
AWS
re:Invent

NET313

Maintaining security and availability on the unpredictable internet

Shailu Mishra
Software Development Manager
Amazon Web Services

Jorge Vasquez
Principal Software Engineer
Amazon Web Services

AWS re:Invent

aws

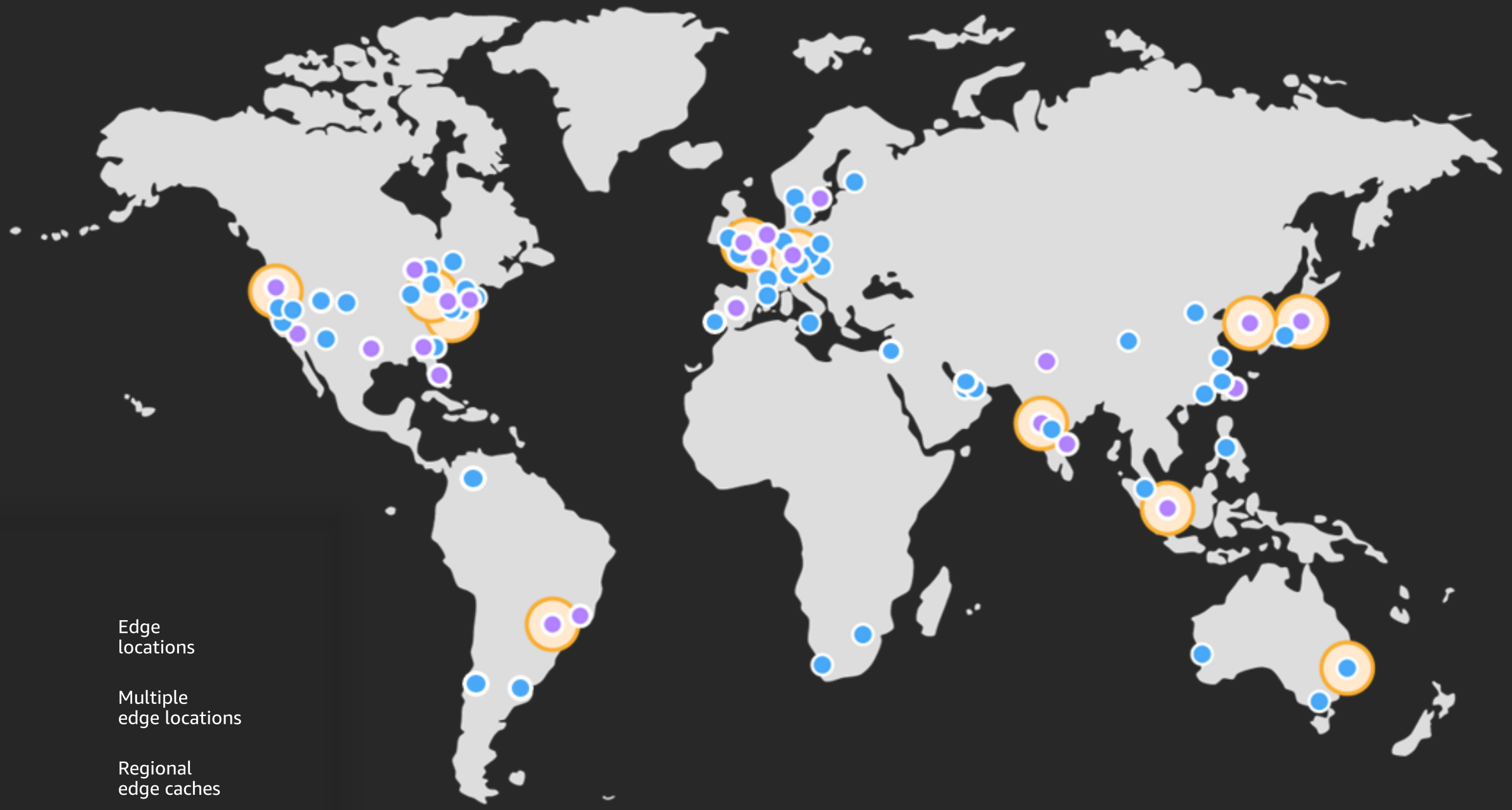
Agenda

Introduction to Amazon CloudFront

CloudFront tenets

Threats to availability

- Engineers
- Internet weather
- Flash crowds
- Dependencies


Edge
locations
Multiple
edge locations
Regional
edge caches

CloudFront tenets: In that order

Security

Availability

Performance

Failure is an option

Complex, massive-scale, highly available system

Unreliable components

- SSDs
- Power supplies
- Network links

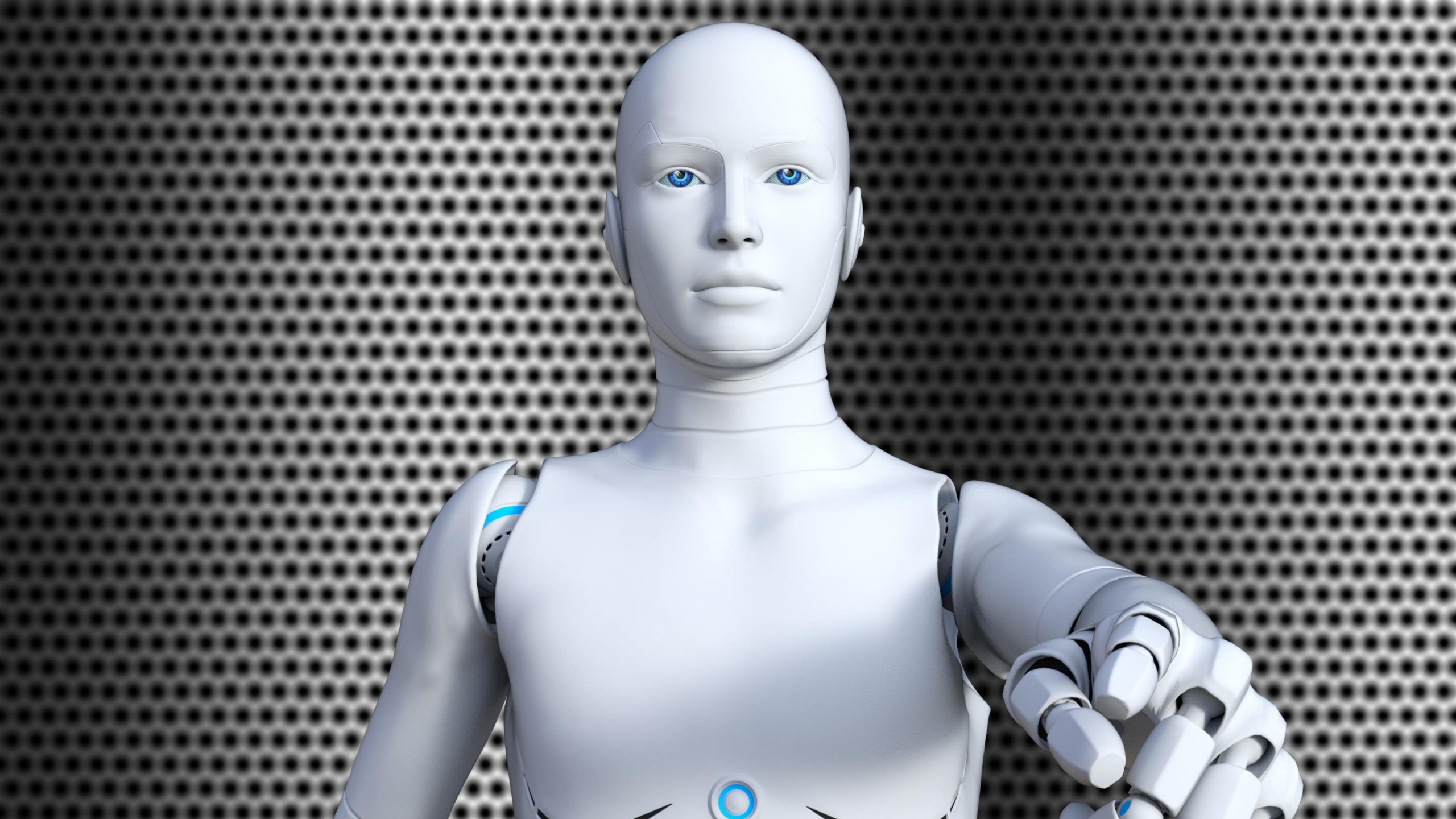


Image attributed to:
https://commons.wikimedia.org/wiki/File:Edgar_Bergen_and_Mortimer_Snerd_1941.JPG

Failure is an option

Complex, massive-scale, highly available system

Built from:

Unreliable components

- SSDs
- Power supplies
- Network links

People

- New software
- Configuration changes

Blast radius

Extent to which a failure impacts the service or its customers

Blast radius

Unreliable components

- SSDs: 1 server
- Power supplies: 1 server
- Network links: Multiple POPs in same city, for a single or multiple ISPs (10 or 100 gbps)

People

- New software: All of CloudFront
- Configuration changes: All of CloudFront

Biggest threat to CloudFront

Reducing blast radius of code changes

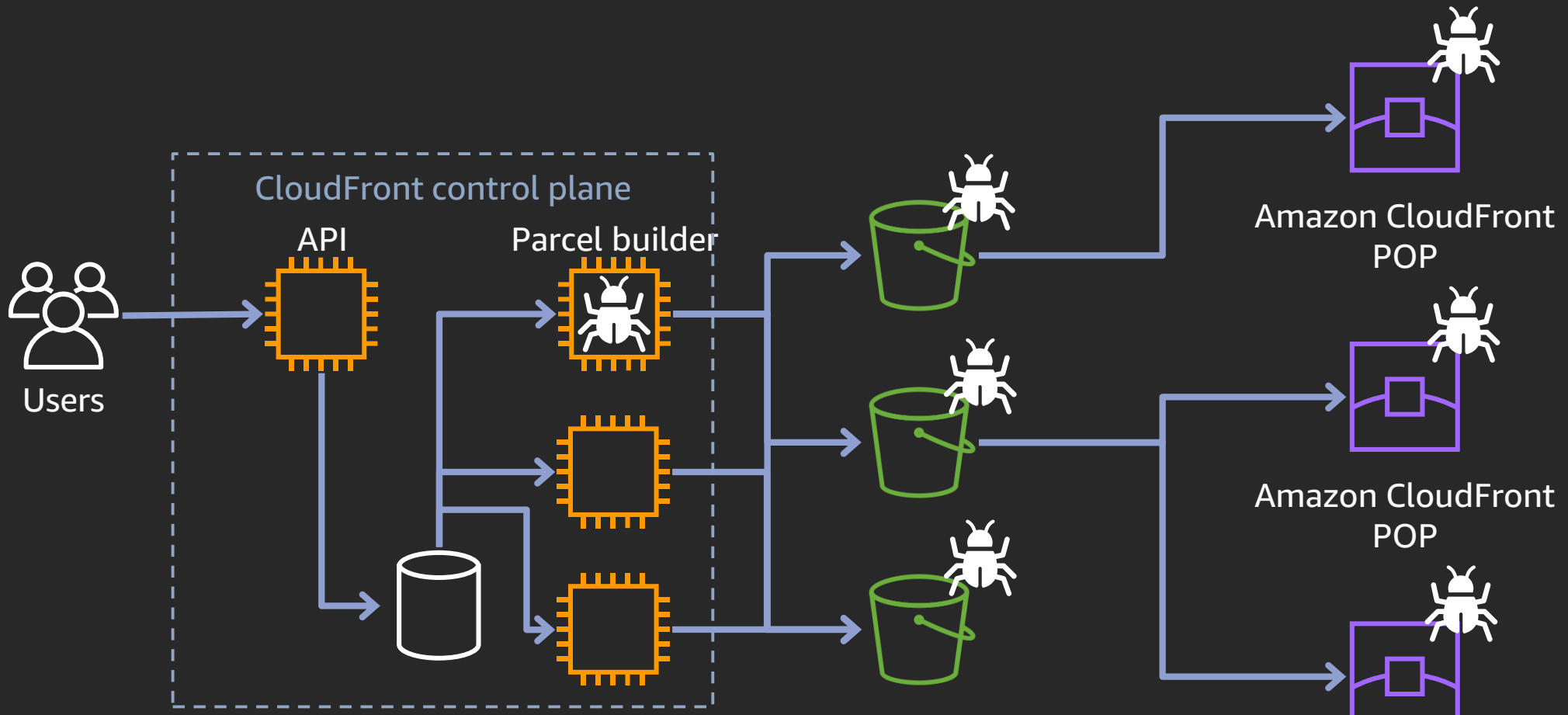
Good engineering discipline

- Code reviews
- Tests
- CI/CD
- Waved deployments

AWS CodePipeline

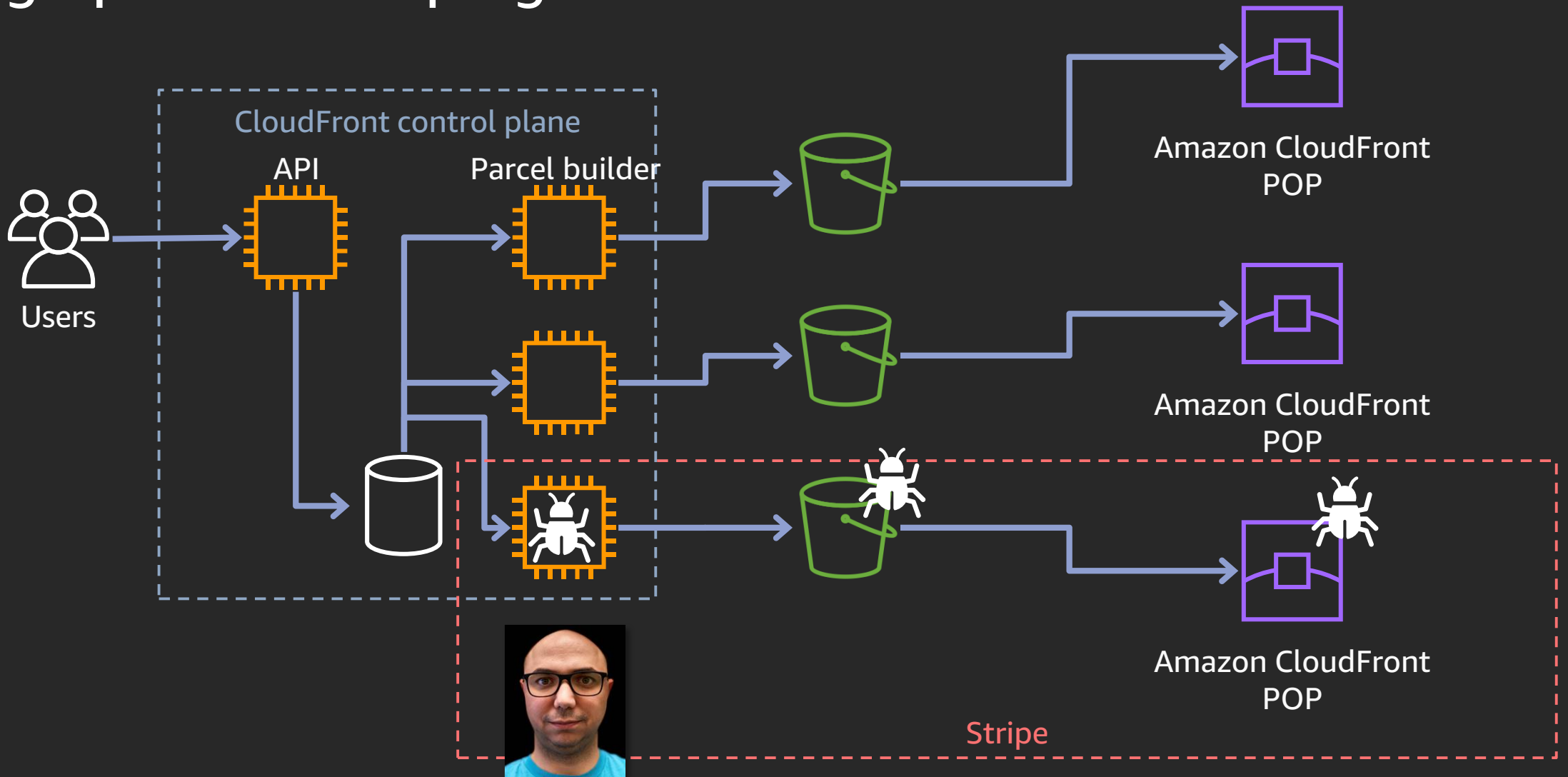
AWS CodeDeploy

Reducing blast radius of code changes



Reducing blast radius of code changes

Design pattern: Striping



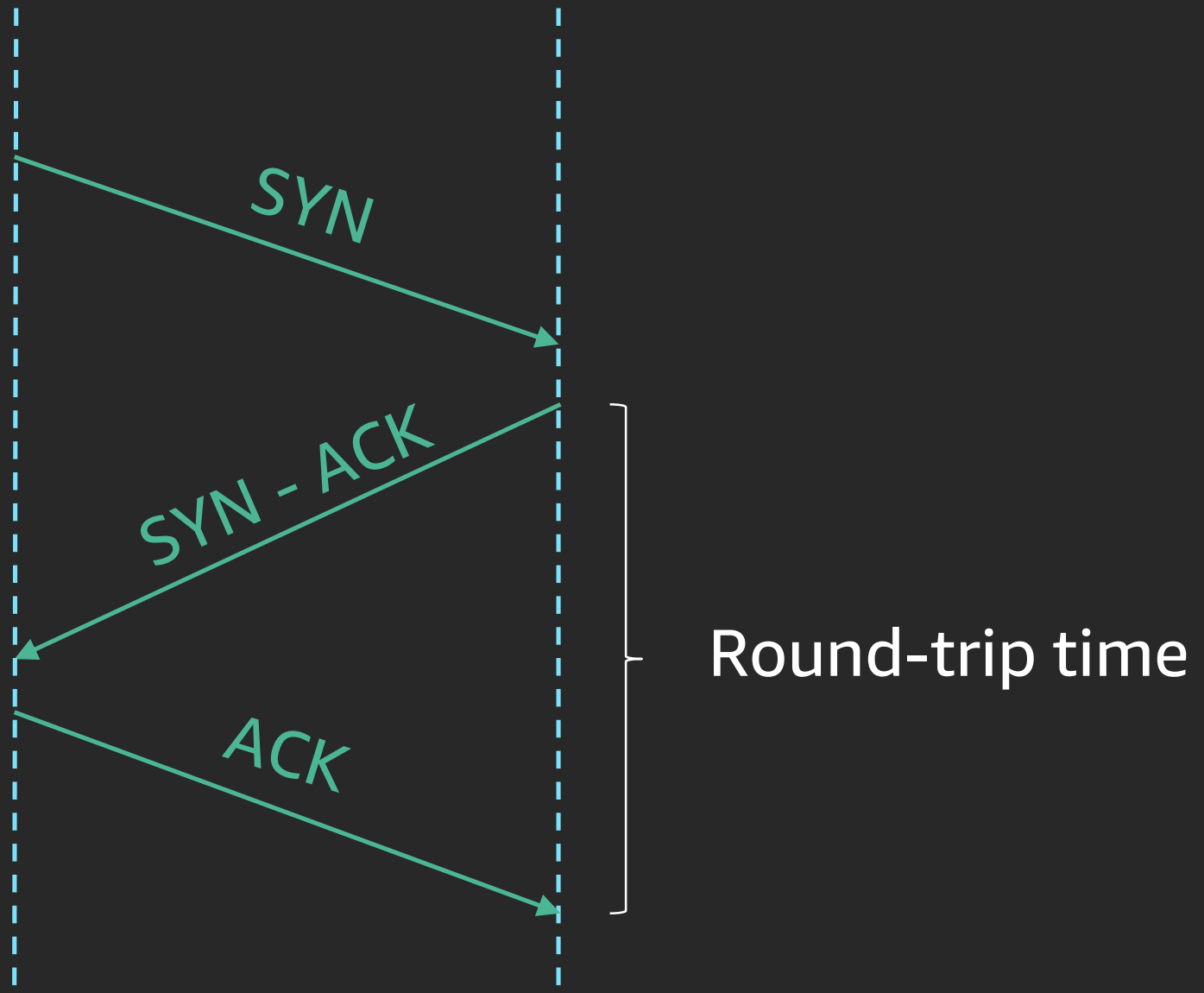
Case study: Poisoned latency measurements

Context

- CloudFront measures latencies from all networks towards all POPs
- Data is aggregated every five minutes and sent to DNS servers
- DNS servers use it to route traffic to the best POPs

The change

- New entries for collections of POPs in the same location



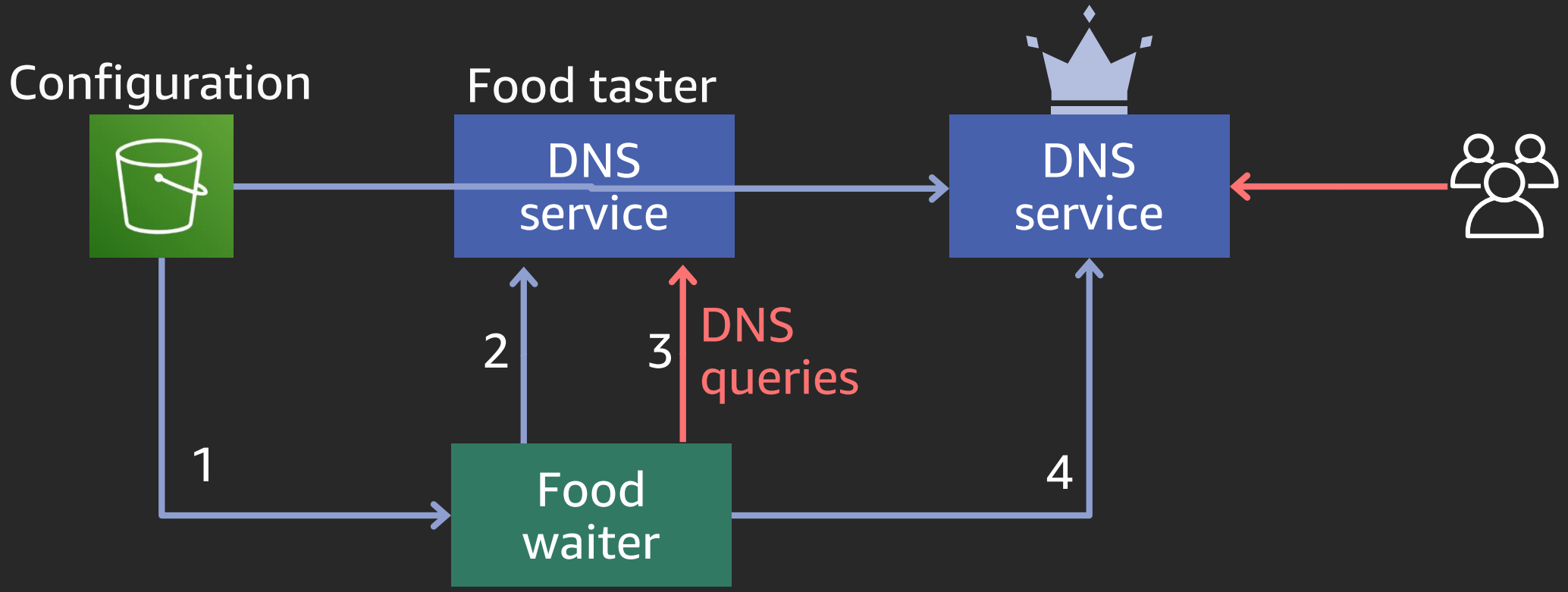
Case study: Poisoned latency measurements

Step by step

1. Deployment of the change
2. Velocity check failure (file size grew too much, too fast)
3. Operator overrode check, the failure was planned
4. New files get sent to all DNS servers
5. Alarms fire: All change propagation to DNS servers has been halted

Case study: Poisoned latency measurements

The food taster



Case study: Bad latency measurements

What went wrong?

- There was a bug
- Food taster was choking, no configurations could be ingested

What went well?

- Food taster did its job, died in place of the king DNS server
- No data plane impact

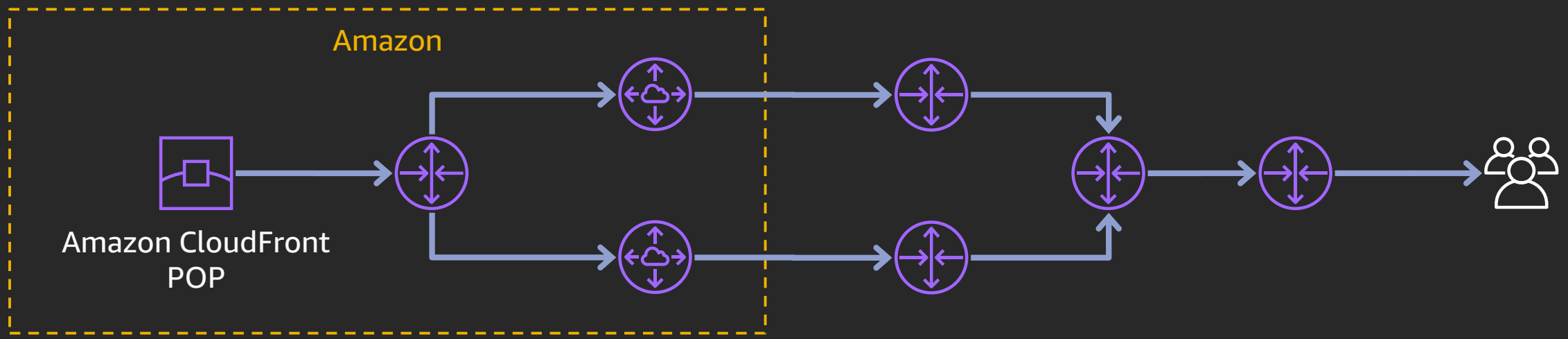
Bad weather


FALL:
46 F, grey, gloomy, probably raining.
(Basically a Tim Burton movie)
K..K..KK.. KILL ME
WINTER:
(See fall)
SPRING:
(See winter)
..KK.. KILL ME
2 SECONDS LATER
K..K..KK.. KILL ME
SUMMER:
Finally shows up in late July.
The whole city gets all manic about how AMAAAAAZING our weather is
OH GOD... IT'S HERE
IT'S FINALLY HERE

Bad weather: Links congested

Amazon-controlled links

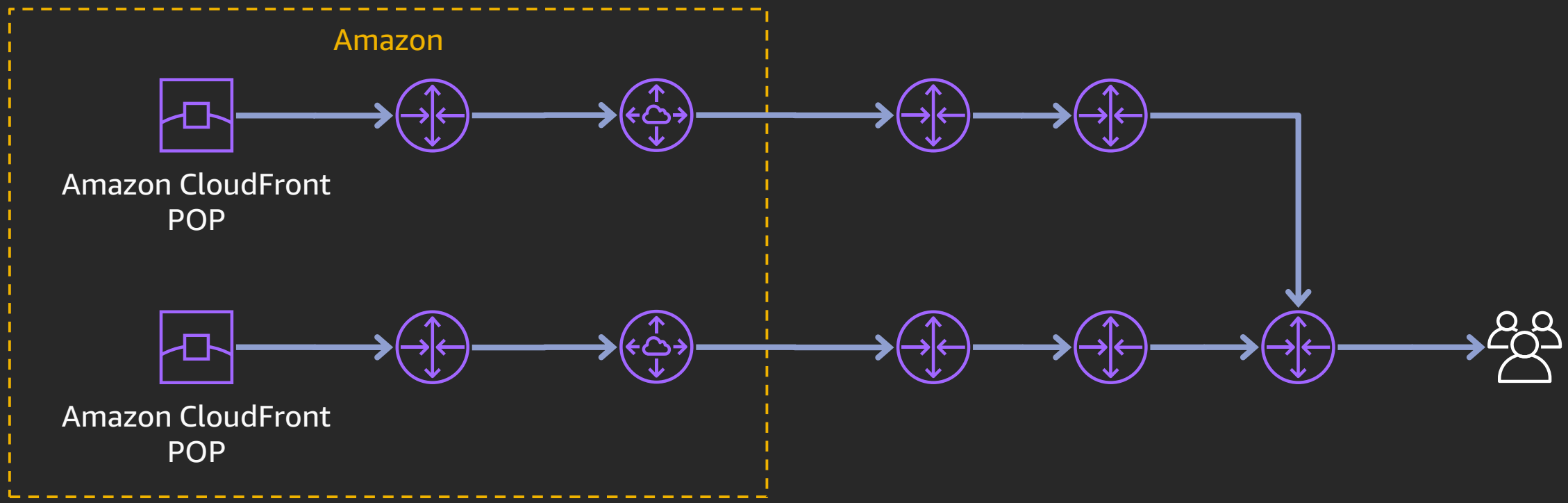
- Active network capacity management
- Automatic traffic engineering
- Current traffic ≠ Demand
- Last resource (<1%): Engage engineers



Bad weather: Links congested

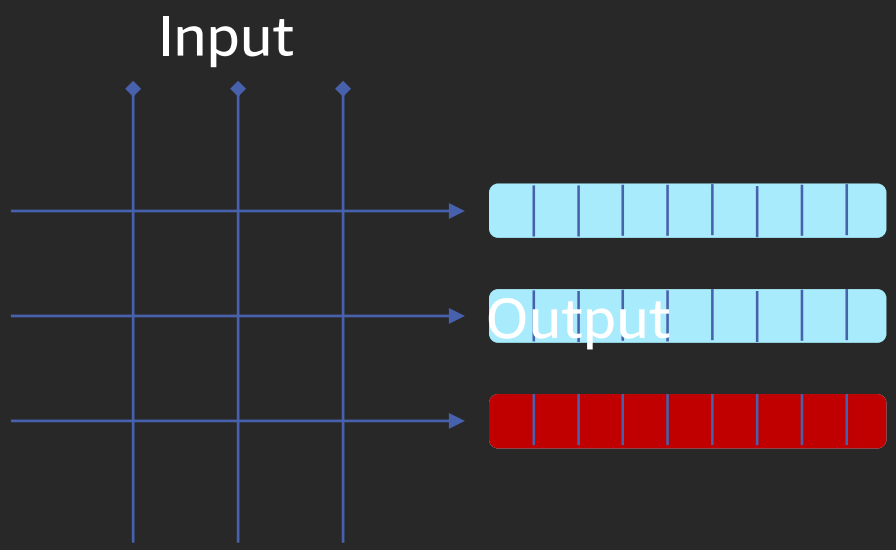
Third-party links

Latency measurements



Bad weather: Links congested

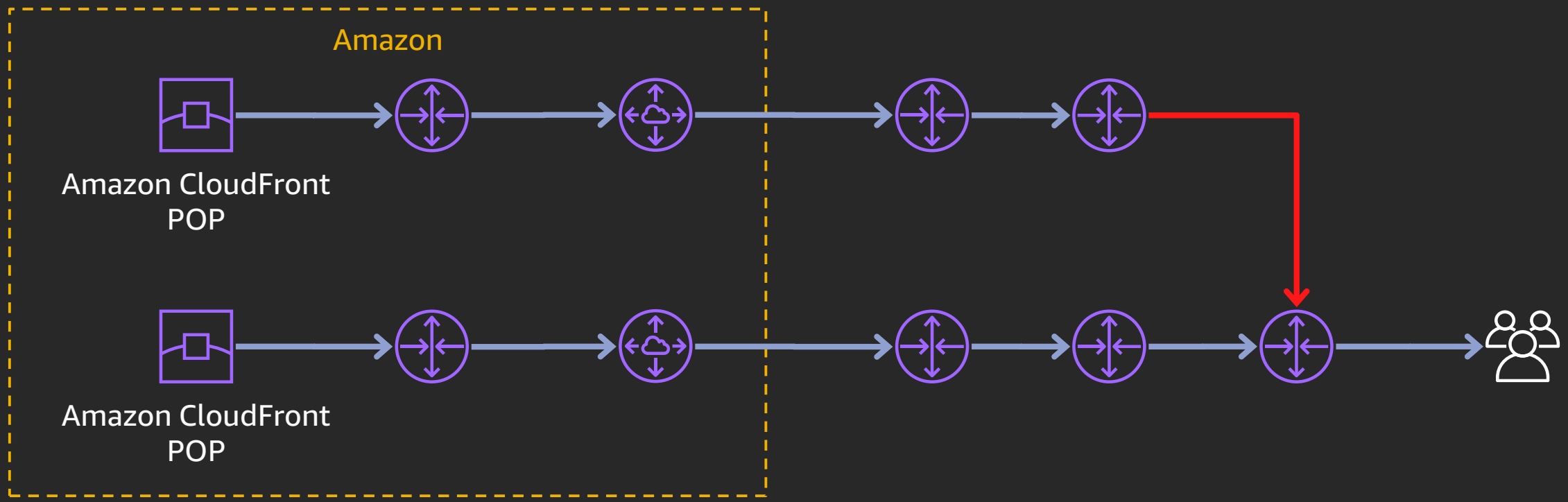
Routers & latency



Bad weather: Links congested

Third-party links

Latency measurements



Bad weather: Links down

"On June 7, 2018 between 18:38 and 21:06 UTC, a fire in a manhole caused a fiber cut on multiple fibers in the New York area, resulting in 2 hours, 28 minutes of congestion."

Internal Postmortem

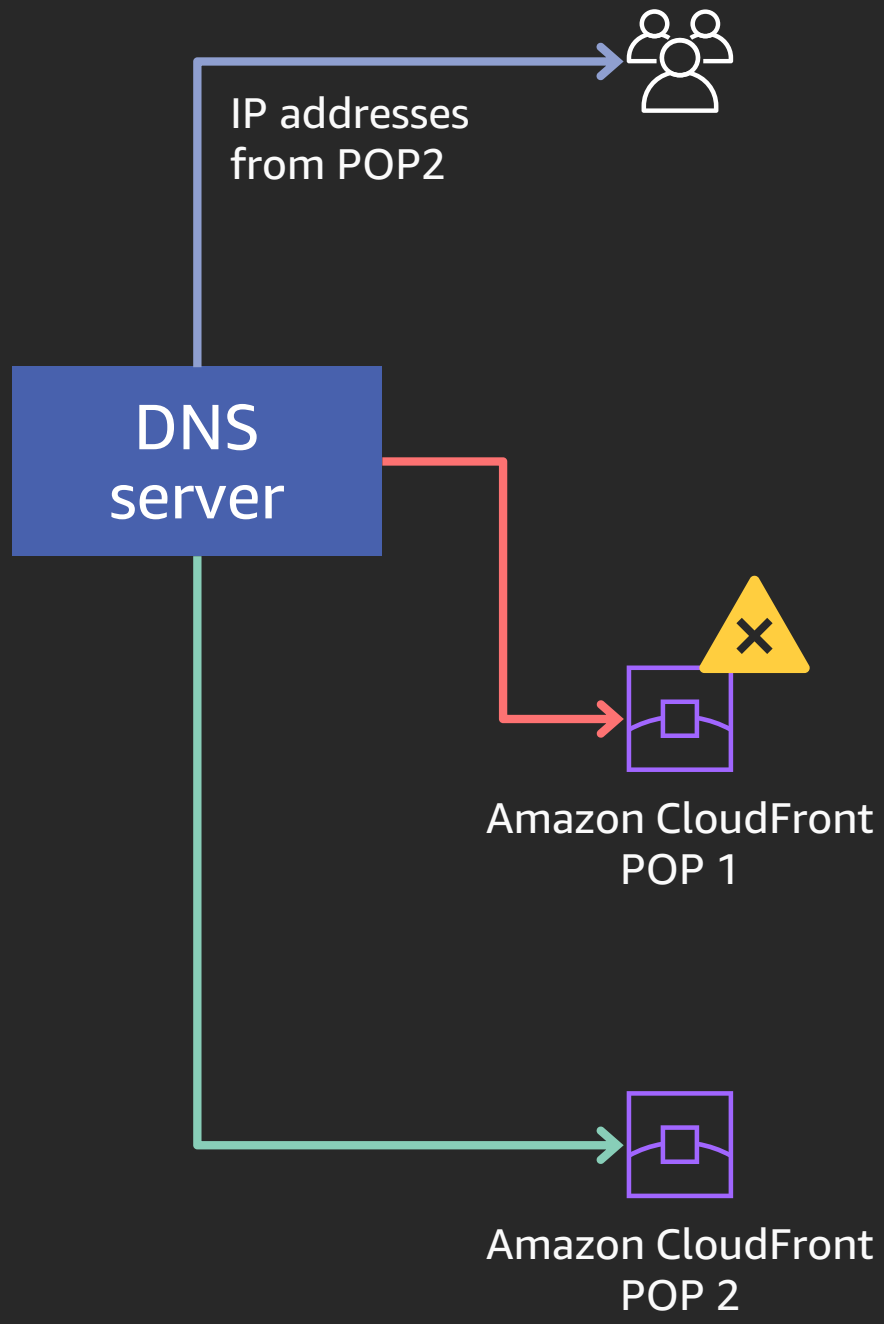
Bad weather: Links down

Single links

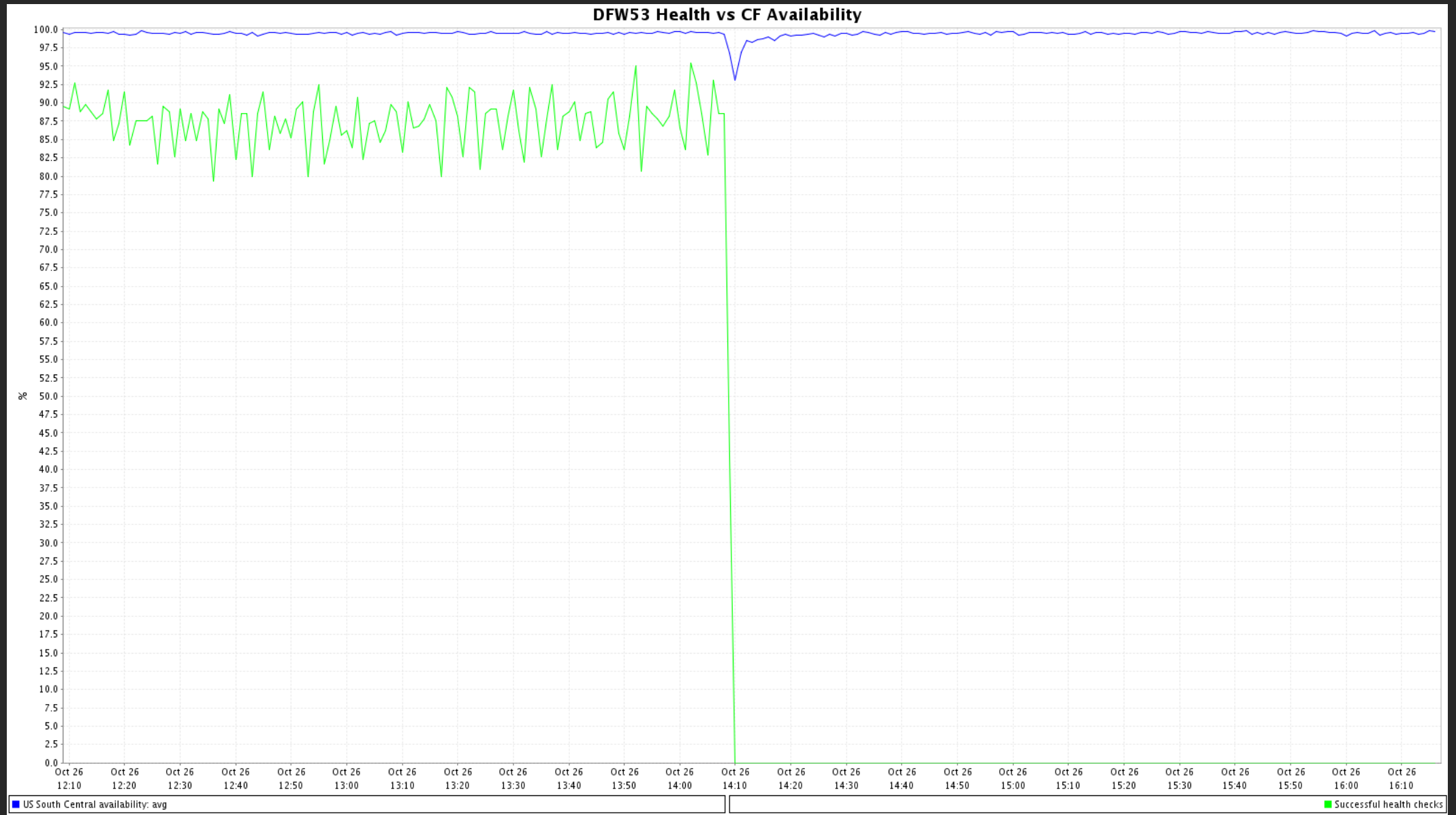
- Common
- Redundancy of links and routers
- BGP failover
- Transforms an availability issue into congestion

Multiple links/isolation

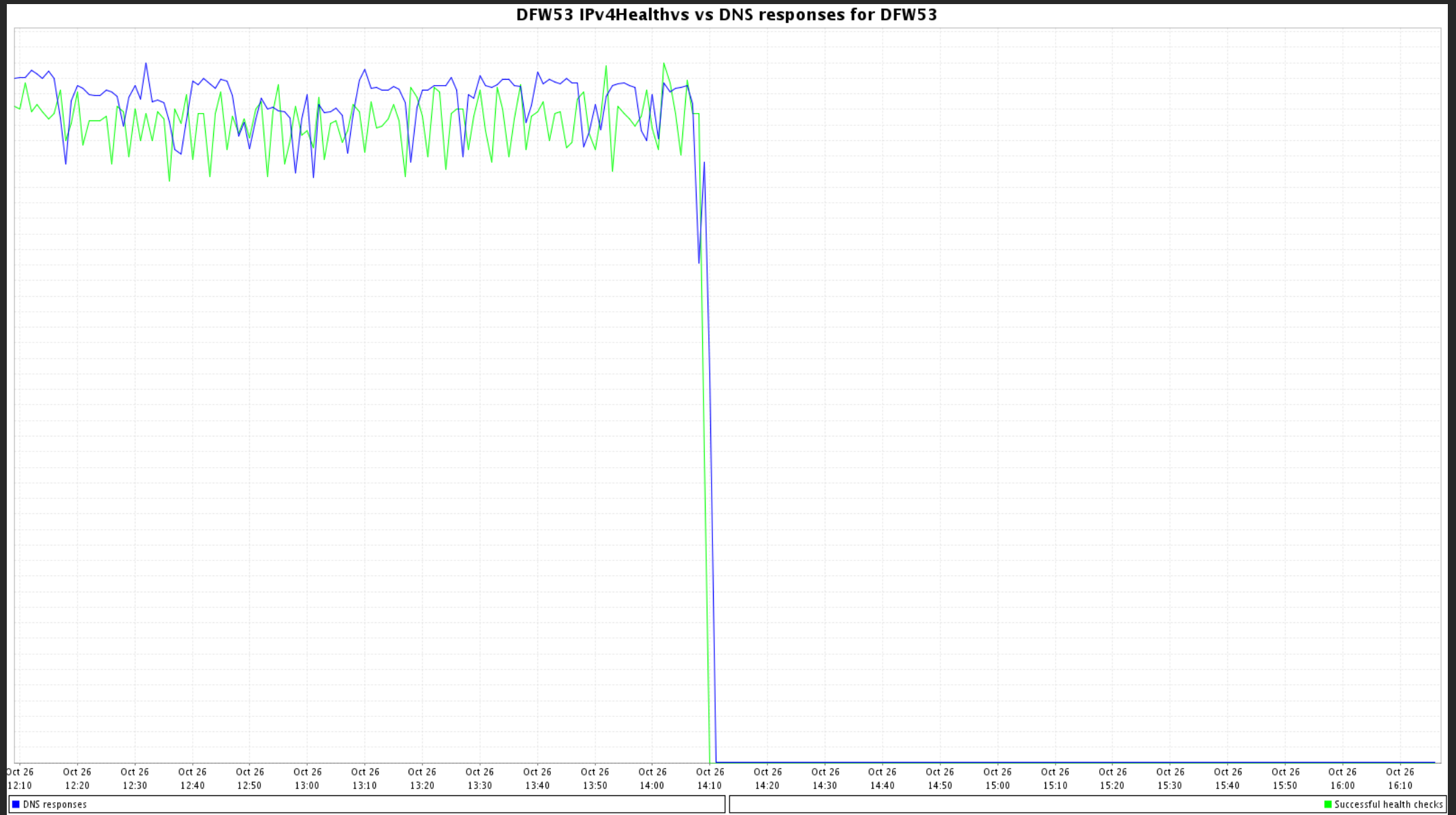
- Rare
- Automated health checks react in less than one minute
  - Achievable with Amazon Route 53
- Manual engineer action after the fact



Case study: POP unreachable



Case study: POP unreachable



Bad weather: Environmental issues

Very rare, just like multi-link failures

- Fire suppression CO_2 equipment increased datacenter pressure and made the hard drives' heads crash-land on the platters
- Power outages
- Cooling failure

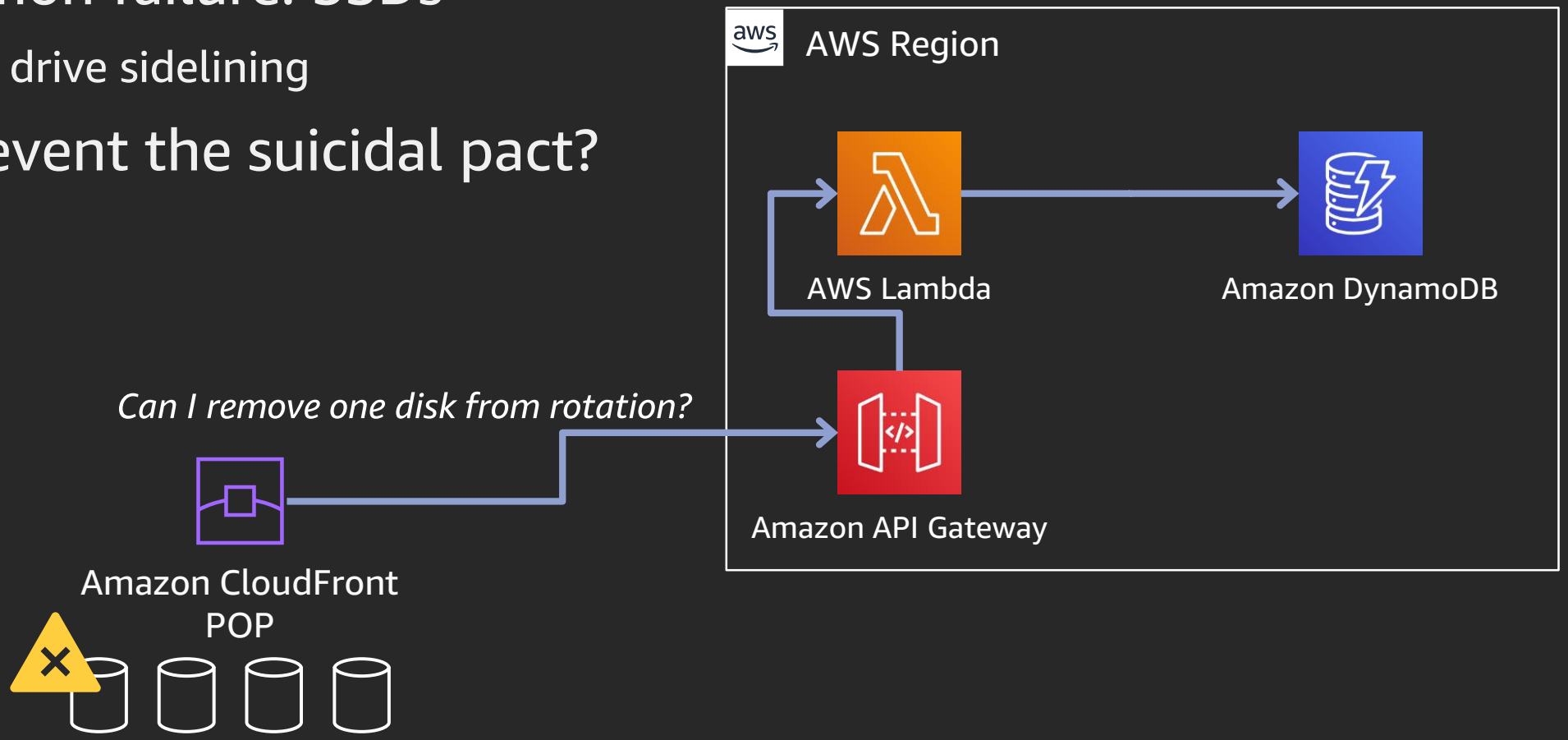
From an operational perspective, it's just like a multi-link down

Bad weather: MTBF

Most common failure: SSDs

- Automated drive sidelining

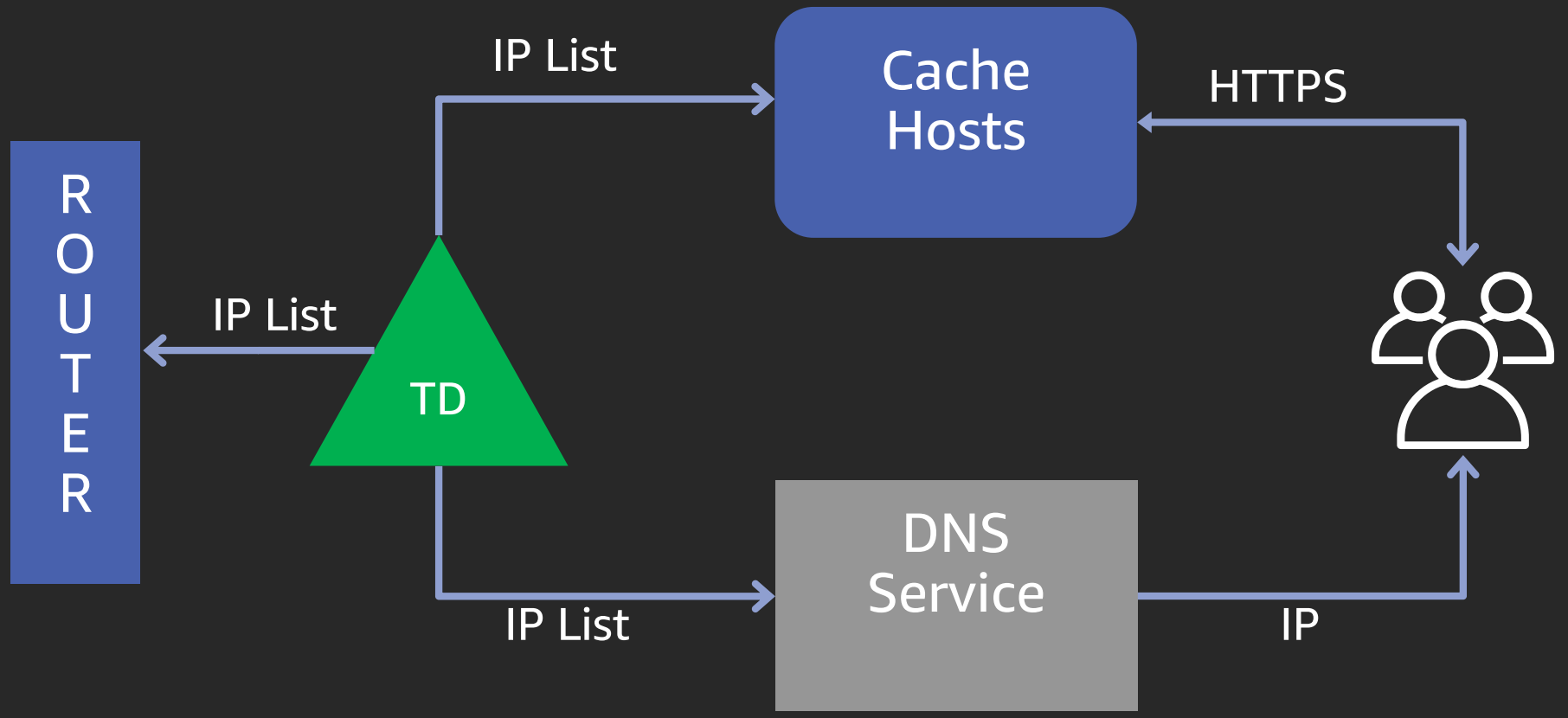
How to prevent the suicidal pact?



Defense in depth

Biggest threat to CloudFront

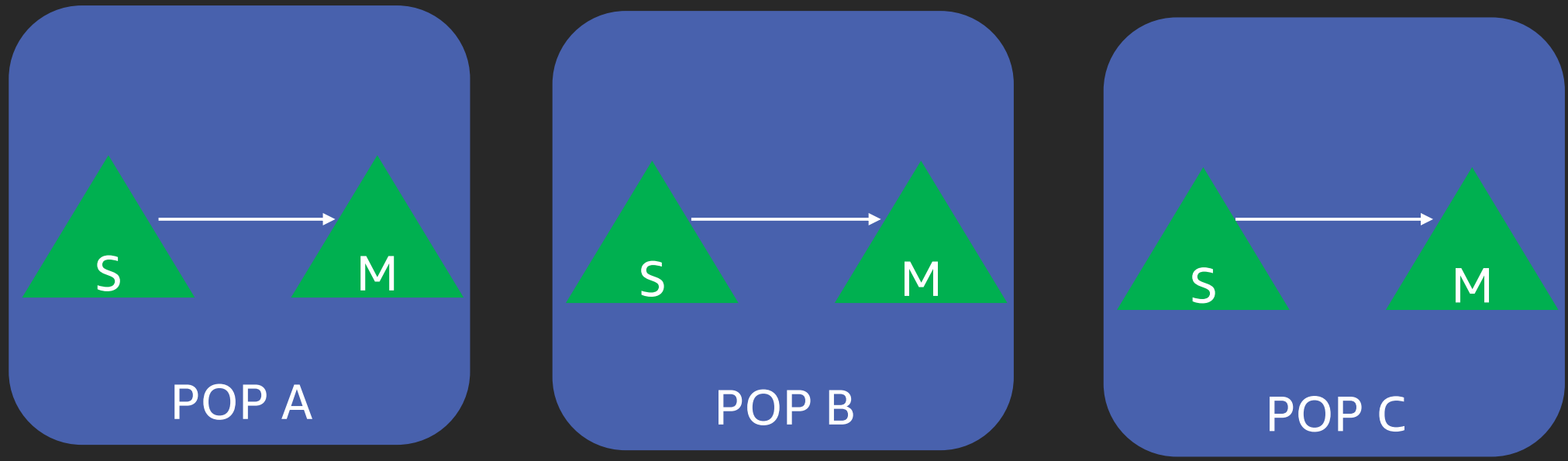
Case study: Traffic Director world domination



Case study: Traffic Director world domination

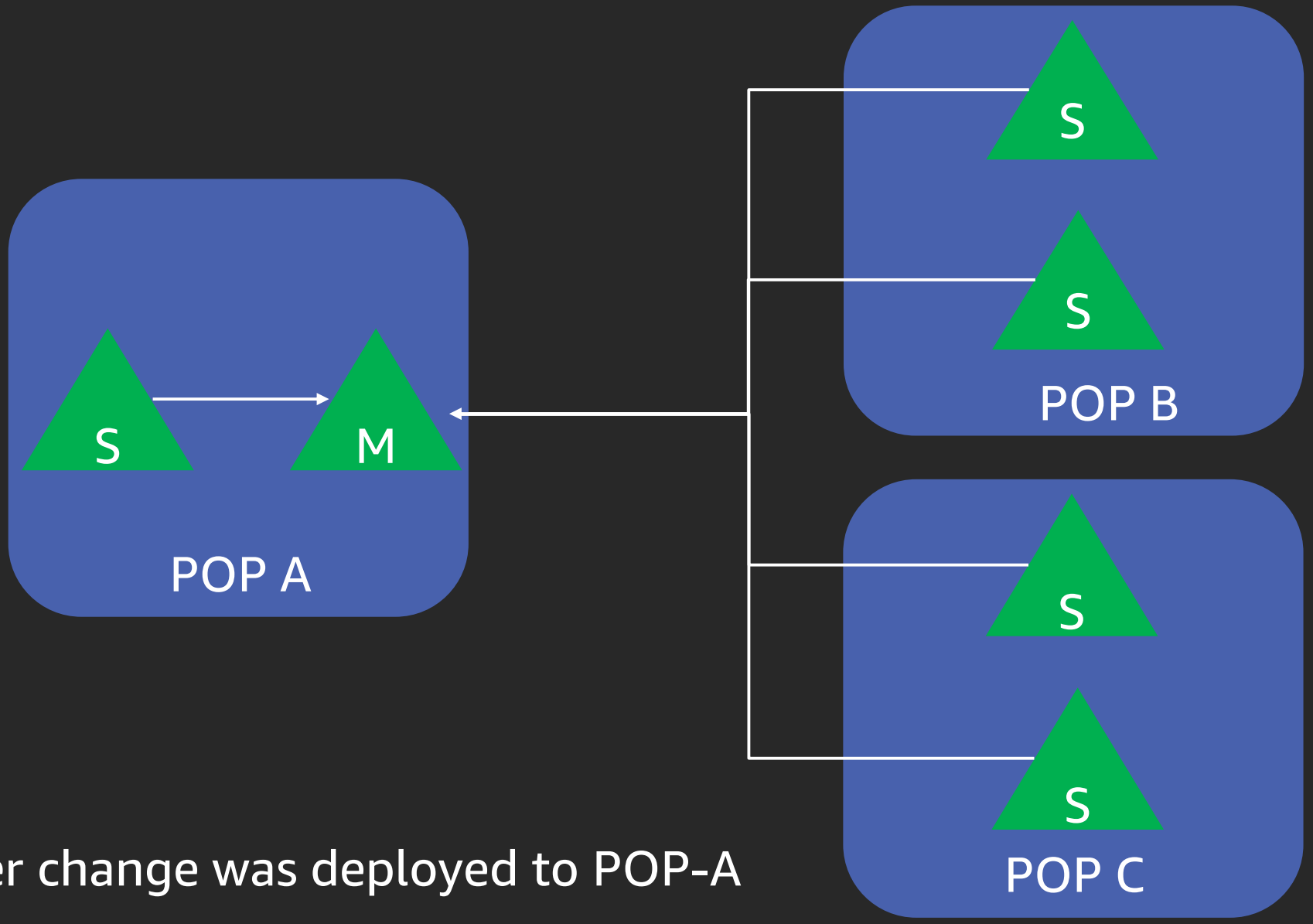
- Unintended deployment of pre-preproduction code to one POP-A
- POP-A started announcing itself as master to other TD hosts outside the POP-A
- TD hosts from other POPs started using the IP list from POP-A

Case study: Traffic Director world domination



Before change was deployed to POP-A

Case study: Traffic Director world domination

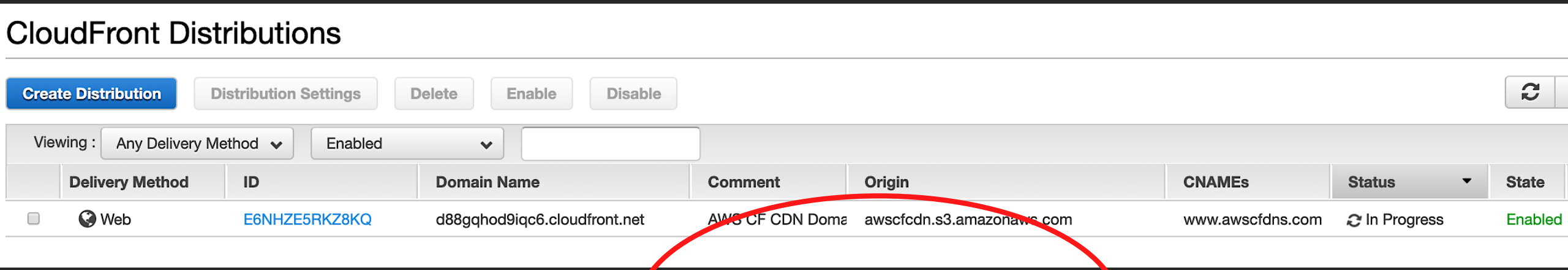


After change was deployed to POP-A

Case study: Traffic Director world domination

- Defense in depth
  - DNS servers failed validation of IP list
  - TD validates routes it publishes to routers
  - All change propagation was halted
  - No data plane impact

Case study: Traffic Director world domination



"A flash crowd is a large spike or surge in traffic to a particular Web site. Major news web sites experience this problem during major world events. Sometimes unpopular Web sites instantly become extremely popular after being mentioned in a popular news feed, also called the Slashdot effect."

Ismail Ari, Bo Hong, Ethan L. Miller, Scott A. Brandt, Darrell and D. E. Long
Managing flash crowds on the internet

Case study: Flash crowd

Examples:

- Super Bowl, Game of Thrones: Surge in traffic at the start of show
- Interactive TV game shows with companion app
- Synchronized internet devices downloading content on fixed schedule (don’t do it ☺)

Case study: Flash crowd

How does CloudFront handle a flash crowd

Before the event: Run POPs with some spare capacity

Before the event: Manual configuration to disperse traffic

Within seconds: Customer level throttling, protects against busy neighbors

Within 10 minutes: Flash crowd mitigator service kicks in and disperses traffic

Within 20 minutes: Routing system generates new routes based on latency and load

Jitter

Case study: RAM disk filling up

Context

- Configuration files in RAM drives for faster access
- Configuration files grow as the service grows
- One day, space ran out

What worked well?

- Every single host has a different RAM disk size
- Event was fully mitigated by sidelining the smallest host

Jitter

Should be applied to:

- Periodic timers
- Retry timers
- Artificial limits (e.g. ram disk size, timeouts)

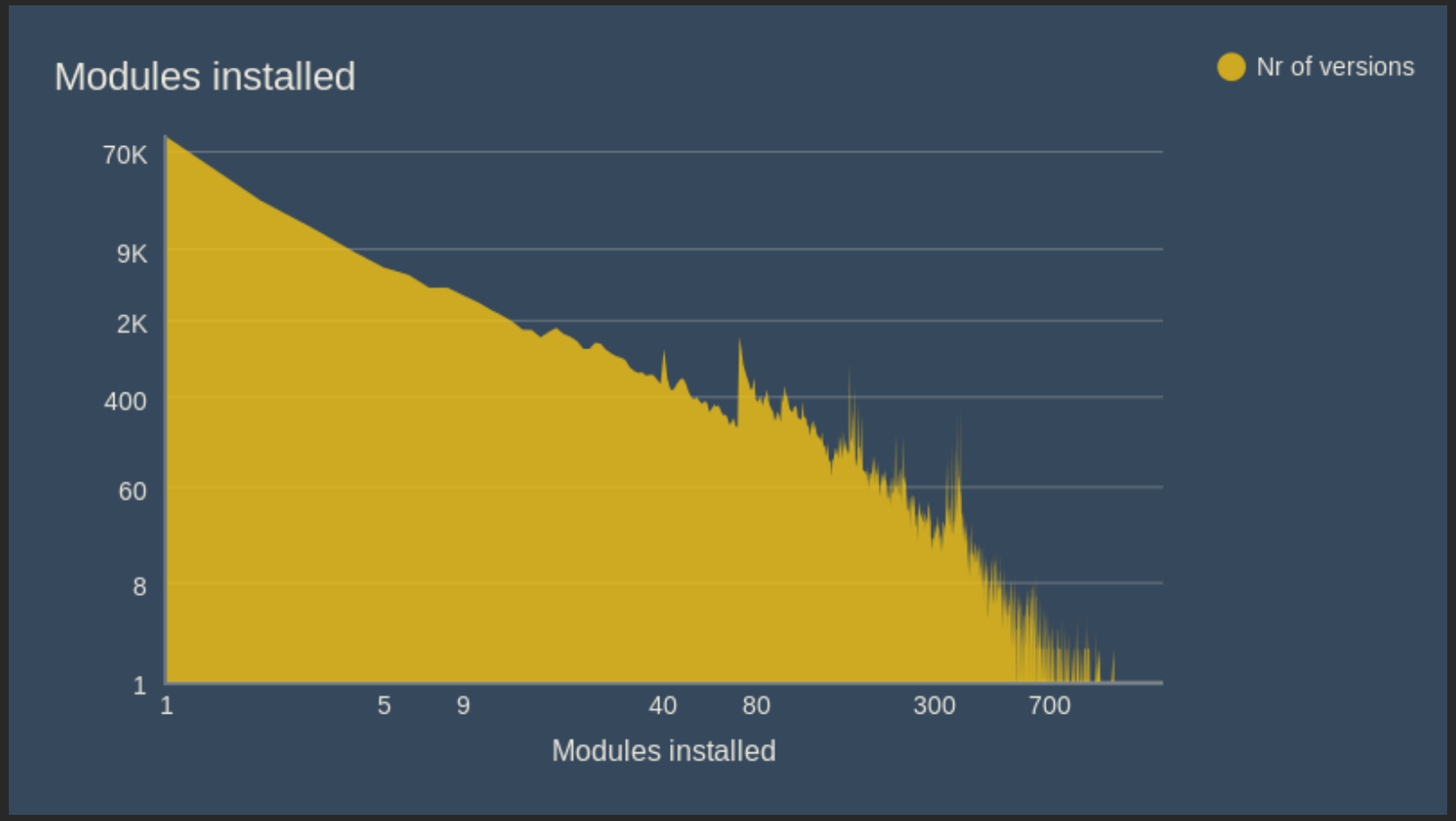
Deterministic

- Example: hash(hostname)
- Avoids hard-to-debug random behavior

The larger picture: Monoculture

Dependencies

Modern software engineering practices



Dependencies

Hardware

- Is also software (CPUs microarchitecture is an example)
- Fails on its own

What about cloud services?

- Cloud = Hardware + Software + People

Dependencies are a necessary evil

- Don’t take them lightly
- Understand how you depend on them and what happens if they fail

Recap

Good CI/CD practices: waves and striping-based deployment

Velocity checks

Defense in depth: validate data and config before ingesting

Bad weather: use Route 53 health checks

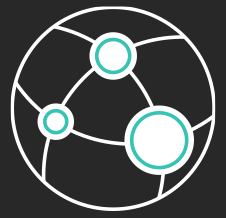
Prevent suicidal pacts

Introduce jitter to prevent synchronized failures

Dependencies: use with caution

Learn networking with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate networking skills



Free digital courses cover topics related to networking and content delivery, including Introduction to Amazon CloudFront and Introduction to Amazon VPC

Validate expertise with the
AWS Certified Advanced Networking - Specialty exam

Visit aws.amazon.com/training/paths-specialty

aws training and certification

Thank you!

AWS re:Invent

aws

Please complete the session survey in the mobile app.