AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

CON314

Building event-driven services on Amazon ECS

Vibhav Agarwal (he/him)
Sr. Product Manager
AWS

Maish Saidel-Keesing (he/him)
Sr. Developer Advocate
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

What is an event-driven service?

Building an event-driven service on ECS

Considerations and takeaways

Q&A

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Key use cases we hear from customers

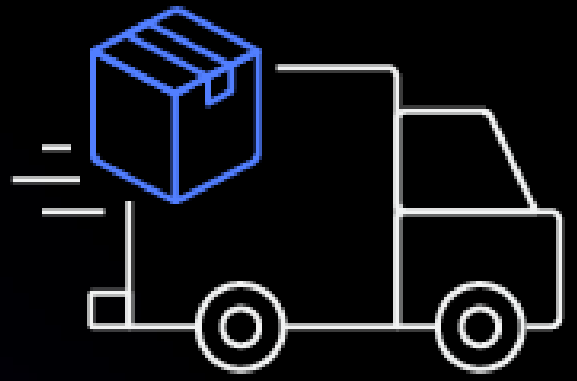
Data validation



Fraud detection



Inventory management



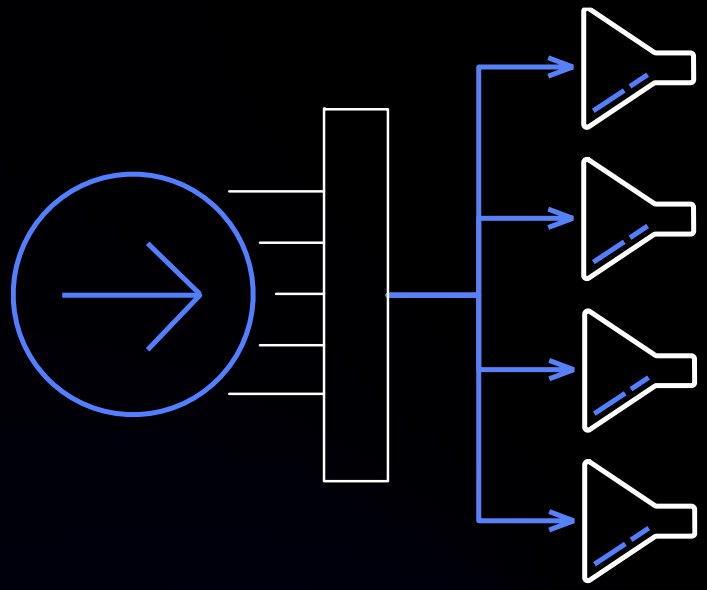
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Event-driven services perform work in response to events

Generally asynchronous

Decouples components

Extensible and resilient



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3 reasons to use ECS for event-driven services

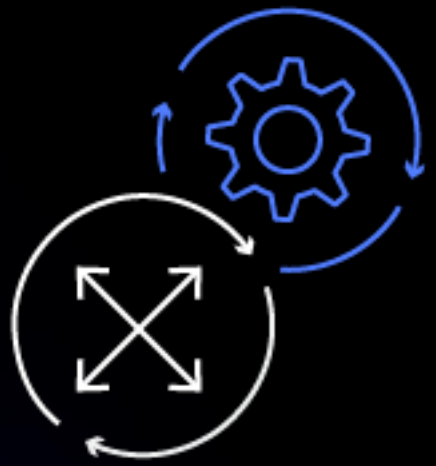
Balance abstraction & power



Reduce costs



Flexible scaling



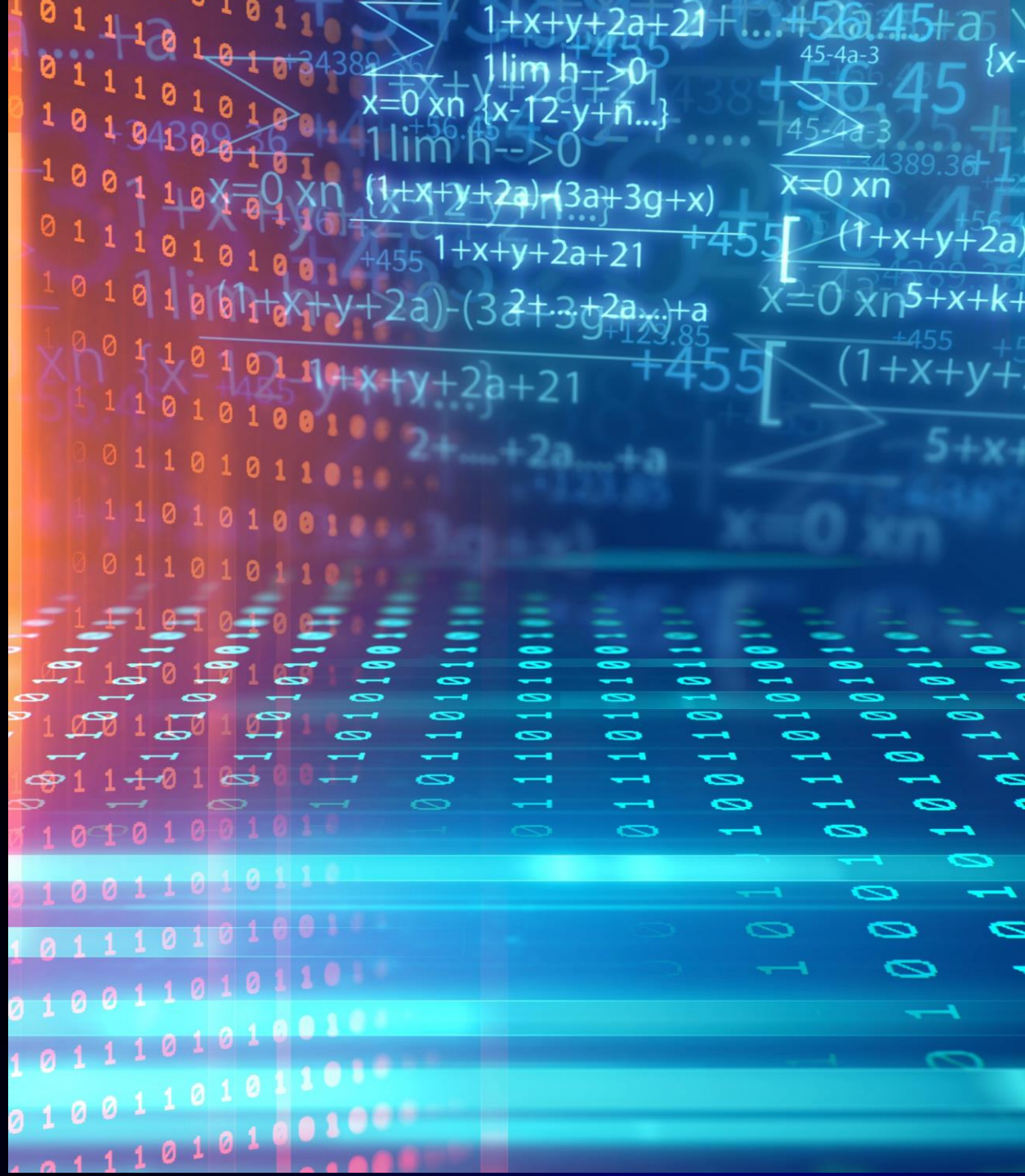
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example use case

Monitoring transactions for fraud

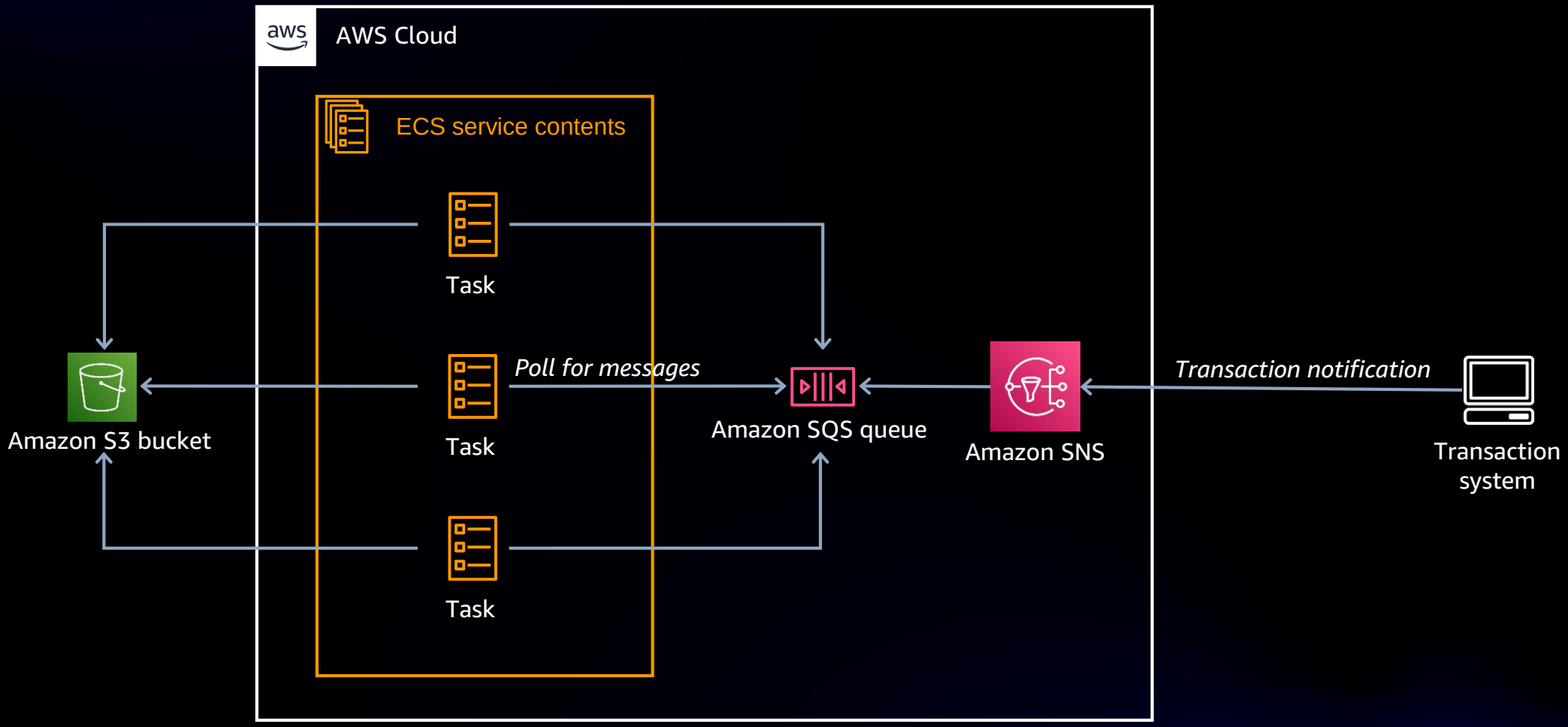
Fraudulent transactions are harmful for both consumers and merchants

Detecting suspicious transactions early and automatically can help save time and money, and build trust with customers



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Transaction monitoring system



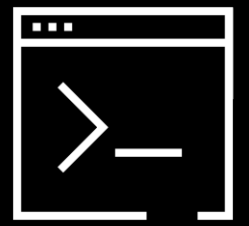
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Whiteboard

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Powerful simplicity with CDK and Copilot

Use patterns for services that process SQS queues in AWS Copilot and ECS CDK library



AWS Copilot
Worker Service pattern

ECS manages configuration of the task definition, the SQS queue, and autoscaling policies



AWS CDK
QueueProcessingServiceBase

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Considerations

Streams or queues

- Streams allow for near real-time information flow
- Queues allow for more decoupled and asynchronous communication

Be aware of your message queue limits

- For example, Amazon SQS maximum visibility timeout

Proper event design for better decoupling

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Takeaways

CDK/Copilot abstractions for added simplicity

Optimize costs with AWS Fargate and Spot

Configurable scaling based on your needs

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

References

Event-driven applications with Copilot

Whitepaper for event-driven applications

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Vibhav Agarwal
agvibhav@amazon.com

Maish Saidel-Keesing
maishsk@amazon.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.