Setting up a big data streaming service involves integrating various tools and services to ensure efficient and reliable data processing. In this scenario, we will utilize Kafka, Spark, NiFi, and Airflow to manage data from transactional SQL databases and analytical NoSQL databases. Here's a step-by-step guide on how I would set up the big data streaming service:

1. Install and Configure Kafka

- Download and install Kafka from the Apache Kafka website.

- Configure the Kafka broker properties, including the broker ID, listeners, and advertised listeners.

- Set up the ZooKeeper ensemble to manage the Kafka brokers.

- Start the Kafka server and verify its functionality.

1. Integrate Kafka with Transactional Databases

- Configure Kafka Connect to connect to the SQL-based transactional databases.

- Use appropriate connectors such as JDBC Source and JDBC Sink connectors to ingest data from and write data to the databases.

- Set up appropriate configurations for the connectors, including connection details and data mapping.

1. Implement Spark for Data Processing

- Install and configure Apache Spark to process the data from Kafka topics.

- Utilize the Kafka-Spark integration to read data from Kafka topics in real-time.

- Develop Spark Streaming applications to perform transformations and analytics on the streaming data.

- Integrate Spark with SQL to perform complex analytics and queries on the data.

1. Configure NiFi for Data Flow Management

- Install and configure Apache NiFi to orchestrate the data flow between different systems.

- Set up NiFi processors to handle data ingestion, routing, transformation, and storage.

- Configure NiFi to interact with Kafka for both data ingestion and delivery.

- Implement appropriate data processors for data conversion, enrichment, and routing.

1. Use Airflow for Workflow Management

- Install and configure Apache Airflow for workflow management and scheduling.

- Create DAGs (Directed Acyclic Graphs) to define the workflow for data processing.

- Integrate Airflow with Kafka, Spark, and NiFi to trigger data processing tasks based on predefined schedules or event-based triggers.

- Monitor and manage the data pipeline using Airflow's UI and logging functionalities.

1. Integrate Analytical NoSQL Databases

- Configure Kafka Connect to integrate with NoSQL databases such as MongoDB, Cassandra, or HBase.

- Utilize appropriate connectors to stream data from Kafka topics to the NoSQL databases for analytics and storage.

- Set up data mappings and configurations to ensure seamless data transfer and compatibility between Kafka and NoSQL databases.

1. Ensure Security and Fault Tolerance

- Implement security measures such as encryption, authentication, and authorization for data transfer and access.

- Set up Kafka clusters with replication and fault-tolerant configurations to ensure data reliability and availability.

- Implement backup and recovery strategies for both the transactional and analytical databases.

1. Monitor and Optimize Performance

- Set up monitoring tools to track the performance and health of the data pipeline components.

- Monitor Kafka, Spark, NiFi, and NoSQL databases for data throughput, latency, and errors.

- Optimize the data pipeline by fine-tuning configurations, adjusting resource allocation, and improving data processing algorithms.

By following these steps, I can effectively set up a robust big data streaming service using Kafka, Spark, NiFi, and Airflow to manage data from both transactional SQL databases and analytical NoSQL databases. It's crucial to continuously monitor and optimize the data pipeline to ensure efficient and reliable data processing and analysis.