

# Grafana Loki: Like Prometheus, but for logs.

Tom Wilkie, Feb 2019



# Demo





**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



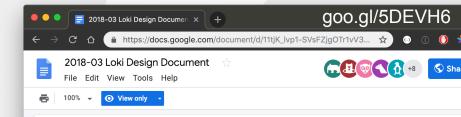




Loki is a horizontally-scalable, highly-available, multitenant log aggregation system inspired by Prometheus.

- 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



### Loki: like Prometheus, but for logs.

Design Document Tom Wilkie & David Kaltschmidt, March 2018

This document aims to explain the motivations for, and design of, the Grafana Loki service. This document does not attempt to describe in depth every possible detail of the design, but hopefully explains the key points and should allow us to spot any obvious mistakes ahead of time.

This document aims to answer question not only about how we're going to build this, but also why we're building it, what it will be used for, and who will be using it.

#### **Background & Motivations**

#0 Simple and cost effective to operate

#1 Integrated with existing observability tools

#2 Cloud Native and Airplane Friendly

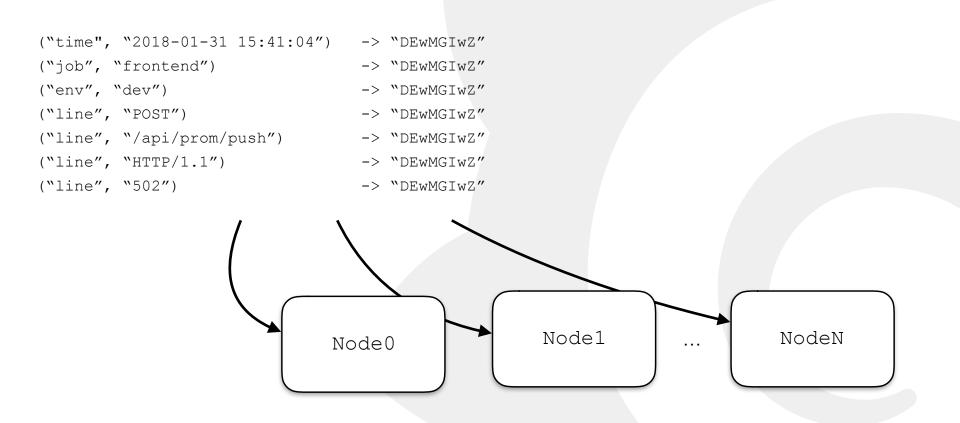
# #0 Simple to scale



```
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



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

### {job="frontend", env="dev"} => {

```
time: "2018-01-31 15:41:04",
```

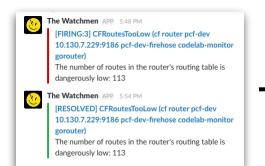
line: "POST /api/prom/push HTTP/1.1 502 0"

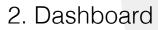
Loki doesn't index the text of the logs, instead grouping entries into "streams" and indexing those with labels.

# #1 Integrated with existing tools



## 1. Alert







## 3. Adhoc Query





Duration: (201222		ins: 🔁			180N
Expand All Collopse	M Filter Service Se *				
short of Taxabaran a	12 Insuing service came x2 Monade server x2 Monade serv	HT IT I I I I I I I I I I I I I I I I I			
Services	41.864ms	83.229ms	125.5Kims	167.458ms	209.5
- ciert	181.128ms : client-calls-server-via-get				
<ul> <li>Task-server</li> </ul>	-180.527ms : get .				
- fask-server	- B35µ : mysqldbconnect				
- fask-server	54.152ms : mysgido:select				
- fask-server	and the second	204 µ : mysqldbconnect			
<ul> <li>Task-server</li> </ul>	<ul> <li>A second sec second second sec</li></ul>	48u: mysejidix.begir_transaction			
<ul> <li>Task-server</li> </ul>	and the second se	40.610ne i mysqidoselect			
<ul> <li>Task-server</li> </ul>	and the second		1.000ms : mysqldbroemmit		
<ul> <li>tomado-server</li> </ul>	and the second		41.194ms : get	1 A A A A A A A A A A A A A A A A A A A	
<ul> <li>tomado-server</li> </ul>	and the second		32.659ms : get_root		
<ul> <li>tornado-server</li> </ul>	and the second		- Otz.deans col-dev	reboare	
<ul> <li>tomade-server</li> </ul>	and the second		11.402mm ( get		
<ul> <li>tomade-server</li> </ul>	and the second		105µ : formade		
<ul> <li>tomado-server</li> </ul>	a second s		· Ottelene : cal-do		
<ul> <li>tomado-server</li> </ul>	Provide the second s		10.51 true : get		
<ul> <li>tomado-server</li> </ul>			BSp : tomado		
<ul> <li>tomade-server</li> </ul>	and the second		· Q29.816ms : call to		
<ul> <li>tomade-server</li> </ul>	and the second			163ms cal_in_request_center	
<ul> <li>Scharchel-server</li> </ul>				time: endpoint	

5. Distributed Tracing



4. Log Aggregation





Prometheus' data model is very simple:

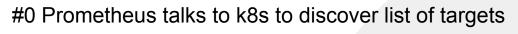
<identifier>  $\rightarrow$  [ (t0, v0), (t1, v1), ... ]

Timestamps are millisecond int64, values are float64

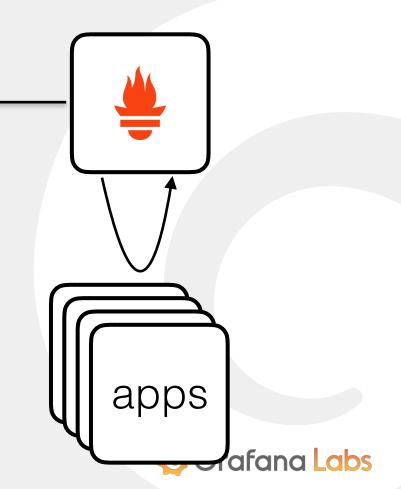
Identifiers are bags of (label, value) pairs:

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

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



- #1 Target information is "relabelled" to build labels
- #2 Metrics are pulled from apps
- #3 Target labels added to series labels



# What is Relabelling?





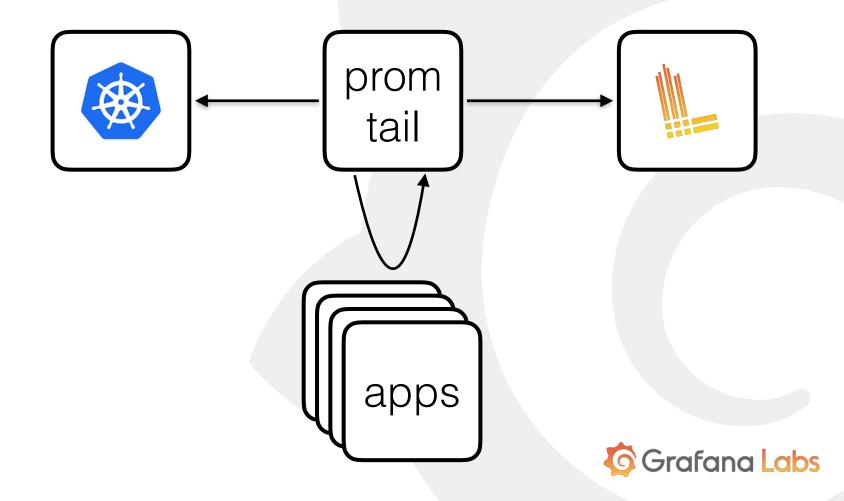
Loki's data model is very similar:

```
<identifier> \rightarrow [ (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



#### Explore

< O 2018-12-28 11:49:00 to 2018-12-28 12:31:22 >

Clear All Run Query +\*

× + -

rate((job="tempo-dev/prontail",\_\_name\_\_="http\_requests\_total")(Sm))



(code="2017 handler="prometheus" instance="promial 7sars" job="tempo de=!promial" method="get" namespace="tempo des") (code="200" handler="prometheus" instance="promtail #24n8" job="hampo dev/promtail" method="get" namespace="hampo dev") (code="2017 handler="prometheus" instance="promial 99cqv" job="tempo devipromial" imethod="get" namespace="tempo dev") (code='200')andler='prometheus'/instance='promtail.cfsg4'job='tempo.des/promtail',method='get',namespace='tempo.des'} (code="200" handler="prometheus" instance="promtal-cp87rf job="hempo-dev/promtal",method="gef",namespace="hempo-dev") (code="200"/sendle="prometheus"/setance="promtail-dbj?k".job="tempo-des/promtail",method="get",namespace="tempo-des") loode-"200" handler-"prometheus" instance-"promial (Fight job-"hengo dev/promial" method="get" namespace-"hengo dev") (code='200') handler='prometheus' instance='promitail gdfn5' job='hampo dev/promitail',method='get',namespace='hempo dev') (code="200" handler="prometheus" instance="promitall gi25s" job="tampo dev/promital",method="get",xamespace="tampo dev") (code="200")ander="prometheus"/instance="promital gkzn"/job="tempo des/promital",method="get",namespace="tempo des") (code="2017 handler="prometheus" instance="promtall jtiple" job="hempo dev/promtall" method="get" namespace="hempo dev") (code="200"/handler="prometheus"/initance="promial kins" job="hempo dev/promial".method="get".namespace="hempo dev") (oode-"2017 handler-"prometheus" (instance-"promtali (7d56" job-"henpo-dev(promtali",method-"get",namespace-"henpo-dev") (code="2017 handler="prometheus" instance="promial rli2ns" job="hempo dev/promial",method="get",nemespace="hempo dev") (zode="2017 handler="prometheus" instance="promitali v=7rs" job="tempo de=/promitali",method="pet",nemespace="tempo de=") (code="200"/sandier="prometheus"/instance="promiall-wc(b)".job="tempo-des/promiall".method="pet".nemespace="tempo-des") (code="200" handler="prometheus" instance="promtail upp45" job="tempo dev/promtail",method="get",namespace="tempo dev") (code="200"/handle="prometheus"/instance="promisil arkgo"/job="hempo dev/promisil"/method="get"/namespace="hempo dev/)

#### Table

ode	handler	Instance	jab .	method	namespace	Value #1
22	prometheus	promtell-7serve	tempo-dev/promt_	98	tempo-dev	0.2
22	prometheus	promtail-824n8	tempo-dev/promt_	96	tempo-der	0.2000007017568
92	prometheus	promtall-99cov	tempo-dev/promt_	96	tempo-der	02
<u>80</u>	prometheus	econtal-chip4	tempo-dev/promt_	98	tempo-dev	0.2
<b>\$</b> 2	prometheus	promtelico#2n	tempo-dev/promt_	96	temposter	0.1929973541817_
92	prometheus	econtail-d8/2k	tempo-dev/promt_	96	tempo-dex	0.1930061586510



Common labels: tempo-dev/promtail tempo-dev stderr Limit 1000 (20 returned)

level-info ts=2018-12-20111:14:58.2694072622 coller-target.go:179 msg="stopping tailing file"
<pre>"/var/log/pods/336c2c25-00e5-11e0-a5e5-42018u06002e/querier/0.log level=info ts=2018-12-20T11:14:50.267611013Z caller=targetmanager.go:155 msg="Removing target nstance=\"querier=640df00b47-fkrvt\", job=\"tempo-dev/querier\", namespace=\"tempo-dev\")"</pre>
<pre>level=info ts=2018-12-20111:14:50.2146405172 caller=target.go:179 msg="stopping tailing file" =/var/log/pods/336c2c25-09e5-11e9-a5e5-42018466002e/querier/1.log</pre>
2018/12/28 11:14:57 Seeked /var/log/pods/ccd6ded8-0a91-11e9-a5e5-42010a96002e/querier/0.log - t:0 Whence:0)
<pre>bevelwinfo ts=2018-12-2011114157.4015022072 caller=target.go:182 msg="start tailing file" fi ar/log/pods/ccd6ded=4x91=11x8=a5e5=42018x86042a/querier/0.log</pre>
<pre>bevel=info ts=2018-12-2011114157.4012540002 caller=targetmanager.goil42 msg="Adding target" tance=\"querier=640df60647-x4fdf\", job=\"tempo-dev/querier\", namespace=\"tempo-dev\"}"</pre>
2018/12/20 11:14:00 Seeked /var/log/pods/ab975435-0x91-11e9-a5e5-42010x96002e/querier/0.log - tr0 Mhencer0)
<pre>levelwinfo ts=2018-12-2011:14:00.942959185Z caller=target.go:182 msg="start tailing file" fi ar/log/pods/a0975435-0091-11e9-a5e5-42010e95002e/querier/0.log</pre>
<pre>bevelwinfo ts=2018-12-2011114:00.941890684Z caller=targetmanager.go142 msg="Adding target" tance=\"guerier=649df69047=ku7gh\", job=\"tempo-dev/querier\", mamespace=\"tempo-dev\")"</pre>
<pre>level=info ts=2018-12-2011113:59.4092570072 caller=target.go:179 msg="stopping tailing file" =/var/log/pods/5627631c-09e7-11e9-a5e5-42010096002e/querier/0.log</pre>
<pre>level=info ts=2018-12-2011113:59.4886636252 caller=targetmanager.go:155 msg="Removing target nstance=\"querier=648df69947=jhskb\", job=\"tempo=dev/querier\", namespace=\"tempo=dev\")"</pre>
2018/12/20 11:13:59 Seeked /var/log/pods/01f1caf0-0407-11e9-a2af-420100500133/prometheus/1.1c set:0 Mience:0)
<pre>level=info ts=2018-12-20111:13:59.2019270002 caller=target.go:182 msg="start tailing file" fi ar/log/pods/01ficaf0-0407-11e0-a2af-420100500133/prometheus/1.log</pre>
2018/12/20 11:13:41 Seeked /var/log/pods/8b0eac56-0ale-11e9-a5e5-42010a96002e/querier/1.log - t:0 Mmence:0)
<pre>level=info ts=2018-12-2011:13:41.1573502452 caller=target.go:182 msg="start tailing file" fi ar/log/pods/8b0ex56-0ale=11e9=a5e5-42010a06002e/querier/1.log</pre>
2018/12/20 1113/30 Seeked /var/log/pods/050eba0b-0407-11e0-x2af-42010a060133/prometheus/1.14 set:0 Whence:0)

## 1. Alert

Fix!

## 2. Dashboard

## 3. Adhoc Query



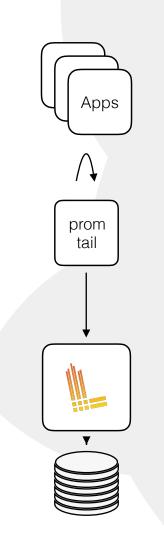
5. Distributed Tracing

4. Log Aggregation

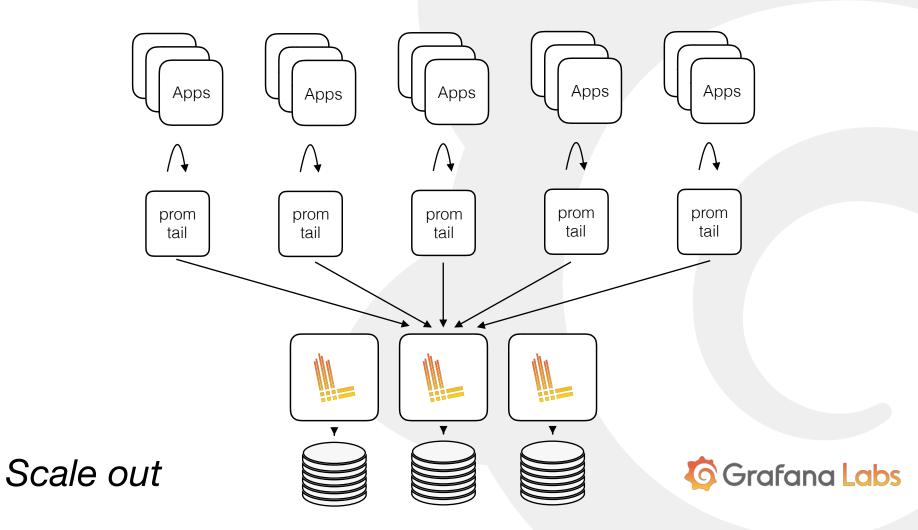
# #2 Cloud Native and Airplane Friendly

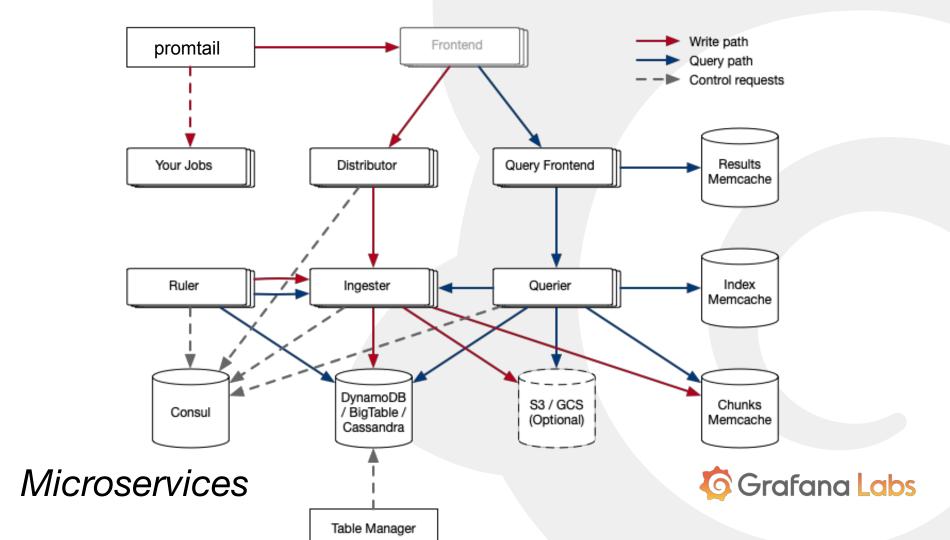














Containerised



## Kubernetes Native



# **Cloud Storage**

#0 Simple and cost effective to operate

#1 Integrated with existing observability tools

#2 Cloud Native and Airplane Friendly

# Demo



# Whats next?



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

extract({job="default/nginx"}, "code=(\d+)", "code=(\d+)", "code=

sum(extract({job="app"}, "code=(\d+)"))





Improve clustering & durability

Add Alerts & Rules off logs

Make it easier to get context, ad hoc filtering

Launch first beta in ~April



# Thanks! Questions?

(we're hiring)

