

ENDPOINT VISIBILITY WITH OSQUERY AND LOKI

WHO WE ARE

GEORGE ADAMS IV & ED WELCH

USE CASE

DETECTING SSH CONNECTIONS AND ALERTING.

FOLLOW ALONG

[HTTPS://GITHUB.COM/GEOWA4/LEARN-LOKI](https://github.com/GEOWA4/LEARN-LOKI)

OSQUERY

RELEASED BY FACEBOOK IN 2014

```
commit 73a32b729403b2f5a7c204b0f7cfb86fdfdd0a85
```

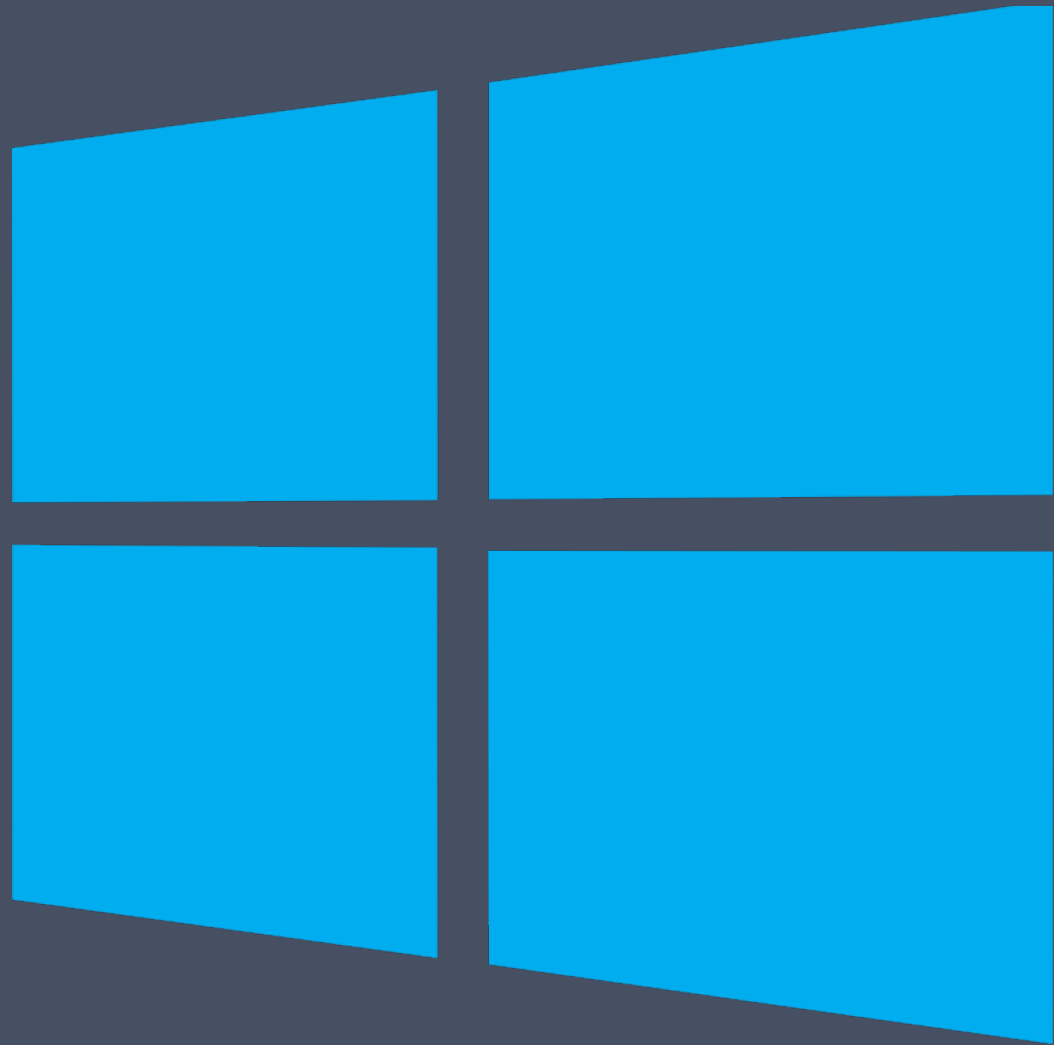
```
Author: mike@arpaia.co <mike@arpaia.co>
```

```
Date:    Wed Jul 30 17:35:19 2014 -0700
```

Initial commit

OSQUERY

WORKS ON MY MACHINE



OSQUERY

SQL INTERFACE TO YOUR ENDPOINTS

.schema processes

```
CREATE TABLE processes(  
  `pid` BIGINT, `name` TEXT, `path` TEXT, `cmdline` TEXT,  
  `disk_bytes_read` BIGINT, `disk_bytes_written` BIGINT,  
  ...  
  PRIMARY KEY (`pid`)  
) WITHOUT ROWID;
```

OSQUERY

```
select max(disk_bytes_read), pid, name, cmdline, cwd from processes;
```

max(disk_bytes_read)	pid	name	cmdline	cwd
10465280	5478	bash	-bash	/vagrant/demo

OSQUERY

```
select p.name, l.port, l.protocol
from processes p inner join listening_ports l on p.pid = l.pid
where p.name = 'VBoxHeadless' and l.port <> 0;
```

name	port	protocol
VBoxHeadless	3000	6
VBoxHeadless	3100	6
VBoxHeadless	9080	6
VBoxHeadless	9090	6
VBoxHeadless	2222	6

OSQUERY

```
select c.name, p.port, p.host_port
from docker_containers c inner join docker_container_ports p
on c.id = p.id;
```

name	port	host_port
/demo_loki_1	80	0
/demo_loki_1	3100	3100
/demo_grafana_1	3000	3000
/demo_prometheus_1	9090	9090
/demo_promtail_1	9080	9080

OSQUERY – LAST SCHEMA

.schema last

```
CREATE TABLE last(  
  `username` TEXT, `time` INTEGER, `host` TEXT,  
  `pid` INTEGER, `tty` TEXT, `type` INTEGER  
);
```

OSQUERY – LAST QUERY

```
select * from last;
```

username	tty	pid	type	time	host
reboot	~	0	2	1566866107	4.15.0-52-generic
runlevel	~	53	1	1566866118	4.15.0-52-generic
	ttyS0	859	5	1566866119	
LOGIN	ttyS0	859	6	1566866119	
	tty1	879	5	1566866119	
LOGIN	tty1	879	6	1566866119	
vagrant	pts/0	5396	7	1566869034	10.0.2.2
vagrant	pts/1	6465	7	1566870878	10.0.2.2
	pts/1	6465	8	1566870880	
vagrant	pts/1	6571	7	1566870886	10.0.2.2
	pts/1	6571	8	1566870910	

OSQUERY – PACKS

```
{  
  "queries": {  
    "last": {  
      "query": "select * from last;",  
      "interval": "60",  
      "platform": "posix",  
      "version": "1.4.5",  
      "description": "..."  
    }  
  }  
}
```

OSQUERY – DECORATORS

```
{  
  "decorators": {  
    "load": [  
      "SELECT uuid AS host_uuid FROM system_info;",  
      "SELECT user AS username FROM logged_in_users ORDER BY time DESC LIMIT 1;"  
    ]  
  }  
}
```

OSQUERY – RESULTS

/var/log/osquery/osqueryd.results.log

```
{
  "name": "pack_incident-response_last",
  "hostIdentifier": "ubuntu-bionic",
  "calendarTime": "Tue Aug 27 01:55:13 2019 UTC",
  "decorations": {
    "host_uuid": "2401CCE9-23EA-4D4D-8C84-D5C8437EBE15",
    "username": "vagrant"
  },
  "columns": {
    "host": "10.0.2.2",
    "pid": "6465",
    "time": "1566870878",
    "tty": "pts/1",
    "type": "7",
    "username": "vagrant"
  },
  "action": "added"
}
```

OSQUERY – RECAP

IT'S BEEN AROUND A WHILE

CROSS-PLATFORM

IT'S JUST SQL

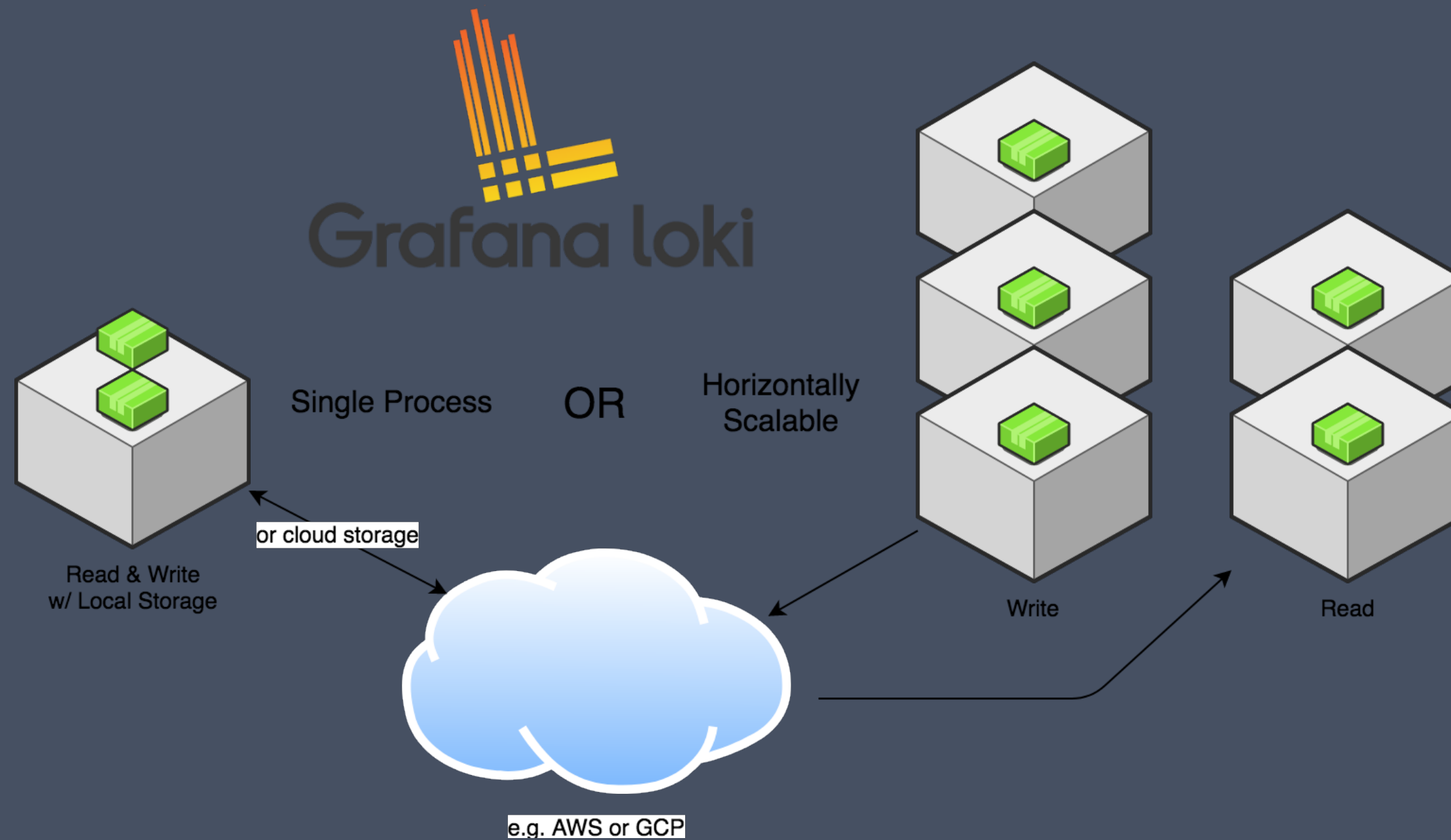
SCHEDULE QUERIES WITH 'PACKS'

LOKI

PROMETHEUS-INSPIRED LOGGING FOR CLOUD NATIVES.

MADE BY GRAFANA

LOKI



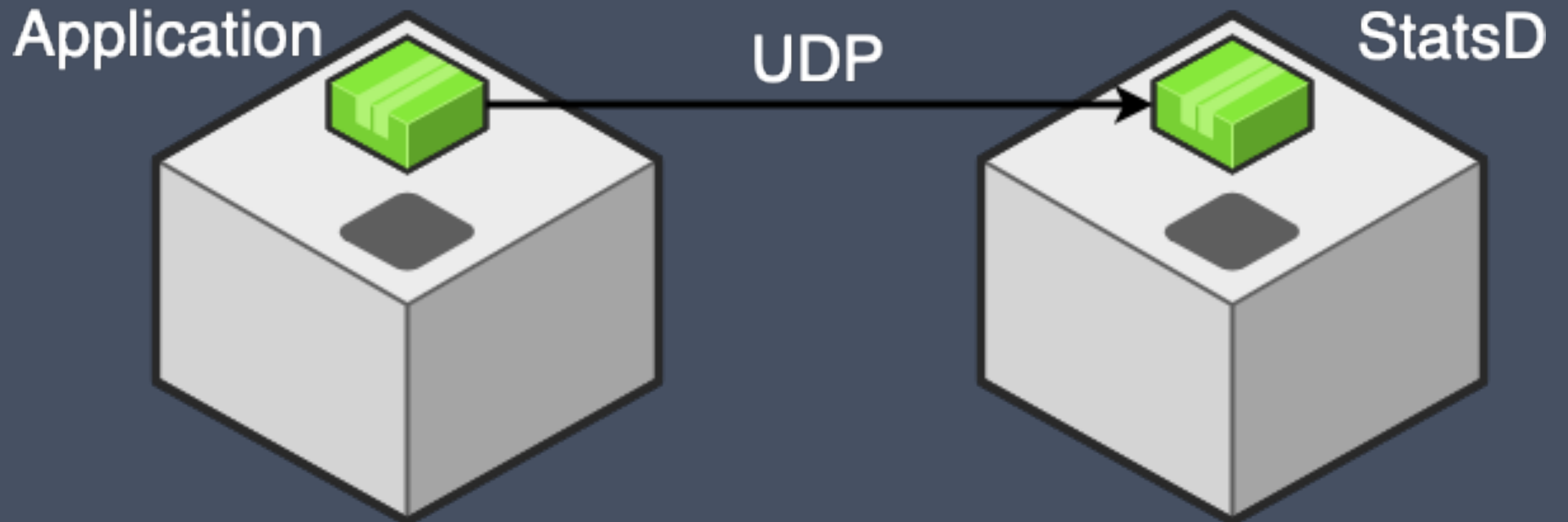
PROMETHEUS

FIRST A LITTLE BACKGROUND ON PROMETHEUS:

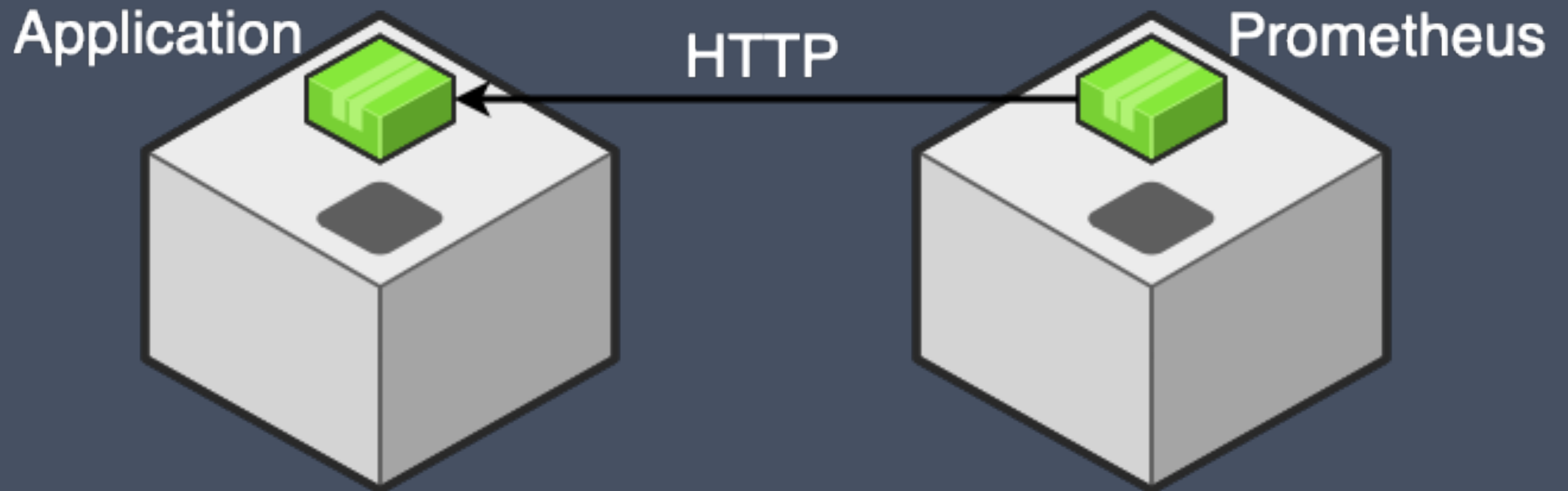
A TIME-SERIES METRIC COLLECTION, STORAGE AND QUERYING
APPLICATION.

PROMETHEUS – PUSH

STATSD



PROMETHEUS - PULL



PROMETHEUS – TIME-SERIES

QUERYABLE VIA PROMQL

```
sum by (instance) rate(http_requests_total{cluster="us-central1",app="loki"}[5m])  
increase(promtail_custom_last_logins{name="pack_incident-response_last"}[5m])
```

PROMETHEUS – SCRAPING

```
scrape_configs:  
  - job_name: "promtail"  
    static_configs:  
      - targets:  
        - promtail:9080
```

PROMETHEUS – DATA STRUCTURE

METRICS HAVE LABELS IN ADDITION TO VALUES.

```
rss_attendance_total{track="tech", talk="osquery_loki"} 30  
sum(rss_attendance_total{track="tech"}) 100
```


LOKI – DATA STRUCTURE

LOG ENTRIES HAVE LABELS, TOO.

```
{track="tech", talk="osquery_loki"} "best talk ever"  
{track="tech", talk="osquery_loki"} "i want to know more"  
{track="tech", talk="osquery_loki"} "i hear that one guy runs rocdev"
```

LOKI – STORAGE



```
{component="printer",location="f2c16",level="error"} "Printing is not supported by this printer"
```

Label key/values hashed to form **Stream ID**: 3b2cea09797978fc

The log entry is added to a "chunk"

Additional log messages with the same labels are added to the same "chunk":

```
{component="printer",location="f2c16",level="error"} "Out of paper"  
{component="printer",location="f2c16",level="error"} "Too much paper"
```

Chunks are filled then compressed and stored:

```
Printing is not supported by this printer  
Out of paper  
Too much paper
```



```
Printing is not supported by this printer  
Out of paper  
Too much paper
```

A separate and small index is kept to lookup chunks

Different label keys or values will hash to a different stream and different chunk:

```
{component="printer",location="f2c16",level="info"} "Consider the environment before printing this log message"
```

fd9a709ddf43a93a

LOKI – QUERY

LOGQL

LABEL MATCHING (REDUCES CHUNKS LOADED FOR QUERIES):

- `{name=~"mysql.+"}`
- `{name!~"mysql.+"}`

LOKI - FILTERING

grep **AND** grep -v

- {job="mysql"} |= "error"
- {name="kafka"} |~ "tsdb-ops.*io:2003"
- {instance=~"kafka-[23]",name="kafka"} != kafka.server:type=ReplicaManager
- {job="mysql"} |= "error" != "timeout"

LOKI – AGGREGATIONS

PROMQL STYLE AGGREGATIONS ($w_c - 1$) OR MORE
COMPLICATED THINGS LIKE RATE

- `count_over_time({job="mysql"}[5m])`
- `rate(({job="mysql"} |= "error" !=
"timeout")[10s]))`

LOKI – ONE MORE NOTE ON LABELS

CONSISTENT LABELING BETWEEN METRICS IN PROMETHEUS AND LOGS IN LOKI ALLOW SWITCHING BACK AND FORTH BETWEEN METRICS AND LOGS FREELY.

LOKI – COLLECTION

PROMTAIL

PROMTAIL

FORWARDS LOGS
AND
EXTRACTS METRICS

PROMTAIL – SCRAPING

```
clients:
```

- url: http://loki:3100/api/prom/push

```
scrape_configs:
```

- job_name: osquery

```
  static_configs:
```

- targets:
 - localhost

```
  labels:
```

```
    job: osquery_results
```

```
    __path__: /var/log/osquery/osqueryd.results.log
```

PROMTAIL – RESULT REMINDER

```
{
  "name": "pack_incident-response_last",
  "hostIdentifier": "ubuntu-bionic",
  "calendarTime": "Thu Aug 29 03:01:37 2019 UTC",
  "unixTime": 1567047697,
  "epoch": 0,
  "counter": 115,
  "decorations": {
    "host_uuid": "661449FD-E11A-462B-9EA9-63A3EE8F9BDC",
    "username": "vagrant"
  },
  "columns": {
    "host": "10.0.2.2",
    "username": "vagrant",
    "type": "7",
    "time": "1567047680",
    "tty": "pts/1",
    "pid": "7404"
  },
  "action": "added"
}
```

PROMTAIL – PIPELINES

```
pipeline_stages:  
- json:  
    expressions:  
        timestamp: unixTime  
        name: name  
- timestamp:  
    source: timestamp  
    format: Unix  
- labels:  
    name: name
```

PROMTAIL - METRICS

```
pipeline_stages:  
  - ...  
  - metrics:  
      last_logins:  
        type: Counter  
        description: count last logins  
        source: name  
        config:  
          value: pack_incident-response_last  
          action: inc
```

PROMTAIL – PROMETHEUS

```
scrape_configs:  
  - job_name: "promtail"  
    static_configs:  
      - targets:  
        - promtail:9080
```

WHERE WE ARE NOW

OSQUERY PRODUCING RESULTS

PROMTAIL FORWARDING TO LOKI

QUERY AND TAIL LOGS IN LOKI

PROMTAIL EXTRACTING METRICS

PROMETHEUS SCRAPING PROMTAIL

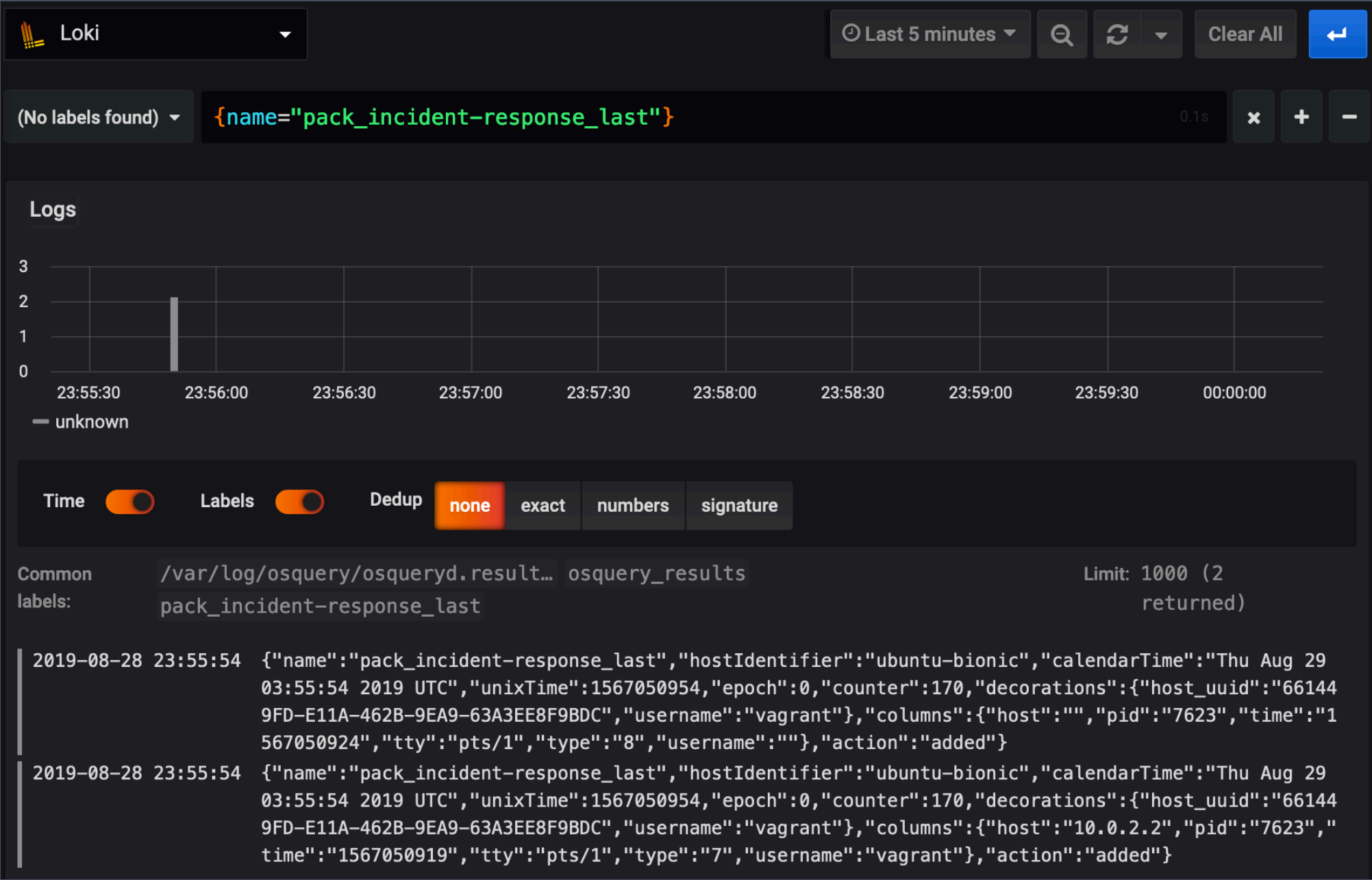
WHAT'S LEFT

CHARTING & ALERTING

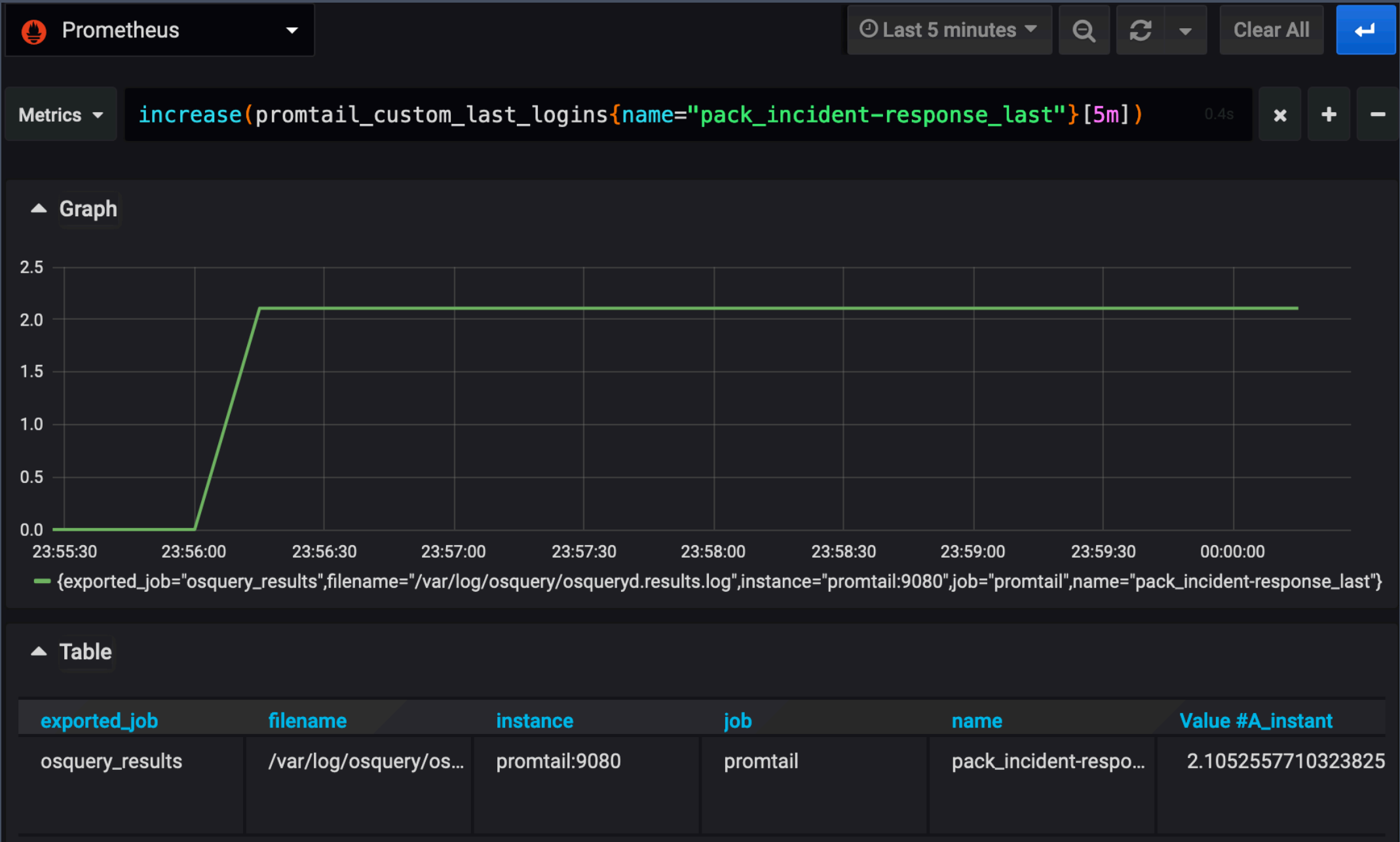
GRAFANA

SUPPORTS BOTH PROMTHEUS AND LOKI AS DATA SOURCES

GRAFANA - LOKI

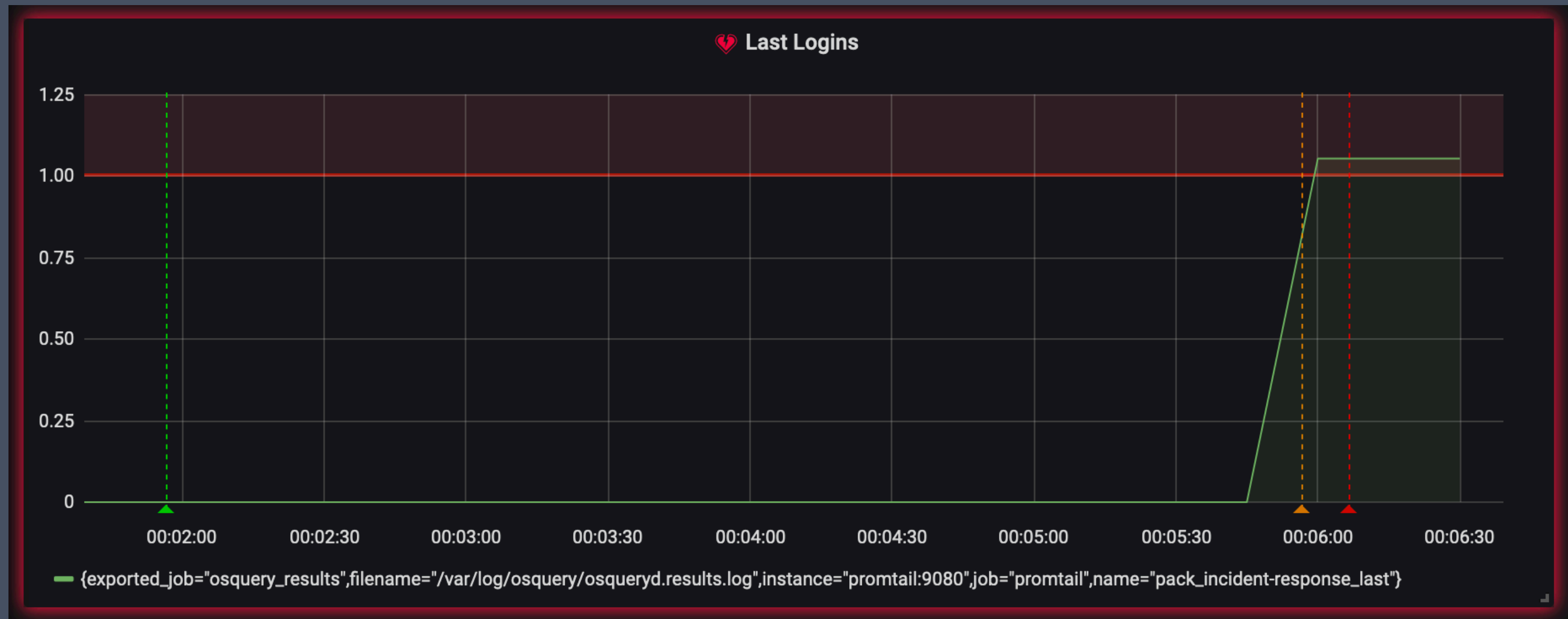


GRAFANA – PROMETHEUS

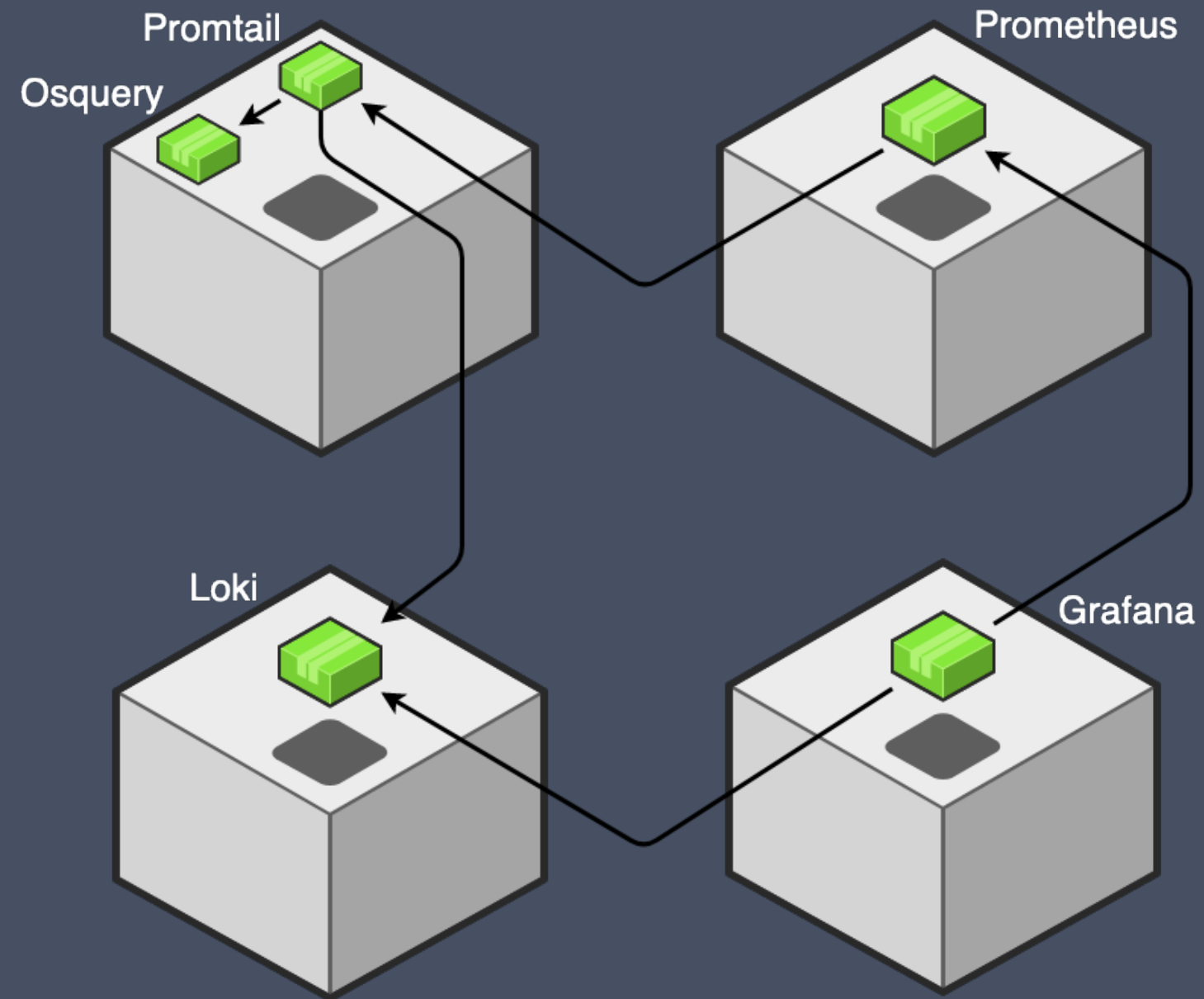


GRAFANA - ALERTING

```
increase(promtail_custom_last_logins{name="pack_incident-response_last"}[5m])
```



PUTTING IT ALL TOGETHER



LINKS & THINGS

- > OSQUERY
- > LOKI
- > PROMETHEUS