Effective Monitoring of Web Services with Heartbeat and Uptime

AJ Pahl
Manager, Solutions Architecture

January 2020

Housekeeping & Logistics

- **Slides and recording** will be available following the webinar
- Chat via IRC #elastic-webinar
  - #elastic-webinar @ Freenode
  - Click "Join the Chat" link, create an IRC account
- Please select high resolution in the YouTube video player

AJ Pahl
@ajpahl1008

elastic

But first...

apm

+

logging

+

metrics

=

elastic observability

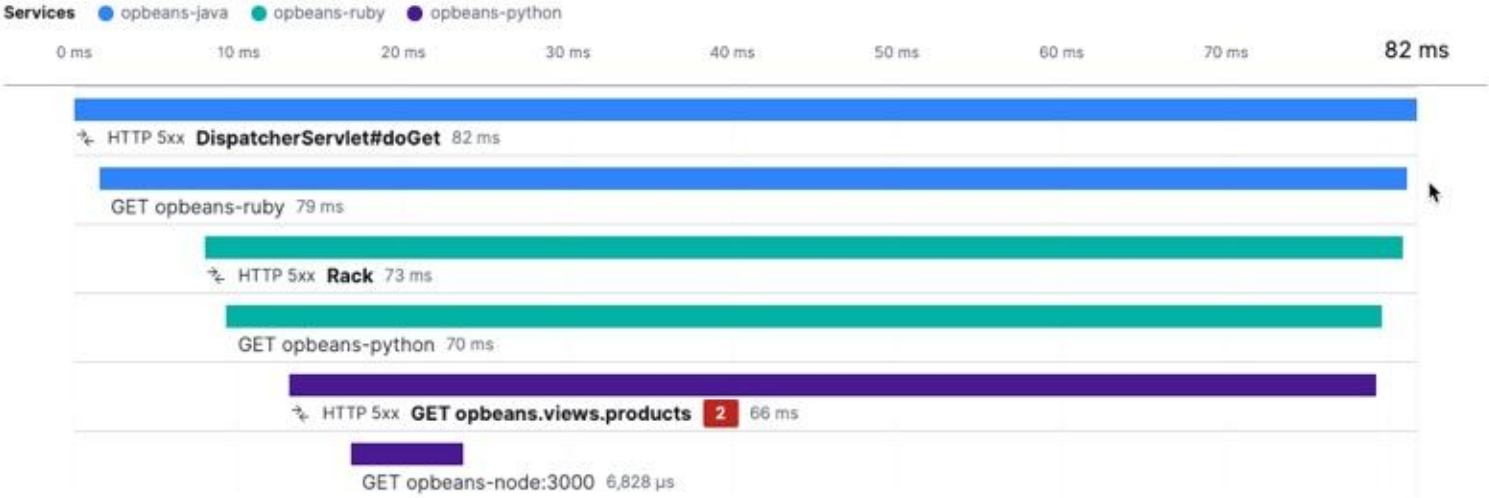
Trace sample

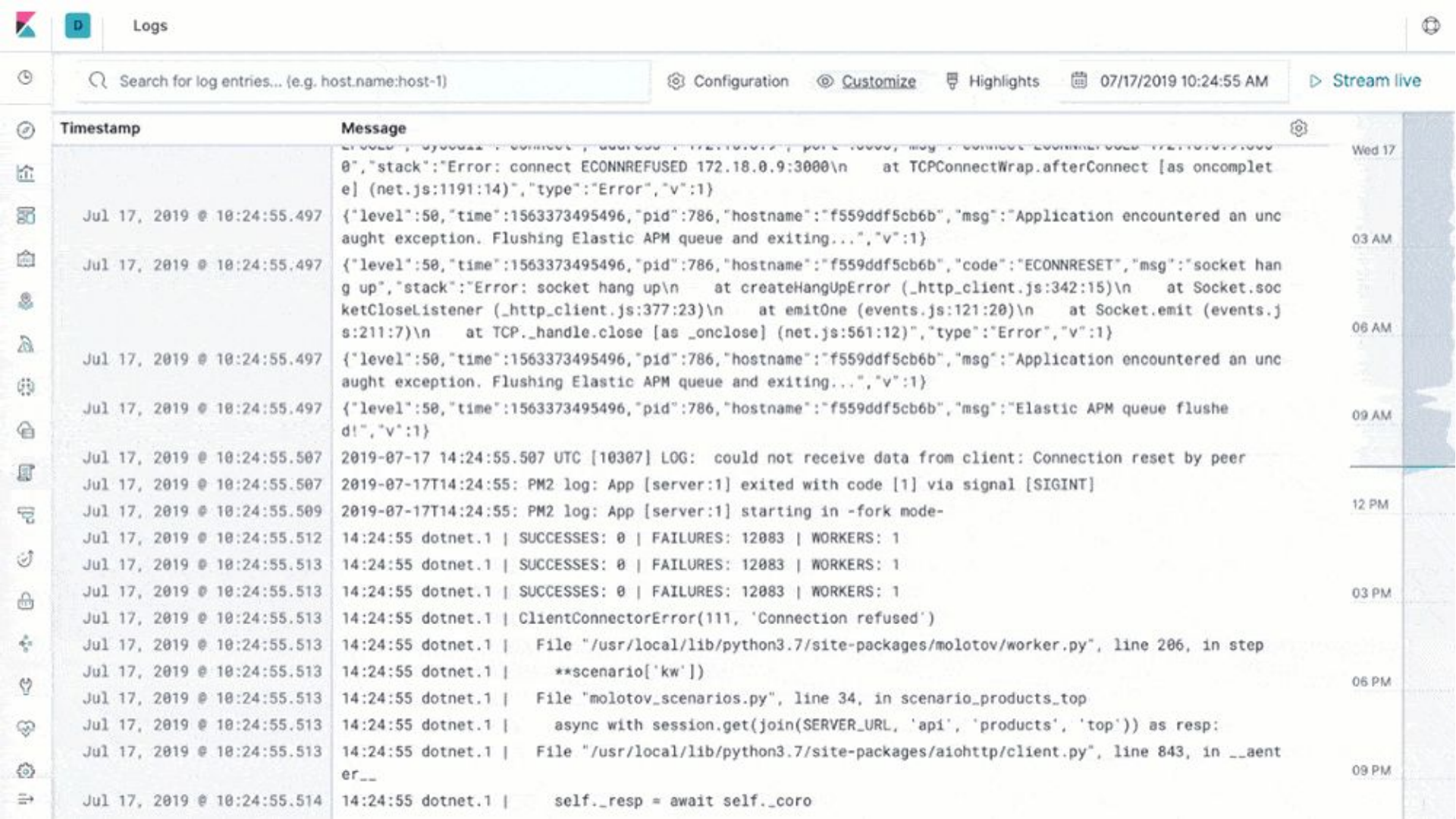
Actions

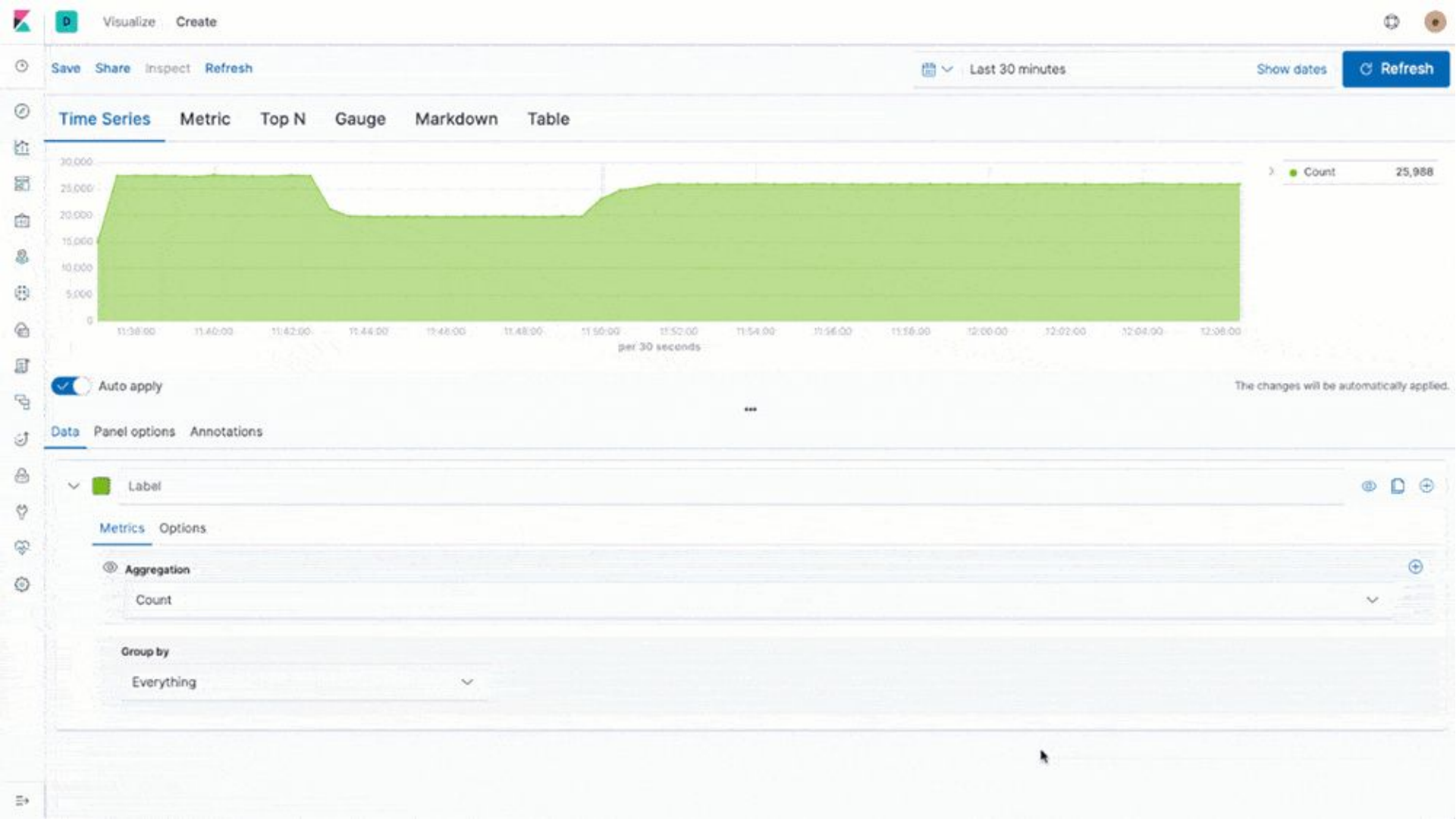
View full trace

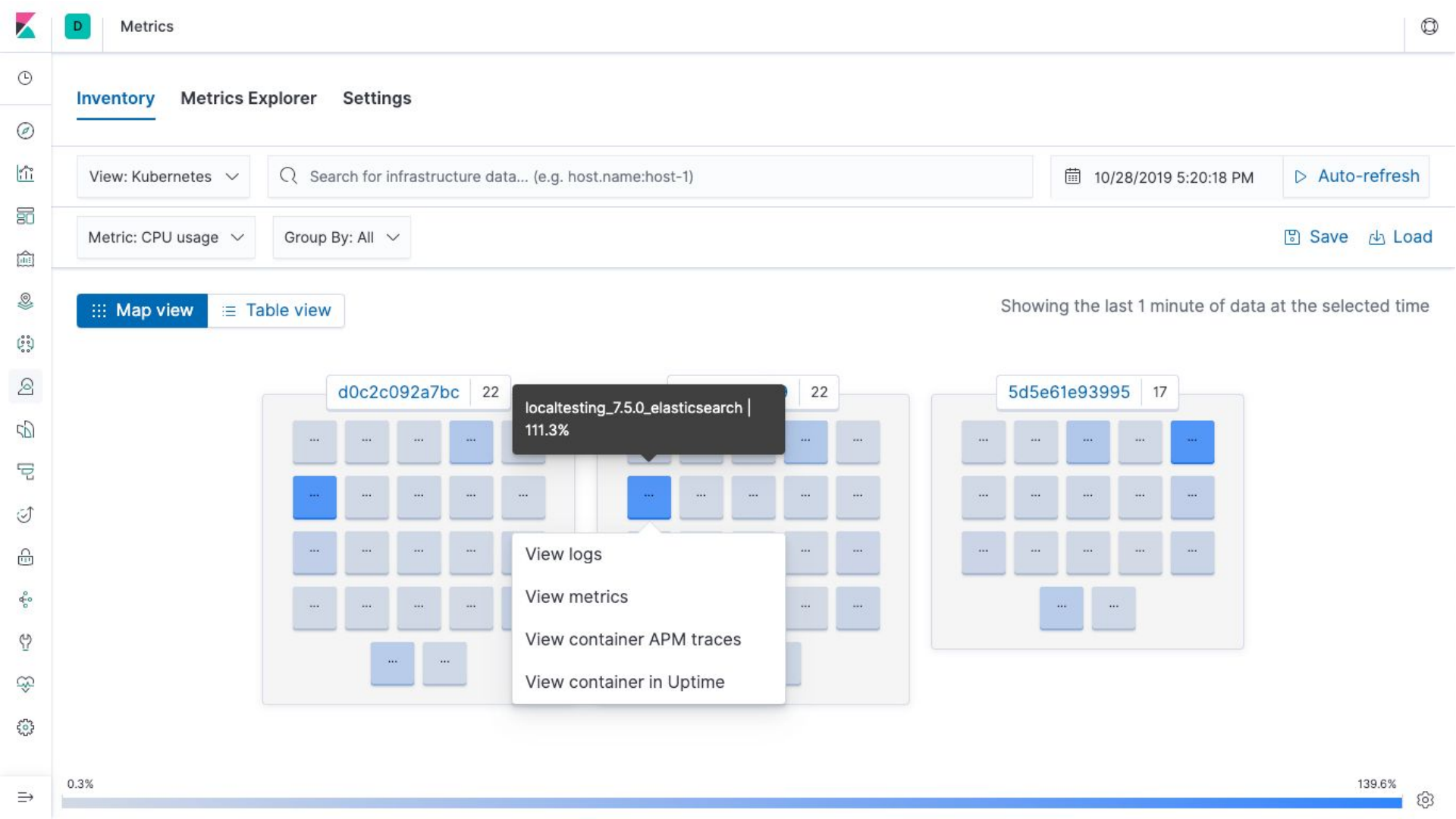
16 minutes ago | 82 ms (89.9% of trace) | GET http://opbeans-java:3000/api/products | 500 Internal Server Error | 2 Errors | Other

Timeline | Metadata




Logs
Search for log entries... (e.g. host.name:host-1)
Configuration
Customize
Highlights
07/17/2019 10:24:55 AM
Stream live
Timestamp
Message
0","stack":"Error: connect ECONNREFUSED 172.18.0.9:3000\n at TCPConnectWrap.afterConnect [as oncomplete] (net.js:1191:14)","type":"Error","v":1}
Jul 17, 2019 @ 10:24:55.497 {"level":50,"time":1563373495496,"pid":786,"hostname":"f559ddf5cb6b","msg":"Application encountered an uncaught exception. Flushing Elastic APM queue and exiting...","v":1}
Jul 17, 2019 @ 10:24:55.497 {"level":50,"time":1563373495496,"pid":786,"hostname":"f559ddf5cb6b","code":"ECONNRESET","msg":"socket hang up","stack":"Error: socket hang up\n at createHangUpError (_http_client.js:342:15)\n at Socket.socketCloseListener (_http_client.js:377:23)\n at emitOne (events.js:121:20)\n at Socket.emit (events.js:211:7)\n at TCP._handle.close [as _onclose] (net.js:561:12)","type":"Error","v":1}
Jul 17, 2019 @ 10:24:55.497 {"level":50,"time":1563373495496,"pid":786,"hostname":"f559ddf5cb6b","msg":"Application encountered an uncaught exception. Flushing Elastic APM queue and exiting...","v":1}
Jul 17, 2019 @ 10:24:55.497 {"level":50,"time":1563373495496,"pid":786,"hostname":"f559ddf5cb6b","msg":"Elastic APM queue flushed!","v":1}
Jul 17, 2019 @ 10:24:55.507 2019-07-17 14:24:55.507 UTC [10307] LOG: could not receive data from client: Connection reset by peer
Jul 17, 2019 @ 10:24:55.507 2019-07-17T14:24:55: PM2 log: App [server:1] exited with code [1] via signal [SIGINT]
Jul 17, 2019 @ 10:24:55.509 2019-07-17T14:24:55: PM2 log: App [server:1] starting in -fork mode-
Jul 17, 2019 @ 10:24:55.512 14:24:55 dotnet.1 | SUCCESSES: 0 | FAILURES: 12083 | WORKERS: 1
Jul 17, 2019 @ 10:24:55.513 14:24:55 dotnet.1 | SUCCESSES: 0 | FAILURES: 12083 | WORKERS: 1
Jul 17, 2019 @ 10:24:55.513 14:24:55 dotnet.1 | SUCCESSES: 0 | FAILURES: 12083 | WORKERS: 1
Jul 17, 2019 @ 10:24:55.513 14:24:55 dotnet.1 | ClientConnectorError(111, 'Connection refused')
Jul 17, 2019 @ 10:24:55.513 14:24:55 dotnet.1 | File "/usr/local/lib/python3.7/site-packages/molotov/worker.py", line 206, in step
Jul 17, 2019 @ 10:24:55.513 14:24:55 dotnet.1 | **scenario['kw'])
Jul 17, 2019 @ 10:24:55.513 14:24:55 dotnet.1 | File "molotov_scenarios.py", line 34, in scenario_products_top
Jul 17, 2019 @ 10:24:55.513 14:24:55 dotnet.1 | async with session.get(join(SERVER_URL, 'api', 'products', 'top')) as resp:
Jul 17, 2019 @ 10:24:55.513 14:24:55 dotnet.1 | File "/usr/local/lib/python3.7/site-packages/aiohttp/client.py", line 843, in __aenter__
Jul 17, 2019 @ 10:24:55.514 14:24:55 dotnet.1 | self._resp = await self._coro
Wed 17
03 AM
06 AM
09 AM
12 PM
03 PM
06 PM
09 PM

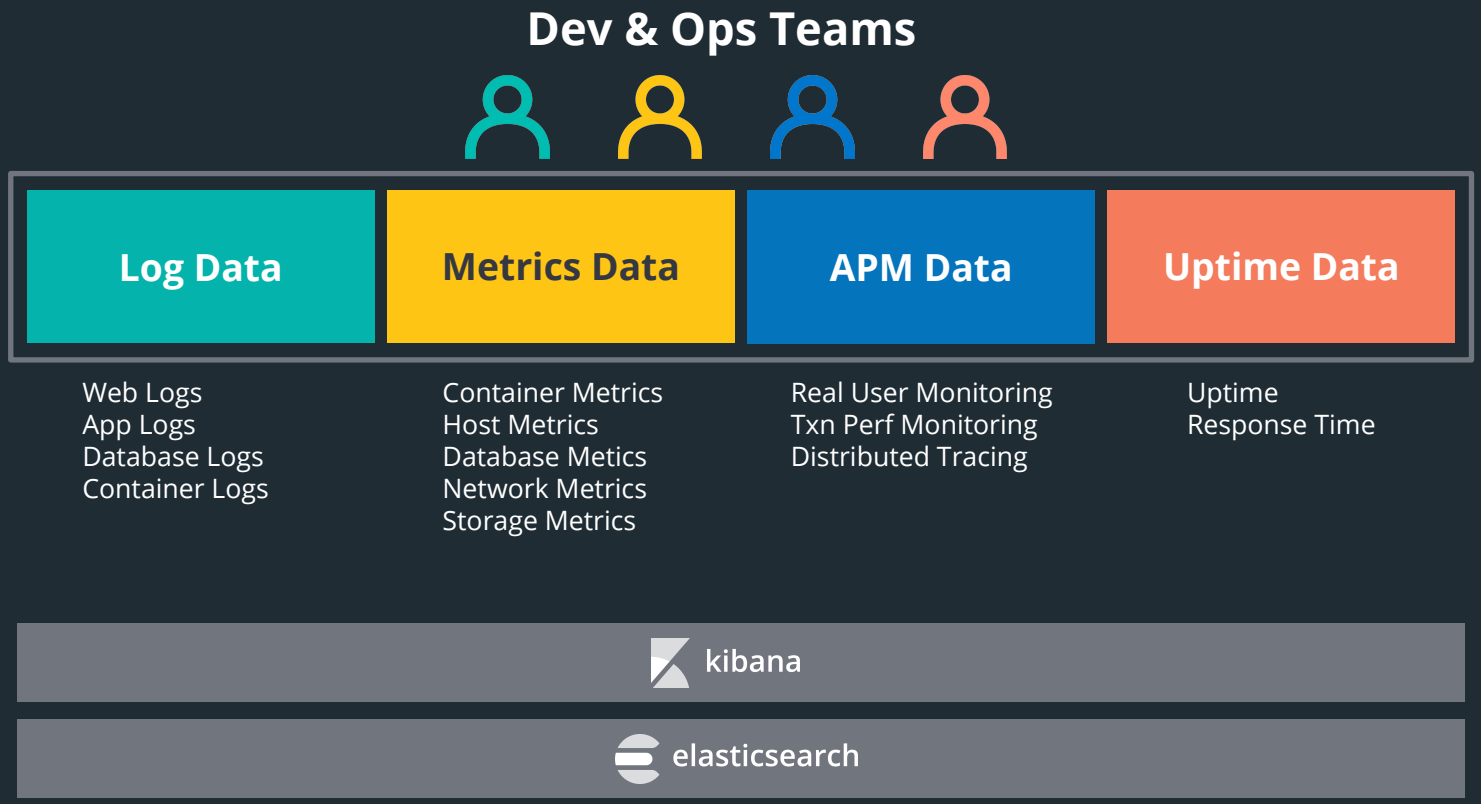

Visualize
Create
Save
Share
Inspect
Refresh
Last 30 minutes
Show dates
Refresh
Time Series
Metric
Top N
Gauge
Markdown
Table
30,000
25,000
20,000
15,000
10,000
5,000
0
11:38:00
11:40:00
11:42:00
11:44:00
11:46:00
11:48:00
11:50:00
11:52:00
11:54:00
11:56:00
11:58:00
12:00:00
12:02:00
12:04:00
12:06:00
per 30 seconds
Count
25,988
Auto apply
The changes will be automatically applied.
Data
Panel options
Annotations
Label
Metrics
Options
Aggregation
Count
Group by
Everything


D
Metrics
Inventory
Metrics Explorer
Settings
View: Kubernetes
Search for infrastructure data... (e.g. host.name:host-1)
10/28/2019 5:20:18 PM
Auto-refresh
Metric: CPU usage
Group By: All
Save
Load
Map view
Table view
Showing the last 1 minute of data at the selected time
d0c2c092a7bc
22
localtesting_7.5.0_elasticsearch
111.3%
22
5d5e61e93995
17
View logs
View metrics
View container APM traces
View container in Uptime
0.3%
139.6%

uptime

elastic

Elastic Approach to Observability



Why is this important?

Weekends

KPI
< Key Performance Indicators >

Home Life

NPS
< Net Promoter Score >

Stop Wasting Time

Focus on Projects

Get Day Job Done

MTTR
<Mean Time to Resolution>

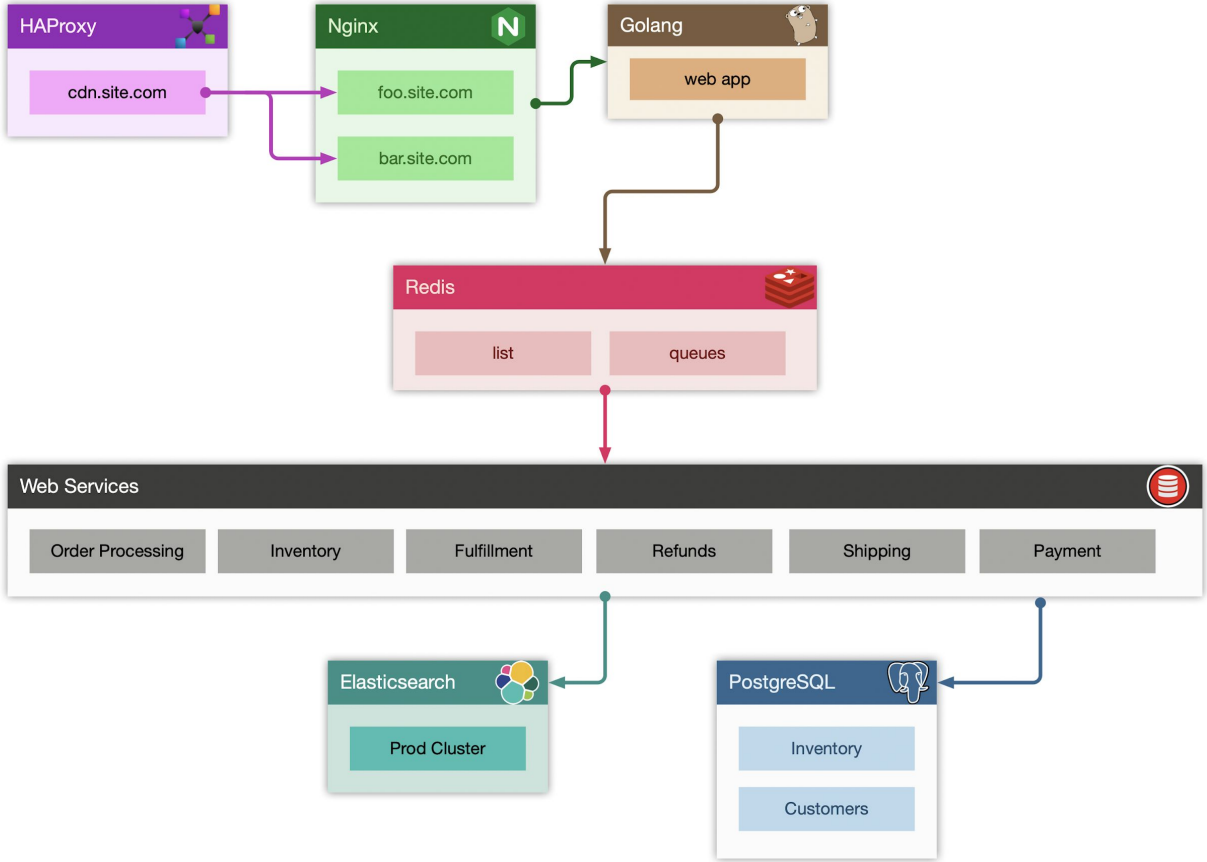
SLA
< Service Level Agreements >

Need Sleep

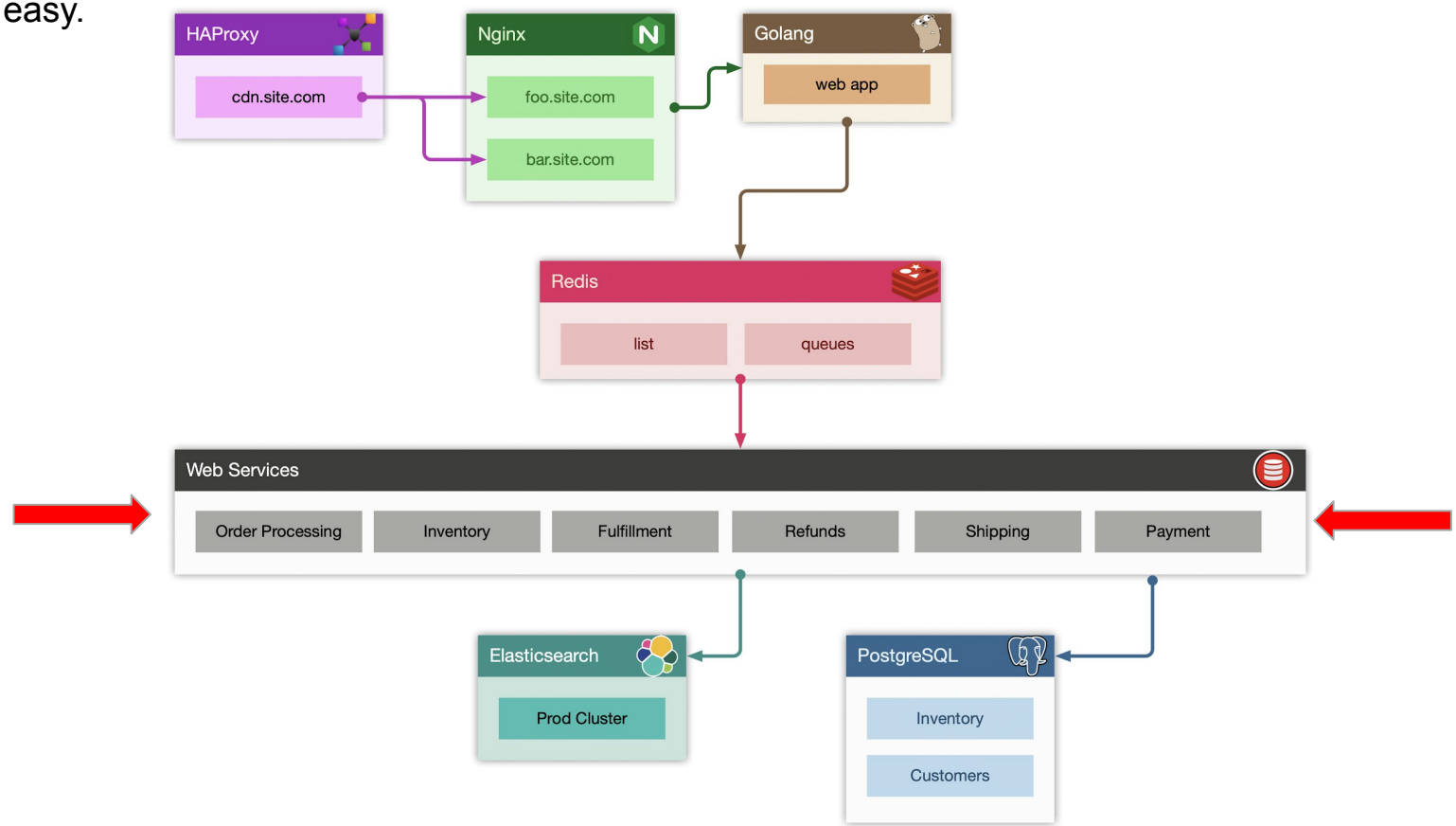
Lost Productivity

Easy.

Not so easy.


HAProxy
cdn.site.com
Nginx
foo.site.com
bar.site.com
Golang
web app
Redis
list
queues
Web Services
Order Processing
Inventory
Fulfillment
Refunds
Shipping
Payment
Elasticsearch
Prod Cluster
PostgreSQL
Inventory
Customers

Not so easy.


HAProxy
cdn.site.com
Nginx
foo.site.com
bar.site.com
Golang
web app
Redis
list
queues
Web Services
Order Processing
Inventory
Fulfillment
Refunds
Shipping
Payment
Elasticsearch
Prod Cluster
PostgreSQL
Inventory
Customers

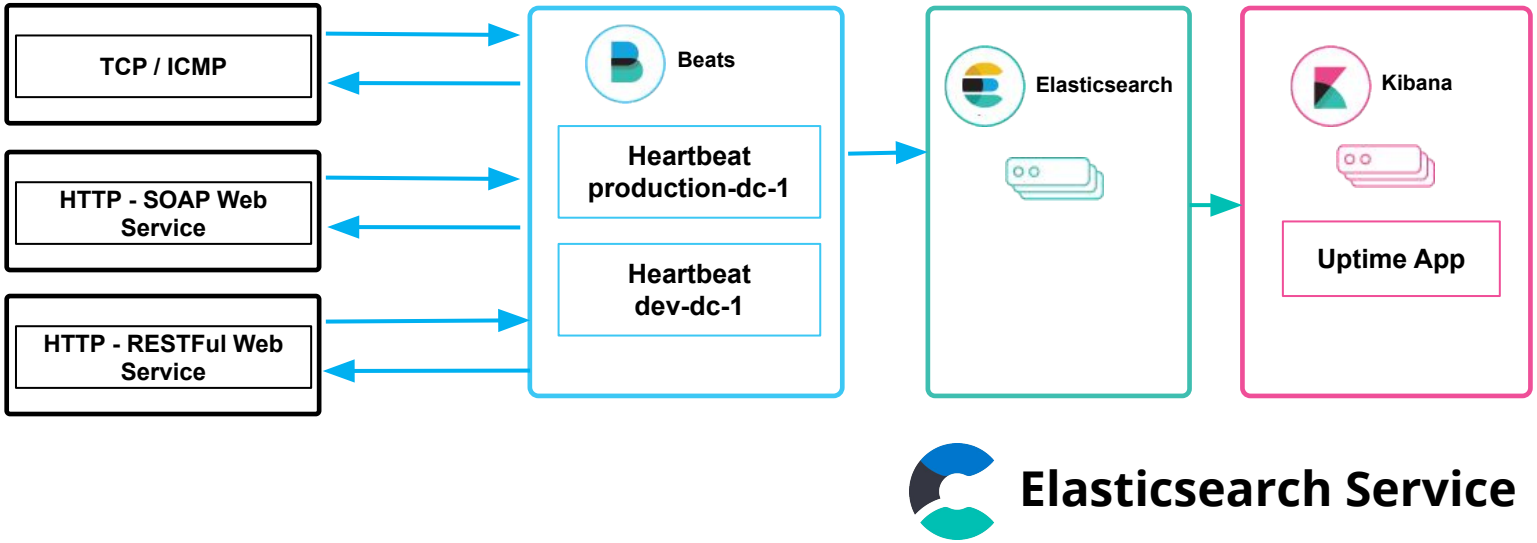
Even still...


Web Services
Web Services
Web Services
Web Services
Order Processing
Inventory
Fulfillment
Refunds
Shipping
Payment

So how can this be made it easy?

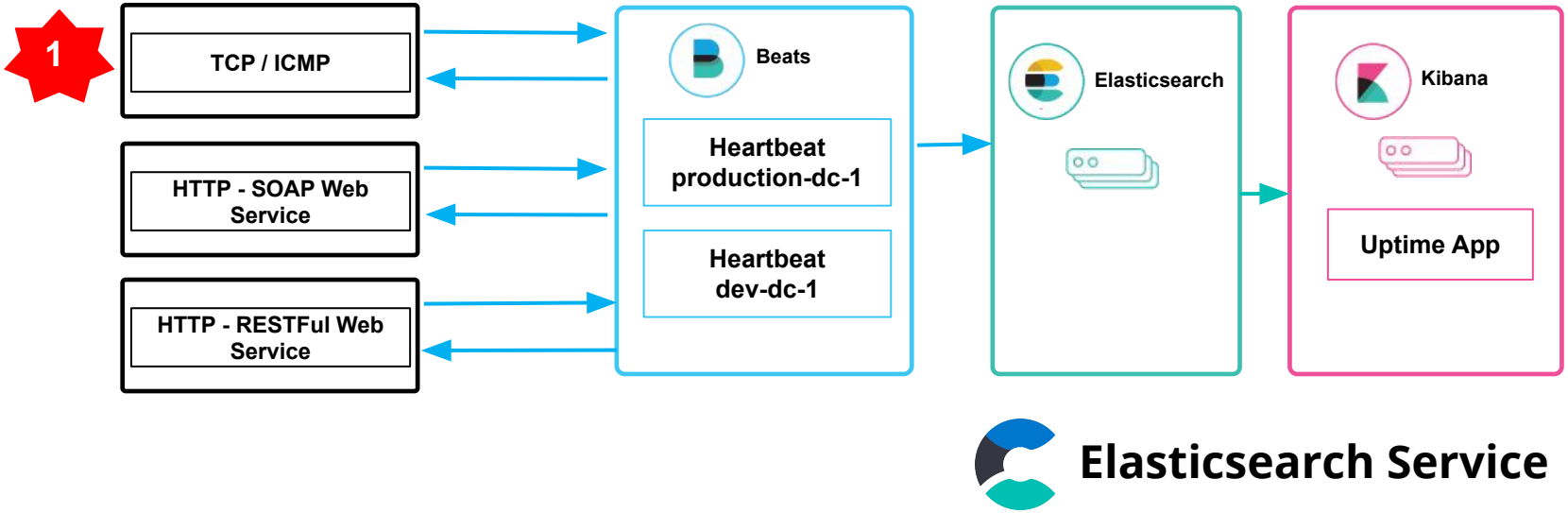
Demo Architecture

Elastic Stack



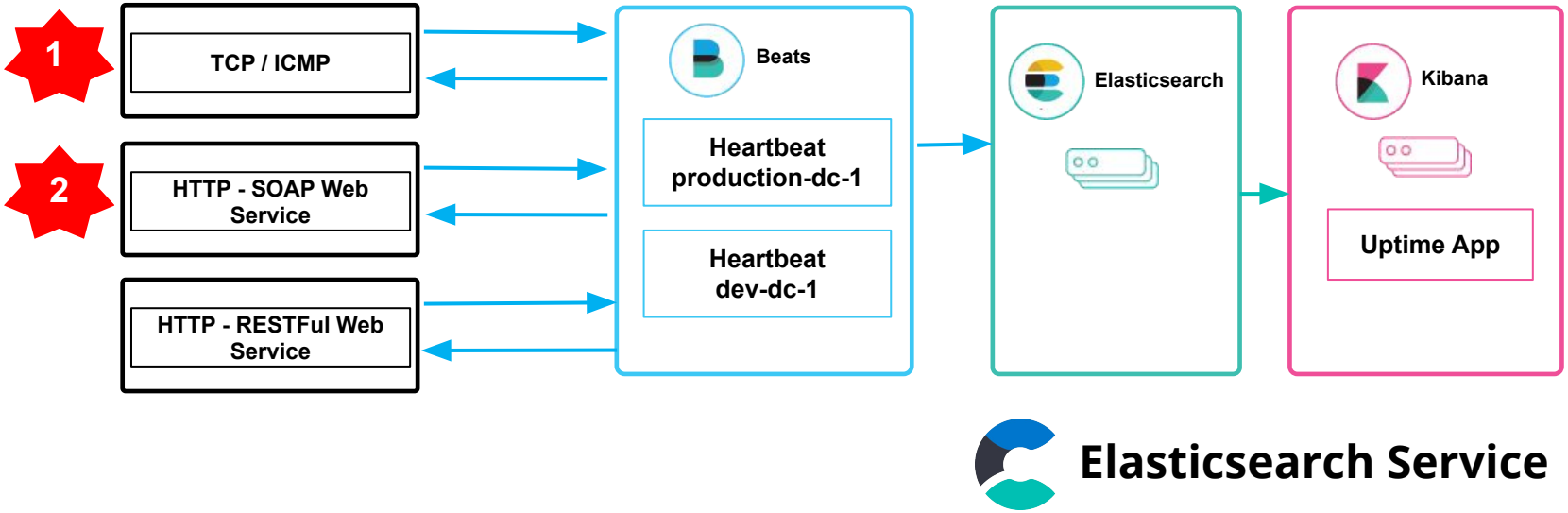
elastic

Elastic Stack

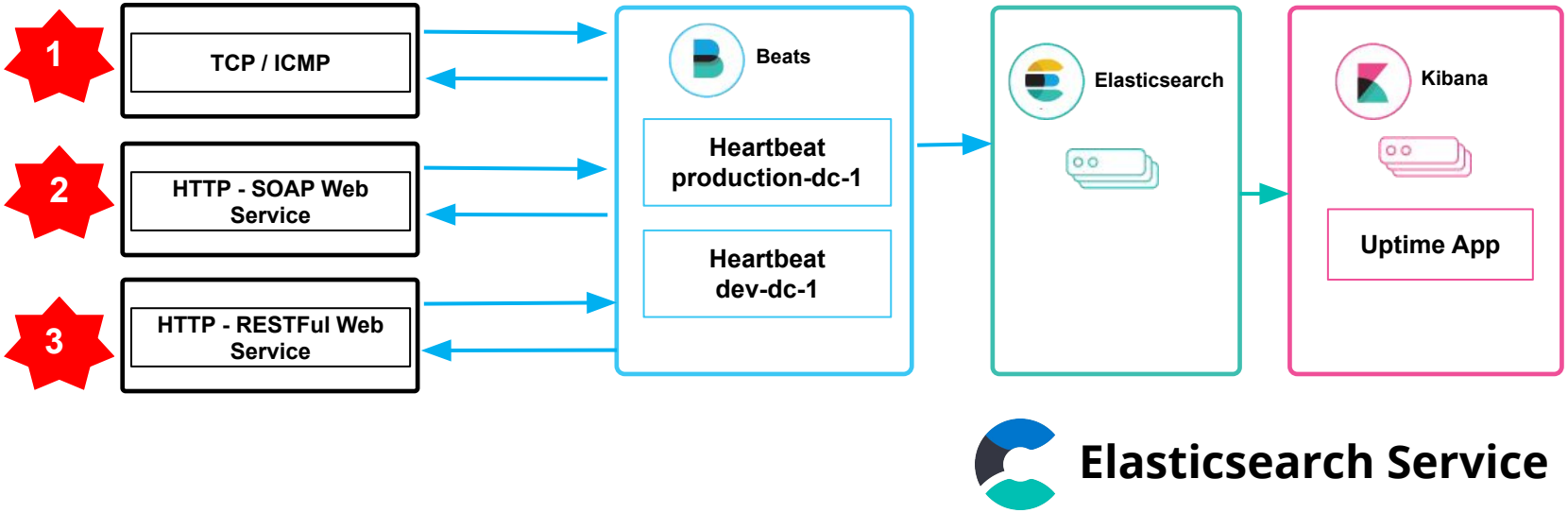


elastic

Elastic Stack



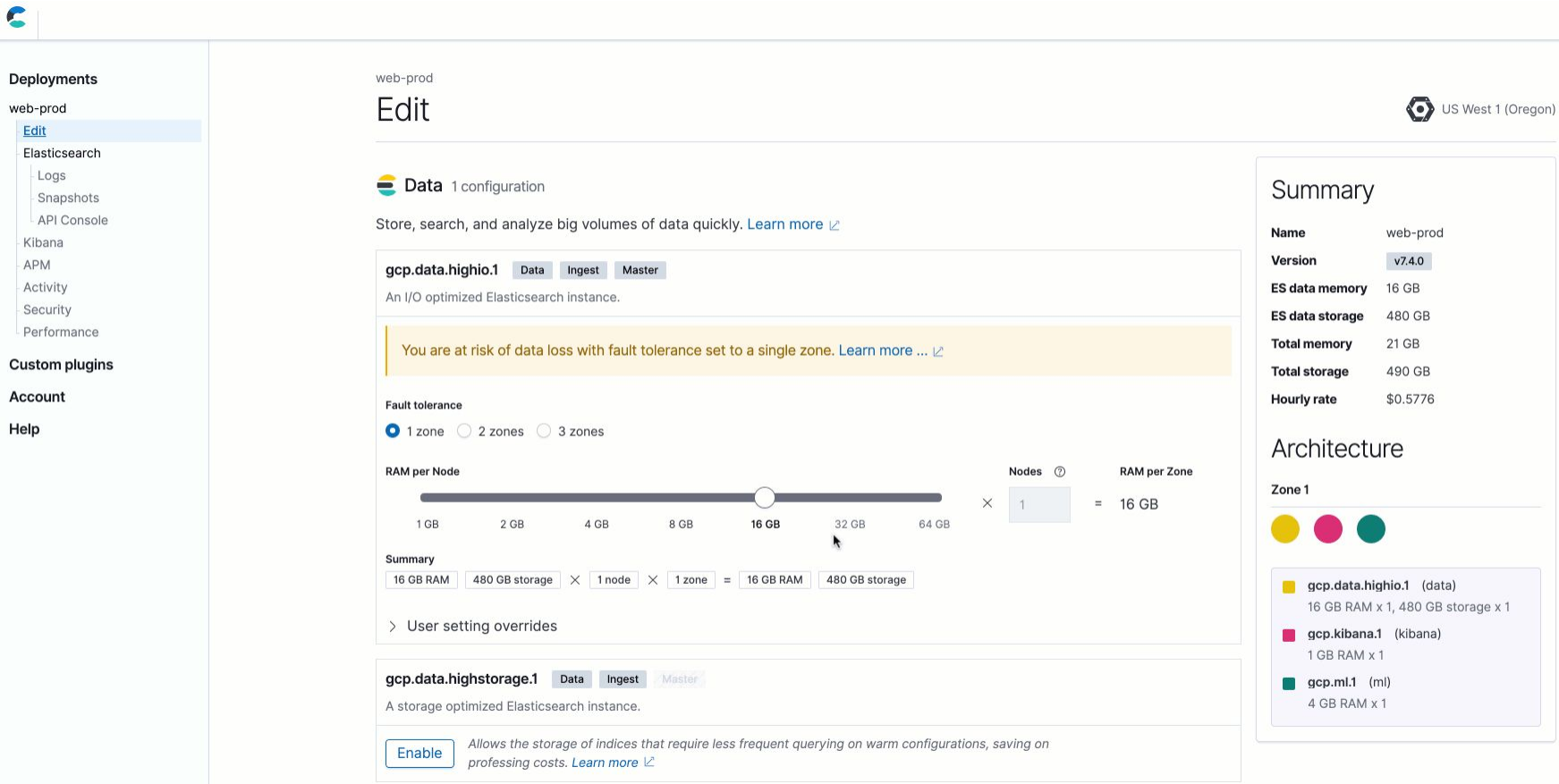
Elastic Stack



elastic

Elasticsearch Service

Hosted Elasticsearch, from the creators. No one hosts it better.



Installation & Start

Downloading Heartbeat

```
wget https://artifacts.elastic.co/downloads/beats/heartbeat/heartbeat-7.5.1-darwin-x86_64.tar.gz

(Other operating environments)
# wget https://artifacts.elastic.co/downloads/beats/heartbeat/heartbeat-7.5.1-linux-x86_64.tar.gz
# wget https://artifacts.elastic.co/downloads/beats/heartbeat/heartbeat-7.5.1-windows-x86_64.zip

tar zxvf heartbeat-7.5.1-darwin-x86_64.tar.gz
cd heartbeat-7.5.1-darwin-x86_64
```

Configure heartbeat monitors

```
heartbeat.monitors:
- type: http
  urls: ["https://httpbin.org"]
  schedule: '@every 30s'

# set name
- type: http
  urls: ["https://httpbin.org"]
  schedule: '@every 30s'
  name: "my-other-httpbin-check"

# add tags
- type: http
  urls: ["https://httpbin.org"]
  schedule: '@every 30s'
  tags: "my-httpbin-tag"

# always fails
- type: http
  urls: ["https://httpbin.org/status/404"]
  check.request.method: "POST"
  schedule: '@every 30s'
```

elastic

Configure heartbeat monitors

```
# expect 404 status
- type: http
  urls: ["https://httpbin.org/status/404"]
  schedule: '@every 30s'
  check.request.method: "POST"
  check.response.status: 404

# send a certain body
- type: http
  urls: ["https://httpbin.org/anything"]
  schedule: '@every 30s'
  check.request.method: "POST"
  check.send: 'Hello World'

# expect a certain body
- type: http
  urls: ["https://httpbin.org/anything/Hello+World"]
  schedule: '@every 30s'
  check.request.method: "POST"
  check.receive: 'Hello World'
```

Configure heartbeat monitors

```
# auth test
- type: http
  urls: ["https://httpbin.org/basic-auth/my_user/my_password"]
  schedule: '@every 30s'
  username: "my_user"
  password: "my_password"
  check.request.method: "POST"

# check response headers
- type: http
  urls: ["https://httpbin.org/response-headers?freeform=my_value"]
  schedule: '@every 30s'
  check.response.headers:
    'freeform': 'my_value'
```

Configure heartbeat monitors

```
# ICMP test against google DNS server
- type: icmp
  schedule: '@every 30s'
  hosts: [ '8.8.8.8' ]

# TCP test against google DNS server
- type: tcp
  schedule: '@every 30s'
  hosts: [ 'tcp://8.8.8.8:53' ]
```

Demo Time!

elastic

Thank You

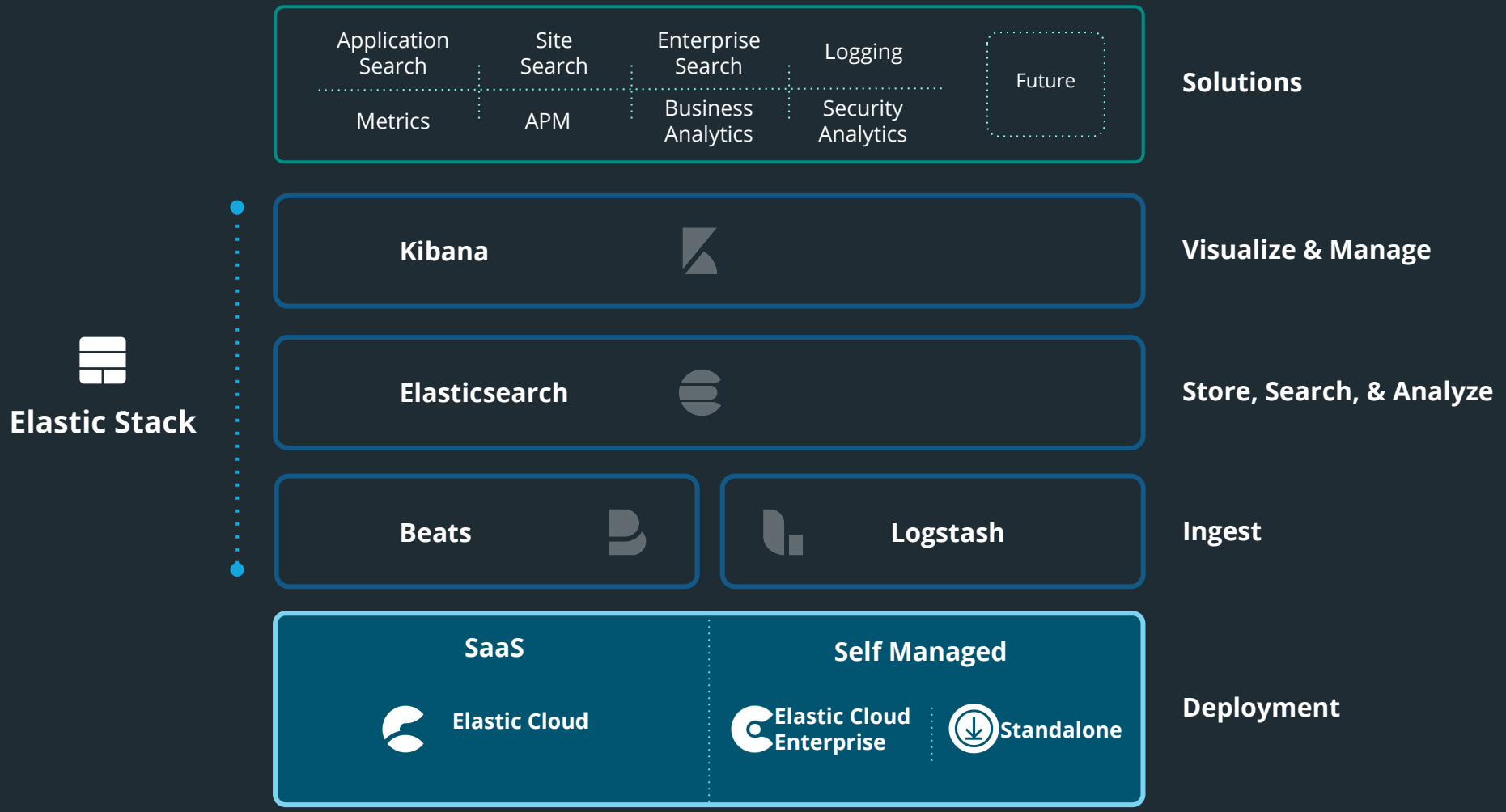
- Web : www.elastic.co
- Demos: demo.elastic.co
- Products : https://www.elastic.co/products
- Forums : https://discuss.elastic.co
- Community : https://www.elastic.co/community/meetups
- Twitter : @elastic

elastic

Questions?

elastic

Thanks Again!!!


Application Search
Site Search
Enterprise Search
Logging
Metrics
APM
Business Analytics
Security Analytics
Future
Solutions
Elastic Stack
Kibana
Visualize & Manage
Elasticsearch
Store, Search, & Analyze
Beats
Logstash
Ingest
SaaS
Elastic Cloud
Self Managed
Elastic Cloud Enterprise
Standalone
Deployment