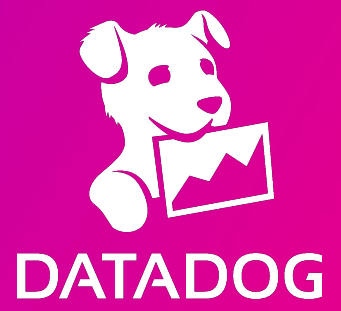
KUBECON CHICAGO 23

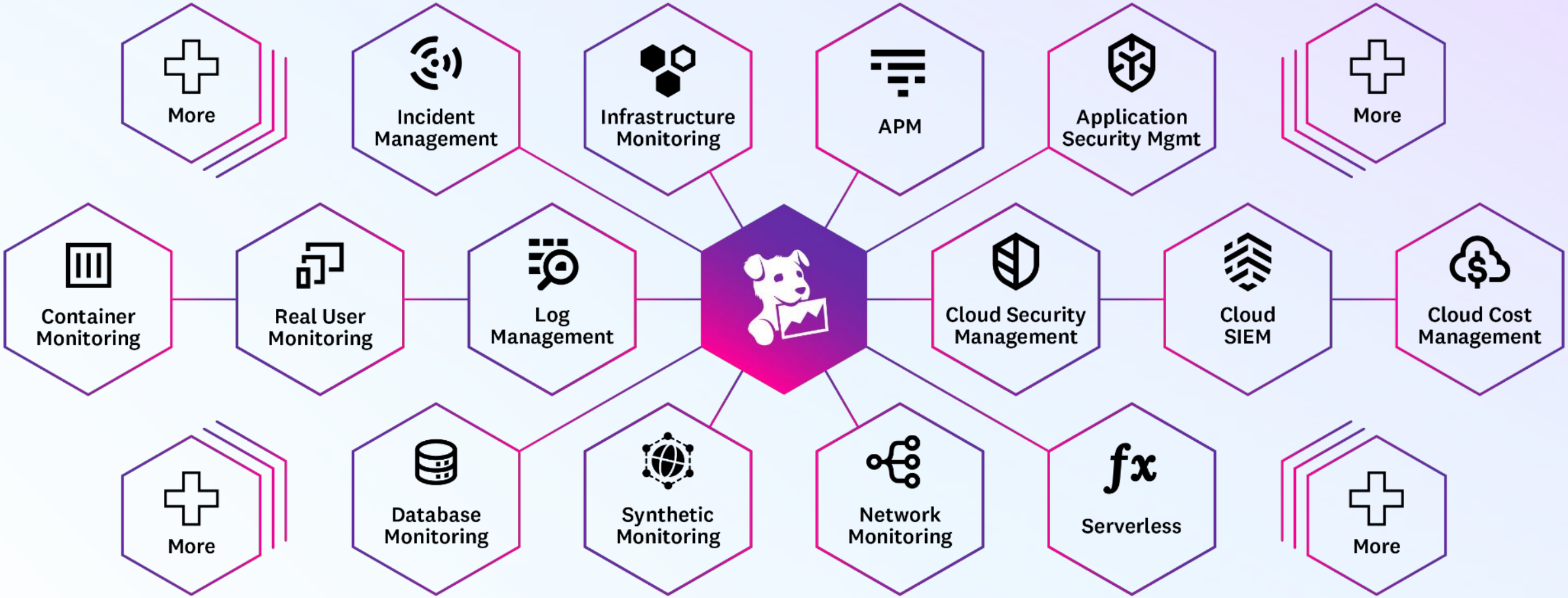
Everything, Everywhere, All at Once

Laurent Bernaille

Hemanth Malla



Datadog


More
Incident Management
Infrastructure Monitoring
APM
Application Security Mgmt
More
Container Monitoring
Real User Monitoring
Log Management
Cloud Security Management
Cloud SIEM
Cloud Cost Management
More
Database Monitoring
Synthetic Monitoring
Network Monitoring
Serverless
More

10,000s of nodes

100s of clusters

4000+ nodes / cluster

Self-managed

March 8th, 2023

Incident and impact

Starting on March 8, 2023, at 06:03 UTC, we experienced an outage that affected the US1, EU1, US3, US4, and US5 Datadog regions across all services.

When the incident started, users could not access the platform or various Datadog services via the browser or APIs and monitors were unavailable and not alerting. Data ingestion for various services was also impacted at the beginning of the outage.

https://www.datadoghq.com/blog/2023-03-08-multiregion-infrastructure-connectivity-issue/

60%

of nodes down in less than 1hr
in all data-centers

Everything, Everywhere, All at Once

Symptoms

Users cannot access the Datadog platform

Issues with Datadog accessing Datadog

Issues accessing Kubernetes Control Planes

Issues with SSHing into nodes to debug

Bad change fleet wide ?

We have explicit policies in place to avoid that !

This can’t happen

01:24 **Zoom** APP

Call

Zoom meeting started by laura.devesine

[redacted]

Meeting ID: [redacted]

9+

493 people joined

Restart FTW !

Unattended upgrades !

Past

Unattended upgrades

Today

100% Node Lifecycle Automation

Past

Unattended upgrades

March 8th

Unattended upgrades
+
Node Lifecycle Automation

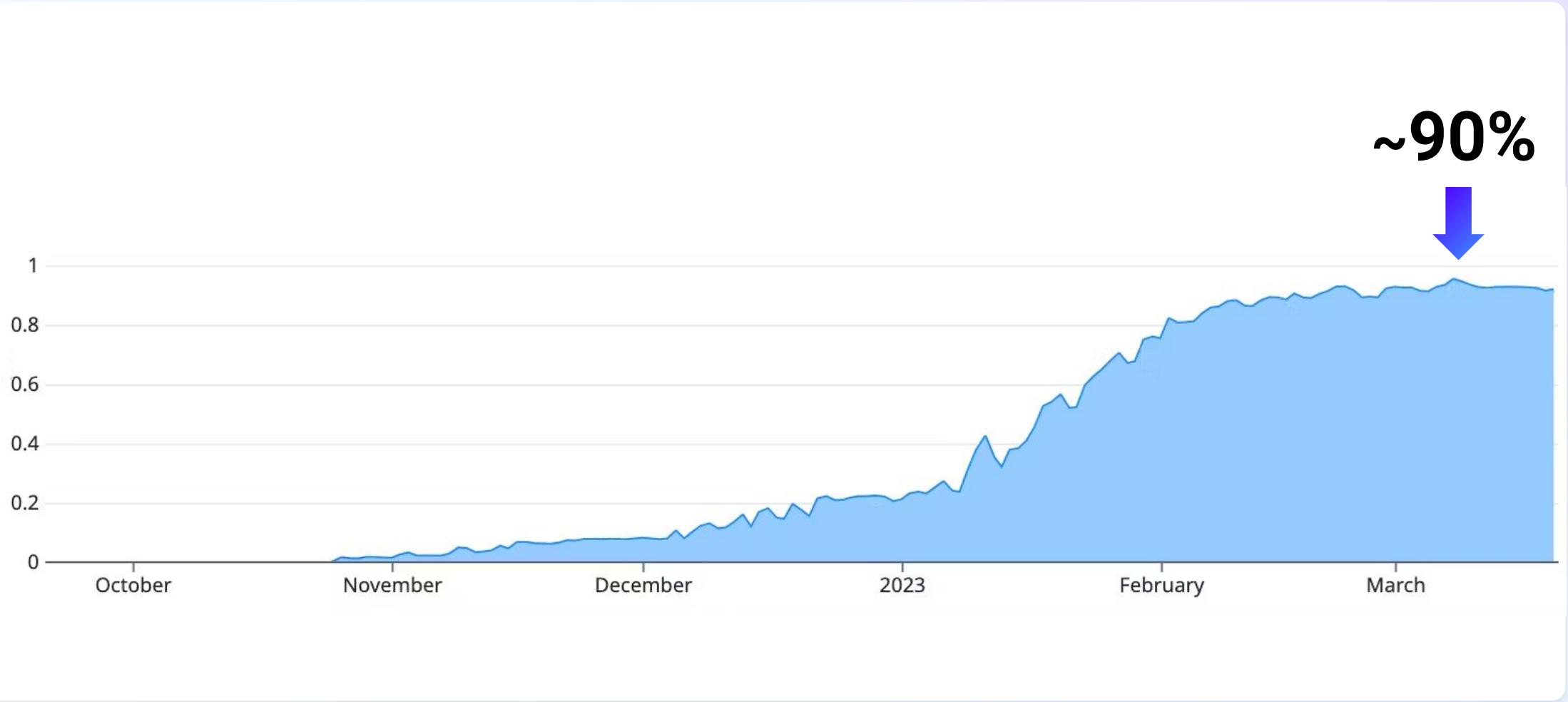
Today

100% Node Lifecycle Automation

How to Carefully Replace Thousands of Nodes Everyday

Adrien Trouillaud & Ryan McNamara, W179, 11:55AM

Ubuntu 22.04 Rollout



USN-5928-1: systemd vulnerabilities

7 March 2023

It was discovered that systemd did not properly validate the time and accuracy values provided to the format_timespan() function. An attacker could possibly use this issue to cause a buffer overrun, leading to a denial of service attack. This issue only affected Ubuntu 14.04 ESM, Ubuntu 16.04 ESM, Ubuntu 18.04 LTS, Ubuntu 20.04 LTS, and Ubuntu 22.04 LTS. (CVE-2022-3821)

It was discovered that systemd did not properly manage the fs.suid_dumpable kernel configurations. A local attacker could possibly use this issue to expose sensitive information. This issue only affected Ubuntu 20.04 LTS, Ubuntu 22.04 LTS, and Ubuntu 22.10. (CVE-2022-4415)

It was discovered that systemd did not properly manage a crash with long backtrace data. A local attacker could possibly use this issue to cause a deadlock, leading to a denial of service attack. This issue only affected Ubuntu 22.10. (CVE-2022-45873)

https://ubuntu.com/security/notices/USN-5928-1

What we know so far

Patched nodes seem broken

Investigation

Patched nodes seem broken

Restarting systemd breaks networking consistently

Investigation

Patched nodes seem broken

Restarting systemd breaks networking consistently

Restarting systemd-network also breaks networking consistently

Investigation

Patched nodes seem broken

Restarting systemd breaks networking consistently

Restarting systemd-network also breaks networking consistently

Only Ubuntu 22.04 is impacted !

Deep in the commit history of systemd-networkd

network: drop unnecessary routing policy rules

```
networkd already drop foreign address, routes, and nexthops on startup,
except those created by kernel. However, previously, routing policy
rules were not. The logic of serialization/deserialization of rules only
works for rules created by previous invocation of networkd, and does not
work for one created by other tools like `ip rule`.

This makes networkd drop foreign routing policy rules except created by
kernel on startup. Also, remove rules created by networkd when the
corresponding links are dropped or networkd is stopping.
```

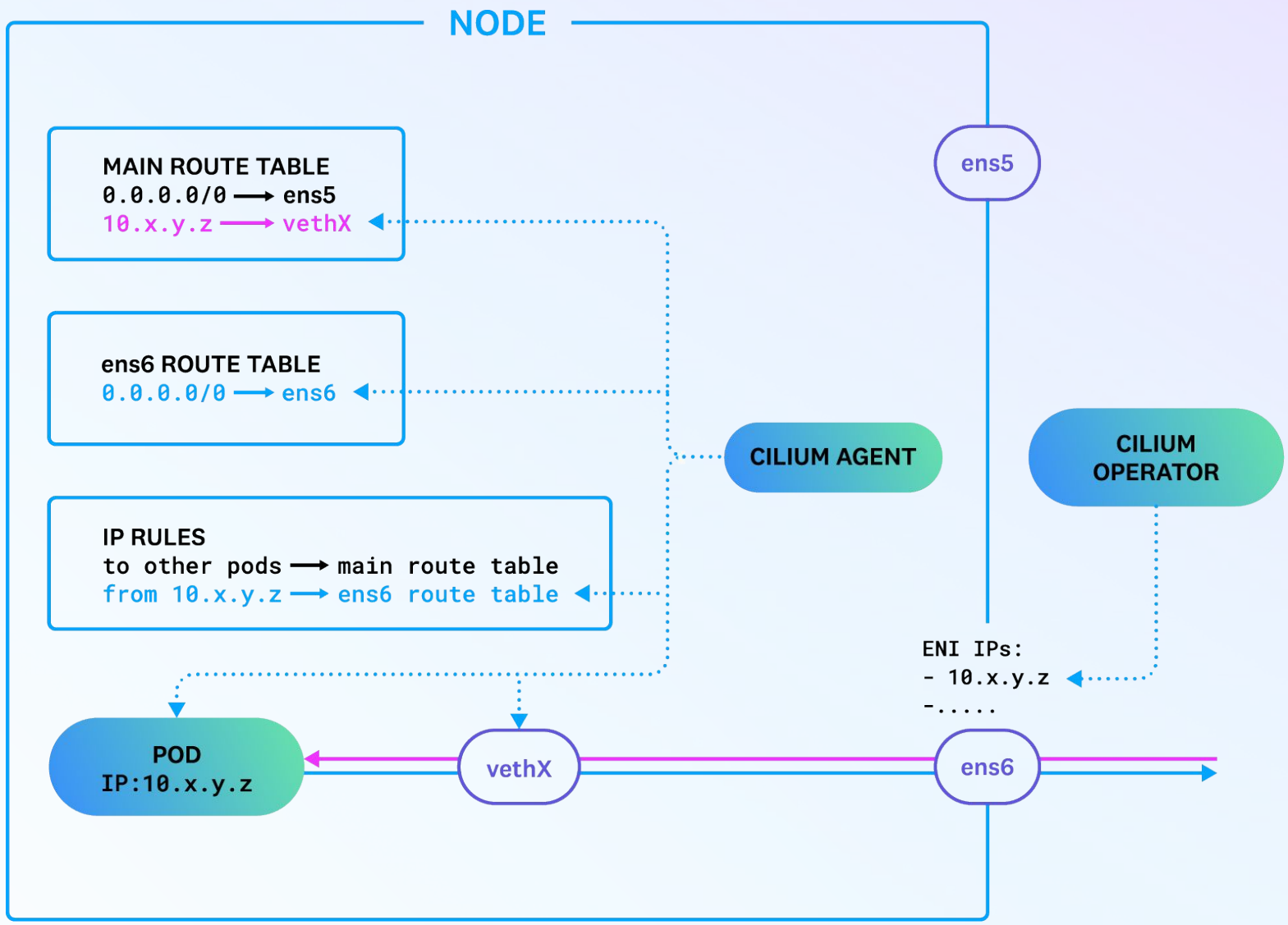
---

main (#17477)

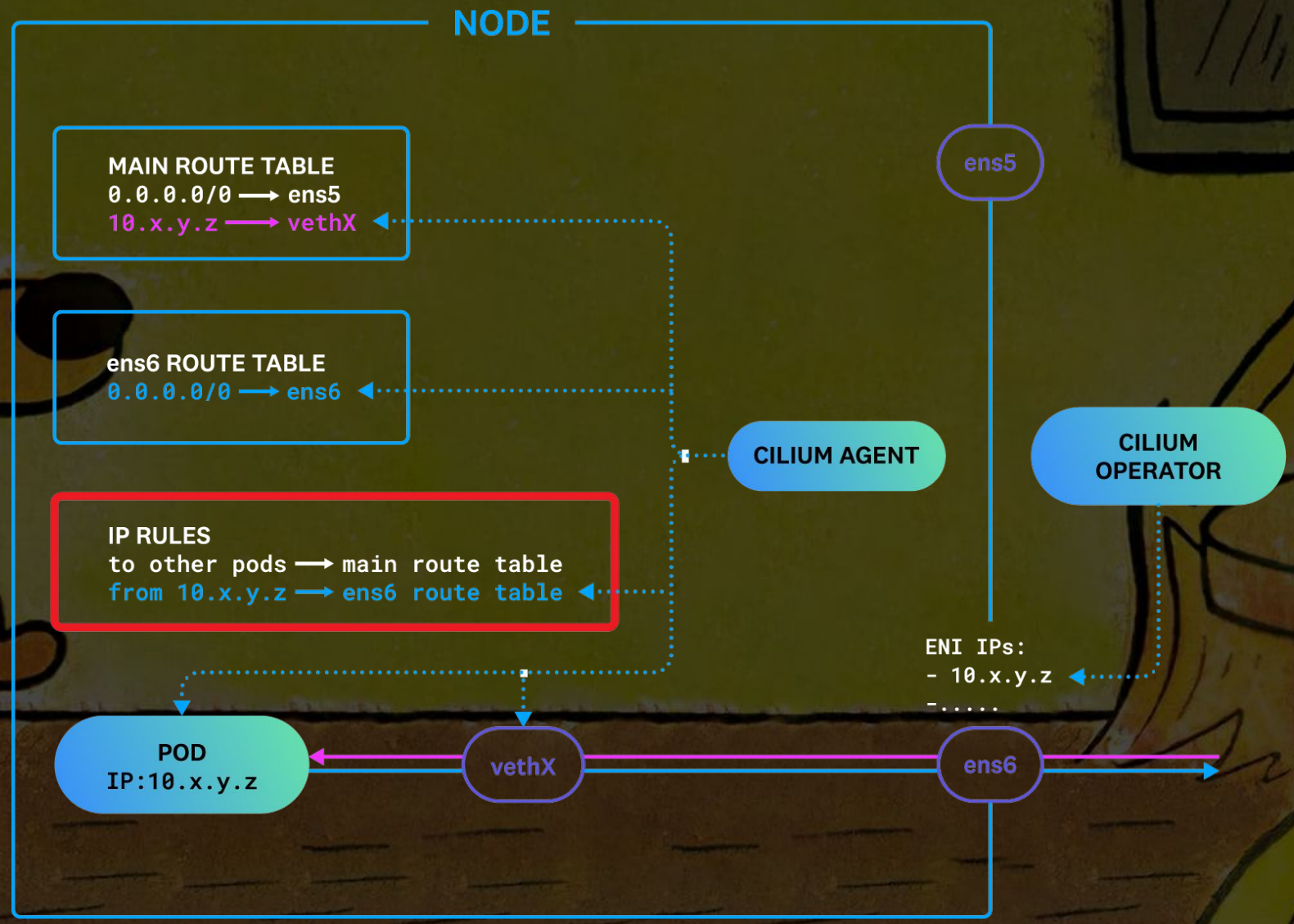
v254 v254-rc3 v254-rc2 v254-rc1 v253 v253-rc3 v253-rc2 v253-rc1 v252 v252-rc3 v252-rc2
v249.12 v249.11 v249.10 v249.9 v249.8 v249.7 v249.6 v249.5 v249.4 v249.3 v249.2 v249.1 v249
v248.3 v248.2 v248.1 v248 v248-rc4 v248-rc3 v248-rc2 v248-rc1 v248-2

Introduced between Ubuntu 20.04 and 22.04

Kubernetes Networking at Datadog



On systemd-networkd restart



Summary

- systemd-networkd never restarts in happy flow
- Patch for an unrelated CVE in systemd triggered systemd-networkd restart
- systemd-networkd restart broke networking

Summary

systemd-networkd never restarts in happy flow

Patch for an unrelated CVE in systemd triggered systemd-networkd restart

systemd-networkd restart broke networking

AND Unattended upgrades *ran* everywhere between 6AM and 7AM

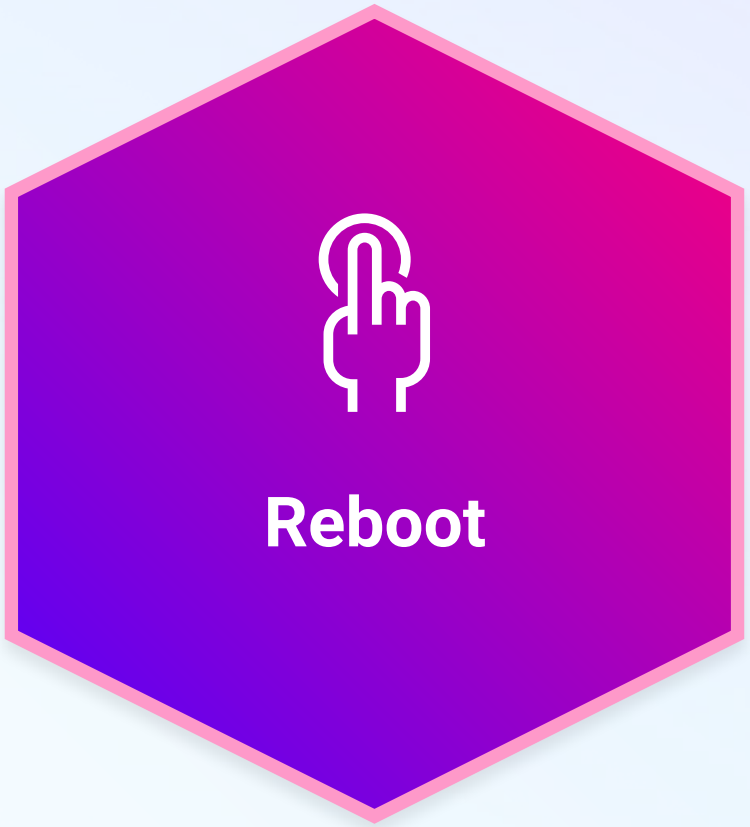
How did we recover from this ?

Recovery

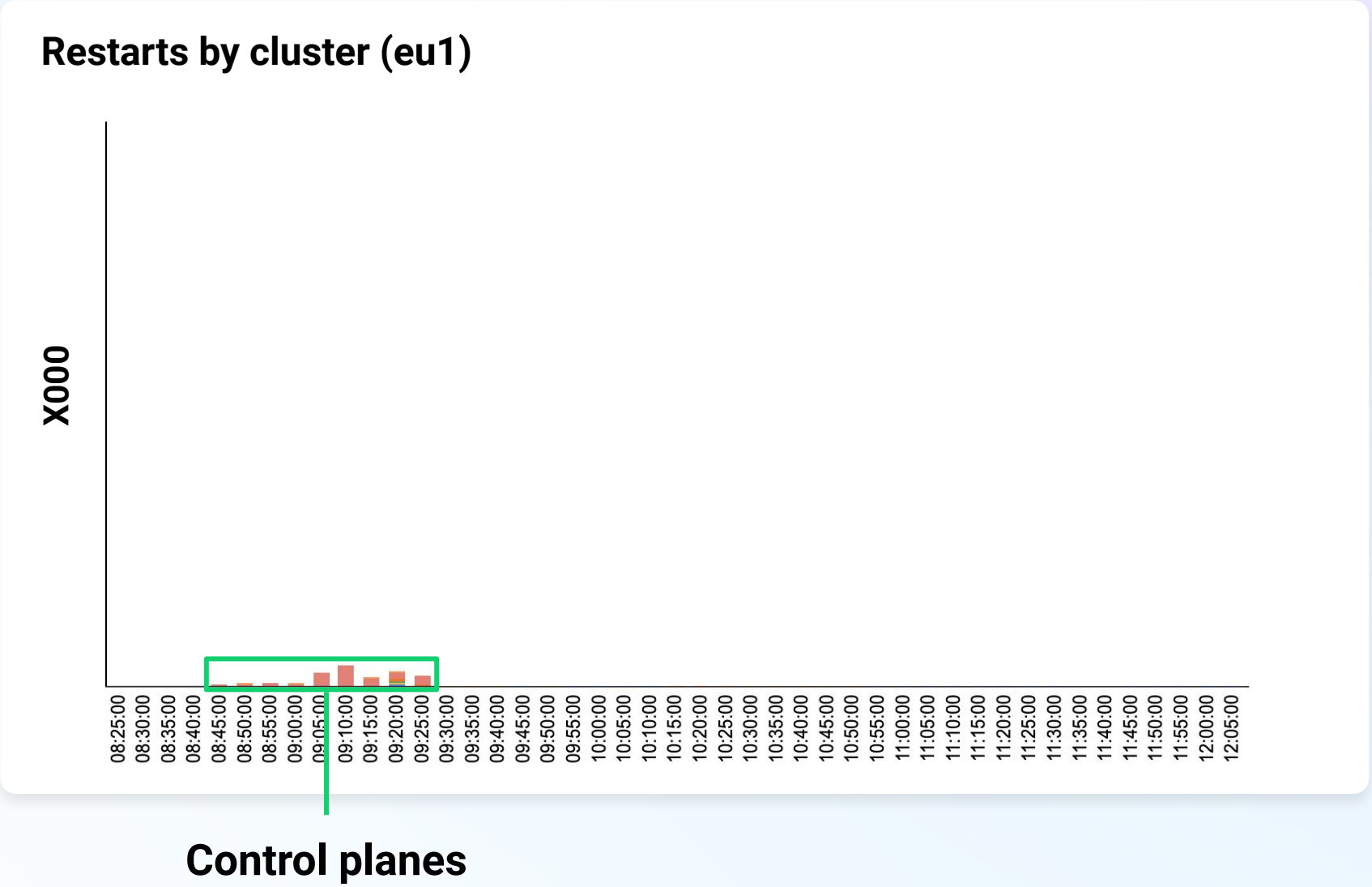
STEP 1:

Get our clusters to a healthy state

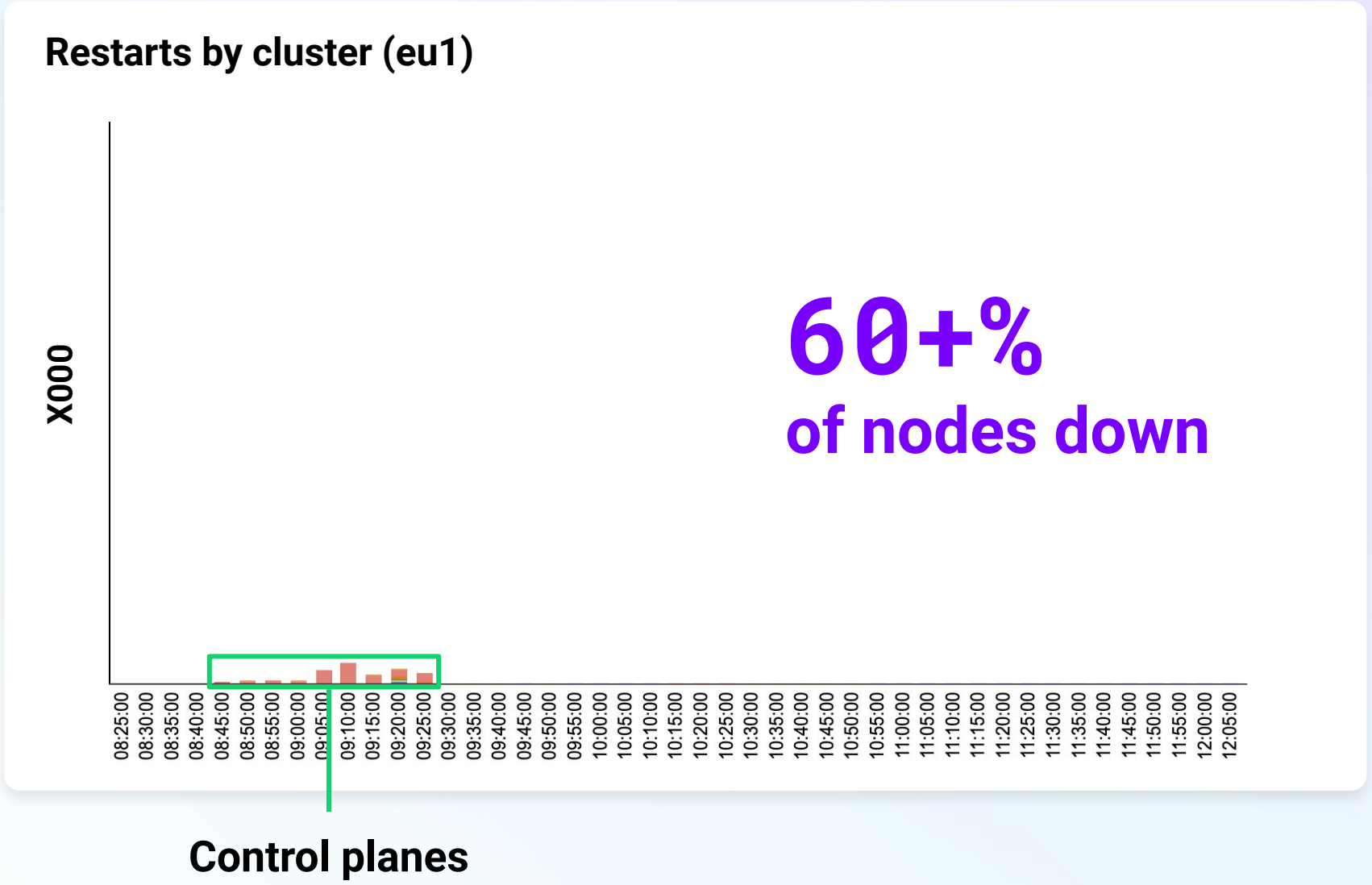
Fixing our GCP region: Simple



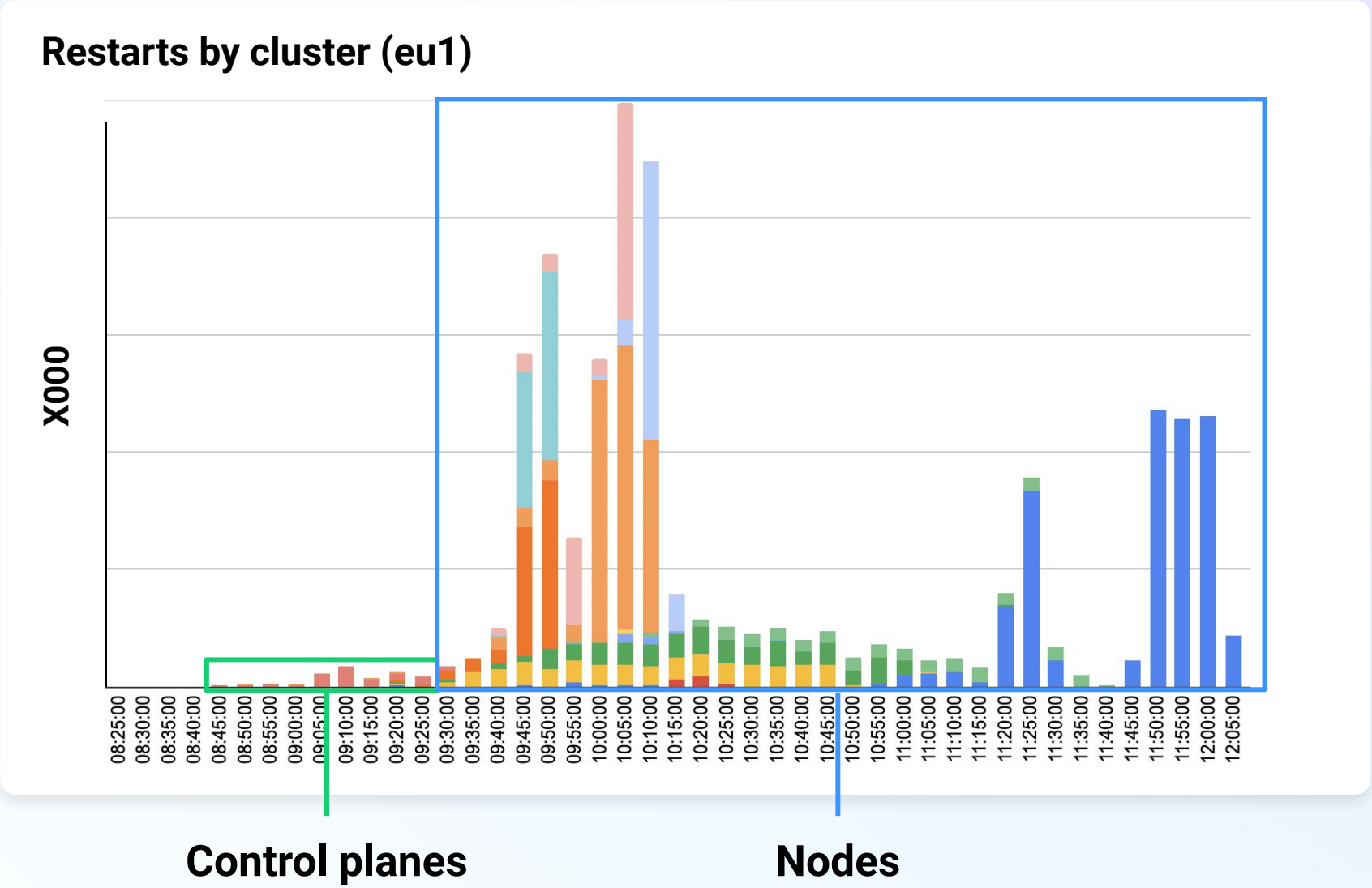
Fixing our GCP region: Simple?



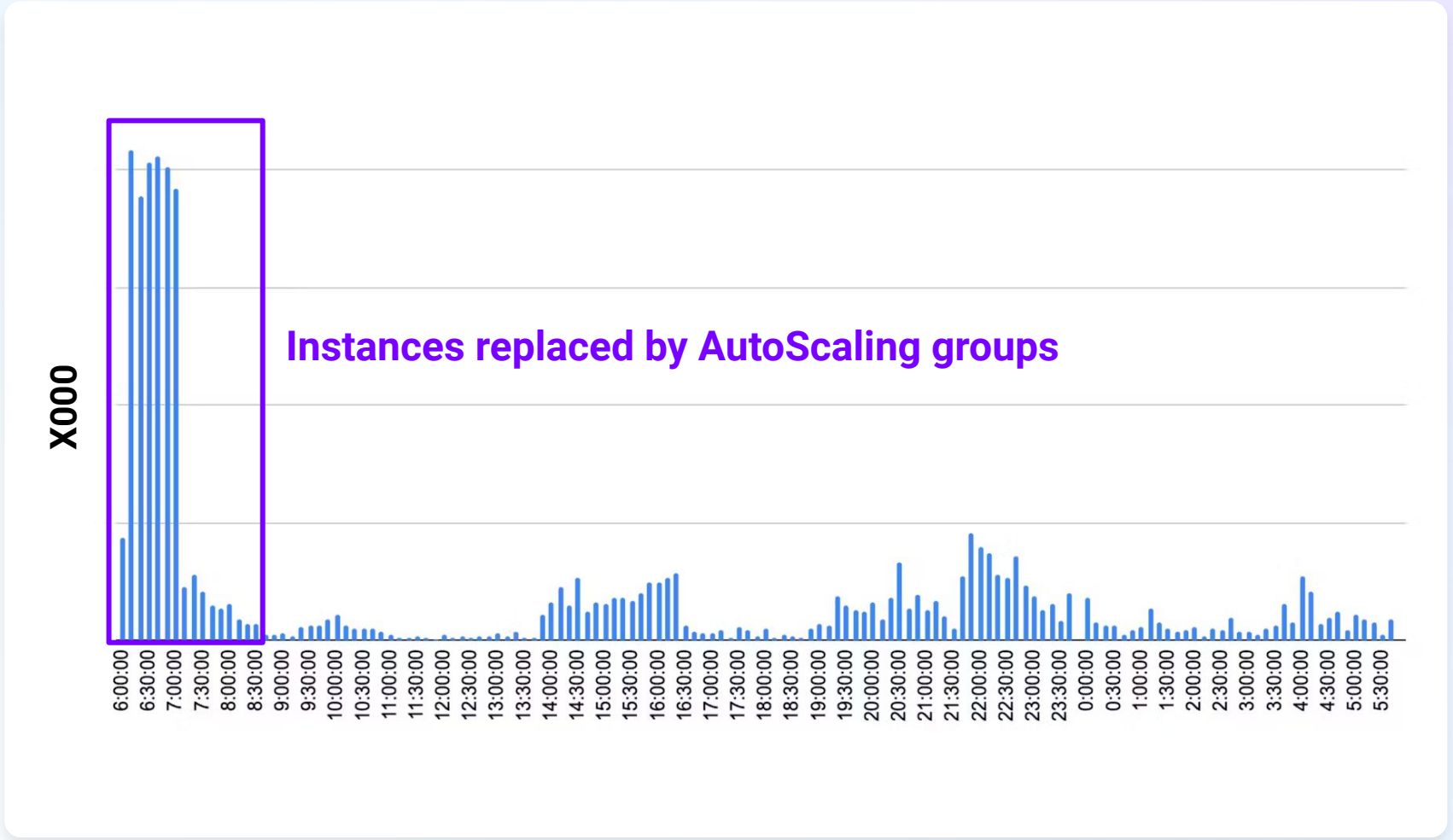
Fixing our GCP region: Simple?



Fixing our GCP region: Simple?

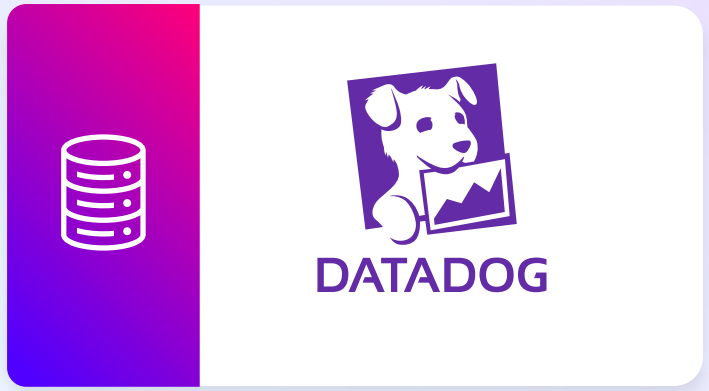


Fixing our AWS region: Auto-heal, great!



Number of instances started by AutoScaling Groups in US1

AWS: Auto-heal, not so great



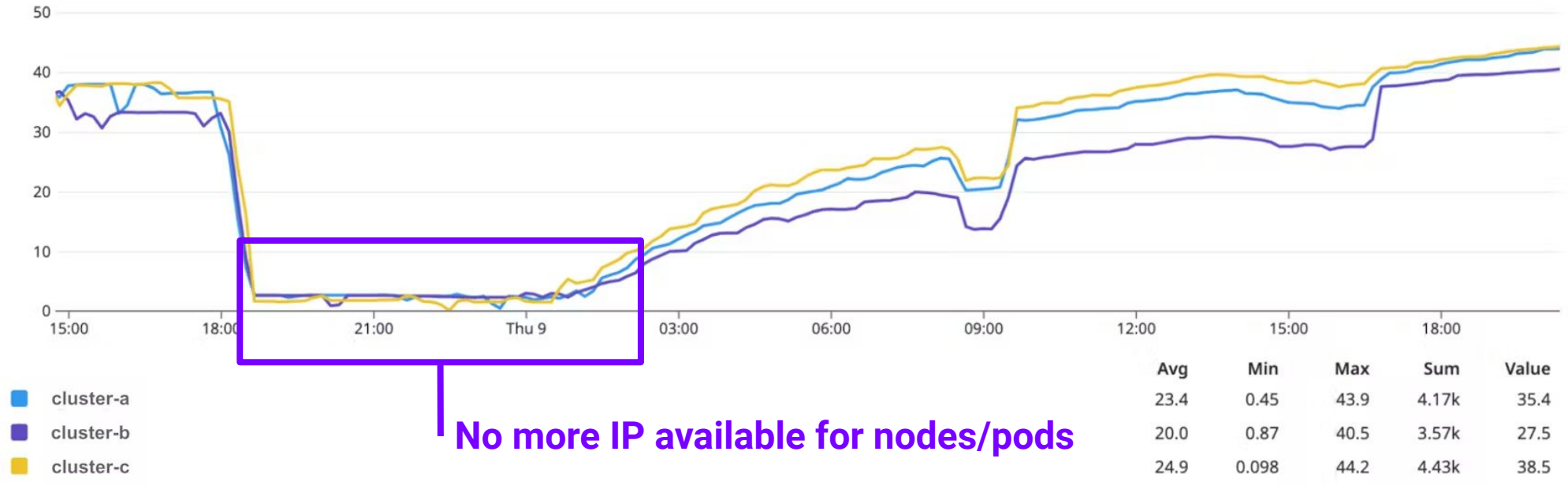
We use local disks almost everywhere

STEP 2:

Getting more capacity

Cloud is elastic, right?

Scaling services by 50+% is *not easy*

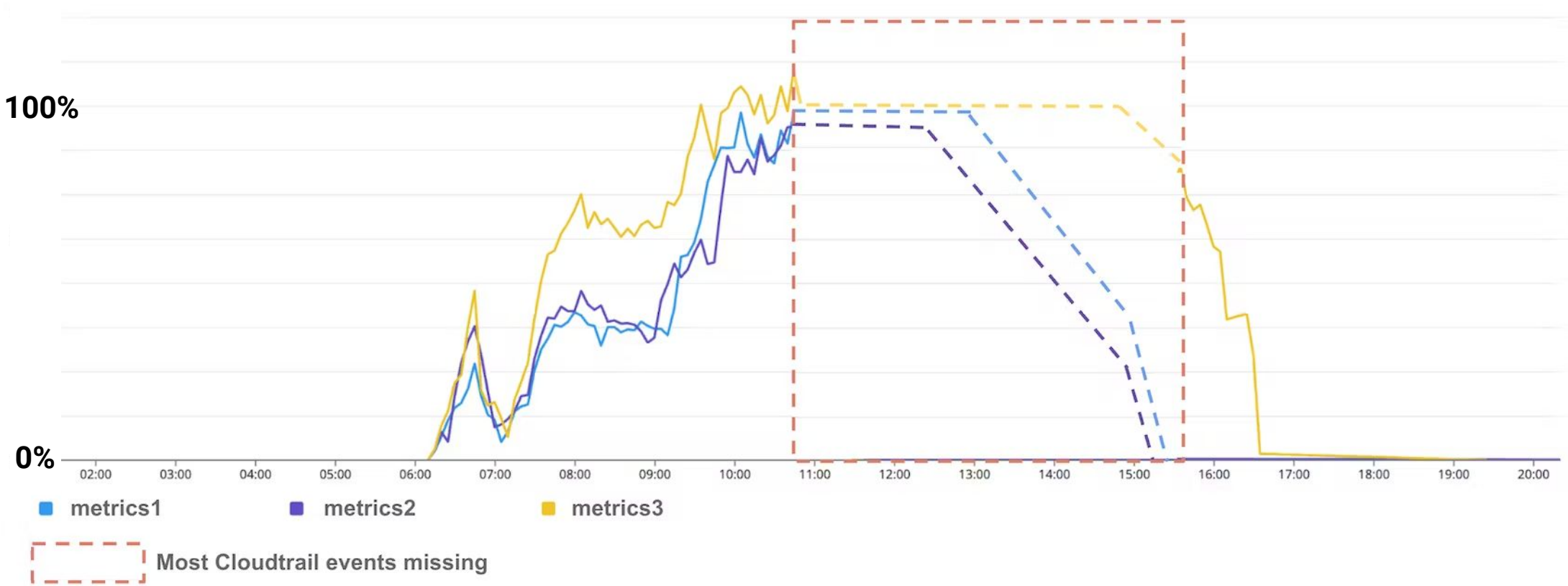


Proportion of IP available

Scaling services by 50+% is *not easy*

GCP: Number of instances running

Scaling services by 50+% is *not easy*

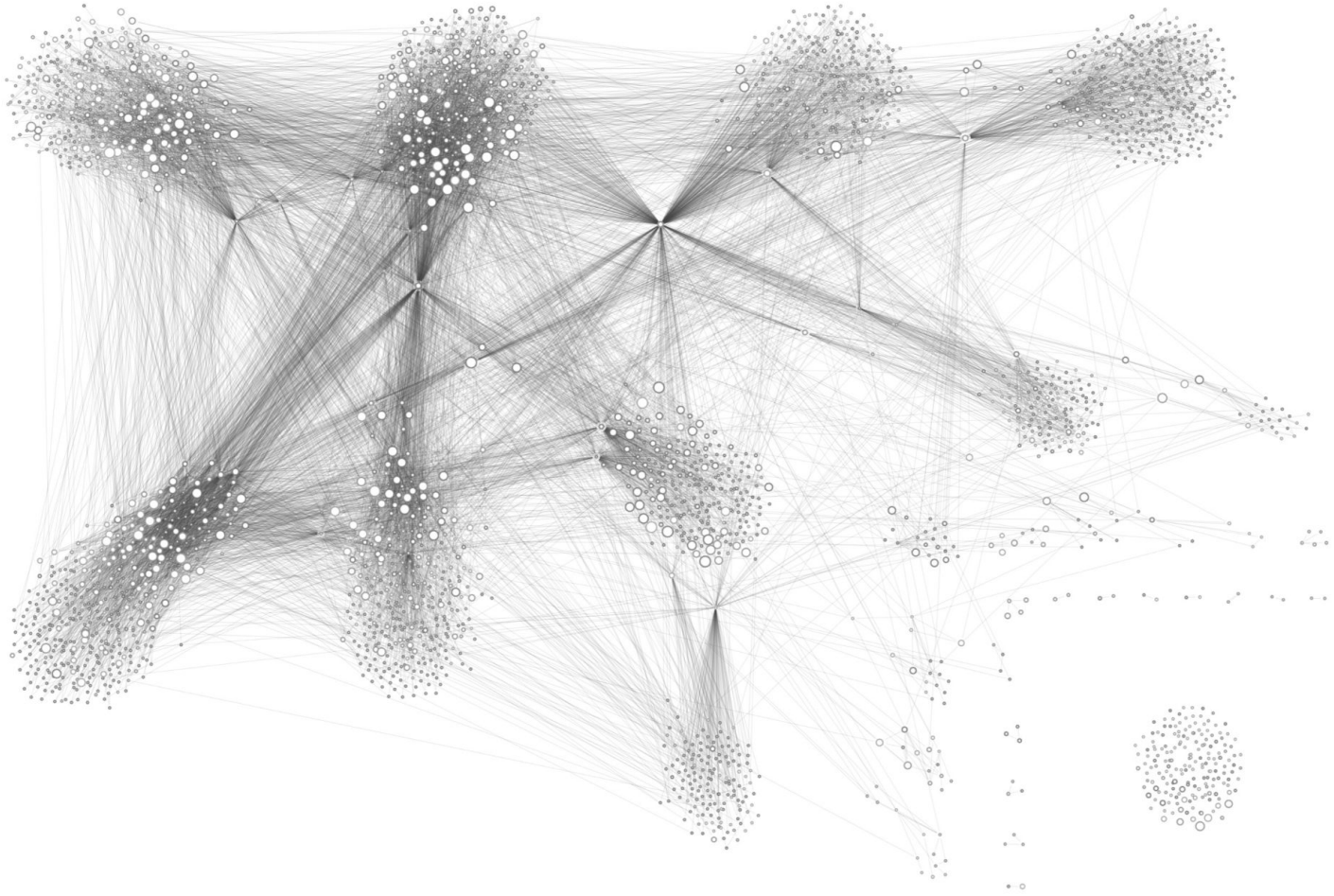


Proportion of CreateNetworkInterface API calls being rate limited

STEP 3:

Recovering applications

The Bootstrap problem



Lessons learned

Lessons learned

Common infrastructure is inherently global, even when it’s not

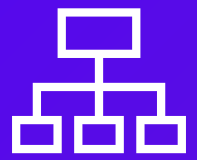
Even simple and common abstractions can leak

Unchecked, systems grow surprising dependencies

Scale is always a challenge

How will we do better in the future?

Continue to decrease blast radius:
Regional / Zonal isolation



Degrade more gracefully:
Prioritize live data

Test resiliency and incident response:
Larger scale chaos tests

Many thanks to our Partners

Thank you

Blog posts: https://www.datadoghq.com/blog/engineering/

We're hiring! https://www.datadoghq.com/careers/

laurent@datadoghq.com
hemanth.malla@datadoghq.com