Building Data Lakes in the Cloud

Enterprise Data Lake & Analytics
March 2017

Eric Anderson, Executive Director, Data  Shyam Konda, Lead Data Engineer
Who We Are

Eric Anderson, Executive Director, Data
- Lead Beachbody Data organization, which includes BI, Data Warehousing, Big Data and Email/CRM Platforms.
- Focused on providing Business Value in the shortest amount of time by leveraging a wide array of technologies to deliver outcomes quickly.
- Delivered on data initiatives for over 15 years with Accenture and Slalom Consulting, and worked for other technology companies in Los Angeles including TrueCar and Edmunds.
Contact: eanderson@beachbody.com or https://www.linkedin.com/in/ericsanderson1

Shyam Konda, Lead Data Engineer
- Well versed in writing sql and PLSQL code against relational databases like Oracle, SQL Server, Teradata, Mysql, Hive, Redshift, Postgres, Netezza, etc.
- Worked on cutting edge technologies like Hadoop, SQOOP, Hive, Pig, Hbase, Elastic Search, S3 and Amazon redshift.
- Strong experience in writing java code to customize data warehouse jobs & help build jobs that can be used as templates for code reusability, etc.
Contact: skonda@beachbody.com or https://www.linkedin.com/in/shyam-sundar-konda-15077418
The Challenge: Do More, Faster, Better, Cheaper

Revenue
Contribution Margin
Returns

Customer Count
EBITDA
Sales

Average Subscription Cycles
Streaming Customers

WHAT WE KNOW...

The Challenge: Do More, Faster, Better, Cheaper

BIG DATA

THE REST...

Need a repository to capture and store the massive amount of data we’re currently ignoring.

How can we personalize offers for each customer?

How can I better match customers with coaches to ensure longevity and mutual success?

What indicators determine if a subscription is about to cancel?

How can we better target and retarget customers leveraging customer purchase and behavioral data?
Project Vision – Open Enterprise Data Lake

- Build an OPEN Enterprise Data Platform

  - **Open Source Technology:** Bring Your Own Tool
  - **Decentralized Data Ownership:** Many teams can publish
  - **Centralized** people, processes & tools available
  - **Capture All Data** as real-time as possible
  - **Access to All raw + processed data** by Authorized Users
  - **HIPAA/PII encrypted** or masked to for compliance
  - **Shift Time** from Collecting Data -> Analyzing Data
People, People, People

Technology alone wasn’t enough

- Hire Talent with related experience
- Train existing Talent in new technologies
- Train stakeholders to leverage the platform to help drive value
- Retain key talent longer with interesting work
- Hire talent in all areas to deliver value (Dev, Dev Ops, Analysts, Scientists)
Technology was Also Important
Data Lake Architecture

- AWS S3 Data Lake folder structure
  - Top-level folders for decentralized ownership
  - Sub-level folders:
    - Raw and processed data
    - Time series data loading
Architectural Design

1. Storage
   - Glacier Cold Storage
   - S3 Data Lake
   - Cross Region Replication: us-west-2/us-east-1

2. Data Pipeline
   - Compute Cluster
     - Spark
       - HDFS-KMS
   - App Servers (Talend, Tableau, etc.)

3. RDBMS
   - RDBMS Engine
     - Amazon RDS
     - AWS-KMS
     - VPC Security Groups

4. Compute
   - AWS-KMS

5. Analytics
   - S3 Data Lake
   - Public Internet
   - External Source Systems
   - Self-Service Client Tools
   - Internal Source Systems
   - AWS Console/Command Line Users
   - AWS Direct Connect

Additional Notes:
- SSE-S3/SSE-KMS
- Region: us-west-2
- VPC Security Groups
- Cross Region Replication
- AWS Console/Command Line Users
- SAML/SSO
Data Lake Component - Storage

- Amazon S3
- All data is encrypted at rest
- Five categories:
  - **Raw**
  - **Pre-Processed** - for efficient consumption
  - **Processed** - Curated data with business rules applied
  - **Sensitive** - Encrypted zone
  - **Work** – Sandbox for projects

SSE-S3 / HDFS-KMS

SSE-KMS / HDFS-KMS
Data Lake Component – Data Pipeline

- Executes processes natively in Talend with Push-Down
- Leverages cluster security
- Will be used to:
  - Ingest & land raw data
  - Transform raw data
  - Orchestrate workflow for data science & analyses
  - Load data into RDBMS

Vendors were compared by capabilities across 6 categories
Data Lake Component - RDBMS

- Amazon Redshift or Presto for analytical database usage?
- Hive for ad-hoc queries against fine-grained data
- Data from “processed” storage is loaded into analytical DB
- Business users connect to RDBMS with data viz or query tools
- Will serve as Enterprise Data Warehouse platform
- All data encrypted at rest and in transit
Data Lake Component - Compute

- Hortonworks HDP used for persistent cluster
  - Available 24/7 to business user community
  - Recent versions of Hadoop ecosystem components
  - Open-source platform tracking closely to Apache releases
- Amazon EMR used for transient clusters
  - Experimental and isolated workloads
  - Optimized for rapid cluster creation and tear-down

Vendors were compared by capabilities across 6 categories
Data Lake Component - Analytics

- Spark compute engine with Machine Learning libraries
- Data science models are run in Spark engine
- Python and Scala used for programming
- Results are landed back into storage for further action
- Data pipeline leverages Spark engine for transformations
- Spark engine encrypts data at rest and in motion
Business Benefits

- Reduced Data Acquisition time by 5x
- Improved Marketing Campaigns
- Reduced site tagging costs
- Improved employee retention and satisfaction
- Automated Customer Self-Service Order status
- Identified Web Funnel Conversion Opportunities (testing now)
Process Flow for Sqoop Ingestor

1. Run create sqoop metadata job
2. Read metadata row with jobid
3. Initialize context variables
4. Check if Hive table exists?
   - Yes: Sqoop load to S3
   - No: Sqoop with Hive
5. Change location to S3
6. Delete local files
7. Alter to external
8. Add Partition
9. Upload to S3
sqoop import --connect 'jdbc:oracle:thin:@00.00.0:1521/mydb' --username user1 --password xxxxxxx --query "select * from TARGET_OWB2.WC_RET_ATTAIN_DT where trunc(bi_update_date) > DATE '2017-02-28' AND \$CONDITIONS" --target-dir 's3a://<>/accesskey:<secret-key>@mybucket/raw/edw/snap/table/wc_ret_attain_dt/year=2017/month=02/day=27' --compress --compression-codec org.apache.hadoop.io.compress.GzipCodec --fields-terminated-by ',' --null-string '\N' --null-non-string '\N' -m1

Example Metadata Columns

- jobid,
- jobname,
- s3_credentials_file,
- database_type,
- jdbc_string,
- db_username,
- db_password_file,
- sqoopsql1,
- sqoopsql2,
- load_type,
- hivetableexists,
- hive_table_name,
- Hive_partition_key
- hive_partition_value,
- split_by_columns,
- sqoophistoricallog,
- localpath,
- compression_codec,
- has_pii,
- s3bucket,
- s3_destinationpath,
- hdfs_destinationpath,
- s3_key_prefix,
- source_name
- Creates metadata rows based on the database types i.e., Oracle and Mysql
- This is a pre job that runs before looping through the data ingestor
- Database source name and number of threads are passed as parameters
- Successfully ingested an oracle schema with 285 tables using five parallel threads in six hours
Data in S3 and Hive

- Business users and Analytics team able to utilize the data loaded to create their own managed
- Hive tables or access the data using tools like Tableau and Alteryx directly
- This is a branch of generic loader which is repeated for different cases
- Used historical log file to avoid duplicate processing of data
Continuous Delivery Integration

Automated software development processes

Plan → Code → Build → Test → Release → Deploy → Operate

Continuous Integration

Continuous Delivery

Accelerates time-to-market. Increased productivity.
Continuous Integration

Jenkins  >  Talend_cmd_ci_qa  >  

Build Environment

Invoke top-level Maven targets
Maven Version  (Default)
Goals  org.talend.ci.builder:6.2.1:generate

Invoke top-level Maven targets
Maven Version  (Default)
Goals  clean test package

Invoke top-level Maven targets
Maven Version  (Default)
Goals  deploy -fn -e

Execute shell
Command  sudo -u talenduser rm -r /opt/talend_621/cmdline_621/cmdline_621_ws/*
Continuous Deployment

- Automate Talend task creation in Talend Administration Console (TAC)
- Built Jenkins workflow which connects to Talend Administration Console (TAC)
- Create Talend tasks using groovy script which calls Talend meta servlet
- Schedule the tasks
- Pass task parameters in JSON format
- Create workflow dependency with Continuous Integration Jenkins workflow
Lessons Learned & Best Practices

- Used Cloud Formation templates to spin up Hadoop clusters and Talend servers
- Created data lake inventory confluence page to provide basic information on data ingestor jobs.
- Created AMIs for Talend and Hadoop servers with prebuilt features
- Using AWS services helped speeding up the Infrastructure and decreased dependency on other teams
Lessons Learned & Best Practices Cont.

- Used Activity Monitoring Console (AMC) in Talend to capture stats and logs
- Implemented Splunk logging on Talend and Hadoop to capture activity logs and runtime logs
- Continuous integration & deployment helped team deploy Talend jobs easily with minimal resources
- Single Talend job can support data sources with structured & unstructured data
Questions?