AWS re:Invent

ANT331

AWS analytics enables fraud prevention for Sony's PlayStation

Eric Krauss

Manager, Platform Engineering – Fraud/Risk

Sony Interactive Entertainment

Tom Lilley

Staff Engineer, Platform Engineering – Fraud/Risk

Sony Interactive Entertainment

Nicholas Walsh

Technical Evangelist

Amazon Web Services

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Introductions

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Know your presenters—Eric Krauss

- Engineering manager in SIE's Commerce Engineering department
- 25 years of experience in financial services and e-commerce
- Been with Sony Interactive Entertainment for 11 months (Love the new dress code)
- Background in software engineering as full stack in Java and .NET
- Active speaker focusing on DevOps, agile, and API management
- Married for 12 years and proud father of 2 boys



Know your presenters—Tom Lilley

- Staff software engineer in SIE's Commerce Engineering department
- Technical lead for the fraud and risk team
  - Building backend services to do real-time fraud and risk analysis as customers interact with the PlayStation store
- 10+ years of software development experience
  - 3 years at SIE
  - Focus on low-latency, high-volume transactional systems
- Fairly new to AWS, only ~2 years of experience on the platform
  - First re:Invent was 2017



Know your presenters—Nicholas Walsh

- Technical evangelist at Amazon Web Services
- Work primarily with AI/ML + game tech
  - But really, I enjoy working with anything that involves solving problems with data
- Unconventional developer history, academia + research
  - Helped build shark species identification system to curb illegal shark fin trafficking
  - Also: neurology software for apathy diagnosis; food science startup trying to make a sustainable egg replacement product
- First taught AWS to college students at hackathons, years before I worked at AWS



About Sony Interactive Entertainment

- Headquartered in San Mateo, California, SIE is a wholly owned subsidiary of Sony Corporation and has regional headquarters in London and Tokyo
- Recognized as a global leader in interactive and digital entertainment, Sony Interactive Entertainment (SIE) is responsible for the PlayStation brand and family of products
- PlayStation has delivered innovative products to market since the launch of the first PlayStation in Japan in 1994
- The PlayStation family of products and services includes PlayStation 4, PlayStation VR, PlayStation Store, PlayStation Now
- SIE also oversees Worldwide Studios, which is responsible for developing exclusive, world-class games for PlayStation



Agenda

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda



- Overview of initiative
- Overview of technology
- Review of results
- Lessons learned
- Q&A

Overview of initiative

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Protecting customers on the PlayStation Store

Vision

To attempt to make PlayStation network a world class place for players to join, to stay, and to enjoy without worries. To protect the PlayStation brand name and revenue against fraud and exploitation.

Mission

We will attempt to do this by employing a real-time data-driven personalized risk screening approach to proactively identify and stop bad actors.

Event **V**erification for **E**nterprise (EVE) is a platform that allows for inline fraud/risk decision on a user initiated transaction within the PlayStation commerce platform.

EVE’s goal for PlayStation Network

- 360-degree view of real-time risk on PSN
- Granular control of our destiny
- World-class customer experience
- Centralized risk management
- Ability to proactively deal with bad actors
- Ability to plug and play with leading-class technology



Use case N

The Challenge:

- For the PlayStation Network to improve its ability to assess events, in real time, on the commerce platform for risky or fraudulent activity

The Goal:

- To create a multi-tenant system that can be consumed as SaaS by different transactional events on the PSN platform

The Solution:

- **EVE** has been created to be a one-stop shop risk-management system



Use Case 1—purchase

The Challenge:

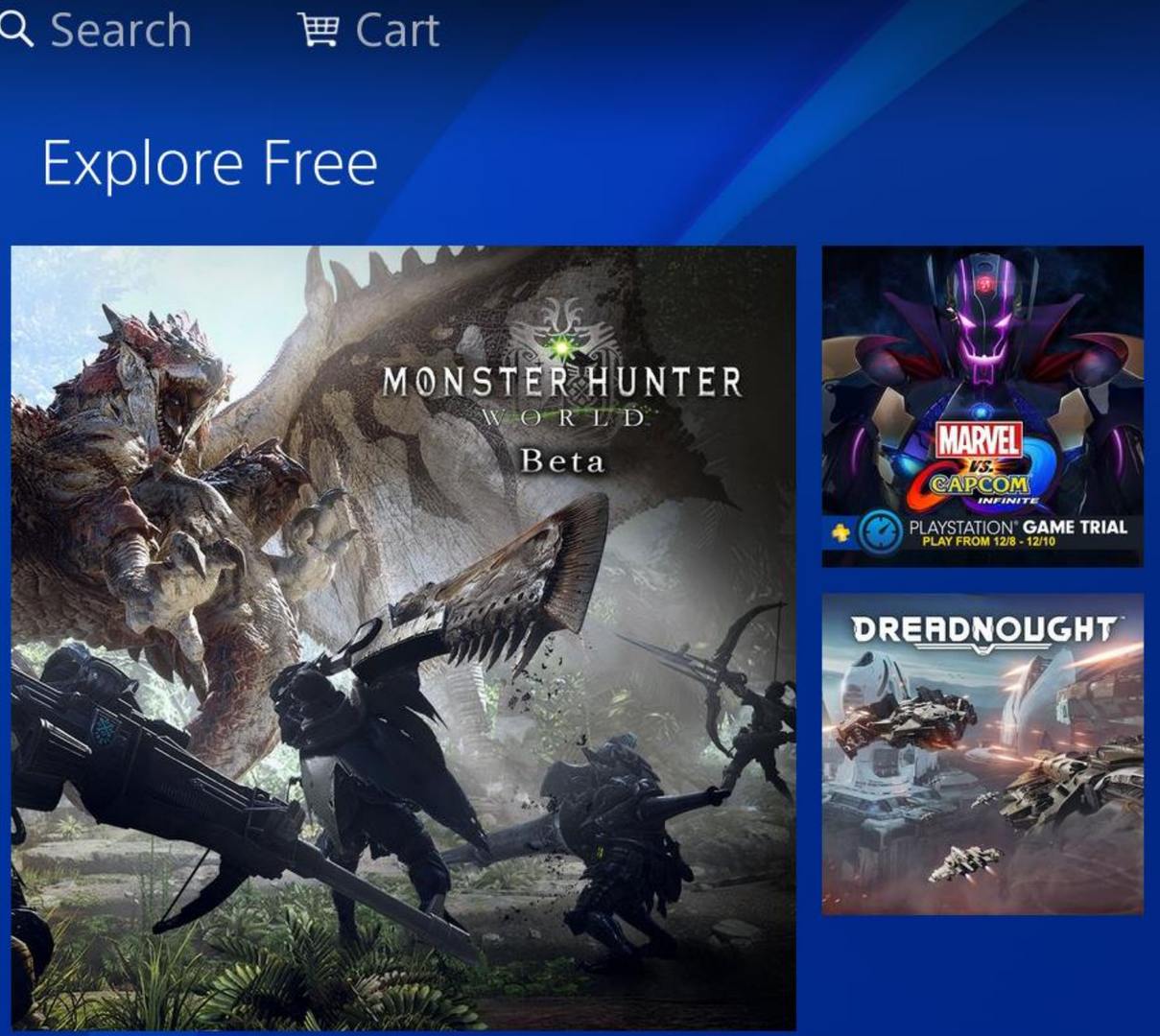
- To further prevent fraudsters, who would use stolen or fraudulent card and user information to purchase digital goods on the PlayStation Network

The Goal:

- Have the highest transaction approval rate possible, while having the lowest chargeback rate possible

The Solution:

- **EVE Purchase** evaluates each purchase transaction initiated from the PS Store and provides approval/denial leveraging Big Data and advanced ML



Use Case 2—login

The Challenge:

- To further prevent fraudsters who would use stolen account information to perform an Account Take-Over (ATO)

The Goal:

- To protect the network and users from malicious intent

The Solution:

- **EVE Login** evaluates each user login on the web or mobile platform and provides approval/challenge/denial, leveraging Big Data and advanced ML



Why AWS technologies?

- SIE successfully migrated their core systems to AWS in 2018
- SIE's journey to the cloud was happening at the same time the EVE project was getting underway
- Unique opportunity to leverage managed resources and focus on business use cases rather than managing infrastructure
- Saw an opportunity to accelerate timeline for delivery to market



Amazon EC2

Amazon Kinesis

AWS Glue

AWS Lambda

Amazon DynamoDB

Amazon Simple Storage Service (Amazon S3)

Amazon CloudWatch

AWS CloudFormation

AWS managed services benefits



- Scalable
  - All components in the system horizontally scale (and to do so is very easy operationally)
- Maintainable
  - Engineers and operations teams are not tied up in the day-to-day maintenance of these services (upgrades, patches, etc.)
  - Simple CloudWatch alarms in place to notify us of issues or the need for more scale

AWS serverless architecture core values



- No infrastructure provisioning or management
- Pay for value (per action or invocation)
- Automatic scaling, high concurrency and availability
- Architecture should consist of “small pieces, loosely joined”
- Reduce scope per microservice to keep complexity manageable

Data storage

Amazon S3

- Highly durable object storage (99.999999999%) for datasets of any size
- Put or access data anywhere, supports cross-region replication



Amazon S3

DynamoDB

- Key-value (NoSQL) database with single-digit millisecond queries
- High throughput at scale (10 trillion requests/day and 20 million r/s)



DynamoDB

Compute

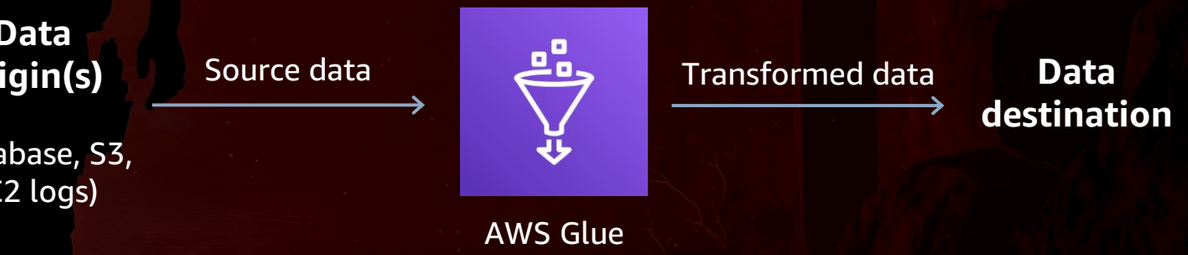
AWS Lambda

- Serverless, event-driven code execution
- Runtimes for all languages
- Set data sources as triggers



AWS Glue

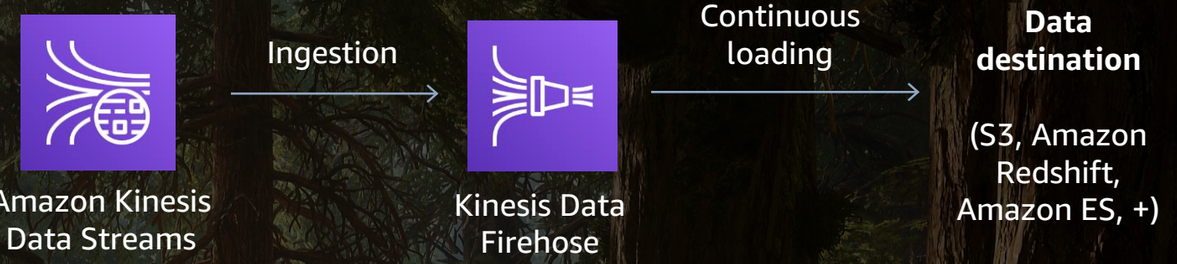
- Serverless ETL, create custom transformation jobs for Spark
- Automatic schema detection from data source



Data transport

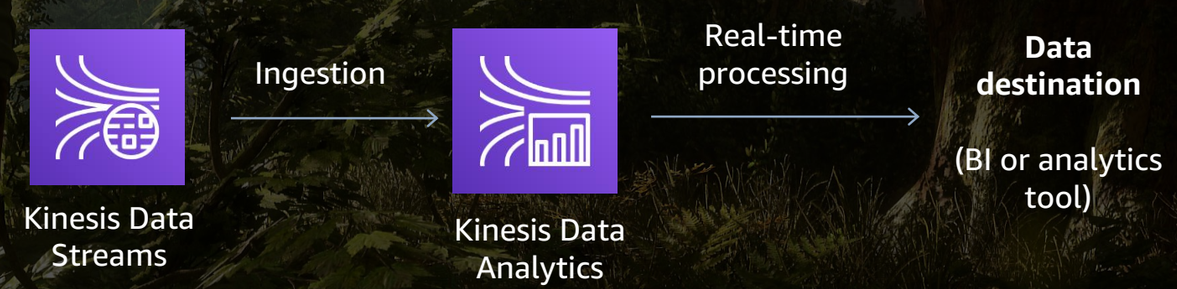
Amazon Kinesis Data Firehose

- Reliably load streaming data to data stores or analytics tools
- Ability to batch, transform, encrypt, and more



Amazon Kinesis Data Analytics

- Easily process streaming data in real time
- Reduces complexity by streamlining analysis as it’s transferred



Architecture management

CloudWatch

- Monitoring and observability for applications and infrastructure
- Set custom metrics for resource utilization and operational health from logs, metrics, and events



CloudWatch

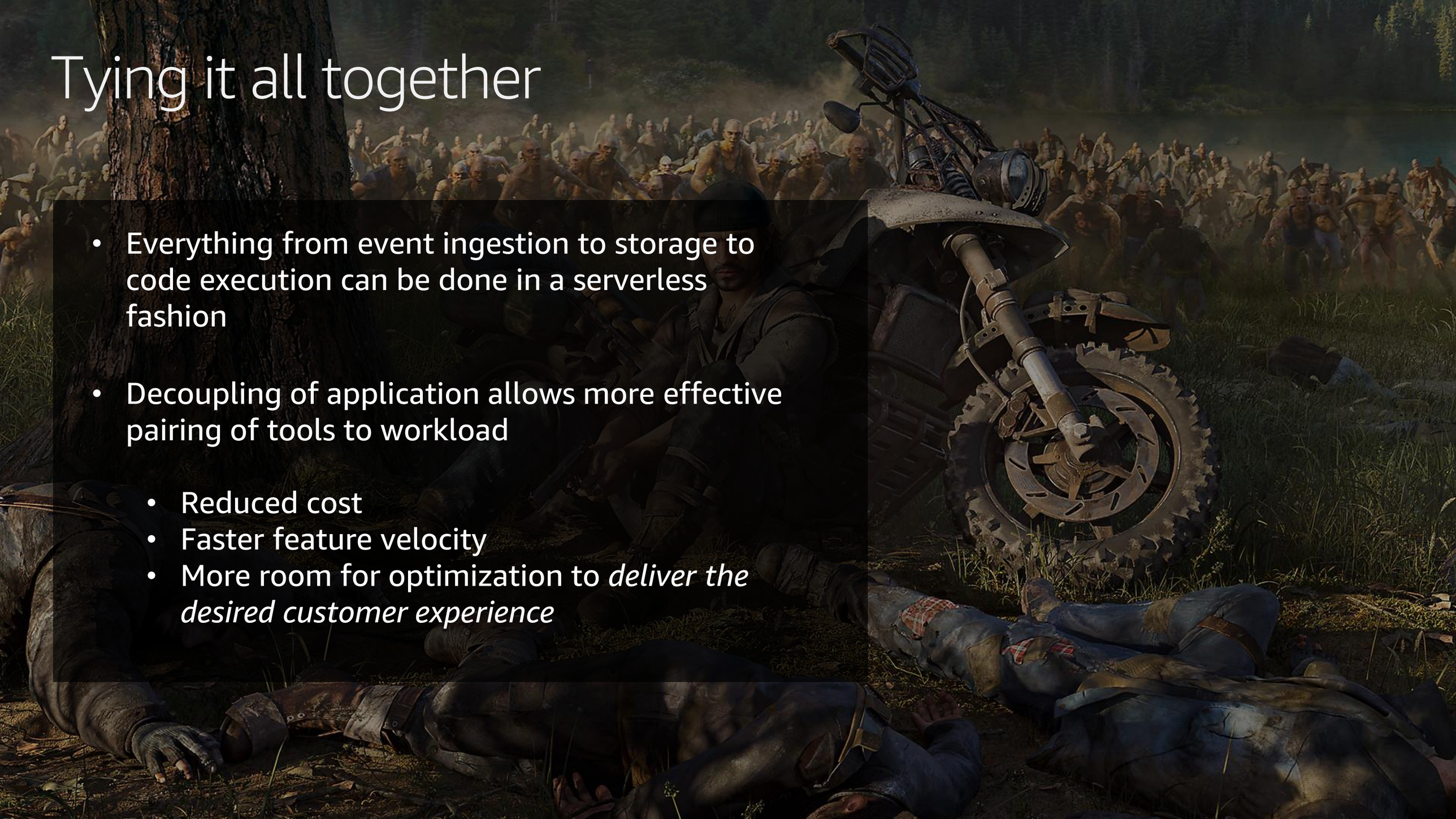
AWS CloudFormation

- Infrastructure as Code (IaC), streamline the provisioning and deployment process
- Config file defines architecture, enabling automated and reproducible deployments for your entire stack



AWS CloudFormation

Tying it all together



- Everything from event ingestion to storage to code execution can be done in a serverless fashion

- Decoupling of application allows more effective pairing of tools to workload

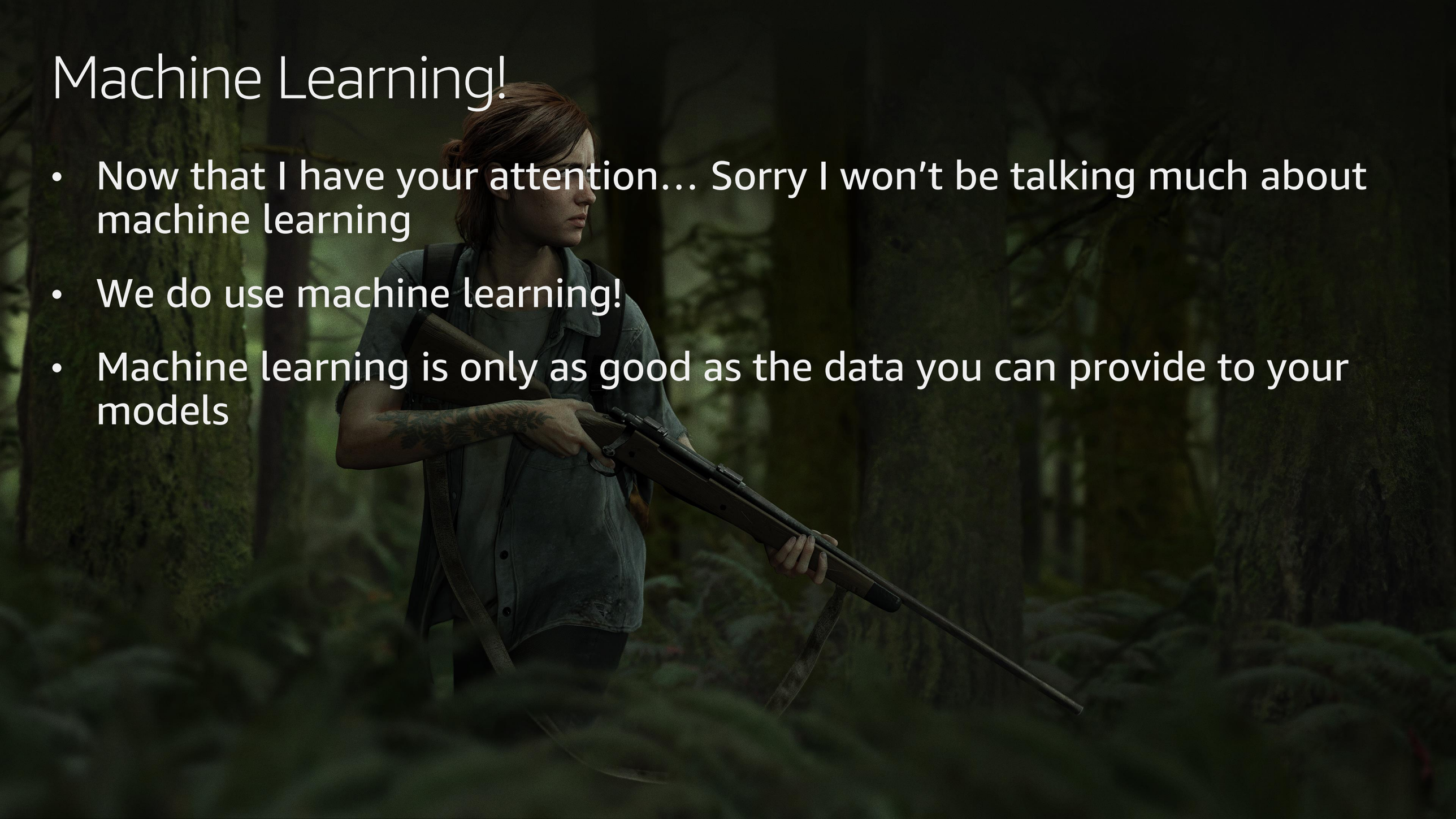
    - Reduced cost
    - Faster feature velocity
    - More room for optimization to *deliver the desired customer experience*

Overview of technology

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Machine Learning!

- Now that I have your attention… Sorry I won’t be talking much about machine learning
- We do use machine learning!
- Machine learning is only as good as the data you can provide to your models

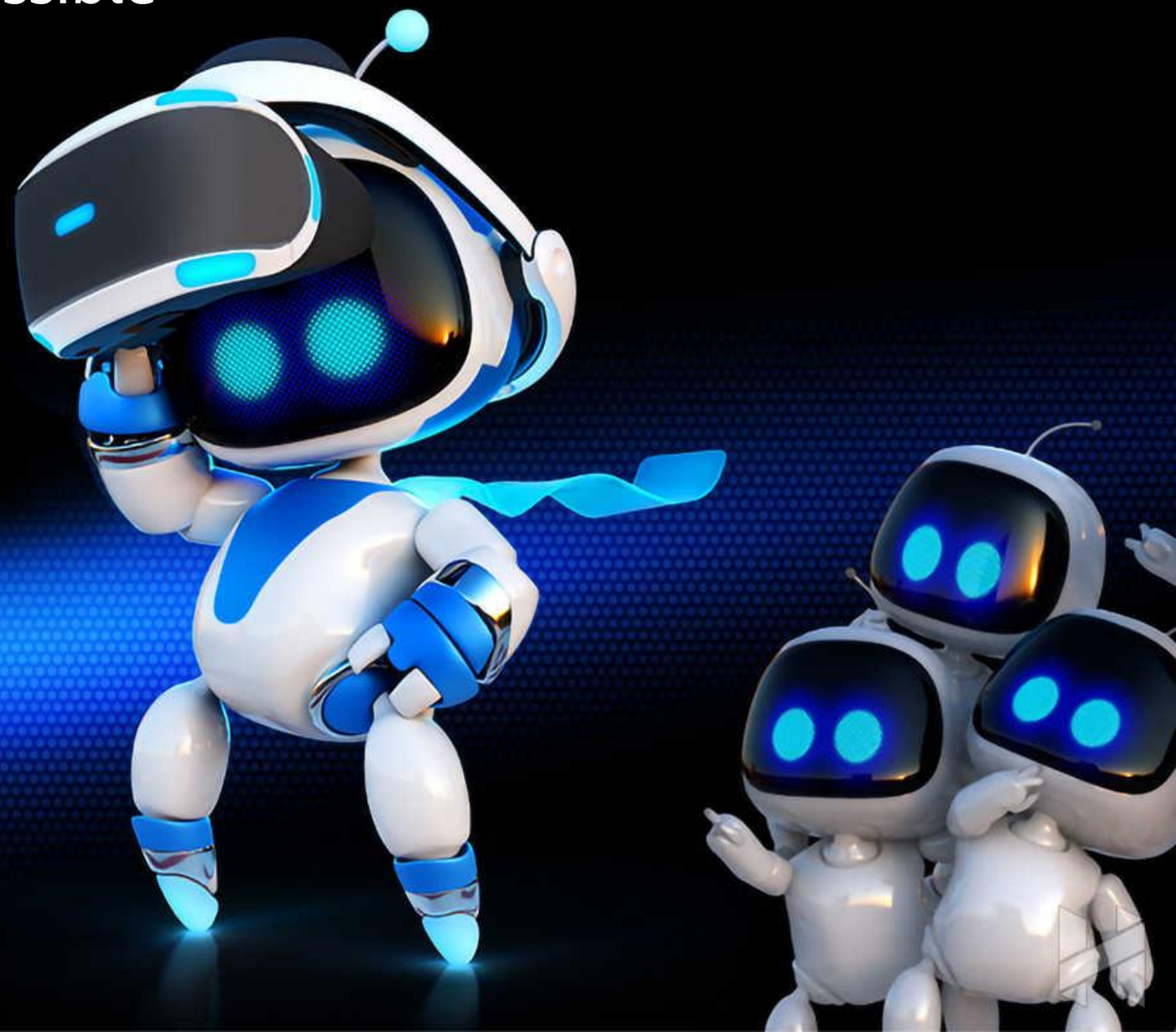


Aggregating data over time

- Most powerful indicator for risk is based on contextual data
- Our real-time analytics system provides sets of aggregated data of events on our platform over time.
- Examples:
  - How many unique IP addresses have I seen for this account in the last 7 days?
  - How many times has this device attempted to buy something in the last hour?
- The system we built on AWS creates and delivers these aggregations to our machine learning models in real time for multiple events on our platform

Key system requirements

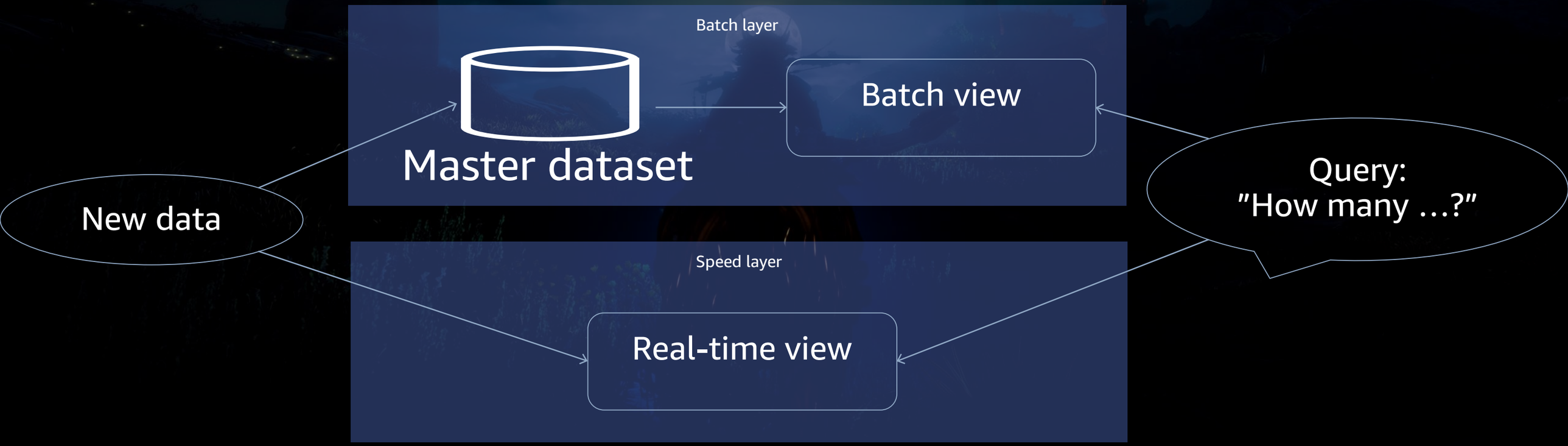
- ***Speed***—We want to allow our good customers to interact with our platform as quickly as possible
  - I want my new bunny suit now!
- ***Scale***—We want to be able to grow the system as traffic patterns change and new events are added
  - “Certain games” have very spiky traffic when they release new seasons. Infrastructure needs to be able to handle these scenarios.
- ***Resilience***—We want a system that is fault tolerant for both human and machine errors
  - Need to let Tom write code even though he admits to writing the occasional bug



Real-time analytics—lambda architecture

Lambda architecture (not to be confused with AWS Lambda)

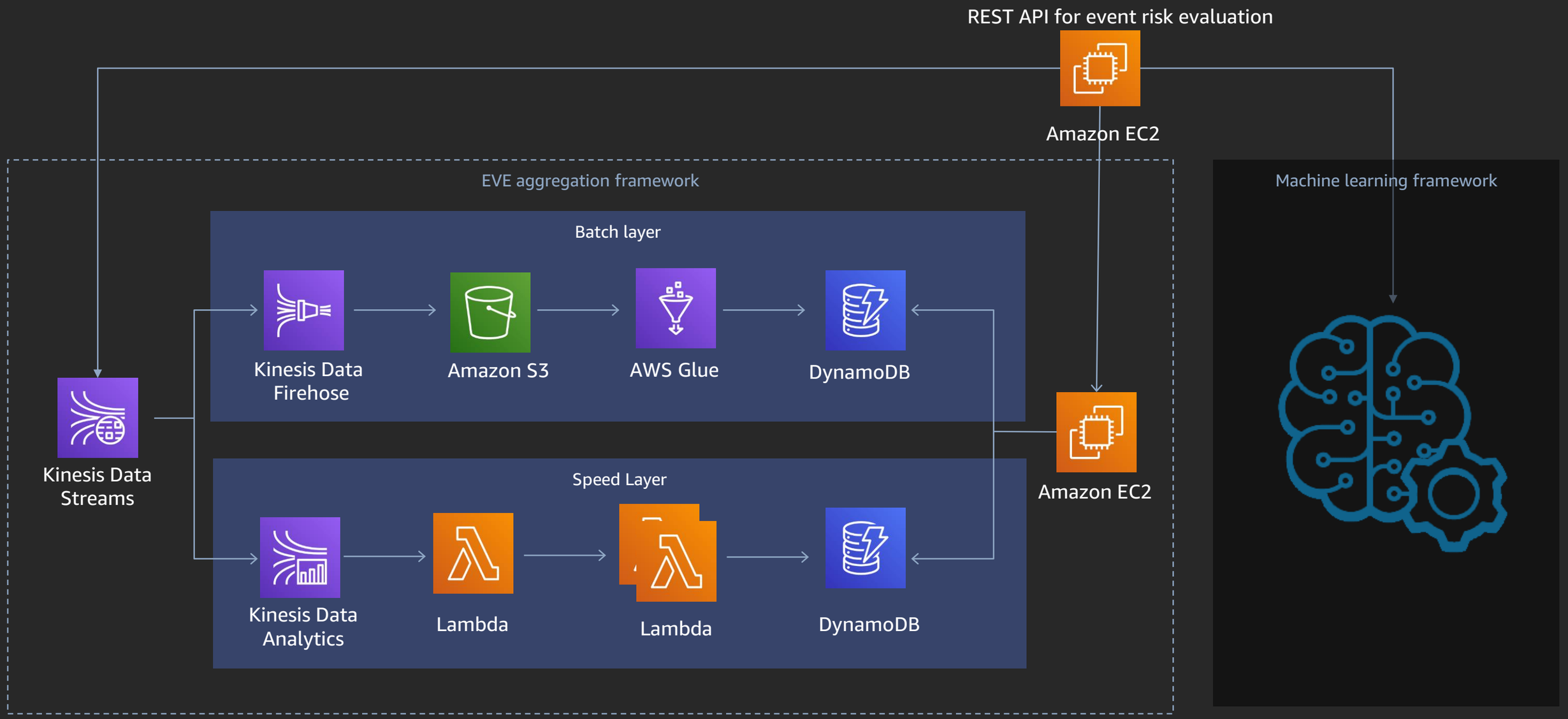
- Best practice for a data processing architecture
- System that uses batch processing to provide comprehensive and accurate views, while using real-time stream processing to provide results more quickly



Lambda architecture benefits

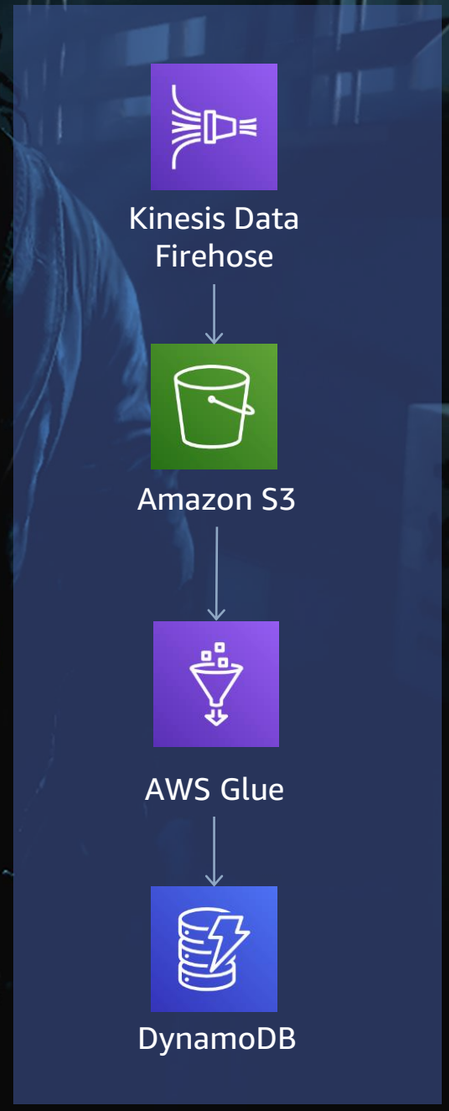
- Complexity isolation
  - Complexity of incremental updates is isolated to the speed layer, whose results are only temporary
  - Once the data is persisted into the batch views from the master dataset, real-time views are disregarded
- Eventual accuracy
  - Some algorithms are difficult to compute incrementally (e.g., unique counts)
  - Use the exact algorithm on the batch layer and an approximate algorithm on the speed layer
- Fault-tolerance (both machine and human)
  - If anything ever drastically goes wrong, regenerate the batch views and discard the entire real-time views and everything will be back to normal

AWS services and the lambda architecture



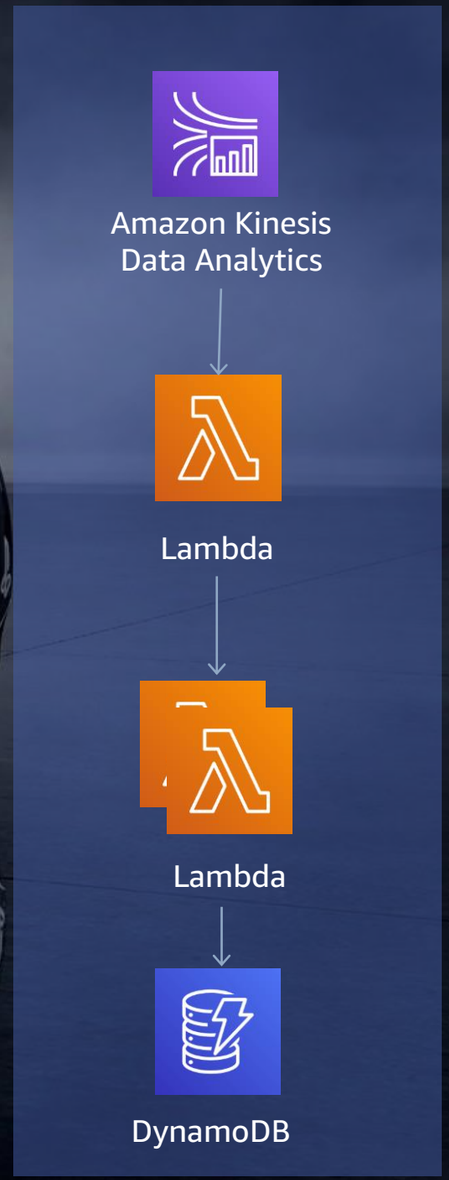
Batch layer

- Kinesis Data Firehose
  - Converts data from JSON to Parquet to optimize for ETL using AWS Glue table schema and delivers to Amazon S3
- Amazon S3
  - The master dataset of the lambda architecture, immutable append-only event log
- AWS Glue
  - Reusable Spark job that can run hourly, daily, and on-demand to generate (and potentially regenerate) the aggregated data from Amazon S3 to DynamoDB
- DynamoDB
  - Data written all at once in batch insert operations, no incremental updates



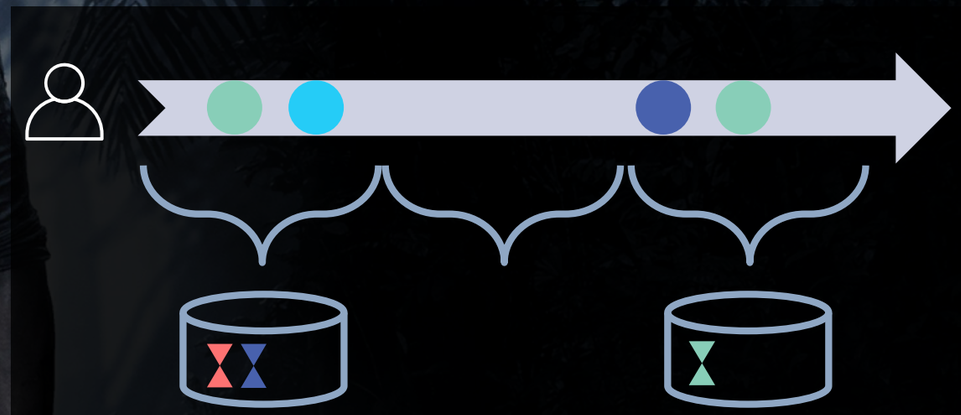
Speed layer

- Kinesis Data Analytics
  - Custom SQL aggregating across 1-second time intervals
- Lambda
  - Organizes the data by key for writes to the database
  - Fans out to multiple lambdas to parallelize writes to the database for better throughput at high traffic volumes
- DynamoDB
  - Data is written as updates, incrementally changing the aggregated values



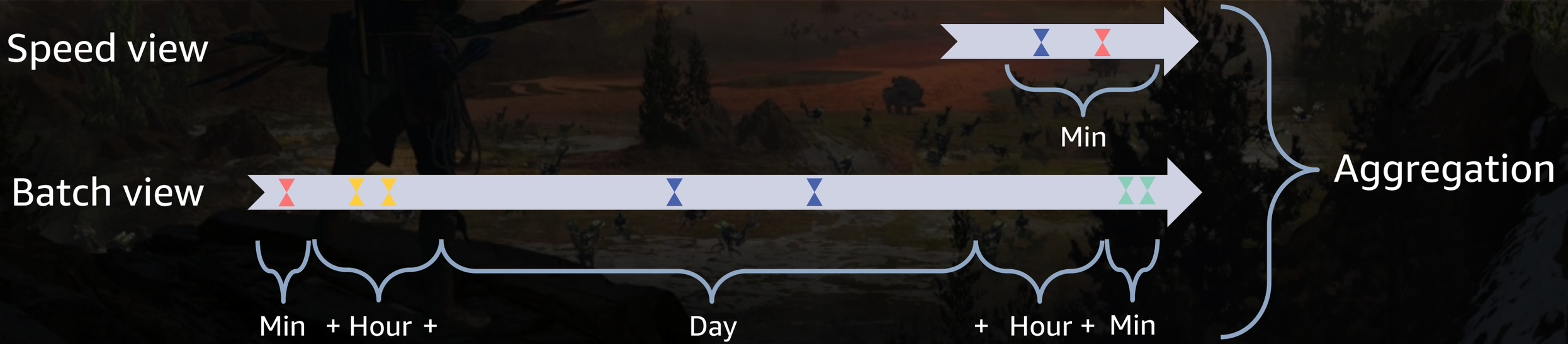
Technical challenge—optimize for reads

- Aggregation retrieval speed is key
  - The customer is waiting!
- Organize data into time “intervals”
  - Events occur in the system in real time
  - Day, hour, and minute intervals
- Keys
  - Hash key—interval type and what data is being aggregated
  - Sort key—the time interval
  - Examples
    - accountid_123XYZ_hour | 2019-11-02 21
    - ip_192.168.0.0_minute | 2019-11-02 21:05



Technical challenge—optimize for reads

- Query appropriate intervals given time is always moving to the right and the intervals to use are changing
- Query example: How many times has this account attempted to buy something in the last 30 days?
    - Minute intervals today (including speed view) and 30 days ago
        - Favor batch view interval if it exists
    - Hour intervals today and 30 days ago
    - Day intervals in the middle



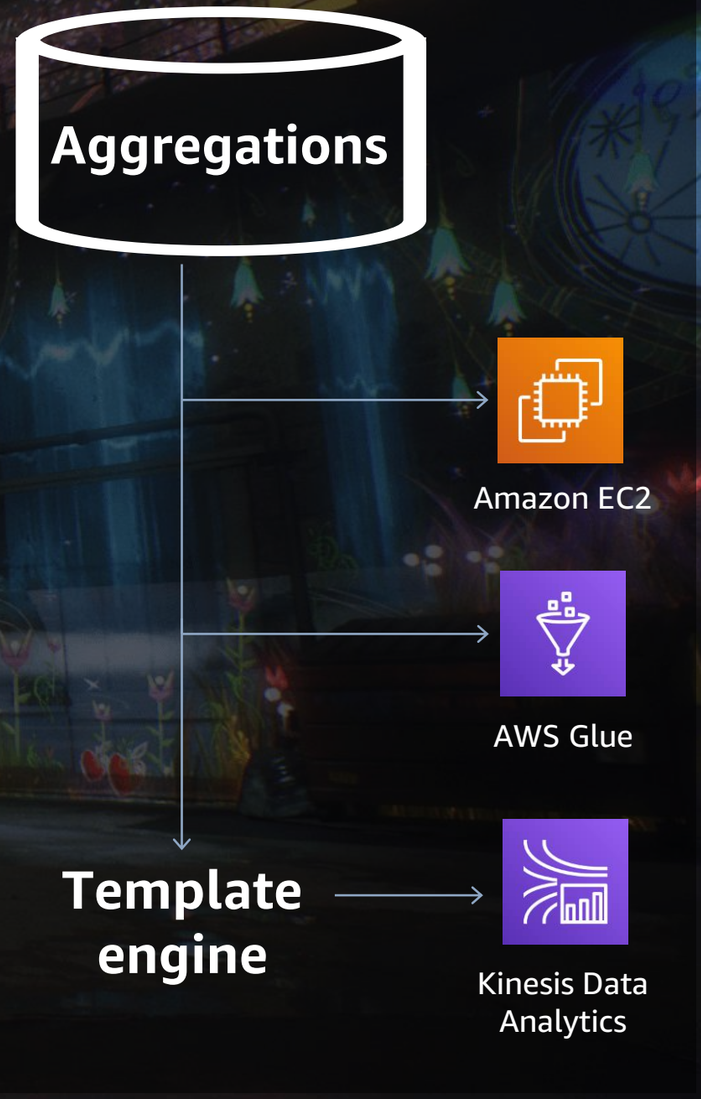
Technical challenge—extensibility

- Key component of our system is designing around a generic metamodel for an aggregation
  - Example
    - How many times has this account attempted to buy something in the last 7 days?
- Break it down into a repeatable structure
  - An operation (sum, count, unique count) on a key (account, device) and value (amount, IP) for an event (purchase, login) over time (1 day, 1 minute, 30 days)



AWS services and the metamodel

- Software has no knowledge of the aggregations, pulls from a configuration store
- Amazon EC2
  - What queries to perform against DynamoDB
- AWS Glue
  - What columns to operate on
  - What data to persist in DynamoDB
- Kinesis Data Analytics
  - Template engine to auto-generate SQL



Technical challenges—Big Data

- Millions of transactions a day, unique count aggregations can create large set sizes for longer time intervals
  - DynamoDB has item size limits, not recommended to use for very large datasets

- Introduced approximations to reduce size of items while maintaining a high enough accuracy for risk analysis
  - HyperLogLog+—Algorithm to approximate the number of distinct elements in a set

	Unique Data Point in Day	Size	Accuracy
Raw Set	{DataPoint1, DataPoint2, …, DataPoint5000}	Huge! >400KB item size limit	100%
Approximation	0100101010101010111010….	Manageable! ~20KB for a large set	~99%

Review of use case results

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use case 1—purchase

Overwhelming success!

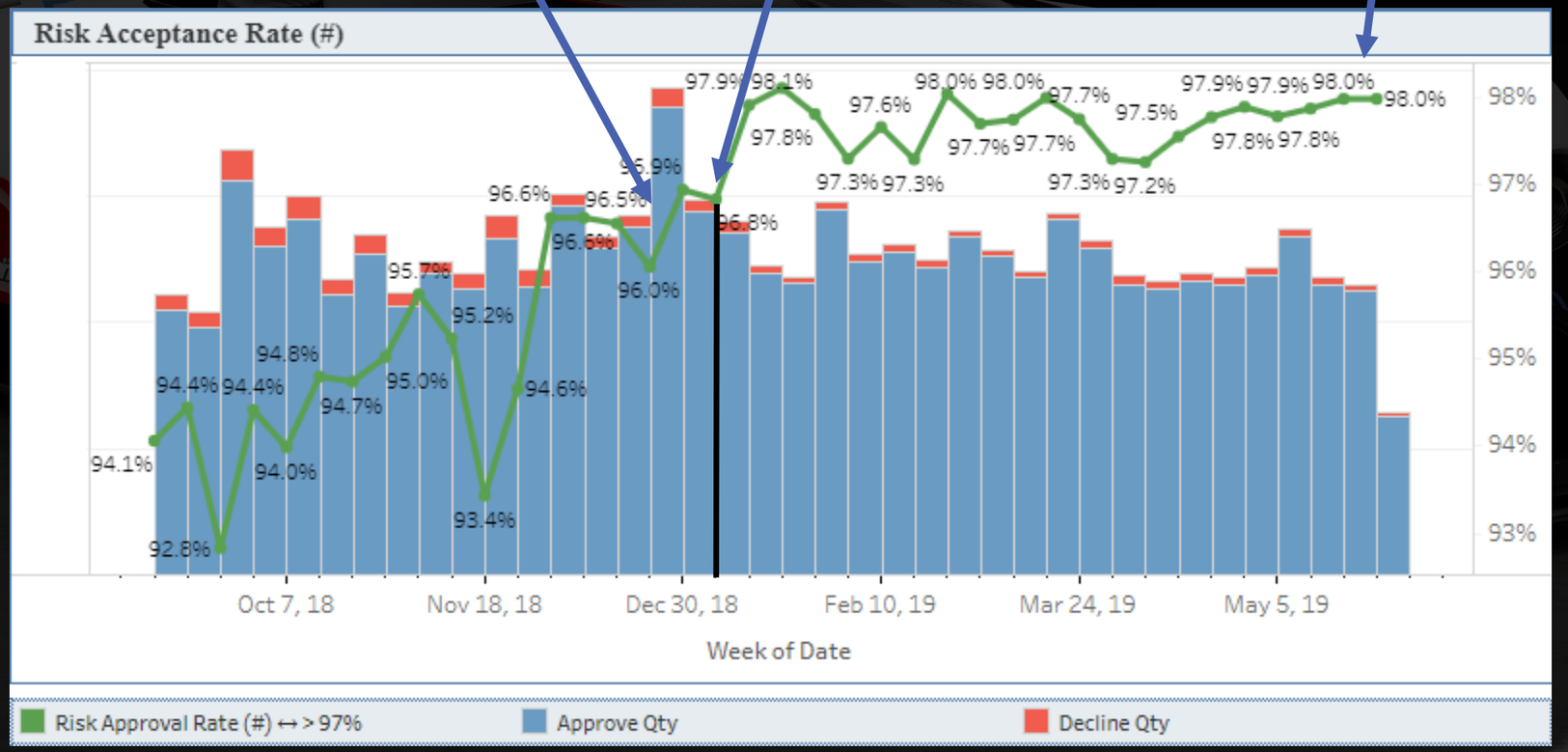
- We were able to improve approval rates by more than 3%, which translates to substantial monthly sales
- We were able to increase our approval rate while staying flat on our chargeback rate
- We reduced the response time from ~2 seconds to under ~195 ms
- We improved our risk decline inaccuracy rate by 40%



**Stats as of 9/29/2019

Use case 1—purchase results

EVE Purchase went live in the middle of January, with noticeable results

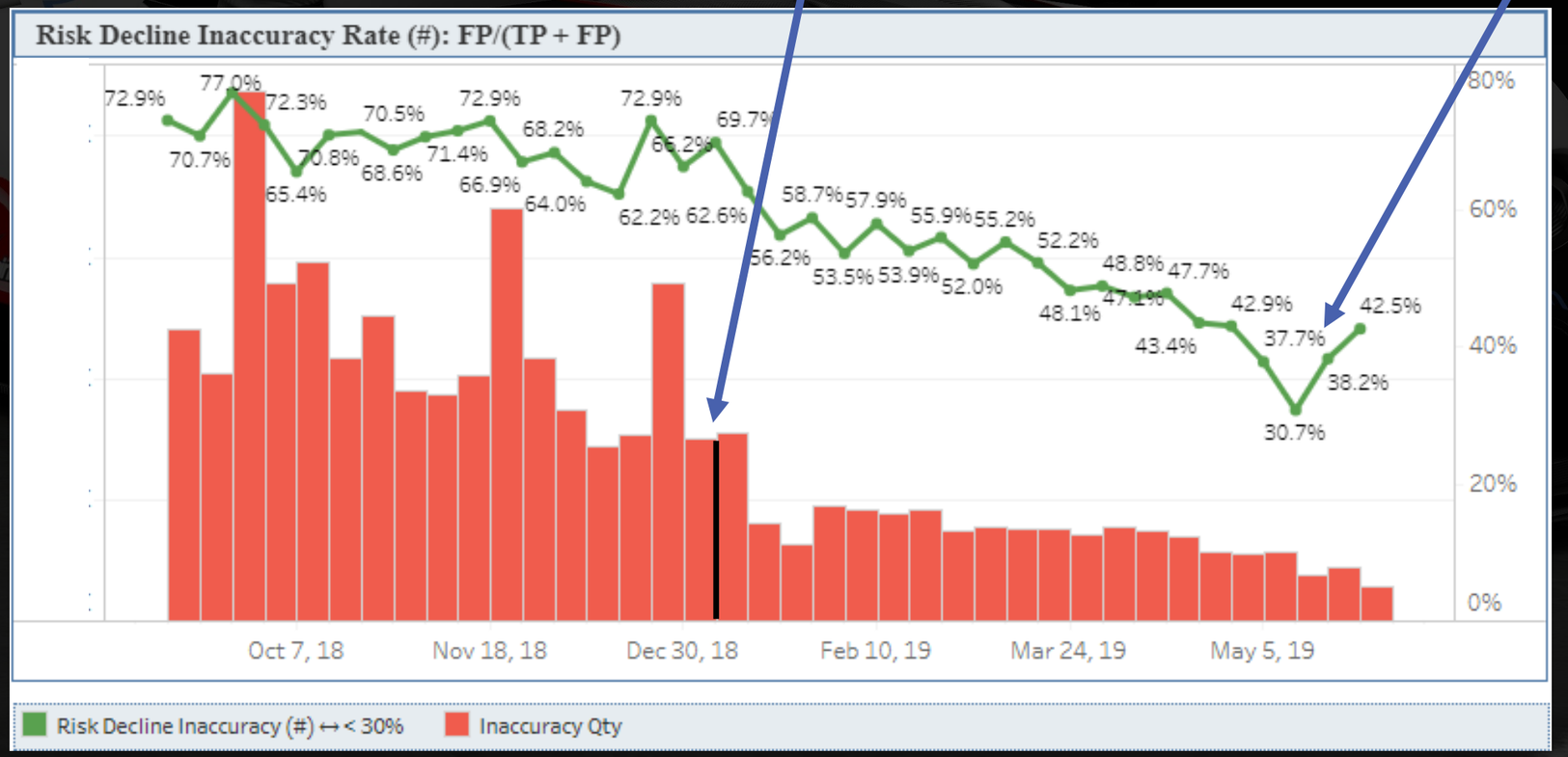


Use case 1—purchase results

EVE Purchase went live in the middle of January, and the results were a clear victory

EVE Purchase install providing a significant reduction in false positives

Additional performance gains as we tune the system



Use case 2—login

So far, we are bringing the fight!

- We have been able to proactively warn customers of suspicious activity
- Out of the warning emails we have sent, ~20% of the recipients have changed their password**
- Potentially ~15K customers have been protected from an ATO since launching in August



**Stats as of 9/29/2019

Lessons learned

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Challenges

- Accuracy vs cost
  - Organizing data into buckets leads to less than 100% accuracy as you operate within the current 1-minute time bucket
    - Shortening to 1-second buckets increases number of items and cost
- Infrastructure as code
  - AWS CloudFormation did not support all the required deployment automation
  - Created custom lambdas to configure resources after deployment
    - Examples—Starting the Kinesis Data Analytics application, configuring Kinesis Data Firehose with the appropriate AWS Glue schema
- Security
  - Sensitive platform data
  - IAM and encryption is very important
  - Needed to ensure least possible access to each resource via very detailed policies

What’s next

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Exciting things to come!

Expanding our risk management footprint

- Over the next 12-18 months we will be creating many more event interceptors, which will bring us closer to a real-time 360-degree risk management platform
- We will be exploring exciting technology using AWS managed services for our fraud management systems

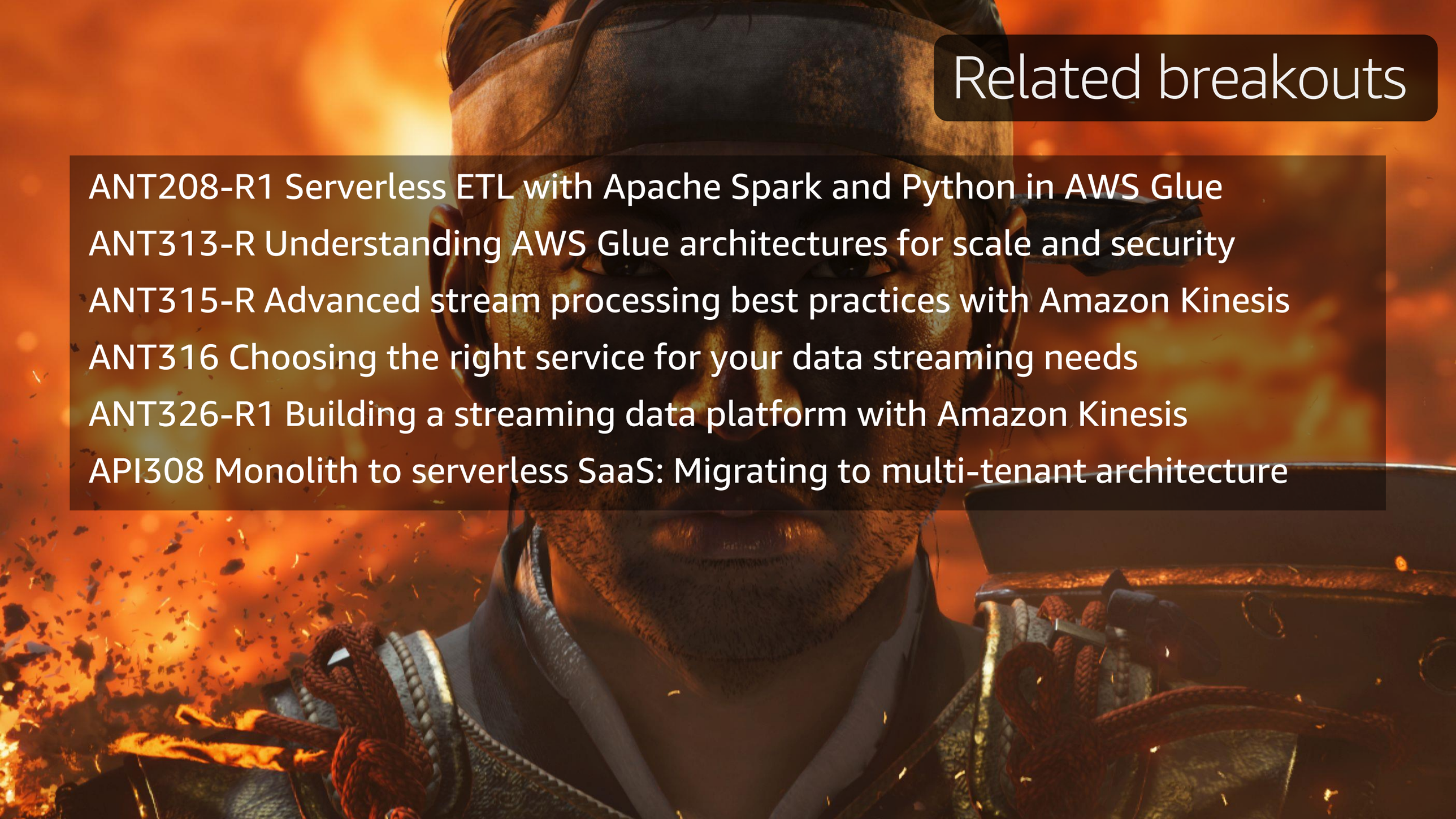
Growing our teams

- We will be expanding our team in our Bay Area and Southern California sites



Q&A

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Related breakouts

ANT208-R1 Serverless ETL with Apache Spark and Python in AWS Glue

ANT313-R Understanding AWS Glue architectures for scale and security

ANT315-R Advanced stream processing best practices with Amazon Kinesis

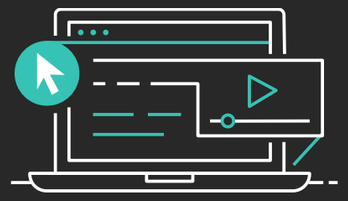
ANT316 Choosing the right service for your data streaming needs

ANT326-R1 Building a streaming data platform with Amazon Kinesis

API308 Monolith to serverless SaaS: Migrating to multi-tenant architecture

Learn big data with AWS Training and Certification

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



New free digital course, Data Analytics Fundamentals, introduces Amazon S3, Amazon Kinesis, Amazon EMR, AWS Glue, and Amazon Redshift



Classroom offerings, including Big Data on AWS, feature AWS expert instructors and hands-on labs



Validate expertise with the **AWS Certified Big Data - Specialty** exam or the new **AWS Certified Data Analytics - Specialty** beta exam

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

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws training and certification

Thank you!

Eric Krauss
eric.krauss@sony.com

Tom Lilley
thomas.Lilley@sony.com

Nicholas Walsh
nmwalsh@amazon.com

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws



Please complete the session survey in the mobile app.

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.