ABD312

AWS re:Invent

Deep Dive: Migrating Big Data Workloads to AWS

Bruno Faria, Sr. EMR Solution Architect, AWS
Ritesh Shah, Sr. Program Manager, Vanguard

November 28, 2017

Agenda

- Deconstructing current big data environments
- Identifying challenges with on-premises or unmanaged architectures
- Migrating components to Amazon EMR and Amazon Web Services (AWS) analytics services
  - Choosing the right engine for the job
  - Architecting for cost and scalability
- Customer migration story
  - How Vanguard migrated their big data workload to AWS
- Q&A

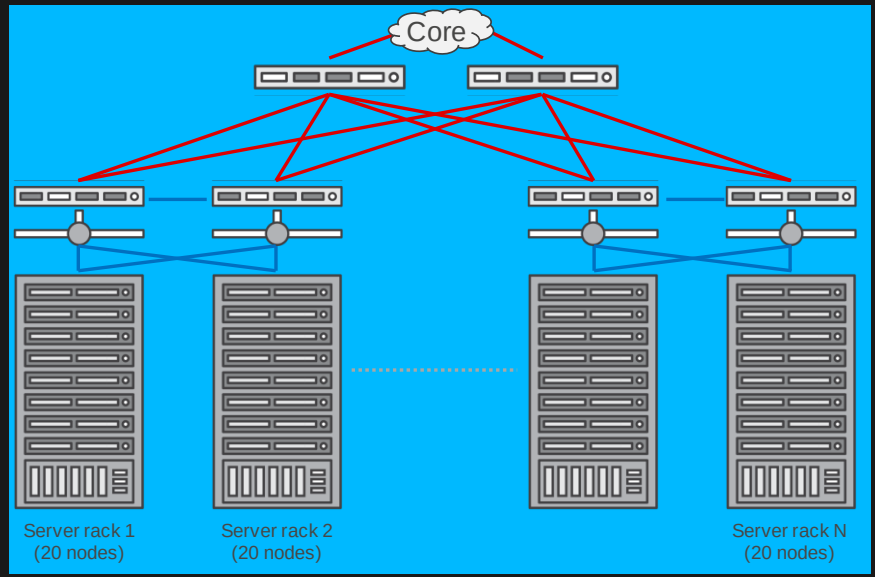
Deconstructing on-premises big data environments

AWS re:Invent

aws

On-premises Hadoop clusters

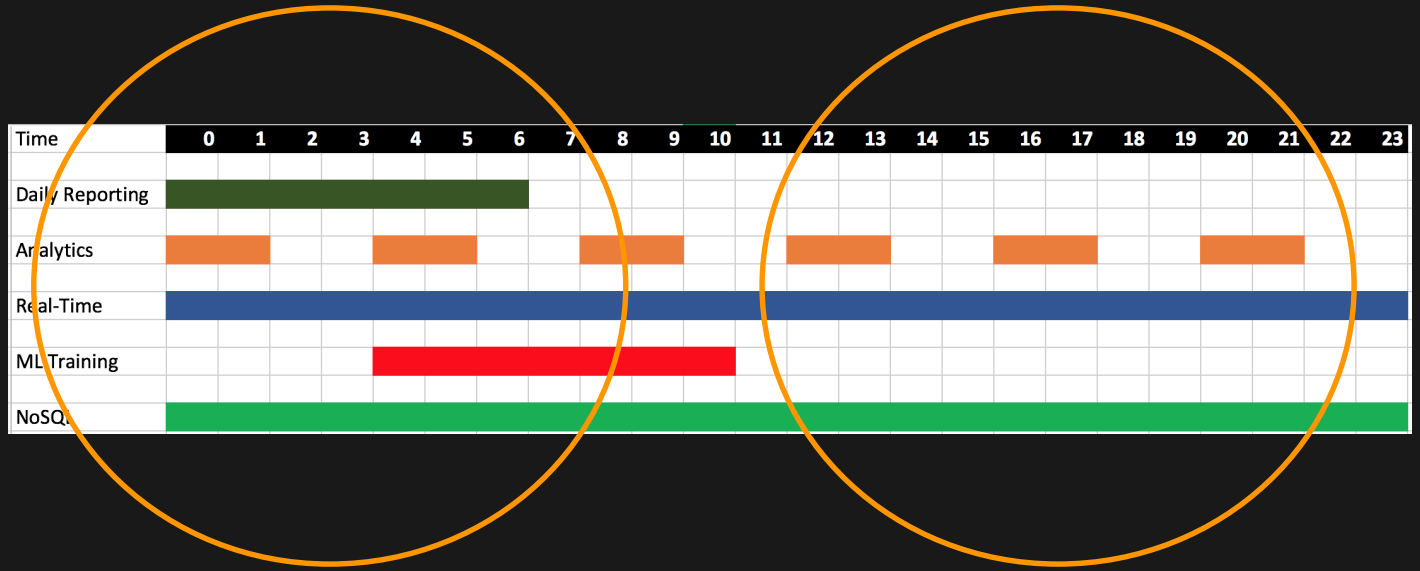
- A cluster of 1U machines
- Typically 12 cores, 32/64 GB RAM, and 6-8 TB of HDD ($3-4K)
- Networking switches and racks
- Open-source distribution of Hadoop or a fixed-licensing term by commercial distributions
- Different node roles
- HDFS uses local disk and is sized for 3x data replication



On premises: Workload types running on the same cluster

- Large-scale ETL
- Interactive queries
- Machine learning and data science
- NoSQL
- Stream processing
- Search
- Data warehouses

On premises: Swim lane of jobs



Over-utilized

Under-utilized

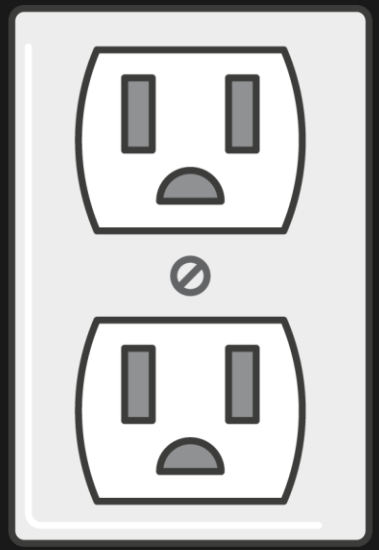
On premises: Role of a big data administrator

- Management of the cluster (failures, hardware replacement, restarting services, expanding cluster)
- Configuration management
- Tuning of specific jobs or hardware
- Managing development and test environments
- Backing up data and disaster recovery

Identifying on-premises challenges

On premises: Over-utilization and idle capacity

- Tightly coupled compute and storage requires buying excess capacity
- Can be over-utilized during peak hours and under-utilized at other times
- Results in high costs and low efficiency



On premises: System management difficulties

- Managing distributed applications and availability
- Durable storage and disaster recovery
- Adding new frameworks and doing upgrades
- Multiple environments
- Need team to manage cluster and procure hardware

Migrating workloads to AWS

AWS re:Invent

aws

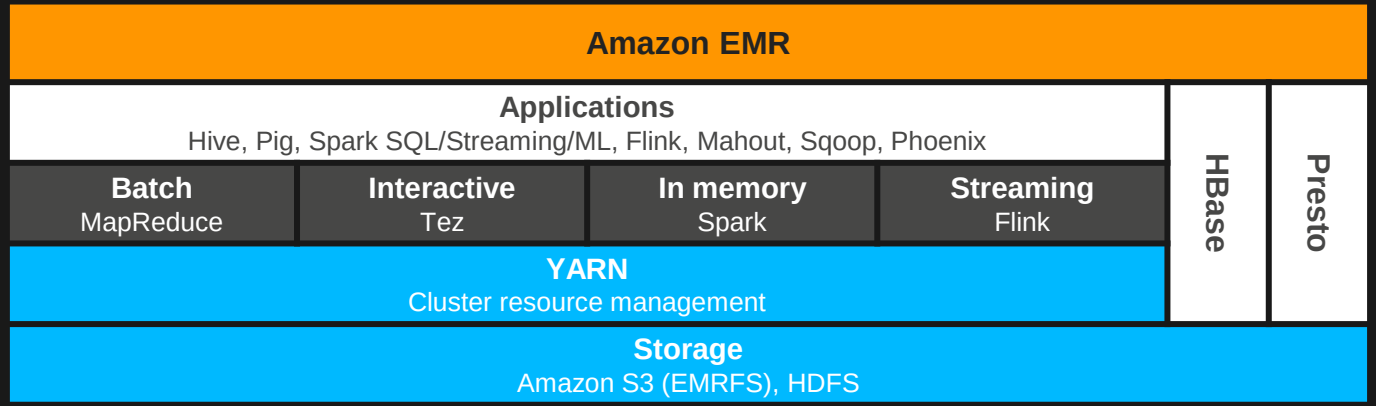
Key migration considerations

- Do not lift and shift
- Deconstruct workloads and use the right tool for the job
- Decouple storage and compute with Amazon Simple Storage Storage Service (Amazon S3)
- Design for cost and scalability

Deconstruct workloads—analytics types & frameworks

Batch	Takes minutes to hours Example: Daily/weekly/monthly reports Amazon EMR (MapReduce, Hive, Pig, Spark), AWS Glue
Interactive	Takes seconds Example: Self-service dashboards Amazon Redshift, Amazon Athena, Amazon EMR (Presto, Spark)
Stream	Takes milliseconds to seconds Example: Fraud alerts, 1-minute metrics Amazon EMR (Spark Streaming, Flink), Amazon Kinesis Analytics, KCL, Storm
Artificial intelligence	Takes milliseconds to minutes Example: Fraud detection, forecast demand, text to speech Amazon AI (Amazon Lex, Amazon Polly, Amazon ML, Amazon Rekognition), Amazon EMR (Spark ML), AWS Deep Learning AMI

Translate use cases to the right tools



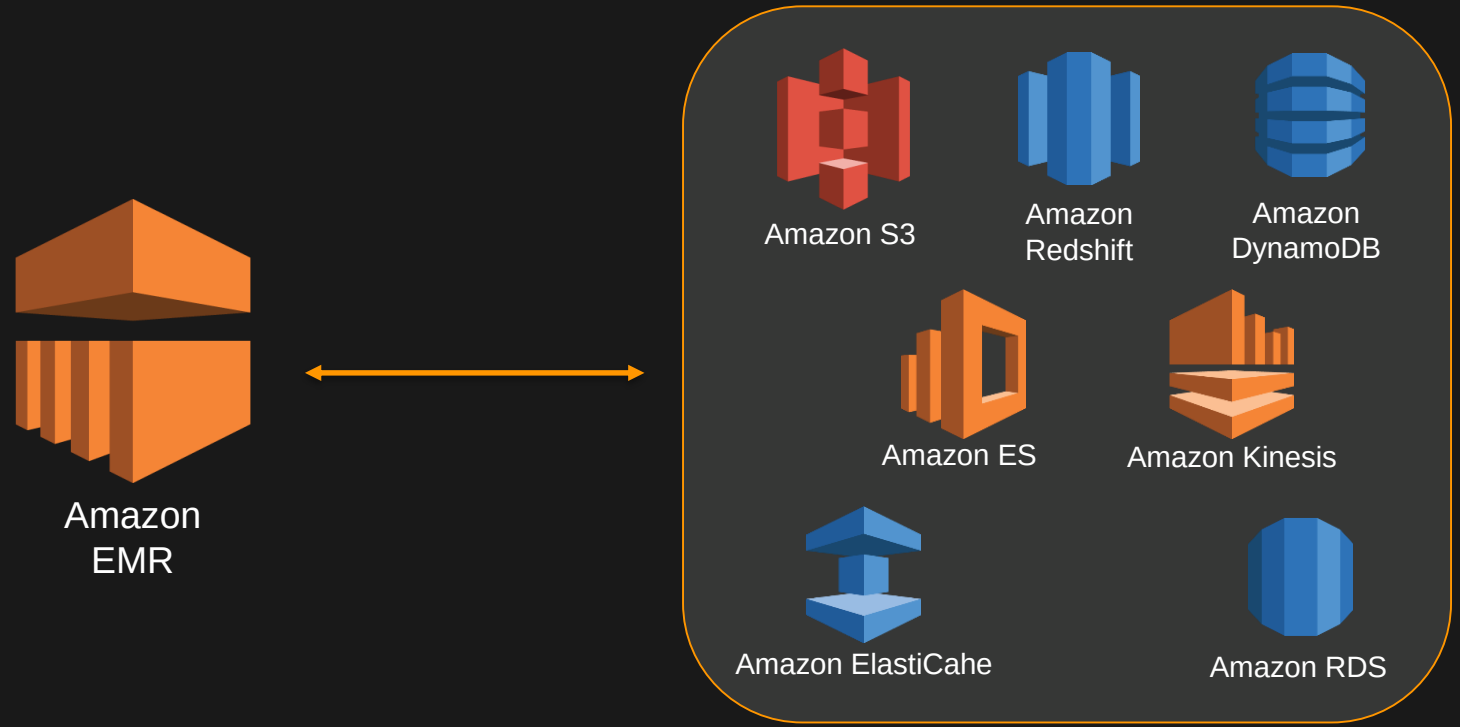
AWS Glue

Amazon Athena

Amazon Redshift/Spectrum

- Low-latency SQL -> Amazon Athena, Presto, Amazon Redshift/Spectrum
- Data warehouse/reporting -> Spark, Hive, AWS Glue, Amazon Redshift
- Management and monitoring -> Amazon CloudWatch, AWS console, Ganglia
- HDFS -> Amazon S3
- Notebooks -> Zeppelin Notebook, Jupyter (via bootstrap action)
- Query console -> Amazon Athena, Hue
- Security -> Ranger (CF template), HiveServer2, AWS IAM roles

Many storage layers to choose from



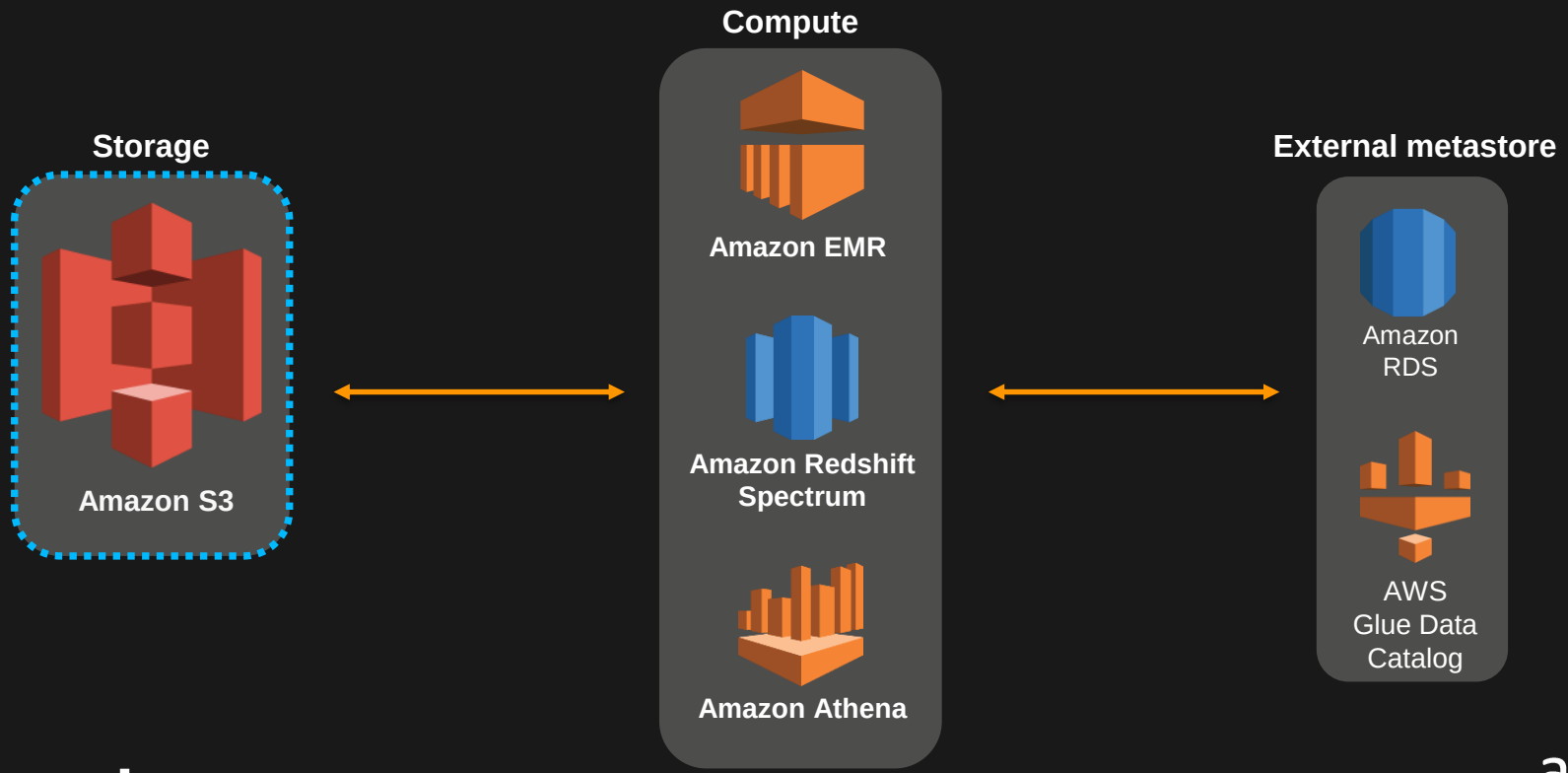
Amazon S3 as your persistent data store

- Natively supported by big data frameworks (Spark, Hive, Presto, and others)
- Decouple storage and compute
    - No need to run compute clusters for storage (unlike HDFS)
    - Can run transient Amazon EMR clusters with Amazon EC2 Spot Instances
    - Multiple and heterogeneous analysis clusters and services can use the same data
- Designed for 99.999999999% durability
- No need to pay for data replication
- Secure—SSL, client/server-side encryption at rest
- Low cost

What about HDFS and data tiering?

- Use HDFS for very frequently accessed (hot) data
- Use Amazon S3 Standard for frequently accessed data
- Use Amazon S3 Standard – IA for less frequently accessed data
- Use Amazon Glacier for archiving cold data

Decouple storage and compute



Amazon S3 tips: Partitions, compression, and file formats

- Partition your data for improved read performance
  - Read-only files the query needs
  - Reduce amount of data scanned
- Optimize file sizes
  - Avoid files that are too small (generally, anything less than 128 MB)
- Fewer files, matching closely to block size
  - Fewer calls to Amazon S3 (faster listing)
  - Fewer network/HDFS requests
- Compress data set to minimize bandwidth from Amazon S3 to Amazon EC2
  - Make sure you use splittable compression or have each file be the optimal size for parallelization on your cluster
- Columnar file formats like Parquet can give increased performance on reads

Data partitioning—examples

```
ALTER TABLE app_logs ADD PARTITION (year='2015',month='01',day='01') LOCATION
's3://bucket_name/app/plaintext/year=2015/month=01/day=01/';

ALTER TABLE elb_logs ADD PARTITION (year='2015',month='01',day='01') LOCATION
's3://bucket_name/elb/plaintext/2015/01/01/';

ALTER TABLE orders DROP PARTITION (dt='2014-05-14',country='IN'),
PARTITION (dt='2014-05-15',country='IN');

ALTER TABLE customers PARTITION (zip='98040', state='WA') SET LOCATION
's3://bucket_name/new_customers/zip=98040/state=WA';

MSCK REPAIR TABLE table_name;  ← Only works with Hive-compatible partitions
```

File formats

Columnar—Parquet and ORC

- Compressed
- Column-based read-optimized
- Integrated indexes and stats

Row—Avro

- Compressed
- Row-based read-optimized
- Integrated indexes and stats

Text—xSV, JSON

- May or may not be compressed
- Not optimized
- Generic and malleable

File Format – Examples

SELECT count(*) as count FROM examples_csv

Run time: 36 seconds, Data scanned: 15.9GB

SELECT count(*) as count FROM examples_parquet

Run time: 5 seconds, Data scanned: 4.93GB

File formats—considerations

Scanning

- xSV and JSON require scanning entire file
- Columnar ideal when selecting only a subset of columns
- Row ideal when selecting all columns of a subset of rows

Read performance

- Text – SLOW
- Avro – Optimal (specific to use case)
- Parquet and ORC – Optimal (specific to use case)

Write performance

- Text – SLOW
- Avro – Good
- Parquet and ORC – Good (has some overhead with large datasets)

* Highly dependent on the dataset

External hive metastore

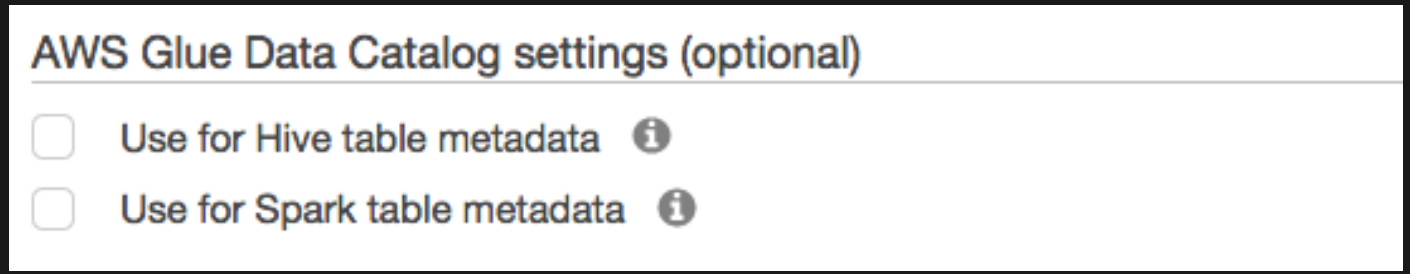
Use an external metastore when you require a persistent metastore or a metastore shared by different clusters, services, and applications. There are two options for an external metastore:

- Amazon Relational Database Service (Amazon RDS)/Amazon Aurora
- AWS Glue Data Catalog (Amazon EMR version 5.8.0 or later only)
    - Search metastore
    - Utilize crawlers to detect new data, schema, partitions
    - Schema and version management

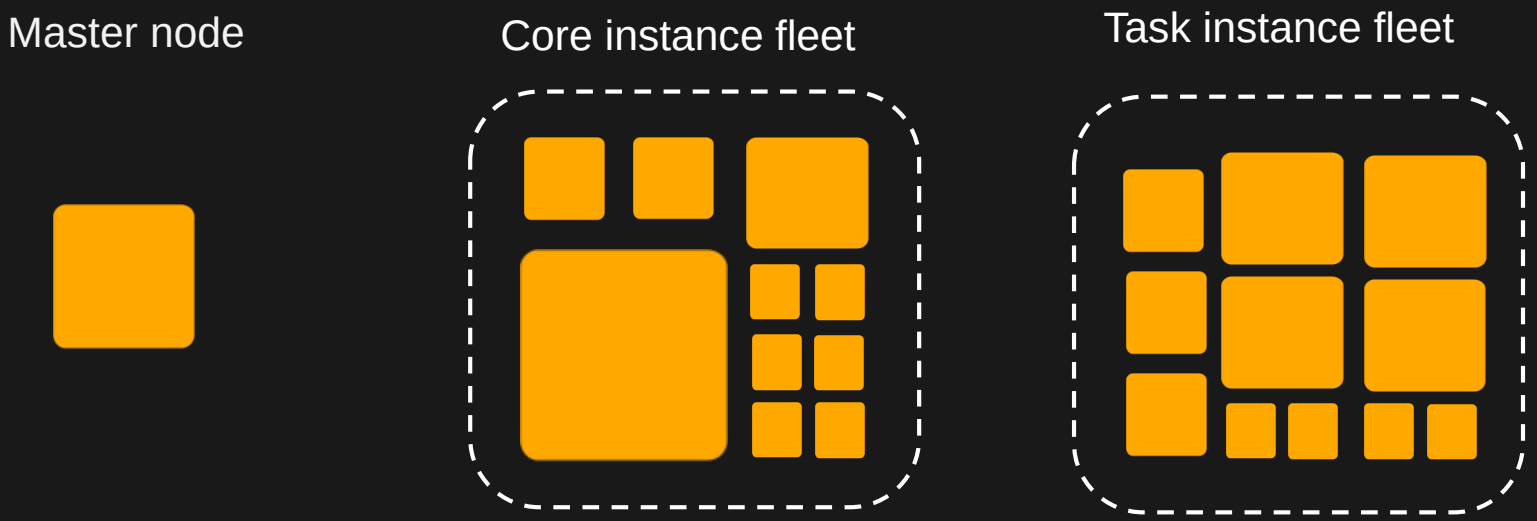
External metastore—AWS Glue Data Catalog

You can choose to use the AWS Glue Data Catalog to store external table metadata for Hive and Spark instead of utilizing an on-cluster or self-managed Hive metastore. This allows you to more easily store metadata for your external tables on Amazon S3 outside of your cluster.

You can configure your Amazon EMR clusters to use the AWS Glue Data Catalog from the Amazon EMR console, AWS Command Line Interface (CLI), or the AWS SDK with the Amazon EMR API.

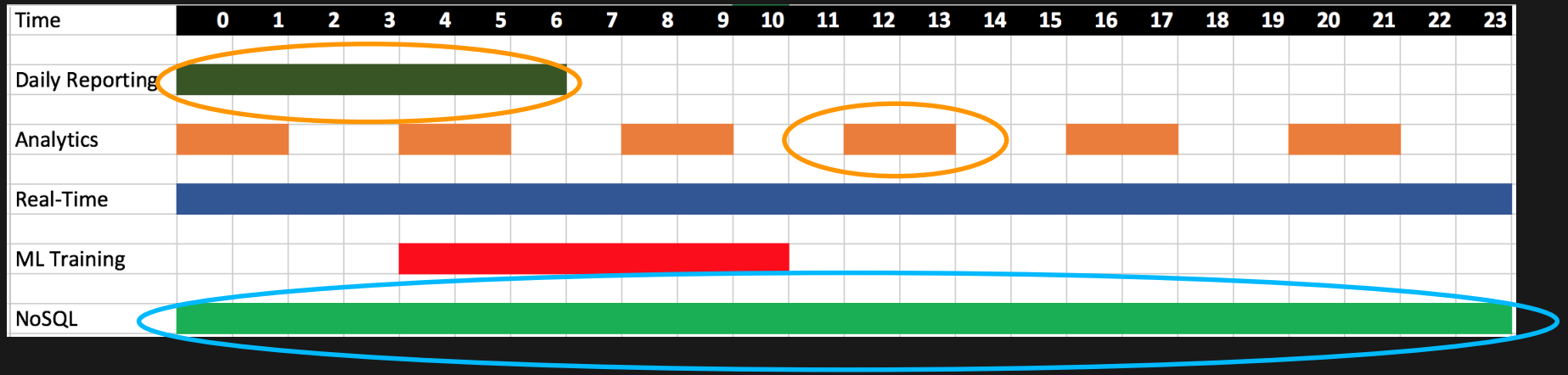


Instance fleets for advanced Spot provisioning



- Provision from a list of instance types with Spot and On-Demand
- Launch in the most optimal Availability Zone based on capacity/price
- Spot block support

Transient or long-running workloads



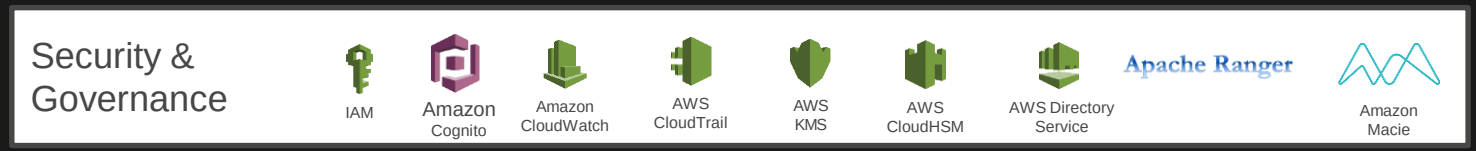
Transient

Long-running

Amazon EMR: Lower costs with Auto Scaling

Security—governance and auditing

- AWS Identity and Access Management (IAM)
- Amazon Cognito
- Amazon CloudWatch and AWS CloudTrail
- AWS Key Management Service (AWS KMS)
- AWS Directory Service
- Amazon S3 access logs for cluster Amazon S3 access
- Amazon Macie
- YARN and application logs
- Apache Ranger



Migrating workloads to AWS at Vanguard

Ritesh Shah, Sr. Program Manager

AWS re:Invent

aws

Vanguard—background

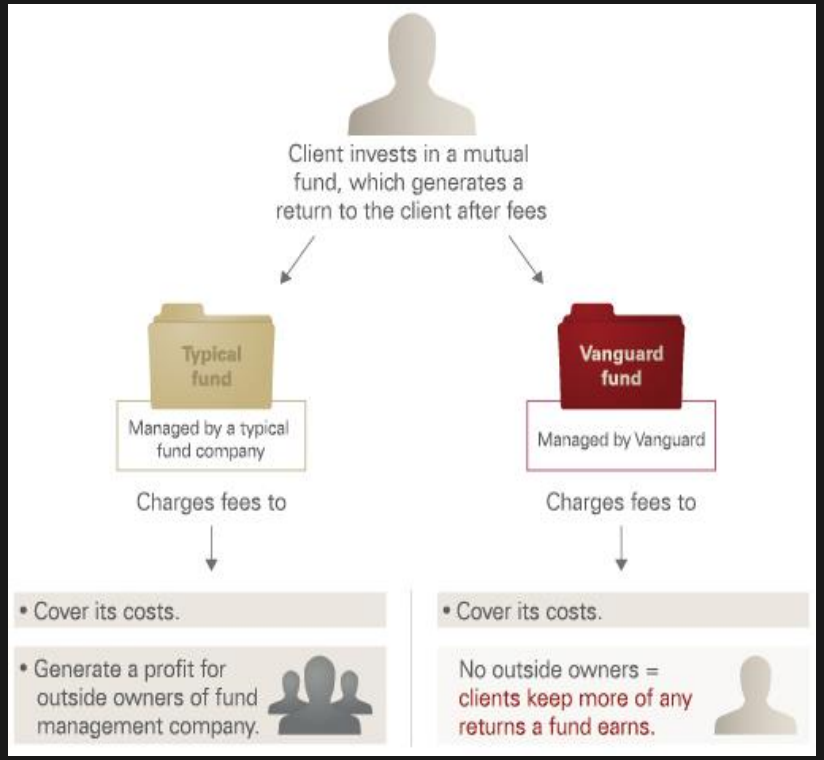
Vanguard is one of the world's largest investment companies, offering a large selection of low-cost mutual funds, ETFs, advice, and related services

Core purpose – To take a stand for all investors, to treat them fairly, and to give them the best chance for investment success

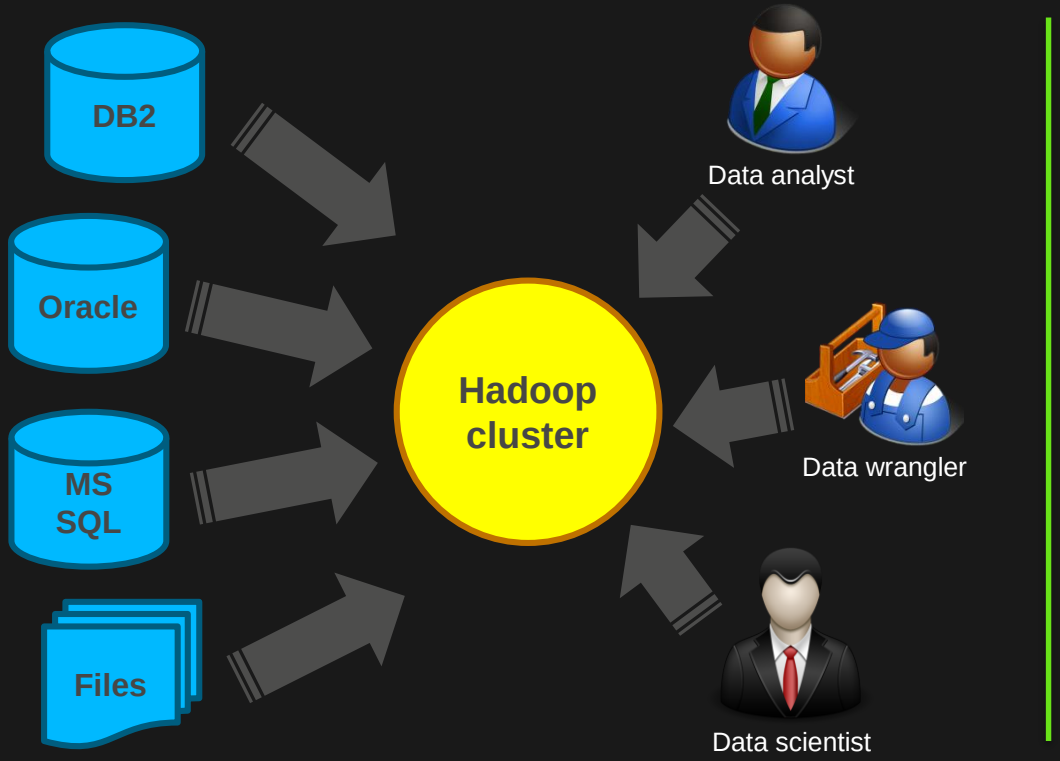
Oldest fund – Wellington Fund (inception 1929)

Began operations – May 1, 1975 in Valley Forge, PA

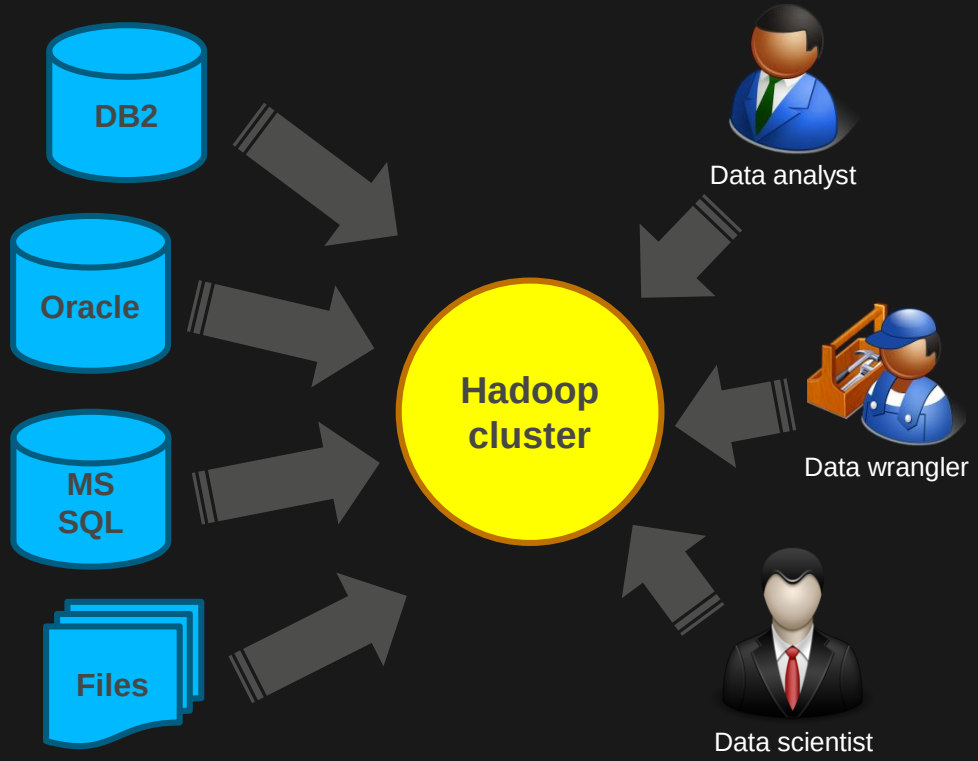
Funds – Over 180 U.S. funds (including variable annuity portfolios) and 190 additional funds in markets outside the United States



Deconstructing on-premises workloads



Deconstructing on-premises workloads



Hadoop
Hive
Oozie
Sqoop
Sentry
Impala
Hue
Spark
Python
Tableau

Why migrate from on premises to AWS?

- Tightly coupled compute and storage
- Over-utilized during peak hours and under-utilized at other times
- Cross impact from different types of workloads
- Lack of DR environment
- Dedicated team to maintain cluster

Why migrate from on premises to AWS?

- Tightly coupled compute and storage
- Over-utilized during peak hours and under-utilized at other times
- Cross impact from different types of workloads
- Lack of DR environment
- Dedicated team to maintain cluster
- Long procurement cycles for hardware
- Long setup time of hardware
- Complex upgrade coordination needs leading to infrequent upgrades
- High total cost of ownership (TCO)
- Difficult to charge back LOB

Demystifying EMR workloads

AWS re:Invent

aws

Foundational requirements

Secure
Encryption in flight & at rest

Flexible
Customize per workload

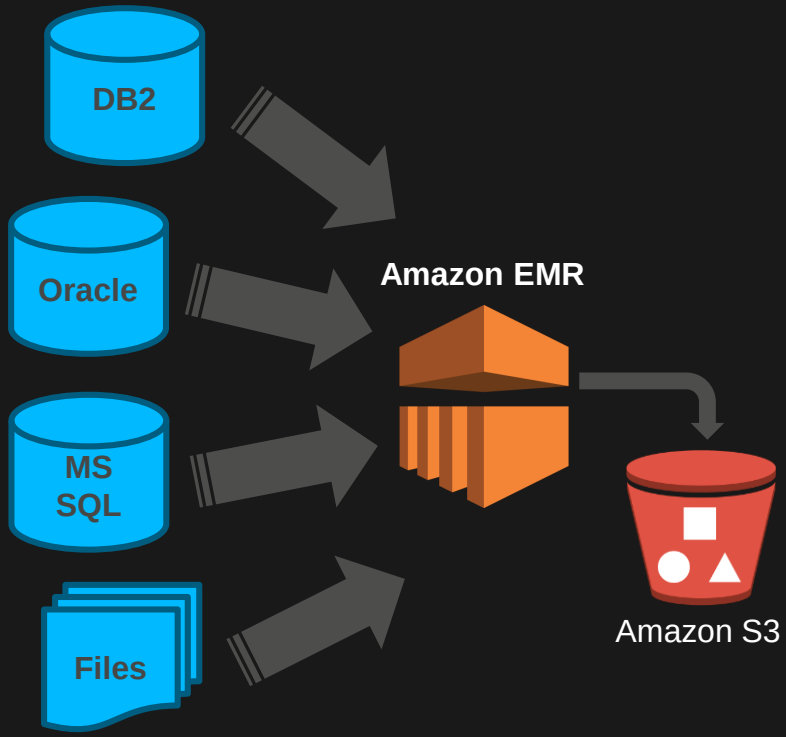
Managed
Reduced administration

Lower TCO
Pay as per usage

Full control
Infrastructure as code

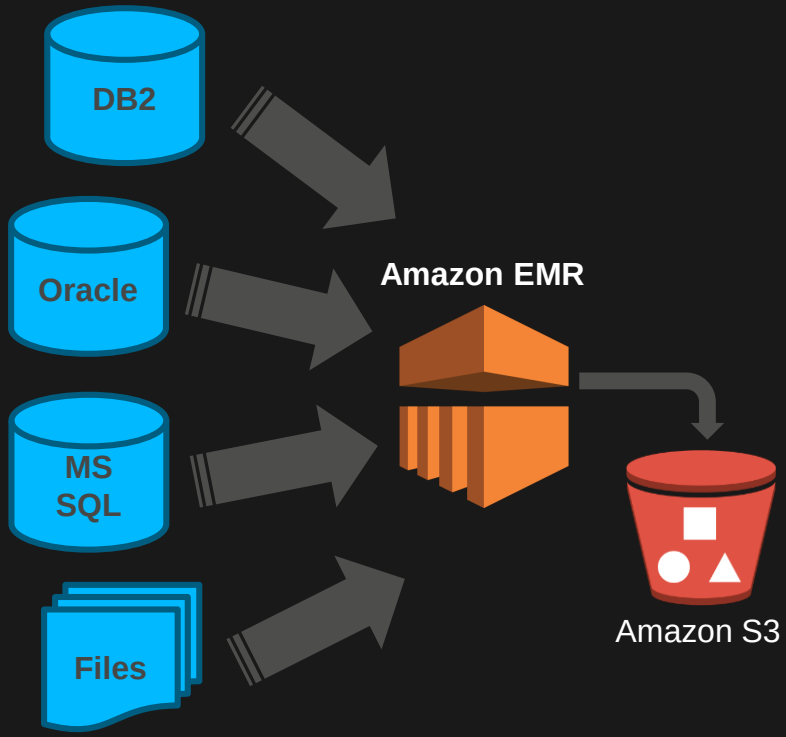
Scalable
Compute & storage

Ingestion workloads



- Ephemeral EMR–based ingestion clusters
- Amazon S3–based data lake
- In-flight and at-rest data encryption using KMS
- S3 bucket access control policies using IAM
- Step API used to launch Oozie–based workflows
- Sqoop, Snowball, and CDC (Attunity)–based data ingestion
- Hive/Tez and Spark–based data processing
- CloudWatch plus Splunk–based monitoring

Ingestion workloads

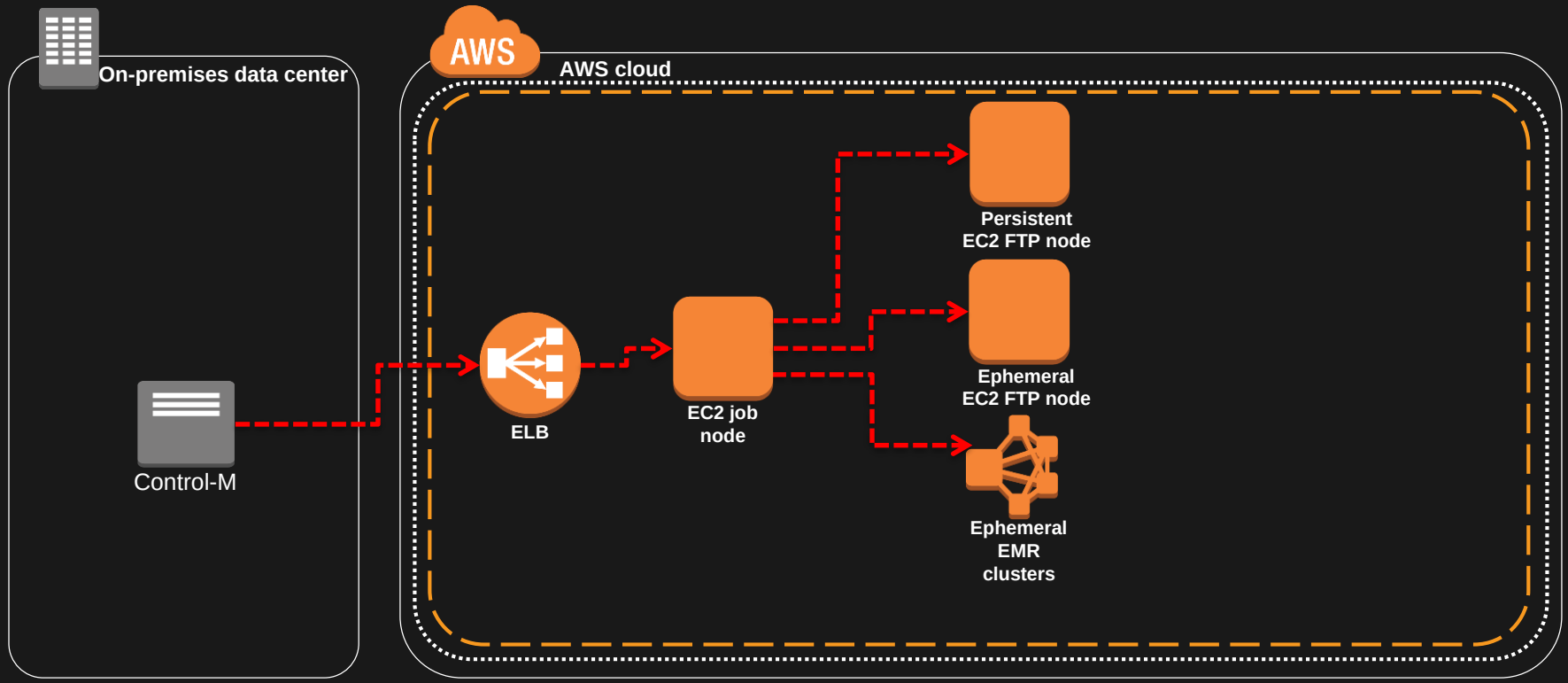


- Ephemeral EMR–based ingestion clusters
- Amazon S3–based data lake
- In-flight and at-rest data encryption using KMS
- S3 bucket access control policies using IAM
- Step API used to launch Oozie–based workflows
- Sqoop, Snowball, and CDC (Attunity)–based data ingestion
- Hive/Tez and Spark–based data processing
- CloudWatch plus Splunk–based monitoring

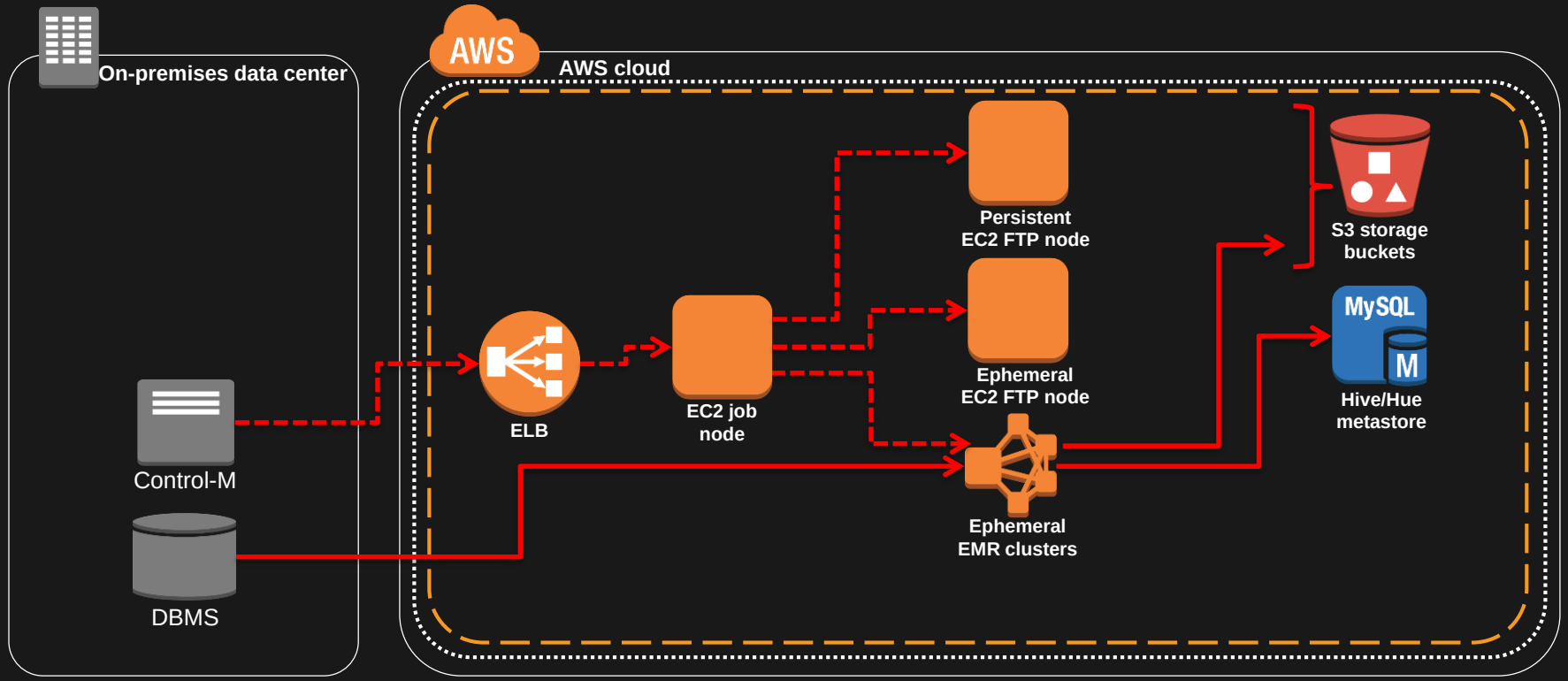
Lessons learned

- Segregation of infrastructure and ingestion code
- Instance type and cluster sizing
- Obfuscation of PII data
- Segregation of data based on LOB rules

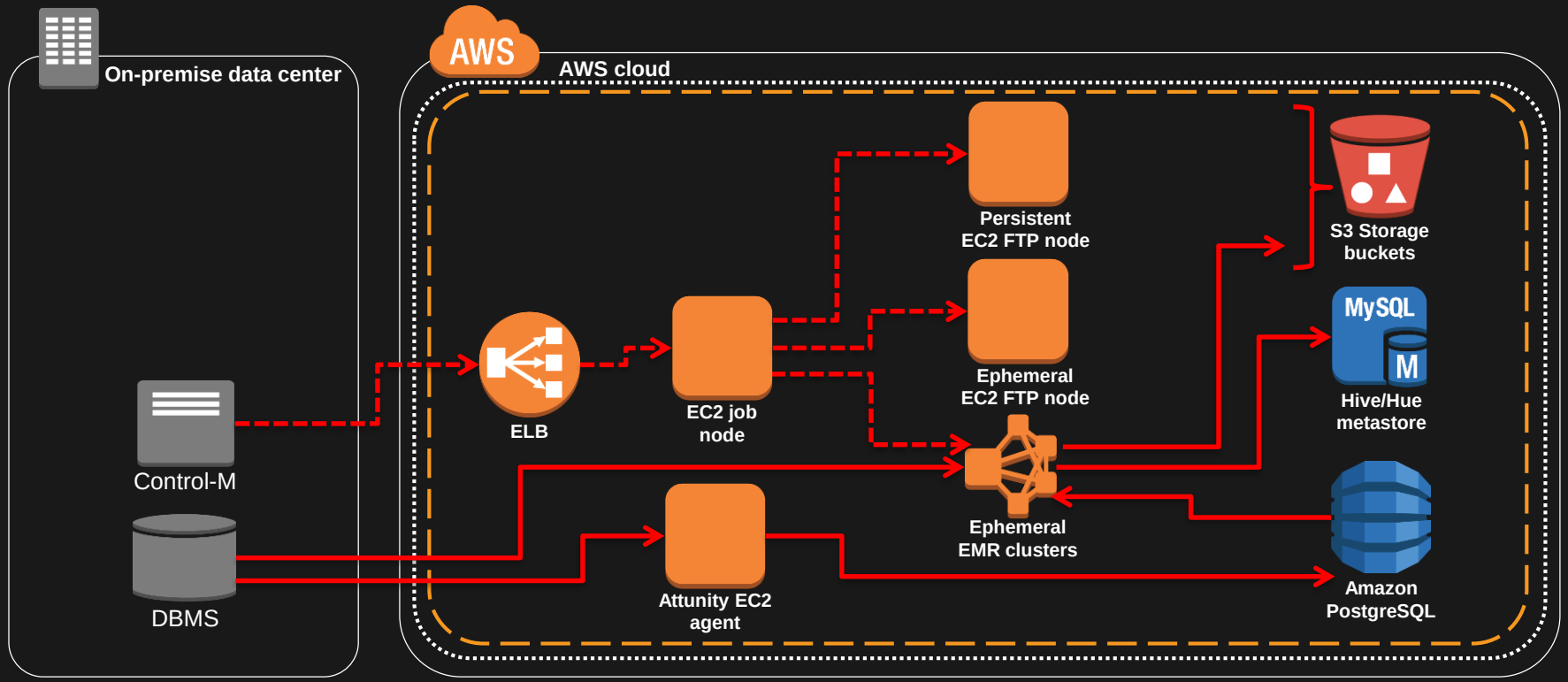
Conceptual diagram—ingestion workload



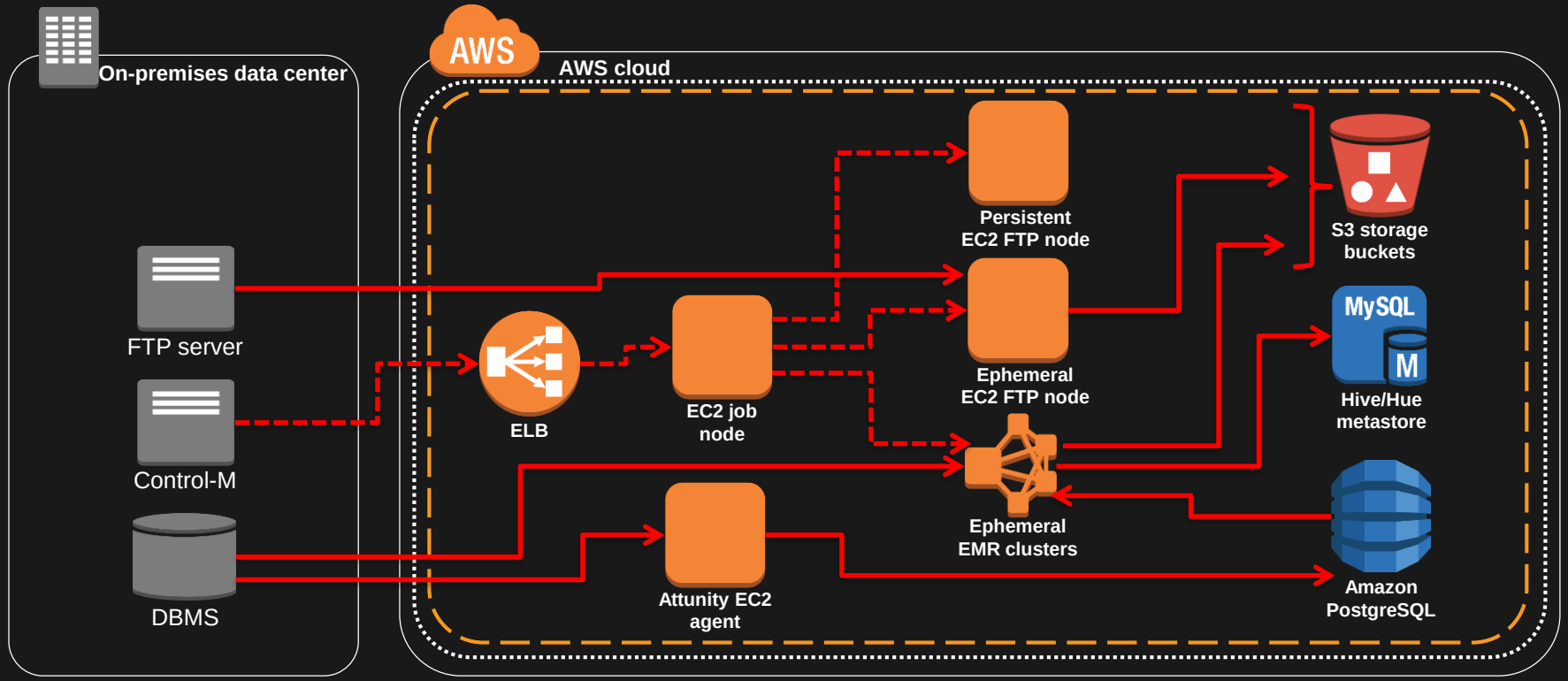
Conceptual diagram—ingestion workload



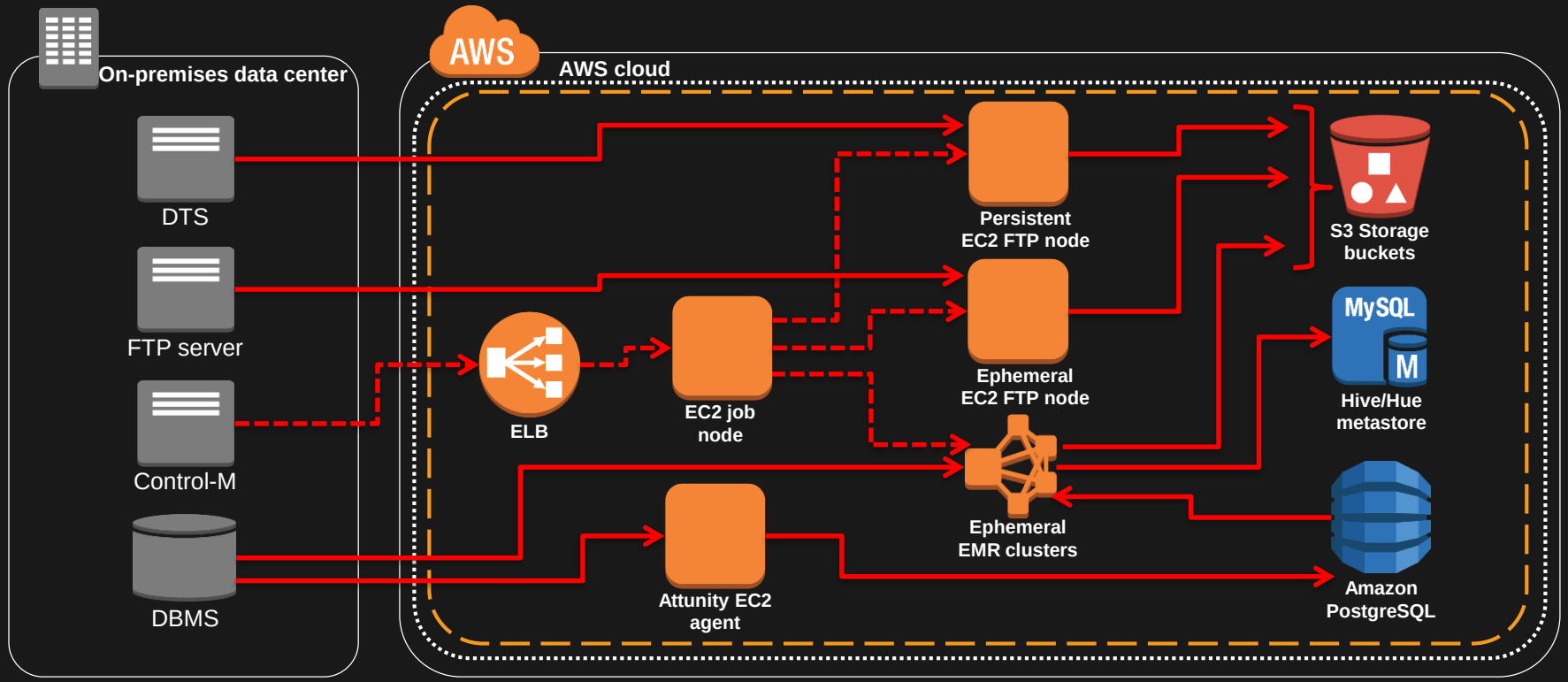
Conceptual diagram—ingestion workload



Conceptual diagram—ingestion workload



Conceptual diagram—ingestion workload

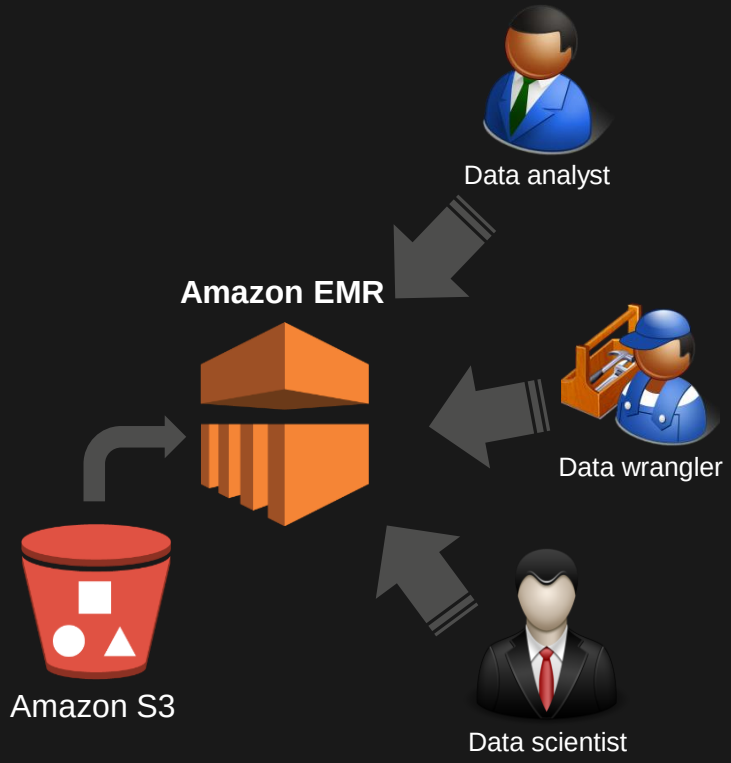


AWS re:Invent

AWS Snowball | Amazon CloudWatch | IAM | AWS KMS

aws

Analytics workloads



- Persistent EMR-based analytics clusters
- Hive/Tez and Presto as query engines
- Spark and Python-based distributed processing
- Hue and Tableau as user interactive tools
- Zeppelin and Jupyter-based notebook for interactive and collaborative data exploration
- Authorization using Hive SQL Auth and IAM bucket policies for Amazon S3

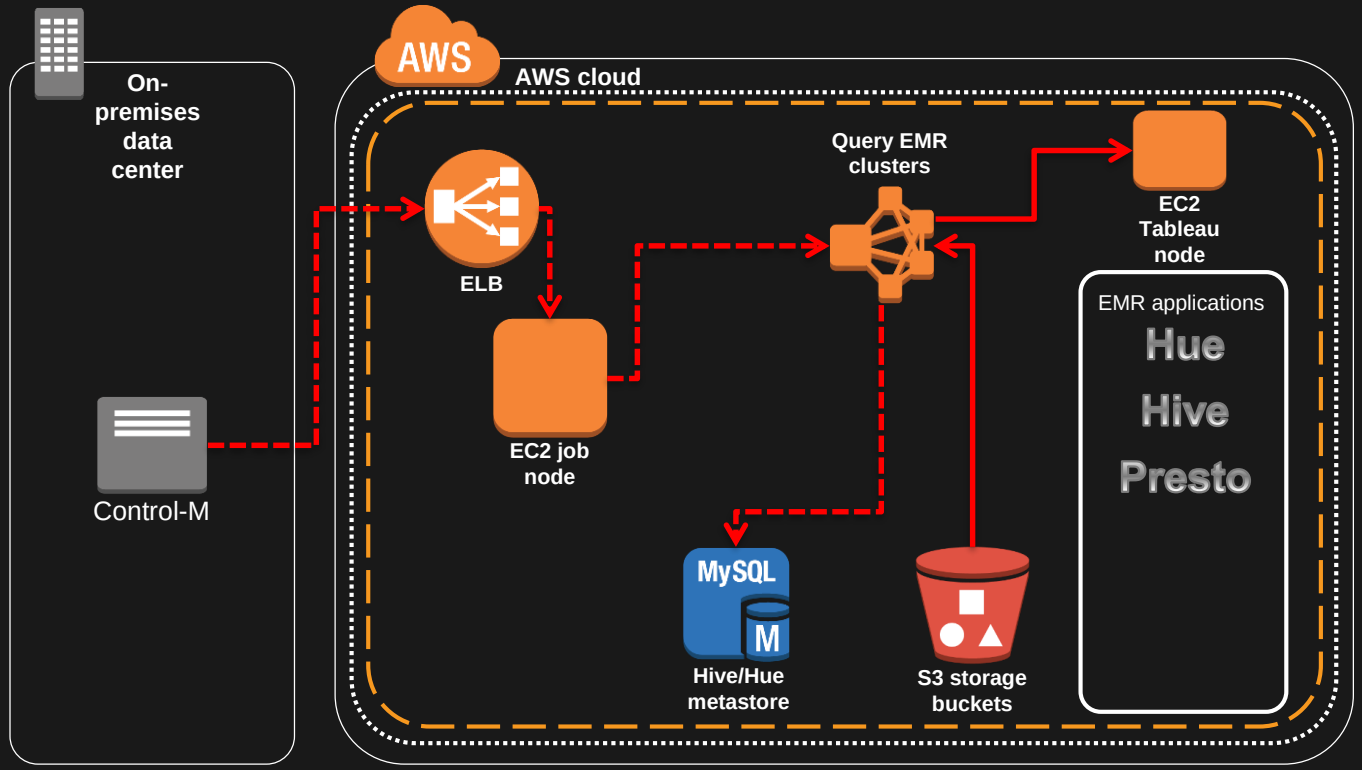
Analytics workloads

- Persistent EMR-based analytics clusters
- Hive/Tez and Presto as query engines
- Spark and Python-based distributed processing
- Hue and Tableau as user interactive tools
- Zeppelin and Jupyter-based notebook for interactive and collaborative data exploration
- Authorization using Hive SQL Auth and IAM bucket policies for Amazon S3

Lessons learned

- Data governance (metadata, lineage, and others)
- Audit of access to data
- Instance type and cluster sizing
- Integration of user tools with query and processing engines

Conceptual diagram—analytics workload

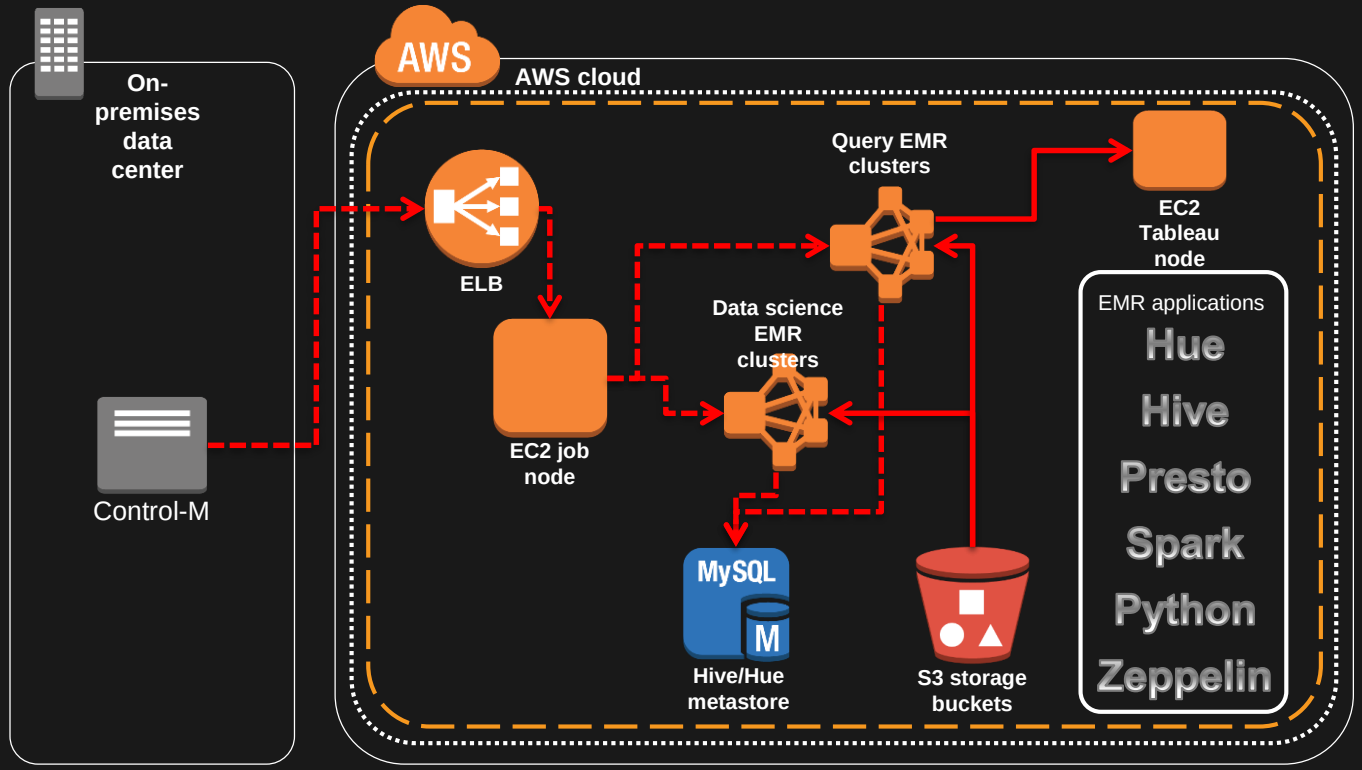


Amazon CloudWatch

IAM

AWS KMS

Conceptual diagram—analytics workload

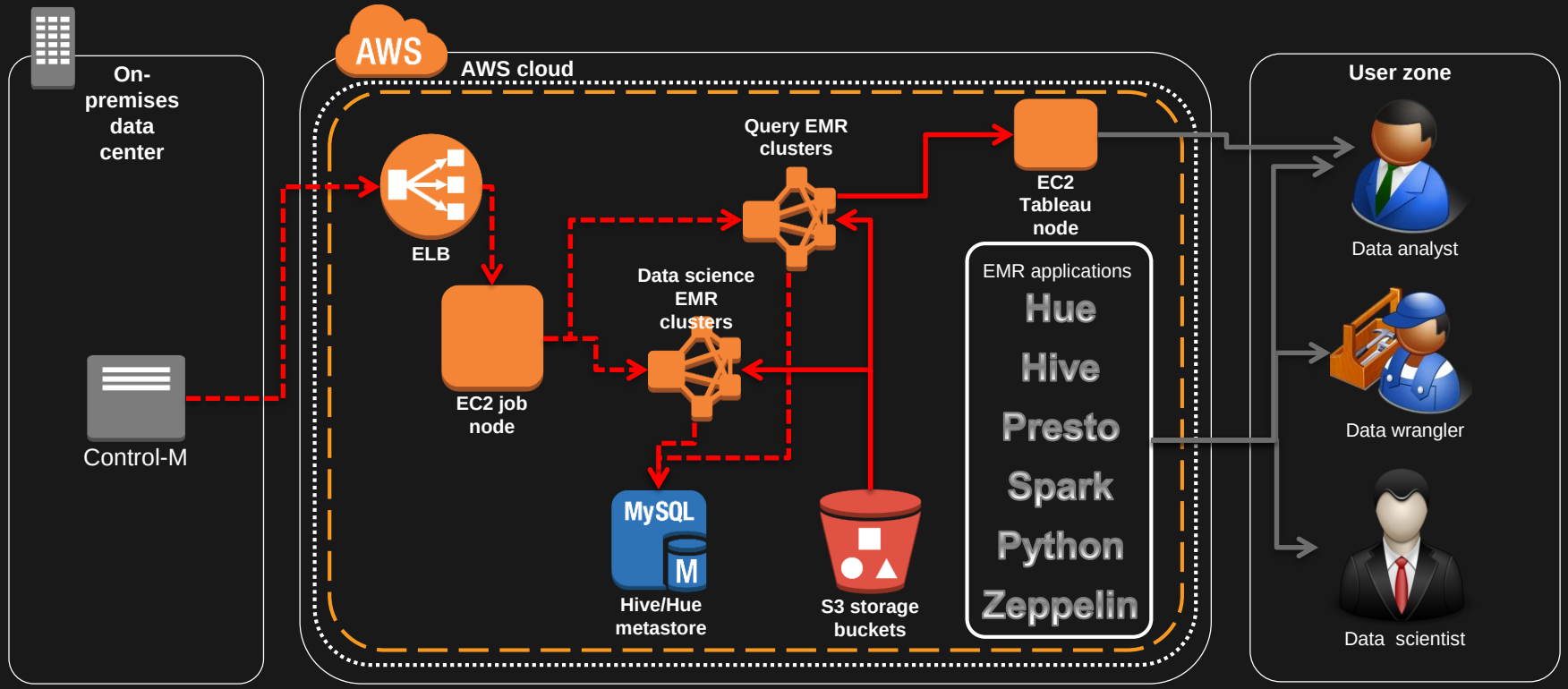


Amazon CloudWatch

IAM

AWS KMS

Conceptual diagram—analytics workload



Amazon CloudWatch

IAM

AWS KMS

What's next

Enable additional AWS services for:

- Real-time data ingestion
- Enhanced data processing pipelines
- Visualization and analytics use case
- Experimentation with AI and ML

AWS re:Invent

Q&A

Thank you!

aws.amazon.com/emr

blogs.aws.amazon.com/bigdata

Please remember to rate this session in the mobile app