**Building a Data Processing Pipeline with Hadoop and Hive**

**Hadoop Learning Path Capstone**

**Overview:**

Welcome to your capstone project! In this project, you will build a data processing pipeline using Apache Hadoop and Apache Hive. This project will empower you to:

* **Set Up a Hadoop Cluster**: Install and configure a single-node Hadoop cluster with HDFS, YARN, and MapReduce.
* **Integrate External Data Sources**: Fetch and store data from an external API or dataset into HDFS.
* **Implement Data Transformation**: Use Apache Hive to perform ETL operations, transforming raw data into meaningful insights.
* **Create Analytical Queries**: Write and execute HiveQL queries to analyze the data and generate reports.

By the end of this project, you'll have hands-on experience with Hadoop's ecosystem, data ingestion, and data analysis using Hive.

**Project Structure:**

Your project will be organized into the following components:

1. **Hadoop Configuration Files**:
   * core-site.xml: Configures HDFS settings.
   * hdfs-site.xml: Sets HDFS-specific configurations.
   * yarn-site.xml: Configures YARN resource management.
   * mapred-site.xml: Configures MapReduce settings.
2. **Data Ingestion Script** (data\_ingestion.py or data\_ingestion.ipynb):
   * Fetches data from an external API or dataset.
   * Processes and stores the data into HDFS.
3. **Hive Scripts**:
   * **Hive Table Creation Script** (create\_tables.hql): Defines the schema and creates Hive tables.
   * **Data Transformation Script** (transform\_data.hql): Performs ETL operations using HiveQL.
   * **Analytical Queries Script** (analysis\_queries.hql): Contains queries for data analysis and reporting.
4. **Supporting Documentation**:
   * **Setup Guide** (SETUP.md): Provides detailed instructions for setting up the Hadoop environment.
   * **README File** (README.md): Offers an overview, usage guidelines, and explanations of the project's components.

**Functional Requirements:**

To successfully complete this project, ensure that your application meets the following functional requirements:

**1. Hadoop Cluster Setup**

* **a. Install Java Development Kit (JDK)**:
  + Ensure that JDK 8 or later is installed on your system.
* **b. Install Hadoop**:
  + Download the latest stable version of Hadoop.
  + Extract and configure Hadoop in a specified directory.
* **c. Configure Environment Variables**:
  + Set HADOOP\_HOME, JAVA\_HOME, and update the PATH variable accordingly.
* **d. Configure Hadoop Files**:
  + **core-site.xml**:
    - Set the default filesystem to HDFS running on localhost:9000.
  + **hdfs-site.xml**:
    - Set the replication factor to 1 (since it's a single-node cluster).
  + **yarn-site.xml**:
    - Configure YARN to enable the MapReduce shuffle service.
  + **mapred-site.xml**:
    - Set the MapReduce framework to run on YARN.
* **e. Start Hadoop Services**:
  + Start HDFS and YARN daemons.
  + Verify that the NameNode and ResourceManager web interfaces are accessible.

**2. Data Ingestion Script (data\_ingestion.py or data\_ingestion.ipynb)**

* **a. External Data Integration**:
  + **Choose a Dataset**: Select a public API or dataset that interests you (e.g., weather data, stock prices, social media data).
  + **API Configuration**: Set up any necessary API keys or access tokens.
* **b. Data Fetching and Processing**:
  + Implement functionality to fetch data from the chosen source.
  + Parse and preprocess the data as needed (e.g., JSON parsing, data cleaning).
* **c. Store Data in HDFS**:
  + Save the processed data to a local file in a suitable format (e.g., CSV, JSON, text).
  + Create necessary directories in HDFS.
  + Upload the data file to HDFS for use with Hive.
* **d. Error Handling and Logging**:
  + Include error handling for network issues, API rate limits, and data parsing errors.
  + Log the status of data fetching and storage operations.

**3. Hive Setup and Data Transformation**

* **a. Install Apache Hive**:
  + Download and install the latest stable version of Apache Hive.
  + Configure Hive environment variables (HIVE\_HOME, update PATH).
* **b. Configure Hive Metastore**:
  + Use the embedded Derby database for the metastore.
  + Configure hive-site.xml accordingly.
  + Create the Hive warehouse directory in HDFS.
* **c. Create Hive Tables**:
  + **External Table**:
    - Define an external table that maps to the data file stored in HDFS.
    - Specify the correct data formats and delimiters.
  + **Managed Tables**:
    - Create managed tables to store transformed data.
* **d. Data Transformation using HiveQL**:
  + Write HiveQL scripts to perform ETL operations, such as:
    - Filtering records based on specific criteria.
    - Aggregating data (e.g., calculating averages, counts).
    - Joining tables if multiple datasets are used.
  + Save these scripts as .hql files.
* **e. Execute Hive Scripts**:
  + Run the Hive scripts to transform and analyze the data.
  + Ensure that the output is stored in Hive tables or exported to HDFS directories.

**4. Data Analysis and Visualization**

* **a. Analytical Queries**:
  + Develop HiveQL queries to extract meaningful insights from the data.
  + Examples include trend analysis, statistical summaries, or pattern detection.
* **b. Retrieve Query Results**:
  + Export the results of analytical queries to local files or HDFS.
* **c. Visualization (Optional but Recommended)**:
  + Use tools like Apache Zeppelin, Jupyter Notebook, or external visualization libraries to create charts or graphs.
  + Visualize trends, patterns, or anomalies in the data.

**5. Supporting Documentation**

* **a. Setup Guide (SETUP.md)**:
  + Provide step-by-step instructions to set up the Hadoop and Hive environment.
  + Include prerequisites, installation commands, and configuration steps.
  + Troubleshooting tips for common setup issues.
* **b. README File (README.md)**:
  + Offer an overview of the project objectives and components.
  + Explain how to run the data ingestion script and Hive scripts.
  + Document any dependencies and how to install them.
  + Include usage examples and expected outcomes.

**Additional Requirements and Best Practices:**

* **Version Control**:
  + Use Git to track changes in your project.
  + Include a .gitignore file to exclude unnecessary files (e.g., logs, temporary files).
* **Code Quality**:
  + Write clean, readable code with appropriate comments.
  + Follow best practices for Python scripting if using Python.
* **Modularity**:
  + Organize your scripts and configurations logically.
  + Separate configuration variables from code logic where possible.
* **Error Handling**:
  + Implement robust error handling in your scripts.
  + Ensure that exceptions are caught, and meaningful messages are logged.
* **Performance Considerations**:
  + Optimize Hive queries for better performance (e.g., use partitioning, indexing if applicable).
  + Be mindful of resource utilization when working with large datasets.

**Optional Enhancements:**

* **Use Advanced Hive Features**:
  + Implement partitions or buckets in your Hive tables.
  + Explore using Hive UDFs (User-Defined Functions) for custom transformations.
* **Connect to a Relational Database**:
  + Configure Hive to use an external metastore like MySQL or PostgreSQL.
  + Migrate the embedded Derby metastore to a production-ready database.
* **Implement Security Features**:
  + Set up user authentication and authorization for your Hadoop cluster.
  + Configure network policies or firewall settings to secure your data.
* **Automate Workflow with Scripts**:
  + Create shell scripts to automate the starting and stopping of Hadoop services.
  + Script the execution of Hive queries for batch processing.

**Deliverables:**

1. **Configuration Files**:
   * Modified Hadoop configuration files (core-site.xml, hdfs-site.xml, yarn-site.xml, mapred-site.xml).
   * Hive configuration file (hive-site.xml).
2. **Data Ingestion Script**:
   * A Python script (data\_ingestion.py) or Jupyter Notebook (data\_ingestion.ipynb) for data fetching and storage.
3. **Hive Scripts**:
   * create\_tables.hql: Script to create Hive tables.
   * transform\_data.hql: Script for data transformation.
   * analysis\_queries.hql: Script containing analytical queries.
4. **Documentation**:
   * SETUP.md: Detailed setup guide.
   * README.md: Project overview and usage instructions.
5. **Visualization Outputs** (Optional):
   * Plots, charts, or dashboards created from the analyzed data.

**Evaluation Criteria:**

Your project will be evaluated based on the following criteria:

* **Completeness**: All functional requirements are met, and deliverables are provided.
* **Correctness**: The Hadoop cluster and Hive are correctly configured, and scripts run without errors.
* **Code Quality**: Scripts are well-written, documented, and follow best practices.
* **Documentation**: Clear and comprehensive setup instructions and project explanations are provided.
* **Analysis Depth**: The data analysis demonstrates a good understanding of the dataset and provides meaningful insights.
* **Optional Enhancements**: Implementation of optional features will be considered positively.

**Getting Started:**

1. **Plan Your Project**:
   * Choose a dataset or API that interests you.
   * Outline the ETL process and identify the transformations you wish to perform.
2. **Set Up the Environment**:
   * Install Java, Hadoop, and Hive following the setup guide.
   * Configure all necessary environment variables and configuration files.
3. **Develop the Data Ingestion Script**:
   * Write the script to fetch and store data in HDFS.
   * Test the script to ensure data is correctly stored.
4. **Create Hive Tables and Perform ETL**:
   * Write Hive scripts to create tables and transform data.
   * Execute the scripts and verify the results.
5. **Analyze and Visualize the Data**:
   * Run analytical queries to extract insights.
   * Create visualizations to represent your findings.
6. **Document Your Work**:
   * Update the SETUP.md and README.md files with detailed instructions and explanations.
   * Ensure that your documentation is clear and helpful for someone replicating your project.

**Submission Guidelines:**

* **Repository**: Host your project on a GitHub repository or similar platform.
* **Files**: Include all scripts, configuration files, and documentation.
* **Instructions**: Ensure that your README.md provides clear instructions on how to run your project.
* **Screenshots** (Optional): Include screenshots of your Hadoop web interfaces, Hive CLI, or visualizations.

**Conclusion:**

This capstone project is an excellent opportunity to apply what you've learned about Hadoop and Hive in a practical scenario. By integrating external data sources and performing ETL operations, you'll gain valuable experience in big data processing and analytics.

Good luck, and enjoy the journey of building your data processing pipeline!

**Resources:**

**Key Code:** [**https://github.com/JUMP-TA/HadoopExercises/blob/main/HadoopWithHive.ipynb**](https://github.com/JUMP-TA/HadoopExercises/blob/main/HadoopWithHive.ipynb)