aws SUMMIT

Managing a Database Migration Project

Blair Layton

Business Development Manager, Database, Amazon Web Services

aws SUMMIT

ย้ายเลย! ย้ายข้อมูลไปยัง AWS (ระดับ 200): จัดการโครงการย้ายข้อมูล DB ด้วยวิธีที่ดีที่สุด

Surawut Phornthabthong

Solutions Architect, Amazon Web Services

What to Expect from the Session

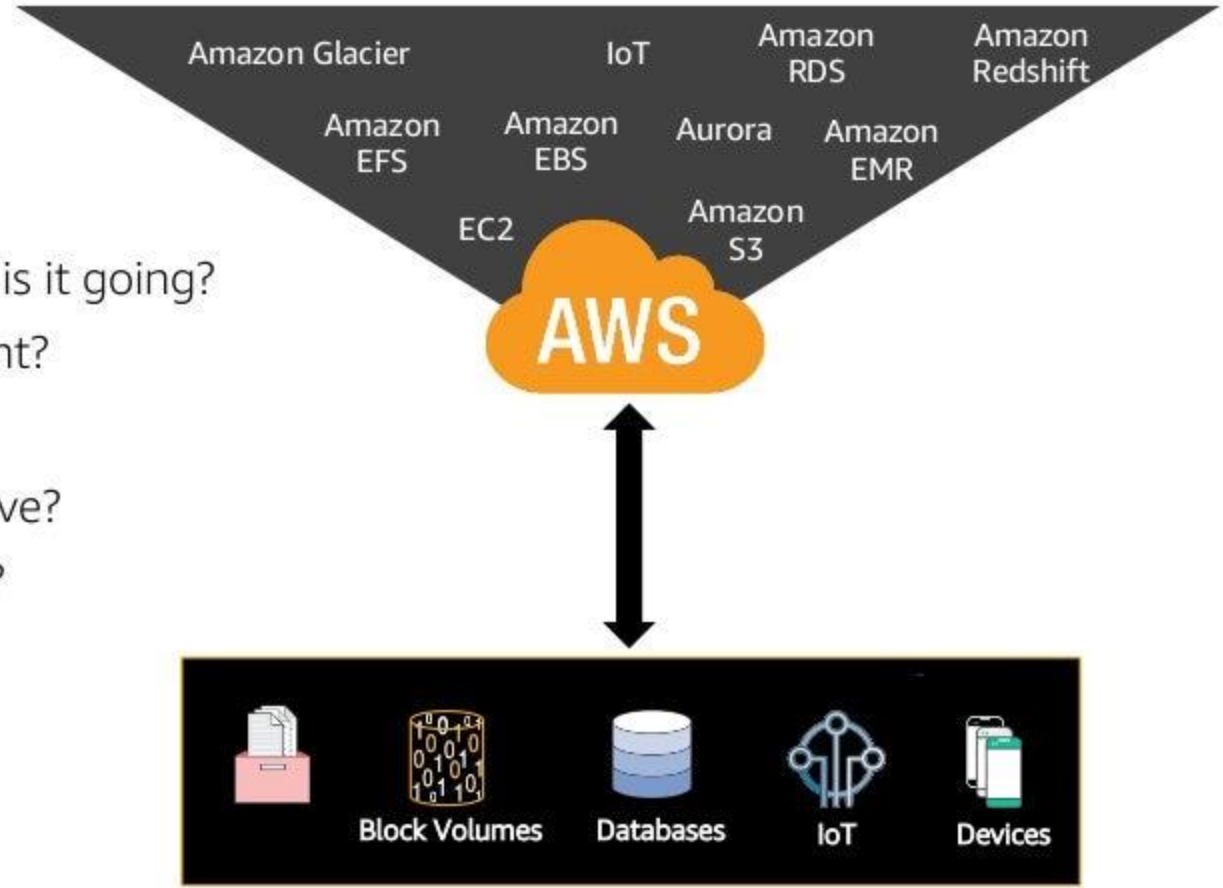
- Database Migration Context
- Database Migration Tools
- Introduction to the AWS Migration Framework
- Database Migration Effort
- Customer References
- Next Steps

aws SUMMIT

Database Migration Context

Migrating Data: Five Key Questions

1) What kind of data is it, and where is it going?
2) One-time or continuous movement?
3) One-way or bidirectional access?
4) How much data & time do you have?
5) How might your WAN be a factor?



Amazon RDS Engines

Aurora

Open source

Commercial

Microsoft SQL Server

Amazon DynamoDB

Fast and flexible NoSQL database service for any scale

Highly scalable

Automatic scaling to hundreds of terabytes of data that serve millions of requests per second

Fast, consistent performance

Consistent single-digit millisecond latency; DAX in-memory performance reduces response times to microseconds

Fully managed

Automatic provisioning, infrastructure management, scaling, and configuration with zero downtime

Business critical reliability

Data is replicated across fault-tolerant Availability Zones, with fine-grained access control

Migration Was Costly, Complex, & Slow

- Required commercial migration & replication software

- Caused long application downtime

- Was complex to set up & manage

- Required DB-specific application code

aws SUMMIT

Database Migration Tools

AWS Database Migration Service (AWS DMS)

DMS migrates databases to AWS easily and securely with minimal downtime. It can migrate your data to and from most widely used commercial and open-source databases.

PostgreSQL

ORACLE

Amazon Aurora

MariaDB

mongoDB.

DynamoDB

Amazon Redshift

MySQL

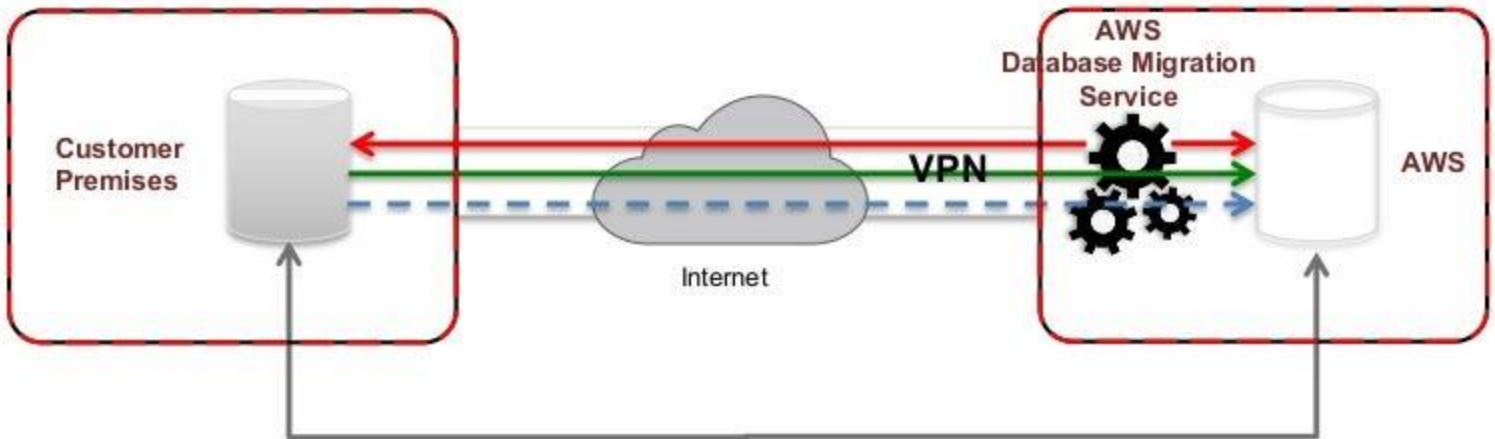
Microsoft SQL Server

Microsoft SQL Azure

SAP ASE

S3 Bucket

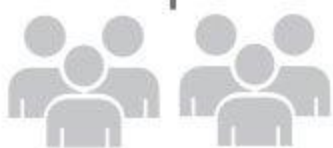
Keep Your Apps Running During the Migration



Start a replication instance

Connect to source and target databases

Select tables, schemas, or databases

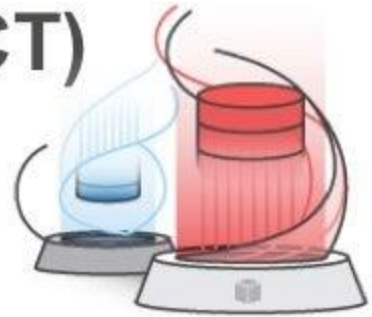


Application Users

Let AWS DMS create tables, load data, and keep them in sync

Switch applications over to the target at your convenience

AWS Schema Conversion Tool (AWS SCT)



SCT helps automate many database schema and code conversion tasks when migrating between database engines or data warehouse engines

ORACLE
Microsoft SQL Server
MariaDB
PostgreSQL
MySQL

MariaDB
Amazon Aurora
MySQL
PostgreSQL

TERADATA
ORACLE
NETEZZA
Greenplum
Microsoft SQL Server
VERTICA

AMAZON REDSHIFT

SCT can tell you how hard the migration will be

Database Migration Assessment Report

Executive Summary

We completed the analysis of your Oracle source database and estimate that 91% of the database storage objects and 100% of database code objects can be converted automatically or with minimal changes if you select Amazon Aurora as your migration target. Database storage objects include schemas, tables, columns, constraints, indexes, sequences, synonyms, user define types and types. Database code objects include functions, procedures, packages, triggers, views, materialized views, events, SQL scalar functions, SQL inline functions, SQL table functions, attributes, variables, constants, table types, public types, private types, cursors, exceptions, parameters and other objects. Based on our analysis of SQL syntax elements of your source database schema, we estimate that 99.9% of your entire database schema can be converted automatically to Amazon Aurora. To complete the migration, we recommend 597 conversion action(s) ranging from simple tasks to medium-complexity actions to significant conversion actions.

Database Objects with Conversion Actions for Amazon Aurora

Of the total 1,576 database storage object(s) and 155 database code object(s) in the source database, we were able to identify 1,427 (91%) database storage object(s) and 155 (100%) database code objects that can be converted automatically or with minimal changes to Amazon Aurora.

149 (9%) database storage object(s) required 149 significant user action(s) to complete the conversion.

Figure: Conversion statistics for database storage objects

Figure: Conversion statistics for database code objects

Detailed Recommendations for Amazon Aurora Migrations

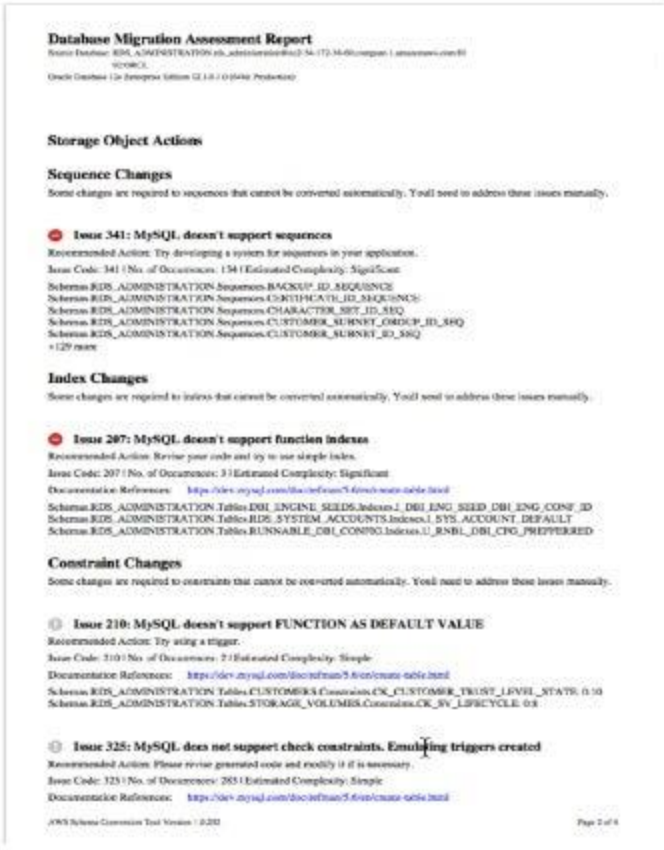
If you choose to migrate your Oracle database to Amazon Aurora, we recommend the following actions.

1. Connect SCT to Source and Target databases.

2. Run Assessment Report.

3. Read Executive Summary.

4. Follow detailed instructions.



Database Migration Assessment Report

Storage Object Actions

Sequence Changes

Some changes are required to sequences that cannot be converted automatically. You'll need to address these issues manually.

Issue 341: MySQL doesn't support sequences

Recommended Action: Try developing a system for sequences in your application.

Issue Code: 341 | No. of Occurrences: 134 | Estimated Complexity: Significant

Schemas.RDS_ADMINISTRATION.Sequences.BACKUP_ID_SEQUENCE
Schemas.RDS_ADMINISTRATION.Sequences.CHANGE_NOTIFICATION_ID_SEQUENCE
Schemas.RDS_ADMINISTRATION.Sequences.CHARACTER_SET_ID_SEQ
Schemas.RDS_ADMINISTRATION.Sequences.CUSTOMER_SUBNET_GROUP_ID_SEQ
Schemas.RDS_ADMINISTRATION.Sequences.CUSTOMER_SUBNET_ID_SEQ
+129 more

Index Changes

Some changes are required to indexes that cannot be converted automatically. You'll need to address these issues manually.

Issue 207: MySQL doesn't support function indexes

Recommended Action: Revise your code and try to use simple index.

Issue Code: 207 | No. of Occurrences: 3 | Estimated Complexity: Significant

Documentation References: https://dev.mysql.com/doc/refman/5.6/en/create-table.html

Schemas.RDS_ADMINISTRATION.Tables.DBI_ENGINE_SEEDS.Indexes.I_DBI_ENG_SEED_DBI_ENG_CONF_ID
Schemas.RDS_ADMINISTRATION.Tables.RDS_SYSTEM_ACCOUNTS.Indexes.I_SYS_ACCOUNT_DEFAULT
Schemas.RDS_ADMINISTRATION.Tables.RUNNABLE_DBI_CONFIG.Indexes.U_RNB1_DBI_CFG_PREFFERED

Constraint Changes

Some changes are required to constraints that cannot be converted automatically. You'll need to address these issues manually.

Issue 210: MySQL doesn't support FUNCTION AS DEFAULT VALUE

Recommended Action: Try using a trigger.

Issue Code: 210 | No. of Occurrences: 2 | Estimated Complexity: Simple

Documentation References: https://dev.mysql.com/doc/refman/5.6/en/create-table.html

Schemas.RDS_ADMINISTRATION.Tables.CUSTOMERS.Constraints.CK_CUSTOMER_TRUST_LEVEL_STATE: 0:10
Schemas.RDS_ADMINISTRATION.Tables.STORAGE_VOLUMES.Constraints.CK_SV_LIFECYCLE: 0:8

Issue 325: MySQL does not support check constraints. Emulating triggers created

Recommended Action: Please review generated code and modify it if it is necessary.

Issue Code: 325 | No. of Occurrences: 285 | Estimated Complexity: Simple

Documentation References: https://dev.mysql.com/doc/refman/5.6/en/create-table.html

aws SUMMIT

When to use DMS and SCT?

Tools for Migration Project Phases

Phase	Service/Tool	Notes
Assessment	AWS Schema Conversion Tool	Reports on the database objects, complexity and types of migration issues
Schema Migration	AWS Schema Conversion Tool	Copies a schema or migrates a schema depending on whether it is a homogeneous or heterogeneous migrations
Data Migration	AWS Database Migration Service AWS Schema Conversion Tool	Bulk load and change data capture (CDC) options Extraction and load for large data warehouses, including AWS Snowball integration
Application Migration	AWS Schema Conversion Tool	SQL statement migration in application code
Data Validation	AWS Database Migration Service	Ensure data is the same on source and target
Functional Testing	Various Tools on Marketplace	Ensure the application runs as intended
Performance Testing	Various Tools on Marketplace	Ensure the application performance as intended

Tools for Migration Scenarios

Scenario	Example	Recommendation
Homogeneous migration to the same database version and edition	Migration of Oracle Database 11gR2 Enterprise Edition from on-premise to EC2	Use the native replication technology to create a standby database and then failover to the standby database
Homogeneous migration to a different version	Migration of MySQL 5.5 to MySQL 5.7	AWS Schema Conversion Tool and AWS Database Migration Service
Homogeneous migration to a different edition	Migration of SQL Server Enterprise Edition to Standard Edition	AWS Schema Conversion Tool and AWS Database Migration Service
Heterogeneous migration	Migration from Oracle Database to PostgreSQL	AWS Schema Conversion Tool and AWS Database Migration Service

That’s the tools, but how to manage migration projects?

aws SUMMIT

Introducing the AWS Migration Framework

AWS Migration Framework

READINESS AND PLANNING

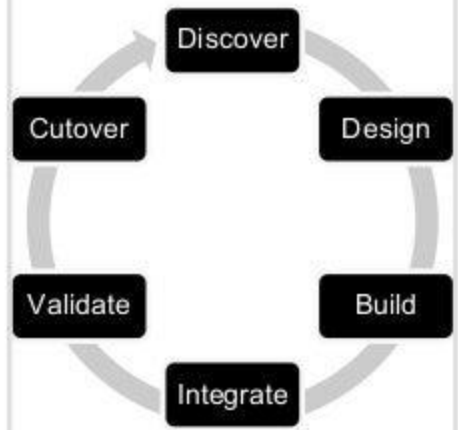
- Project Control
  - Strategy (business driver)
  - Key Stakeholders and Team
  - Plan (Scope, Schedule, Resources)
  - Cost Estimation
- Portfolio discovery
- Migration plan
- Operations Integration
- Security

Prioritized Backlog (PLAN)

ACTIVATE

- Prioritized Backlog
  - Application groups
  - Migration strategy
  - Success criteria
- Ops Integration – Foundation and Landing Zone (target zone setup)
- Setup Factory (Tools, Teams, Processes)
- Pilot Migration

EXECUTE



OPTIMIZE

- Application optimization
- Process optimization
- Operational optimization
- Cost optimization

AWS Migration Framework- Readiness and Planning

AWS Migration Framework - Readiness & Planning

READINESS AND PLANNING

- **Project Control**
  - **Strategy (business driver)**
  - **Key Stakeholders and Team**
  - **Plan (Scope, Schedule, Resources)**
  - **Cost Estimation**
- Portfolio discovery
- Migration plan
- Operations Integration
- Security

Project Control focuses on ensuring there is a migration strategy in place that is supported by key stakeholders in the organisation. Additionally, we look at defining the team that will carry out the work, with associated timelines and cost estimations.

Sample decision points:

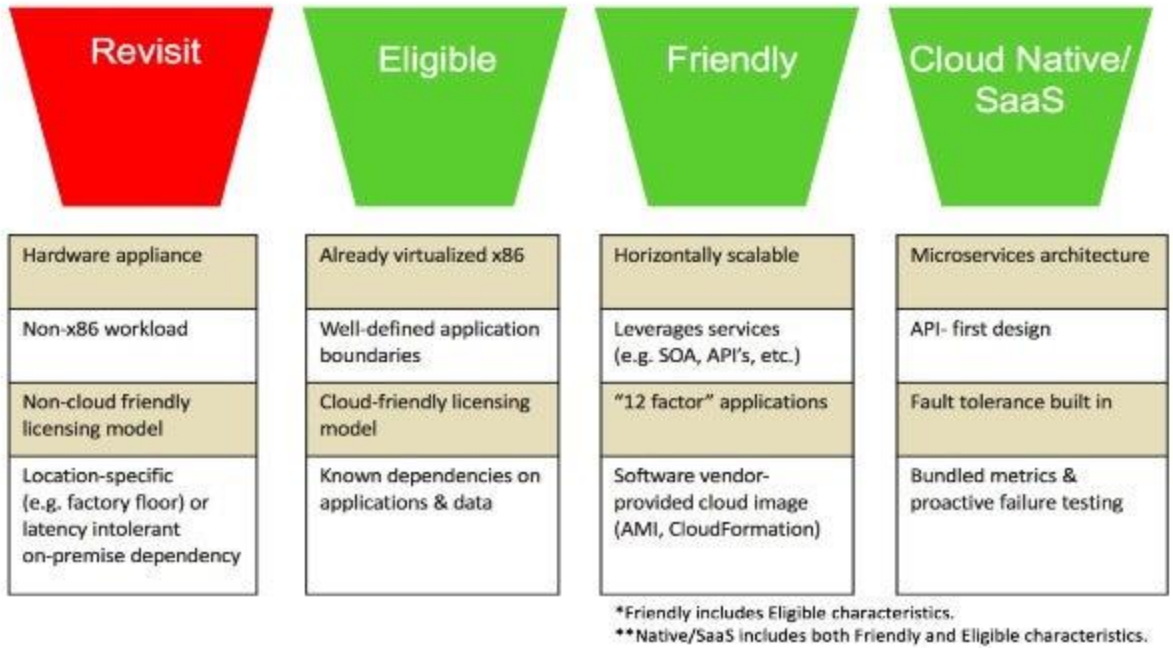
- Who is the executive sponsor?
- Are there any compelling events that will affect the migration strategy?
- Do we have the right resources? How are they organized?
- What are the timeframes we are working with?
- Do we have the necessary budget?

AWS Migration Framework - Readiness & Planning

READINESS AND PLANNING

- Project Control
  - Strategy (business driver)
  - Key Stakeholders and Team
  - Plan (Scope, Schedule, Resources)
  - Cost Estimation
- **Portfolio discovery**
- Migration plan
- Operations Integration
- Security

Quickly understand which applications are Cloud Eligible, Cloud Friendly, or Cloud Native and then execute a dive deep analysis on just that subset of applications.



*Friendly includes Eligible characteristics.
**Native/SaaS includes both Friendly and Eligible characteristics.

Application Assessment

- Business driver and intended ROI?
- Migration sponsor (business owner, C-level)?
- ISV application? Does the ISV support the target?
- Maintenance window for the migration?
- Design documentation?
- Original developers/DBAs still available?

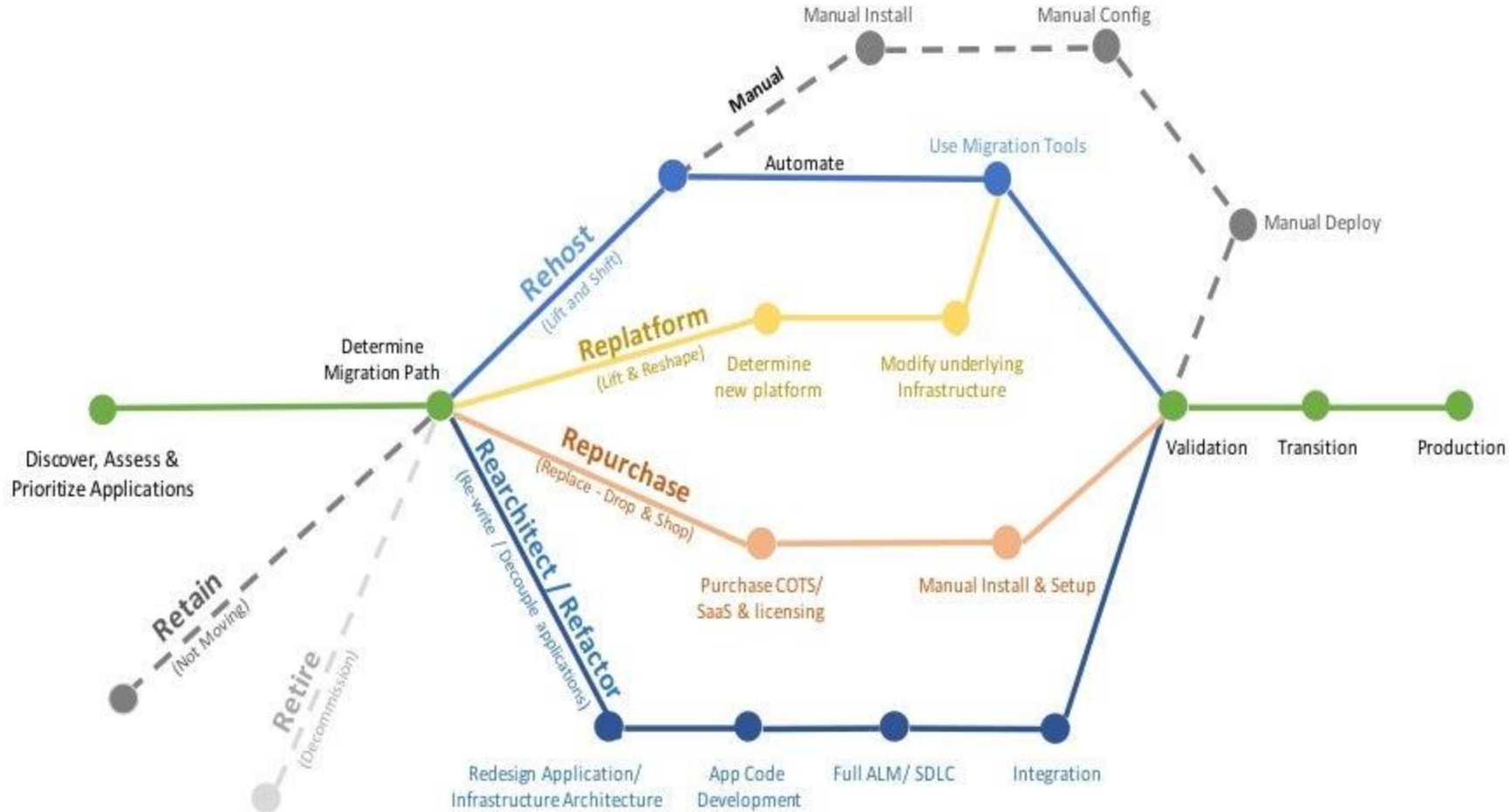
Database Assessment

- How many database objects (tables, triggers, SPs, users, etc.)?
- How much data?
- Complexity of the SPs and triggers?
- Proprietary DB features?
- Non-standard or custom data types?
- Character set conversions?
- Time zone or UTC?
- User authentication method?
- Licensing mechanism (cores, users, ULA etc.)

Application Technical Assessment

- Database Access:
  - SQL statements throughout the code?
  - Calls to a data abstraction layer?
  - API calls?
- ANSI SQL used where possible?
- SQL complexity, e.g. analytics with many joins or simple CRUD?
- Number of lines of SQL code?
- Application access, e.g. LDAP, DB Users, etc.

The 6Rs of Migration Planning



AWS Migration Framework - Activate

AWS Migration Framework - Activate

ACTIVATE

- Prioritized Backlog
  - Application groups
  - Migration strategy
  - Success criteria
- Ops Integration – Foundation and Landing Zone (target zone setup)
- Setup Factory (Tools, Teams, Processes)
- Pilot Migration

- Determine your application priorities and group integrated applications together
- Outline the success criteria for each application migration
- Create your AWS landing zone (accounts, VPC, subnets, IAM roles, VPN/Direct Connect, etc.)
- Configure DMS, SCT and other migration tools
- Team creation
- POC/pilot

Building a Migration Team

Application architect/developer: Application expert who can identify what components are important, complex, redundant, etc.

Source DBA: Knows the database design, schema, features used and what must be migrated to the target.

Target DBA: An expert in the target database to help map features from the source DB with the Source DBA.

AWS Solution Architect: Determines the correct target architecture in AWS and is familiar with DMS/SCT.

Application/Database Developers: Customer and/or partner resources to migrate the stored procedures, triggers and application code.

Hiring and Developing Talent

New skills are needed for the target DB and often AWS if migrating from on-premises

Develop training plans for existing employees

Hire in required skills if necessary

Retrain, redeploy or make people redundant who's skills are no longer relevant

Pilot/POC

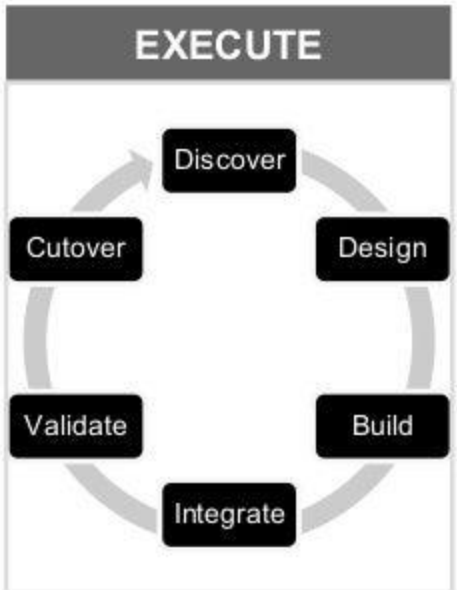
Choose a reasonably complex module/component to migrate to validate your assumptions in the Activate phase

You should:

- Obtain more accurate migration assessments
- Determine what can be automated
- Learn how the migration tools behave (limitations, bugs, improvements needed)
- Learn what skills are missing from your team

AWS Migration Framework - Execute

AWS Migration Framework - Execute



- Always have a back up plan!
- Execute according to lessons from Pilot/POC
- Typically the same amount of time to migrate the DB as to migrate the application (assuming DAL)
- Determine how you will cutover
  - Parallel run: expensive and difficult
  - Minimal downtime: DMS+CDC
  - Large maintenance window: application and data verification before go live

AWS Migration Framework - Optimize

AWS Migration Framework - Optimize

OPTIMIZE

- Application optimization
- Process optimization
- Operational optimization
- Cost optimization

- DMS instance and task optimization
- Database tuning
- Database instance right sizing
- Application tuning
- Application instance right sizing
- Use EC2 and RDS stop/start to optimize costs
- Purchase reserved instances and use spot instances
- Look for contention and evaluate caching, NoSQL, federation and adoption of a microservices strategy
- Perform HA/DR scenarios and optimize the use of AWS managed services to help, e.g. RDS MAZ, Auto-scaling

aws SUMMIT

Migration Effort

Database migration – multi phase process

Phase	Description	Automation	Effort (%)
1	Assessment	SCT	2
2	Database Schema Conversion	SCT/DMS	14
3	Application Conversion/Remediation	SCT	25
4	Scripts Conversion	SCT	7
5	Integration with 3rd party applications		3
6	Data Migration	DMS	4
7	Functional testing of the entire system		29
8	Performance tuning	SCT	2
9	Integration and deployment		7
10	Training and knowledge		2
11	Documentation and version control		2
12	Post production support		3

Database Migration Process

STEP 1:

Conversion

Source: Oracle Database on-premises, in EC2 or RDS

AWS Schema Conversion Tool

Target: Amazon Aurora Database

STEP 2:

Data migration

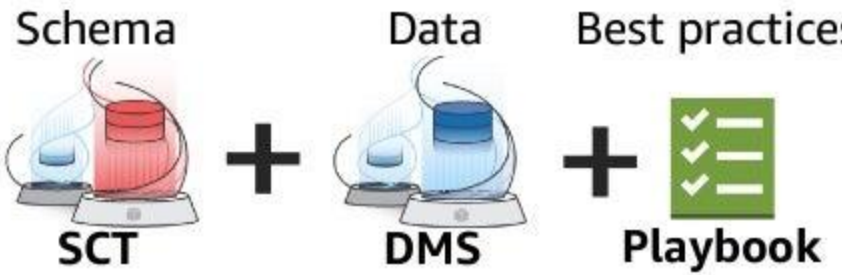
Source: Oracle Database on-premises, in EC2 or RDS

AWS Database Migration Service

Target: Amazon Aurora Database

Oracle to Aurora Migration Playbook

- Topic-by-topic overview of Oracle to Aurora PostgreSQL migrations and “hands-on” best practices
- How to migrate from proprietary features and the different database objects
- Migration best practices



	Oracle Feature	PostgreSQL Feature	Compatibility
Link	Index Organized Tables (IOTs)	PostgreSQL "Cluster" Tables	Yes*
Link	Common Data Types	Common Data Types	Yes
Link	Table Constraints	Table Constraints	Yes
Link	Table Partitioning including: RANGE, LIST, HASH, COMPOSITE, Automatic LIST	Table Partitioning including: RANGE, LIST	Yes*
Link	Exchange & Split Partitions	N/A	None
Link	Temporary Tables	Temporary Tables	Yes*
Link	Unused Columns	ALTER TABLE DROP COLUMN	Yes
Link	Virtual Columns	Views and/or Function as a Column	Yes*
Link	User Defined Types (UDTs)	User Defined Types (UDTs)	Yes
Link	Read Only Tables & Table Partitions	Read Only Roles and/or Triggers	Yes*
Link	Index Typ		
Link	B-Tree Inc		
Link	Composit		
Link	BITMAP Ir		
Link	Function-I		
Link	Global an Indexes		
Link	Identity C		
Link	MVCC (Table & F		
Link	Character		
Link	Transacti		
Link	Recovery Manager (RMAN)	AWS Aurora Snapshots	Yes
Link	Flashback Database	AWS Aurora Snapshots	Yes
Link	12c Multi-tenant architecture: PDBs and CDB	Databases	Yes*
Link	Tablespaces & DataFiles	Tablespaces	Yes*
Link	Data Pump	pg_dump & pg_restore	Yes
Link	Resource Manager	Separate AWS Aurora Clusters	Yes
Link	Database Users	Database Roles	Yes
Link	Database Roles	Database Roles	Yes
Link	SGA & PGA Memory	Memory Buffers	Yes
Link	V$ Views & Data Dictionary	System Catalog Tables, Statistics Collector, AWS Aurora Performance Insights	Yes*
Link	Log Miner	Logging Options	Yes
Link	Instance & Database Parameters (SPFILE)	AWS Aurora Parameter Groups	Yes
Link	Session Parameters	Session Parameters	Yes
Link	Alert.log (error log)	Error Log via AWS Console	Yes
Link	Automatic and Manual Statistics Collection	Automatic and Manual Statistics Collection	Yes
Link	Viewing Execution Plans	Viewing Execution Plans	Yes

https://aws.amazon.com/dms/getting-started/

aws SUMMIT

Customer / Partner References

>50,000 Databases Migrated with DMS

Expedia migrated from SQL Server to AWS

Needed real-[illegible] market pricing

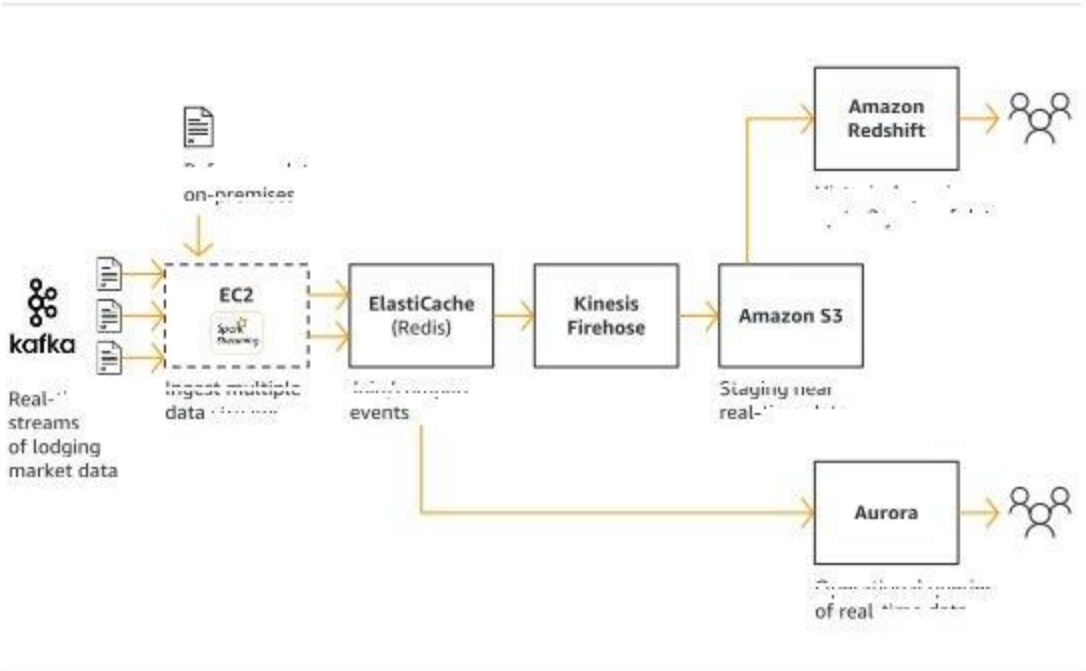
Migrated from Microsoft SQL Server

Use Amazon Aurora, Amazon Redshift, Kinesis, and ElastiCache

Process high-volume pricing and availability data

Query execution times reduced 80%–95%

Database has >15B rows and continues to grow



Trimble migrated from Oracle to Amazon RDS PostgreSQL

"The SCT Assessment Report was the key enabler to allow us to understand the scope of effort required to complete an Oracle to PostgreSQL migration.

What was originally thought to be a largely manual task that no one was particularly excited about having to do became a very straight forward quick and easy process."

- Todd Hofert, Director of Infrastructure Operations, Trimble

aws SUMMIT

Next Steps

Next Steps

- Talk to your AWS account team and AWS Partner
- Ask us about funding for POCs and commercial DB migrations (e.g. Oracle Database to Aurora)
- Read Documentation, White Papers, Playbooks
- Links:
    - DMS & SCT: https://aws.amazon.com/dms/
    - Getting Started Guides and Playbooks: https://aws.amazon.com/dms/getting-started/

aws SUMMIT

Thank you!

aws SUMMIT

ข้อมูลเชิงลึกของลูกค้าและ Machine Learning (Level 200 -300): การสร้าง Data Lake แบบไร้เซิร์ฟเวอร์บน AWS

Surawut Phornthabthong

Solutions Architect, Amazon Web Services

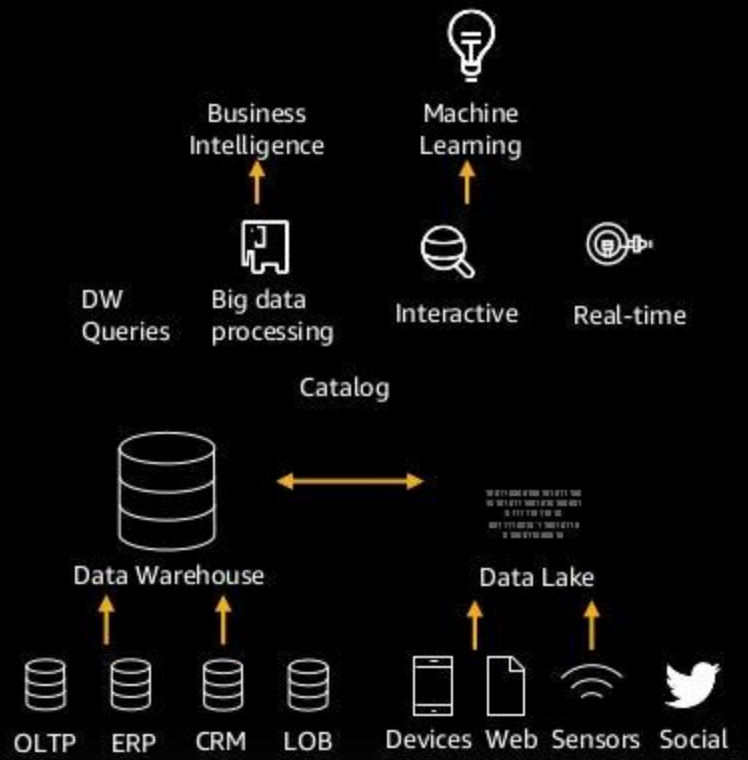
aws SUMMIT

Architecting a Serverless Data Lake on AWS

Surawut Phornthabthong
Solutions Architect, Amazon Web Services

What is a Data Lake ?

A data lake is an **architectural approach** that allows you to store **massive amounts** of data into a **central location**, so it's readily available to be categorized, processed, analyzed and **consumed by diverse group of users** within an organization.



Challenges faced by data teams

aws SUMMIT

Exponential growth in data


Transactions


Billing


ERP


Web Logs


Sensor Data


Infrastructure Logs


Social

Diversified consumers


Data Scientists

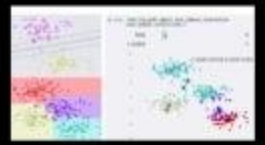

Applications


Business Analyst


External Consumers

Multiple access mechanisms


API Access


Notebooks


BI Tools

Characteristics of a data lake

aws SUMMIT

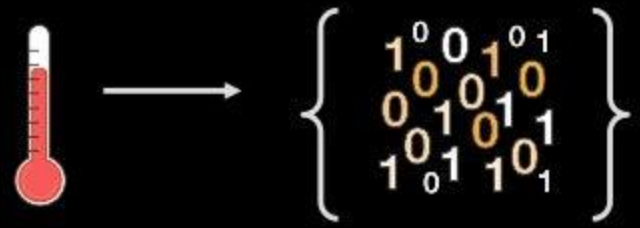
Collect Anything

Dive in Anywhere

Flexible Access

Future Proof

Let's take an example



Device Data

Record-level data

Design Outcomes

Serverless

Ingest & **store** data in real-time

Discover and **catalog** data stored in the lake

Enable batch & real-time **processing**

Consume raw & processed data

Scalable, Highly Available, Pay what you use

aws SUMMIT

Data Lake Ingestion

Multiple data lake ingestion methods

aws SUMMIT

AWS Snowball and AWS Snowmobile
- PB-scale migration

Amazon Kinesis Firehose
- Ingest device streams
- Transform and store on Amazon S3

AWS Storage Gateway
- Migrate legacy files

AWS Direct Connect
- On-premises integration

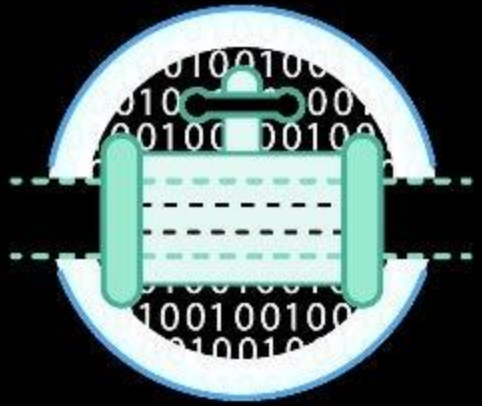
Native/ISV Connectors
- Ecosystem integration

Amazon S3 Transfer Acceleration
- Long-distance data transfer

Amazon Kinesis - real-time analytics

aws SUMMIT

Easily collect, process, and analyze video and data streams in real time



Kinesis Data Streams

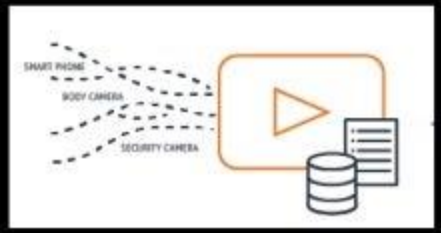
Build custom applications that analyze data streams

Kinesis Data Firehose

Load data streams into AWS data stores

Kinesis Data Analytics

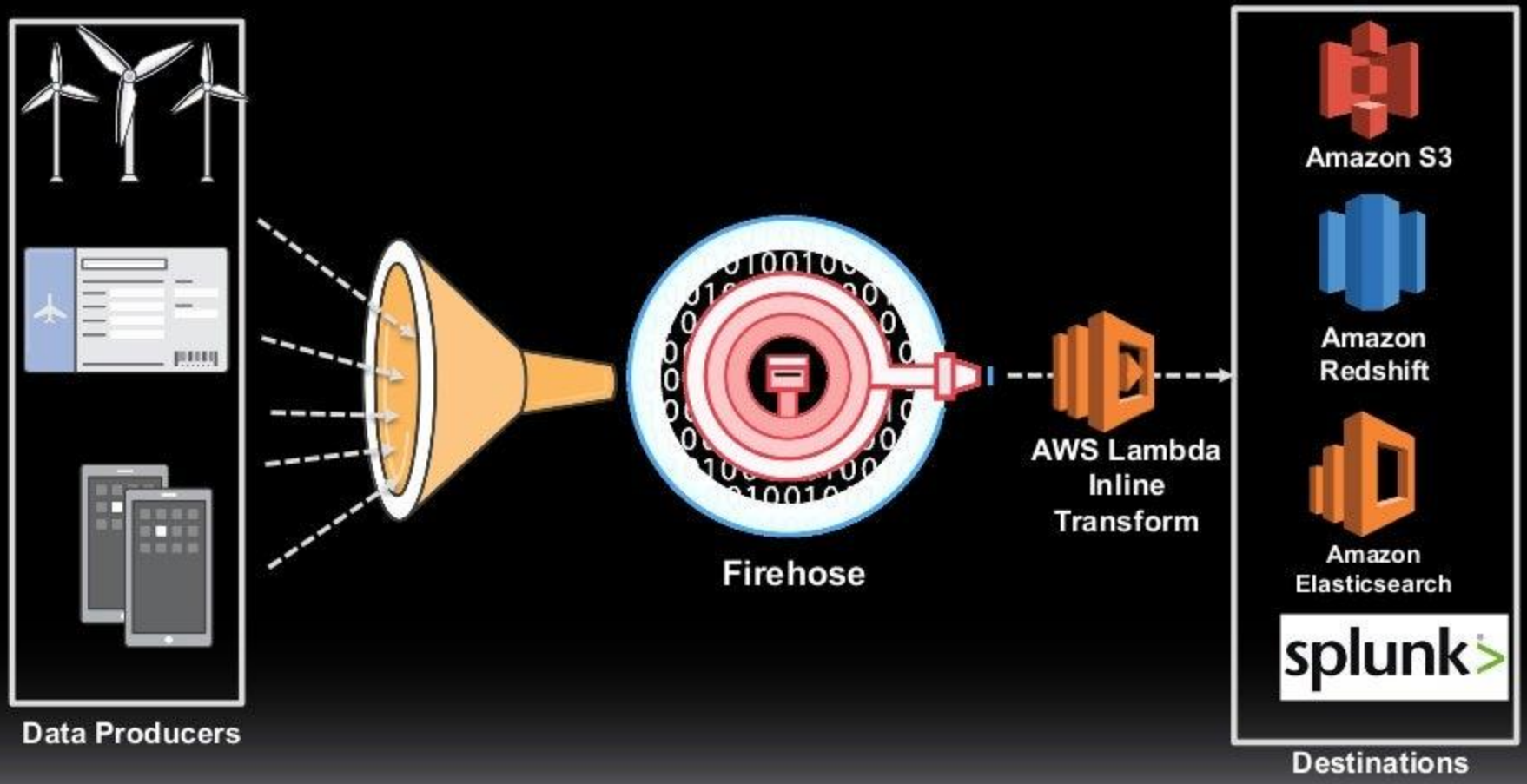
Analyze data streams with SQL



Kinesis Video Streams

Capture, process, and store video streams for analytics

Serverless data delivery with Kinesis Firehose



Architecture

aws SUMMIT

Data Lake Storage

Amazon S3 - Infinite, Durable & Cost Effective Storage

Unmatched durability, availability, and scalability

Best security, compliance, and audit capability

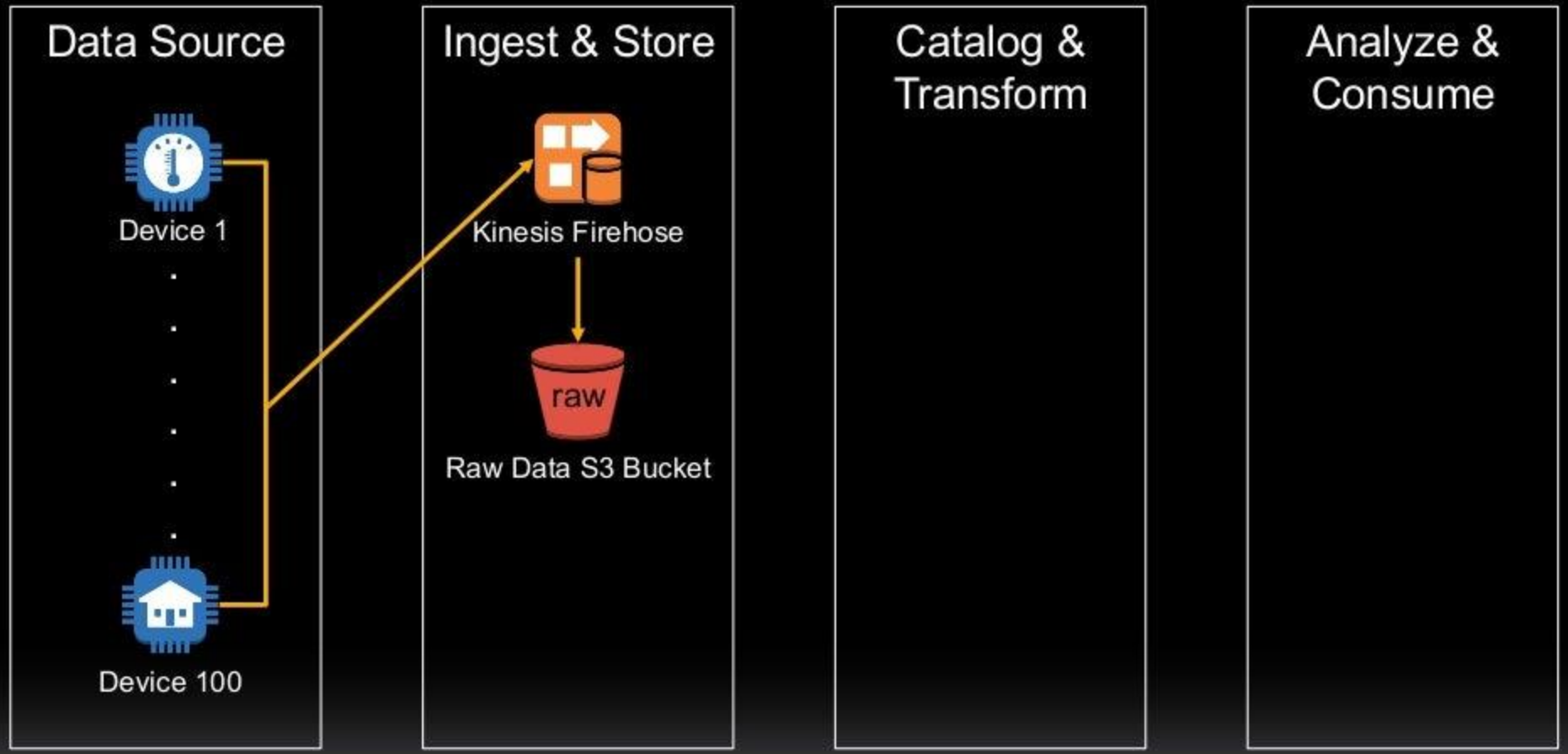
Object-level control at any scale

Business insight into your data

Most ways to bring data in

Twice as many partner integrations

Architecture



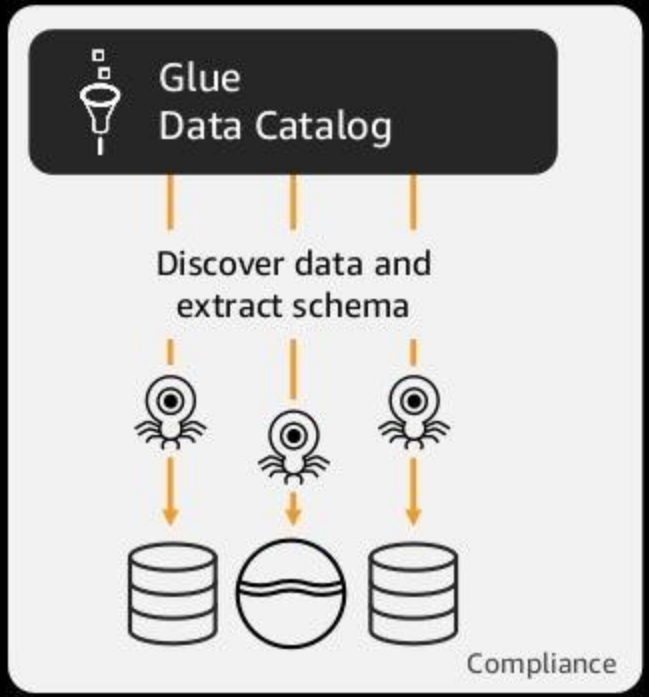
aws SUMMIT

Data Lake Metadata Management

Discover, Catalog & ETL

AWS Glue—data catalog

Make data discoverable



Automatically **discovers data** and **stores schema**

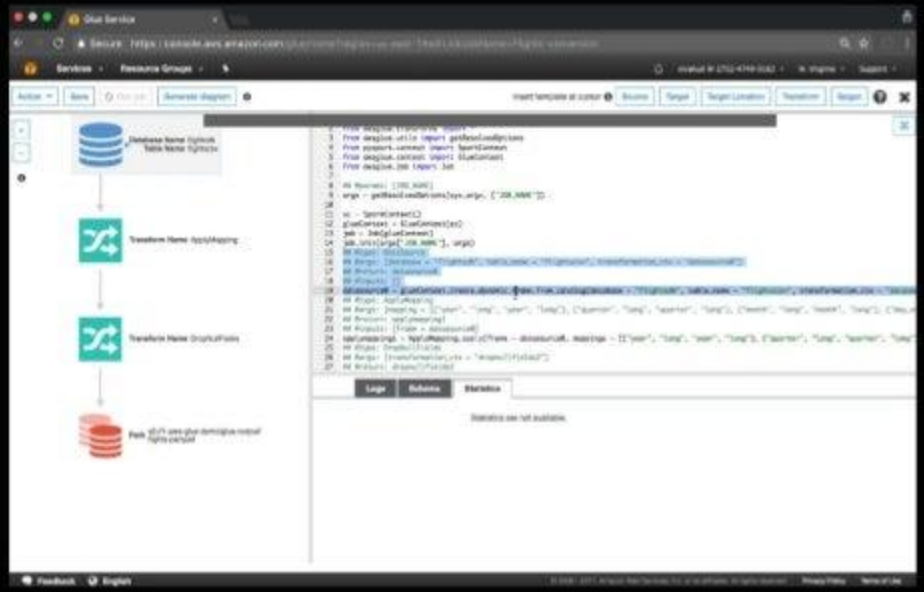
Catalog makes **data searchable**, and available for ETL

Catalog contains table and job definitions

Computes statistics to make queries efficient

AWS Glue—ETL service

Make ETL scripting and deployment easy



Serverless Transformations

Based on **Apache Spark**

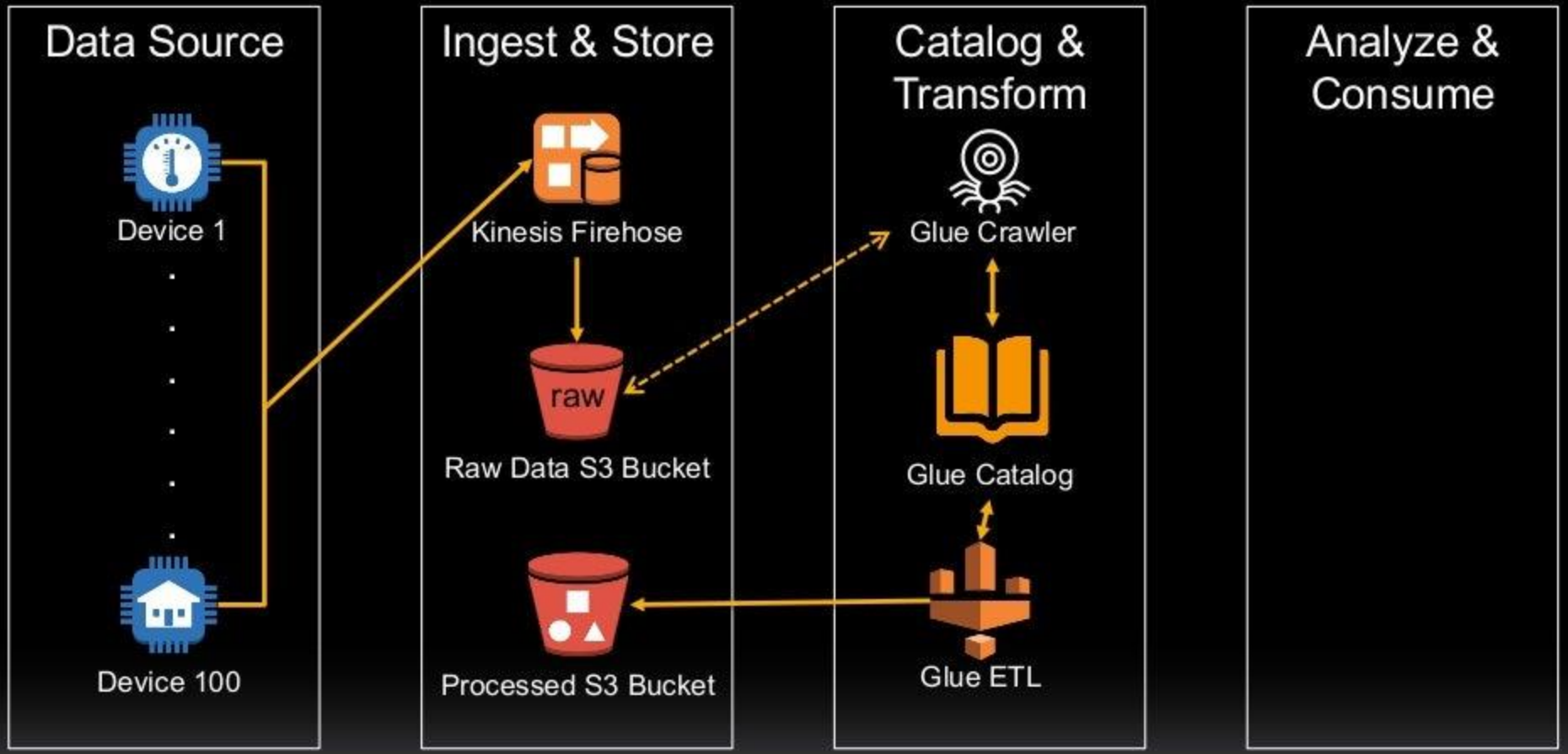
Automatically **generates ETL code**

Code is customizable with **PySpark** and **Scala**

Endpoints provided to edit, debug, test code

Jobs are **scheduled** or **event-based**

Architecture



aws SUMMIT

Data Lake Analytics

Amazon EMR - Big Data Processing

Fully managed – Hadoop Framework

19 Apps : Hadoop, Hive , **Spark**, HBase, **Presto**, and more

S3 Integration – **Decouple Compute and Storage**

Low Cost – **Transient Clusters**, Per Second Pricing, Spot Instances

Amazon Redshift - Modern Data Warehousing

Fast, scalable, **fully managed EDW** at **1/10th the cost** of other EDWs

Massively parallel, scales from gigabytes to **exabytes**

Access data across your Redshift DW and Amazon S3 **data lake**

Amazon Redshift

Amazon Athena—Interactive Analysis

aws SUMMIT

Interactive query service to analyze data in Amazon S3 using standard SQL

No infrastructure to set up or manage and no data to load

Query Instantly

Zero setup cost; just point to S3 and start querying

Pay per query

Pay only for queries run; save 30–90% on per-query costs through compression

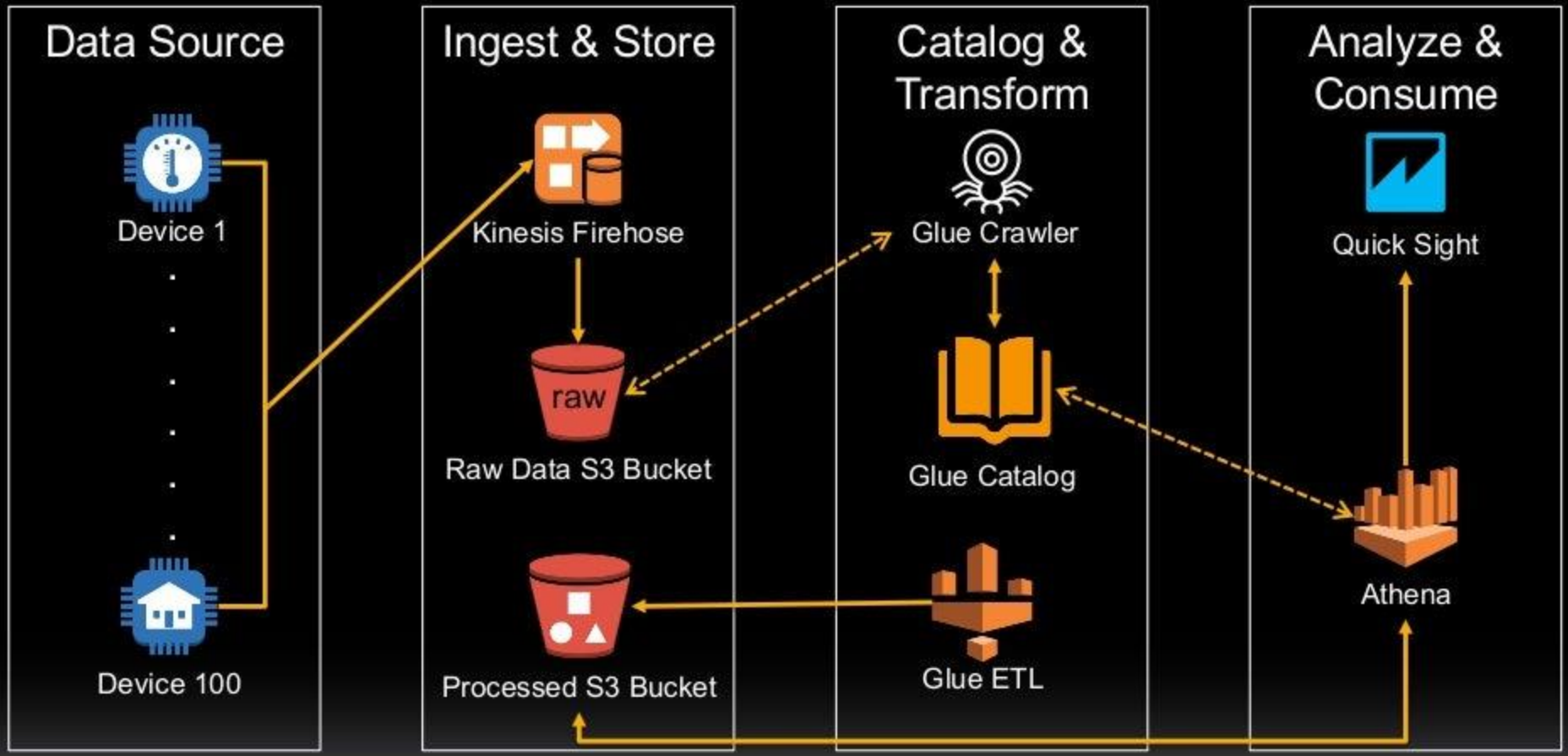
Open

ANSI SQL interface, JDBC/ODBC drivers, multiple formats, compression types, and complex joins and data types

Easy

Serverless: zero infrastructure, zero administration
Integrated with QuickSight

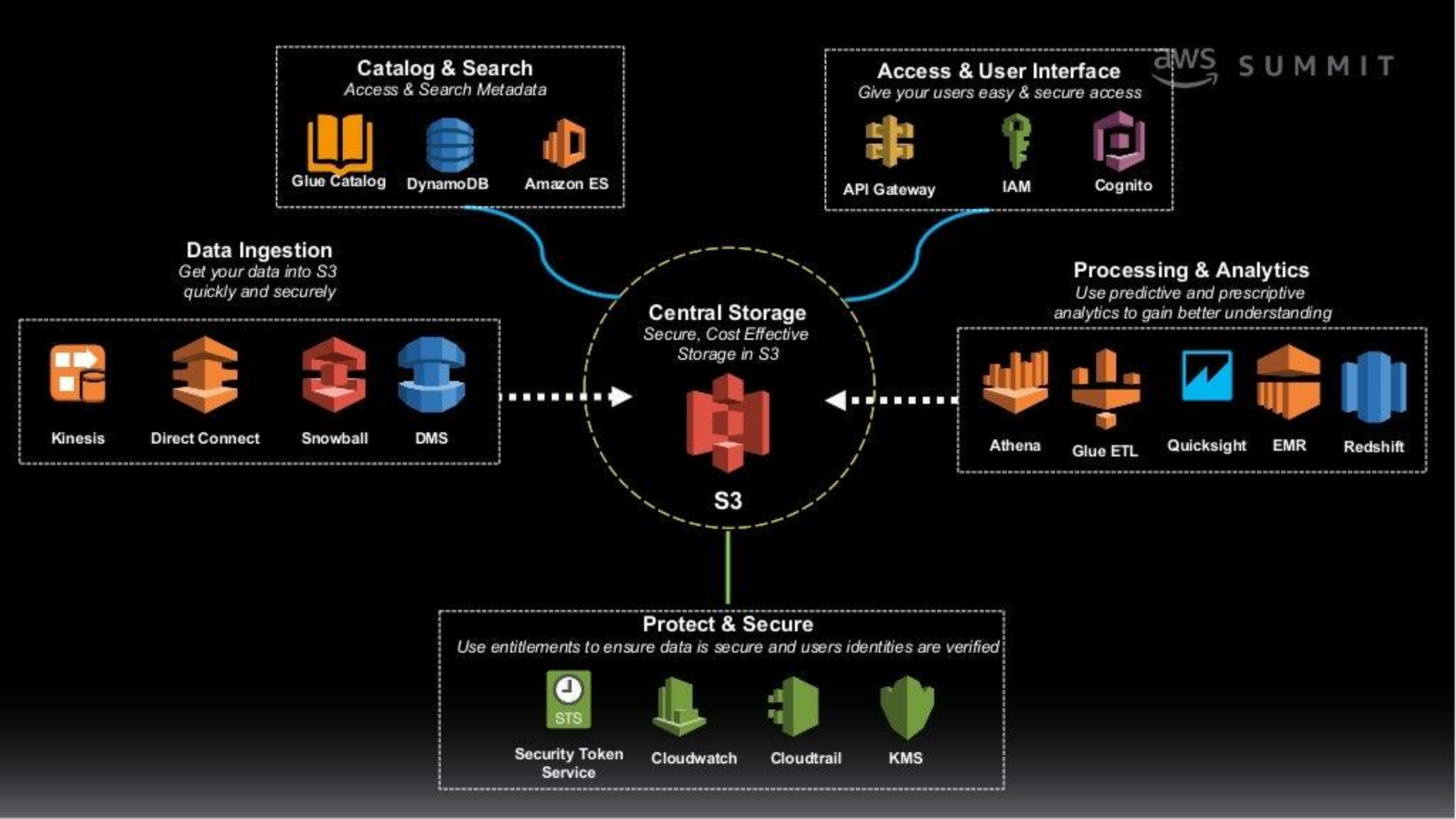
Architecture



Take the demo home…

aws SUMMIT

http://bit.ly/sg-summit-datalake-demo


Catalog & Search
Access & Search Metadata
Glue Catalog
DynamoDB
Amazon ES
Access & User Interface
Give your users easy & secure access
API Gateway
IAM
Cognito
aws SUMMIT
Data Ingestion
Get your data into S3 quickly and securely
Kinesis
Direct Connect
Snowball
DMS
Central Storage
Secure, Cost Effective Storage in S3
S3
Processing & Analytics
Use predictive and prescriptive analytics to gain better understanding
Athena
Glue ETL
Quicksight
EMR
Redshift
Protect & Secure
Use entitlements to ensure data is secure and users identities are verified
STS
Security Token Service
Cloudwatch
Cloudtrail
KMS

aws SUMMIT

spuul

Daniel Muller
Head of Cloud Infrastructure, Spuul
danielmullerch

spuul

Watch unlimited
Bollywood Movies.
Anytime. Anywhere.

- Leading OTT player
- Indian Movies, Shows and Live TV
- 50 millions registered users
- Users in 5 continents
- Content served on Mobile, connected TVs, SetTop Boxes
- (Coming Soon..!) Non-Indian content

Why we built a serverless data lake ?

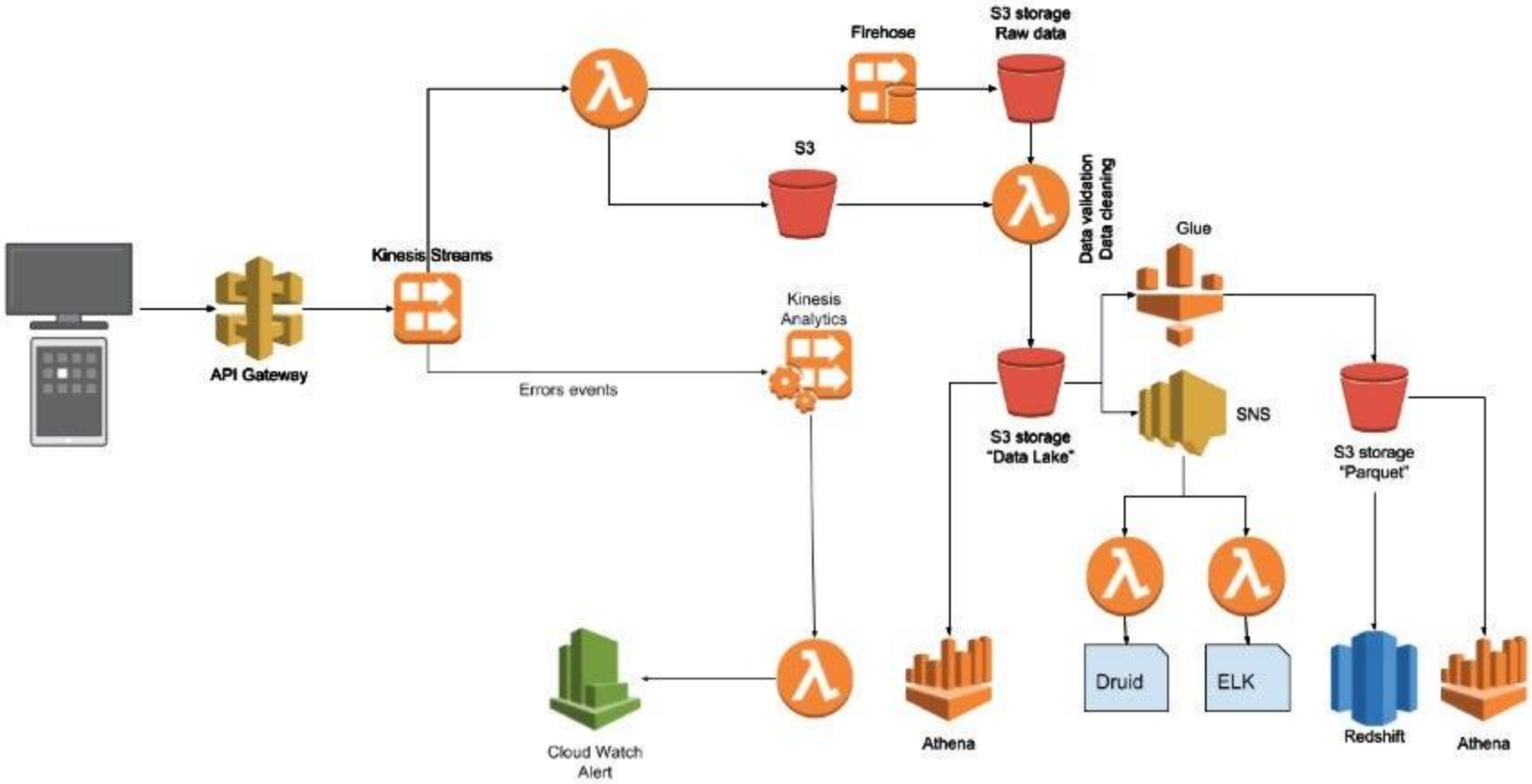
aws SUMMIT

- 100+ event types - across microservices & devices
- Flexibility - Ingestion, Consumption
- Bottomless storage - Cheap & Reliable
- Ad-hoc querying – Analyze data without a data-warehouse
- Future use cases - 3rd party integrations
- We hate managing servers..! #NoMoreServers

spüul

Architecture

spuul aws SUMMIT



Lessons Learnt & Best Practices

aws SUMMIT

Use a framework - SAM, Serverless, Apex/Up

Store data in raw format - debugging and re-processing

Convert to Columnar Formats - Optimized for reads

Partition data - Based on your filters

Specify columns to load - Reduce data transfer

Create files of ~100MB - Reduces S3 list API calls

Compress Data in Lake – Reduce network transfers

Use Lambda for Automation – Wire things together

spuul

Summary

aws SUMMIT

Thank You !

Take the demo home

http://bit.ly/sg-summit-datalake-demo

#AWSSummitSG
#NoMoreServers
#DataLake

unni_k_pillai

aws SUMMIT

Thank You

You will receive today's webinar recording and presentation deck, look out for it in your inbox.