


Application Monitoring using Datadog

Mukta Aphale
(DevOps Practice Head, WhiteHedge Technologies)

12 Dec 2015, DevOps Meetup, Pune

WhiteHedge Technologies

Started 2003 | Focused Agile Product Development

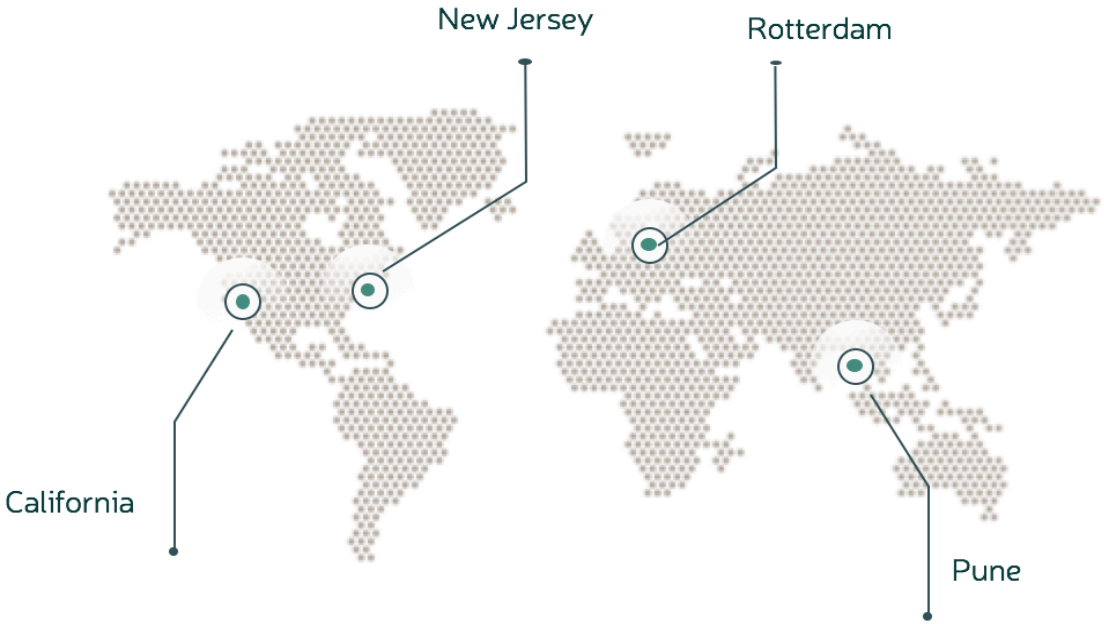
Envision Products | Convert into businesses

100+ employees | 50+ live products world-wide

The best of the Talent and Infrastructure

Self funded | Well funded | Profitable

Global Presence



More about us ...

What defines us ?

- Agile + Flexible
- Thorough + Quick Learner
- Competitive + Comprehensive
- Honest + Transparent
- Young + Mature
- Innovative + Creative

Application Monitoring

- Ensure that a software application processes and performs in an expected manner and scope

APM

Runtime Metrics

Health of the application

Health of infrastructure

Provide system/application feedback

Continuous Improvement

Datadog

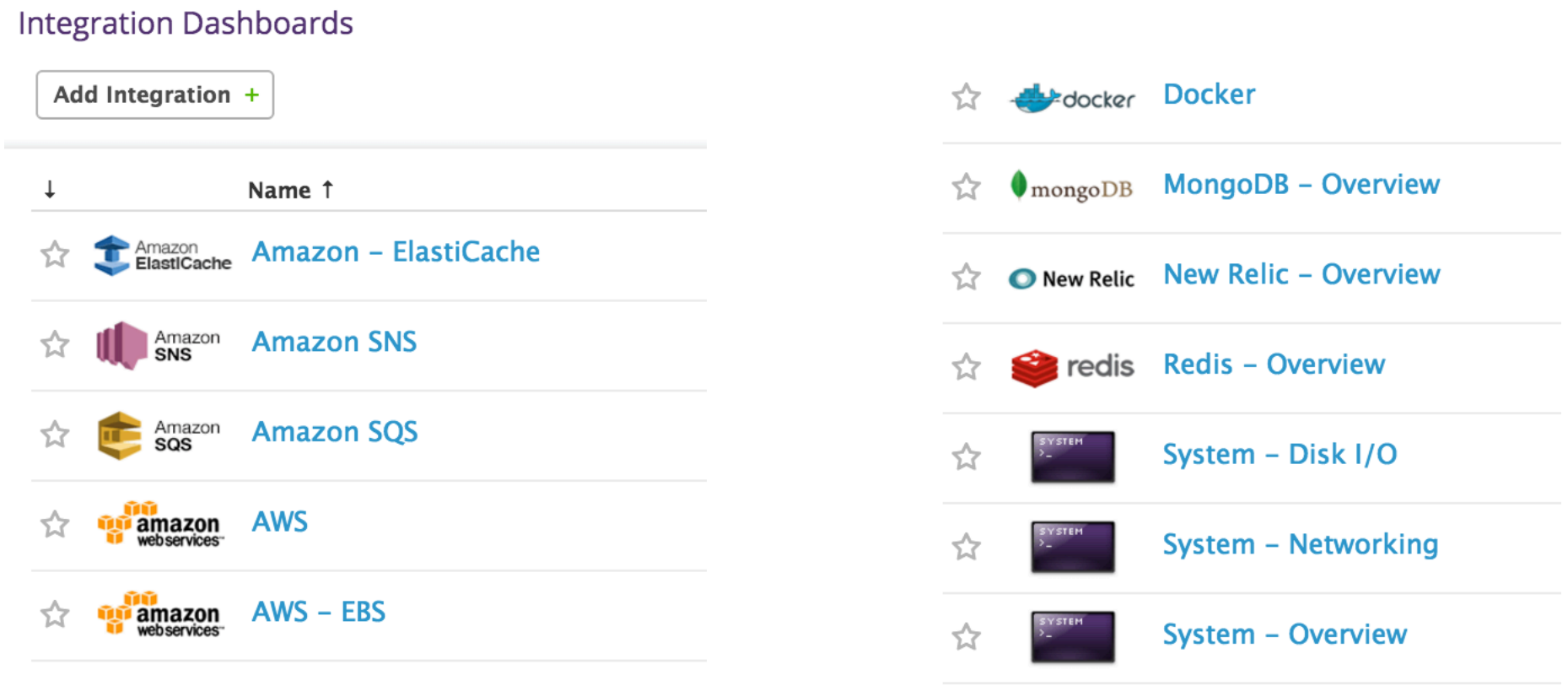
- Monitoring as a Service
- Agent Based
- Python
- Integrations
- Dashboards
- Tagging
- Alerts
- Checks

Environment

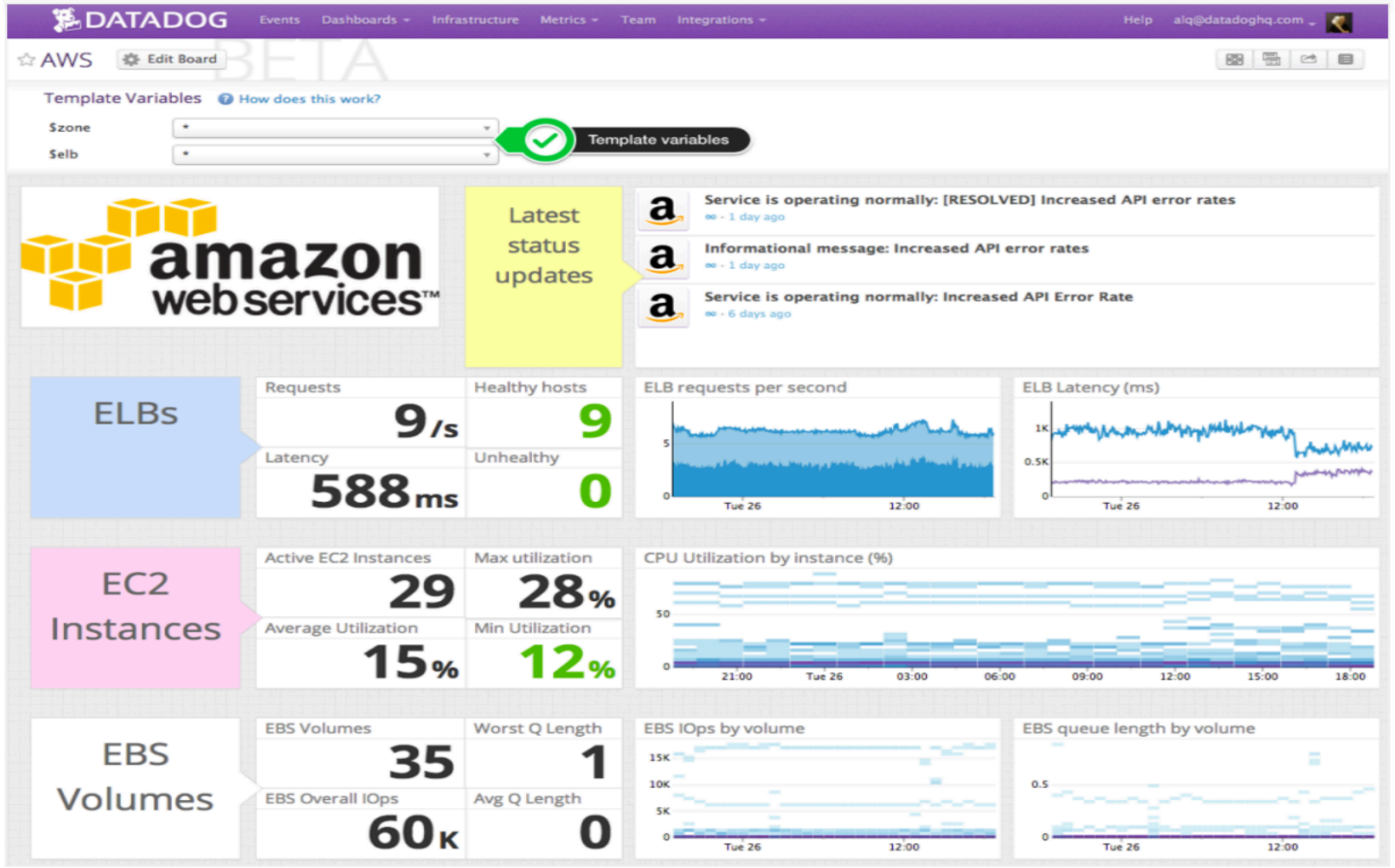
Challenge

- Monitor System Health
- Monitor Redis, MongoDB
- Application (API Server) runs as docker container
- Monitor application performance
- Logging slowed down performance
- Rapid development, No feedback mechanism
- Insight in application needed by management
- Insight needed by support, devops and developers

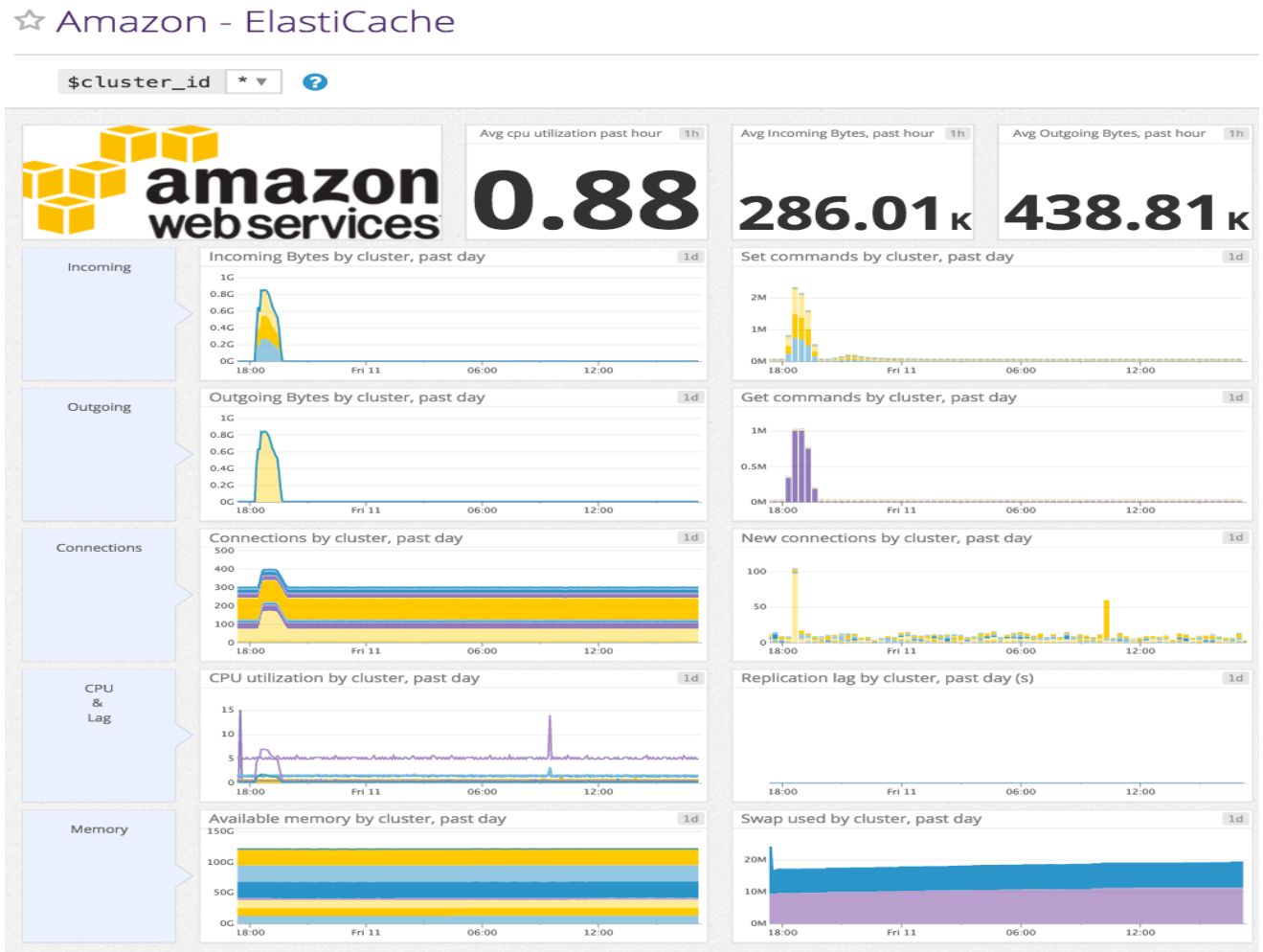
Integration Dashboards



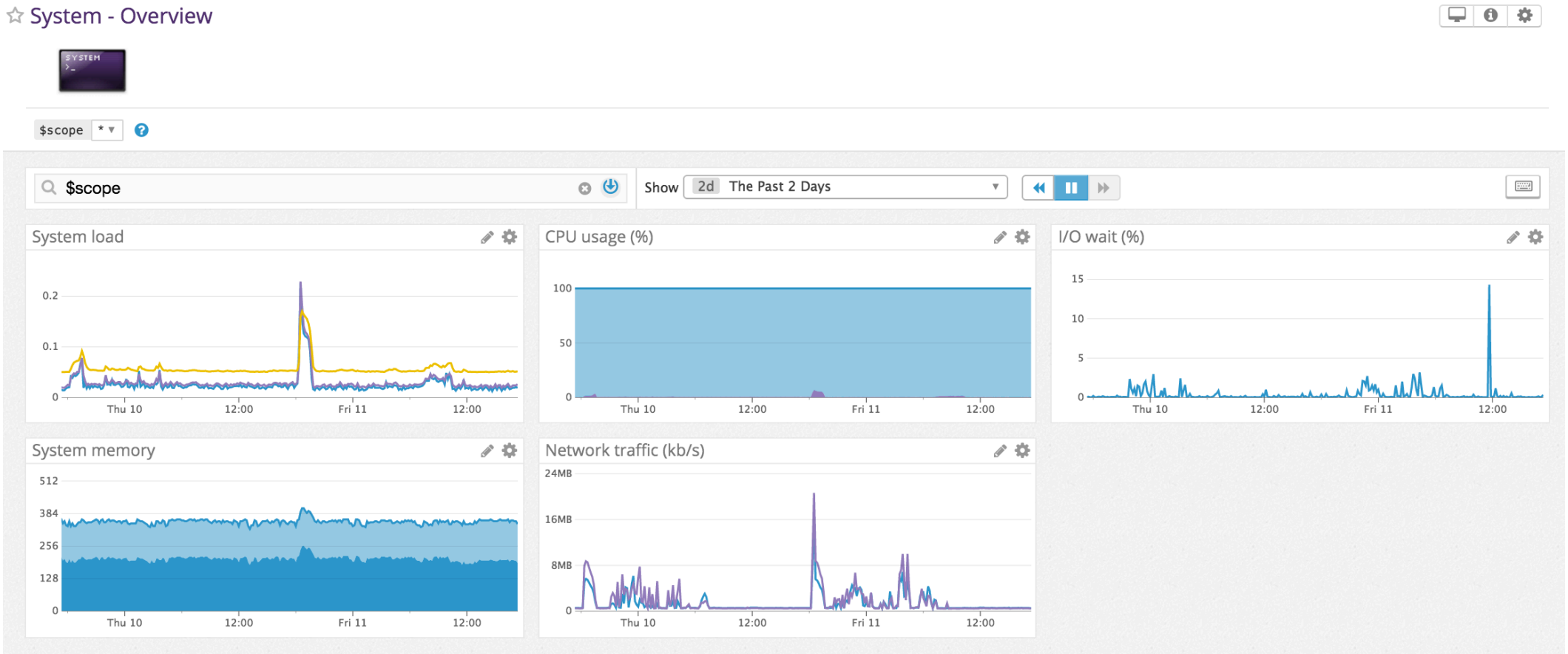
AWS


DATADOG
Events
Dashboards
Infrastructure
Metrics
Team
Integrations
Help
alq@datadoghq.com
AWS
Edit Board
BETA
Template Variables
How does this work?
$zone
$elb
Template variables
amazon web services™
Latest status updates
Service is operating normally: [RESOLVED] Increased API error rates
1 day ago
Informational message: Increased API error rates
1 day ago
Service is operating normally: Increased API Error Rate
6 days ago
ELBs
Requests
9/s
Healthy hosts
9
Latency
588ms
Unhealthy
0
ELB requests per second
ELB Latency (ms)
Tue 26
12:00
EC2 Instances
Active EC2 Instances
29
Max utilization
28%
Average Utilization
15%
Min Utilization
12%
CPU Utilization by instance (%)
21:00
Tue 26
03:00
06:00
09:00
12:00
15:00
18:00
EBS Volumes
EBS Volumes
35
Worst Q Length
1
EBS Overall IOps
60K
Avg Q Length
0
EBS IOps by volume
EBS queue length by volume

Amazon - ElastiCache



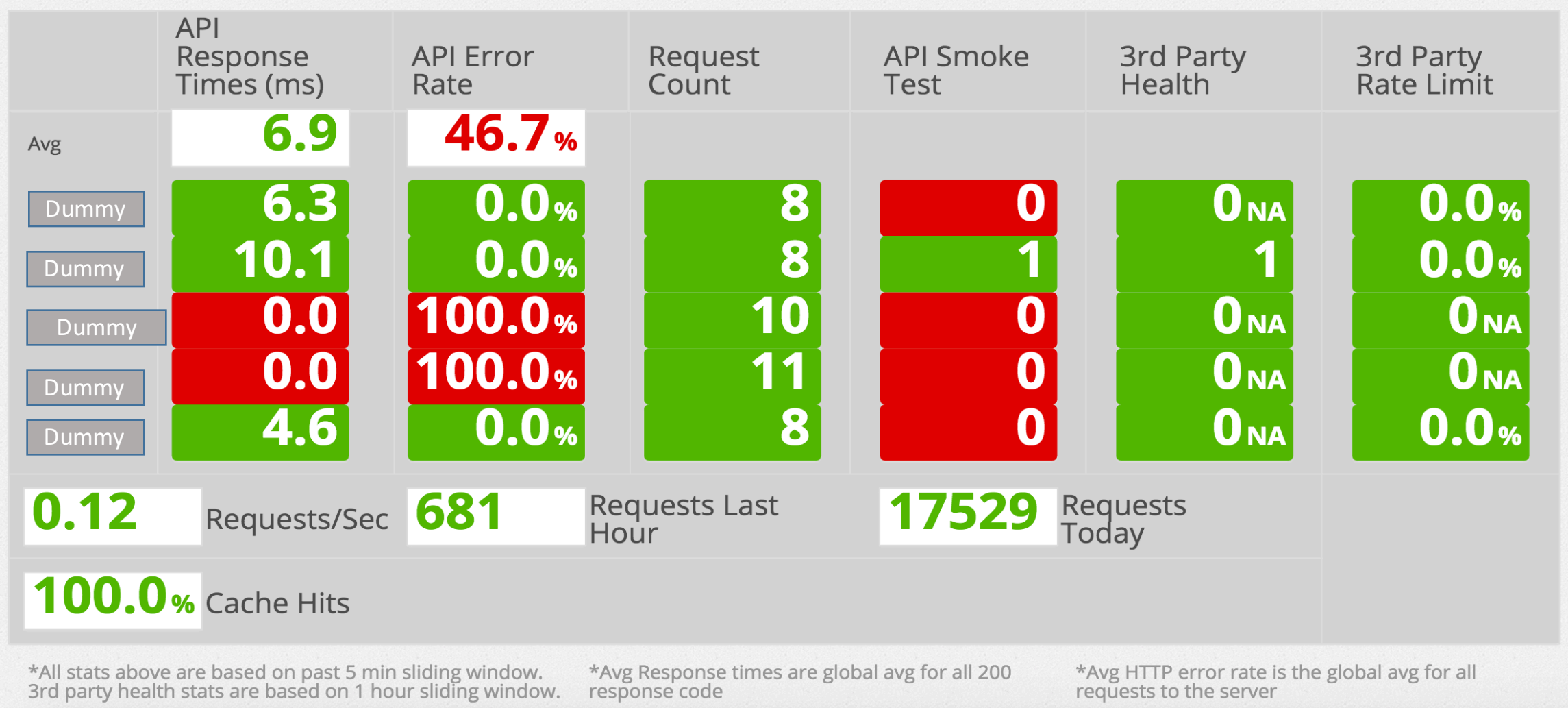
System Overview



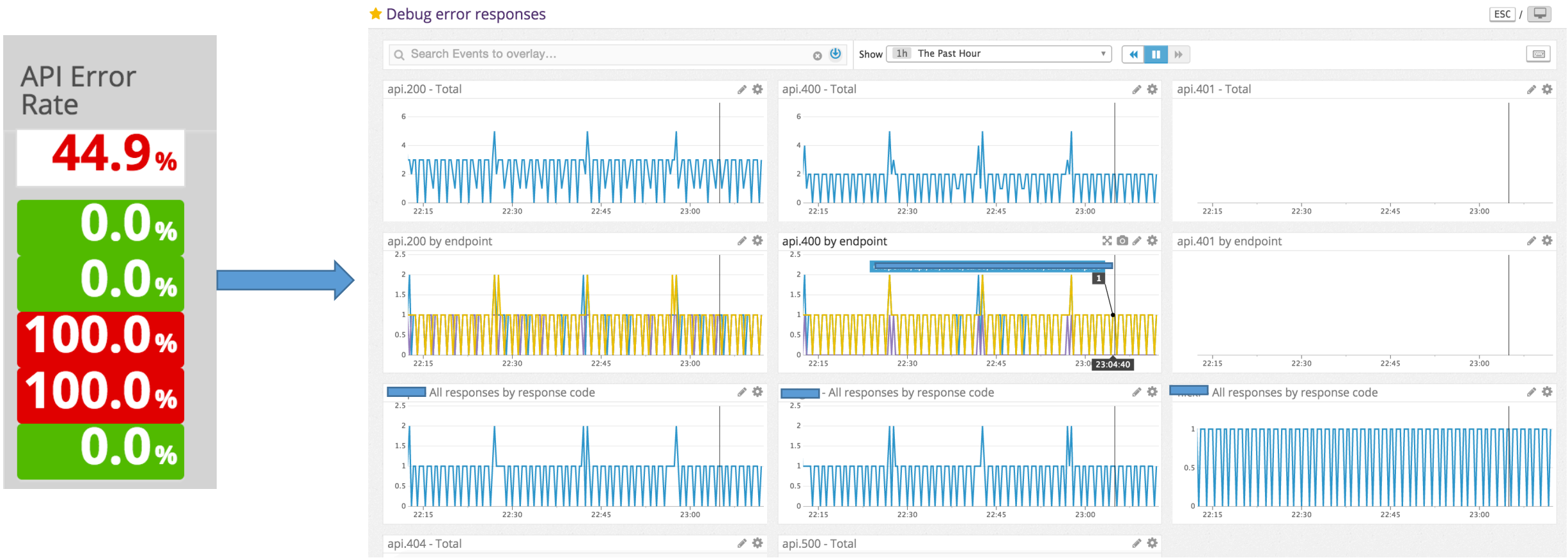
But…

- How to impress the management?! ☺
- How to deal with X environments and Y versions?
- How to measure performance, without affecting performance?

Overview Dashboard



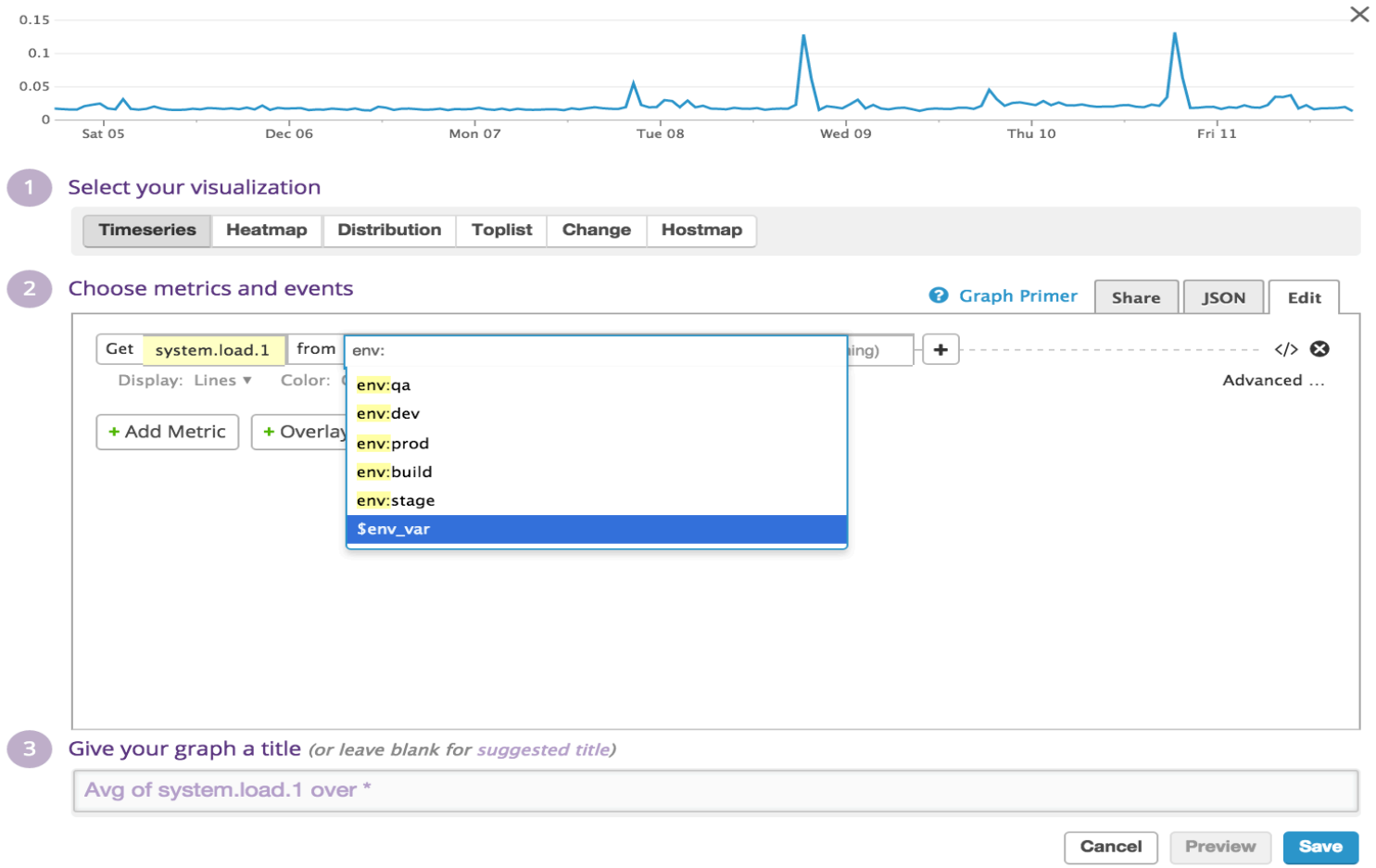
Debug: API Error Rate



Which endpoint is having more errors? Which error codes are being thrown? Search logs in Loggly if needed.

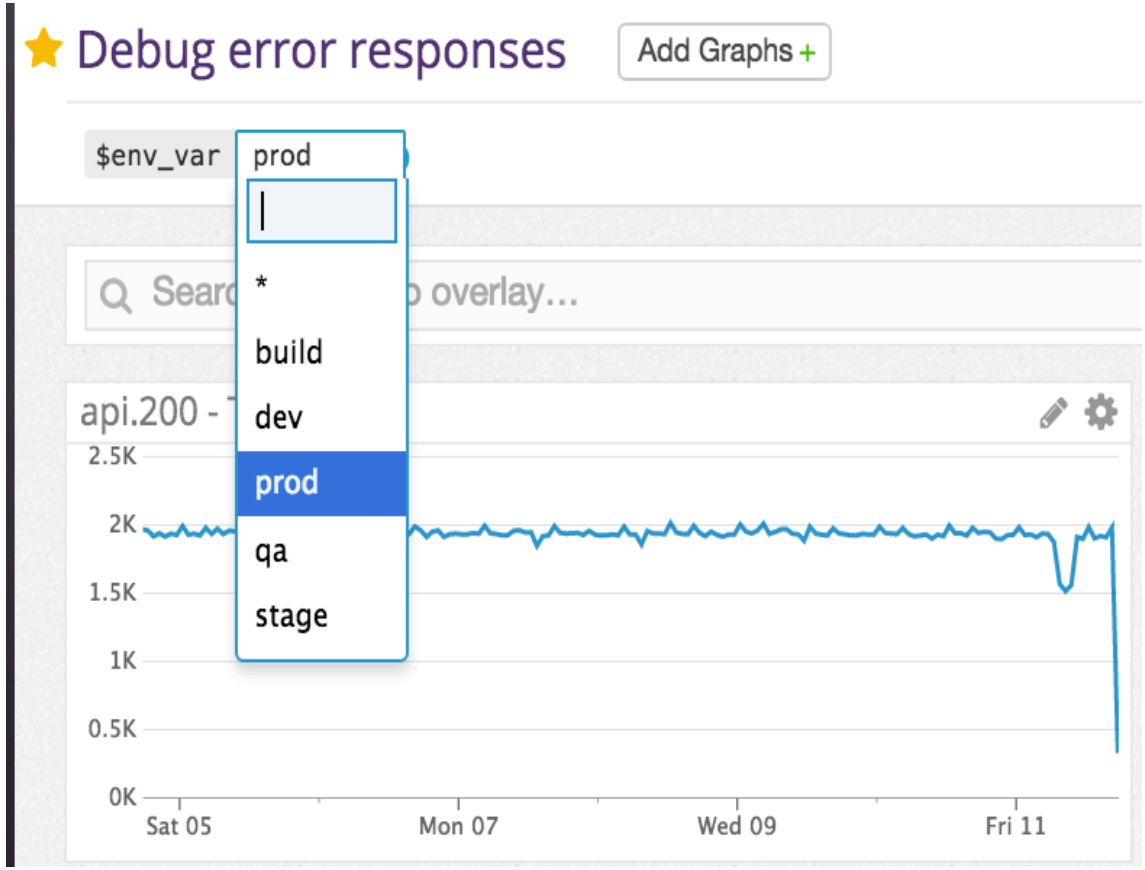
Datadog Tags

- Inherited from Integrations
- Custom tags



Templated Dashboards

- Dashboard variables
- Dynamically explore metrics



Datadog API

- We can code:
  - Instance configuration
  - Infrastructure
  - Deployments
- Why not monitoring?!
- Datadog has great API

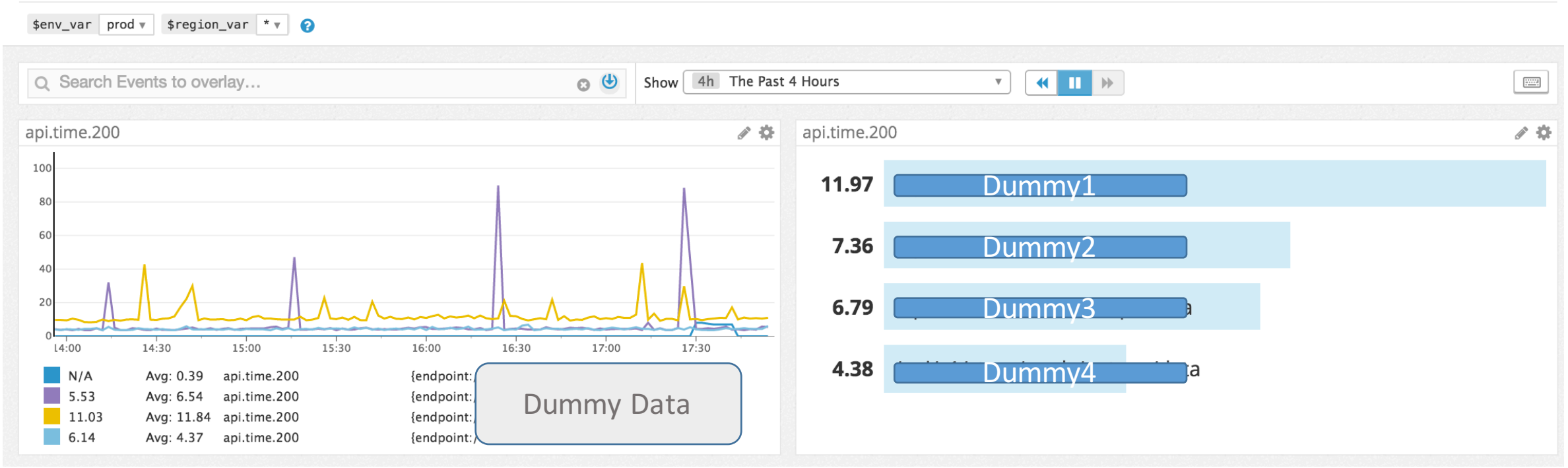
Datadog module for nodejs

```
1   var StatsD = require('node-dogstatsd').StatsD;
2   var dogstatsd = new StatsD();
3
4   var tag1 = "env:" + process.env.NODE_ENV;
5   var tag2 = "endpoint:dummy";
6
7   dogstatsd.increment("task.hits", 1, [ tag1, tag2 ]);
```

Measuring http response times from application code

```
1   if (!(api.config.env.isLocal || api.config.env.isTest)) {
2       var tag1 = "response_code:" + raw.responseHttpCode;
3       var tag2 = "env:" + process.env.NODE_ENV;
4       var tag3 = "endpoint:" + raw.parsedURL.pathname;
5
6       var metric = raw.parsedURL.pathname;
7
8       // Send API metrics (count)
9       metric = "api." + raw.responseHttpCode;
10      dogstatsd.increment(metric, 1, [ tag3, tag2 ]);
11
12      // Send response times
13      metric = "api.time." + raw.responseHttpCode;
14      dogstatsd.gauge(metric, data.duration, [ tag2, tag3 ]);
15  }
```

Plot it!


$env_var prod
$region_var *
Search Events to overlay...
Show 4h The Past 4 Hours
api.time.200
100
80
60
40
20
0
14:00
14:30
15:00
15:30
16:00
16:30
17:00
17:30
N/A Avg: 0.39 api.time.200 {endpoint:
5.53 Avg: 6.54 api.time.200 {endpoint:
11.03 Avg: 11.84 api.time.200 {endpoint:
6.14 Avg: 4.37 api.time.200 {endpoint:
Dummy Data
api.time.200
11.97 Dummy1
7.36 Dummy2
6.79 Dummy3
4.38 Dummy4

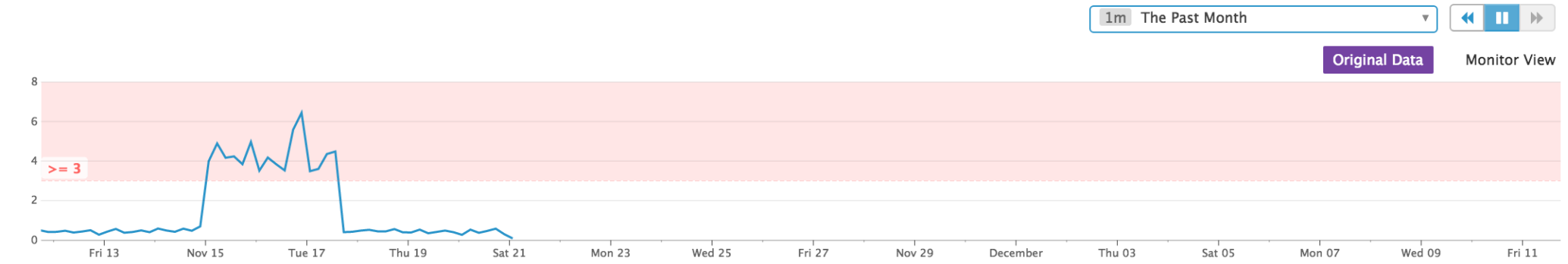
What about performance while tracking performance?

- Metrics from code are sent to local datadog agent using UDP
- Local datadog agent syncs the metrics to the datadog server
- Datadog dashboard reflects the metrics with some delay
- Application performance does not get affected

Alerts

API Server is responding very slowly on PROD env on {{host.name}}

Edit | Status



1 Define the metric

Source | Edit

Define the monitor's query.

```
(avg:api.time.400{prod} + sum:api.time.401{prod} + avg:api.time.402{prod}  + avg:api.time.409{prod} + avg:api.time.200{prod} + avg:api.time.500{prod} + 0) / 1000
```

2 Set alert conditions

Threshold Alert | Change Alert — An alert is triggered whenever a metric crosses a threshold.

Trigger when the metric is [above or equal to] the threshold [in total] during the last [5 minutes]

Alert threshold: 3 (3)

Datadog Checks

- Collect metrics from datadog agent check
- Out of the box agent checks
- Custom agent checks in Python
- Interesting use cases:
  - Keep alive check (Service is up)
  - Network Check (HTTP, TCP)
  - Validate response for expected data

Using Datadog we could...

- Monitor dynamic infrastructure
- Monitor system health
- Monitor application availability
- Monitor application performance
- Show application & infra health graphically
- Provide feedback about health of system

Thank You!

Questions?

We are Authorized Datadog Partners!

You can write to me at:
maphale@whitehedge.com
Twitter: @muktaa