Grafana @ eBay - Moving Monitoring Visualizations from custom UIs to Grafana Plugins

Vijay Samuel

Github: @vjsamuel Twitter: @vjsamuel_


Many moons ago…

Monitoring and Logging UIs were custom

Logs

Built Custom Log Viewer tools

Developers would build custom interfaces using HTML CSS and JS libraries.

Users instrument code with internal frameworks to ship logs.

The Custom UIs allowed users to search and view behaviors of their applications.

Metrics

No different from logs

Users instrument code with internal frameworks to ship metrics.

Custom UIs allowed users to view their metrics and set up alerts.

Quality of UIs were greatly dependent on the person building

ebay

I’m not a frontend guy …

How my frustrations with building UIs led to a PoC on Grafana!

A First Attempt At Grafana!

The Hack

Modify the OpenTSDB data source on Grafana HEAD to understand our internal TSDB's request/ response formats

Dashboards

Build out scripted dashboards for one of our internal monitoring components.

Distribute the dashboards to folks who do on call to see if it really helps

What was missing?

It wasn't clean

The PoC was used within the team and it was a dirty hack to modify source code directly. It wasn't maintainable.

Missing Features

Basic features like drawing a graph and aliases were present.

Still lacked support for features like templating, annotations.

The PoC was lost in the sands of time.

Until…

Folks from DB Ops loved Grafana so much that they decided to help out!

Thank you Steven West, Auston McReynolds!!!

ebay

How did they help?

Clean things up

What was a dirty hack was now a proper Grafana data source plugin.

Plugin was still a modified grunt generated file ☺

Add Docker Support

Created a Dockerfile and build scripts to create a container with the data source loaded.

How I took it and ran with it!

Rebase often

No more dirty changes. Rebase often and roll out new containers with most recent version of Grafana.

Also had Kubernetes specs for folks to easily deploy.

Build a community

Grafana was still a pet project at this point

Evangelize the product and gain more adoption.

ebay

The big break through…

SRE decides to revamp all their monitoring dashboards with the help of Grafana!

Thank you Satish Sambasivan!!!

SRE takes it to the next level

Overlay data

Overlay application metrics with change executions in the system.

Provides hosted dashboards

Hosted dashboards with key indicators for site applications provided as templated dashboards for product developers by SRE.

Contribute code

SRE not only increased adoption to the system but also submitted PRs to enhance the plugin.

(Still a grunt generated file)

Requests Monitoring team to support Grafana!

Life has now become so easy for SRE that they want the monitoring team to officially support Grafana!

Pool Health (By Four Golden Signals)

Last 3 hours

> Errors (Host Level) *(6 panels)*

> Errors(Montage) : Choose "Montage Pool" *(3 panels)*

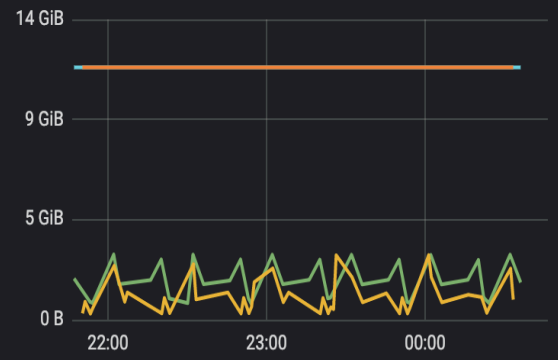
> Traffic (Colo Level) *(6 panels)*

> Traffic (Source: OLAP2) *(3 panels)*

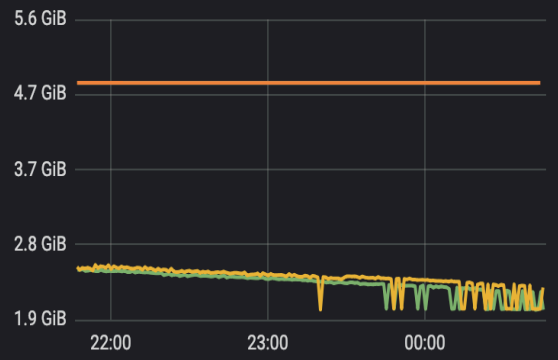
> Traffic (Host Level) *(3 panels)*

Saturation (Colo Level)

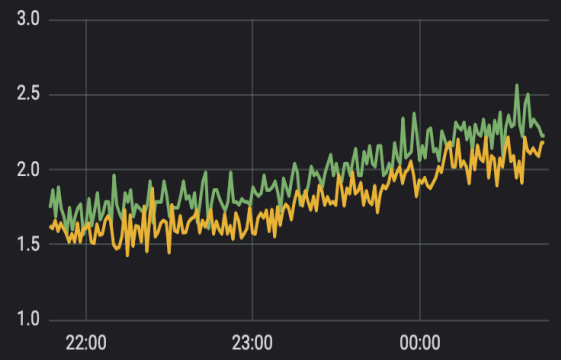
Physical Memory (Total vs Available)



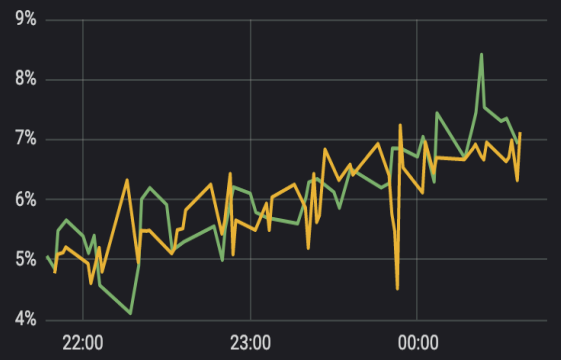
JVM Memory (Total vs Available)



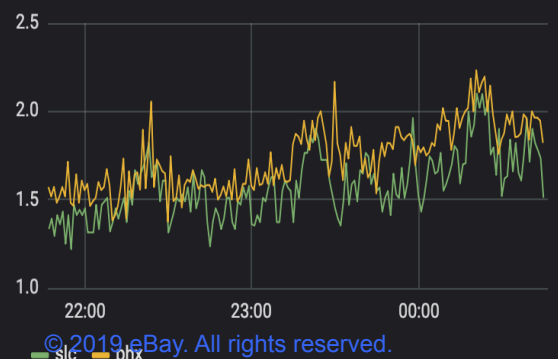
GC Count



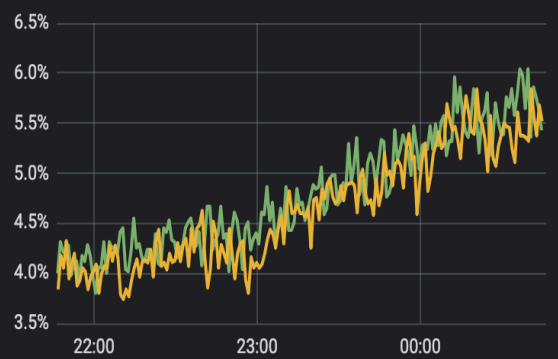
Busy Application Threads Pct



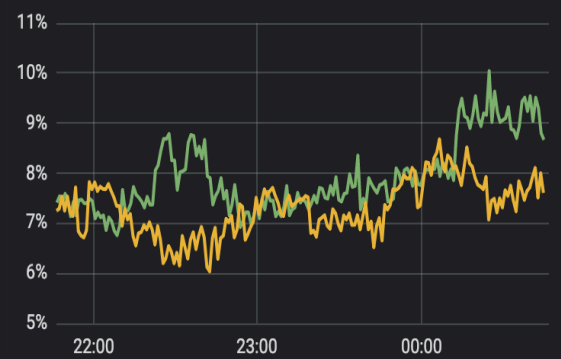
GC Overhead (x 10)



JVM CPU Usage



CPU Usage



LB Connections (Colo)

We said YES!

A whole new make over

Throw away that JS file!

Remove grunt generated code and do it right this time.

Provide a hosted solution

Move away from a model where folks spin up their own in favor of a single hosted solution that is managed by the monitoring team.

Add widgets for logs/ events

Provide a single interface for logs/ events/metrics on Grafana with the help of internal widgets.

Log viewer plugin provides a tail like experience.

Build out more features

Use Grafana's custom apps to build out alerting experiences based on internal alerting APIs.

Implement annotations, eBay specific authentication etc.

ebay

What happened in the backend

Move away from custom APIs

Instead of building custom APIs to push logs/metrics/ events we decided to use cloud native mechanisms.

A newbie can log to file

Anyone should be able to write logs to either stdout or to a log file. Our platform would collect and ship the data.

This allows new hires to onboard faster.

Same would apply for metrics

Instrument code with well known libraries like Prometheus.

The platform would scrape metrics and store them!

Do all of this as a community player!

Contribute to various projects like Elastic Beats, Flink etc. with features and bug fixes.

Keep custom code in plugins.

ebay

Lessons learned…

Be closer to community

Never resort to in house code changes unless working with community fails.

Give code back to the community when new features are built.

Creating dashboards is easy

The cost of building out data source plugins is far cheaper than building out entire UIs.

Rome was not built in a day but our custom data source was!

Cloud Native is everything

Moving away from proprietary APIs and into a cloud native model helped onboard use-cases that weren't conventionally possible.

We should've adopted Grafana sooner!

This was almost 4 years in the making.

This should've been a no brainer ☺

ebay

Thank You