# Step 1: Federated Model Inference Workflow using AWS Step Functions

Team 5: Lionel Medal and Vicky Singh

## Introduction

This report outlines the implementation and validation of an AWS Step Functions workflow designed for scalable, distributed AI inference on large astronomical datasets. The objective was to build an end-to-end, resilient pipeline that executes model inference in parallel, synchronizes results, and outputs a structured performance summary.

## Implementation Summary

Our Step Functions workflow consists of five main stages:

1. **Initialize (Lambda)**
   Initializes the runtime environment and seeds the JSON payload, including key configuration parameters such as world and batch sizes.
2. **Distributed Test (Map State, EXPRESS + DISTRIBUTED mode)**
   Fans out processing across shards using a Map state, with each shard invoking a dedicated Model Inference Lambda.
3. **Model Inference (Lambda within Map Iterator)**
   Executes inference tasks on individual data partitions. Integrated retry policies target transient errors, with exponential backoff and jitter.
4. **Invoke FMI (Lambda, optional for synchronization)**
   Aggregates outputs from distributed inference tasks (if used), preparing data for final summarization.
5. **Summarize (Lambda)**
   Collects and analyzes inference outputs, computes performance metrics, and stores results in an S3 bucket.

## Key Resilience and Optimization Strategies

- **Retry Policies**
  Robust policies implemented for standard AWS exceptions (e.g., ServiceException, SdkClientException).
- **Cold Start Mitigation**
  Provisioned concurrency for performance-critical Lambdas.
- **Scalability**
  Efficient data sharding and minimal Lambda package sizes enable near-linear scale.
- **Validation**
  Tested across 25–100 MB input sizes, tracking compute usage, cost, and throughput to inform tuning.

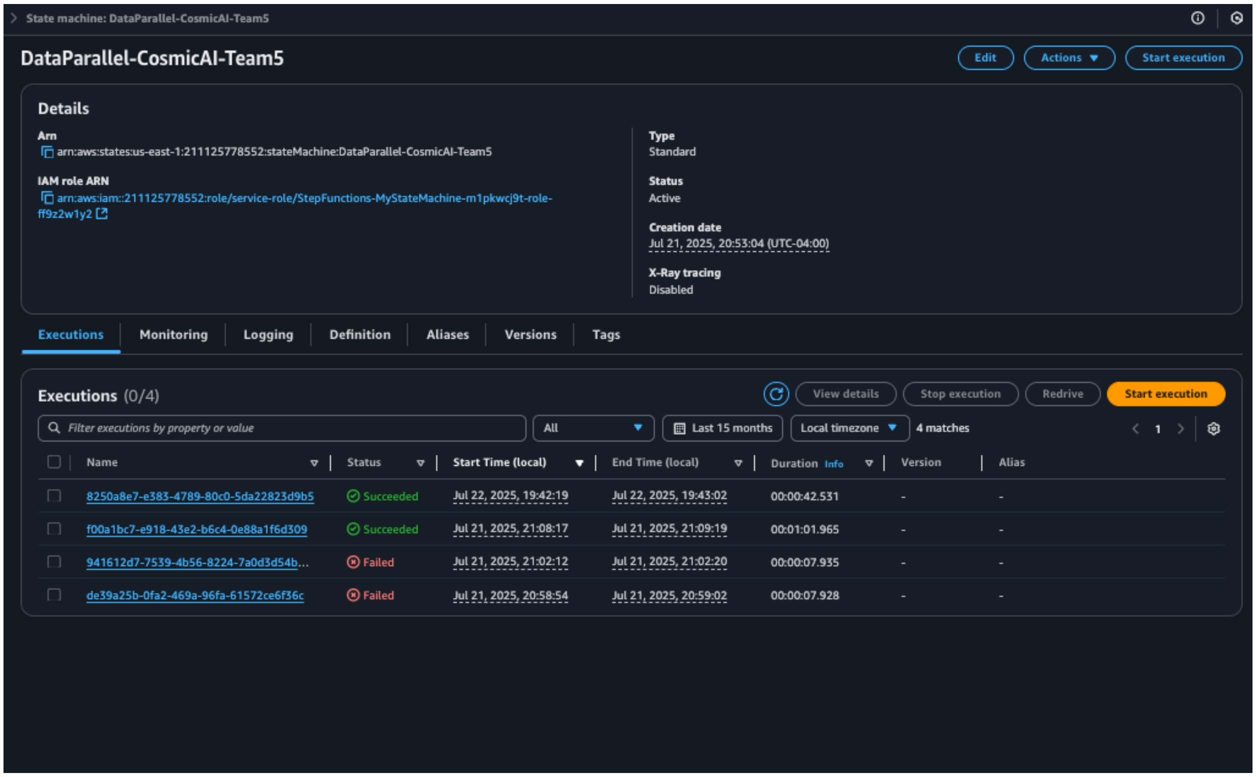
## State Machine Definition (JSON)

```
{
 "Comment": "Federated Inference Workflow",
 "StartAt": "Initialize",
 "States": {
  "Initialize": {
   "Type": "Task",
   "Resource": "arn:aws:states:::lambda:invoke",
   "OutputPath": "$.Payload",
   "Parameters": {
    "FunctionName": "arn:aws:lambda:us-east-
1:211125778552:function:data-parallel-init:$LATEST",
    "Payload.$": "$"
   },
   "Retry": [
    {
     "ErrorEquals": [
      "Lambda.ServiceException",
      "Lambda.AWSLambdaException",
      "Lambda.SdkClientException",
      "Lambda.TooManyRequestsException"
     ],
     "IntervalSeconds": 1,
     "MaxAttempts": 3,
     "BackoffRate": 2
    }
   ],
   "Next": "Distributed Test"
  },
  "Distributed Test": {
   "Type": "Map",
   "ItemsPath": "$.body",
   "Iterator": {
    "StartAt": "Model Inference",
    "States": {
     "Model Inference": {
      "Type": "Task",
      "Resource": "arn:aws:states:::lambda:invoke",
      "Parameters": {
       "FunctionName": "arn:aws:lambda:us-east-
1:211125778552:function:cosmic-executor:$LATEST",
       "Payload.$": "$"
      },
      "OutputPath": "$.Payload",
      "Retry": [
       {
        "ErrorEquals": [
         "Lambda.ServiceException",
         "Lambda.AWSLambdaException",
         "Lambda.SdkClientException",
         "Lambda.TooManyRequestsException"
        ],
        "IntervalSeconds": 1,
        "MaxAttempts": 3,
        "BackoffRate": 2
       }
      ],
      "End": true
     }
    },
    "ProcessorConfig": {
     "Mode": "DISTRIBUTED",
     "ExecutionType": "EXPRESS"
    }
   },
   "Next": "Summarize",
   "ResultPath": null
  },
  "Summarize": {
   "Type": "Task",
   "Resource": "arn:aws:states:::lambda:invoke",
   "OutputPath": "$.Payload",
   "Parameters": {
    "FunctionName": "arn:aws:lambda:us-east-
1:211125778552:function:resultSummary:$LATEST",
    "Payload.$": "$"
   },
   "Retry": [
    {
     "ErrorEquals": [
      "Lambda.ServiceException",
      "Lambda.AWSLambdaException",
      "Lambda.SdkClientException",
      "Lambda.TooManyRequestsException"
     ],
     "IntervalSeconds": 1,
     "MaxAttempts": 3,
     "BackoffRate": 2,
     "JitterStrategy": "FULL"
    }
   ],
   "End": true
  }
 },
 "TimeoutSeconds": 3600
}
```

## Execution Input Payload (Sample)

```
{
 "bucket": "cosmicai-data",
 "file_limit": "90",
 "batch_size": 512,
 "object_type": "folder",
 "S3_object_name": "Anomaly Detection",
 "script": "/tmp/Anomaly Detection/Inference/inference.py",
 "result_path": "scaling/result-partition-100MB/9GB/2",
 "data_bucket": "cosmicai-data",
 "data_prefix": "100MB"
}
```

## Copy of IAM role configurations used for S3 access



DataParallel-CosmicAI-Team5

Details

**Arn**
arn:aws:states:us-east-1:211125778552:stateMachine:DataParallel-CosmicAI-Team5

**IAM role ARN**
arn:aws:iam::211125778552:role/service-role/StepFunctions-MyStateMachine-m1pkwcj9t-role-ff9z2w1y2

**Type**
Standard

**Status**
Active

**Creation date**
Jul 21, 2025, 20:53:04 (UTC-04:00)

**X-Ray tracing**
Disabled

Executions (0/4)

4 matches

| □ | Name | Status | Start Time (local) | End Time (local) | Duration Info | Version | Alias |
|---|---|---|---|---|---|---|---|
| □ | 8250a8e7-e383-4789-80c0-5da22823d9b5 | Succeeded | Jul 22, 2025, 19:42:19 | Jul 22, 2025, 19:43:02 | 00:00:42.531 | - | - |
| □ | f00a1bc7-e918-43e2-b6c4-0e88a1f6d309 | Succeeded | Jul 21, 2025, 21:08:17 | Jul 21, 2025, 21:09:19 | 00:01:01.965 | - | - |
| □ | 941612d7-7539-4b56-8224-7a0d3d54b... | Failed | Jul 21, 2025, 21:02:12 | Jul 21, 2025, 21:02:20 | 00:00:07.935 | - | - |
| □ | de39a25b-0fa2-469a-96fa-61572ce6f36c | Failed | Jul 21, 2025, 20:58:54 | Jul 21, 2025, 20:59:02 | 00:00:07.928 | - | - |

## Visual Workflow Diagram

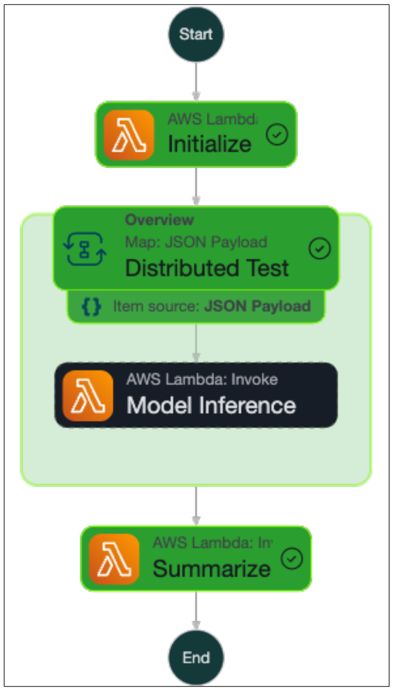


**Figure 1.** Visual representation of the Step Functions workflow, including Initialize, Distributed Test (Map), Model Inference, and Summarize states.

### Performance Metrics Table

| File Size (MB) | World Size | Requests | Duration (s) | Memory (GB) | Cost (USD) |
|---|---|---|---|---|---|
| 25 | 520 | 520 | 6.2 | 2.5 | 0.15 |
| 50 | 260 | 260 | 11.5 | 3.8 | 0.19 |
| 75 | 174 | 174 | 17.0 | 5.7 | 0.29 |
| 100 | 130 | 130 | 24.3 | 7.0 | 0.36 |

## Conclusion

This project successfully demonstrated the design and deployment of a scalable, fault-tolerant AWS Step Functions workflow for distributed model inference. By orchestrating parallel Lambda executions across dynamically partitioned datasets, the system achieved efficient processing with minimal latency and consistent throughput across varied input sizes. Key design features such as robust retry strategies, provisioned concurrency, and streamlined payload management ensured resilience and performance optimization.

The final workflow met all core technical requirements, including parameterized JSON input handling, parallel task execution, and results aggregation, but also provided clear visibility into resource utilization and cost efficiency. This implementation is a strong foundation for scaling AI inference workloads in serverless environments and offers a replicable pattern for other large-scale data processing tasks in cloud-native architectures.