REAL TIME ETHEREUM BLOCK DATA DASHBOARD

Introduction

The Ethereum Block Data Analytics Application is a robust and comprehensive solution designed to fetch, store, and analyze Ethereum blockchain data. Leveraging the Alchemy API, MongoDB, and Streamlit, this project provides real-time and historical insights into Ethereum block data, enabling users to explore various blockchain metrics and trends through an interactive web interface.

Purpose

The primary goal of this project is to offer a seamless and efficient way to monitor and analyze Ethereum block data. By continuously retrieving block information and storing it in a database, the application allows users to visualize and understand key blockchain metrics, aiding in research, development, and decision-making processes related to blockchain technology.

Components

1. **Backend:**
    - Retrieves Ethereum block data from the Alchemy API.
    - Stores the data in MongoDB.
    - Scheduled to run every 15 seconds to ensure data is up-to-date.
    - FastAPI
2. **Database:**
    - MongoDB is used to store block data as documents, enabling efficient querying and analysis.

3. **Frontend:**
   - Two Streamlit applications for data visualization and analysis:
     - block_data.py: Displays data related to a specific block.
     - dashboard.py: Provides a broader view of blockchain metrics over time.

Technologies Used

- **Alchemy API:** For accessing Ethereum blockchain data.
- **MongoDB:** As a NoSQL database to store block data.
- **Streamlit:** For creating interactive Frontend.
- **Docker:** For containerizing the backend and frontend applications to ensure consistent and isolated environments.
- **Python:** The primary programming language for backend data retrieval and frontend applications.
- **(FastAPI) :** Web framework
- **Plotly:** To visualize data and plot graphs

How to Run

1. **Clone the repository** **(https://github.com/ATHARVA1202/Real-Time-Ethereum-Block-Dashboard)**

2. **Set up the environment variables**

   a. **In the 'Backend' folder :**

      i. modify the **.env** file with your API keys and MongoDB URI

      ii. Also modify the '**config.py**' file present in the '**app**' folder by inserting **your 'DB_NAME'** and **'COLLECTION_NAME'**

   b. **In the 'Frontend' folder :**

      In files **blockdata.py** and **dashboard.py** modify the **line no. 8, 9 and 10** with **your MongoDB URI, DB Name and Collection Name respectively**

3. **Run the containers using Docker**

   a. Run Docker Desktop to run the Docker engine

   b. Go to the project directory in the terminal and run "**docker-compose up**" command

   c. Now you will see the project running and data being stored in the database periodically at an interval of 15 sec

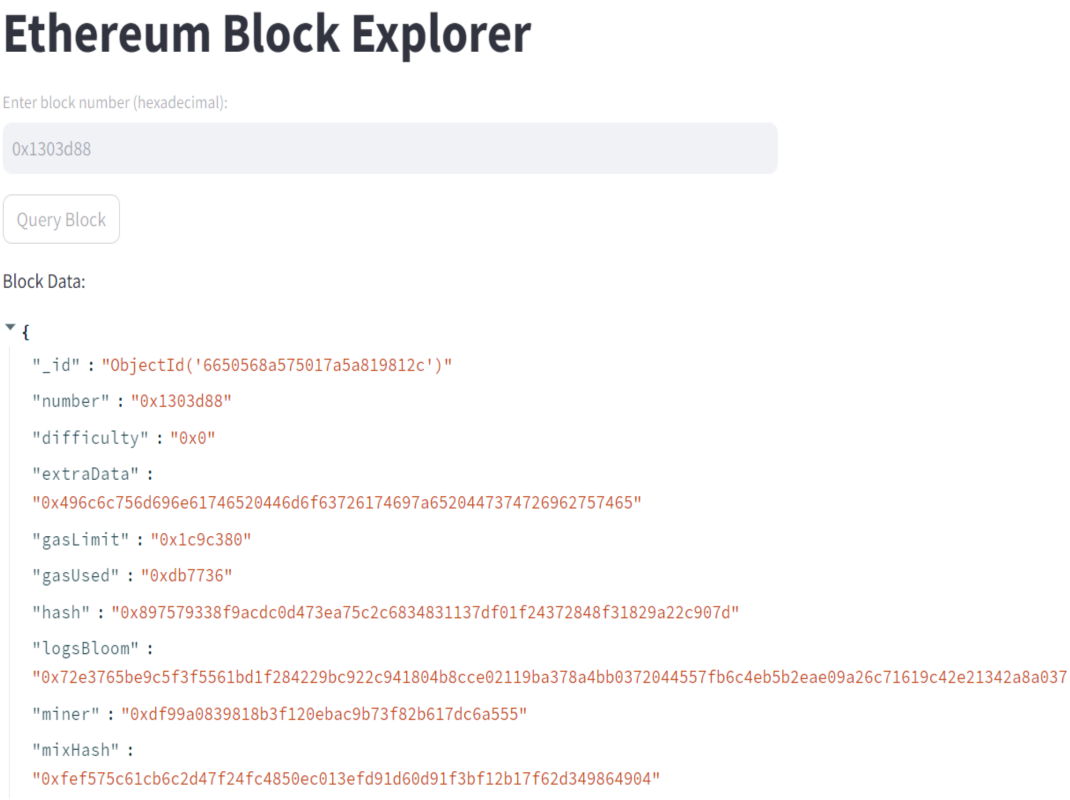
      ```
      backend-1   | DEBUG:root:Block data stored in MongoDB
      ```

   d. To access the frontend, data visualizations and the graphs :

      i. **For block-data : http://localhost:8501/**

ii. **For the blockchain data and metrics across all Ethereum blocks : http://localhost:8502/**

Screenshots :



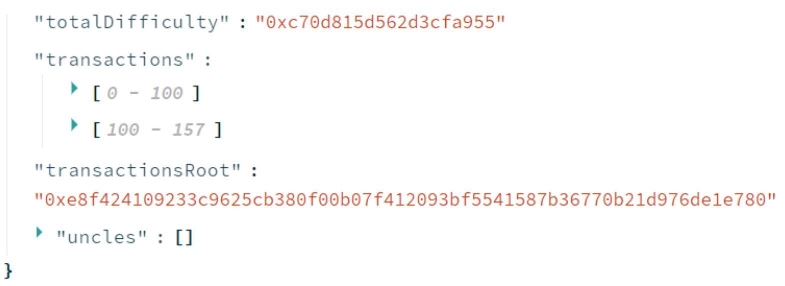
Ethereum Block Explorer

Enter block number (hexadecimal):

0x1303d88

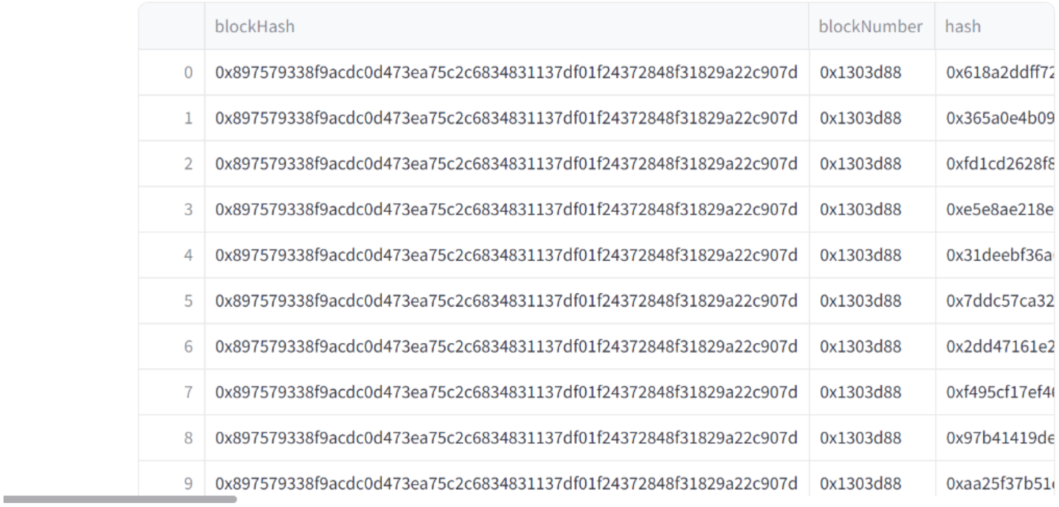
Block Data:

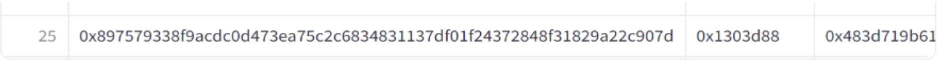
```
▾ {
    "_id" : "ObjectId('6650568a575017a5a819812c')"
    "number" : "0x1303d88"
    "difficulty" : "0x0"
    "extraData" :
    "0x496c6c756d696e61746520446d6f63726174697a6520447374726962757465"
    "gasLimit" : "0x1c9c380"
    "gasUsed" : "0xdb7736"
    "hash" : "0x897579338f9acdc0d473ea75c2c6834831137df01f24372848f31829a22c907d"
    "logsBloom" :
    "0x72e3765be9c5f3f5561bd1f284229bc922c941804b8cce02119ba378a4bb0372044557fb6c4eb5b2eae09a26c71619c42e21342a8a037
    "miner" : "0xdf99a0839818b3f120ebac9b73f82b617dc6a555"
    "mixHash" :
    "0xfef575c61cb6c2d47f24fc4850ec013efd91d60d91f3bf12b17f62d349864904"
```



```
  "totalDifficulty" : "0xc70d815d562d3cfa955"
  "transactions" :
     [ 0 - 100 ]
     [ 100 - 157 ]
  "transactionsRoot" :
  "0xe8f424109233c9625cb380f00b07f412093bf5541587b36770b21d976de1e780"
   "uncles" : []
}
```

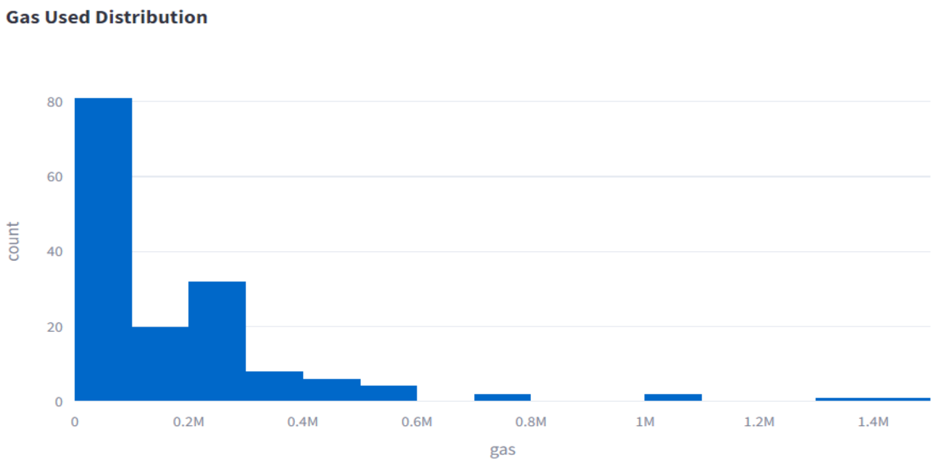
Transactions DataFrame:





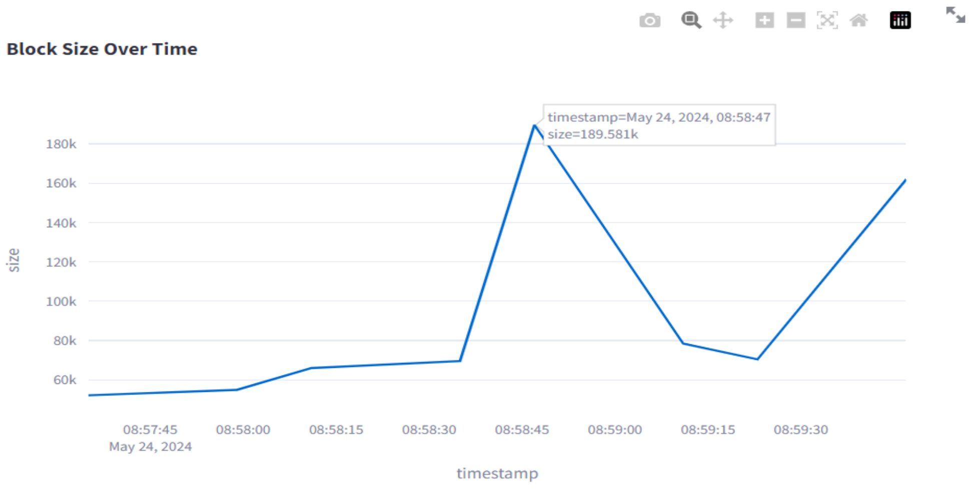
	blockHash	blockNumber	hash
0	0x897579338f9acdc0d473ea75c2c6834831137df01f24372848f31829a22c907d	0x1303d88	0x618a2ddff72
1	0x897579338f9acdc0d473ea75c2c6834831137df01f24372848f31829a22c907d	0x1303d88	0x365a0e4b09
2	0x897579338f9acdc0d473ea75c2c6834831137df01f24372848f31829a22c907d	0x1303d88	0xfd1cd2628f
3	0x897579338f9acdc0d473ea75c2c6834831137df01f24372848f31829a22c907d	0x1303d88	0xe5e8ae218e
4	0x897579338f9acdc0d473ea75c2c6834831137df01f24372848f31829a22c907d	0x1303d88	0x31deebf36a
5	0x897579338f9acdc0d473ea75c2c6834831137df01f24372848f31829a22c907d	0x1303d88	0x7ddc57ca32
6	0x897579338f9acdc0d473ea75c2c6834831137df01f24372848f31829a22c907d	0x1303d88	0x2dd47161e2
7	0x897579338f9acdc0d473ea75c2c6834831137df01f24372848f31829a22c907d	0x1303d88	0xf495cf17ef4
8	0x897579338f9acdc0d473ea75c2c6834831137df01f24372848f31829a22c907d	0x1303d88	0x97b41419de
9	0x897579338f9acdc0d473ea75c2c6834831137df01f24372848f31829a22c907d	0x1303d88	0xaa25f37b51
25	0x897579338f9acdc0d473ea75c2c6834831137df01f24372848f31829a22c907d	0x1303d88	0x483d719b61

Gas Used


Gas Used Distribution

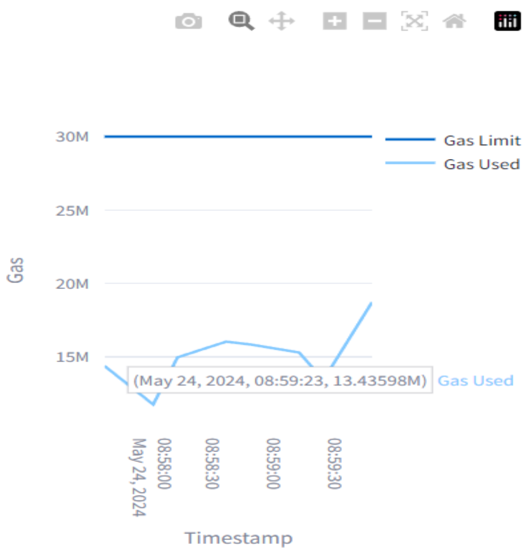
Ethereum Block Data Visualization

Block Size Over Time

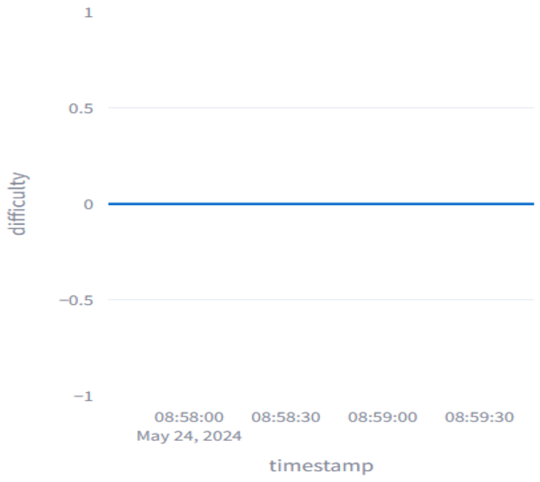


Gas Usage and Difficulty Over Time

Gas Used vs Gas Limit

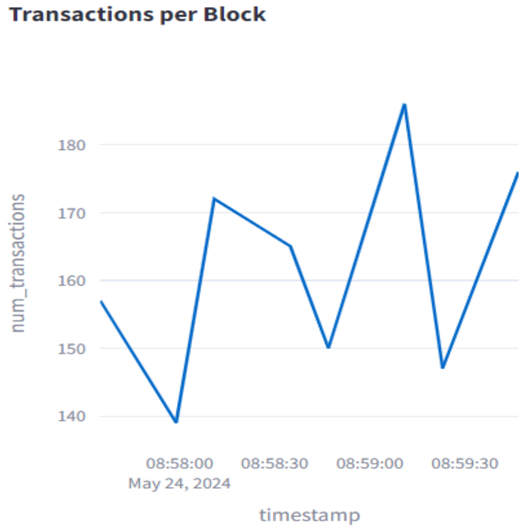


Difficulty Over Time

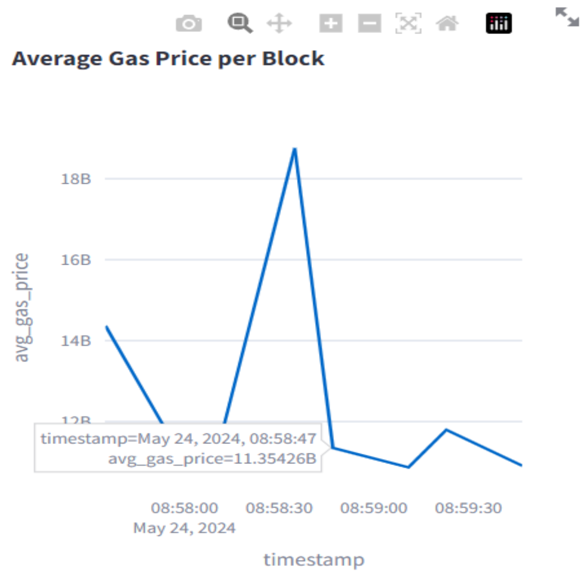


Transactions and Gas Price

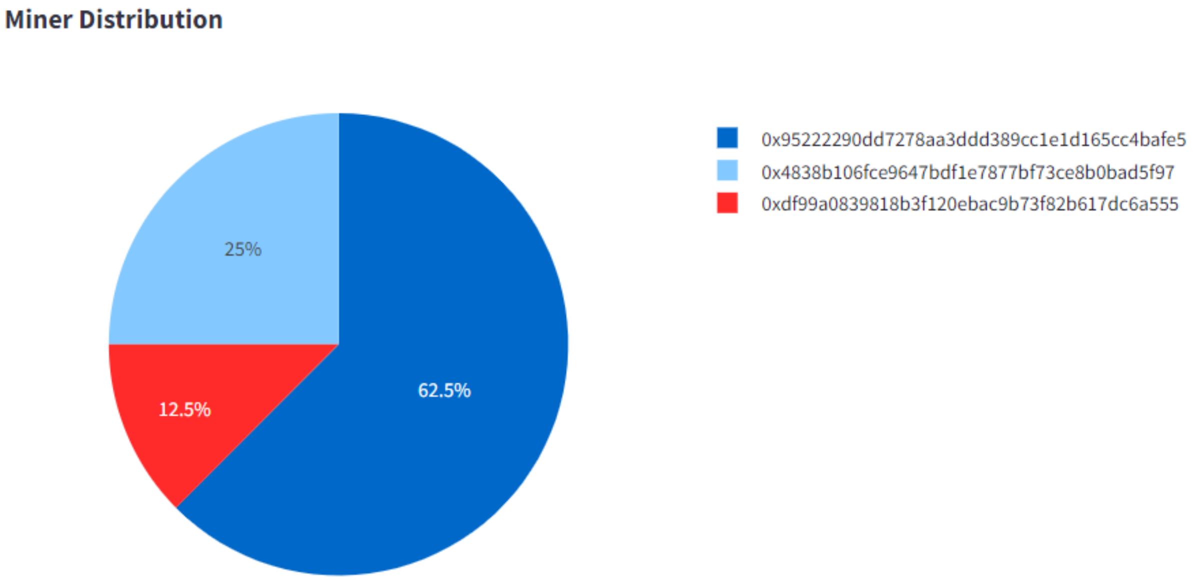
Transactions per Block



Average Gas Price per Block



Miner Distribution



Demonstration Video : https://drive.google.com/drive/folders/1WzgUvJJhIi-KobgWkiboEK10tGjqJ4uM