aws INNOVATE

MODERN APPLICATIONS EDITION

27&28 October 2021

Improve observability with AWS App Mesh and Amazon ECS

Thirumalai Aiyalu (Thiru)

Partner Solutions Architect
Amazon Web Services

By the end of this session, we can understand...

- What is observability?
- What is Amazon Elastic Container Service (Amazon ECS)
- Observability with Amazon ECS
- Observability redux, with Amazon ECS, AWS App Mesh, and AWS X-Ray!
- End-to-end observability
- Demo!

Why is observability so important?

We need to talk about modern apps

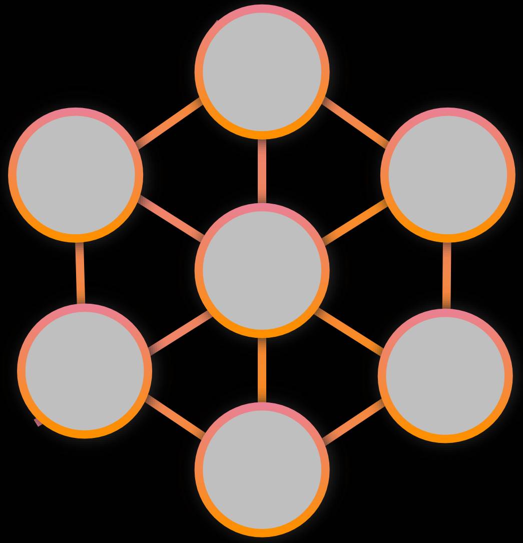
- Social media
- Online gaming
- Shared economy
- e-commerce
- Media streaming
- Ride hailing
- Dating

Users	1M+
Data volume	Terabytes–petabytes
Locality	Global
Performance	Microsecond latency
Request rate	Millions per second
Access	Mobile, IoT, devices
Scale	Virtually unlimited
Economics	Pay-as-you-go
Developer access	Instance API access
Development	Apps and storage are decoupled

We need to talk about modern apps

Monolith

Does everything

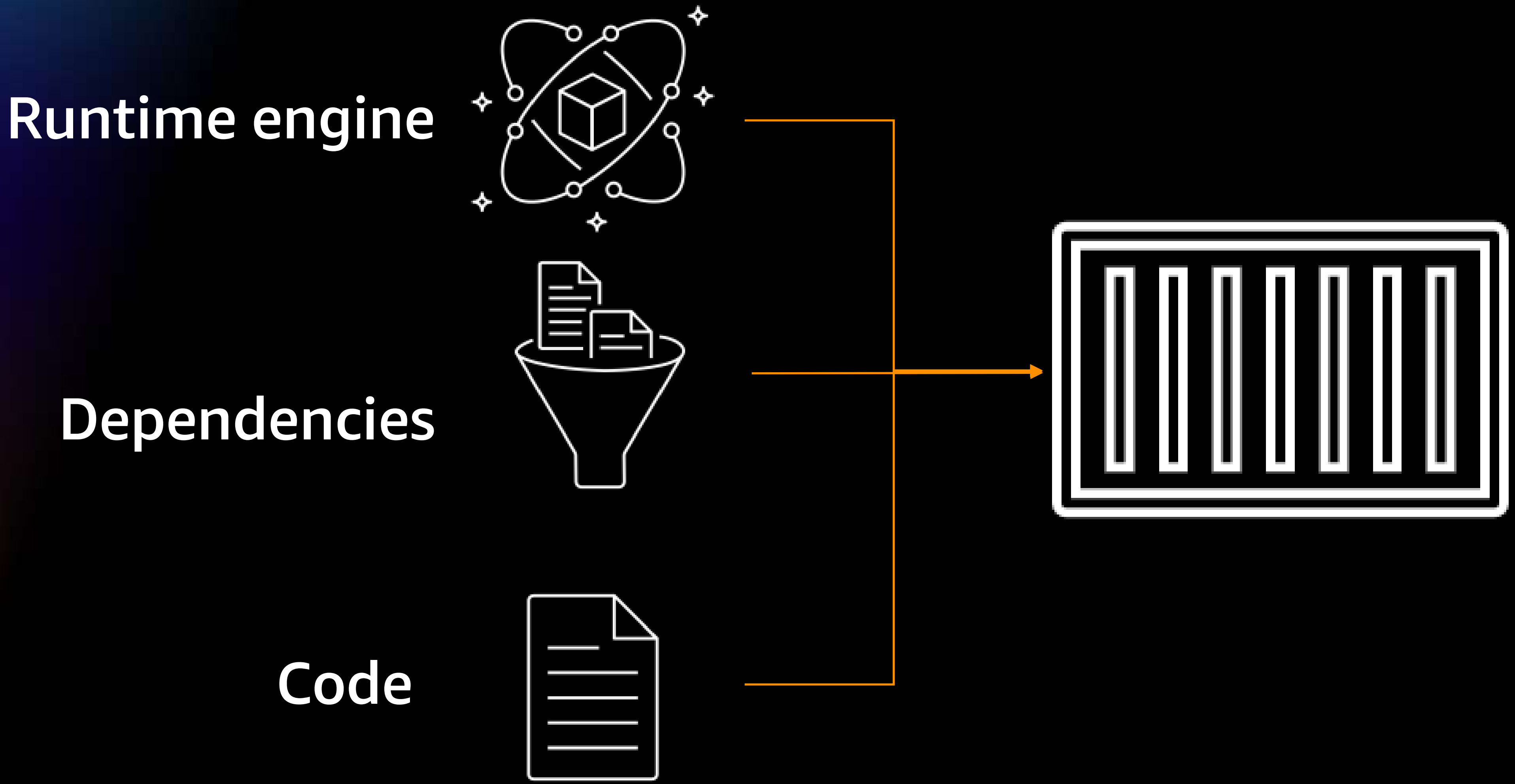


Microservices

Do one thing

Modern apps need “modern” solutions

Containers to the rescue!



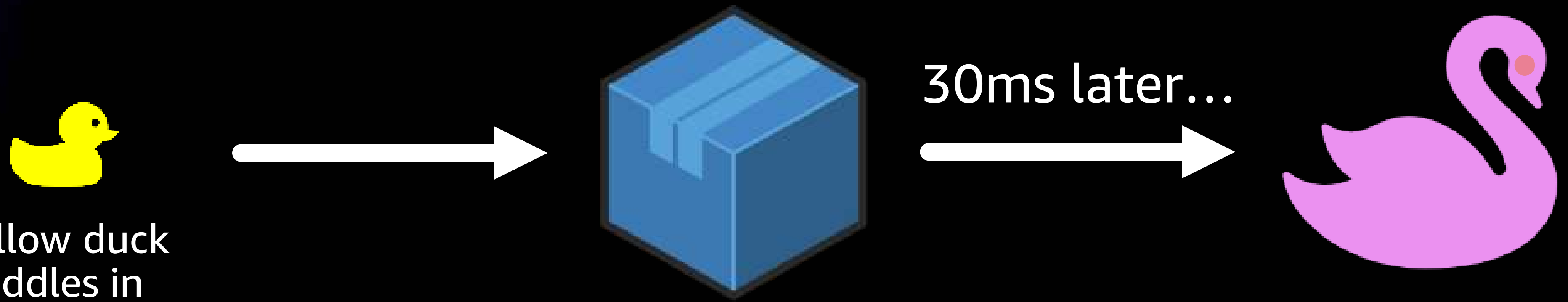
Runtime engine

Dependencies

Code

Observability in real life

Magic ~~box~~ container

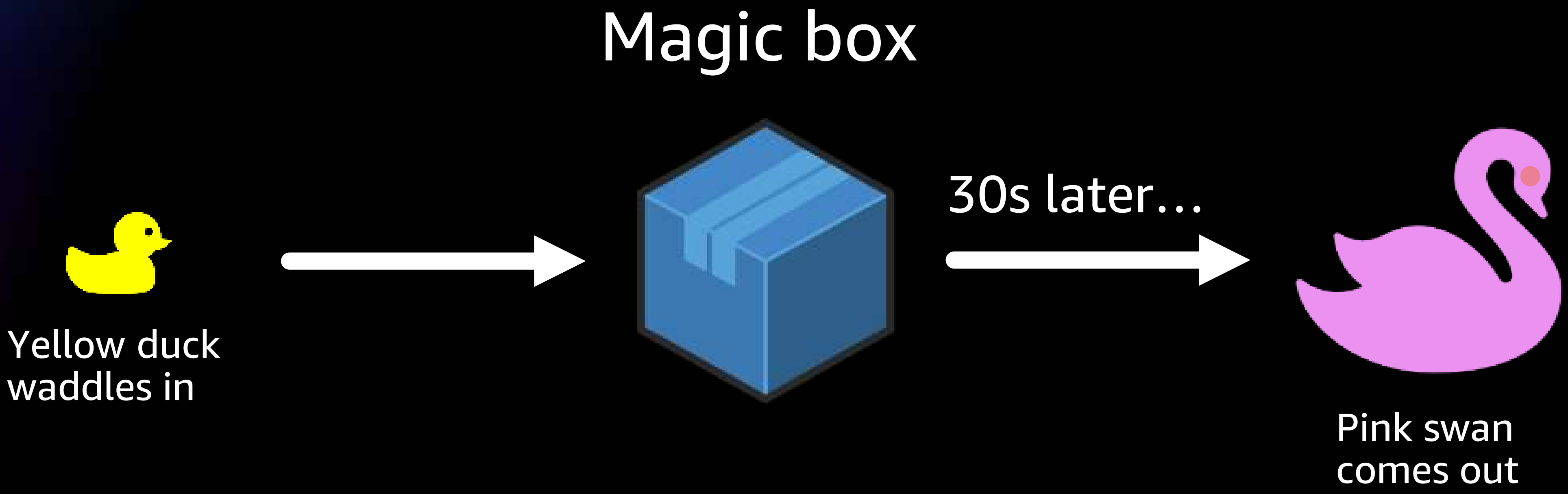


Yellow duck waddles in

30ms later…

Pink swan comes out

Observability in real life



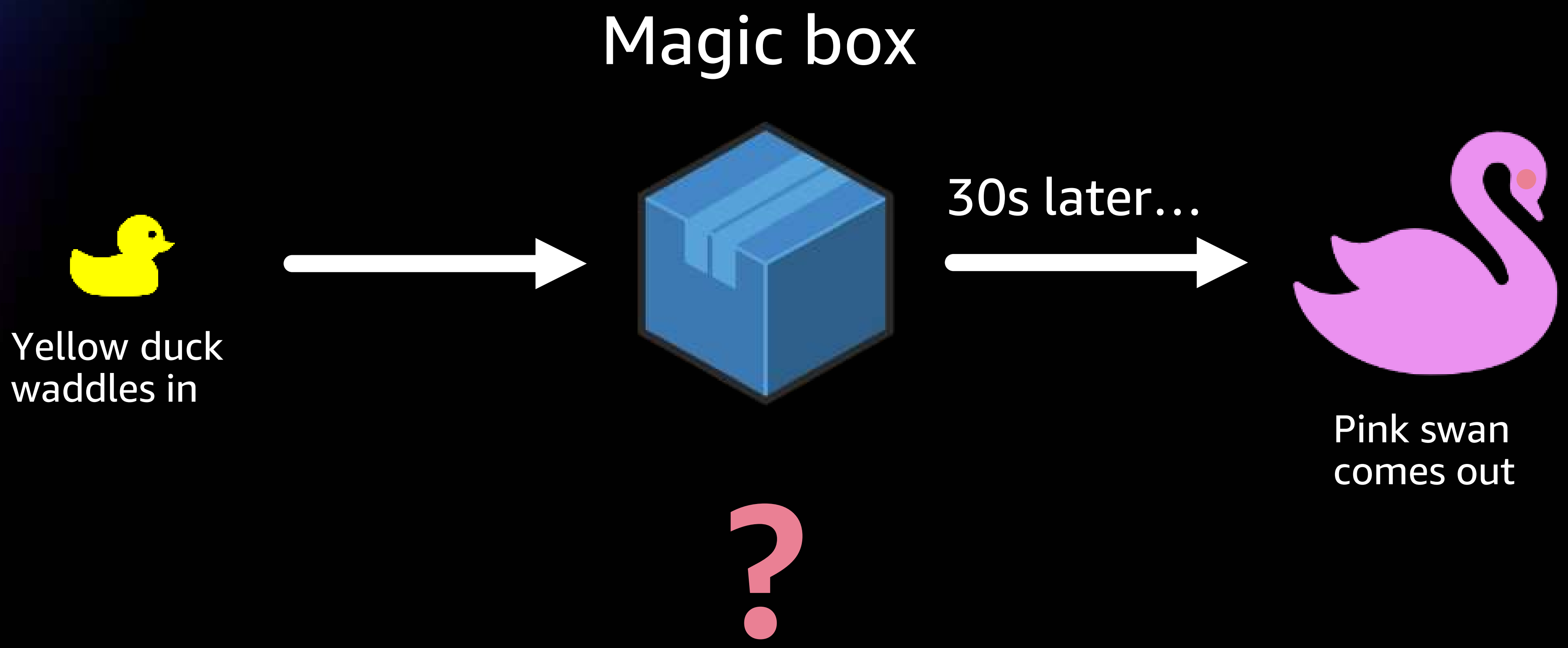
Magic box

Yellow duck
waddles in

30s later…

Pink swan
comes out

Observability in real life



Magic box

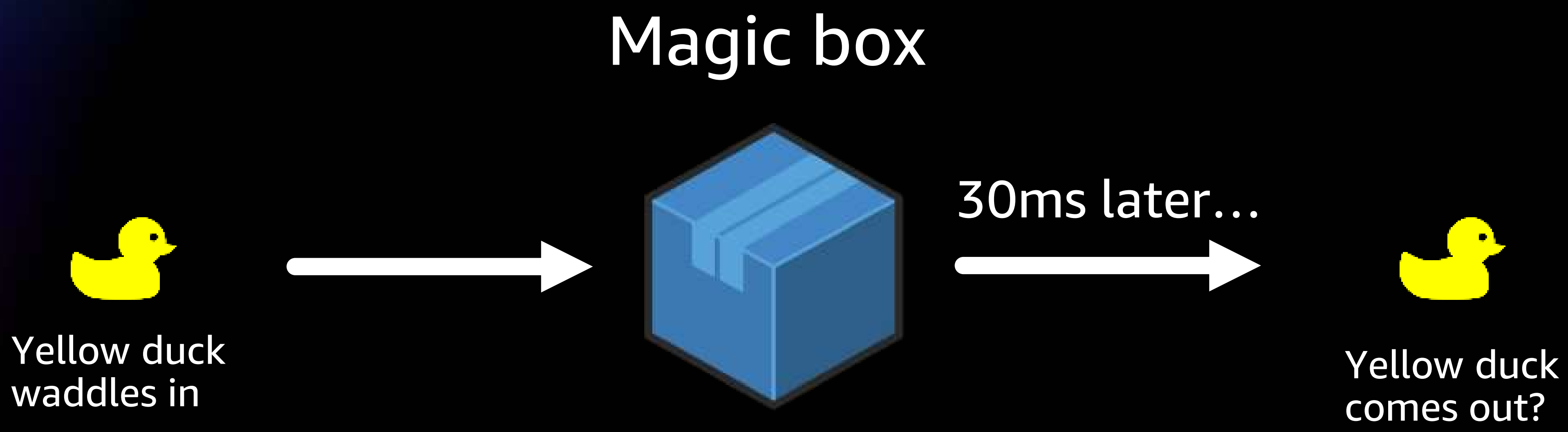
Yellow duck waddles in

30s later…

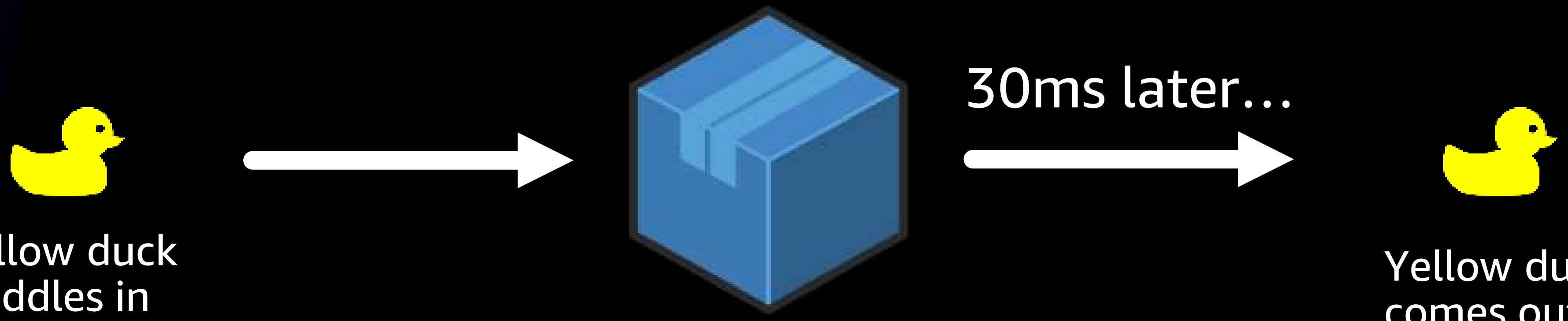
Pink swan comes out

?

Observability in real life



Observability in real life



Magic box

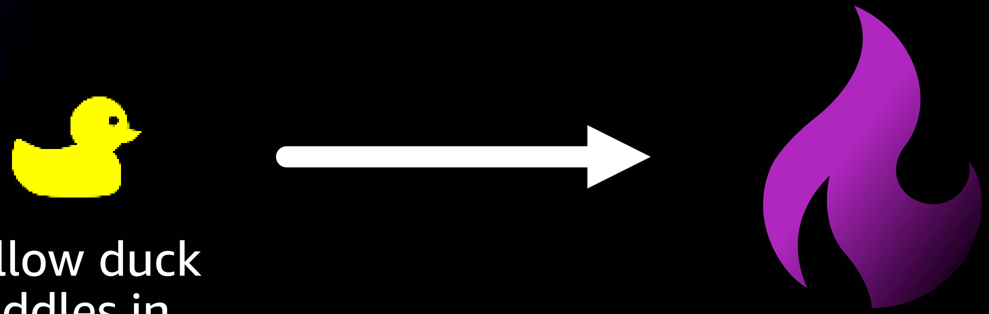
Yellow duck waddles in

30ms later…

Yellow duck comes out?

Oh no!

Observability in real life

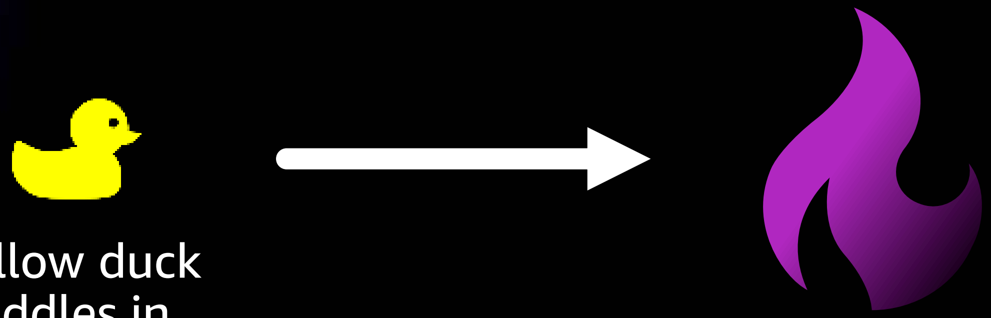


Magic box

Yellow duck
waddles in

Observability in real life

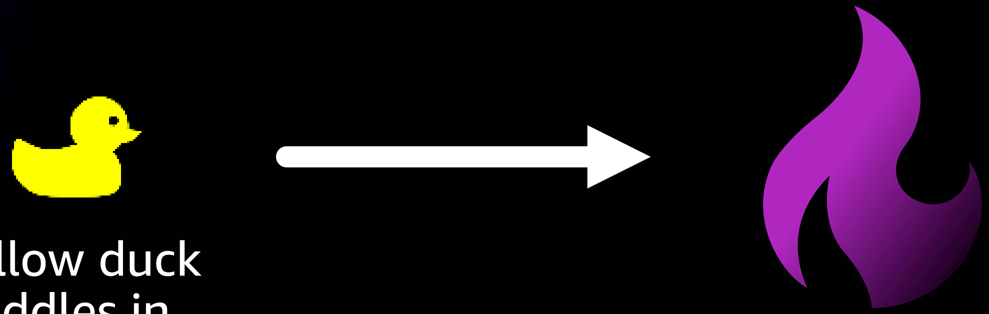
Magic box



Yellow duck waddles in

Box is on fire now

Observability in real life



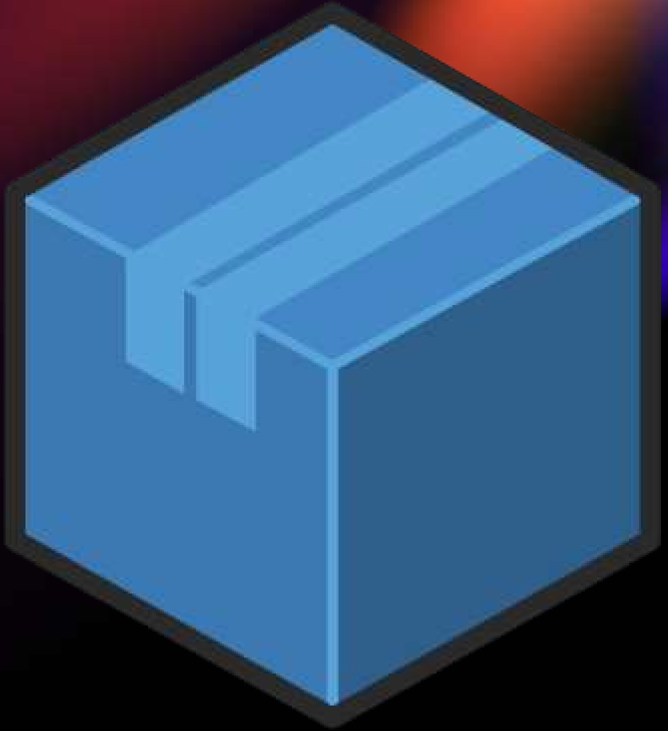
Magic box

Yellow duck waddles in

No duck, no swan, only smoke!

Why did purple smoke come out?

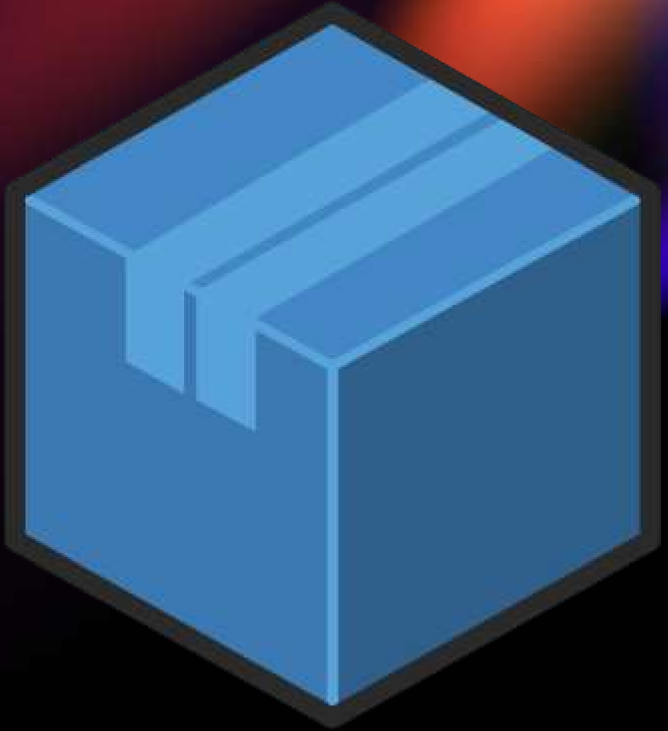
- There's no observability, so we don't know what goes on
- Why does it take 30ms normally?
- Why did it start taking 30s?
- What led to this behaviour?
- How do we stop purple smoke from coming out again?



Magic box?

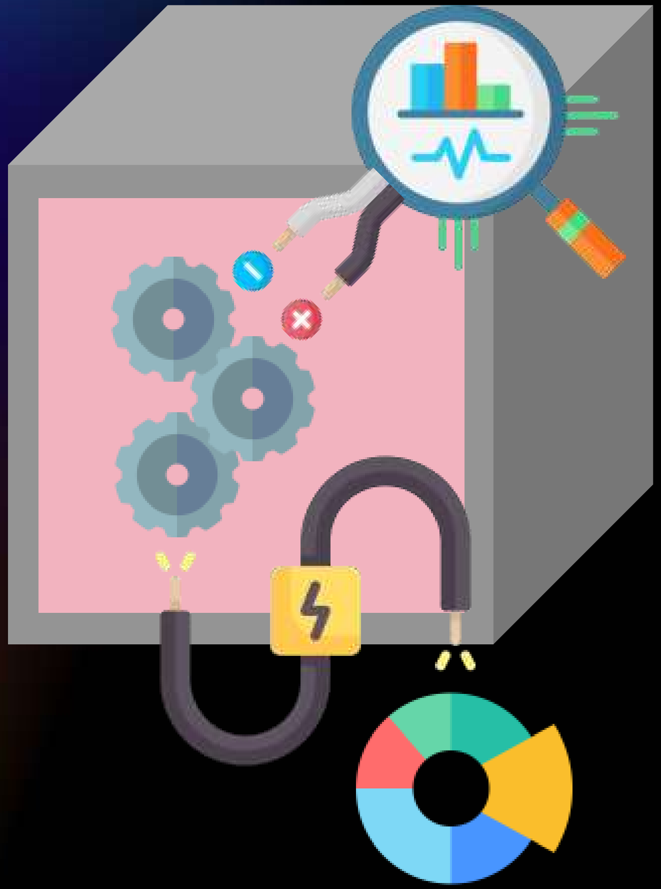
Why did purple smoke come out?

- There’s no observability, so we don’t know what goes on
- Why does it take 30ms normally?
- Why did it start taking 30s?
- What led to this behaviour?
- How do we stop purple smoke from coming out again?



Not really…

Observability needs to be preemptive

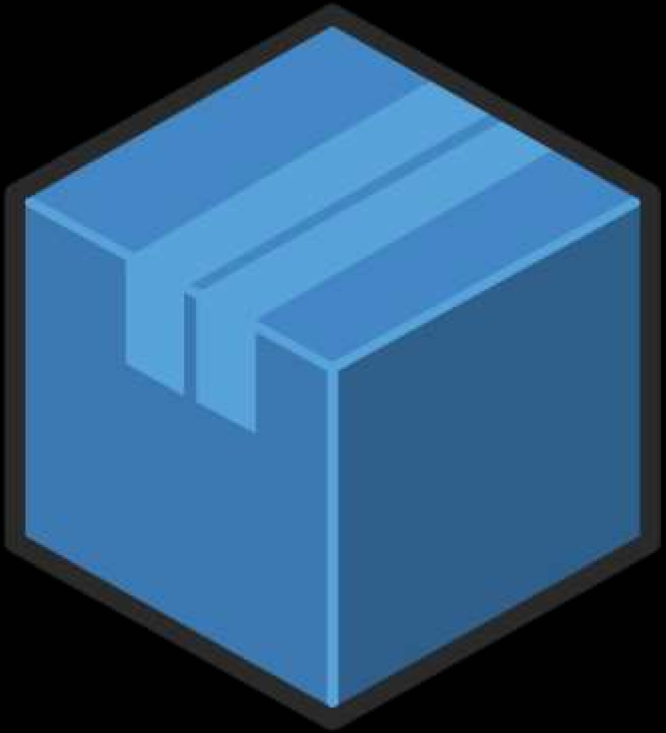


Good observability lets you to answer questions you didn’t even know you needed to ask.

When you ask questions ahead of time, you will already have the answers when those problems arise.

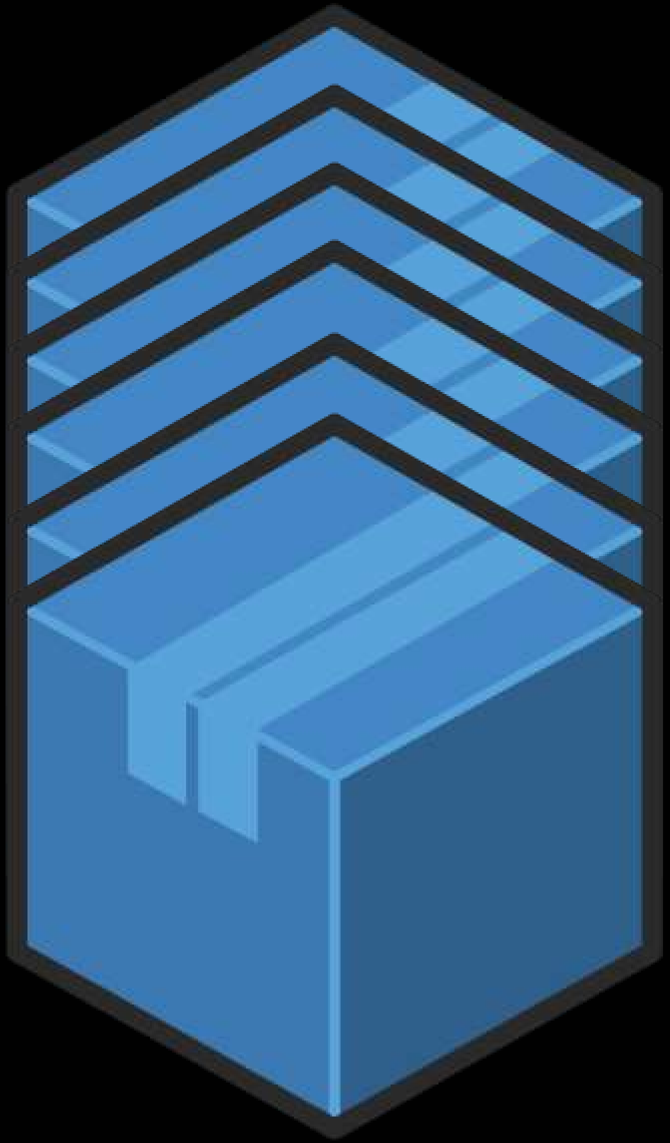
Modern apps – running at scale

- Modern apps primarily run with microservices
- And microservices run well with containers
- But we need many, many containers



Modern apps – running at scale

- Modern apps primarily run with microservices
- And microservices run well with containers
- But we need many, many containers



What is Amazon ECS?

Amazon Elastic Container Service (Amazon ECS)

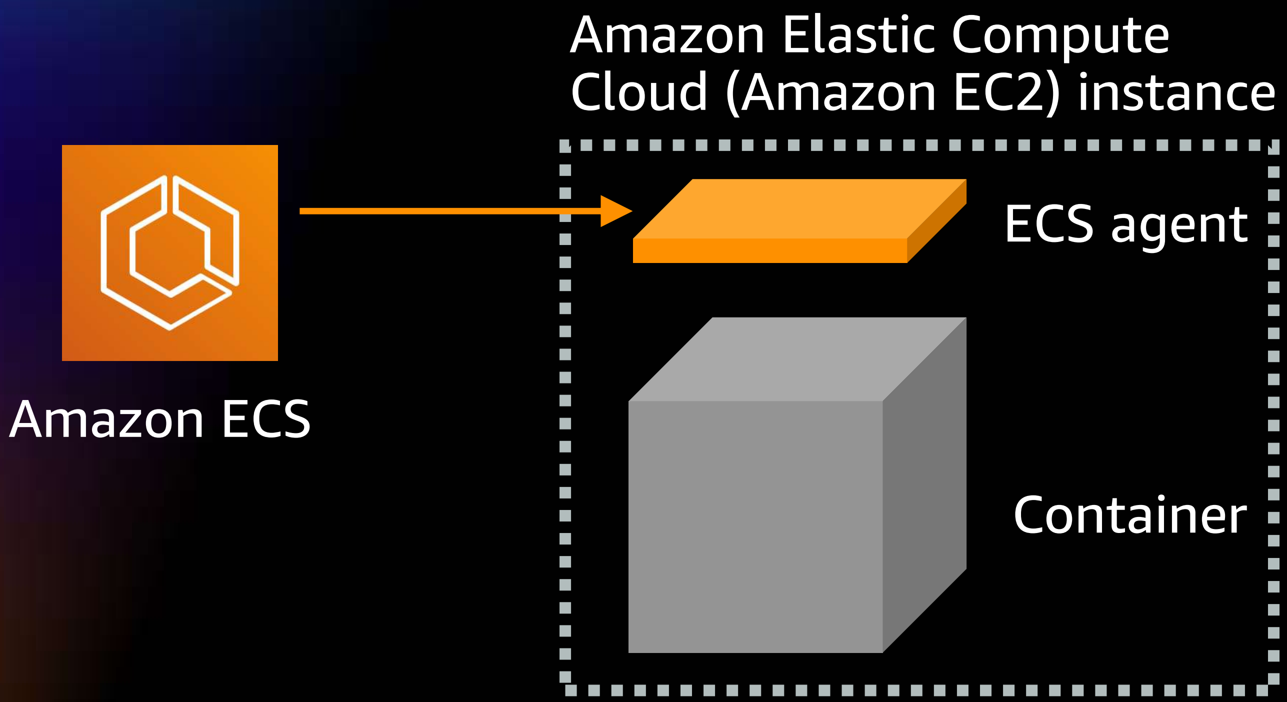
AWS-opinionated way to run containers at scale

Reduce decisions without sacrificing scale or features

Reduce time to build, deploy, and migrate applications

What is Amazon ECS?

Running containers on Amazon ECS



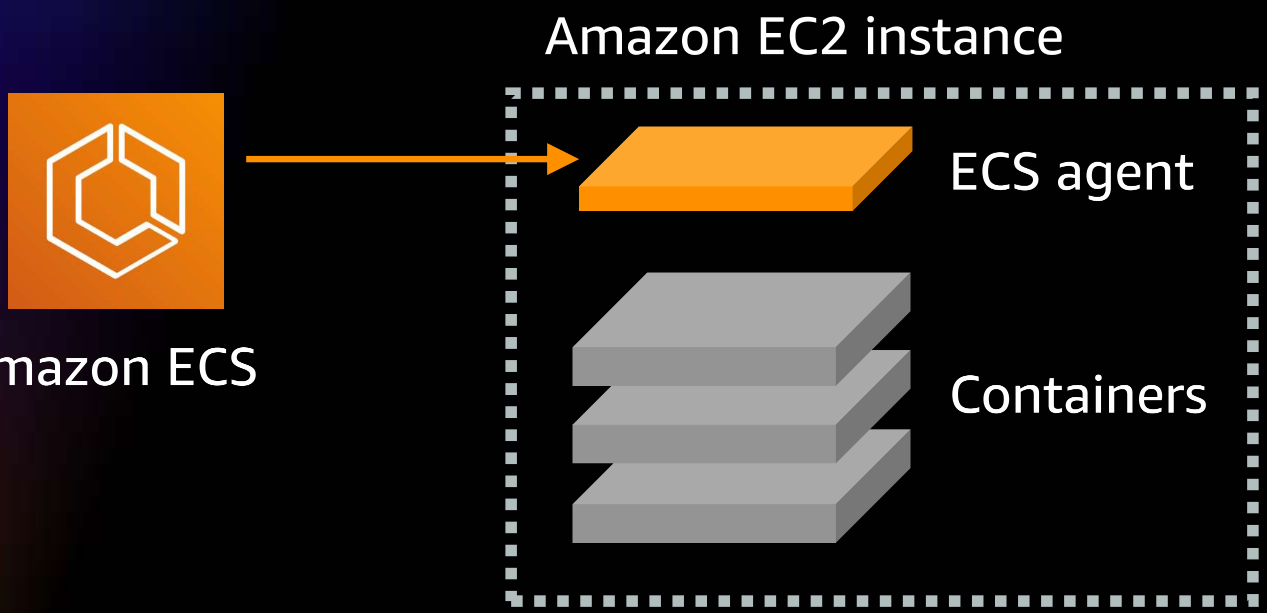
Give Amazon ECS your Docker Container

Amazon ECS runs as an agent on top of Amazon EC2 instances (or AWS Fargate)

Handles container lifecycle

What is Amazon ECS?

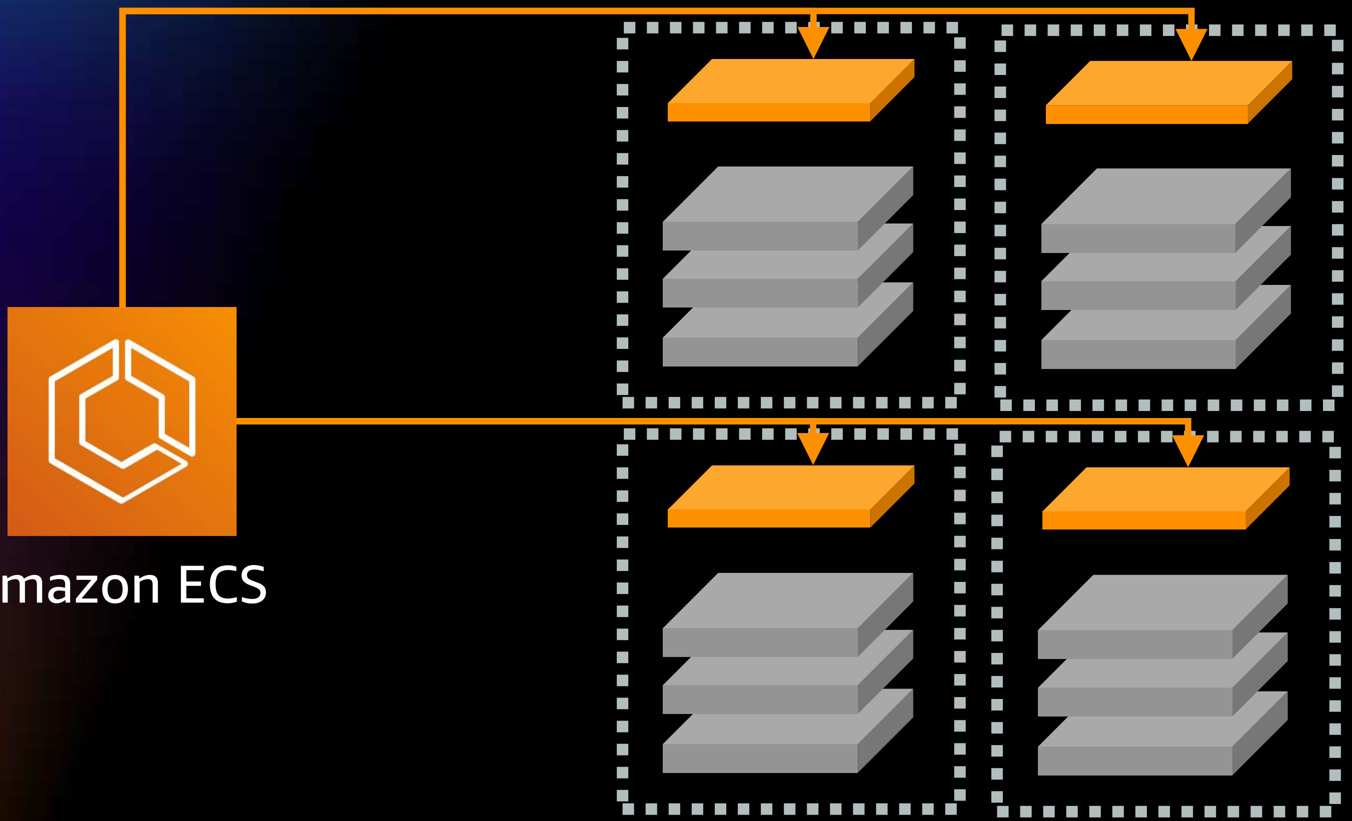
How about multiple containers?



Amazon ECS launches, controls and maintains multiple containers for each Amazon EC2 instance in the cluster

What is Amazon ECS?

But I've got so many instances and containers!

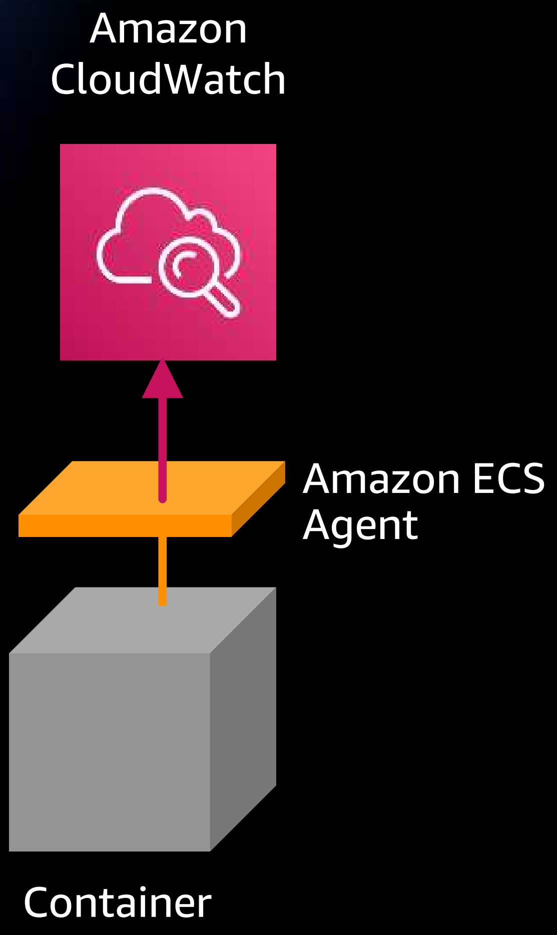


Amazon ECS scales with ease over many containers across many Amazon EC2 instances.

Even thousands of them!

How does Amazon ECS help me with observability?

Amazon ECS has out-of-the-box observability

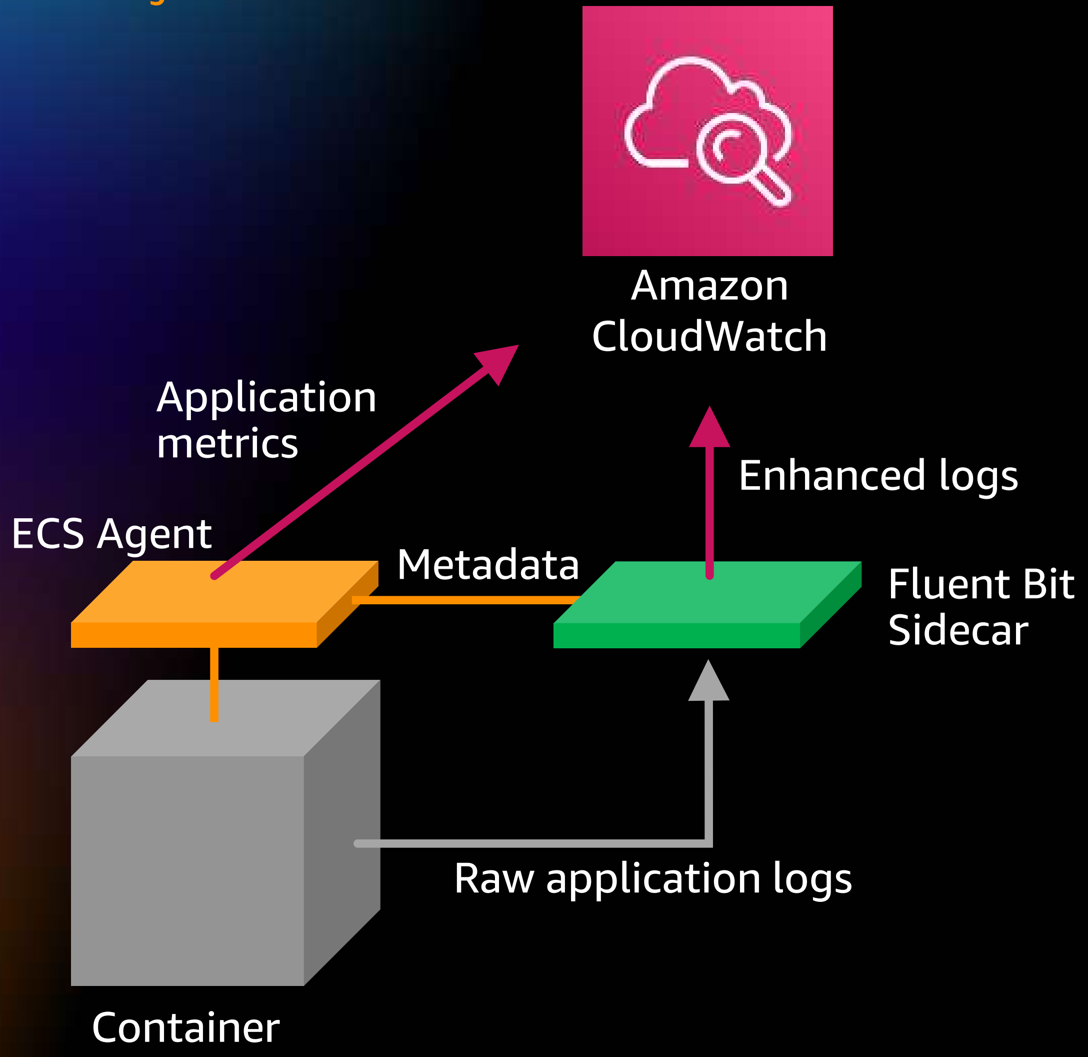


Amazon ECS’ container agent can get:

- CPU & memory usage
- standard output (stdout) & standard error (stderr)
- Health check (HTTP 200 etc)
- Container exit code

How does Amazon ECS help me with observability?

FireLens log router

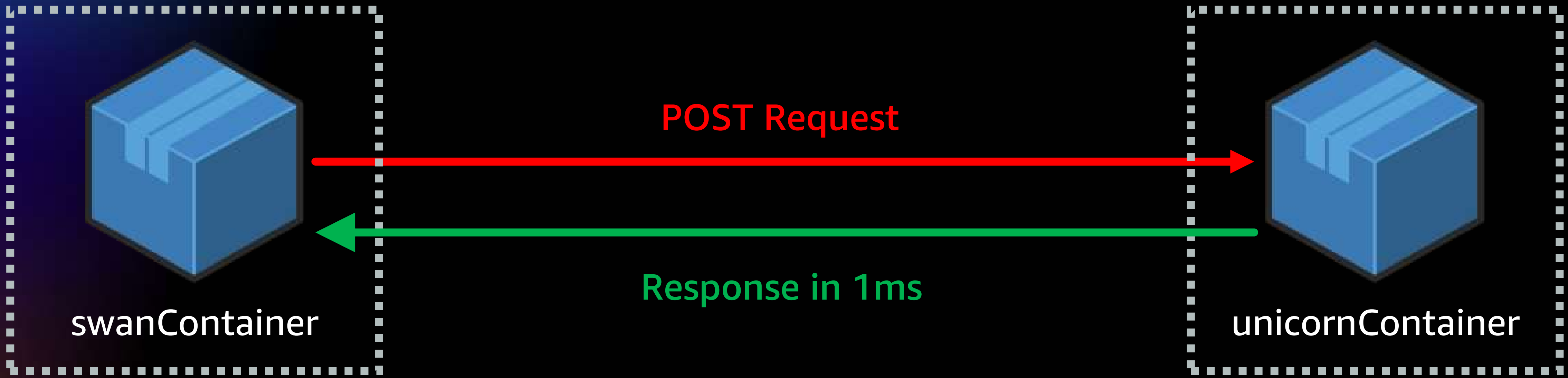


FireLens is a log router for Amazon ECS running as a *Fluent Bit* “sidecar” alongside the app to filter and route logs.

FireLens also enhances logs with additional information like Amazon EC2 metadata, task ID, app version and more!

Service-to-service communications

Typical service communication



POST Request

Response in 1ms

swanContainer

unicornContainer

Amazon EC2

Amazon EC2

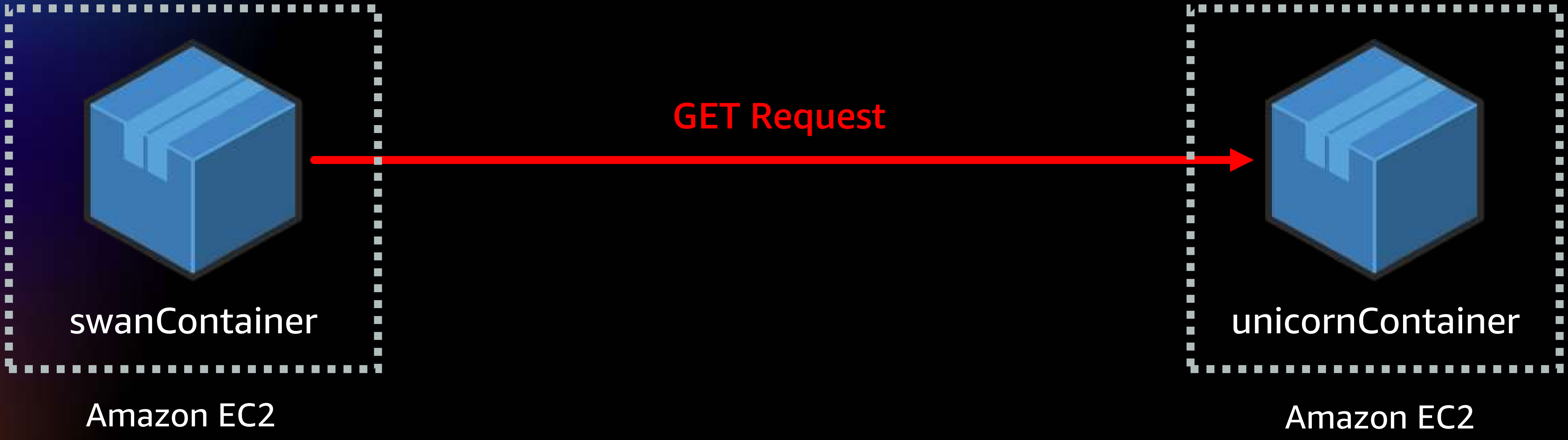
Access logs:

POST /unicorn – 8ms

Access logs:

POST /unicorn – 9ms

But logs only tell us so much



swanContainer

Amazon EC2

GET Request

unicornContainer

Amazon EC2

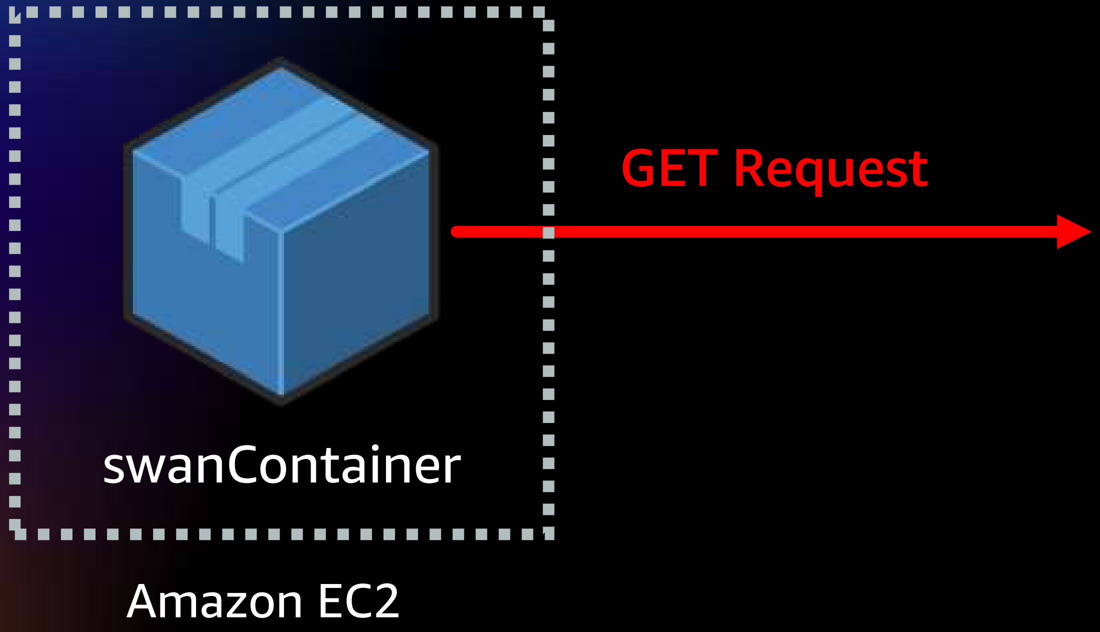
Access logs:

GET /unicorn –
timed out

Access logs:

No logs

What is going on?



Access logs:

GET /unicorn – timed out

- Networking issues?
- Instance/Host issues?
- Container issues?

A service mesh can help

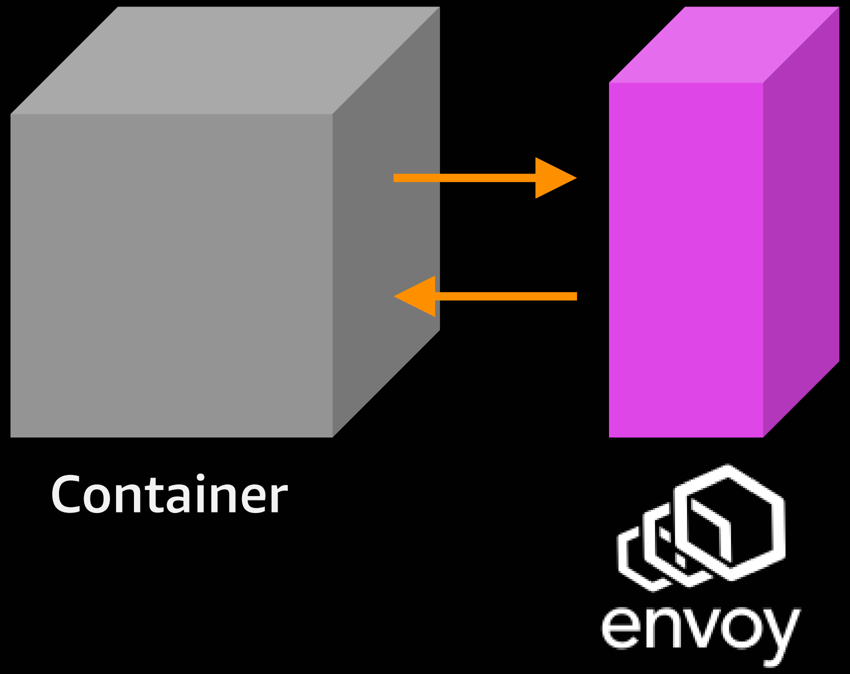
A service mesh provides a means of monitoring all interservice traffic and abstracting its configuration

The mesh is aware of all the data on the wire, and we can leverage that to solve for many needs

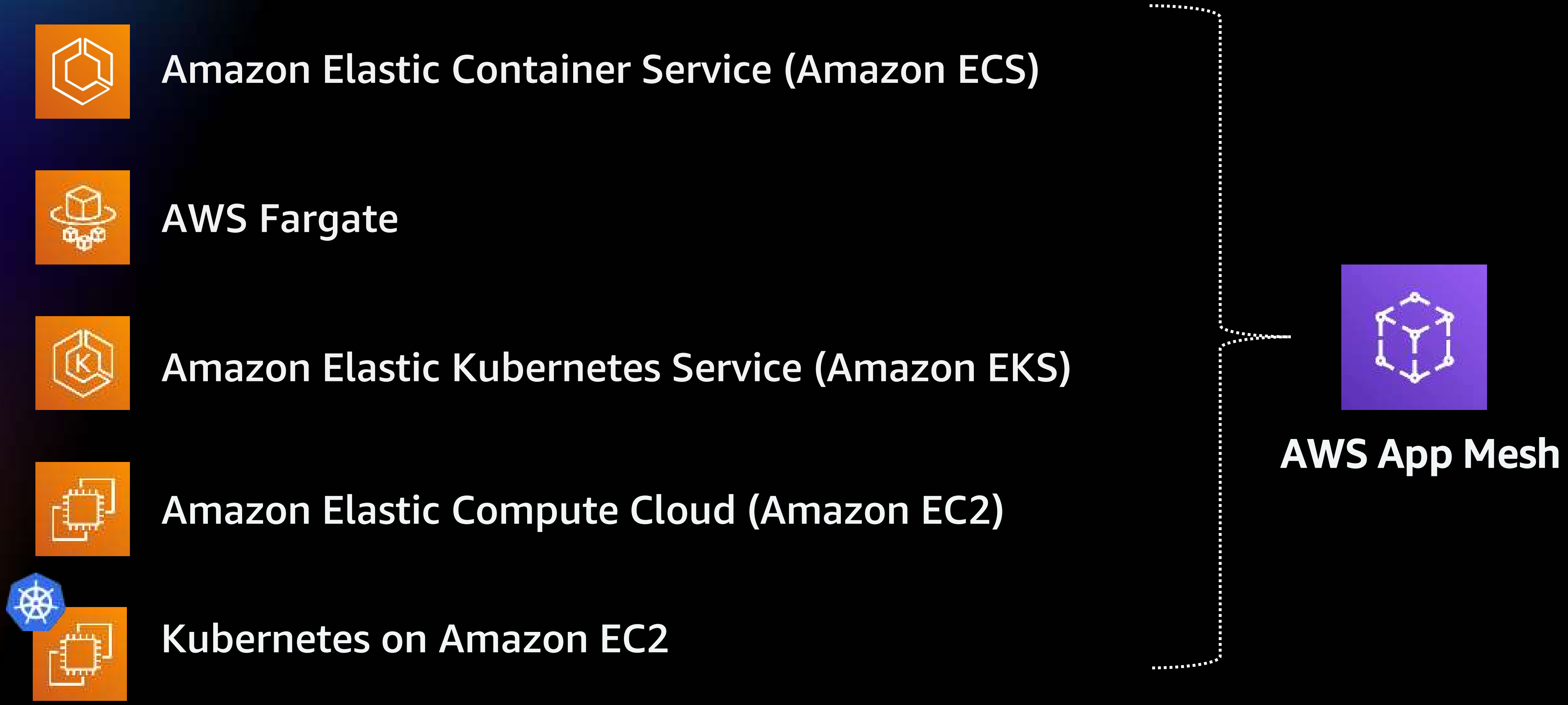
AWS App Mesh can help

AWS App Mesh is a fully managed service mesh

- Intelligent, application-aware traffic management
- Security & policy management and enforcement
- Observability support for numerous solutions (Prometheus/Grafana…)
- Powered by Envoy proxy
- All inbound and outbound traffic is proxied through Envoy
- Envoy routes and observes all connections to and from your application



AWS App Mesh, not just on Amazon ECS



Amazon Elastic Container Service (Amazon ECS)

AWS Fargate

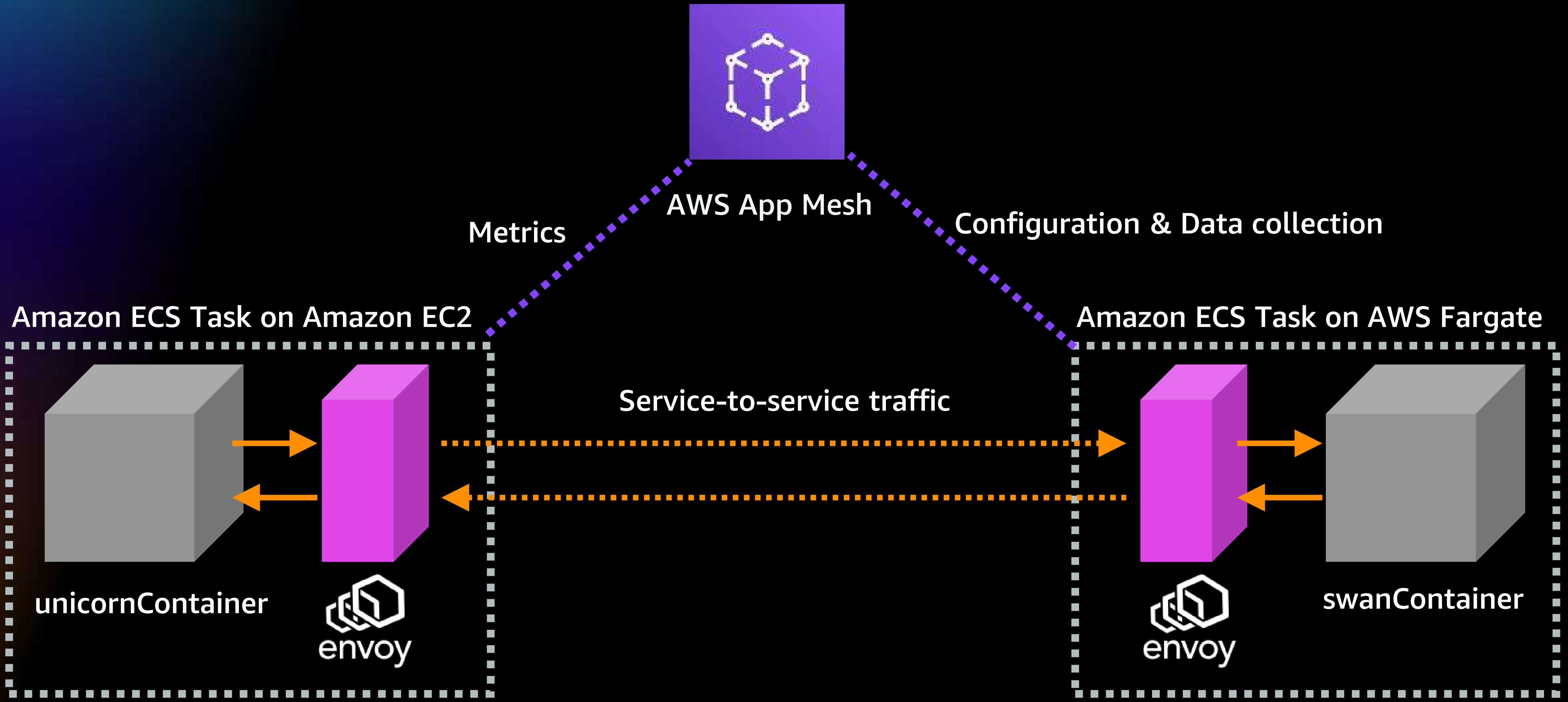
Amazon Elastic Kubernetes Service (Amazon EKS)

Amazon Elastic Compute Cloud (Amazon EC2)

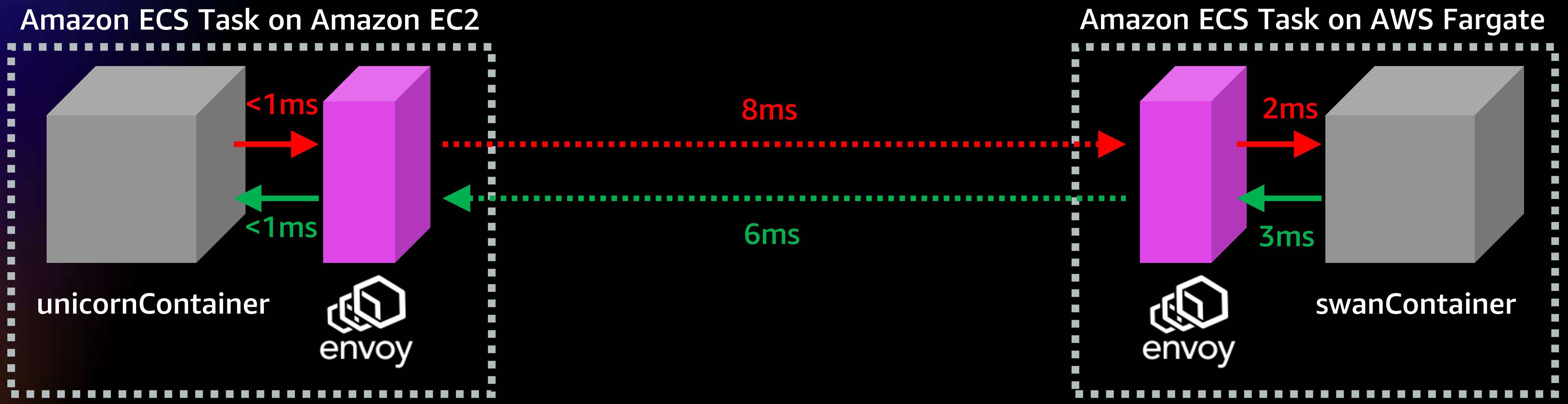
Kubernetes on Amazon EC2

AWS App Mesh

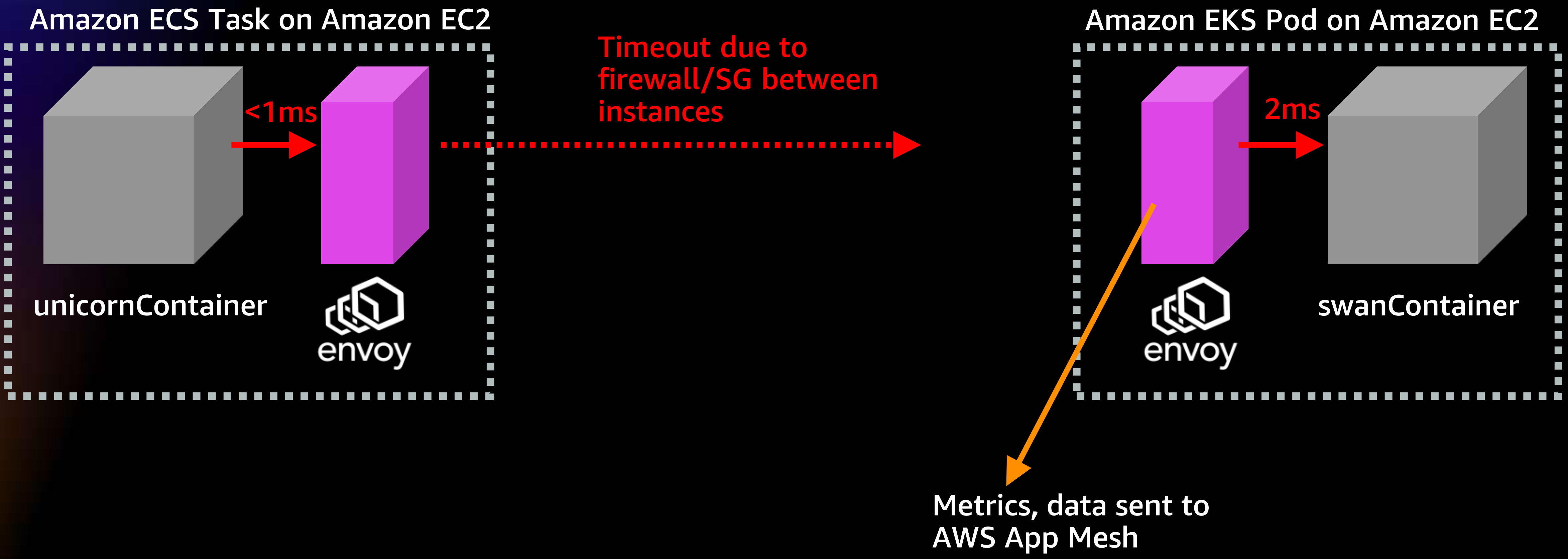
AWS App Mesh 30k feet view


AWS App Mesh
Metrics
Configuration & Data collection
Amazon ECS Task on Amazon EC2
Amazon ECS Task on AWS Fargate
Service-to-service traffic
unicornContainer
envoy
envoy
swanContainer

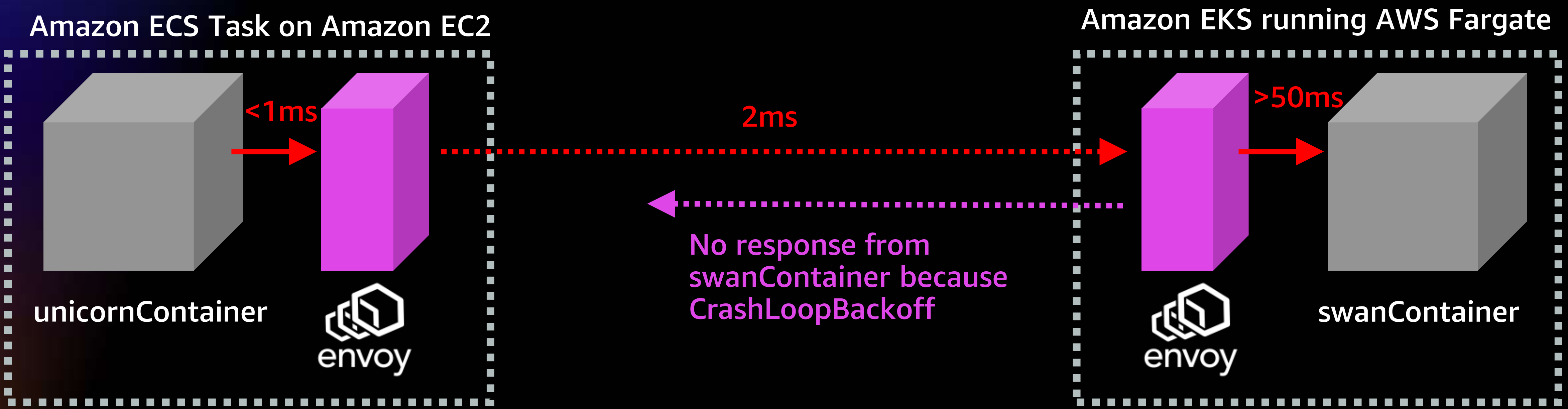
Observe point-to-point traffic



And know exactly where things break down


Amazon ECS Task on Amazon EC2
<1ms
unicornContainer
envoy
Timeout due to firewall/SG between instances
Amazon EKS Pod on Amazon EC2
2ms
envoy
swanContainer
Metrics, data sent to AWS App Mesh

And know exactly where things break down



Amazon ECS Task on Amazon EC2

Amazon EKS running AWS Fargate

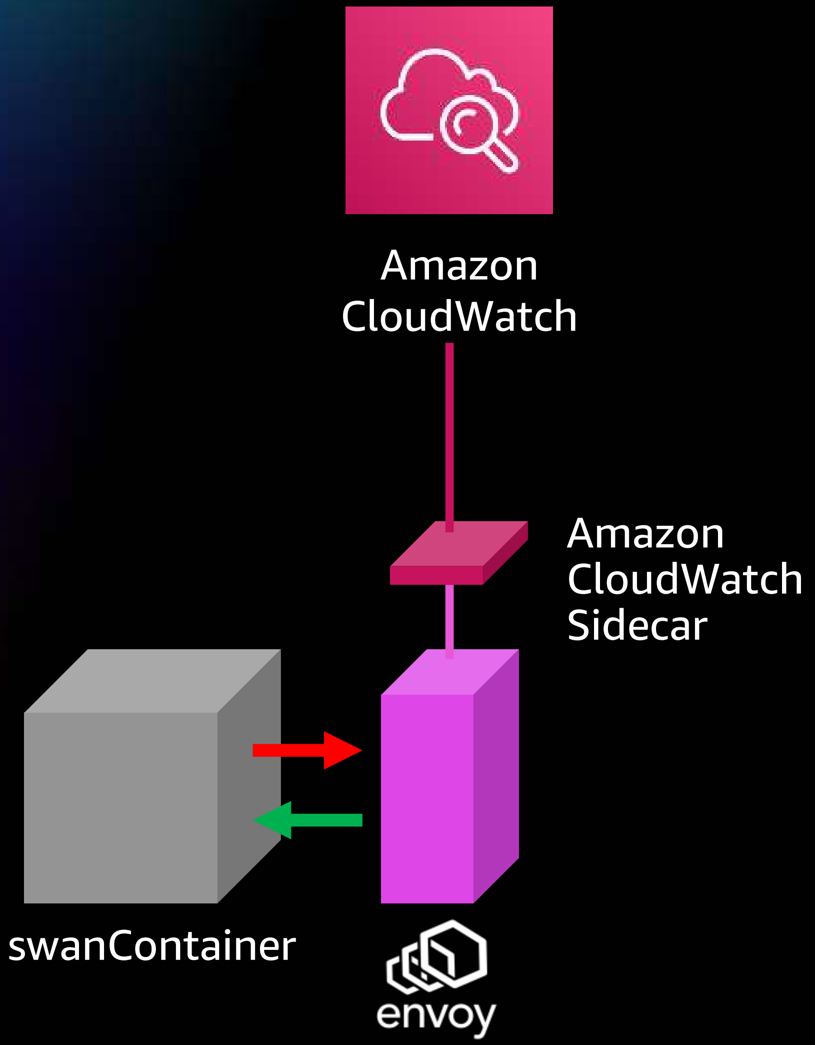
No response from swanContainer because CrashLoopBackoff

AWS App Mesh can also

- Shape traffic between services
- Help with deployment strategies
- Enforce rate limiting and circuit breakers
- Parameter-based routing
- And more!

AWS App Mesh

Envoy logs to Amazon CloudWatch



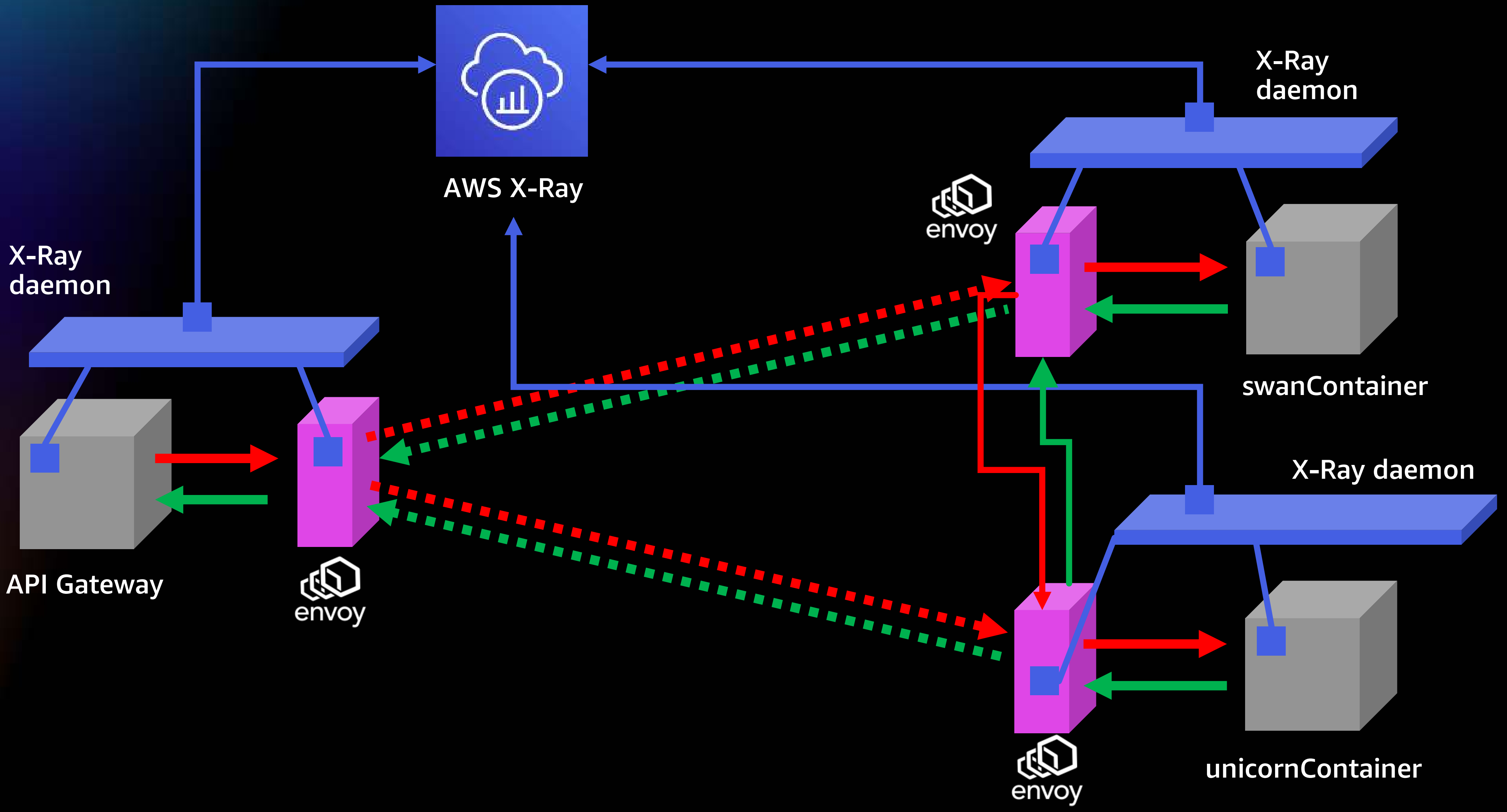
- Envoy proxies collect over 3000 unique metrics per proxy!
- These are in standard ***StatsD*** format
- Along with Envoy, Amazon CloudWatch also has an agent sidecar running along the app that consumes these logs, and stores it on Amazon CloudWatch

Deep tracing with AWS X-Ray

- Analyze and debug production, distributed applications
- Identify performance bottlenecks at every hop
- Troubleshoot root cause with code level tracing
- Trace user requests
- Popular language/SDK support
- Easy integration with your current app

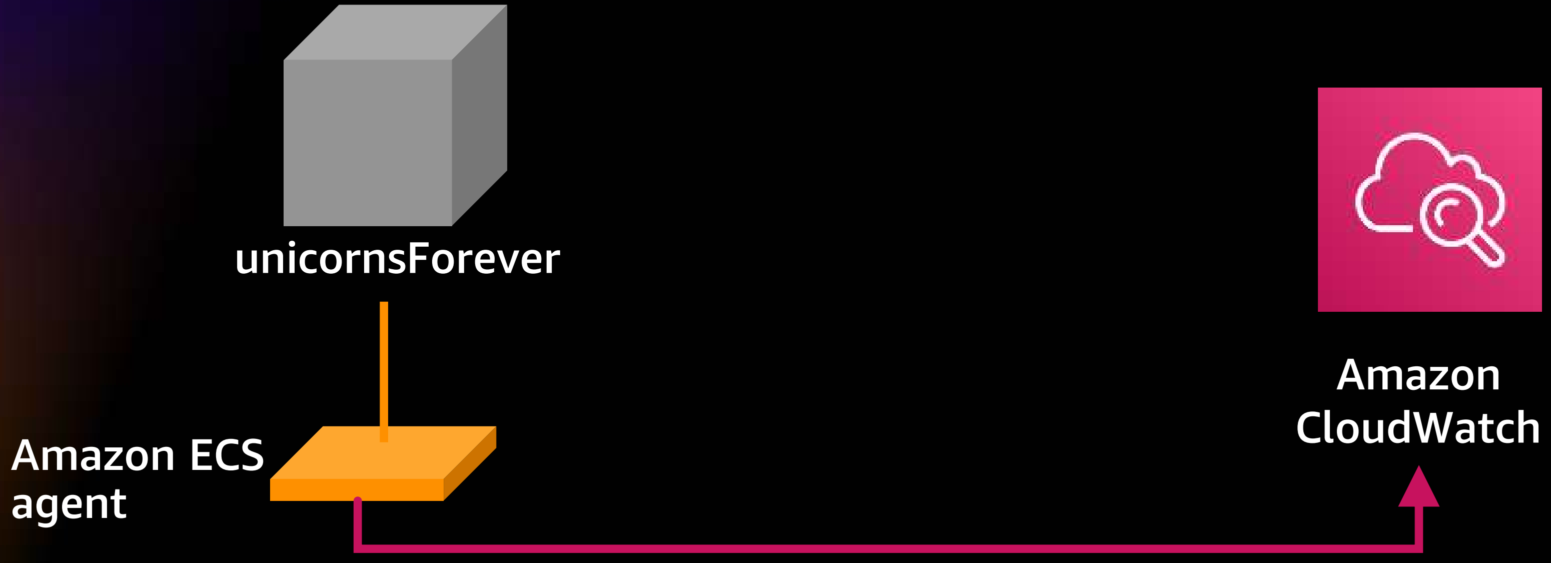
AWS X-Ray

Deep tracing with AWS X-Ray



Observability everywhere

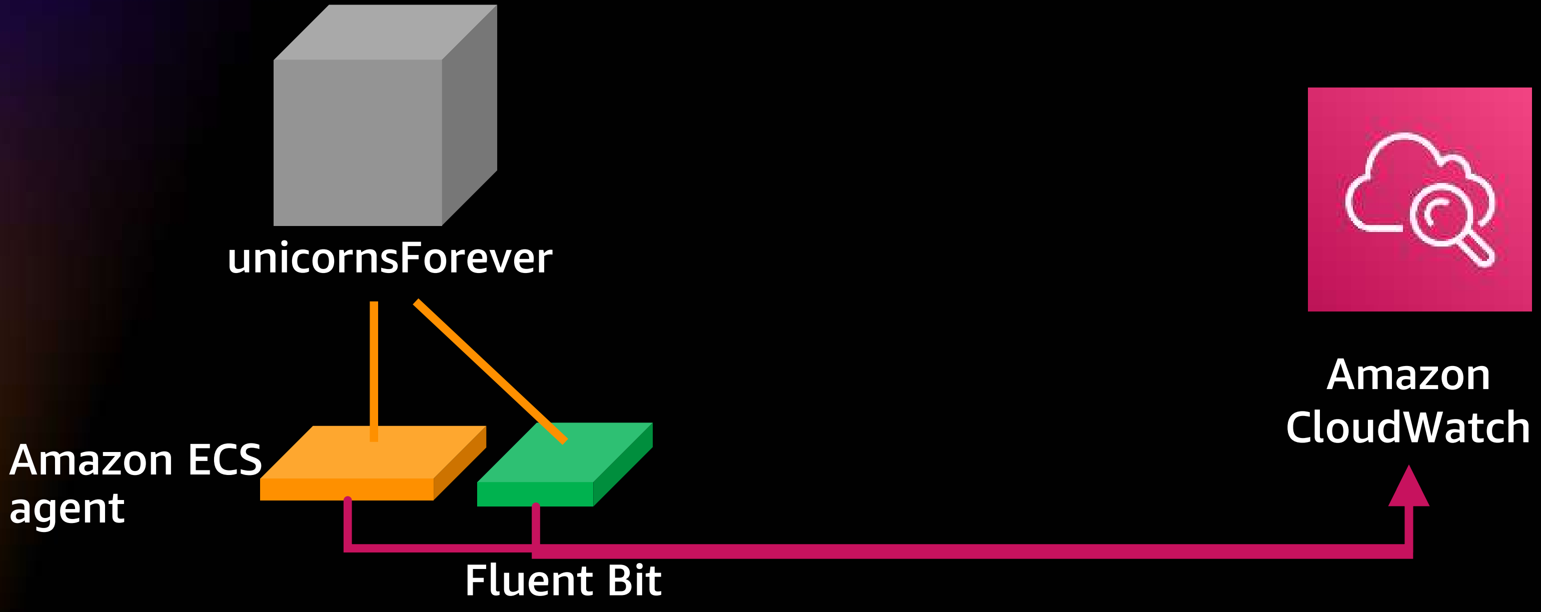
End-to-end observability



Logs and metrics

Alerts, Dashboards, Insights

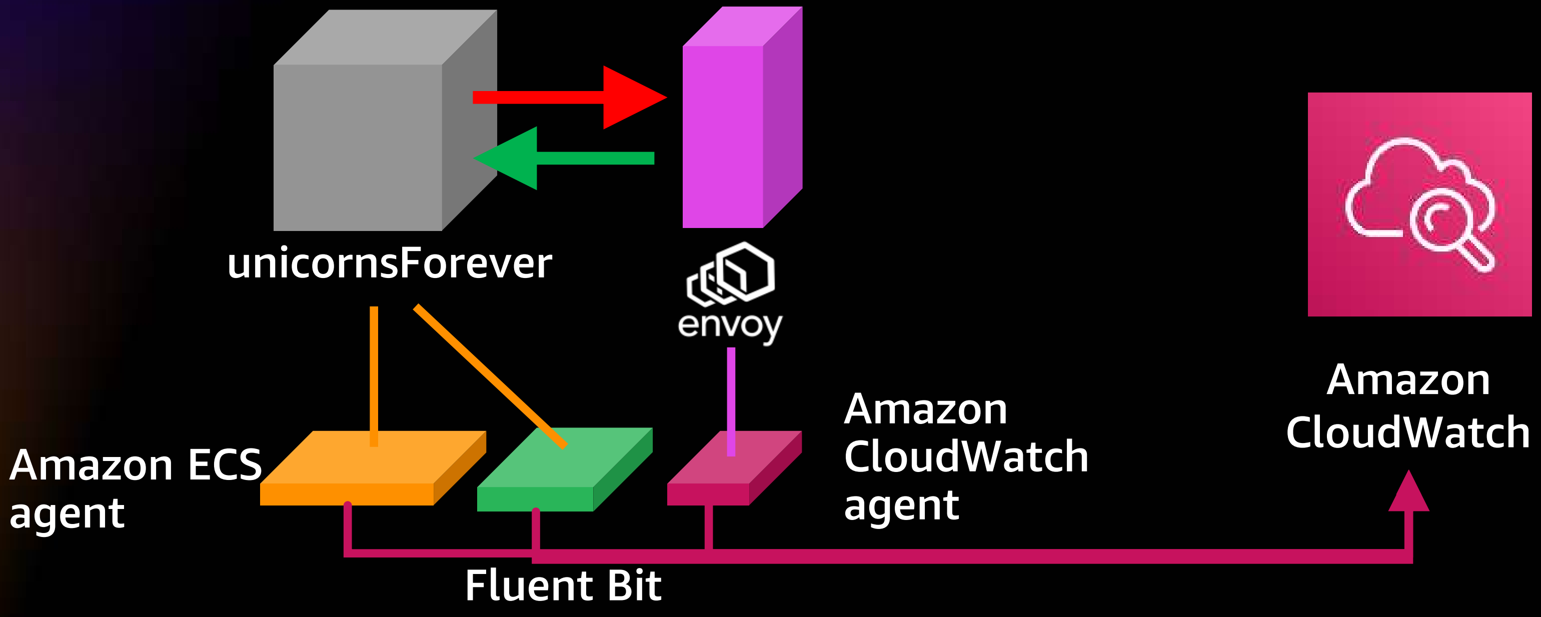
End-to-end observability



Logs and metrics

Filter logs, get more insights, and finer grained information on the application

End-to-end observability



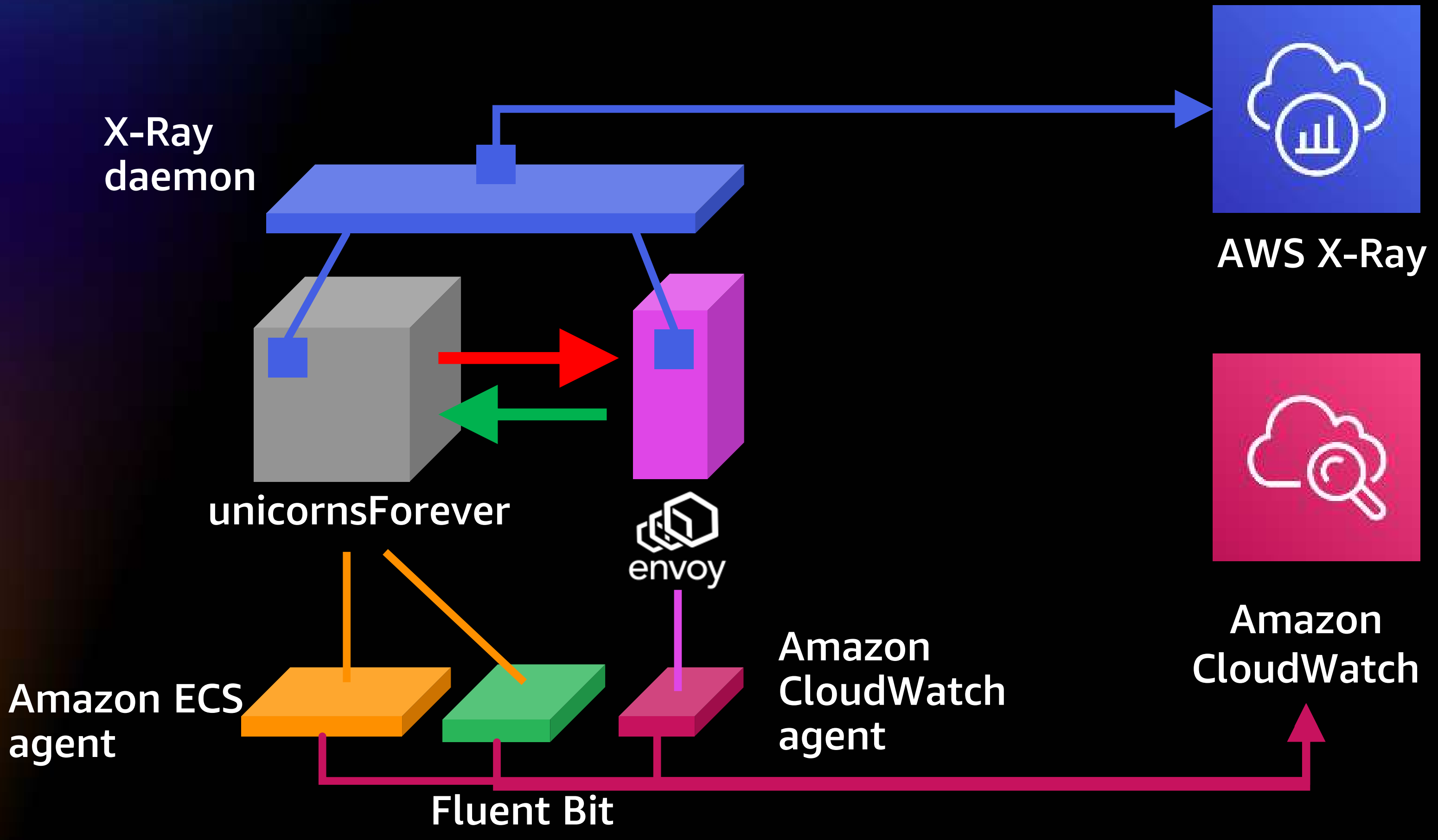
Logs and metrics + Envoy

Dashboards, monitoring, alerts & more

Insights

ServiceLens

End-to-end observability



Trace network requests in every service/node/container

Dive deep into errors, tracing from origin to destination

Service Map

Logs and metrics + Envoy

Dashboards, monitoring, alerts & more

Insights

ServiceLens

Demo time!

Recap

- Modern apps need modern solutions with microservices
- We can run microservices on containers with Amazon ECS
- Basic observability with Amazon ECS, Fluent Bit and Amazon CloudWatch
- Traffic shaping and monitoring with AWS App Mesh
- Tracing of network requests with AWS X-Ray
- Extensive investigative ability with AWS observability tools

Other resources

- Try the Observability workshop at: https://observability.workshop.aws/
- Try the AWS App Mesh workshop at: https://www.appmeshworkshop.com/
- AWS App Mesh Samples - https://github.com/aws/aws-app-mesh-examples/
- AWS Controllers for Kubernetes - https://github.com/aws-controllers-k8s/community/projects/1
- Envoy Proxy 101 - https://www.getambassador.io/learn/envoy-proxy/

Visit the Modern Applications Resource Hub for more resources

Dive deeper with these resources to help you develop an effective plan for your modernization journey.

- Build modern applications on AWS e-book
- Build mobile and web apps faster e-book
- Modernize today with containers on AWS e-book
- Adopting a modern Dev+Ops model e-book
- Modern apps need modern ops e-book
- Determining the total cost of ownership: Comparing Serverless and Server-based technologies paper
- Continuous learning, continuous modernization e-book
- … and more!

https://bit.ly/3yfOvbK

Visit resource hub »

AWS Training and Certification

Accelerate modernization with continuous learning

Free digital courses, including:
Architecting serverless solutions
Getting started with DevOps on AWS

Earn an industry-recognized credential:
AWS Certified Developer – Associate
AWS Certified DevOps – Professional

Hands-on classroom training (available virtually) including:
Running containers on Amazon Elastic Kubernetes Service (Amazon EKS)
Advanced developing on AWS

Create a self-paced learning roadmap
AWS ramp-up guide - Developer
AWS ramp-up guide - DevOps

Take Developer and DevOps training today

Learn more about Modernization training for you and your team

Thank you for attending AWS Innovate Modern Applications Edition

We hope you found it interesting! A kind reminder to **complete the survey.**
Let us know what you thought of today's event and how we can improve the event experience for you in the future.

- aws-apj-marketing@amazon.com
- twitter.com/AWSCloud
- facebook.com/AmazonWebServices
- youtube.com/user/AmazonWebServices
- slideshare.net/AmazonWebServices
- twitch.tv/aws

Thank you!

Thirumalai Aiyalu (Thiru)

taiyalu@amazon.com

@taiyalu85