**Level 1 – Collecting Your Data**

We began by installing the Datadog agent on an Ubuntu 16.04 server running on VMWare Workstation.

What is the Agent?

Very simply, the agent is a piece of software that runs on the host machine and collects information. This information includes metrics or events, and can be collected from various sources, as well as by various means, on the host machine.

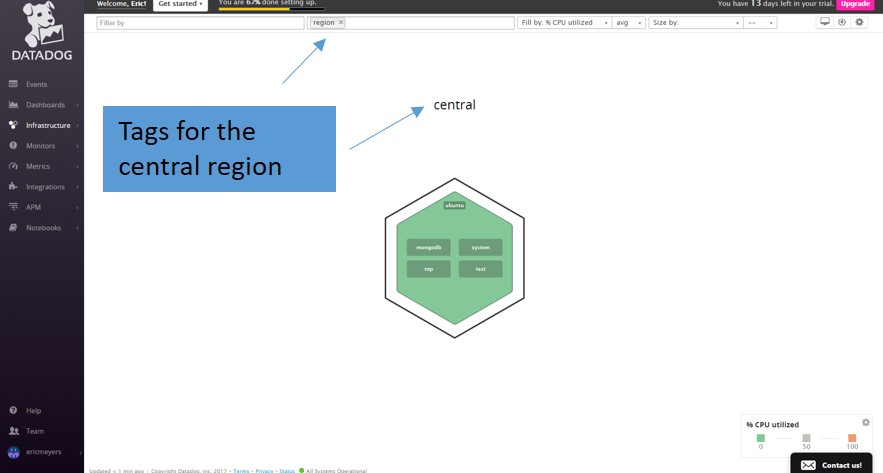
Tags

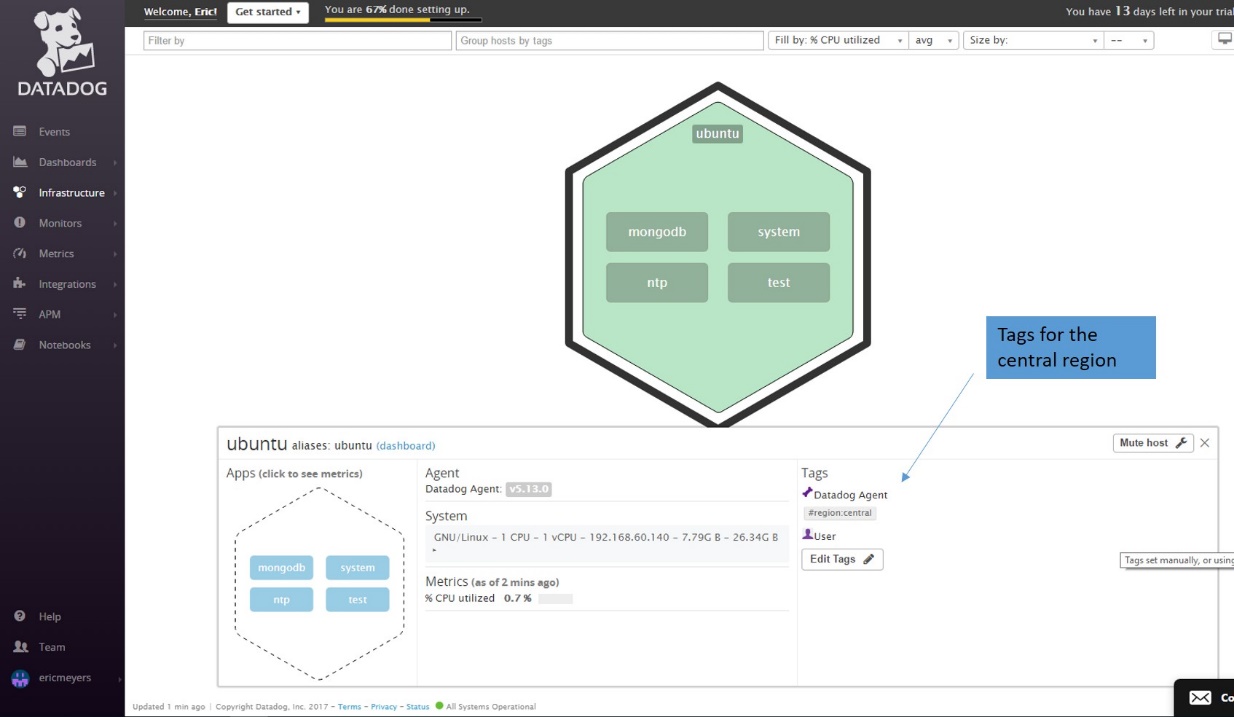
Tags are used to make it easier to subset and query the metrics that you have to monitor. You can label and organize entities so they roll up into meaningful categories. This, in turn, will make it easier to subset and query those metrics, so that you can find the right metrics you need at the right time.

There are many ways to tag and tagging can be done for the overall agent as well as for each integration. For this exercise, we tagged the datadog.conf file, which is a high level configuration file for the agent. We added a simple line of code to show that our host is in the Central region:

tags: region:central

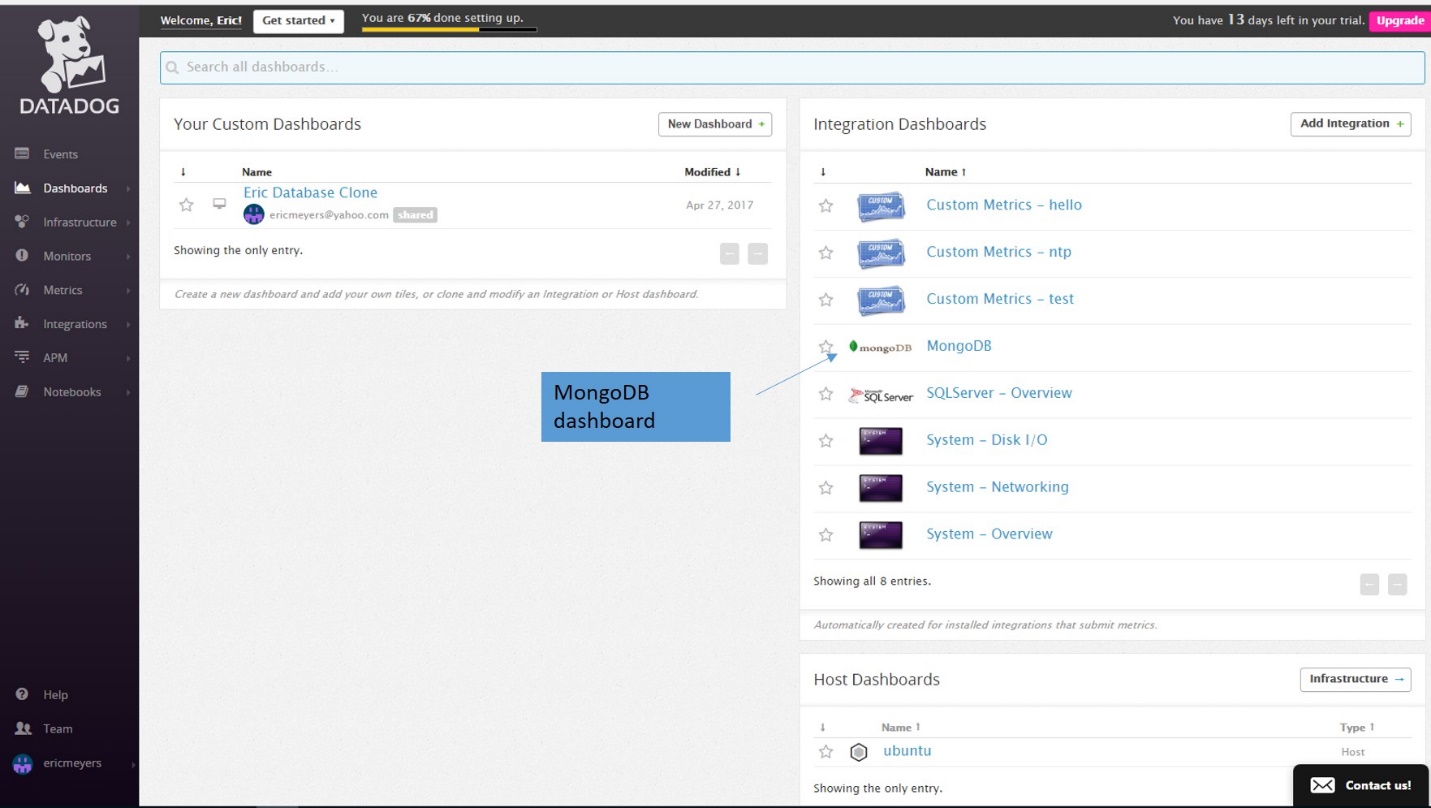
From the Host Map page in Datadog, you will see a screenshot of the host as well as the tags. In particular, you will see that we are focused particular on the Central region, which includes our Ubuntu server.

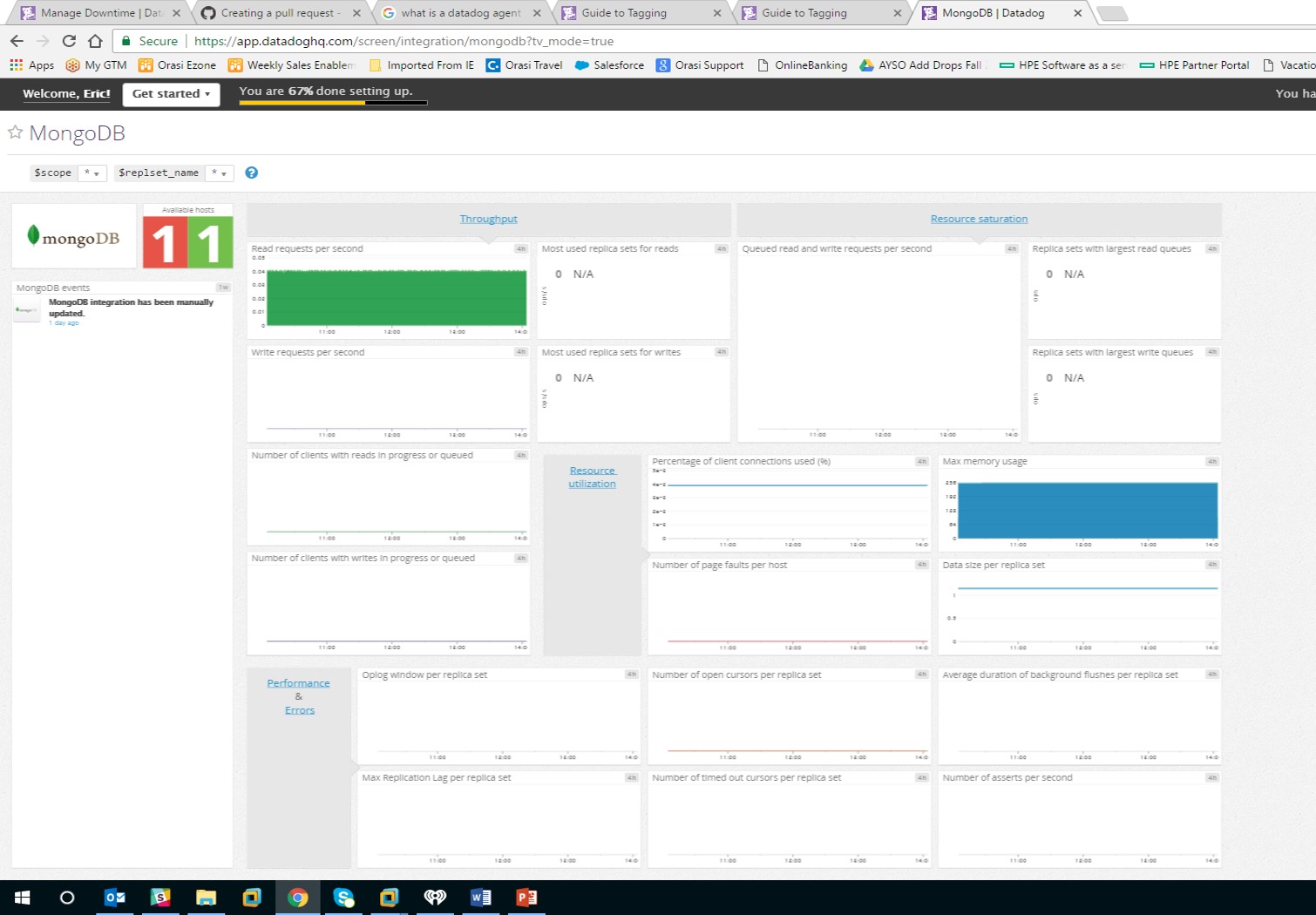




Database

We installed the latest MongoDB, and then installed the MongoDB integration for this database. We created a new user in the “admin” database, and this user has administrative privileges to collect server statistics. We then began collecting data against this database and it is reporting into a dashboard.



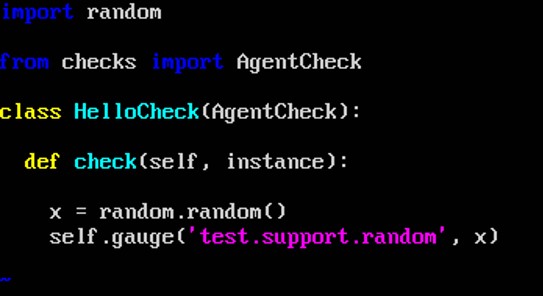


Custom Agent Check

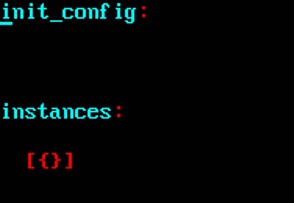
We decided to create a custom agent check that samples a random value between 0 and 1. To do this, we created 2 different files.

Under the checks.d directory, we created a Python file called “eric.py”. Why did we call it “eric” – well I think that’s a great name for a file! This is the code itself that grabs the random number and we assigned it to a metric called “test.support.random”. Under the conf.d directory, we created a yaml file called “eric.yaml”.

The eric.py file looks like:



The eric.yaml code looks like:



**Level 2 – Visualizing Your Data**

The MangoDB dashboard was cloned, we added the test.support.random graph to the dashboard as well. You will find that dashboard here:

<https://p.datadoghq.com/sb/abc79fbd6-3274f27563>

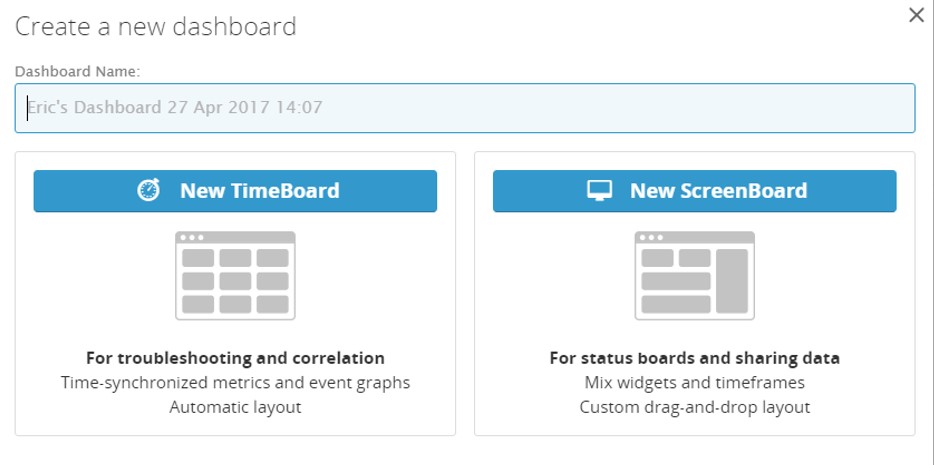
Timeboards and Screenboards

At Datadog, we have 2 different types of dashboards. These are called timeboards and screenboards. You’ll see that you can choose the type of dashboard you want when you select a new dashboard.

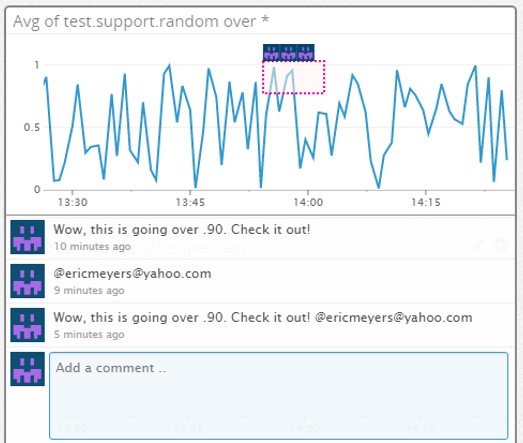
See Screenshot “New Dashboard”

A timeboard is intended to use in troubleshooting efforts, and has all of the metrics time-synced to try to show correlation between events on different systems.

A screenboard is more flexible than dashboards, and each graph can be on its own timeframe and the graphs can have different sizes and different feel. In addition, there is more flexibility in the sharing options of a screenboard.



In the test.support.random graph, you will see that it periodically goes about .90. We are able to have some collaboration on this graph – which can be very useful to resolving an issue.



Level 3 – Alerting on your Data

A monitor was set up that alerts me when the value goes above .90 at least one time in the last 5 minutes. In addition, this was set up as a multi-alert by host, so if the infrastructure scales, we won’t have to recreate the alert every time we add an additional host.

A screenshot of setting up the alert

A screenshot of the email that is received during the alert.

In addition, since this monitor alerts quite often, we set up scheduled downtime, so we are not alerted at certain times.

A screenshot of setting up the scheduled downtime

A screenshot of the notification