Adopting Prometheus the Hard Way

Tim Simmons - Engineer - DigitalOcean
@timsimlol

Who am I

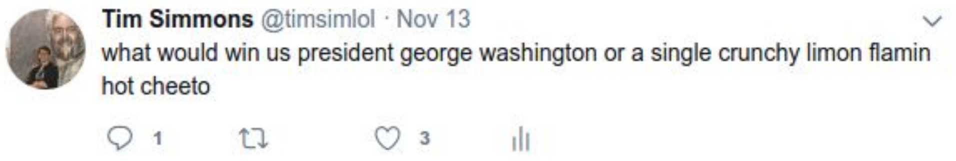
Tim Simmons

Engineer/Prometheus person

Observability Platforms

DigitalOcean

@timsimlol



Tim Simmons @timsimlol · Nov 13

what would win us president george washington or a single crunchy limon flamin hot cheeto

#goals

Understand what Prometheus **is**

Understand the **value** of a healthy Observability culture

How Prometheus (and friends) can **help**

Learn the **nuances** of scaling Prometheus and its adoption

Make Prometheus **successful** for you

Prometheus

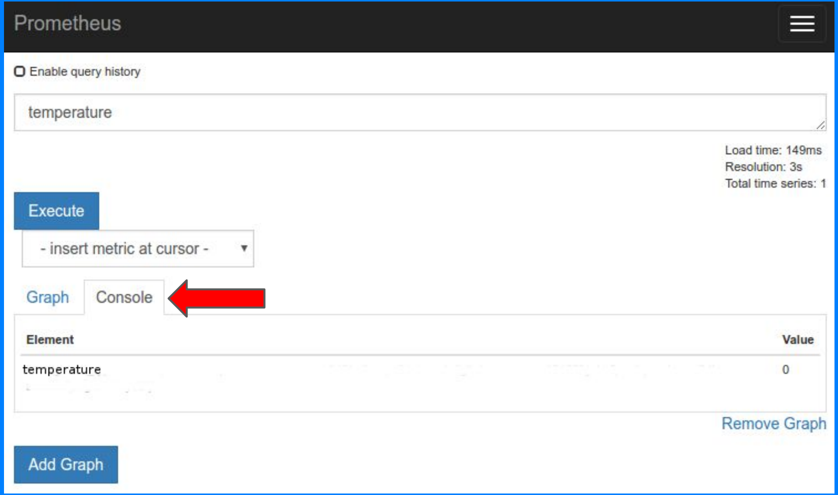
timeseries data (metrics)

timeseries = {(t0, v0), (t1, v1) .... }
temperature = {(00:00, 20), (00:30, 22), ... (12:00, 25)}

queries (PromQL)

temperature = 25
temperature[1h] = [(11:30, 23), (12:00, 25)]
avg_over_time(temperature[12h]) = 20.94734

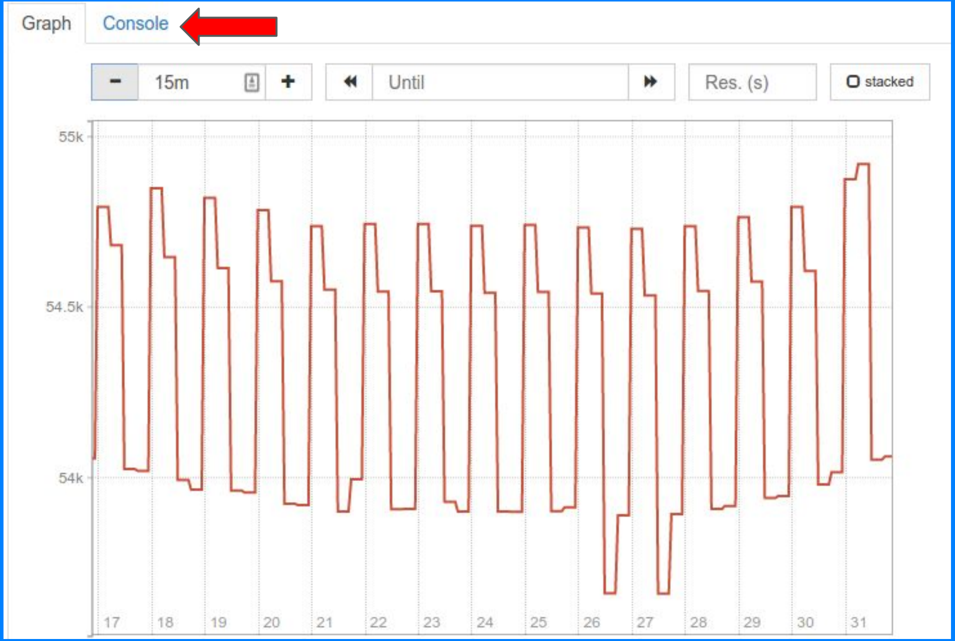
Prometheus - Querying



temperature

Load time: 149ms
Resolution: 3s
Total time series: 1

Element	Value
temperature	0



instant queries (console, /query)
latest timeseries sample

range queries (graph, /query_range)
evaluate query at many times over the timescale of the query (useful for making graphs)

digitalocean.com

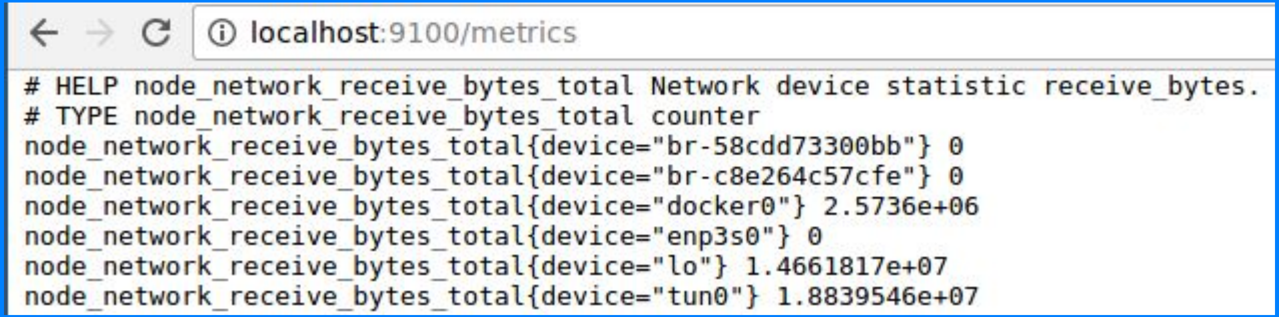
Prometheus - Labels

query	results
temperature	temperature{city="seattle", state="WA"} = 11 temperature{city="tacoma", state="WA"} = 10 temperature{city="san francisco", state="CA"} = 16 temperature{city="san jose", state="CA"} = 12 temperature{city="austin", state="TX"} = 15
temperature{state="CA"}	temperature{city="san francisco", state="CA"} = 16 temperature{city="san jose", state="CA"} = 12
avg(temperature) by (state)	temperature{state="WA"} = 10.5 temperature{state="CA"} = 14 temperature{state="TX"} = 15

Prometheus - Metrics

Pull based

Applications expose metrics HTTP endpoint



```
# HELP node_network_receive_bytes_total Network device statistic receive_bytes.
# TYPE node_network_receive_bytes_total counter
node_network_receive_bytes_total{device="br-58cdd73300bb"} 0
node_network_receive_bytes_total{device="br-c8e264c57cfe"} 0
node_network_receive_bytes_total{device="docker0"} 2.5736e+06
node_network_receive_bytes_total{device="enp3s0"} 0
node_network_receive_bytes_total{device="lo"} 1.4661817e+07
node_network_receive_bytes_total{device="tun0"} 1.8839546e+07
```

Prometheus - Exporters

JMX
Consul
ElasticSearch
Memcached
MongoDB
MSSQL
MySQL
PostgreSQL
ProxySQL
Redis
node/system metrics
NVIDIA
Ubiquiti
Kubernetes

Kafka
MQTT
RabbitMQ
Ceph
Gluster
Hadoop
Apache
HAProxy
Nginx
Varnish
Cloudflare
DigitalOcean
Docker
Fluentd

Go
Java or Scala
Python
Ruby
Bash
C++
Common Lisp
Elixir
Erlang
Haskell
Lua for Nginx/Tarantool
.NET / C#
Node.js
Perl
PHP
Rust

Prometheus - Cool Stuff

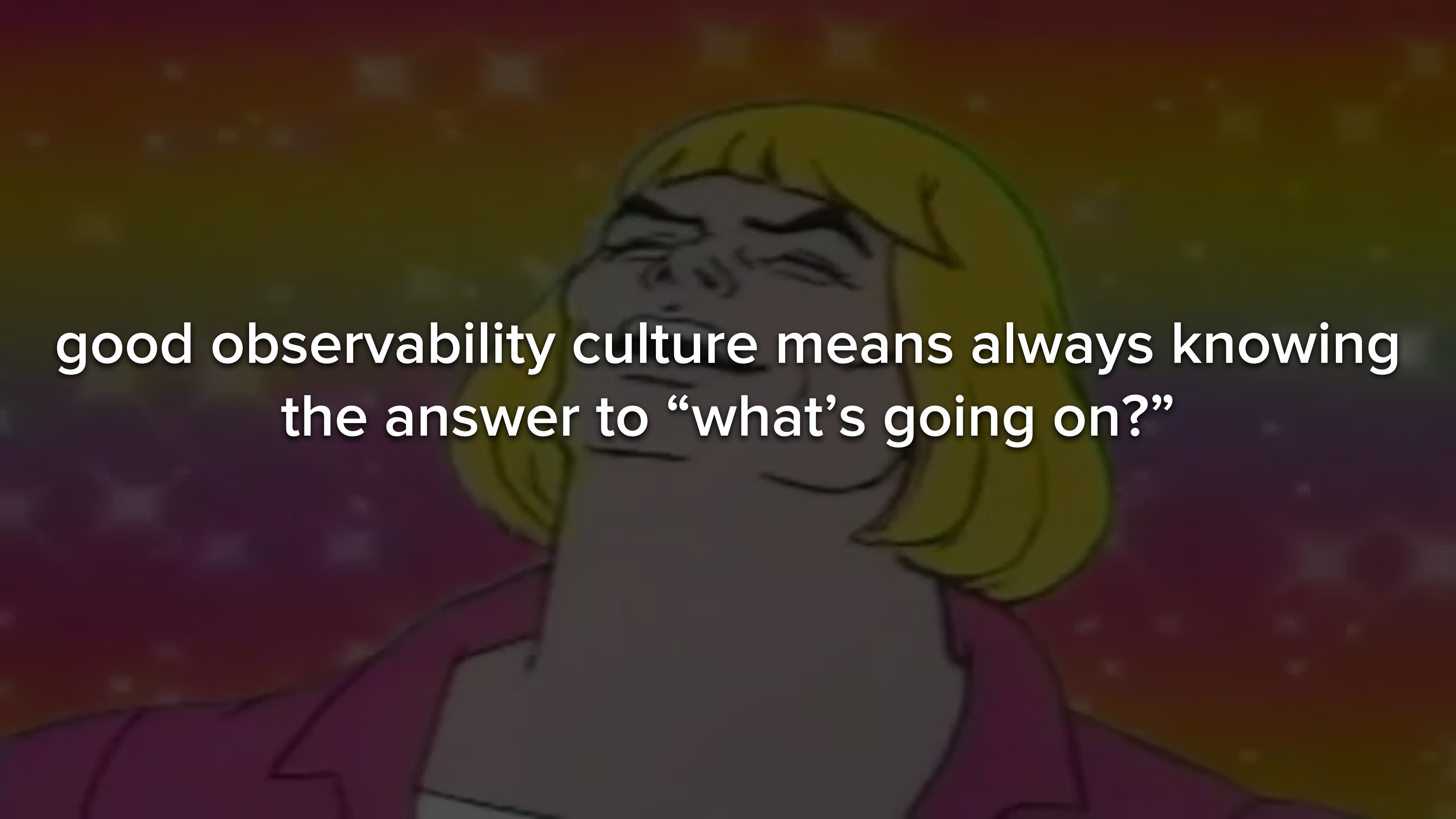
Alerts
if temperature > 30 for 3 days, send me an alert!

Relabeling
temperature{location=”Seattle, WA”} => temperature{city=”seattle”, state=”WA”}

Recording Rules
compute expensive queries regularly and save results to new metrics

Robust, Extendable Service Discovery
Kubernetes, GCE, Azure, OpenStack, EC2, Consul, DNS, Custom, etc

Endless Customization
federation, alerting routes/receivers/inhibits/integrations, scraping, remote read/write, limits



good observability culture means always knowing the answer to “what’s going on?”

Measure everything, and then use that data to make your life better

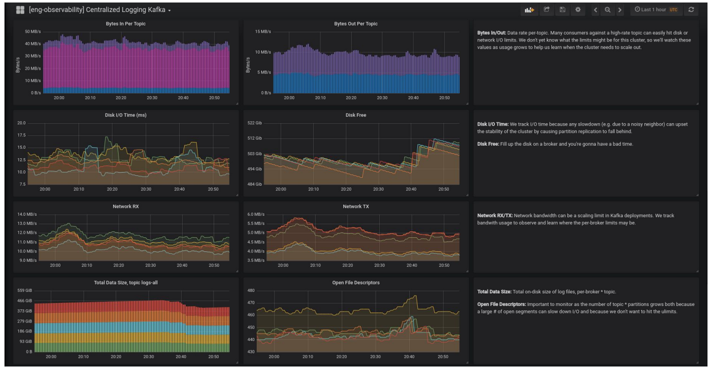
Dashboards are good

In an incident, “what’s going on” stands out 💸

Democratizing knowledge of “what’s going on” 😎

Onboarding 👶

More than just graphs! 🗣 📎



Bytes In/Out: Data rate per-topic. Many consumers against a high-rate topic can easily hit disk or network I/O limits. We don't yet know what the limits might be for this cluster, so we'll watch these values as usage grows to help us learn when the cluster needs to scale out.

Disk I/O Time: We track I/O time because any slowdown (e.g. due to a noisy neighbor) can upset the stability of the cluster by causing partition replication to fall behind.

Disk Free: Fill up the disk on a broker and you're gonna have a bad time.

Network RX/TX: Network bandwidth can be a scaling limit in Kafka deployments. We track bandwith usage to observe and learn where the per-broker limits may be.

Total Data Size: Total on-disk size of log files, per-broker * topic.

Open File Descriptors: Important to monitor as the number of topic * partitions grows both because a large # of open segments can slow down I/O and because we don't want to hit the ulimits.

Friends of Prometheus - Grafana

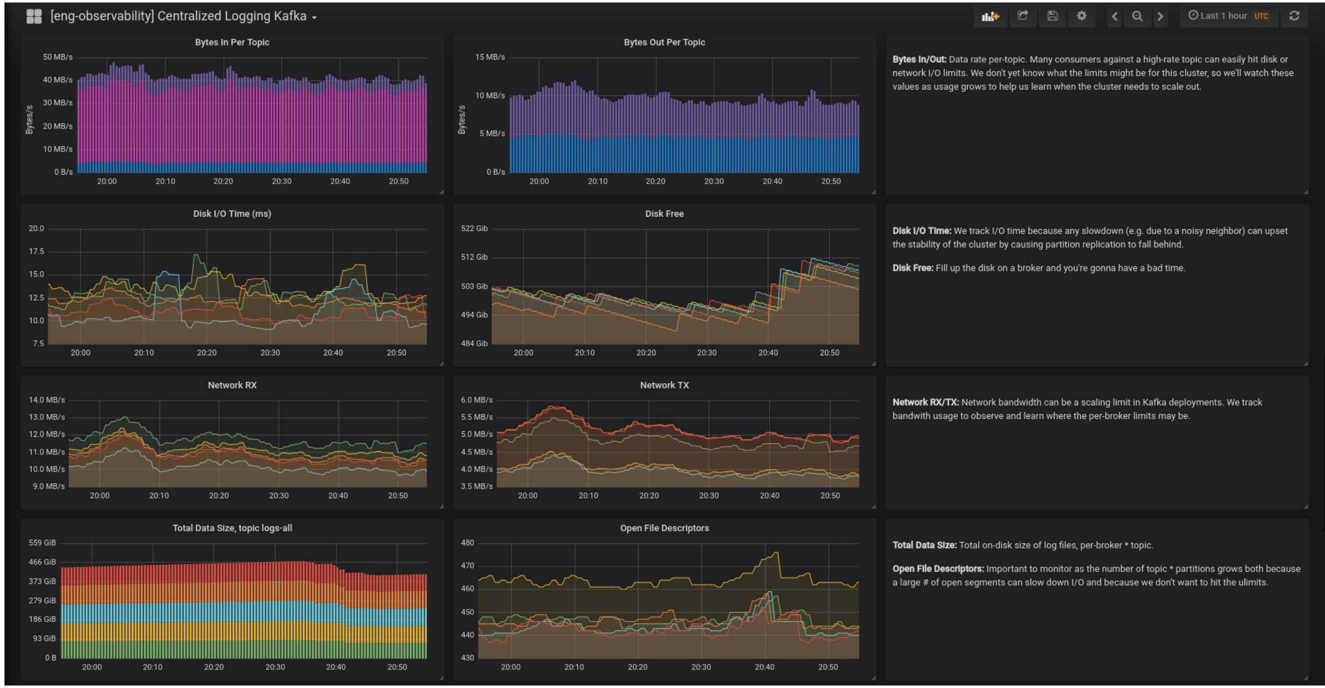
Dashboards

JS ✨

login/teams/rbac

API

More than just Prometheus!



Bytes In/Out: Data rate per-topic. Many consumers against a high-rate topic can easily hit disk or network I/O limits. We don't yet know what the limits might be for this cluster, so we'll watch these values as usage grows to help us learn when the cluster needs to scale out.

Disk I/O Time: We track I/O time because any slowdown (e.g. due to a noisy neighbor) can upset the stability of the cluster by causing partition replication to fall behind.

Disk Free: Fill up the disk on a broker and you're gonna have a bad time.

Network RX/TX: Network bandwidth can be a scaling limit in Kafka deployments. We track bandwidth usage to observe and learn where the per-broker limits may be.

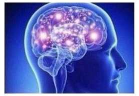
Total Data Size: Total on-disk size of log files, per-broker * topic.

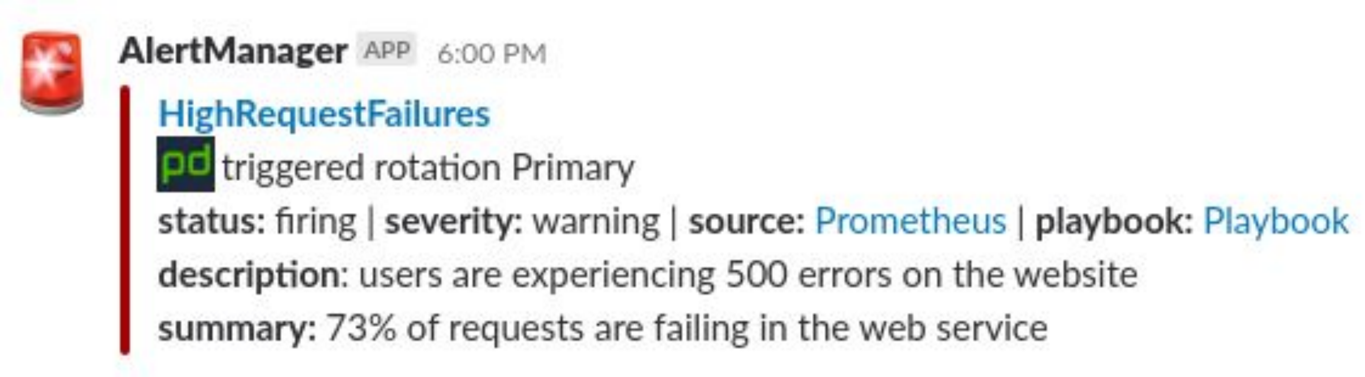
Open File Descriptors: Important to monitor as the number of topic * partitions grows both because a large # of open segments can slow down I/O and because we don't want to hit the ulimits.

Alerts are good

You know when things break 🙄

A good alert can be handled by anyone 📱

More alerts -> more context 


AlertManager APP 6:00 PM

HighRequestFailures

pd triggered rotation Primary

status: firing | **severity:** warning | **source:** Prometheus | **playbook:** Playbook

description: users are experiencing 500 errors on the website

summary: 73% of requests are failing in the web service

“On-call doesn’t have to suck” -Cindy Sridharan @copyconstruct https://medium.com/@copyconstruct/on-call-b0bd8c5ea4e0

Friends of Prometheus - Alertmanager

Alertmanager is an **awesome** friend

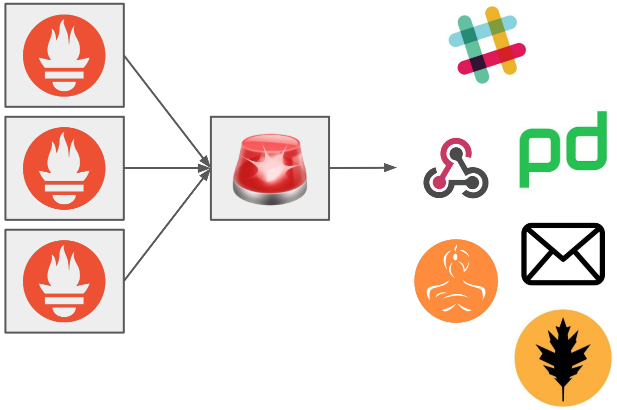
Alerts are PromQL queries

Prometheus evaluates alerts on a cadence

If alert is true, send to Alertmanager

Deduplicate, Group, Notify

HA capabilities mean you never miss an alert



Observability Culture - Software

Build with Observability in mind rather than adding it later

Increasing returns as you build new systems

Clearer code organization/architectures

Quantify customer experience

SLOs and Error Budgets are right there

Make data driven decisions

@

DigitalOcean

STORY TIME!

Stories - Service Specific Dashboards/Alerts

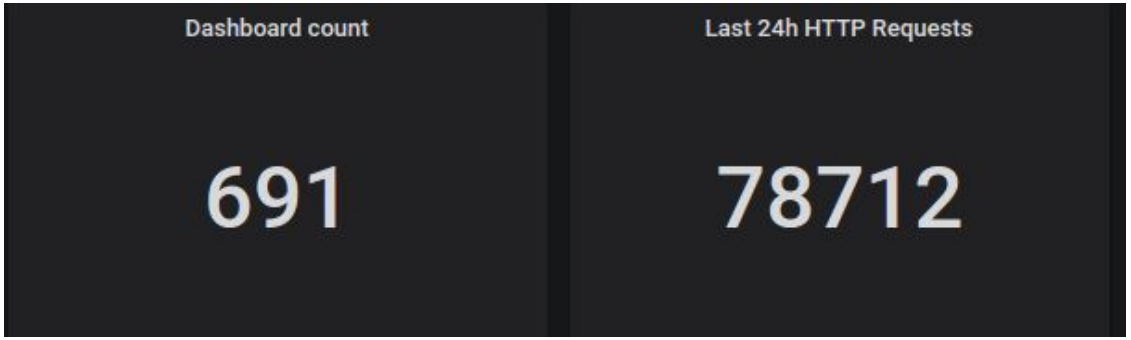
Grafana

30+ Team Directories

700+ Dashboards

Alerts!

Searchability 💯



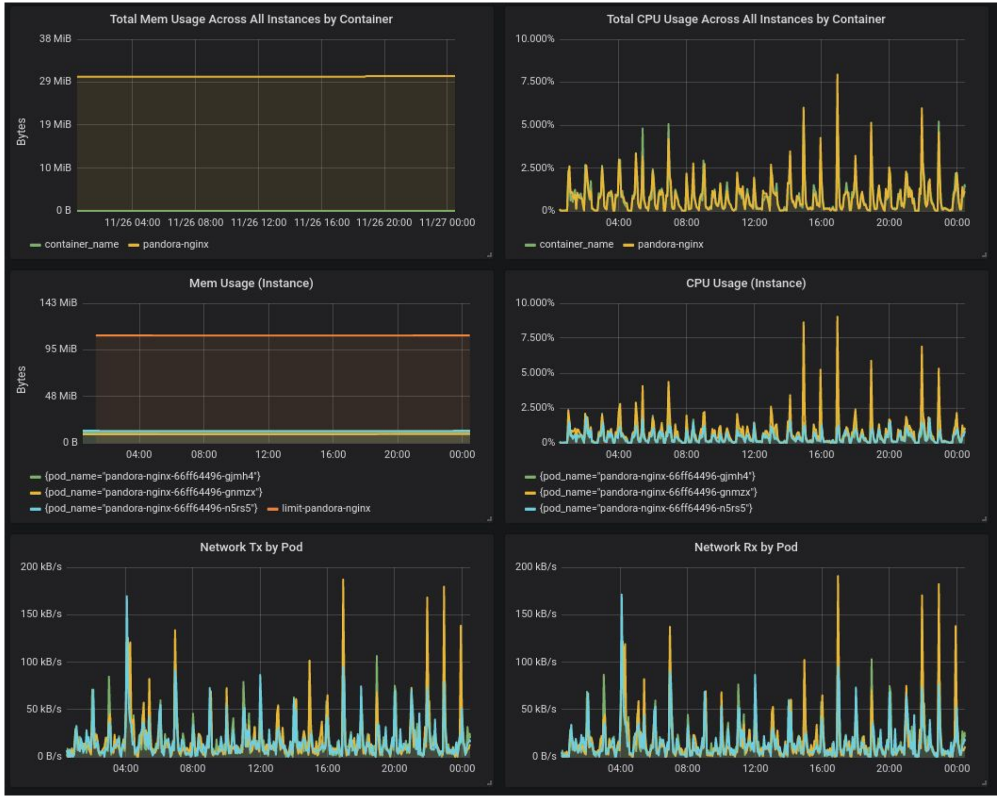
Dashboard count	Last 24h HTTP Requests
691	78712

Stories - Shared Dashboards

Grafana templating allows systems with similar metrics to share a dashboard

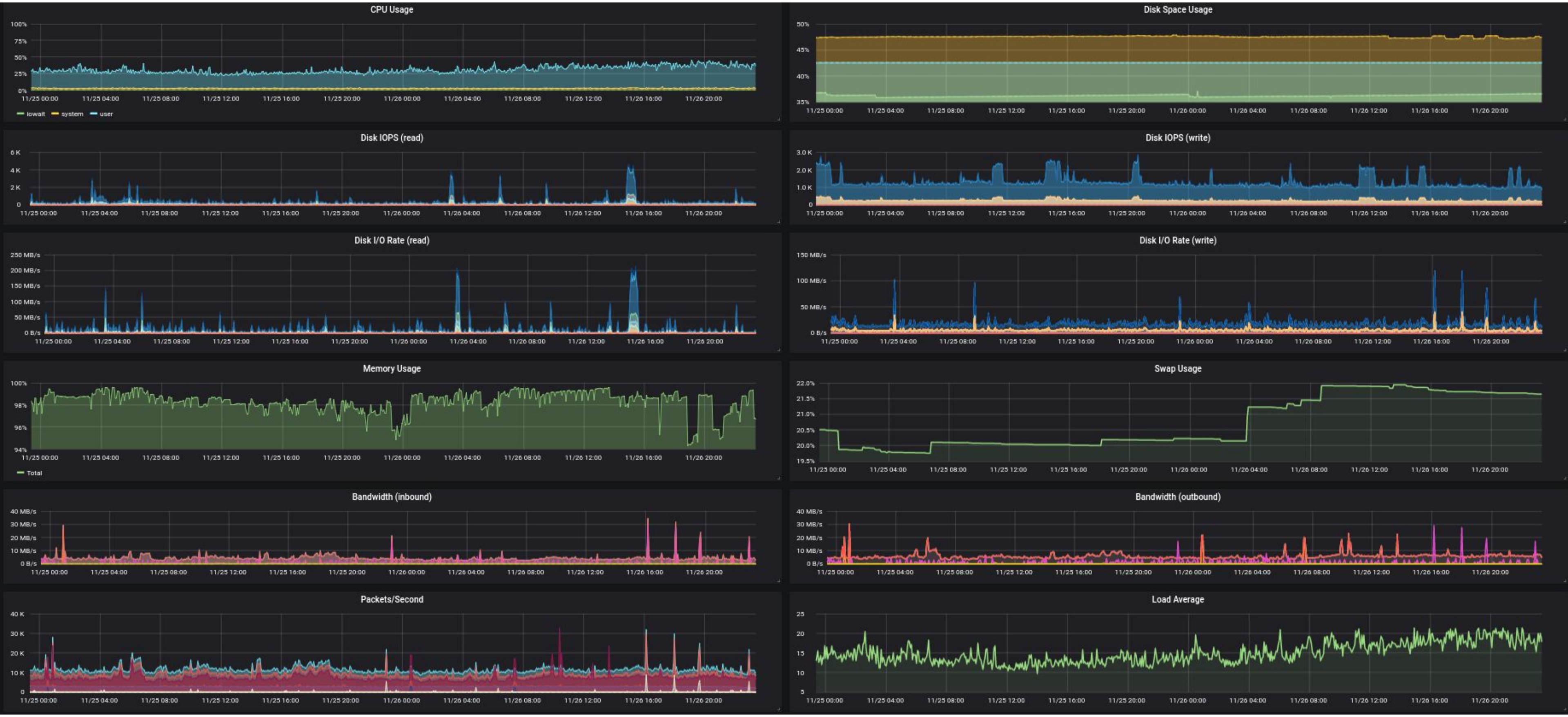
k8s applications CPU/RAM/Net

Databases

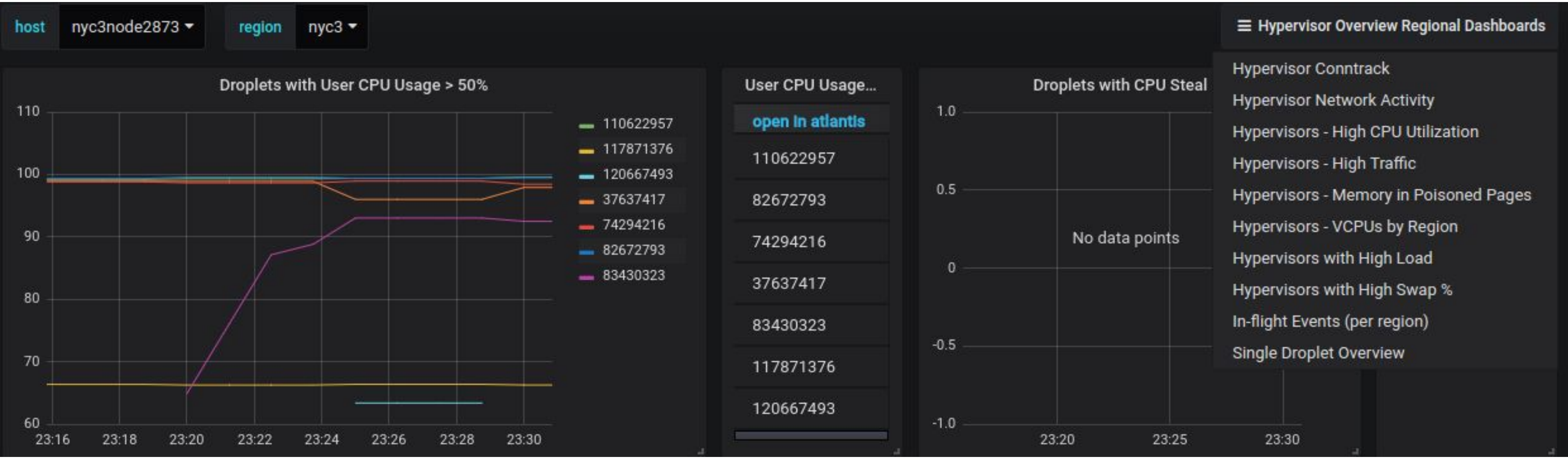

Total Mem Usage Across All Instances by Container
Total CPU Usage Across All Instances by Container
Mem Usage (Instance)
CPU Usage (Instance)
Network Tx by Pod
Network Rx by Pod
Bytes
container_name
pandora-nginx
{pod_name="pandora-nginx-66ff64496-gjmh4"}
{pod_name="pandora-nginx-66ff64496-gnmzx"}
{pod_name="pandora-nginx-66ff64496-n5rs5"}
limit-pandora-nginx

digitalocean.com

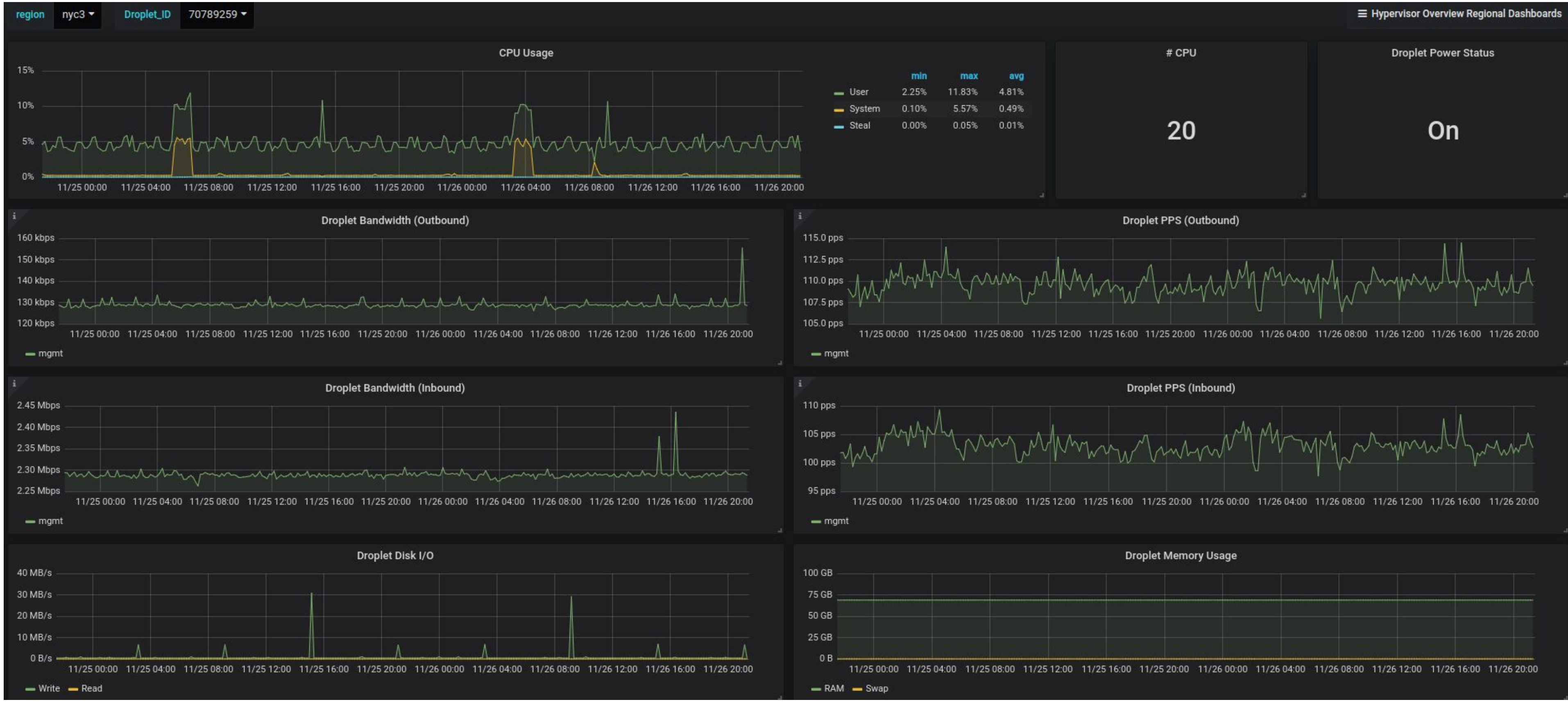
Stories - Hypervisor Metrics



Stories - Fleet Overview Graphs

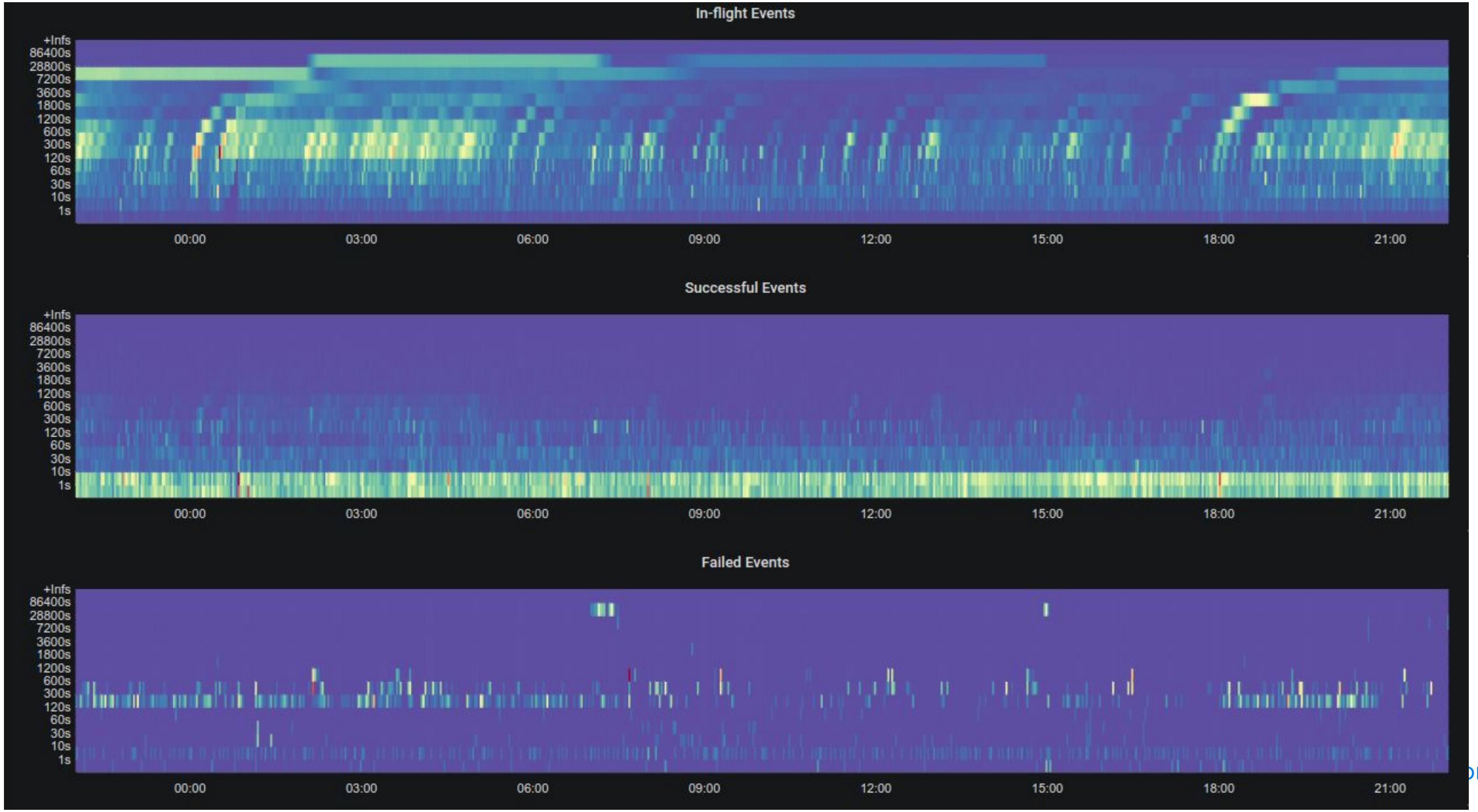


Stories - Droplet Metrics



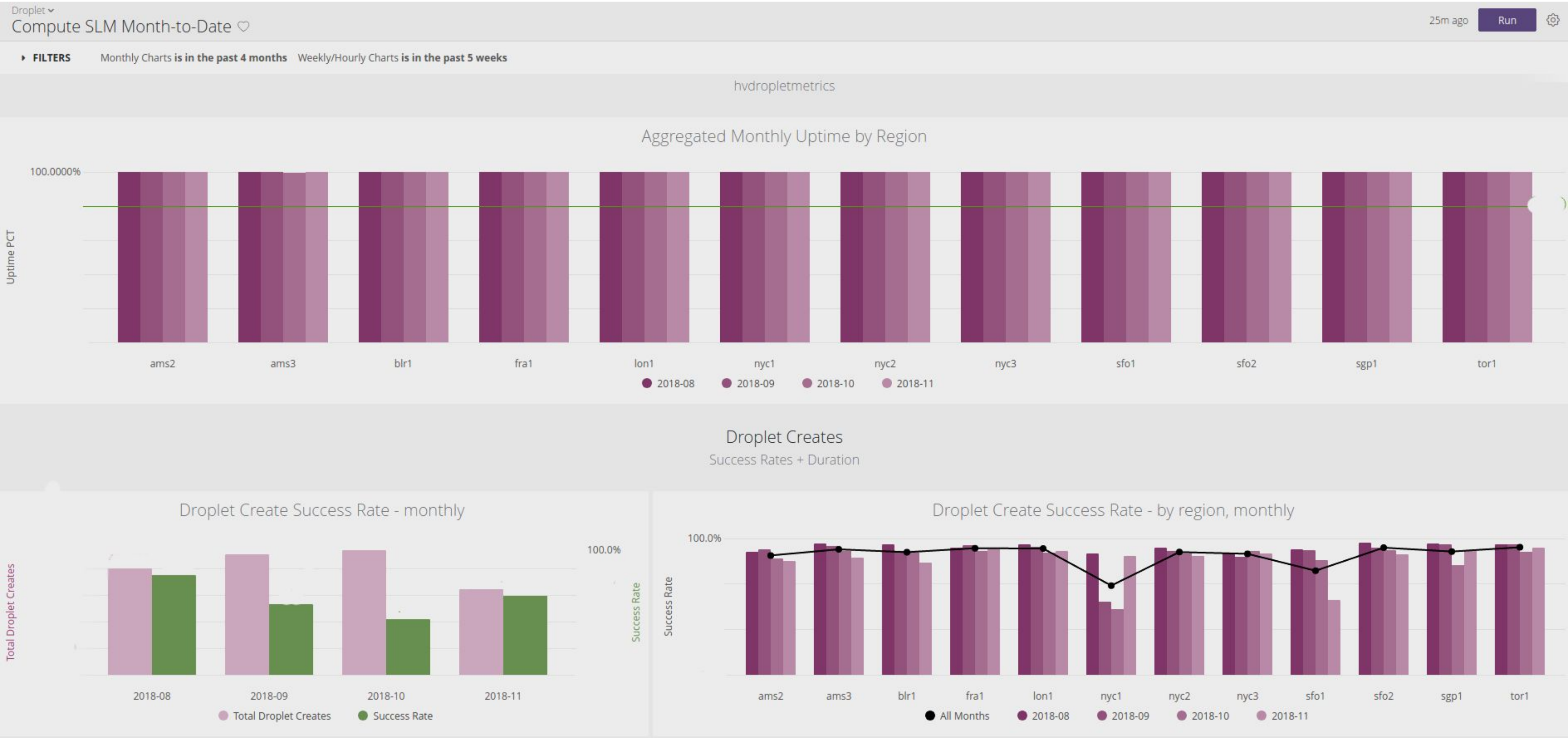
	min	max	avg
User	2.25%	11.83%	4.81%
System	0.10%	5.57%	0.49%
Steal	0.00%	0.05%	0.01%

Stories - Event Visualization



om

Stories - SLM/SLOs

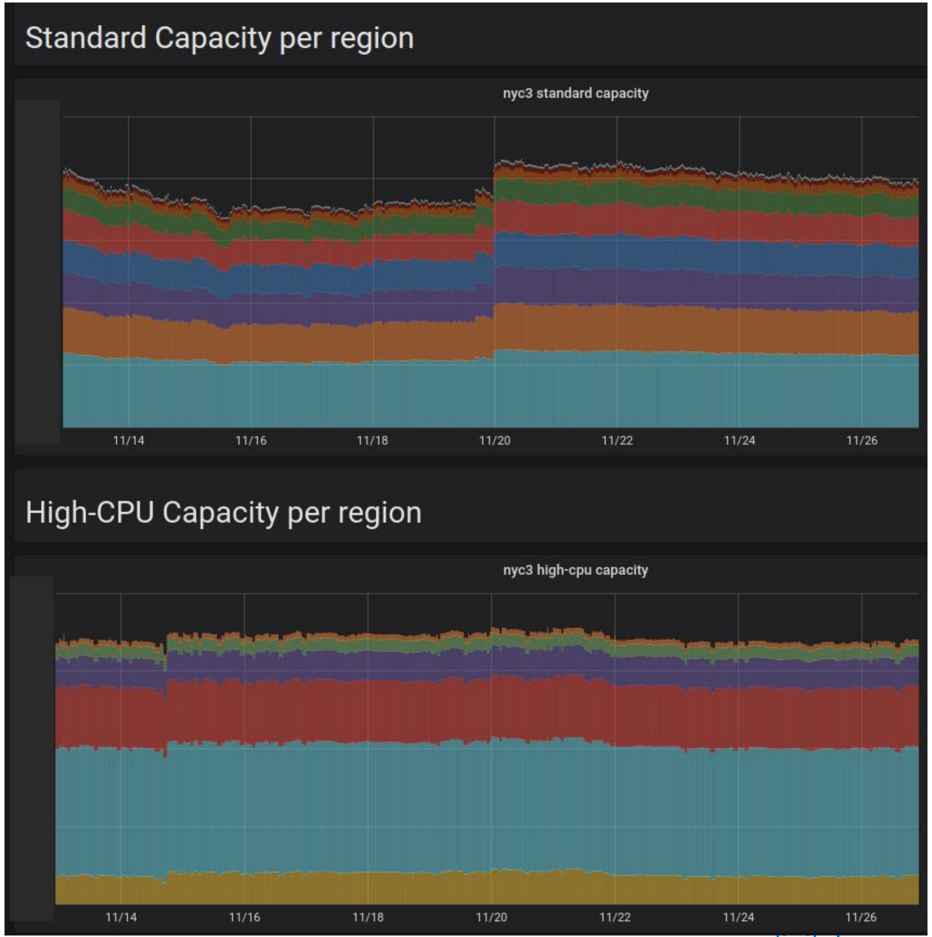


Stories - Capacity

Custom exporters

Relabel configs

Alerts!



Prometheus at DigitalOcean

192 Prometheus servers

200M+ time series

2M+ samples/second

Scaling Prometheus

If you take one thing away from this talk

Every **permutation of labels** in Prometheus creates a new time series

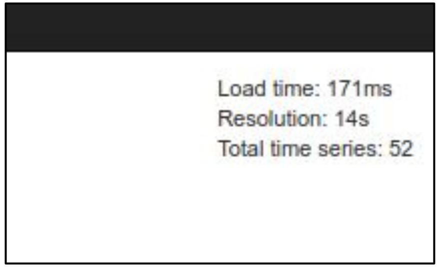
Individual queries should use **hundreds not thousands** of time series **(at most)**

Queries that **operate on** thousands of time series will overload Prometheus

Work out your query in the **Console** before graphing

Avoid **high cardinality** labels*

*unless you *really* know what you're doing



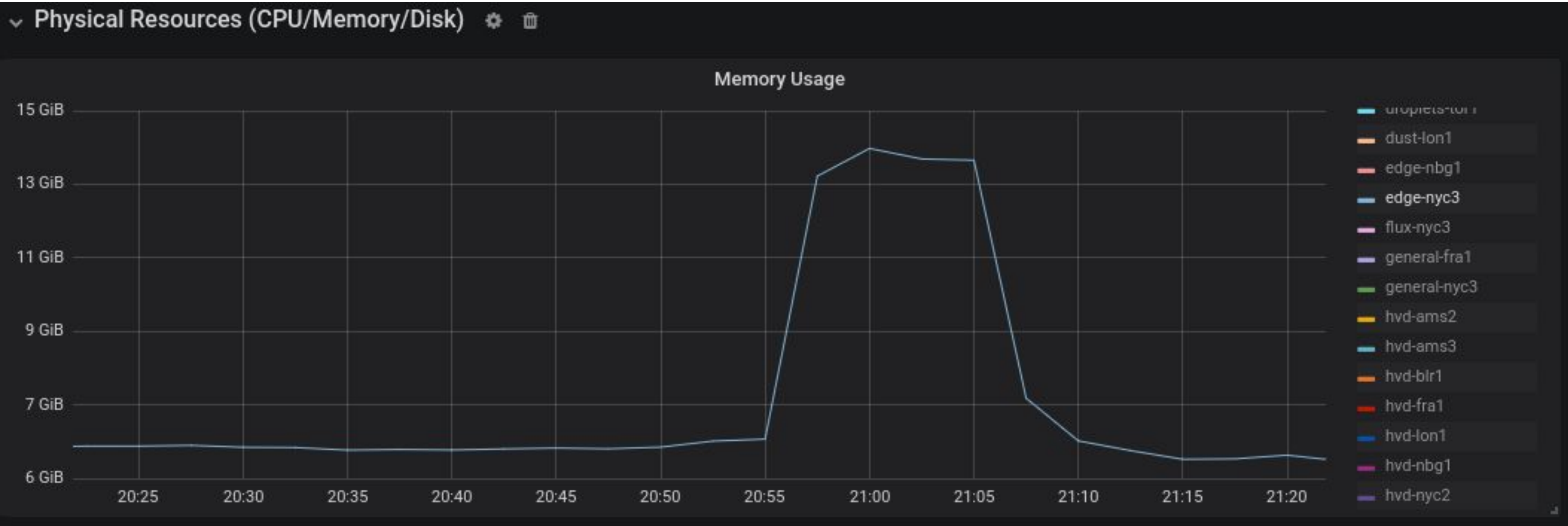
Load time: 171ms
Resolution: 14s
Total time series: 52

query: requests_total{path=~"(/status|/)", method=~"(GET|POST)"}
{__name__="requests_total", path="/status", method="GET", instance="10.0.0.1:80"}
{__name__="requests_total", path="/status", method="POST", instance="10.0.0.3:80"}
{__name__="requests_total", path="/", method="GET", instance="10.0.0.2:80"}

Leave Headroom

Prometheus needs **memory** headroom to execute queries

High cardinality metrics, big dashboards, long retention, concurrent queries



Shard Metrics Functionally

Metrics will get too big to query

10s of millions of timeseries on a single server

You will need to shard the metrics across multiple Prometheus servers

Pick a dimension that is a query boundary

Never split metrics that you want to query together

Split on region, service, team

Not instance, application

HA Prometheus Pairs

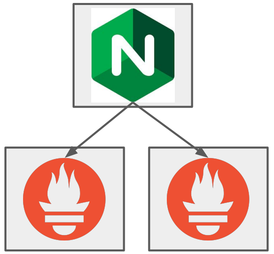
Deploy Prometheus in pairs (at least?)

Scrape the same metrics

Proxy queries Active-Active, Active-Passive

Proxies are good

If something bad happens, you’re good(ish)!



Find Bad Metrics

https://www.robustperception.io/which-are-my-biggest-metrics

```
topk(10, count by (__name__)({__name__=~".+"}))
```

```
topk(10, count by (__name__, job)({__name__=~".+"}))
```

```
topk(10, count by (job)({__name__=~".+"}))
```

```
sum(scrape_samples_scraped) by (job)
```

Expensive, but worth it

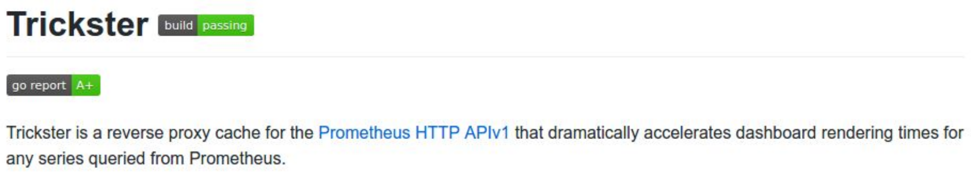
Friends of Prometheus - Trickster

github.com/Comcast/trickster

Caches query results

(Partial, or full)

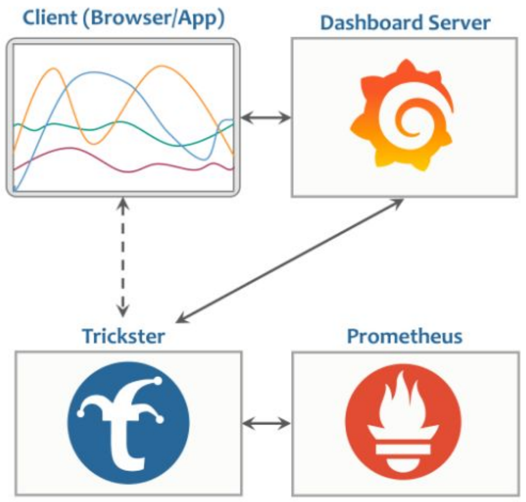
Step boundary normalization



Trickster build passing

go report A+

Trickster is a reverse proxy cache for the Prometheus HTTP APIv1 that dramatically accelerates dashboard rendering times for any series queried from Prometheus.



Scaling Prometheus Adoption

Creating Value

Engineers (Dev/SRE/Support) are paid to create business value

New Features

Maintenance

Every engineer has customers

Maintenance

Noun
the process of maintaining or preserving something

“revenue protection”*

Measure performance

Debug poor performance

Support users

This is what **Observability tools do!**

*https://twitter.com/DarkAndNerdy/status/1070690952472350720

Revenue Protection

Systems are rarely built with “revenue protection” in mind

Chasing features to provide value is natural

New shiny can tarnish the old dull

Observability tools help protect the 💰

So if you want to provide all the value...

Build Observability into your application in the **beginning** under *pressure* to deliver the new shiny

OR

Retrofit Observability **later**, under *pressure* because something is broken or hard to maintain

Adding Observability with Prometheus

With limited time be efficient as possible, Ctrl+c, Ctrl+v!

In an organization, **patterns proliferate**

Prometheus metrics are custom for every application

Metrics data quality is **more** dependent on your decisions

Prometheus is deceptively simple

Basics are breezy

Creating meaningful custom metrics is harder

Operations are easy at first

Defining/Delivering quality alerts

Making great dashboards

Dealing with expensive queries

Be careful copying those patterns

If anyone gets this one come talk to me after so we can be best friends

On the Path to Maintenance

Every decision you make is a multiplier on your ability to efficiently maintain a system.

But it is OK

Prometheus is great!

You will get results, no matter what.

Done well, you can 10000x those results tho

Especially in medium+ size orgs

You need people for this.

You need people to do this

What

Ensure engineers can *effectively utilize* limited Observability time

Accelerate *basic* Prometheus understanding

How

Operate

Centralize Knowledge and Configuration

Consult

Operate

“own” Prometheus and friends

Make sure no one else has to operate Prometheus

Upgrade and improve

Be on-call

Watch out for metric/alerting regressions

Make sharding decisions

Centralize Configuration

Make it easy to copy good patterns

Chef, Ansible, Git, Docker

Provide a **good** abstraction

Centralize Knowledge

Standardize and document what to measure (Golden Signals, RED, USE)

Establish patterns around similar use cases (RPC, Physical Resources, k8s)

Document good examples (client usage, dashboards, alerts)

Document how to alert good ("On-call doesn't have to suck" -Cindy Sridharan @copyconstruct https://medium.com/@copyconstruct/on-call-b0bd8c5ea4e0)

Document how to use your internal tools

What I Did

```
---
hosts:
 prod-elauneind-nova:
   port: 9123
   kubernetes:
     cluster: nova
     namespace: observability
     application: elauneind
```

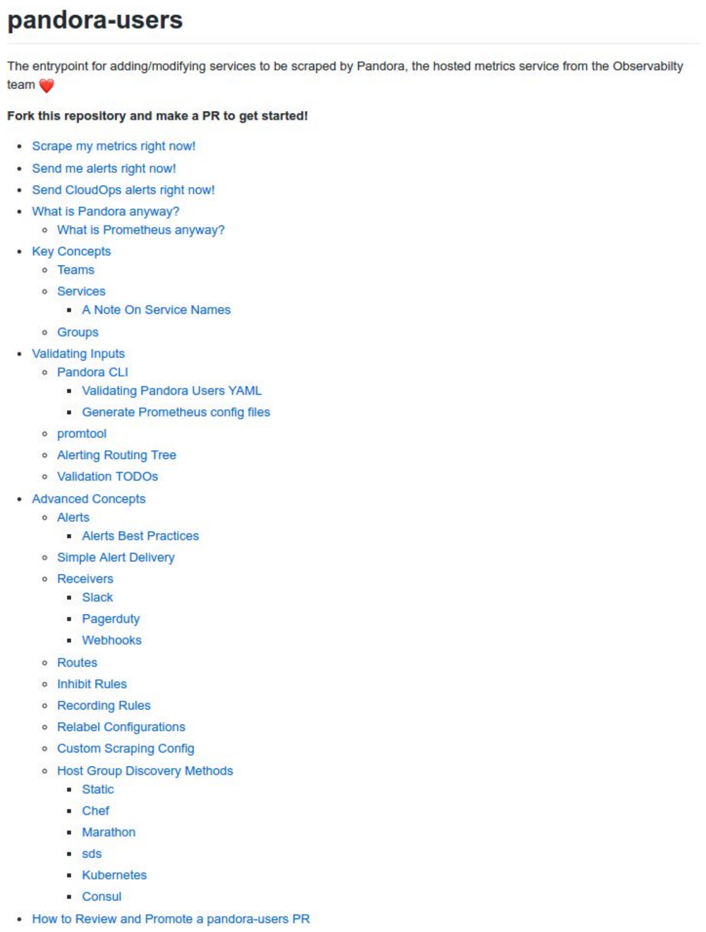
- YAML abstractions
- GitOps
- Observability reviews PRs
- Created confusion and delay

```
alerts:
- alert: Elauneind Prod Down
 expr: |-
   sum(up{job="prod-elauneind"}) == 0
 for: 10m
 labels:
   team: eng-observability
   service: elauneind
   severity: warning
 annotations:
   description: "elauneind down in prod {{ $labels.cluster }}"
```

```
---
owner: eng-observability@digitalocean.com
slack_channels:
 - "#observability-alerts"
pagerduty_keys:
 - momsspaghetti
severity_routing:
 warning:
   - "#observability-alerts"
 critical:
   - "#observability-alerts"
   - momsspaghetti
```

What I Did

Wrote a lot of documentation

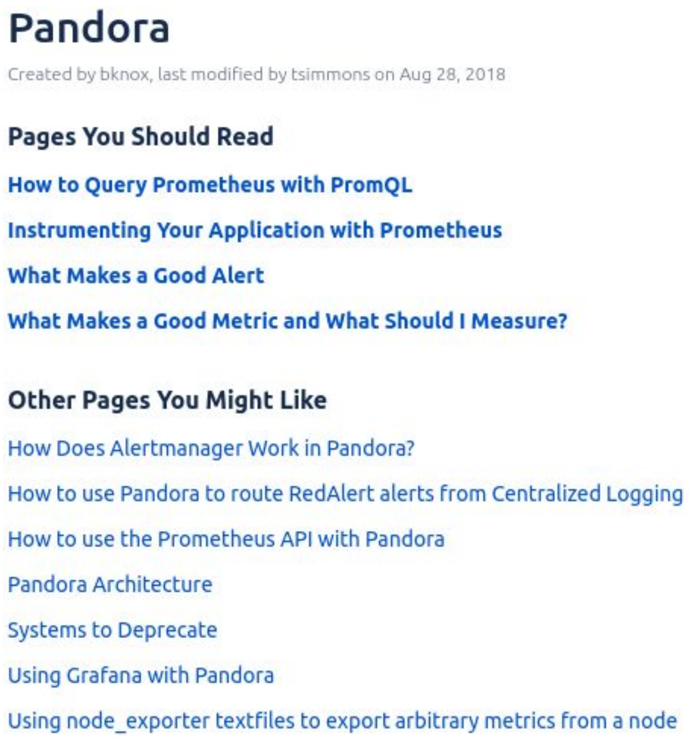


pandora-users

The entrypoint for adding/modifying services to be scraped by Pandora, the hosted metrics service from the Observabilty team ❤️

Fork this repository and make a PR to get started!

- Scrape my metrics right now!
- Send me alerts right now!
- Send CloudOps alerts right now!
- What is Pandora anyway?
  - What is Prometheus anyway?
- Key Concepts
  - Teams
  - Services
    - A Note On Service Names
  - Groups
- Validating Inputs
  - Pandora CLI
    - Validating Pandora Users YAML
    - Generate Prometheus config files
  - promtool
  - Alerting Routing Tree
  - Validation TODOs
- Advanced Concepts
  - Alerts
    - Alerts Best Practices
  - Simple Alert Delivery
  - Receivers
    - Slack
    - Pagerduty
    - Webhooks
  - Routes
  - Inhibit Rules
  - Recording Rules
  - Relabel Configurations
  - Custom Scraping Config
  - Host Group Discovery Methods
    - Static
    - Chef
    - Marathon
    - sds
    - Kubernetes
    - Consul
- How to Review and Promote a pandora-users PR



Pandora

Created by bknox, last modified by tsimmons on Aug 28, 2018

Pages You Should Read

How to Query Prometheus with PromQL

Instrumenting Your Application with Prometheus

What Makes a Good Alert

What Makes a Good Metric and What Should I Measure?

Other Pages You Might Like

How Does Alertmanager Work in Pandora?

How to use Pandora to route RedAlert alerts from Centralized Logging

How to use the Prometheus API with Pandora

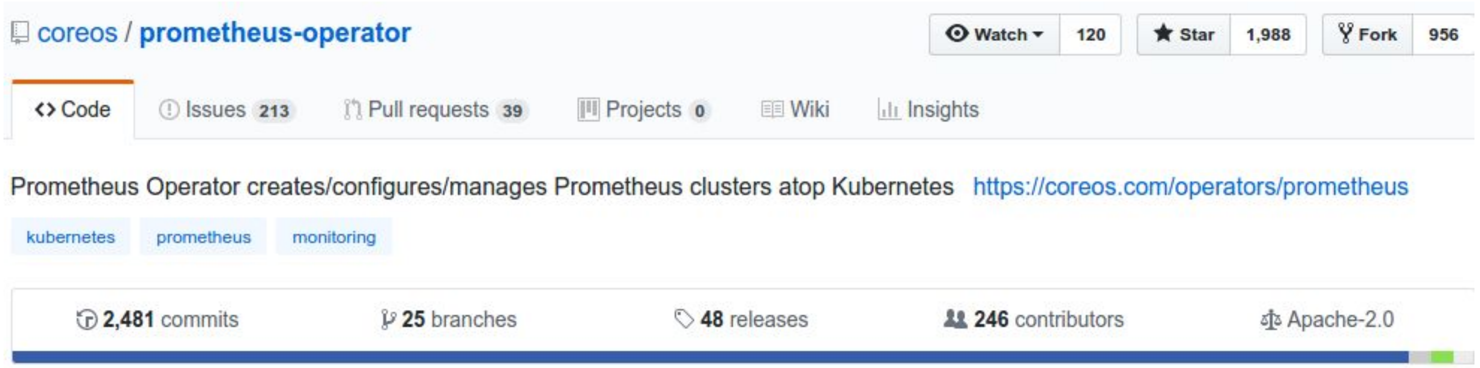
Pandora Architecture

Systems to Deprecate

Using Grafana with Pandora

Using node_exporter textfiles to export arbitrary metrics from a node

What I Would Do Today


coreos / prometheus-operator

Watch 120 | Star 1,988 | Fork 956

Code | Issues 213 | Pull requests 39 | Projects 0 | Wiki | Insights

Prometheus Operator creates/configures/manages Prometheus clusters atop Kubernetes https://coreos.com/operators/prometheus

kubernetes | prometheus | monitoring

2,481 commits | 25 branches | 48 releases | 246 contributors | Apache-2.0

Build administrative app on top of prometheus-operator

Configure scraping by pushing info from local app manifests (w/ CLI) into a stateless application that verifies/shards/places metrics by creating operator compatible CRDs

Consult

Point out getting started resources

Coach initial attempts

Enforce good patterns

Suggest advanced configuration

Lend a hand in an incident

Help solve larger organizational issues

You need people to do this

Operate

Centralize Knowledge

Centralize Configuration

Consult

Anyone can do this!

Manage Centralized Logging? Tracing? Exceptions?

What does adoption look like?

Some people will get it, love it, and get involved

Most will copypasta patterns

Some people will want the moon
- 1s sampling
- infinite retention
- massive label cardinality

What do you want? You want the moon?
Just say the word and I'll throw a lasso
around it and pull it down.

Friends of Prometheus - LTS - Thanos/M3

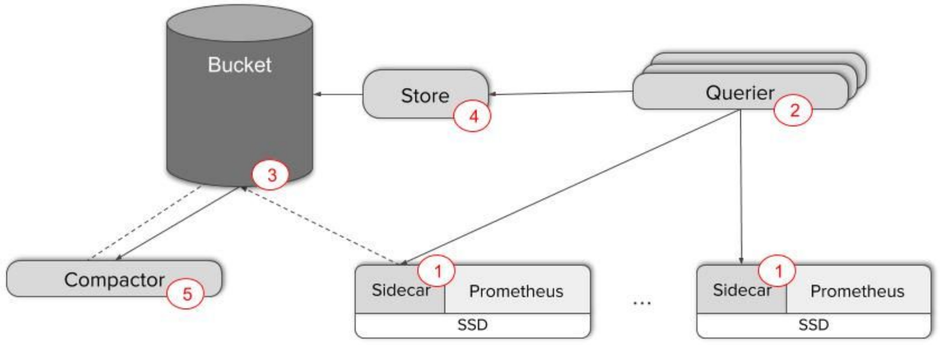
Thanos

Prometheus sidecars

Backed by block storage

Query Prometheus and block storage

Downsampling! Global query view!



M3

Prometheus remote read/write

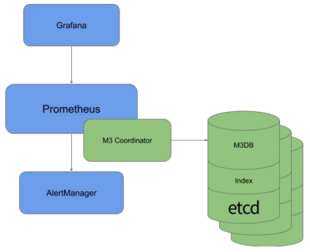
Replicated, distributed storage of metric data

Configurable time resolutions, metric durability

Support for multiple clusters based on etcd

Downsampling!

Global view of metrics



Conclusion

Prometheus is good, and has friends that make it even better

Having a healthy Observability culture has enormous value

Using Prometheus to successfully power an Observability culture is possible

You need people dedicated to making it good

The Greek Titan Prometheus stole fire, and gave it to humanity.

The computer thing Prometheus is trying to give you the ability to debug, monitor, and understand your systems.

Christian Griepenkerl (1839-1916): Die Strafe. Photo © Maicar Förlag - GML

The Greek Titan Prometheus was punished. He was chained to a rock and an eagle was sent to eat his regenerating liver every day for eternity.

Don’t punish your Prometheus by letting it go unmanaged.

DigitalOcean