Day-3(Hadoop)

1. Write a Python program to read a Hadoop configuration file and display the core components of Hadoop.
2. Implement a Python function that calculates the total file size in a Hadoop Distributed File System (HDFS) directory.
3. Create a Python program that extracts and displays the top N most frequent words from a large text file using the MapReduce approach.
4. Write a Python script that checks the health status of the NameNode and DataNodes in a Hadoop cluster using Hadoop's REST API.
5. Develop a Python program that lists all the files and directories in a specific HDFS path.
6. Implement a Python program that analyzes the storage utilization of DataNodes in a Hadoop cluster and identifies the nodes with the highest and lowest storage capacities.
7. Create a Python script that interacts with YARN's ResourceManager API to submit a Hadoop job, monitor its progress, and retrieve the final output.
8. Create a Python script that interacts with YARN's ResourceManager API to submit a Hadoop job, set resource requirements, and track resource usage during job execution.
9. Write a Python program that compares the performance of a MapReduce job with different input split sizes, showcasing the impact on overall job execution time.

Submission Guidelines:

- Answer all the questions in a single Jupyter Notebook file (.ipynb).
- Include necessary code, comments, and explanations to support your answers and implementation.
- Ensure the notebook runs without errors and is well-organized.
- Create a GitHub repository to host your assignment files.
- Rename the Jupyter Notebook file using the format "date_month_topic.ipynb" (e.g., "12_July_Hadoop.ipynb").
- Place the Jupyter Notebook file in the repository.

- Commit and push any additional files or resources required to run your code (if applicable) to the repository.
- Ensure the repository is publicly accessible.
- Submit the link to your GitHub repository as the assignment submission.

Grading Criteria:

1. Understanding and completeness of answers: 40%
2. Clarity and depth of explanations: 25%
3. Correct implementation and evaluation of optimizer techniques: 15%
4. Analysis and comparison of different optimizers: 10%
5. Proper code implementation and organization: 10%

Note:- Create your assignment in Jupyter notebook and upload it to GitHub & share that uploaded assignment file link through your dashboard. Make sure the repository is public.