

This is a step-by-step guide for running this project. Follow each step as outlined here.

1. Prerequisites and dataset

Note

The project root directory refers to: `C:\Users\XXX\Desktop\realtime_fraud_detection` or similar to this

Device requirement

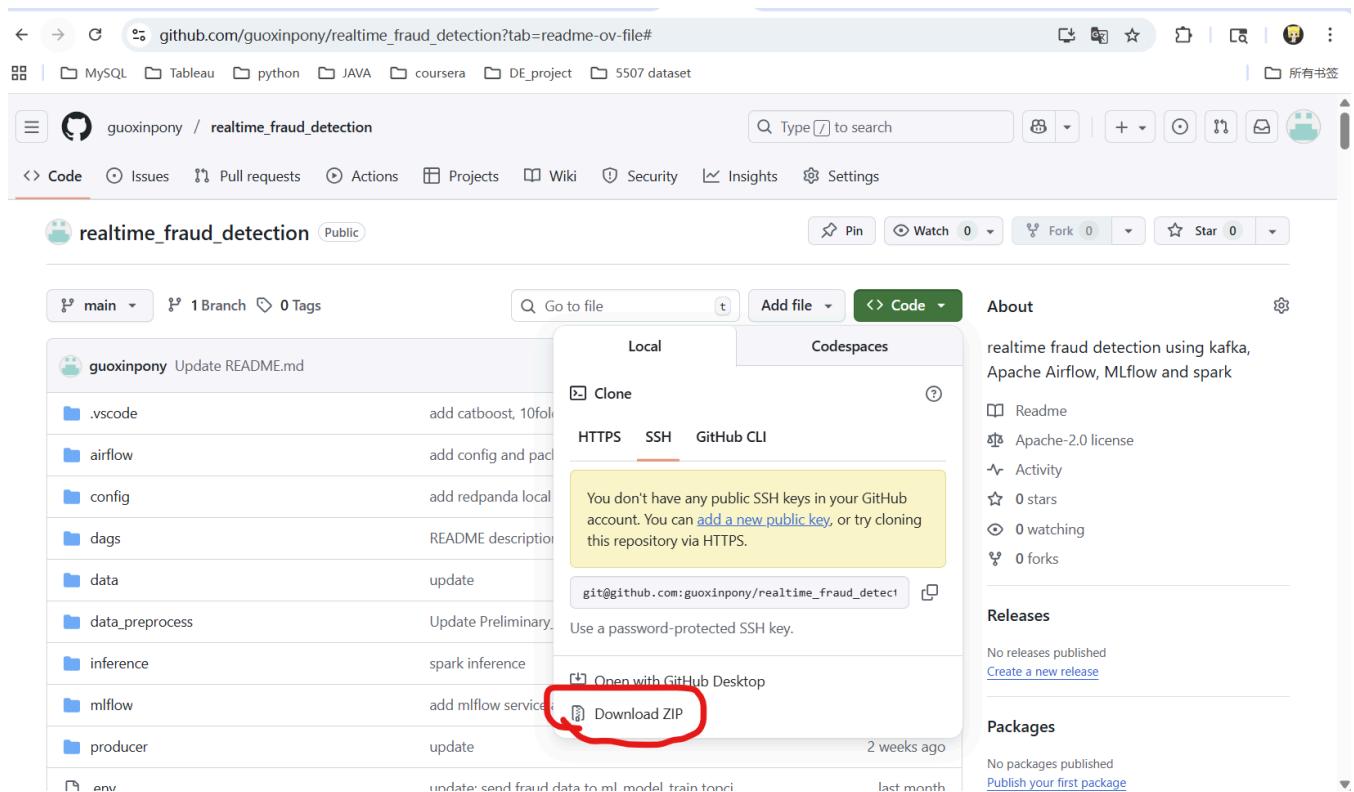
- **CPU:** 4+ cores recommended
- **Memory:** 8GB minimum, 16GB recommended
- **OS:** Linux, macOS, or Windows with WSL2

Clone or download project:

```
git clone https://github.com/guoxinpony/realtime_fraud_detection.git
```

OR

Download ZIP:



Install Docker Desktop:

1. download package from: <https://www.docker.com/products/docker-desktop/>
2. Install docker desktop. If you are using Windows system, the installation process will prompt you to enable WSL2. Follow the instructions to enable WSL2 and restart your computer.
3. Launch Docker Desktop

Download Dataset:

1. Download from: https://drive.google.com/file/d/1y1QqL1BdJKMpEu4dOB5OKANPeUxIkI3X/view?usp=drive_link, and place the dataset in the data directory within the project.

OR

2. Download from Kaggle: <https://www.kaggle.com/datasets/kartik2112/fraud-detection>, including two datasets: fraudTest.csv and fraudTrain.csv; and place the two datasets in the data directory within the project; and RUN in location of project root directory:

```
python ./data_preprocess/merge_data.py
```

MAC OS/ Linux required: Change permission of Script in location of project root directory:

```
chmod +x wait-for-it.sh
```

2.Docker image build

The initial build takes some time, which includes downloading the official image, necessary packages, and the build process itself. The exact duration depends on your computer's performance and network connection.

In the project root directory, build the following four images airflow-webserver, mlflow-server, producer, and inference. Input in the terminal:

```
docker compose build airflow-webserver
```

```
docker compose build mlflow-server
```

```
docker compose build producer
```

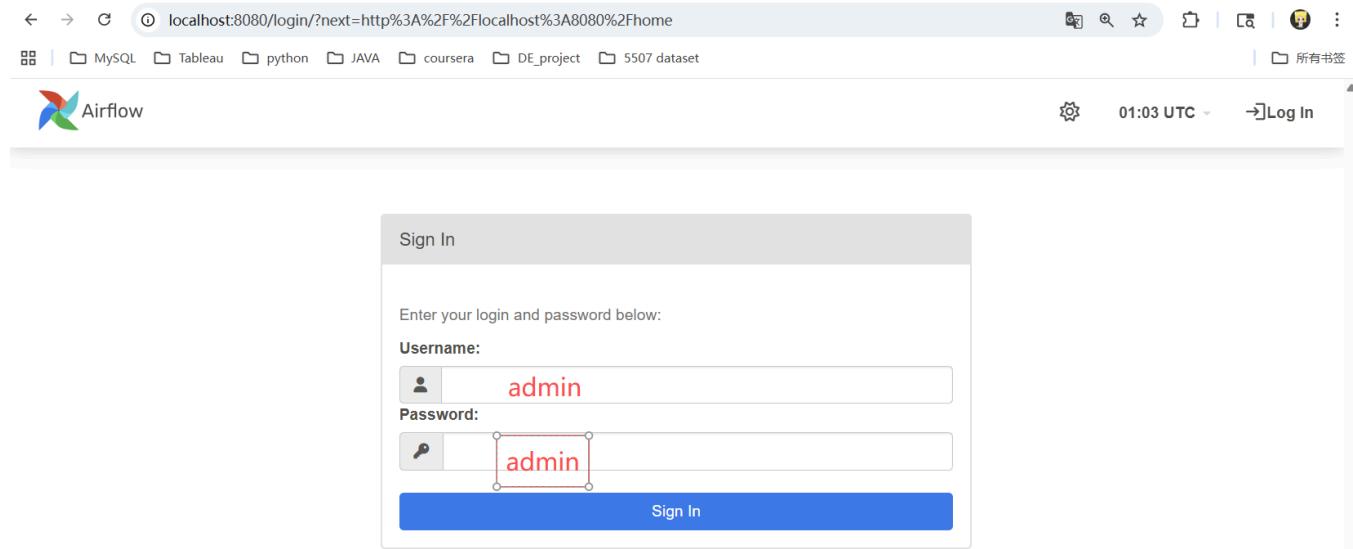
```
docker compose build inference
```

3.Start Services and Check

(1)Use the following command to start all services(containers), and waiting for all container run successfully:

```
docker compose up -d
```

(2) Open <http://localhost:8080/home> in your browser, then enter the username admin and password admin; If the page fails to open, it indicates that the service has failed to run:



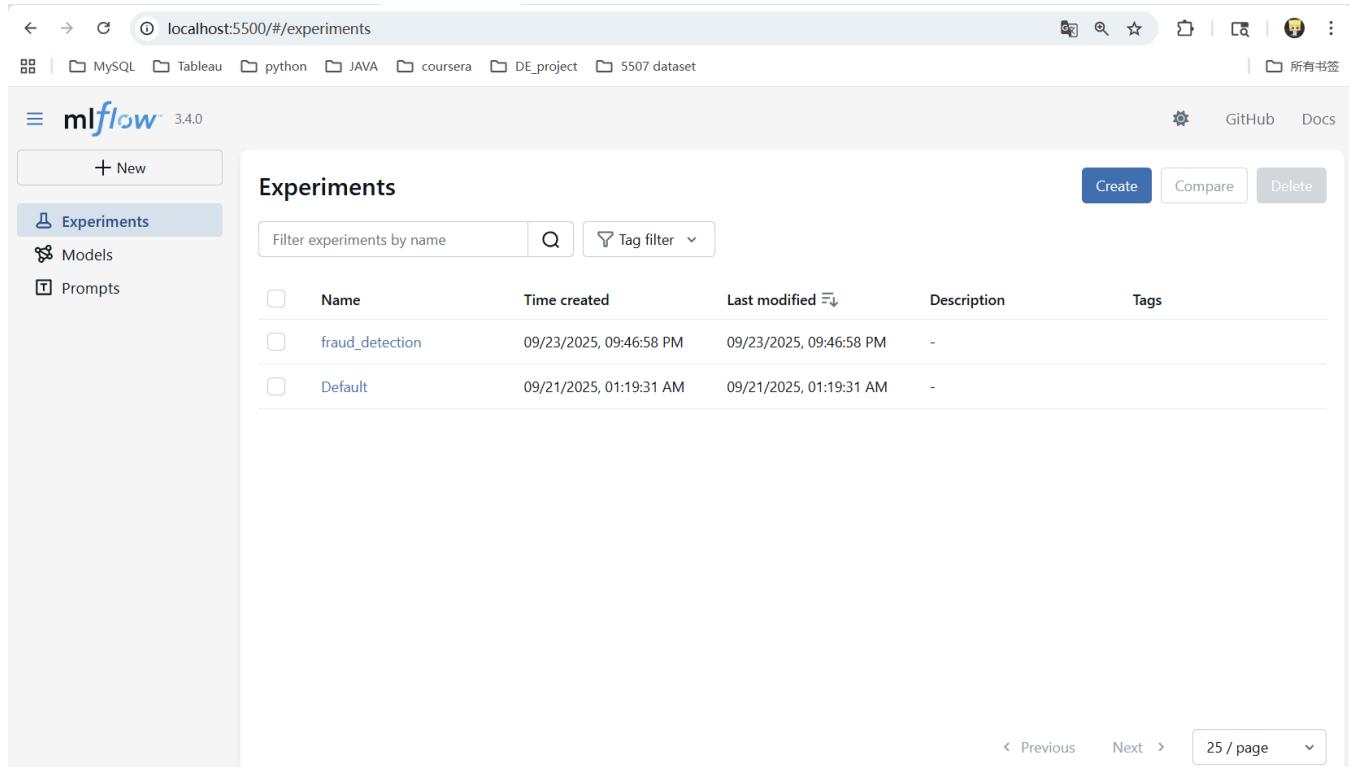
You should be able to see:

The screenshot shows the Airflow home page with the 'DAGs' tab selected. The page displays a list of DAGs with their details. The DAGs listed are: `fraud_detection_catboost_training`, `fraud_detection_decision_tree_training`, `fraud_detection_extra_tree_training`, `fraud_detection_lightgbm_training`, `fraud_detection_logistic_training`, `fraud_detection_mip_training`, and `fraud_detection_random_forest_training`. Each DAG row includes columns for status (Paused or Active), owner (fraud_detection_project), number of runs, schedule, last run date, next run date, and recent tasks. The 'DAGs' tab is highlighted in blue.

(3) Open <http://localhost:5500/> in your browser, If this is the first time opening it, this interface should be blank, and the `fraud_detection` directory does not exist:

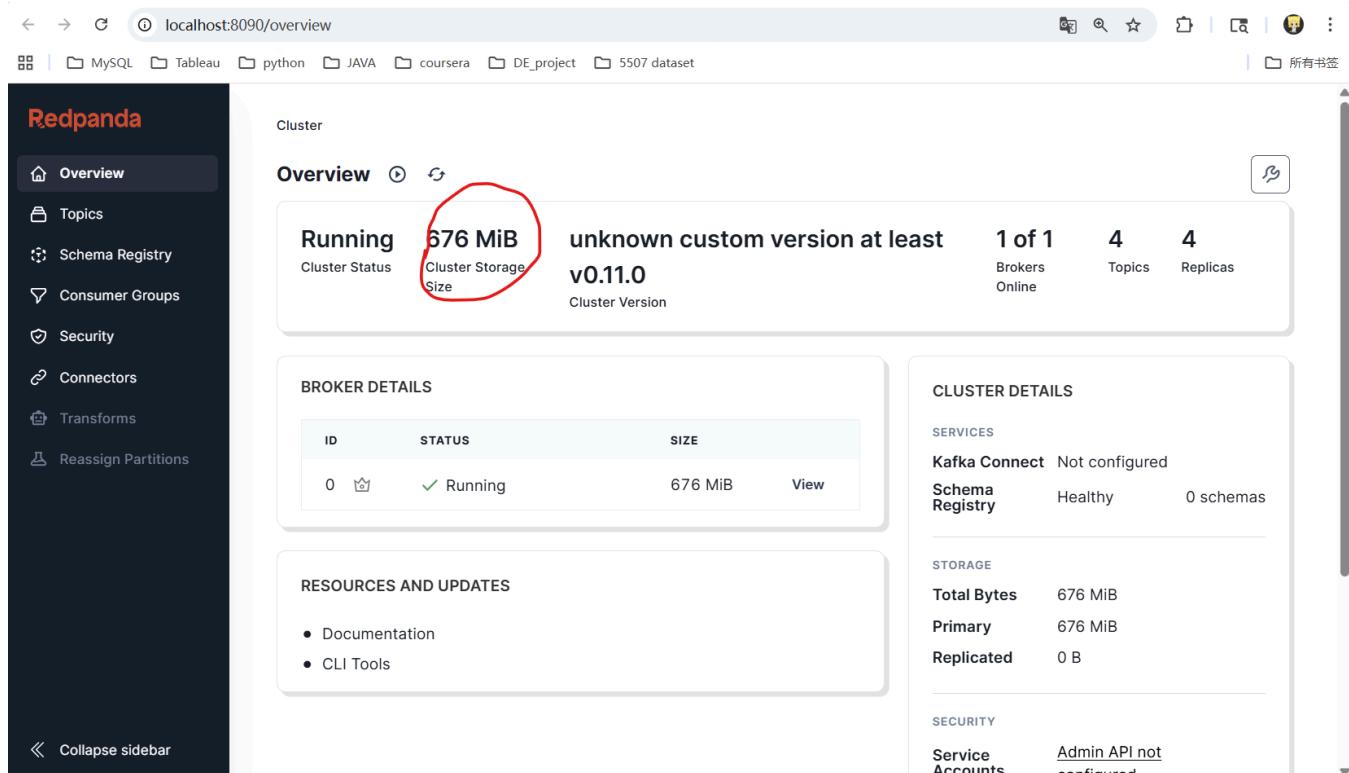
 Note

After running a task that trains a model in Airflow, a `fraud_detection` directory will appear.



The screenshot shows the mlflow UI at <http://localhost:5500/#/experiments>. The left sidebar has tabs for Experiments (selected), Models, and Prompts. The main area is titled "Experiments" with a table. The table has columns: Name, Time created, Last modified, Description, and Tags. It lists two entries: "fraud_detection" (created on 09/23/2025, last modified on 09/23/2025, no description, no tags) and "Default" (created on 09/21/2025, last modified on 09/21/2025, no description, no tags). There are buttons for Create, Compare, and Delete at the top right. At the bottom right, there are navigation links for Previous, Next, and a page size selector set to 25 / page.

(4) Open <http://localhost:8090/overview> in your browser, you may notice the growing volume of data.



The screenshot shows the Redpanda UI at <http://localhost:8090/overview>. The left sidebar has tabs for Overview (selected), Topics, Schema Registry, Consumer Groups, Security, Connectors, Transforms, and Reassign Partitions. The main area is titled "Overview". It shows cluster status: "Running" (Cluster Storage Size: 676 MiB, circled in red), "unknown custom version at least v0.11.0" (Cluster Version), and "1 of 1" brokers online, with 4 topics and 4 replicas. Below this are sections for BROKER DETAILS (Broker ID 0, Status: Running, Size: 676 MiB) and CLUSTER DETAILS (Services: Kafka Connect Not configured, Schema Registry Healthy, 0 schemas). There are also sections for RESOURCES AND UPDATES (Documentation, CLI Tools) and SECURITY (Service Accounts: Admin API not configured).

(5) Open <http://localhost:5555/> in your browser, You should be able to see two workers online.

The screenshot shows the Flower web interface at <http://localhost:5555/>. The page title is "Flower". The main content is a table titled "Workers" showing two workers: "celery@e38e3c996c37" and "celery@7d213b742818". Both workers are listed as "Online". The table includes columns for Worker, Status, Active, Processed, Failed, Succeeded, Retried, and Load Average. The "Load Average" column lists values 0.38, 0.37, 0.37 for each worker. A summary row "Total" shows 0 Active, 3 Processed, 0 Failed, 3 Succeeded, and 0 Retried. Below the table, it says "Showing 1 to 2 of 2 workers". Navigation buttons "Previous" and "Next" are shown, with "1" selected.

Worker	Status	Active	Processed	Failed	Succeeded	Retried	Load Average
celery@e38e3c996c37	Online	0	1	0	1	0	0.38, 0.37, 0.37
celery@7d213b742818	Online	0	2	0	2	0	0.38, 0.37, 0.37
Total		0	3	0	3	0	

(5) Open <http://localhost:9001/> in your browser, Username is minio, password is minio123:

The screenshot shows the MinIO Object Store login page at <http://localhost:9001/login>. The page has a dark background with a blue wavy graphic. The title "High-Performance Object Store" is displayed. To the right, the MinIO logo and "OBJECT STORE" text are visible, along with "AGPLv3 LICENSE". The login form contains fields for "Username" (minio) and "Password" (minio123), both highlighted in red. A "Login" button is below the form. At the bottom, there is a "Documentation | GitHub | Support | Download" link and a "Other Authentication Methods" dropdown.

After logging in, you will see:

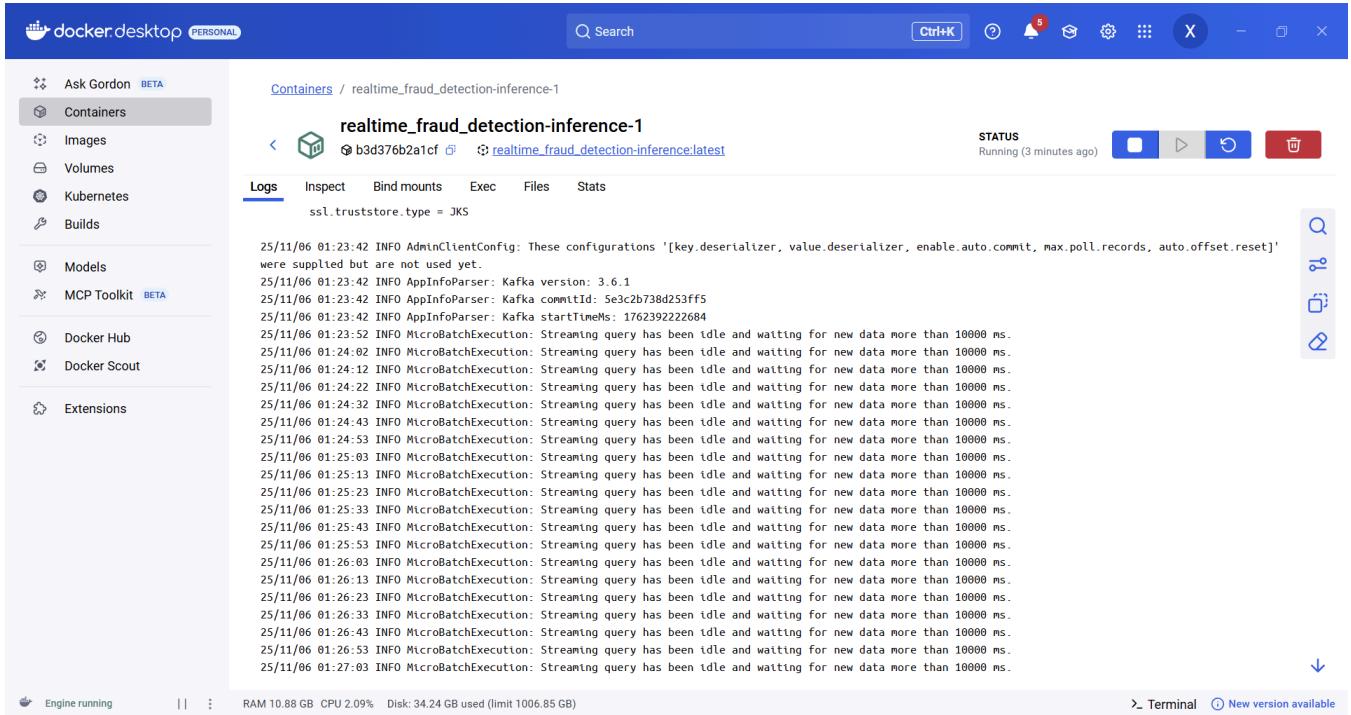
The screenshot shows the MinIO Object Store Object Browser. On the left, there's a sidebar with sections for User (Object Browser, Access Keys, Documentation), Administrator (Buckets, Policies, Identity, Monitoring, Events, Configuration), and a search bar labeled 'Filter Buckets'. The main area is titled 'Object Browser' and contains a table with four columns: Name, Objects, Size, and Access. A single row is visible for the bucket 'mlflow', which has 346 objects and a size of 55.3 MB, with access set to R/W.

(6) In Docker Desktop, click the Containers panel on the left and navigate to the inference container:

The screenshot shows the Docker Desktop interface. The left sidebar has sections for Ask Gordon (BETA), Containers (selected), Images, Volumes, Kubernetes, Builds, Models, MCP Toolkit (BETA), Docker Hub, Docker Scout, and Extensions. The main area is titled 'Containers' and shows CPU and memory usage statistics: Container CPU usage (105.55% / 2200%) and Container memory usage (8.35GB / 11.13GB). It also has a 'Show charts' link. Below these are search and filter options. The main table lists running containers with columns for Name, Container ID, Image, Port(s), and Actions. One container, 'inference-1', is highlighted with a red circle around its name. The table shows 19 items.

Note

Under the logs bar, if you see the log entry “`INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms`,” it indicates that the system’s inference component is ready and waiting for new credit card data to be pushed, utilizing the machine learning model to make inferences!



The screenshot shows the Docker Desktop interface. On the left, there's a sidebar with various tabs: Ask Gordon (BETA), Containers (selected), Images, Volumes, Kubernetes, Builds, Models, MCP Toolkit (BETA), Docker Hub, Docker Scout, and Extensions. The main area shows a container named "realtime_fraud_detection-inference-1" with a status of "Running (3 minutes ago)". Below the container name, there are tabs for Logs (selected), Inspect, Bind mounts, Exec, Files, and Stats. The logs pane displays several lines of log output:

```
25/11/06 01:23:42 INFO AdminClientConfig: These configurations '[key.deserializer, value.deserializer, enable.auto.commit, max.poll.records, auto.offset.reset]' were supplied but are not used yet.  
25/11/06 01:23:42 INFO AppInfoParser: Kafka version: 3.6.1  
25/11/06 01:23:42 INFO AppInfoParser: Kafka commitId: 5e3cb738d253ff5  
25/11/06 01:23:42 INFO AppInfoParser: Kafka startTimeMs: 1762392222684  
25/11/06 01:23:52 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:24:02 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:24:12 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:24:22 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:24:32 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:24:43 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:24:53 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:25:03 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:25:13 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:25:23 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:25:33 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:25:43 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:25:53 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:26:03 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:26:13 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:26:23 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:26:33 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:26:43 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:26:53 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.  
25/11/06 01:27:03 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.
```

At the bottom of the screen, there are status indicators for Engine running, RAM usage (10.88 GB), CPU usage (2.09%), and Disk usage (34.24 GB used / limit 1006.85 GB). There are also links for Terminal and New version available.