Adopting Prometheus the Hard Way

Tim Simmons - Engineer - DigitalOcean
@timsimlol

Who am I

Tim Simmons
Engineer/Prometheus person
Observability Platforms
DigitalOcean

@timsimlol

#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

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

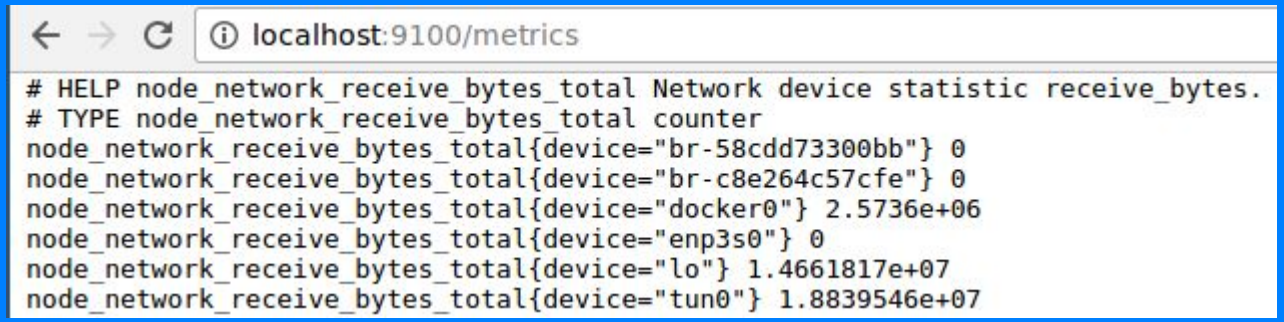
Prometheus - Labels

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



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


why do we observe

Creating Value

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

New Features vs. Maintenance **(Revenue Protection)**

Measure, Debug, Support

Systems are rarely built with “revenue protection” in mind

Observability tools are built for this

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


you should measure everything, and then use that data to make your life better

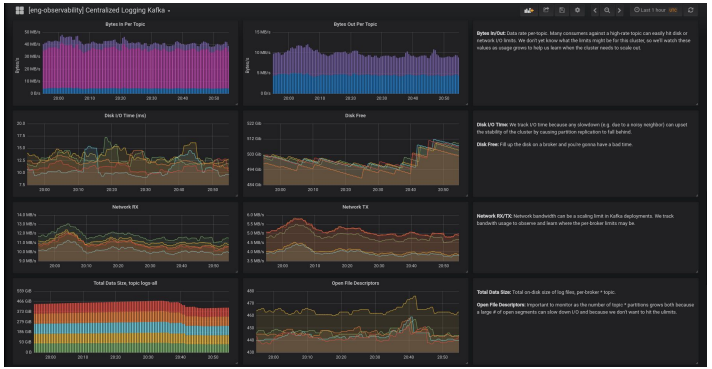
Dashboards are good

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

Democratizing knowledge of “what’s going on” 😎

Onboarding 👶

More than just graphs! 🗣️📎



Friends of Prometheus - Grafana

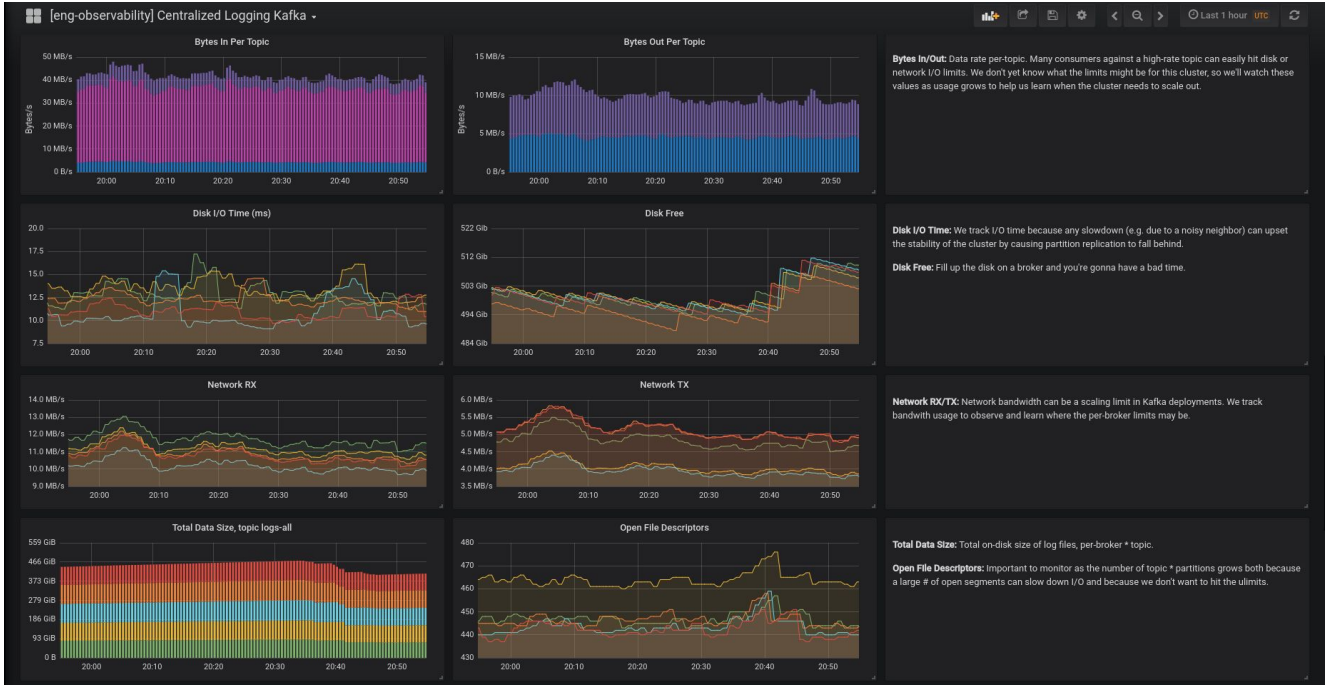
Dashboards

JS ✨

login/teams/rbac

API

More than just Prometheus!



Alerts are good

You know when things break 🙄

A good alert can be handled by anyone 📱

More alerts -> more context 

digitalocean.com

“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

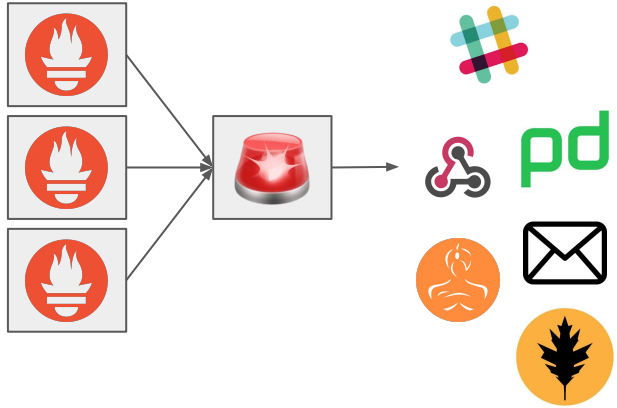
Alerts are PromQL queries

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.com

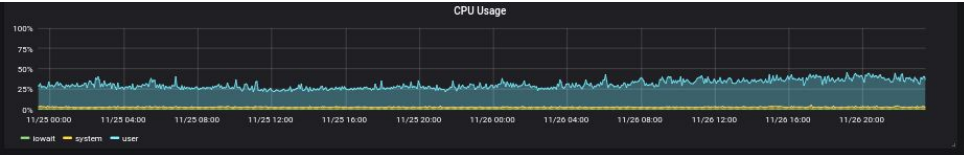
@

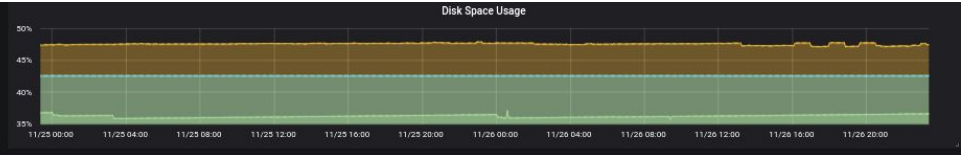
DigitalOcean

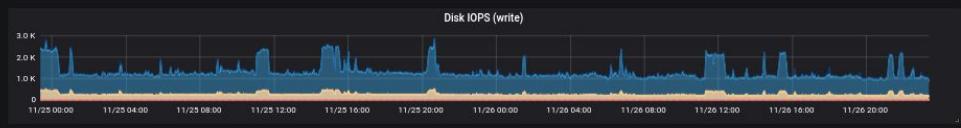
digitalocean.com

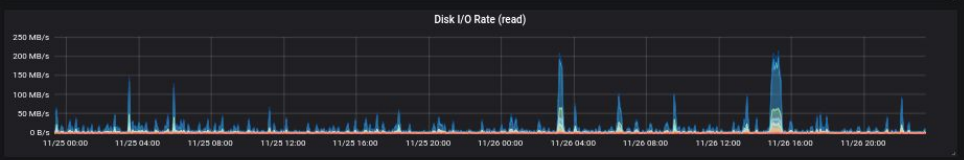
STORY TIME!

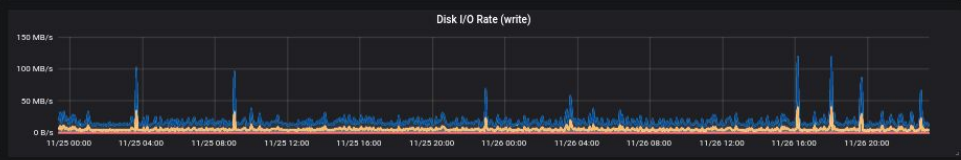
Stories - node_exporter

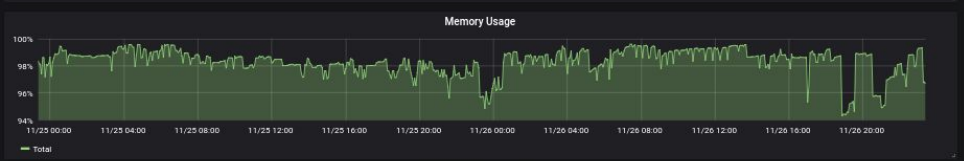


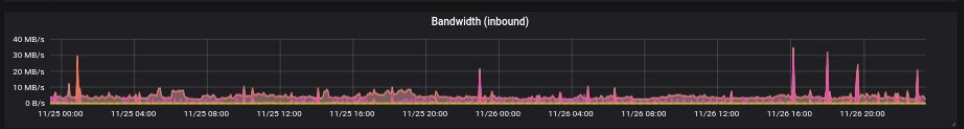


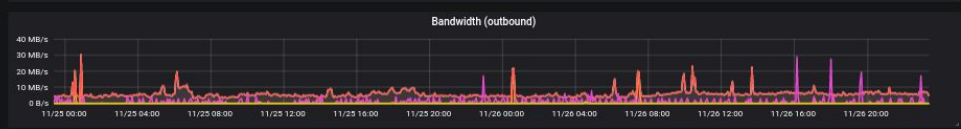


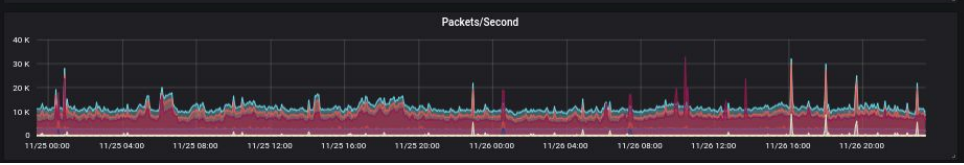


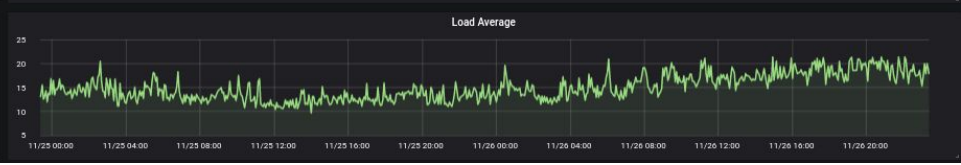






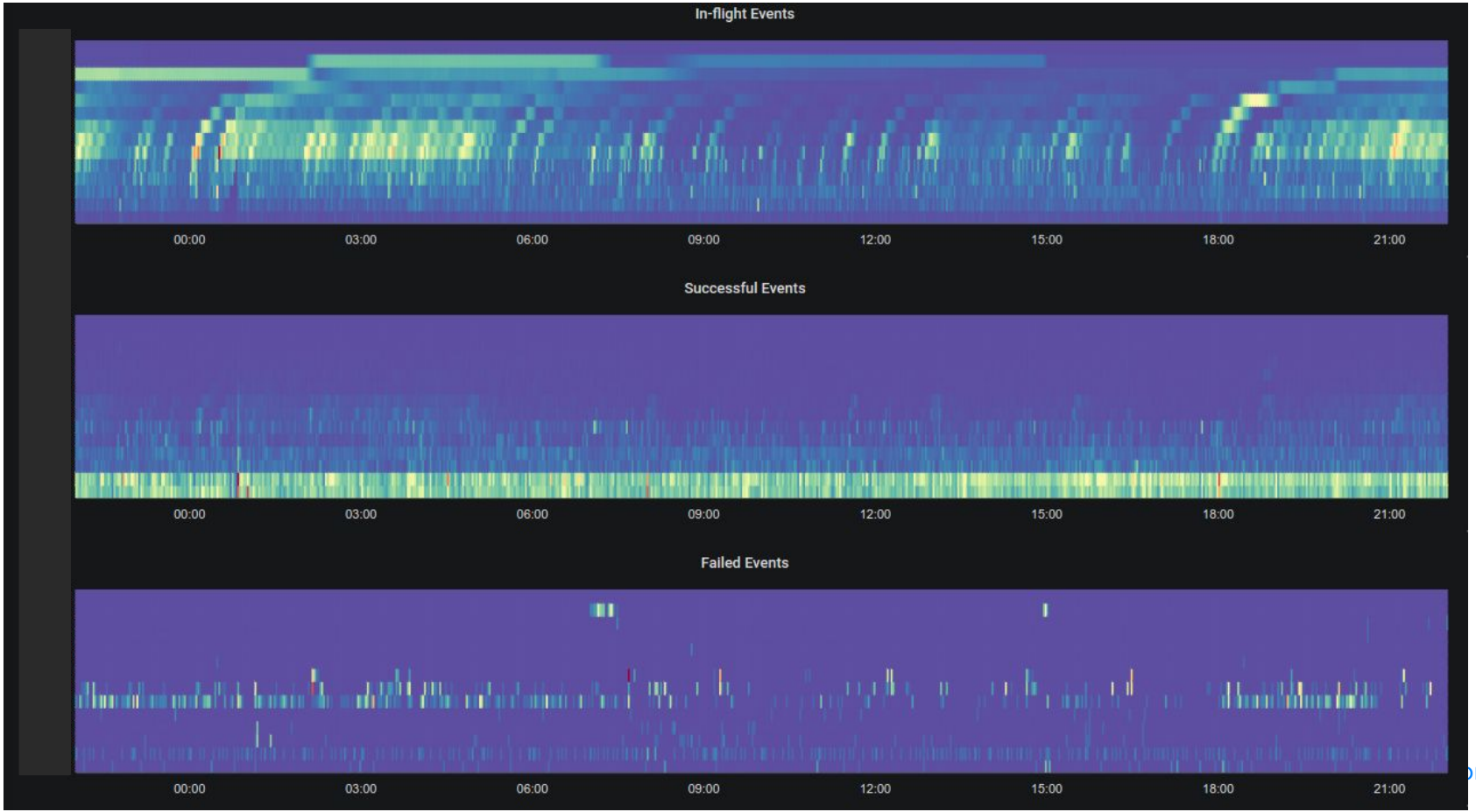






Stories - Droplet Metrics

Stories - Event Visualization



Stories - SLM/SLOs

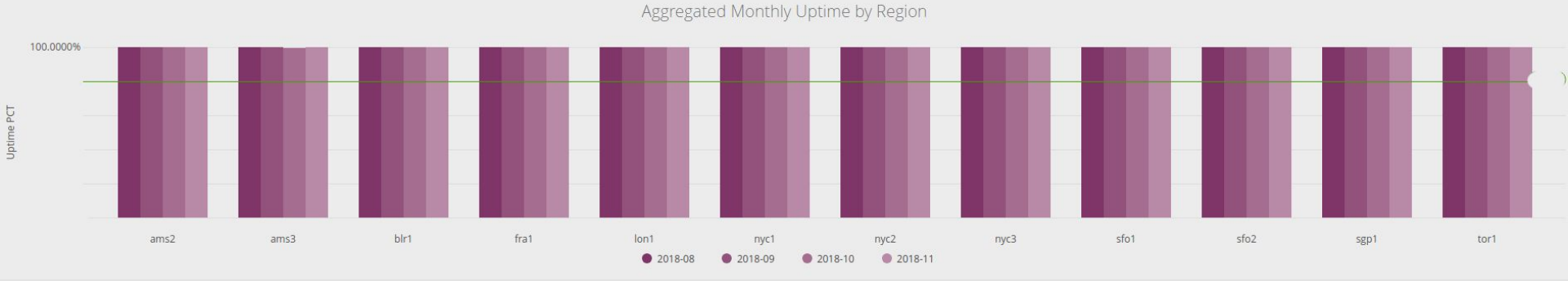
Droplet

Compute SLM Month-to-Date ♡

25m ago Run

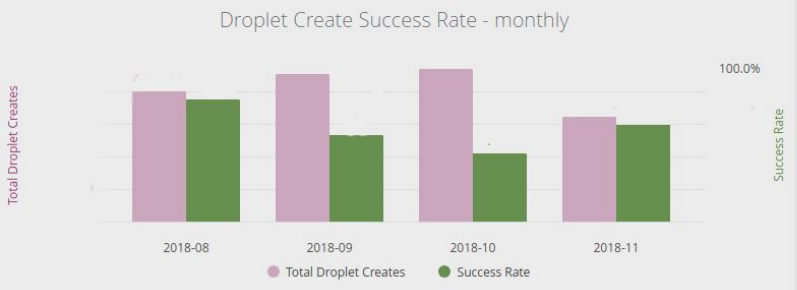
FILTERS Monthly Charts **is in the past 4 months** Weekly/Hourly Charts **is in the past 5 weeks**

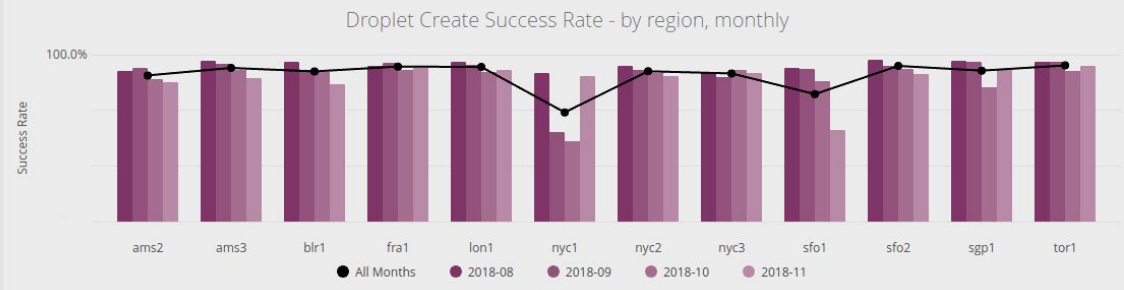
hvdropletmetrics



Droplet Creates

Success Rates + Duration





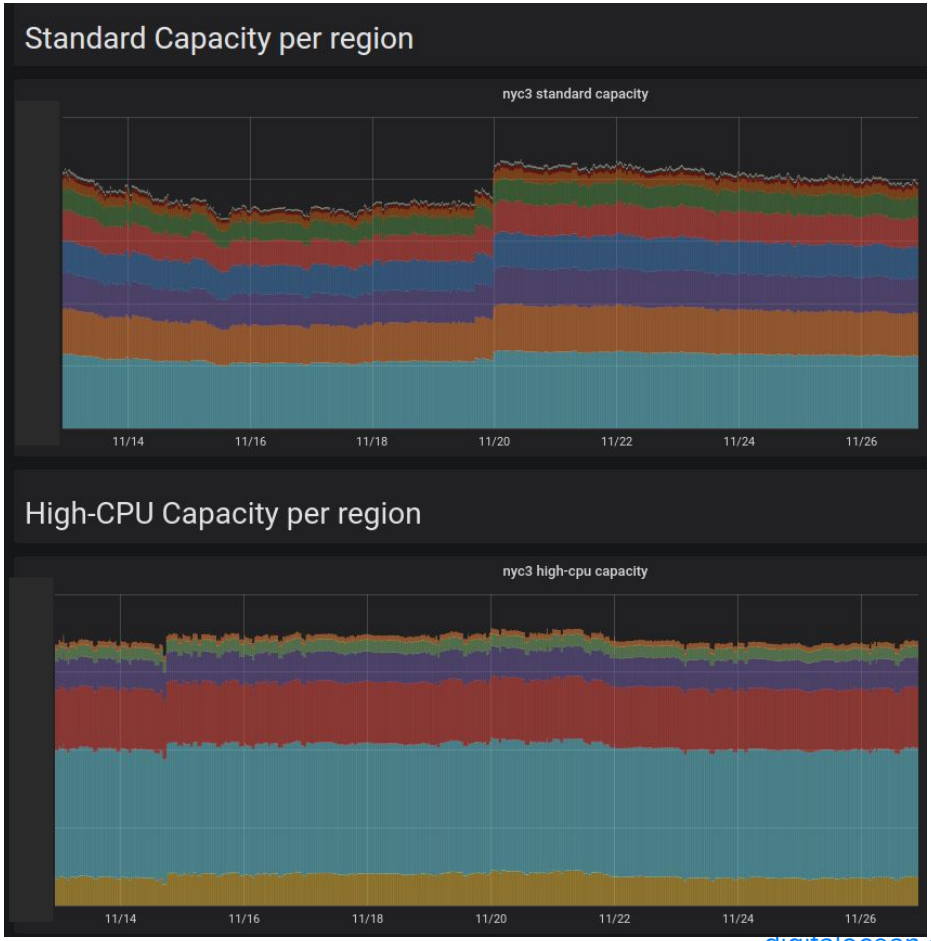
Stories - Capacity

Custom exporters

Relabel configs

Alerts!

Custom exporters <3



Prometheus at DigitalOcean

192 Prometheus servers

200M+ time series

2M+ samples/second

Scaling Prometheus - More Metrics, More Problems

High cardinality metrics

Memory usage spikes

Reasonable metric sharding

High Availability

Finding abusive metrics

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*

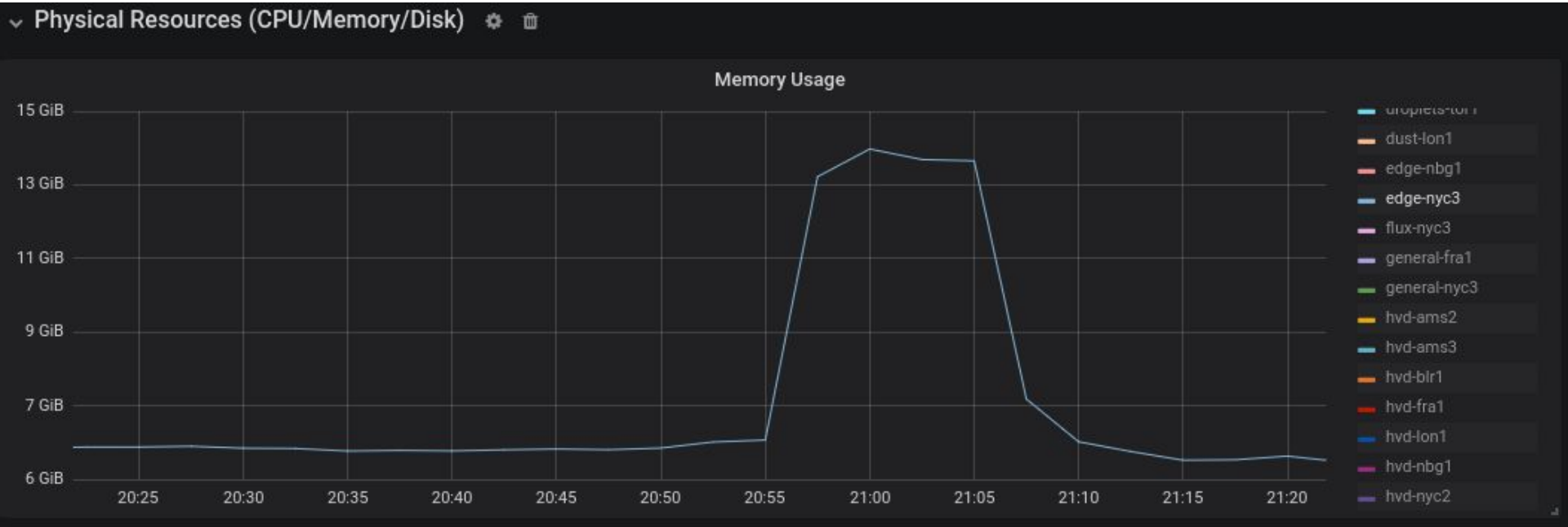
*unless you *really* know what you're doing

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

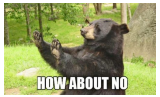
Prometheus needs **memory** headroom to execute queries

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

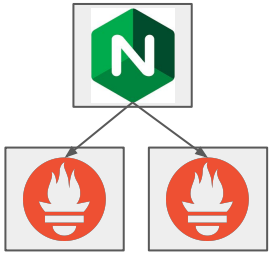
Deploy Prometheus in pairs (at least?)

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

Scaling Prometheus Adoption

Providing *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.

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 for Prometheus

Watch out for metric/alerting regressions

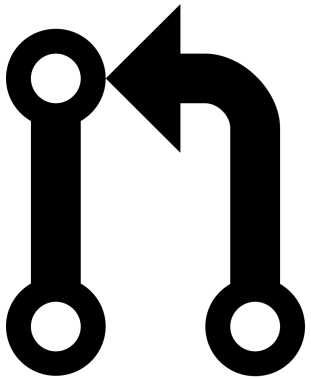
Make sharding decisions

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

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

What I Tried To Do

Centralize configuration to enable copying

Provide a good alerting framework

Expand service discovery options

Ensure metric/alert quality through review

Reduce work to get started

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

pandora-users

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

Wrote a lot of documentation

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

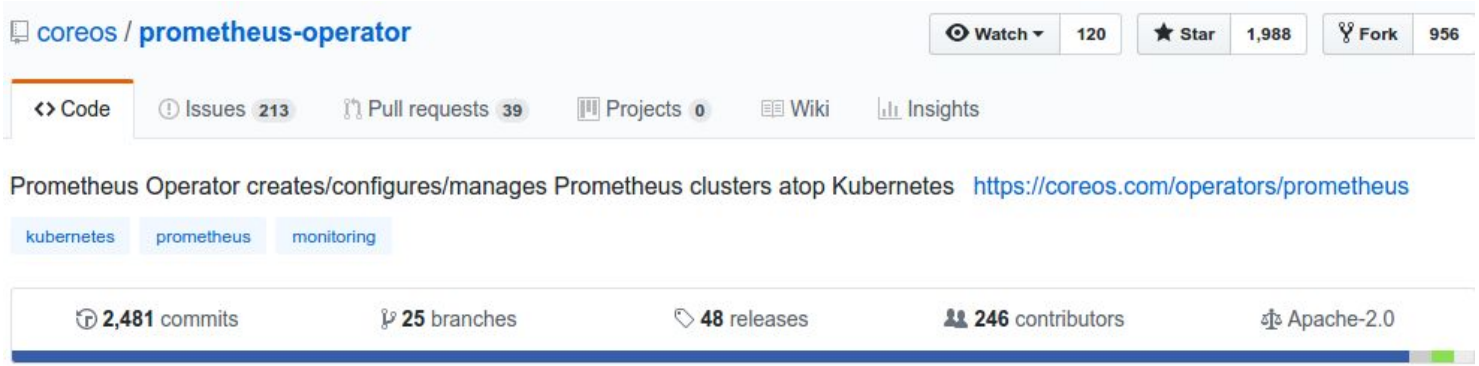
Pandora Architecture

Systems to Deprecate

Using Grafana with Pandora

Using node_exporter textfiles to export arbitrary metrics from a node

What I Would Do Today



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

You need people to do this

Operate

Centralize Knowledge

Centralize Configuration

Consult

Anyone can do this!

Manage Centralized Logging? Tracing? Exceptions?

thank u, next

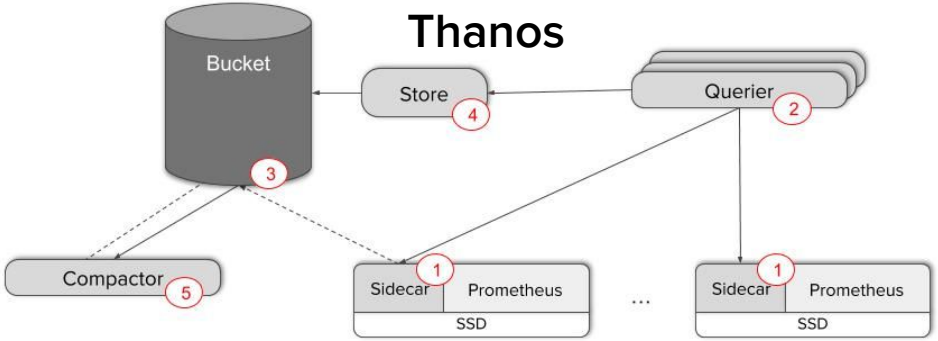
Some people will get it, love it, and get involved

Most will copypasta patterns

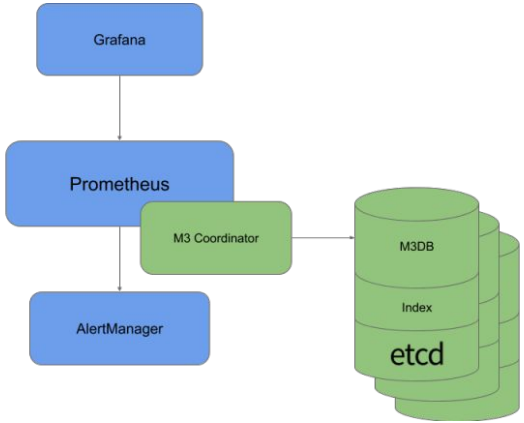
Some people will want **moar**
- 1s sampling
- infinite retention
- massive label cardinality

Friends of Prometheus - Thanos/M3/Cortex

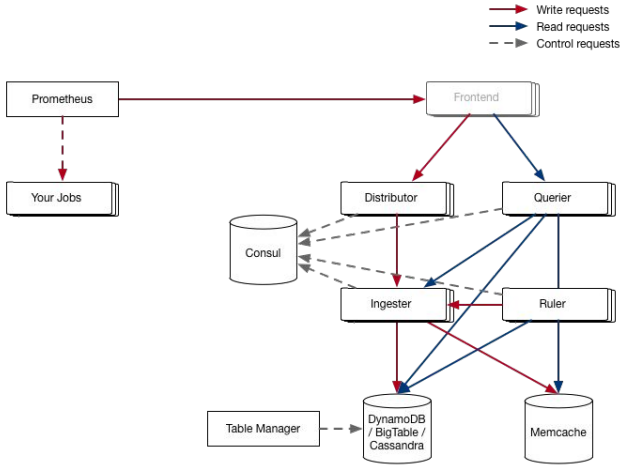
Thanos



Uber M3



Weaveworks Cortex



digitalocean.com

Questions

Why would I do all this when I can just **buy my Observability** from <vendor>?

YOU ARE WRONG about <something>!

What was the deal with those numbers you showed?

At what **specific point of cardinality** can you no longer do queries?

Why didn't you talk about <all the stuff I cut out>?

What is **DigitalOcean** anyway?

Why didn't you leave time for questions?