Fool-proof K8s dashboards for sleep-deprived on-calls

David Kaltschmidt
@davkals

Kubecon 2019

I'm David

Working on Explore, Prometheus, and Loki at Grafana Labs

Previously:
Unifying Metrics/Logs/Traces at Kausal,
Work on WeaveScope

david@grafana.com

Twitter: @davkals

Grafana Labs

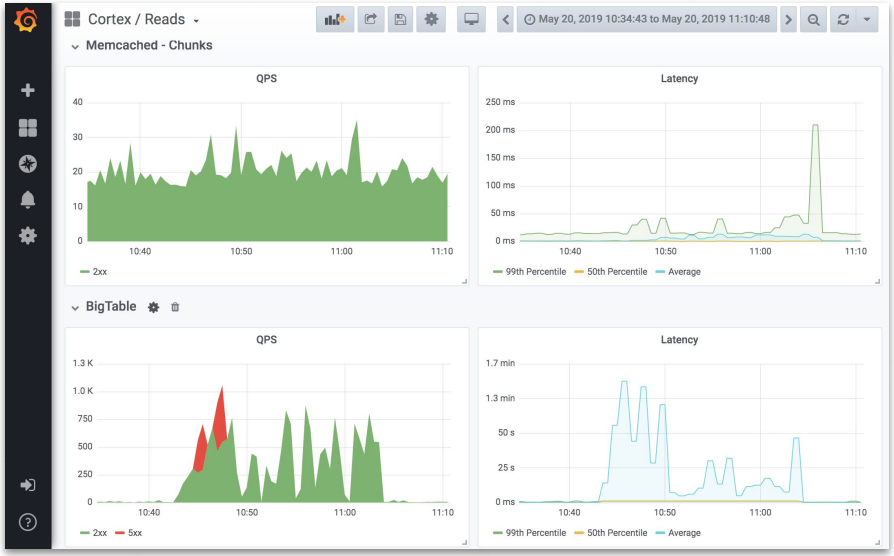
Cognitive load

In which direction do I have to pull the little lever to open the metro door?

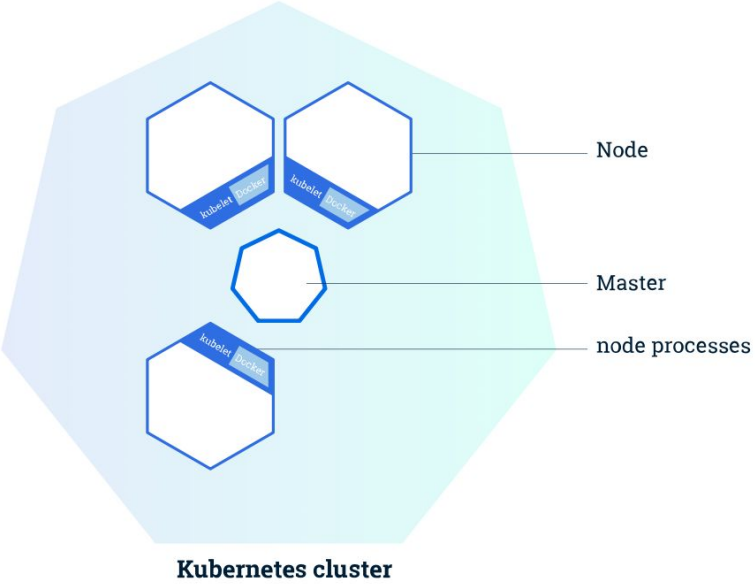
Dashboarding for Kubernetes on-calls

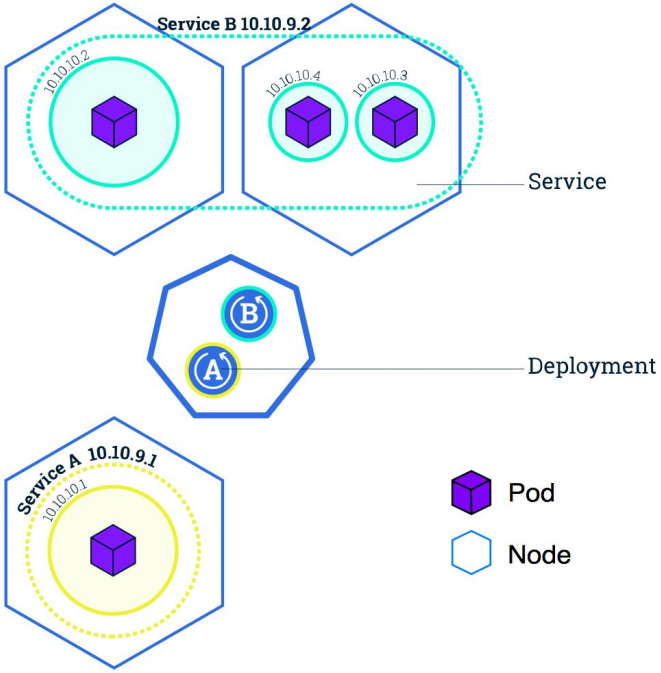
On-call

- Good on-call is debugging and follow-up, improving things for the rest.
- Bad on-call is mostly incident response where every minute counts



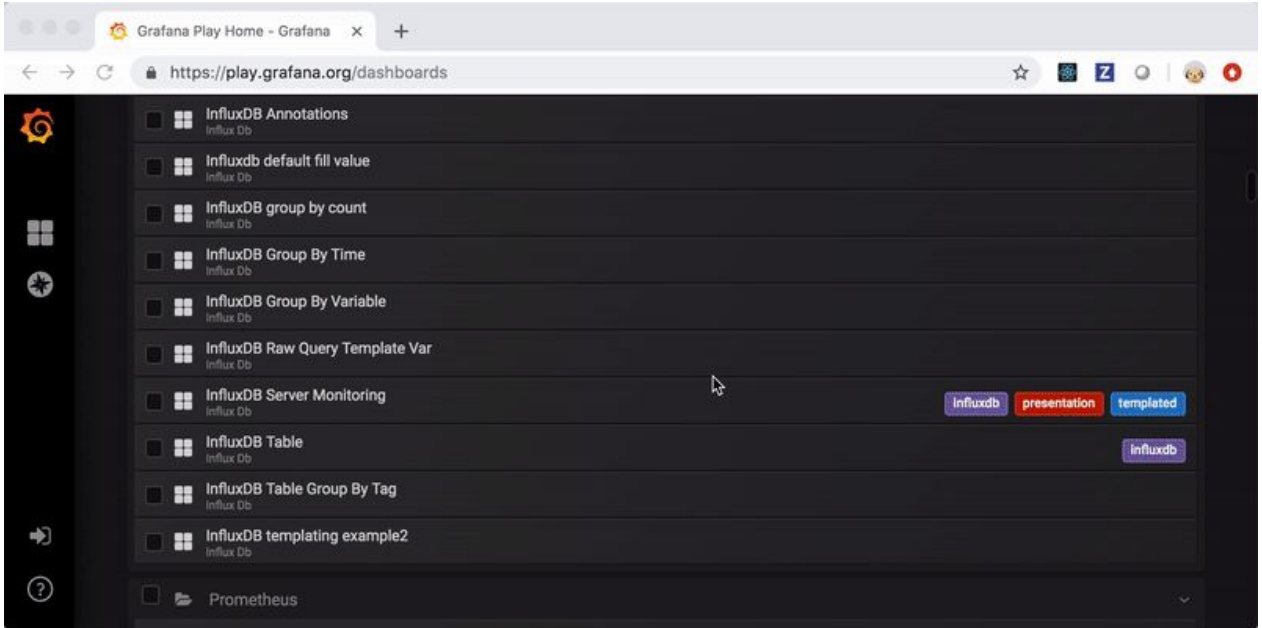
On-call for Kubernetes





https://kubernetes.io/docs/tutorials/kubernetes-basics/

The path to 1,000 dashboards



Introducing DMM:

Dashboarding Maturity Model

Dashboarding maturity levels

Low

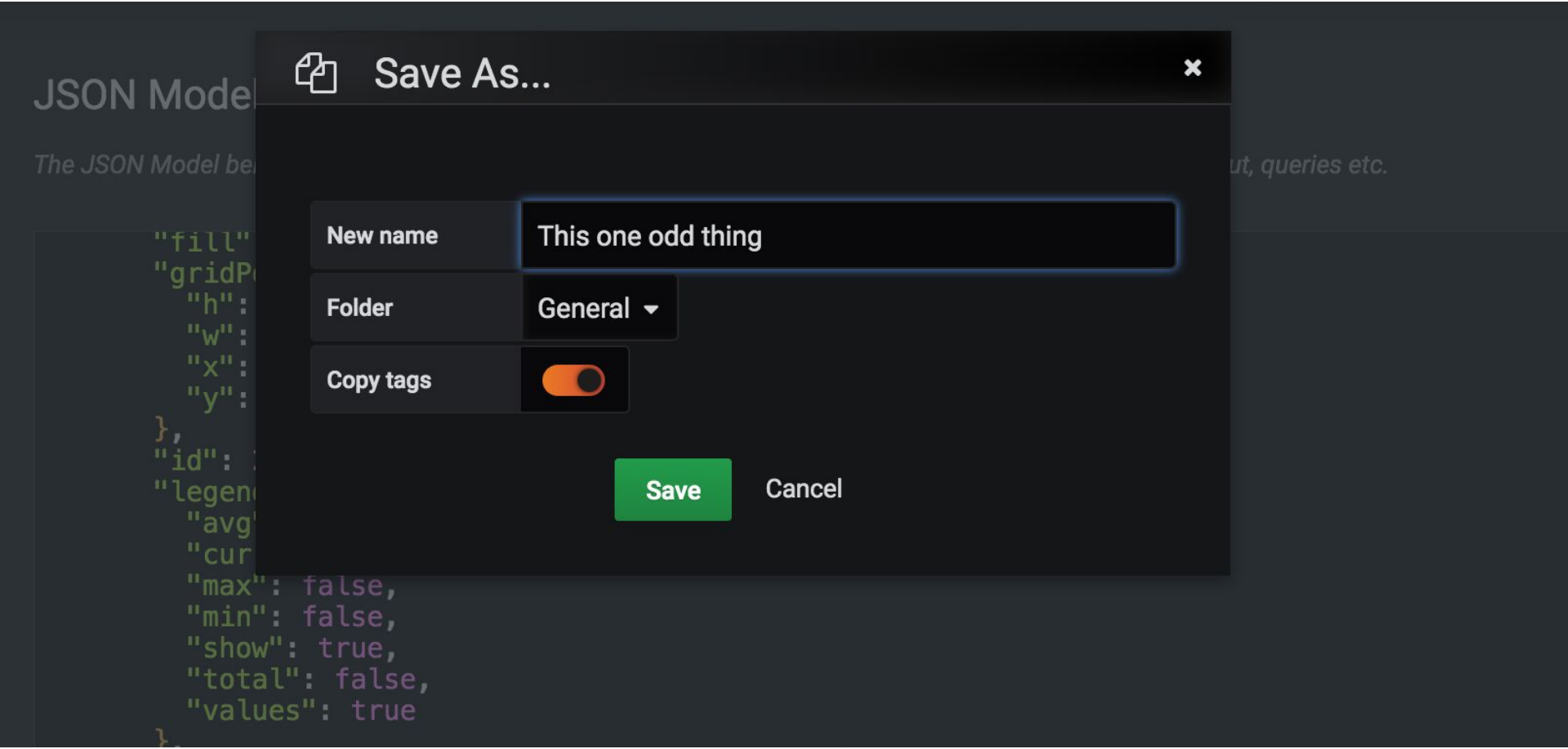
Default state
(no strategy)

Medium

Managing use of
methodical dashboards

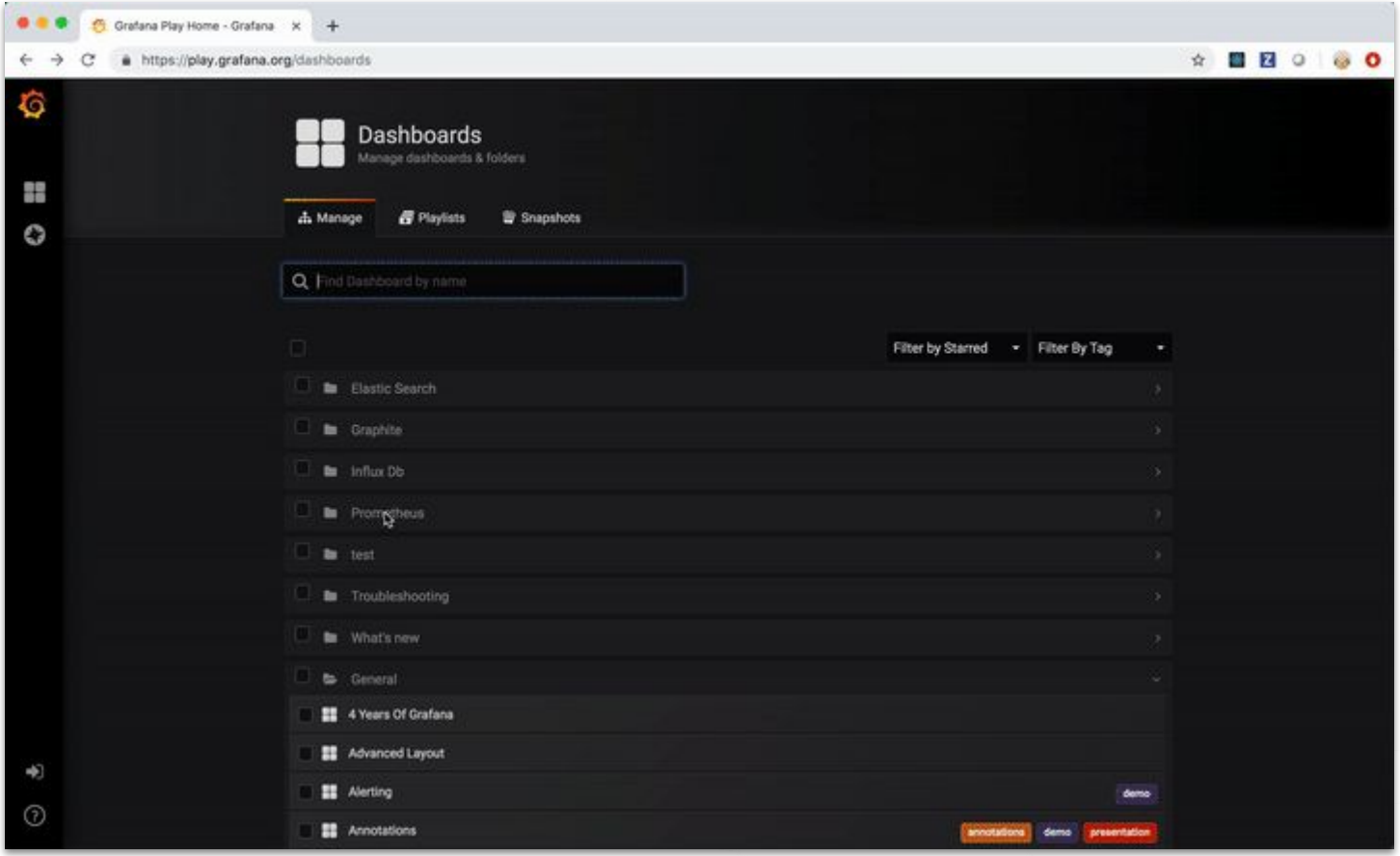
High

Optimizing use,
consistency by design



Low maturity: Sprawl

Low maturity: No version control



Low maturity: Browsing for dashboards

Dashboarding maturity levels

Low

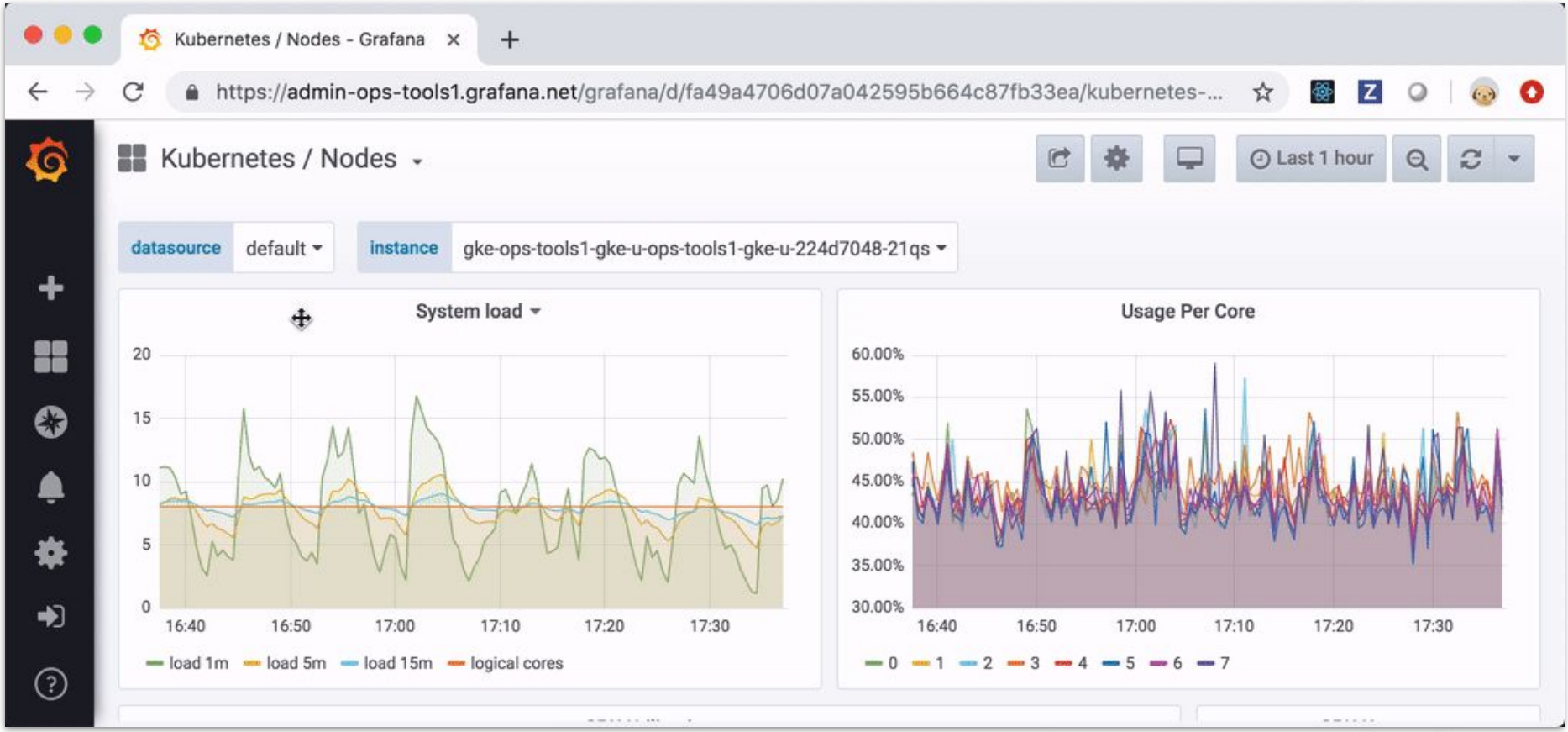
No strategy
(default state)

Medium

Managing the use of
methodical dashboards

High

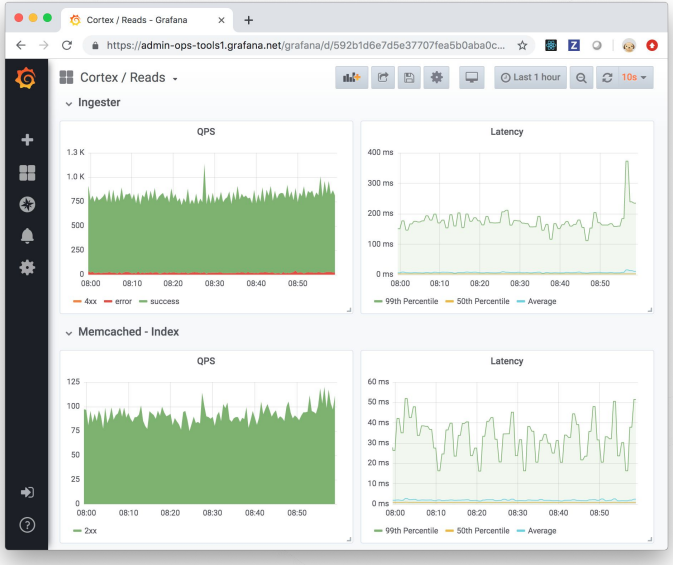
Optimizing use,
consistency by design

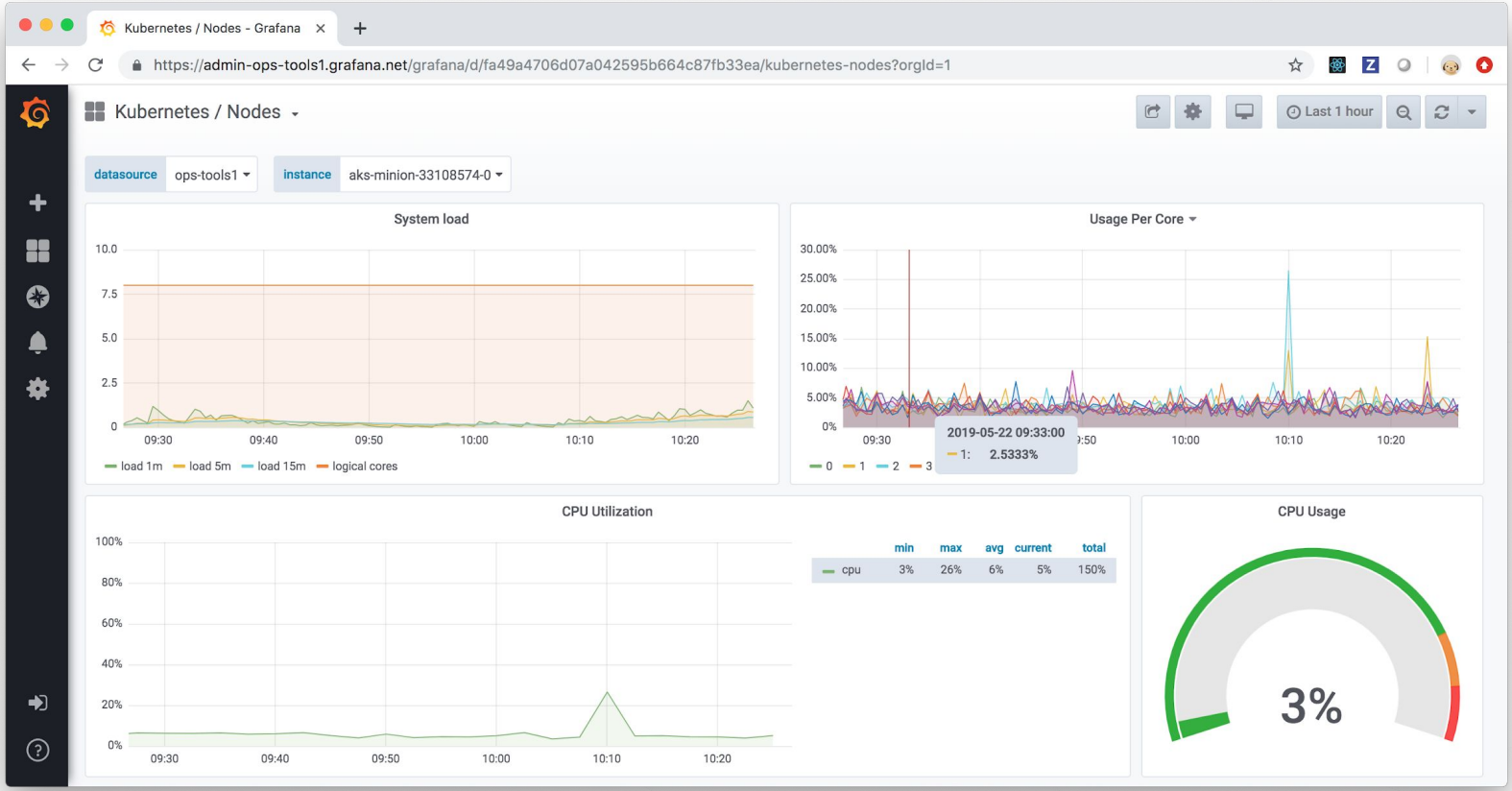


Medium maturity: Prevent sprawl by using template variables [Docs]

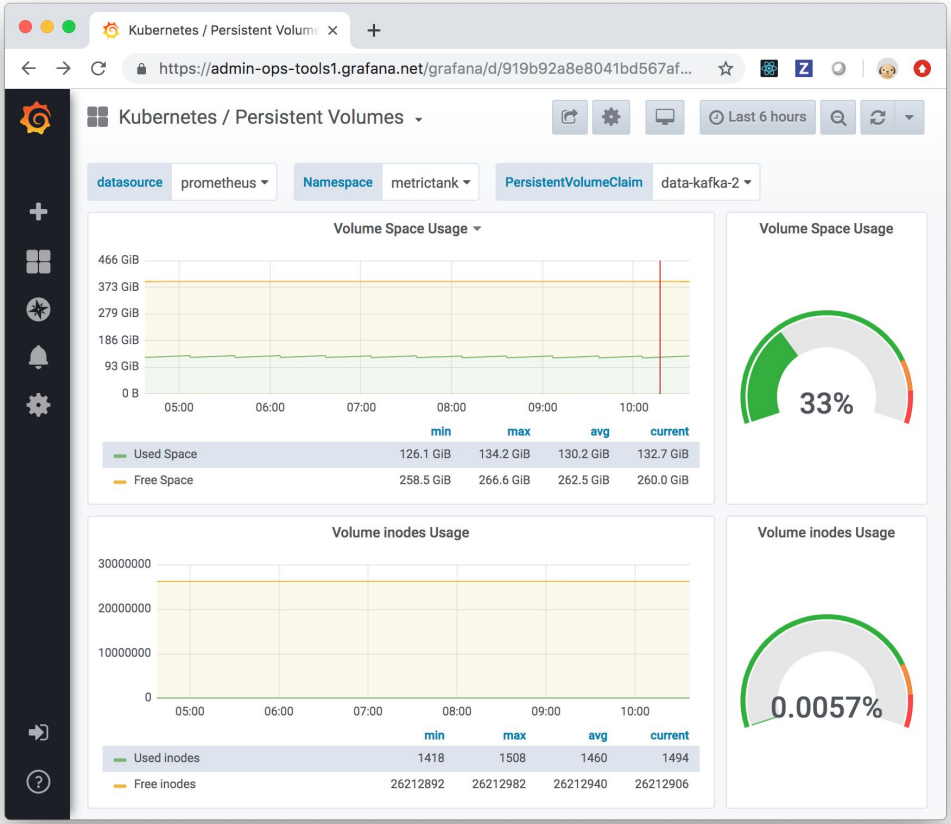
Medium maturity: Methodical dashboards

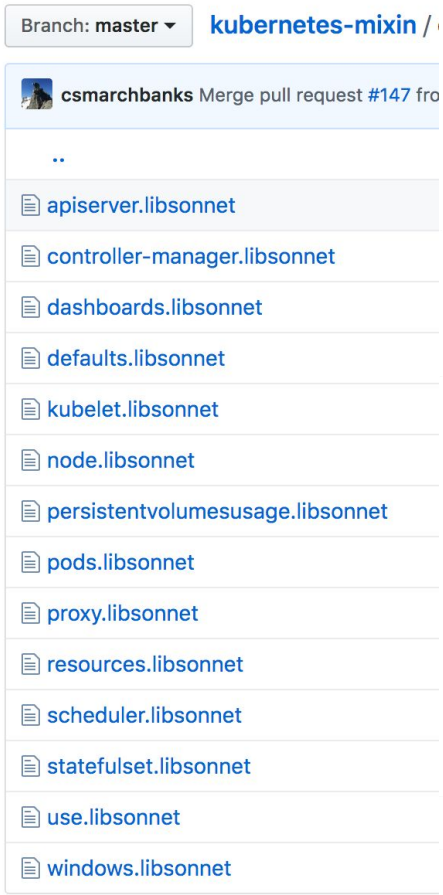
- USE method for resources:
  For each resource measure utilization, saturation, errors
- RED method for services:
  For each service measure request and error rate, and duration
- Your own method





Medium maturity: USE method dashboards (part of the Kubernetes mixin)

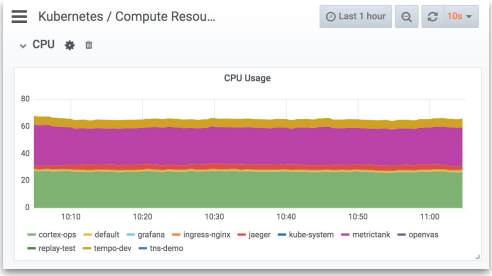


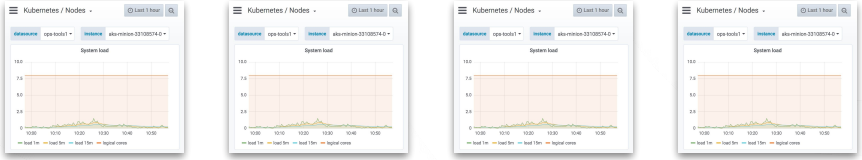


Medium maturity: **Peer-reviewed** K8s dashboards in the Kubernetes mixin

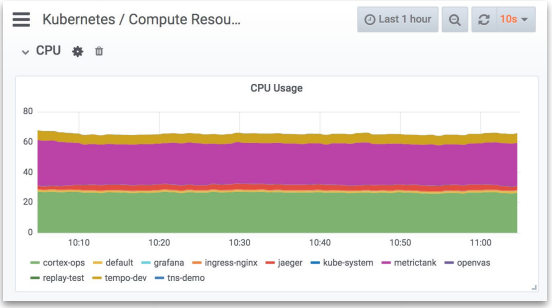
Medium maturity: Hierarchical dashboards

- Summary views with aggregate queries
- Queries have breakdown by next level
- Tree structure reflecting the k8s hierarchies

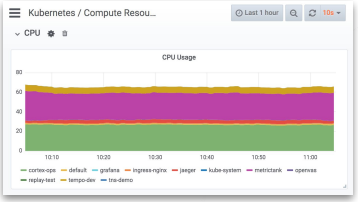


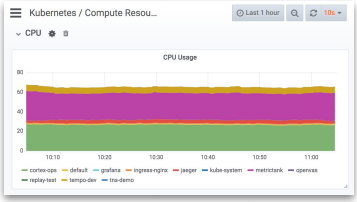


Cluster

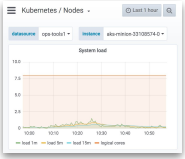


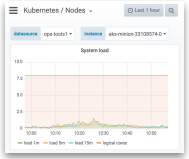
Namespace

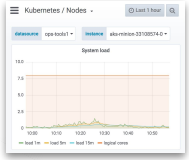


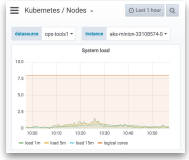


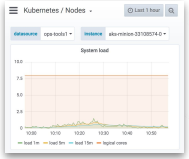
Pod

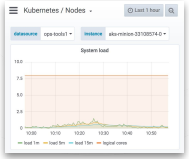










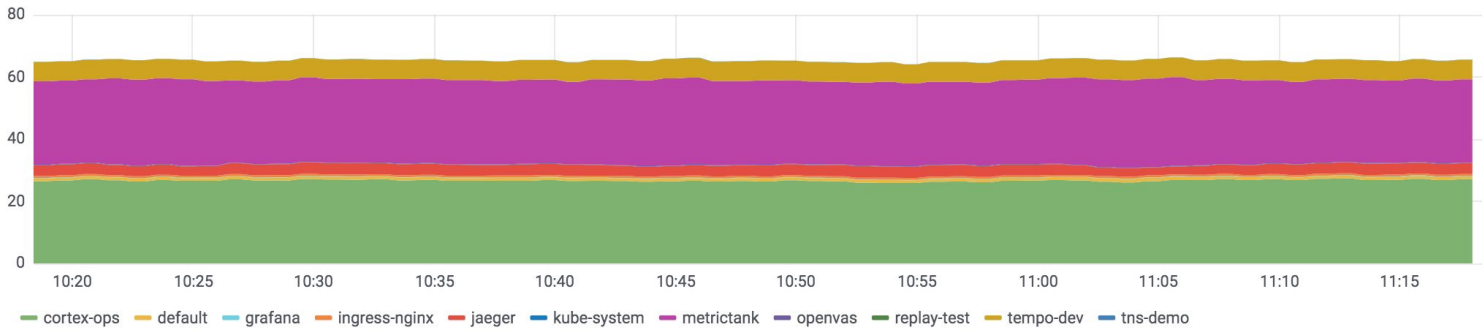


Medium maturity: Hierarchical dashboards along K8s hierarchies

Kubernetes / Compute Resources / Cluster

Last 1 hour

CPU Usage



CPU Quota

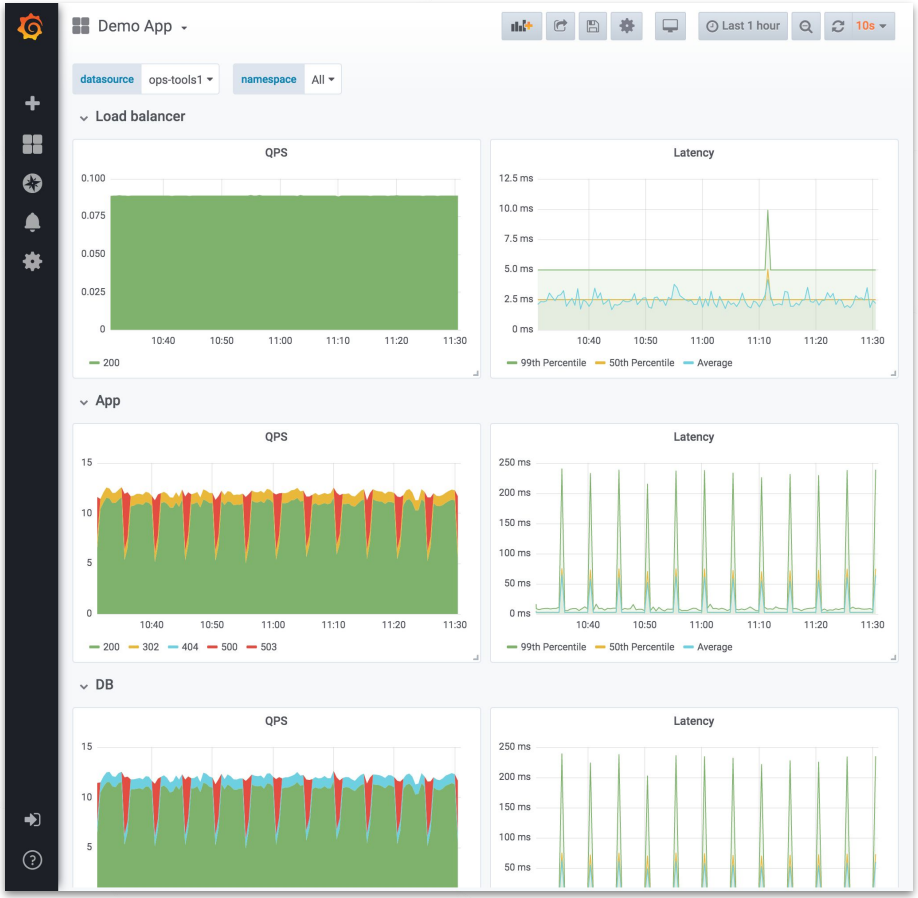
CPU Quota

Namespace	Pods	Workloads	CPU Usage	CPU Requests	CPU Requests %	CPU Limits	CPU Limits %
cortex-ops (Drill down to namespace)	-	-	26.99	63.30	42.64%	31.30	86.24%
openvas	-	-	0.00	1.00	0.13%	2.00	0.06%
tns-demo	-	-	0.07	-	-	-	-
tempo-dev	-	-	6.29	14.20	44.27%	46.20	13.61%
kube-system	-	-	0.16	2.14	7.30%	0.25	63.10%
jaeger	-	-	3.31	0.15	2204.07%	3.00	110.20%

Medium maturity: Hierarchical dashboards with drill-down to next level

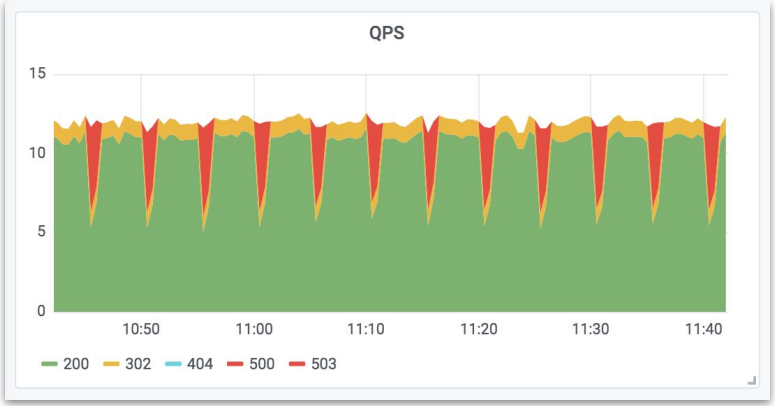
Medium maturity: Service hierarchies

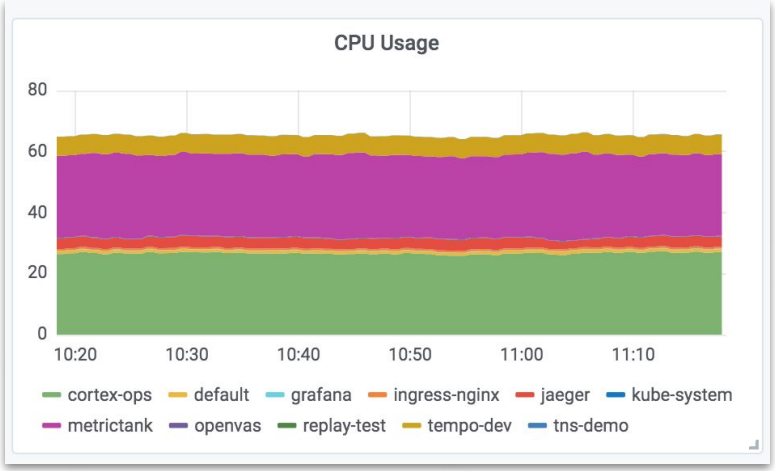
- RED method
- One row per service
- Row order reflects data flow



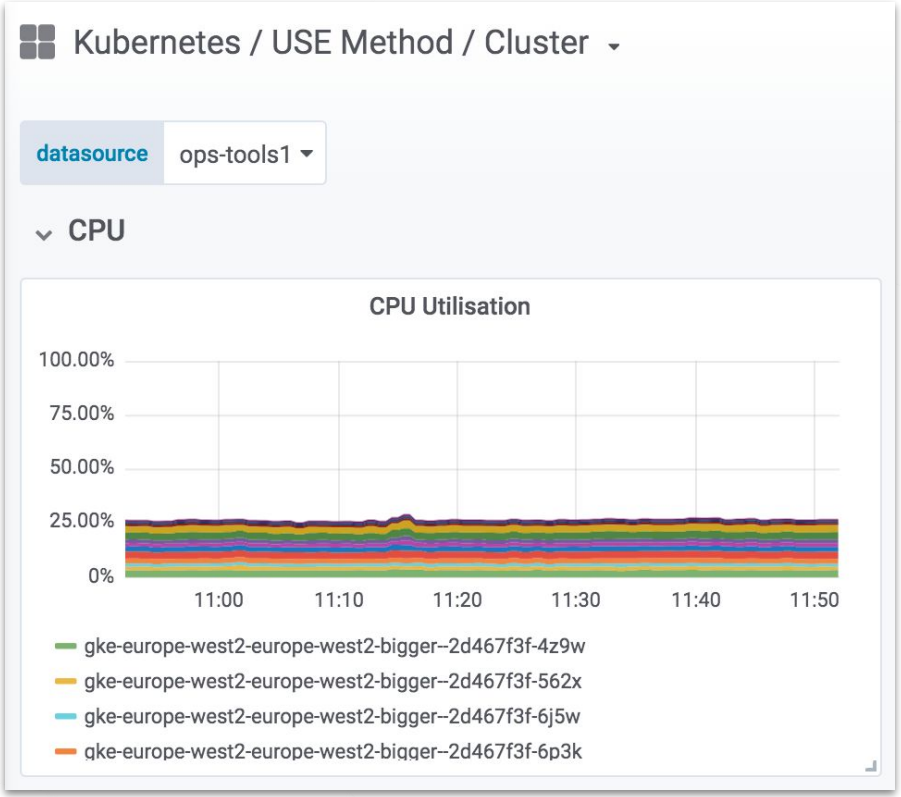
Medium maturity: Expressive charts

- Meaningful use of color
- Normalize axis where you can
- Understand the underlying metrics

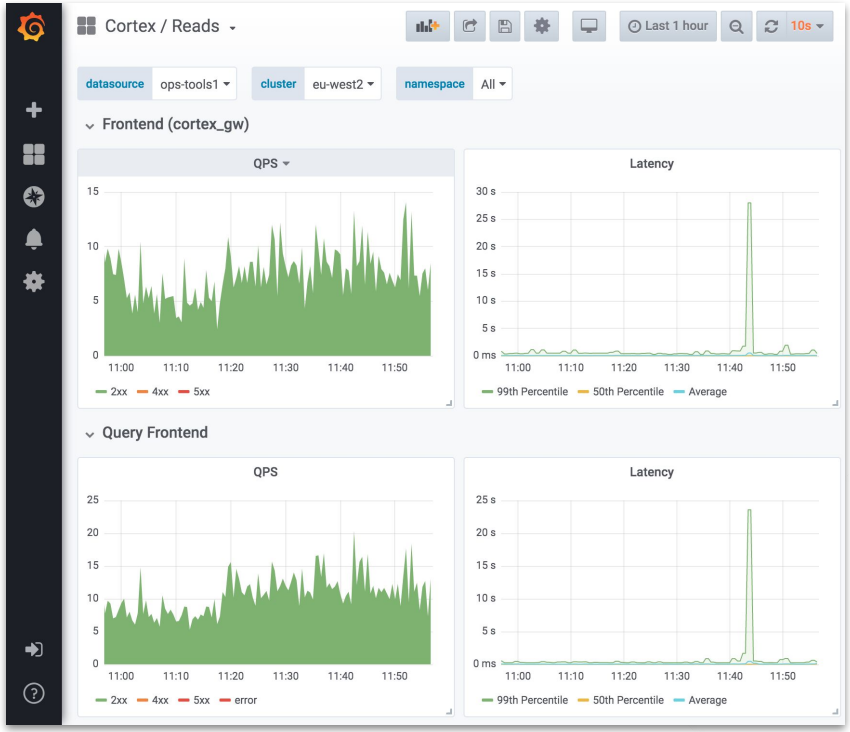




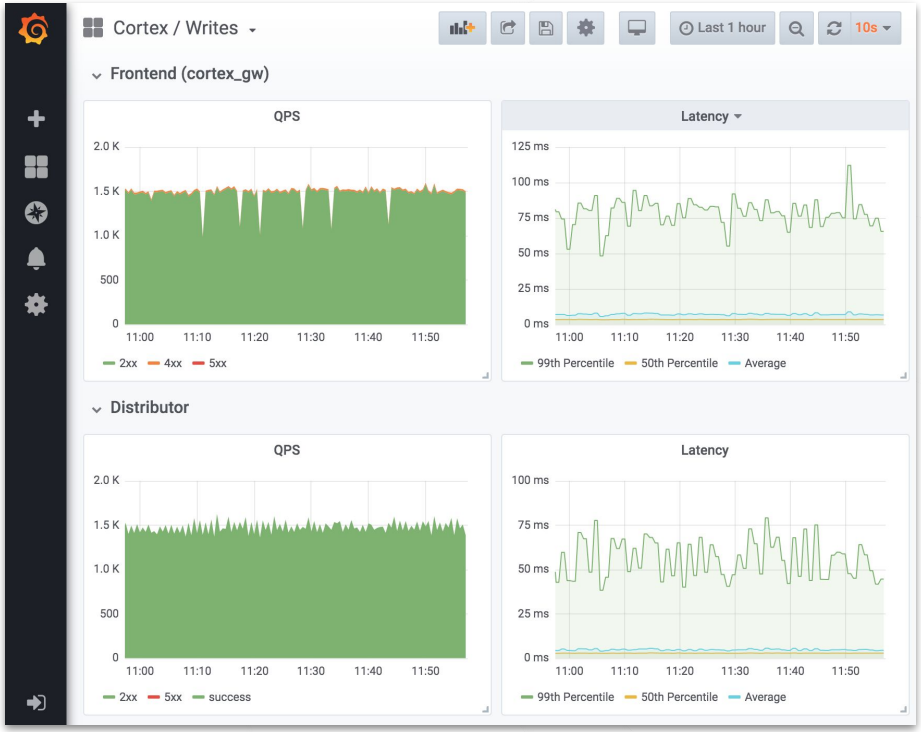
```
{
  // CPU utilisation per node, normalized by cluster-wide CPUs
  record: 'node:cluster_cpu_utilisation:ratio',
  expr: |||
    node:node_cpu_utilisation:avg1m
      *
    node:node_num_cpu:sum
      /
    scalar(sum(node:node_num_cpu:sum))
  ||| % $._config,
},
```



Medium maturity: Normalized charts (part of Kubernetes mixin)



Read API

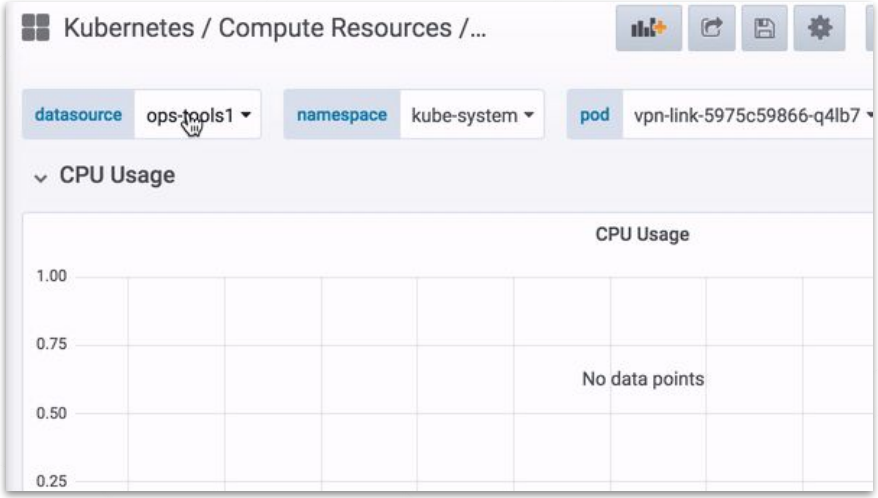


Write API (1000x)

Expressive dashboards: Split service dashboards where magnitude differs

Medium maturity: Directed browsing

- Template variables make it harder to “just browse”
- Most dashboards should be linked to by alerts
- Browsing is directed (drill-down)



Medium maturity: Managing dashboards

- Version controlled dashboard sources
- Currently by copy/pasting JSON
- RFC in our [design doc]

grafana / grafana / grafana

Watch 1,214 Star

Code | Issues 1,746 | Pull requests 88 | ZenHub | Projects 4 | Wiki | Insights

Dashboard workflow for provisioned dashboards #13823

Open bergquist opened this issue on 25 Oct 2018 · 13 comments

bergquist commented on 25 Oct 2018 • edited Member

Grafana currently allows operators to load dashboard JSON files from the filesystem into Grafana using provisioning. The user can edit the dashboard in the browser but cannot save it. To update the dashboard the user has to copy the edited JSON to the files themselves.

One idea that we have is to allow Grafana to...

1. Provision dashboards directly from Github
2. Create pull requests within Grafana
3. Review pull requests within Grafana by rendering dashboards from pull requests.

Cognitive load

On which side do you usually swipe your tickets at the turnstile?

Dashboarding maturity levels

Low

Default state
(no strategy)

Medium

Managing use of
methodical dashboards

High

Optimizing use,
consistency by design

High maturity: Optimizing use

- Actively reducing sprawl
- Regularly reviewing existing dashboards
- Tracking use

High maturity: Consistency by design

- Use of scripting libraries to generate dashboards
  - grafonnet (Jsonnet)
  - grafanalib (Python)
- Consistent attributes and styles across all dashboards
- Smaller change sets

```
g.dashboard('Cluster').addRow(
  g.row('CPU').addPanel(
    g.panel('CPU Utilisation') +
    g.queryPanel('node:cluster_cpu_utilisation:ratio') +
    g.stack +
    { yaxes: g.yaxes({ format: 'percentunit', max: 1 }) },
  ).addPanel(
    g.panel('CPU Saturation (Load1)') +
    g.queryPanel(|||
      node:node_cpu_saturation_load1: /
scalar(sum(min(kube_pod_info) by (node)))
    |||) +
    g.stack +
    { yaxes: g.yaxes({ format: 'percentunit', max: 1 }) },
  )
)
```

Prometheus Monitoring Mixins
Talk at PromCon 2018
by Tom Wilkie
https://www.youtube.com/watch?v=GDdnL5R_l-Y

High maturity: Use of mixins or other peer-reviewed templates

Future workflow: Dashboard as code

- Live edit JSON and preview dashboards
- Live edit Jsonnet or Python sources and preview in browser
- Open PR directly from Grafana

Dashboarding maturity levels

Low

No strategy
(default state)

- Everyone can modify
- Duplicate used regularly
- One-off dashboards
- No version control
- Lots of browsing

Medium

Managing use of methodical dashboards

- prevention of sprawl
- use of template variables
- methodical dashboards
- hierarchical dashboards
- expressive charts
- version control
- directed browsing

High

Optimizing use,
consistency by design

- active sprawl reduction
- use of scripting libraries
- use of mixins
- no editing in the browser
- browsing is the exception

DMM for oncalls:

Your dashboarding practices should reduce cognitive load, not add to it.

Thank you.

Don't be the Barcelona Metro of dashboards!

UX feedback to
david@grafana.com
@davkals

Grafana Labs