Explore your Prometheus data in Grafana

David Kaltschmidt
@davkals

PromCon 2018
I’m not Carl

Carl is having a great time on parental leave
I’m David

All things UX at Grafana Labs

If you click and are stuck, reach out to me.

david@grafana.com

Twitter: @davkals
Grafana

From
Dashboarding solution

To
Observability platform
**Instant query**

queries only the last datapoint best displayed in table panel

<table>
<thead>
<tr>
<th>Table</th>
<th>General</th>
<th>Metrics</th>
<th>Options</th>
<th>Column Styles</th>
<th>Time range</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
</tr>
</tbody>
</table>

```plaintext
sum(kube_pod_container_resource_requests_cpu_cores) by (namespace)
```

Format as: Table

Instant: ✔️
<table>
<thead>
<tr>
<th>Namespace</th>
<th>CPUs (Requests)</th>
<th>CPUs (Limits)</th>
<th>CPU Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>prod</td>
<td>40.35</td>
<td>38.00</td>
<td>16.89</td>
</tr>
<tr>
<td>dev</td>
<td>4.65</td>
<td>0.50</td>
<td>0.68</td>
</tr>
<tr>
<td>kube-system</td>
<td>1.24</td>
<td>-</td>
<td>0.07</td>
</tr>
<tr>
<td>default</td>
<td>0.74</td>
<td>0.69</td>
<td>0.13</td>
</tr>
</tbody>
</table>

```go
A. sum(kube_pod_container_resource_requests_cpu_cores) by (namespace)

B. sum(kube_pod_container_resource_limits_cpu_cores) by (namespace)

C. sum(rate(container_cpu_usage_seconds_total[5m])) by (namespace)
```
Step alignment

“Jumpy” rate graphs

Align start/end parameters to step
Heatmap panel released in 5.1

Query: rate(foo_metric_bucket[10m])
Legend format: {{le}}
Format as: Heatmap
Variable expansion in template variables

Summer intern project (Tobias, tak!!)
New built-in variable: $__range
Dashboard-range-based variables
Template variable expansion
Use range-based template variables in queries
Provisioning API

Define data sources and dashboards in files

Auto-reload on change

Allows version control of files

Provisioning Grafinana

In previous versions of Grafana, you could only use the API for provisioning data sources and dashboards. But that required the service to be running before you started creating dashboards and you also needed to set up credentials for the HTTP API. In v5.0 we decided to improve this experience by adding a new active provisioning system that uses config files. This will make GitOps more natural as data sources and dashboards can be defined via files that can be version controlled. We hope to extend this system to later add support for users, orgs and alerts as well.

Config File

Checkout the configuration page for more information on what you can configure in grafana.ini

Config File Locations

- Default configuration from $WORKING_DIR/conf/defaults.ini
- Custom configuration from $WORKING_DIR/conf/custom.ini
- The custom configuration file path can be overridden using the \(--config\) parameter

http://docs.grafana.org/administration/provisioning/
Grafana is now fully CI’ed

Test new features that are in master:

docker run -d --name=grafana -p 3000:3000 grafana/grafana:latest

https://hub.docker.com/r/grafana/grafana/
Prometheus adoption
PromCon 2016: 2.8K
PromCon 2017: 16K
PromCon 2018: 54K

PromCon 2016: 36K
PromCon 2017: 92K
PromCon 2018: 186K
New: Explore UI (Alpha)
Motivation

Dashboard UI is made for building dashboards, not query iteration.

“I just want to quickly...”
Motivation

once panel is found, difficult to interact with

overwhelming style and display options
Motivation

Incident response currently happens elsewhere!

User story:
As an SRE, I want to investigate a certain aspect or incident of my infrastructure. I’m already looking at the right dashboard panel and have formed a hypothesis. From here I want to quickly run variations of the panel’s query to verify my hypothesis. The panel I started from should stay unaffected.
Explore UI wireframes

<table>
<thead>
<tr>
<th>First tab</th>
<th>Second tab</th>
<th>3rd tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAPH</td>
<td>TABLE</td>
<td>BOTH</td>
</tr>
<tr>
<td>RUN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>rate(http_requests_total[5m])</th>
<th>4.2s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - rate(http_requests_total[5m])</td>
<td>3.2s</td>
</tr>
</tbody>
</table>

My tab

<table>
<thead>
<tr>
<th>GRAPH</th>
<th>TABLE</th>
<th>BOTH</th>
<th>Last 1 hour, Refresh 10s</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>rate(http_requests_total[5m])</th>
<th>4.2s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - rate(http_requests_total[5m])</td>
<td>3.2s</td>
</tr>
</tbody>
</table>
Demo: http://localhost:3333/explore
Enable Explore UI

Behind feature flag. To enable, edit Grafana config ini file

[explore]
enabled = true

Set up a datasource that supports Explore, e.g., Prometheus.

Will be released in 6.0
What we're working on
Explore UI needs to be refined

still behind feature flag, feedback welcome: @davkals or david@grafana.com

Prometheus metric metadata from HELP line in the exposition

support for completion for
foo_metric / on( ) group_left( ) bar_metric

design doc for explore components for other datasources

bring query field over to panel editor
MultiStat panel [https://github.com/grafana/grafana/grafana/pull/12620](https://github.com/grafana/grafana/grafana/pull/12620)
Dashboard management

Improve Grafana’s integration with Mixins (see Tom Wilkie’s talk)

Git integration to save changes back

Defaults for panel configs
One more thing...
Logging is coming to Grafana

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Line</th>
<th>Message</th>
<th>File</th>
<th>Class</th>
<th>Method</th>
<th>Parameters</th>
<th>Duration</th>
<th>Memory Usage</th>
<th>Status</th>
</tr>
</thead>
</table>
Tak for listening

UX feedback to david@grafana.com
@davkals
Tak for listening

UX feedback to
david@grafana.com
@davkals