Humane On-Call

Alerting doesn't have to be painful

Bad alerting adds friction to operating production software

Bad alerting adds friction to operating production software

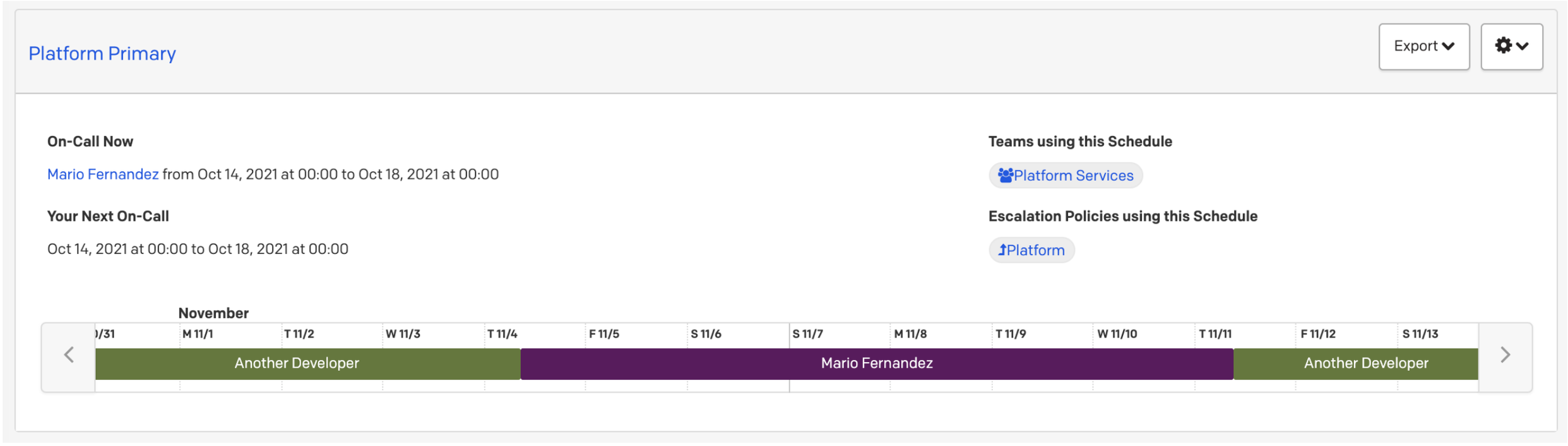
We can do better!

On-Call in a Nutshell

✅ Being *available* to handle an incident

❌ Constantly monitor production

As part of a *rotation*



Why do this to ourselves?

Reason 1

Users expect constant availability


Google


facebook


NETFLIX

Reason 2

You build it, you run it

stevesmith.tech/blog/you-build-it-you-run-it/

Alerting 101

Detect problems in production before your customers do

Monitoring + Notification

The Four Golden Signals

Latency

Traffic

Errors

Saturation

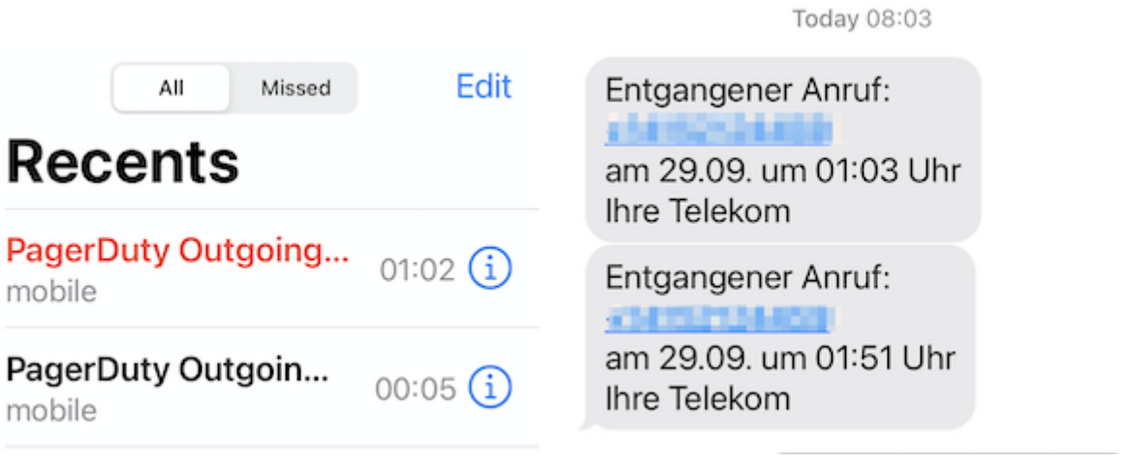
sre.google/sre-book/monitoring-distributed-systems/


555-3215

Constant Interruptions

User	Time on Call	High-Urgency Incidents	Acknowledged	Timeout Escalations	Manual Escalations	Reassignments	MTTA
[redacted]	168h 0m	2	100% (2)	50% (1)			57m
[redacted]	168h 0m	3	100% (3)			33% (1)	2m
[redacted]	168h 0m	1					
[redacted]	168h 0m	10	80% (8)	10% (1)			4m
[redacted]	123h 0m	1	100% (1)	100% (1)			13m
Mario Fernandez	96h 0m	20	40% (8)	30% (6)			5m
[redacted]	96h 0m	10	50% (5)	40% (4)			12m
[redacted]	72h 0m	17	71% (12)	6% (1)			2m
[redacted]	45h 0m						
[redacted]	45h 0m						
[redacted]		37		19% (7)			

Bad Night(s)


All
Missed
Edit
Recents
PagerDuty Outgoing...
mobile
01:02
PagerDuty Outgoin...
mobile
00:05
Today 08:03
Entgangener Anruf:
am 29.09. um 01:03 Uhr
Ihre Telekom
Entgangener Anruf:
am 29.09. um 01:51 Uhr
Ihre Telekom

Events Watchdog

[P1] [Recovered on {resource_name:get_/v1/get/min-layout}] service prod / Error Rate [Change%]

created_by:terraform env:prod monitor resource_name:get_/v1/get/min-layout

Error Rate is at 9.577% for the route[get_/v1/get/min-layout]

Show More

Fri Oct 08 2021 08:33:05 CEST (2 days and 3 hours ago)

Flakiness

[P1] [Triggered on {resource_name:get_/v1/get/min-layout}] service prod / Error Rate [Change%]

created_by:terraform env:prod monitor resource_name:get_/v1/get/min-layout

Error Rate is at 13.551% for the route[get_/v1/get/min-layout]

Show More

Fri Oct 08 2021 08:31:05 CEST (2 days and 3 hours ago)

Alerting Dysfunctions

Mixed abstraction levels

Lack of automation

Noisy alerts

Inadequate tools

Mismatched tuning

Mixed Abstraction Levels

What are you monitoring?

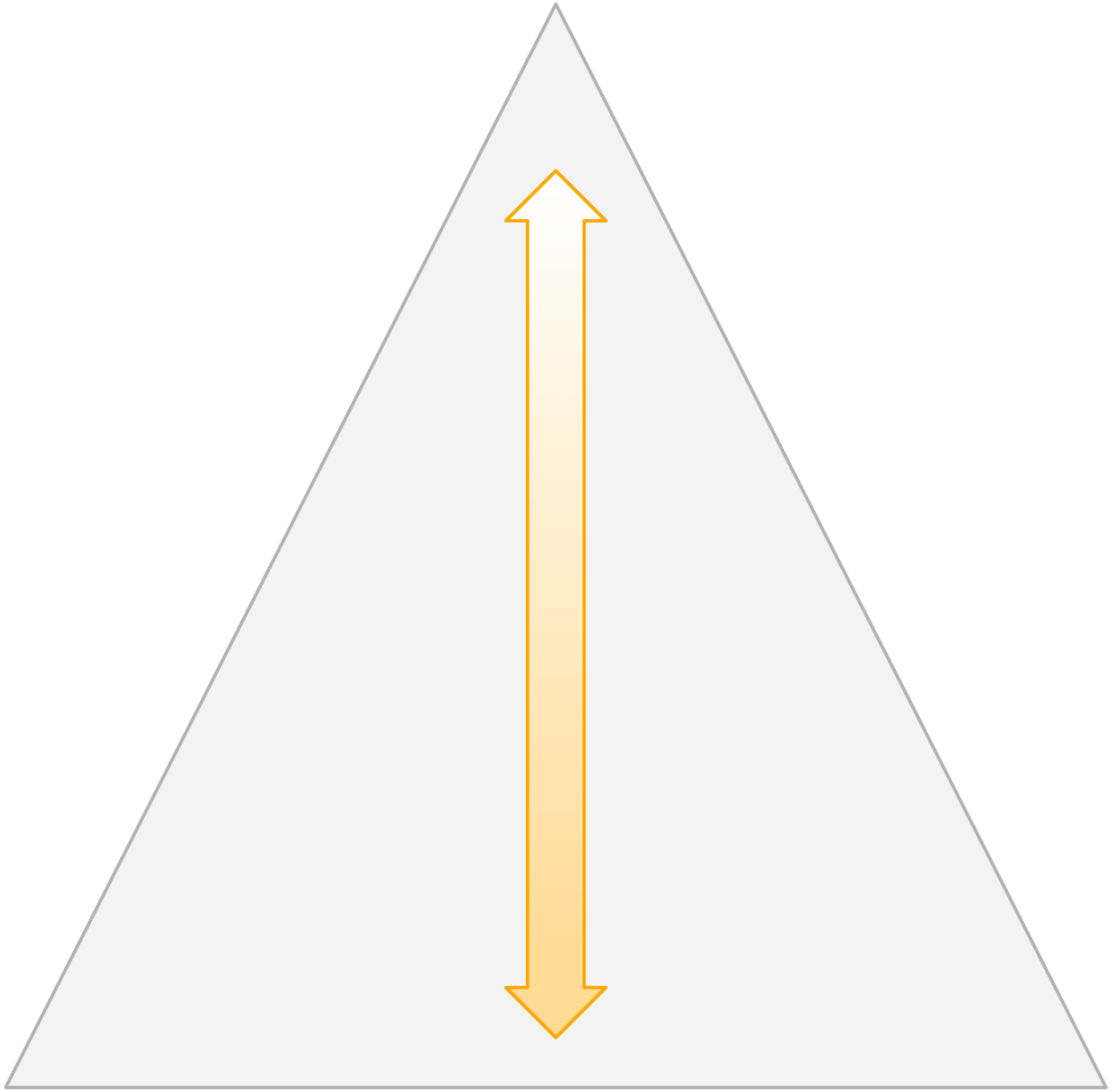
What are you monitoring?

High-Level: Close to the user

What are you monitoring?

High-Level: Close to the user

Low-Level: Close to the infrastructure



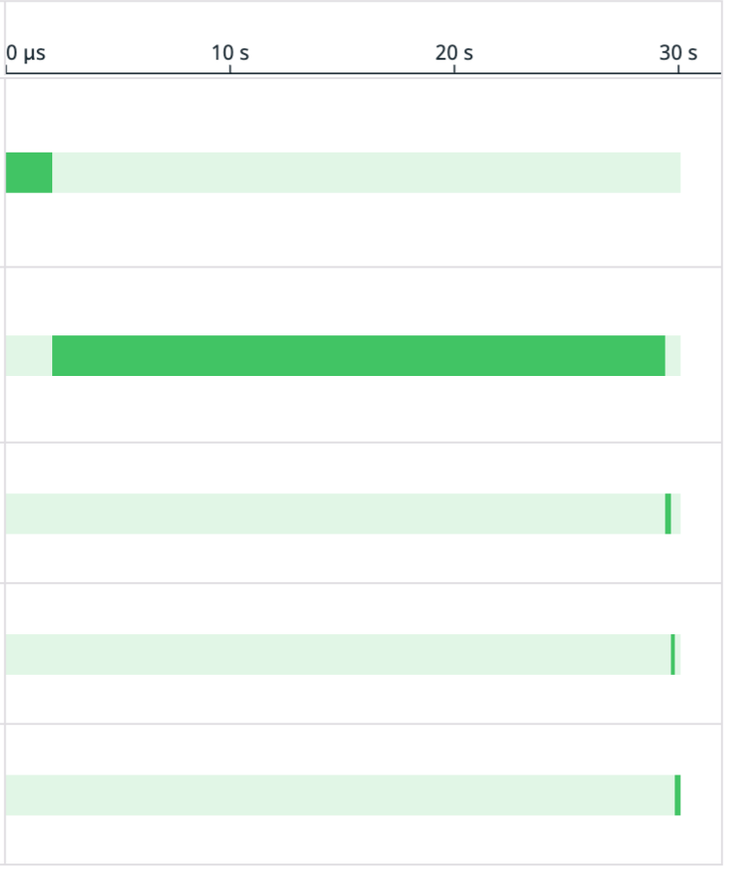

Synthetic
Monitoring
Real User
Monitoring
Application Performance
Monitoring
Infrastructure
Monitoring

OK Last ran 1 min ago (Oct 18, 2021, 5:24 pm)

Navigation Available

STATUS	STEPS	DURATION	LOCATION	DEVICE	BROWSER	TIME RAN	RUN TYPE
OK	5 / 5	29s	Frankfurt (AWS)	Laptop Large	edge 95.0.1020.9	3 mins ago (Oct 18, 2021, 5:19 pm)	Scheduled

	SCREENSHOT	ACTION	DURATION
0 ✔		Navigate to start URL Started at about:blank	2 s
1 ✔		Login > Expand 10 steps	26.2 s
2 ✔		✔ Navigation Panel present	207 ms
3 ✔		✔= User displayed Should not be empty	194 ms
4 ✔		✔ "Help & Support" is present	211 ms



OK Monitor status since 19 weeks ago (6 Jun, 18:30:51 CEST)

Mute | Declare Incident

CPU load is very high on {{host.name}}

Properties

Metric Monitor
ID: 11872480

TAGS

PRIORITY Not Defined

QUERY `avg(last_5m):100 - avg:system.cpu.idle{w[redacted]} by {host} > 60`

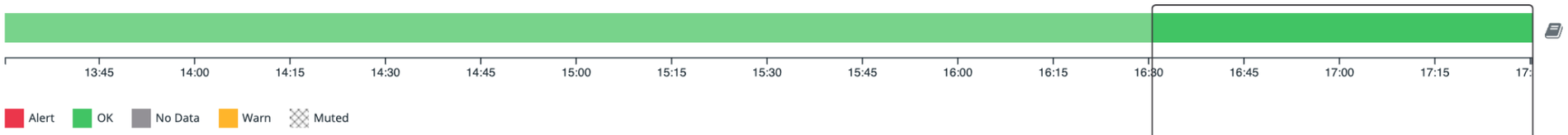
Filter monitor groups and their events

Alert 0 | Warn 0 | No Data 0 | OK 1 | **1 of 1 groups**

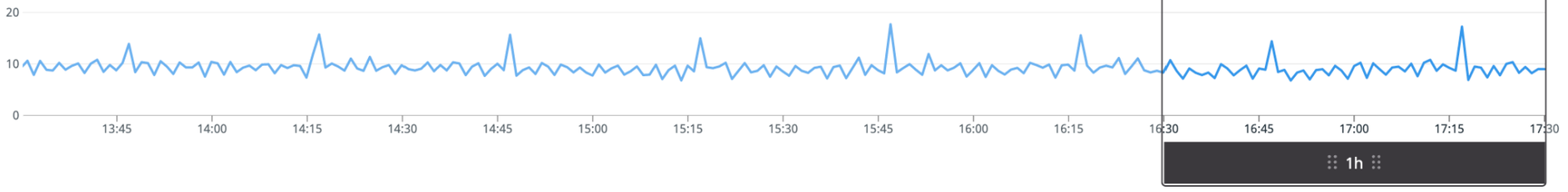
Status & History

GROUP STATUS

NAME	VALUE
host:[redacted]..	9.22



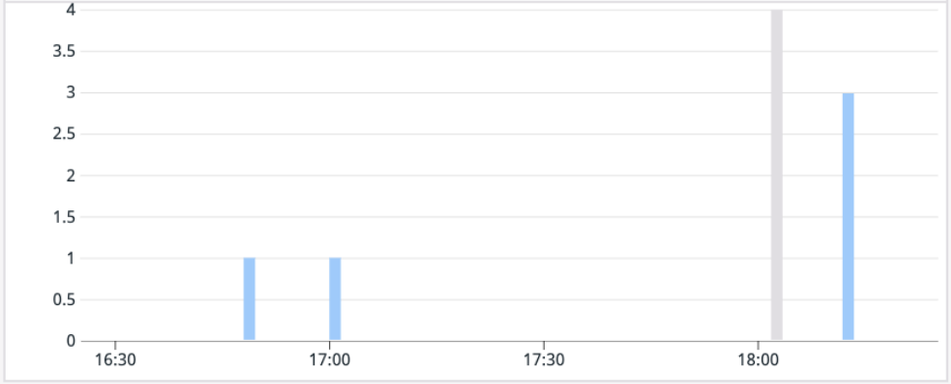
HISTORY



Choose the alert based on what you want to monitor

User Impact

RUM: User visible errors



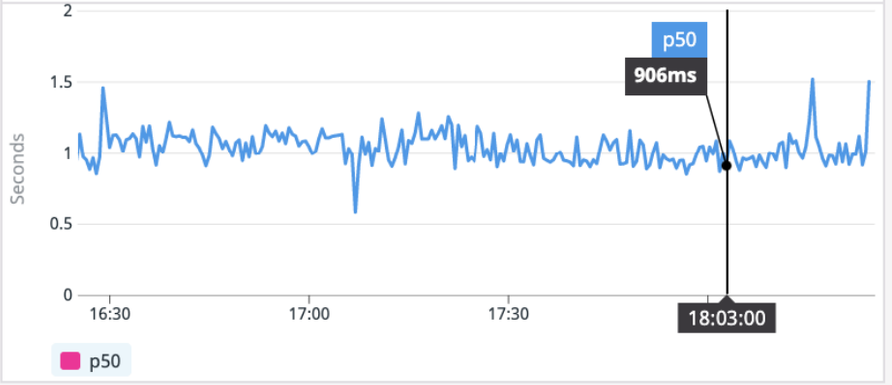
% of Errors w/o Cache

51.6

Total Errors

31

Duration



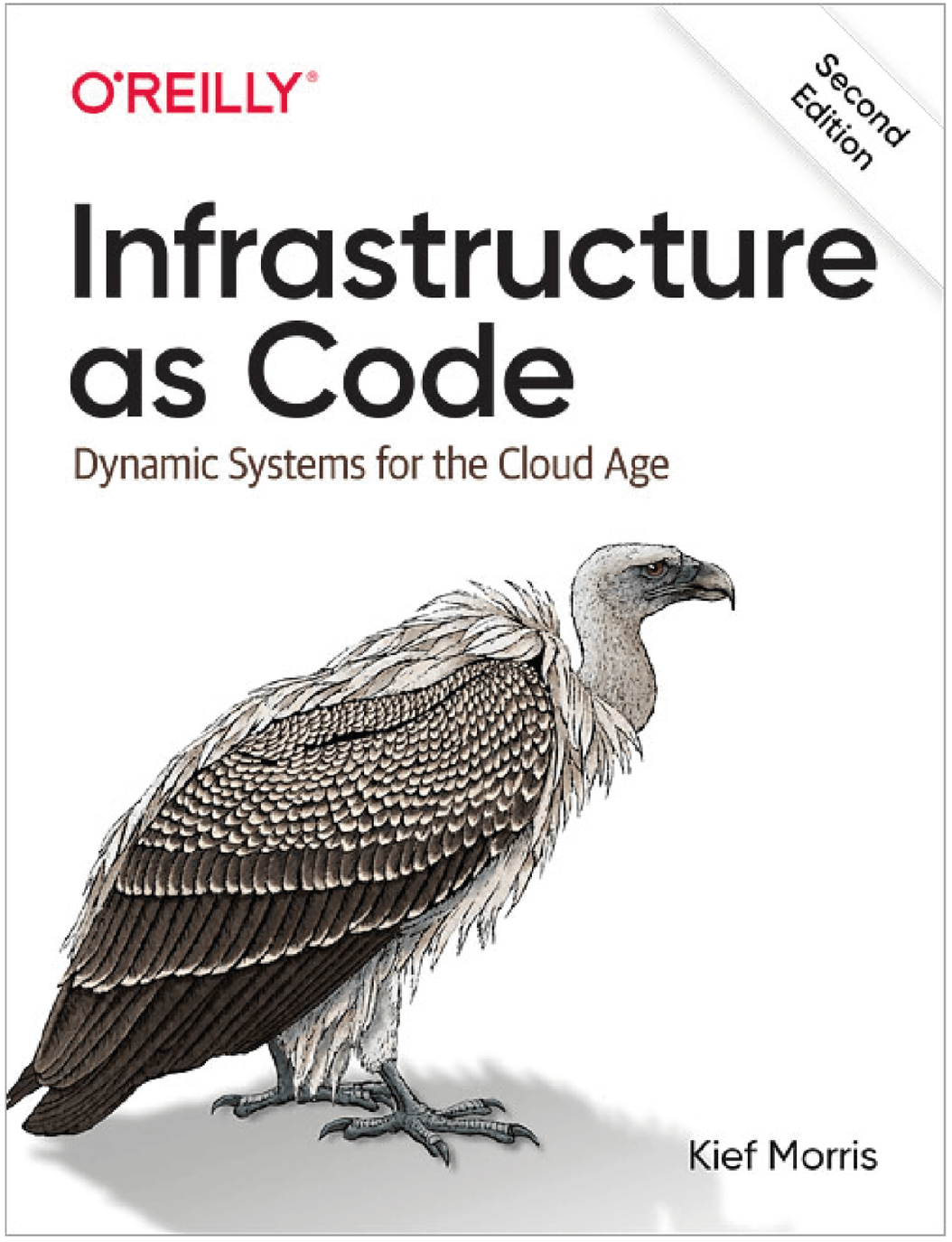
Learning 1️⃣

Use the right alert

Lack of Automation

Manual work is not reproducible

Automation is crucial in alerting


O'REILLY
Second Edition
Infrastructure as Code
Dynamic Systems for the Cloud Age
Kief Morris

Reduce maintenance

Reduce maintenance

Spread best practices

HashiCorp

Terraform

https://registry.terraform.io/browse/providers

```
resource "datadog_monitor" "monitor" {
  count = try(var.enabled, false) == true ? 1 : 0

  name               = var.name
  type               = var.type
  query              = var.query
  message            = var.message

  monitor_thresholds {
    warning = lookup(var.monitor_thresholds, "warning", null)
    ok = lookup(var.monitor_thresholds, "ok", null)
    critical = lookup(var.monitor_thresholds, "critical", null)
  }

  priority            = var.priority
}
```

```
resource "pagerduty_team" "team" {
  name = "Service Team"
}

resource "pagerduty_schedule" "schedule" {
  name      = "Weekly Engineering Rotation"
  time_zone = "Europe/Berlin"

  layer {
    name                         = "Night Shift"
    start                        = "2015-11-06T20:00:00-05:00"
    rotation_virtual_start       = "2015-11-06T20:00:00-05:00"
    rotation_turn_length_seconds = 86400
    users                        = [pagerduty_user.team.id]
  }

  teams = [pagerduty_team.example.id]
}
```

What about more complex stuff?

Multiwindow, Multi-Burn-Rate Alerts


Error Rate SLO
Short Window
Fast Burn Rate
Long Window
Short Window
Medium Burn Rate
Long Window
Short Window
Slow Burn Rate
Long Window

```
module "error_rate_slo_burn_rate_long_window" {
  source  = "../../monitor"
  enabled = var.enabled

  name    = <<-EOT
  ${var.service_name} ${var.environment} -
  Error Rate SLO [Long Window, ${var.burn_rate}x Burn rate]"
EOT

  query = <<-EOT
sum(${local.period}):(
  sum:trace.${local.metric}.request.errors{env:prod,service:${var.service_name}}
    .rollup(sum, ${var.window_in_seconds}) /
  sum:trace.${local.metric}.request.hits{env:prod,service:${var.service_name}}
    .rollup(sum, ${var.window_in_seconds})
) * 100 > ${local.threshold}
EOT

  monitor_thresholds = {
    critical = local.threshold
  }
}
```

```
module "error_rate_slo_quick_burn" {
  source  = "../error_rate_slo_burn_rate"
  enabled = var.enabled

  service_name = var.service_name
  environment  = var.environment

  slo                = var.slo
  burn_rate          = 14.4
  window_in_seconds = 3600

  language = var.language
}
```

```
module "error_rate_slo" {
  source  = "../../monitor"
  enabled = var.enabled

  name    = "${var.service_name} - Error Rate SLO [${var.slo}%]"
  type = "composite"

  query = <<-EOT
(module.error_rate_slo_quick_burn.long_window_id &&
    module.error_rate_slo_quick_burn.short_window_id)
||
(module.error_rate_slo_slow_burn.long_window_id &&
    module.error_rate_slo_slow_burn.short_window_id)
EOT

  message = templatefile("${path.module}/message.md", {
    service     = var.service_name
    environment = var.environment
    notify      = local.notify
  })
}
```

Scale alerts inside a team

Scale alerts across teams

Learning 2️⃣

Leverage automation at scale

Noisy Alerts

When the signal is drowned by the noise

Alert fatigue leads to ignored alerts

atlassian.com/incident-management/on-call/alert-fatigue

A lot of the noise is *accidental*

OK Monitor status since 1 week and 1 day ago (19 Oct, 10:35:10 CEST)

Service [redacted] has low pod count

Integration Monitor

TAGS env:prod

PRIORITY P1 (Critical)

QUERY

```
avg(last_5m):sum:kubernetes_state.deployment.replicas_available{env:prod} < 1
```

Status & History

GROUP STATUS

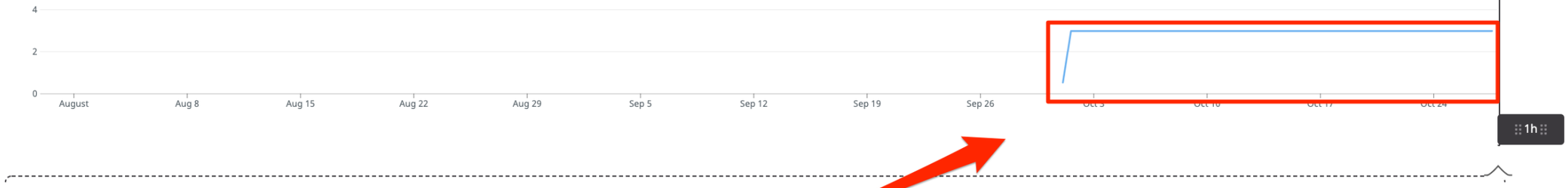
NAME **VALUE**

* (Entire Infrastructure) 3

August Aug 8 Aug 15 Aug 22 Sep 19 Oct 24

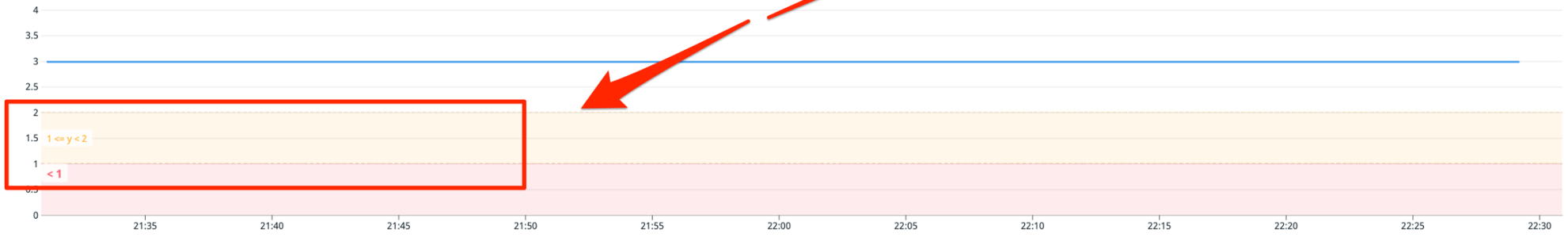
Alert OK No Data Warn Muted

HISTORY



Isn't this Kubernetes job?

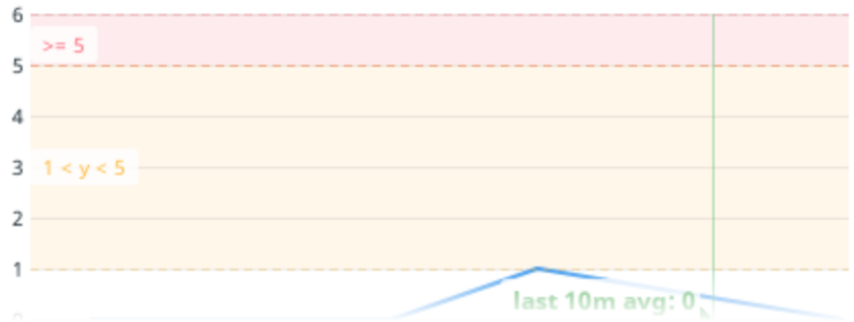
1h **EVALUATION GRAPH** Each point in the graph reflects an evaluation window of **5min**



Events & Watchdog

Events Watchdog

[P4] [Recovered] eDocs Signing Failed ?



What should I do?

Tue Oct 19 2021 12:30:50 CEST (9 hours and 12 minutes ago)

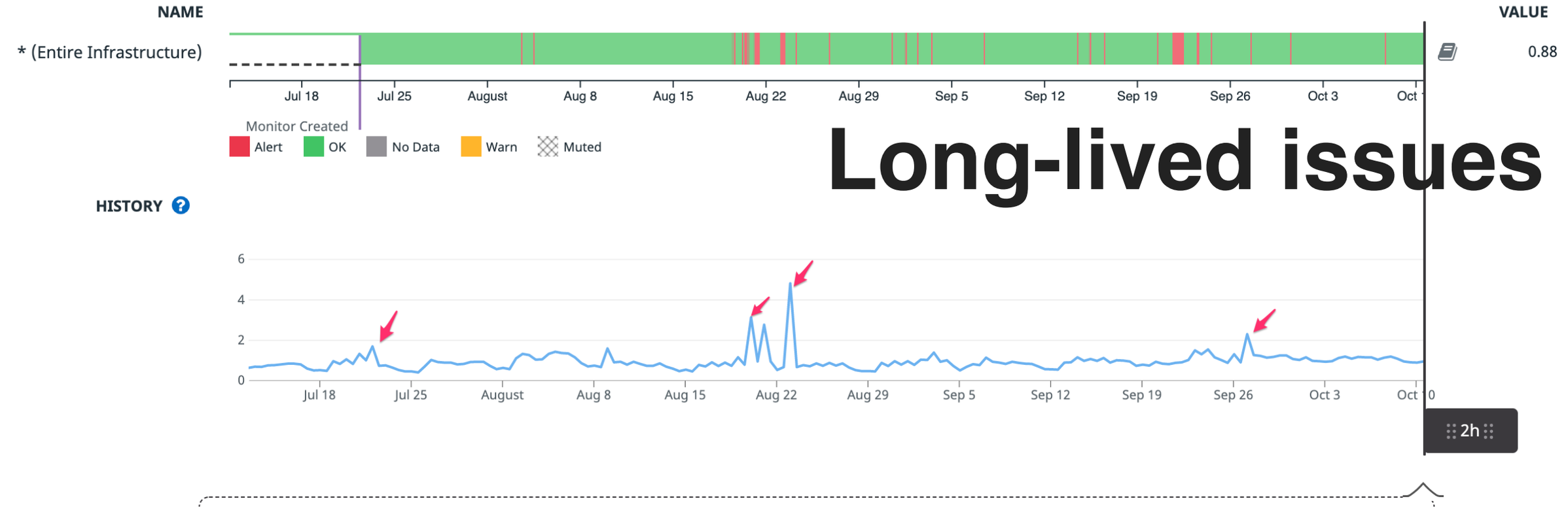
Warnings are evil

Make them failures or ignore them

When that alert wakes me up I just snooze it and keep sleeping

OK Monitor status since 22 hours and 2 minutes ago (9 Oct, 16:13:59 CEST)

Status & History



Delete things you don't need!

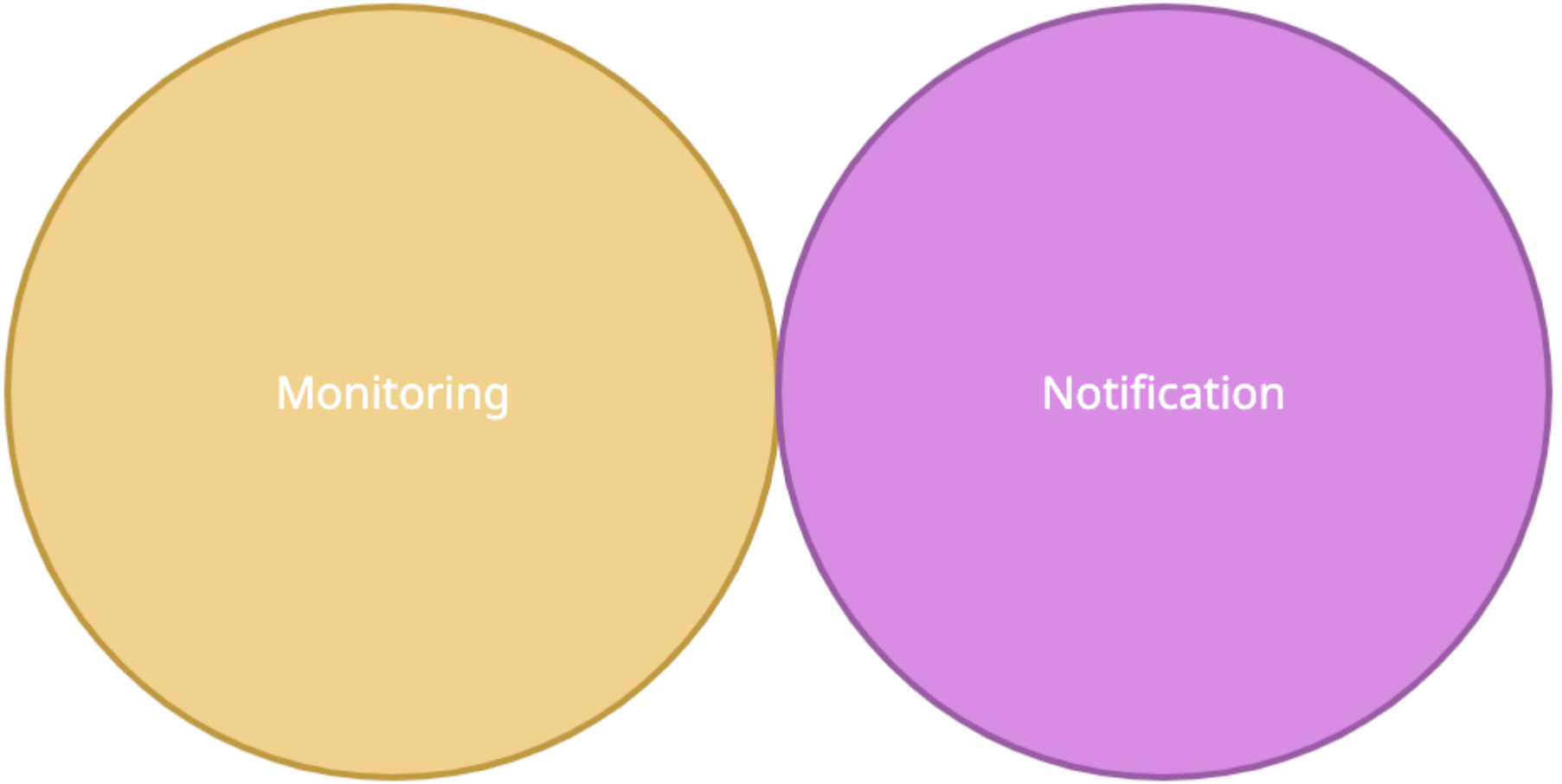
Like, seriously

Learning 3️⃣

Reduce noise ruthlessly

Inadequate Tools

Crappy tools are extra infuriating at 3 in the morning


Monitoring
Notification

The market is full of high-quality tools


DATADOG

PagerDuty

Do you take good 🛠️ for granted?

I've learned not to 😩

Project at a German client

Ticketing system without features such as

Auto closing

Auto closing

Auto grouping

Auto closing

Auto grouping

Conditional routing

Auto closing

Auto grouping

Conditional routing

Learning 4️⃣

Adopt tools that support you

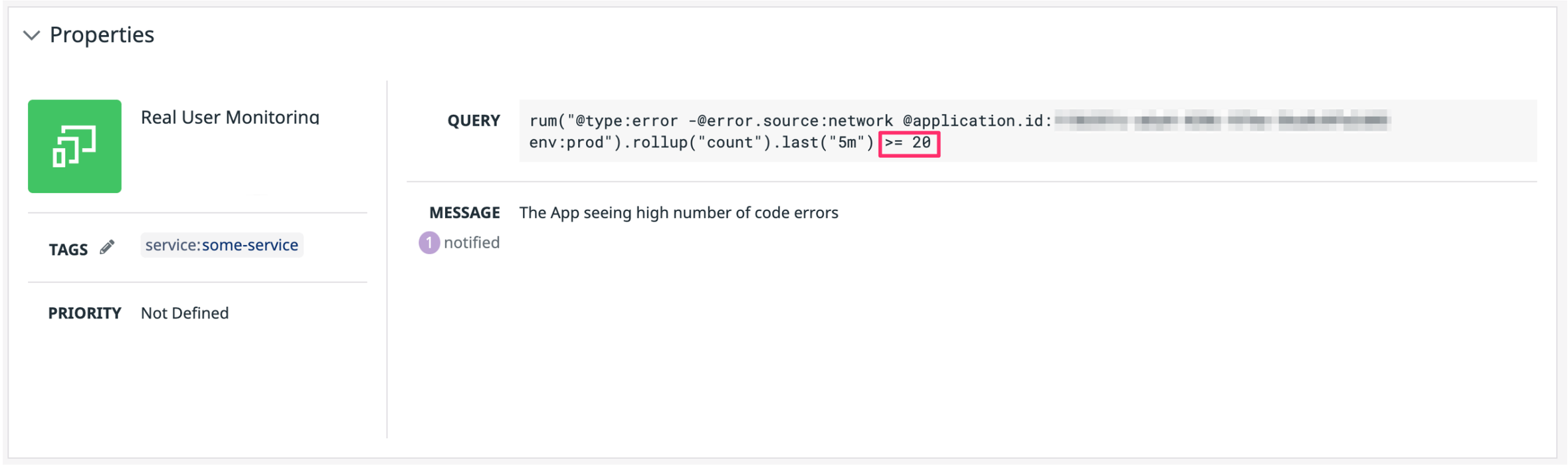
Mismatched Tuning

Minimize

false *positives*

false *negatives*

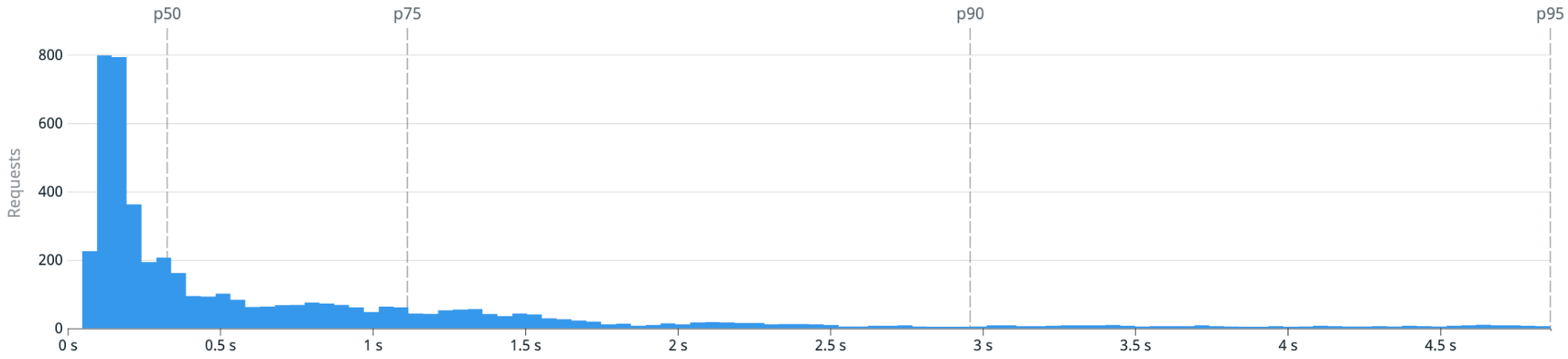
Use ratios for your metrics



Get enough data

supplier_media

Latency Distribution



5.07k requests

max	31.85 min
p99	16,547 ms
p95	4,862 ms
p90	2,960 ms
p75	1,115 ms
p50	327 ms

Endpoints 61 Endpoints

NAME	↓ REQUESTS	TOTAL TIME	P50 LATENCY	P90 LATENCY	P99 LATENCY	ERRORS	ERROR RATE	REQUESTS/S	API TE...
☆ POST product_media/imag...	624	17 min 41 s	1250 ms	3150 ms	6795 ms	0	0%	0.2 req/s	
☆ POST catalog/media/rich_...	495	5 min 38 s	690 ms	1187 ms	1634 ms	0	0%	0.1 req/s	

Multiple small alerts

PRIORITY	STATUS ↑	MUTED LEFT	NAME	
P1	OK		navigation-service prod / P1 - Error Rate SLO [99,9%]	2
P1	OK		navigation-service prod - Error Rate SLO [Short Window, 6x Burn rate]	
P1	OK		navigation-service prod - Error Rate SLO [Long Window, 14.4x Burn rate]	
P1	OK		navigation-service prod - Error Rate SLO [Short Window, 14.4x Burn rate]	
P1	OK		navigation-service prod - Error Rate SLO [Long Window, 6x Burn rate]	

Reliable alerts are hard!

hceris.com/monitoring-alerts-that-dont-suck/

Learning 5️⃣

Tune alerts often

Dysfunctions and Learnings Side by Side

Mixed abstraction levels

Lack of automation

Noisy alerts

Inadequate tools

Mismatched tuning

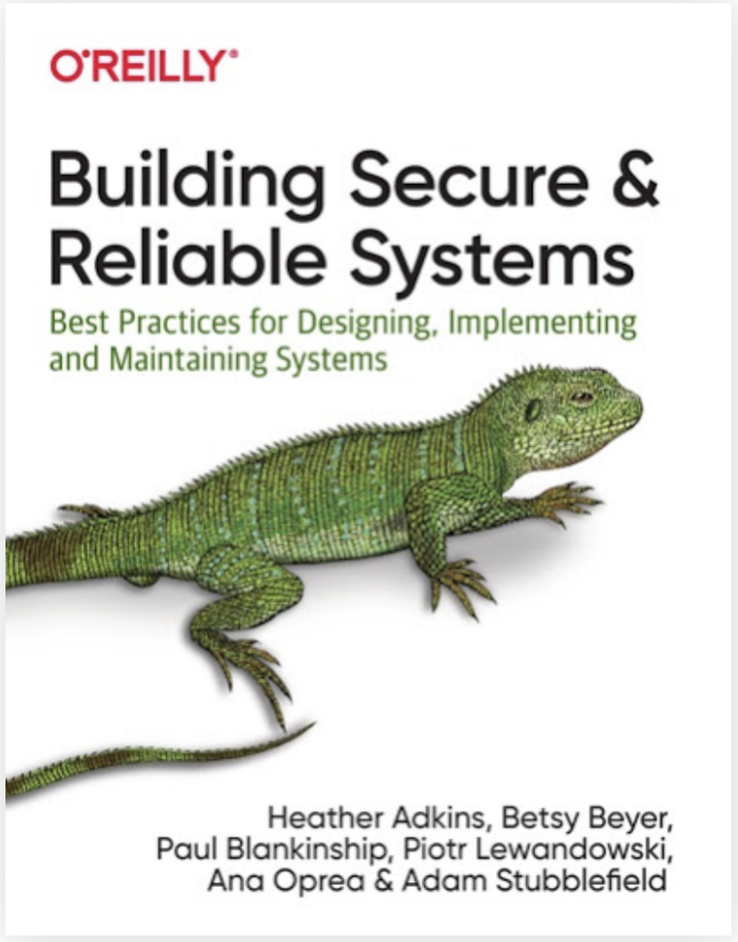
Use the right alert

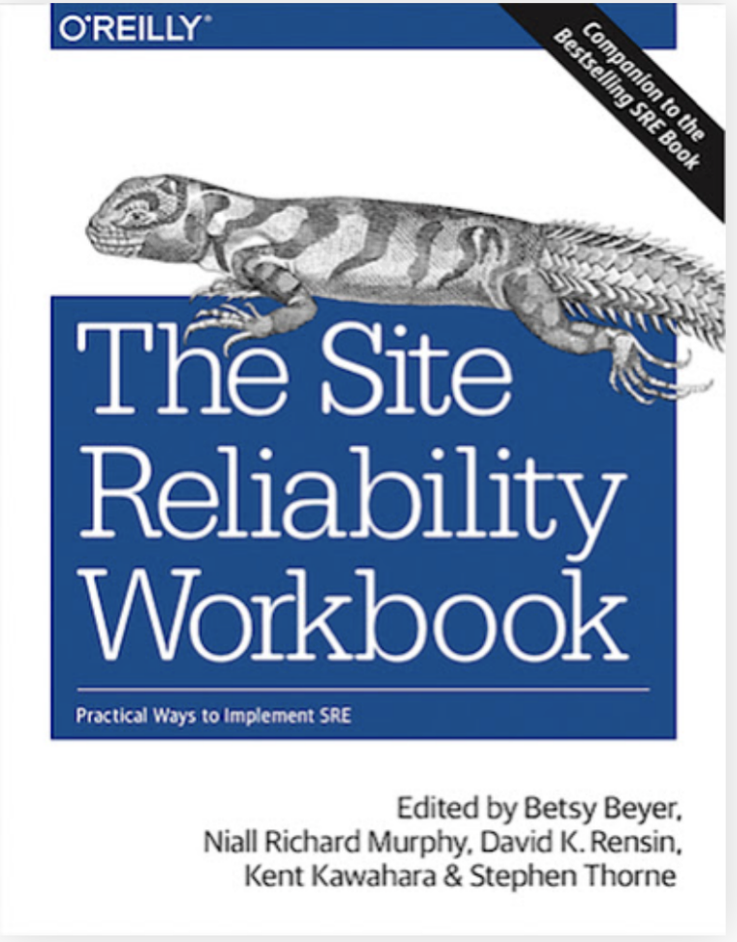
Leverage automation at scale

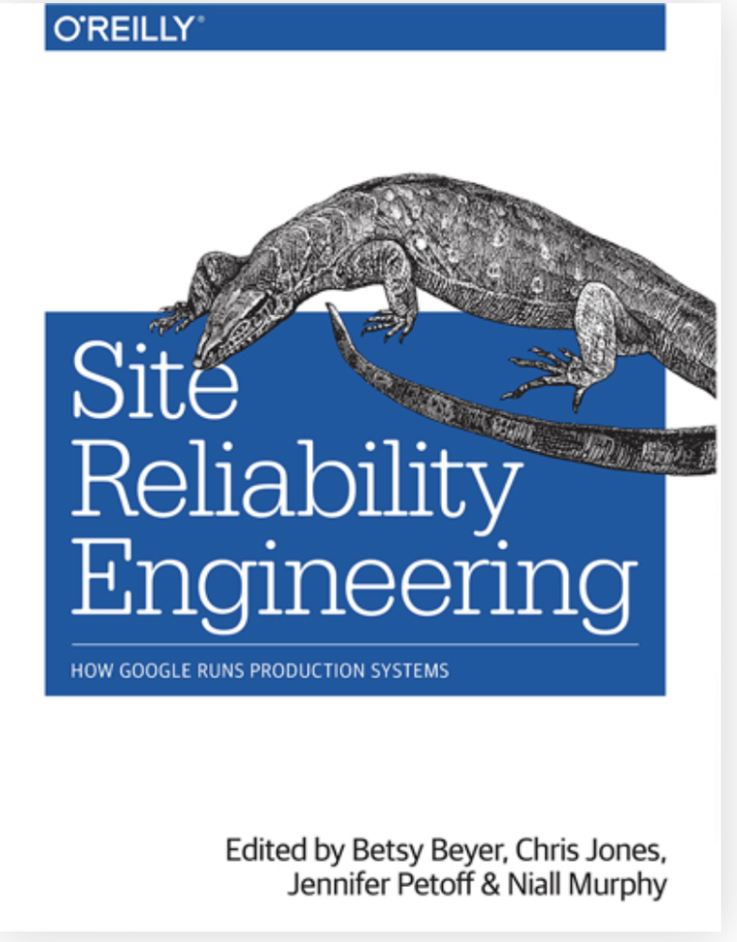
Reduce noise ruthlessly

Adopt tools that support you

Tune alerts often


O'REILLY
Building Secure & Reliable Systems
Best Practices for Designing, Implementing and Maintaining Systems
Heather Adkins, Betsy Beyer, Paul Blankinship, Piotr Lewandowski, Ana Oprea & Adam Stubblefield


O'REILLY
Companion to the Bestselling SRE Book
The Site Reliability Workbook
Practical Ways to Implement SRE
Edited by Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara & Stephen Thorne


O'REILLY
Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS
Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Murphy

Mario Fernandez

Staff Engineer

Wayfair