

Project: Weather Stations Monitoring

Designing Data-Intensive Applications

Name	ID
Toka Ashraf Abo Alwafa	19015539
Rowan Nasser Edress	19015686
Nada Mohamed Ibrahim	19016782

Problem Statement

You are required to simulate a distributed weather monitoring system with multiple components, including weather stations, data processing, storage, and analysis. This system will utilize Kafka for messaging, BitCask Riak for key-value storage, and Parquet files for archiving. Additionally, the system must be deployed using Docker and Kubernetes.

Requirements

A) Weather Station Mock

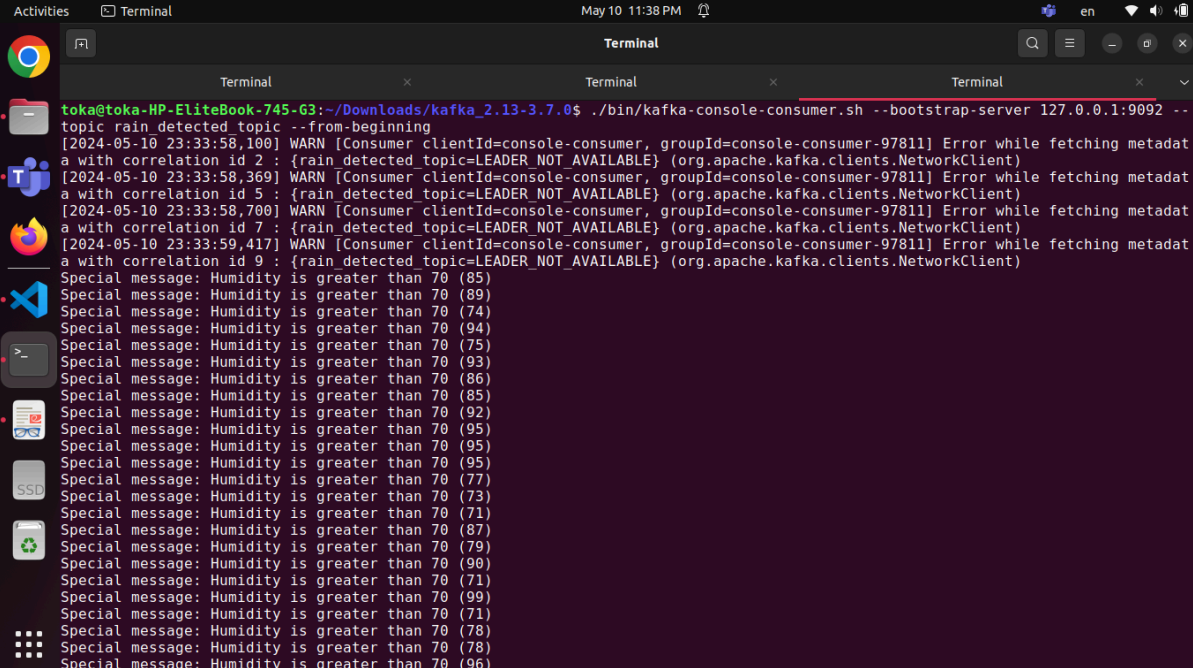
- The weather station ID is retrieved, and a WeatherStation task is scheduled to run every second using a scheduled executor service.
- The WeatherStation class, which implements Runnable, runs periodically and generates a weather status message each time it executes.
- Within each execution:
 - A Unix timestamp is generated.
 - Random values for humidity, temperature, and wind speed are created.
 - The battery status is randomly updated.
 - Occasionally, the message is dropped to simulate message loss.
 - If not dropped, a JSON string is constructed with the weather data.

B) Kafka Integration:

Weather Station Connection to use Kafka's produce API to send status messages to certain topic (weather_topic) in the Kafka server.

C) Rain Detection using Kafka Processors:

- Use Kafka Processors to detect if humidity > 70% by consuming from the topic that contains all weather statuses and extract the humidity to see if it more than 70 print special message in another topic (rain_detected_topic).

A terminal window titled 'Terminal' showing the output of a Kafka console consumer. The user is running the command: `./bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic rain --from-beginning`. The output shows several warning messages: `WARN [Consumer clientId=console-consumer, groupId=console-consumer-97811] Error while fetching metadata for a with correlation id 2: {rain_detected_topic=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)`. This is followed by a series of 'Special message: Humidity is greater than 70 (85)' through '(96)'.

```
toka@toka-HP-EliteBook-745-G3:~/Downloads/kafka_2.13-3.7.0$ ./bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --
topic rain detected topic --from-beginning
[2024-05-10 23:33:58,100] WARN [Consumer clientId=console-consumer, groupId=console-consumer-97811] Error while fetching metadat
a with correlation id 2 : {rain_detected_topic=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
[2024-05-10 23:33:58,369] WARN [Consumer clientId=console-consumer, groupId=console-consumer-97811] Error while fetching metadat
a with correlation id 5 : {rain_detected_topic=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
[2024-05-10 23:33:58,700] WARN [Consumer clientId=console-consumer, groupId=console-consumer-97811] Error while fetching metadat
a with correlation id 7 : {rain_detected_topic=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
[2024-05-10 23:33:59,417] WARN [Consumer clientId=console-consumer, groupId=console-consumer-97811] Error while fetching metadat
a with correlation id 9 : {rain_detected_topic=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
Special message: Humidity is greater than 70 (85)
Special message: Humidity is greater than 70 (89)
Special message: Humidity is greater than 70 (74)
Special message: Humidity is greater than 70 (94)
Special message: Humidity is greater than 70 (75)
Special message: Humidity is greater than 70 (93)
Special message: Humidity is greater than 70 (86)
Special message: Humidity is greater than 70 (85)
Special message: Humidity is greater than 70 (92)
Special message: Humidity is greater than 70 (95)
Special message: Humidity is greater than 70 (95)
Special message: Humidity is greater than 70 (95)
Special message: Humidity is greater than 70 (77)
Special message: Humidity is greater than 70 (73)
Special message: Humidity is greater than 70 (71)
Special message: Humidity is greater than 70 (87)
Special message: Humidity is greater than 70 (79)
Special message: Humidity is greater than 70 (90)
Special message: Humidity is greater than 70 (71)
Special message: Humidity is greater than 70 (99)
Special message: Humidity is greater than 70 (71)
Special message: Humidity is greater than 70 (78)
Special message: Humidity is greater than 70 (78)
Special message: Humidity is greater than 70 (96)
```

D) Central Station Implementation:

1. BitCask Riak for Status Storage:
 - Maintain an updated key-value store of each station's latest status using BitCask Riak LSM. and implementing hint files to help in rehash for recovery also schedule the compaction process to help to reclaim space by removing old versions of keys and consolidating data, ensuring that the storage remains efficient and does not degrade over time.
 - Bitcask can offer the following benefits in our project:
 1. The stations can generate a lot of data in real-time. Bitcask's high write performance ensures that all this data can be logged efficiently without slowing down the system.
 2. Quick Data Access as when querying the status of a particular station, the in-memory index allows for very fast data retrieval, making the system responsive.
 3. In case of system crashes, the hint files help in quickly recovering the state, ensuring that the system can get back online with minimal data loss.
 4. Scalability as the number of stations increases, the system can handle the increased load without a significant drop in performance, thanks to the scalable nature of Bitcask.

5. Efficient Storage Management: as regular compaction ensures that the storage remains efficient and does not bloat with outdated data, keeping the system performance optimal.

2. Historical Data Archiving:

- Archive all weather statuses that are stored in kafka into Parquet files.
- Partition Parquet files by time and station ID by making a folder with the date and this contains the ten stations each one contains parquet files partitioned by the time.
- Write records in batches to avoid i/o blocking by making the operation of writing not start until the size of the batch is reached.

E) Historical Weather Status Analysis:

1. ElasticSearch Integration:
 - Direct all weather statuses to ElasticSearch for indexing.
 - Make Query to get the percentage of the dropped messages by dividing the total messages that arrived to elastic search by the total messages without dropping that calculated using the distinct values of s_no and multiply them by the number of stations.
2. Kibana is used to visualize and confirm the percentage of the battery status and dropped messages.

F) Deployment using Kubernetes:

1. Docker Configuration:

<input type="checkbox"/>	weather-station c555b8477450	latest	In use	5 minutes ago	572.61 MB			
<input type="checkbox"/>	es-indexer 56ab2b6e8e8e	latest	Unused	1 hour ago	291.16 MB			
<input type="checkbox"/>	central-station bc68ae8056d2	latest	In use	15 hours ago	572.53 MB			
<input type="checkbox"/>	bitnami/zookeeper 16c83222214f	latest	In use	4 days ago	576.34 MB			
<input type="checkbox"/>	bitnami/kafka bc7700d07f0b	latest	In use	4 days ago	622.91 MB			
<input type="checkbox"/>	docker/desktop-kubernetes 15340d8e9882	kubernetes-v1.29.2-cni-v1.4.0-critools-v	Unused	2 months ago	438.87 MB			
registry.k8s.io/kube-apiserver								

- Write Dockerfile for the central server.

The Dockerfile sets up an OpenJDK environment, configures it to use a specific security policy, and then runs a Java application within that environment.

- Write Dockerfile for the weather stations.

The Dockerfile sets up an OpenJDK environment, copies the application JAR




































file to the appropriate directory, sets the working directory, and runs the Java application.

- Write Dockerfile for the Indexer

This Dockerfile sets up a Docker image with Python installed, installs the necessary dependencies, copies the Elasticsearch script into the container, and specifies the command to run the script when the container starts.

2. Kubernetes Setup:

- Deploy 10 weather station services using their Docker image.

<input type="checkbox"/>		k8s_weather-station-7_weather- 256d22508ceb 	sha256:4c8d9 Running	73.04%	12 seconds ago			
<input type="checkbox"/>		k8s_weather-station-5_weather- fa090ccaec8 	sha256:4c8d9 Running	69.48%	13 seconds ago			
<input type="checkbox"/>		k8s_weather-station-9_weather- 1e32fd1b45bc 	sha256:4c8d9 Running	54.46%	11 seconds ago			
<input type="checkbox"/>		k8s_weather-station-8_weather- 3bcee315fb54 	sha256:4c8d9 Running	53.6%	12 seconds ago			
<input type="checkbox"/>		k8s_kafka kafka-0_default_382 144586ed2428 	sha256:bc770 Running	52.67%	30 seconds ago			
<input type="checkbox"/>		k8s_weather-station-10_weather 359eaae98369 	sha256:4c8d9 Running	50.05%	11 seconds ago			
<input type="checkbox"/>		k8s_weather-station-3_weather- 62b45678a42b 	sha256:4c8d9 Running	42.73%	15 seconds ago			

This YAML file defines 10 Deployments and corresponding Services for 10 instances of the weather station application. Each instance has its own unique identifier and is exposed through a separate service.

- Deploy a central server service using its Docker image.

```
89, "temperature": 48, "wind_speed": 13}}
2024-05-18 10:19:10 Processing message: {"station_id": 2, "s_no": 9, "battery_status": "medium", "status_timestamp": 1716016750, "weather": {"humidity": 89, "temperature": 48, "wind_speed": 13}}
2024-05-18 10:19:10 Processed message for station ID: 2
2024-05-18 10:19:10 Fetched 1 records
2024-05-18 10:19:10 Processing record: {"station_id": 4, "s_no": 10, "battery_status": "low", "status_timestamp": 1716016750, "weather": {"humidity": 55, "temperature": 2, "wind_speed": 4}}
2024-05-18 10:19:10 Processing message: {"station_id": 4, "s_no": 10, "battery_status": "low", "status_timestamp": 1716016750, "weather": {"humidity": 55, "temperature": 2, "wind_speed": 4}}
2024-05-18 10:19:10 Processed message for station ID: 4
2024-05-18 10:19:10 Fetched 1 records
2024-05-18 10:19:10 Processing record: {"station_id": 7, "s_no": 11, "battery_status": "low", "status_timestamp": 1716016750, "weather": {"humidity": 31, "temperature": 65, "wind_speed": 23}}
2024-05-18 10:19:10 Processing message: {"station_id": 7, "s_no": 11, "battery_status": "low", "status_timestamp": 1716016750, "weather": {"humidity": 31, "temperature": 65, "wind_speed": 23}}
2024-05-18 10:19:10 Processed message for station ID: 7
2024-05-18 10:19:10 Fetched 1 records
2024-05-18 10:19:10 Processing record: {"station_id": 3, "s_no": 11, "battery_status": "high", "status_timestamp": 1716016750, "weather": {"humidity": 31, "temperature": -7, "wind_speed": 30}}
2024-05-18 10:19:10 Processing message: {"station_id": 3, "s_no": 11, "battery_status": "high", "status_timestamp": 1716016750, "weather": {"humidity": 31, "temperature": -7, "wind_speed": 30}}
2024-05-18 10:19:10 Processed message for station ID: 3
2024-05-18 10:19:11 Fetched 1 records
2024-05-18 10:19:11 Processing record: {"station_id": 9, "s_no": 10, "battery_status": "medium", "status_timestamp": 1716016751, "weather": {"humidity": 51, "temperature": 26, "wind_speed": 26}}
2024-05-18 10:19:11 Processing message: {"station_id": 9, "s_no": 10, "battery_status": "medium", "status_timestamp": 1716016751, "weather": {"humidity": 51, "temperature": 26, "wind_speed": 26}}
2024-05-18 10:19:11 Processed message for station ID: 9
```

✓ home	MODIFIED		2 minutes ago	drwxr-xr-x
✓ data	MOUNT		1 minute ago	drwxrwxrwx
✓ bitcask	MOUNT		19 seconds ago	drwxr-xr-x
0.bitcask	MOUNT	9.8 kB	1 minute ago	-rw-r--r--
0.bitcask.hint	MOUNT	861 Bytes	1 minute ago	-rw-r--r--
1_compact.bitcask	MOUNT	9.8 kB	46 seconds ago	-rw-r--r--
1_compact.bitcask.hint	MOUNT	941 Bytes	46 seconds ago	-rw-r--r--
10_compact.bitcask	MOUNT	7.5 kB	1 second ago	-rw-r--r--
2.bitcask	MOUNT	9.8 kB	45 seconds ago	-rw-r--r--
2.bitcask.hint	MOUNT	861 Bytes	45 seconds ago	-rw-r--r--
3.bitcask	MOUNT	9.8 kB	44 seconds ago	-rw-r--r--
3.bitcask.hint	MOUNT	861 Bytes	44 seconds ago	-rw-r--r--
4_compact.bitcask	MOUNT	9.8 kB	39 seconds ago	-rw-r--r--
parquet	MOUNT		1 minute ago	drwxr-xr-x
2024-MAY-18	MOUNT		1 minute ago	drwxr-xr-x
Station1	MOUNT		36 seconds ago	drwxr-xr-x
.7-18.parquet.crc	MOUNT	24 Bytes	1 minute ago	-rw-r--r--
.7-19.parquet.crc	MOUNT	32 Bytes	36 seconds ago	-rw-r--r--
.7-20.parquet.crc	MOUNT	0 Bytes	36 seconds ago	-rw-r--r--
7-18.parquet	MOUNT	1.8 kB	1 minute ago	-rw-r--r--
7-19.parquet	MOUNT	2.7 kB	36 seconds ago	-rw-r--r--
7-20.parquet	MOUNT	0 Bytes	36 seconds ago	-rw-r--r--
Station10	MOUNT		35 seconds ago	drwxr-xr-x
Station2	MOUNT		36 seconds ago	drwxr-xr-x
Station3	MOUNT		35 seconds ago	drwxr-xr-x
Station4	MOUNT		36 seconds ago	drwxr-xr-x

This YAML file ensures that the "central-station" component runs as a single replica, with access to Kafka for messaging and persistent storage through shared volumes. The service exposes port 8080 to facilitate communication with other components within the Kubernetes cluster.

- Deploy Kafka and Zookeeper as two services.

```

2024-05-18 10:21:08 [2024-05-18 07:21:08,445] INFO [GroupCoordinator 0]: Dynamic member with unknown member id joins group rain-detection-app in PreparingRebalance state. Created a new member id rain-detection-app-83fd607a-5de9-4baf-9830-5b3f320a2e22-StreamThread-1-consumer-1e170bac-4d49-474c-875e-46bcb
b674ae7 and request the member to rejoin with this id. (kafka.coordinator.group.GroupCoordinator)
2024-05-18 10:21:08 [2024-05-18 07:21:08,525] INFO [GroupCoordinator 0]: Dynamic member with unknown member id joins group rain-detection-app in PreparingRebalance state. Created a new member id rain-detection-app-7ac11ab8-c340-435a-b42e-8b16df624d82-StreamThread-1-consumer-355c1727-b161-4c33-ab61-bd1db
51946f6 and request the member to rejoin with this id. (kafka.coordinator.group.GroupCoordinator)
2024-05-18 10:21:08 [2024-05-18 07:21:08,839] INFO [GroupCoordinator 0]: Dynamic member with unknown member id joins group rain-detection-app in PreparingRebalance state. Created a new member id rain-detection-app-c53ee3d5-10cd-44a5-9c41-63fb19cfaf25-StreamThread-1-consumer-df97bf73-321f-4df3-9cd69-89d32
b90975b and request the member to rejoin with this id. (kafka.coordinator.group.GroupCoordinator)
2024-05-18 10:21:08 [2024-05-18 07:21:08,913] INFO [GroupCoordinator 0]: Dynamic member with unknown member id joins group rain-detection-app in PreparingRebalance state. Created a new member id rain-detection-app-a7268dd3-b1d0-45a1-8cfd-a13f81efde2a-StreamThread-1-consumer-bd3ad841-e881-4140-b1ba-a7968
1a4b983 and request the member to rejoin with this id. (kafka.coordinator.group.GroupCoordinator)
2024-05-18 10:21:08 [2024-05-18 07:21:08,927] INFO [GroupCoordinator 0]: Dynamic member with unknown member id joins group rain-detection-app in PreparingRebalance state. Created a new member id rain-detection-app-d331bd33-c785-41d7-8255-7a27326a5eb7-StreamThread-1-consumer-030cbecb-d7cf-4c19-841d-f3ddd
54e69ea and request the member to rejoin with this id. (kafka.coordinator.group.GroupCoordinator)
2024-05-18 10:21:09 [2024-05-18 07:21:09,034] INFO [GroupCoordinator 0]: Dynamic member with unknown member id joins group rain-detection-app in PreparingRebalance state. Created a new member id rain-detection-app-d15d606d-46a9-4601-b0ae-91cc77798ea7-StreamThread-1-consumer-b39413d4-97b8-4e34-ad5d-41968
ea9c05e and request the member to rejoin with this id. (kafka.coordinator.group.GroupCoordinator)
2024-05-18 10:21:09 [2024-05-18 07:21:09,362] INFO [GroupCoordinator 0]: Dynamic member with unknown member id joins group rain-detection-app in PreparingRebalance state. Created a new member id rain-detection-app-8058d89f-3199-4db0-a941-33bd68086a4a-StreamThread-1-consumer-0357380d-2e3f-4d21-bc75-63914
a0877ca and request the member to rejoin with this id. (kafka.coordinator.group.GroupCoordinator)
2024-05-18 10:21:09 [2024-05-18 07:21:09,438] INFO [GroupCoordinator 0]: Dynamic member with unknown member id joins group rain-detection-app in PreparingRebalance state. Created a new member id rain-detection-app-49a69ac3-c478-4423-aa2e-9f4942793faf-StreamThread-1-consumer-eb79c1b1-dc4a-400b-aa9a-7c378
a5fe42f and request the member to rejoin with this id. (kafka.coordinator.group.GroupCoordinator)

```

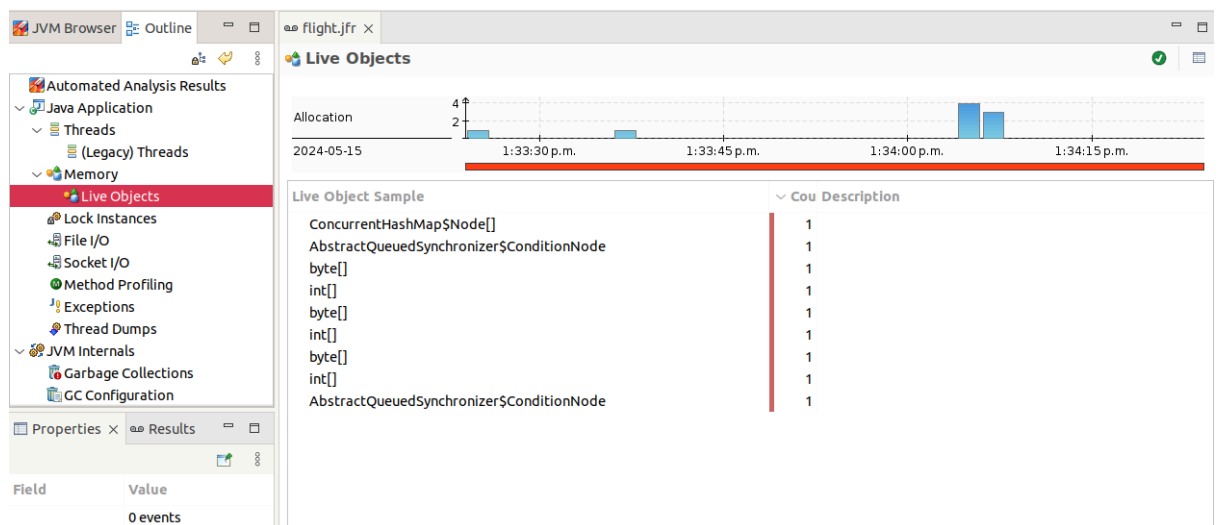
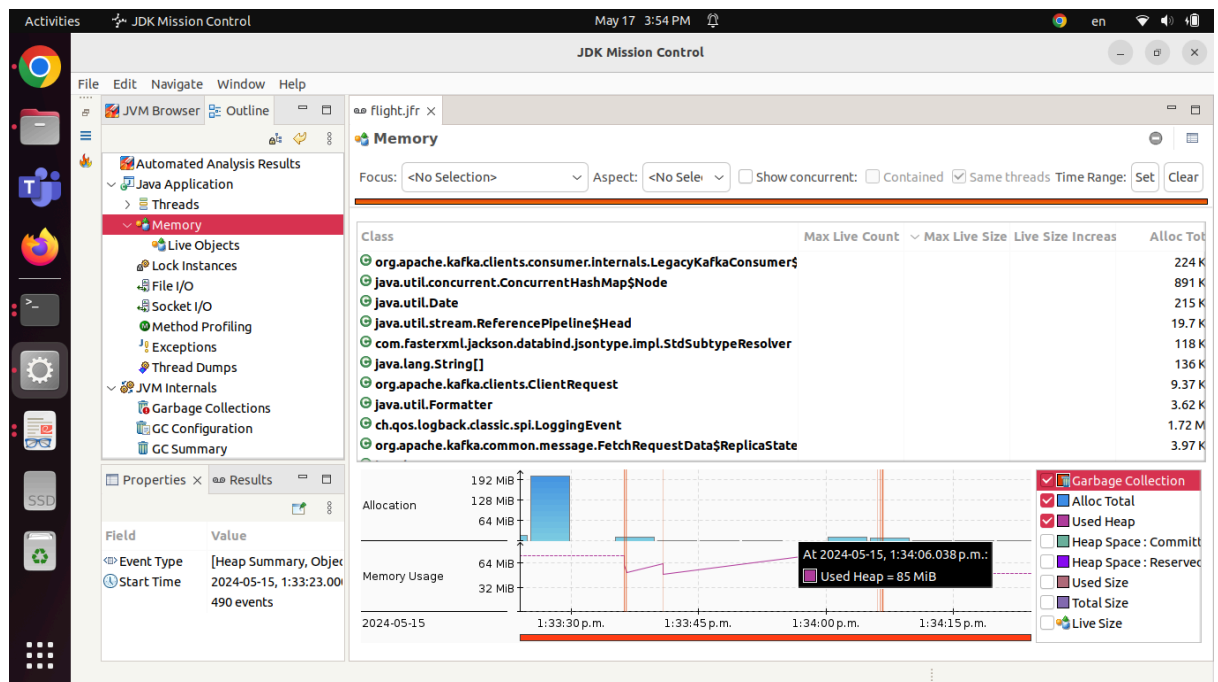
This YAML file sets up a Zookeeper service and StatefulSet, a Kafka service

and StatefulSet, along with a shared volume for data storage, and a ConfigMap for Kafka topics.

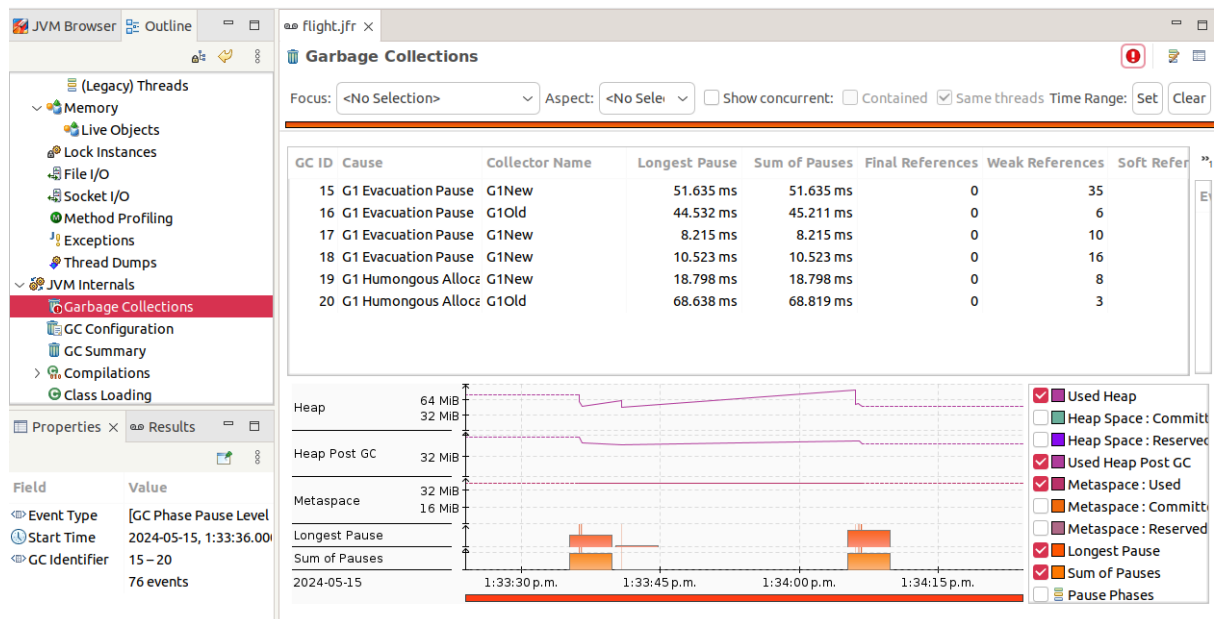
- Configure shared storage for Parquet files and BitCask Riak LSM.
This YAML file creates a PV named "shared-volume" with a capacity of 10 gigabytes, accessible by multiple nodes with read-write access. It also creates a corresponding PVC with the same name and access mode, allowing pods to dynamically claim and use the storage provided by the PV.

G) Profile Central Station using JFR

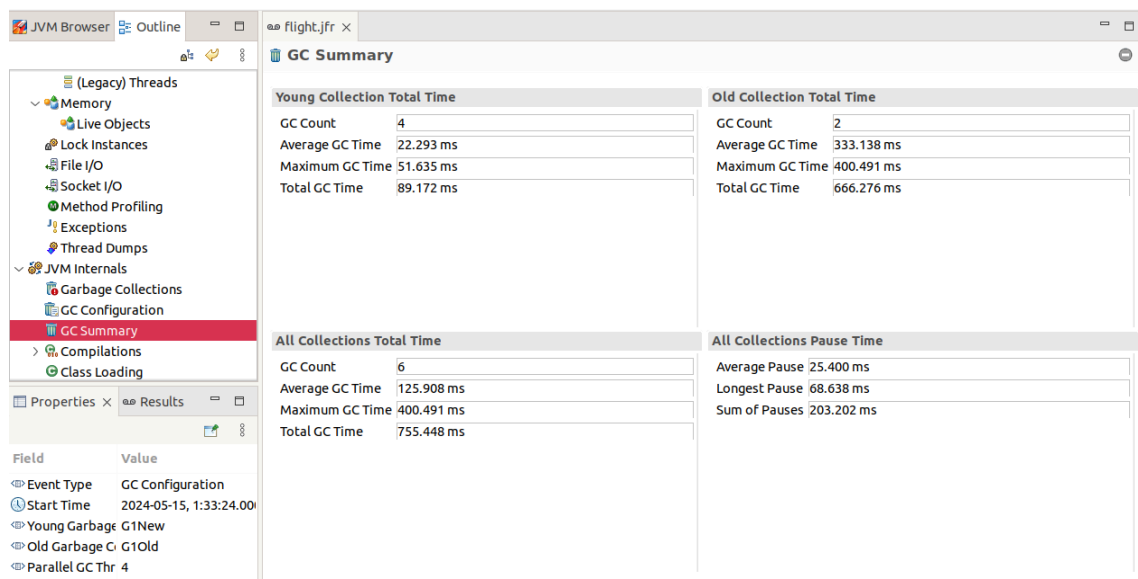
1. Top 10 Classes with highest total memory



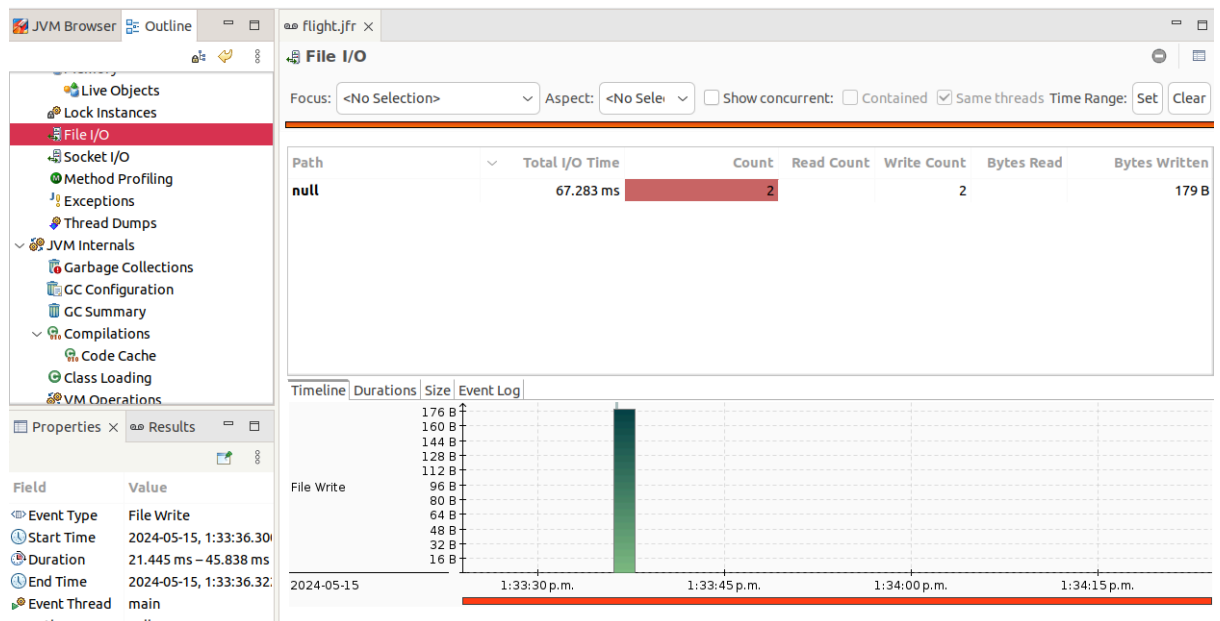
2. GC pauses count



3. GC maximum pause duration



4. List of I/O operations



Results :

Docker & Kubernetes files:

- K8s yaml file: [Link](#)
- Weather Station Dockerfile

```
# Use an OpenJDK base image
FROM openjdk:23-slim

# Copy necessary files
COPY Weather_Stations_Monitoring.jar /app/Weather_Station.jar

# Set the working directory
WORKDIR /app

# Command to run the application
CMD ["java", "-jar", "Weather_Station.jar"]
```

- Central Server Dockerfile

```
# Use an official OpenJDK runtime as a parent image
FROM openjdk:23-slim

# Set the working directory
WORKDIR /app

# Copy the application JAR file and any other necessary files
COPY Weather_Stations_Monitoring.jar /app/Weather_Stations_Monitoring.jar

# Copy the policy file to the container
COPY java.policy /app/java.policy

# Set the Java options to use the policy file
ENV JAVA_OPTS="-Djava.security.manager -Djava.security.policy=/app/java.policy"

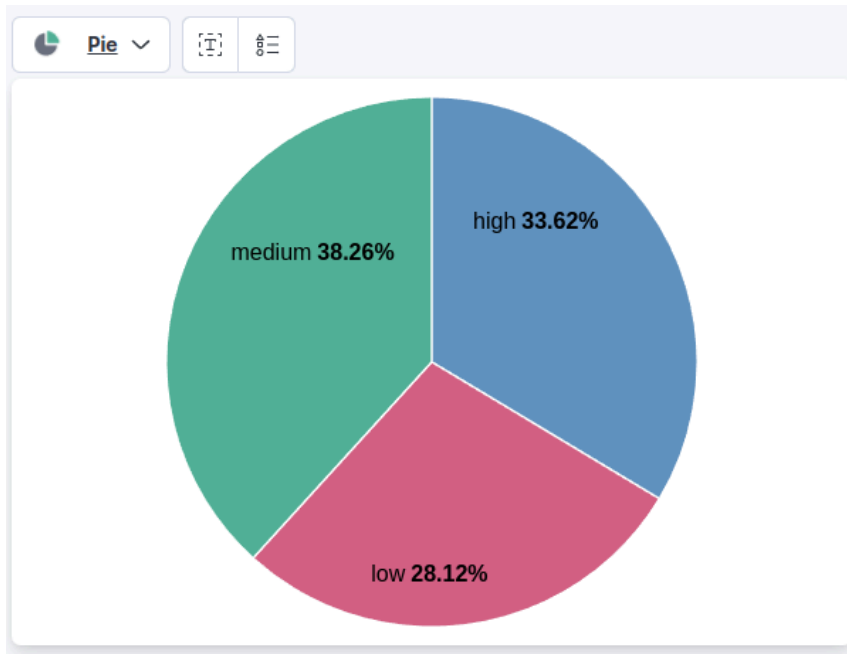
# Run the application
CMD ["sh", "-c", "java $JAVA_OPTS -jar Weather_Stations_Monitoring.jar"]
```

- Indexer Dockerfile

```
1  # Dockerfile for Elasticsearch Script
2  FROM python:3.8-slim
3
4  WORKDIR /app
5
6  # Install required Python packages
7  COPY requirements.txt requirements.txt
8  RUN pip install --no-cache-dir -r requirements.txt
9
10 # Copy the script into the container
11 COPY elastic_search.py elastic_search.py
12
13 # Set the entrypoint for the container
14 CMD ["python", "elastic_search.py"]
15
```

- Screenshots for Kibana visualization confirming:

- Battery status distribution (30% low - 40% medium - 30% high)



- 10% dropped messages

Documents (90)		Field statistics	
Type	Name	Documents (%)	Distinct values
✓	# s_no	90 (100%)	10
DOCUMENTS STATS		SUMMARY	TOP VALUES
count	90	min	1
percentage	100%	median	5
distinct values	10	max	10
		Calculated from 90 records.	
		1	10 (11.1%)
		2	10 (11.1%)
		3	10 (11.1%)
		4	10 (11.1%)
		5	10 (11.1%)
		6	10 (11.1%)
		7	10 (11.1%)
		8	10 (11.1%)
		9	9 (10%)
		10	1 (1.1%)

- [Sample Parquet File Link](#)

- [Sample BitCask Riak LSM directory](#)

Bonus: Open Meteo

- It is designed to fetch specific hourly weather data from an API for a given date and station, handling necessary date formatting and JSON parsing to retrieve and return the desired information of the weather that represents humidity, temperature and wind speed.
- Channel Adapter is designed to take the json object that resulted from the api to open meteo and extract the desired data to make an object of the weather information to use in the weather station mock.

<



k8s_weather-station-7_weather-station-7-6b7d6986f7-xk9zs_default_3d20dc23-64ed-4264-af9d-21eb41063fc2_0

[sha256:c555b84f7450be7f929efe3b533628a131a0e27c59a61641567b0fee662dcc8b507efe78ae86](#)

STATUS
Running (19 seconds ago)



LogsInspectBind mountsExecFilesStats

```
2024-05-18 10:29:06 https://api.open-meteo.com/v1/forecast?latitude=9&longitude=13.41&start_date=2024-05-18&end_date=2024-05-18&hourly=temperature_2m,relativehumidity_2m,wind_speed_10m
2024-05-18 10:29:07 ===== Message is Sent =====
2024-05-18 10:29:08 https://api.open-meteo.com/v1/forecast?latitude=9&longitude=13.41&start_date=2024-05-18&end_date=2024-05-18&hourly=temperature_2m,relativehumidity_2m,wind_speed_10m
2024-05-18 10:29:08 ===== Message is Sent =====
2024-05-18 10:29:09 https://api.open-meteo.com/v1/forecast?latitude=9&longitude=13.41&start_date=2024-05-18&end_date=2024-05-18&hourly=temperature_2m,relativehumidity_2m,wind_speed_10m
2024-05-18 10:29:10 ===== Message is Sent =====
2024-05-18 10:29:11 https://api.open-meteo.com/v1/forecast?latitude=9&longitude=13.41&start_date=2024-05-18&end_date=2024-05-18&hourly=temperature_2m,relativehumidity_2m,wind_speed_10m
2024-05-18 10:29:13 https://api.open-meteo.com/v1/forecast?latitude=9&longitude=13.41&start_date=2024-05-18&end_date=2024-05-18&hourly=temperature_2m,relativehumidity_2m,wind_speed_10m
2024-05-18 10:29:13 ===== Message is Sent =====
2024-05-18 10:29:14 https://api.open-meteo.com/v1/forecast?latitude=9&longitude=13.41&start_date=2024-05-18&end_date=2024-05-18&hourly=temperature_2m,relativehumidity_2m,wind_speed_10m
2024-05-18 10:29:15 ===== Message is Sent =====
2024-05-18 10:29:16 https://api.open-meteo.com/v1/forecast?latitude=9&longitude=13.41&start_date=2024-05-18&end_date=2024-05-18&hourly=temperature_2m,relativehumidity_2m,wind_speed_10m
2024-05-18 10:29:17 ===== Message is Sent =====
2024-05-18 10:29:18 https://api.open-meteo.com/v1/forecast?latitude=9&longitude=13.41&start_date=2024-05-18&end_date=2024-05-18&hourly=temperature_2m,relativehumidity_2m,wind_speed_10m
2024-05-18 10:29:19 ===== Message is Sent =====
2024-05-18 10:29:20 https://api.open-meteo.com/v1/forecast?latitude=9&longitude=13.41&start_date=2024-05-18&end_date=2024-05-18&hourly=temperature_2m,relativehumidity_2m,wind_speed_10m
2024-05-18 10:29:21 https://api.open-meteo.com/v1/forecast?latitude=9&longitude=13.41&start_date=2024-05-18&end_date=2024-05-18&hourly=temperature_2m,relativehumidity_2m,wind_speed_10m
```

```
2024-05-18 10:29:47 Processing message: {"station_id": 9, "s_no": 28, "battery_status": "low", "status_timestamp": 1716017387, "weather": {"humidity": 30, "temperature": 28, "wind_speed": 7}}
2024-05-18 10:29:47 Processed message for station ID: 9
2024-05-18 10:29:47 Fetched 1 records
2024-05-18 10:29:47 Processing record: {"station_id": 2, "s_no": 29, "battery_status": "medium", "status_timestamp": 1716017387, "weather": {"humidity": 88, "temperature": 24, "wind_speed": 8}}
2024-05-18 10:29:47 Processing message: {"station_id": 2, "s_no": 29, "battery_status": "medium", "status_timestamp": 1716017387, "weather": {"humidity": 88, "temperature": 24, "wind_speed": 8}}
2024-05-18 10:29:47 Processed message for station ID: 2
2024-05-18 10:29:47 Fetched 1 records
2024-05-18 10:29:47 Processing record: {"station_id": 6, "s_no": 30, "battery_status": "medium", "status_timestamp": 1716017387, "weather": {"humidity": 75, "temperature": 28, "wind_speed": 7}}
2024-05-18 10:29:47 Processing message: {"station_id": 6, "s_no": 30, "battery_status": "medium", "status_timestamp": 1716017387, "weather": {"humidity": 75, "temperature": 28, "wind_speed": 7}}
2024-05-18 10:29:47 Processed message for station ID: 6
2024-05-18 10:29:47 Fetched 1 records
2024-05-18 10:29:47 Processing record: {"station_id": 4, "s_no": 29, "battery_status": "low", "status_timestamp": 1716017387, "weather": {"humidity": 86, "temperature": 24, "wind_speed": 9}}
2024-05-18 10:29:47 Processing message: {"station_id": 4, "s_no": 29, "battery_status": "low", "status_timestamp": 1716017387, "weather": {"humidity": 86, "temperature": 24, "wind_speed": 9}}
2024-05-18 10:29:47 Processed message for station ID: 4
2024-05-18 10:29:47 Fetched 1 records
2024-05-18 10:29:47 Processing record: {"station_id": 7, "s_no": 27, "battery_status": "low", "status_timestamp": 1716017387, "weather": {"humidity": 74, "temperature": 30, "wind_speed": 13}}
2024-05-18 10:29:47 Processing message: {"station_id": 7, "s_no": 27, "battery_status": "low", "status_timestamp": 1716017387, "weather": {"humidity": 74, "temperature": 30, "wind_speed": 13}}
2024-05-18 10:29:47 Processed message for station ID: 7
```