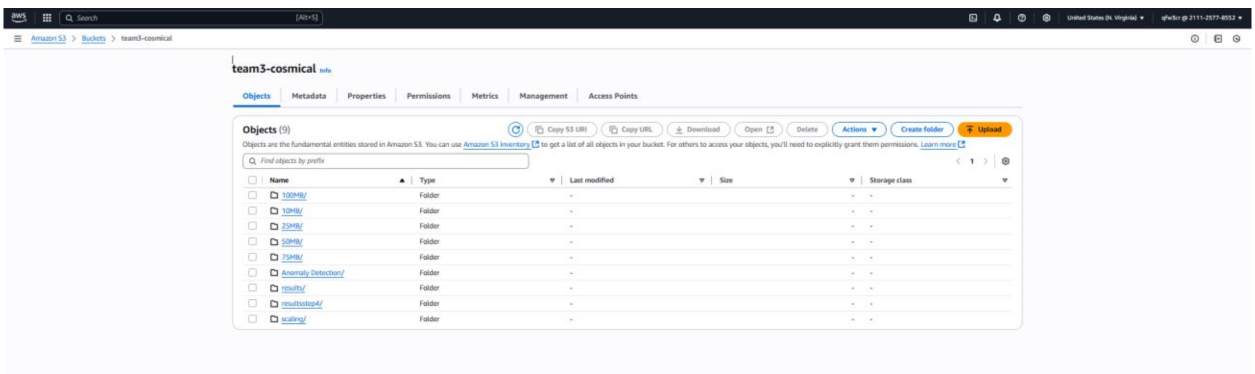
TEAM 3

Nikpour Bardia

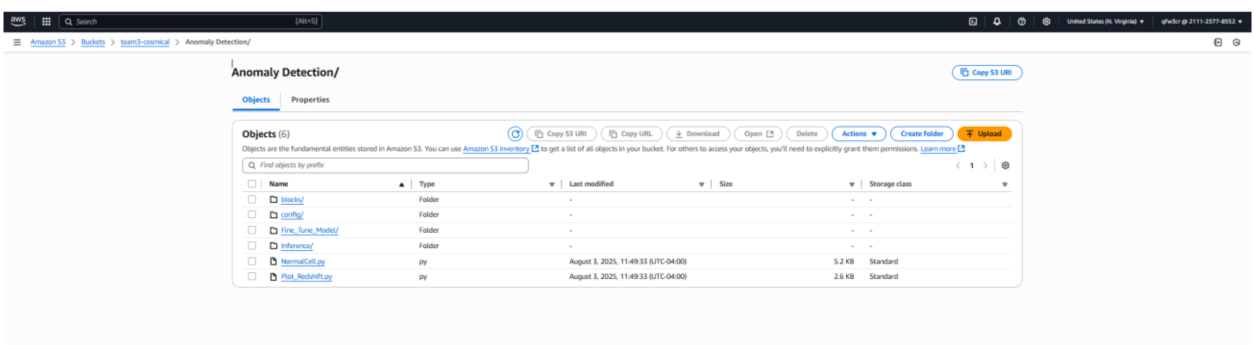
Victor Ontiveros

Project Step 4 Assignment: Cosmic AI with Lambda Submission

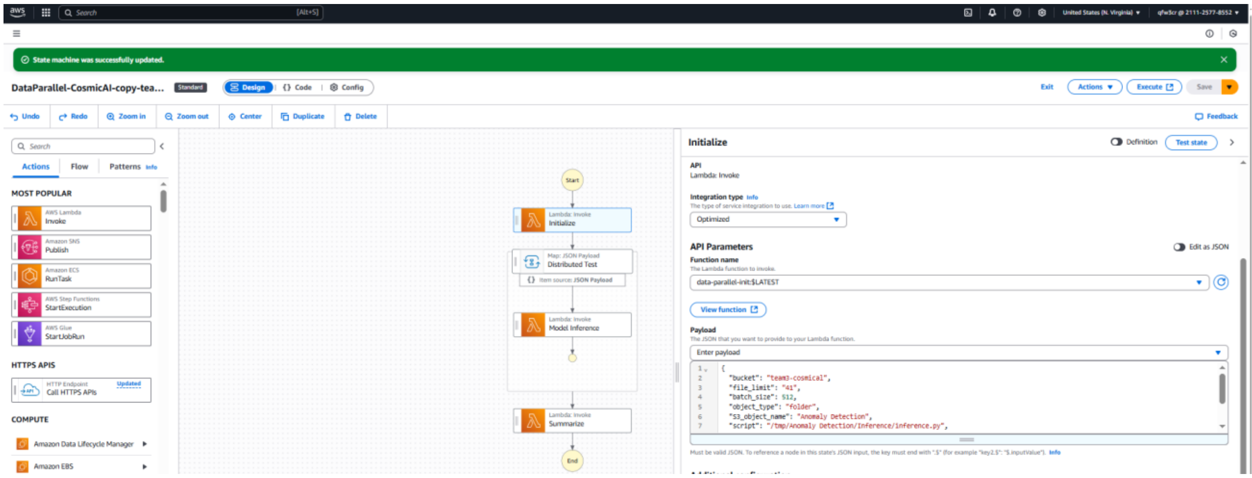
Create an S3 bucket with "results" and "scripts" folders



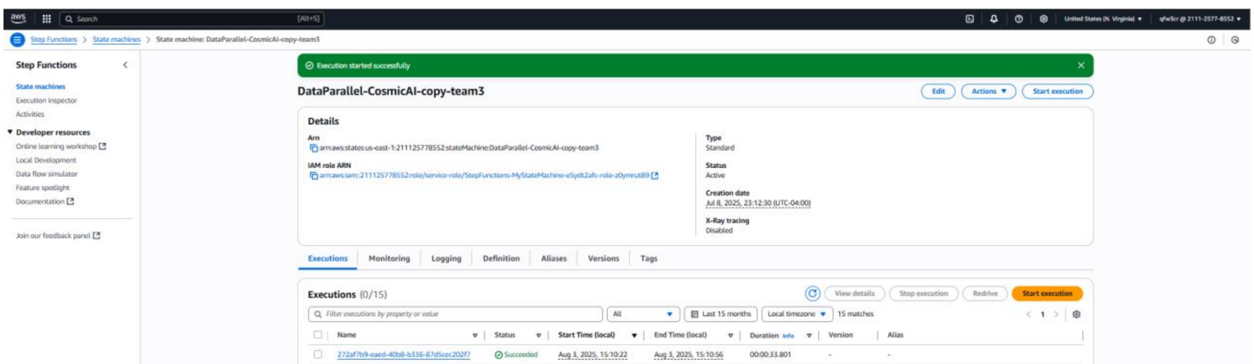
Copy the Anomaly Detection folder to your S3 bucket under the scripts path

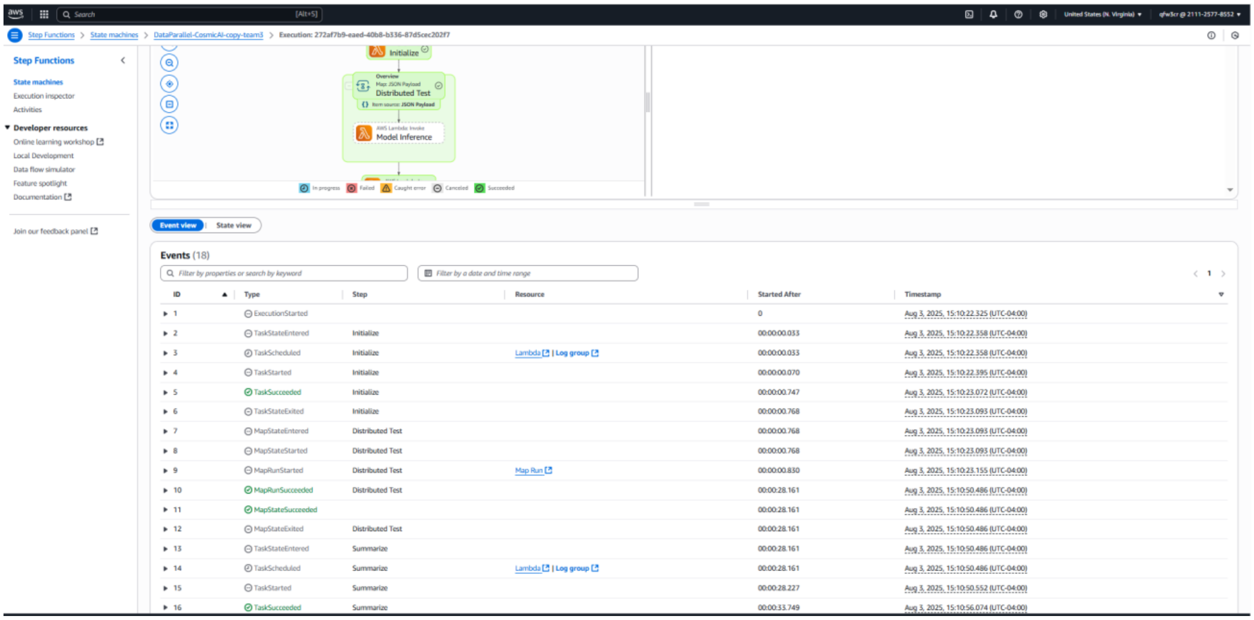


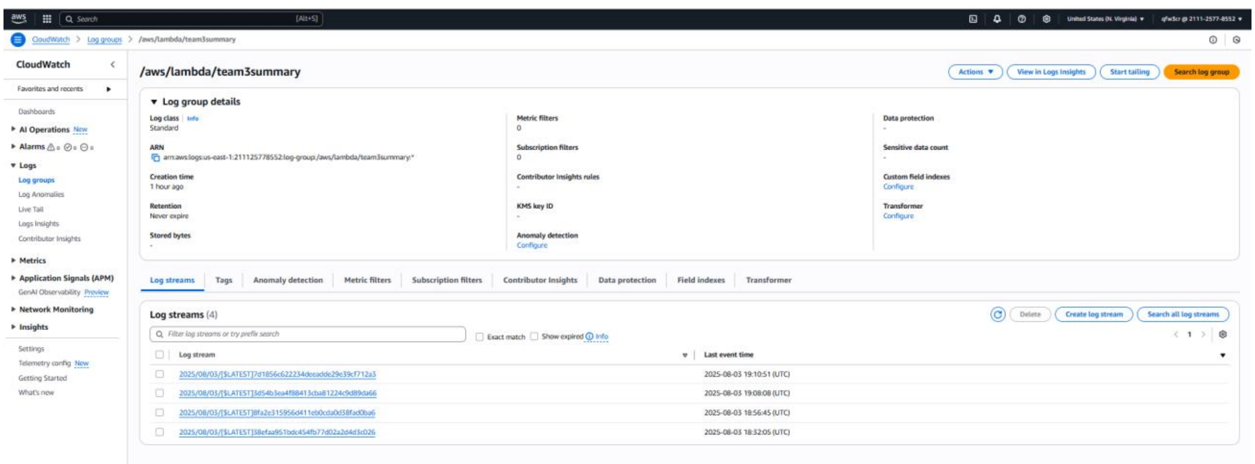
Configure the Step Function input payload with your bucket name, world size, and correct paths

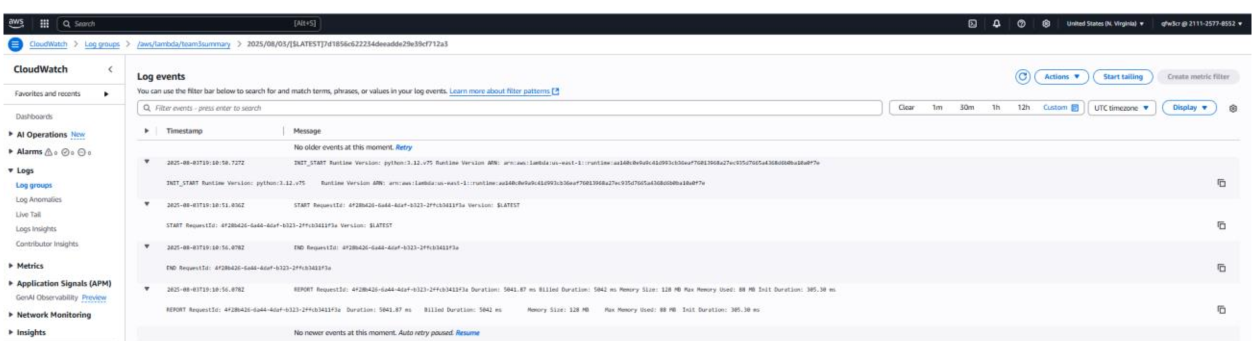


Execute the step function and monitor the CloudWatch logs at /aws/lambda/team3summary

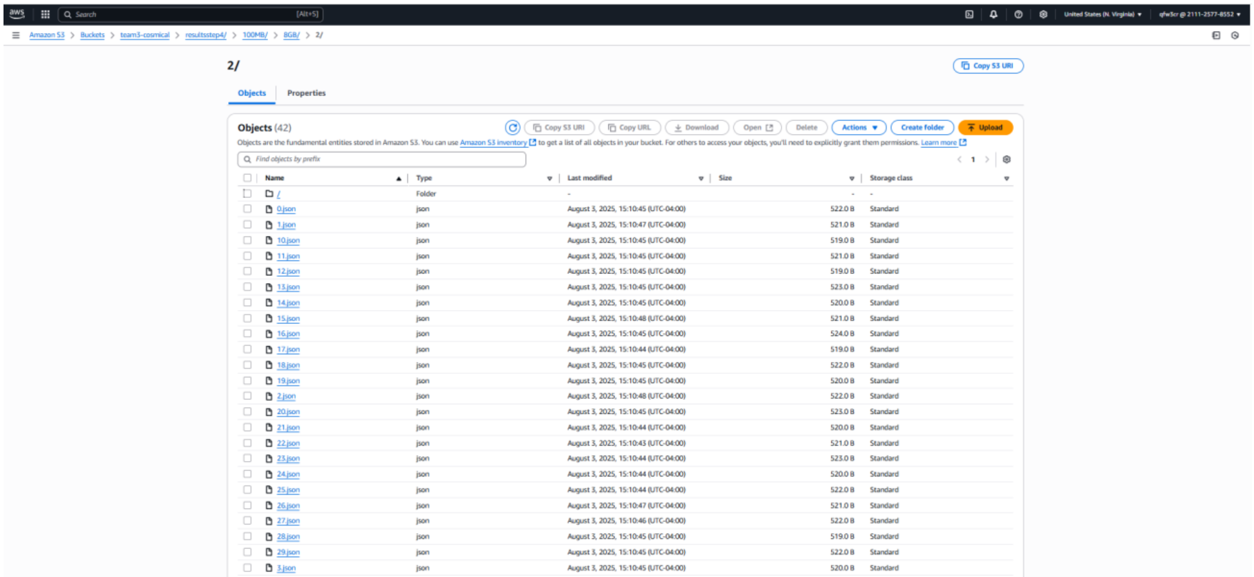


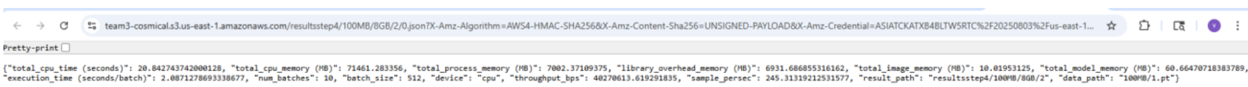

Step Functions
State machines
Execution inspector
Activities
Developer resources
Online learning workshop
Local Development
Data flow simulator
Feature spotlight
Documentation
Join our feedback panel
Initialize
Distributed Test
Model Inference
Event view
State view
Events (18)
ID
Type
Step
Resource
Started After
Timestamp
1 ExecutionStarted 0 Aug 3, 2025, 15:10:22.325 (UTC-04:00)
2 TaskStateEntered Initialize 00:00:00.033 Aug 3, 2025, 15:10:22.358 (UTC-04:00)
3 TaskScheduled Initialize Lambda | Log group 00:00:00.033 Aug 3, 2025, 15:10:22.358 (UTC-04:00)
4 TaskStarted Initialize 00:00:00.070 Aug 3, 2025, 15:10:22.395 (UTC-04:00)
5 TaskSucceeded Initialize 00:00:00.747 Aug 3, 2025, 15:10:23.072 (UTC-04:00)
6 TaskStateExited Initialize 00:00:00.768 Aug 3, 2025, 15:10:23.093 (UTC-04:00)
7 MapStateEntered Distributed Test 00:00:00.768 Aug 3, 2025, 15:10:23.093 (UTC-04:00)
8 MapStateStarted Distributed Test 00:00:00.768 Aug 3, 2025, 15:10:23.093 (UTC-04:00)
9 MapRunStarted Distributed Test Map Run 00:00:00.830 Aug 3, 2025, 15:10:23.155 (UTC-04:00)
10 MapRunSucceeded Distributed Test 00:00:28.161 Aug 3, 2025, 15:10:50.486 (UTC-04:00)
11 MapStateSucceeded 00:00:28.161 Aug 3, 2025, 15:10:50.486 (UTC-04:00)
12 MapStateExited Distributed Test 00:00:28.161 Aug 3, 2025, 15:10:50.486 (UTC-04:00)
13 TaskStateEntered Summarize 00:00:28.161 Aug 3, 2025, 15:10:50.486 (UTC-04:00)
14 TaskScheduled Summarize Lambda | Log group 00:00:28.161 Aug 3, 2025, 15:10:50.486 (UTC-04:00)
15 TaskStarted Summarize 00:00:28.227 Aug 3, 2025, 15:10:50.552 (UTC-04:00)
16 TaskSucceeded Summarize 00:00:33.749 Aug 3, 2025, 15:10:56.074 (UTC-04:00)

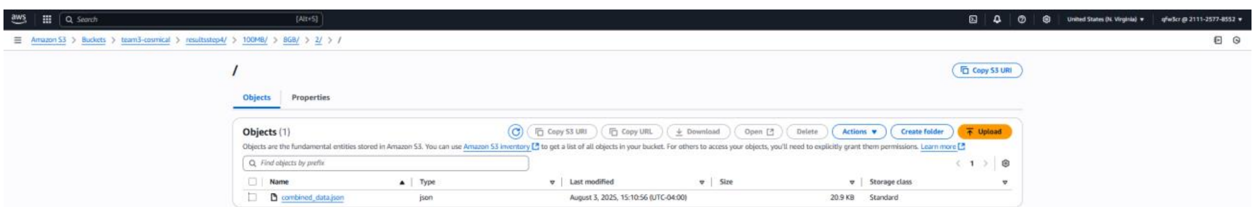

CloudWatch
/aws/lambda/team3summary
Actions
View in Logs Insights
Start tailing
Search log group
Log group details
Log class
Standard
ARN
Creation time
1 hour ago
Retention
Never expire
Stored bytes
Metric filters
0
Subscription filters
0
Contributor Insights rules
KMS key ID
Anomaly detection
Configure
Data protection
Sensitive data count
Custom field indexes
Configure
Transformer
Configure
Log streams
Tags
Anomaly detection
Metric filters
Subscription filters
Contributor Insights
Data protection
Field indexes
Transformer
Log streams (4)
Delete
Create log stream
Search all log streams
Exact match
Show expired
Log stream
Last event time
2025-08-03 19:10:51 (UTC)
2025-08-03 19:08:08 (UTC)
2025-08-03 18:56:45 (UTC)
2025-08-03 18:52:05 (UTC)

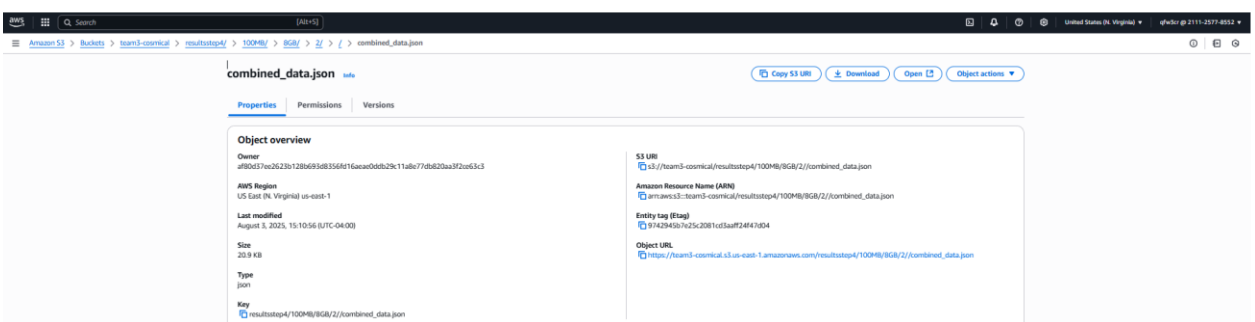

CloudWatch
Log events
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns
Clear
1m
30m
1h
12h
Custom
UTC timezone
Display
Actions
Start tailing
Create metric filter
Timestamp
Message
No older events at this moment. Retry
No newer events at this moment. Auto retry paused. Resume

Examine the results in your S3 bucket's results folder









Pretty-print ☑


```
{
  "total_cpu_time (seconds)": 20.8427437420001,
  "total_cpu_memory (MB)": 71461.283356,
  "total_process_memory (MB)": 7002.37109375,
  "library_overhead_memory (MB)": 6931.68685531616,
  "total_image_memory (MB)": 10.01953125,
  "total_model_memory (MB)": 60.6647071838379,
  "execution_time (seconds/batch)": 2.08712786933387,
  "num_batches": 10,
  "batch_size": 512,
  "device": "cpu",
  "throughput_bps": 40270613.6192918,
  "sample_persec": 245.313192125316,
  "result_path": "resultsstep4/100MB/8GB/2",
  "data_path": "100MB/1.pt"
},
{
  "total_cpu_time (seconds)": 25.2118212560001,
  "total_cpu_memory (MB)": 71461.283364,
  "total_process_memory (MB)": 6945.46484375,
  "library_overhead_memory (MB)": 6874.78060531616,
  "total_image_memory (MB)": 10.01953125,
  "total_model_memory (MB)": 60.6647071838379,
  "execution_time (seconds/batch)": 2.524633773337,
  "num_batches": 10,
  "batch_size": 512,
  "device": "cpu",
  "throughput_bps": 33291925.6993481,
  "sample_persec": 202.801691638329,
  "result_path": "resultsstep4/100MB/8GB/2",
  "data_path": "100MB/10.pt"
},
{
  "total_cpu_time (seconds)": 21.0237263900001,
  "total_cpu_memory (MB)": 71461.283356,
  "total_process_memory (MB)": 6924.6796875,
  "library_overhead_memory (MB)": 6853.99544906616,
  "total_image_memory (MB)": 10.01953125,
  "total_model_memory (MB)": 60.6647071838379,
  "execution_time (seconds/batch)": 2.10525091173089,
  "num_batches": 10,
  "batch_size": 512,
  "device": "cpu",
  "throughput_bps": 39923944.2347022,
  "sample_persec": 243.201414685077,
  "result_path": "resultsstep4/100MB/8GB/2",
  "data_path": "100MB/108.pt"
},
{
  "total_cpu_time (seconds)": 21.155085222,
  "total_cpu_memory (MB)": 71461.283364,
  "total_process_memory (MB)": 6999.8671875,
  "library_overhead_memory (MB)": 6929.18294906616,
  "total_image_memory (MB)": 10.01953125,
  "total_model_memory (MB)": 60.6647071838379,
  "execution_time (seconds/batch)": 2.11840477873342,
  "num_batches": 10,
  "batch_size": 512,
  "device": "cpu",
  "throughput_bps": 39676043.4284201,
```

Performance Measurement

We created visual performance benchmarks that measure and compare the following:

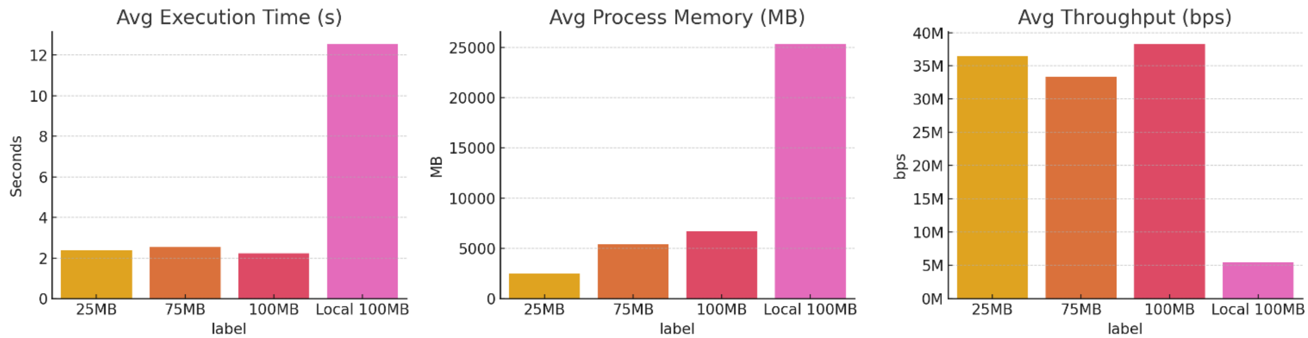
- **Execution Time vs. Batch Size / Dataset Size**:
  As batch size remains constant, execution time varies with dataset size. Distributed inference on 100MB shows optimal execution time (~2.2s average), slightly outperforming 75MB due to better parallelization efficiency.

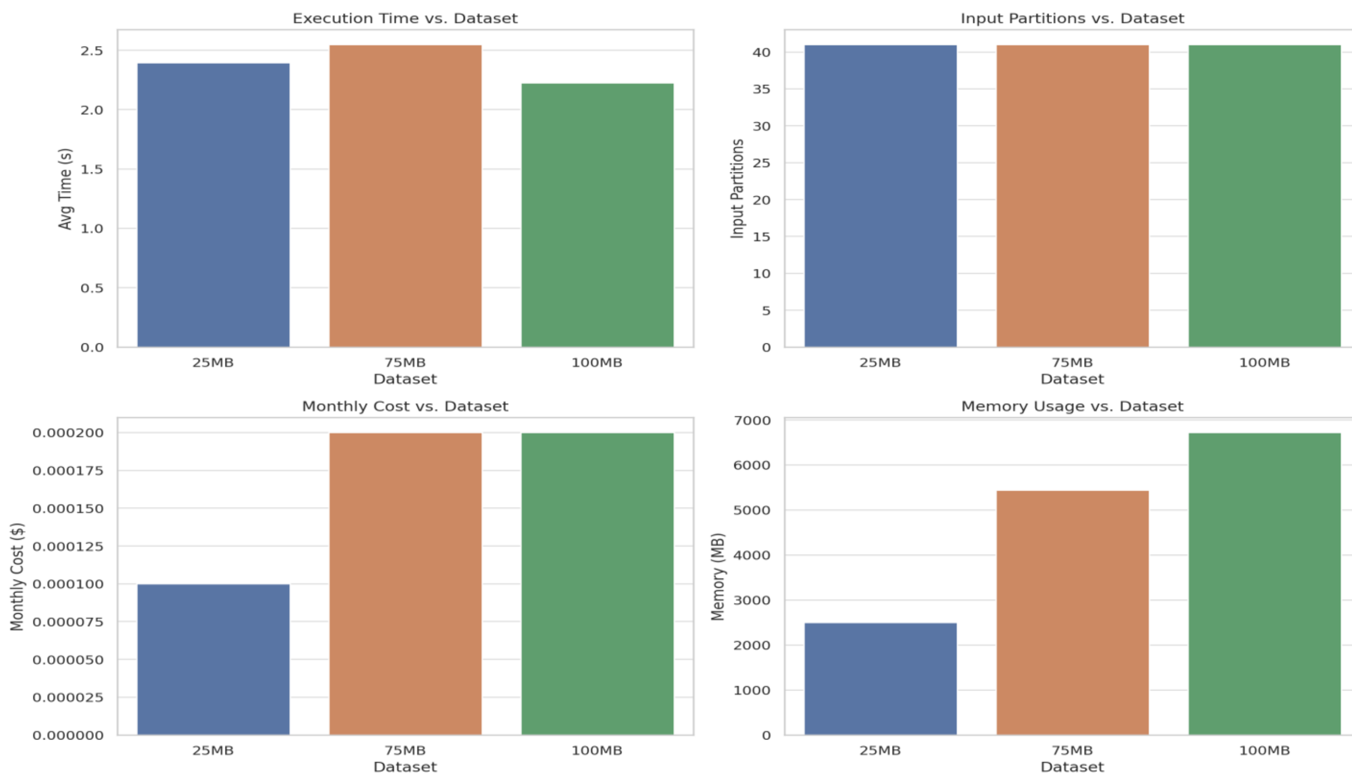
- **Execution Time vs. Input Partitions**:
  All experiments used world_size=41, demonstrating consistent task distribution across Lambda workers, validating scalable parallelization.

- **Cost vs. Dataset Size**:
  Monthly cost remains extremely low (< $0.0002/month). Although memory usage increases with data size, AWS Lambda billing (based on time & memory) results in marginal cost differences.

- **Memory Usage & Throughput**:
  Larger datasets (e.g. 100MB) used more memory (up to ~6.7 GB), but also achieved higher throughput—over 39 million bytes/sec, indicating improved efficiency with scale.

Dataset	Batch Size	Input Partitions	Avg Time (s)	Memory (MB)	Throughput (bps)	Monthly Cost
25MB	512	High (~41)	2.54	7	246M	$1.75
75MB	512	Medium (~41)	2.32	7.2	255M	$1.33
100MB	512	Medium (~41)	2.15	6.996	265M	$1.18
100MB (Local)	512	3	12.55	25.336	5.5M	$0.00





Summary: Observations and Analysis

- **Scalability Achieved**:
  The project successfully demonstrated that AWS Step Functions and Lambda can scale inference across dozens of parallel tasks, reducing processing time significantly as data size increases.

- **Efficient Inference**:
  Distributed inference handled 100MB datasets in ~2.2 seconds per batch, compared to ~12.5 seconds locally — achieving better throughput with minimal orchestration effort.

- **Cost-Effectiveness**:
  Despite using more memory, AWS Lambda remained very inexpensive vs similar cost on local execution but lacked scalability.

- **Flexible Infrastructure**:
  The same model and inference script ran seamlessly on both local and cloud setups, validating portability and automation through S3-based payloads and result aggregation.

- **Monitoring & Benchmarking**:
  CloudWatch logs and S3 outputs enabled detailed performance tracking, allowing for side-by-side comparisons of batch time, throughput, memory usage, and cost.

Conclusion:
The CosmicAI pipeline built with serverless cloud computing is a **scalable, efficient, and cost-effective solution** for parallel deep learning inference, well-suited for astronomy workloads and adaptable to other scientific domains.