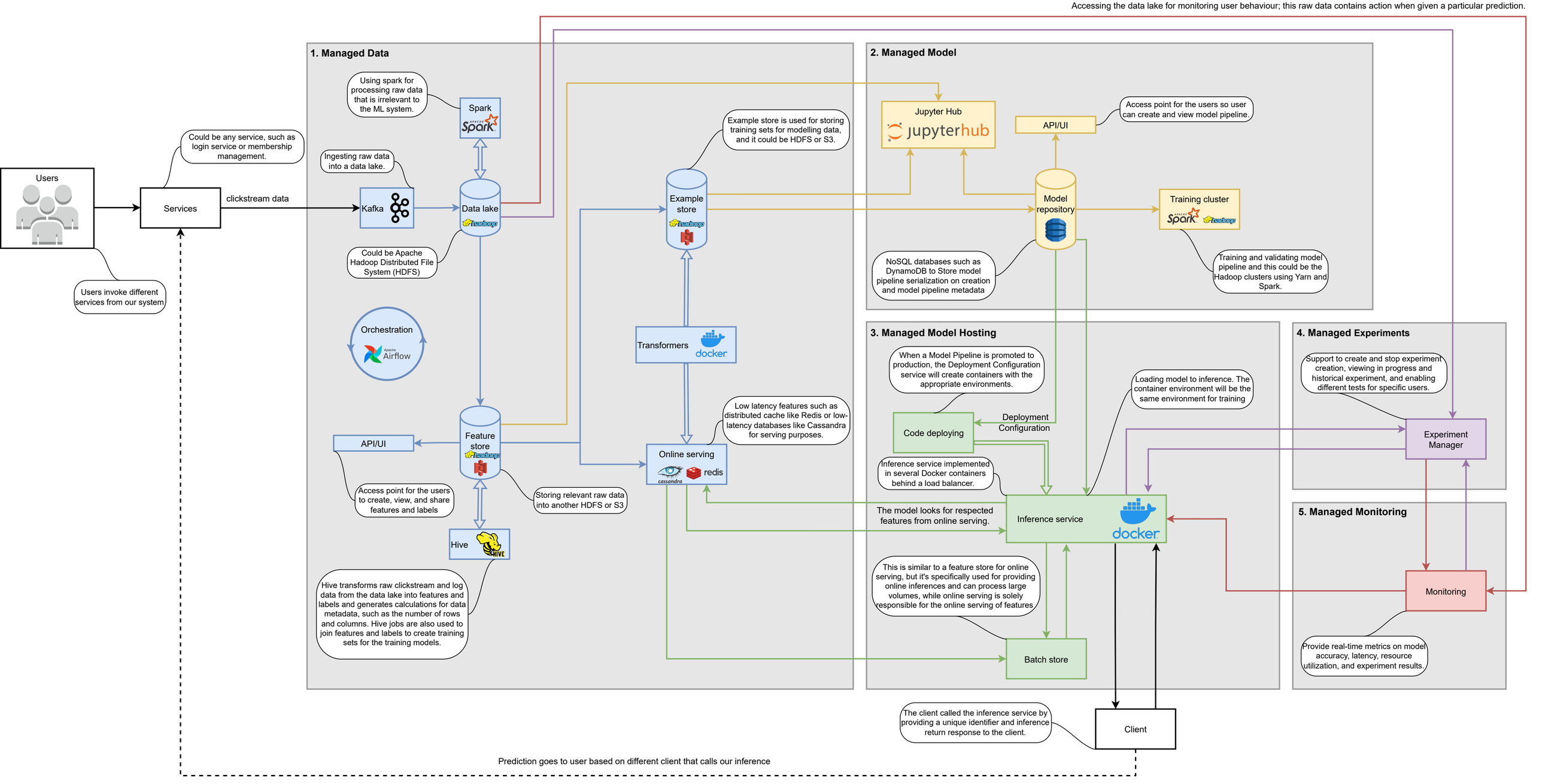
Accessing the data lake for monitoring user behaviour; this raw data contains action when given a particular prediction.
1. Managed Data
2. Managed Model
3. Managed Model Hosting
4. Managed Experiments
5. Managed Monitoring
Users
Users invoke different services from our system
Could be any service, such as login service or membership management.
Services
clickstream data
Using spark for processing raw data that is irrelevant to the ML system.
Spark
Ingesting raw data into a data lake.
Kafka
Data lake
Could be Apache Hadoop Distributed File System (HDFS)
Orchestration
Apache Airflow
Example store is used for storing training sets for modelling data, and it could be HDFS or S3.
Example store
Transformers
docker
Feature store
API/UI
Access point for the users to create, view, and share features and labels
Storing relevant raw data into another HDFS or S3
Low latency features such as distributed cache like Redis or low-latency databases like Cassandra for serving purposes.
Online serving
cassandra
redis
Hive
Hive transforms raw clickstream and log data from the data lake into features and labels and generates calculations for data metadata, such as the number of rows and columns. Hive jobs are also used to join features and labels to create training sets for the training models.
Jupyter Hub
jupyterhub
API/UI
Access point for the users so user can create and view model pipeline.
Model repository
Training cluster
Training and validating model pipeline and this could be the Hadoop clusters using Yarn and Spark.
NoSQL databases such as DynamoDB to Store model pipeline serialization on creation and model pipeline metadata
When a Model Pipeline is promoted to production, the Deployment Configuration service will create containers with the appropriate environments.
Code deploying
Deployment Configuration
Loading model to inference. The container environment will be the same environment for training
Inference service implemented in several Docker containers behind a load balancer.
The model looks for respected features from online serving.
Inference service
docker
This is similar to a feature store for online serving, but it's specifically used for providing online inferences and can process large volumes, while online serving is solely responsible for the online serving of features
Batch store
Support to create and stop experiment creation, viewing in progress and historical experiment, and enabling different tests for specific users.
Experiment Manager
Monitoring
Provide real-time metrics on model accuracy, latency, resource utilization, and experiment results.
The client called the inference service by providing a unique identifier and inference return response to the client.
Client
Prediction goes to user based on different client that calls our inference