

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 requiemment

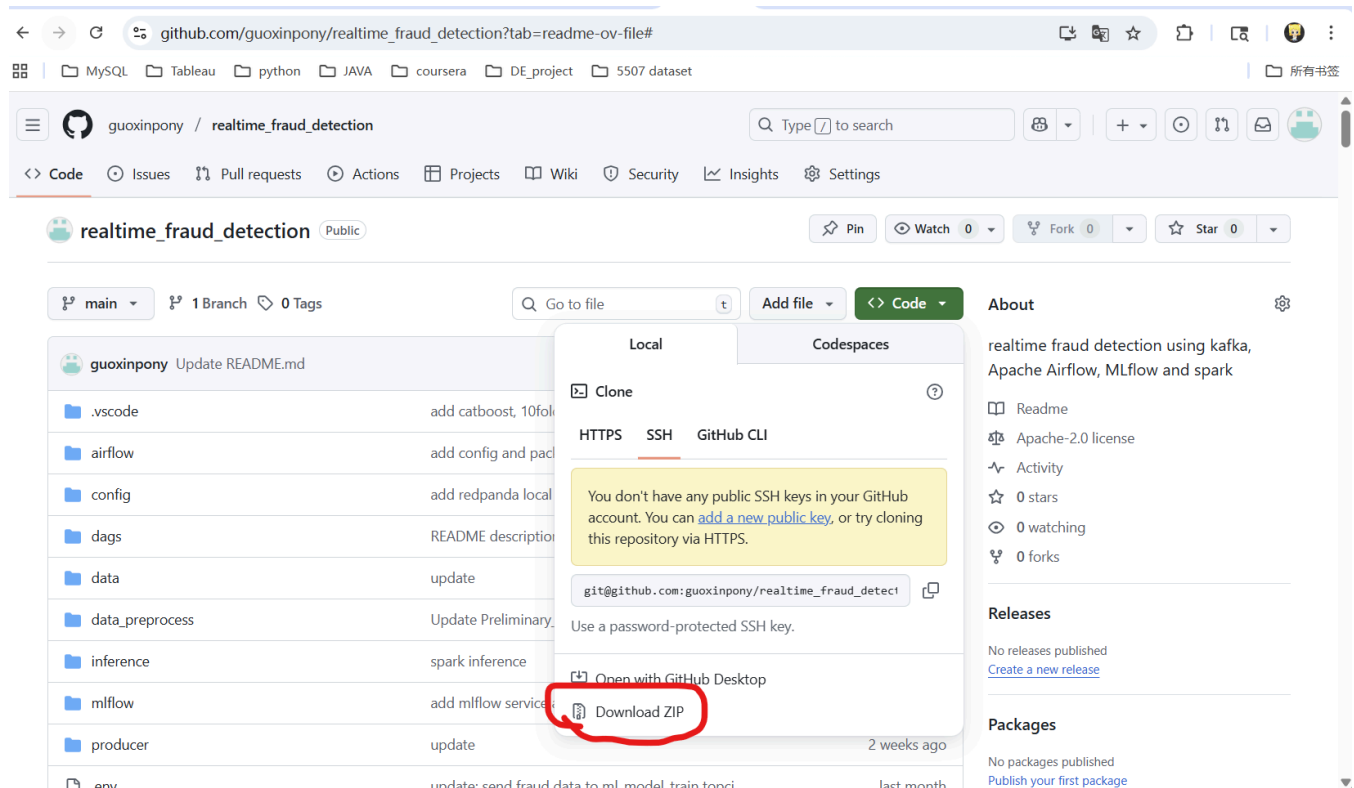
- **CPU:** 4+ cores recommended
- **Memory:** 8GB minimum, 16GB recommended
- **OS:** Linux, macOS, or Windows with WSL2

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
```

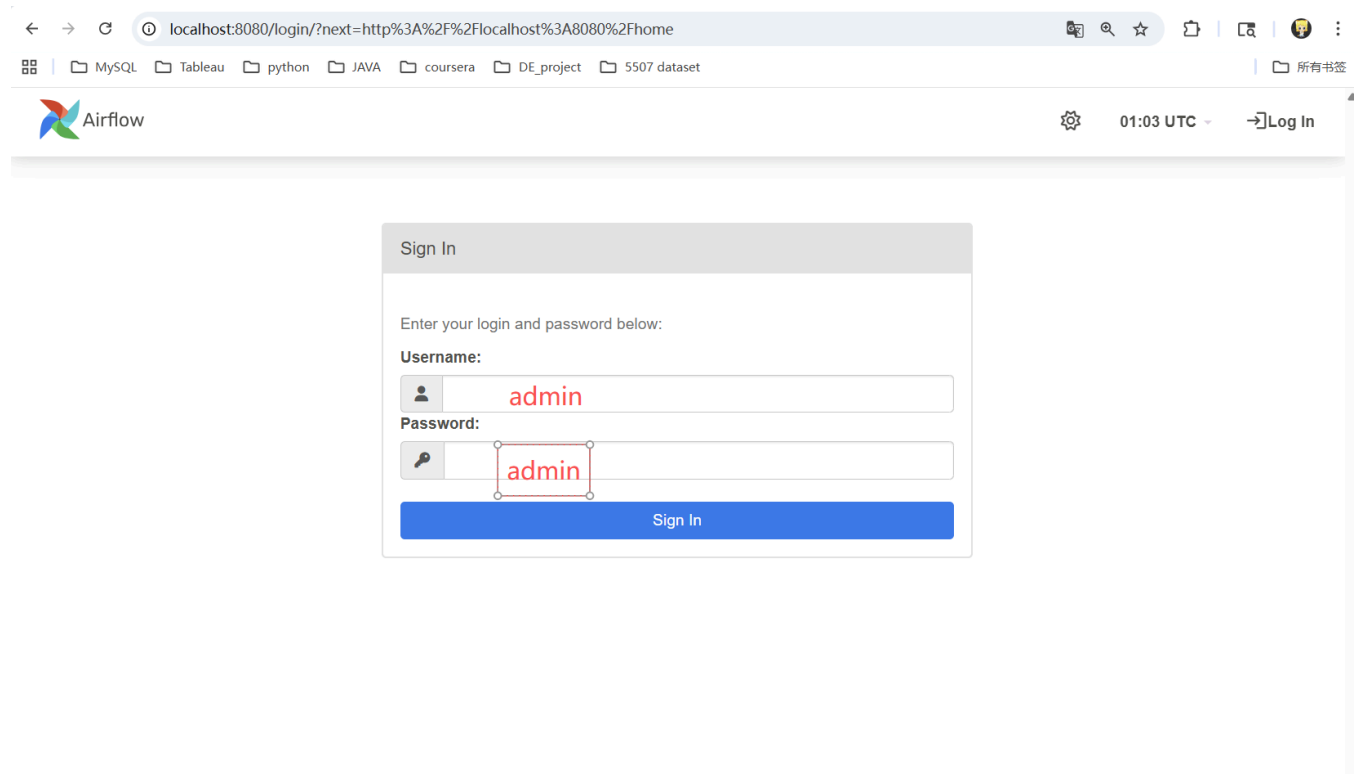
```
docker compose build inference
```

3.Start Services and Check

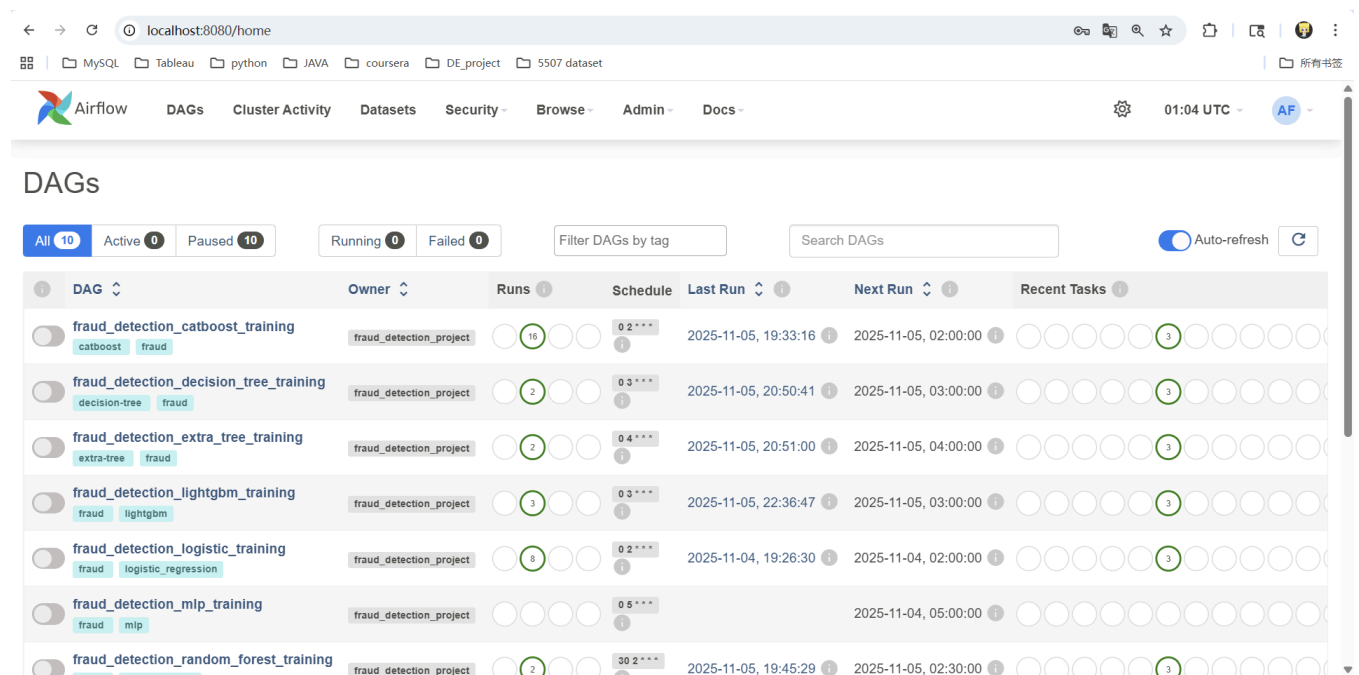
(1)Use the following command to start all services(containers), and waiting for all container run successfully:

```
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:



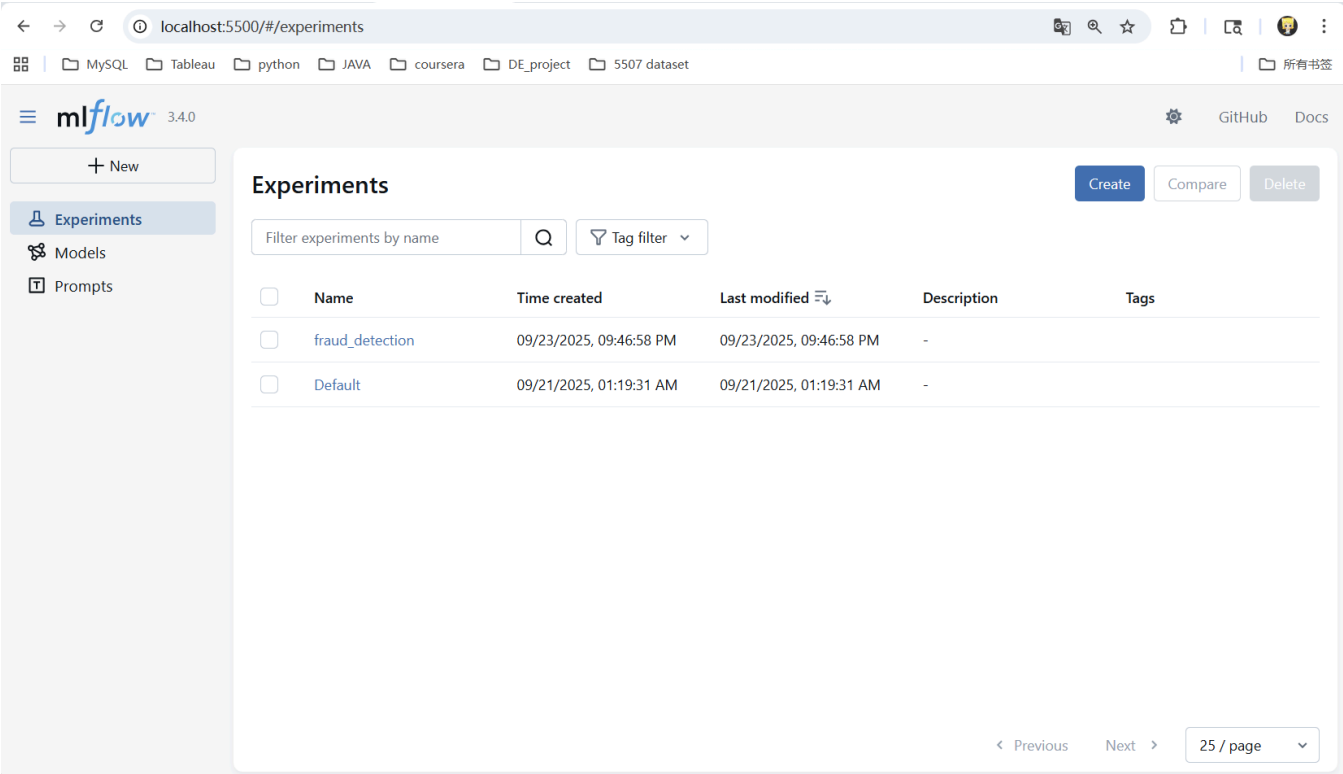
You should be able to see:



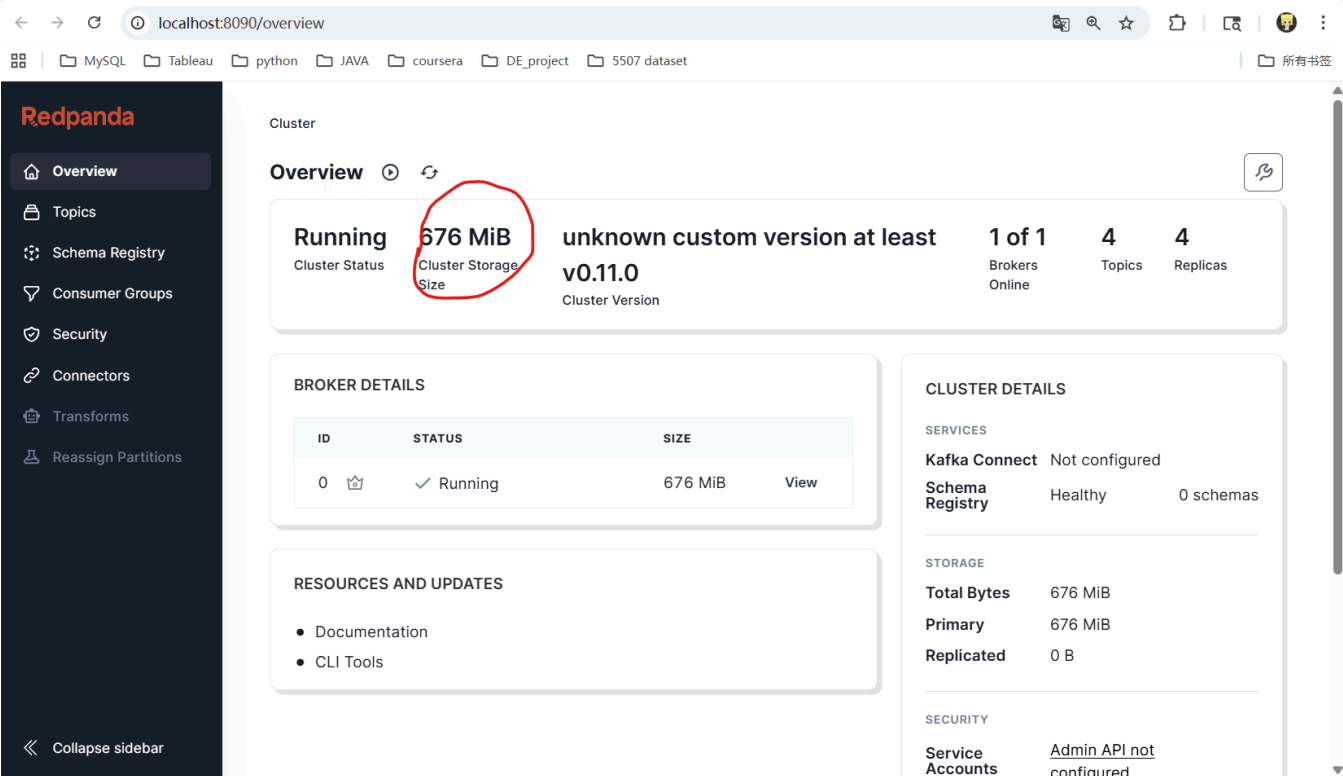
(3) Open <http://localhost:5500/> in your browser, If this is the first time opening it, this interface should be blank, and the `fraud_detection` directory does not exist:

Note

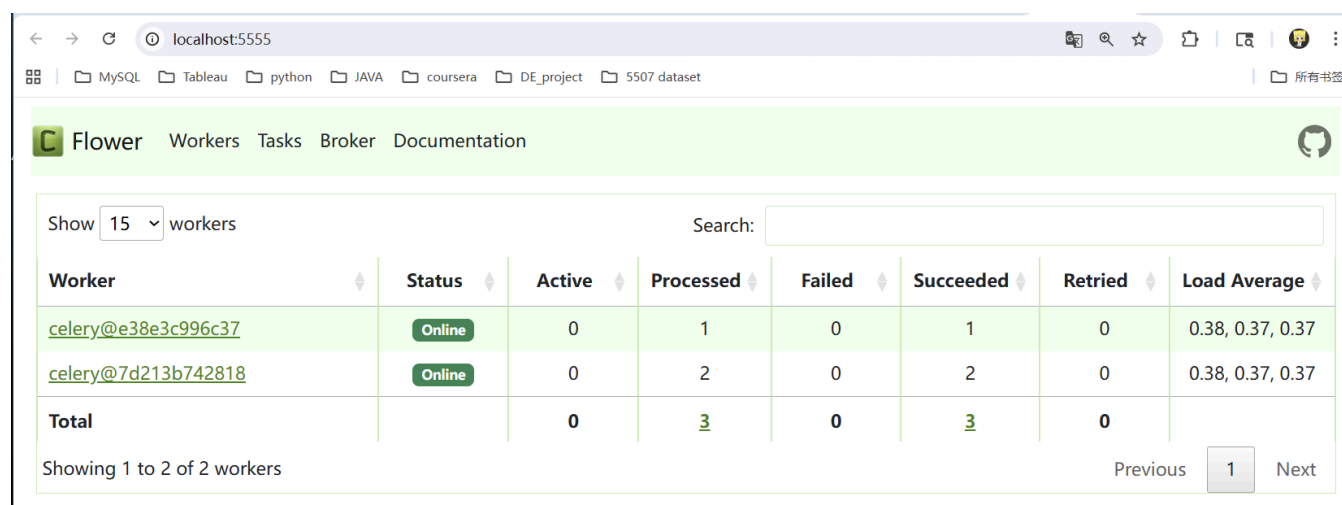
After running a task that trains a model in Airflow, a `fraud_detection` directory will appear.



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



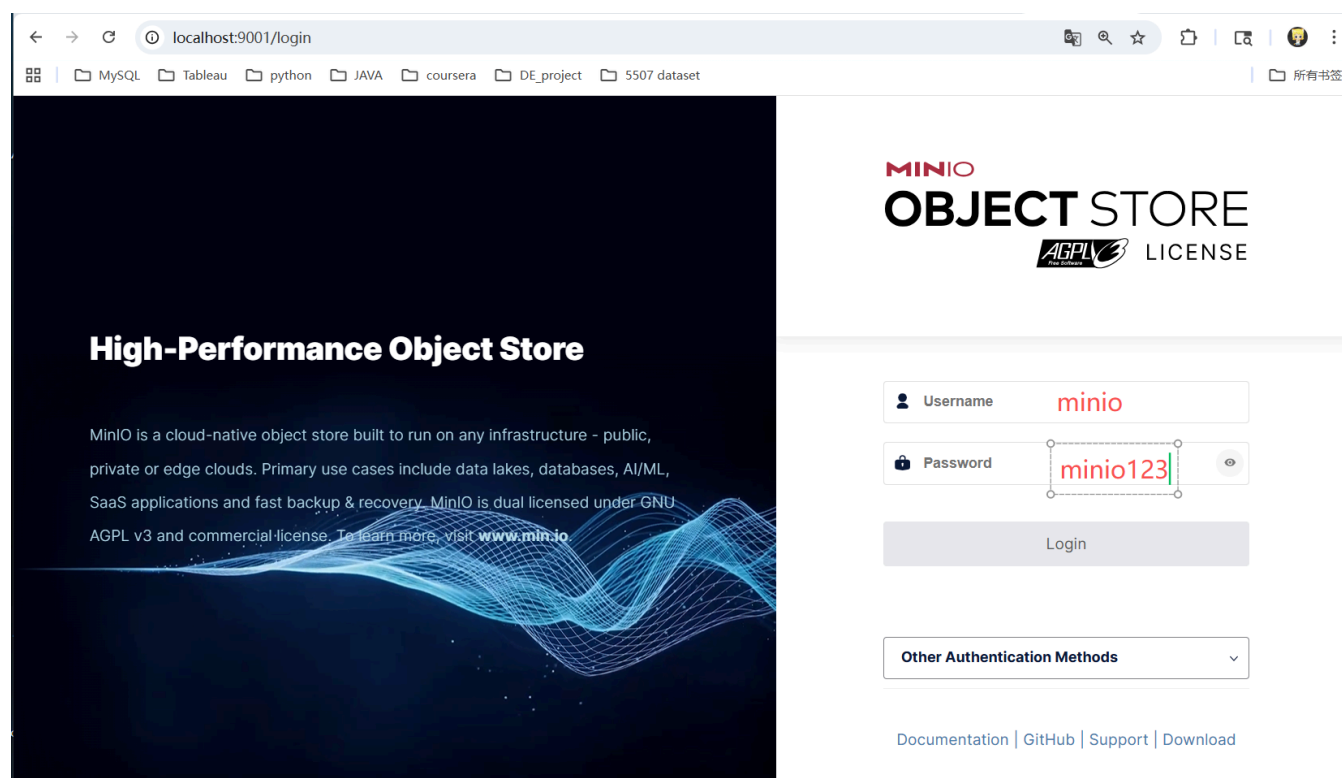
(5) Open <http://localhost:5555/> in your browser, You should be able to see two workers online.



The screenshot shows the Flower web interface in a browser window. The address bar displays 'localhost:5555'. The interface has a green header with the 'Flower' logo and navigation links: 'Workers', 'Tasks', 'Broker', and 'Documentation'. Below the header, there's a section to 'Show 15 workers' with a search bar. A table lists the workers with columns for Worker, Status, Active, Processed, Failed, Succeeded, Retried, and Load Average. Two workers are listed, both with a status of 'Online'. A 'Total' row shows 0 active, 3 processed, 0 failed, 3 succeeded, and 0 retried workers. At the bottom, it says 'Showing 1 to 2 of 2 workers' with 'Previous' and 'Next' buttons.

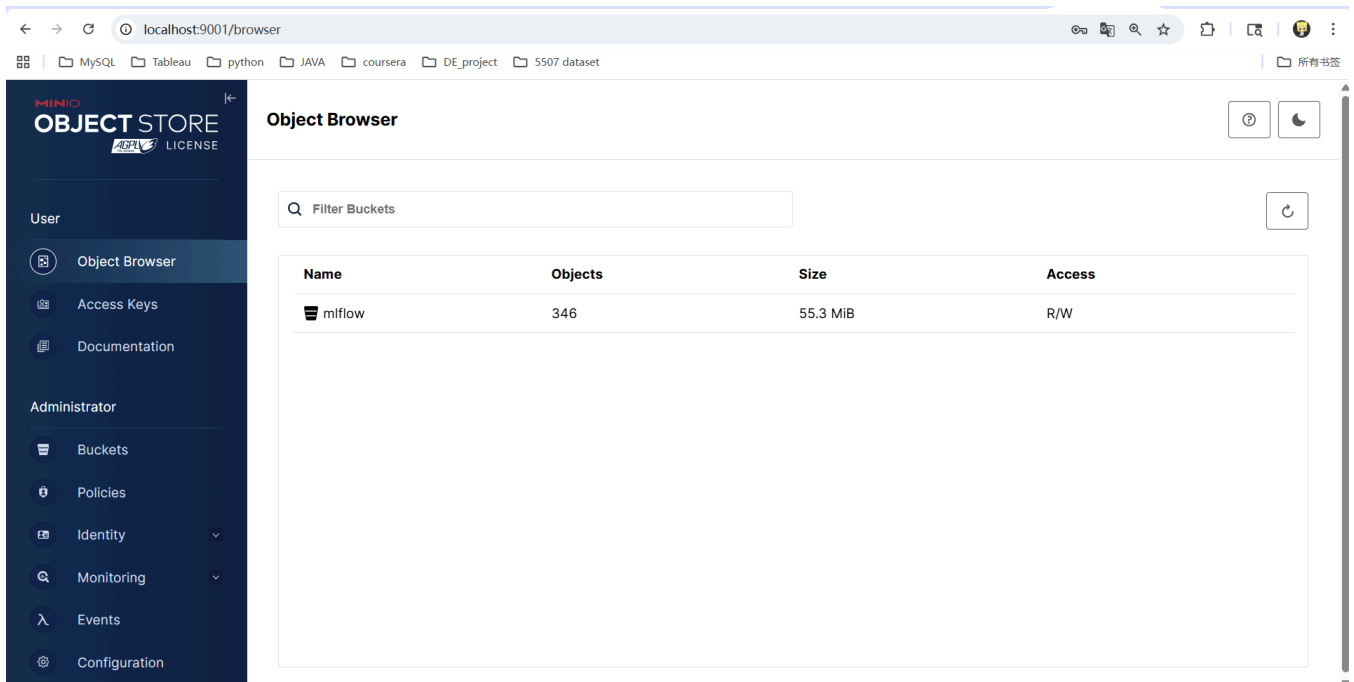
Worker	Status	Active	Processed	Failed	Succeeded	Retried	Load Average
celery@e38e3c996c37	Online	0	1	0	1	0	0.38, 0.37, 0.37
celery@7d213b742818	Online	0	2	0	2	0	0.38, 0.37, 0.37
Total		0	3	0	3	0	

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

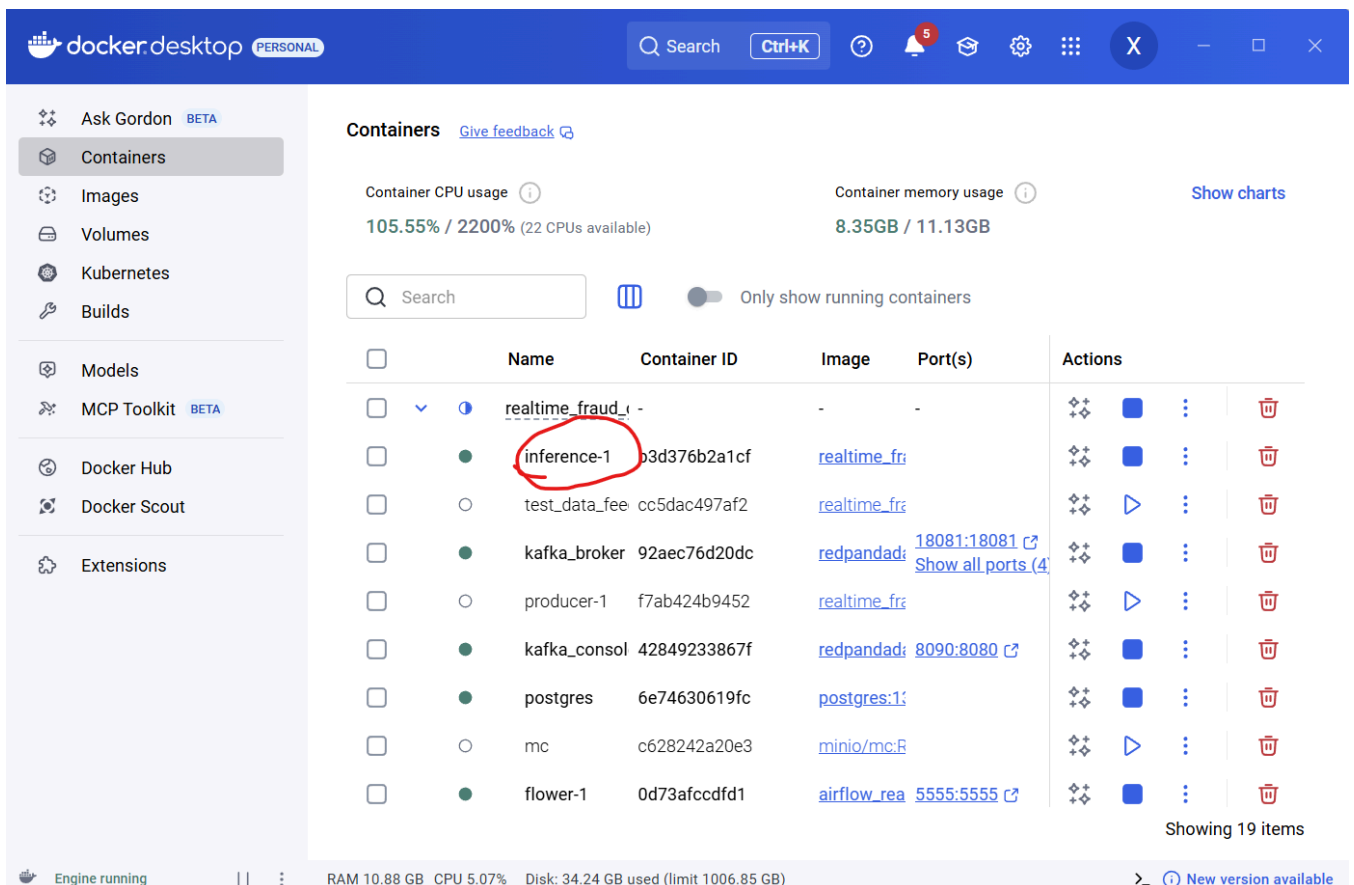


The screenshot shows the MinIO Object Store login page in a browser window. The address bar displays 'localhost:9001/login'. The page has a dark blue header with the 'MINIO OBJECT STORE' logo and 'AGPL LICENSE'. Below the header, there's a section titled 'High-Performance Object Store' with a description of MinIO. On the right, there's a login form with fields for 'Username' (minio) and 'Password' (minio123), a 'Login' button, and a dropdown for 'Other Authentication Methods'. At the bottom, there are links for 'Documentation', 'GitHub', 'Support', and 'Download'.

After logging in, you will see:



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



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 is a sidebar with navigation options: Ask Gordon (BETA), Containers, Images, Volumes, Kubernetes, Builds, Models, MCP Toolkit (BETA), Docker Hub, Docker Scout, and Extensions. The main area displays the 'Containers' view for a container named 'realtime_fraud_detection-inference-1'. The container ID is 'b3d376b2a1cf' and it is running the image 'realtime_fraud_detection-inference:latest'. The status is 'Running (3 minutes ago)'. Below this, there are tabs for 'Logs', 'Inspect', 'Bind mounts', 'Exec', 'Files', and 'Stats'. The 'Logs' tab is selected, showing a list of log entries. The first entry is '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.' followed by '25/11/06 01:23:42 INFO AppInfoParser: Kafka version: 3.6.1'. The subsequent entries are '25/11/06 01:23:42 INFO AppInfoParser: Kafka commitId: 5e3c2b738d253ff5' and '25/11/06 01:23:42 INFO AppInfoParser: Kafka startTimeMs: 176239222684'. The following 18 entries are '25/11/06 01:24:02 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.' The final entry is '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 interface, there is a status bar showing 'Engine running', system resources 'RAM 10.88 GB CPU 2.09% Disk: 34.24 GB used (limit 1006.85 GB)', and buttons for 'Terminal' and 'New version available'.

docker desktop PERSONAL

Search Ctrl+K

Containers / realtime_fraud_detection-inference-1

realtime_fraud_detection-inference-1

b3d376b2a1cf realtime_fraud_detection-inference:latest

STATUS Running (3 minutes ago)

Logs Inspect Bind mounts Exec Files Stats

ssl.truststore.type = JKS

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: 5e3c2b738d253ff5

25/11/06 01:23:42 INFO AppInfoParser: Kafka startTimeMs: 176239222684

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 01:24:02 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 01:24:42 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:24:52 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:25:02 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:25:12 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:25:22 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:25:32 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:25:42 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:25:52 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:26:02 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:26:12 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:26:22 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:26:32 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:26:42 INFO MicroBatchExecution: Streaming query has been idle and waiting for new data more than 10000 ms.

25/11/06 01:26:52 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.

Engine running | | RAM 10.88 GB CPU 2.09% Disk: 34.24 GB used (limit 1006.85 GB) > Terminal New version available