
AWS
re:Invent
DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

NTA302

How Abrigo transformed its business with AWS: Lessons learned

Jason Perlewitz
Vice President, Cloud Operations
Abrigo

Phil Schoon
Sr. Software Architect
Abrigo

Nayan Karumuri
Sr. Solutions Architect
AWS

- Overview of Abrigo’s business
- Understanding Abrigo’s environment
- Abrigo’s cloud journey in AWS
  - Migration
  - Optimization
  - Modernization

Overview of Abrigo’s business

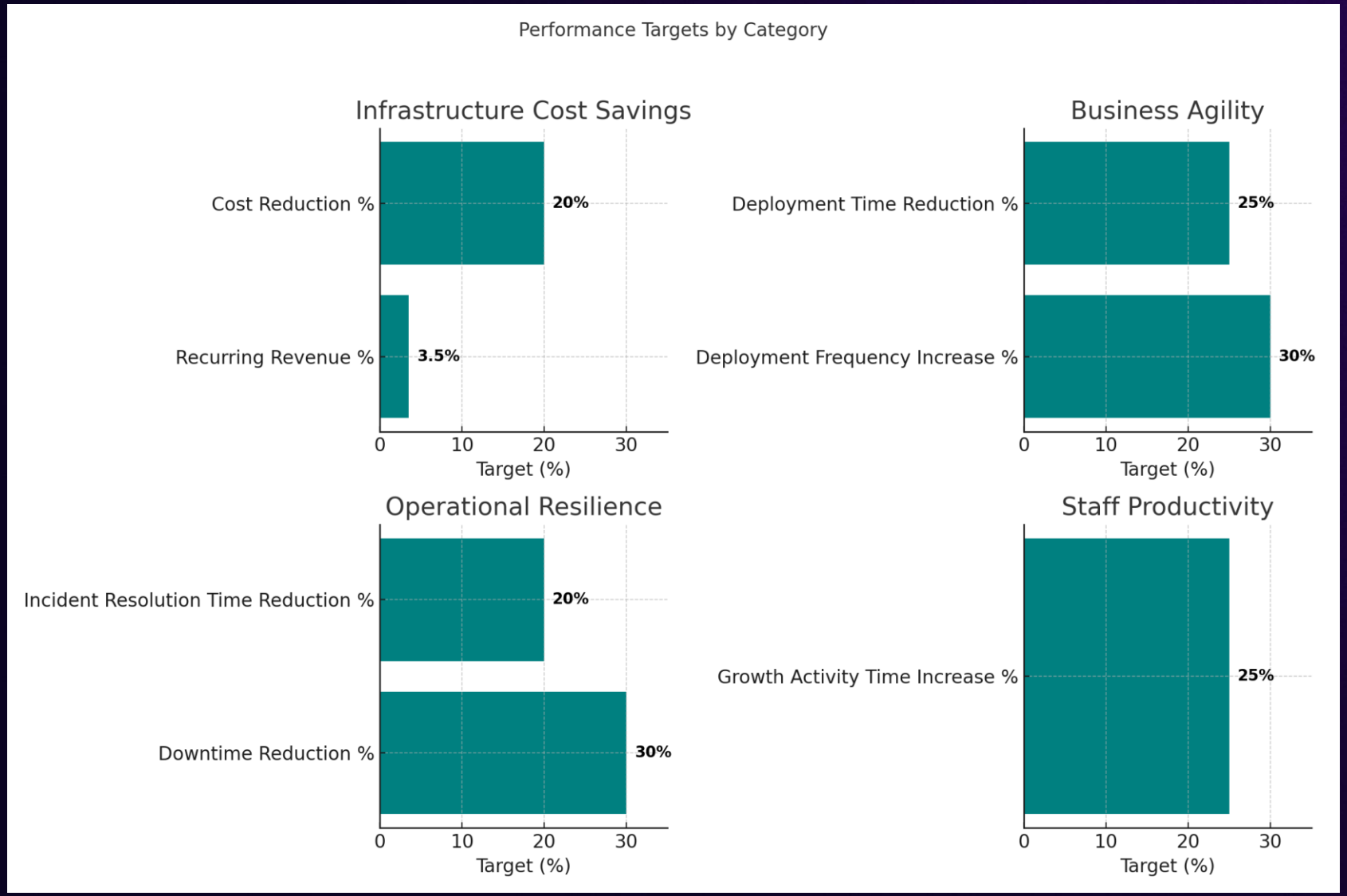
- Abrigo provides market-leading compliance, credit risk, and lending software solutions to their customers to fight financial crime, prevent fraud, process loans more quickly, and leverage data to strengthen their portfolios
- Leading provider of banking software and consulting to 2,400+ community financial institutions
- Serving community banking and credit unions for more than 20 years
- Built through various means – merger and acquisition, in-house development, and external partnerships

Abrigo’s pre-cloud hosting environment

- All SaaS servers hosted out of two geographically diverse collocated data centers – one as primary, one for disaster recovery and internal development
- Infrastructure: ~1,500 virtual servers, 5 PB of storage

  90% of the infrastructure built on the Microsoft stack: (Windows 2016–2019, MS SQL Server Standard and Enterprise, IIS App Server, .NET Framework and .NET Core)
- $7.5M/year in capital costs, hardware, and subscriptions
- 50+ unique hosted SaaS applications

Target outcomes in the cloud



Migration

Abrigo’s migration journey – The drivers

- Create momentum to accelerate migration to AWS without any disruption of services to Abrigo’s customers
- Reduce manual work and create automation to support the migration process
- Create a foundation for infrastructure, product, and database modernization to advance the business and drive down costs
- Equip teams to operate in the cloud, improve internal capabilities, and improve security posture
- Institute IaC-only change management and DevSecOps best practices
- Agility to innovate and adopt new technologies, including AI more quickly

Abrigo’s migration journey – The challenges

- 50 unique applications with non-standard operational patterns
- Migration deadline of less than 16 months
- Historical lack of hygiene and large volumes of data
- Tech debt
- Strict downtime limitations to minimize customer impact

Abrigo’s migration journey – The solutions

- AWS Professional Services to assist in building fit-for-purpose landing zones and security architecture that meet business needs and follow best practices
- Investment in training to upskill Abrigo teams
- Focus on curating “culture of the future”
- “Lift and shine” approach

  Copy existing VMs using AWS migration manager

  Make value-add opportunistic small changes for large benefit

  Eliminating unneeded environments and data
- Bi-directional storage layer synchronization
- Common engineering patterns for databases and attached storage for reuse
- AWS Managed Services (AMS) for initial augmented operational support

Migration techniques

AWS Application Discovery Service

AWS Migration Hub

Leverage AWS Cloud Migration Factory (CMF) solution with application migration services to simplify Abrigo's migration agent deployment, migration wave planning, and tag management at scale

AWS Server Migration Service (AWS SMS)

AWS Application Migration Service

Migration Evaluator

Real-time replication for on-premises applications to AWS, reducing cutover times for Abrigo's most sensitive workloads

Gave Abrigo detailed cost and usage insights, helping to **right-size AWS resources**

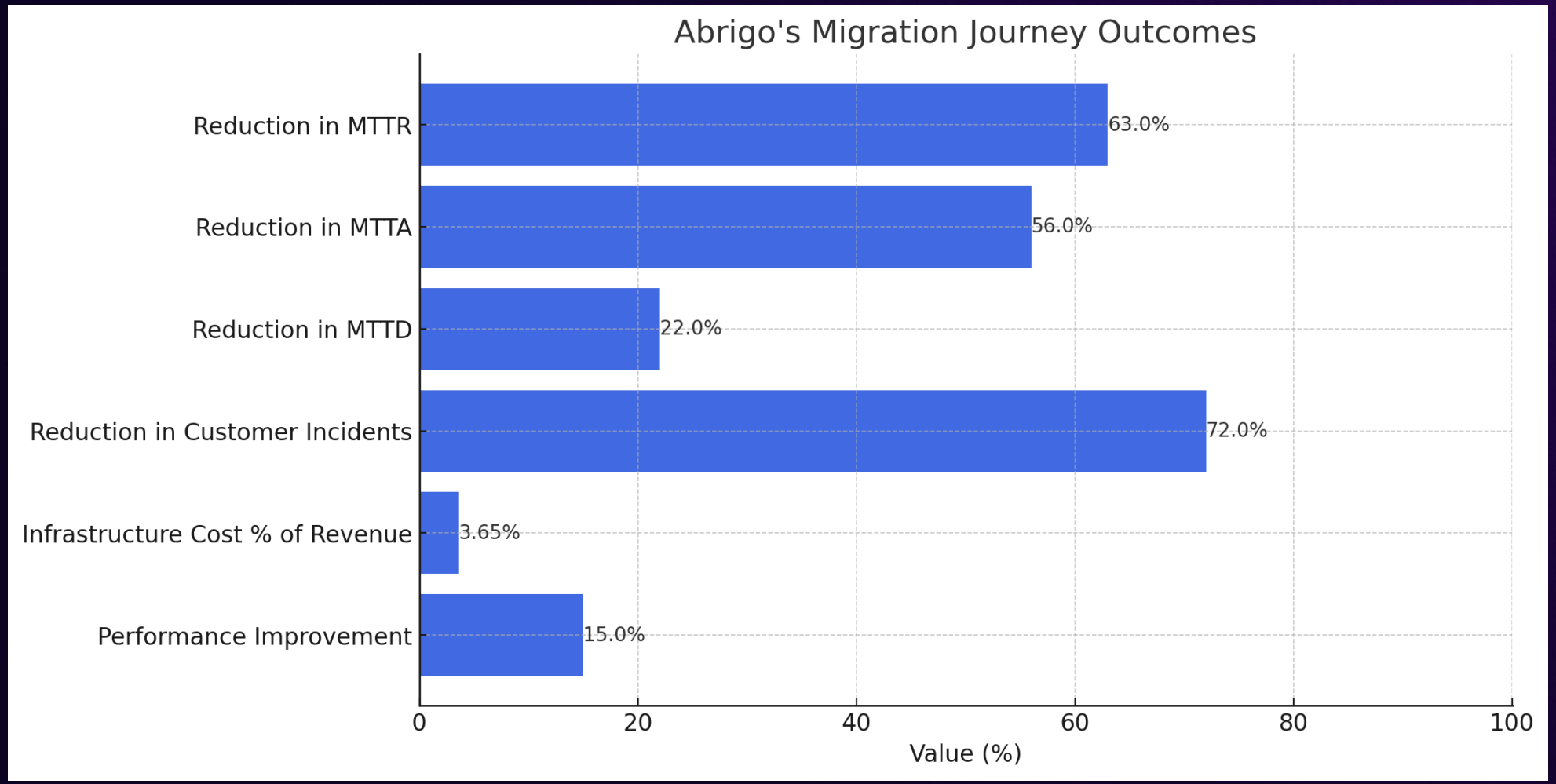
AWS Elastic Disaster Recovery (AWS DRS)

AWS Training and Certification

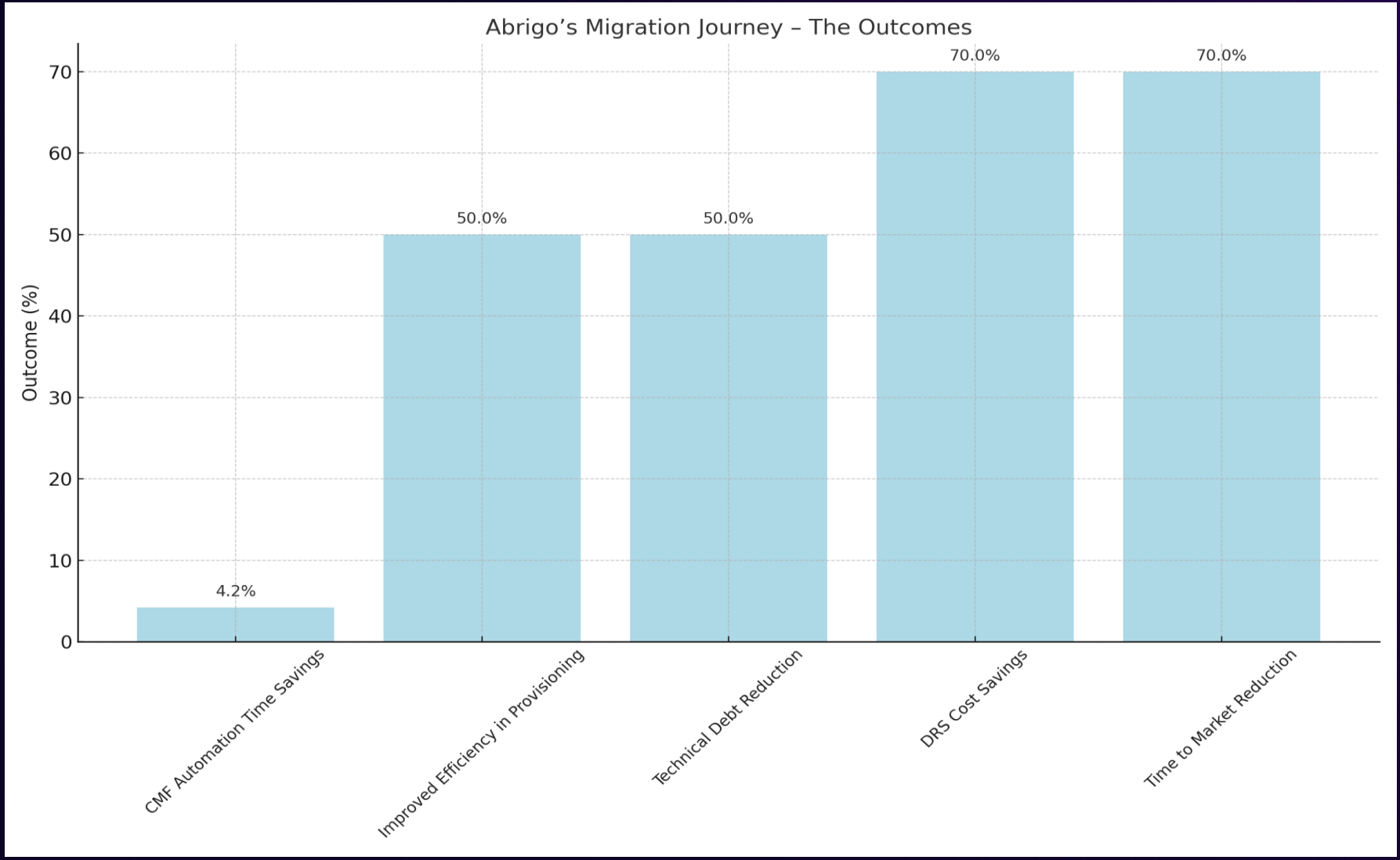
Leverage AWS DRS to simplify Abrigo's disaster recovery strategy at a lower cost

Engage AWS Learning Needs Analysis team to upskill with cloud essential trainings

Abrigo’s migration journey – The outcomes



Abrigo’s migration journey – The outcomes



Migration tips and tricks

LEARNED FROM THE SCHOOL OF HARD KNOCKS

Pre-Cutover

- There is no substitute for planning and preparation
  - "An ounce of prevention is worth a pound of cure." – Benjamin Franklin
- Implement the learnings from pain points in non-production environments before proceeding up stack
- Develop a standardized tagging strategy up front

Cutover

- Do a dry run of production cutover
- Have a backout plan during cutover and after cutover
- Give yourself a buffer of time in cutover windows
- Be in a room together during cutover (cross-functional)

Post-Cutover: Hypercare period

- Implement a FinOps practice on Day 1 to monitor costs
- Understand your Savings Plans (SP) and Reserved Instances (RIs) early

Optimization

Optimization – The challenges

- Initial bubble costs and inefficient spend after migration
- Cultural resistance to change within the organization
- Performance tuning and right-sizing based on real usage data (we were often over-provisioned, but it could take a couple months to hit busy cyclical usage patterns to verify changes)

Optimization – The solutions

Goal is to get the most efficient operating profile without major architectural changes that require large time and resource investment

- Right-sizing compute
- Optimizing cost-advantaged compute resources (SP and RI)
- Storage optimization and data purging
- Decommissioning systems no longer in use
- Scheduling systems to turn off in low-volume times and turn on
- Preferential movement to managed systems/services (third-party SSH/FTPS to AWS Transfer Family)
- Re-platforming (Red Hat Enterprise Linux (RHEL) → Amazon Linux, .NET applications on Windows → Linux)
- Limited transition to employ auto scaling where there is low barrier to entry

Optimization techniques – Compute

R6g instance

C6g instance

Migrated legacy M4/C4/R4 instances to C6g/R6g (AWS Graviton2) instances for better price-to-performance ratios

AWS Graviton2 instances provided a 40% improvement in performance per dollar compared to x86-based instances

C6a instance

R6i instance

R5a instance

C6a: Improved AI/ML performance and reduced latency for real-time analytics with better price-performance

R6i: Optimized memory throughput for fraud detection with 25% more I/O

R5a: Lowered costs by 15% for memory-intensive AI/ML and security workloads

Spot Instance

Stateless, fault-tolerant workloads

Reduced compute costs for batch processing and machine learning training in their analytics workloads

Optimization techniques – Compute

AWS Lambda

FinOps automation with AWS Lambda

Leveraged AWS Lambda functions triggered by Amazon CloudWatch Events to automatically shut down non-critical resources

Saving an additional $50K annually

AWS Compute Optimizer

Amazon EC2 Auto Scaling

Monitored compute utilization and provided right-sizing recommendations for Amazon EC2

Scaling policies using target tracking based on Amazon CloudWatch custom metrics

Auto Scaling groups with a combination of Spot and On-Demand Instances

Automated recommendations using AWS Systems Manager

Amazon Elastic Container Service (Amazon ECS)

AWS Fargate

Leveraged Amazon ECS with Fargate Spot for containerized applications to further optimize cost with serverless containers

Optimization techniques – Cost management

AWS Budgets

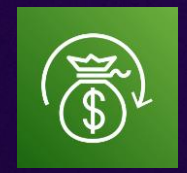
Deployed **AWS Budgets** and **AWS Cost Categories** for granular cost control

AWS Cost & Usage Report

AWS Cost Explorer

Integrated cost anomaly detection and daily cost insights for granular tracking

Reserved Instance (RI) Reporting



Savings Plans

Implemented Savings Plans (SP) with auto-renewal scripts for predictable workloads, locking in lower rates with utilization monitoring at 99.5%

Optimize cost efficiency by combining Reserved Instances with Auto Scaling

Optimization techniques – Storage management

Amazon Elastic Block Store (Amazon EBS)

Volume gp3

Amazon FSx for NetApp ONTAP

Shifted from **gp2 to gp3** volumes, reducing costs by 20% while maintaining throughput and IOPS

FSx ONTAP environment, Abrigo optimized the read/write IOPS profile, tailored caching mechanisms, and adjusted tiering policies

Amazon Simple Storage Service (Amazon S3)

S3 Intelligent-Tiering

S3 Glacier Flexible Retrieval (successor to S3 Glacier)

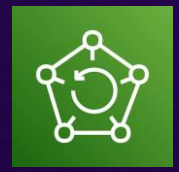
S3 Intelligent-Tiering reduced costs by up to 40% for infrequently accessed data

S3 Storage Lens identified unused data for migration to S3 Glacier

Optimized disaster recovery costs with S3 Standard-IA and Glacier Deep Archive

Amazon Aurora

Amazon DocumentDB



AWS Backup

Transitioning to **Amazon Aurora PostgreSQL**, reducing licensing costs by 40% and improving scalability

Replaced MongoDb databases with **Amazon DocumentDB**, achieving a reduction in management overhead

Streamlined backup management with **AWS Backup**, automating snapshot retention and reducing costs

Optimization techniques – Other

AWS Key Management Service (AWS KMS)

AWS Identity and Access Management (IAM)

Post-migration, Abrigo applied a **least privilege model** by auditing permissions using **AWS Identity and Access Management Access Analyzer**

AWS IAM Identity Center across all environments, reducing the administrative overhead for access management

Optimized encryption of data at rest across EBS, S3, and Amazon RDS using **AWS KMS** with custom key policies

AWS CloudFormation

Amazon CloudWatch

Template modularization

Parameter and output management

Automated resource scaling

CloudWatch Log Insights and anomaly detection

Optimization – The outcomes

$140K

Cost savings
Rightsizing EC2 Instances

30%

Processor generation
Upgrade performance uplift

$250K

Cost savings
FSx ONTAP & EBS Storage Optimization

85%

SPS and RI utilization

$60K

Cost savings
Automated power cycling

Modernization

Modernization – The challenges

- Tech debt
- Monolithic architectures
- Getting prioritization (trading off features for architecture)
- One-time cost to implement
- Longer times to realize value
- Articulating “customer value”

Modernization – The solutions

Adoption of AWS cloud-native services


AWS Lambda


Amazon Elastic Container Service (Amazon ECS)


AWS Transfer Family


Amazon Aurora


AWS Fargate


AWS Database Migration Service (AWS DMS)

Adoption of AWS supporting services for operational management


AWS Backup


Fleet Management


AWS Elastic Disaster Recovery (AWS DRS)

Modernization – The outcomes

$1.0M

Cost savings DRS
Reduced EC2 Footprint

60%

DRS faster RTO times

$1.3M

Database modernization
Aurora Postgres savings

15%

Database modernization
operational overhead reduction

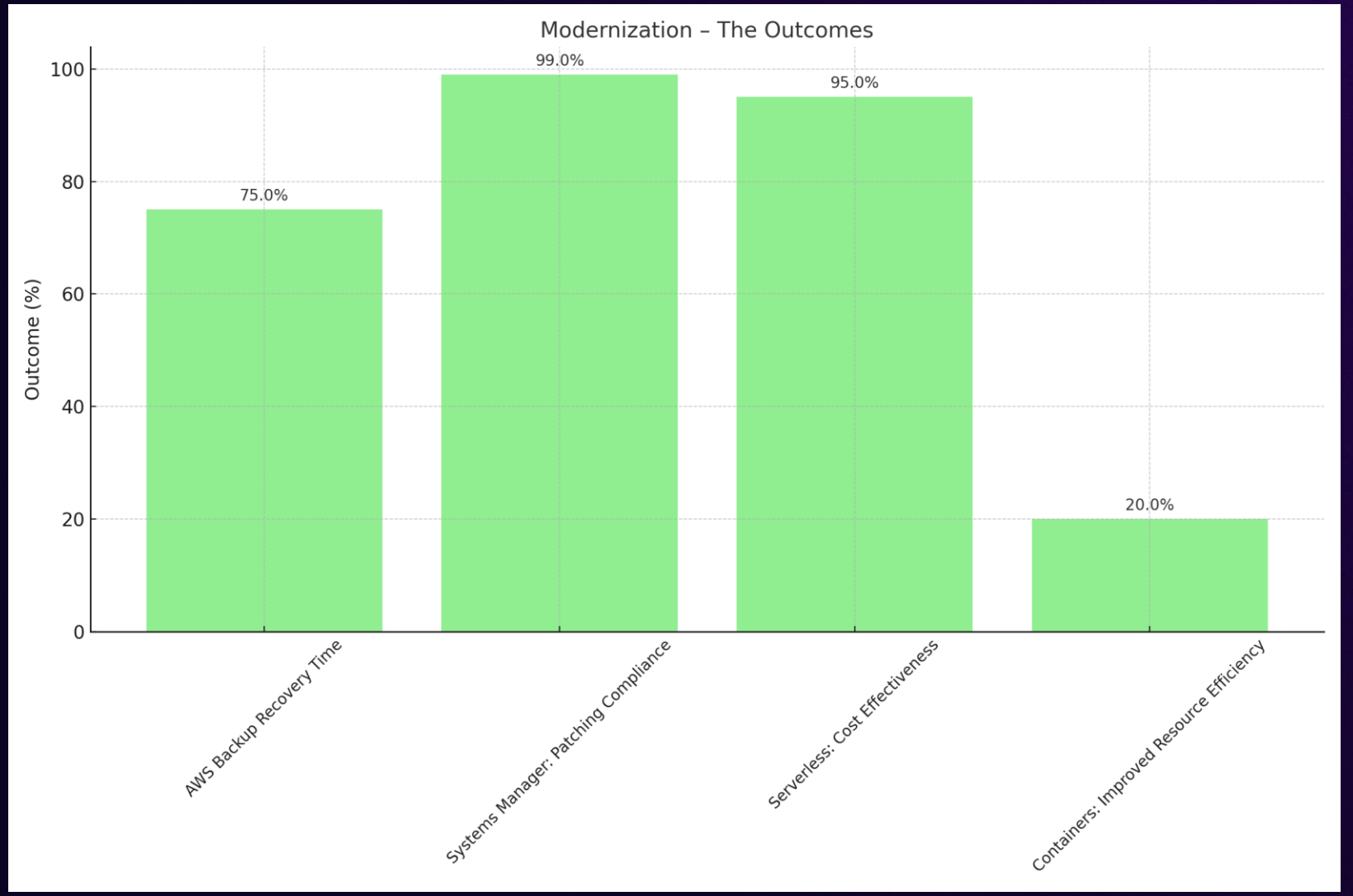
50%

Aurora Postgres
Performance improvement

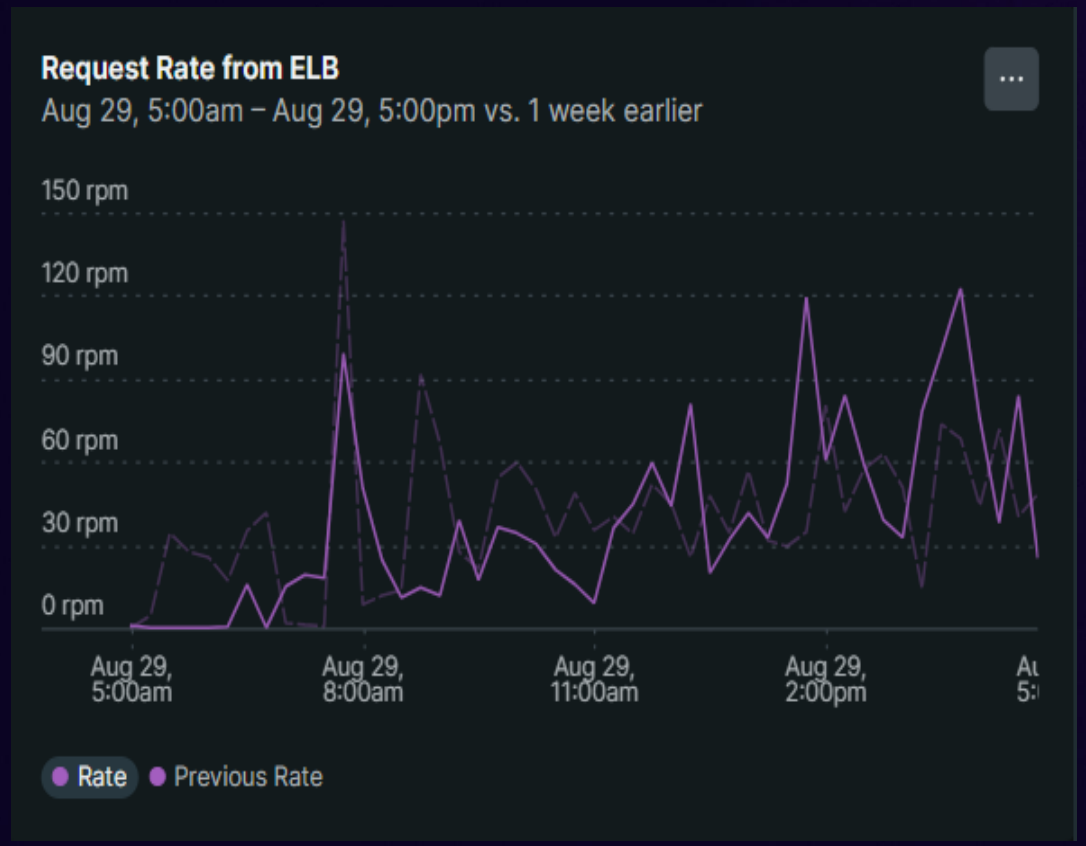
80%

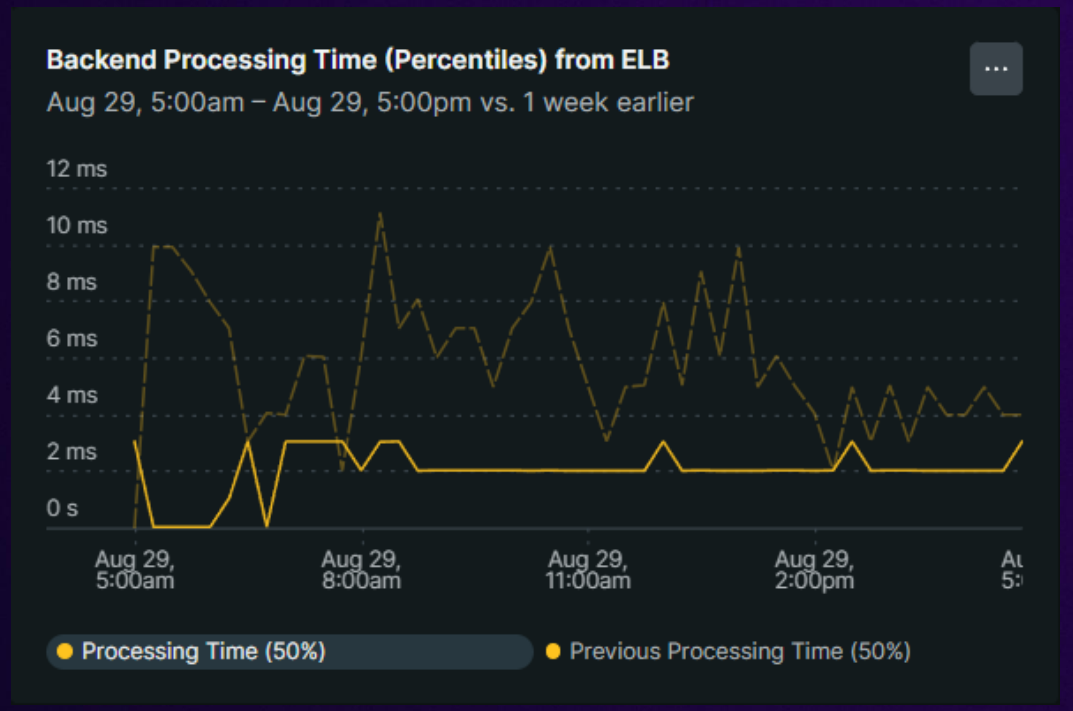
Babelfish development
Cost savings

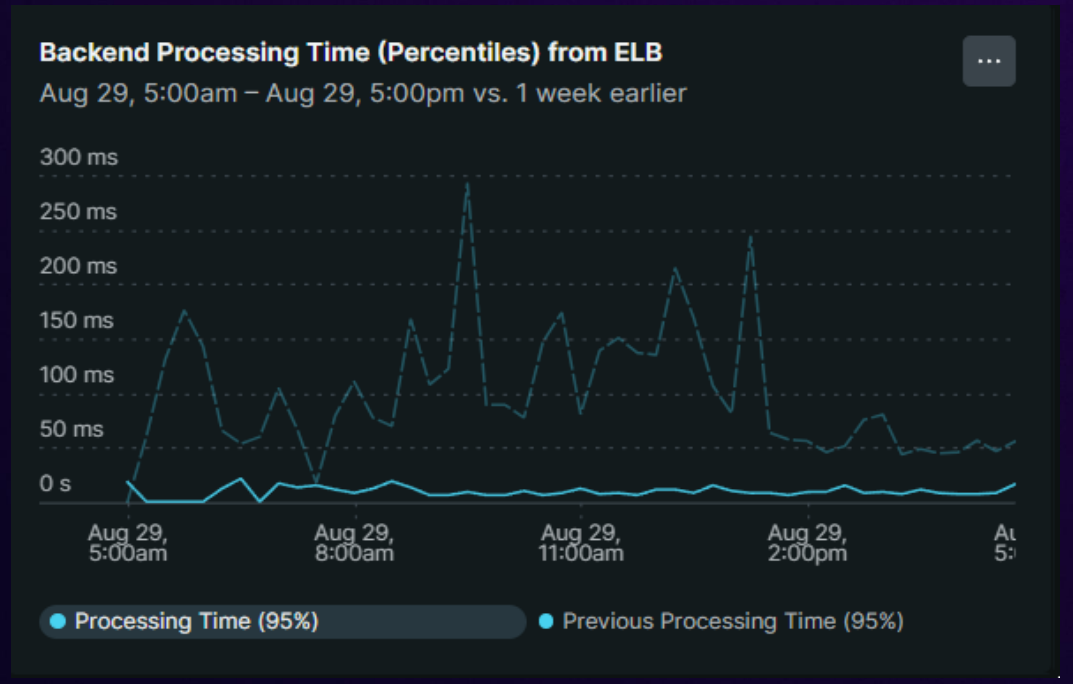
Modernization – The outcomes



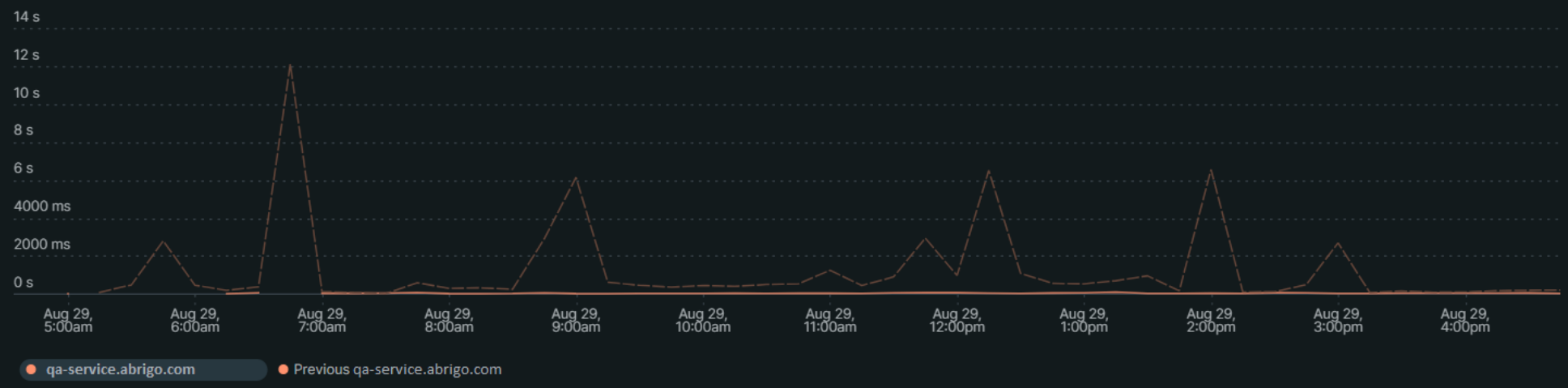
Container efficiencies



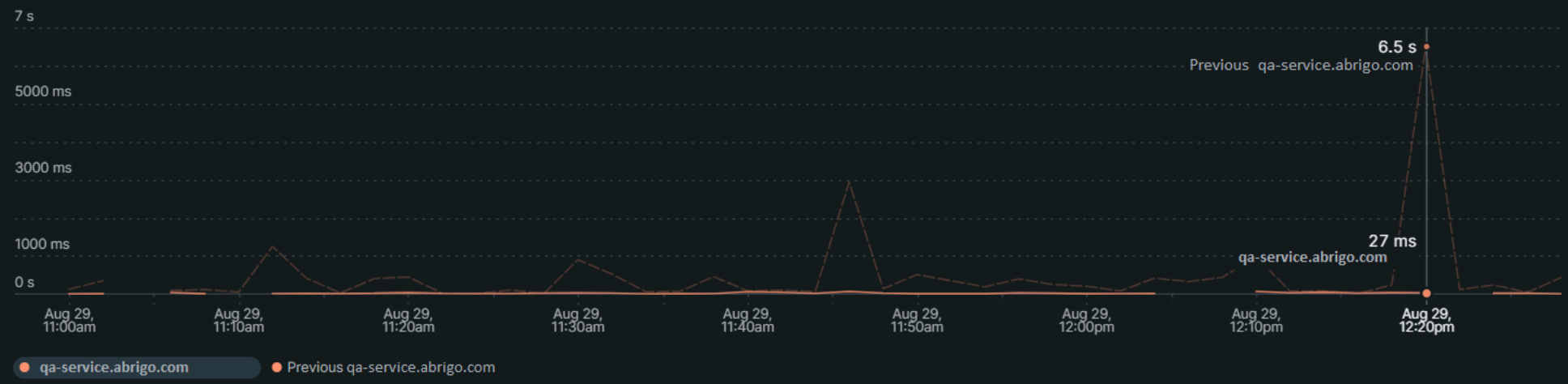




Max Backend Processing Time from ELB
Aug 29, 5:00am – Aug 29, 5:00pm vs. 1 week earlier

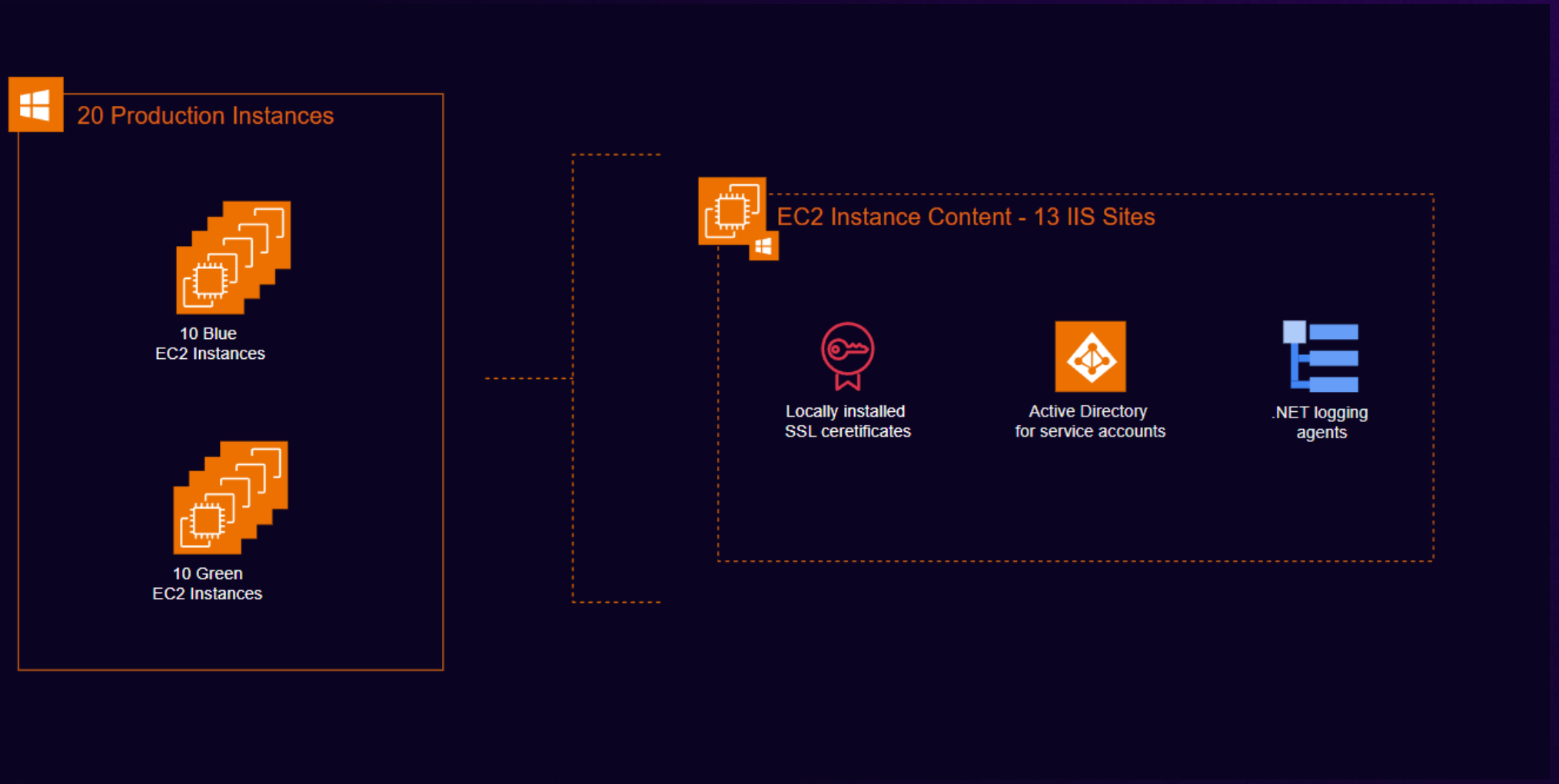


Max Backend Processing Time from ELB
Aug 29, 11:00am – Aug 29, 12:30pm vs. 1 week earlier



ASP.NET Windows IIS monolith

POST-AWS MIGRATION



Modernization to ECS Fargate (Goldilocks)

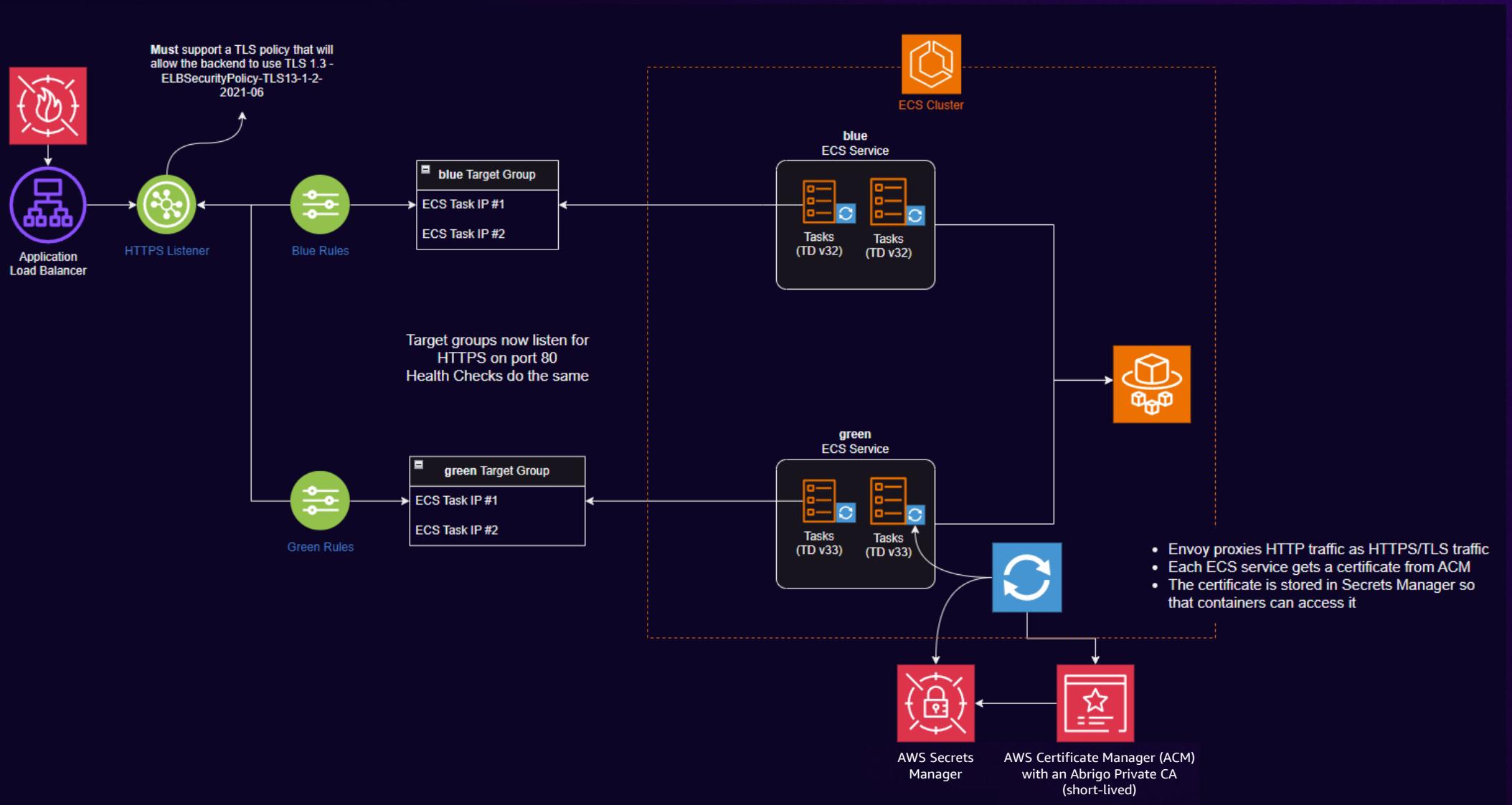
Benefits

- Autoscaling
- Zero-downtime code deployment
- No EC2 management

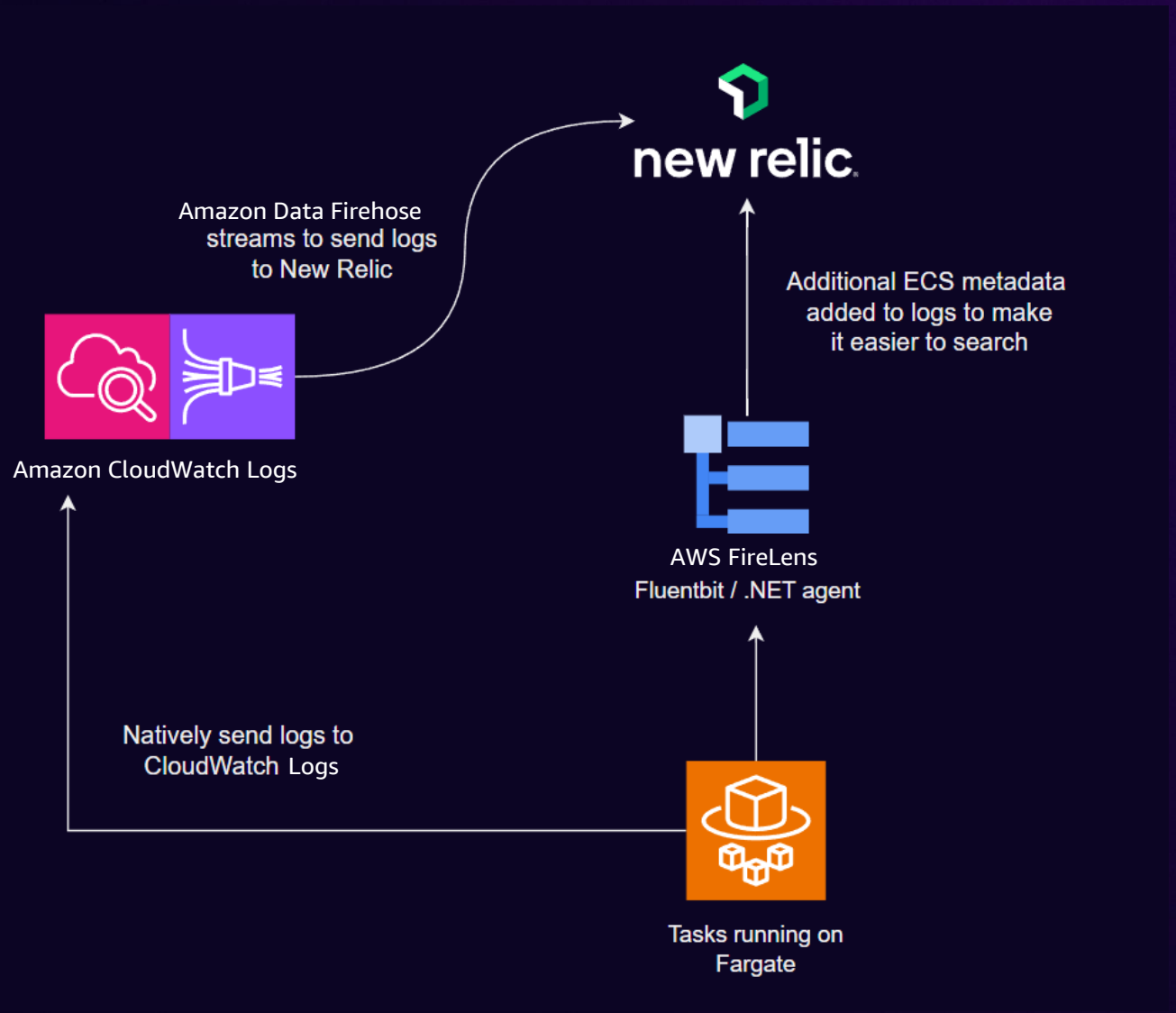
Challenges

- Certificate management
- Application logging
- Access to MS SQL via AD
- Deployment pipeline/releases

Certificate management

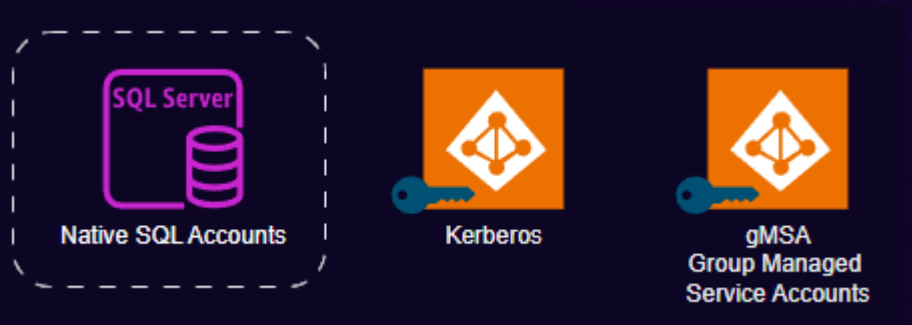


Application logging

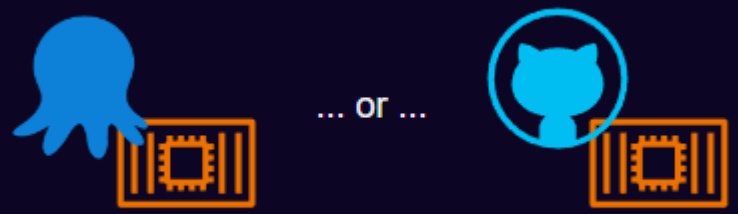


Modernization ECS Fargate solutions

Access to MS SQL via AD



Deployment pipeline/releases



Thank you!

Please complete the session survey in the mobile app

Jason Perlewitz
jason.perlewitz@abrigo.com

Phil Schoon
philip.schoon@abrigo.com

Nayan Karumuri
nayanpk@amazon.com