Building a Production Data Lake in the Cloud

Prat Moghe
Founder and CEO
Cazena
Agenda

• Defining Data Lake
• Cloud vs. on-Premises
• 6 steps to production-ready Cloud Data Lake
• Demo
Defining Data Lake

Why Data Lake?

- Agility – Need data to drive new products
- Capture dark or external data
- Surface and discover dark or external data
- Query, transform and extract to pipeline

Definition

- Data is the value (Data capture and ingest)
- Staging area yet actionable
- BUT NOT an EDW!
On-premises Data Lakes have Huge Drag

Top 3 Challenges

- **Infrastructure slows you down**
  - 6-9 months to procure and deploy
  - Iteration hard if you get it wrong

- **Stack and technology complexity**
  - Which technology – what’s the half life? How do you POC?
  - Tremendous churn.
    Open source projects exploding.
    Expertise limited.
  - Pipeline connectivity inflexible

- **CAPEX and Utilization**
  - Have to buy upfront for next 3-5 years
  - Built-in headroom for 20-30% growth – wasted capacity
  - Enterprise DWs and clusters – 50% idle
Cloud Promises Agility, but Lots of Questions…

- Project scoping: where do you begin?
- Provisioning is still a black art.
- How do you integrate cloud with your data flow?
- How to integrating security, compliance controls into policies
- How to monitor workload and infrastructure and ensure SLA?
- How do you build a pipeline connecting your data lakes, warehouses and tools?
6 steps to Production Cloud Data Lake

1. Planning
2. Provisioning
3. Enterprise Integration
4. Security, Compliance and Governance
5. SLA - Processing/Analytics
6. Pipeline around the Data Lake
#1 Planning

- **Which cloud provider(s)?**
  - Approved vendors
  - Economics vs. on-premises
  - Coverage
  - Stack capabilities
  - Price/performance

- **Build direct – DIY?**
  - How long?
  - Expertise shortage
  - One-off vs. Platform

- **Build on Big Data as Service platform**
  - Cloud agnostic
  - Automated
  - Managed
# AWS and Azure – Parallel Building Blocks

<table>
<thead>
<tr>
<th></th>
<th>Azure</th>
<th>AWS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compute</strong></td>
<td>Virtual Machines</td>
<td>Elastic Compute Cloud (EC2)</td>
</tr>
<tr>
<td><strong>Object Storage</strong></td>
<td>Azure Blob Storage</td>
<td>Amazon Simple Storage (S3)</td>
</tr>
<tr>
<td><strong>Disk Storage</strong></td>
<td>Azure Page Blobs and Disks</td>
<td>Elastic Block Storage (EBS)</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td>Virtual Network</td>
<td>Virtual Private Cloud (VPC)</td>
</tr>
<tr>
<td><strong>Key Management</strong></td>
<td>Azure Key Vault</td>
<td>AWS Key Management Service</td>
</tr>
</tbody>
</table>
#2 Provisioning - Workload and Nodes

- It all begins with the workload
  - Tools for analysis
  - Storage vs. Compute
  - Archiving, Discovery, Querying, Batch or Interactive Analytics
  - Concurrency
  - Data ingest cycle requirements
- Cost – loaded cost
- How many nodes? What types?
- Keep up with changing node types

![40+ choices of Amazon machines!](image)
#2 Provisioning – Workload and Nodes

- Bigger nodes are not always better – there is a sweet spot
- Which is better: c4.large or c4.xlarge?

<table>
<thead>
<tr>
<th>Instance</th>
<th># cores</th>
<th>RAM</th>
<th>EBS Pipe (mbps)</th>
<th>Cost ($/hour)</th>
</tr>
</thead>
<tbody>
<tr>
<td>c4.large</td>
<td>2</td>
<td>3.75</td>
<td>500</td>
<td>0.11</td>
</tr>
<tr>
<td>c4.xlarge</td>
<td>4</td>
<td>7.5</td>
<td>750</td>
<td>0.22</td>
</tr>
</tbody>
</table>

- In this case, I/O scales up 50% for 100% cost
- Pick c4.large if your workload is driven by EBS I/O
- Pick the best price/”balanced performance” ratio for YOUR workload
#2 Provisioning - Workload and Technology

- Which technology? Which vendor?
  - Hadoop (catch all)
  - MPP SQL
  - Spark

- Pipeline architecture?
  - Centralized
  - Distributed (Hub and Spoke)

- Storage architecture
  - Persistent object store vs. instance store? Price/performance impact
  - Data collection, movement and ingest/extract architecture critical
#2 Performance graphs can be misleading

- 61 queries on 100 GB of data for both Impala and Redshift.
- Clusters had similar numbers of CPUs (16 nodes, 2 vCPUs each) and storage, but the **Redshift cluster had 4x the RAM**.
- Both technologies run at interactive speeds - except for **RAM intensive jobs**. Providing Impala with additional RAM would result in huge performance gains.
#2 Provisioning - Workload and Technology

• The real experience is driven by actual workload and tool support
  – ANSI SQL?
  – UDF?
  – What happens when queries run out of memory? Performance cliff?
  – Concurrency

• Configuration Management
  – Lots of knobs for each distribution plus workload management
  – Even if you get them right, have to continually tune depending on usage
  – Packaged scripts or management tools are a good way to start, but may not offer end-to-end scope
  – Ultimately you have to configure end-to-end infrastructure – including data ingest, storage, cluster, tools etc.
#2 Provisioning - Storage Performance

- Object store vs. instance store
- Price/performance impact on Data Lake

<table>
<thead>
<tr>
<th></th>
<th>Monthly $/1TB*</th>
<th>Guaranteed Throughput (MBps)</th>
<th>Consistency</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS S3</td>
<td>$30</td>
<td>No guarantees (~40)</td>
<td>Eventual</td>
</tr>
<tr>
<td>AWS Magnetic EBS</td>
<td>$150</td>
<td>62 – 500 per instance</td>
<td>Strict</td>
</tr>
<tr>
<td>Azure Blob</td>
<td>$30</td>
<td>60</td>
<td>Strict</td>
</tr>
<tr>
<td>Azure Premium Disks</td>
<td>$369</td>
<td>100 – 200 per disk</td>
<td>Strict</td>
</tr>
</tbody>
</table>

* Dollars per 1 TB, replicated three times.
#3 Enterprise Integration: Data Movement/Ingest

- **Data Sources**
  - Enterprise/Cloud/Third party
  - Structured, logs/files, batching/streaming

- **Data Movement**
  - Reliability
  - Compression – impact on pipe
  - Ingest Rate
  - Extract back to enterprise sources (DW)
  - Ease of use: UI driven
  - Flexibility: REST API driven by enterprise workflow
#3 Enterprise Integration: Data Movement/Ingest

- **Schema management**
  - Logs/files to data lake, associate meta data
  - Store in the “right” format
- **Monitor status**
  - What are failure scenarios?
  - Can you recover?
  - Undo?
- **Often underestimated – takes a long time to get right!**
#3 Architect Ingest with Parallelism

- General Principle
- Use parallelism end-to-end
- Move data from object store to Hadoop worker nodes
- Compress
- Convert to the right format

```sql
• CREATE EXTERNAL TABLE my_csv_table (a int, b string) LOCATION 's3a://PATH_TO_CSV/';
• SET COMPRESSION_CODEC=snappy

• CREATE TABLE parquet_table LIKE my_csv_table STORED AS PARQUET;
• INSERT INTO parquet_table SELECT * FROM my_csv_table;
```
#4 Security & Compliance

- Cloud has plenty of security controls.
- Lock down the cloud – single tenant VPC
- Encrypt data going in and out, encrypt data at rest
- Role segregation
- Audit and Log everything….export to enterprise
- Compliance – treat lake like any enterprise asset
#5 Monitoring Workloads and SLA

- Need end-to-end monitoring of:
  - Health
  - Performance
  - Dashboards and logs for each platform can create confusion
  - Dashboard for the entire pipeline
- On-going SLA optimization
  - Capacity issues
  - End-user environmental changes
  - New tools and technologies
#6 Data Lake feeds Pipeline

- Data Lake is the “start” of data lifecycle – attach it to a pipeline
- Data Marts & Sandboxes
- Customer Lake/Marts
- Extraction to enterprise
- Single platform for diverse workloads
#6 Pipeline vs. centralized

- Distributed scales better as users grow
- Easier workload management
- Not one size fits all – Allows exploitation of technology sweet spots and best overall price performance – reduces vendor lock-in
- Data needs to be copied but only what is needed
- Ultimately more agile (lessons from 30 years of EDW)
#4 Pipeline provides governance control

- Strike balance between agility and control
- Instantiate Data Lake with set of users (X)
  - IT(X)
  - Data(X)
  - Analysts(X)
- Spawn marts and sandboxes with other users (Y)
  - Either IT(X) and Data (X) in control
  - or not….  
- Pipeline knob offers governance control
Cazena Demo: Big Data as a Service

• Future-proofed
  – Abstract cloud & workload engines – pick best of breed cloud & technology for all workloads

• Simple
  – End-to-end platform includes data movement/ingest + processing

• Integrated
  – Access cloud within an enterprise security tunnel

• Optimized
  – Continual optimization for best SLA
Thanks!

Prat Moghe
Founder and CEO
Cazena