

The courtship dance of product discovery



● 2016–Present

Datadog

● 2015–2016 (Co-founder & CPO)

Opsee (Heavybit)

● 2013–2015 (Acq. by New Relic)

Opsmatic

● 2011–2013 (Acq. by BMC)

Boundary

Datadog

EMPLOYEES

2,000+

CUSTOMERS

DevOps

FOUNDED

2010

BUSINESS

SaaS



Let's start with the obvious

- Dev tools companies often founded by devs
- Many dogfood their own products
- Early on your product does one thing and you're really close to it
- **This is what I love as a designer**
- Engineers who care about the UX since they are themselves users
- Chances are **making software will never feel more personal to you than it does right now**

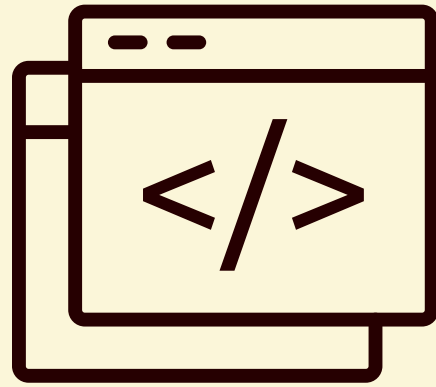
HEAVYBIT
INDUSTRIES

You're already connected with customers

- In the early days you're defining **core assumptions** about your users
- These stay with you essentially forever
- Everything you build connects to them
- You may know these as **Personas**
- I've never liked the dogma around Personas, User Stories, etc—but the need to understand and evangelize these definitions is crucial

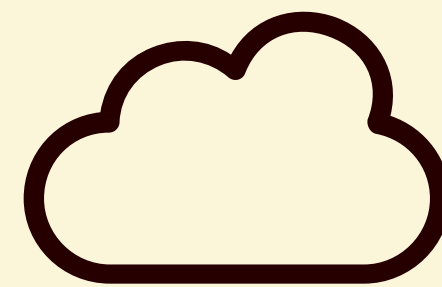


What this looks like at Datadog



DevOps

- Devs managing their own infrastructure
- Breaking down silos



Cloud-native

- Hybrid infrastructure
- Dynamic, ephemeral environments
- Using a wide range of technologies



Trust us

- Trust our expertise
- “Show me the best way to monitor X”
- Look to us for best practices

Recruiting and managing design partners

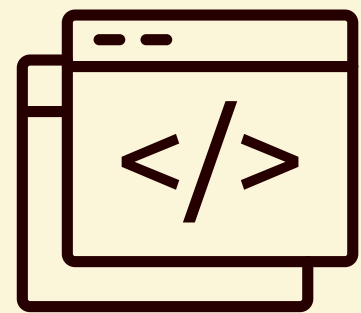
One of our most well-worn processes

1. Identify a small number customers interested in the problem you're trying to solve
 - Big users of product
 - Support tickets about problem
 - CSM knows customer is concerned about problem
2. Recruit the most vocal/helpful as Design Partners
 - These people are special*
3. Show them ideas, mockups
4. Run a private beta, treat all feedback as high-priority
5. If you're on the right track, release it more widely

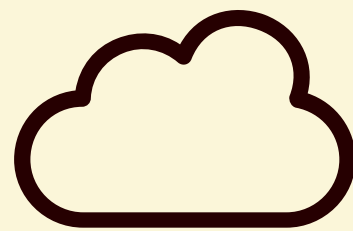
Example: Dashboards

Persona: engineer at a software company

- **User story:** “I need to troubleshoot issues with a service(s) I work on”



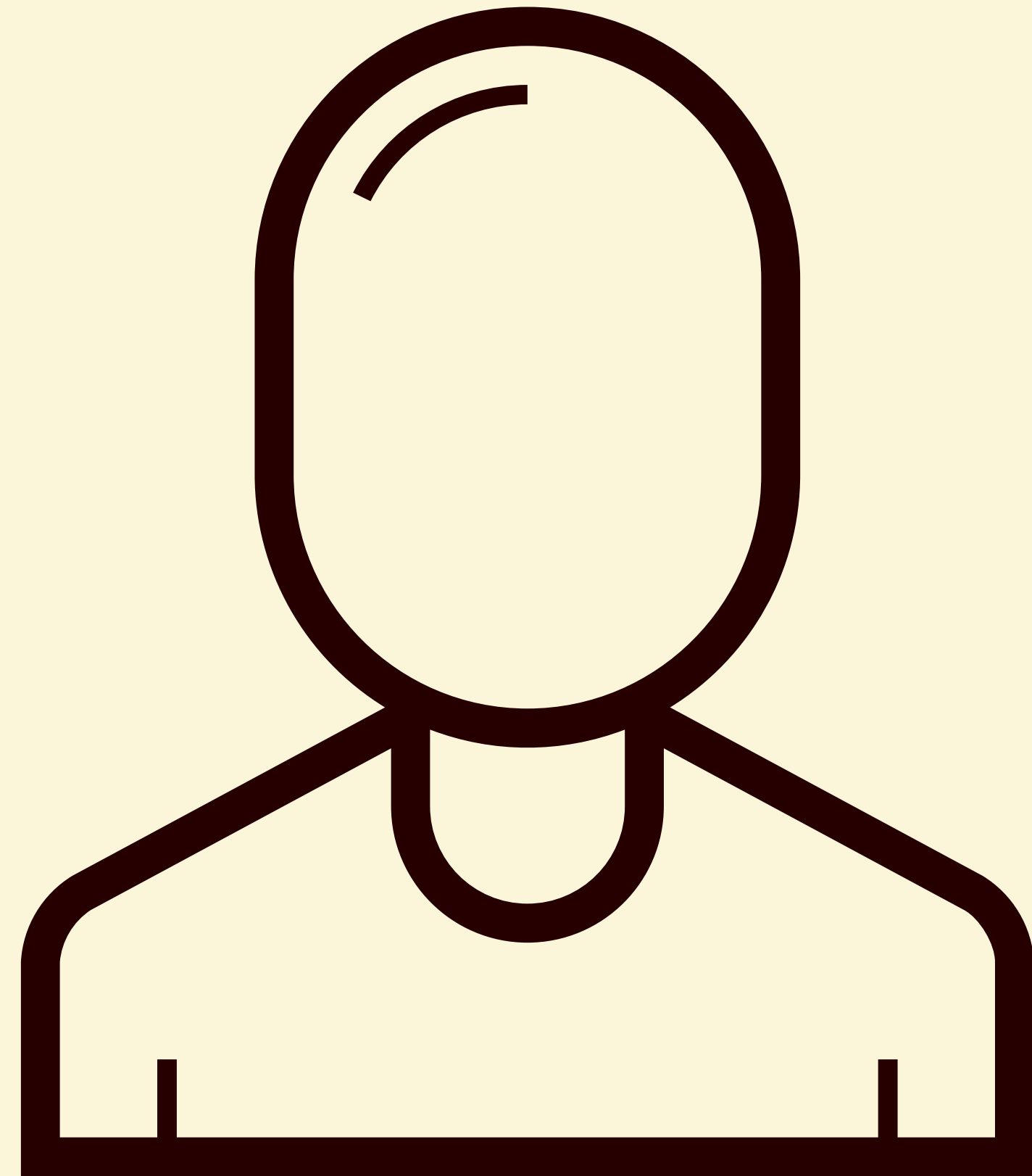
DevOps



Cloud-native



Trust us



\$PSHARD

?

Q Search events to overlay...



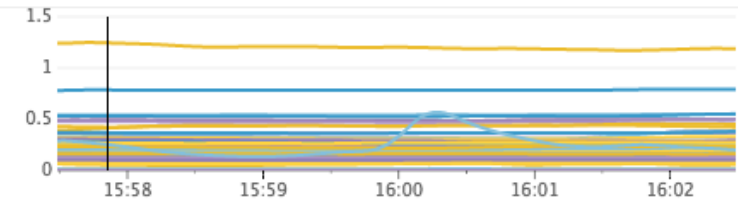
Show

5m

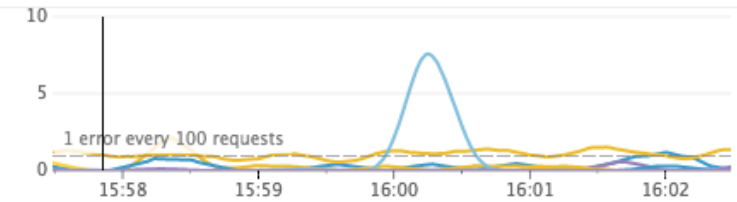
Oct 31, 3:57PM – Oct 31, 4:02PM



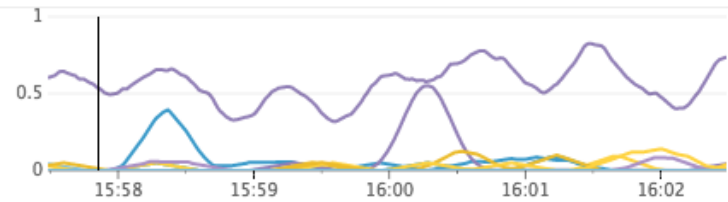
Resp time by haproxy svc & elb (avg)



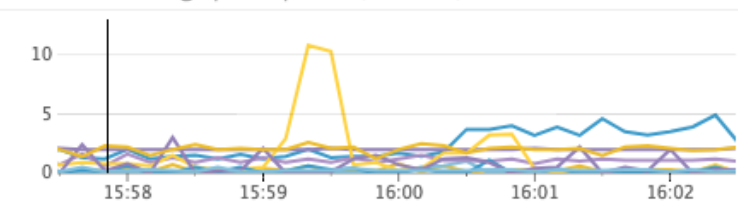
5xx resps by haproxy svc (%)



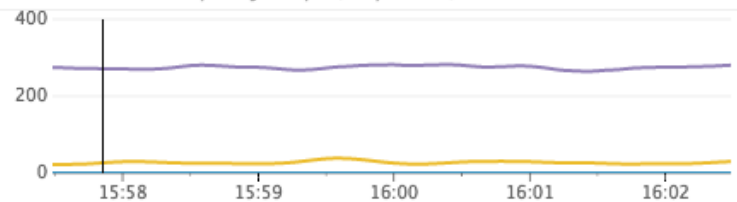
5xx resp by haproxy svc & elb (reqs / sec)



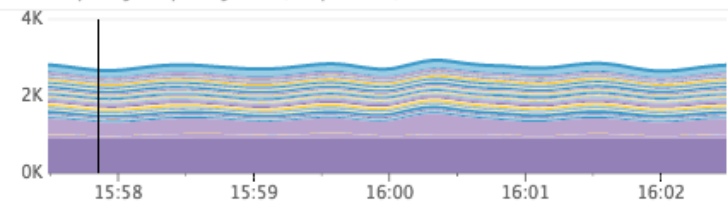
10 most erroring api endpoints (err / sec)



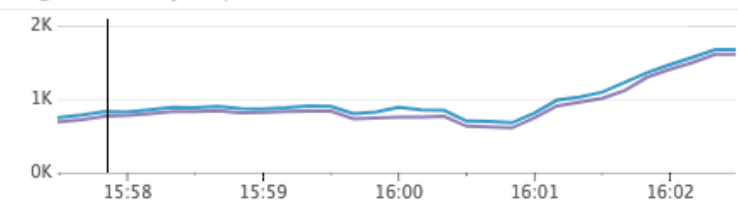
1xx, 3xx, & 4xx haproxy resps (reqs / sec)



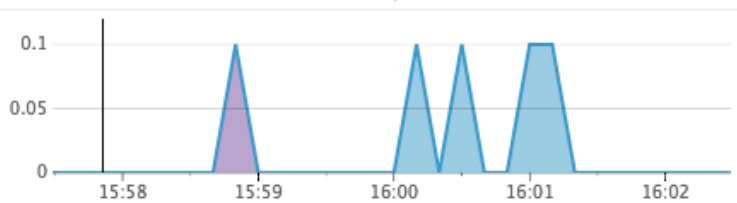
2xx resps by haproxy svc (reqs / sec)



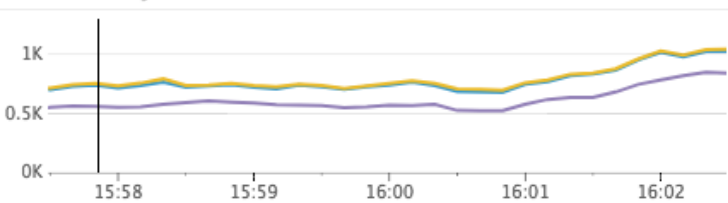
Dogweb latency 95 percentile + Median



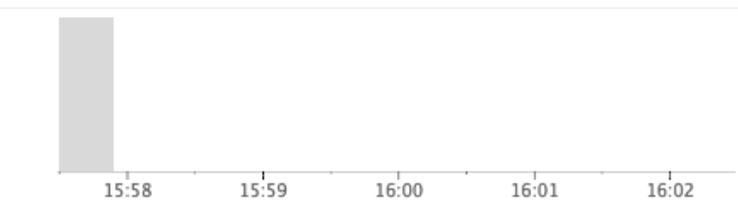
Gunicorn 500, 504 timeout, 512 resps (rate)



Paster latency



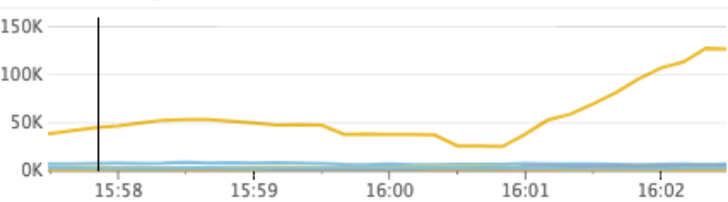
Paster errors



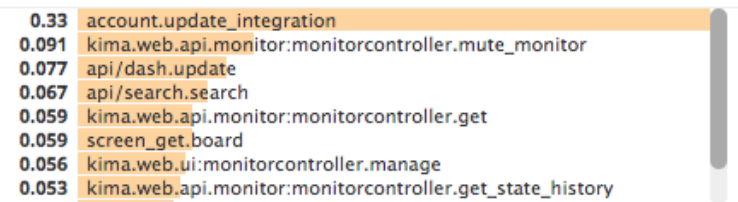
Requests (dogweb stats)



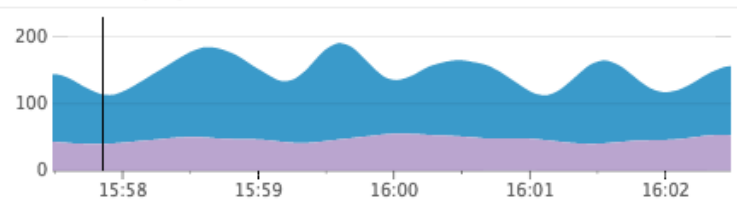
10 Slowest Pages



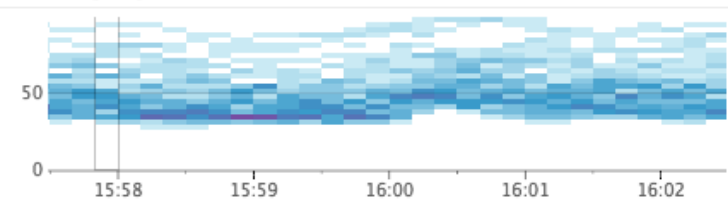
Slow Requests By Page Per Second



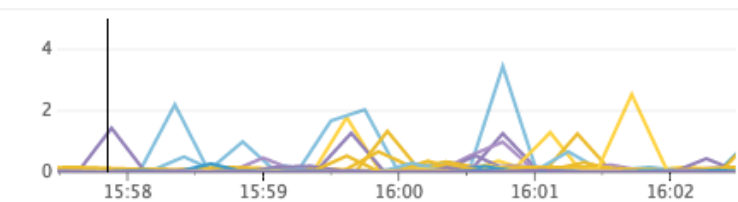
Static file resp by status (rate? count? who knows!)



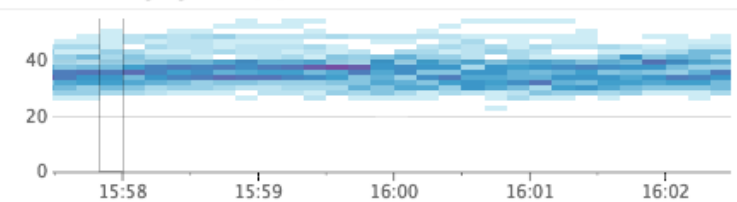
CPU usage by host (%)



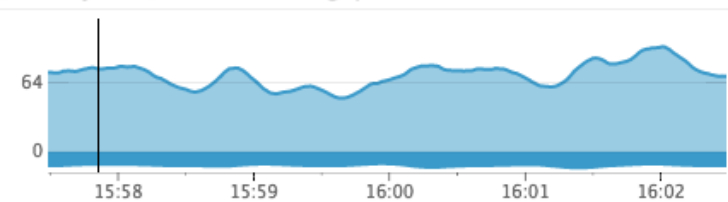
CPU iowait max



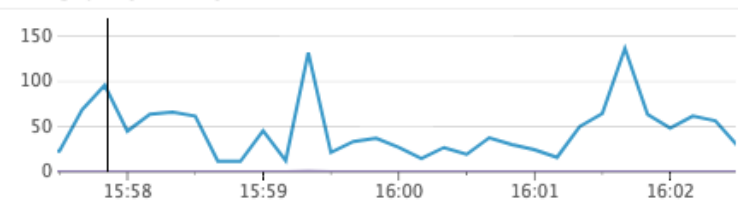
Used memory by host (%)



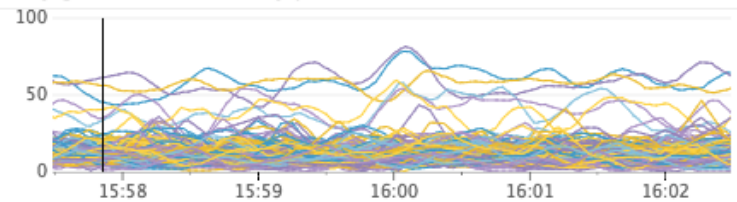
[mcnulty-web] Network throughput



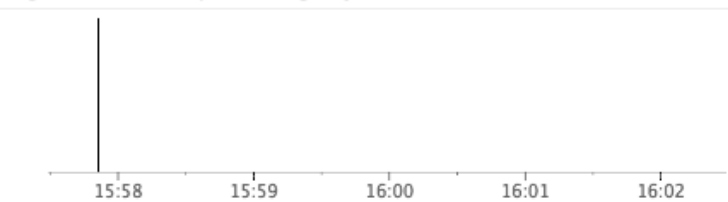
String query latency



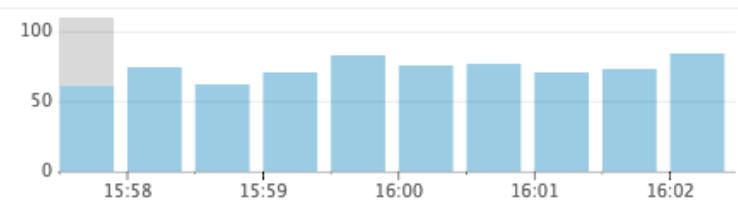
Busy gunicorn workers by pool & az (%)



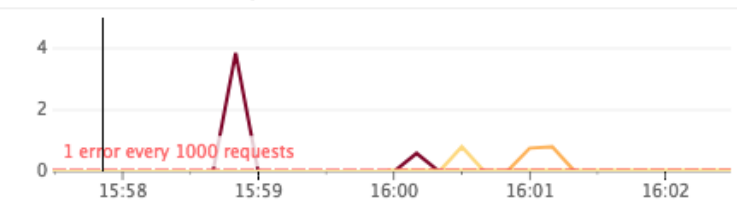
dogweb cache hit percentage by cache



Restarts



Gunicorn errors (% reqs / status / host)



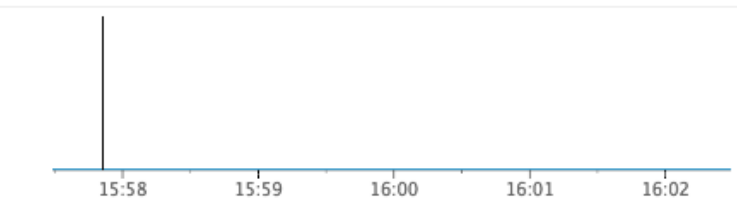
Dogweb errors (rate / host)



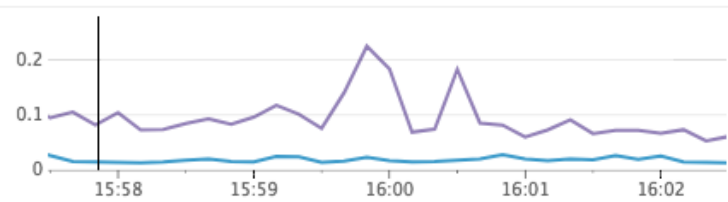
nginx down pages (maybe; i got a down page and didn't see it h...



Query cache errors (rate)



Query cache get (+) & set (TODO -) latency (avg p50 & p95)



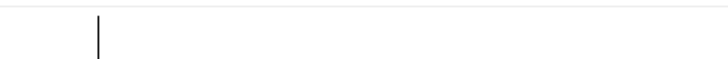
Session get (+) & set (-) latency (avg p50 & p95 ms; cassandra)



Session Storage Latency (ms)- app.datadoghq.com/dash/41814



Frontend index errors (rate)



“I make one dashboard per service”

“I include clusters of widgets for dependent services/infra”

“I design a dashboard as a 2-column matrix”

“I visually cluster and organize widgets”

“I put the most important metrics on top”

Customer quotes

“I make one dashboard per service”

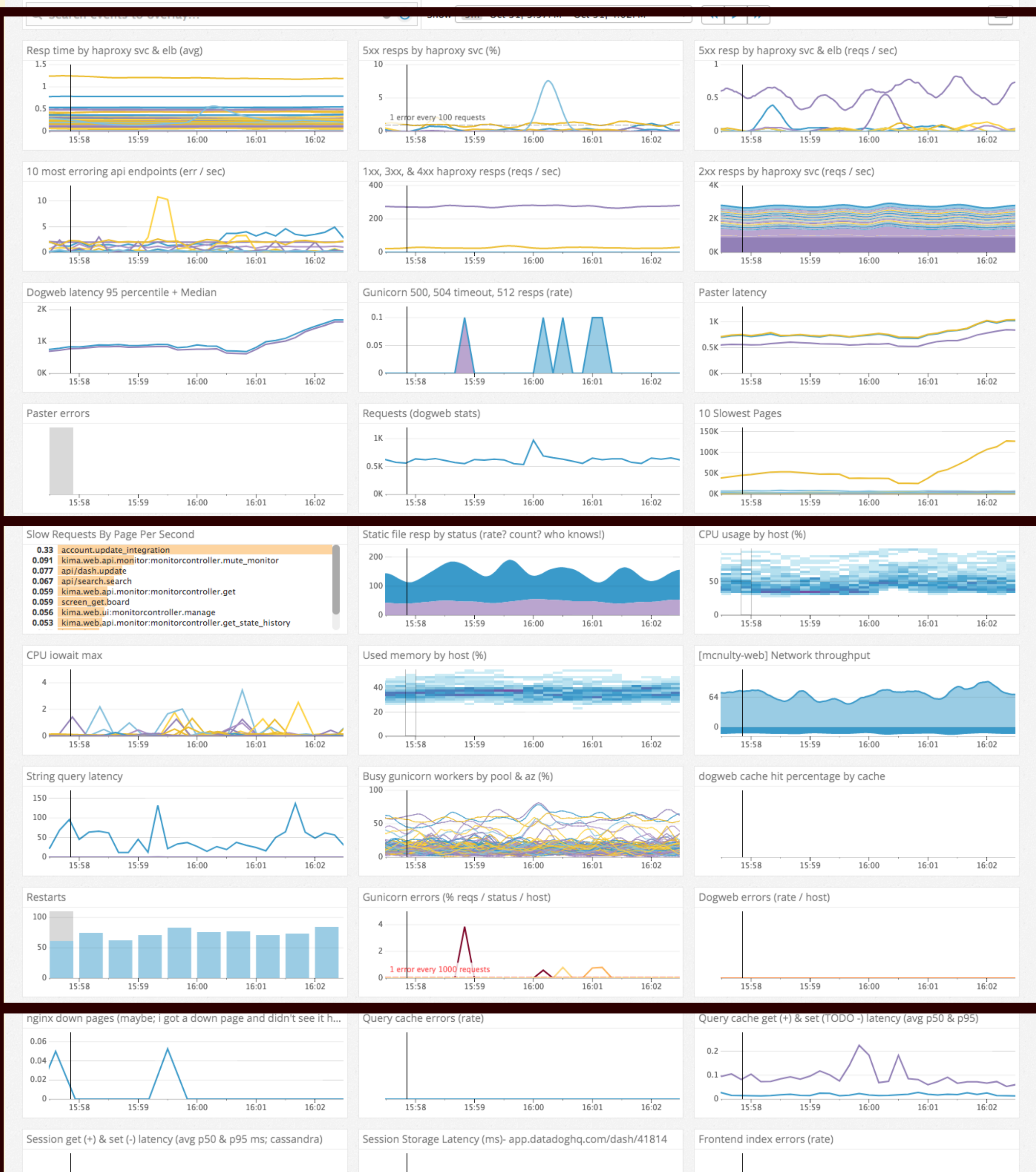
“I include clusters of widgets for dependent services/infra”

“I design a dashboard as a 2-column matrix”

“I visually cluster and organize widgets”

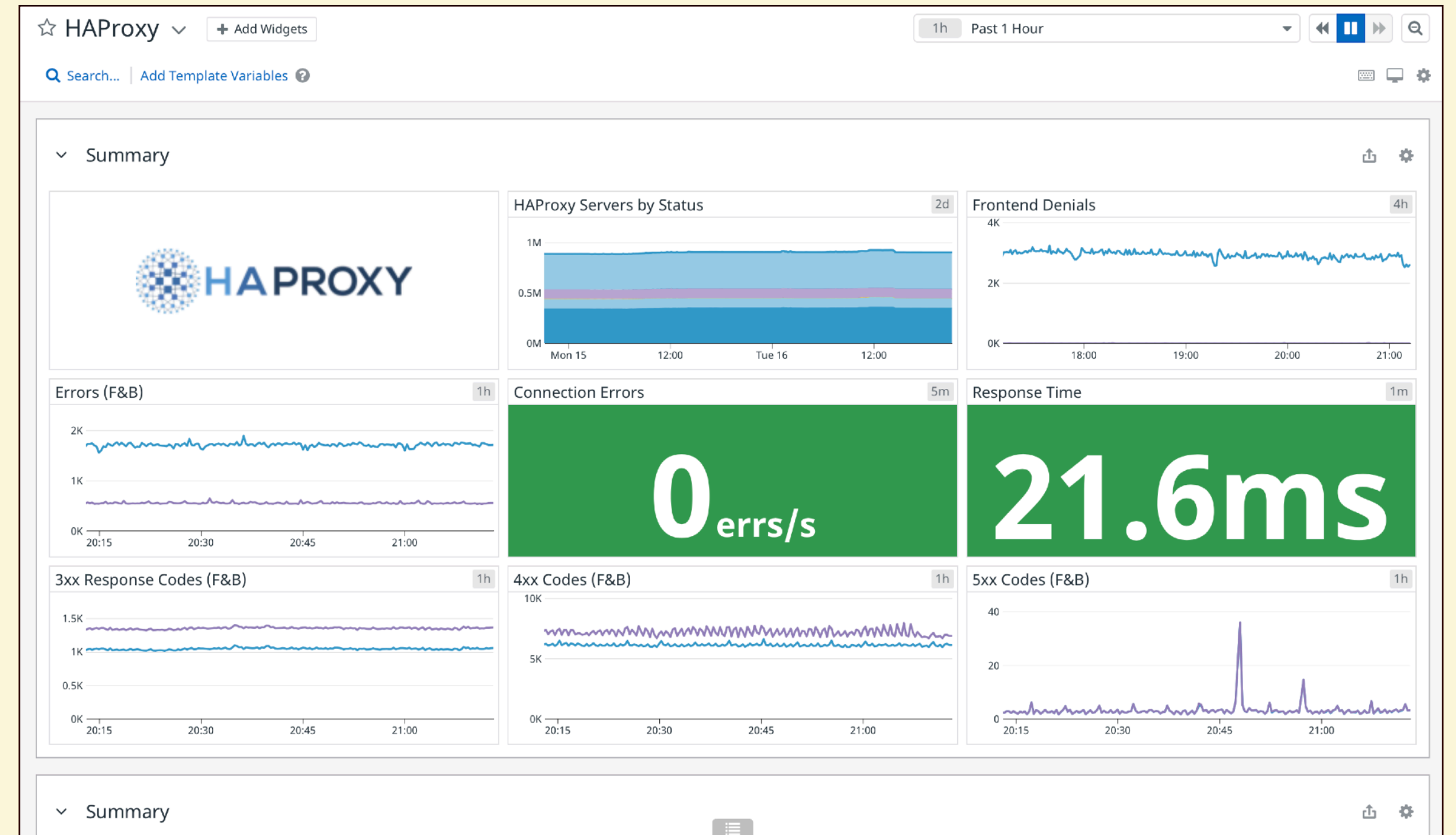
“I put the most important metrics on top”

Cloud-native + DevOps



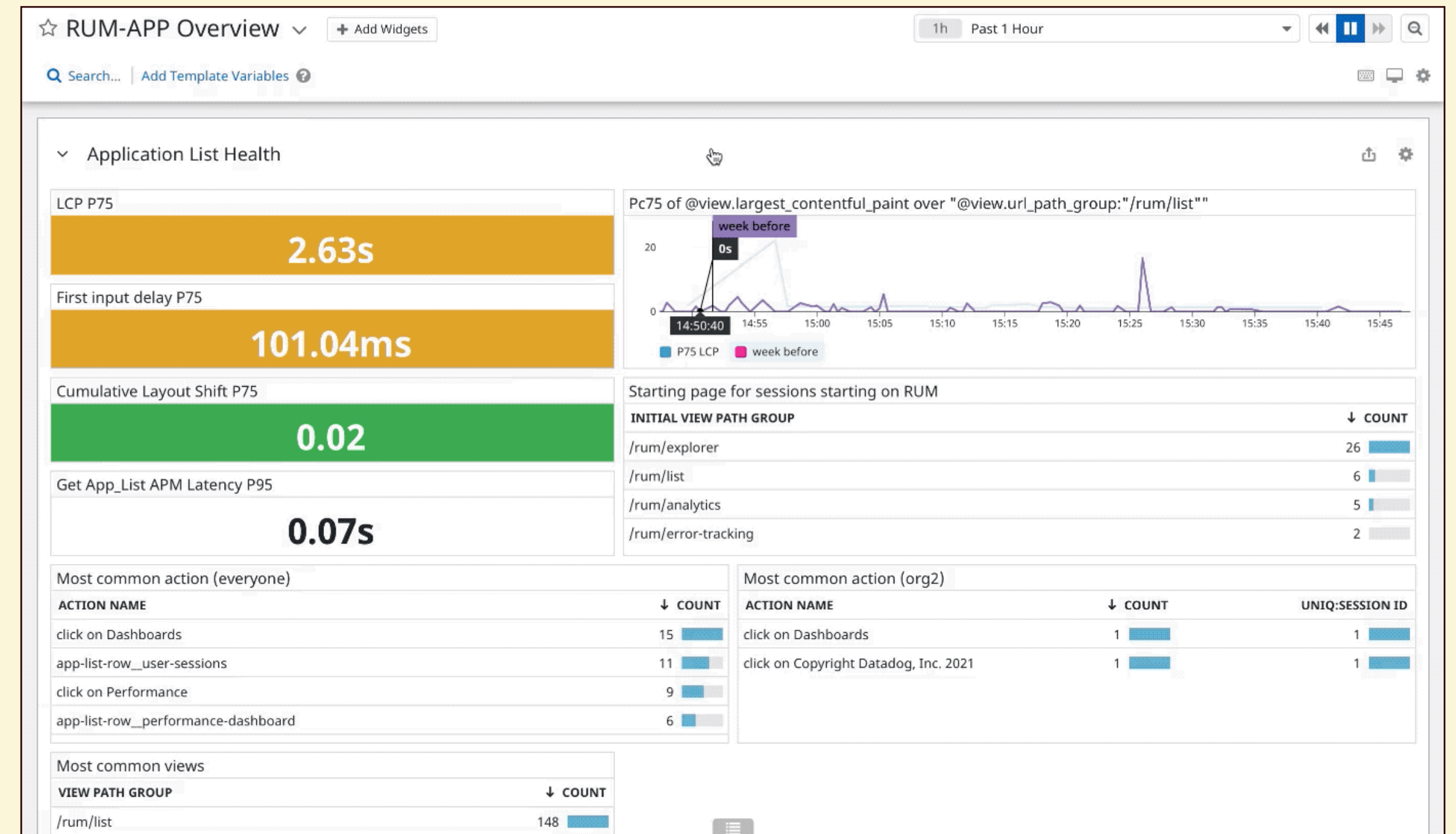
Trust us

- 400+ integrations
- Out-of-the-box dashboards for most integrations



What we built

- Widget groups
- Revamped OOTB dashboards using widget groups
- Copy and paste for widgets & groups



What we said

Blog post on clipboard

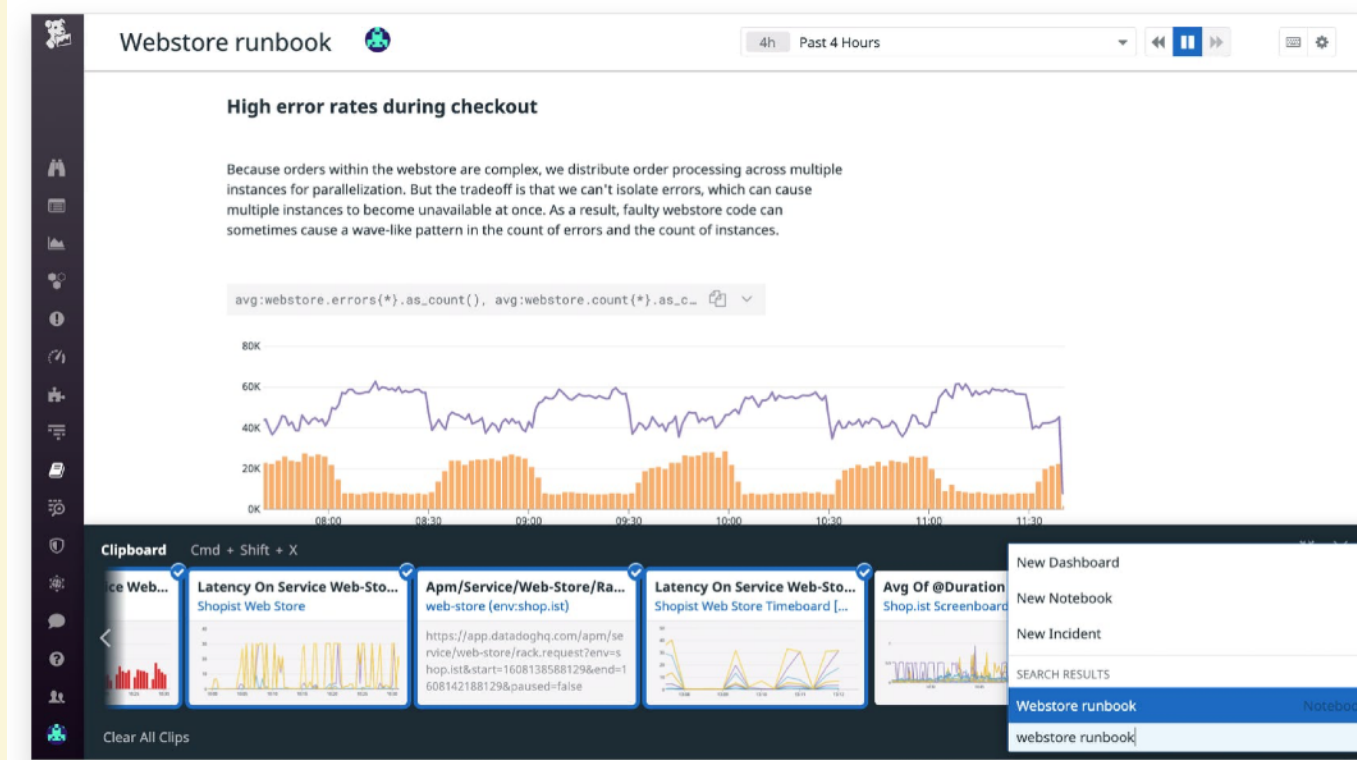
- Shared resources
- Easier dashboard creation
- Frictionless troubleshooting

From explorations to shared resources

Whether you have joined a new team or are designing a new service, you often need to gather data from different contexts while building an overall picture of your infrastructure and applications. The Datadog Clipboard streamlines this process by letting you gather relevant signals without leaving your current Datadog tab.

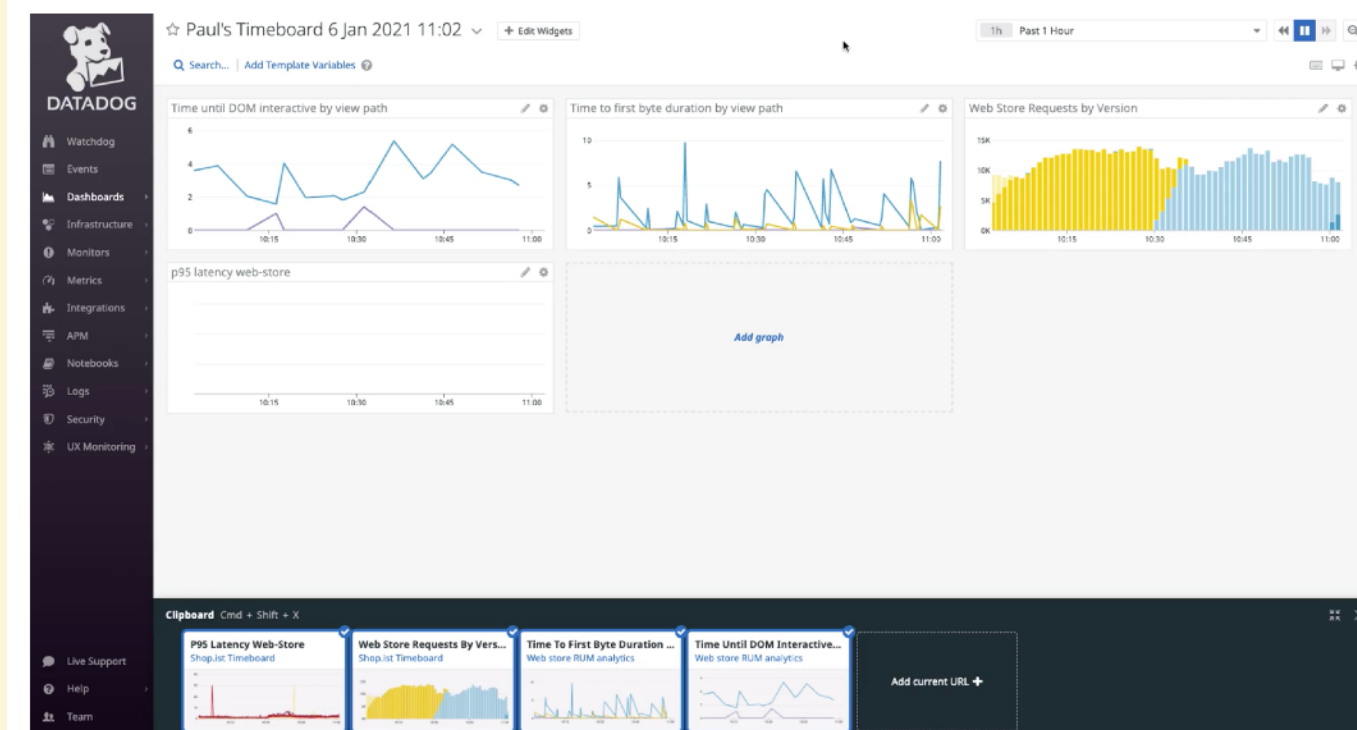
As you explore your team's resources, any graphs you copy are automatically stored in the Clipboard. You can easily export these graphs to a Notebook. Notebooks can serve as personal guides or resources to share with your teammates, who can edit them in real time and add comments for discussion. This way, you can use the Clipboard to easily create runbooks, allowing teammates to benefit from your research when deploying a new service, scaling up a cluster, or verifying that a service can communicate with a new dependency.

In this example, we are adding signals to an existing runbook from the Clipboard as we learn more about one of our services.



Easier dashboard creation

You can also use the Datadog Clipboard to create dashboards more efficiently. For example, let's say you are deploying a new feature and want to understand how your users interact with it. While browsing through Datadog —e.g., graphs of RUM analytics data or browser logs—you copy widgets to the Clipboard. Once you've amassed a complete picture of your data, you select the signals you want to export, then click "Add to a new dashboard." Datadog will automatically create a dashboard from the selected signals.



The dance

Courting stakeholders



Product and design briefs

The inside-out approach, court your team

- Start with the persona(s) and use cases
- Define successful outcomes and workflows
- We started creating these in response to “design failures”
 - 7+ iterations of design without agreement on solution
- Our biggest launch failures have nearly always been due to lack of clarity in persona development

What launch failure looks like

- We model the persona in the wrong way
- Private beta users “don’t get it”
- Tear-down as opposed to design iteration
- Months of development time lost
- Sunken cost fallacy makes this worse
 - Be willing to accept medium-scale failure to avoid large-scale failure



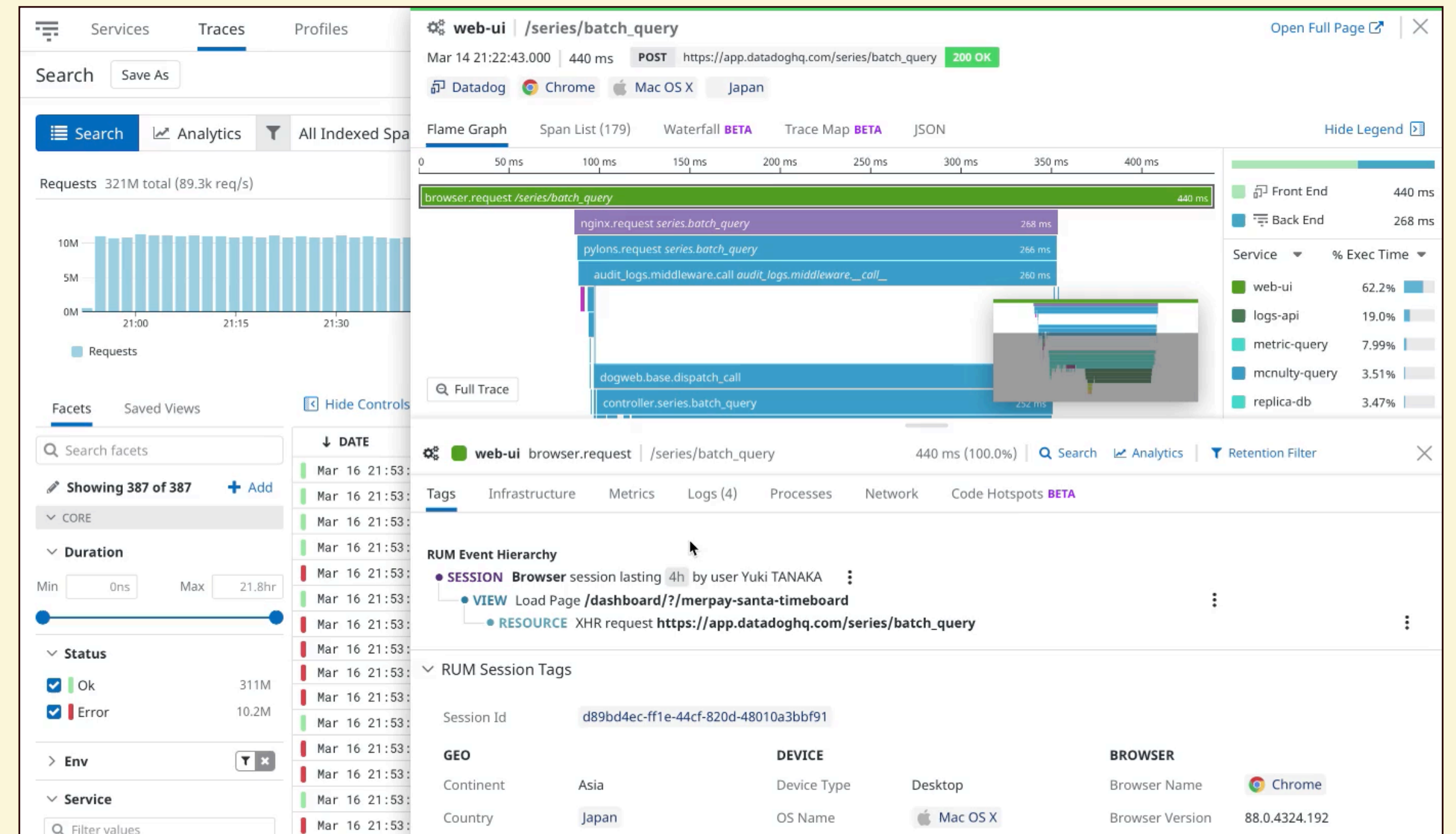
Press release-driven development

The outside-in approach, court your customers

- Amazon thing: “working backwards”
- Imagine how your customers will understand the feature up front
- We don’t formally practice this
- We tried this a few times
 - Product teams cannot work from **only** a press release
 - I have found myself advocating against them
- If you like your designers and engineers you will write proper product briefs also

Bundle & upsell

- Adding to the customer model is the core dynamic of the business
- Augment the persona with new stories, upsell on those stories
- We added APM and Logs: three pillars of observability
 - Complete picture now means connecting Logs to Traces
- Security, RUM and other products starting to stretch into new user types



This also defines how teams are organized

- You add more and more user types and personas
- Pods exist to own distinct customer types
- We organize the Design org around this
 - Design teams structured around related products for distinct customer types

In conclusion

- Start with a small number of assumptions about our core user types
- Recruit small numbers of people into design partner groups based on strong matches from those assumptions
- Over time, bolt on more assumptions and add features/products to meet them
 - Upsell based on those value-adds
- Organize teams around subgroups of those assumptions
- That's basically where we are today as a \$25B+ company

Thanks!