

Scaling monitoring with Datadog



Agenda

- . **What** is Datadog?

- . **How** does it help me **scale** monitoring?

Thank you #\$\$%! Peter and Denise...



Thank you Dear Sponsor !



STACKDRIVER

Thank you Dear Host !



whoami

Service
Source®

Alexis Midon

Backend Engineer

a year in DevOps, by accident

The logo for Renew OnDemand features a stylized white cloud icon above the word "Renew" in a bold, sans-serif font. Below "Renew" is the word "OnDemand" in a smaller, lighter font. A trademark symbol (TM) is positioned to the upper right of "Renew".

Renew™
OnDemand

environment



What is Datadog?



Datadog

- . Monitoring **service**
- . agent based
- . integrated with **AWS**
- . resource tagging

Datadog - cont'd

- . Metrics and Alerts
- . Correlation features
- . Collaboration features
- . Custom Dashboards

Event Stream

Show 4h Mar 2, 10:20PM - Mar 3, 2:20AM

SHOW

All

What I'm Following

FROM

All

- Amazon Web Services
- Chef
- Datadog
- Feed
- HAProxy
- Jenkins
- Metric Alert
- My Apps
- Nagios
- Pagerduty
- Users
- Webmetrics

PRIORITY

All

Normal

Low

22:24 7 events

22:48

23:12

23:36

0:00

68 matching events from Mar 2, 10:20PM - Mar 3, 2:20AM

Service Source

Leave a status update...

N Nagios Check Node.js is critical on `APP-i-5fb69e7f--i-5fb69e7f`

CRITICAL - Socket timeout after 10 seconds

7 hours ago · [Add comment](#) · [Lower priority](#) · [Follow thread](#) [Claim Incident](#)

38 events (8 in timeframe)

N Nagios Mongo Free Connections on `DB-S8-RS6--i-4d828e25`

Service appears to have started flapping (20.0% change >= 20.0% threshold)

8 hours ago · [Add comment](#) · [Lower priority](#) · [Follow thread](#)

23 events (6 in timeframe)

N Nagios haproxy is ok on `LBWEB-i-ff7019d1--i-ff7019d1`

HAPROXY OK - nginx (Active: 1/1) nginx-insecure (Active: 0/1) gateway_ext (Active: 0/1) node (Active: 8/8)

8 hours ago · [Add comment](#) · [Lower priority](#) · [Follow thread](#)

8 events

N Nagios Check Node.js is ok on `APP-i-04dcf424--i-04dcf424`

HTTP OK: HTTP/1.1 200 OK - 483 bytes in 0.007 second response time

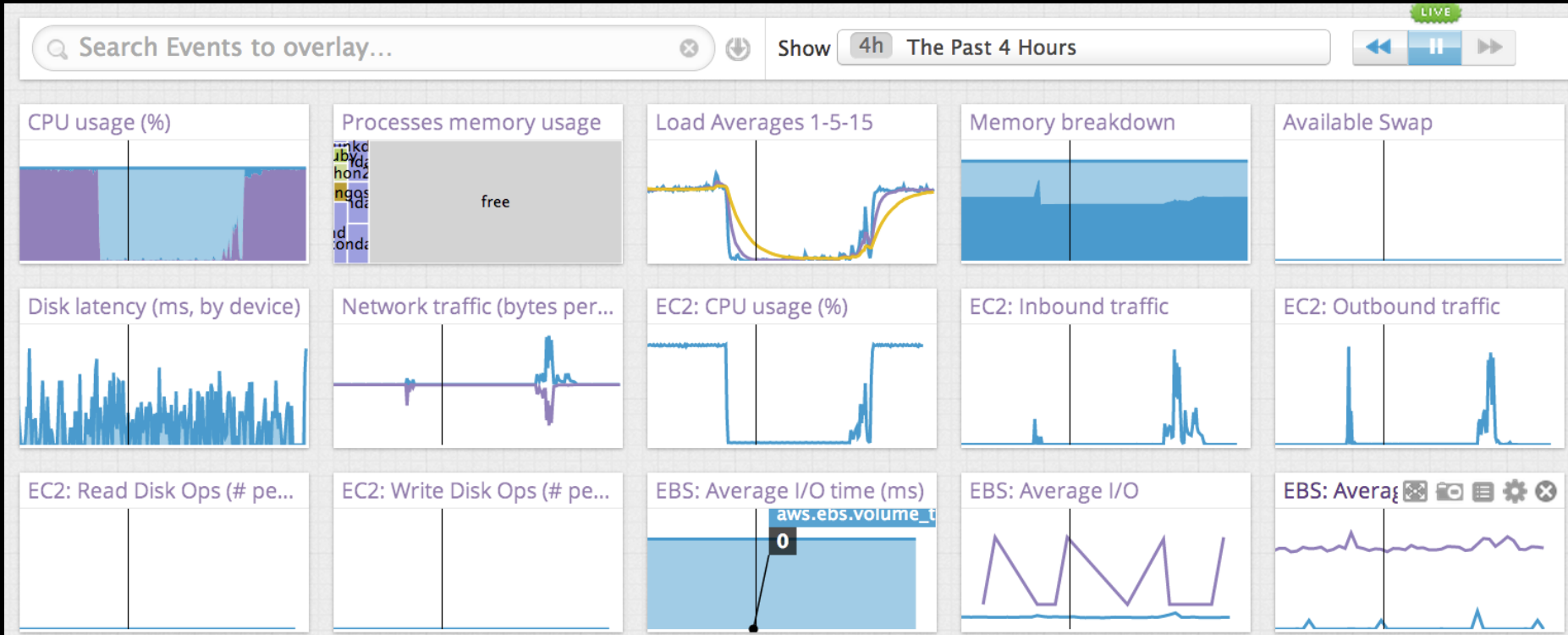
8 hours ago · [Add comment](#) · [Lower priority](#) · [Follow thread](#)

PRAGER DUTY Resolved: Webmetrics Alert `#service:product-support-oncall`

[see the full incident log](#)

9 hours ago · [Add comment](#) · [Lower priority](#) · [Follow thread](#) [Solved by](#)

Default Instance Dashboard



BETA

Template Variables [How does this work?](#)

Szone

Selb

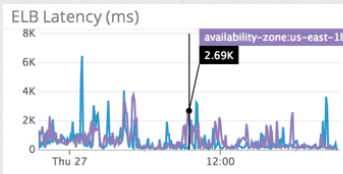
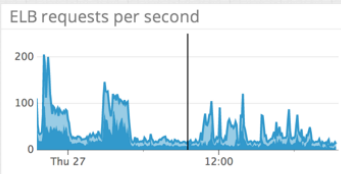


Latest status updates

ELBs

Requests	200/s
Latency	4047 ms

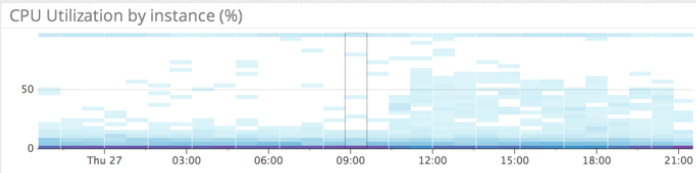
Healthy hosts	54
Unhealthy	0



EC2 Instances

Active EC2 Instances	146
Average Utilization	11%

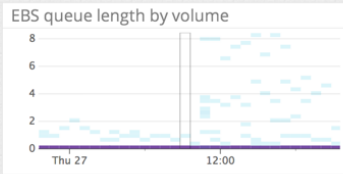
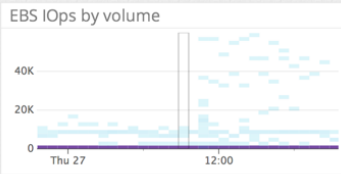
Max utilization	26%
Min Utilization	5%



EBS Volumes

EBS Volumes	158
EBS Overall IOps	372K

Worst Q Length	16
Avg Q Length	0



nice, but...



How do you help me deal with:

N components: mongo, redis, nodejs, ...

x P environments: prod-1, prod-2, staging, ...

x Q versions: app-blue, app-green, etc

x R users

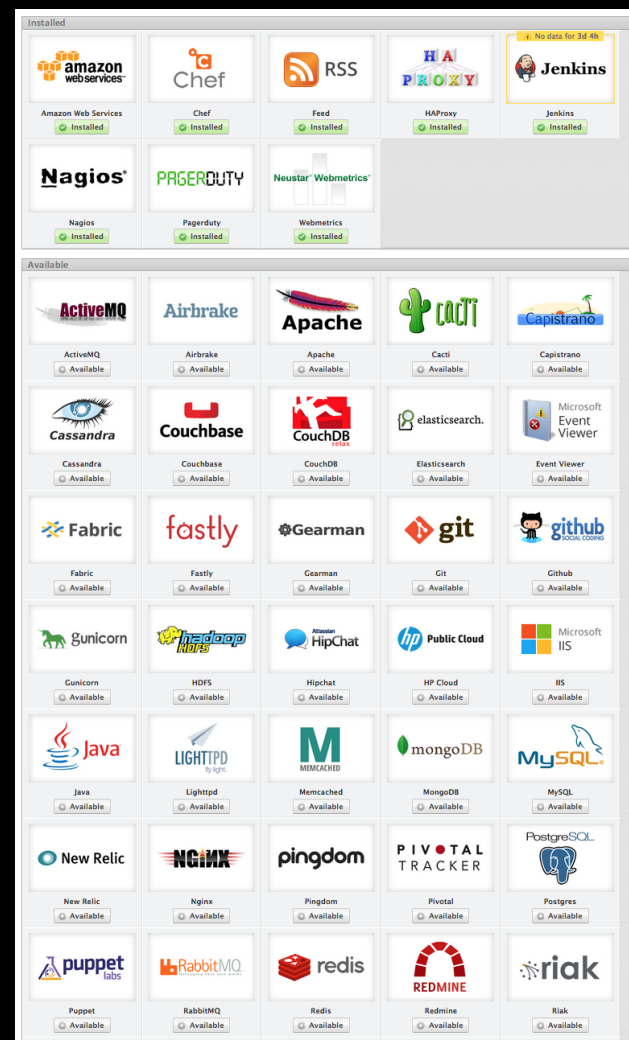
help me scale!

How does Datadog help?

#1 pre-canned tools

pre-canned integrations

pre-canned dashboards



#2 Templated Dashboards

A dashboard can have multiple **variables**.

Edit once, and **re-use**.

```
$environment $zone $tier $asg ...
```

a template example

The screenshot shows a Datadog dashboard for an AWS environment. The top navigation bar includes the Datadog logo, menu items for Events, Dashboards, Infrastructure, Metrics, Team, and Integrations, and a user profile for alq@datadoghq.com. The dashboard is titled 'AWS' and features a 'Template Variables' section with dropdown menus for '\$zone' and '\$elb', both set to '*'. A green checkmark icon and a 'Template variables' button are visible next to these dropdowns.

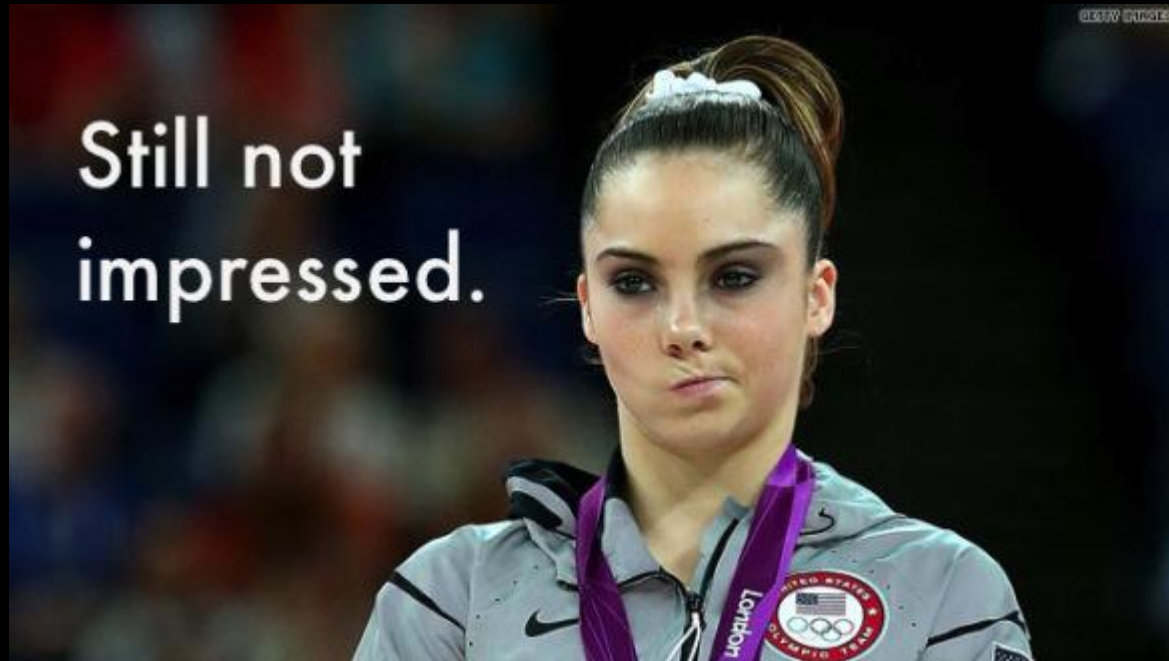
The main content area is divided into several sections:

- amazon web services™** logo and a yellow box labeled 'Latest status updates'.
- A list of three status messages from Amazon, all related to 'Increased API error rates'. The first message is resolved and dated 1 day ago. The second is informational and dated 1 day ago. The third is a warning dated 6 days ago.
- ELBs** (Elastic Load Balancing) section with a blue header. It contains a table of metrics and two line charts.
- EC2** (Elastic Compute Cloud) section with a pink header, containing a table of metrics and a bar chart.

Metric	Value
Requests	9/s
Latency	588ms
Healthy hosts	9
Unhealthy	0
Active EC2 Instances	29
Max utilization	28%

The ELB charts show 'ELB requests per second' and 'ELB Latency (ms)' over time. The EC2 chart shows 'CPU Utilization by instance (%)' as a horizontal bar chart.

not bad but



Gimme API !!!

We can code:

- . instance configuration
- . infrastructure

Why not monitoring?

#3 Datadog has a great API.

- . events, metrics, event, tags, dashboard, alerts, ...
- . bindings for python, ruby, node.js, etc
- . command-line

plain json + curl

```
{
  "template_variables": [
    "ai",
    "sb"
  ],
  "title": "by Stack",
  "description": "Environment Metrics by Stack",
  "graphs": [
    {
      "definition": {
        "events": [],
        "requests": [
          {
            "q": "avg:aws.ec2.cpuutilization{$ai,stack:app}"
          },
          {
            "q": "avg:system.linux.memory_utilization{$ai,stack:app}"
          },
          {
            "q": "avg:system.disk.in_use{$ai,stack:app}"
          }
        ]
      }
    }
  ],
}
```

using the **ruby gem**

```
title = 'by Stack'
description = 'Environment Metrics by Stack'
template_variables=['ai']

stacks = %w{app cube db dbc lbweb redis seca secc}
graphs = stacks.map do |stack|
  {
    "definition" =>
      {
        "events" => [],
        "requests" => [
          {
            "q" => "avg:aws.ec2.cpuutilization{$ai,stack:#{stack}}"
          },
          {
            "q" => "avg:system.linux.memory_utilization{$ai,stack:#{stack}}"
          },
          {
            "q" => "avg:system.disk.in_use{$ai,stack:#{stack}}"
          }
        ],
        "viz" => "timeseries"
      },
    "title" => "#{stack.upcase} - CPU and Memory Utilization (%)"
  }
end
dog.create_dashboard(title, description, graphs, template_variables)
```

I have the power!



SodaHeadGraphics.com

Now, I can:

- . **version control** my dashboards, alerts
- . **code** my monitoring resources
- . **integrate** with my provisioning tool

Integration example:

CloudFormation++

CloudFormation++

- . a CFN template usually has related dashboards and alerts.
- . same life-cycle

e.g. app tier:

- . dashboards for ELB and front-end instances
- . alerts on HTTP errors, etc

CloudFormation++

stack = CloudFormation + Datadog

```
$ rake stack:app:create
```

```
executing stack:app:cloudformation:create
```

```
executing stack:app:datadog:create
```

in git

```
/stacks
```

```
  /app
```

```
    app_cfn_template.json
```

```
    app_datadog.rb
```

```
    app_http_alerts.json
```

be creative

datadog.rb is evaluated in a rich context.
It has access to everything.

very flexible.

Cons / Pain points :-)

- . still have to deal with some json
- . room for drift - if users manually edit resources
- . resource tracking can be tricky

Pros

- . monitoring has code
- . all the benefits of using code:
 - tests, versioning, tracking, DRY, bugs, ...

Summary

Go code your Monitoring,
with the awesome Datadog API.

impressed ?



Thank you!



Scaling monitoring with Datadog

