Grafana Loki: *Like Prometheus, but for logs.*

Tom Wilkie, Feb 2019

Grafana Labs

Demo

Grafana Labs

Tom Wilkie *VP Product, Grafana Labs*

Previously*: Kausal, Weaveworks, Google, Acunu, Xensource*

Prometheus & Cortex maintainer, mixins authors etc

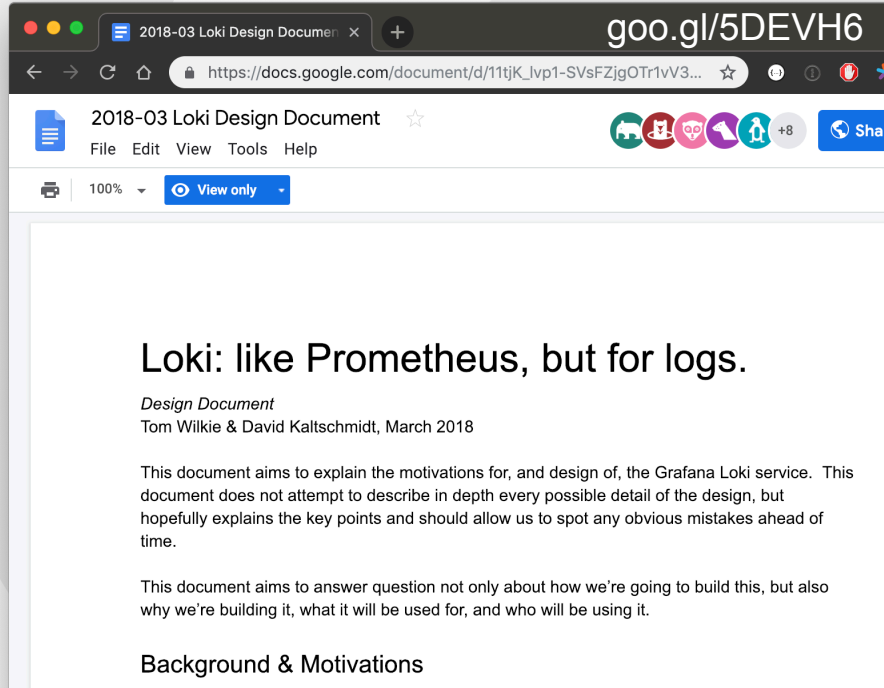
Twitter: *@tom_wilkie* Email: *tom@grafana.com*

Grafana Labs

Loki is a horizontally-scalable, highly-available, multi-tenant log aggregation system inspired by Prometheus.

03/18	Project started
12/18	Launched at KubeCon
12/18	#1 on HN for ~12hrs!
01/19	~5k GitHub stars

https://github.com/grafana/loki



#0 Simple and cost effective to operate

#1 Integrated with existing observability tools

#2 Cloud Native and Airplane Friendly

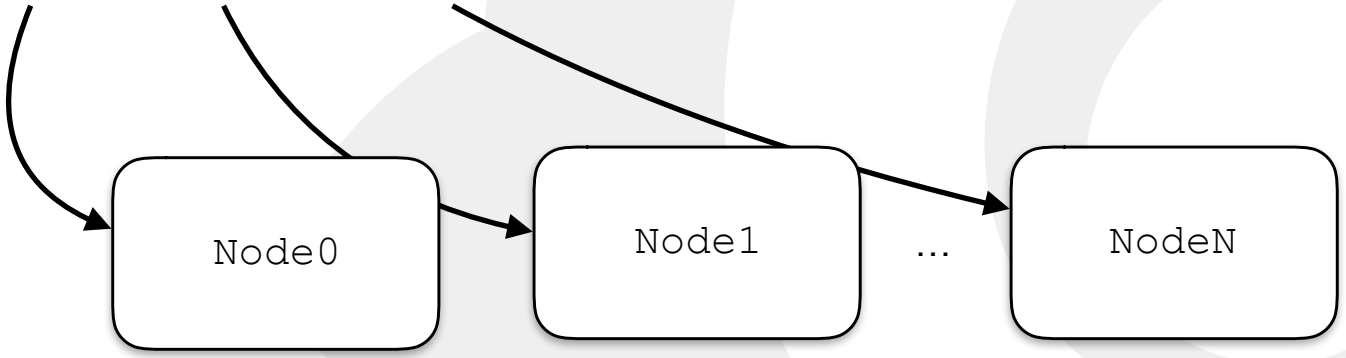
#0 Simple to scale

```
DEwMGIwZ => {
   time:  “2018-01-31 15:41:04”,
   job:   “frontend”,
   env:   “dev”,
   line:  “POST /api/prom/push...”
}
```

```
(“time", “2018-01-31 15:41:04”)   -> “DEwMGIwZ”
(“job”, “frontend”)               -> “DEwMGIwZ”
(“env”, “dev”)                    -> “DEwMGIwZ”
(“line”, “POST”)                  -> “DEwMGIwZ”
(“line”, “/api/prom/push”)        -> “DEwMGIwZ”
(“line”, “HTTP/1.1”)              -> “DEwMGIwZ”
(“line”, “502”)                   -> “DEwMGIwZ”
```

Existing log aggregation systems do full text indexing and support complex queries

```
("time", "2018-01-31 15:41:04")   -> "DEwMGIwZ"
("job", "frontend")               -> "DEwMGIwZ"
("env", "dev")                    -> "DEwMGIwZ"
("line", "POST")                  -> "DEwMGIwZ"
("line", "/api/prom/push")        -> "DEwMGIwZ"
("line", "HTTP/1.1")              -> "DEwMGIwZ"
("line", "502")                   -> "DEwMGIwZ"
```



Existing log aggregation systems do full text indexing and support complex queries

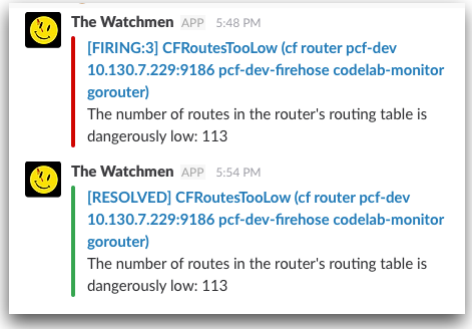
```
{job="frontend", env="dev"} => {
  time:  "2018-01-31 15:41:04",
  line:  "POST /api/prom/push HTTP/1.1 502 0"
}
```

Loki doesn't index the text of the logs, instead grouping entries into "streams" and indexing those with labels.

#1 Integrated with existing tools

Grafana Labs

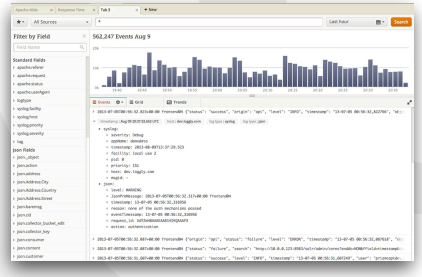
1. Alert



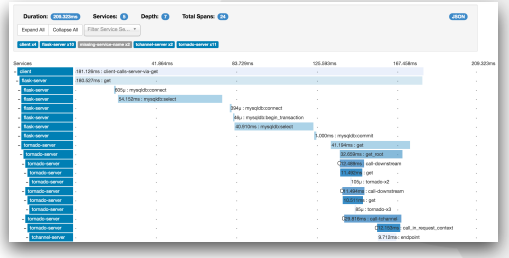
2. Dashboard

3. Adhoc Query

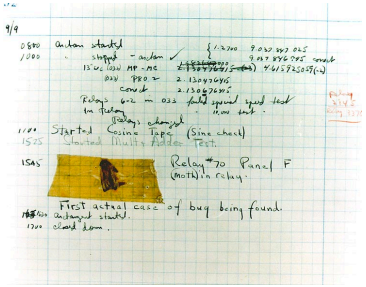
4. Log Aggregation



5. Distributed Tracing



Fix!



Prometheus' data model is very simple:

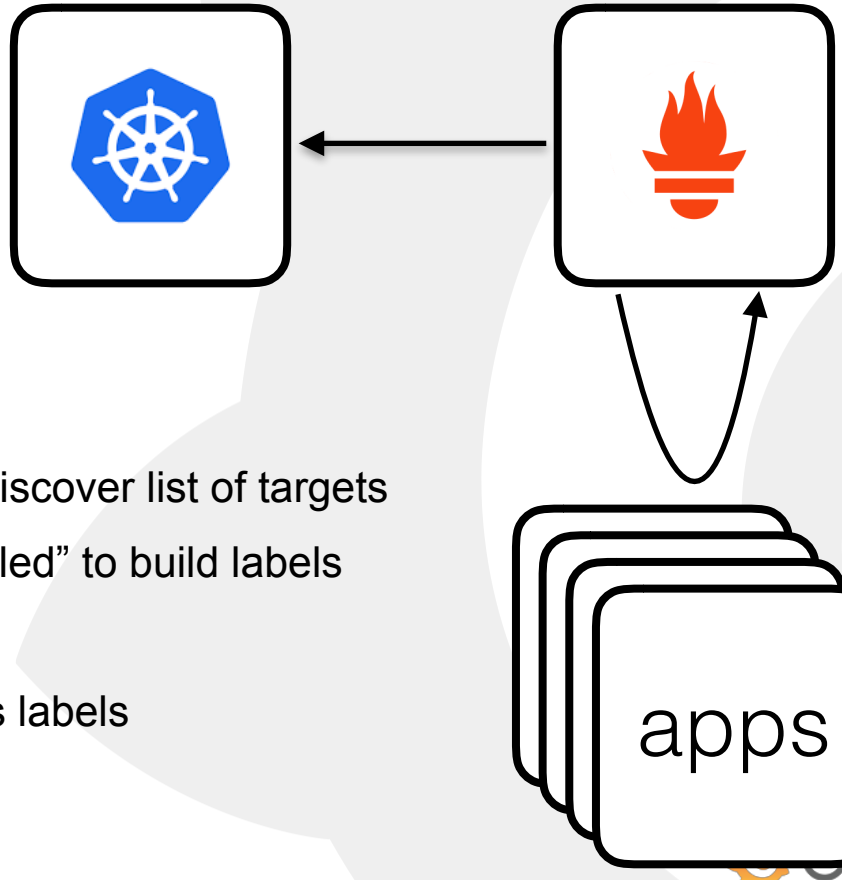
```
<identifier> → [ (t0, v0), (t1, v1), ... ]
```

Timestamps are millisecond int64, values are float64

Identifiers are bags of (label, value) pairs:

```
{job="foo", instance="bar", ... }
```

https://www.slideshare.net/Docker/monitoring-the-prometheus-way-julius-voltz-prometheus



#0 Prometheus talks to k8s to discover list of targets

#1 Target information is “relabelled” to build labels

#2 Metrics are pulled from apps

#3 Target labels added to series labels

What is Relabelling?

Grafana Labs

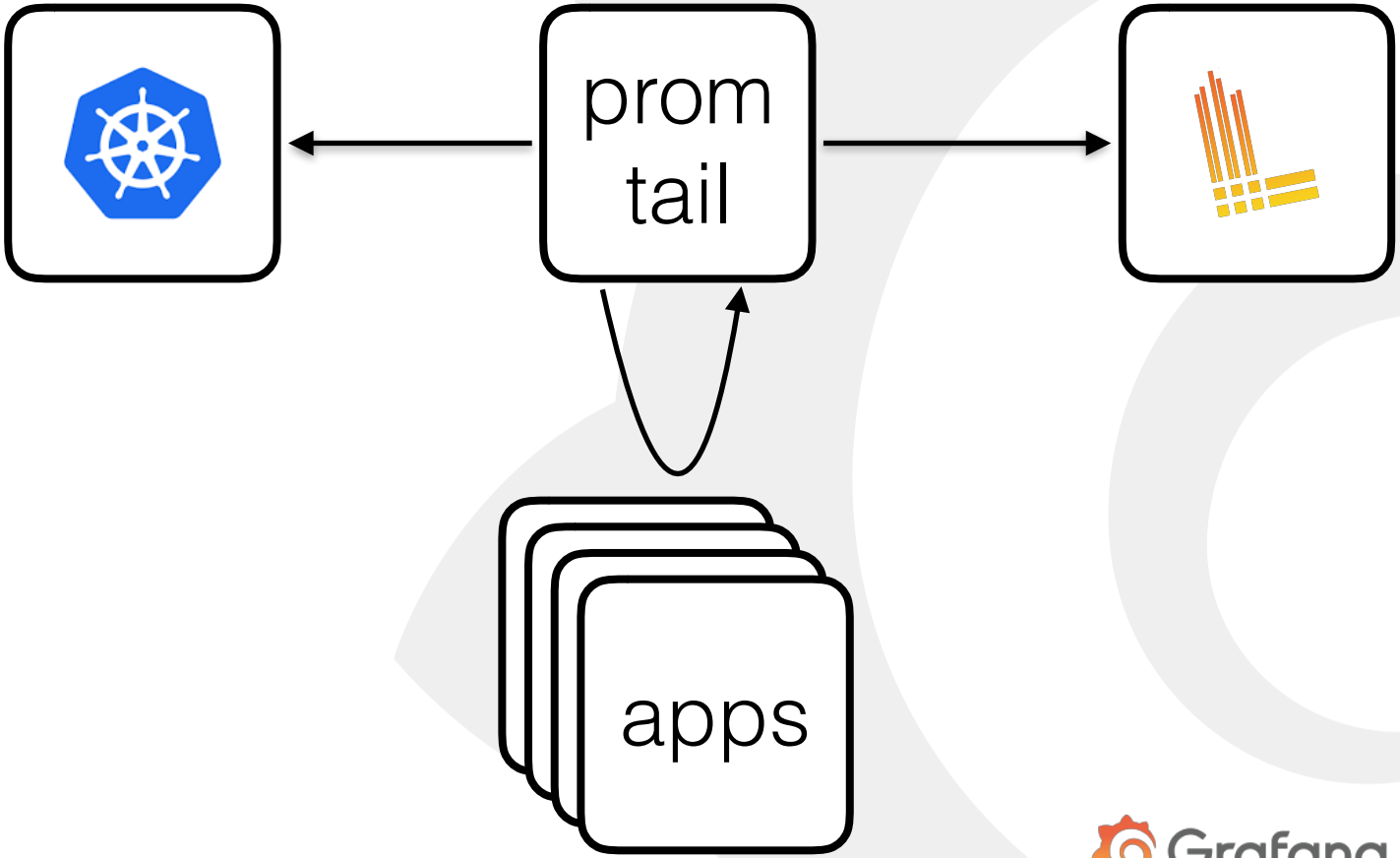
Loki’s data model is very similar:

```
<identifier> → [ (t0, v0), (t1, v1), ... ]
```

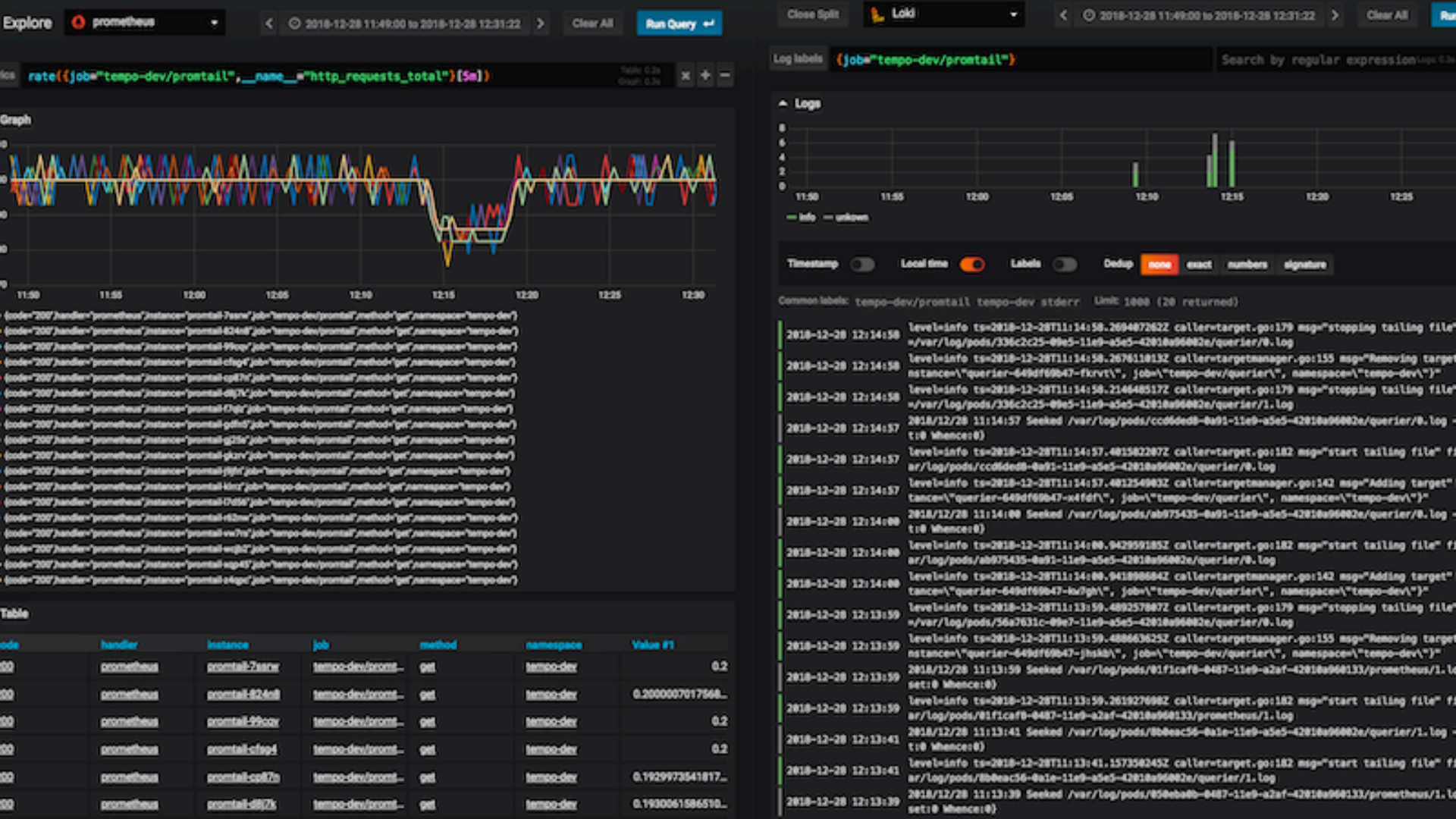
Timestamps are nanosecond floats, values are byte arrays.

Identifiers are the same - label sets.

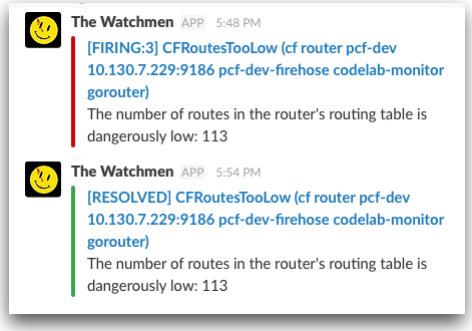
https://www.slideshare.net/Docker/monitoring-the-prometheus-way-julius-voltz-prometheus



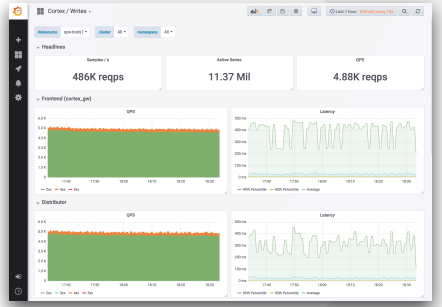
Grafana Labs


Explore
prometheus
2018-12-28 11:49:00 to 2018-12-28 12:31:22
Clear All
Run Query
rate({job="tempo-dev/promtail",__name__="http_requests_total"}[5m])
Graph
11:50
11:55
12:00
12:05
12:10
12:15
12:20
12:25
12:30
Table
code
handler
instance
job
method
namespace
Value #1
200 prometheus promtail-7sxrw tempo-dev/proms... get tempo-dev 0.2
200 prometheus promtail-824n8 tempo-dev/proms... get tempo-dev 0.2000007017568...
200 prometheus promtail-99ccv tempo-dev/proms... get tempo-dev 0.2
200 prometheus promtail-cfsg4 tempo-dev/proms... get tempo-dev 0.2
200 prometheus promtail-cp87n tempo-dev/proms... get tempo-dev 0.1929973541817...
200 prometheus promtail-d8j7k tempo-dev/proms... get tempo-dev 0.1930061586510...
Close Split
Loki
2018-12-28 11:49:00 to 2018-12-28 12:31:22
Clear All
Log labels
{job="tempo-dev/promtail"}
Search by regular expression
Logs
8
6
4
2
0
11:50
11:55
12:00
12:05
12:10
12:15
12:20
12:25
info
unknown
Timestamp
Local time
Labels
Dedup
none
exact
numbers
signature
Common labels: tempo-dev/promtail tempo-dev stderr
Limit: 1000 (20 returned)
2018-12-28 12:14:58 level=info ts=2018-12-28T11:14:58.269487262Z caller=target.go:179 msg="stopping tailing file" =/var/log/pods/336c2c25-00e5-11e9-a5e5-42010a96002e/querier/0.log
2018-12-28 12:14:58 level=info ts=2018-12-28T11:14:58.267611813Z caller=targetmanager.go:155 msg="Removing target nstance=\"querier-649df69b47-fkrvt\", job=\"tempo-dev/querier\", namespace=\"tempo-dev\"}"
2018-12-28 12:14:58 level=info ts=2018-12-28T11:14:58.214648517Z caller=target.go:179 msg="stopping tailing file" =/var/log/pods/336c2c25-00e5-11e9-a5e5-42010a96002e/querier/1.log
2018-12-28 12:14:57 2018/12/28 11:14:57 Seeked /var/log/pods/ccd6ded0-0a91-11e9-a5e5-42010a96002e/querier/0.log - t:0 Whence:0}
2018-12-28 12:14:57 level=info ts=2018-12-28T11:14:57.401582207Z caller=target.go:182 msg="start tailing file" f ar/log/pods/ccd6ded0-0a91-11e9-a5e5-42010a96002e/querier/0.log
2018-12-28 12:14:57 level=info ts=2018-12-28T11:14:57.401254983Z caller=targetmanager.go:142 msg="Adding target" tance=\"querier-649df69b47-x4fdf\", job=\"tempo-dev/querier\", namespace=\"tempo-dev\"}"
2018-12-28 12:14:00 2018/12/28 11:14:00 Seeked /var/log/pods/ab975435-0a91-11e9-a5e5-42010a96002e/querier/0.log - t:0 Whence:0}
2018-12-28 12:14:00 level=info ts=2018-12-28T11:14:00.942959185Z caller=target.go:182 msg="start tailing file" f ar/log/pods/ab975435-0a91-11e9-a5e5-42010a96002e/querier/0.log
2018-12-28 12:14:00 level=info ts=2018-12-28T11:14:00.941890684Z caller=targetmanager.go:142 msg="Adding target" tance=\"querier-649df69b47-kw7gh\", job=\"tempo-dev/querier\", namespace=\"tempo-dev\"}"
2018-12-28 12:13:59 level=info ts=2018-12-28T11:13:59.489257807Z caller=target.go:179 msg="stopping tailing file" =/var/log/pods/56a7631c-00e7-11e9-a5e5-42010a96002e/querier/0.log
2018-12-28 12:13:59 level=info ts=2018-12-28T11:13:59.488663625Z caller=targetmanager.go:155 msg="Removing target nstance=\"querier-649df69b47-jhskb\", job=\"tempo-dev/querier\", namespace=\"tempo-dev\"}"
2018-12-28 12:13:59 2018/12/28 11:13:59 Seeked /var/log/pods/01f1caf0-0487-11e9-a2af-42010a960133/prometheus/1.log set:0 Whence:0}
2018-12-28 12:13:59 level=info ts=2018-12-28T11:13:59.261927698Z caller=target.go:182 msg="start tailing file" f ar/log/pods/01f1caf0-0487-11e9-a2af-42010a960133/prometheus/1.log
2018-12-28 12:13:41 2018/12/28 11:13:41 Seeked /var/log/pods/8b0eac56-0a1e-11e9-a5e5-42010a96002e/querier/1.log - t:0 Whence:0}
2018-12-28 12:13:41 level=info ts=2018-12-28T11:13:41.157350245Z caller=target.go:182 msg="start tailing file" f ar/log/pods/8b0eac56-0a1e-11e9-a5e5-42010a96002e/querier/1.log
2018-12-28 12:13:39 2018/12/28 11:13:39 Seeked /var/log/pods/850eba0b-0487-11e9-a2af-42010a960133/prometheus/1.log set:0 Whence:0}

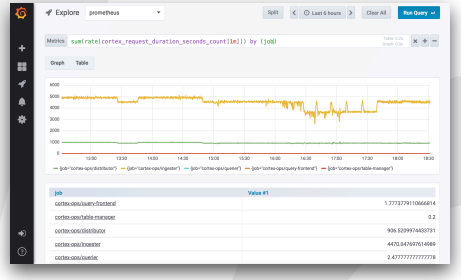
1. Alert



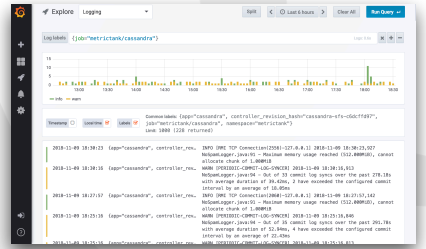
2. Dashboard



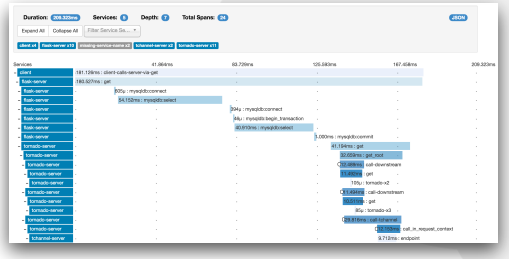
3. Adhoc Query



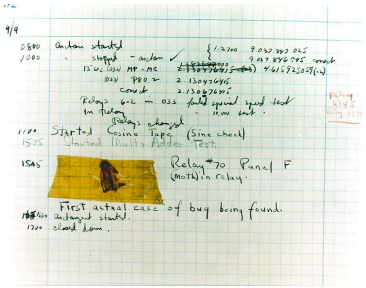
4. Log Aggregation



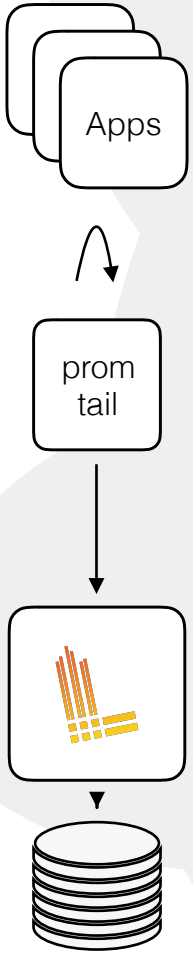
5. Distributed Tracing



Fix!

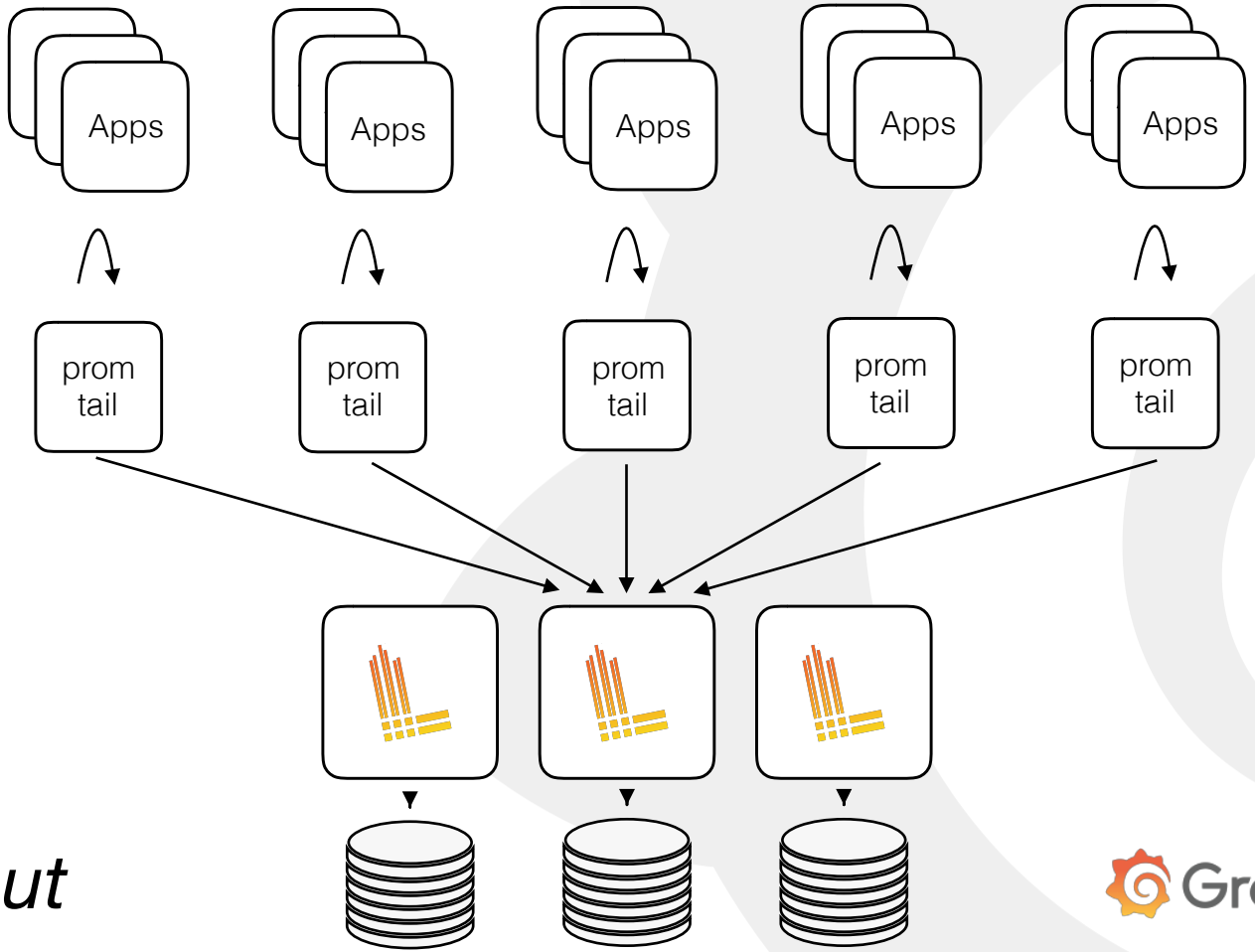


#2 Cloud Native and Airplane Friendly



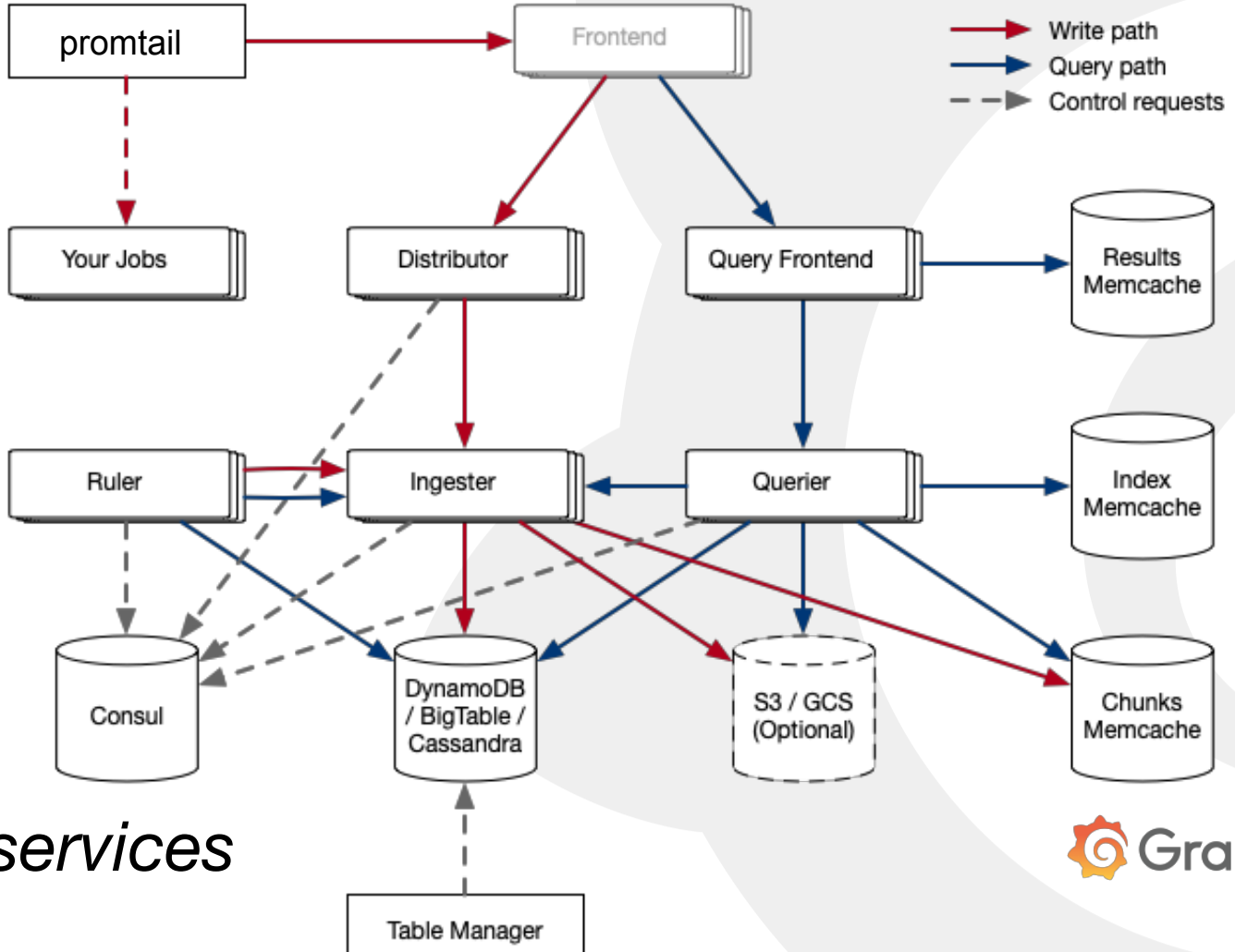
Airplane Friendly

Grafana Labs



Scale out

Grafana Labs

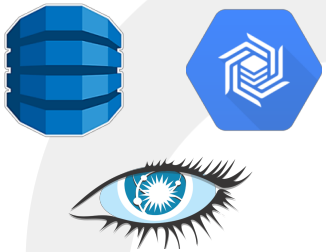


Microservices

Grafana Labs

Containerised

Kubernetes Native



Cloud Storage

#0 Simple and cost effective to operate

#1 Integrated with existing observability tools

#2 Cloud Native and Airplane Friendly

Demo

Grafana Labs

Whats next?

https://goo.gl/8h58zu

```
rate(({job=”app”} | “/foo” ! “/foo/bar”)[1m])

extract({job=”default/nginx”}, “code=(\d+)”, “co
    |> {code >= 500}

sum(extract({job=”app”}, “code=(\d+)”))
```

LogQL

Grafana Labs

Improve clustering & durability

Add Alerts & Rules off logs

Make it easier to get context, ad hoc filtering

Launch first beta in ~April

Grafana Labs

Thanks! Questions?

(we're hiring)