ADC-D2

McAfee Re-Imagined From RDBMS to NoSQL using Amazon DynamoDB

Kanniah V J

Senior Solution Architect
AWS

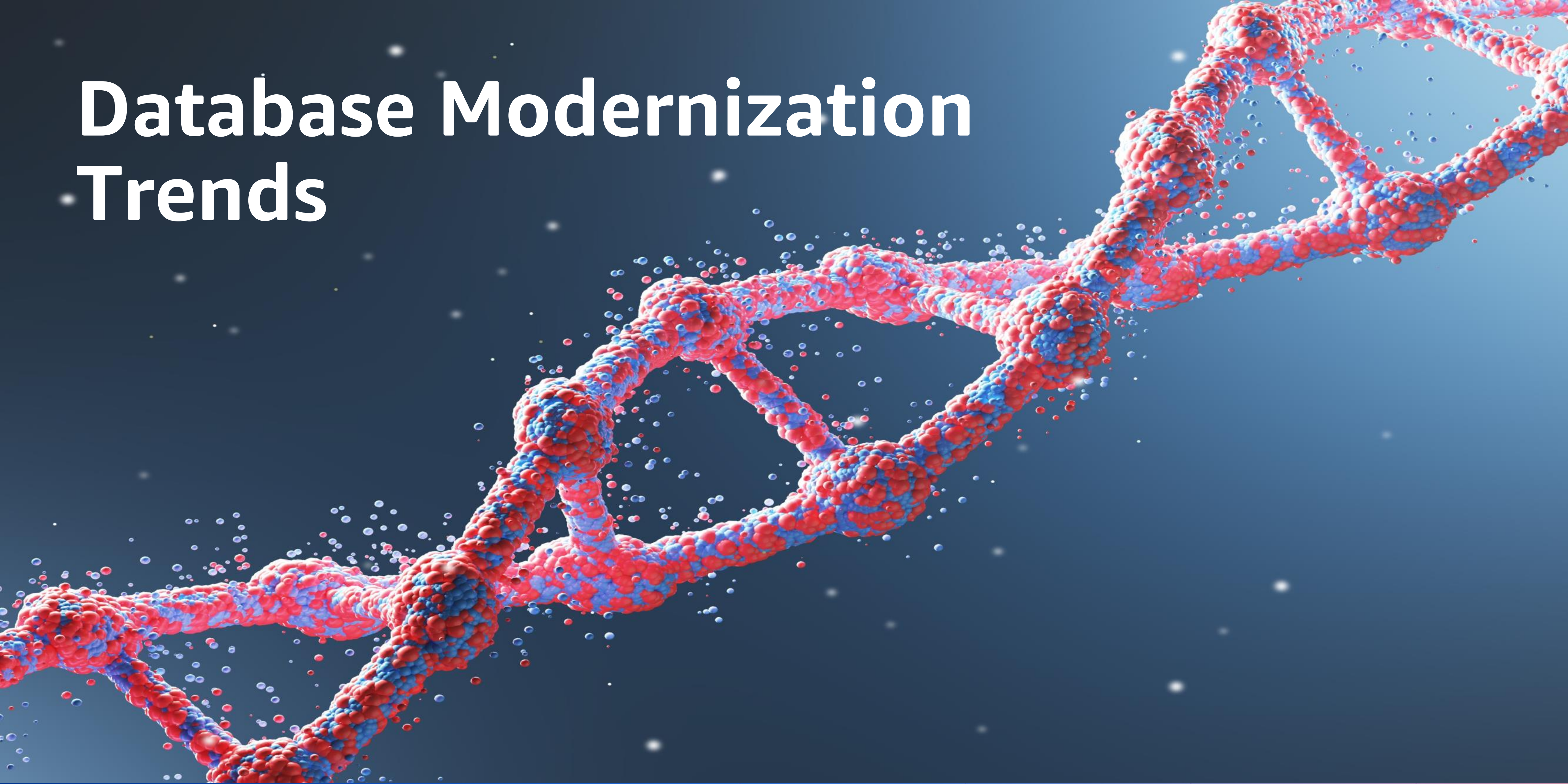
Dmitrii Gmyzin

Cloud Software Architect
McAfee

Agenda

- Database Modernization Trends
- Amazon DynamoDB at Glance
- Database Modernization Lifecycle
- NoSQL Data Modeling
- Productive NoSQL Tools
- Takeaways

Database Modernization Trends



ADC-D2

Three ways customers consume AWS database services

1

Purpose built for modern applications

Build new cloud-native applications by using the right tool for the right job

2

Move to managed services

Migrate legacy applications and use the same database engine in a managed cloud service

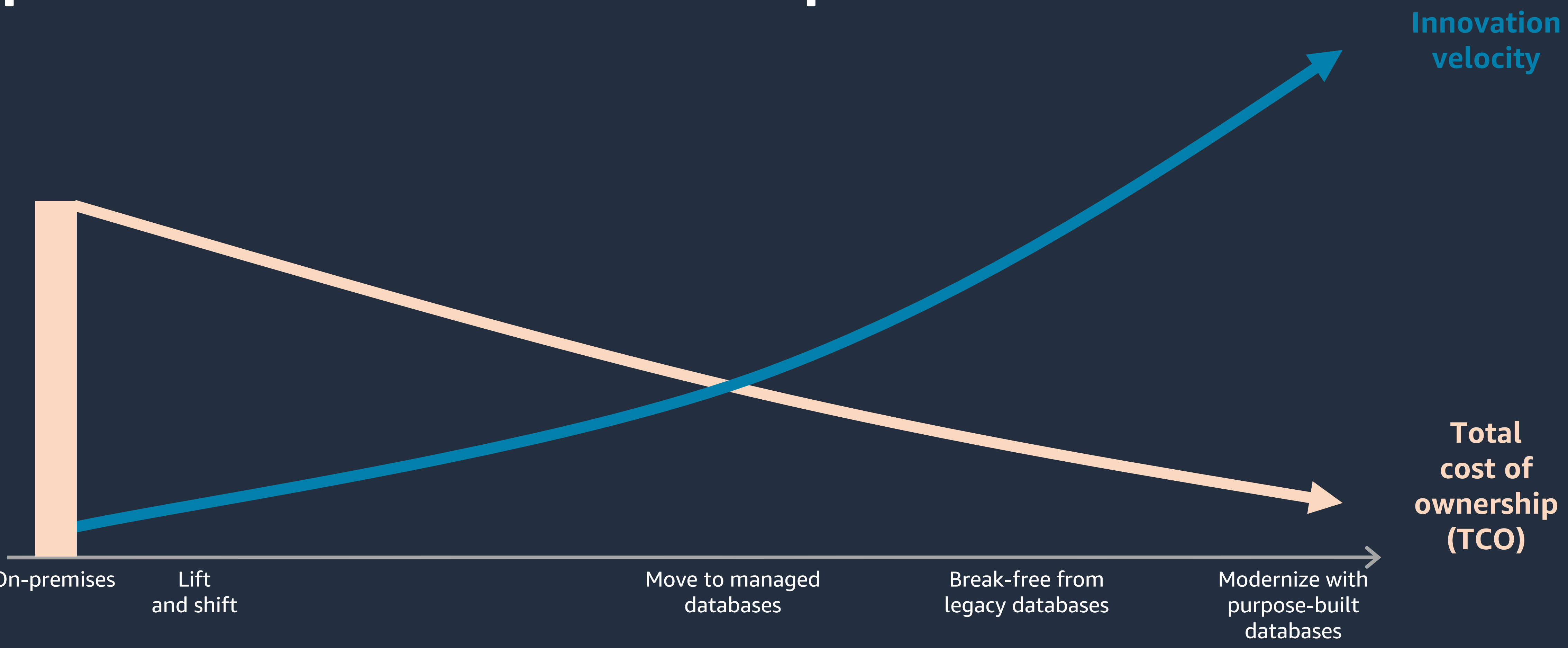
3

Break free from legacy databases

Migrate “old guard” commercial databases to open-source engines (for example, Oracle to DynamoDB)

ADC-D2

Modernizing leads to maximum innovation velocity and optimal total cost of ownership



ADC-D2

McAfee SampleDB

- SampleDB – Internal McAfee database
- Used for:
  - Metadata storage
  - Reporting
  - Business logic

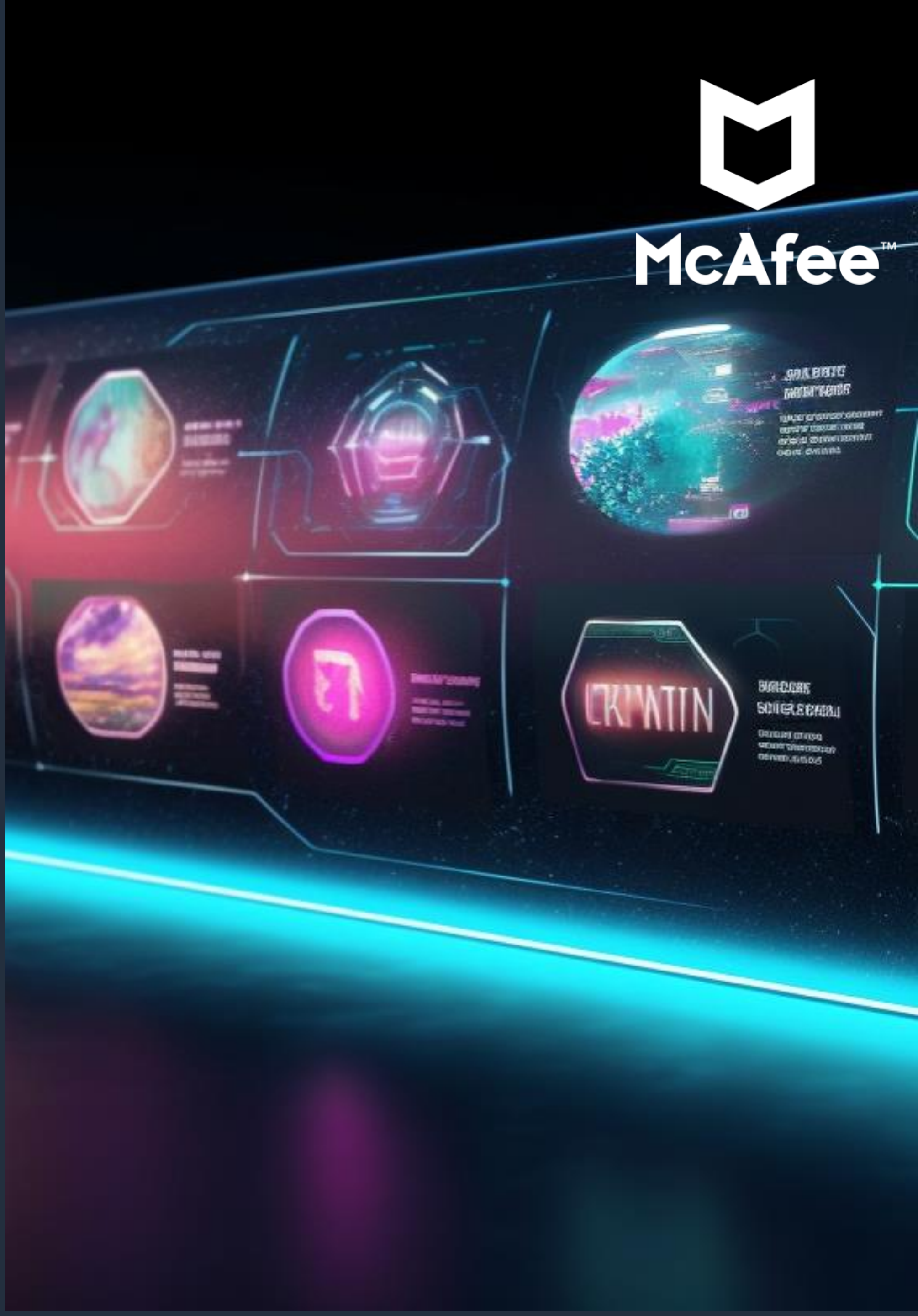
- Server size:
  - On-premise SQL Server (OLTP)
  - 3-region cluster
  - ~50 additional servers for varied roles
  - Over 80 TB of data
  - Tables with up to 150 billion records

ADC-D2

McAfee Modernization Approach

- **Why migrate and modernize?**
  - 15 years old legacy database, but high value data
  - Complex web of interconnected servers
  - Monolithic architecture with a lack of scalability

- **Why move from SQL server**
  - Be able to scale horizontally
  - Satisfy key-value access pattern



Amazon DynamoDB at Glance

ADC-D2

Amazon DynamoDB

Performance at scale

- Consistent, single-digit millisecond read and write performance
- Nearly unlimited throughput and storage
- Massive scalability

Secure and Resilient

- Data encryption at rest
- Global replication
- Up to 99.999% availability SLA

Serverless

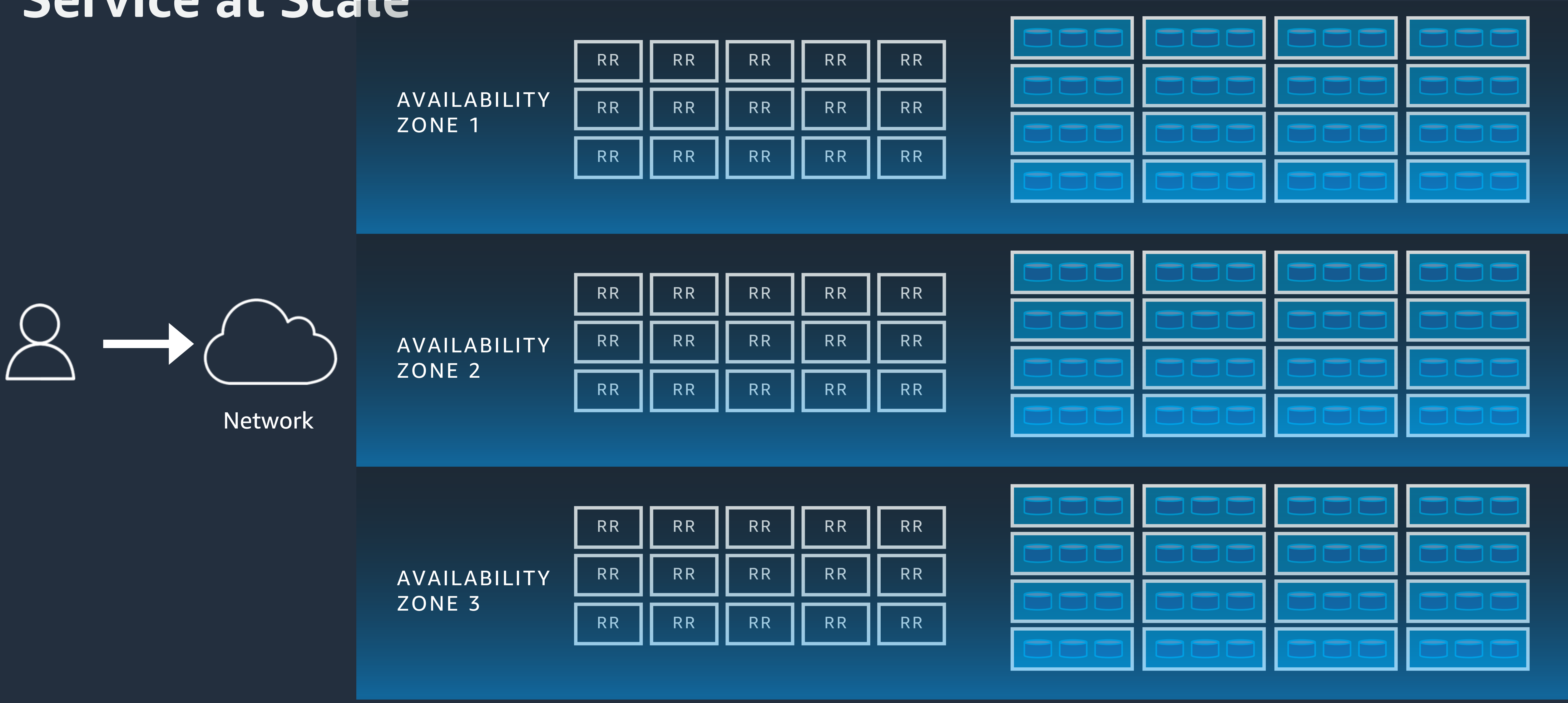
- Performance at scale with the ability to scale-up and scale-to-zero
- No downtime maintenance, no maintenance windows
- No provisioning or capacity management
- Pay-per-request billing

Built-in integration with others AWS services

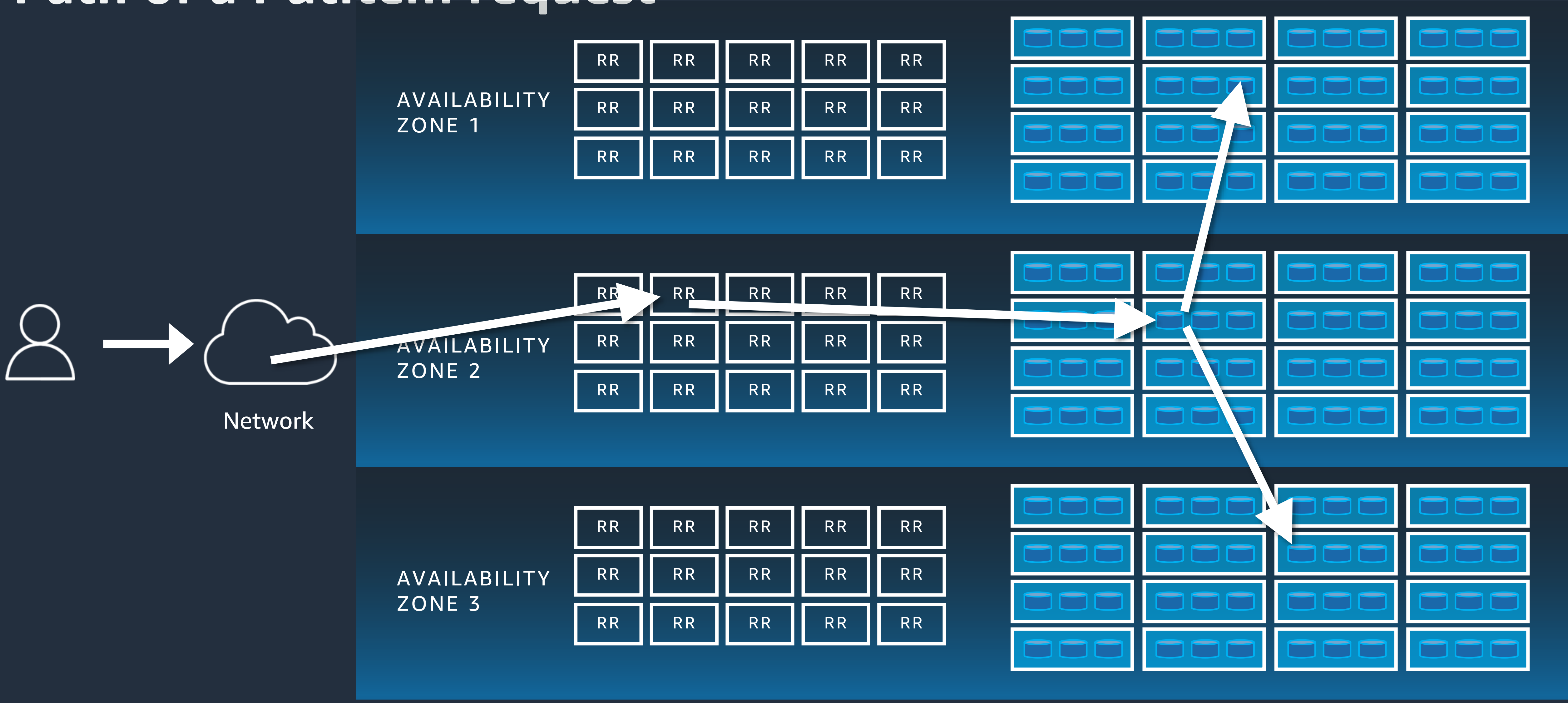
- Logging, monitoring, and analytics
- Import and export data to Amazon S3

ADC-D2

Service at Scale



Path of a PutItem request



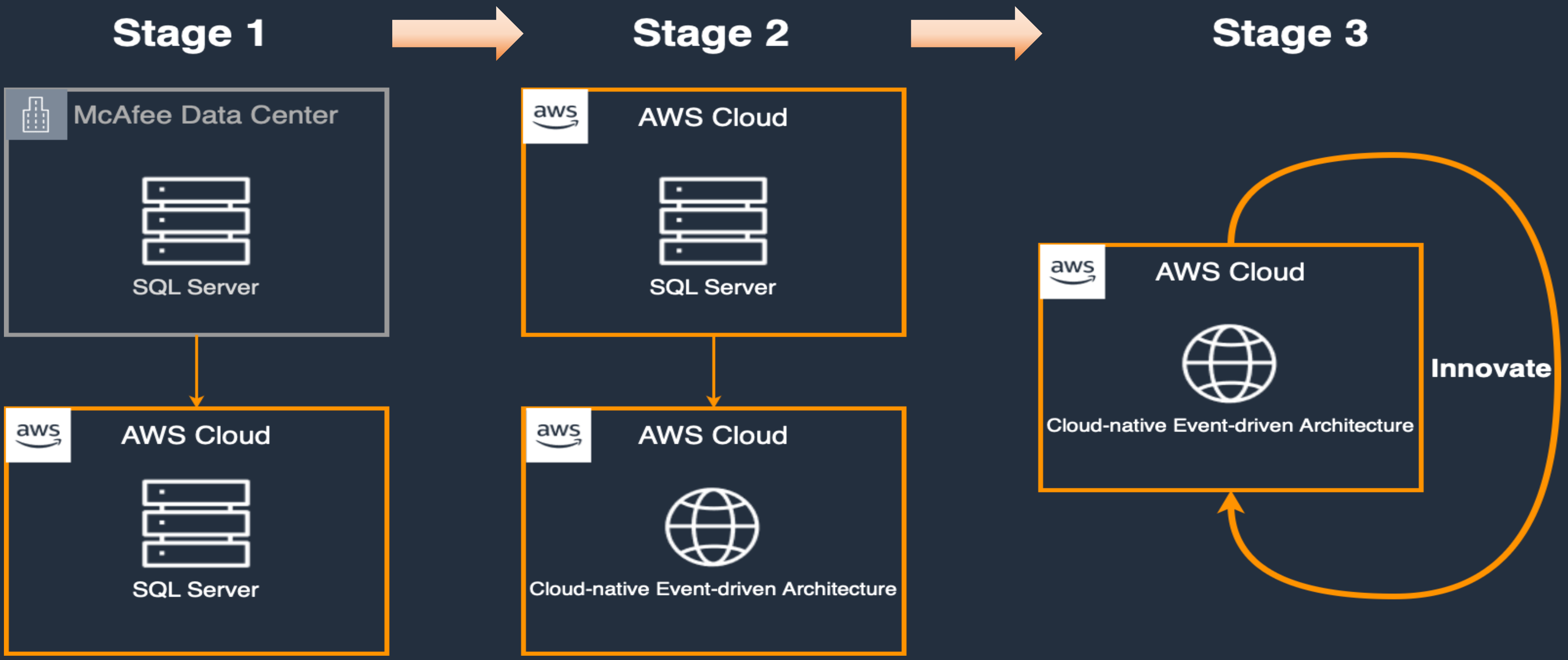
Database Modernization Lifecycle

ADC-D2

Modernization Lifecyle

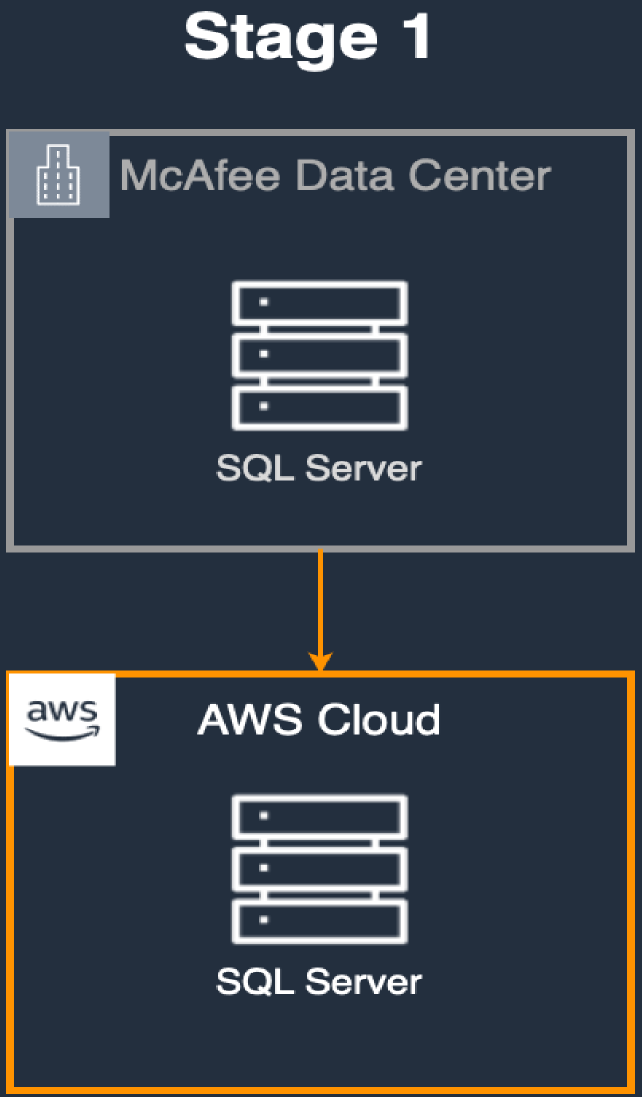
- Planning
- Data Analysis
- Data Modeling
- Testing
- Migration

McAfee Journey



Stage 1 : Migration

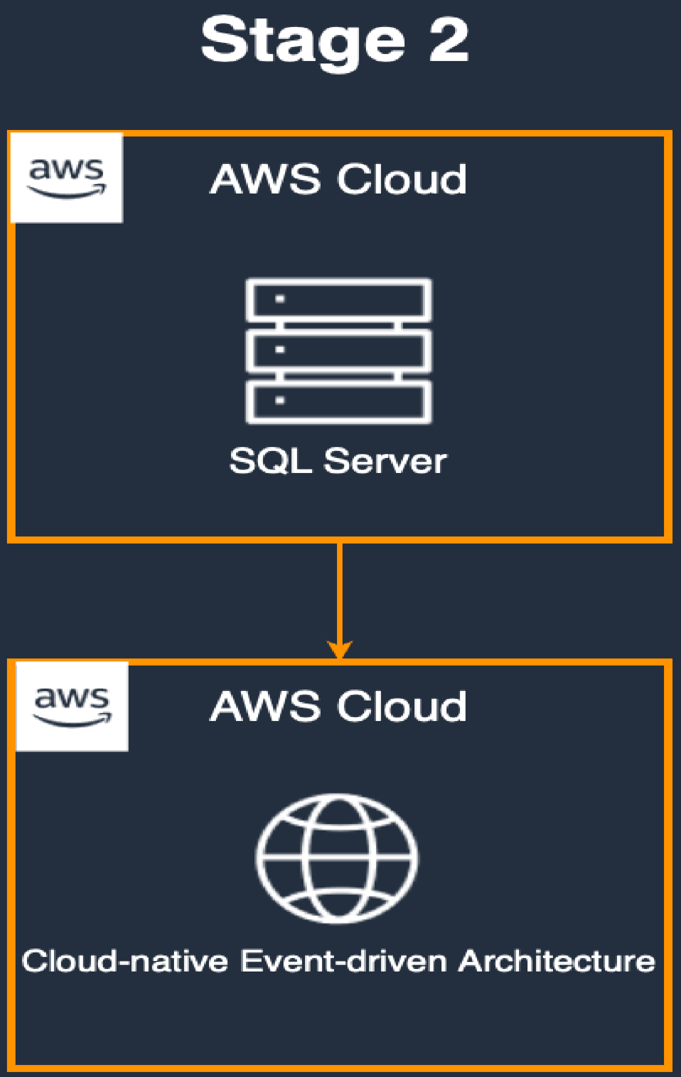
- Approach:
  - Lift and Shift
- Configuration:
  - Multi-AZ x2iedn.32/24xlarge
- Steps:
  - Copy data to AWS using DD Boost
  - Establish RO instance within Always On Failover Cluster
  - Failover Master from on-premises to AWS



ADC-D2

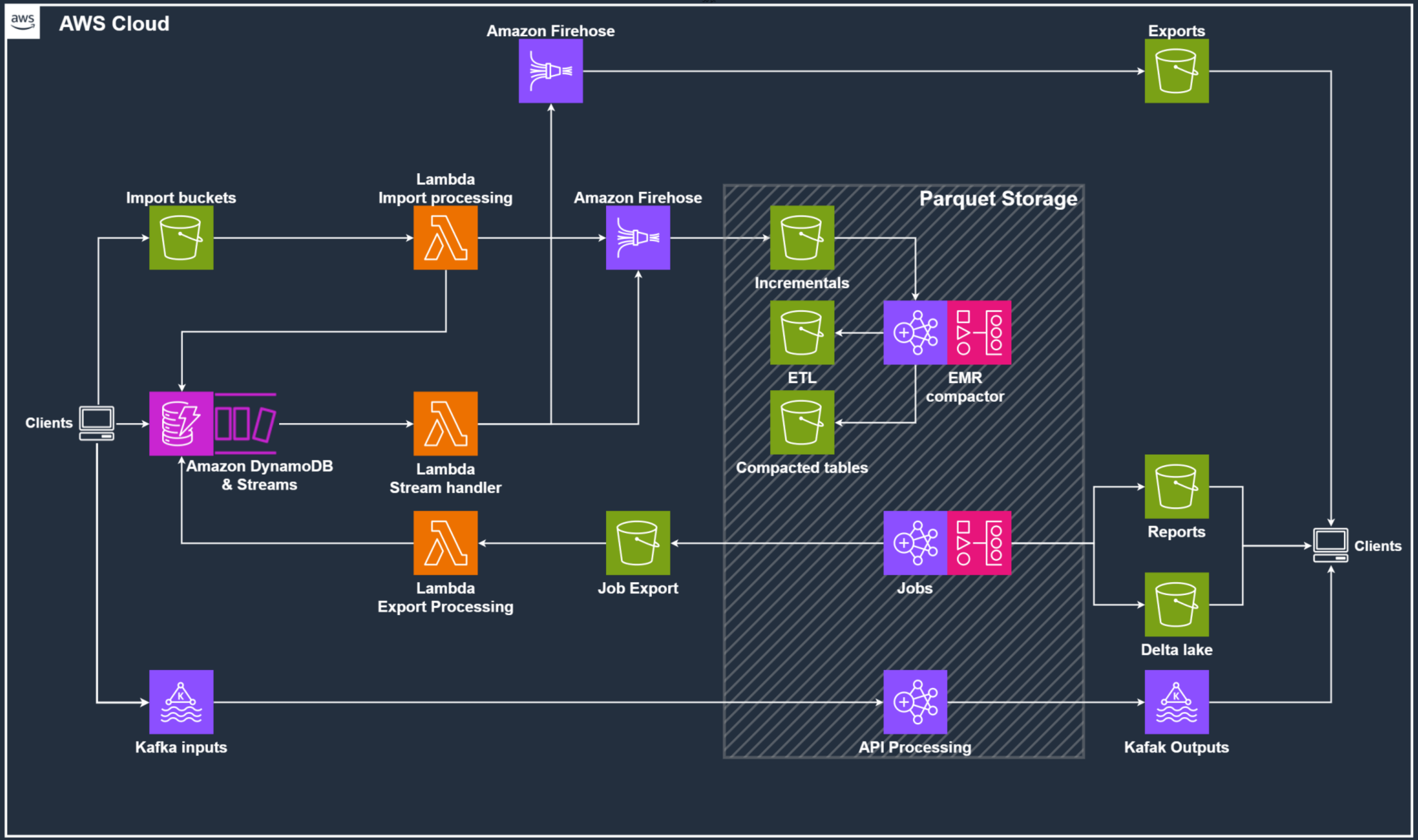
Stage 2 : Modernization

- Start by understanding system thoroughly
- Initiate Proof of Concept (PoC)
- Do gradual transition
- Reduce load to scale down clusters
- Deliver early value for the business



ADC-D2

Modernized Current Architecture



NoSQL Data Modeling

SQL and NoSQL side by side

SQL	NoSQL
Optimized for storage	**Optimized for compute**
Normalized/relational	Denormalized/hierarchical
Ad hoc queries	Instantiated views
Scale vertically	Scale horizontally
Good for OLAP	Built for OLTP* at scale

(*) DynamoDB is. Some NoSQL databases are built for analytical workloads.

Designing a Data Modeling

ADC-D2

Access patterns and schema

- Denormalized tables
- Keys
- Stored Procedures replacements
- Used DDB streams as triggers
- S3 partitioned tables for EMR

Productive NoSQL Tools

NoSQL Workbench

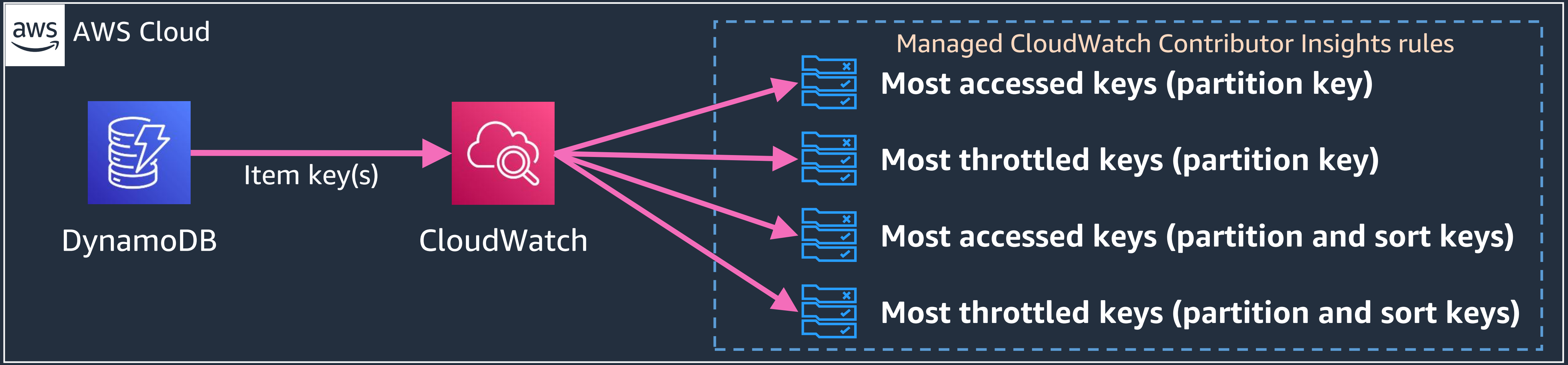
Enterprise ready

Data modeler

Visualizer

Operation builder

Amazon CloudWatch Contributor Insights (CCI) for DynamoDB

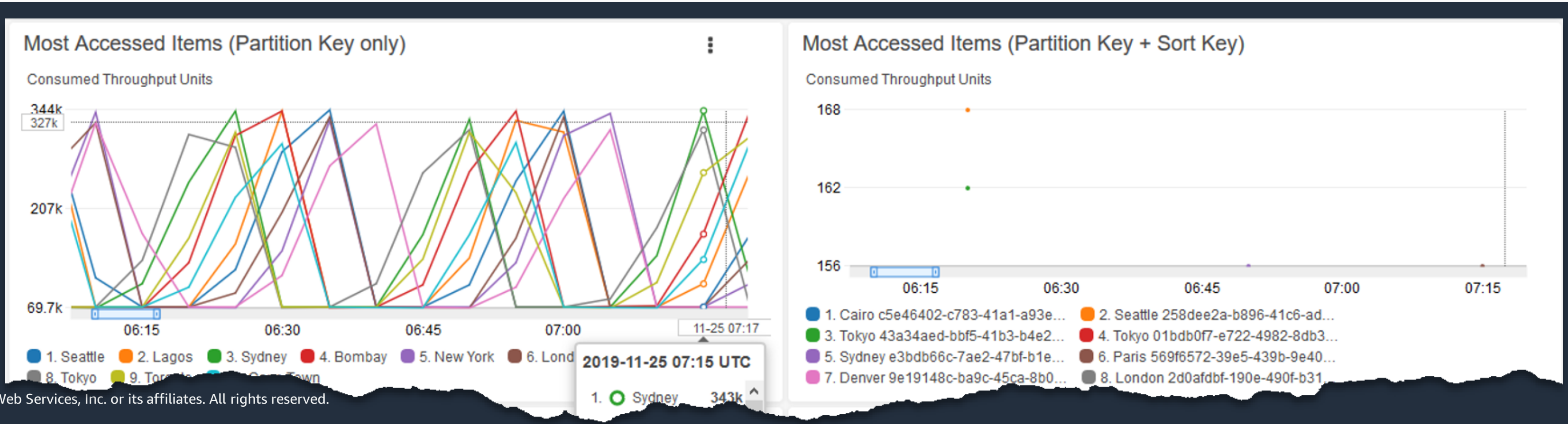


Most accessed keys (partition key)

Most throttled keys (partition key)

Most accessed keys (partition and sort keys)

Most throttled keys (partition and sort keys)



Most Accessed Items (Partition Key only)

Consumed Throughput Units

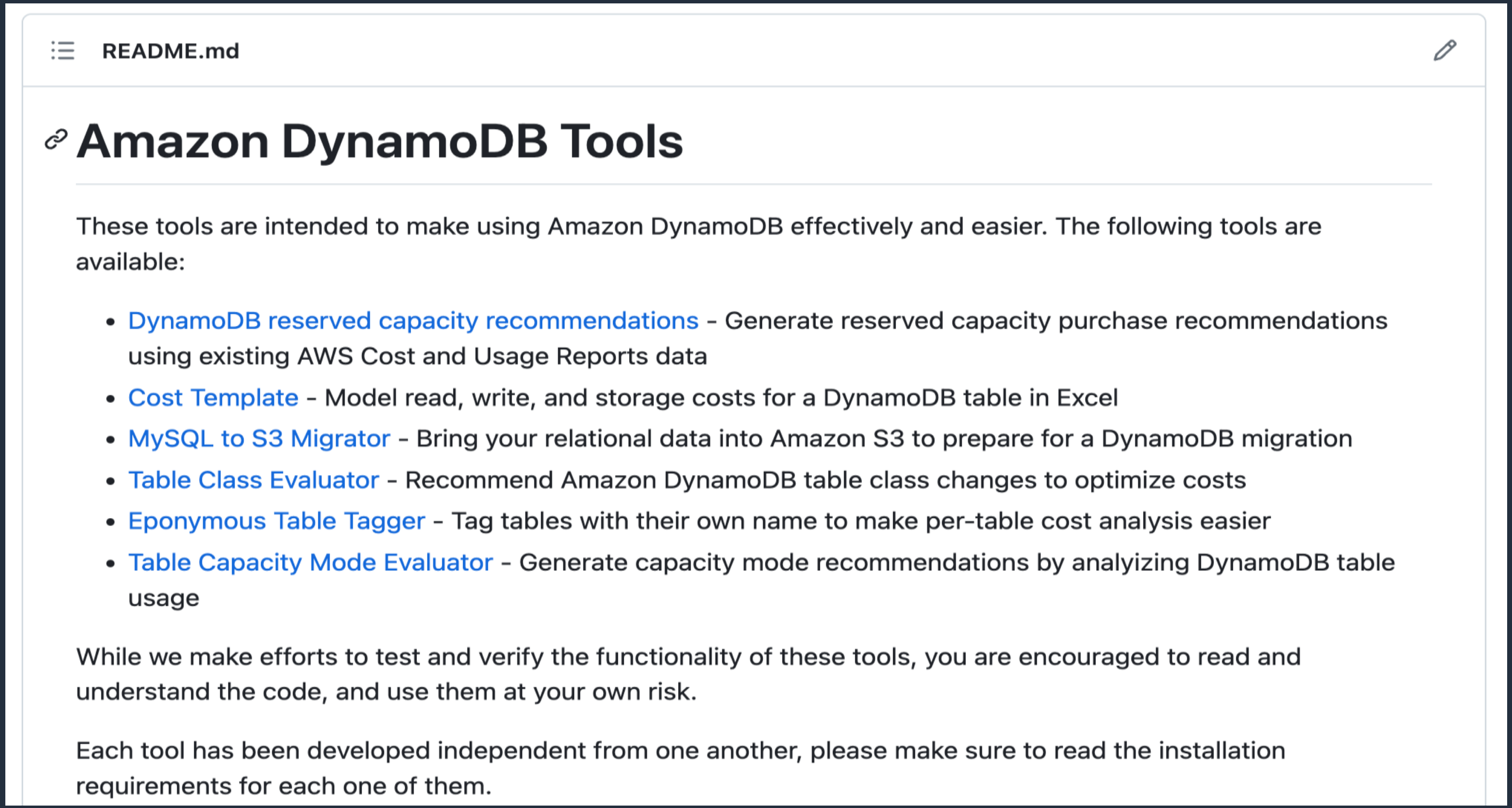
344k
327k
207k
69.7k
06:15
06:30
06:45
07:00
11-25 07:17
1. Seattle
2. Lagos
3. Sydney
4. Bombay
5. New York
6. Lond
8. Tokyo
9. Tor
2019-11-25 07:15 UTC
1. Sydney 343k

Most Accessed Items (Partition Key + Sort Key)

Consumed Throughput Units

168
162
156
06:15
06:30
06:45
07:00
07:15
1. Cairo c5e46402-c783-41a1-a93e...
2. Seattle 258dee2a-b896-41c6-ad...
3. Tokyo 43a34aed-bbf5-41b3-b4e2...
4. Tokyo 01bdb0f7-e722-4982-8db3...
5. Sydney e3bdb66c-7ae2-47bf-b1e...
6. Paris 569f6572-39e5-439b-9e40...
7. Denver 9e19148c-ba9c-45ca-8b0...
8. London 2d0afdbf-190e-490f-b31

Amazon DynamoDB Tools/Resources



README.md

Amazon DynamoDB Tools

These tools are intended to make using Amazon DynamoDB effectively and easier. The following tools are available:

- DynamoDB reserved capacity recommendations - Generate reserved capacity purchase recommendations using existing AWS Cost and Usage Reports data
- Cost Template - Model read, write, and storage costs for a DynamoDB table in Excel
- MySQL to S3 Migrator - Bring your relational data into Amazon S3 to prepare for a DynamoDB migration
- Table Class Evaluator - Recommend Amazon DynamoDB table class changes to optimize costs
- Eponymous Table Tagger - Tag tables with their own name to make per-table cost analysis easier
- Table Capacity Mode Evaluator - Generate capacity mode recommendations by analyizing DynamoDB table usage

While we make efforts to test and verify the functionality of these tools, you are encouraged to read and understand the code, and use them at your own risk.

Each tool has been developed independent from one another, please make sure to read the installation requirements for each one of them.

- https://github.com/awslabs/amazon-dynamodb-tools

Takeaways

ADC-D2

Business/Solution Outcomes

- Cost reduction:
  - Exit datacenter | Optimise License cost | Utilize suitable storage tiers

- Productivity enhancements:
  - Accelerated deployment
  - Comprehensive testing and faster development

- Scalability & Reliability

- Improved system visibility

- Additional recovery options

- Empowering data scientists

ADC-D2

Roadmap

- Derive more data from existing database
- Increase speed of data delivery
- Optimise DDB storage

Amazon DynamoDB Resources to get started !

- Getting started with Amazon DynamoDB
    - aws.amazon.com/dynamodb/getting-started/
- Resources for developers
    - aws.amazon.com/dynamodb/resources/
- Data modeling examples :
- https://github.com/aws-samples/amazon-dynamodb-design-patterns

twitter.com/DynamoDB

linkedin.com/showcase/aws-databases

youtube.com/AmazonWebServices

youtube.com/serverlessland

amazondynamodbofficehrs.splashthat.com

twitch.tv/aws