Fibr Corp Data Engineer Assessment

1. You are a Data Engineer at a large marketplace company that processes massive amounts of transactional data, including user interactions, sales, and inventory management. Your task is to design the data lifecycle from source data ingestion to the data warehouse, utilizing a Big Data Distributed System within the Hadoop Ecosystem.
    a. What tools would you choose to build your Big Data Distributed System, and why?
    b. Please create a diagram of your Big Data Distributed System using draw.io.
    c. How would you monitor the performance of your system? Explain the tools and methods you would use.

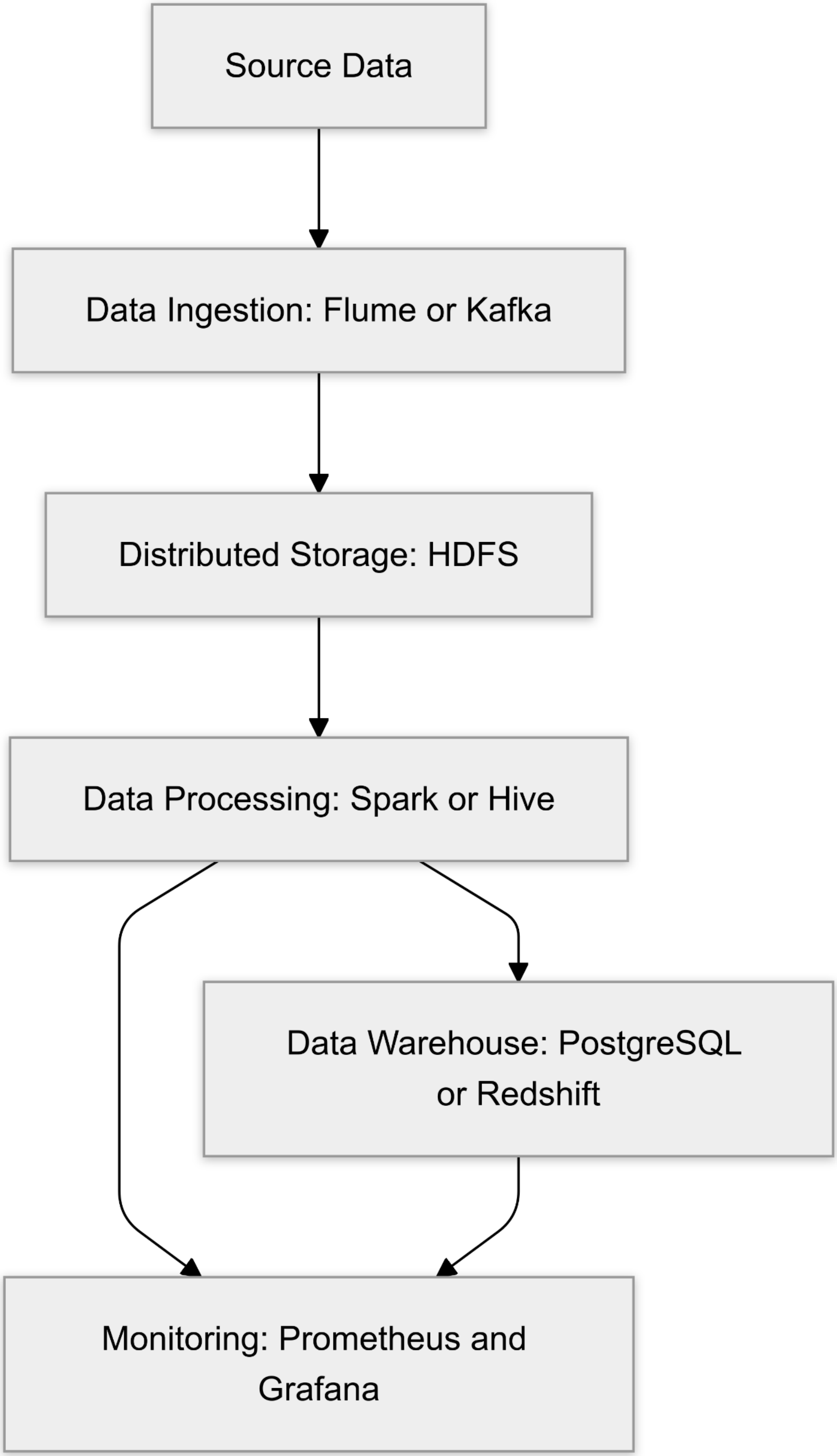
2. Your company requires regular access to exchange rate data between the US Dollar (USD) and the Indonesian Rupiah (IDR) to support financial analytics and decision-making processes.
   Your task is to identify a reliable online data source, scrape the data, and ingest it into your data warehouse. You are required to write a Python script to accomplish this.

1. A. To build a Big Data Distributed System within the Hadoop Ecosystem, I would use the following tools:
    - HDFS (Hadoop Distributed File System)**:** For distributed storage of large datasets.
    - Apache Kafka**:** For real-time data ingestion and streaming.
    - Apache Flume or Sqoop**:** For batch data ingestion from relational databases or log files.
    - Apache Hive**:** For data warehousing and running SQL-like queries over large datasets.
    - Apache Spark**:** For distributed data processing and analytics due to its speed and efficiency over MapReduce.
    - Apache Oozie**:** For scheduling and managing workflows in the data pipeline.
    - Zookeeper**:** For distributed system coordination, ensuring fault tolerance and synchronization.

    Why?

    - These tools are scalable, fault-tolerant, and integrate seamlessly within the Hadoop Ecosystem.
    - They support both batch and real-time data processing, which is critical for handling transactional data.
    - They are open-source and widely adopted in the industry.

B.


Source Data
Data Ingestion: Flume or Kafka
Distributed Storage: HDFS
Data Processing: Spark or Hive
Data Warehouse: PostgreSQL or Redshift
Monitoring: Prometheus and Grafana

C. To monitor system performance, I would use:

- Apache Ambari**:** To manage and monitor Hadoop clusters, providing an easy-to-use interface for performance metrics.
- Prometheus and Grafana**:** For collecting metrics like CPU usage, memory consumption, and processing latency, and visualizing them in dashboards.
- Logs**:** Using tools like the ELK Stack (Elasticsearch, Logstash, Kibana) for log analysis and alerting.
- Hadoop-specific tools**:** Metrics provided by the NameNode, ResourceManager, and DataNode.

Methods:

- Alerts: Set thresholds for critical metrics (e.g., HDFS storage utilization, Spark job failures) and configure alerts.
- Dashboards: Create real-time dashboards in Grafana to track system health.
- Automated Recovery: Use scripts or managed services to automatically recover from failures.