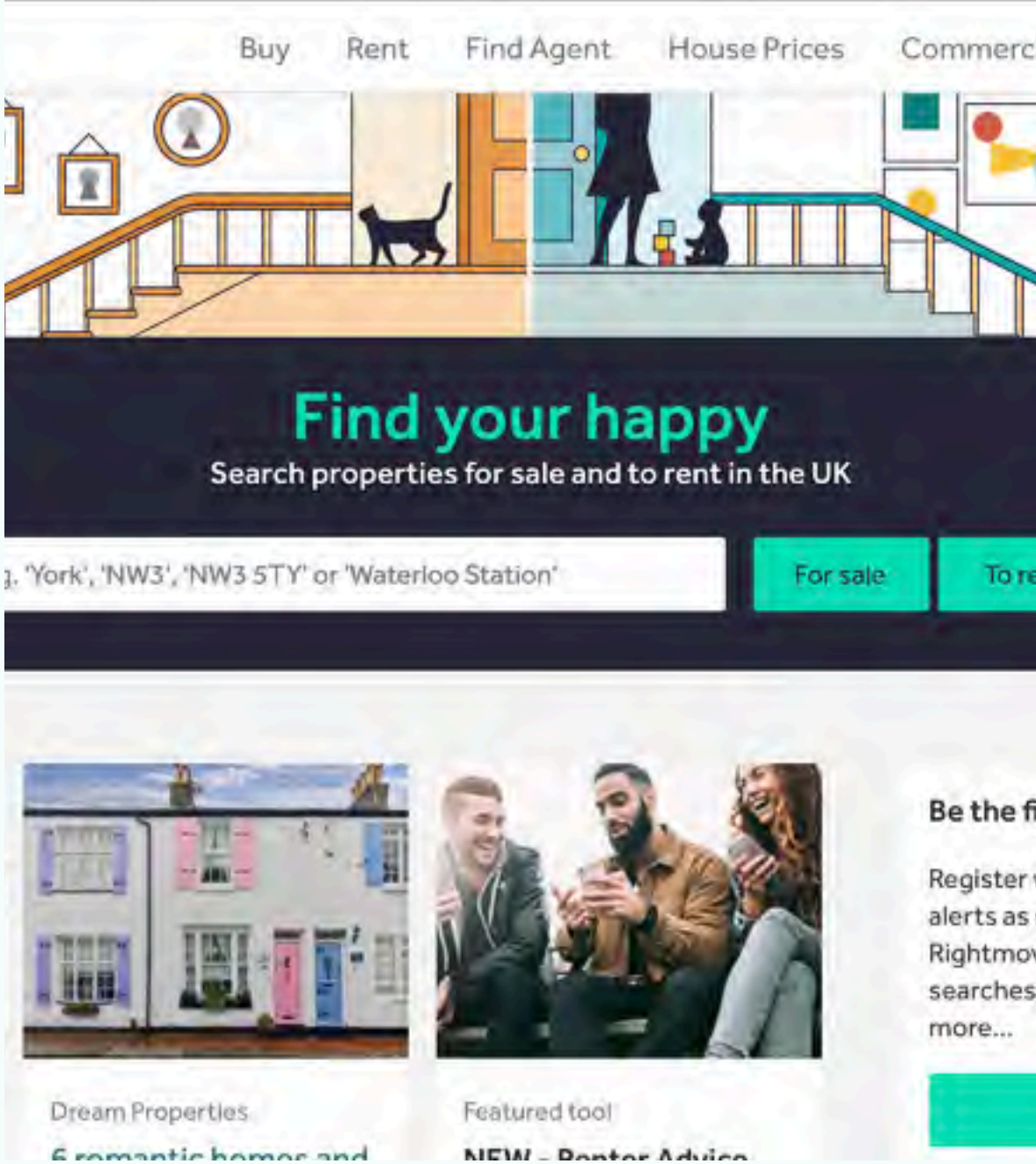
Practical Advice For Monitoring Microservices

Adrian McMichael
@trev_boxmonster
adrian.mcmichael@rightmove.co.uk

rightmove
find your happy

Introduction

- Adrian McMichael
- Lead Application Architect at Rightmove.
- UK’s Biggest Property Portal
  - Established in 2000.
  - Around 60M requests a day.
  - Around 1.2 Million Properties on Site.
  - 90% of all estate agent listings in the country.
  - Around 7.3 Billion Log messages a day.



What I'll cover

- A short Rightmove history lesson
- Best Practices
  - Observable Events
- How We Monitor
  - Logging Pipeline
  - Alerting
- Results

A Bit of History

Before Microservices

- Before 2014 if an application failed
  - ssh onto application server
  - cd to the correct directory
  - Hope the logs contain the data you need in the right format
  - Begin the awk/sed wizardry
  - if answer present:
    - Repeat for each application instance affected
  - else
    - Increase logging and wait for reoccurrence

```
gzcat access_log.h2-
api05.20130506 | awk -
F'"' ' $7 > 3000000
{print $2}' | grep '\?'
|sed 's/.* \(\/.*\?\).*/
\1/g' | sed 's/\(\/
api\/.*\/sync\)/\/api\/
sync/g' | sort | uniq -c
| sort
```

- An ancient incantation for grouping slow pages

Enter Project Odin

- Investigated a new search engine
- Decided to replace our core flow with microservices
  - Gives us more flexibility
  - Improve ownership
  - Improve maintainability
- Given the time it could take to look at issues we needed better tools.

What We Wanted to Achieve

- Take advantage of the wider surface area of microservices to pinpoint issues better
- Have a self service approach to logging and investigating how services are behaving
- Support microservice ownership.
- Provide access to data about our systems in a way that is friendlier to non-developers.

Observable Events

Obligatory definition time!

"'Monitoring' refers to repeatedly checking a system and its outputs to make sure they are within known-good ranges..."

Observability ... is about being able to understand the inner workings of your software and systems by asking questions and observing the answers on the outside…"

- Charity Majors, @mipsytipsy, 2018

https://bit.ly/2Ovf2ji

Bad Event Logging

- Is an afterthought.
- Is autogenerated or relies purely on 3rd Party agents and plugins.
- Is anaemic and lacks context
- Uses a human readable format which makes ingestion hard
- Uses a message field that contains all the information.
- Describes the system as we expect it to work!

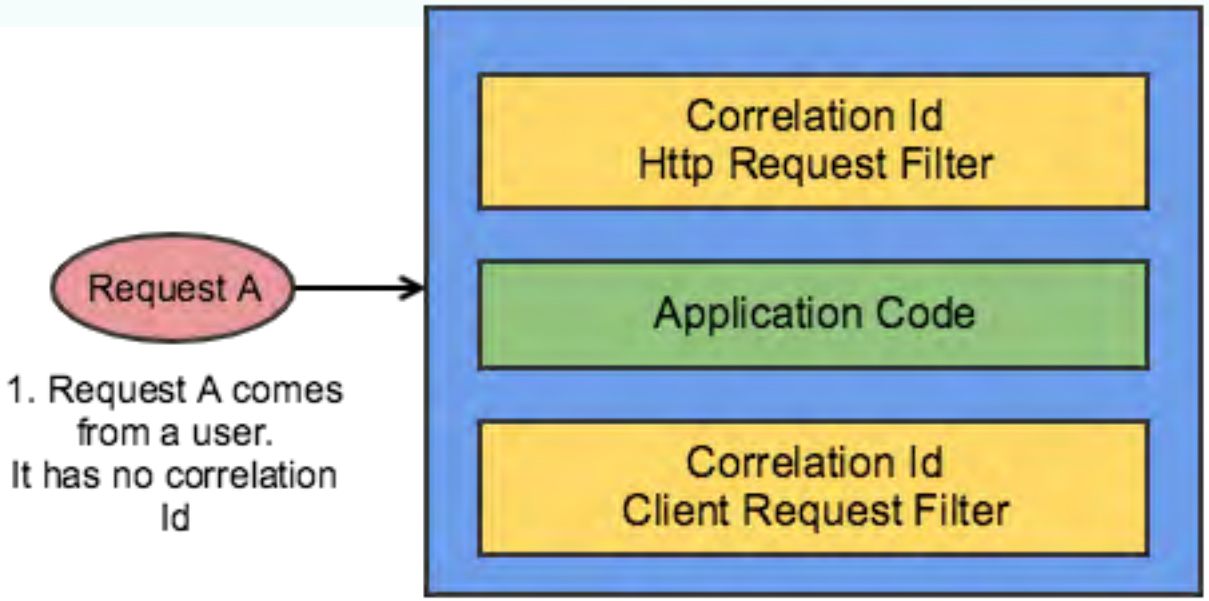
Good Event Logging

- Is a stream of events that can be followed across boundaries to describe how a system behaves.
- Shares a common specification
- Is designed to allow us to ask questions of a system.
- Has messages that help discoverability but are not the source of contextual data.
- Is testable
- Evolves with the system
- Accepts failure!

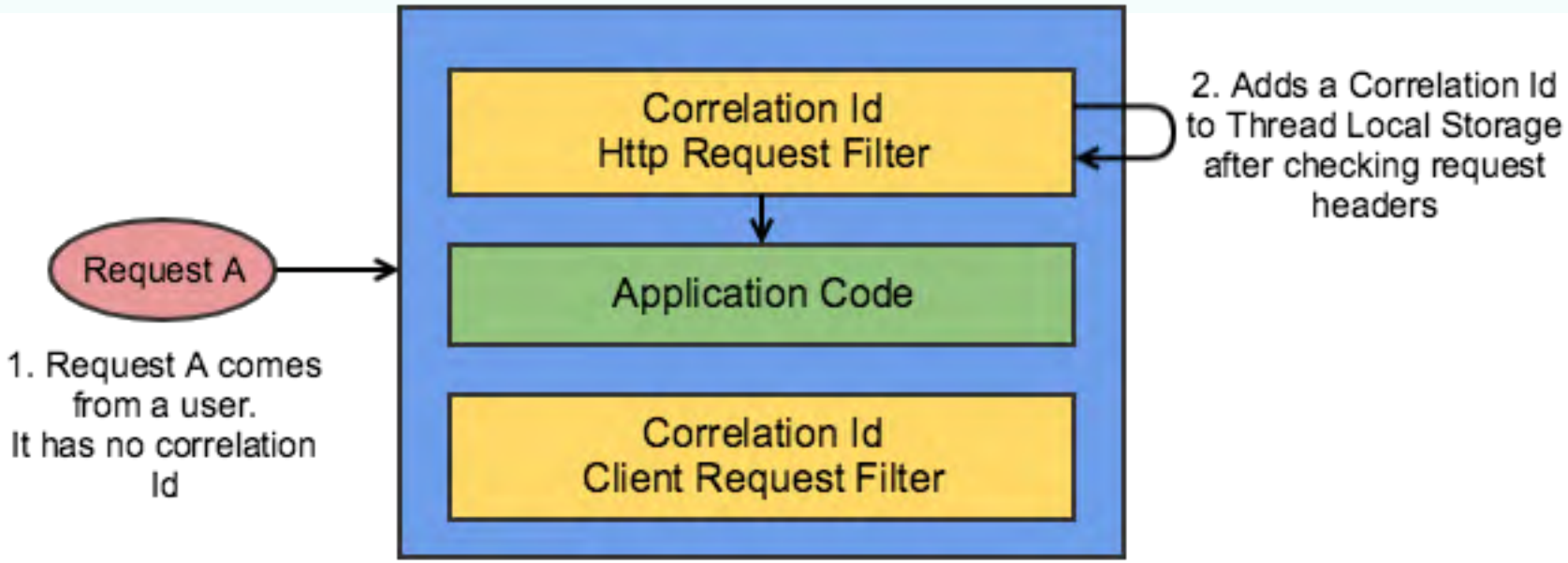
How we structure logging

- A bit about our event structure
  - Transactions are correlatable across application boundaries
  - Log data is machine readable and in most cases JSON based.
  - Supports thread local dimensional key-values pairs, timing and tags.
  - Supports passing of contextual data across application boundaries to keep APIs clean.
- We use a custom made Log4j2 Java Library
  - See Also:
    - Open Tracing - https://opentracing.io/
    - Open Census - https://opencensus.io/
    - Brave - https://github.com/openzipkin/brave
    - Zipkin - https://zipkin.io/
    - Honeycomb - https://www.honeycomb.io/

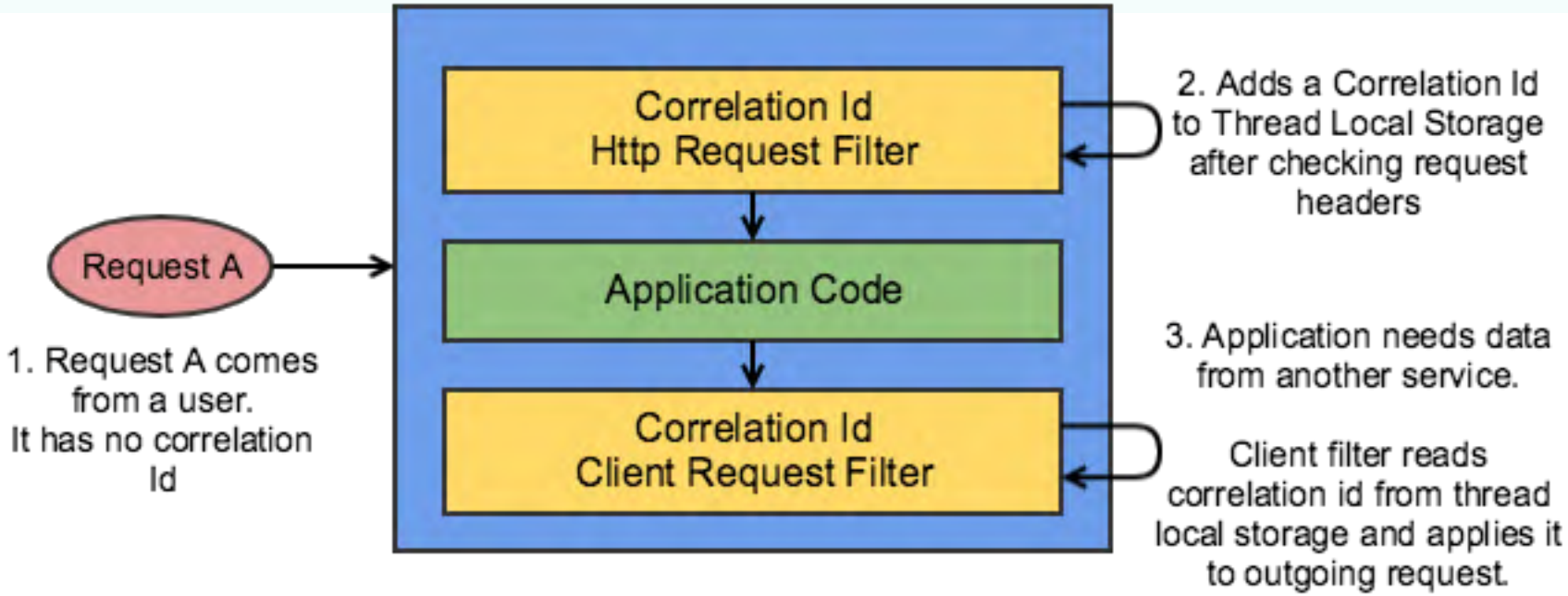
Correlation Ids



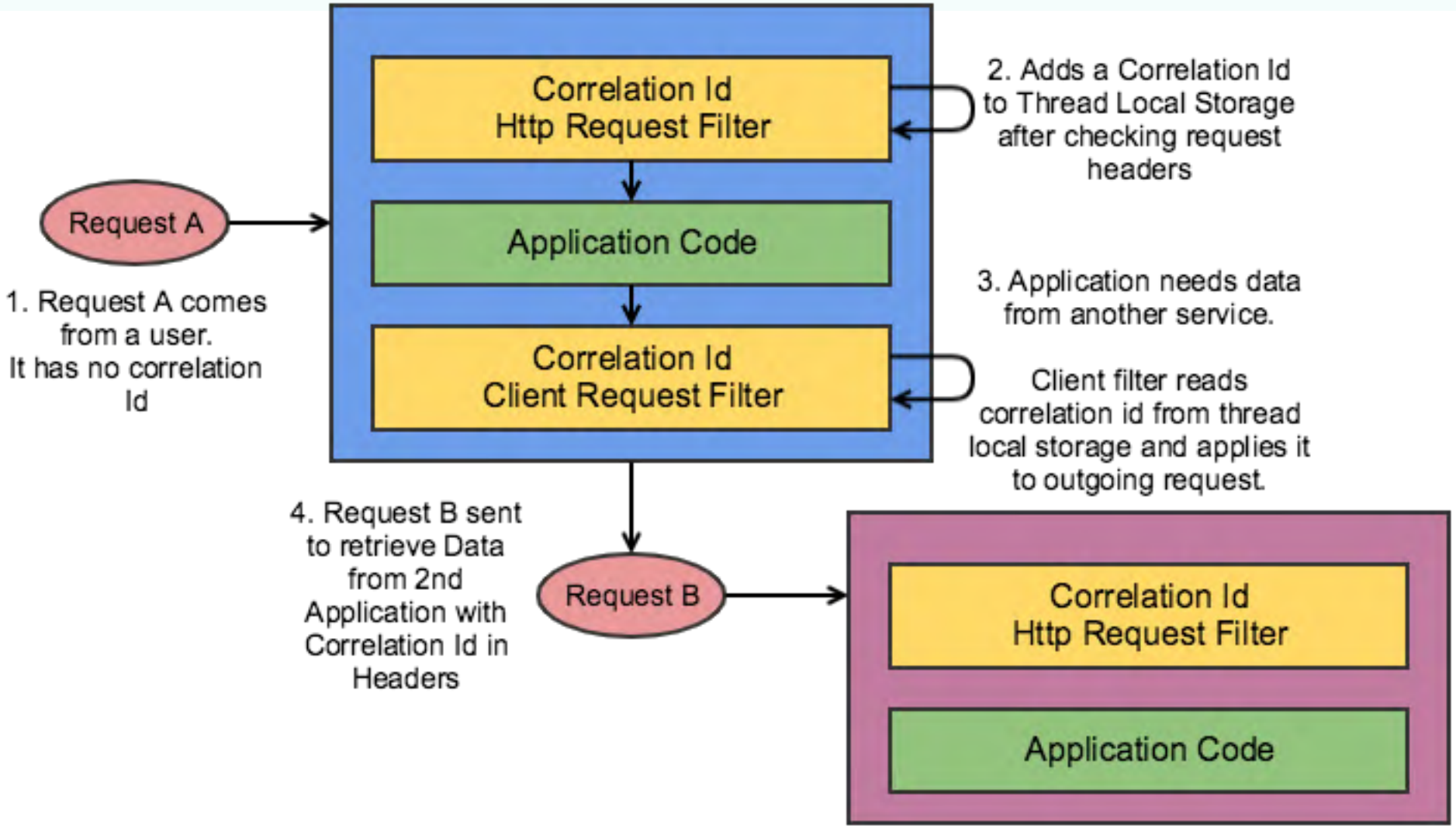
Correlation Ids



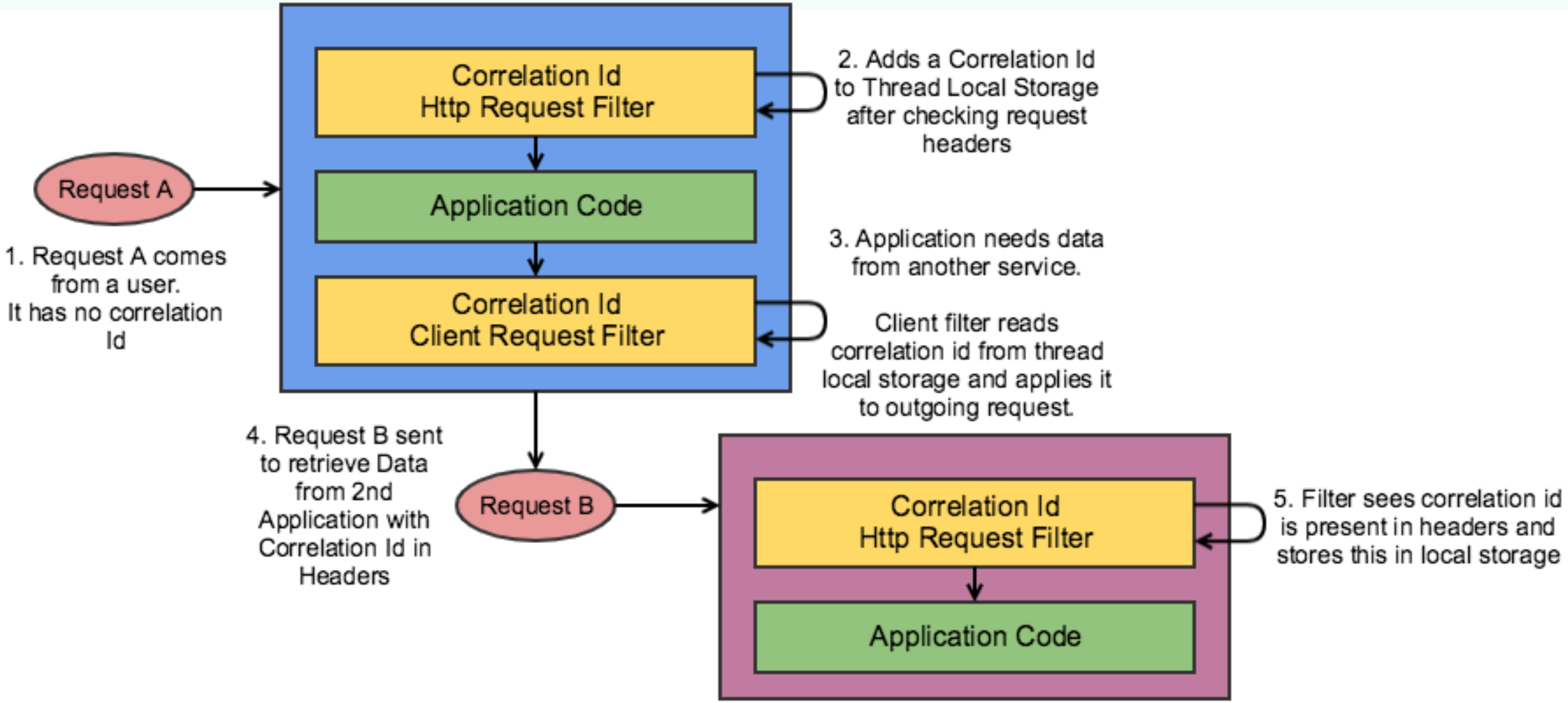
Correlation Ids



Correlation Ids



Correlation Ids



Event Ids

- Identify a discrete event within a transaction
  - Usually a numeric id/enumeration which allows to query across families
  - Are unique per activity not per transaction like correlation ids.
  - Can be used to query across event types when exploring data.
  - Can be technical
    - Like an Application Starting
  - Can be domain driven
    - Adding an item to a basket, saving a property
  - https://oreil.ly/2yi9KO9 – Matthew Skelton, Velocity London, 2017

Common Metadata

- Request parameters
  - Search filters
  - User ids
  - User-agent details
- Application instance details
- Timings
  - Broken down for multiple calls
  - Share a common timing denomination
- Can pass important information for logging purposes via request headers to avoid polluting APIs
  - E.g. Customer Ids, Human readable Search terms, user-agent

Anemic Events vs Fat Events

```
{
  “level”: “INFO”
  “message”: “property search”,
  “duration”: 100
}
```

```
{
  “message”: “sales property search complete”,
  “eventId”: 20000,
  “correlationId”: “a4229...
  “duration”: 100,
  “containerId”: “abcd098098”,
  “metadata”: {
    “locationId”: 12345
    “locationType”: “region”,
    “minBeds”: 2,
    “maxPrice”: 1000000,
    “keywords”: [“sea view”]
    “containerLabels”: {
      “language”: “java”
      ...
    }
    ...
  }
}
```

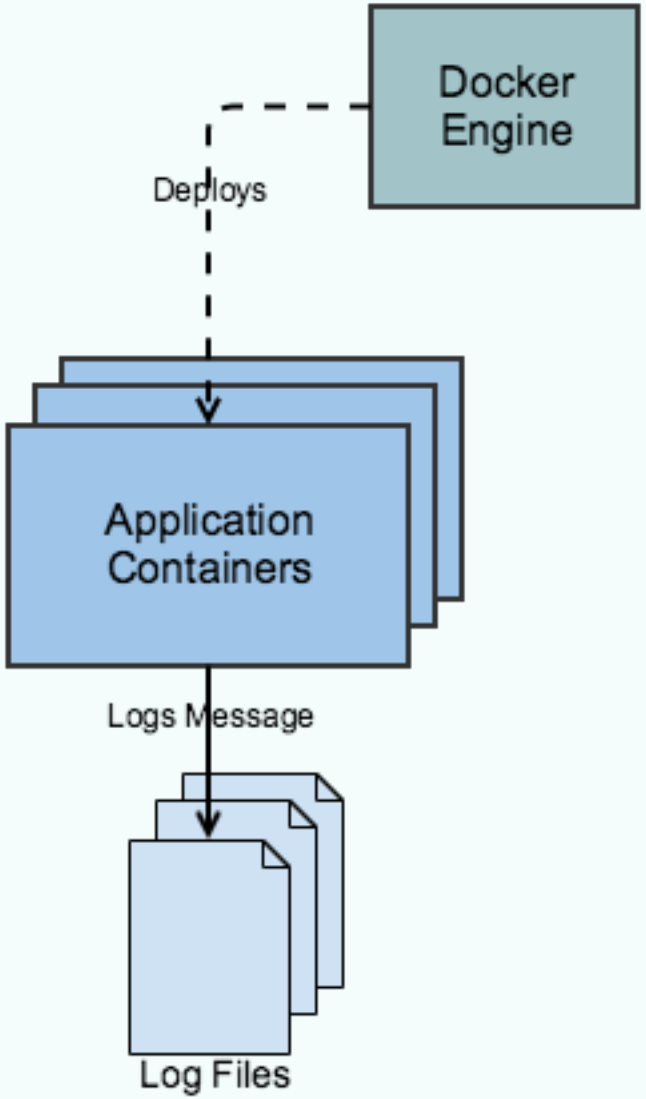
What about Aggregated Metrics?

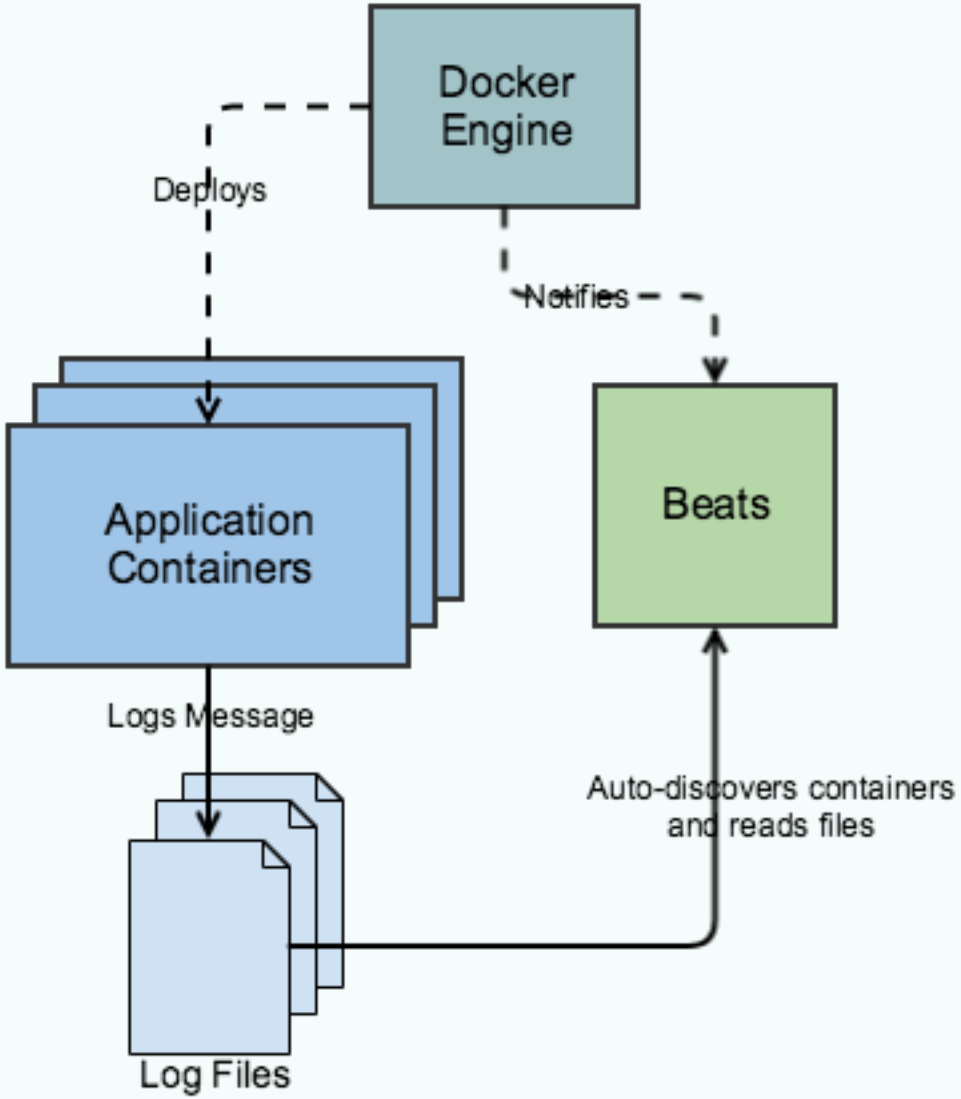
- Aggregated metrics are cheap and quick to store but lack context.
- Indicative of faults.
- Aggregated metrics are good for tracking a fluctuating numeric value.
  - Connection Pool Usage
  - JVM Memory
  - CPU Usage
  - Request rate
  - Error Rate
- Check out micrometer for JVM based metric collection:
  - https://micrometer.io/
  - Supports tags on metrics
  - Spring Boot 2+ library of choice

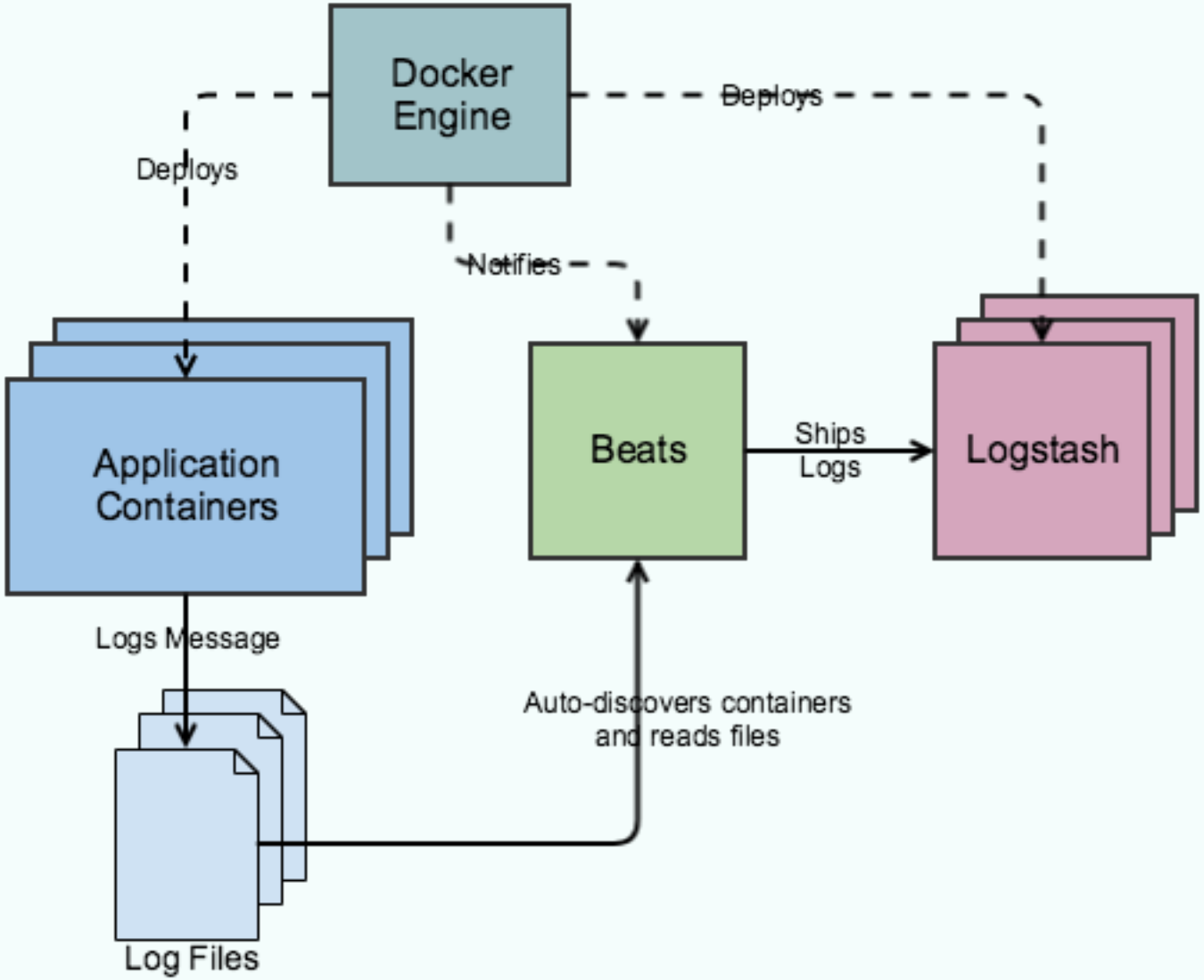
Events Strategy

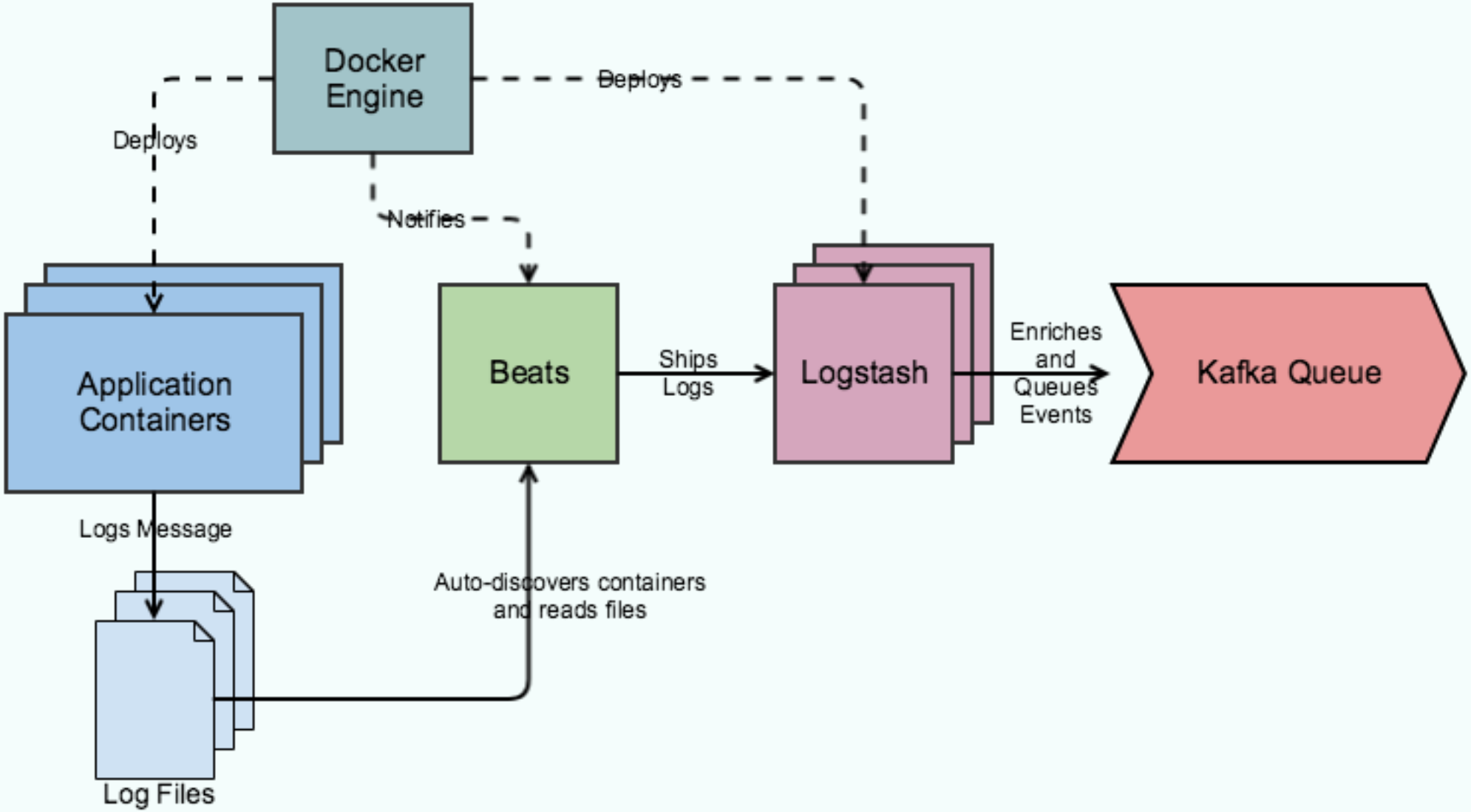
- Focus on state changes.
- Consider using spans to break down transactions.
- Start with a sensible coverage then iterate as needed.
- Practice Continuous Delivery.
- Consider thread local metadata storage to make sharing context easy.
- Watch out for edge cases and errors and ensure metadata is present in all cases.

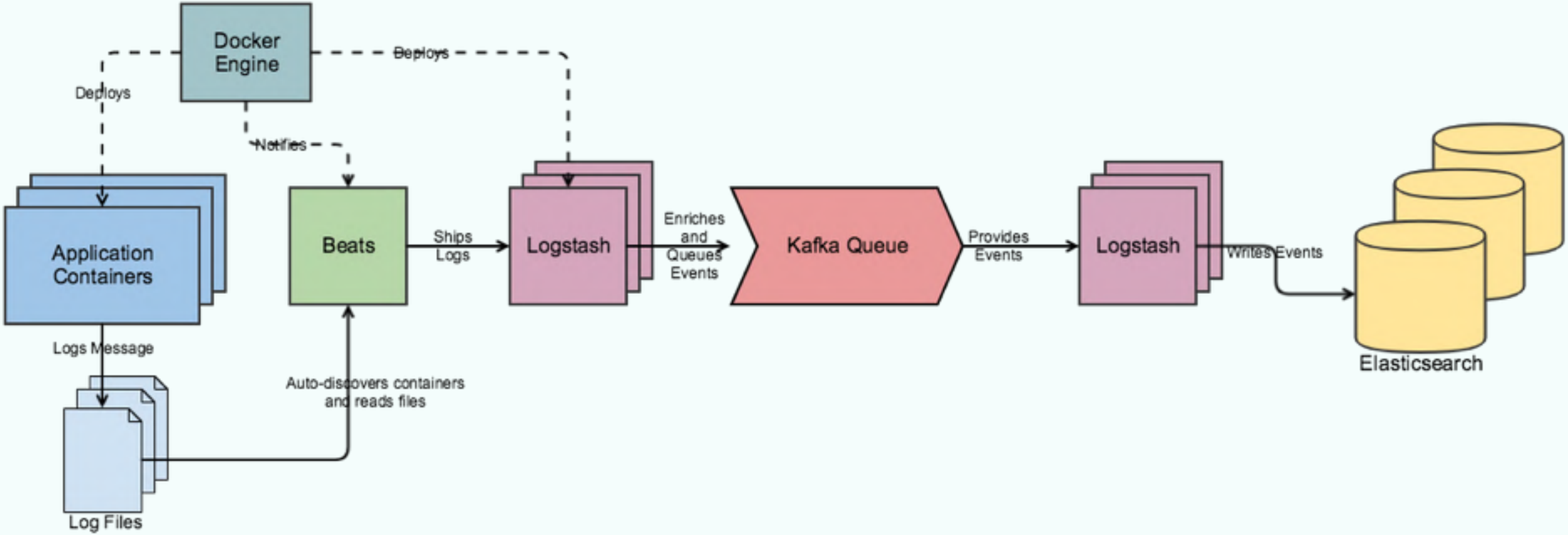

Logging Pipeline


Docker Engine
Deploys
Application Containers
Logs Message
Log Files


Docker Engine
Deploys
Notifies
Application Containers
Beats
Logs Message
Auto-discovers containers and reads files
Log Files


Docker Engine
Deploys
Deploys
Notifies
Application Containers
Beats
Ships Logs
Logstash
Logs Message
Auto-discovers containers and reads files
Log Files


Docker Engine
Deploys
Deploys
Notifies
Application Containers
Beats
Ships Logs
Logstash
Enriches and Queues Events
Kafka Queue
Logs Message
Auto-discovers containers and reads files
Log Files


Docker Engine
Deploys
Deploys
Notifies
Application Containers
Beats
Ships Logs
Logstash
Enriches and Queues Events
Kafka Queue
Provides Events
Logstash
Writes Events
Elasticsearch
Logs Message
Auto-discovers containers and reads files
Log Files

Beats and Container Auto-discovery

- A lightweight log shipper written in Go
- Has the ability to Autodiscover Kubernetes/Docker hosts based on listening to docker engine events.
- Augments events with metadata like container names, ids images, Docker Labels and Kubernetes Annotations
- https://www.elastic.co/guide/en/beats/filebeat/current/configuration-autodiscover.html

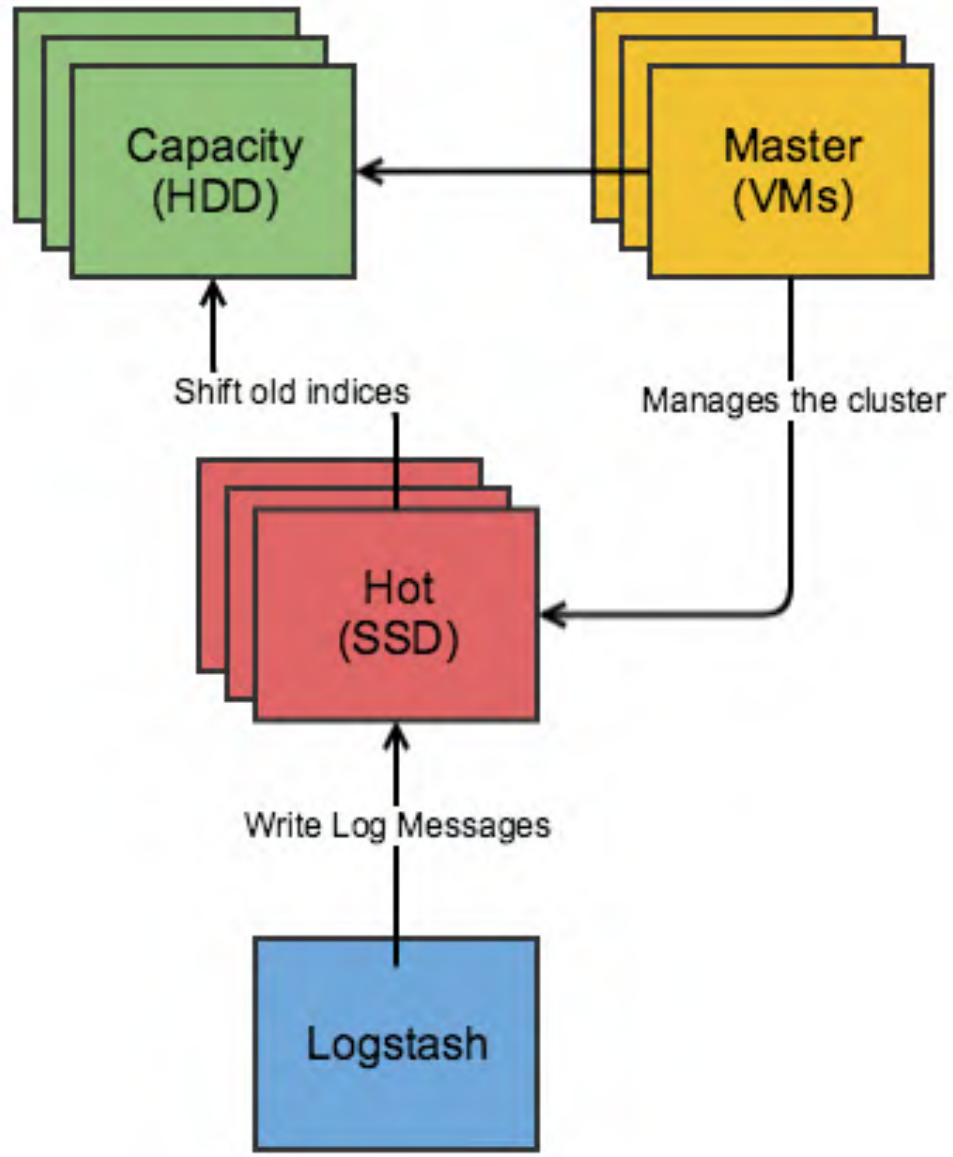
```
filebeat.autodiscover:
  providers:
    - type: docker
      templates:
        - condition:
            contains: docker.container.image: redis
          config:
            - type: docker
              containers.ids:
                - "${data.docker.container.id}”
```

Logstash as a service

- Logstash commonly used in a sidecar pattern.
- Can also act as a clustered service
- Beats can be configured to communicate to a list of Logstash servers
- Allows centralized enrichment and processing of log messages
  - User-agent normalising
  - Geo IP lookups
- Codify your Logstash setup
  - We use Pebble templates - https://github.com/PebbleTemplates/pebble

Hot-Warm Architectures

- Logstash writes to smaller faster Elasticsearch nodes sized for 24 hours
  - SSDs for fast I/O
- After 24 Hours indices are moved to slower but larger capacity nodes
  - HDDs that are cheaper and much larger
- Need to leave 30-40% capacity to allow for datacenter failure



Elasticsearch Advice

- Indexes all fields so everything is searchable.
- Don’t use dynamic schemas!
  - Use mappings to avoid type clashes and unwanted analysing.
  - Custom fields that are more dynamic can be mapped with dynamic templates to keep types and analysing consistent.
- Analyse free-text.
  - Users will search for partial stack traces and error names and expect this to work.
- Consider data roll-ups to track trends.
- Don’t map 1000s of fields in one index unless you want heap issues.

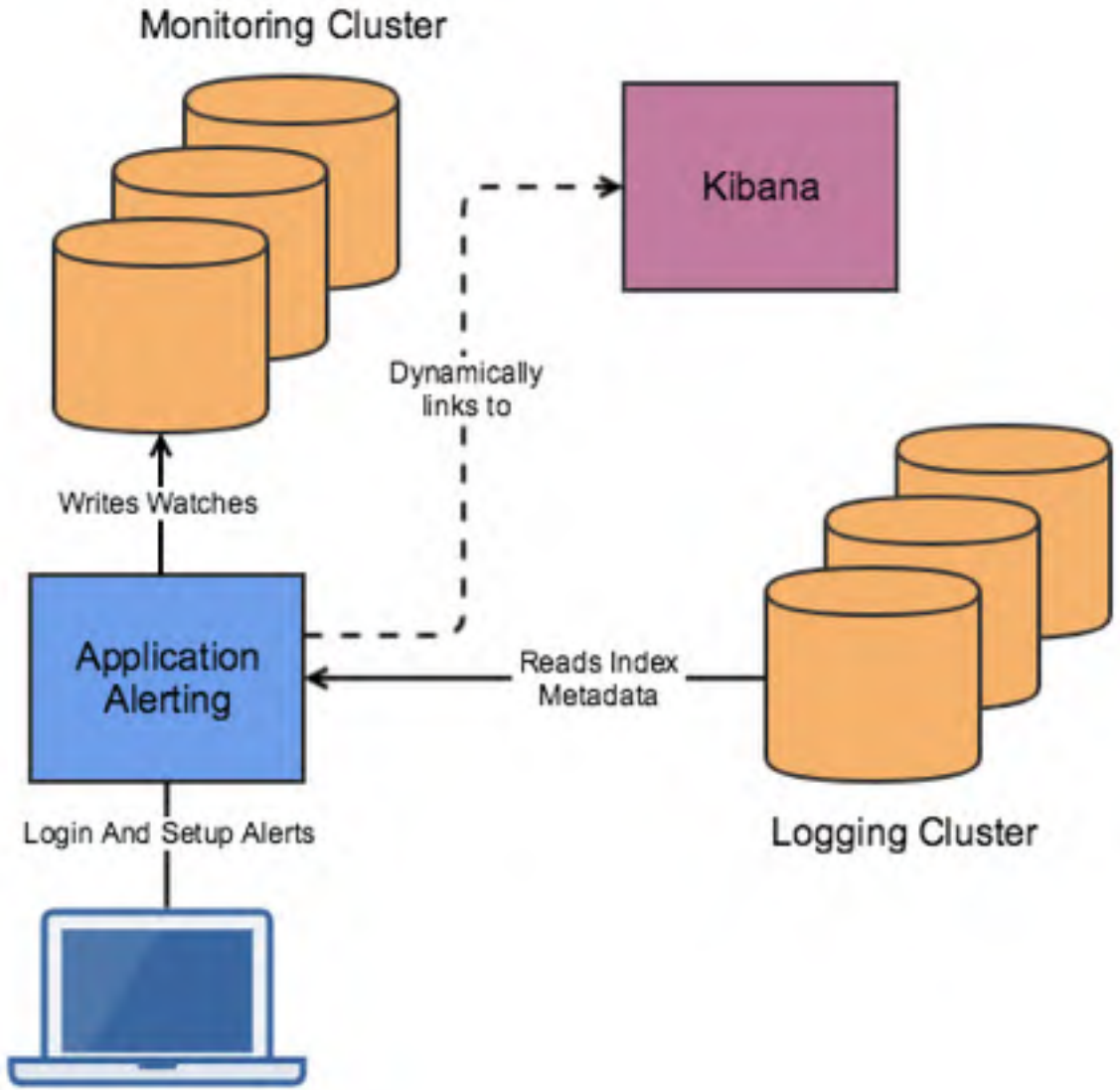
Pipeline Advice

- Codify your configuration.
- Measure your message latency.
- Set a sensible retention policy for raw data.
- Backup important metrics.
- Make your pipeline continuously deliverable and separate from your application delivery.
- Provide a test environment for developers.
- Log in pre-production!

Alerting

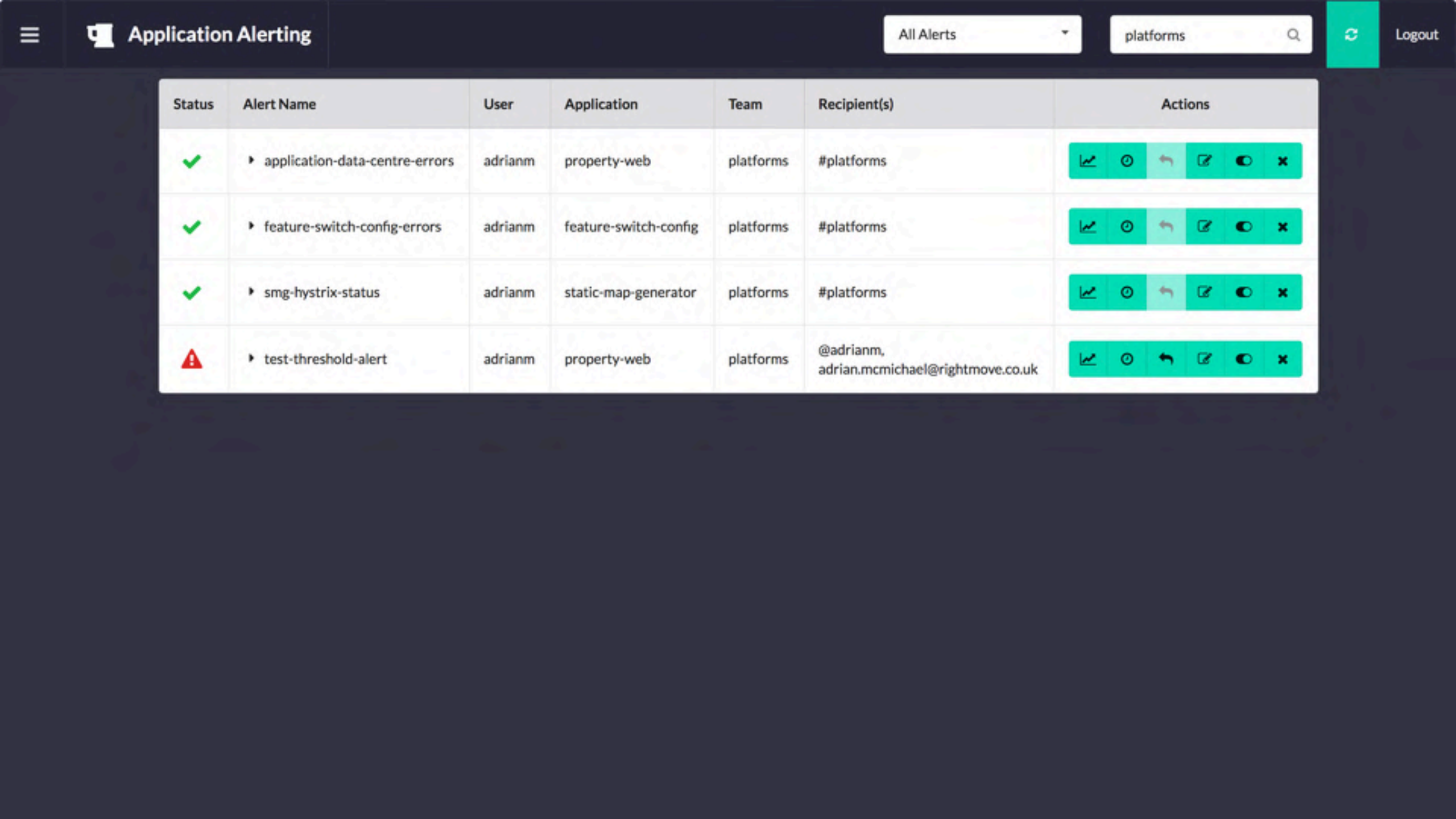
Alerting Architecture

- 2 Elasticsearch Clusters
  - One for monitoring
  - One for collecting logs
- Custom Web-App for setting up and managing alerts
  - Self-service
  - Covers complexity
  - Uses Kibana where possible instead of reinvention



Watcher/X-Pack Alerting

- Part of Elastic's X-Pack suite
- Allows us to alert based on our logging data directly
- REST API based setup
- Backed by a configurable data context object
- Uses a groovy-like scripting language called Painless
- A watch consists of:
  - Input – Adds any input data needed to check the alert condition
  - Trigger – How often the alert should run
  - Condition – The condition to check and alert on
  - Actions – What to do when alerting, e.g. send a slack message
  - Transforms – Allows the optional transformation of data for use in actions


Application Alerting
All Alerts
platforms
Logout
Status
Alert Name
User
Application
Team
Recipient(s)
Actions
application-data-centre-errors
adrianm
property-web
platforms
#platforms
feature-switch-config-errors
adrianm
feature-switch-config
platforms
#platforms
smg-hystrix-status
adrianm
static-map-generator
platforms
#platforms
test-threshold-alert
adrianm
property-web
platforms
@adrianm,
adrian.mcmichael@rightmove.co.uk

Set up a query alert

Pick a team to own the alert

platforms

Name your alert

test-query-alert

Select an index

access_log_*

Select an application

static-map-generator

Define your alert query

response:[400 TO *]

Alert query currently returns 3,321 hits in the last 24hrs

Setup alert condition

Will alert if the percentage of documents that matches the query is **above** 0 % for 5 minute(s).

Setup alert interval

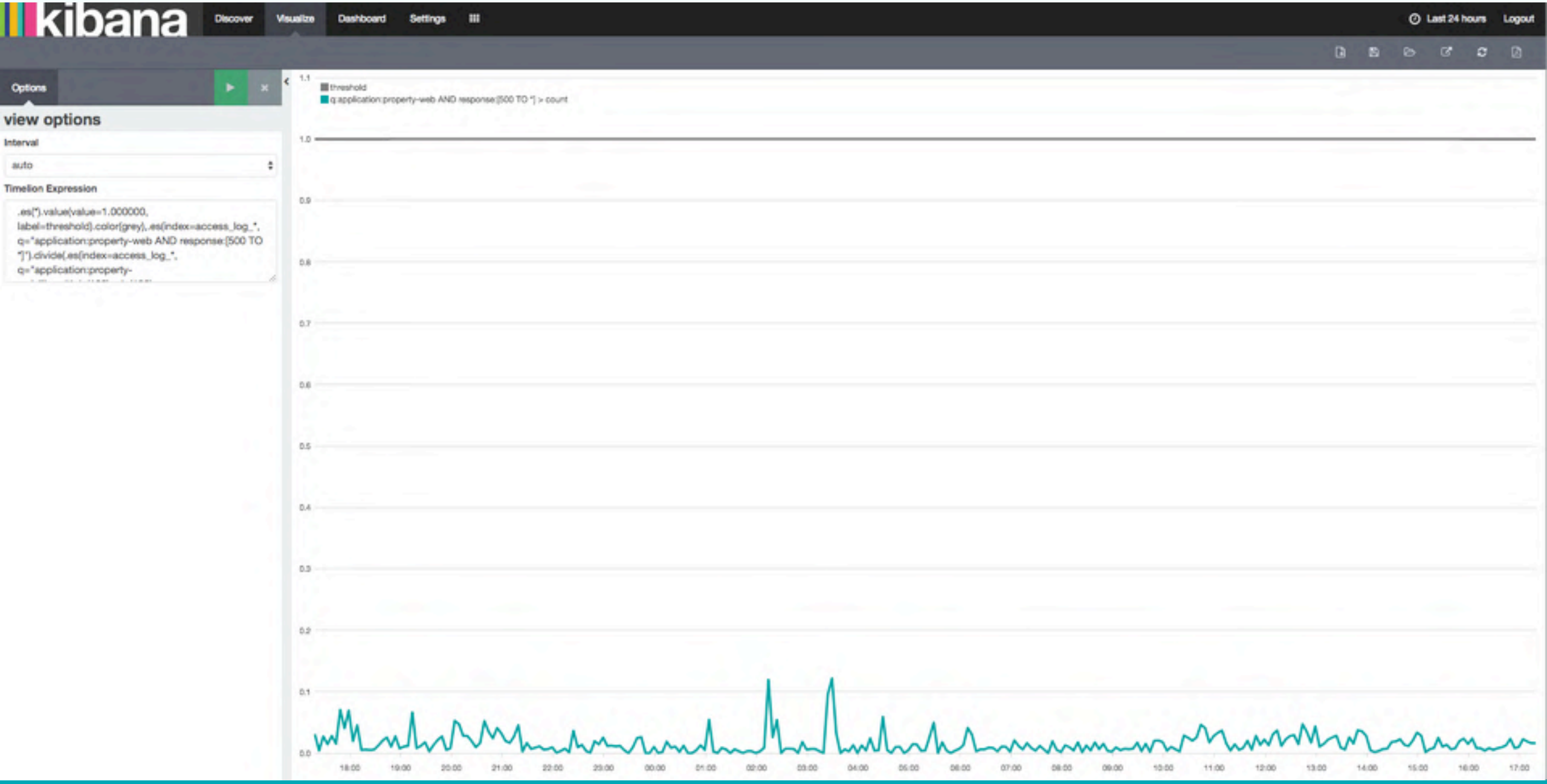
Run this alert every 5 minute(s).

Setup your alert actions

- [ ] Alert using email
- [x] Alert using slack

@adrianm

Throttle actions after an alert for 15 minute(s).


kibana
Discover
Visualize
Dashboard
Settings
Last 24 hours
Logout
Options
view options
Interval
auto
Timelion Expression
.es(*).value(value=1.000000,
label=threshold).color(grey),.es(index=access_log_*,
q="application:property-web AND response:[500 TO
*]").divide(.es(index=access_log_*,
q="application:property-
threshold
q:application:property-web AND response:[500 TO *] > count
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
18:00
19:00
20:00
21:00
22:00
23:00
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00

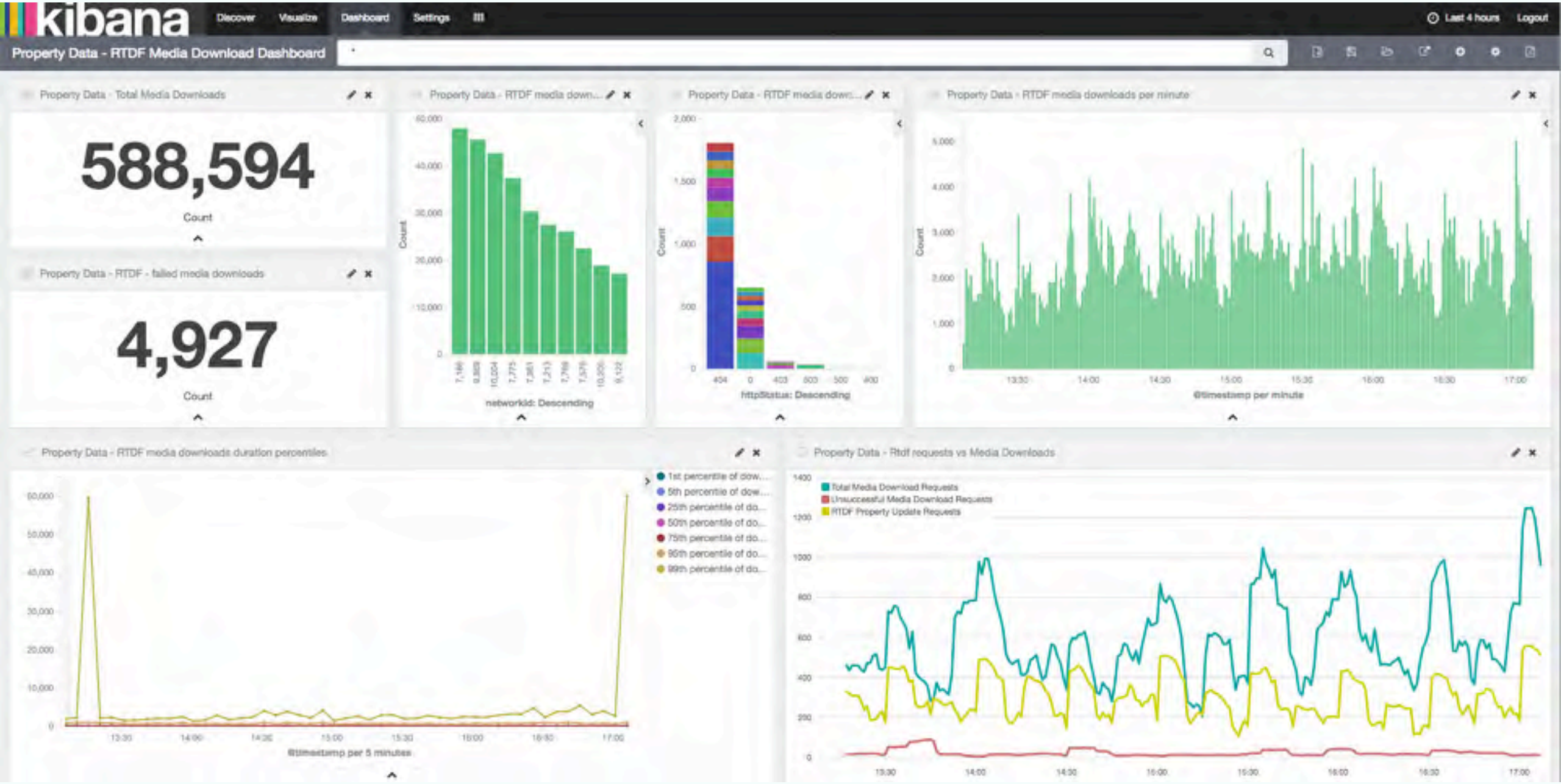
Alerting Advice

- Try to focus on what matters!
  - Traffic
  - Error Rate
  - Duration of important types of requests
  - Important KPIs
- Give teams power to configure themselves but be prepared to offer guidance.
- Health is a sliding scale!
  - Understand what healthy looks like for your system.
- Fix issues as they arise!
- Building a system isn't enough!

The Results

Cultural Change

- Developers naturally reach out to the tooling when issues occur.
- Workshops have helped spread the knowledge amongst teams.
- Other areas of the business are looking to Kibana dashboard for support processes.
- Queuing events has led to exploration of other Data Processing use-cases
- Made a difference when starting new projects.


kibana
Discover
Visualize
Dashboard
Settings
Last 4 hours
Logout
Property Data - RTDF Media Download Dashboard
Property Data - Total Media Downloads
588,594
Count
Property Data - RTDF media down...
networkId: Descending
Property Data - RTDF media down...
httpStatus: Descending
Property Data - RTDF media downloads per minute
@timestamp per minute
Property Data - RTDF - failed media downloads
4,927
Count
Property Data - RTDF media downloads duration percentiles
1st percentile of dow...
5th percentile of dow...
25th percentile of do...
50th percentile of do...
75th percentile of do...
95th percentile of do...
99th percentile of do...
@timestamp per 5 minutes
Property Data - Rtdf requests vs Media Downloads
Total Media Download Requests
Unsuccessful Media Download Requests
RTDF Property Update Requests

rightmove

Final Advice!

- Sounds like a lot of work!
  - It is but that's okay
  - Think of this like your testing
  - Make time for it - its easy to show the benefits to management
- Treat it with respect and care
  - Crappy logging and alerting helps no-one and erodes trust
- Share with Others
  - Show them how you figured out problems
  - Discuss KPIs and health
  - Hold Reviews!
- Keep trying!

Big Shout Out to…

- Matthew Skelton
  - https://twitter.com/matthewpskelton
- Charity Majors
  - https://twitter.com/mipsytipsy
- Cindy Sridharan
  - https://twitter.com/copyconstruct
- O11ycast
  - https://www.heavybit.com/library/podcasts/o11ycast/
- My team at Rightmove
  - Especially Alex Palmer who helped with the Lego photography

The End

Adrian McMichael
@trev_boxmonster
adrian.mcmichael@rightmove.co.uk

rightmove
find your happy