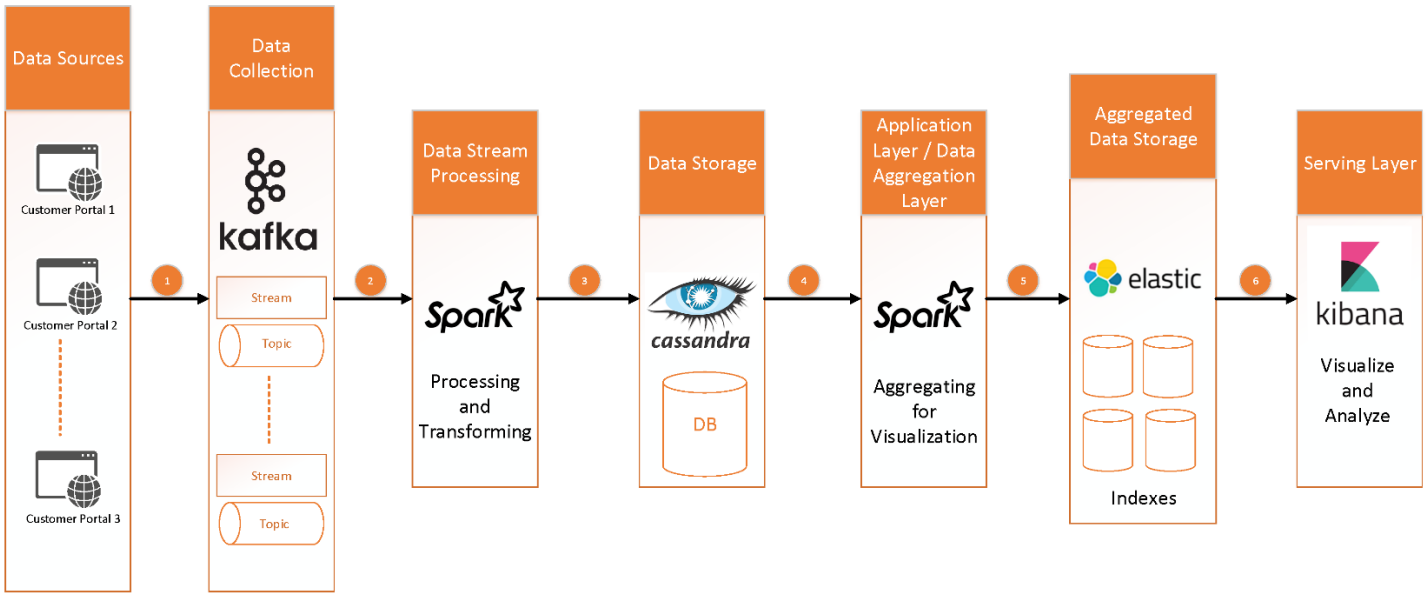
Design Question

Design A Google Analytic like Backend System. We need to provide Google Analytic like services to our customers. Please provide a high level solution design for the backend system. Feel free to choose any open source tools as you want.

Requirements

- Handle large write volume: Billions of write events per day.
- Handle large read/query volume: Millions of merchants wish to gain insight into their business. Read/Query patterns are time-series related metrics.
- Provide metrics to customers with at most one hour delay.
- Run with minimum downtime.
- Have the ability to reprocess historical data in case of bugs in the processing logic.

Proposed High Level Solution Architecture



Details of proposed solution

Layer Name	Description	Product / Software Details	Key Design Considerations
Data Sources	Customer Portals	JavaScript Framework	• A JavaScript framework will be created by us as a provider of the analytics services. • This framework will be consumed by the customer and will take care of submitting customer's analytics data to our service. • Customer is expected to make a simple function call defined within our JavaScript framework. An example could be AnalyticsService.track()
Data Collection	Streaming Topics	Kafka	• Kafka cluster(s) will serve the purpose of reliably processing the messages.

			• This layer also provides the ability to replay messages in case of failures in the system. • Details of the retention policy will be a key input to the overall sizing. • Depending on the potential sizing of the ingestion dataset (to be worked out at a granular level), the size of the cluster will be worked out. • Kafka topics will be created based on the identified categorizations. Categorizations could be on event type, region, etc.
Data Stream Processing	Processing and Transforming	Spark	• For any custom processing that requires cleaning or transforming of the data, custom services will be created in Spark. • These services will be leveraged for submission of data for the storage
Data Storage	Data Store	Cassandra	• Cassandra scales very well for large amounts of writes and reads, supporting our requirement to handle billions of write and millions of read events per day. • Cassandra is based on a master less cluster, supporting our needs of minimum downtime. • Cassandra is the overall source of truth. If for example, it is required to rebuild the indexes in Elastic, Cassandra can serve that purpose. • Cassandra can further be used to act as a source of truth for custom analytics application if requirements deem so.
Application Layer / Data Aggregation Layer	Data Aggregator	Spark	• For faster response to common reporting needs, data will be aggregated by Spark and submitted in Elastic. • Pre-aggregations will support large reads for most common requirements. • Additional aggregations can be added as needed. • This layer will support the need for reprocessing historical data in case of bugs in the processing logic.
Aggregated Data Storage	Reporting	Elastic	• Indexes will be created for respective event types. • Indexes will be designed to support the reporting needs. • Support for additional enrichment benefitting the visualization for aggregation, filtering of data.
Serving Layer	Visualization	Kibana	• Customers will be able to analyze their data using preconfigured Kibana Dashboards. • Out of box support for multiple dashboards, mingling of multiple data types, filtering, etc.

Data Flow

Step #	Step	Step Description	Component
1	Data Submission	Customer data will be submitted (published) from their web portals to the respective API endpoint.	A custom JavaScript framework will be provided to customers for submission.
2	Data Streaming	Kafka Streams for identified topics will receive the streaming data.	Kafka Cluster with multiple topics will be created.
3	Data Processing	Topic subscribers in the Spark Application (Data processing cluster) will process the streaming data.	Processing Layer will be created in Spark Application that will process and transform the data to be saved in the database.
4	Data Storage	Wide column storage will be used for storing the customers data.	Cassandra will be used to handle the billions of write events per day.
5	Data Aggregation	Spark Application (Data aggregation cluster) will read the data from Cassandra and will aggregate that data according to the visualization requirements and will store that data in Elastic Search.	Indexes will be created in Elastic Search for visualization and analysis.
6	Data Visualization	Customer can access the Data Analytics including but not limited to Graphs, charts, etc.	Kibana will be used to visualize the data retrieved from Elastic Search.

Out of Scope Considerations:

Following points were not analyzed deeply in order to define the above solution. There is a support for all of the above choices.

- Infrastructural Monitoring of the above components
- Infrastructural sizing, i.e. how many clusters, nodes etc.
- Coordination management, status tracking of cluster nodes with Zookeeper
- Security for the nodes. For instance – public facing Kafka may require to be fronted with proxy or other solutions securing the public services.