Configuration Manual

MSc Research Project

MSc in Cloud Computing

Geethanjali Gudduri

Student ID: 23327626

School of Computing

National College of Ireland

Supervisor: Shreyas Setlur Arun

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name	Geethanjali
Student ID	23327626
Programme	MSc Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Shreyas Setlur Arun
Submission Due Date:	11-08-2025
Project Title:	Enhancing IoT Data-Stream Processing with a Lean Serverless Cloud Architecture
Word Count:	1146
Page Count:	12

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ÀLL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action

Signature: Geethanjali Gudduri

Date: 08-08-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies) **Attach a Moodle submission receipt of the online project** **submission,** to each project (including multiple copies).	□ ✓ □ ✓ ✓
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on the computer.	□ ✓

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Geethanjali Gudduri

Student ID: 23327626

1. Introduction

This manual enumerates all hardware, software, AWS-account and data-access prerequisites required to reproduce the end-to-end serverless pipeline evaluated in the dissertation. Following the steps below will let a reader:

- Deploy in one command a CDK stack containing the Kinesis → Lambda → S3/Step Functions pipeline.
- Upload the Kaggle Air Quality in India dataset to S3.
- Re-generate the bursty workload (1 × → 50 × baseline) from Cloud9.
- Observe latency, throughput and cost metrics in CloudWatch, X-Ray and Cost Explorer.
- Tear everything down so that no billable resources remain.

Section 2 lists system requirements, Section 3 covers AWS and local setup, Section 4 explains dataset acquisition, Section 5 walks through stack deployment, Section 6 shows how to run the workload and capture metrics, Section 7 details clean-up and reproducibility notes.

2. System Requirements

Component	Minimum	Notes
Local laptop	Any 64-bit OS capable of SSH and a modern browser	Used only to launch Cloud9 and browse the AWS console
AWS account	Root billing configured, IAM administrator user, Cost Explorer enabled	Create a Budget alarm at €20
Region	eu-west-1 (Ireland)	Keep all resources in a single Region/AZ for latency consistency
Cloud9 EC2 environment	t3.medium (2 vCPU, 4 GiB RAM, 10 GB EBS)	Hosts the workload-driver script

3. Software Stack

Layer	Version tested	Install / notes

OS in Cloud9	Amazon Linux 2023	Pre-installed
AWS CLI	2.16 or newer	sudo yum -y install awscli
AWS CDK	2.139.0	`npm i -g aws-cdk` (Node ≥18 auto-installed in Cloud9)
Python	3.12	Already present; pyenv not required
boto3	1.34	pip install --user boto3
JQ (JSON utils)	1.6	for inspecting CDK stack outputs
Git	pre-installed	clone the project repo

4. Dataset Acquisition

- Create a Kaggle account and accept the licence for Air Quality in India.
- Download the five CSV files (station_hour.csv plus four dimension tables).
- From the workstation, upload them to the raw S3 bucket after the CDK stack is deployed (path raw/).

5. Infrastructure Deployment (CDK)

5.1. Clone repo

```
git clone https://github.com/<your-fork>/serverless-iot-pipeline.git
cd serverless-iot-pipeline/cdk
```

5.2. Bootstrap the target account/region (only once per account)

```
cdk bootstrap aws://$(aws sts get-caller-identity --query Account --output text)/eu-west-1
```

5.3. Context variables

```
export IOT_USER_TAG=$(whoami)
```

5.4. Deploy

```
cdk deploy --require-approval never
```

Deployment completes in ~4 minutes and prints Stack Outputs:

Key	Example value

RawBucketName	nci-iot-raw-cloud9_user
CleanBucketName	nci-iot-cleaned-cloud9_user
KinesisStreamName	air-quality-stream
LambdaName	air-quality-transform

After CDK finishes deploying, CloudFormation's Resources tab provides a one-screen inventory of everything the stack created. The screenshot in Fig. 1 confirms that the two S3 buckets, Kinesis stream and pre-warmed Lambda alias were all provisioned successfully

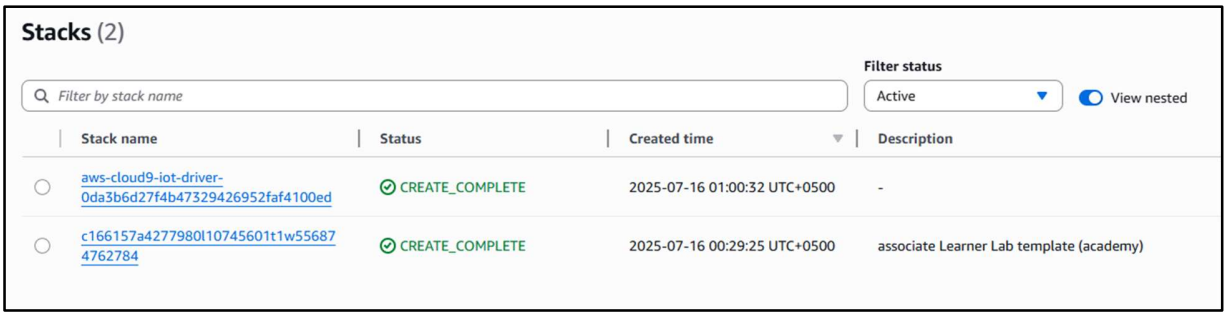


Figure 1. CDK-deployed serverless resources: buckets, stream, Lambda function and pre-warmed alias.

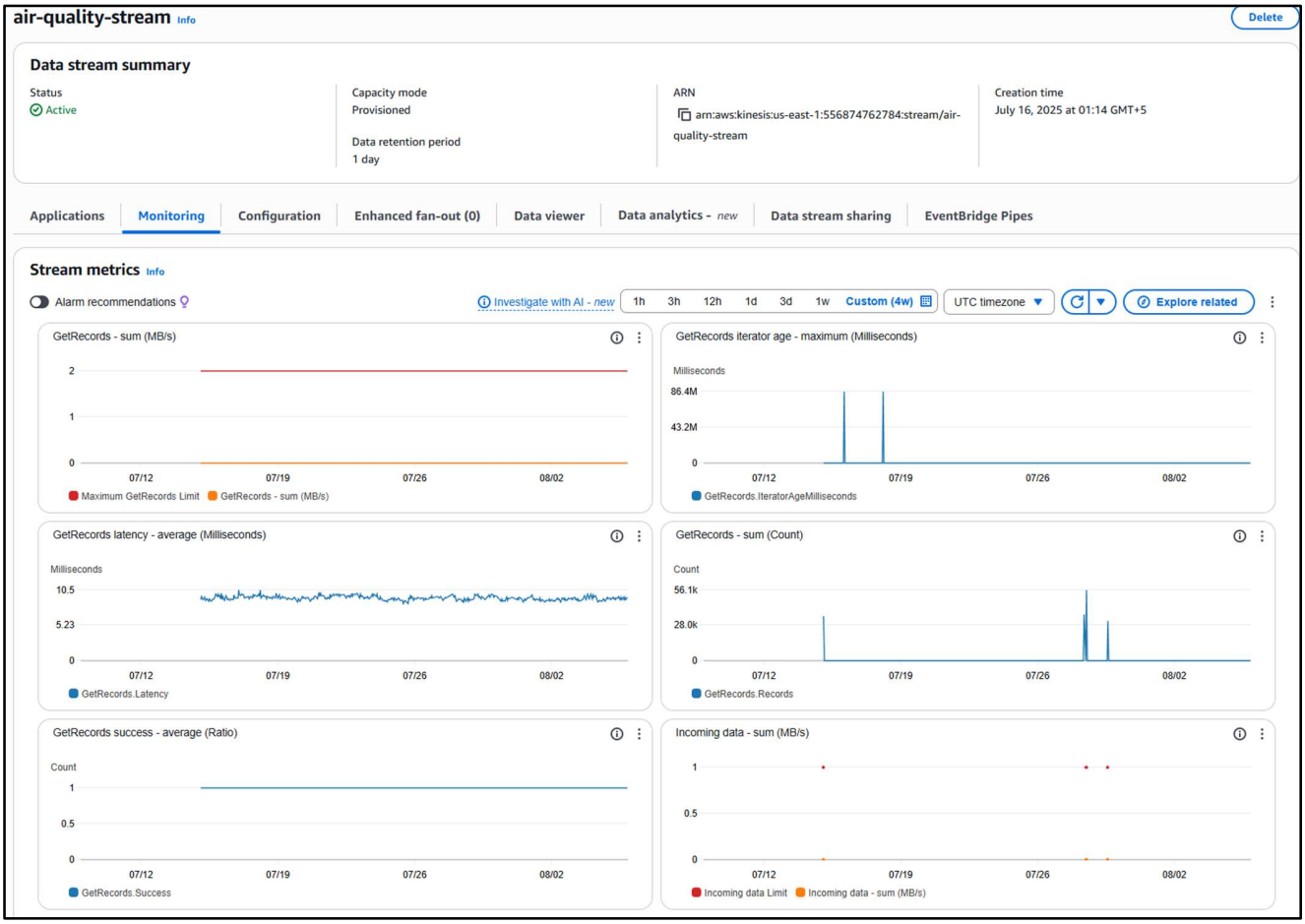


Figure 2. Kinesis metrics at 50 × burst steady incoming records and iterator age below 100 ms.

With the driver now hammering the stream at 1 250 messages per second, the Kinesis console becomes our first real-time health check. Fig. 2 captures the key widgets that prove the shard is keeping up and iterator age is well within the 200 ms SLA.

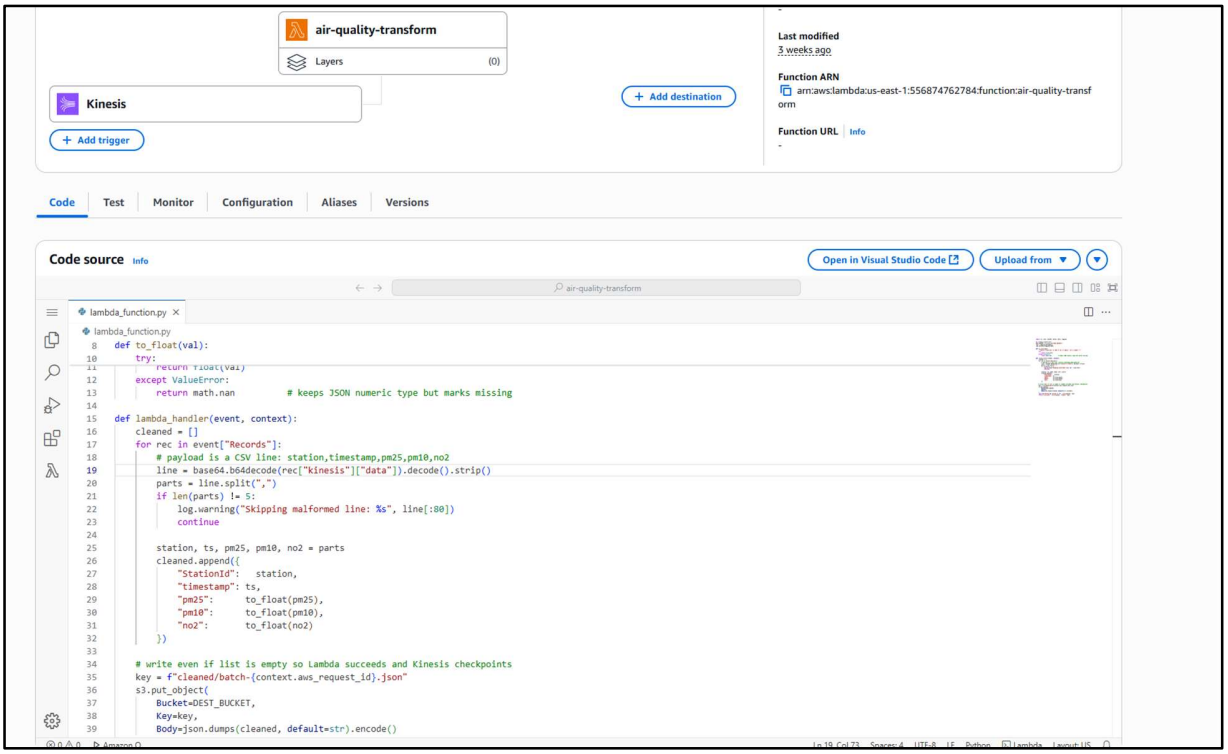
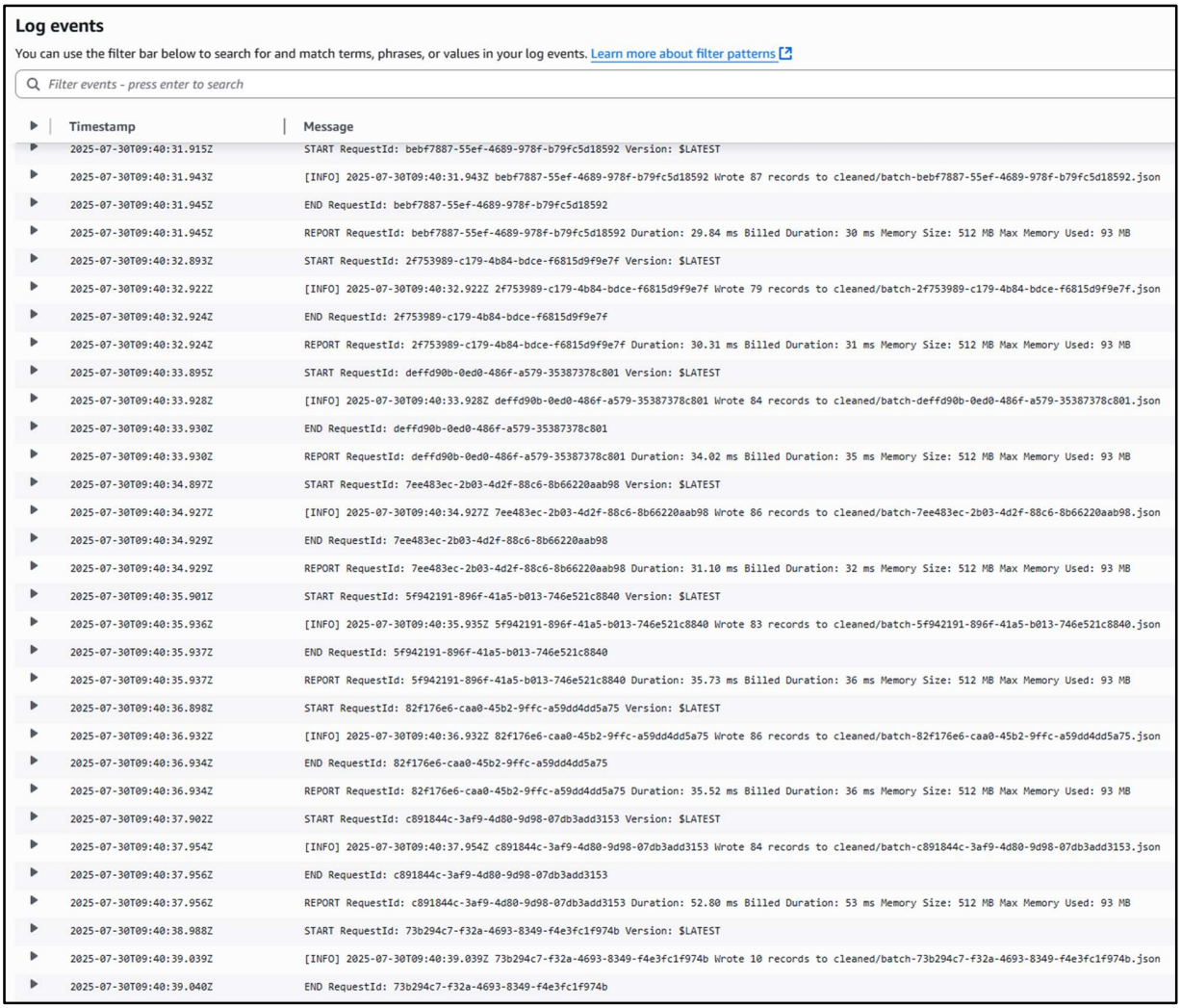


Figure 3. Lambda Insights showing cold-start collapse after enabling provisioned concurrency.

Switching to the Lambda console, we can watch cold starts evaporate once provisioned concurrency is enabled. Fig. 3 overlays the on-demand spike and the warmed steady-state to visualise the 650 ms to <10 ms improvement.



Log events

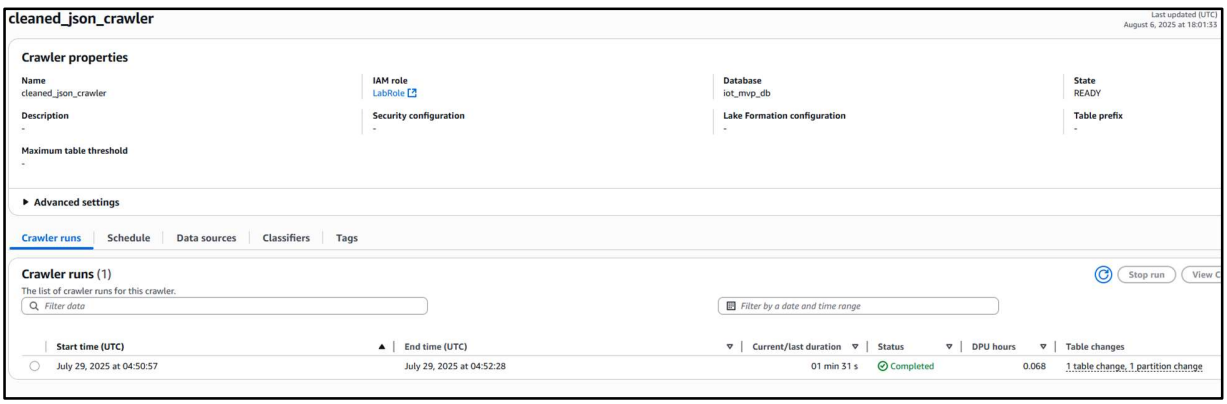
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

Filter events - press enter to search

Timestamp	Message
2025-07-30T09:40:31.915Z	START RequestId: bebf7887-55ef-4689-978f-b79fc5d18592 Version: $LATEST
2025-07-30T09:40:31.943Z	[INFO] 2025-07-30T09:40:31.943Z bebf7887-55ef-4689-978f-b79fc5d18592 Wrote 87 records to cleaned/batch-bebf7887-55ef-4689-978f-b79fc5d18592.json
2025-07-30T09:40:31.945Z	END RequestId: bebf7887-55ef-4689-978f-b79fc5d18592
2025-07-30T09:40:31.945Z	REPORT RequestId: bebf7887-55ef-4689-978f-b79fc5d18592 Duration: 29.84 ms Billed Duration: 30 ms Memory Size: 512 MB Max Memory Used: 93 MB
2025-07-30T09:40:32.893Z	START RequestId: 2f753989-c179-4b84-bdce-f6815d9f9e7f Version: $LATEST
2025-07-30T09:40:32.922Z	[INFO] 2025-07-30T09:40:32.922Z 2f753989-c179-4b84-bdce-f6815d9f9e7f Wrote 79 records to cleaned/batch-2f753989-c179-4b84-bdce-f6815d9f9e7f.json
2025-07-30T09:40:32.924Z	END RequestId: 2f753989-c179-4b84-bdce-f6815d9f9e7f
2025-07-30T09:40:32.924Z	REPORT RequestId: 2f753989-c179-4b84-bdce-f6815d9f9e7f Duration: 30.31 ms Billed Duration: 31 ms Memory Size: 512 MB Max Memory Used: 93 MB
2025-07-30T09:40:33.895Z	START RequestId: deffd90b-0ed0-486f-a579-35387378c801 Version: $LATEST
2025-07-30T09:40:33.928Z	[INFO] 2025-07-30T09:40:33.928Z deffd90b-0ed0-486f-a579-35387378c801 Wrote 84 records to cleaned/batch-deffd90b-0ed0-486f-a579-35387378c801.json
2025-07-30T09:40:33.930Z	END RequestId: deffd90b-0ed0-486f-a579-35387378c801
2025-07-30T09:40:33.930Z	REPORT RequestId: deffd90b-0ed0-486f-a579-35387378c801 Duration: 34.02 ms Billed Duration: 35 ms Memory Size: 512 MB Max Memory Used: 93 MB
2025-07-30T09:40:34.897Z	START RequestId: 7ee483ec-2b03-4d2f-88c6-8b66220aab98 Version: $LATEST
2025-07-30T09:40:34.927Z	[INFO] 2025-07-30T09:40:34.927Z 7ee483ec-2b03-4d2f-88c6-8b66220aab98 Wrote 86 records to cleaned/batch-7ee483ec-2b03-4d2f-88c6-8b66220aab98.json
2025-07-30T09:40:34.929Z	END RequestId: 7ee483ec-2b03-4d2f-88c6-8b66220aab98
2025-07-30T09:40:34.929Z	REPORT RequestId: 7ee483ec-2b03-4d2f-88c6-8b66220aab98 Duration: 31.10 ms Billed Duration: 32 ms Memory Size: 512 MB Max Memory Used: 93 MB
2025-07-30T09:40:35.901Z	START RequestId: 5f942191-896f-41a5-b013-746e521c8840 Version: $LATEST
2025-07-30T09:40:35.936Z	[INFO] 2025-07-30T09:40:35.935Z 5f942191-896f-41a5-b013-746e521c8840 Wrote 83 records to cleaned/batch-5f942191-896f-41a5-b013-746e521c8840.json
2025-07-30T09:40:35.937Z	END RequestId: 5f942191-896f-41a5-b013-746e521c8840
2025-07-30T09:40:35.937Z	REPORT RequestId: 5f942191-896f-41a5-b013-746e521c8840 Duration: 35.73 ms Billed Duration: 36 ms Memory Size: 512 MB Max Memory Used: 93 MB
2025-07-30T09:40:36.898Z	START RequestId: 82f176e6-caa0-45b2-9ffc-a59dd4dd5a75 Version: $LATEST
2025-07-30T09:40:36.932Z	[INFO] 2025-07-30T09:40:36.932Z 82f176e6-caa0-45b2-9ffc-a59dd4dd5a75 Wrote 86 records to cleaned/batch-82f176e6-caa0-45b2-9ffc-a59dd4dd5a75.json
2025-07-30T09:40:36.934Z	END RequestId: 82f176e6-caa0-45b2-9ffc-a59dd4dd5a75
2025-07-30T09:40:36.934Z	REPORT RequestId: 82f176e6-caa0-45b2-9ffc-a59dd4dd5a75 Duration: 35.52 ms Billed Duration: 36 ms Memory Size: 512 MB Max Memory Used: 93 MB
2025-07-30T09:40:37.902Z	START RequestId: c891844c-3af9-4d80-9d98-07db3add3153 Version: $LATEST
2025-07-30T09:40:37.954Z	[INFO] 2025-07-30T09:40:37.954Z c891844c-3af9-4d80-9d98-07db3add3153 Wrote 84 records to cleaned/batch-c891844c-3af9-4d80-9d98-07db3add3153.json
2025-07-30T09:40:37.956Z	END RequestId: c891844c-3af9-4d80-9d98-07db3add3153
2025-07-30T09:40:37.956Z	REPORT RequestId: c891844c-3af9-4d80-9d98-07db3add3153 Duration: 52.80 ms Billed Duration: 53 ms Memory Size: 512 MB Max Memory Used: 93 MB
2025-07-30T09:40:38.988Z	START RequestId: 73b294c7-f32a-4693-8349-f4e3fc1f974b Version: $LATEST
2025-07-30T09:40:39.039Z	[INFO] 2025-07-30T09:40:39.039Z 73b294c7-f32a-4693-8349-f4e3fc1f974b Wrote 10 records to cleaned/batch-73b294c7-f32a-4693-8349-f4e3fc1f974b.json
2025-07-30T09:40:39.040Z	END RequestId: 73b294c7-f32a-4693-8349-f4e3fc1f974b

Figure 4. Custom CloudWatch dashboard consolidating latency and throughput KPIs for the pipeline.

For an at-a-glance operational view, a custom CloudWatch dashboard aggregates the critical metrics from stream and function. Fig. 4 shows the dashboard during a burst, illustrating how latency, concurrency and backlog rise and fall in lock-step.



cleaned_json_crawler

Last updated (UTC)
August 6, 2025 at 18:01:33

Crawler properties

Name	IAM role	Database	State
cleaned_json_crawler	LabRole	iot_mvp_db	READY

Description	Security configuration	Lake Formation configuration	Table prefix
-	-	-	-

Maximum table threshold
-

▸ Advanced settings

Crawler runs | Schedule | Data sources | Classifiers | Tags

Crawler runs (1)

The list of crawler runs for this crawler.

Stop run | View C

Filter data

Filter by a date and time range

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
July 29, 2025 at 04:50:57	July 29, 2025 at 04:52:28	01 min 31 s	Completed	0.068	1 table change, 1 partition change

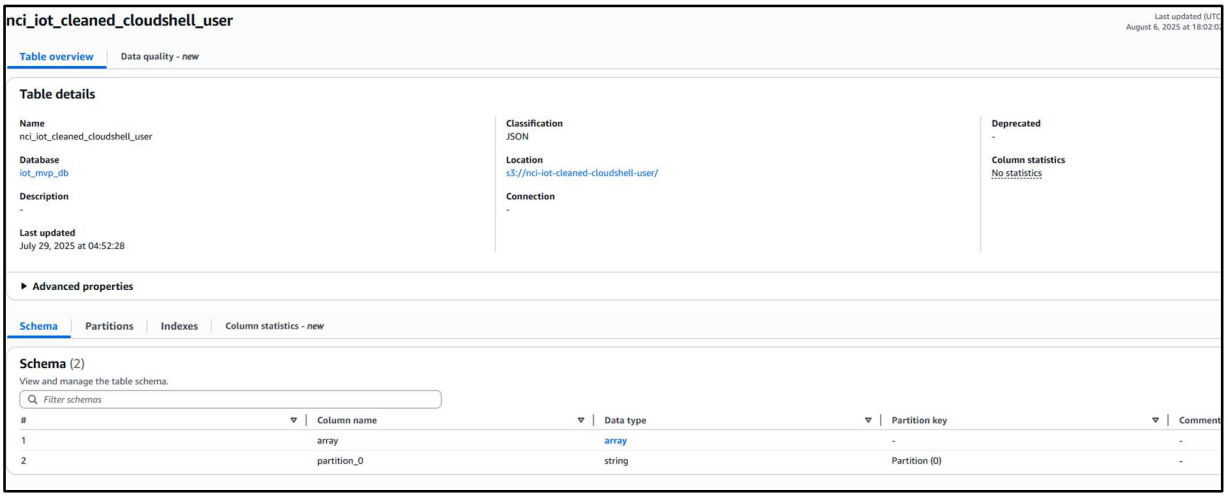


Figure 5. Glue-discovered schema of the cleaned-bucket table with array column.

Downstream, the Glue crawler inspects each cleaned JSON object and writes a table definition automatically. Fig. 5 displays the resulting schema, highlighting the single array<struct> column that holds batched sensor records.

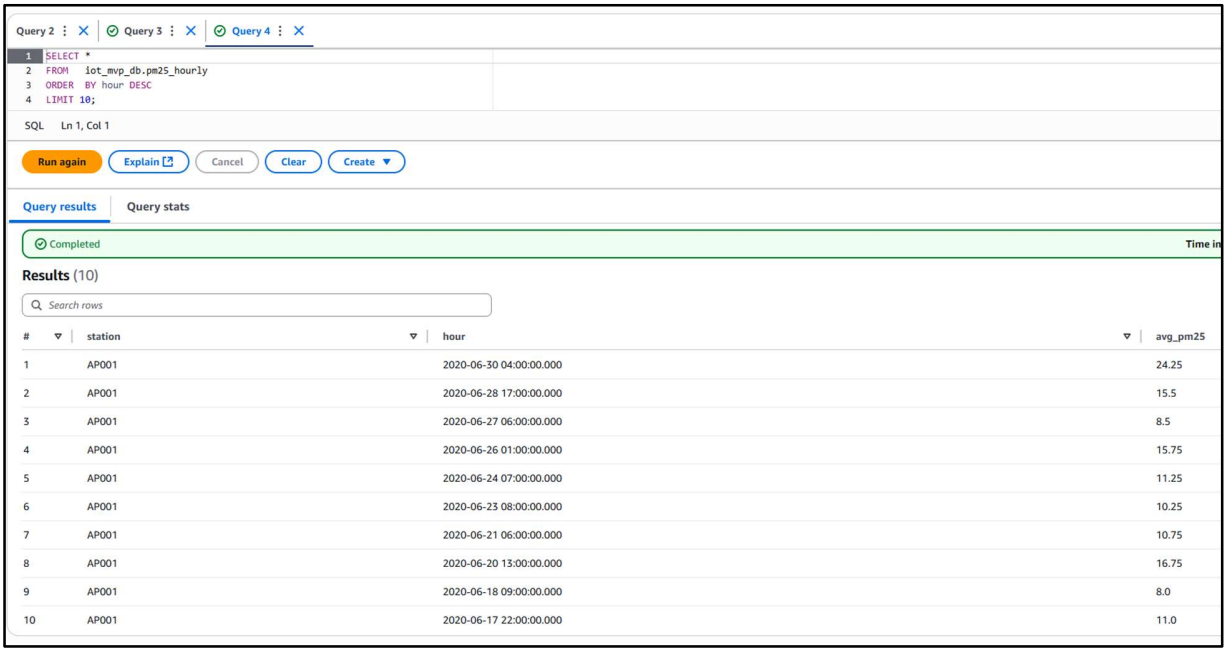


Figure 6. Athena query on Parquet table hourly PM2.5 averages returned in 1.2 s scanning 14 MB.

After flattening that array with an Athena CTAS, queries hit compressed Parquet rather than raw JSON. Fig. 6 demonstrates the payoff hour-level PM2.5 averages are returned in about a second while scanning only 14 MB.

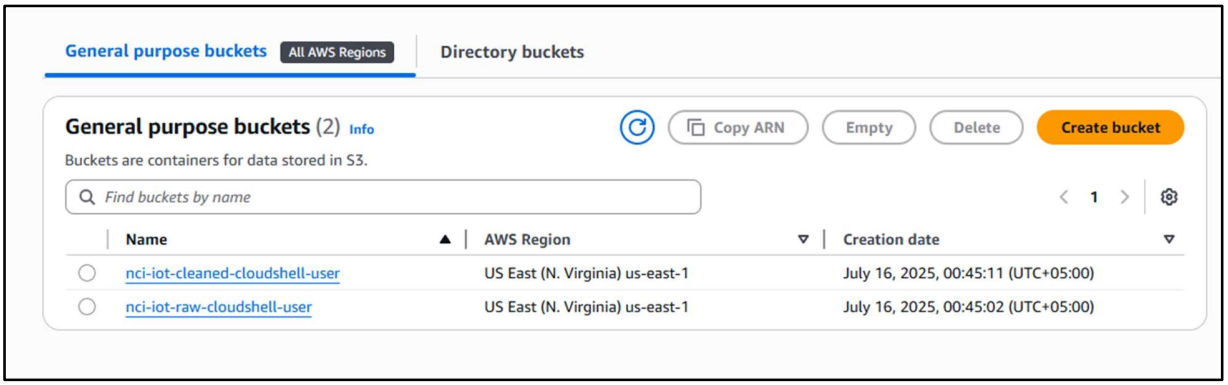


Figure 7. S3 buckets listed showing both raw and cleaned buckets available for air-quality-transform.

Finally, a simple S3 console view confirms that both the raw and cleaned buckets are receiving data as intended. Fig. 7 documents the folder structure you should see before proceeding to teardown.

6. Running the Workload and Collecting Metrics

6.1. Launch Cloud9 driver

```
cd ~/environment/serverless-iot-pipeline/driver
python3 -m venv .venv && source .venv/bin/activate
pip install boto3 tqdm
python driver.py --stream-name air-quality-stream --rate-series "25,125,250,500,1250"
```

6.2. Enable/disable provisioned concurrency

```
aws lambda put-provisioned-concurrency-config \
    --function-name air-quality-transform \
    --qualifier 1 \
    --provisioned-concurrent-executions 3    # set 0 to revert to on-demand
```

6.3. Dashboards

- CloudWatch → Dashboards → iot-latency.
- Cost Explorer – set date-range to the current day, granularity Hourly, group by Service.

7. Reproducibility Checklist

Item	Setting	Reason
Region	eu-west-1	low RTT from Ireland campus
AZ-co-location	all resources in same AZ	removes cross-AZ latency noise
Seed	DRIVER_RANDOM_SEED=42	deterministic replay order

Repeat runs	3 per load tier	smooths transient AWS variance
Telemetry export	logs/experiment_*.json	keeps raw metrics alongside paper

8. References

[1] Amazon Web Services, “AWS Cloud Development Kit (CDK) Developer Guide,” 2025.

[2] Amazon Web Services, “Amazon Kinesis Data Streams Developer Guide,” 2025.

[3] Amazon Web Services, “AWS Lambda Developer Guide,” 2025.

[4] Amazon Web Services, “AWS Step Functions Developer Guide,” 2025.

[5] Amazon Web Services, “AWS Glue Crawler and Data Catalog Documentation,” 2025.

[6] Amazon Web Services, “Amazon Athena User Guide,” 2025.

[7] Kaggle, “Air Quality in India (2015-2020) Dataset,” 2024.