SUMMIT

INDIA | MAY 25, 2023

GSAWS007

Building next-gen applications using resilient serverless architectures

Santhakumar Rajangam
Enterprise Solutions Architect
AWS India

Varun Kumar
Senior Startup Solutions Architect
AWS India

What the Suez Canal obstruction event tells us about resilience?

- Single point of failure
- Understanding the dependencies
- Lack of anticipation

Expectations for today's talk

Building a resilient application is not just about the design patterns – it's about your entire application pipeline and the way you build in resilience at every stage

We are going to offer you guidance that will help you shape the way you approach architecting on AWS

We don't believe in only providing a few specific design patterns

Resiliency

The ability of your workload to withstand partial and intermittent failures across components

Reliability

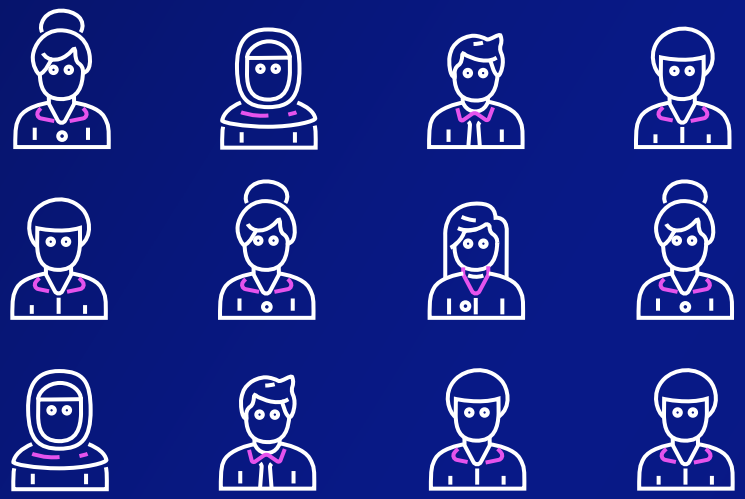
A reliable workload performs its intended function correctly and consistently

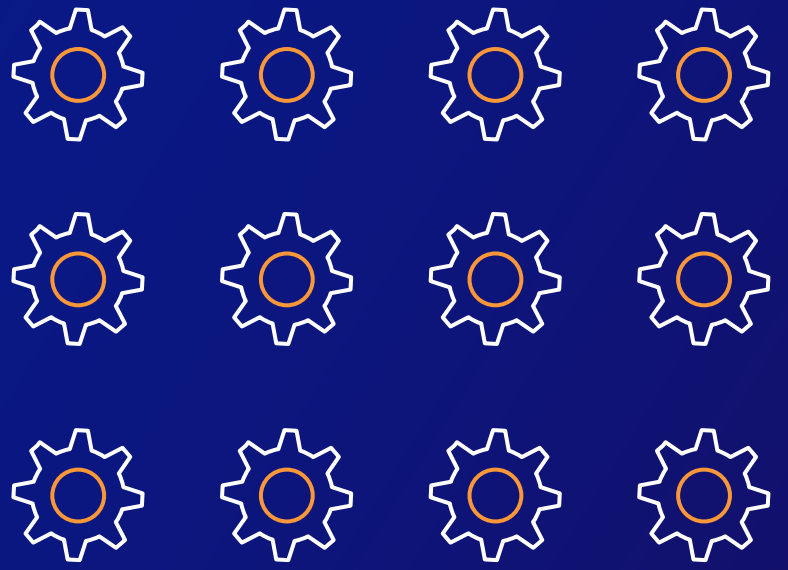
Operational Excellence

The ability to run your workloads effectively, gain insight into your operations, and continuously improve processes and procedures

Serverless adoption is growing fast

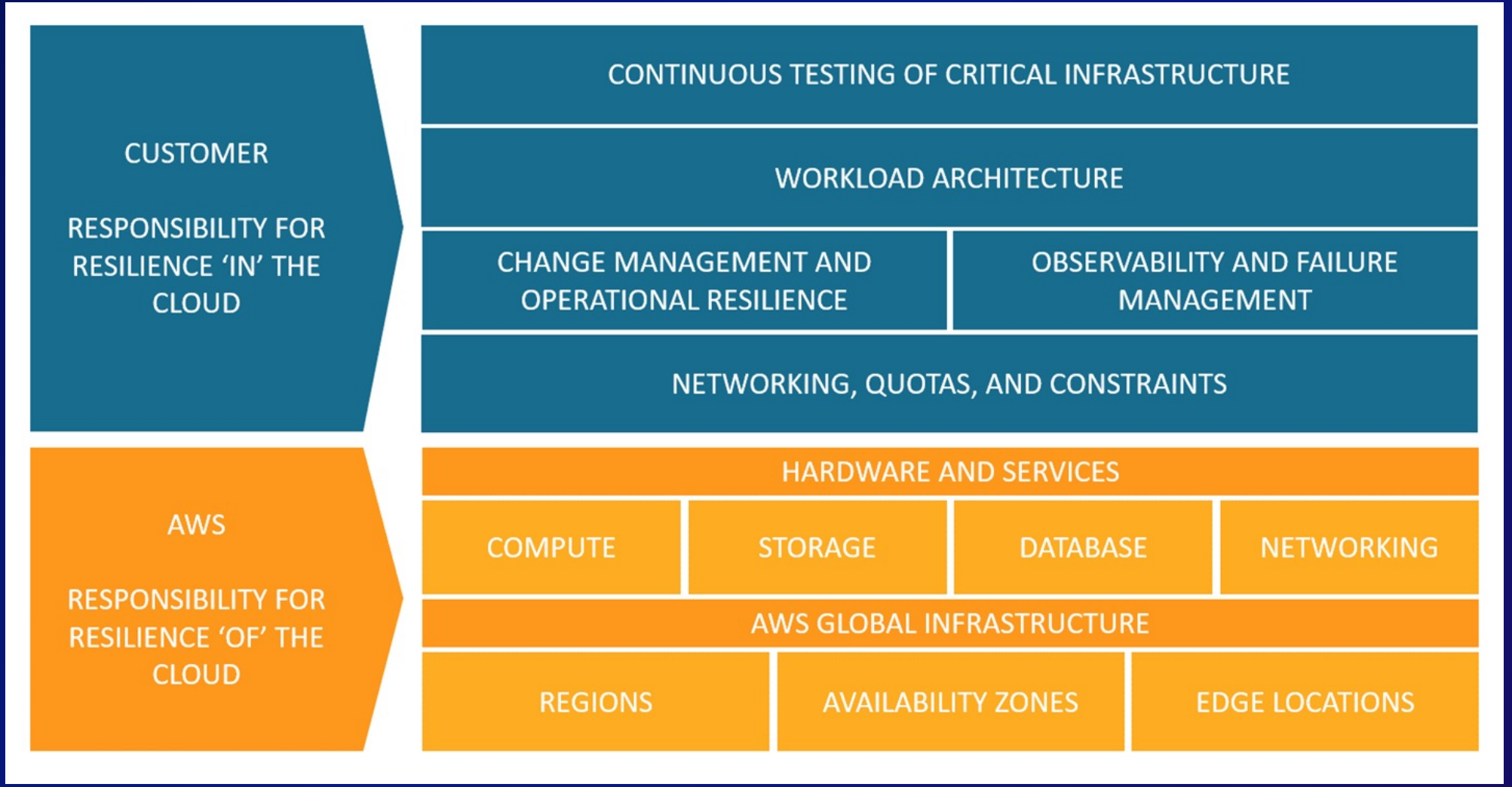
Hundreds of thousands
of customers





Trillions
of Lambda executions per month

Shared responsibility model for Resilience



What about Resilience goals?

How much data can you afford to recreate or lose?

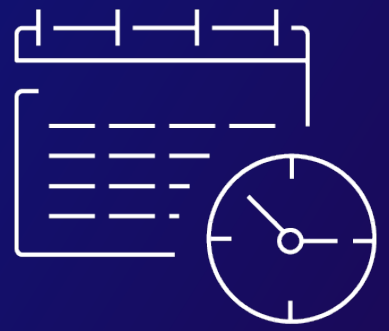
How quickly must you recover? What is the cost of downtime?

What is AWS Resilience Hub?

A new application resilience service that provides customers a central place to **define, validate, and track** the resilience of their applications on AWS

AWS Resilience Hub – Key capabilities

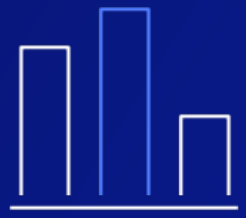
- Define application RTO and RPO in a **resilience policy**
- Assessment to **uncover resilience weaknesses**
- **Resilience score** indicates likelihood of meeting RTO and RPO
- Recommendations on **SOPs and alarms**
- **Resilience testing** and verification
- Dashboard to view **resilience posture**



Four essential capabilities in a resilient system

Anticipate

Understanding what to expect; imagining potential failures and mitigating those in advance



Monitoring

Understanding what to look for, in both internal and external conditions

Responding

Understanding what to do and adjusting responses, if necessary, in a flexible way

Learning

Understanding and sharing what has happened to promote learning and changes

Erik Hollnagel, “Epilogue,” in *Resilience Engineering in Practice*, https://amzn.to/3tVMTEx

Anticipate

CONTINUOUS RESILIENCE

AWS Lambda has fixed resources per unit of work

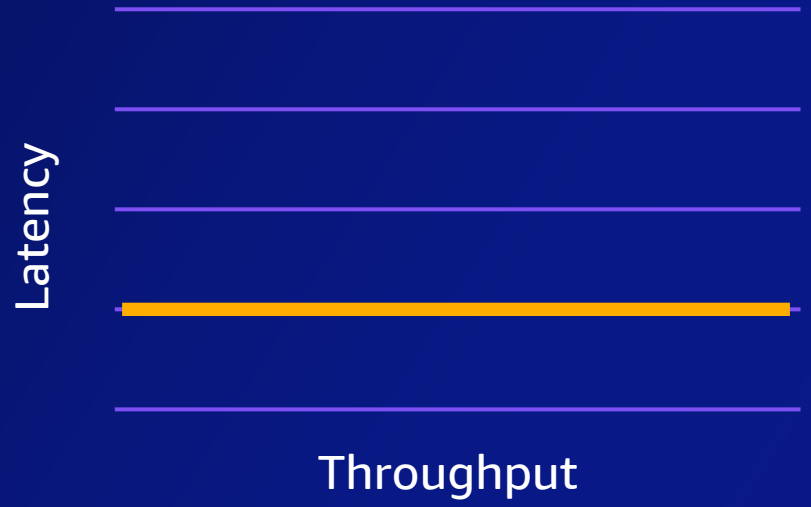
SCALING IS ONE OF THE MOST IMPORTANT – AND HIGHEST – CAUSES OF DISRUPTIONS

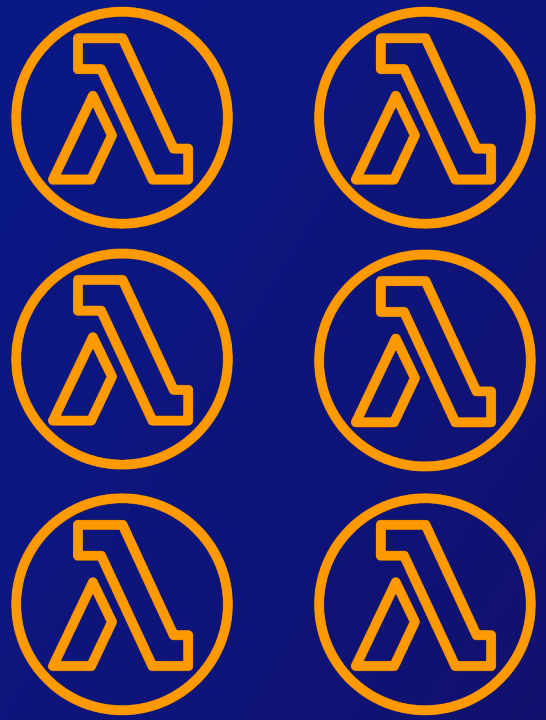
- Each **execution environment** has the same finite resources
- Each execution environment handles **one, and only one event** at a given time
- Each request gets the same resources

As work increases, scale concurrent environments*

Workload isolation means predictable performance

Latency and throughput are constant

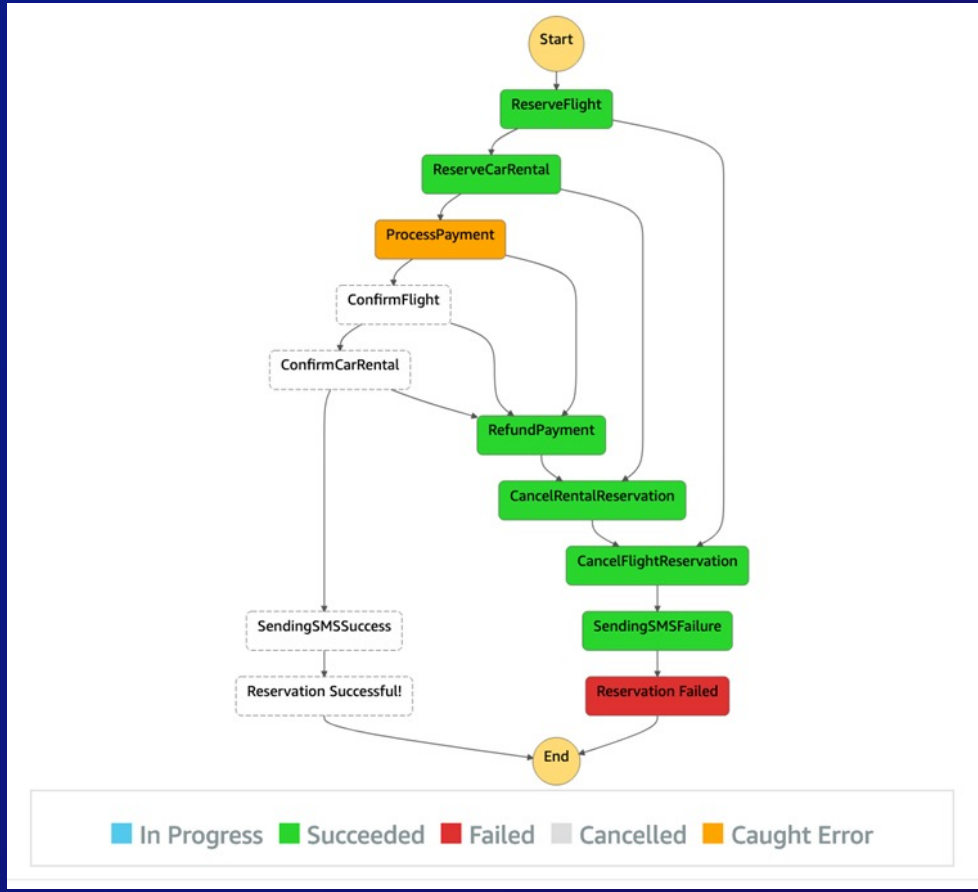




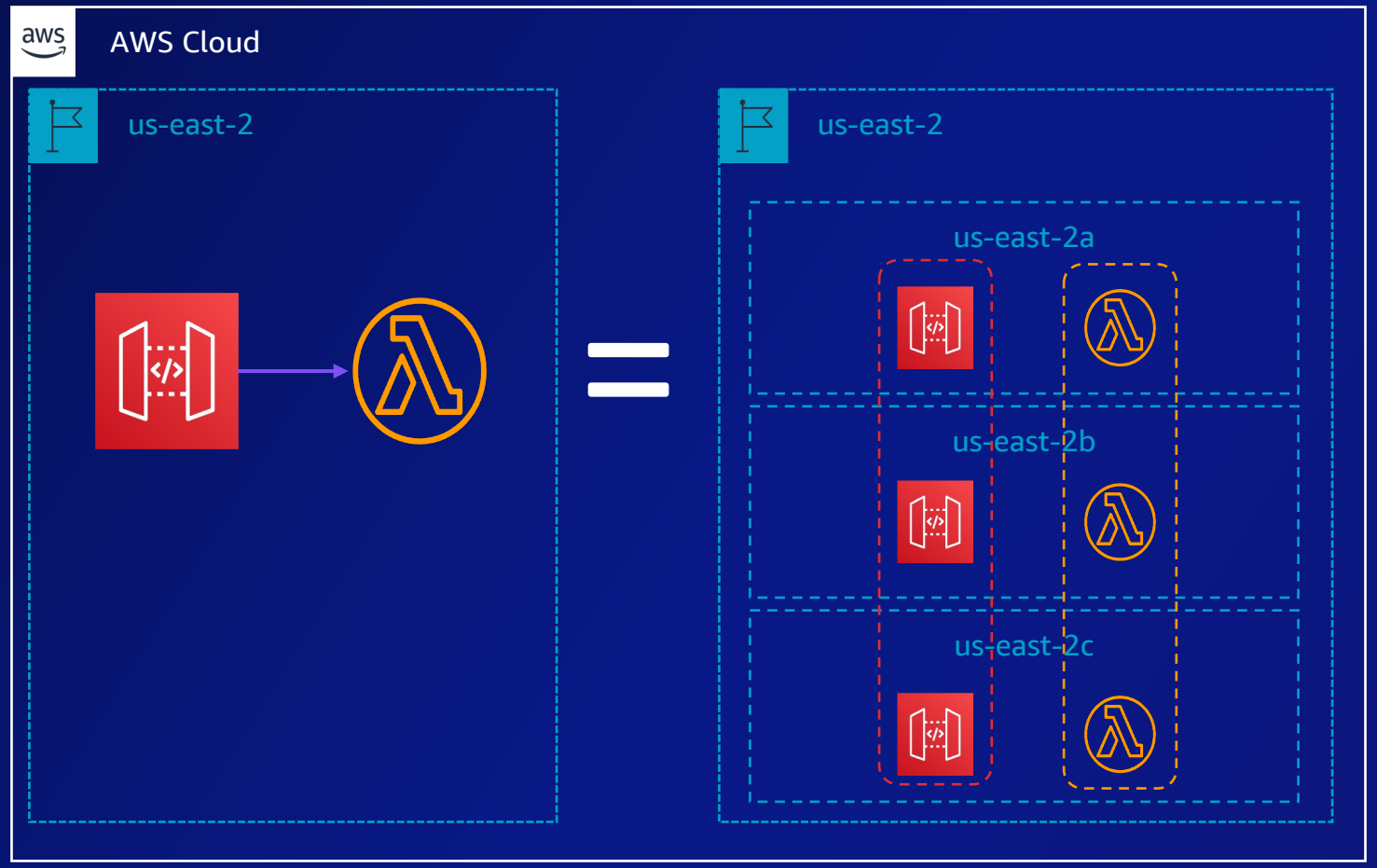
*Concurrency limits apply

If you have long-running transactions, use AWS Step Functions

- AWS Lambda can run up to 15 minutes per execution
- With AWS Step Functions you can string multiple Lambda functions together …
- This has resilience benefits too!



With serverless, multi-AZ is included



However, If you are attaching Lambda to a VPC, you do need to include subnets in multi-AZ – this is not automatic

Is there anything we can do to anticipate in a more automated way?

Using Resilience Hub to diagnose serverless applications

AWS Resilience Hub can analyze an existing workload containing serverless and provide the estimated RTO and RPO for:

Regional – A region becomes unavailable	**Infrastructure – Hardware becomes unavailable**
AZ – An AZ becomes unavailable	**Application** – Application code or data issues

RTO — Resiliency recommendations

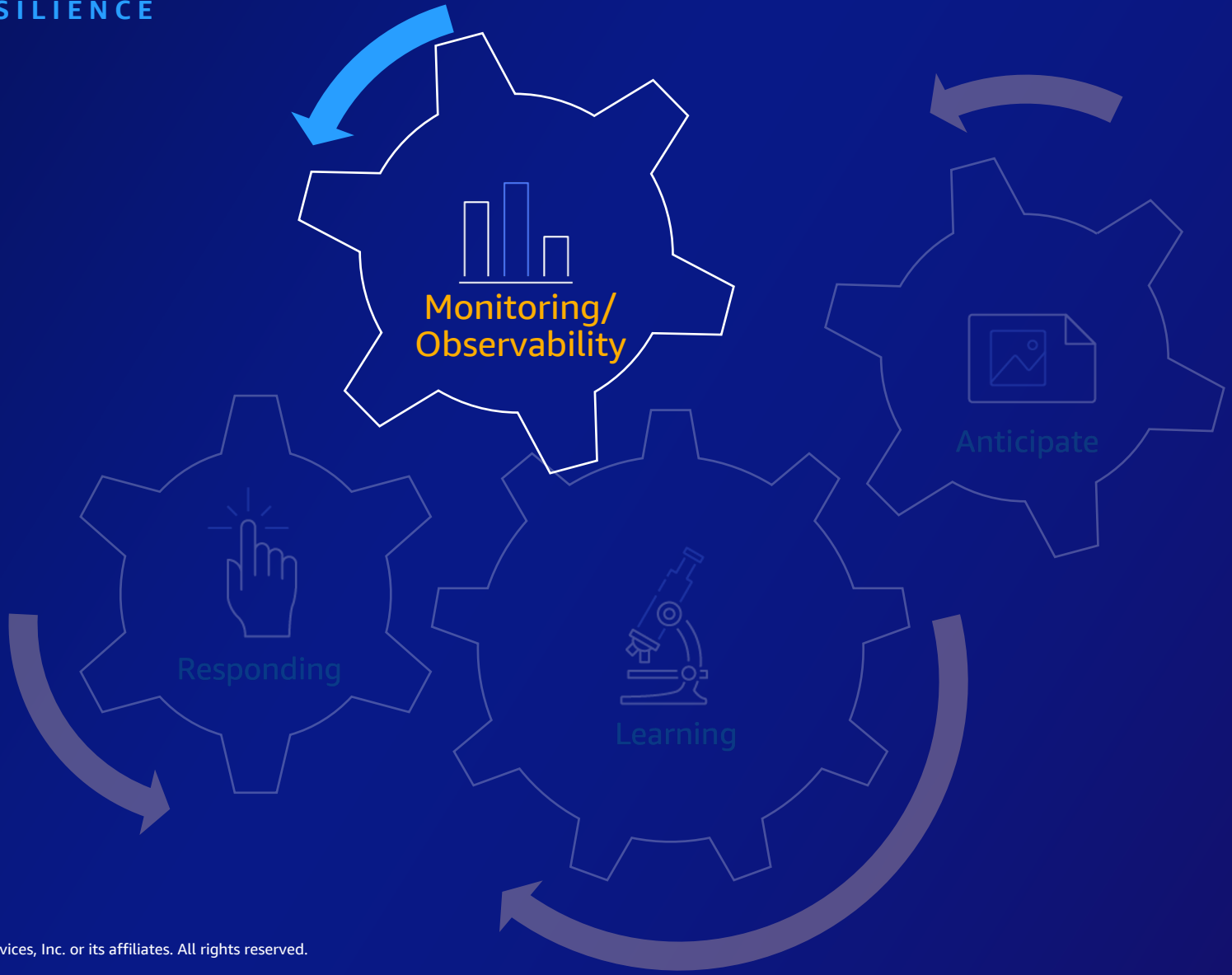
Disruption type	Targeted	Estimated
Application	1h	10m
Infrastructure	1h	0s
Availability Zone	1h	0s
Region	-	-

RPO — Resiliency recommendations

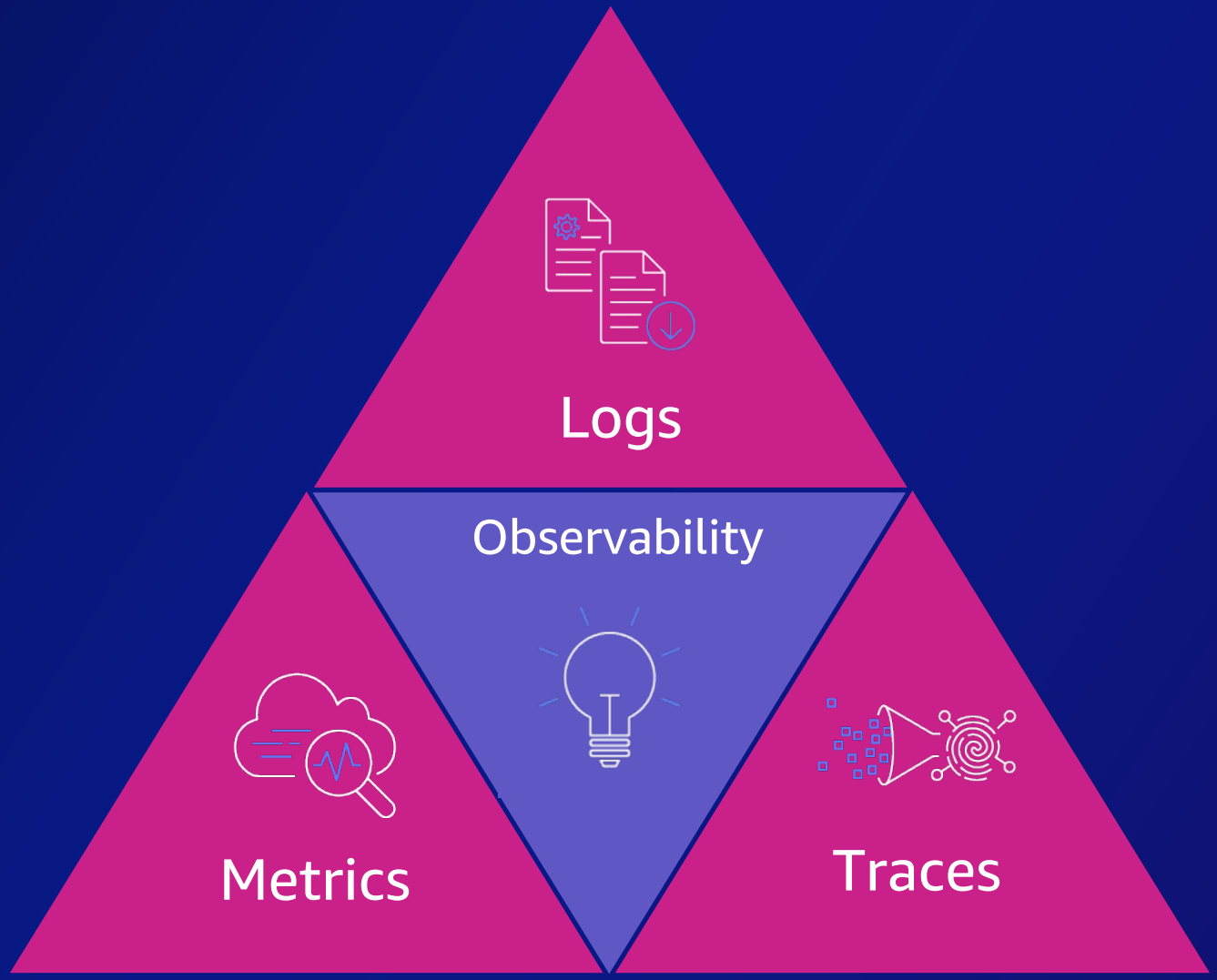
Disruption type	Targeted	Estimated
Application	15m	unrecoverable
Infrastructure	15m	0s
Availability Zone	15m	0s
Region	-	-

Monitoring

CONTINUOUS RESILIENCE

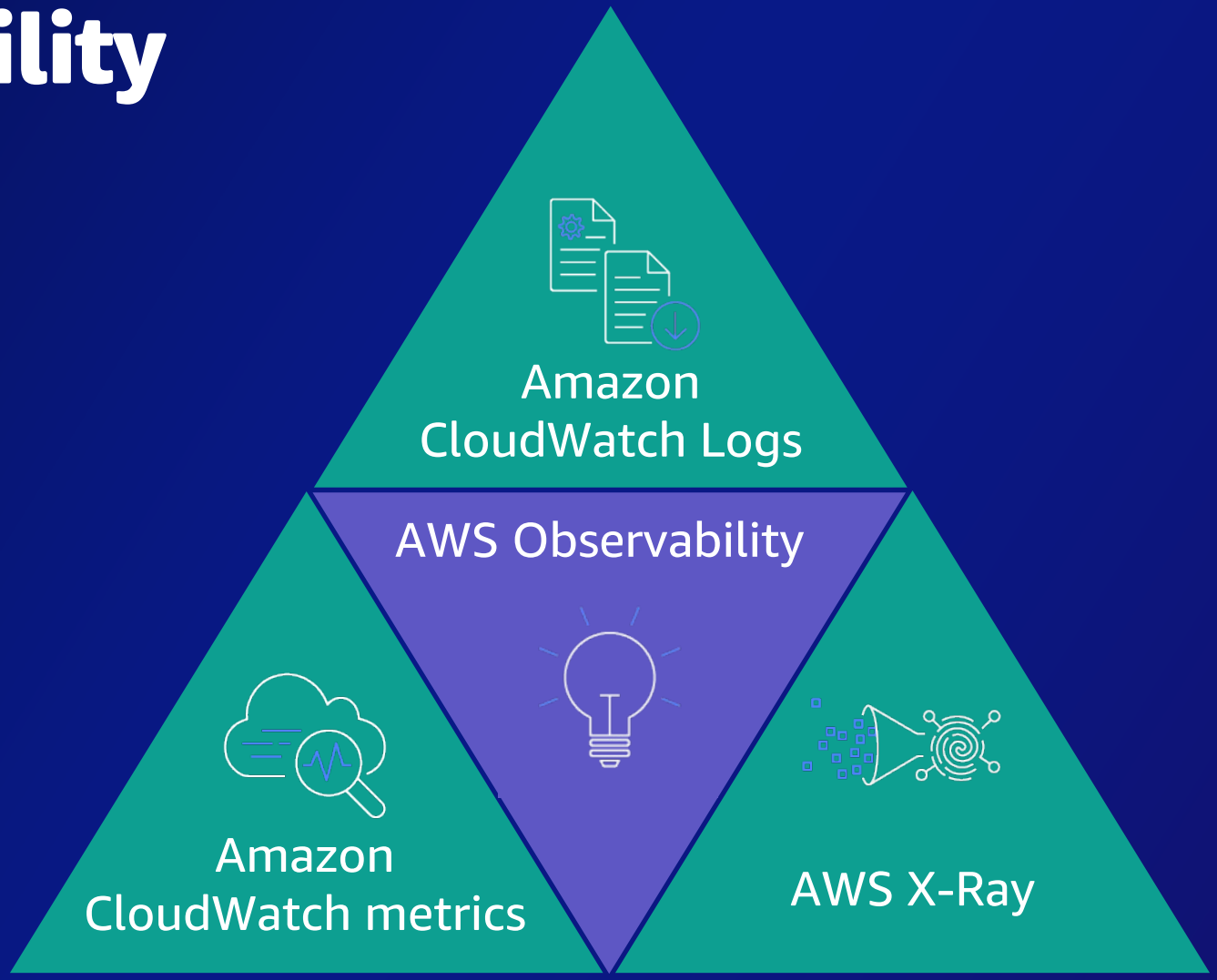


The 3 pillars of observability



Cindy Sridharan, *Distributed Systems Observability*, https://bit.ly/3AEB9Kp

AWS has 3 main services to help with observability



Cindy Sridharan, *Distributed Systems Observability*, https://bit.ly/3AEB9Kp

Take action with Amazon CloudWatch

AWS Metrics

AWS Lambda

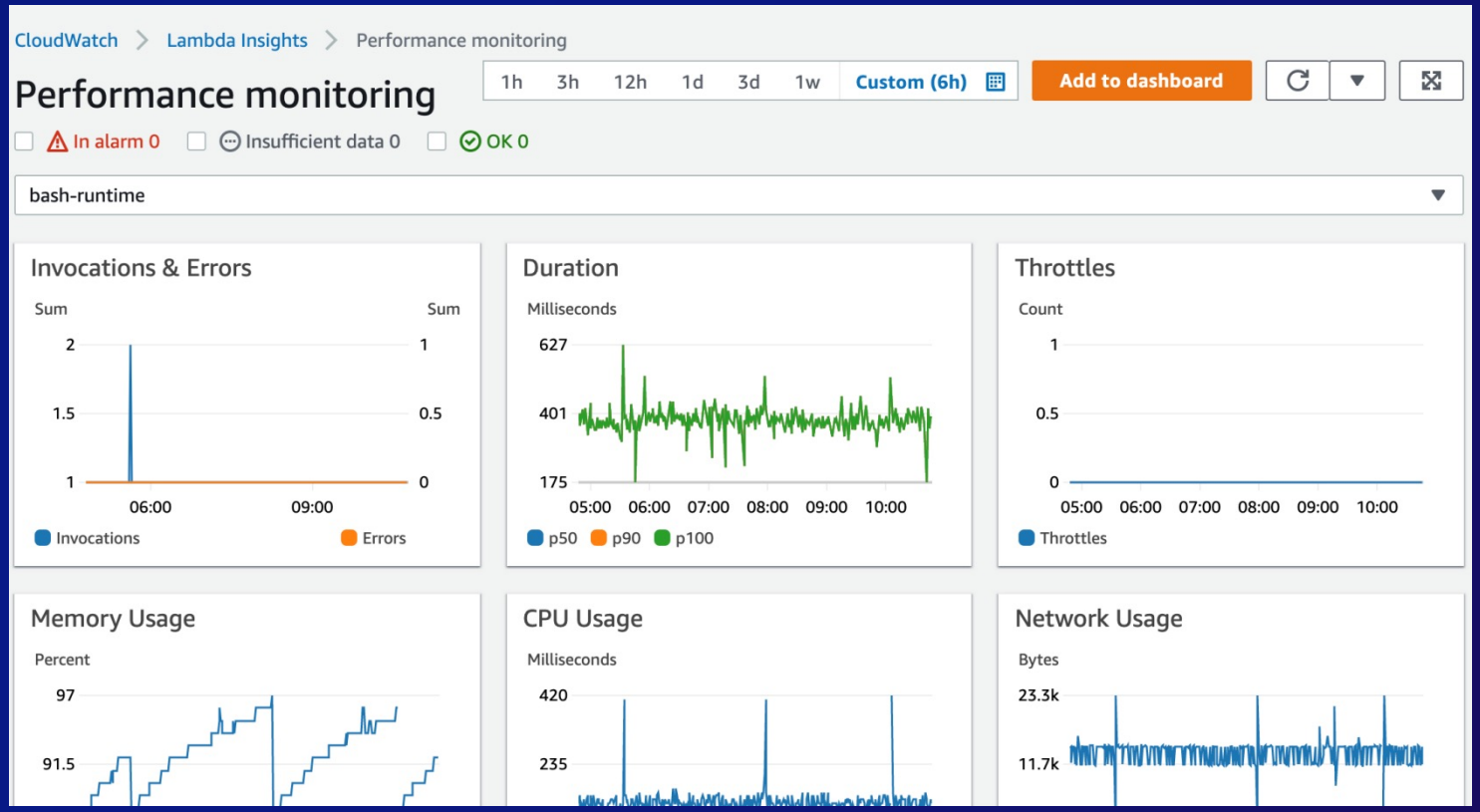
- Errors, throttles, iterator age, concurrency

Amazon API Gateway

- Success rate
- Latency (tail p90, p95, p99)

Amazon SQS

- Message age



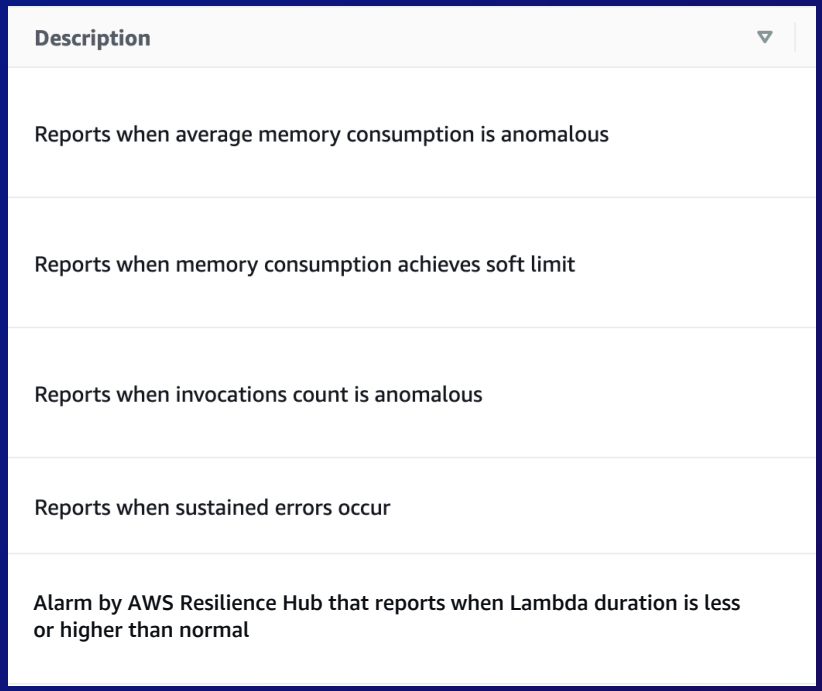
AWS X-Ray enables tracing of distributed applications

- Scales for microservice and serverless architectures
- Identify the root cause of performance issues and errors
- X-Ray provides a cross-service view of requests made to the application

AWS Lambda metrics and alarms via Resilience Hub

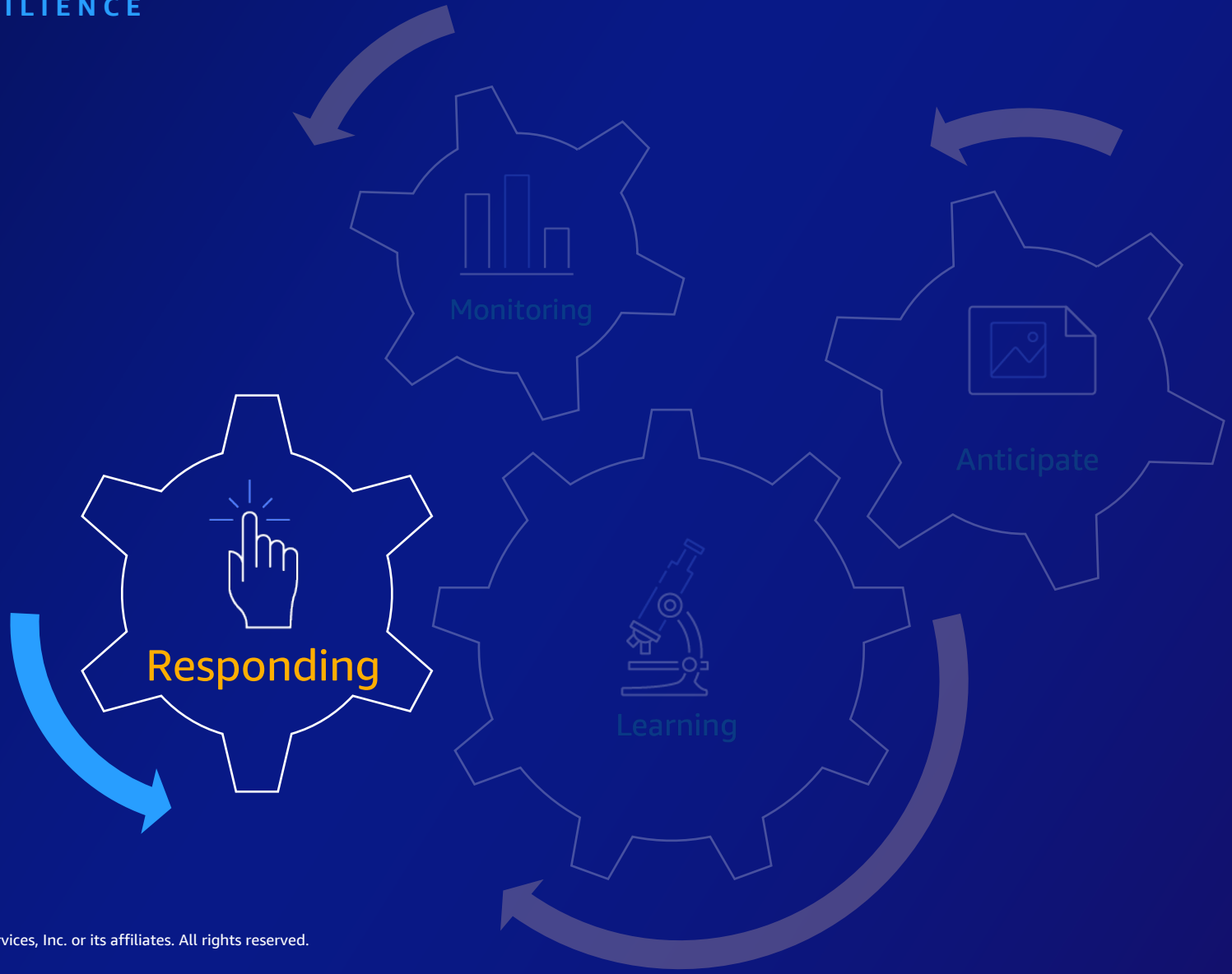
Resilience Hub recommends different metrics that should be monitored

- Memory
- Invocation
- Sustained errors
- Duration



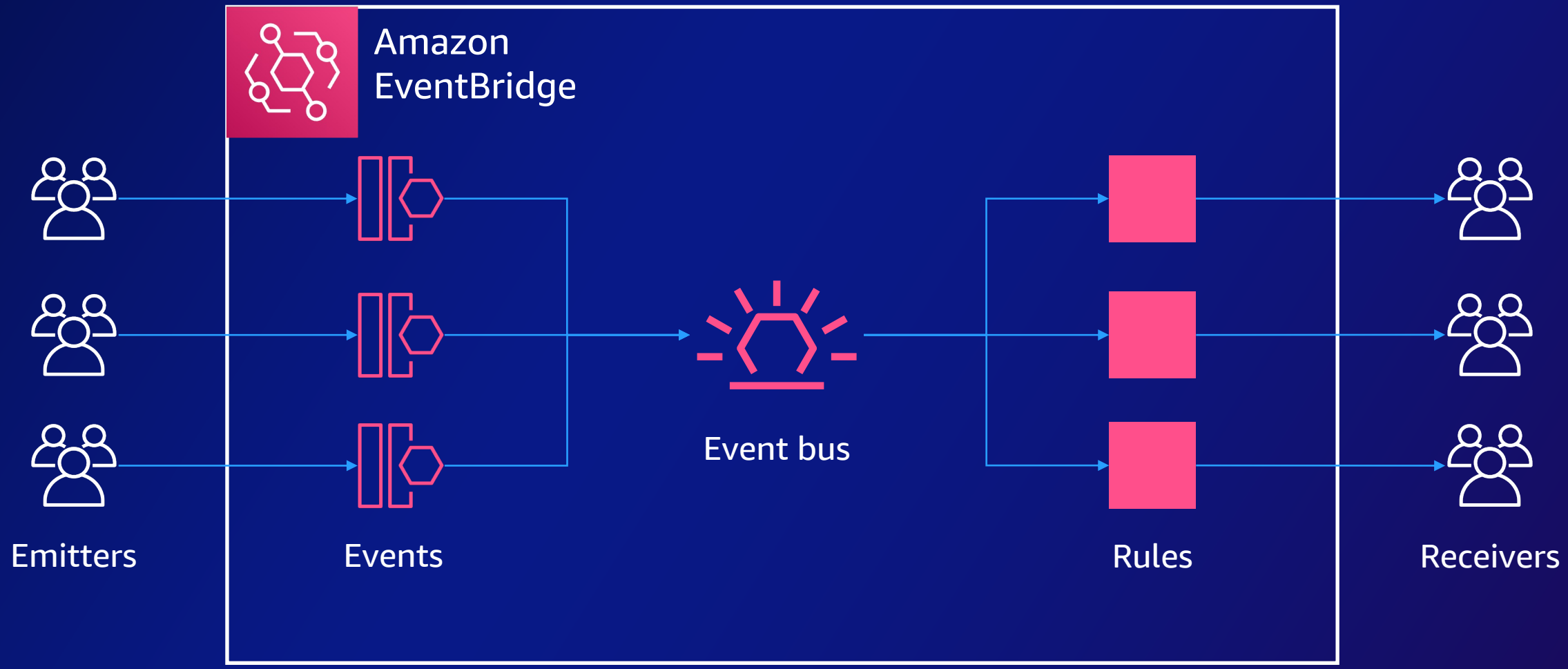
Responding

CONTINUOUS RESILIENCE



Event-driven patterns

AUTOMATED RESPONSES THAT RUN IN RESPONSE TO EVENTS



Lambda and Amazon SQS recovery procedures from Resilience Hub

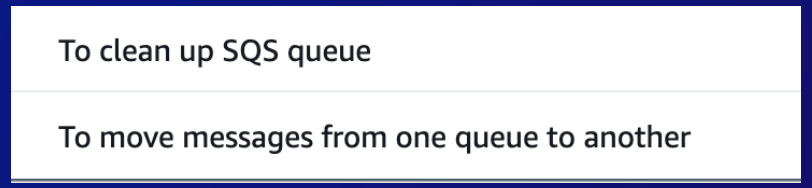
Resilience Hub recommends automated procedures to invoke a recovery if needed

Lambda

- Switch alias
- Restore memory size
- Restore execution time
- Restore provisioned concurrency

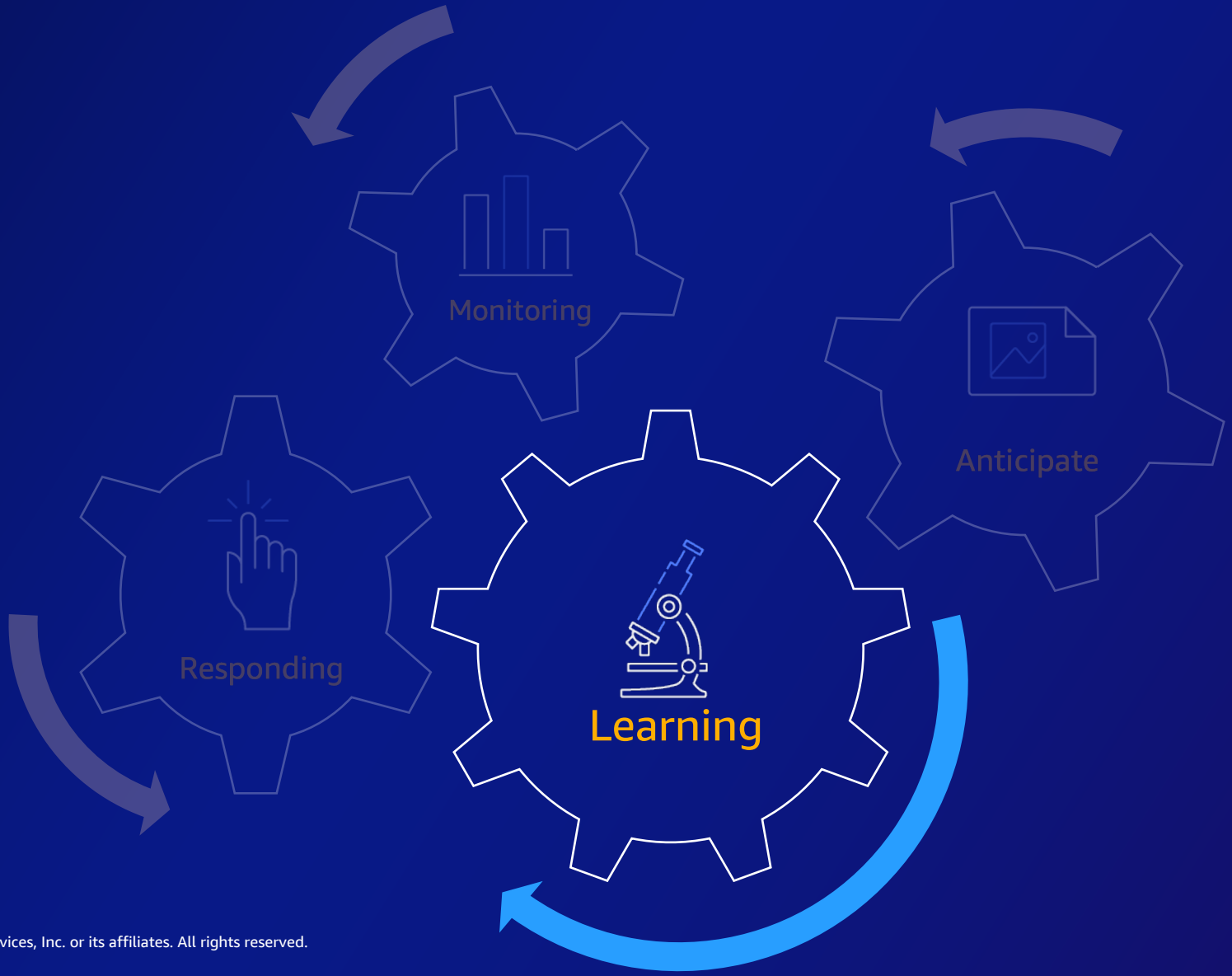
Amazon SQS

- Move messages
- Clean up queue



Continuous resilience

LEARNING



Chaos engineering

WITH AWS FAULT INJECTION SIMULATOR (AWS FIS)

Improve resilience and performance

Uncover hidden issues

Expose blind spots
Monitoring, observability, and alarm

Fix failures before they become outages

Lambda and Amazon SQS testing from Resilience Hub

Resilience Hub recommends AWS Fault Injection Simulator (AWS FIS) tests to verify resilience under stress

- Lambda thresholding
- Amazon SQS deletion failure
- Unsent messages
- Message size
- Allowable tries

Test Lambda behavior when hitting ReservedConcurrentExecutions value
Test behavior when messages are not deleted from a specific queue
Test behavior when messages cannot be sent to an SQS queue
Test SQS behavior after sending a message of size larger than threshold
Test standard Queue behavior after receiving a message maximum allowed times

Demo

What you should take away from this !!

CALL TO ACTION: APPLY TO ALL YOUR SERVERLESS WORKLOADS

Anticipate

- What are you trying to anticipate?
- Understand your RTO/RPO
- Understand your own architecture

Monitoring

- Use the tools out there for your specific business needs
- Design metrics and traces that matter for your scenario

Responding

- Take automated action when something does occur

Learning

- Test, test, test!
- Test the unknown with chaos engineering
- Use this data to improve your architecture

Resources

Serverless on AWS

https://aws.amazon.com/serverless/

AWS Resilience Hub resources

https://aws.amazon.com/resilience-hub/resources/

AWS Monitoring and Observability

https://aws.amazon.com/cloudops/monitoring-and-observability

AWS Fault Injection Simulator

https://aws.amazon.com/fis/

skillbuilder.aws

Your time is now

Build in-demand cloud skills your way

Thank you!

Please complete the session survey

Santhakumar Rajangam
Startup Solutions Architect
AWS India

Varun Kumar
Enterprise Solutions Architect
AWS India