What does Datadog do?

 technically.substack.com/p/what-does-datadog-do

Justin

The TL;DR

Datadog is **monitoring software** - developers use it to get operational visibility (what's broken? What's taking too long?) into their servers and applications.

- As applications and infrastructure have gotten more complex, developers need to know what's **going on under the hood** so they can pre-empt and troubleshoot
- Datadog provides a product and series of SDKs (little code libraries) that let you **instrument your application**, record your metrics, and analyze them visually
- Generally, the Datadog platform has **4 major components**: server monitoring, application monitoring, logs, and Other Stuff™

The New York based (surprisingly enough) company IPOd back in 2019, and is worth $23B (!) at the time of writing. So they're definitely onto something.

Servers, apps, and metrics

This is the first time we're writing about monitoring and observability, so it's worth taking a step back to understand (a) why developers need this *visibility* and all of these *metrics*, and (b) how we got here (why didn't Datadog exist 10 years ago?).

Every app that you use on the internet is running on a server somewhere. Until recently, that used to literally mean one server - a giant computer - so you had whatever computing power you had, and you had one place to look if you wanted to know why things were slow, and why your users were sending you infinite "fuck you" loops in Java. On your server, there are a few metrics you want to keep an eye on to make sure everything is running smoothly:

- **CPU** – how much processing your server is doing as a function of its total processing power (e.g. 3 out of 4 CPUs)
- **RAM** - how much memory your server is using (e.g. 3GB/4GB memory)
- **Disk** - how much storage your server is using (e.g. 450GB/500GB stored)
- **IO** - how fast your server is reading and writing things from memory and disk

There were basic utilities in Linux for monitoring a lot of this stuff, like the htop command – which is still used a lot, mind you – but this was mostly a reactive process. Something would go wrong, and you'd check why.

Then a few things changed:

1. **Infrastructure got easier, but more complicated**

Once Docker and Kubernetes entered the scene, things changed a lot - the concept of "servers" got a lot more complicated, because there was a thick layer of abstraction between your code and what infrastructure it was running on. That meant more surface area for problems – you could be having an issue with Docker *or* your server – but it also meant nicher things to worry about, like your Kubernetes cluster having a hard time restarting pods and other things I don't fully understand.

2. **The internet got bigger**

The other thing that happened is that the internet became widely available, so apps are now used by like, billions of people. So when 2 billion people are loading Facebook.com every hour as opposed to 200, a lot more things can go wrong, and critically, *it's more important to fix them, and fix them quickly*.

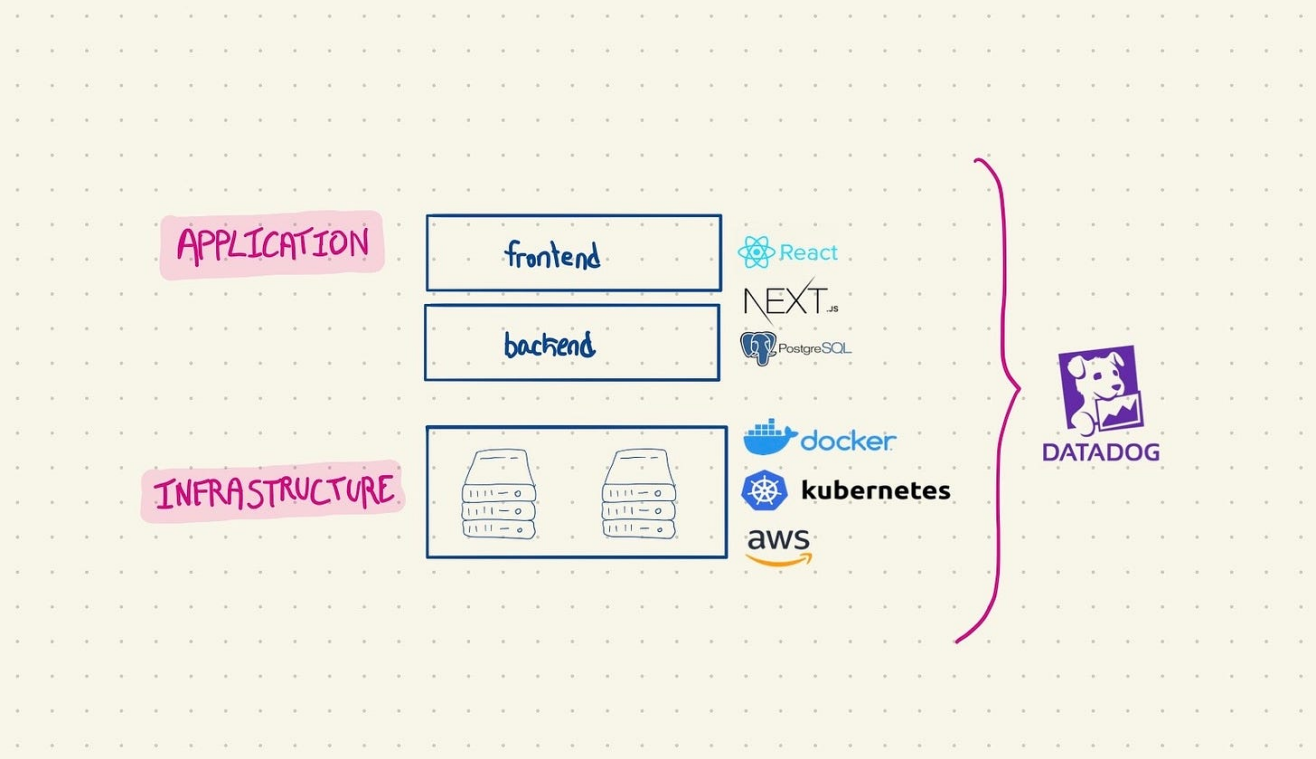
> 🔍 **Deeper Look**🔍
>
> Beyond servers, developers also needed to start monitoring their **application performance**. How long are requests to our backend taking to fulfill? Are users getting any errors on their profile page? Problems with any of these could be the server's fault *or* the application's fault, which makes investigating all the more tedious.
>
> 🔍 **Deeper Look**🔍

These are the two big ones, but a lot of other stuff was happening behind the scenes too that contributed to an environment where Datadog could grow fast. The move into microservices from monolithic apps created more individual entry points for monitoring. And as companies moved from on-prem to the cloud, building native integrations to save time became more feasible.

So in summary, developers were faced with more complex infrastructure *and* more pressure to understand, monitor, and keep that infrastructure running smoothly. This is part of why DevOps (development operations) started to become its own discipline – companies were employing teams of developers *just to deploy and monitor infrastructure*. And from the ashes of 3AM pager buzzes and angry managers, a giant was born.

The basic Datadog product

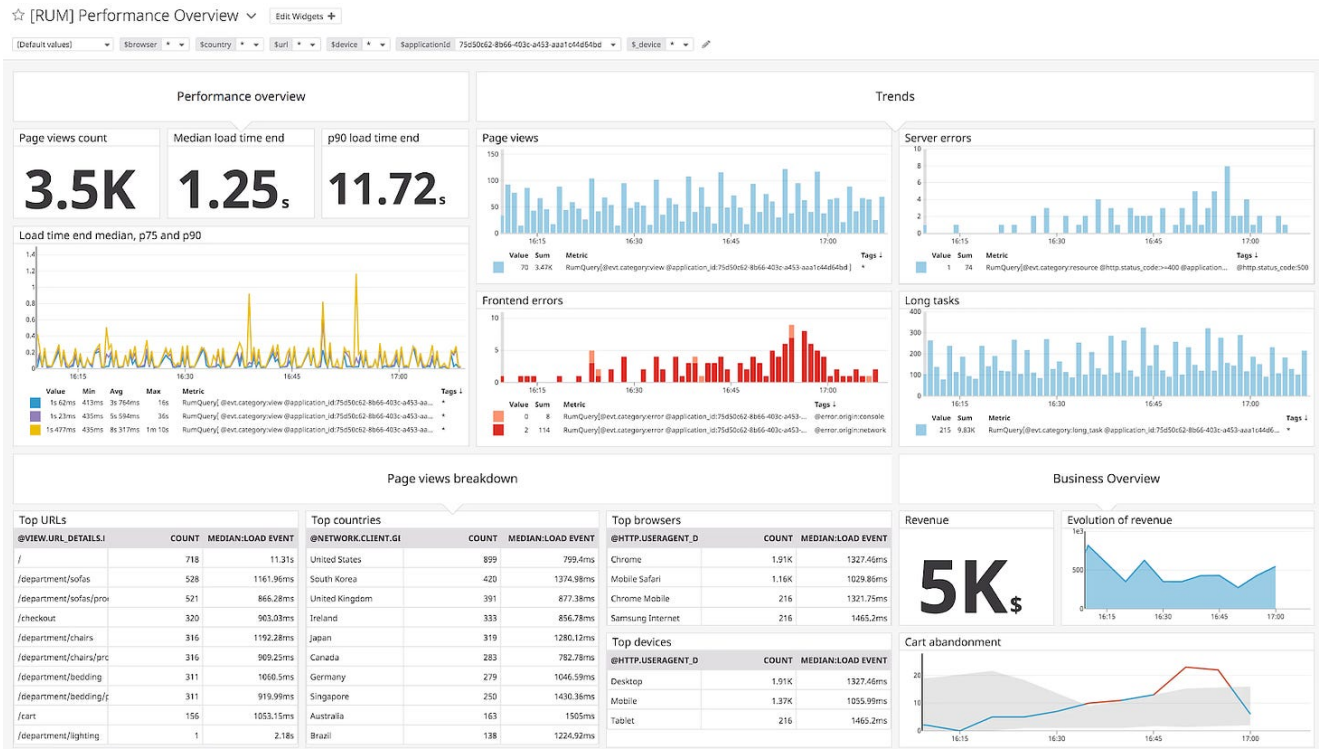
Datadog is a godsend for DevOps teams (and before you're large enough to have a DevOps team, regular full stack developers). It hooks up to your infrastructure (Docker, Kubernetes, plain Linux, etc.) and automatically pulls metrics like CPU and Disk usage.



Here's how the basic product works:

1. Install Datadog on your servers (they call this the "agent")
2. Datadog collects your performance data and stores it
3. You visualize and set alerts on that stored data as you please

Like most developer tools, the Datadog product is a combination of code libraries (SDKs) that you have to integrate with your application, as well as a web interface that you use for admin tasks, dashboarding, setting alerts, etc. Here's what a basic Datadog dashboard might look like:



You'll notice that it's tracking page views, frontend errors, server errors, and long running server tasks - you can use the Datadog SDK to instrument these specific metrics in your app.

Pricing is predictably complicated (you pay per host), but you can expect to pay $1K+ per month at least for an application with meaningful usage. Datadog *only works in the cloud*, which is interesting, because with their market cap and customer list, you'd assume they're selling into organizations for whom cloud is a non-starter. It's possible these companies are more comfortable using the cloud for monitoring since there's little/no PII involved, but this thread is worth following (note: big competitor New Relicalso doesn't deploy on prem).

The Datadog product *suite*

No $34B enterprise software company has just one product, and Datadog is no exception. Generally, their product suite fits into 4 distinct buckets. Dashboards and alerting is sort of like a meta-layer that works on top of all of this stuff.

1. **Infrastructure monitoring**

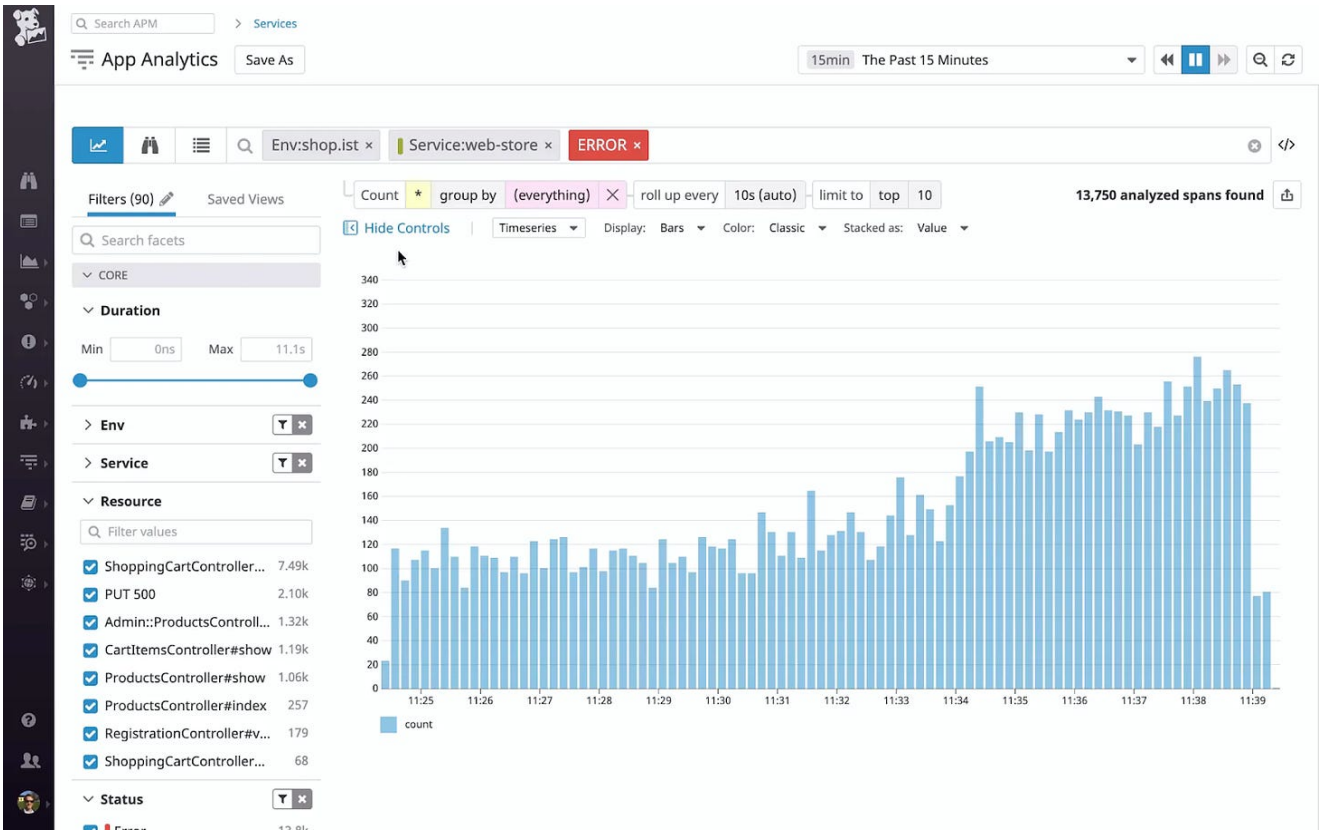
Infrastructure monitoring is keeping an eye on your core infrastructure, *below* the application level. Datadog has agents (integrations) for Docker, Kubernetes, Heroku, Ubuntu, etc. – pretty much everything you'd ever use if you're not stuck on some legacy data center run by a guy named Karl who wears cargo pants and a Slayer tee to work . For example, here are the metrics that Datadog collects by default for their Docker agent:

CHECK	METRICS
CPU	System
Disk	Disk
Docker	Docker
File Handle	System
IO	System
Load	System
Memory	System
Network	Network
NTP	NTP
Uptime	System

Again, this data isn't very valuable sitting around - the real oomph of Datadog is the ability to pull this into a dashboard, visualize it, and set alerts to let you know when something is wrong.

2. **Application monitoring (APM)**

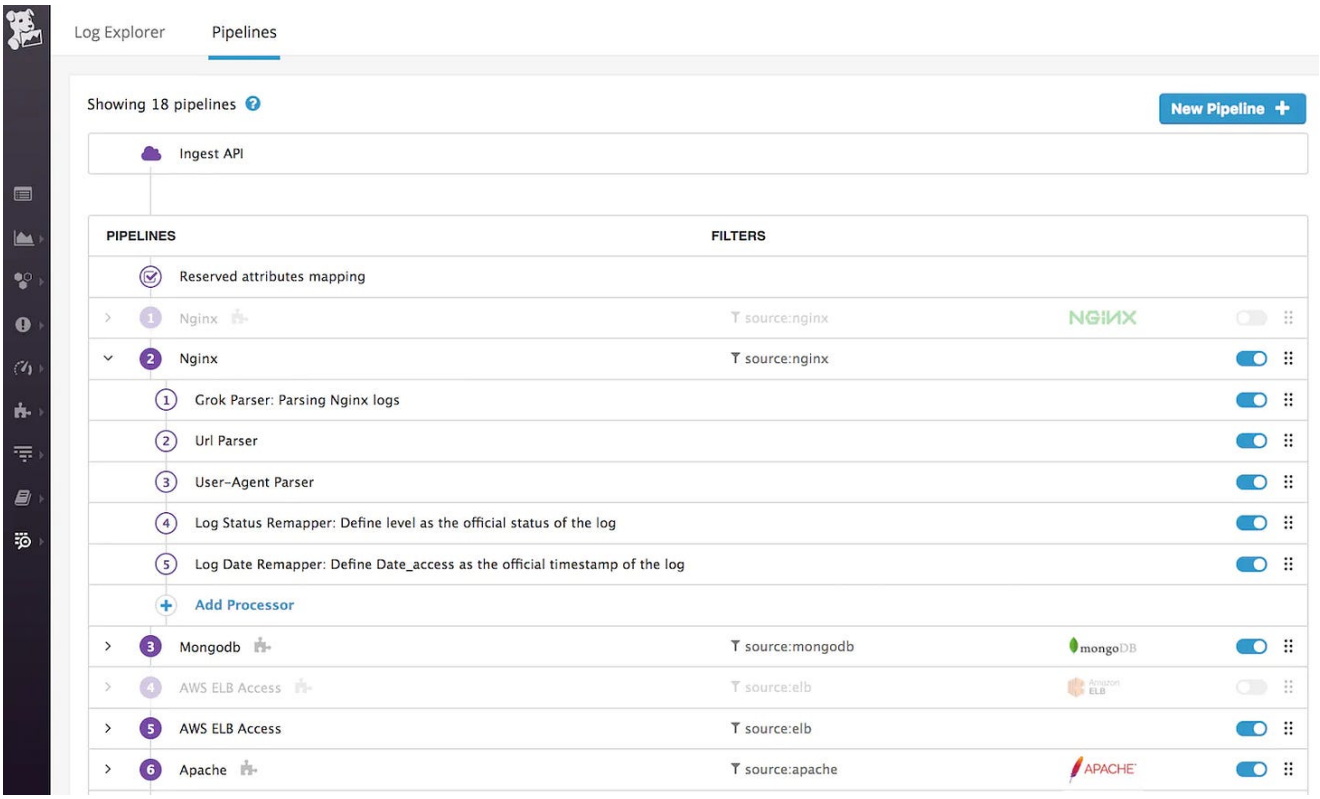
Application monitoring is one level above server monitoring - it's how developers look at their frontend and backend performance. If you have an endpoint that returns user information that you use to populate a profile page, you might want to monitor how fast the request goes, and if it returns the kind of data you expect it to.



APM is a category in and of itself, and there are entire other companies like AppDynamics, DynaTrace, and Sentry that make most of their money doing this.

3. **Log management**

Recently, Datadog released a log management product - it gathers all of the logs that your server generates, stores them, and lets you analyze and search them. Using logs is generally the next step *after monitoring* - if Datadog shows you that something is going wack with your server, you'll find the logs the server is emitting, which should contain useful info on what's happening.



This is directly competitive with what Splunk and Elastic make their money on, so it will be interesting to see where it goes.

These first 3 categories - infrastructure metrics, APM, and logs - are often referred to as the 3 pillars of observability (is in some greek pantheon or whatever).

4. **Other Stuff™**

Sort of like Segment, and to an extent all maturing developer focused companies, Datadog has continued to release new products that aren't groundbreaking on their own, and probably represent a tiny percentage of revenue, but help build their ecosystem and make the entire Datadog suite that much more attractive. A few examples:

- Continuous Profiling: analyzing code in production automatically
- Security Monitoring: real time thread detection across your app
- Real User Monitoring: looking at user journeys across apps

Some of these are bound to sunset at some point; every enterprise software company eventually becomes an ecosystem, as the value of each individual product is a lot higher when they're integrated together into a single package (or at least that's what my strategy textbooks told me).

Further reading

- My friend Adam wrote a great piece on the history of DevOps and how Datadog came to be
- Datadog's documentation is generally just ok, but their getting started guide is useful for breaking down the different product lines

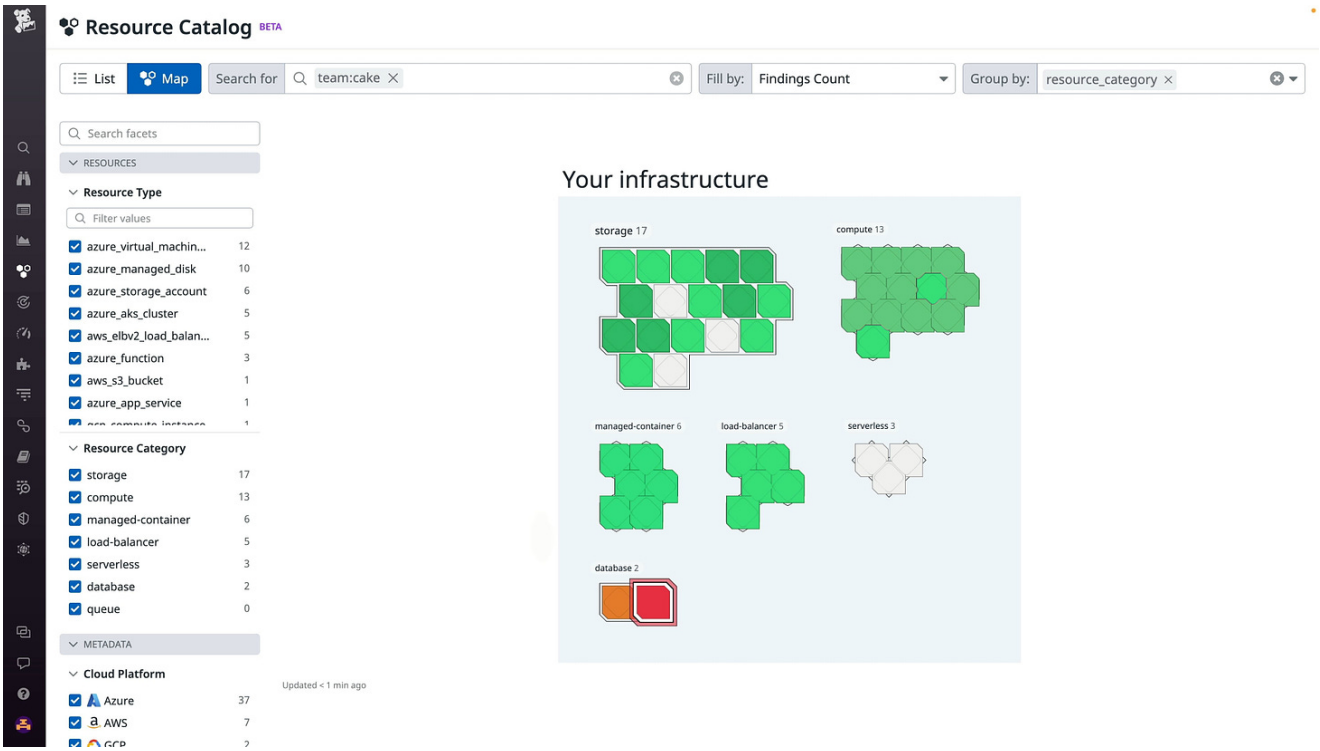
Update: Datadog's new security products

Since the time of writing, Datadog has been investing heavily in new security products. Here's a quick lowdown on what they are and what they do.

The way to think about these products is that they add a **security view** to the data that the product was already collecting. E.g. Datadog knows the commands running on your servers and what they're doing because you installed it for performance monitoring: it's easy for them to organize that data in a different way (different charts, different alerts, etc.) that help with *security* stuff instead of performance stuff.

1. **Cloud security management**

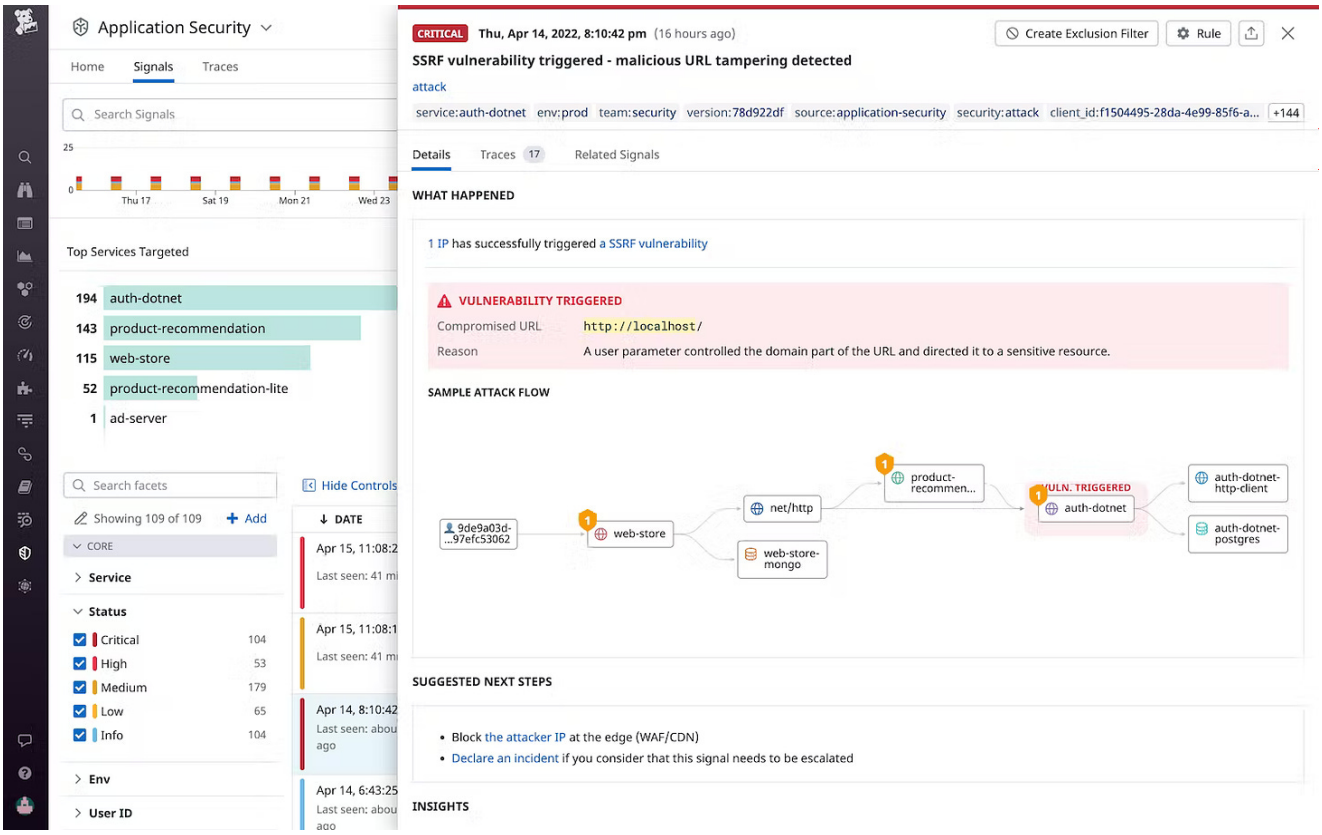
Datadog CSM helps teams find any vulnerabilities in their infrastructure: unsecured endpoints that a hacker could get into, errant commands run on a machine that might be from a hacker, etc. The first thing you get is a dashboard that shows the status of each piece of your infrastructure, and how all of it relates to each other:



For each piece of infrastructure (a server, a database, etc.) you get a view of what commands are running in it, status, etc. If a suspicious command gets run (e.g. changing permissions on a server), Datadog will notify you. There's also a facility for managing **incidents** – when a breach actually happens, or infrastructure goes down.

2. **Application security management**

This product is similar to the above, but with respect to your *application* instead of your *infrastructure*. The basic idea is that Datadog will detect when someone is trying to attack your application, and notify you about what's happening and where.

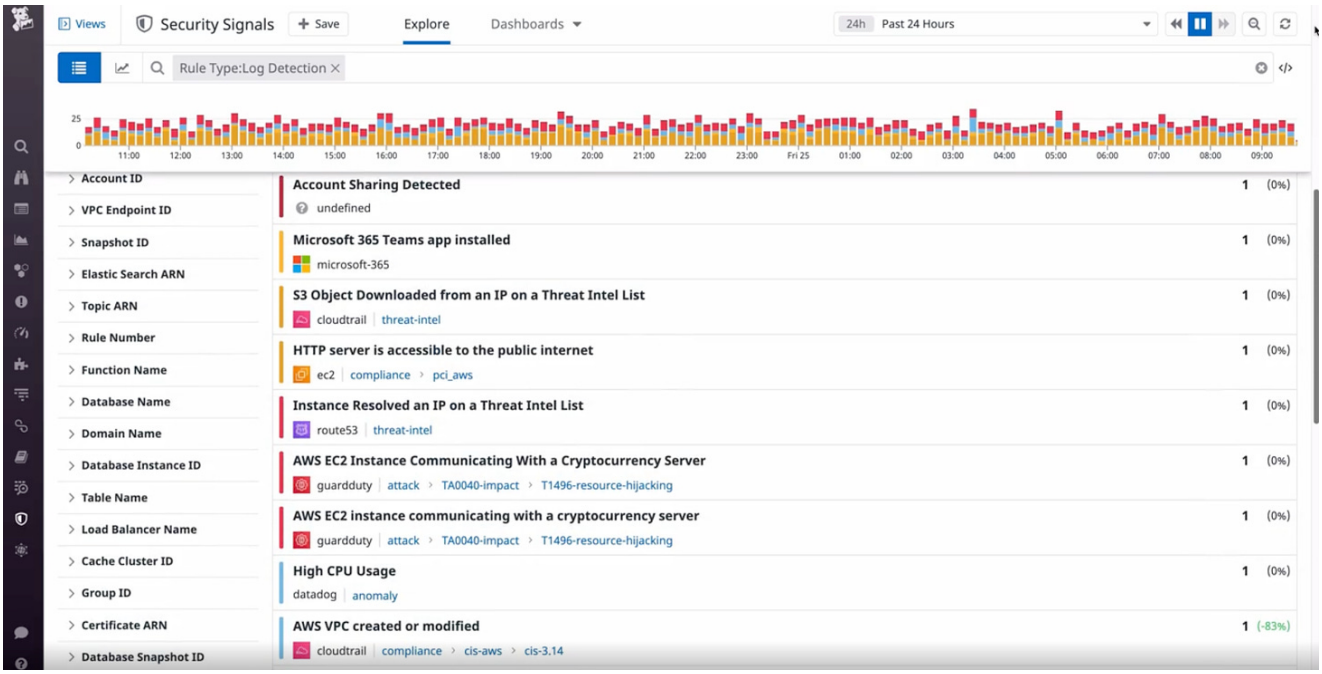


You're paged at 3AM because some asshole was fucking around with your application's URL trying to find sensitive resources

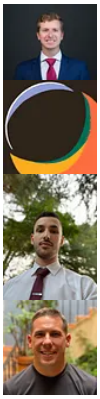
3. **Cloud SIEM**

This one is interesting, since Datadog is getting into a space with some big players in it (like Splunk). SIEM (Security Information and Event Management) is the tried and true practice of ingesting massive amounts of mostly useless logs from your infrastructure and seeing what the fuck is going on with them.

Every time *anything* happens on a server, or in a more enterprise-focused SaaS product, a **log** is generated. That log is just a bunch of text – plus a timestamp – describing what happened. User logged in. Server restarted. Configuration changed. You get it. Anyway there are so many of these – and most are innocuous – that looking through them manually, especially for large companies, is all but impossible. SIEM products help teams ingest, transform, and analyze this big data to see what's actually going on:



It's beyond me to say if this is a *good* SIEM product, but they seem to positioning it as a cost effective, simpler alternative to products like Splunk.



24 Likes

24

9

Share

Previous

Next

9 Comments

7 more comments...