

WEEK 4: Finding out the directory where the messages are getting stored and find out how they are being assimilating over time.

- Kafka messages are stored inside “/tmp/kafka-logs”

How to find the location:

```
cd /tmp/kafka-logs/node_metrics-0
```

Here, **node_metrics** is the topic name & 0 indicates it's first partition.

```
ubuntu@ip-172-31-21-218:~$ cd /tmp/kafka-logs/node_metrics-0
ubuntu@ip-172-31-21-218:/tmp/kafka-logs/node_metrics-0$ ls -a
.                                00000000000000000000.timeindex
..                               000000000000000005376.snapshot
00000000000000000000.index      leader-epoch-checkpoint
00000000000000000000.log        partition.metadata
ubuntu@ip-172-31-21-218:/tmp/kafka-logs/node_metrics-0$
```

What do these files mean?

File Name	Purpose
*.log	The actual data (messages)
*.index	Offset index (maps offsets to physical positions in the log file)
*.timeindex	Timestamp index (maps timestamps to offsets)

Each *.log is a **segment** — Kafka creates new ones as data grows or time/size limits are reached.

Python Script using flask to monitor the messages over time and ship the metrics in a json format to a endpoint:

```
import os
import time
from flask import Flask, jsonify
from threading import Thread

# --- Config ---
LOG_DIR = "/tmp/kafka-logs/node_metrics-0" # Update this path
CHECK_INTERVAL = 600 # 10 minutes

# Store size history
log_size_history = []

def get_directory_size(directory):
    total = 0
    for dirpath, dirnames, filenames in os.walk(directory):
        for f in filenames:
            fp = os.path.join(dirpath, f)
            total += os.path.getsize(fp)
    return total

def monitor_kafka_logs():
    while True:
        size_bytes = get_directory_size(LOG_DIR)
        timestamp = int(time.time())
        log_size_history.append({
            "timestamp": timestamp,
            "size_bytes": size_bytes
        })
```

```

        print(f"[{time.strftime('%Y-%m-%d %H:%M:%S')}] Size:
{size_bytes / 1024:.2f} KB")

        time.sleep(CHECK_INTERVAL)

# --- API Server (for Grafana etc.) ---
app = Flask(__name__)

@app.route("/metrics")
def get_metrics():
    return jsonify(log_size_history)

def run_flask():
    app.run(host="0.0.0.0", port=5000)

# --- Main ---
if __name__ == "__main__":
    Thread(target=monitor_kafka_logs, daemon=True).start()
    run_flask()

```

Using **nohup** to run it as a background service:

```
nohup python3 kafka_log_monitor.py &
```

Expose the metrics at:

<http://<your-EC2-IP>:5000/metrics>

Monitoring the growth of size of kafka logs:

```
[2025-04-05 18:25:27] Size: 20879.26 KB
WARNING: This is a development server. Do not use it in a production deployment
Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.31.21.218:5000
128.185.112.58 - - [05/Apr/2025 18:26:58] "GET / HTTP/1.1" 404 -
128.185.112.58 - - [05/Apr/2025 18:26:59] "GET /favicon.ico HTTP/1.1" 404 -
128.185.112.58 - - [05/Apr/2025 18:27:25] "GET /metrics HTTP/1.1" 200 -
[2025-04-05 18:27:27] Size: 20879.26 KB
128.185.112.58 - - [05/Apr/2025 18:27:47] "GET /metrics HTTP/1.1" 200 -
128.185.112.58 - - [05/Apr/2025 18:28:00] "GET /metrics HTTP/1.1" 200 -
128.185.112.58 - - [05/Apr/2025 18:28:10] "GET /metrics HTTP/1.1" 200 -
[2025-04-05 18:29:27] Size: 20879.26 KB
[2025-04-05 18:31:27] Size: 20879.26 KB
[2025-04-05 18:33:27] Size: 20879.26 KB
128.185.112.58 - - [05/Apr/2025 18:34:55] "GET /metrics HTTP/1.1" 200 -
[2025-04-05 18:35:27] Size: 20879.26 KB
[2025-04-05 18:37:27] Size: 20879.26 KB
[2025-04-05 18:39:27] Size: 20879.26 KB
128.185.112.58 - - [05/Apr/2025 18:39:53] "GET /metrics HTTP/1.1" 200 -
[2025-04-05 18:41:27] Size: 20879.26 KB
[2025-04-05 18:43:27] Size: 20879.26 KB
^Cubuntu@ip-172-31-21-218:~$
```