

Grafanalib

a Python EDSL for Grafana dashboards
<https://github.com/weaveworks/grafanalib>

Bryan Boreham, Director of Engineering



@bboreham

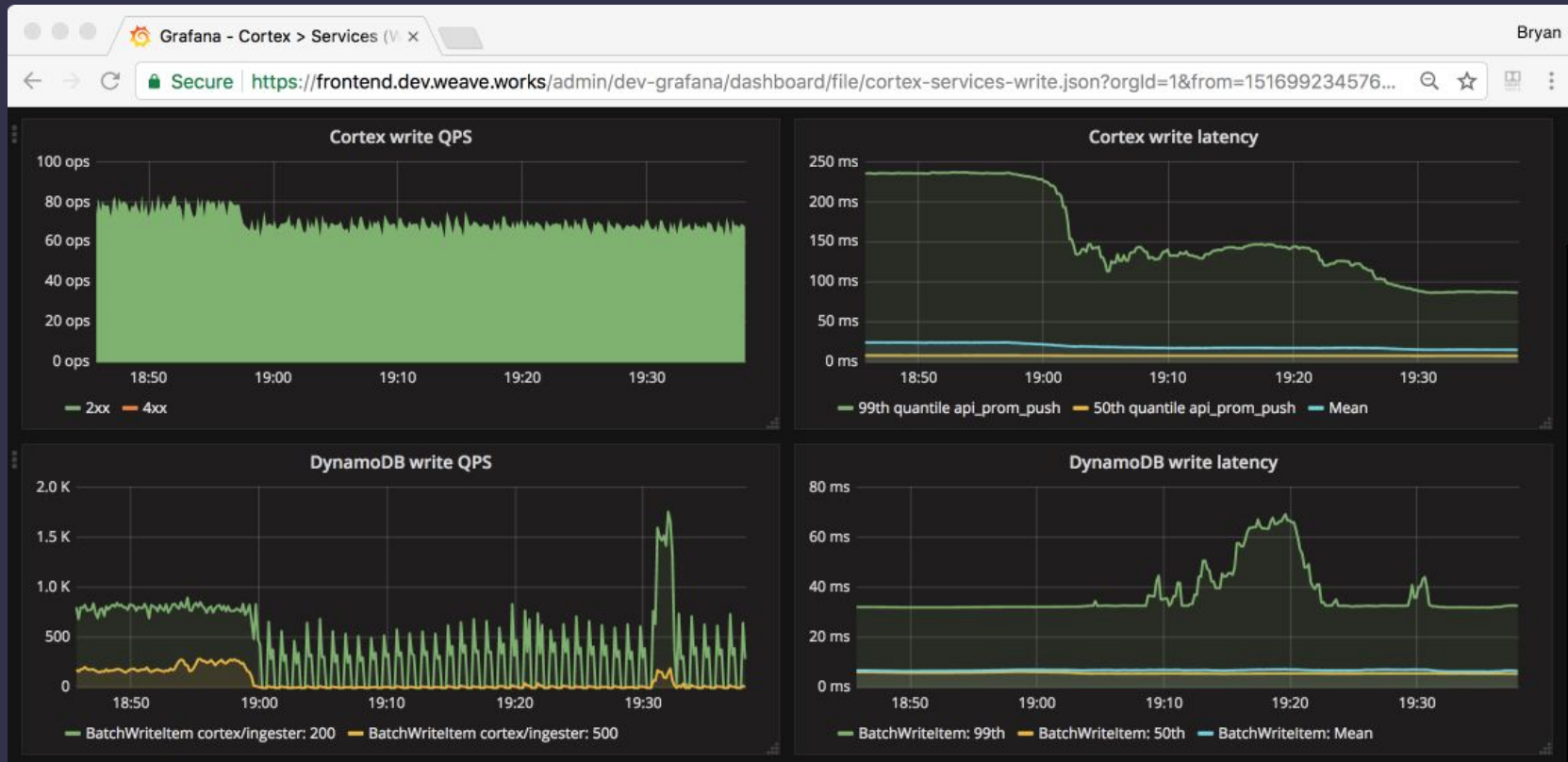


Who is working with...

- Grafana
- Prometheus
- Weaveworks



Grafana



Gitops

- All config is checked into a git repo
- All changes to infrastructure are a git commit
- Continuous sync from git to cluster

Gitops Benefits

- Complete history of who changed what
- Roll back config changes by reverting commits
- Run review processes on ops just like code

Grafanalib Graph definition

```
PromGraph(  
    title='Cortex write Latency' % (name,),  
    expressions=[  
        ('99th quantile', 'request_duration:99quantile'),  
        ('50th quantile', 'request_duration:50quantile'),  
    ],  
    yAxes=[  
        G.YAxis(format=G.MILLISECONDS_FORMAT),  
        G.YAxis(format=G.SHORT_FORMAT),  
    ],  
)
```

Higher-level functions

```
QPSGraph(  
  title="Cortex write QPS",  
  data_source=common.PROMETHEUS,  
  expressions=[  
    'request_rate {status_code=~"1.."}',  
    'request_rate {status_code=~"2.."}',  
    'request_rate {status_code=~"3.."}',  
    'request_rate {status_code=~"4.."}',  
    'request_rate {status_code=~"5.."}',  
  ]  
)
```

Write your own functions!

```
dashboard = G.Dashboard(  
    title="Scope > Services",  
    rows=[  
        scope_row('Collection', 'scope/collection'),  
        scope_row('Query', 'scope/query'),  
        scope_row('Control', 'scope/control'),  
        scope_row('Demo', 'extra/demo')  
    ]
```


Run the generator

```
$ generate-dashboard -o mydash.json \  
  mydash.dashboard.py
```

Grafanalib Summary

- Graph, Gauge, Sparkline, SingleStat
- Dashboard, Row, Column, Table, Grid
- Styles, Axes, Templates
- Prometheus, OpenTSDB, Zabbix
- `gfdatasource` `sidecar`

Questions?

<https://github.com/weaveworks/grafanalib>

Contributions welcome!

We are hiring!

Engineers in Berlin & SF

weave.works/hiring