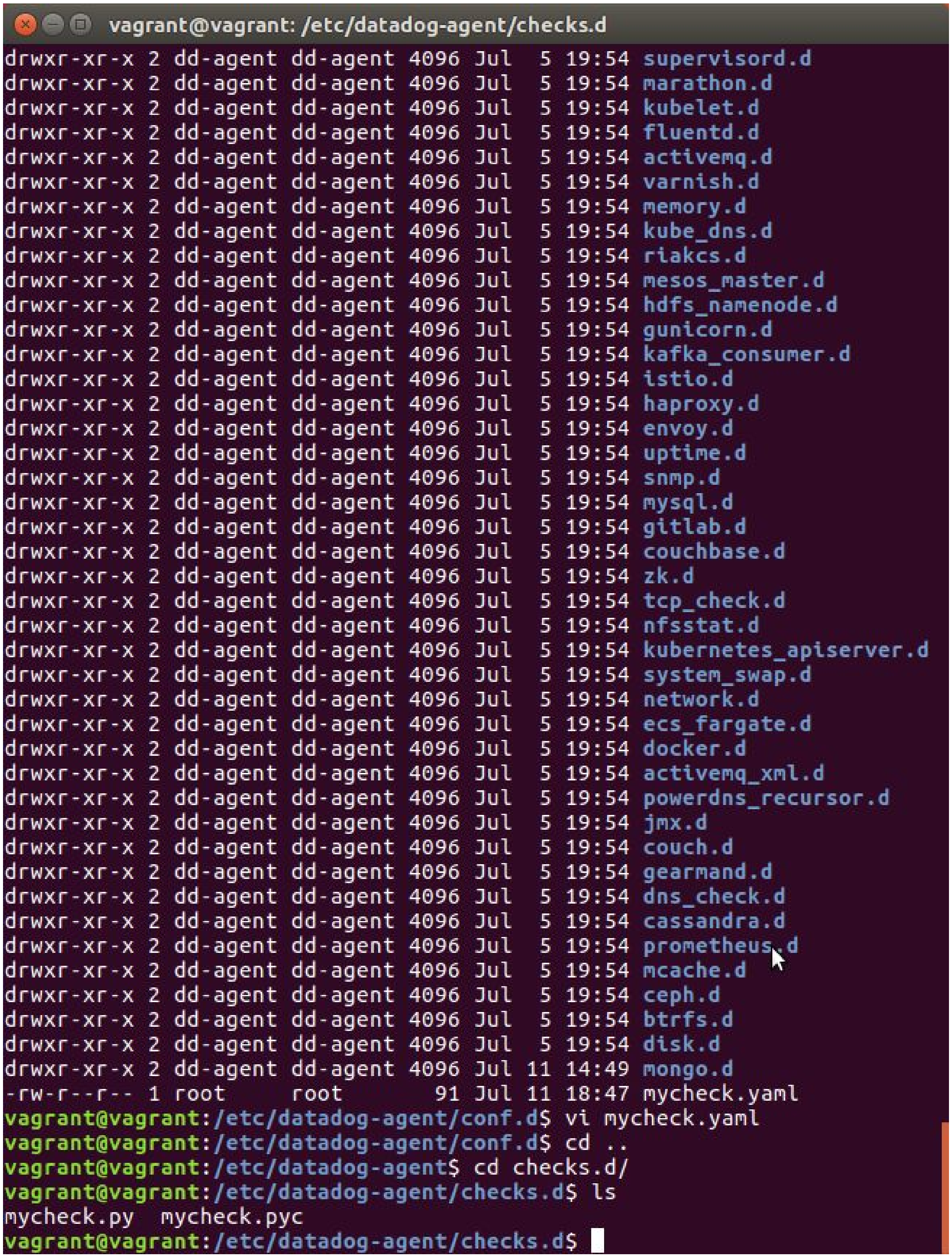
- Create a custom Agent check that submits a metric named my_metric with a random value between 0 and 1000.

Same comment about Agent Checks as for the integrations, will be great to document the purpose and how it helps in monitoring.
https://docs.datadoghq.com/developers/agent_checks/

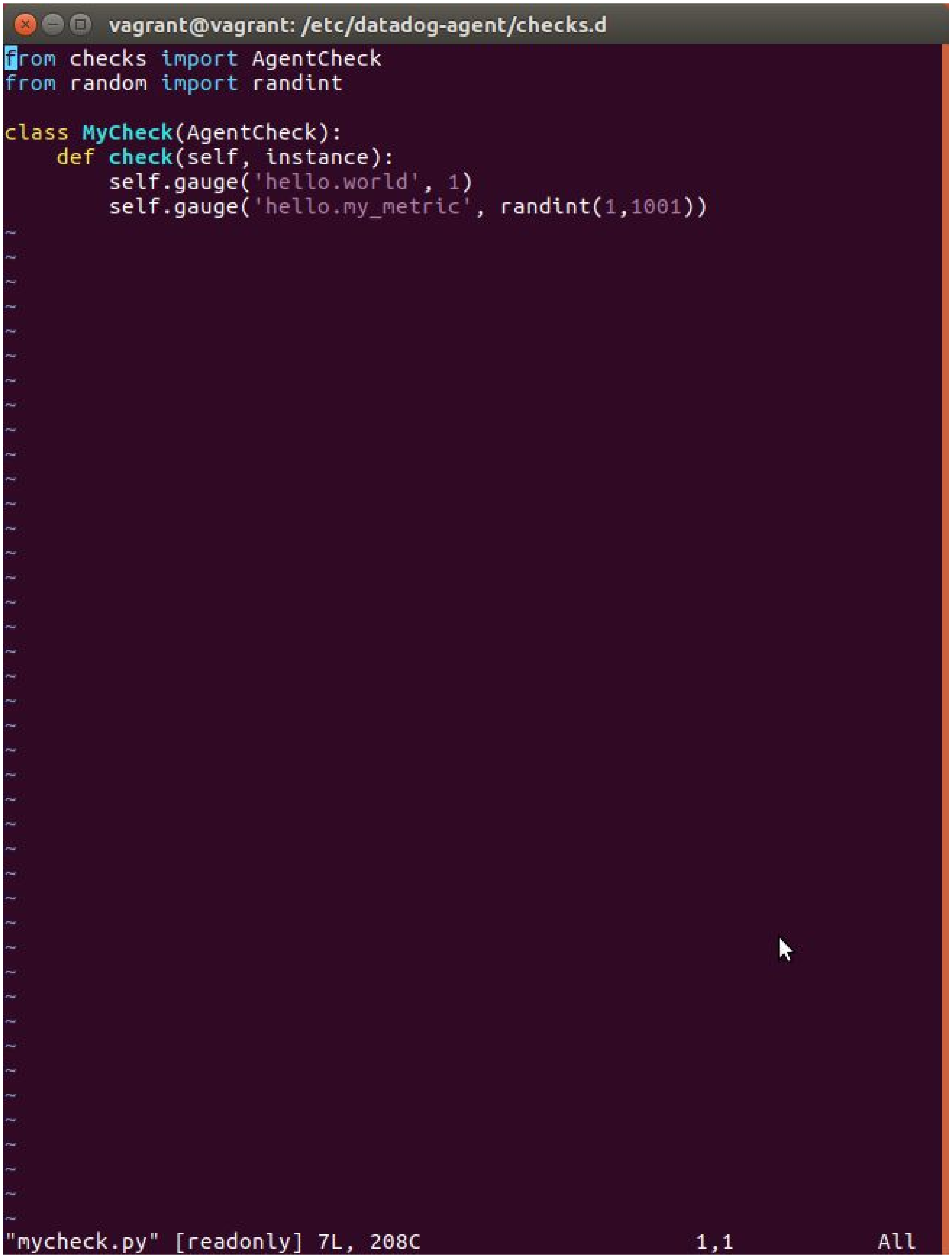


vagrant@vagrant: /etc/datadog-agent/conf.d

```
init_config:
    min_collection_interval: 45

instances:
    - min_collection_interval: 45
~
~
~
~
                                                      1,1           All
```


vagrant@vagrant: /etc/datadog-agent/checks.d

```
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 supervisord.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 marathon.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 kubelet.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 fluentd.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 activemq.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 varnish.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 memory.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 kube_dns.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 riakcs.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 mesos_master.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 hdfs_namenode.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 gunicorn.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 kafka_consumer.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 istio.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 haproxy.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 envoy.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 uptime.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 snmp.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 mysql.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 gitlab.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 couchbase.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 zk.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 tcp_check.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 nfsstat.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 kubernetes_apiserver.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 system_swap.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 network.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 ecs_fargate.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 docker.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 activemq_xml.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 powerdns_recursor.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 jmx.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 couch.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 gearmand.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 dns_check.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 cassandra.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 prometheus.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 mcache.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 ceph.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 btrfs.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul  5 19:54 disk.d
drwxr-xr-x 2 dd-agent dd-agent 4096 Jul 11 14:49 mongo.d
-rw-r--r-- 1 root     root       91 Jul 11 18:47 mycheck.yaml
vagrant@vagrant:/etc/datadog-agent/conf.d$ vi mycheck.yaml
vagrant@vagrant:/etc/datadog-agent/conf.d$ cd ..
vagrant@vagrant:/etc/datadog-agent$ cd checks.d/
vagrant@vagrant:/etc/datadog-agent/checks.d$ ls
mycheck.py  mycheck.pyc
vagrant@vagrant:/etc/datadog-agent/checks.d$ 
```

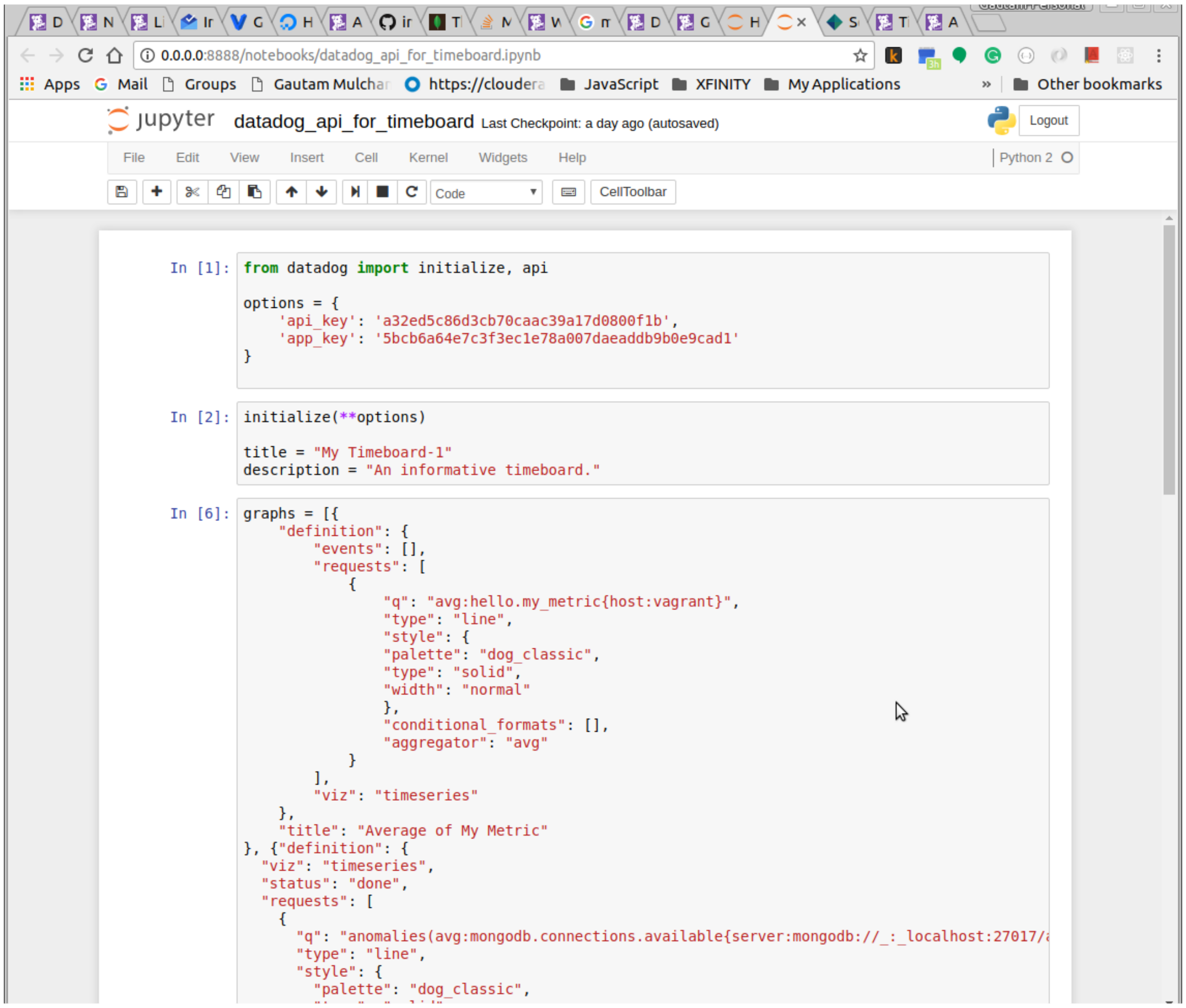


vagrant@vagrant: /etc/datadog-agent/checks.d

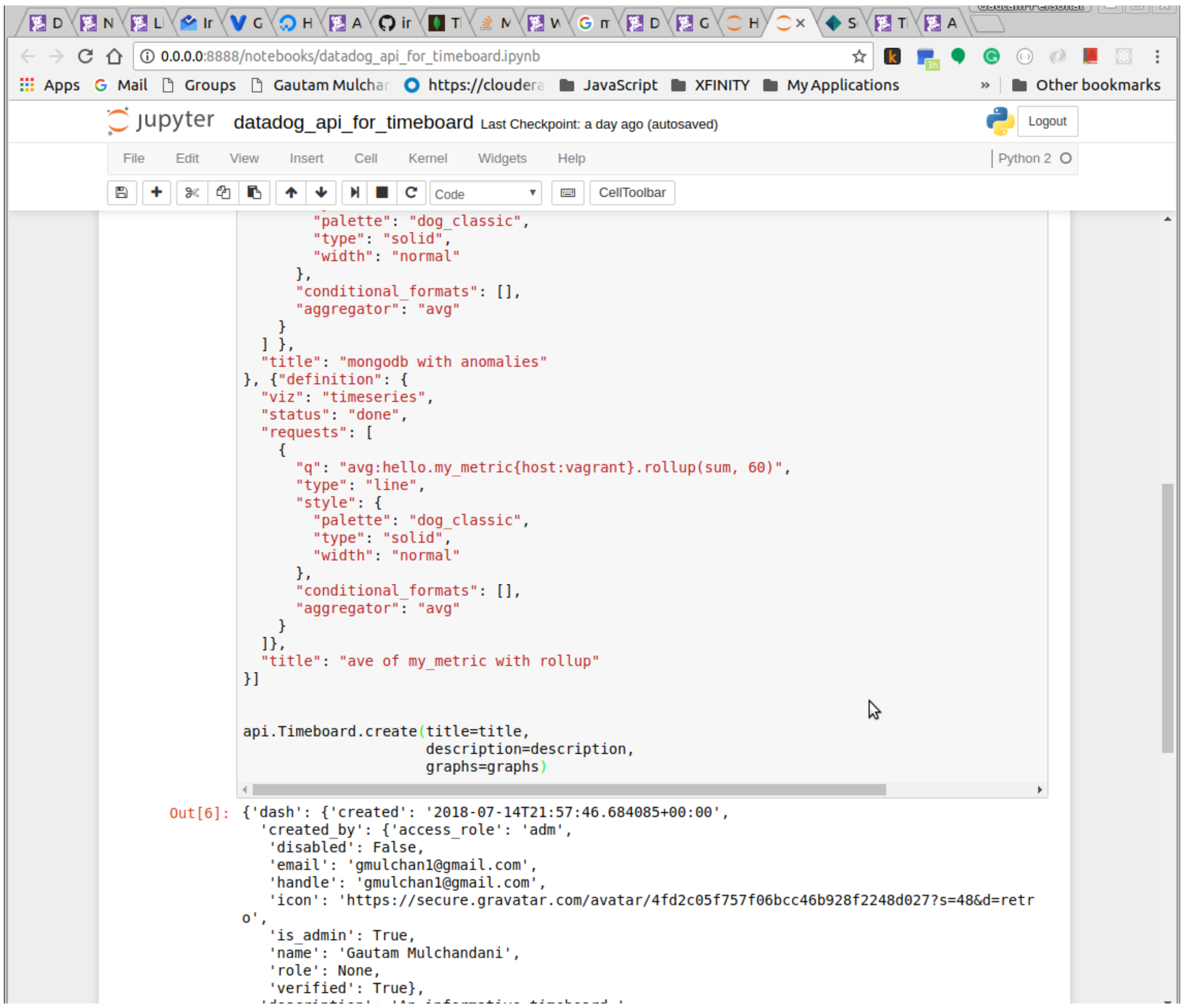
```
from checks import AgentCheck
from random import randint

class MyCheck(AgentCheck):
    def check(self, instance):
        self.gauge('hello.world', 1)
        self.gauge('hello.my_metric', randint(1,1001))
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"mycheck.py" [readonly] 7L, 208C                  1,1           All
```

Visualizing Data:

Datadog dashboarding capabilities look impressive. However creating them through APIs required some reading and understand of the various parameters. So it was very nice to create a dashboard in the UI and then plug in the JSON of each graph into the API call. Testing done in Jupyter notebook and screenshots below (Code also attached in different files):

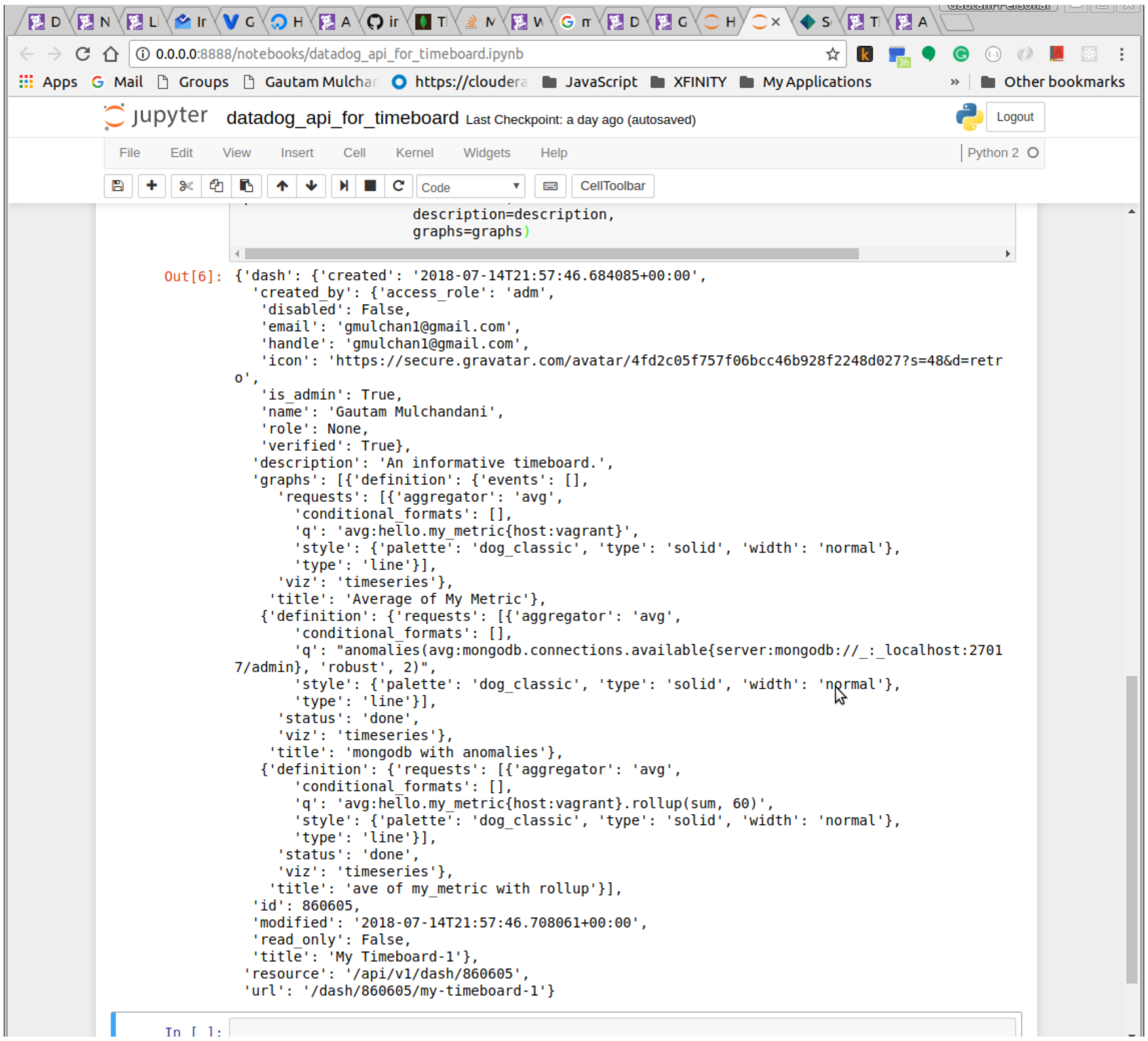


```
In [1]: from datadog import initialize, api

        options = {
            'api_key': 'a32ed5c86d3cb70caac39a17d0800f1b',
            'app_key': '5bcb6a64e7c3f3ec1e78a007daeaddb9b0e9cad1'
        }

In [2]: initialize(**options)

        title = "My Timeboard-1"
        description = "An informative timeboard."

In [6]: graphs = [{
            "definition": {
                "events": [],
                "requests": [
                    {
                        "q": "avg:hello.my_metric{host:vagrant}",
                        "type": "line",
                        "style": {
                        "palette": "dog_classic",
                        "type": "solid",
                        "width": "normal"
                        },
                        "conditional_formats": [],
                        "aggregator": "avg"
                    }
                ],
                "viz": "timeseries"
            },
            "title": "Average of My Metric"
        }, {"definition": {
          "viz": "timeseries",
          "status": "done",
          "requests": [
            {
              "q": "anomalies(avg:mongodb.connections.available{server:mongodb://_:_localhost:27017/a
              "type": "line",
              "style": {
                "palette": "dog_classic",
```


jupyter
datadog_api_for_timeboard Last Checkpoint: a day ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Python 2
Code
CellToolbar

```
          "palette": "dog_classic",
          "type": "solid",
          "width": "normal"
        },
        "conditional_formats": [],
        "aggregator": "avg"
      }
    ] },
    "title": "mongodb with anomalies"
}, {"definition": {
    "viz": "timeseries",
    "status": "done",
    "requests": [
      {
        "q": "avg:hello.my_metric{host:vagrant}.rollup(sum, 60)",
        "type": "line",
        "style": {
          "palette": "dog_classic",
          "type": "solid",
          "width": "normal"
        },
        "conditional_formats": [],
        "aggregator": "avg"
      }
    ]},
    "title": "ave of my_metric with rollup"
}]


api.Timeboard.create(title=title,
                     description=description,
                     graphs=graphs)
```

```
Out[6]: {'dash': {'created': '2018-07-14T21:57:46.684085+00:00',
          'created_by': {'access_role': 'adm',
           'disabled': False,
           'email': 'gmulchan1@gmail.com',
           'handle': 'gmulchan1@gmail.com',
           'icon': 'https://secure.gravatar.com/avatar/4fd2c05f757f06bcc46b928f2248d027?s=48&d=retr
        o',
           'is_admin': True,
           'name': 'Gautam Mulchandani',
           'role': None,
           'verified': True},
```



0.0.0.0:8888/notebooks/datadog_api_for_timeboard.ipynb
jupyter datadog_api_for_timeboard Last Checkpoint: a day ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Python 2
Code
CellToolbar

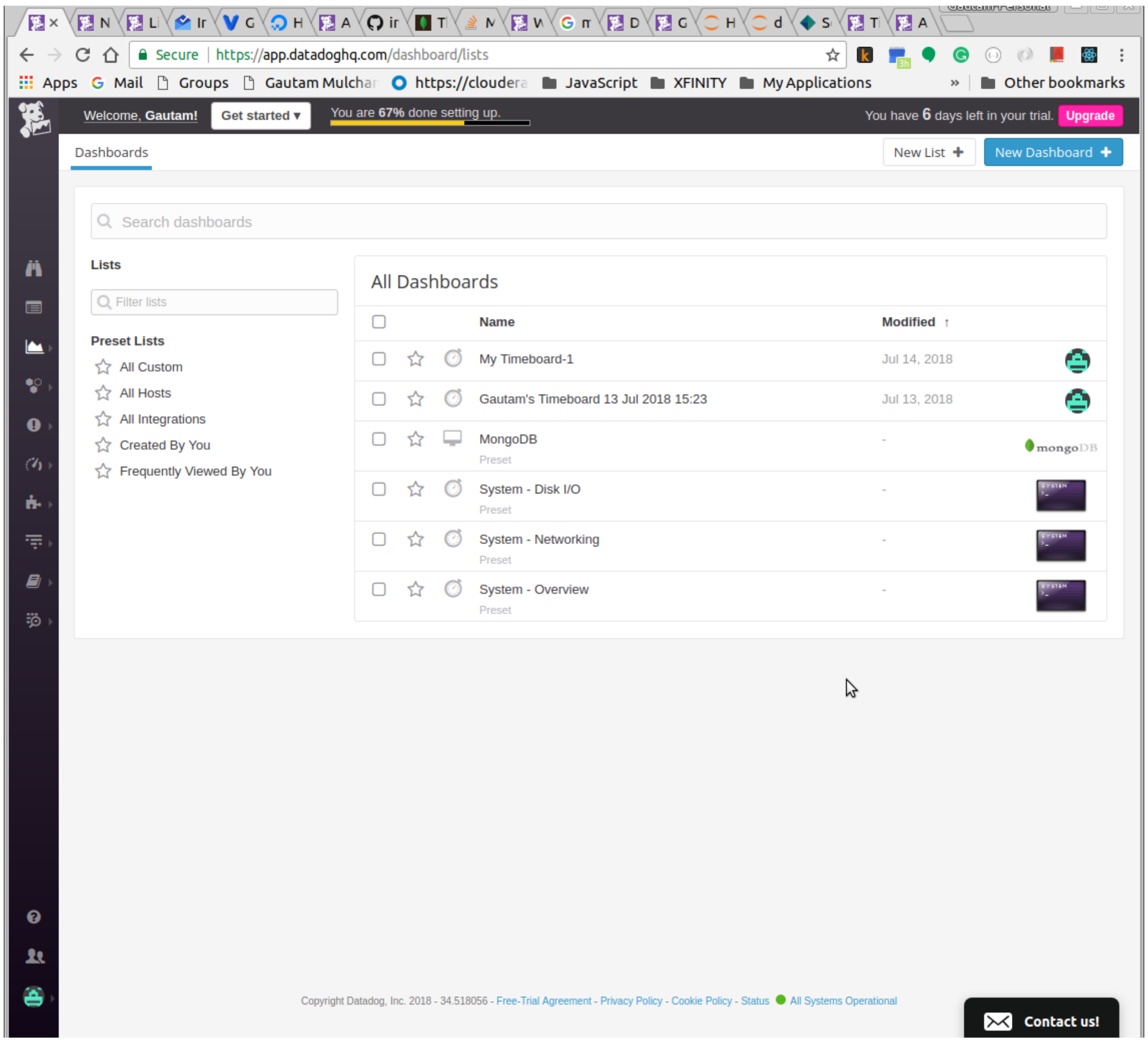
```
                    description=description,
                    graphs=graphs)
```

Out[6]:

```
{'dash': {'created': '2018-07-14T21:57:46.684085+00:00',
  'created_by': {'access_role': 'adm',
   'disabled': False,
   'email': 'gmulchan1@gmail.com',
   'handle': 'gmulchan1@gmail.com',
   'icon': 'https://secure.gravatar.com/avatar/4fd2c05f757f06bcc46b928f2248d027?s=48&d=retr
o',
   'is_admin': True,
   'name': 'Gautam Mulchandani',
   'role': None,
   'verified': True},
  'description': 'An informative timeboard.',
  'graphs': [{'definition': {'events': [],
     'requests': [{'aggregator': 'avg',
       'conditional_formats': [],
       'q': 'avg:hello.my_metric{host:vagrant}',
       'style': {'palette': 'dog_classic', 'type': 'solid', 'width': 'normal'},
       'type': 'line'}],
     'viz': 'timeseries'},
    'title': 'Average of My Metric'},
   {'definition': {'requests': [{'aggregator': 'avg',
       'conditional_formats': [],
       'q': "anomalies(avg:mongodb.connections.available{server:mongodb://_:_localhost:2701
7/admin}, 'robust', 2)",
       'style': {'palette': 'dog_classic', 'type': 'solid', 'width': 'normal'},
       'type': 'line'}],
     'status': 'done',
     'viz': 'timeseries'},
    'title': 'mongodb with anomalies'},
   {'definition': {'requests': [{'aggregator': 'avg',
       'conditional_formats': [],
       'q': 'avg:hello.my_metric{host:vagrant}.rollup(sum, 60)',
       'style': {'palette': 'dog_classic', 'type': 'solid', 'width': 'normal'},
       'type': 'line'}],
     'status': 'done',
     'viz': 'timeseries'},
    'title': 'ave of my_metric with rollup'}],
  'id': 860605,
  'modified': '2018-07-14T21:57:46.708061+00:00',
  'read_only': False,
  'title': 'My Timeboard-1'},
 'resource': '/api/v1/dash/860605',
 'url': '/dash/860605/my-timeboard-1'}
```

In [ ]:

Below, the My Timeboard-1 is created using datadog API and Gautam's Timeboard was created directly in UI.


Welcome, Gautam! Get started ▾ You are 67% done setting up. You have 6 days left in your trial. Upgrade

Dashboards New List New Dashboard

Search dashboards

Lists

Filter lists

Preset Lists
- All Custom
- All Hosts
- All Integrations
- Created By You
- Frequently Viewed By You

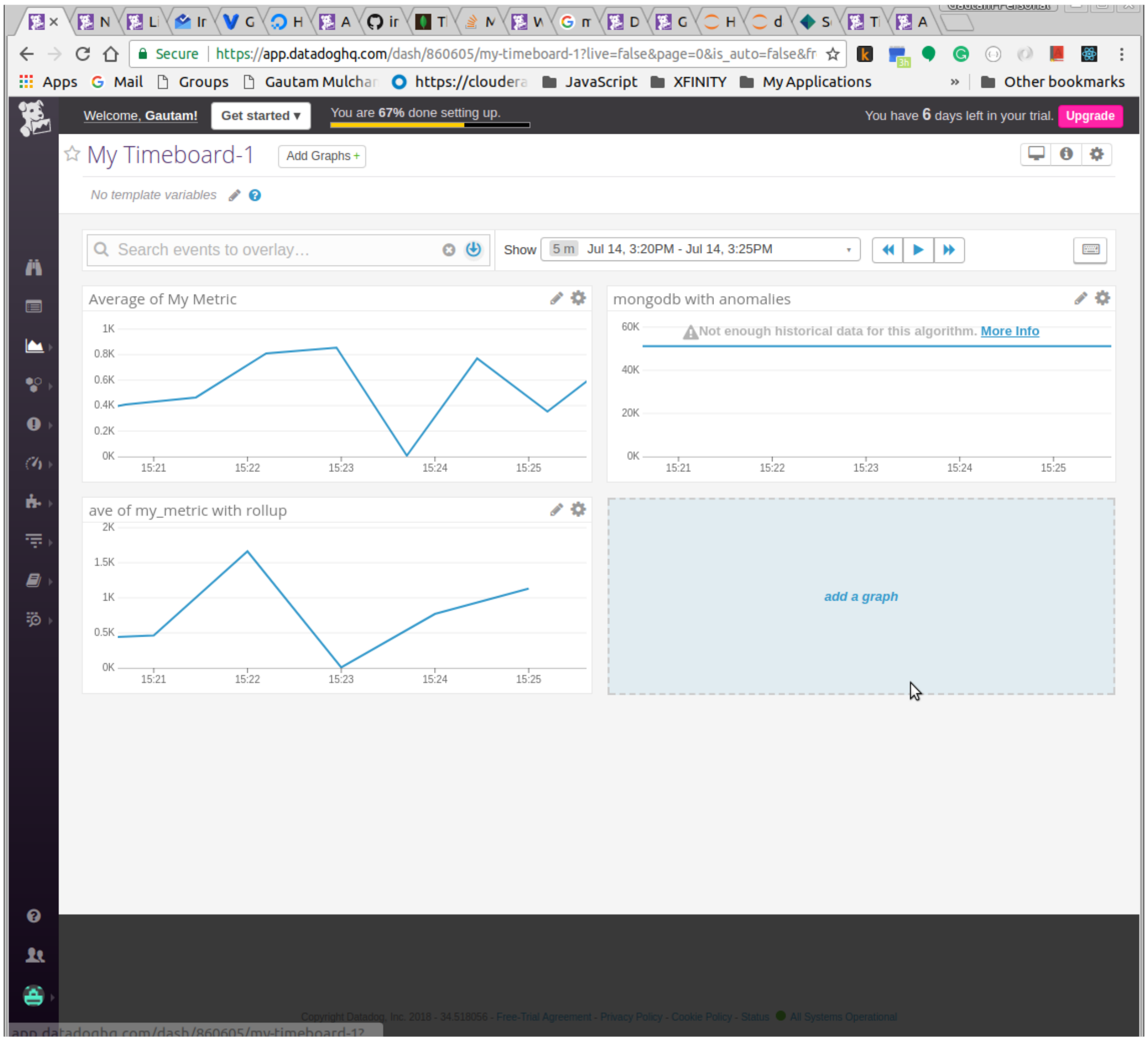
All Dashboards

Name	Modified ↑
My Timeboard-1	Jul 14, 2018
Gautam's Timeboard 13 Jul 2018 15:23	Jul 13, 2018
MongoDB Preset	-
System - Disk I/O Preset	-
System - Networking Preset	-
System - Overview Preset	-

Copyright Datadog, Inc. 2018 - 34.518056 - Free-Trial Agreement - Privacy Policy - Cookie Policy - Status ● All Systems Operational

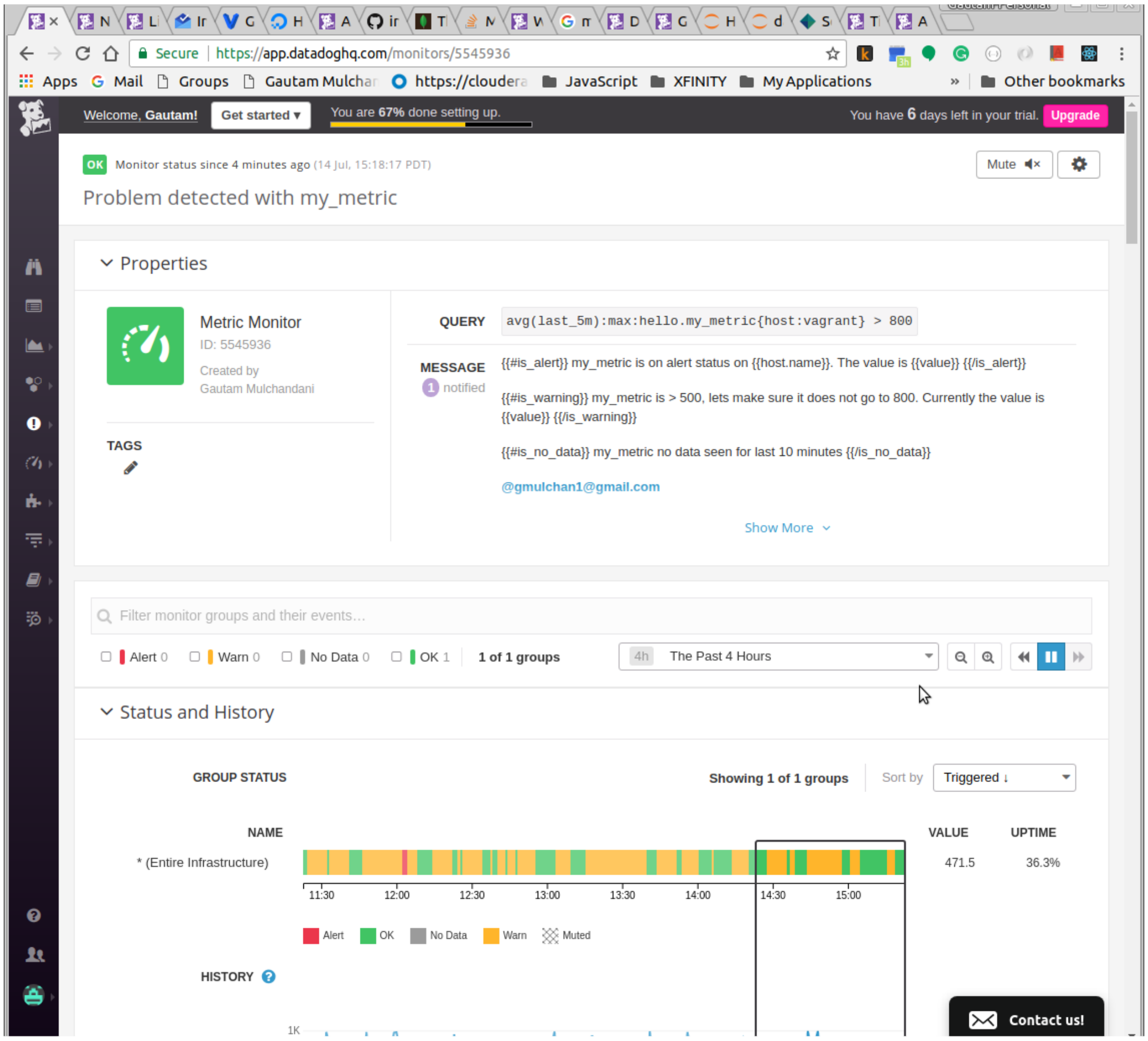
Contact us!

Last 5 minutes on graph:


Welcome, Gautam!
Get started
You are 67% done setting up.
You have 6 days left in your trial.
Upgrade
My Timeboard-1
Add Graphs +
No template variables
Search events to overlay...
Show 5 m Jul 14, 3:20PM - Jul 14, 3:25PM
Average of My Metric
mongodb with anomalies
Not enough historical data for this algorithm. More Info
ave of my_metric with rollup
add a graph

Can send a graph in email by annotating it with @email address but not the entire timeboard. The anomaly graph just shows not enough data - probably because it needs sufficient historical data points to compare to and make predictions on what is "normal".

Monitoring Data.


OK Monitor status since 4 minutes ago (14 Jul, 15:18:17 PDT)

Problem detected with my_metric

Properties

Metric Monitor
ID: 5545936
Created by
Gautam Mulchandani

TAGS

QUERY
```
avg(last_5m):max:hello.my_metric{host:vagrant} > 800
```

MESSAGE
1 notified

{{#is_alert}} my_metric is on alert status on {{host.name}}. The value is {{value}} {{/is_alert}}

{{#is_warning}} my_metric is > 500, lets make sure it does not go to 800. Currently the value is {{value}} {{/is_warning}}

{{#is_no_data}} my_metric no data seen for last 10 minutes {{/is_no_data}}

@gmulchan1@gmail.com

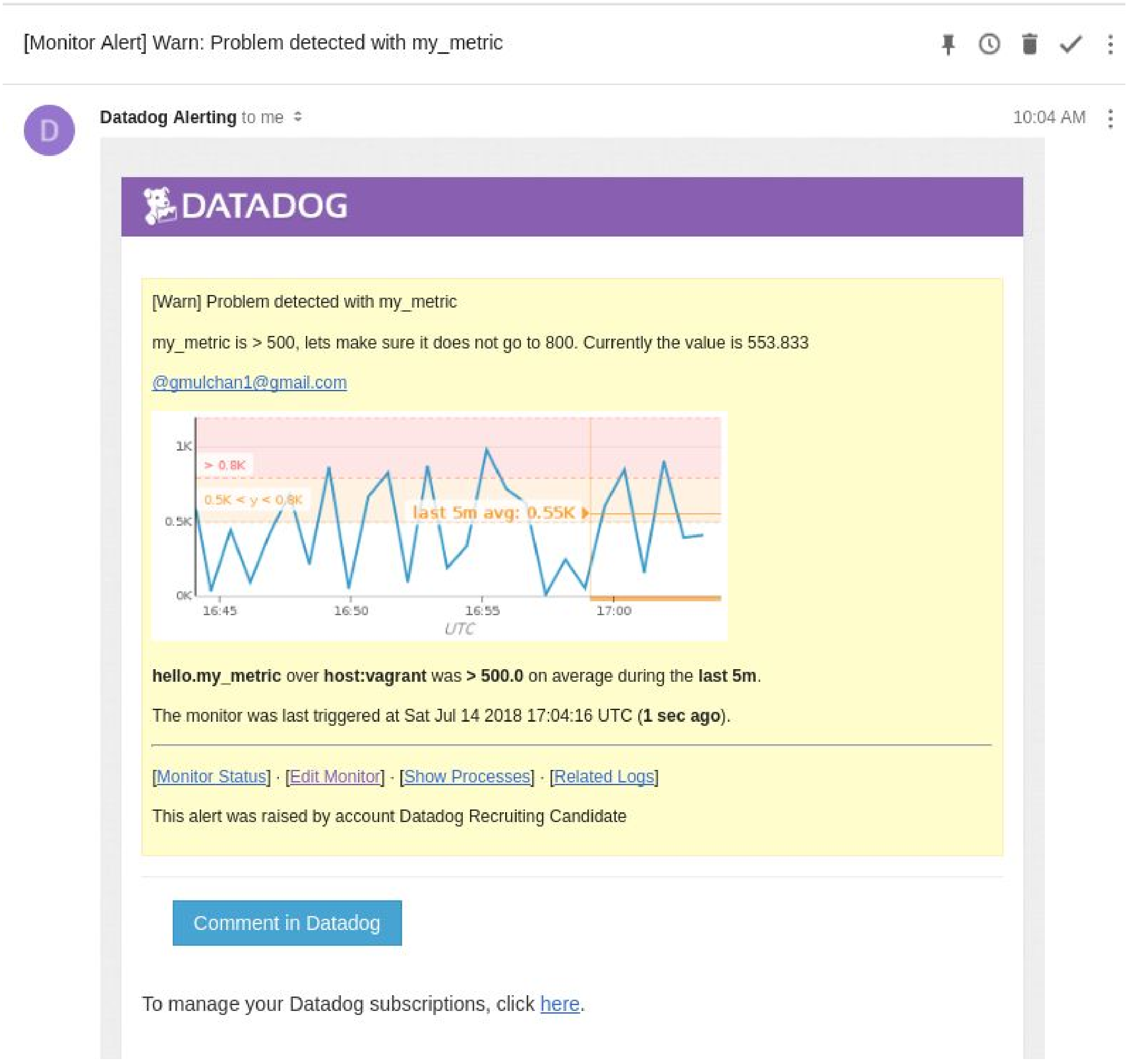
Status and History

Alert 0 | Warn 0 | No Data 0 | OK 1 | 1 of 1 groups | 4h The Past 4 Hours

GROUP STATUS — Showing 1 of 1 groups — Sort by Triggered ↓

NAME	VALUE	UPTIME
* (Entire Infrastructure)	471.5	36.3%

The alert from the monitor that was setup is below. Setting up monitor and alert quite simple, even with different messages for warning, critical and no data.


[Monitor Alert] Warn: Problem detected with my_metric

Datadog Alerting to me 10:04 AM

[Warn] Problem detected with my_metric

my_metric is > 500, lets make sure it does not go to 800. Currently the value is 553.833

@gmulchan1@gmail.com

hello.my_metric over **host:vagrant** was **> 500.0** on average during the **last 5m**.

The monitor was last triggered at Sat Jul 14 2018 17:04:16 UTC (**1 sec ago**).

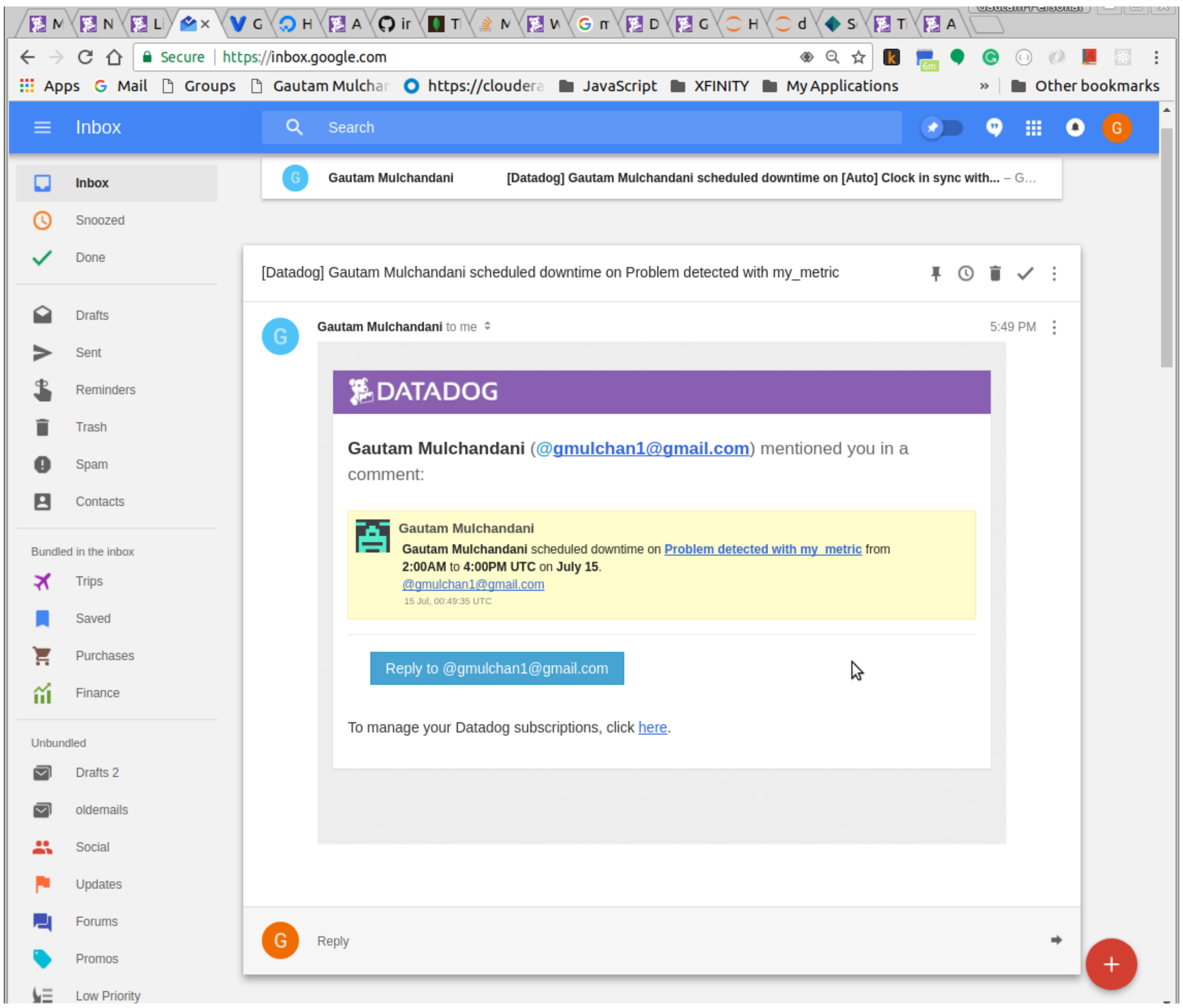
[Monitor Status] · [Edit Monitor] · [Show Processes] · [Related Logs]

This alert was raised by account Datadog Recruiting Candidate

To manage your Datadog subscriptions, click here.

Scheduling downtime:

Important to provide a way to schedule downtime. Downtime can occur as a result of regular maintenance, for example during upgrade or changing hardware resources. If downtime is not scheduled, it can result in an alert storm where the alerts are meaningless. Also, does not help the operations team in reporting their SLAs.



Gautam Mulchandani [Datadog] Gautam Mulchandani scheduled downtime on [Auto] Clock in sync with... – G...

[Datadog] Gautam Mulchandani scheduled downtime on Problem detected with my_metric

Gautam Mulchandani to me 5:49 PM

DATADOG

Gautam Mulchandani (@gmulchan1@gmail.com) mentioned you in a comment:

Gautam Mulchandani
Gautam Mulchandani scheduled downtime on Problem detected with my_metric from 2:00AM to 4:00PM UTC on July 15.
@gmulchan1@gmail.com
15 Jul, 00:49:35 UTC

Reply to @gmulchan1@gmail.com

To manage your Datadog subscriptions, click here.

Collecting APM Data

Combining both APM and infrastructure monitoring into single dashboards is very powerful as one can quickly make time based correlations between the Business Services that are served by the Applications and the infrastructure components the applications rely upon. This helps in triaging and even diagnosing problems instead of multiple teams in development, operations, networks, databases pointing fingers at each other. Adding APM metrics to the existing dashboard was simple by just going to the APM screen and selecting Export to Timeboard and then selecting the Timeboard.

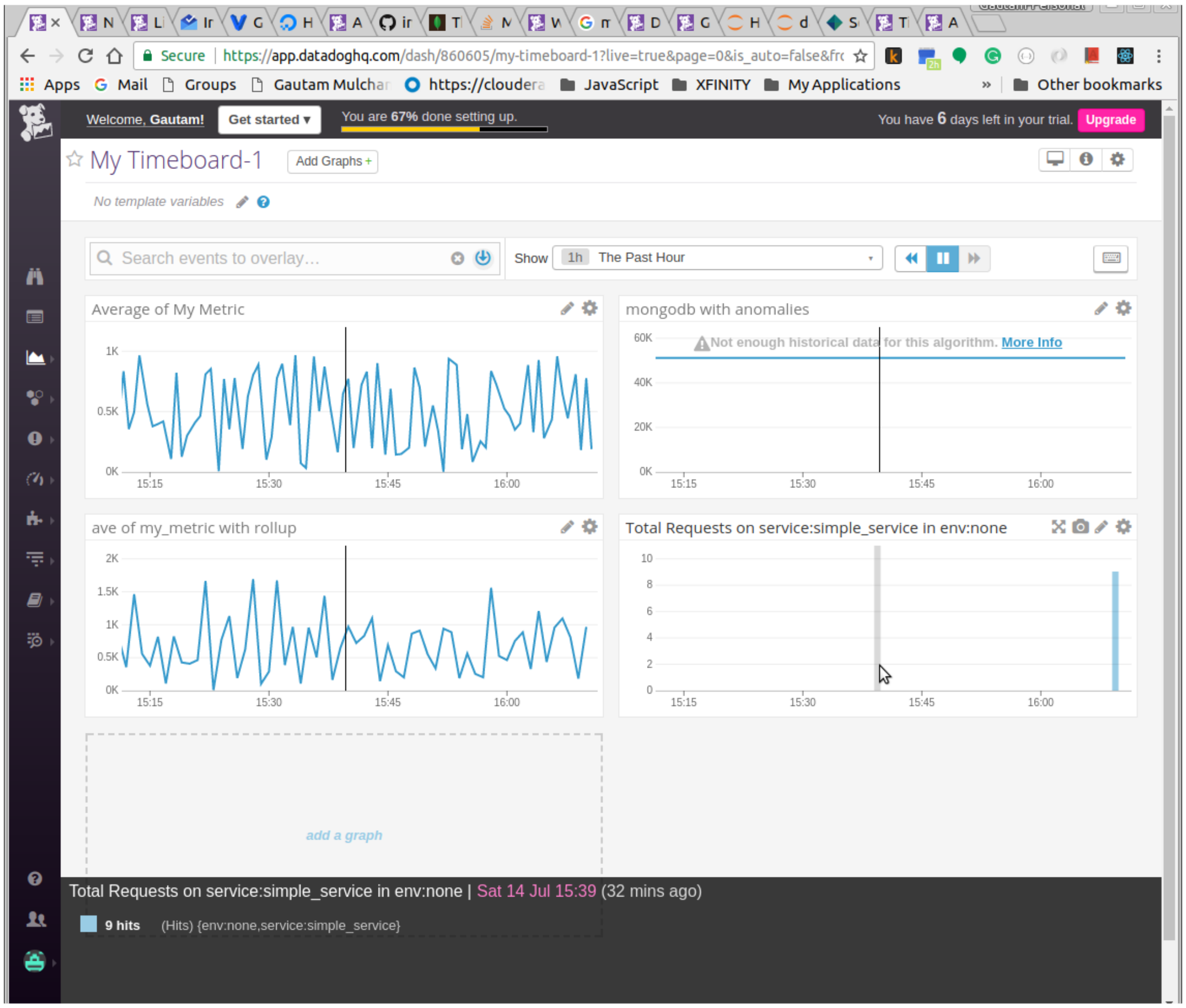
Also liked that there is no need for code changes. The datadog agent handles the instrumentation automatically. However, to add traces have to put the annotation in the

source code. There might be another way to achieve the same through monkey patching.
The modified source code for the Flask application is also uploaded.

Link to Timeboard with both APM and Infra metrics.

https://app.datadoghq.com/dash/860605/my-timeboard-1?live=true&page=0&is_auto=false&from_ts=1531606346480&to_ts=1531609946480&tile_size=m

Screenshot:


Welcome, Gautam!
Get started
You are 67% done setting up.
You have 6 days left in your trial. Upgrade
My Timeboard-1
Add Graphs +
No template variables
Search events to overlay…
Show 1h The Past Hour
Average of My Metric
mongodb with anomalies
Not enough historical data for this algorithm. More Info
ave of my_metric with rollup
Total Requests on service:simple_service in env:none
add a graph

Total Requests on service:simple_service in env:none | Sat 14 Jul 15:39 (32 mins ago)

9 hits (Hits) {env:none,service:simple_service}

Difference between Service and Resource:

https://help.datadoghq.com/hc/en-us/articles/115000702546-What-is-the-Difference-Between-Type-Service-Resource-and-Name-

At a high level, a Service is a logical entity that is implemented by an application. The service itself provides resources, each resource implemented by an API call.

Is there anything creative you would use Datadog for?

Looking at capabilities of datadog while performing the exercises, I would consider using it for predicting scenarios and taking action that are beneficial.

1. For example, Mumbai city gets flooded during monsoon season as it happened in the last 1 week. Based on the amount of rainfall, water level of of certain streets, level of the sea tide, one can predict exactly which neighbourhood and streets will get flooded next and take preventive measures such as diverting traffic, pumping water out at certain intersections. This can bring relief to people who have to guess while navigating the heavy rainfall and still get to work.
2. Measure the temperature, general weather patterns, vaccinations given to population and use that as a measure of how many patients will be admitted to hospitals. That will determine how many personnel are required - nurses, doctors as well as medications. Proactively have the resources ready or alerted so that the right amount of resources are available to people who need help.