Ravi.K
ravi2105.k@gmail.com
331-356-2184

Professional Summary:

- Over 8+ years of experience in Data Engineering, Data Pipeline Design, Development and Implementation as a Sr. Data Engineer/Data Developer.
- Involved in creating database objects like tables, views, procedures, triggers, and functions using T-SQL to provide definition, structure and to maintain data efficiently.
- Hands-on use of Spark and Scala APIs to compare the performance of Spark with Hive and SQL, and Spark SQL to manipulate Data Frames in Scala.
- Experience in developing Map Reduce Programs using Apache Hadoop for analyzing the big data as per the requirement.
- Experience in setting up Hadoop clusters on cloud platforms like AWS and GCP.
- Extensive experience working with AWS Cloud services and AWS SDKs to work with services like AWS API Gateway, Lambda, S3, IAM and EC2.
- Strong experience in Software Development Life Cycle (SDLC) including Requirements Analysis, Design Specification and Testing as per Cycle in both Waterfall and Agile methodologies.
- Strong experience on Druid which is built to handle high-velocity, high-dimensional data, making it ideal for real-time monitoring, user behavior analysis, and ad optimization. It is designed to handle both event-driven data and time series data.
- Experience in Data Security and Privacy to protect the data from Unauthorized access and to ensure compliance with privacy regulations, implement encryption measures and to establish data management and Governance.
- Have Extensive Experience in IT data analytics projects, Hands on experience in migrating on premise ETLs to Google Cloud Platform (GCP) using cloud native tools such as BIG query, Cloud Data Proc, Google Cloud Storage, Composer.
- Experience working in different Google Cloud Platform Technologies like Big Query, Dataflow, Data proc, Pub sub, Airflow.
- Building and deploying data processing pipelines that can scale to handle high data volumes is possible with Flink.
- Experience in designing star schema, Snowflake schema for Data Warehouse, ODS architecture.
- Experience working with GitHub/Git 2.12 source and version control systems.
- Experienced in using Alteryx to connect and integrate data from various sources, including databases, files, APIs, and other data repositories.
- Implementing Data Modeling that Identifies the data that will be used in the project is the first step. This can be achieved by looking at the project requirements and figuring out what data is required to satisfy them. The relationships between the data elements must be understood after

the data has been identified. This can be achieved by building a data model that illustrates the connections between the data elements.

- Implementing data access in application because Spring Boot supports a variety of data access technologies, such as JDBC, JPA, and NoSQL databases. Authentication, authorization, and secure communication are just a few of the many security features that Spring Boot offers to make sure that application is safe and secure from unauthorized access.
- Using Alteryx to handle complex data engineering challenges efficiently, streamline workflows, and deliver high-quality data for analysis and decision-making and can improve data integration, transformation, and quality assurance processes by leveraging its capabilities, resulting in better insights and outcomes for the organization.
- Using orchestration tools like Apache Airflow, Luigi, or Prefect to schedule and manage data workflows and also Ensure dependencies and task execution order are properly managed.
- Used Databricks XML plug-in to parse the incoming data in the XML format and generate the required XML as output.
- Using GCE which can host your data warehouse infrastructure, supplying the compute resources required for storage and analysis. On VM instances, you can run data warehouse solutions such as Google BigQuery, Google Cloud SQL, or other relational databases. GCE's performance and scalability ensure that your data warehouse environment can meet data analytics and reporting requirements.
- The streaming SQL engine for Apache Kafka is called KSQL. It offers a user-friendly yet potent interactive SQL interface for stream processing on Kafka without requiring programmers to write code in a language like Java or Python. Scalable, elastic, fault-tolerant, real-time, and KSQL.
- Hands on Spark MLlib utilities such as including classification, regression, clustering, collaborative filtering, dimensionality reduction.
- Using Flink Processed the Real time streaming data, which helps for use cases like processing IoT data, Fraud detection, real-time analytics.
- Data can be ingested and processed from a variety of sources, including cloud storage, Kafka and Pub/sub using Flink’s stream processing capabilities.
- HDFS, Hive, Sqoop, Pig, Oozie, HBase, NiFi, Spark, Scala, Kafka and Zookeeper and ETL (Data Stage).
- Experience in designing star schema, Snowflake schema for Data Warehouse, ODS architecture.
- Using Google Cloud Data Fusion, a fully managed data integration service, to build and manage ETL pipelines visually.
- Experience in Data Analysis, Data Profiling, Data Integration, Migration, Data governance and Metadata Management, Master Data Management and Configuration Management.
- Designing and developing MDM solutions using Informatica MDM tool. Developing Custom software components to support MDM.

- Experience in work with tools like AWK, GREP, SED and other Unix shell scripting techniques to extract, transform and Load data from various sources.
- Data engineering tasks may benefit from Scala's support for functional programming paradigms. Higher-order functions, which can aid in data transformations, filtering, and aggregations, can be used, along with immutability, help to write clear, concise code. Knowledge about cloud dataflow and Apache beam.
- Expert in building Enterprise Data Warehouse or Data warehouse appliances from Scratch using both Kimball and Inmon’s Approach.
- Experienced
- Experienced in supply chain applications which assist businesses in managing their inventory, procurement, logistics, transportation, and overall supply chain operations.
- Experienced with Docker and Kubernetes on multiple cloud providers, from helping developers build and containerize their application (CI/CD) to deploying either on public or private cloud.
- Looking to create and manage containerized data processing and storage systems, OpenShift can be a powerful tool. Using that can facilitate the development and administration of data engineering systems due to its scalability, ease of deployment, resource management features, security, and integration capabilities.
- Evaluating the current on-prem environment before moving to the new one and to identify any potential problems. Identifying dependencies, assessing compatibility with the new environment, and making sure that data is adequately backed up and secured are a few examples of what this may entail.
- A thorough understanding of data processing, distributed systems, and real-time analytics is necessary for working with Druid. Additionally, it calls for knowledge of data modeling, data warehousing, and data visualization techniques as well as proficiency in programming languages like Java, Python, and SQL.
- Involved in daily SCRUM meetings to discuss the development/progress of Sprints and was active in making scrum meetings more productive.
- Analyzed data and provided insights with R Programming and Python Pandas
- Expertise in Business Intelligence, Data warehousing technologies, ETL and Big Data technologies.
- Experience in Creating ETL mappings using Informatica to move Data from multiple sources like Flat files, Oracle into a common target area such as Data Warehouse.
- Experience in writing PL/SQL statements - Stored Procedures, Functions, Triggers and packages.

Experience:

Verizon- Irving, TX De2022- Till date

Migrating the data from On-premises to GCP Bigquery. Creating some data pipelines, Normalize the data and store in GCS buckets. We are running some Python code using

Dataproc and pushing the final processed data in to Bigquery. In the pipelines we implemented data quality checks also.

Role: Senior GCP Data Engineer

Responsibilities:

- Developed pipelines for auditing the metrics of all applications using GCP Cloud functions, and dataflow for a plot project.
- Using rest API with Python to ingest Data from and some other site to BIGQUERY.
- Build a program with Python and Apache beam and execute it in cloud Dataflow to run Data validation between raw source file and Big query tables.
- Process and load bound and unbound Data from Google pub/subtopic to Big query using cloud Dataflow with Python.
- Design and implement large scale distributed solutions in GCP.
- Implemented Linux features on windows through VM's, UI's and cloud shell in GCP.
- As part of GCP implemented a spring boot app using IntelliJ IDE and deployed it to Google APP Engine
- Developed end-to-end pipeline, which exports the data from parquet files in Cloud Storage to GCP Cloud SQL.
- Loading salesforce Data every 15 min on incremental basis to BIGQUERY raw and UDM layer using SOQL, Google Data Proc, GCS bucket, HIVE, Spark, Scala, Python, Gsutil and Shell Script.
- Files extracted from Hadoop and dropped on daily hourly basis into S3.
- Diagnosed and resolve issues related to Alteryx workflows, ensuring the stability and reliability of data processes and also Schedule and monitor data pipelines to ensure timely and accurate data processing.
- Automated the data processing with Oozie to automate data loading into the Hadoop Distributed File System.
- Using rest API with Python to ingest Data from and some other site to BIGQUERY.
- Good experience in using Relational databases Oracle, Devops, SQL, Server and PostgreSQL
- Using Flink performing some Data transformation and enrichment tasks like Data cleaning, aggregation and joining can be performed. The Flink APIs can be used to create unique enrichments and transformations that can be applied to streaming data.
- Using Unix Shell scripting which enables to automate the tasks by writing scripts that execute multiple commands and using some tools like 'cron' to schedule the execution of scripts at some intervals of time.
- Utilizing GCP services like Cloud Dataflow, Data proc and Kubernetes engine, Flink can be deployed so this makes it possible to deploy and scale Flink Based data processing pipelines using GCP's managed services.

- Developed a Python Script to load the CSV files into the S3 buckets and created GCP S3buckets, performed folder management in each bucket, managed logs and objects within each bucket.
- Implemented monitoring with Prometheus and Grafana to visualize pipeline performance and detect failures.
- Developing the architecture of Druid is built around the concepts of data ingestion, indexing, and query processing. Loading data into Druid's memory-based data stores, which are designed for fast write performance, is known as data ingestion. A column-oriented storage format and a distributed indexing architecture are used to index and optimize data for fast query performance. Query processing entails running queries on the indexed data and returning results in milliseconds.
- Working as a Data engineer with GCP may entail a variety of tasks such as data ingestion, transformation, processing and analysis. Scala can be used in the conjustion with GCP services to efficiently complete these tasks..
- To create meaningful visualisations, tableau frequently relies on centralised data sources. we can assist in the development of data integration pipelines that extract data from various systems, transform it into a unified format, and load it into a centralised repository like a data warehouse.
- Using Kafka and integrating with the Spark Streaming.
- Developed Airflow DAGs in python by importing the Airflow libraries.
- Expertise in Snowflake to create and maintain tables and views.
- Experienced in accessing additional Google services such as Google Cloud Storage, Data Flow, Pub/sub and others. These services work in Tandem with Google Compute Engine, allowing you to create comprehensive data engineering solutions with my work environment.
- Creating GCP bucket using Google Cloud Console, the command line interface.
- Experience in using Cloud composer which benefit from scalability, reliability and automation it offers and manage the data very effectively.
- Experiencing In Continuous Integration/Continuous Deployment (CI/CD) pipelines can be used to streamline the development process and guarantee that code changes are successfully tested, built, and deployed to production. To manage code changes, make use of a version control system (VCS), such as Git. This will make it easier to keep track of changes over time, work with team members, and go back to earlier versions if necessary. Example Like we can store our code in Git repositories using Cloud Source Repositories, build and package it is using Cloud Build, and store artifacts in Cloud Storage.
- In order to help visualize data and offer insights for decision-making, I had used this power BI in this project which can benefit from the use of Power BI, a potent business intelligence tool. I had used Data fusion can be used to assemble data from different sources into a single view. This can assist in locating data dependencies and relationships, which is helpful when organizing a migration.

- Using Alteryx to design and orchestrate data pipelines that use GCP services. Alteryx's visual workflow capabilities can be combined with GCP services such as Google Cloud Dataflow or Cloud Composer to create and manage data processing workflows.
- Used Apache airflow in GCP composer environment to build data pipelines and used various airflow operators like bash operator, Hadoop operators and python callable and branching operators.
- Developed Sqoop Jobs to load data from RDBMS, External Systems into HDFS and HIVE.
- Druid is designed to handle real-time and batch data streams, as well as provide analytical queries with sub-second query latencies. It uses a distributed architecture to scale horizontally across multiple nodes and stores data in a column-oriented fashion.
- Using Scala which is a Programming language that can be used to create monitoring and alerting systems for GCP infrastructure. I can easily build scalable and reactive applications that monitor GCP resources and trigger alerts based on predefined criteria using libraries like Akka or play Framework.
- Using Flink to process data in batches which helps for dealing with large amounts of data that cannot be processed in real-time. Flink's Batch processing features can be used to process data from variety of sources including Cloud storage and Big query.
- Building a Scala and spark based configurable framework to connect common Data sources like MYSQL, Oracle, Postgres, SQL Server, Salesforce, Big query and load it in big query.
- Creating and implementing data pipelines, which entails putting the data ingestion procedure in place to allow data to be brought from various sources into the Druid data store. Designing and implementing ETL procedures might be necessary for this.
- Maintained and developed Docker images for a tech stack including Cassandra, Kafka, Apache and several in house written Java services running in Google Cloud Platform (GCP) on Kubernetes.
- Mainly Druid works with a wide range of data sources, including real-time event streams, logs, and databases, and it integrates with other tools in the data analytics ecosystem, including Apache Kafka, Apache Spark, and Apache Superset.

Environment: Pig, Kafka, Alteryx, HBase, Cassandra, GCP, SQL, Python, Mongo DB, Spark, Hive, Scala, Hadoop, Oozie, PySpark, NOSQL, CI/CD Pipelines, Snowflake, Airflow, Spark-SQL, Sqoop, Tableau, Power BI, Redshift, SQL, Spring boot, Server and PostgreSQL

WellCare Health Plans - Tampa, FL Dec 2021-Dec 2022

I am responsible for delivering Data warehouse, ETL and reporting solutions for all the users. My job was to build a data pipeline that combines two data sources, normalizes them, and places them in three different layers in AWS. The goal was to centralize and improve efficiency by consolidating all data into a single SQL Server database in Amazon RDS. The key challenge in this project is Data Compatibility and Schema Mapping. The use of complex ETL pipelines and Jenkins automation allowed for efficient data movement, transformation, and deployment. The result was a centralized and scalable

database solution for HealthEquity, which improved data management and operational efficiency.

Role: Big Data Engineer

Responsibilities:

- Extensively worked with Spark-SQL context to create data frames and datasets to pre-process the model data.
- Involved in data migration project for multiple applications from on-prem to AWS.
- Design and Develop ETL Processes in AWS Glue to migrate Campaign data from external sources like S3, ORC/Parquet/Text Files into AWS Redshift.
- Data Extraction, aggregations and consolidation of Adobe data within AWS Glue using PySpark.
- Developed and executed a migration strategy to move Data Warehouse from an Oracle platform to AWS Redshift.
- Create external tables with partitions using Hive, AWS Athena and Redshift
- Ingested terabytes of click stream data from external systems like FTP Servers and S3 buckets into HDFS using custom Input Adaptors.
- Created a multi-threaded Java application running on edge node for pulling the raw click stream data from FTP servers and AWS S3 buckets.
- Used HDFS File System API to connect to FTP Server and HDFS, S3 AWS SDK for connecting to S3 buckets.
- Implemented installation and configuration of multi-node cluster on the cloud using Amazon Web Services (AWS) on EC2.
- Responsible for building and configuring distributed data solution using Map distribution of Hadoop.
- Created Partitioning, Bucketing, and Map Side Join, Parallel execution for optimizing the hive queries decreased the time of execution from hours to minutes.
- Implemented data pipeline using Spark, Hive, Sqoop and Kafka to ingest customer behavioral data into Hadoop platform to perform user behavioral analytics.
- Worked with cloud provisioning team on a capacity planning and sizing of the nodes (Master and Slave) for an AWS EMR Cluster.
- Involved in build applications using Maven and integrated with CI servers like Jenkins to build jobs.
- Experience in Microservice Architecture with Spring Boot and Docker
- Developed and maintained batch data flow using HiveQL and Unix scripting.
- Here I had used Data Validation techniques which is used to make sure there are no data discrepancies, Power BI can be used to compare the migrated data to the source data which can support ensuring the data migration's accuracy.
- Involved in converting Hive/SQL queries into Spark transformations using Spark RDD, Scala and Python.

- Using Containerization techniques Like Docker, Kubernetes, LXC/LXD can be very useful for deploying and managing Data processing and storage systems. Especially when dealing with big data processing requirements and complex data processing workflows.
- Specified the cluster size, allocating Resource pool, Distribution of Hadoop by writing the specification texts in JSON File format.
- Worked as L1 support on Jira requests for Kafka.
- Worked on Topic partitioning and replication.
- Configured documentations for Kafka to operate effectively.
- Created a Producer application that sends API messages over Kafka.
- Defined API security key and other necessary credentials to run Kafka architecture.
- Wrote python code that tracks Kafka message delivery.
- Developed Map Reduce programs in Java for parsing the raw data and populating staging Tables.
- Experienced in working with spark ecosystem using Spark SQL and Scala queries on different formats like text file, CSV file.
- Using scala used to create data ingestion pipelines for integrating data from multiple sources in to your big data infrastructure. Scala libraries such as Apache kafka and Apache NiFi can be used to consume and process data streams before storing them in distributed storage systems such as Hadoop Distributed File System, Apache Hbase or Apache Cassandra.
- Using Open Shift containerization to control sizable data processing and storage systems. We can easily deploy and manage containers across a cluster of machines with the aid of OpenShift, which is advantageous for dealing with growing data loads and processing demands.
- Used Hadoop YARN to perform analytics on data in Hive.
- Developed spark code and spark-SQL/streaming for faster testing and processing of data.
- Build a program with Python and Apache beam and execute it in cloud Dataflow to run Data validation between raw source file and big query tables.
- Worked with ETL tools Including Talend Data Integration, Talend Big Data, Pentaho Data Integration and Informatica
- Involved in building a real time pipeline using Kafka and Spark streaming for delivering event messages to downstream application team from an external rest-based application.
- Developed Spark jobs using Scala for faster real-time analytics and used Spark SQL for querying.
- Implemented Spark using Scala and utilizing Data frames and Spark SQL API for faster processing of data.

  Environment: Python, NOSQL, ETL, Hadoop, Scala, Spring boot, SQL, Parquet, Spark, Hive, Json, Kafka, PowerBI, map reduce Docker, Containerization techniques, Open-shift Containerization, Pig, Jira.

Citi Bank- Pleasanton, CA Sep 2019- Nov 2021

Role: Data Engineer
Responsibilities:

- Implemented Partitioning, Dynamic Partitions, Buckets in HIVE.
- Installed/Configured/Maintained Apache Hadoop clusters for application development and Hadoop tools like Hive, Pig, Zookeeper and Sqoop.
- Used AWS EMR to transform and move large amounts of data into and out of other AWS data stores and databases, such as Amazon Simple Storage Service (Amazon S3) and Amazon Dynamo DB.
- I had used Data Visualization techniques which Power BI can be used to develop interactive reports and dashboards that are visually appealing and offer insights into the migrated data. It helps me to aid decision-makers in comprehending the data and making wise choices.
- Configured, deployment, and support of cloud services in Amazon Web Services (AWS).
- Experienced working on cloud AWS using EMR Performed operations on AWS using EC2 instances, S3 storage, performed RDS, analytical Redshift operations and wrote various data normalization jobs for new data ingested into Redshift by building multi-terabyte of data frame.
- Developed Airflow DAGs in python by importing the Airflow libraries.
- Configured Spark Streaming to receive real time data from the Kafka and store the stream data to HDFS.
- Worked on to retrieve the data from FS to S3 using spark commands.
- Experienced in working with spark ecosystem using Spark SQL and Scala queries on different formats like text file, CSV file.
- Created AWS Lambda, EC2 instances provisioning on AWS environment and implemented security groups, administered Amazon VPC's.
- Created Hive base script for analyzing requirements and for processing data by designing cluster to handle huge amount of data for cross examining data loaded in Hive and Map Reduce jobs.
- Implemented End to End solution for hosting the web application on AWS cloud with integration to S3 buckets.
- Worked on AWS CLI Auto Scaling and Cloud Watch Monitoring creation and update.
- Allotted permissions, policies and roles to users and groups using AWS Identity and Access Management (IAM).
- Developed server-side software modules and client-side user interface components and deployed entirely in Compute Cloud of Amazon Web Services (AWS).
- Implemented Lambda to configure Dynamo DB Auto scaling feature and implemented Data Access Layer to access AWS Dynamo DB data.
- Creating structured and scalable data pipelines to capture, process, and store data generated by supply chain applications. This may entail using technologies such as Apache Spark or Apache Kafka to handle large amounts of data, as well as ensuring data quality and optimizing pipeline performance.

- Worked on AWS Services like AWS SNS to send out automated emails and messages using BOTO3 after the nightly run.
- Configured AWS Lambda with multiple functions.
- Perform Data Cleaning, features scaling, and features engineering using pandas and NumPy packages in python.
- Implementations of generalized solution model using AWS Sage Maker
- Integrated spark streaming service with Kafka to load the data into an HDFS location.
- Used Kafka HDFS connector to export data from Kafka topic to HDFS files in a variety of formats and integrates with Apache hive to make data immediately available for HQL querying.
- Good experience in writing Spark applications using Python.
- Performed ETL using AWS Glue.
- Developed spark code and spark-SQL/streaming for faster testing and processing of data.
- Selected and generated data into csv files and stored them into AWS S3 by using AWS EC2 and then structured and stored in AWS Redshift.
- Ensuring Data Integrity and quality within supply chain applications by implementing some data cleansing techniques like Data Validation checks and Data profiling processes.
- Designed and developed Security Framework to provide fine grained access to objects in AWS S3 using AWS Lambda, Dynamo DB.
- Developed data pipeline using Flume, Sqoop, Pig and Java Map Reduce to ingest customer behavioral data into HDFS for analysis.

Environment: Scala, NOSQL, Hadoop, Map reduce, Spark-SQL, Spark, Kafka, SQL, Power BI, HDFS, Hive.

Optum, India Aug 2016- Feb 2019

Role: Hadoop Developer

Responsibilities:

- Responsible for data extraction and data ingestion from different data sources into Hadoop Data Lake by creating ETL pipelines using Pig, and Hive.
- Responsible for importing data to HDFS using Sqoop from different RDBMS servers and exporting data using Sqoop to the RDBMS servers after aggregations for other ETL operations.
- Experience in designing and developing applications in Spark using python to compare the performance of Spark with Hive.
- Used Jira for ticketing and tracking issues and Jenkins for continuous integration and continuous deployment.
- Tested Apache TEZ, an extensible framework for building high performance batch and interactive data processing applications, on Pig and Hive jobs.
- Implemented Partitioning, Dynamic Partitions and Buckets in HIVE for efficient data access.

- Experience in using GCP which provides management tools like stack driver which allows to monitor the performance and health of Hadoop clusters.
- Create/Modify shell scripts for scheduling various data cleansing scripts and ETL load process.
- Developed testing scripts in Python and prepare test procedures, analyze test results data and suggest improvements of the system and software.
- Responsible for importing data to HDFS using Sqoop from different RDBMS servers and exporting data using Sqoop to the RDBMS servers.
- Involved in establishing automated Hadoop Integration testing system and implementing Oozie workflow.
- Created Hadoop jobs for processing and analyzing millions of records of data.
- Developed shell scripts to validate Hadoop daemon services and reported accordingly to any warning or failure conditions.
- Validated the data load process for Hadoop using the HiveQL queries.
- Created Partitioned and Bucketed Hive tables in Parquet File Formats with Snappy compression and then loaded data into Parquet hive tables from Avro hive tables.
- Involved in running all the hive scripts through hive. Hive on Spark and some through Spark SQL.
- Written Kafka REST API to collect events from front end.
- Developed data pipeline using Sqoop to ingest customer behavioral data and purchase histories into HDFS for analysis.

Environment: HDFS, GCP, Hive, SQL, Spark, Kafka, Pig, Hadoop.

TECHNICAL SKILLS:

Data Modeling Tools: Erwin Data Modeler, ER Studio v17

Programming Languages: SQL, PL/SQL, UNIX, Spring boot

Methodologies: RAD, JAD, System Development Life Cycle (SDLC), Agile

Cloud Platform: AWS, Azure, Google Cloud.

Databases: Oracle 12c/11g, Teradata R15/R14.

OLAP Tools: Tableau, SSAS, Business Objects, and Crystal Reports 9

ETL/Data warehouse Tools: Informatica 9.6/9.1, Altryx and Tableau.

Operating System: Windows, Unix, Sun Solaris, Mac OS

Big Data Tools: Hadoop

Ecosystem: Map Reduce