Graphical user interface, application

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **LAYER NAME** | **DESCRIPTION** | **PRODUCT DETAILS** | **DESIGN DETAIL** |
| 1 – Data sources | On-premise data | SQL Server, Files with different format, Other data sources. | We create an integration framework to connect data sources to Kafka.  Kafka-Connect provides us a connector API that gives us readily available configuration to establish JDBC connection pools with database directly and keep long running operations in terms of consistent data reads. |
| 2 – Data ingest | Streaming topics | Kafka.  Zookeeper. | Kafka is a distributed system, which is able to be scaled quickly and easily without incurring any downtime. Apache Kafka is able to handle many terabytes of data without incurring much at all in the way of overhead.  Due to the sizing of the ingestion dataset, the size of the cluster will be worked out.  Kafka topics will be created based on the identified category. In this scenario, category would be event type, products, region, transaction ID etc. |
| 3 – Data processing and Transformation | Processing and Transformation | Spark | Dataproc cluster with at least three worker nodes.  We create three Spark jobs to handle:   1. Ingesting streaming topics from Kafka cluster into GCS as arvo file format. 2. Data processing that requires cleaning or basic transformation of the data in real-time, that data is sent to bigtable as sink. 3. Extract, process and send curated/refined data to bigquery tables for Data consumers.   Few things to note:  **Late Data Handling**  We can use the .***withWatermark*** operator to handle any Late Data arriving. Although we would have to agree on the threshold for processing time. |
| 4 – Data storage | Storage | BigTable | BigTable will serve as our Data warehouse. |
| 5 - Curated Data storage | Ready-to-use Data | BigQuery.  Optional -BigQuery ML. | Curated/refined data will be stored as tables on bigquery for Data consumers. Since bigquery offers large read/query volume, this was the best choice.  Visit to see the benefits of using BQ: <https://xo.xello.com.au/blog/google-bigquery-5-benefits-cloud-data-warehouse>  Visit to see the benefits of using bigquery ML: <https://cloud.google.com/bigquery-ml/docs/introduction> |
| 6 – Data Consumers | Virtualization tools, ML applications, Business intelligence | PowerBi, Tableau, ML models. etc | Consumers will be able to analyze data using configured Dashboards. |
| 7 – ML Engine | Machine Learning Engine | Google Cloud ML Engine | Optional – I believe we can further get insights on the data to help promote business decisions.  This Layer is for advanced enterprises on AI journey who have data scientists and ML experts inhouse to train ML models using TensorFlow, Cloud TPUs, GPUs and ML engine |

Note: In this design, the following were out of scope. I have added a few considerations.

* Infrastructural Monitoring of each components.

We can use google cloud monitoring and other third party tools like Site24x7, Dynatrace, PagerDuty(with GCP Stackdriver integration)

* Infrastructural sizing i.e., Number of clusters, nodes etc.
* Security within the above pipeline and On-premise to GCP connection.

Google Cloud KMS, Google Cloud IAM, Google Cloud Identity, Stackdriver Logging are all tools that help with security. We can discuss this further as most security practices depends on the organization policy. I will list some best practices for consideration:

* Use fully managed google accounts which are tied to your corporate domain name through Cloud Identity.
* Control access to resources using google groups instead of individual accounts and use Service Accounts where applicable.
* Use Custom VPC Network, avoid using ‘Default’ firewall rules instead create your own custom firewall rules which brings loads of control about how you can group and isolate related resources with combination of the network and firewall rules.