

XYZ POV Report
Datadog

Dates: 5/9/2018 - 5/11/2018

Datadog Contacts:

Tre' Sellari – Sales Engineer

XYZ Contacts:

Dustin Lawler – VP of App

Summary

For the XYZ POV we selected three different hosts to monitor. A Windows 7 laptop, a Linux host running Docker, and a Windows 10 desktop. Agents were installed on each host, a PostgreSQL database and Application Performance Monitoring (APM) data was collected from a python application and a Java application.

Using metrics collected from the hosts, databases, and applications a Timeboard was created using the Datadog API. With this data centralized into Datadog, XYZ will now be able to monitor data from the front in to the backend of their applications. This will allow XYZ to have insight into potential issues, anomalies, and improve overall quality and performance of their applications. Which in the end will help drive customer satisfaction and increase revenue potential.

POV Business Cases and Objects:

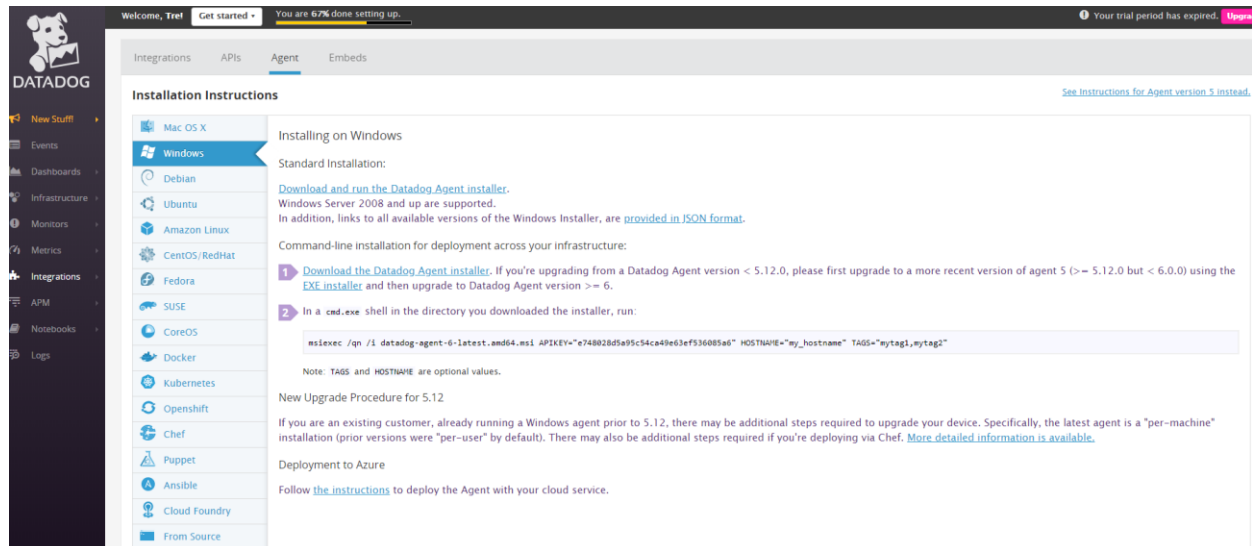
- Installation and setup of agents, APM, and integrations on hosts, database, and applications
- Prove out native integrations to high value components of XYZ infrastructure
- Prove out Datadog's ability to collect metrics from multiple sources into a single interface
- Prove out Datadog's ability to collect Application Performance Metrics
- Show Datadog's ability to identify anomalies
- Demonstrate Datadog's ability create customized alerts via Metric Monitors

Day 1

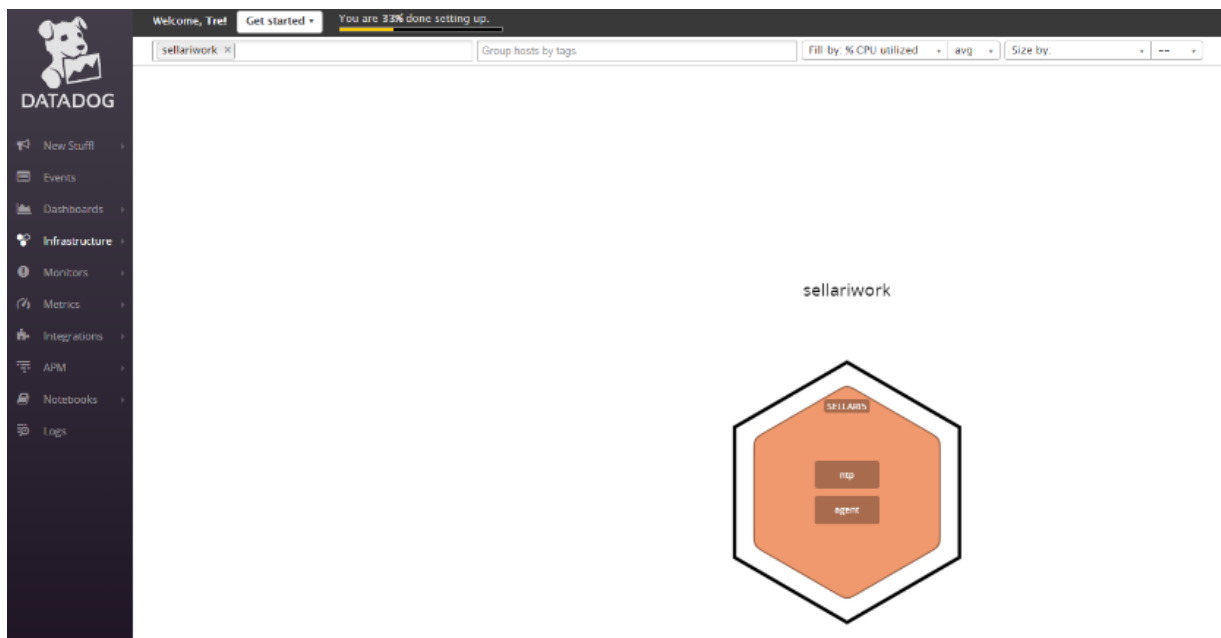
- Agent Installation on Hosts
- Database integration

Agent Installation on Hosts

- From the Integrations tab in Datadog, simply select the appropriate Agent and follow instructions. Here is an example of the Windows installation page:



- Once the Agent has been installed, it will automatically begin to report metric data. You will see information about the host in the Infrastructure tab:



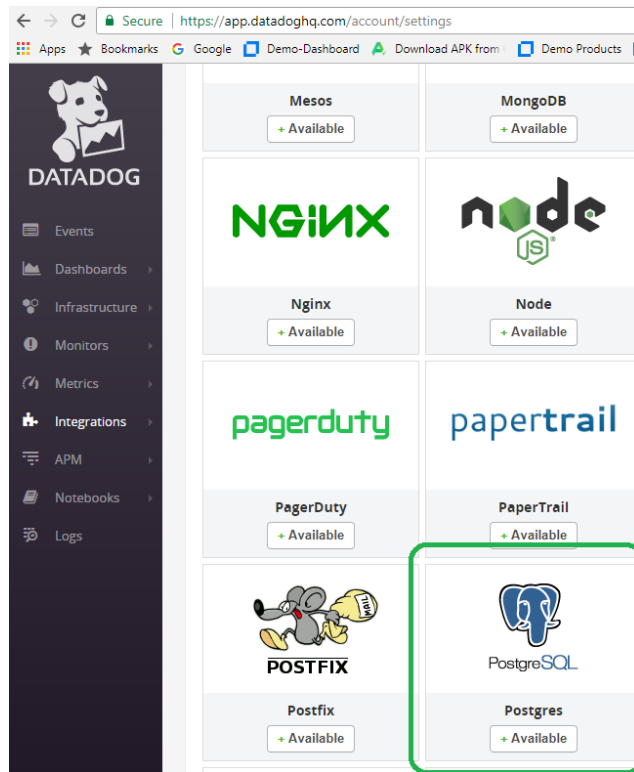
- With Datadog's ability to use tags, you can make life easier by giving hosts tags in the agent config file.

Here is an example:

```
95     server_timeout: 15
96     skip_ssl_validation: false
97     statsd_forward_host: ""
98     statsd_forward_port: 0
99     statsd_metric_namespace: ""
100    syslog_pem: ""
101    syslog_rfc: false
102    syslog_tls: false
103    syslog_uri: ""
104    tags: key_first_tag:SellariWork
105    use_dogstatsd: true
106    use_service_mapper: true
107    use_v2_api:
108      events: false
109      series: false
110      service_checks: false
```

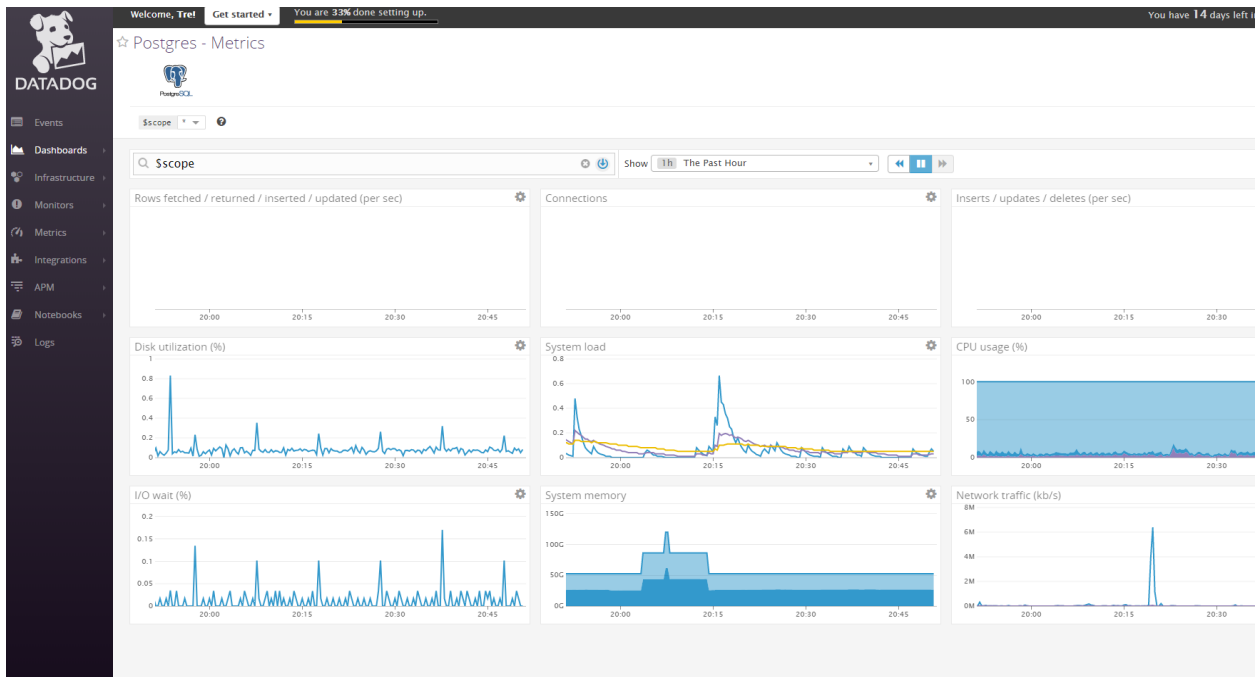
Database integrations Setup

- From the Integrations tab in Datadog select Integrations. There you will see all the native integrations Datadog provides out of the box.



- For the XYZ POV, we integrated with PostgreSQL
 - Here is the summary of the steps taken to accomplish the integrations
 - Configured Datadog user for PostgreSQL
 - Created postgres.yaml file from example in directory and added it to the conf.d directory.
 - Added needed connection information to yaml file
 - Stopped/restarted PostgreSQL

- Once the integration was completed metrics began to automatically be relayed into Datadog



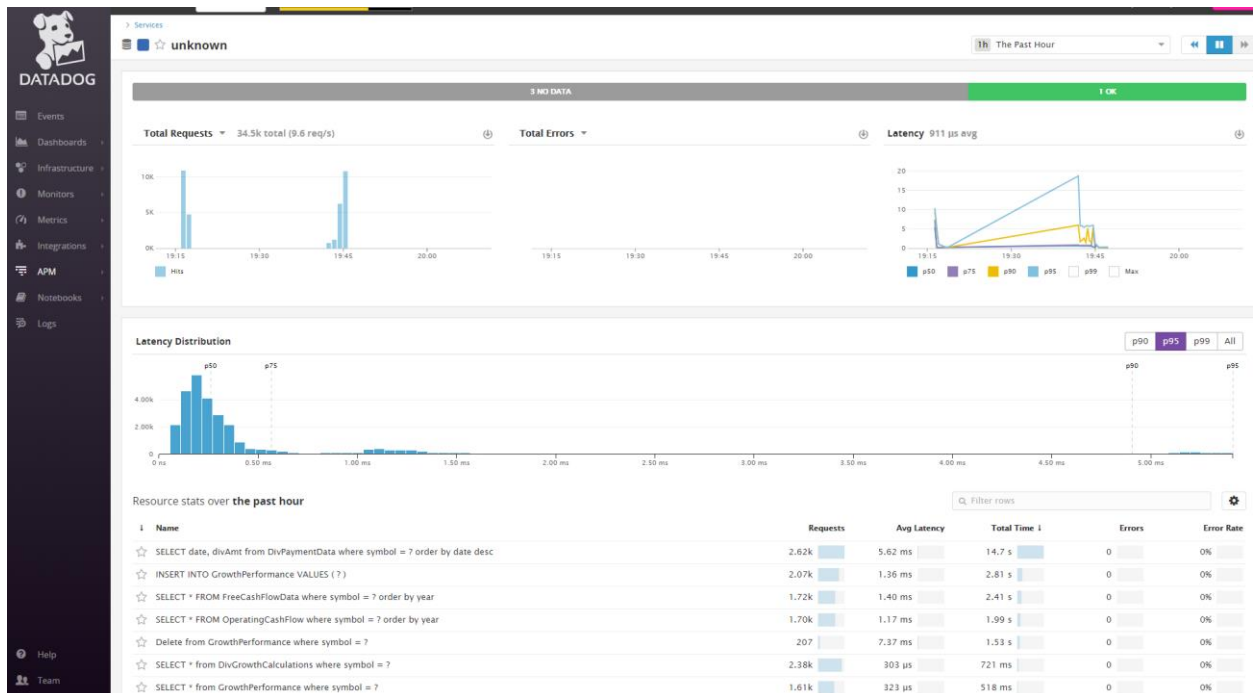
- **Key Value Point**
 - With Datadog you will be able to see metrics from all aspects of the application. From UI, infrastructure, and down to the database. This is important because creating a high quality application with a valuable experience is not just dependent on the UI. All levels of the applications must be functioning correctly.

Day 2

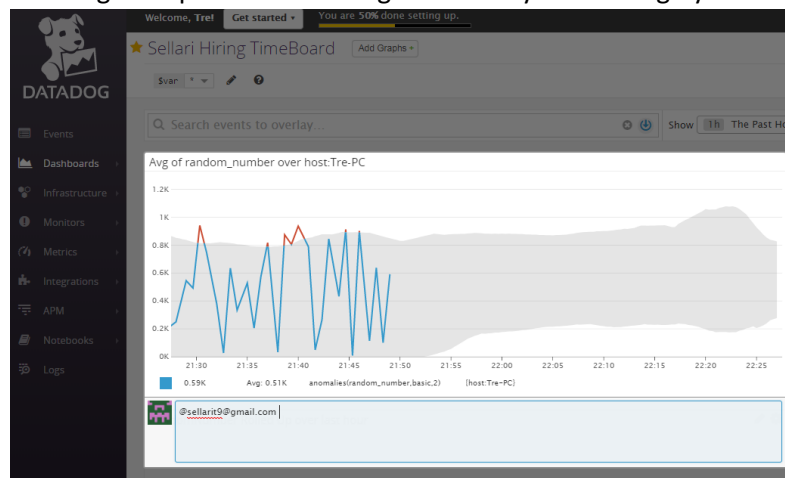
- Creating Dashboards
 - Identifying anomalies
- Setup Monitoring and Alerting

Creating Valuable Dashboards

- In Datadog XYZ has the ability to create dashboards that have metrics from all levels of the application and infrastructure.
For example, here you can see information from the database, with the query calls, information about the number of requests, and also the latency in the application:

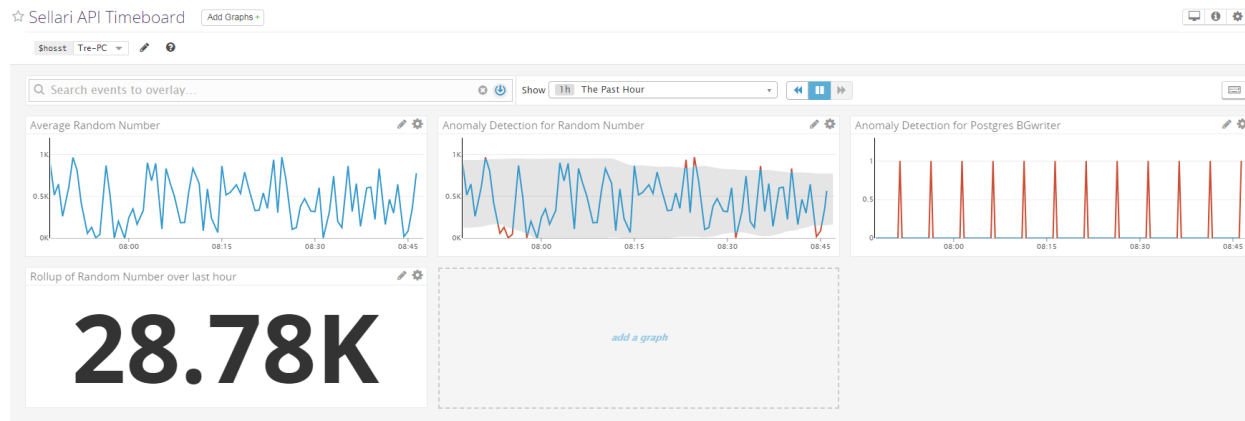


- Once dashboards have been created you have multiple ways to share this valuable information
 - Via a shareable link - https://p.datadoghq.com/sb/bd421a7ea-cf8af2ebc0f98f2b6ad45bea2bae99fa?tv_mode=false
 - Creating a snapshot and sharing it internally in Datadog by referencing someone with @



This allows for collaboration inside and out of Datadog. You can also integration with chatops solutions, such as slack, HipChat, and others to drive collaboration.

- With metrics in a dashboard, we can now use Datadog to identify anomalies.
 - Here is a Timeboard where two of the graphs are using anomaly detection:

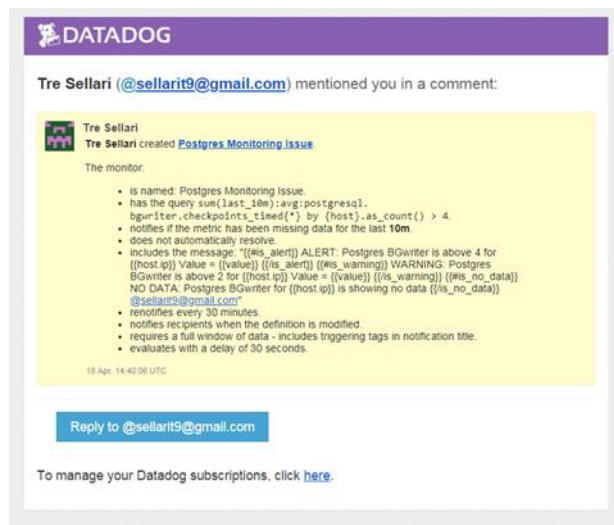


You can see in the graph “Anomaly Detection for Random Number” that there are times with the values are following outside of the expected range. This is indicated by the red parts of the line graph.

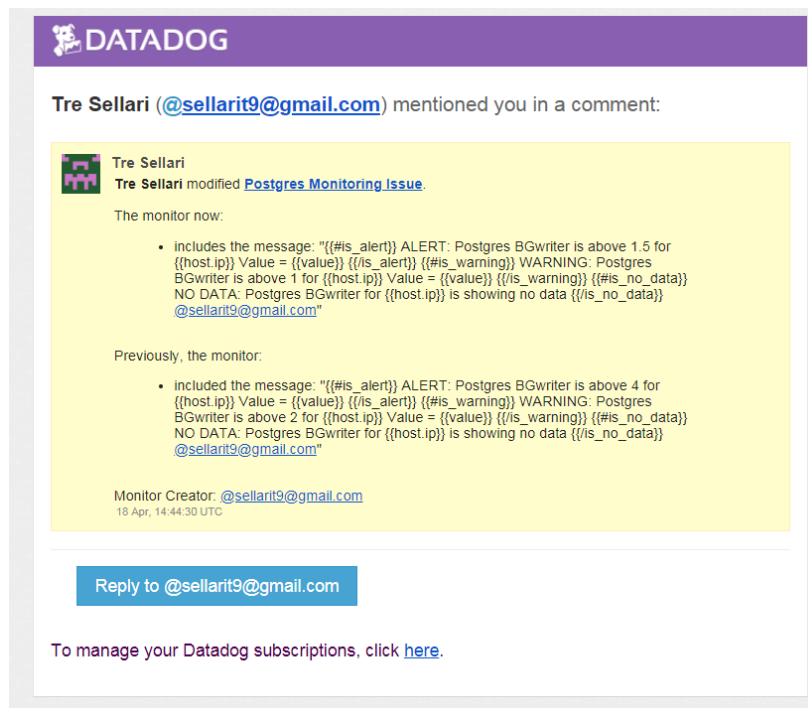
- **Key Value Point**
 - Identifying potential issues in the application will provide greater value than just looking and waiting for the application to break or go down. Being able to isolate anomalies will help XYZ find and prepare for issues. Don’t fall into the “if it ain’t broke, don’t fix it!” mindset. Your application might not be broken, but it could still be functioning at a poor quality, which will drive away revenue, productivity, and success.

Extending the Data in Dashboards to Monitoring and Alerting

- Now that metrics are collected and useable in Datadog setting up monitors and alerts will give XYZ the ability to take action, when issues arise, in a timely and cost savings manor.
- In Datadog XYZ can create custom monitors that alert the appropriate parties, so action can be taken
 - Here is an example of a created monitor



- When the monitor is triggered or modified an email alert can be sent out



- Monitors can also be setup with scheduled downtime
 - Here you can see that our monitor will be down, Monday – Friday from 7am – 9pm

Schedule Downtime

1 Choose what to silence

Monitor: Postgres Monitoring Issue

2 Schedule

One-Time **Recurring**

Start Date: 2018/04/18 Time Zone: America/Chicago

Repeat Every: 1 days

Beginning: 7:00pm

Duration: 14 hours

Repeat Until: No end date

Summary: From 7:00pm to 9:00am Daily

[Preview planned recurrences](#)

3 Add a message

Preview ☒ Edit [Markdown supported](#)

Monitor: Postgres Monitoring Issue will be on downtime M-F 7pm to 9am @sellari9@gmail.com

4 Notify your team

Tre Sellari

- **Key Value Point for Monitoring and Alerting**
 - “without knowledge action is useless and knowledge without action is futile” – Abu Bakr
 - Collecting data is great, but without detection, monitoring, alerts, notifications for someone to get on the issue, then why are we collecting metrics. Action must be taken, and it must be taking timely and effectively.
 - Combining Datadog’s metric collection, anomaly detection, scoping, and correlations, monitoring, and alerting you can resolve issues in efficiently. To break it down a little more.... Time is money! The more time we let issues linger, the more the “cost”.

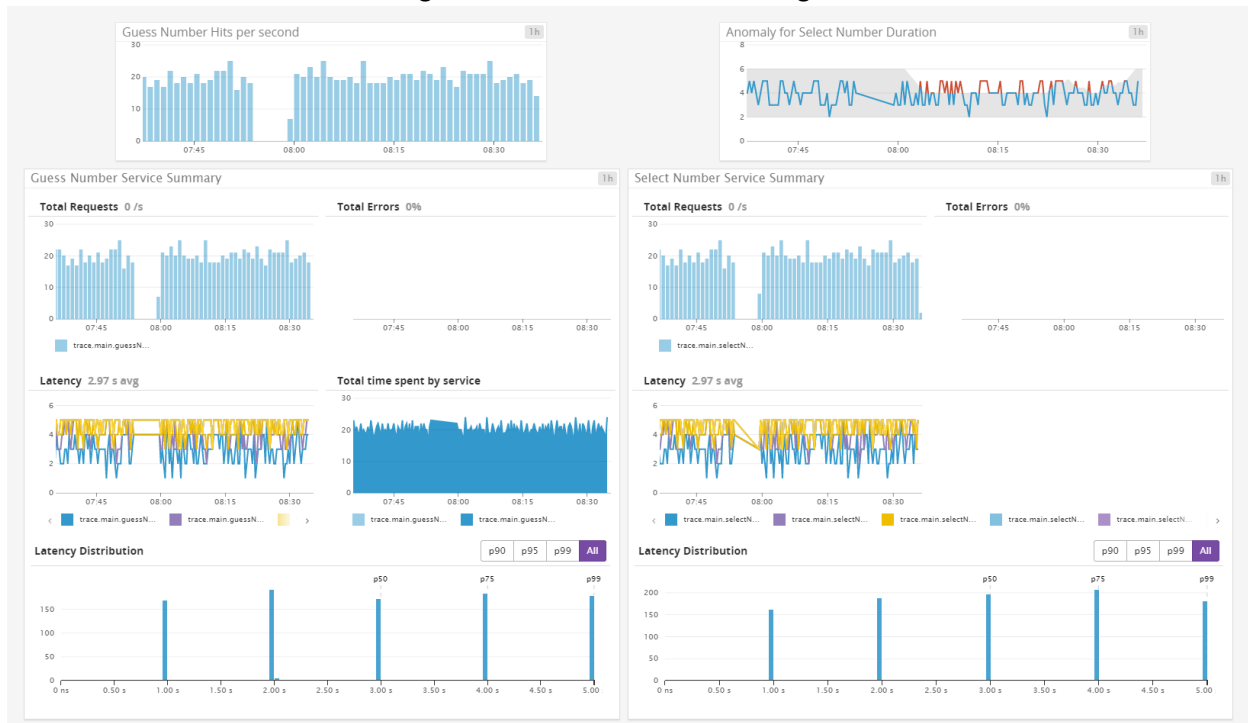
Day 3

- Application Performance Monitoring Setup
 - Python
 - Java

Setup of Python APM

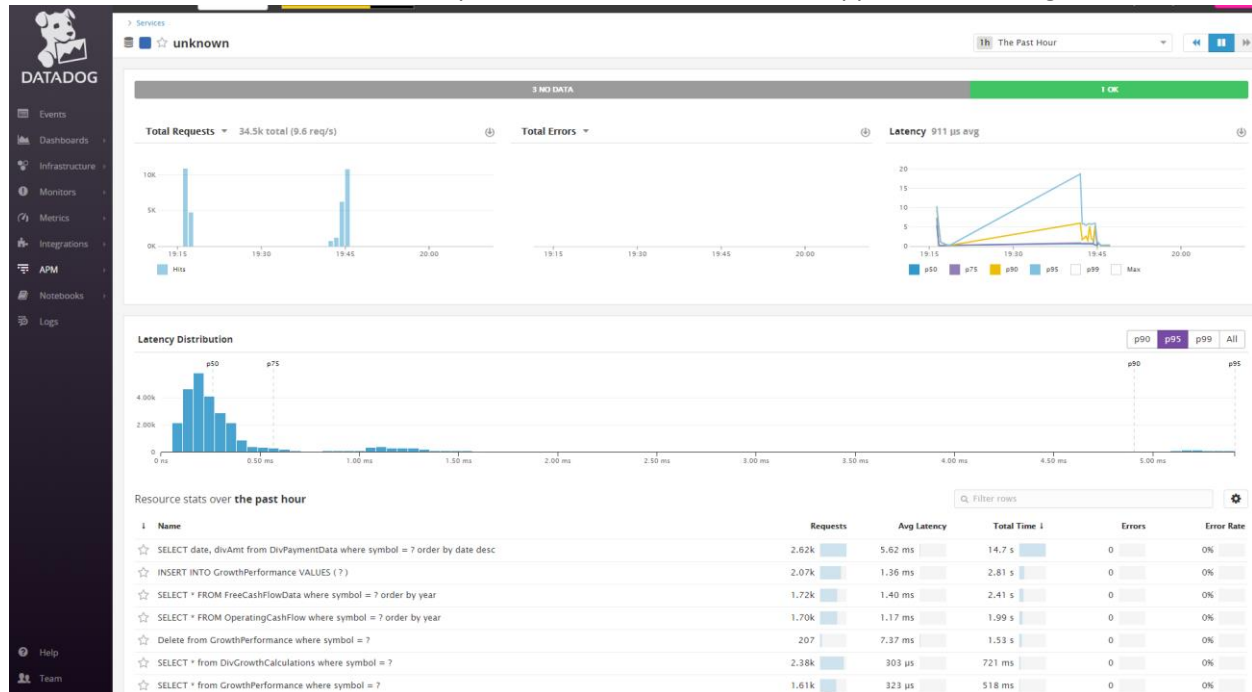
The selected application was a random guessing game Python application, provided by XYZ.

Integrations were imbedded into the application and data was pulled into Datadog. A dashboard was then created to view the data and gather valuable metrics and insight.



The second application used as part of the APM portion was a Java Swing and with a backend database. Using the Java APM jar, which was quickly downloaded from Datadog and added to a local IDE, APM data was immediately pulled into Datadog when running the application. A quote from the developer was, “that was surprising and satisfyingly easy”

Here is a screenshot of the APM data pulled in within 30 mins of the application running:



You will also notice that high level overview data, requests, and single sql queries are all now accessible in a single interface.

Summary

A 3 day POV was fully completed by Datadog at XYZ. All expectations were from XYZ were met and proved by Datadog.

On the first day the Datadog instance was initialized for XY, accounts created, and Host Agents were installed and configured, and a database integration was setup.

The hosts were:

- Windows 7 run on a laptop
- Windows 10 run on a desktop
- Linux host running Docker

Within moments of the agents installed and started, metric data was available and useable in Datadog. This was a quick proof of value to XYZ to see data so quickly from the infrastructure, with very minimal setup and configuration.

Next the PostgreSQL database integration was setup. Just like the host integrations, data was populated in Datadog almost instantly.

On day two dashboards, monitors, and alerts were setup in Datadog. This allows XYZ to have valuable insight metrics from all the components of the application. With anomaly detection XYZ can distinguish between normal and abnormal metric trends. The monitors and alerts allows XYZ to take quick and decisive action to resolve issues that could be cause problems for their end users.

On day 3 the Application Performance Monitors were setup with a Python application and a Java Swing application. With this data now in Datadog XYZ will quickly be able to identify performance bottlenecks. Also, now with data from both the infrastructure and application performance correlations between errors, events, and metrics can be made.