Databricks' Data Pipelines: Journey and Lessons Learned

Yu Peng, Burak Yavuz

07/06/2016

databricks

Who Are We

Yu Peng

Data Engineer at Databricks

Building Databricks' next-generation data pipeline on top of Apache Spark

BS in Xiamen University

Ph.D in The University of Hong Kong

Burak Yavuz

Software Engineer at Databricks

Contributor to Spark since Spark 1.1

Maintainer of Spark Packages

BS in Mechanical Engineering at Bogazici University

MS in Management Science & Engineering at Stanford University

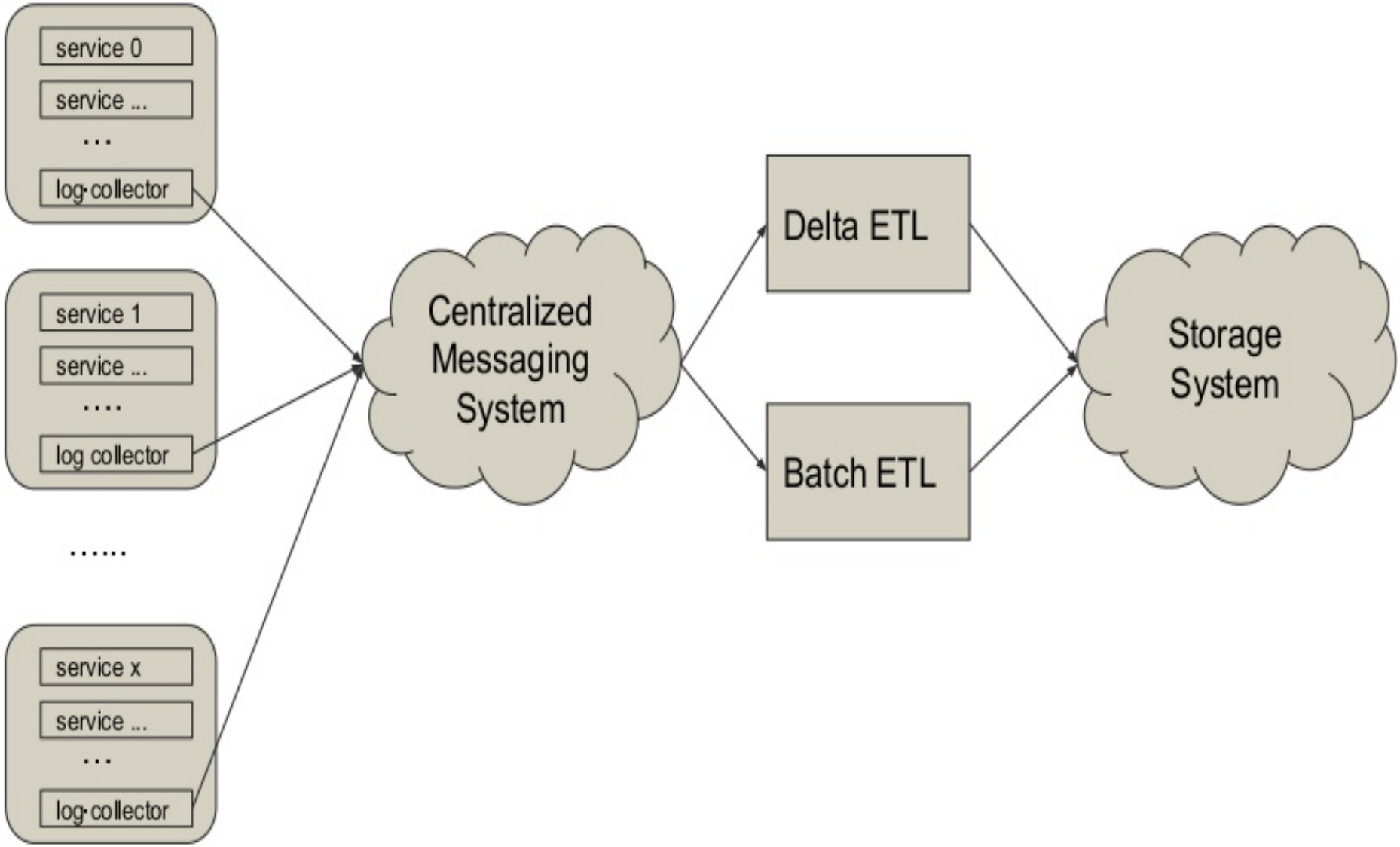
Building a data pipeline is hard

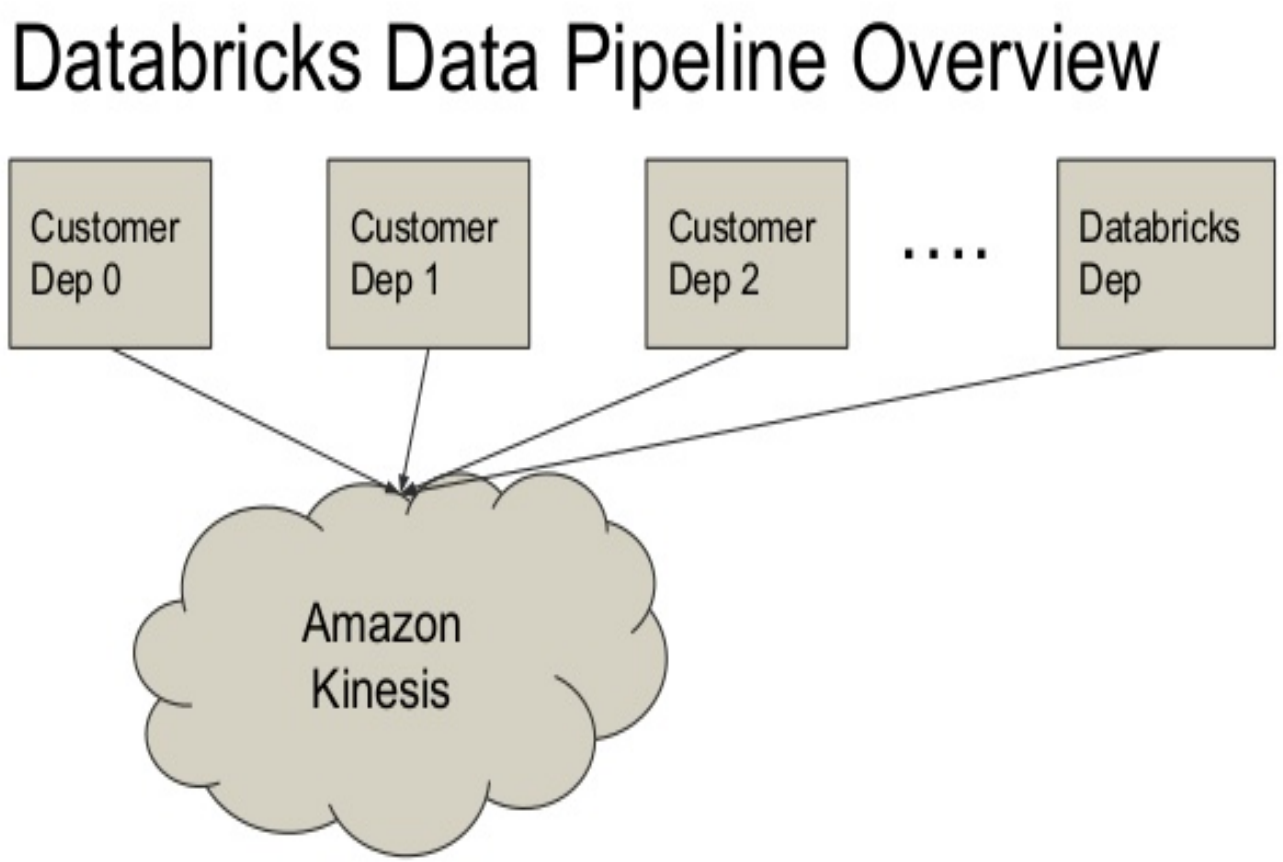
- At least once or exactly once semantics
- Fault tolerance
- Resource management
- Scalability
- Maintainability

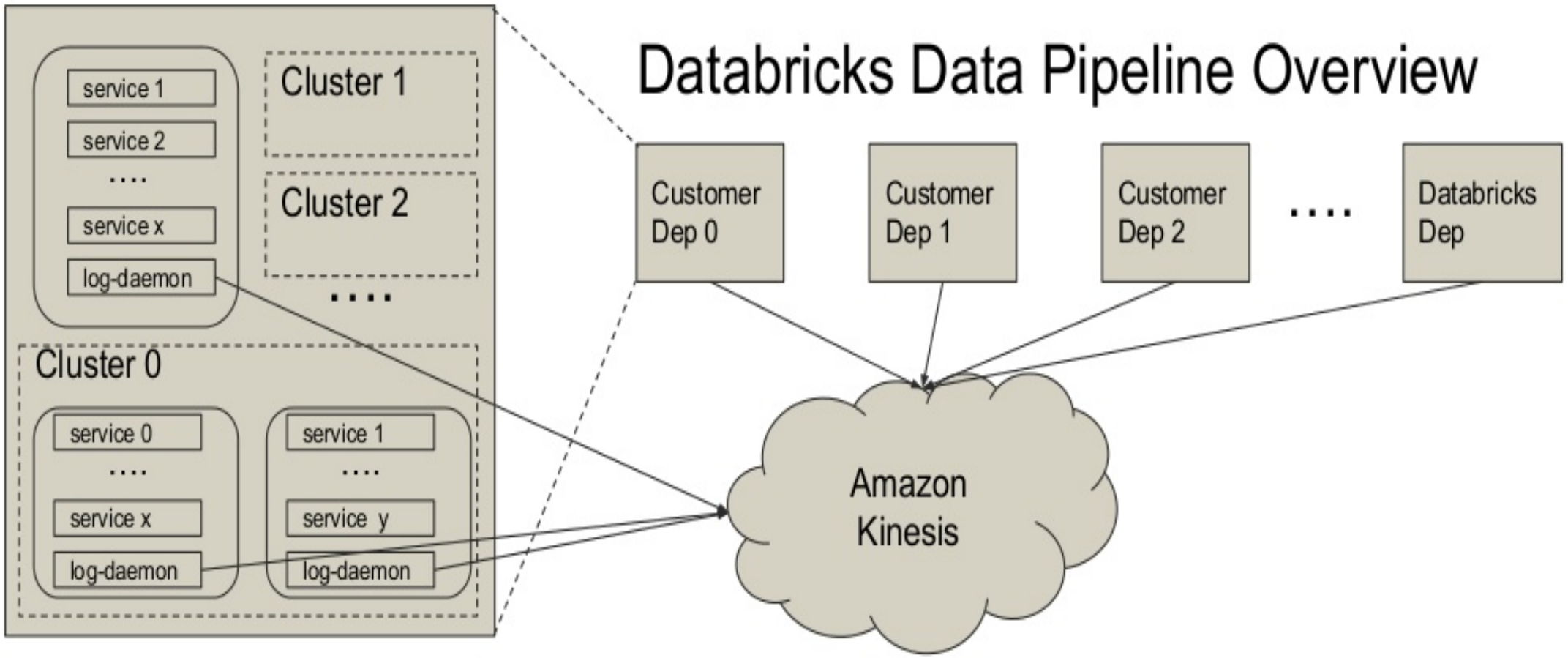
Apache® Spark™ + Databricks = Our Solution

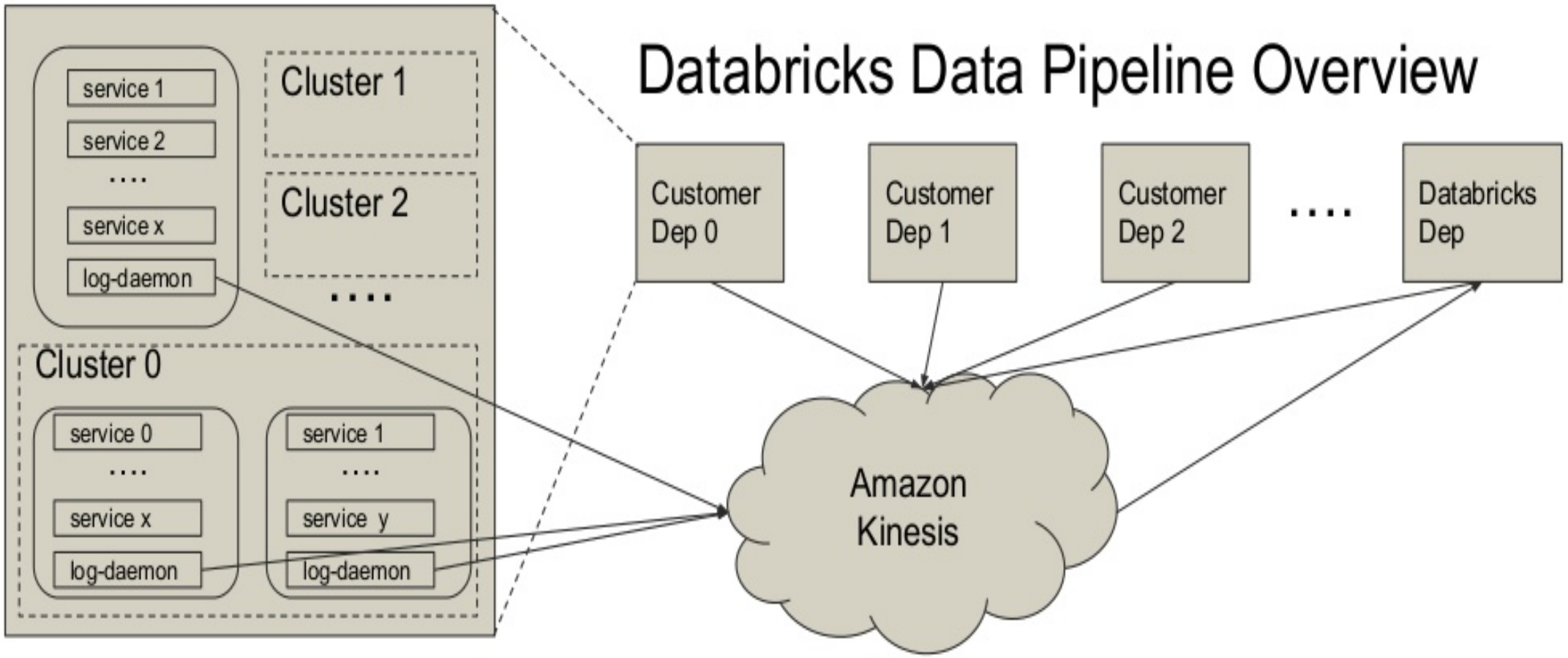
- All ETL jobs are built on top of Apache Spark
  - Unified solution, everything in the same place
- All ETL jobs are run on Databricks platform
  - Platform for Data Engineers and Scientists
- Test out Spark and Databricks new features

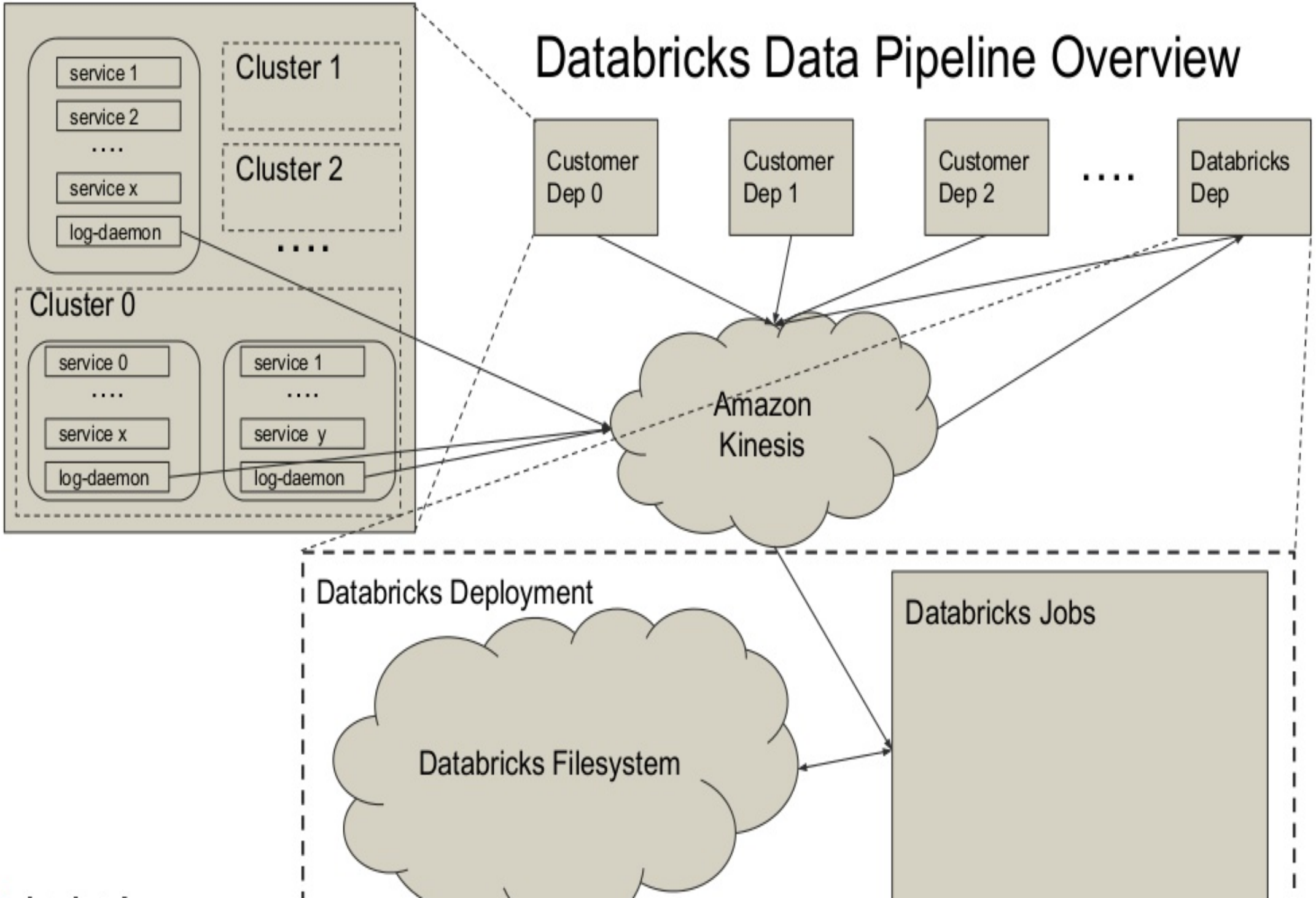
Classic Lambda Data Pipeline

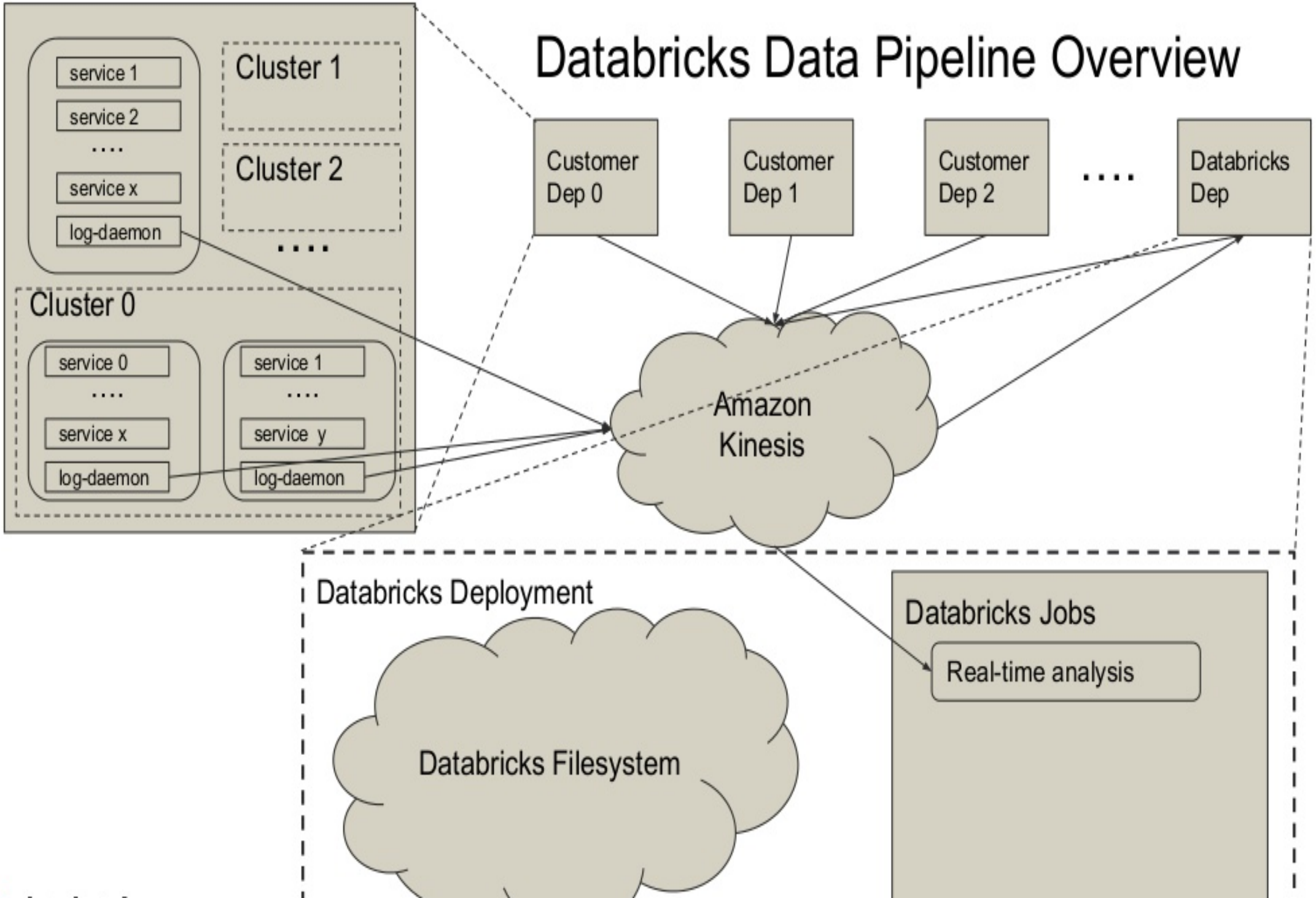


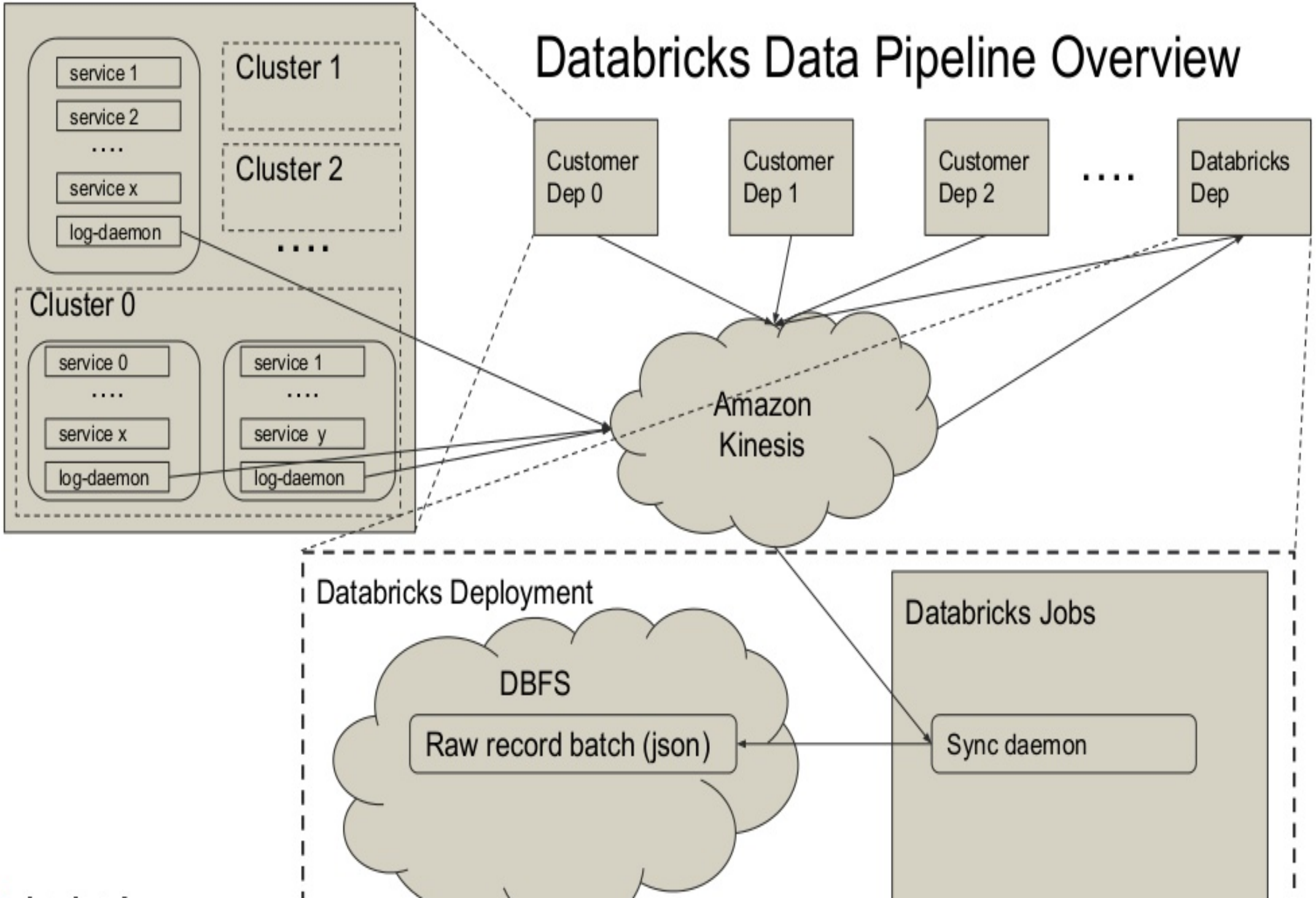

Databricks Data Pipeline Overview
Customer Dep 0
Customer Dep 1
Customer Dep 2
....
Databricks Dep
Amazon Kinesis

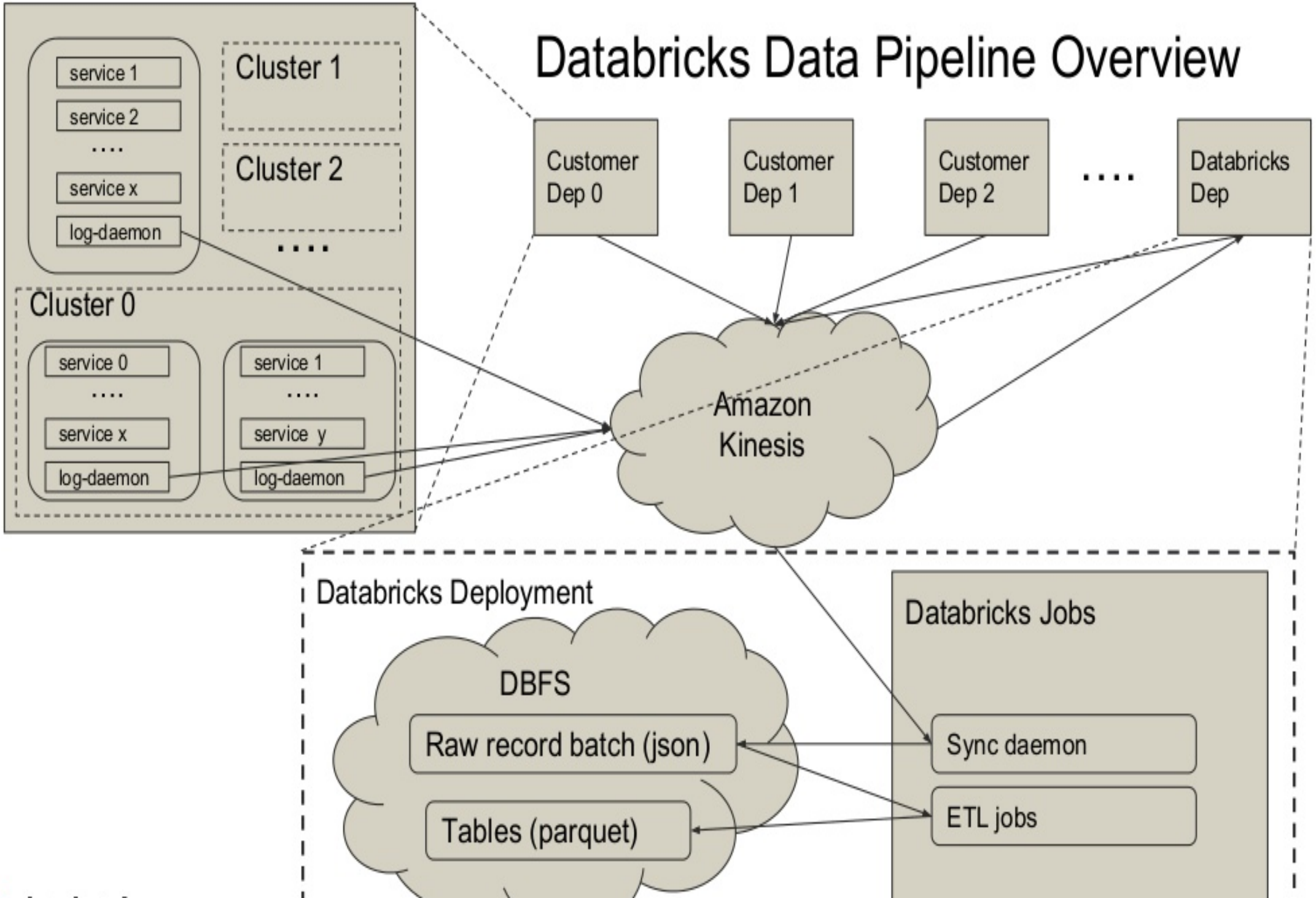

Databricks Data Pipeline Overview
service 1
service 2
....
service x
log-daemon
Cluster 1
Cluster 2
....
Cluster 0
service 0
....
service x
log-daemon
service 1
....
service y
log-daemon
Customer Dep 0
Customer Dep 1
Customer Dep 2
....
Databricks Dep
Amazon Kinesis

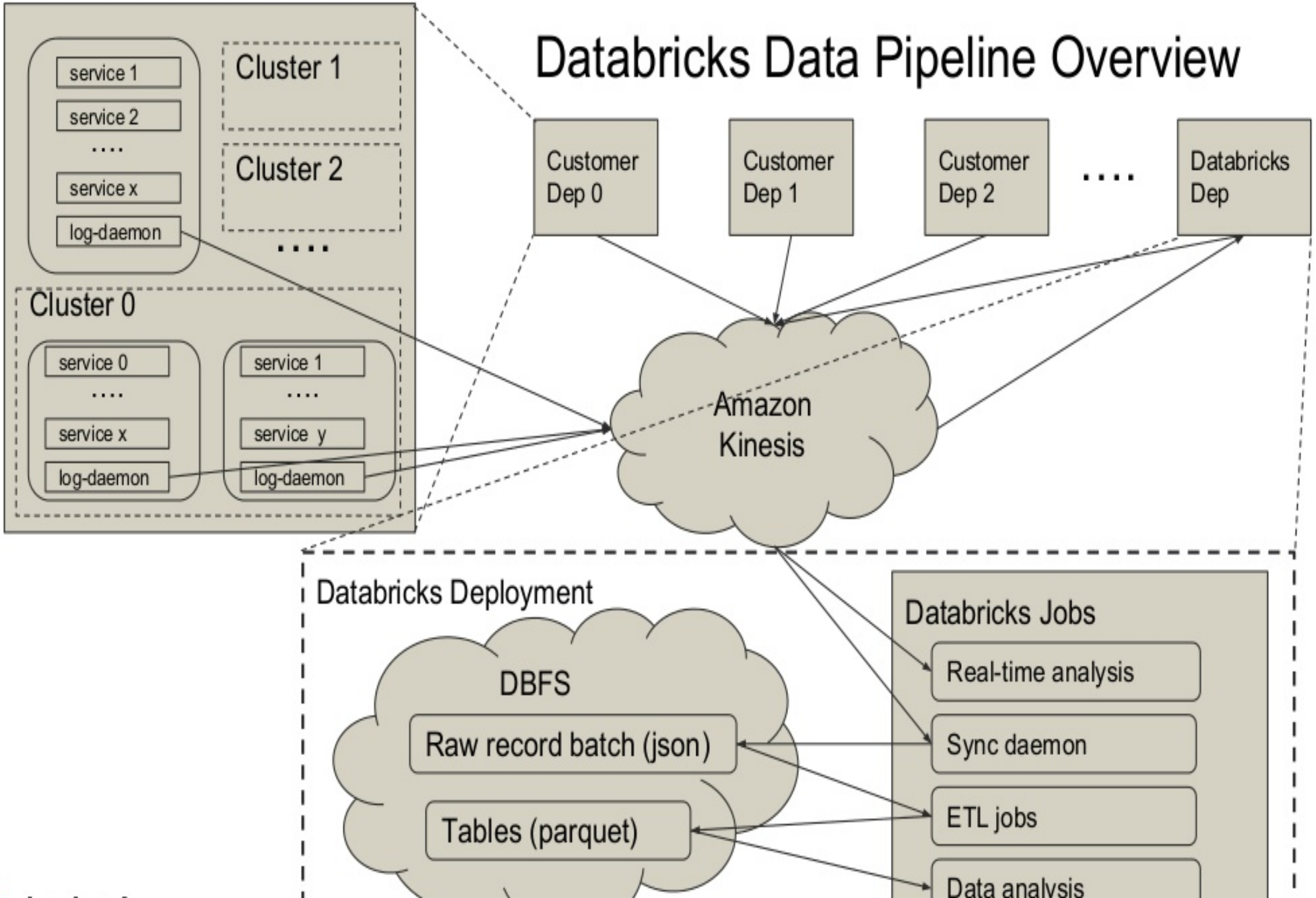

Databricks Data Pipeline Overview
service 1
service 2
....
service x
log-daemon
Cluster 1
Cluster 2
....
Cluster 0
service 0
....
service x
log-daemon
service 1
....
service y
log-daemon
Amazon
Kinesis
Customer
Dep 0
Customer
Dep 1
Customer
Dep 2
....
Databricks
Dep


Databricks Data Pipeline Overview
service 1
service 2
....
service x
log-daemon
Cluster 1
Cluster 2
....
Cluster 0
service 0
....
service x
log-daemon
service 1
....
service y
log-daemon
Customer Dep 0
Customer Dep 1
Customer Dep 2
....
Databricks Dep
Amazon Kinesis
Databricks Deployment
Databricks Filesystem
Databricks Jobs


Databricks Data Pipeline Overview
service 1
service 2
....
service x
log-daemon
Cluster 1
Cluster 2
....
Cluster 0
service 0
....
service x
log-daemon
service 1
....
service y
log-daemon
Customer Dep 0
Customer Dep 1
Customer Dep 2
....
Databricks Dep
Amazon Kinesis
Databricks Deployment
Databricks Filesystem
Databricks Jobs
Real-time analysis


Databricks Data Pipeline Overview
service 1
service 2
....
service x
log-daemon
Cluster 1
Cluster 2
....
Cluster 0
service 0
....
service x
log-daemon
service 1
....
service y
log-daemon
Customer Dep 0
Customer Dep 1
Customer Dep 2
....
Databricks Dep
Amazon Kinesis
Databricks Deployment
DBFS
Raw record batch (json)
Databricks Jobs
Sync daemon

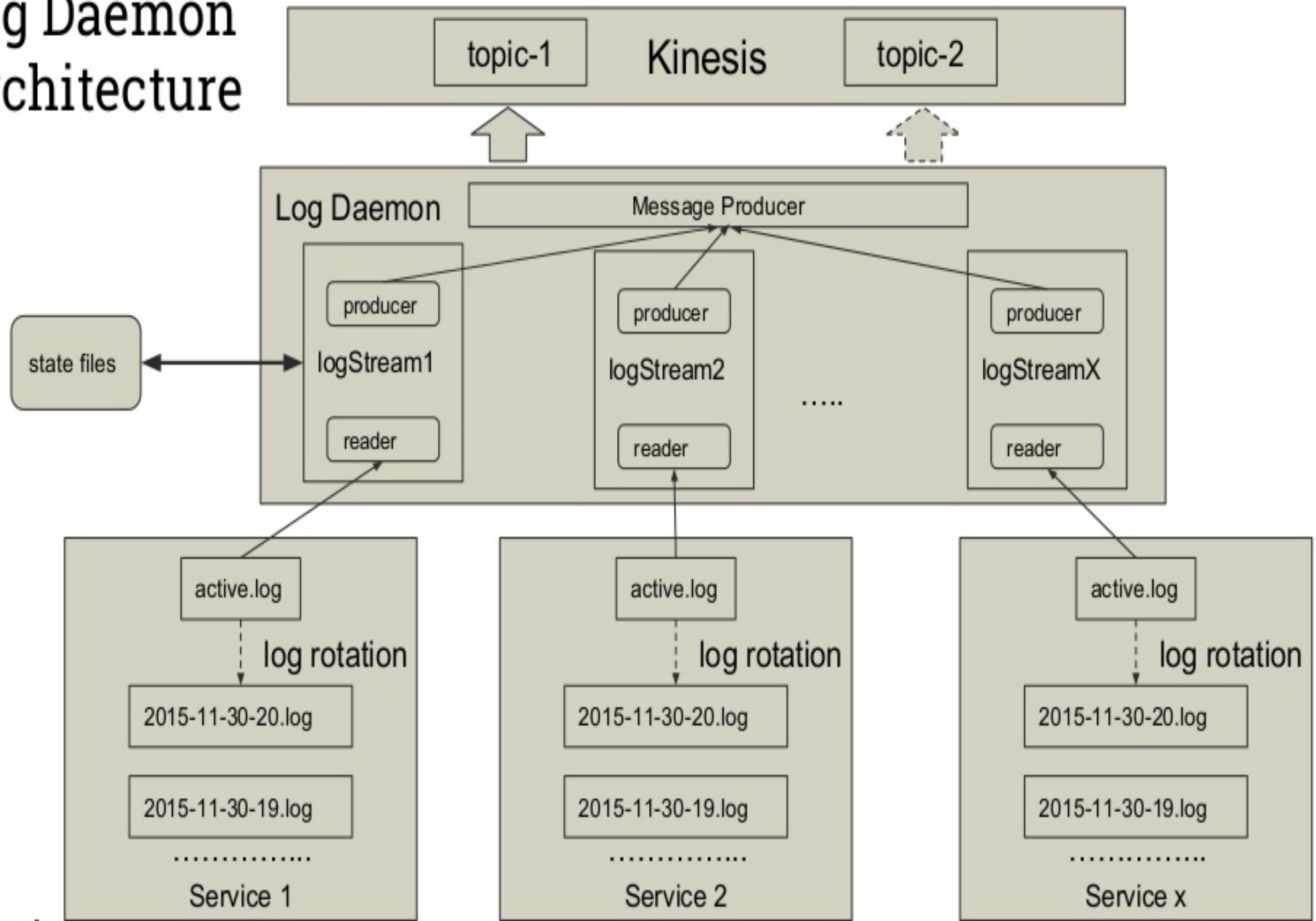

Databricks Data Pipeline Overview
service 1
service 2
....
service x
log-daemon
Cluster 1
Cluster 2
....
Cluster 0
service 0
....
service x
log-daemon
service 1
....
service y
log-daemon
Customer Dep 0
Customer Dep 1
Customer Dep 2
....
Databricks Dep
Amazon Kinesis
Databricks Deployment
DBFS
Raw record batch (json)
Tables (parquet)
Databricks Jobs
Sync daemon
ETL jobs


Databricks Data Pipeline Overview
service 1
service 2
....
service x
log-daemon
Cluster 1
Cluster 2
....
Cluster 0
service 0
....
service x
log-daemon
service 1
....
service y
log-daemon
Customer Dep 0
Customer Dep 1
Customer Dep 2
....
Databricks Dep
Amazon Kinesis
Databricks Deployment
DBFS
Raw record batch (json)
Tables (parquet)
Databricks Jobs
Real-time analysis
Sync daemon
ETL jobs
Data analysis

Log collection (Log-daemon)

- Fault tolerance and at least once semantics
- Streaming
- Batch
  - Spark History Server
- Multi-tenant and config driven
  - Spark container

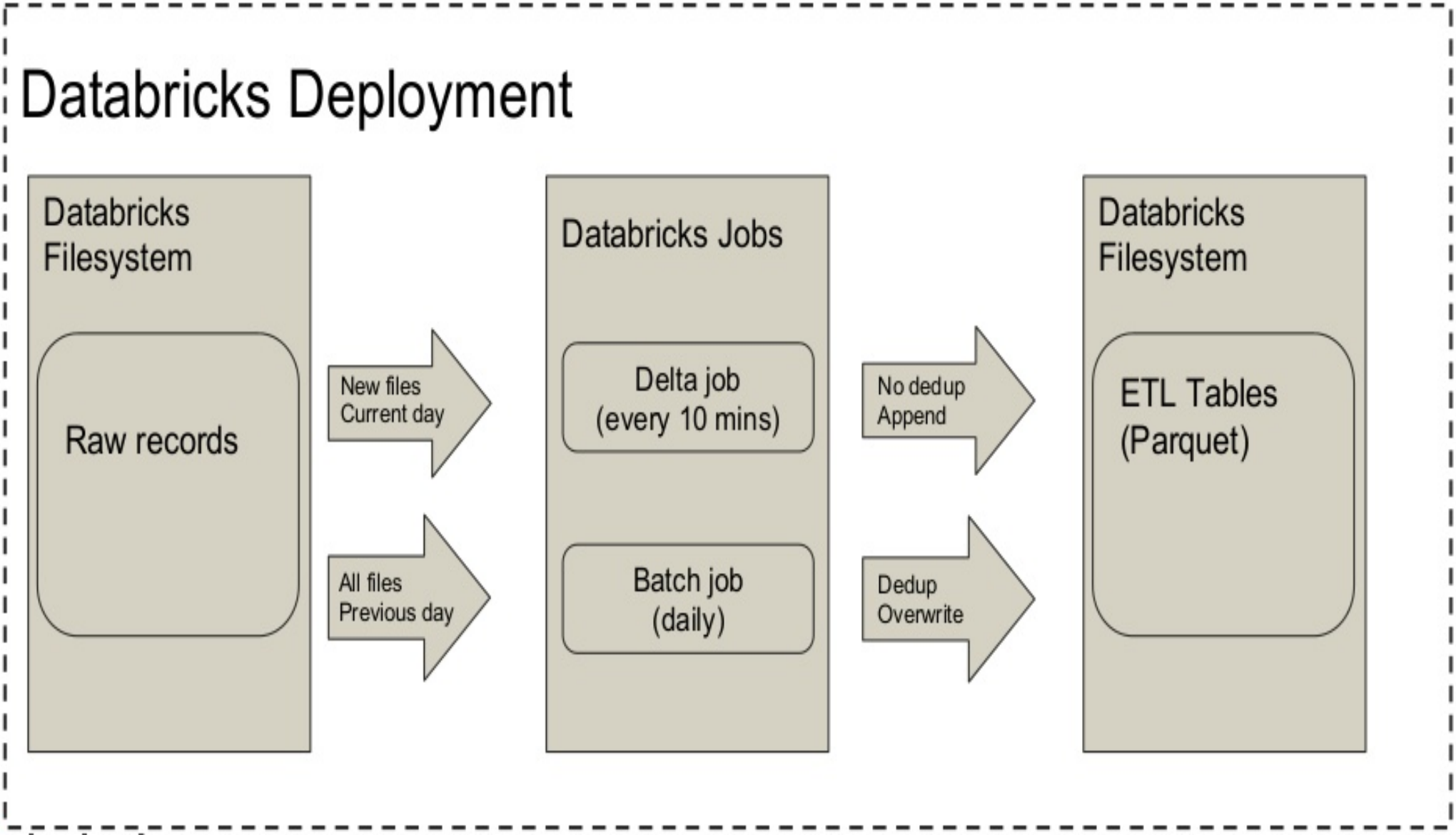
Log Daemon Architecture



Sync Daemon

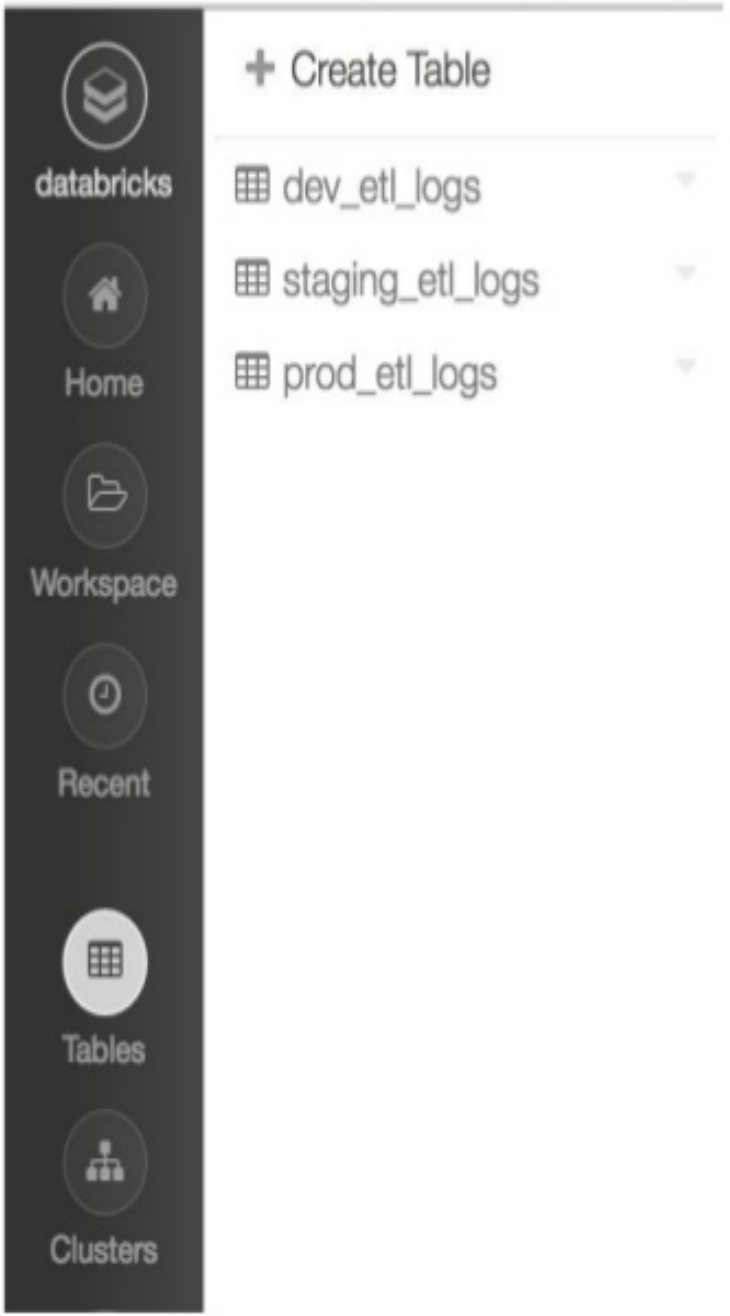
- Read from Kinesis and Write to DBFS
  - Buffer and write in batches (128 MB or 5 Mins)
  - Partitioned by date
- A long running Apache Spark job
  - Easy to scale up and down

ETL Jobs


Databricks Deployment
Databricks Filesystem
Raw records
New files
Current day
All files
Previous day
Databricks Jobs
Delta job
(every 10 mins)
Batch job
(daily)
No dedup
Append
Dedup
Overwrite
Databricks Filesystem
ETL Tables
(Parquet)

ETL Jobs

- Use the same code for Delta and Batch jobs
- Run as scheduled Databricks jobs
- Use spot instances and fallback to on-demand
- Deliver to Databricks as parquet tables



Lessons Learned

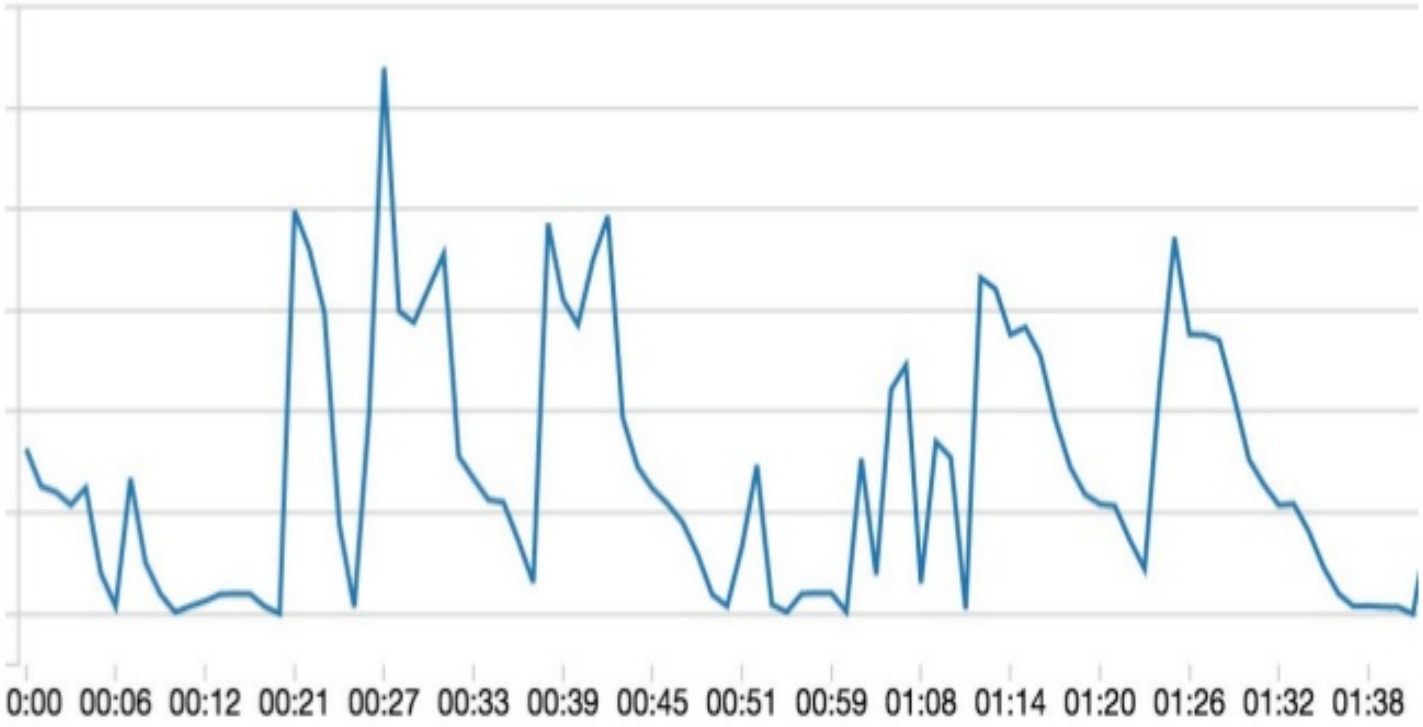
- Partition Pruning can save a lot of time and money

Reduced query time from 2800 seconds to just 15 seconds.

Don’t partition too many levels as it leads to worse metadata discovery performance and cost.

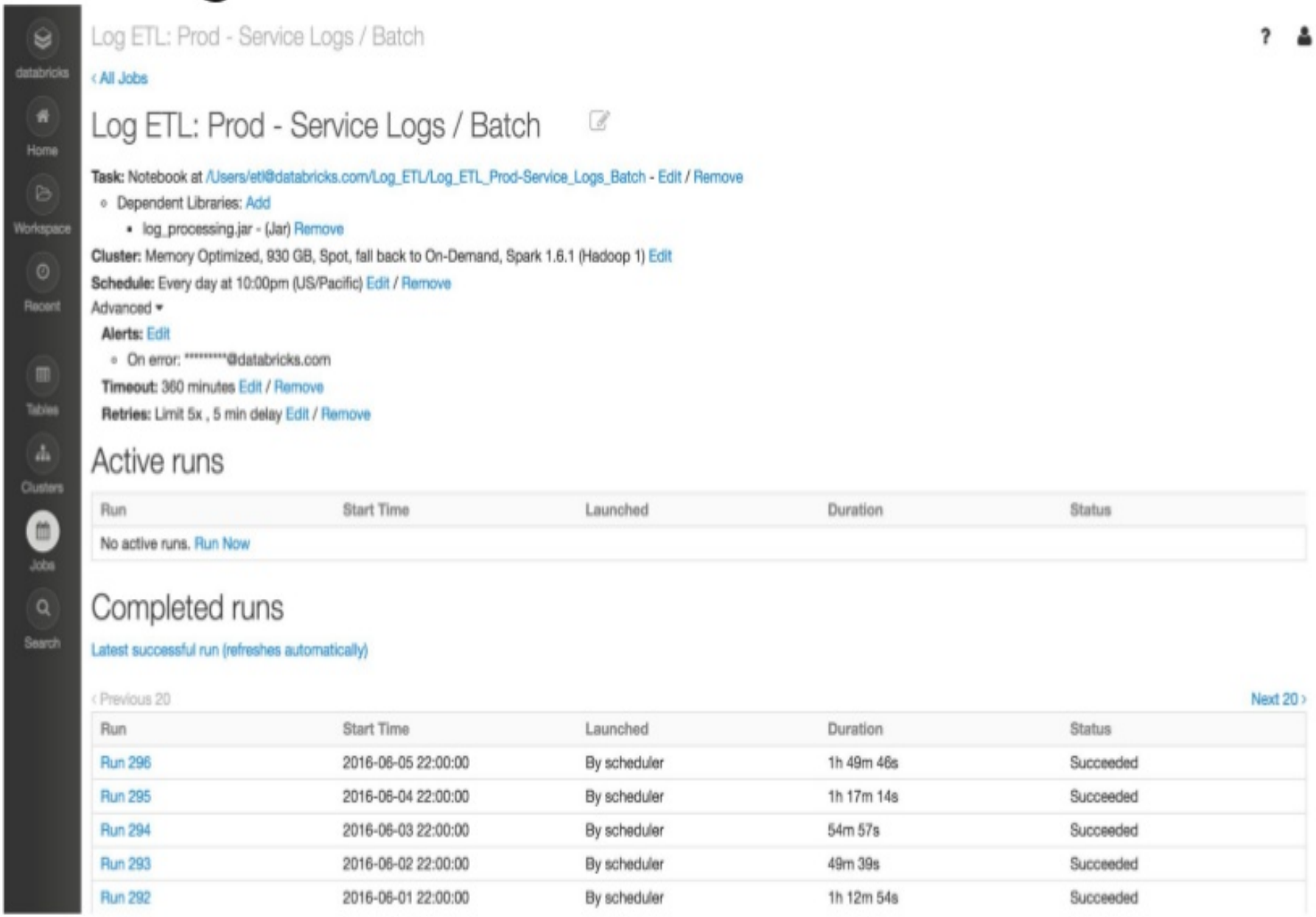
Lessons Learned

- High S3 costs: Lots of LIST Requests

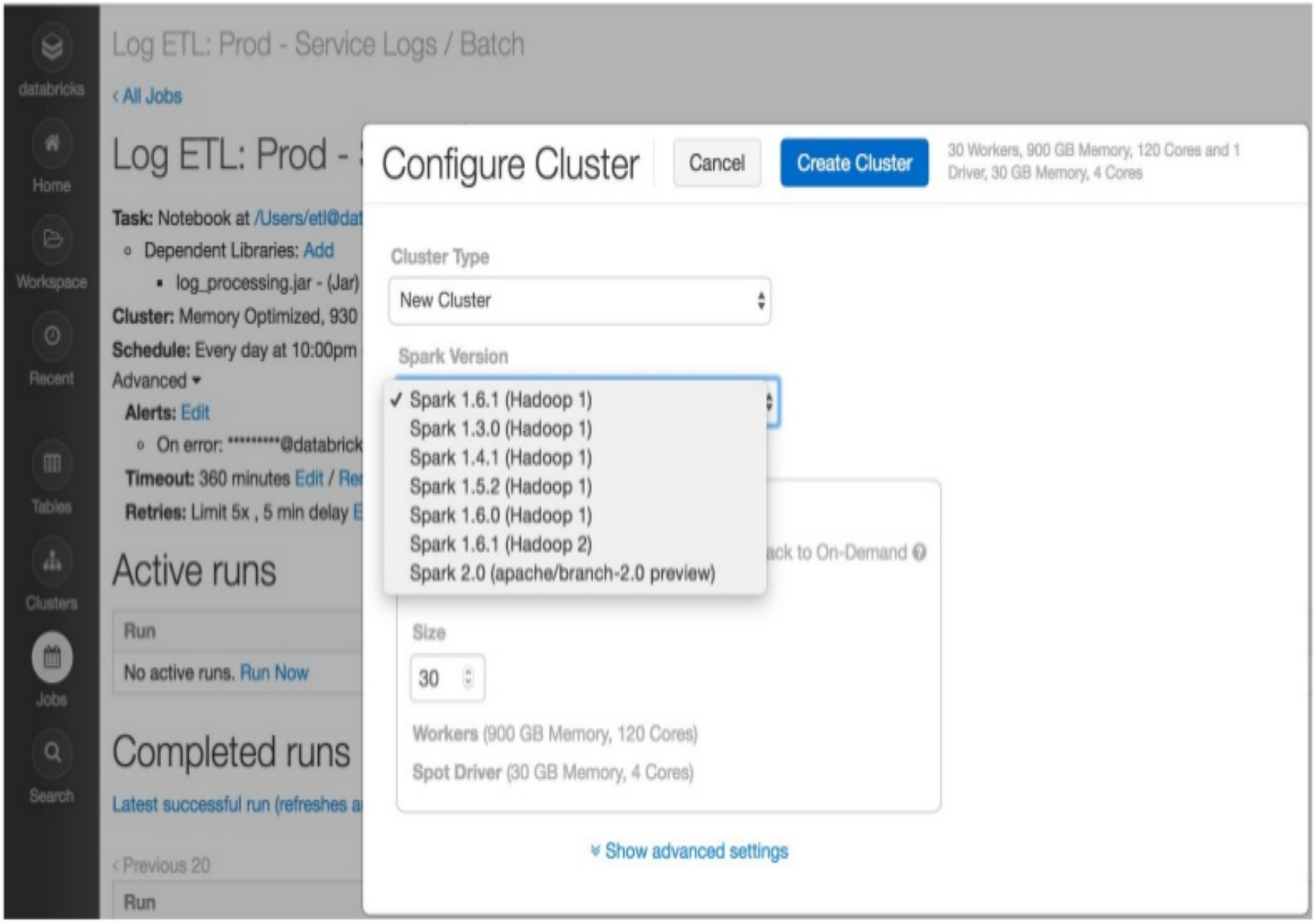


Metadata discovery on S3 is expensive. Spark SQL tries to refresh it's metadata cache even after write operations.

Running It All in Databricks - Jobs



Running It All in Databricks - Spark



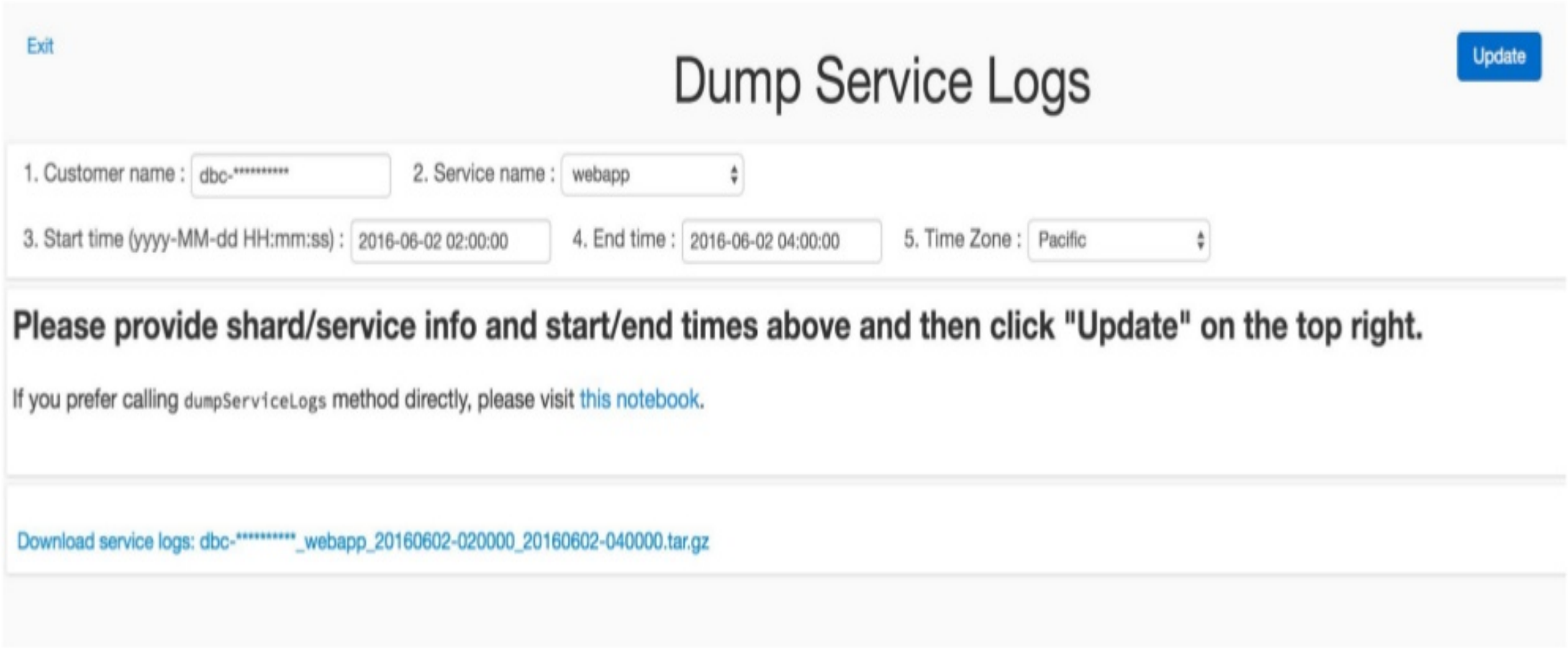
Data Analysis & Tools

We get the data in. What's next?

- Monitoring
- Debugging
- Usage Analysis
- Product Design (A/B testing)

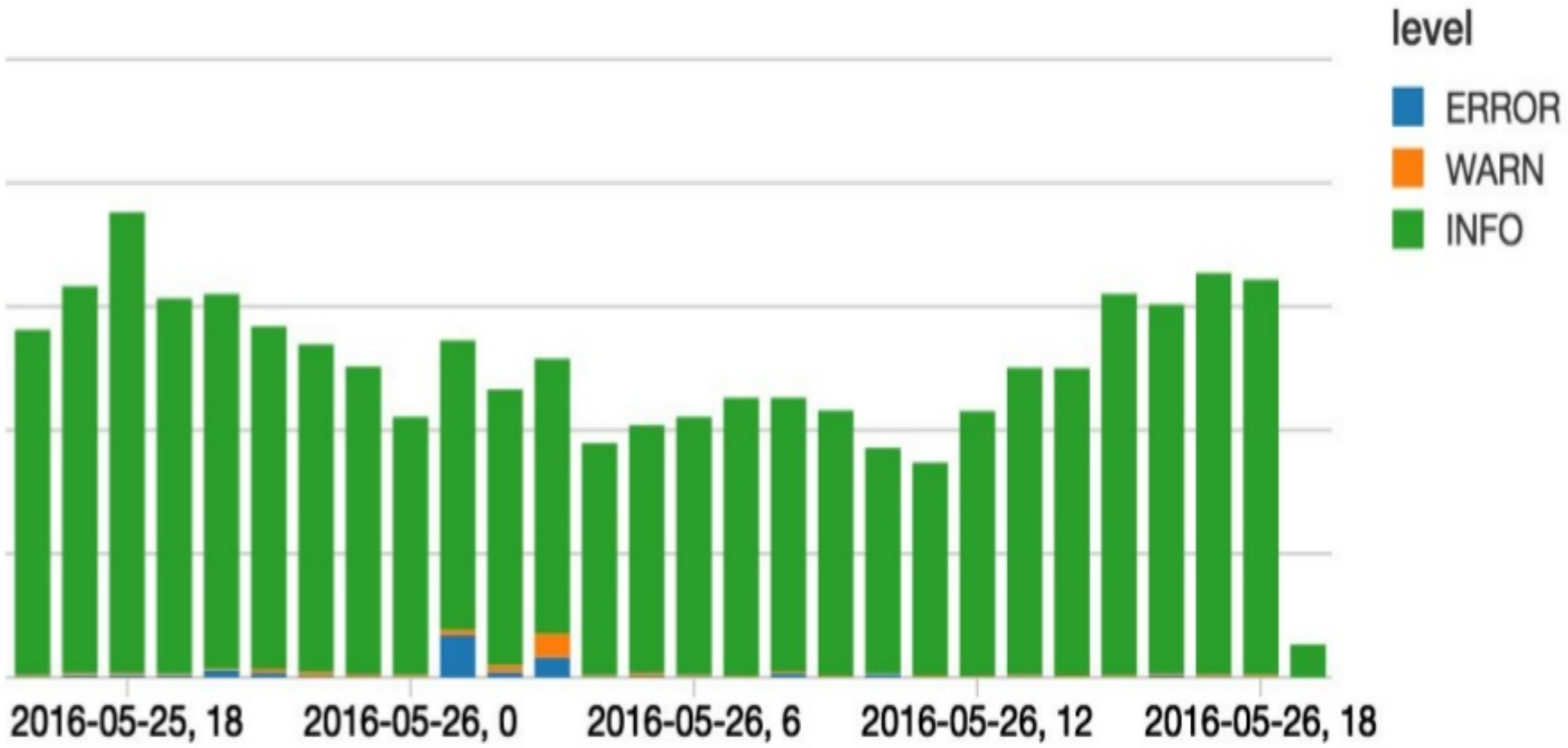
Debugging

Access to logs in a matter of seconds thanks to Apache Spark.



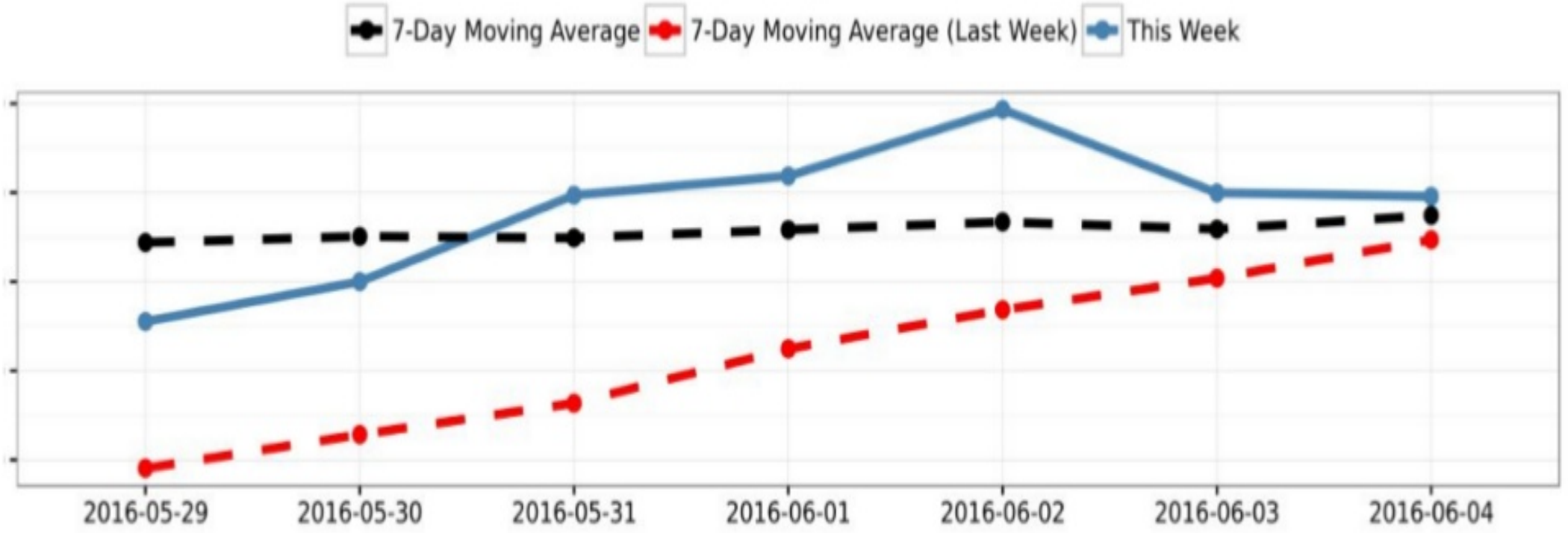
Monitoring

Monitor logs by log level. Bug introduced on 2016-05-26 01:00:00 UTC. Fix deployed in 2 hours.



Usage Analysis + Product Design

SparkR + ggplot2 = Match made in heaven



Summary

Databricks + Apache Spark create a unified platform for:

- ETL
- Data Warehousing
- Data Analysis
- Real time analytics

Issues with DevOps out of the question:

- No need to manage a huge cluster
- Jobs are isolated, they don't cannibalize each other's resources
- Can launch any Spark version

Ongoing & Future Work

Structured Streaming

- Reduce Complexity of pipeline:
  Sync Daemon + Delta + Batch Jobs => Single Streaming Job

- Reduce Latency
  Availability of data in seconds instead of minutes

- Event Time Dashboards

Try Apache Spark with Databricks

http://databricks.com/try

Thank you.

Have questions about ETL with Spark?
Join us at the Databricks Booth 3.45-6.00pm!