

Scenario

Every day in a large global consulting company, all employees enter their timesheets into an online system that stores the data on-premises in a transactional database. At regular intervals during the day, a **batch ETL job load the data into a data warehouse to report on financials.**

Your CEO is concerned about potential incidents of timesheet fraud and is interested in using **machine learning to identify suspect timesheets in timely manner.**

1. How can you feed large volumes of data into your machine learning model for generating predictions?
2. How can you analyze the data and build visualization and/or reports?

Amazon ML for machine learning...? Hmm...

- Create and deploy models to production using your data that is already in the AWS cloud
- Automate model lifecycle with full features APIs and SDKs (Java, Python, .NET, JavaScript, Ruby, and PHP)

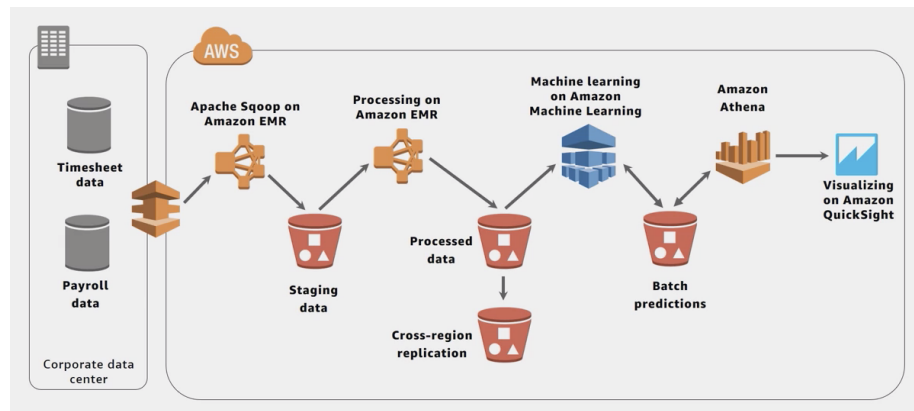


Figure 1: image

Getting On-premises data into the cloud

Push from on-premises, pull from the cloud with single instance, or cluster of instances.

The common pattern is using ETL node to orchestrate and ingest the data from on-premises databases. But it might throttle, since it's depending on single,

stale instance.

So why not use Amazon EMR! Orchestration and ingestion is done on ETL node, but the data processing is done on EMR.

But since you still depend on one ETL node for ingesting, you can encounter networking bottlenecks. You can avoid this network bottleneck by using Apache Sqoop to **ingest** data from on-premises database.

Then, after ingestion is done, and if you need some time-scheduled batching, use different EMR cluster to process the data furthermore.

Amazon EMR cluster

Managed Hadoop framework running on dynamically scaled amazon EC2 instances.

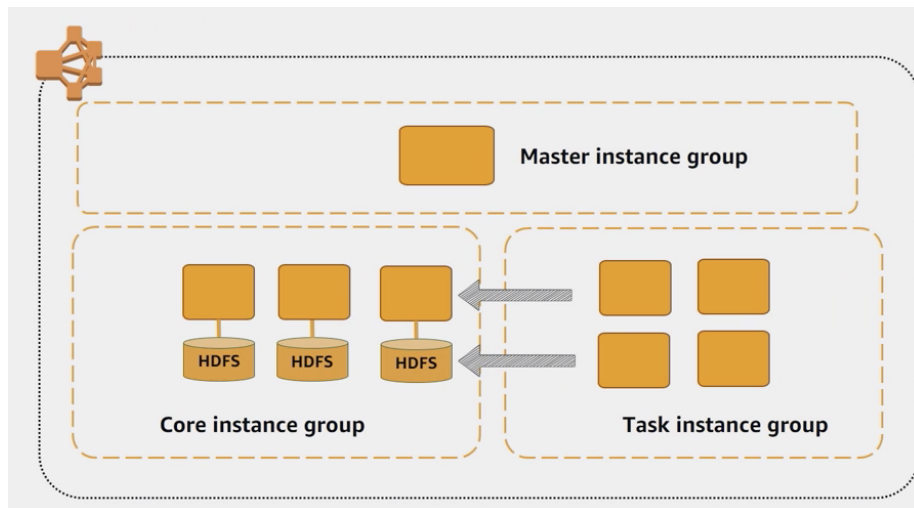


Figure 2: image

Do you still need your data after the cluster is done running? EMR uses EMRFS, which supports both HDFS and S3. For certain scenarios where job outputs HDFS for iterative calculations, use **DistCP to copy the data to S3 before terminating cluster**.

Class Discussion 1

A customer running an analytics platform want to scale out their Amazon EMR cluster. The majority of work involves running ad hoc, low-latency SQL queries on S3. What is the best to scale out the cluster while saving cost?

A: Use automatic scaling

Analyzing the Data and Building Visualization

Your tables will be append-only. You can use Athena to run SQL queries.

Athena

It uses Hive (optimized for query throughput) for DDL (Database Definition Language) functionality

- complex data types
- multitude of formats
- supports data partitioning

Teradata Presto(optimized for latency) for SQL queries

- In-memory distributed query engine
- ANSI-SQL compatible with extensions.

Hive is good for daily,weekly report which must run reliably. Presto is good for small, interactive queries (ad hoc)

Athena provides **Schema-on-read capability**. Query is projected on the data as is in Amazon S3. This gives capability of creating different schemas or table structures with the same data.

Athena : Best Practices

- Partitioning
 - reduce the amount of data scanned
 - Read only files necessary for queries
- Compression and file sizes
 - Splittable files allows Athena's execution engine to split the reading of a file by multiple readers to increase parallelism
- Columnar formats for analytics
 - optimize column-based reads
 - use apache parquet and apache ORC

Athena has both JDBC and ODBC driver. So download the driver and connect your favorite tools for analyzing and visualizing the data. connect athena to your favorite BI tools.

There is also **asynchronous interaction model** using Athena API supports:
- Named queries - column data and metadata - integration with existing data access tools - paginated result sets

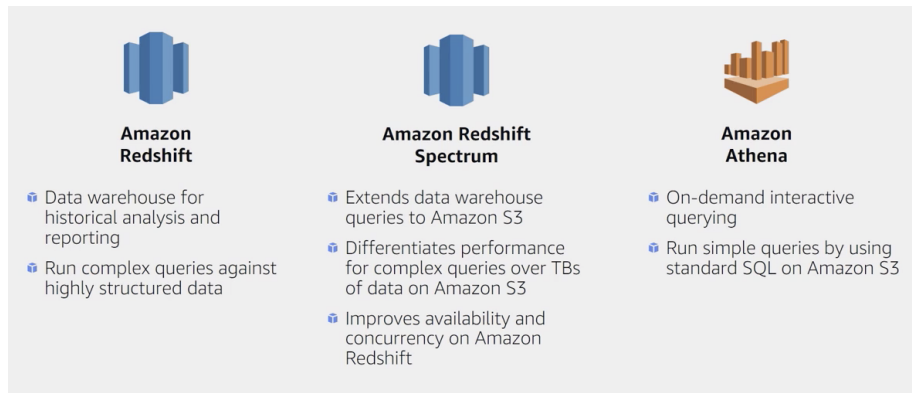


Figure 3: image

Now you can visualize Athena Data using Amazon QuickSight

1. Create an Amazon Athena data source
2. Select an Athena table in QuickSight
3. Do your visualization!

Class Discussion 2

What are some of the optimization techniques that you can implement with Amazon Athena for a faster execution?

A: Partitioning matters the most. It acts as virtual columns. Also, you can restrict the amount of scanned data by specifying filters based on the partition.

Class Discussion 3

You need to run a query using Amazon Athena and use the results of that query to perform further analysis using a second query. How would you accomplish this?

A: create an external table. run a new query on new table.