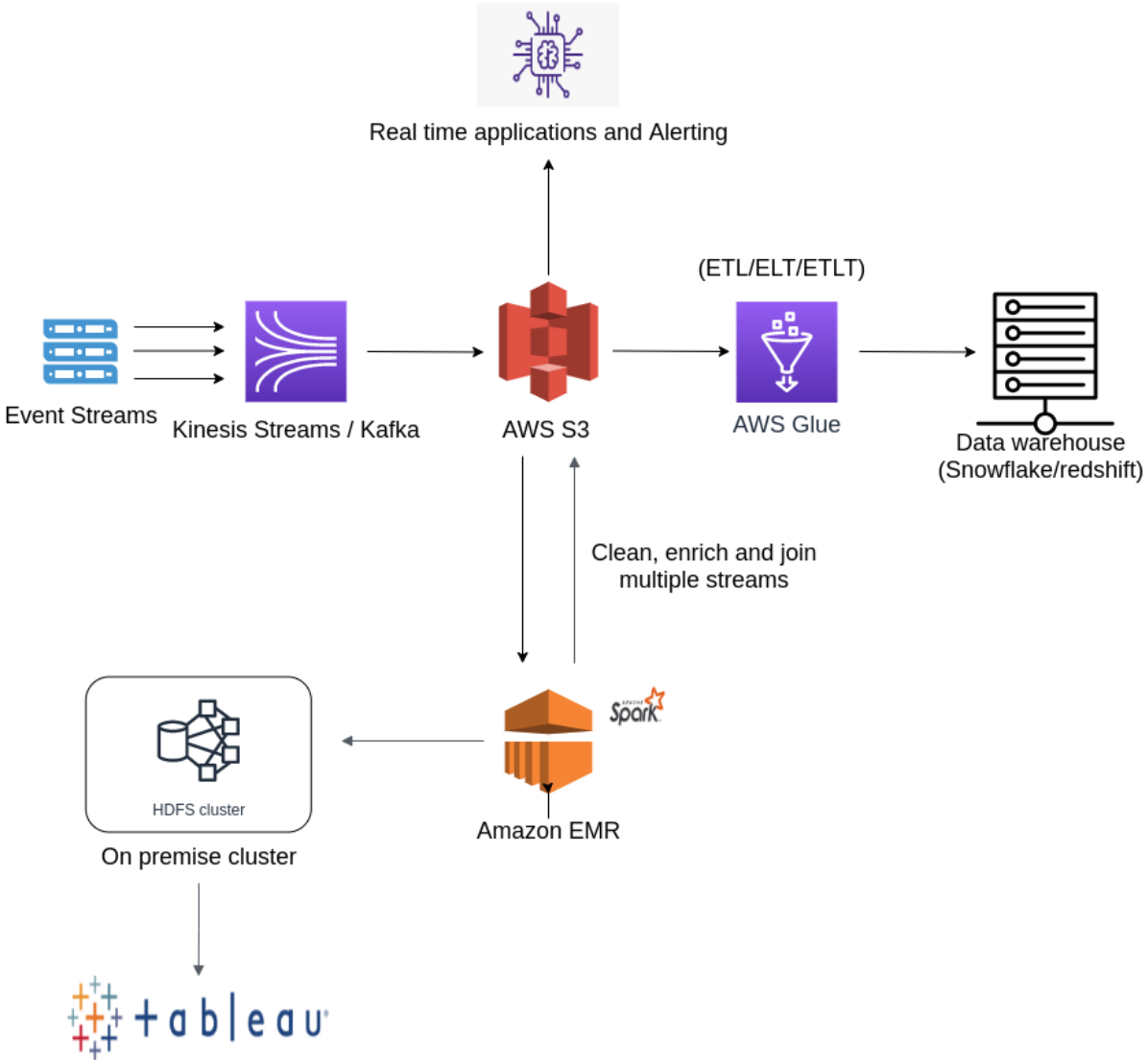

Apache Kafka VS AWS Kinesis

- **Kinesis**
  - Fully managed alternative. No infrastructure maintenance
  - Stores only 24 hours of data by default. This limit can be increased to 365 days for a cost.
  - Can create multiple shards. Each shard has a throughput limit of 1 MB/sec or 1000 PUT messages/sec.
  - Highly available with data replication to multiple availability zones
  - Much easier to set up and pay per read/write operations. No upfront infrastructure setup costs
- **Kafka**
  - Can persist data indefinitely, only limited by disk size
  - High throughputs available
  - Much more flexible than kinesis
  - Need to manage the clusters and infrastructure
  - Costs involved in setting up infrastructure

AWS S3 vs HDFS

- **AWS S3**
    - Fully managed, scales vertically automatically. No limits on storage space
    - Extremely high data durability and persistence at a very cheap cost.
    - File size limited to 5 GB
    - Low performance as latency is higher and data throughput is lower.
- **HDFS**
    - Scales horizontally but need to add more nodes manually which can be complicated and costly.
    - No limitations on file size and storage formats
    - Excellent performance. Extremely fast access and processing speeds as data is stored and processed on the same machines
    - No persistence when EC2 or EMR shuts off.
    - High costs as HDFS stores 3 copies of data as a backup by default

AWS S3 is our data lake in the architectural diagram

AWS Glue

- A fully managed ETL process that extracts, transforms, and load the data
- Glue has a schema inference feature that discovers data types and the schemas in the data using data crawlers
- Glue can also automatically generate the scripts required to move the data from s3 to managed warehouse
- Limited flexibility as it is specifically made to work with AWS services. Need to move data to these services (like S3) for the glue to function

AWS EMR

- A platform for processing large amounts of data using apache spark
- Faster than traditional apache spark
- Ability to autoscale compute and storage clusters automatically based on utilization
- Fully managed. No need to manage/ manually scale infrastructure

ETL vs ELT

- **ETL**
    - Transforming data as its moving
    - Advantage of landing data in its transformed state, which makes it possible to handle real-time scenarios
    - If transformation logic becomes complicated and/or the volume of data increases, ETL becomes very slow and not real-time
    - Hard to automate. A custom ETL process is required for each use case and every change in database and warehouse would require changing the ETL process.

- **ELT**
    - Transforming data after its moved
    - Raw data is loaded into DB. Then the target DB's powerful processing is leveraged using SQLs to do the transformations.
    - Allows us to handle a higher volume of data and perform more complex operations
    - Newer warehouses like snowflake can perform these transformations very quickly.

Snowflake vs Redshift

- **Snowflake**
    - Better support for JSON based queries
    - Pricing model based on usage patterns
    - Fully managed, no manual infrastructure maintenance required
    - Offers instant auto-scaling
- **Redshift**
    - Requires manual maintenance
    - Scaling takes minutes to add new nodes
    - Good integration with other AWS services
    - Offers better discounts for long term engagements